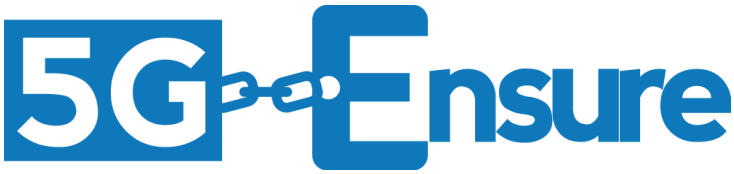


Deliverable D3.6
5G-PPP security enablers open specifications (v2.0)

Project name	5G Enablers for Network and System Security and Resilience	
Short name	5G-ENSURE	
Grant agreement	671562	
Call	H2020-ICT-2014-2	
Delivery date	1.5.2017	
Dissemination Level:	Public	
Lead beneficiary	SICS[1]	Thomas Carnehult, thomas.carnehult@sics.se
Authors	VTT: Kimmo Ahola, Olli Mämmelä, Pekka Ruuska, Jani Suomalainen TS: Théo Combe, Pascal Bisson, Edith Félix ALBLF[2]: Linas Maknavicius EAB: Håkan Englund, Prajwol Kumar Nakarmi IT-Innov: Maxim Bashevoy, Gianluca Correndo, Stefanie Cox, Stephen C. Phillips, Toby Wilkinson LMF: Patrik Salmela NEC: Felix Klaedtke, Alessandro Sforzin Nixu: Tommi Pernila Orange: Ghada Arfaoui, Jean-Philippe Wary, Jose Manuel Sanchez Vilchez SICS[1]: Thomas Carnehult, Nicolae Paladi, Ludwig Seitz, Linus Karlsson, Martin Gunnarsson, Markus Ahlström and Simon Holmberg TASE: David Perez, Willie Betancourt TCS: Sébastien Keller, Frédéric Motte, Filippo Rebecchi TIIT: Baltatu Madalina, Luciana Costa, Dario Lombardo UOXF: Piers O'Hanlon, Ravishankar Borgaonkar	

1 RISE SICS since Jan 1, 2017
2 Nokia Bell Labs since Jan 14, 2016

Executive summary

This document describes the open specifications of 5G Security enablers planned to compose the second and final software release (i.e. v2.0) of 5G-ENSURE Project due in August 2017 (M22). The enablers' open specifications are presented per security areas in scope of the project, namely: Authentication, Authorization and Accounting (AAA), Privacy, Trust, Security Monitoring, and Network management & Virtualisation isolation. For each of these categories the open specifications of all enablers planned in the project's Technical Roadmap for v2.0 and having features for v2.0 are detailed following the same template. Overall, this deliverable leverages the previous deliverable (i.e. D3.2) in that it extends and completes the open specification of security enablers in scope of the second release. It also paves the way towards the development and demonstration of the second and final set of 5G-ENSURE security enablers as planned for v2.0 in the project's Technical Roadmap(Update) (i.e. D3.5).

D3.6 provides valuable input to work both on the 5G Security architecture and 5G Security testbed, since it provides details to link security enablers to the 5G security architecture so far defined (D2.4) and under consolidation (in the context of D2.7 to come), and also, to plan security enablers integration, testing, demonstration, and assessment on the 5G security testbed.

Foreword

5G-ENSURE belongs to the first group of EU-funded projects which collaboratively develop 5G under the umbrella of the 5G Infrastructure Public Private Partnership (5G-PPP) in the Horizon 2020 Programme. The overall goal of 5G-ENSURE is to deliver strategic impact across technology and business enablement, standardisation and vision for a secure, resilient and viable 5G network. The project covers research & innovation - from technical solutions (5G security architecture and testbed with 5G security enablers) to market validation and stakeholders' engagement - spanning various application domains.

Deliverable D3.6 follows deliverable D3.5 5G-PPP security enablers Technical Roadmap (Updated) that has delivered a description of 5G security enablers requested to achieve 5G security vision as illustrated in deliverable D2.1 on Use Cases.

Whereas the objective of D3.5 was mainly focusing on highlighting requested 5G Security enablers (in terms of their high-level product vision) and their scheduling between release one and release two, D3.6 clearly focuses on providing the open specifications of enablers planned to be software released (through their planned features) for the second and full release (i.e. v2.0 due at M22).

Whereas D3.2 was providing the open specifications of enablers with features planned in Release 1 as per early version of Technical Roadmap (i.e. D3.1), D3.6 details the open specifications of each of the enablers in scope of the second release (v2.0) with their planned features as per latest update of the Technical Roadmap (i.e. D3.5).

The 5G-ENSURE security enablers' open specifications are public and royalty free (see Open specification legal notice in Annex A) in order to enable anyone interested to come up with their own "compliant" implementation.

D3.6 will be followed by deliverable D3.7 "5G-PPP Security enablers software release (v2.0)" and deliverable D3.8 "5G-PPP Security enablers documentation (v2.0)" due both at M22, that are respectively devoted to a software release of the 5G Security enablers v2.0 with their planned features (i.e. reference implementations following open specifications) and their accompanying documentation (e.g. installation and administration guide, user & programmer guide, unit testing plan).

This deliverable is also an important input to other technical Work packages of the project to feed into their work. Especially WP2 for what concerns the on-going work on 5G security architecture, since enablers specified in D3.6 would form some of the main building blocks of the final architecture D2.7, but also WP4, as it is responsible for the further development the 5G security testbed where these enablers will be hosted and assessed.

Disclaimer

Copyright notice

Table of Contents

Abbreviations

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
AKA	Authentication and Key Agreement
AMF	Authentication Management Field
API	Application Programming Interface
ASME	Access Stratum Mobile Equipment
AUTN	Authentication Token
AV	Authentication Vector
AVP	Attribute-Value Pair
B-TID	Bootstrapping Transaction Identifier
BSF	Bootstrapping Server Function
BYOI	Bring-your-own-identity
CEP	Complex Event Processing
CK	Ciphering Key
DH	Diffie-Hellman
EPS	Evolved Packet System
GBA	Generic Bootstrapping Architecture
GUSS	GBA User Security Settings
HE	Home Environment
HLR	Home Location Register
HMAC	Hashed Message Authentication Code
HSS	Home Subscriber Server
IDS	Intrusion Detection System
IK	Integrity Key
IoT	Internet-of-Things
IPS	Intrusion Protection System

JSON	JavaScript Object Notation
KDF	Key Derivation Function
LTE	Long Term Evolution
MAC	Message Authentication Code
ME	Mobile Equipment
MITM	Man-In-The-Middle
MME	Mobility Management Entity
MTC	Machine-Type Communication
NAF	Network Application Function
PFS	Perfect Forward Secrecy
PLMN	Public Land Mobile Network
RAND	Random bit string
RES	Response
SDN	Software Defined Network
SGSN	Serving GPRS Support Node
SIEM	Security Information and Event Management
SN	Serving Network
UE	User Equipment
eUICC	Embedded Universal Integrated Circuit Card
UICC	Universal Integrated Circuit Card
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
vGBA	Vertical GBA
VLR	Visitor Location Register
XRES	Expected User Response

1 Introduction

This document describes the open specifications of 5G Security enablers planned to compose the second and final software release (i.e. v2.0) of 5G-ENSURE Project due in August 2017 (M22). As such this document mainly focuses on enablers identified in the project's updated Technical Roadmap (D3.5) and having features for v2.0. The enablers' open specifications are presented per security areas in scope of the project, namely: Authentication, Authorization and Accounting (AAA), Privacy, Trust, Security Monitoring, and Network management & virtualisation isolation. For each of these categories the open specifications of the enablers planned in the updated Technical Roadmap for v2.0 and having features for v2.0 are detailed following same template.

The table below summarizes the 5G-ENSURE technical roadmap for R2 (or v2.0) as described in D3.5. It shows the 5G security enablers in scope providing their (code) name, the category to which they belong to (i.e. AAA, Privacy, Trust, Security Monitoring, Network management & virtualization isolation), as well as the planned features.

Table 1-1: 5G-ENSURE Technical Roadmap

Category	5G-ENSURE security enablers	Features [3]
AAA	Basic AAA enabler	Forward Secrecy (R1/R2)
		AAA aspects of trusted micro-segmentation (R1 /R2)
		Trusted interconnect and authorization (R2)
	Internet of things (IoT)	Group based AKA (R1/R2)
		Non-USIM based AKA (R2)
		BYOI (Bring Your Own Identity) (R2)
		vGBA (Vertical GBA) (R2)
	Fine-grained Authorization	Basic Authorization in Satellite systems (R1)
		Basic distributed authorization Enforcement for RCDs (R1)
		AAA integration with satellite systems (R2)
		Authorization and authentication for RCD based on ongoing IETF standardization (R2)
	Federative authentication and identification enabler	Storage of authentication level (R2)
		Usage of authentication level (R2)
Privacy	Privacy Enhanced Identity Protection	Encryption of Long Term Identifiers **(**IMSI public-key based encryption) (R1)
		Home Network centric IMSI protection (R2)
		IMSI Pseudonymization (R2)
	Device identifier(s) privacy	Enhanced privacy for network attachment protocols (R1)
		Anonymous and optimised address selection for network attachment protocols (R2)

[3] R1 indicates first specified in D3.2 and part of release 1 and R2 indicates specified in D3.6 and part of release 2. R1/R2 indicates specified in R1 and updated in R2.

Category	5G-ENSURE security enablers	Features [3]
	Device-based Anonymization	Format preserving anonymization algorithm (R2)
		Privacy configuration (R2)
	Privacy policy analysis	Privacy policy specification (R2
		Privacy preferences specification (R2)
		Comparison of policies and preferences (R2)
Trust	Trust Builder	5G Asset model (R1/R2)
		Graphical editor (R1/R2)
		5G Threat knowledgebase (R2)
	Trust Metric Enabler	Trust metric based network domain security policy management (R1)
		Improved trust metric based on extended data (R2)
	VNF Certification	VNF Trustworthiness Evaluation (R1)
		VNF Trustworthiness Certification (R2)
	Security Indicator	Security indicator subscriber display (R2)
	Reputation based on Root Cause Analysis for SDN	Root Cause Analysis for SDN (R2)
Security Monitoring	System Security State Repository	Deployment model ontology (also known as 5G asset model) (R1)
		System Security State Repository service (R2)
	Security Monitor for 5G Micro-Segments	Complex Event Processing Framework for Security Monitoring and Inferencing (R1)
		Risk-based adaptation of micro-segments (R2)
		Extended data gathering (R2)
		Cross-domain information exchange (R2)
	Satellite Network Monitoring	Pseudo real-time monitoring (R1)
		Threat detection (R1)
		Active security analysis (R2)
		Pre-emptive mitigation security actions (R2)
	Generic Collector Interface	Log and Event Processing (R1)
	PulSAR: Proactive Security Analysis and Remediation	5G specific vulnerability schema (R1)
		5G specific vulnerability schema implementation (R2)
		PulSAR interface with Generic Collector (R2)
Network Management and Virtualization Isolation	Anti-Fingerprinting	Controller-Switch-Interaction Imitator (R1)
	Access Control Mechanisms	Southbound Reference Monitor (R1)
		Access Requirements for VNF Container Resources (R2)
	Component-Interaction Audits	Basic OpenFlow Compliance Checker (R1)

Category	5G-ENSURE security enablers	Features [3]
		Basic NFV Reconfiguration Compliance Checker (R2)
	Bootstrapping Trust	Integrity Attestation of Virtual Network Components (R1)
		Integrity Attestation of VNFs running in Docker containers (R2)
	Micro Segmentation	Dynamic Arrangement of Micro-Segments (R1)
		Extended Northbound API (R2)
		Support for multi-domain micro-segments (R2)
	Flow Control: in-network Threat Detection and Mitigation for Critical Functions in Virtual Networks	Detection of malicious behaviours in virtual networks (R1)
		Mitigation of detected network threats (R2)

Each of the enablers' Open Specification encompasses a range of information including, amongst others, the description of the enabler (e.g. scope, behaviour and intended use), definition of terms to assist the understanding of the specification, legal notice that applies, architectural sketch depicting components and their main interactions, preliminary description of the API that the enabler will support when it applies, to the description of non-functional features. Taking advantage of the fact that the 5G Security Architecture (draft) has been defined (see D2.4), each of the enablers are also linked to key concepts it encompasses.

1.1 5G Security Architecture

The security architecture is an assembly and categorization of techniques, which collectively protect security and privacy of a 5G network and its domains with different trust relationships. The 5G security architecture [D2.4 and D2.7] adopts terminology, concepts and objectives from the current 3G/4G security architectures from 3GPP [TS33.102, TS33.401] and then extends to support new requirements and characteristics arising from 5G technologies, business and use cases, trust models, as well as regulatory aspects.

1.1.1 Domains, strata and security feature groups

In the security architecture, a **domain** is a grouping of network entities according to physical or logical aspects that are relevant for 5G networks. Figure 1-1 illustrates domains in the 5G architecture. Domains are classified horizontally to (physical) infrastructure domain and (logical) virtual domains as well as vertically to user equipment and network, and IP (internet protocol) service domains.

Infrastructure domain includes physical hardware with different trust anchors. Universal Integrated Circuit Card (UICC) domain that provides conventional tamper-resistant module offering protected storage and processing of long-term subscriber credentials. Mobile equipment hardware (MEHW) domain provides device specific security support, e.g. a trusted execution environment, for other device or user-specific credential management. The Infrastructure Provider Domain (IP Domain) contains the hardware platforms for the compute, storage, and networking resources. The trust anchors in IP domain can be based e.g. on integrity protection and verification that has been enabled by trusted platform module based trusted boot.

Tenant domain contains the software elements spanning from User Subscriber Identity Module (USIM) to Mobile Equipment applications and from different radio access technologies in access network domain to serving, home, transport network as well as Internet Protocol Service provider. New domains introduced

for 5G architecture include identity management (IM), slice, third party (3P), as well as management domains. Identity Management domain enables alternatives to USIM-based authentication. Slice domains enables the network to provide virtual networks dedicated for particular applications, providing e.g. ultra-low latency for critical industry automation or optimization for real-time multimedia. Slices also provide customization and accuracy to security services (by allowing special AAA solutions, unified threat management, or strong isolation of information). Third party domains capture use cases where operator allows trusted parties to provide specific services such as Authentication, Authorization and Accounting (AAA) or credential management. Management domains supports virtualization, management of security and security monitoring in other domains.

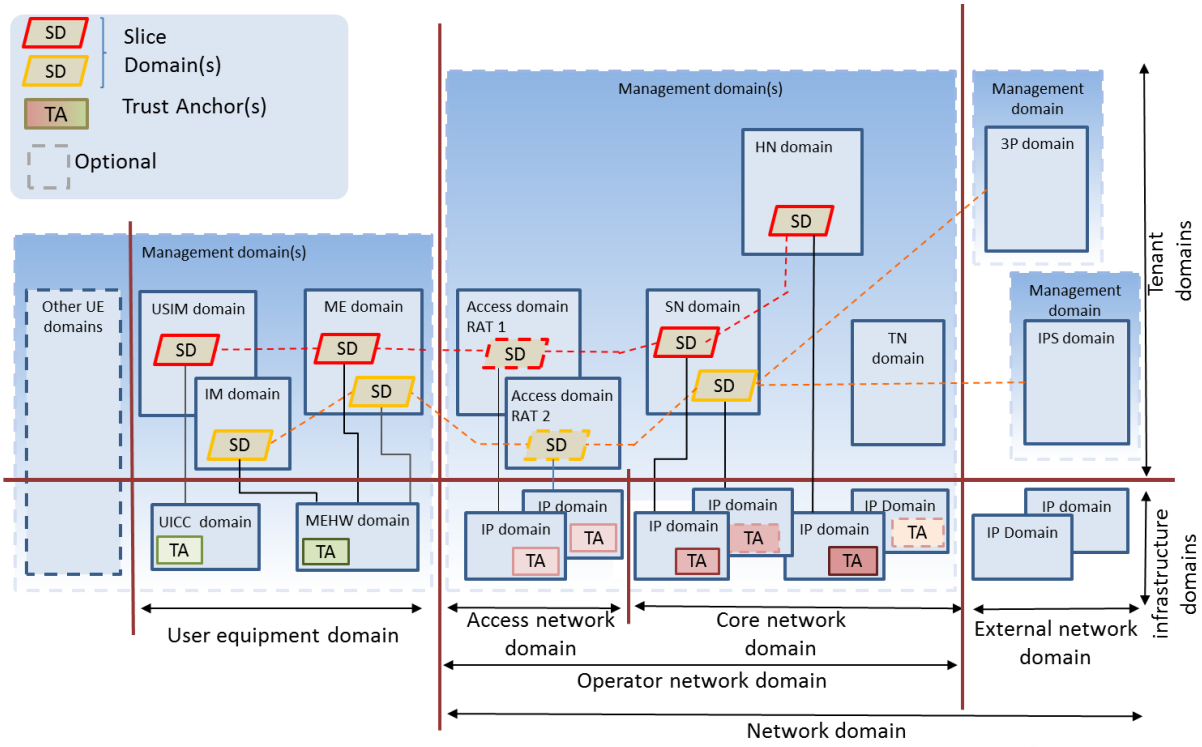


Figure 1-1: Domains in the draft 5G security architecture

The architecture uses concept of **stratum** to group protocols and functions that are related to one aspect of the network services. The architecture has five different strata [Figure 1-2]: access, transport, serving, home, application, and management stratum. The first four are based on 3GPP definitions [TS33.401, TS23.101, TS33.102] and the last one is a new extension introduced for 5G networks. Management stratum comprises aspects related to conventional network management (configuration, software upgrades, user account management, log collection/analysis) and, in particular, security management aspects (security monitoring audit, key and certificate management, etc.). In addition, aspects related to management of virtualization and service creation/composition (orchestration, network slice management, isolation and VM management, etc.) belong to this aspect.

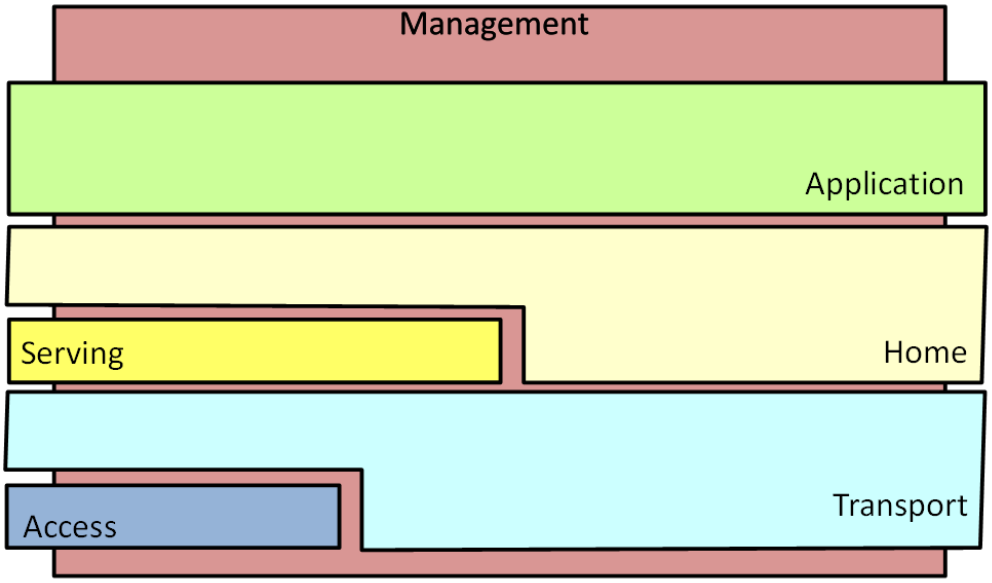


Figure 1-2: Strata in the draft 5G security architecture

1.2 Document structure

This document is organized as follows:

- Section 1 is a general introduction.
- Section 2 is devoted to the AAA category of enablers.
- Section 3 is devoted to Privacy category of enablers.
- Section 4 is devoted to Trust category of enablers.
- Section 5 is devoted to Security monitoring category of enablers.
- Section 6 is devoted to Network Management & Virtualization category of enablers.
- Section 7 concludes the document, while References are provided at the end.

The category descriptions of Sections 3-7 provide the open specification of each of the enablers planned to be software released for v2, this in accordance with features as per Technical Roadmap latest update (i.e. D3.5).

Overall, this deliverable paves the way towards the development and demonstration of the second and final set of 5G-ENSURE security enablers as planned for v2.0 in the project's Technical Roadmap(Update) (i.e. D3.5). As for D3.2, D3.6 provides also valuable input to both work on the 5G Security architecture and 5G Security testbed, since it provides details to link security enablers to 5G security architecture defined and under consolidation, but also to plan their integration, testing, demonstration, and assessment on the 5G security testbed.

2 AAA Security Enablers open specifications

2.1 Basic AAA enabler: Open specifications

2.1.1 Preface

5G brings a wide range of new requirements to the mobile networks. The AAA algorithms from earlier generations will likely remain in a similar form also. This enabler studies potential improvements of AAA while assuming most properties are reused from previous generations.

2.1.2 Copyright

2.1.3 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

2.1.4 Terms and definitions

Authentication, Authorization and Accounting	
Authentication and Key Agreement	Protocol that establishes mutual authentication and a shared key used for further protection of the communication
Authentication Management Field	An input parameter of the AKA' authentication algorithms. (in 4 hex digits)
Access Stratum Mobile Equipment	Notation used to indicate that it is a key shared between the Access Stratum (MME) and the ME (should be UE as it could be on USIM)
Authentication Vector	Set of quadruplets {XRES, CK, IK, AK} user for mutual authentication of network and UE, and for key establishment
Ciphering Key	Key used for confidentiality protection of information
Evolved Packet Core	Is a framework for an evolution or migration of the 3GPP system to a higher-data-rate, lower-latency, packet-optimized system that supports, multiple RATs (From TR21.905 v13.0.0 [1])
Evolved Packet System	It is an evolution of the 3G UMTS characterized by higher-data-rate, lower-latency, packet-optimized system that supports multiple RATs. The Evolved Packet System comprises the Evolved Packet Core together with the evolved radio access network (E-UTRA and E-UTRAN). (From TR21.905 v13.0.0 [1])
Evolved UTRA/Evolved UTRAN	Evolved UTRA is an evolution of the 3G UMTS radio-access technology/network towards a high-data-rate, low-latency and packet-optimized radio-access technology.
Home Environment	Responsible for overall provision and control of the Personal Service Environment of its subscribers (From TR21.905 v13.0.0 [1])
Hashed Message Authentication Code	Integrity check calculated by using hash functions with a secret key as input as well as the message.
Home Subscriber Server	Server managing subscription related information

	including keys and functions for producing authentication vectors
Integrity Key	Key used for integrity protection of information
Long Term Evolution	Name of the 3GPP project for E-UTRAN, has since been adopted as the name for E-UTRAN. Does not include EPS.
Message Authentication Code	A secret key dependent checksum that is used to verify the authenticity and integrity of data
Mobile Equipment	The mobile device excluding the USIM/UICC
Man-In-The-Middle	An attack where a man in the middle can establish connections with both communicating parties without either end point being aware that data is relayed between an entity in between.
Public Land Mobile Network	A network that provides communication services according to the specifications to mobile users in a geographical area
Serving GPRS Support Node	A node responsible for the delivery of data packets from and to the UEs within its geographical service area. Its tasks include packet routing and transfer, mobility management, logical link management, and authentication and charging functions.
Serving Network	The serving network provides the user with access to the services of home environment (From TR21.905 v13.0.0 [1])
User Equipment	ME+USIM
Universal Mobile Telecommunications System	An umbrella term for the third generation radio technologies developed within 3GPP
Universal Subscriber Identity Module	An application residing on the UICC used for accessing services provided by mobile networks, which the application is able to register on with the appropriate security (from TR21.905 v13.0.0 [1])
Visitor Location Register	A database that contains information about roaming subscribers within a locations area.

2.1.5 Overview

The Basic AAA enabler builds on the assumption that the AAA framework utilized by 4G to a large degree is inherited also into 5G. The enabler considers two areas of new functionality or improvements, namely AAA aspects of trusted micro-segmentation in 5G networks; Perfect Forwards Secrecy for the AKA protocol. The features will be described independently in the document.

AAA aspects of trusted micro-segmentation in 5G networks

Micro-segments are virtualized, isolated parts of the 5G network, meant to provide customizable security to subscribers, which can be hospitals, factories, and Industrial Internet and Internet of Things (IoT) based companies. Micro-segmentation aims to provide a more homogeneous and smaller environment to manage by security monitoring. By means of this, better accuracy can be achieved for e.g. anomaly detection. The AAA method in micro-segments depends on the needed security level, which is determined by the micro-segment subscriber.

The micro-segmentation enabler (Section 6.4) incorporated EAP Pre-Shared Key (EAP-PSK) based authentication method in the first release. In the second release of the enabler, two authentication

methods shall be supported: one based on EAP-PSK and the other on EAP Authentication and Key Agreement (EAP-AKA). The latter authentication method is developed in the Privacy Enhanced Identity Protection Enabler (Section 3.2). The main goal of this enabler is to provide stronger protection of user identity, e.g., by hiding the International Mobile Subscriber Identity (IMSI).

Perfect Forward Secrecy

This feature introduces perfect forward secrecy into the AKA protocol to ensure that a compromised key does not affect the secrecy of messages protected by other keys established as a result of previous or future AKA runs. The aim is to only modify the necessary parts of the current AKA protocol and reuse properties, such as mutual authentication.

2.1.6 Basic concepts

2.1.6.1 Perfect Forward Secrecy

Perfect forward secrecy is a property of a protocol which ensures that a key leakage does not compromise sessions using successor keys. In the following we will also consider solutions that provide protection of future sessions against passive attacks, however, we will not consider protection against future active attacks.

2.1.6.2 UMTS / LTE AKA procedure

On a high level, AKA works as in **Figure 2-1**

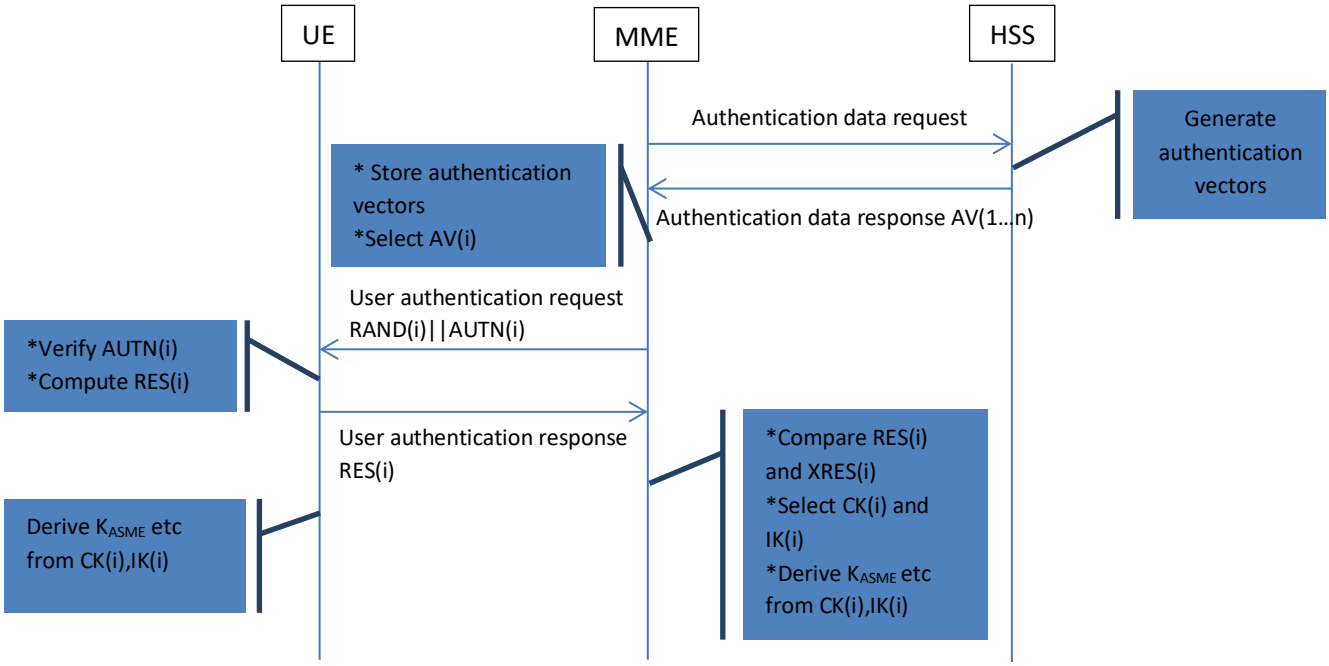


Figure 2-1: AKA procedure

AUTN is a parameter composed of different fields: AMF, MAC and a sequence number indication (SQN, possibly encrypted by an anonymity key AK). The MAC is a Message Authentication Code that protects RAND, SQN and AMF from being forged by a 3rd party through the cryptographic functions implemented by the USIM as shown in **Figure 2-2**. The keys CK and IK are used directly for ciphering /integrity protection in 3G and are used indirectly for these purposes in 4G/LTE by deriving ciphering/integrity keys from CK and IK.

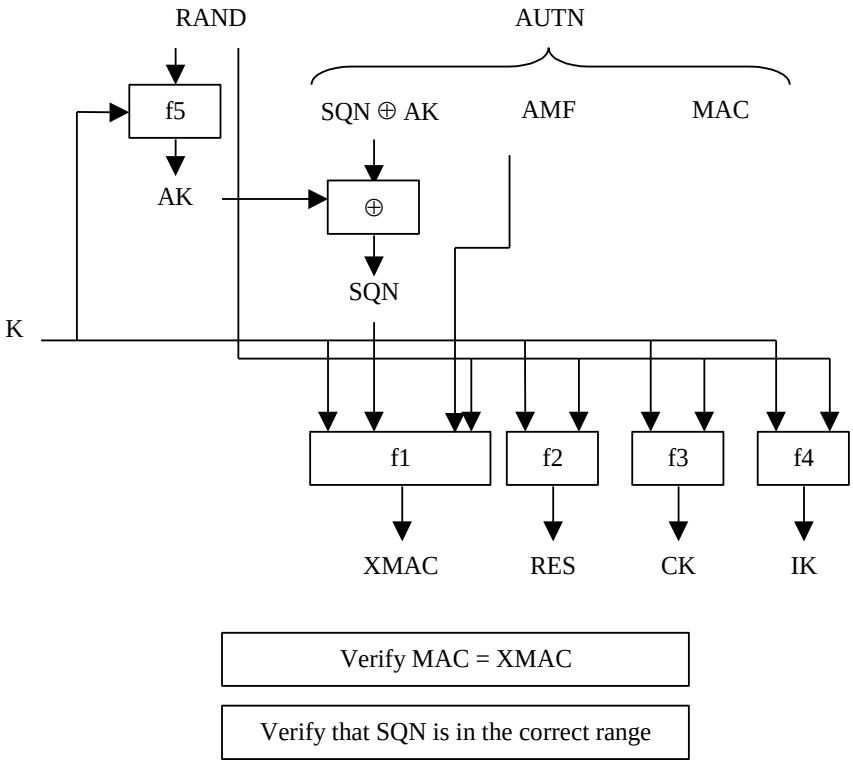


Figure 2-2: USIM AKA calculations

2.1.6.3 Diffie-Hellman key exchange protocol

The original protocol is based on the multiplicative groups of integers module a prime number *p*, the generator element *g* is primitive root modulo *p*. Both parties each randomly generates a secret value, the UE generates value x, and the network generates value y. The UE calculates g^y (which does not need to be kept secret) and sends the result to the network, the network calculates g^x and sends to the UE, see **Figure 2-3**.

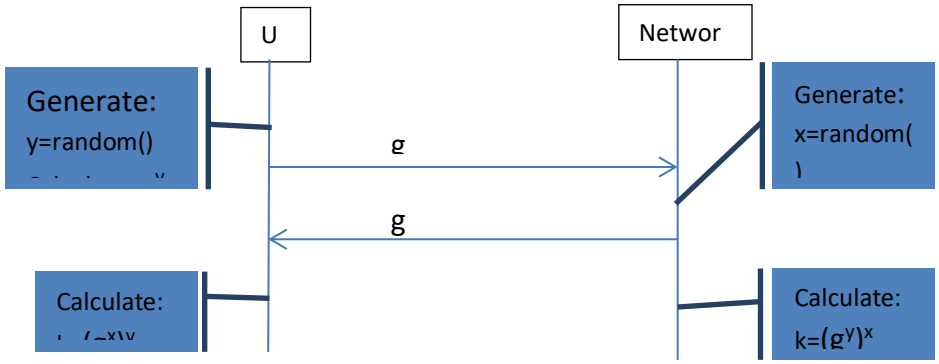


Figure 2-3: Diffie-Hellman key exchange procedure

The UE and the network can then each calculate the shared secret key as $k=(g^x)^y=(g^y)^x$. For an adversary with knowledge of g^x and g^y it is believed difficult to derive g^{xy}.

2.1.7 Main interactions

2.1.7.1 Use cases

AAA aspects of trusted micro-segmentation in 5G networks

Once a micro-segment has been created, nodes can be added to the micro-segment. Before adding nodes to the micro-segment, they need to be authenticated by the micro-segment AAA entity and subsequently authorized to use the micro-segment. As mentioned earlier, the AAA methods in micro-segmentation depend on the used application or the requirements of the micro-segmentation subscriber. The AAA method is thus fully customizable.

An example of a new AAA policy could be the setting of more detailed requirements on the authentication of users. Also, preventing users or group of users from the use of a micro-segment is another possible action.

Perfect Forward Secrecy

In this use case, a subscriber and a MNO want to reduce the security effect of any compromised keys shared between the subscriber and the operator. Current AKA procedures suffer from the fact that a compprompise of the long term keys stored in HSS/USIM can lead to all future, and past messages, sent with keys derived from the compromised key can be decrypted. It is proposed that the key agreement protocol is updated to use the Diffie-Hellman key exchange protocol instead to ensure that a compromised key only affect sessions in which that particular key is used. To prevent man-in-the-middle attacks on the Diffie Hellman protocol, the authentication aspects of the AKA procedure are reused.

2.1.7.2 Components and interaction overview

AAA aspects of trusted micro-segmentation in 5G networks

Each micro-segment could have its own AAA entity, but the AAA functionality of a micro-segment should not add to the overall complexity. This component would be a virtualized resource or function, i.e., Network Function Virtualization (NFV) would be used. It is for further study how to bootstrap the micro-segment AAA entities.

Perfect forward secrecy

The Diffie-Hellman protocols in general requires transportation of parameters that are much larger than the parameters of the AKA protocol (RAND, RES, etc.). Even if it is possible to increase the number of bits signalled over the air interface, it would be desirable to maintain the standardized USIM-ME interface, which implies a bottleneck for the size of the protocol parameters which falls below the level where DH offers strong security. RAND is currently 128 bits, and to reach a security matching the 128-bit strength of AKA:

- DH parameters of at least 256-bits are need for elliptic curve variants of DH
- around 3000-bits for standard discrete log DH modulo a prime, p
- around 6000-bits for supersingular isogeny DH, out of which it is sufficient to transfer roughly 3000 bits (this variant is believed to be quantum computer immune).

In the context of 5G it is important to keep the computational complexity low, but even more important to keep the message lengths short. Hence, it is recommended to use elliptic curve variants of DH short term,

and long-term transmission to quantum immune alternatives. It is also beneficial to select elliptic curves, we recommend the use of X22519 function [8] for ECDH over the elliptic curve Curve22519[6], because it is believed to be easier to implement it correctly compared to e.g., the NISTp256 curve, see [7].

Another aspect to consider is that DH is susceptible to a man-in-the-middle (MITM) attacks, as a result, a mechanism to authenticate the DH parameters is necessary.

The RAND information element can be used to carry a DH value when sent from the network to UE: RAND = gx. (for elliptic curves, the commonly used notation is RAND=xG). The result is likely to be substantially larger than the current size of 128-bits. To maintain the USIM-ME interface, we propose to compress RAND, e.g. by cryptographic hashing: RAND' = H(RAND) = H(gx), where H denotes taking the 128 LSB of SHA256, i.e., H(x)=TRUNC128-LSB(SHA256(x)). H is applied in the ME, before inputting RAND' to the USIM. Consequently, the AKA MAC-field will be computed in dependency of RAND', but through the use of H, it will in effect still be computed in dependency of gx. Therefore, the authenticity of the DH value gx is ensured between the AuC/HSS and the USIM, which prevents MITM attacks, in particular between the serving network and UE.

The UE will need to compute a response DH-value of the form gy, which is computed from a (pseudo) randomly generated value, y, in the UE. Again, to thwart MITM attacks, this value must be authenticated. To this end, however, we propose to use RES and add a standard MAC, i.e., by computing RES' = MAC(RES, gy), and thus the UE responding with RES', gy. It is proposed that the MAC function is based on HMAC.

Note that the AuC/HSS needs to generate the authentication vectors sent to the serving network accordingly, i.e. compute parameters: RAND' = H(gx), before inputting it to the f-functions, etc.

Note: In principle, the HSS does not need to send CK, IK (or keys derived therefrom) to the MME as part of Authentication Vectors (AV) since the resulting shared key will be based on gxy which is not known to the HSS at time of AV generation. However, in EPS/LTE keys are "bound" to the access network through inclusion of a PLMN identifier in the derivation of the KASME key from CK, IK. Since the HSS as noted does not know gxy at the point when AVs are generated, a binding to PLMN ID could be achieved by including the PLMN ID in derivation of some further key from CK, IK and including that derived key in the AV.

Also the MME would need updates, i.e. given XRES in the AV from the HSS, it would compute XRES' = MAC(XRES, gy) before verifying subscriber authenticity, and it would need to derive KASME as KDF(gxy), where KDF denotes the key derivation function as defined in TS 33.401 [2] (accompanied by some newly chosen string S). The UE would compute KASME similarly. The updated AKA procedure is shown in **Figure 2-4**.

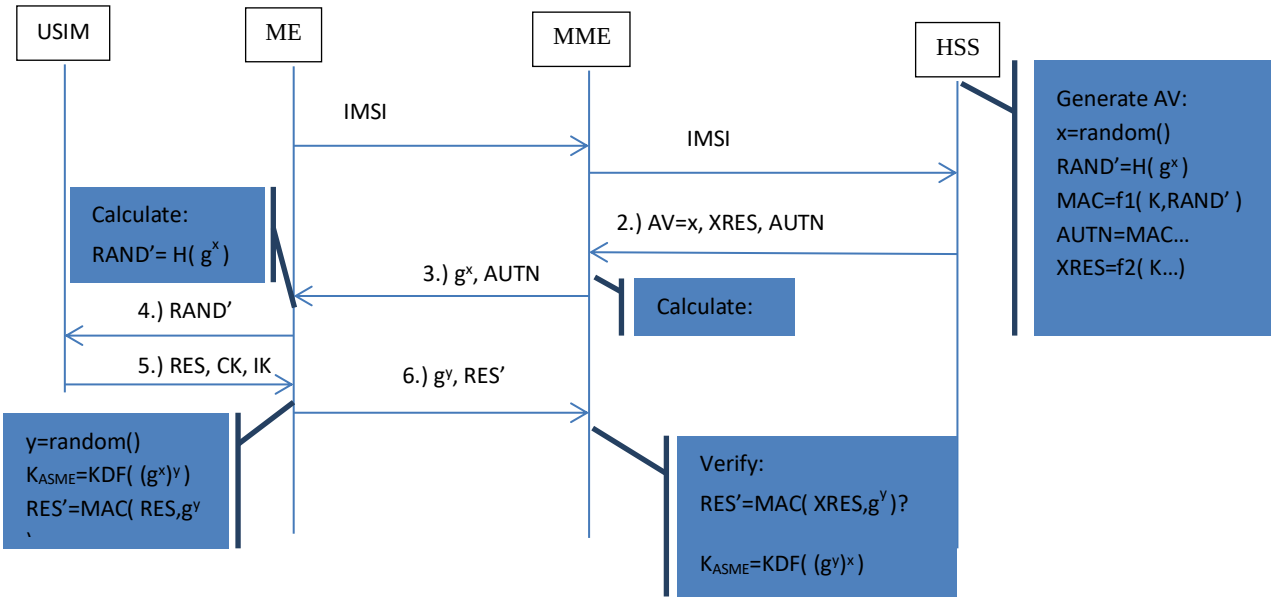


Figure 2-4: Adding PFS to the USIM AKA procedure

2.1.8 Architectural drivers

2.1.8.1 High-Level functional requirements

AAA aspects of trusted micro-segmentation in 5G networks

In general, the AAA methods of a micro-segment should not add to the complexity of the whole system. The authentication protocol of micro-segments depends on the needed security level, which is adjustable. For example, if the application or service requires strong authentication, Extensible Authentication Protocol over LAN (EAPoL) may be used.

Micro-segments may support and require particular authentication mechanisms. For example, if the mobile device has been authenticated strongly to the mobile network by the use of USIM, it can be authorized to use the micro-segment. However, if lighter authentication methods have been used in the mobile network, the device may need to be re-authenticated using a stronger or a second-factor mechanism to authorize use of the micro-segment.

Perfect Forward Secrecy

The AKA procedure shall provide perfect forward secrecy, i.e., a compromised long term keys shall not affect keys established during other AKA procedures.

2.1.8.2 Link to Security Architecture

The PFS feature relates to the following security objectives:

O2.1 - 5G must provide a security and privacy level higher or at least equal to the security and privacy level in 4G.
O4.3 - 5G infrastructure components should support necessary root-of-trust functionality.
O7.4 - 5G systems and components must provide strong mutual authentication and authorization

Figure 2-5 and **Figure 2-6** show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016]

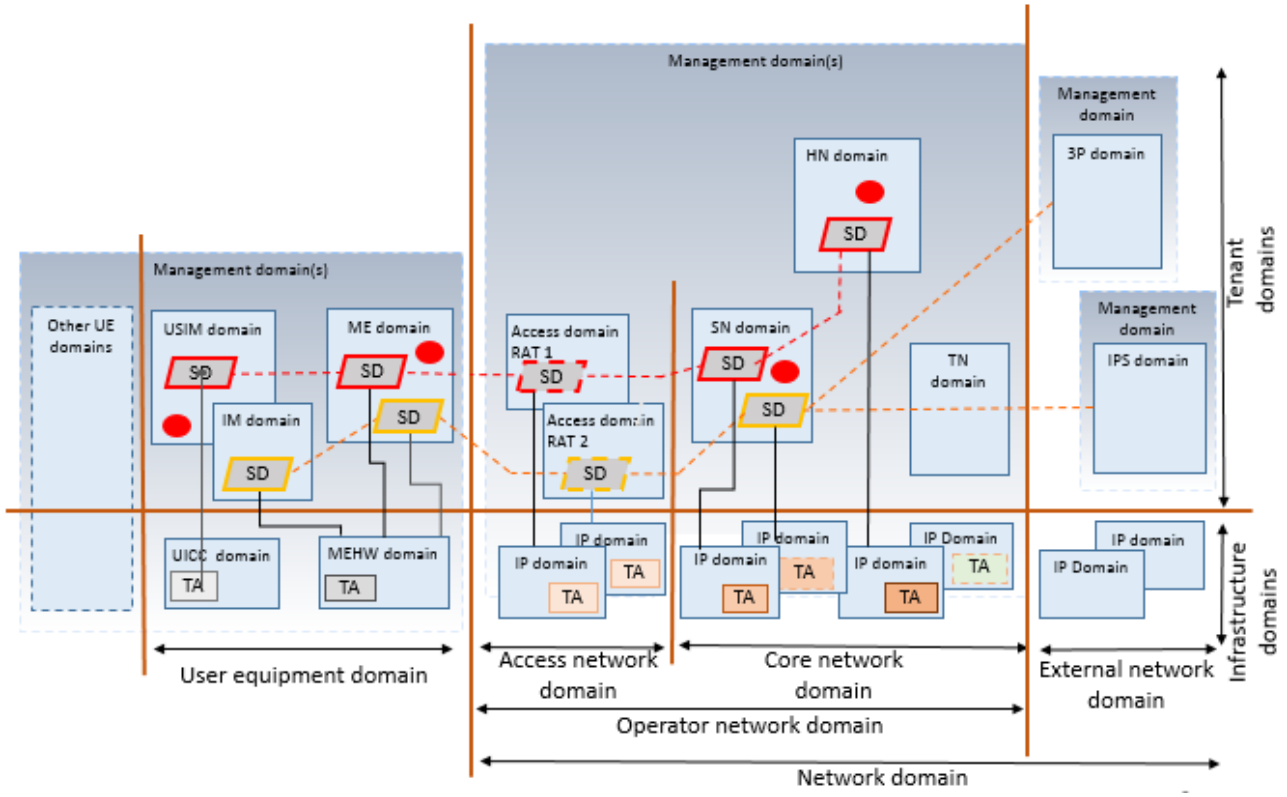


Figure 2-5 Enabler feature relevant domains

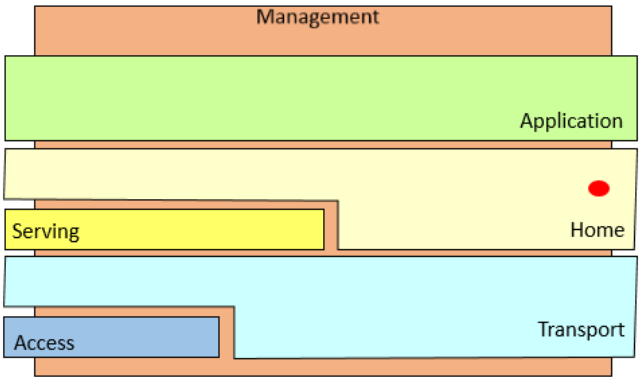


Figure 2-6 Strata of the 5G Security Architecture

2.1.8.3 Quality attributes

A product implementing one or several of the features described in the enabler should be evaluated primarily based on the following quality attributes:

- Security:
  - Confidentiality
  - Integrity
  - Non-repudiation
  - Accountability
  - Authenticity
- Compatibility
  - Co-existence
  - Interoperability
- Performance efficiency
  - Time Behavior
  - Resource Utilization

2.1.8.4 Technical constraints

The enabler is scheduled for release 2 and is currently in a research phase, thus there are no known technical constraints at this stage.

2.1.8.5 Business constraints

No known business constraints.

2.1.9 Detailed specifications

2.1.9.1 Introduction

In this section, detailed information about the required updates to the 3GPP specifications to implement the suggested AAA enhancements are provided.

2.1.9.2 Conformance

An implementation to be reported as conformant should comply with the open specifications here stated for the enabler.

2.1.9.3 API specifications

Perfect Forward Secrecy

For the interface between the MME and the HSS, i.e., S6a, a new Authentication-Info needs to be defined, as documented by TS29.272 [3]

5G-Vector ::= <AVP header: 14xx 10415>

[Item-Number]

{ x }

{ XRES }

{ AUTN }

*[AVP]

In addition, the NAS AUTHENTICATION REQEST message needs to be updated according to **Table 2-1** (see TS24.008 v13.5.0 [4] section 9.2.2 for original values).

Table 2-1: 3GPP TS 24.008: AUTHENTICATION REQUEST message content

IEI	Information element	Type/Reference	Presence	Format	Length
	Mobility management protocol discriminator	Protocol discriminator TS24.008 section 10.2	M	V	½
	Skip Indicator	Skip Indicator TS24.008 section 10.3.1	M	V	½
	Authentication Request message type	Message type TS24.008 section 10.4	M	V	1
	Ciphering key sequence Number	Ciphering key sequence number TS24.008 section 10.5.1.2	M	V	½
	Spare half octet	Spare half octet TS24.008 section 10.5.1.8	M	V	½
NEW	Authentication element is replaced by: g^x (Diffie-Hellman parameter used for key establishment)		M	V	33
20	Authentication Parameter AUTN	Auth. parameter AUTN TS24.008 section 10.5.3.1.1	O	TLV	18

The AUTHENTICATION RESPONSE also needs to be updated according to **Table 2-2** (see TS24.008 v13.5.0 [4] section 9.2.2 for original values).

Table 2-2: TS24.008: AUTHENTICATION RESPONSE message content

IEI	Information element	Type/Reference	Presence	Format	Length
	Mobility management protocol discriminator	Protocol discriminator 10.2	M	V	1/2
	Skip Indicator	Skip Indicator 10.3.1	M	V	1/2
	Authentication Response message type	Message type 10.4	M	V	1
	Authentication Response parameter	Auth. Response parameter 10.5.3.2	M	V	4
NEW	g^x (Diffie-Hellman parameter used for key establishment)		M	V	33
21	Authentication Response Parameter (extension)	Auth. Response parameter 10.5.3.2.1	O	TLV	3-14

Note: This feature is backward compatible with USIM, the IMSI cannot indicate whether the ME supports it or not. It is for further study if it shall be mandated for all 5G accesses, or if we need some “classmark” too, to indicate ME support.

AAA aspects of trusted micro-segmentation in 5G networks

Micro-segmentation does not require an API since it is primarily a study item, and as mentioned in the updated Technical Roadmap (D3.5), the intention is primarily to investigate the usage of micro-segmentation in the context of AAA infrastructure, rather than to specify API:s.

2.1.9.4 References

[1] 3GPP, *TR 21.905* Vocabulary for 3GPP Specifications.

[2] 3GPP, *TS 33.401 ;* 3GPP System Architecture Evolution (SAE);Security architecture; *v13.2.0*

[3] 3GPP, *TS 29.272;* Evolved Packet System (EPS); Mobility Management Entity (MME); Serving GPRS Support Node(SGSN) related interaces based on Diameter protocol*; v13.5.1.*

[4] 3GPP, *TS24.008;* Mobile radio interface Layer 3 specification; Core Network Protocols*; v13.5.0.*

[5] Bernstein, J., "Curve25519: new Diffie-Hellman speed records ", WWW http://cr.yp.to/ecdh/curve25519-20060209.pdf, February 2006.

[6] Daniel J. Bernstein and Tanja Lange. SafeCurves: choosing safe curves for elliptic-curve cryptography. https://safecurves.cr.yp.to, accessed 1 December 2014.

[7] IETF RFC7748, Elliptic Curves for Security

2.2 Internet-of-Things: Open specifications

2.2.1 Preface

Internet-of-things (IoT) is expected to be an essential use case and business area in 5G. This security enabler provides important new features, with their specification, to the existing AKA protocol that is directly aimed at enabling IoT.

The main feature in this enabler is the group-based authentication [1] which was developed for release 1. In release 2 of the group-based authentication it is extended with multi-UE capabilities and perfect-forward secrecy. This enabler also includes features described in R1 but are here updated and explained in more detail in the scope of R2, namely: authentication of USIM-less devices; authentication support for BYOI.

Moreover, the open specification includes an additional feature, vertical Generic Bootstrapping Architecture (GBA), which was not part of the 5G-PPP security enablers' technical roadmap (early vision). This feature has been added to support additional requirements coming from the use case 3.2 Network-Based Key Management for End-to-End Security from deliverable D2.1.

2.2.2 Status

A prototype implementation of the group-based authentication feature was delivered in Release 1 and a new implementation version, for Release 2, is under development. Preliminary specifications for the Release 2 prototype are given in this document. The other features of the enabler will not be implemented in the 5G-ENSURE project but are included in the specifications of the enabler given in this document.

2.2.3 Copyright

2.2.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

2.2.5 Terms and definitions

A group is formed by one or more members that share similar features. Examples of common features include members that do the same task, members located in the same geographical area, or members belonging to the same owner. A group may also share a macro feature that is derived by a combination of single features.

An inverted hash tree [2] is a data structure in which a node is linked to at most two successors (children), and the value of each node is computed using a family of hash functions h_*. The value of the root is given, while the value associated with any other node is derived from the hash value of its parent (Figure 1). Since two siblings use two different hash functions, they get two different hash values. In particular, we consider two hash functions h_0 and h_1, and recursively assign the value of each node n_{ij} located at i^{th} position and j^{th} level as follows.

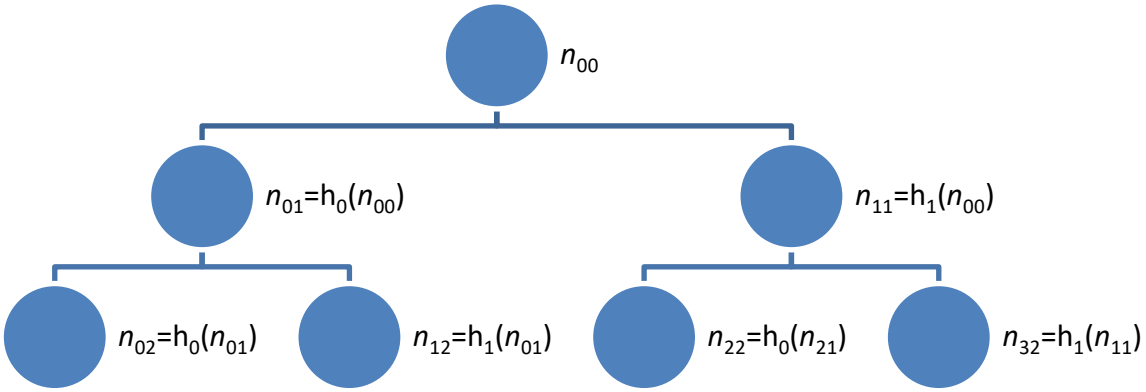


Figure 2-7: **Example of an inverted hash tree of level 2**

2.2.6 Overview

The IoT Enabler consists of four main features: support for authenticating USIM-less devices; enabling authentication based on 3rd party identities, i.e. bring-your-own-identity (BYOI); vertical GBA; and introducing support for group-based authentication, aimed at resource constrained devices. As mentioned above, all features will be introduced in the first chapters, while group-based authentication release 2 will be described in more detail. Each enabler feature will be described independently.

An important aspect of IoT, in the context of 5G, is to be able to use non-USIM/SIM based authentication schemes in a 3GPP network. The feature *authentication of USIM-less* devices considers the adaptions to the AAA framework needed to introduce such a feature in the authentication schemes in a 3GPP network, including aspects of identifying which algorithm to use in the AKA as well as which AAA server to contact.

A similar adaption to authentication in 5G, with the expected boom of devices, is the *group-based authentication* feature, which introduces an extension of the current EPS-AKA, mainly aimed at resource constrained devices. The goal of the group-based AKA protocol is to make it more suitable for massive deployment scenarios by reducing the signaling between the serving and home network.

Thirdly, the *vertical Generic Bootstrapping Architecture (GBA)* feature optimizes the use of GBA (3GPP TS 33.220) for resource constrained devices by having the device implicitly perform the GBA bootstrapping during authentication to the network, in the network attach. Thus, the device does not have to perform the HTTP Digest AKA negotiation with the BSF to bootstrap its GBA credentials. Once attached to the 3GPP core the device can set-up a secure connection, e.g. perform GBA based HTTP Digest authentication, to a NAF without additional signaling to the BSF. The solution is called vertical GBA (vGBA) as the keys generated during AKA are re-used at different layers in the stack. The solution is described using the nodes of 4G networks, as 5G network nodes are yet to be specified.

With 5G's expected "as a service" approach, the IoT enabler will also introduce a framework for enabling an organization to reuse an existing AAA infrastructure to authorize 5G access via a MNO. The access stratum may be provided by either the MNO or the organization itself, it is however assumed that the organization is unable, or find it undesirable, to provide a full core network. From a high level, the setup is technically similar to a Mobile Virtual Network Operator (MVNO) providing networks services via a MNOs network. However, from a trust perspective, it introduces entirely new aspects as the organization may not abide by

the same rules and regulations as a classical operator. It is thus possible that the organization may need to adopt to additional regulatory frameworks, but this is outside the technical scope of the enabler. A new entity, the Industrial Automation Control (IAC), which is functionally similar to an MME, is introduced into the operator network in order for the operator to distinguish between roaming agreements with other MNOs and other external organizations.

2.2.7 Basic concepts

Relevant LTE/EPS signaling entities and interfaces during roaming

The interface between a UE and the eNB is denoted Uu, the interface between an eNB and an MME is denoted S1-MME. The interface between the MME and the HSS is denoted S6a. In a classical roaming scenario, the MME and eNB are provided by a visited network, the MME will request authentication data from the HSS located in the home network of the subscriber.

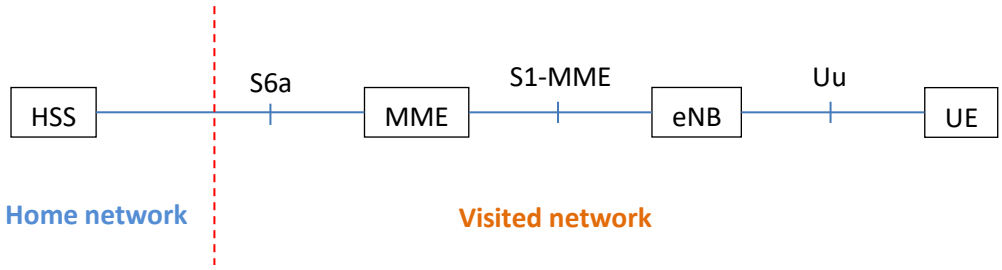


Figure 2-8: Signalling entities and interfaces during roaming

The authentication vector on the S6a reference point

When a UE attaches to the 3GPP core network it performs the AKA exchange for mutual authentication and key agreement with the network. The AKA is performed between the UE and the MME, which fetches authentication vectors (AV) from the HSS over the S6a reference point.

Generic Bootstrapping Architecture

GBA is a way of using the 3GPP subscription credentials for authenticating the UE and getting keys to a service (Network Application Function, NAF). To achieve this, the UE first bootstraps with the Bootstrapping Server Function (BSF), a bootstrapping entity in the 3GPP core, using the 3GPP credentials and the HTTP Digest AKA protocol. The BSF fetches AVs from the HSS over the Zh reference point, in a similar way as the MME over the S6a reference point. After bootstrapping, the UE can, e.g. using HTTP Digest, authenticate to the service (NAF) based on the bootstrapping context with the BSF. The BSF provides the NAF with a service specific key that the NAF can use for authentication of the UE and securing further communication of the session.

Group-based AKA

The functional goal of a group-based AKA protocol is to authenticate a group of devices efficiently, minimizing the cost of repeated message exchanges and communication delays. More specifically, a group-

based AKA protocol aims to reduce the signalling between the mobility management entity (MME) and home subscriber server (HSS) when a large group of machine-type communication (MTC) devices with similar features requires network access simultaneously.

Perfect forward secrecy

Perfect forward secrecy (PFS) is a property of key-exchange protocols that enables session key secrecy in the event of a long-term key exposure after a session [3]. The property is often achieved by introducing Diffie-Hellman-Merkle (DH) exchange to the key-exchange protocol and using an ephemeral secret derived from the DH exchange to establish session keys. A widely used modern variation of the DH exchange is called elliptic curve Diffie-Hellman (ECDH) exchange and works in a similar manner, and can also be used to achieve PFS in key-exchange protocols.

2.2.8 Main interactions

2.2.8.1 Use cases

A 5G PLMN is expected to support a much wider range of devices than previous generations. Additionally, slicing technology will also enable compartmentalization of devices that could potentially be classified into different security categories. It is expected that the USIM is unsuitable for certain categories, e.g. IoT and resource-constrained devices, which will be addressed by the *authentication of USIM-less devices* feature that considers the integration of non-USIM based AKA procedure into the 5G system.

Similarly, group-based authentication aims to increase the capacity for massive deployments of MTC devices in 5G. At large events, e.g. sporting events, we envision that 5G connected sensors and tracking devices will be deployed at a massive scale. We foresee that the existing EPS-AKA is not sufficiently efficient to handle the massive simultaneous authentication requests that these devices will incur. This is especially important in the case of device subscriptions in a roaming setup. With the group-based authentication approach, the signalling between the MTC devices and their HSS are significantly reduced, enabling the serving network to handle almost all the signalling independently from the home network, and reducing the latency drastically.

The MTC is preconfigured with the necessary group terms, such as the group ID. The pre-configuration can be performed via non-3GPP access. The identity of the MTC is stored in the USIM, fulfilling the requirements of securing the identities against attacks such as cloning.

We imagine that the configuration of the groups will be performed using an application programming interface (API) to access the network. It allows for an automated approach for submitting the MTC identities that should belong to the specific group.

To further enhance 5G for resource-constrained devices, the enabler will introduce vertical GBA. The basic use case for vertical GBA is that a constrained UE wishes to use some GBA enabled service(s). Constrained devices are in many cases constrained especially regarding power supply and might be operating solely on battery power. To optimize the performance of these types of devices, reducing the amount of signalling required by the device is an important step for improving the lifetime of the device. With vGBA, the UE does not have to perform an explicit GBA bootstrapping with the BSF, thus using vGBA reduces the amount of signalling needed for authentication to GBA enabled services (NAFs), using the pre-generated GBA credentials.

Finally, the use case for BYOI is to attract enterprises that already have a deployed AAA infrastructure to use 5G. By establishing a contract with a MNO, the enterprise can allow their devices to perform roaming and utilize other aspects of a PLMNO.

2.2.8.2 Components and interaction overview

Authentication of non-USIM based devices

Non-USIM based AKA procedures will impact the AKA protocol as well as the identification protocols utilized by the current 3GPP standards, e.g., in a certificate public key pair based authentication system the standard IMSI may not necessarily be an identifier that is used to identify the subscription.

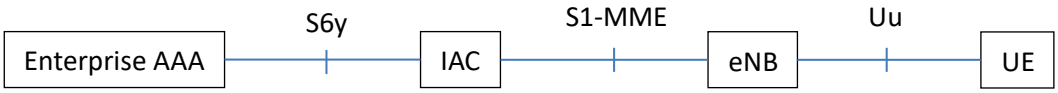


Figure 2-9: Network deployment considered for non-USIM based devices

In the considered solution, it is assumed that there will be no support for negotiation of authentication methods, the subscription will be associated with an identifier that maps uniquely to a provided ID. The ID provider may either be the MNO, that utilize HSS for AKA procedures, or an enterprise that has deployed an Enterprise AAA for the AKA procedures. It is also assumed that the identifier uniquely determines the AKA protocols to be used. A UE may support several AKA protocols and multiple ID providers, e.g., both towards a HSS and an Enterprise AAA. In such a situation, it is assumed that the UE selects which AKA protocol to use, e.g., based on the selected slice ID.

EAP is one natural candidate as it is standardized and can carry most authentication methods. In EAP, the identity type is used to query the identity the correct AAA server, and select a EAP method. It is generally used as the initial request message, but it is not required to be sent within the EAP conversation. Hence, it is possible to send the identity in the ATTACH REQUEST message and introduce new classmarks to identify the EAP methods supported by the UE. In **Error! Reference source not found.**, a certificate based EAP TLS AKA procedure is illustrated. Authentication is done using server certificate as well as UE certificates (with corresponding private keys). It assumed that K_{ASME} is returned from the AAA server to the MME as a part of the EAP Success message, and further key material is derived from K_{ASME} as in 4G. To reduce the number of sent messages, it is assumed that the EAP success message between the MME and the UE is sent in the Security Mode Complete message.

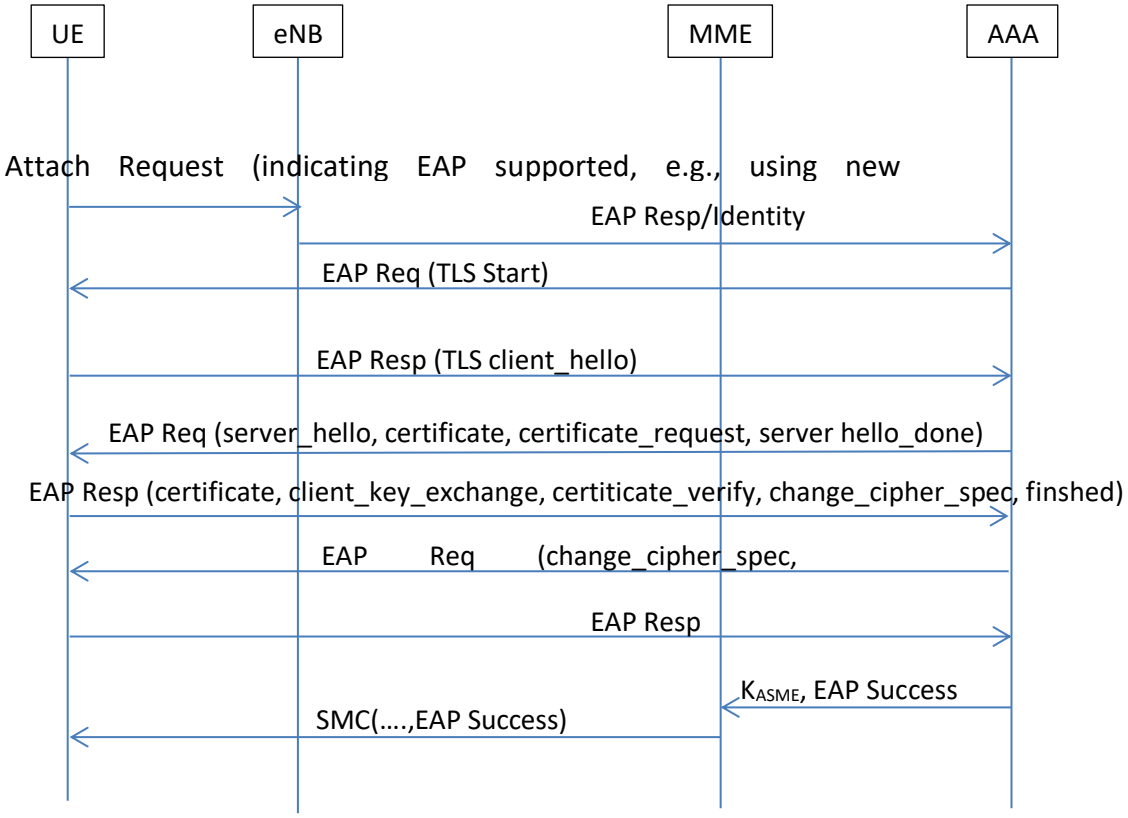


Figure 2-11: Certificate based EAP procedure

When previously established session can be resumed using a slightly more efficient handshake, this is illustrated in Figure 2-11

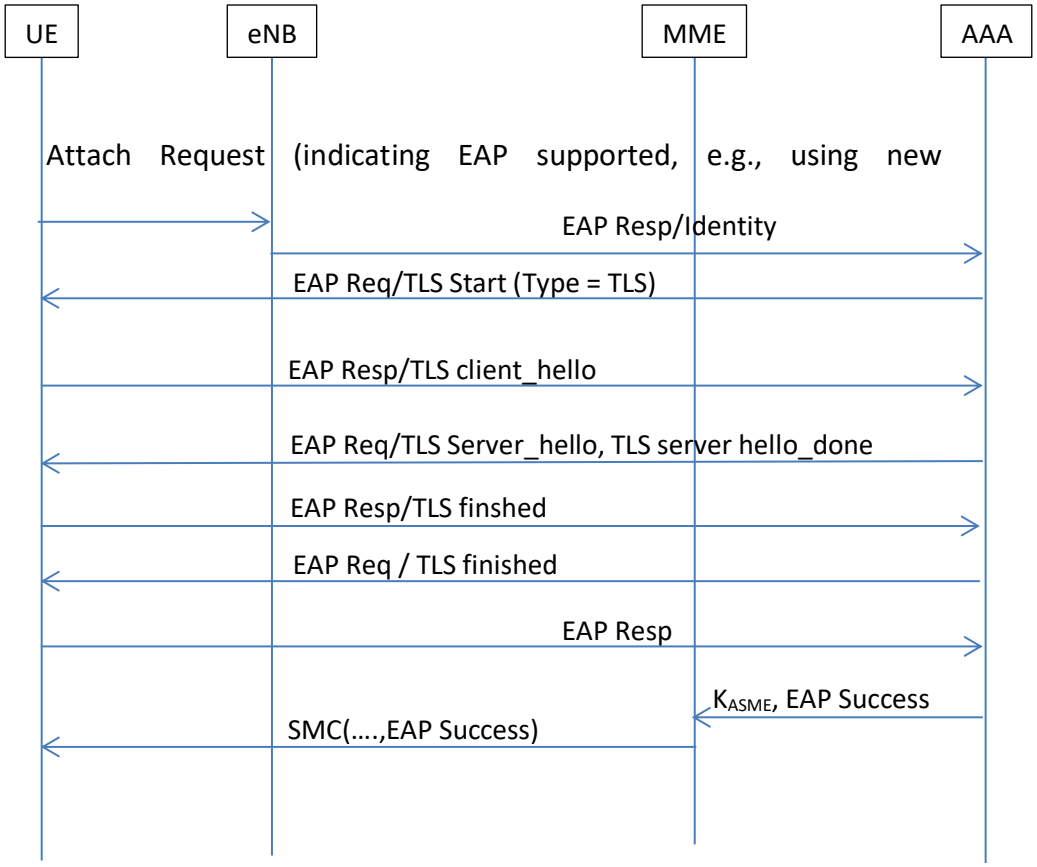


Figure 2-12

EAP always assumes that the authentication occurs in the home network. In 4G, a visiting network can request multiple authentication vectors from the home network, which allows the visiting network to perform multiple authentication rounds without the need to contact the home network each time. This can be seen as a positive property from a performance aspect, but from a security perspective, a requirement to always contact the home network (as is the case for EAP) is a positive property as authorization policy changes are taken into account at every authentication procedure.

Authentication support for BYOI

As illustrated by Use Case 1.1 – Factory Device Identity Management for 5G Access and Use Case 1.2 - Using Enterprise Identity Management for Bootstrapping 5G Access in [4], different levels of enterprise network deployment can be imagined. In all cases, the enterprise is assumed to operate its own AAA. In some scenarios, the organization own and operate their own base stations. However, this is similar to the Home eNodeBs already supported in LTE/EPS and should, as such, not introduce new aspects, beyond the currently defined Home eNB technical solutions. The deployment considered is summarized in figure 4.

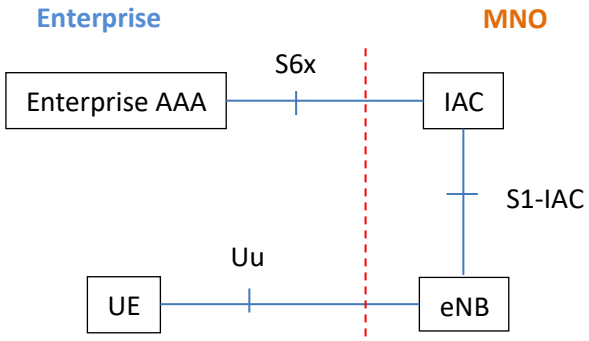


Figure 2-13: Network deployment where Enterprise operates an AAA

The main difference compared to classical roaming, is that the IAC may require some form of security assurance from the UE and Enterprise AAA during the AKA procedure: this could be in the form of a remote attestation procedure, where the device attests to its current security state; or could just be a classification of the security level of the algorithms used that can be used to assess which slice the UE is allowed to utilize. Alternatively, or in addition, a contractual agreement (with liabilities) between the MNO and the Enterprise could stipulate requirements on secure storage/processing of the credentials in the UE.

Vertical GBA

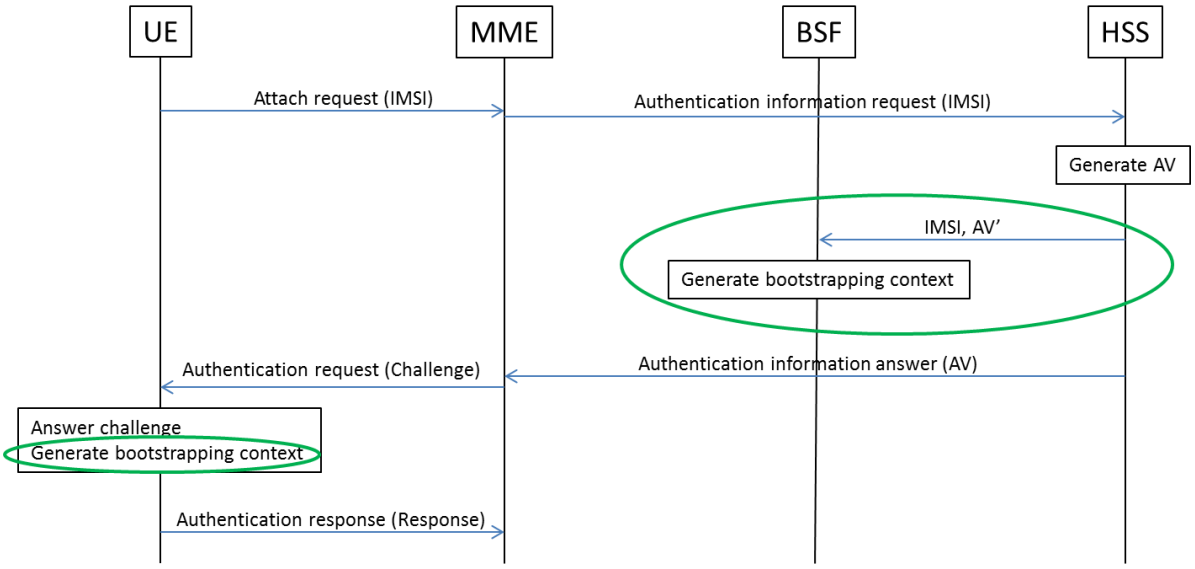


Figure 2-14: **vGBA bootstrapping during AKA**

Figure 2-14 shows the modified AKA flow when vGBA is used. During AKA the UE generates the GBA bootstrapping context and associated credentials. The HSS provides the BSF with a modified version of the AV (AV') for use for GBA, different from what is provided to the MME (AV). AV' is used by the BSF for generating the corresponding bootstrapping context to match a successful GBA bootstrapping end-result.

Group-based authentication

The underlying idea of the proposed group-based AKA is to associate each MTC device to a value of the leaf node of an inverted hash tree. The tree is generated by the HSS, which reveal a sub-root node to the MME so that the MME can authenticate a (sub)group of all MTC descendants without contacting the HSS. The protocol is based on an article by Giustolisi et al. [1].

The central properties of the proposed protocol follow:

Two inverted hash trees are generated by the HSS. One serves as the group key tree (GK tree) and the other a challenge key tree (CH tree) where each leaf in the same relative position of the GK tree and the CH tree are associated to a specific MTC device. Let us call such leaves CH_{MTC} and GK_{MTC}.

The leaves CH_{MTC} and GK_{MTC} can be hashed so that they are coupled with a sequence number N. They are then referred to as $H(CH_{MTC}, N)$ and $H(GK_{MTC}, N)$, respectively.

Each MTC device in a group is pre-loaded with one or more obfuscated values called $O_{(MTC, N)}$ from which the MTC can produce $H(GK_{MTC}, N)$ if $H(CH_{MTC}, N)$ and the MTC long term key K stored in the USIM are known. This is possible since $O_{(MTC, N)}$ = hash(K, $H(CH_{MTC}, N)$) XOR $H(GK_{MTC}, N)$.

An MME which receives (sub)root nodes of a GK tree and a CH tree and a valid sequence number can perform a challenge response scheme with all MTC devices that hold $O_{(MTC, N)}$ values corresponding to the sequence number and the tree leaves of the (sub)root nodes.

The above properties of the protocol would enable an MME that has received a GK tree (sub)root node and a CH tree (sub)root node to mutually authenticate and derive a master session key with MTCs, without contacting the HSS.

The sequence numbers have been introduced in Release 2. They could potentially be used for re-authentication and/or reallocation of authentication parameters to other MMEs. The idea is that an MTC could store many above mentioned obfuscated values $O_{(MTC, N)}$ for a range of sequence numbers. This would enable for several secure runs of the protocol. Another addition in Release 2 of the feature is an ECDH exchange between the MTC and the MME. The ECDH exchange enables PFS in the protocol.

Two cases can be distinguished in the proposed protocol: Case A, in which the MME cannot derive the keys for the MTC without signalling with the HSS; Case B, in which the MME can derive such keys.

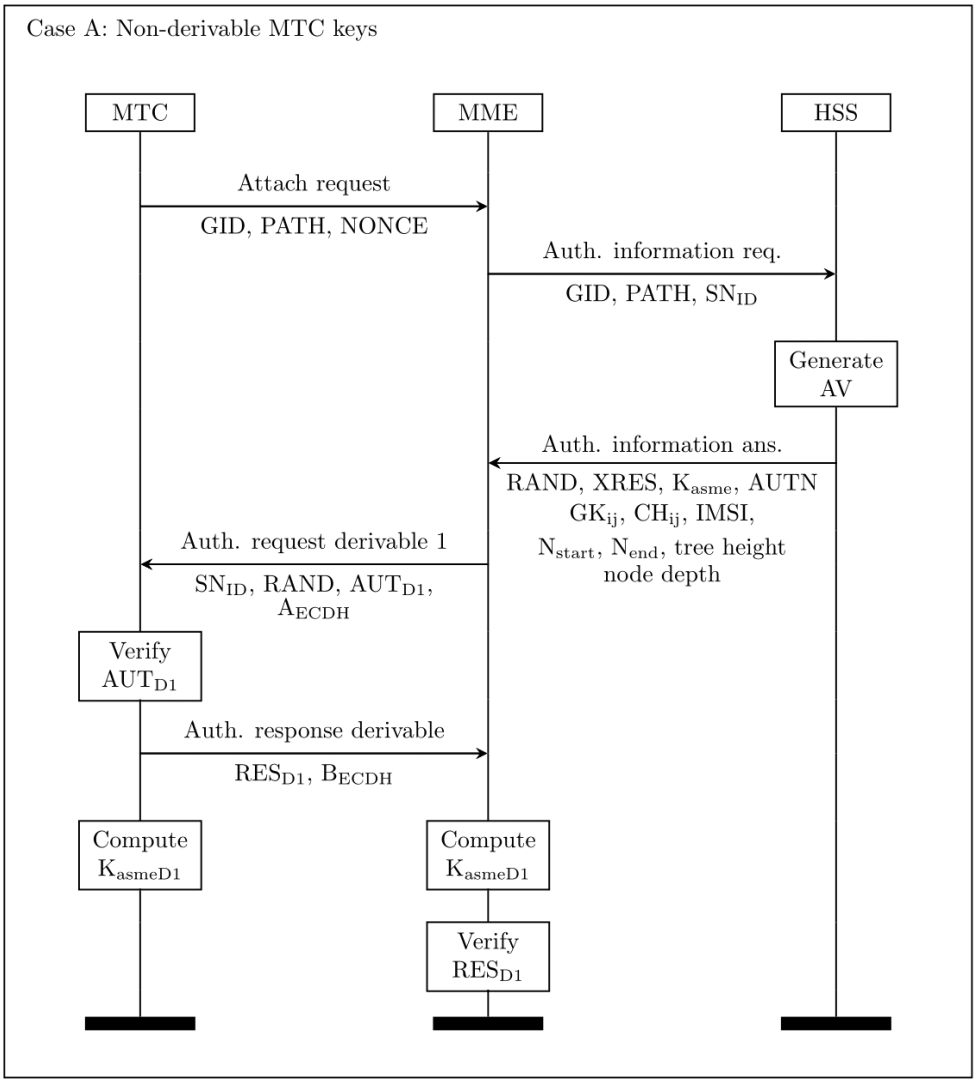


Figure 2-15: The message sequence chart of group-based authentication when the MME cannot derive the keys for the MTC

Case A: Figure 2-15 provides an overview of the initial *Attach Request* from an MTC device for a specific group, thus requiring the MME to retrieve the group elements from the HSS. A Case A message exchange is similar to an EPS-AKA message exchange. The main differences are that additional values are sent from the HSS to the MME and that the communication between the MME and the HSS is modified so that the exchange becomes an ECDH exchange.

The MTC initiates the protocol by issuing the *Attach Request*, containing the additional parameters necessary to specify the group it belongs to, and its position in the inverted hash trees. These parameters are called GID and PATH, respectively. Moreover, a random value named NONCE is sent, it is intended to be used as a challenge by the MTC to the network.

The MME receives the Attach Request message. If the MME is unable to derive keys associated with the MTC device (e.g. the request sent by the first MTC of a specific group), Case A is run and the MME requests authentication vectors from the HSS via an *Authentication Information Request*. The message includes the GID and PATH parameters and the serving network identifier SN_{ID}.

The HSS receives the Authentication Information Request. It verifies that the group id and path are valid and generates the authentication vector, including the new terms intended for the MME to enable group-based authentication for future attach requests. Specifically, the HSS decides sub-root nodes (GK_{ij} and CH_{ij}) of the inverted hash trees and sends two parameters called N_{start} and N_{end} which signify valid sequence numbers that the MME can use for future attach requests. The HSS sends these parameters and other auxiliary parameters to the MME with the *Authentication Information Answer* message.

After receiving the Authentication Information Answer, the MME will continue the message exchange similarly to EPS-AKA. However, the remaining messages also include an authenticated ECDH exchange between the MME and the MTC. An ECDH shared secret is then generated at the MME and at the MTC, which is used to derive the session master key, K_{asmed1}, between them. The K_{asmeD1} is then used as the K_{asme} would be used in the standard AKA procedure.

After Case A has been run, the MME will now be able to derive keys for future authentication requests, based on the additional terms provided in the Authentication Information Answer from the HSS. This scenario is covered by Case B.

Shall an MTC device crash during the protocol run, it can reattempt the authentication procedure using the classic AKA procedure.

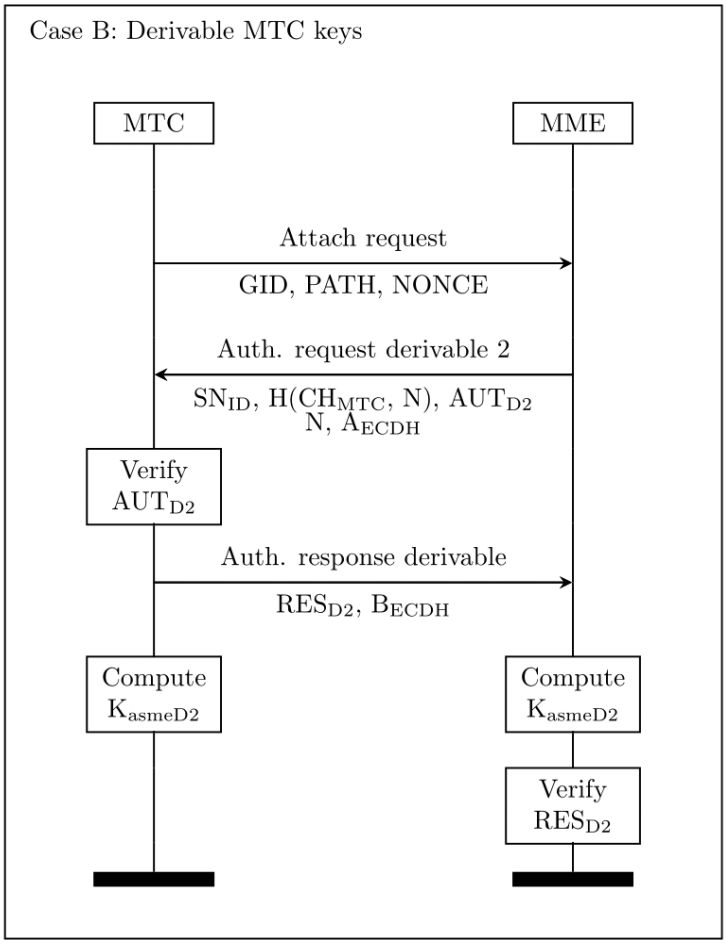


Figure 2-16: **The message sequence chart of group-based authentication when the MME can derive the keys for the MTC**

Case B: Figure 2-16 provides an overview of the succeeding AKA protocol runs for the same group. The MME has stored the necessary terms from the Authentication Information Answer as in Figure 6, and can derive keys for the remaining MTC devices. The protocol is executed solely within the serving network, without the need to interact with the HSS of the group member's home network. Note that Case B, just as Case A, supports an ECDH exchange for session key generation between the MME and the MTC.

Just as in Case A of the proposed protocol, an attach request message is sent by an MTC to the serving network MME.

The MME identifies by the GID parameter that the MTC belongs to the same group as an MTC in a Case A run of the protocol. Next the MME verifies that the PATH parameter can be used to derive leaf nodes from the (sub)root nodes received from the HSS in a Case A run of the protocol. If so, Case B of the protocol continues.

By following the PATH parameter, the MME traverses the inverted hash trees, received from the HSS in Case A, from the (sub)root nodes down to the leaf nodes GK_{MTC} and CH_{MTC}. The MME generates an ECDH public key called A_{ECDH}.

The MME then sends the following to the MTC in the *Authentication Request Derivable 2* message: a valid sequence number N, the SN_{ID}, the $H(CH_{MTC}, N)$ and A_{ECDH} together with a message authentication code (MAC) of NONCE, $H(CH_{MTC}, N)$, GID, SN_{ID} and PATH. The MAC is keyed using $H(GK_{MTC}, N)$ and is contained in the authentication token AUT_{D2}.

After receiving the Authentication Request Derivable 2 message, the MTC de-obfuscates the $O_{(MTC, N)}$ value by using $H(CH_{MTC}, N)$ and its long-term secret K, which produces $H(GK_{MTC}, N)$. Since the MTC now has access to the $H(GK_{MTC}, N)$ value it uses it to verify the received MAC from the MME. The MTC continues by generating an ECDH public key called B_{ECDH}. Moreover, the MTC also generates an ECDH shared secret through the B_{ECDH} and A_{ECDH} parameters.

The MTC then generates a parameter called RES_{D2}, which is a MAC of $H(CH_{MTC}, N)$ and B_{ECDH}, generated with the secret key $H(GK_{MTC}, N)$. The RES_{D2} parameter and B_{ECDH} is sent to the MME in the *Authentication Response Derivable* message.

The MME first verifies the response parameter RES_{D2}. Then it produces the shared ECDH secret by using the B_{ECDH} and A_{ECDH} parameters.

At this stage of the protocol, both the MTC device and the MME can produce a shared session key K_{asmeD2} by using the $H(GK_{MTC}, N)$, the $H(CH_{MTC}, N)$, the sequence number, NONCE and the shared ECDH secret. The K_{asmeD2} is then used as regular K_{asme} would be used in the standard AKA procedure.

2.2.9 Architectural drivers

2.2.9.1 High-Level functional requirements

The main goal of the group-based authentication extension is to make the EPS-AKA protocol more suitable for massive deployment scenarios. This is achieved by removing the existing 1-to-1 correlation between the number of MTC devices and the number of messages in the authentication signaling to the HSS, located in the subscriber's home environment. An arbitrary attribute can define a group.

The group(s) are defined in the HSS, thus in this release the members of a group must be subscribers of the same (virtual) operator, as the HSS must know the individual identity of each MTC device. Additionally, the feature requires provisioning of the group ID, the assigned path of the binary tree, and the obfuscated values given to each MTC. The provisioning procedure is out of scope.

With a configured group in the HSS, only the first authentication request, independent of the actual MTC device that requests it, is sent to the HSS. The HSS will thereafter send the necessary terms to the MME to enable group-based authentication for the succeeding authentication requests.

The group-based authentication feature requires a universal integrated circuit card (UICC or (e)UICC) based authentication in this release to provide a strong identification of the MTC device. To prevent cloning of shared credentials, the feature is built with independent secrets for each MTC device, configured by the HSS. Furthermore, the feature prohibits future, or past, members of a group of being able to decipher data exchanged when they were not part of the specific group. The feature is compliant with lawful interception requirements since all members of a group can be uniquely identified.

In release 2 of the group-based authentication feature PFS is introduced. This property is established by using an authenticated ECDH exchange between the MTC and the MME, the resulting secret of which is used for session key generation. Therefore, it is required that the MTC and the MME must be able to perform elliptic curve functions to enable the ECDH exchange.

2.2.9.2 Link to Security Architecture

Non-USIM based authentication

The Non-USIM based authentication feature relates to the following domain and strata as shown in Figure X and **Figure 2-17Error! Reference source not found.** (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016

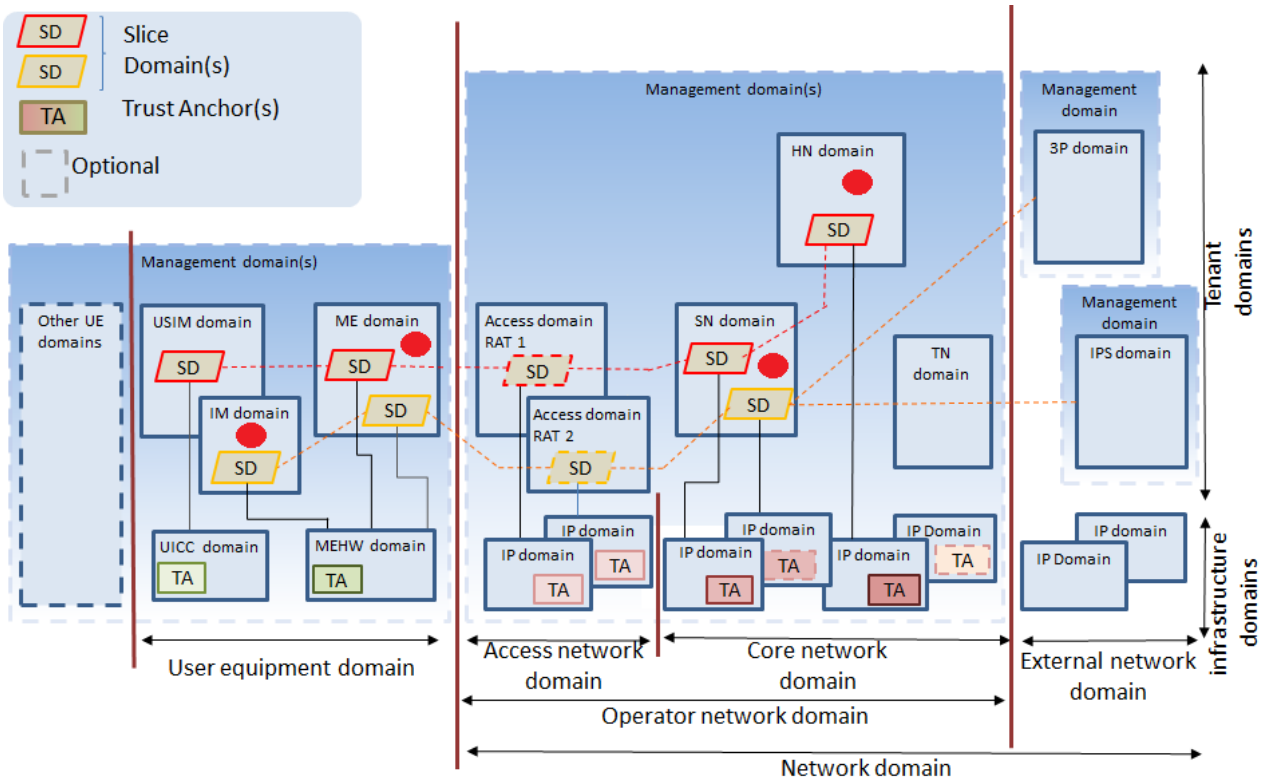


Figure 2-17 Enabler feature relevant domains

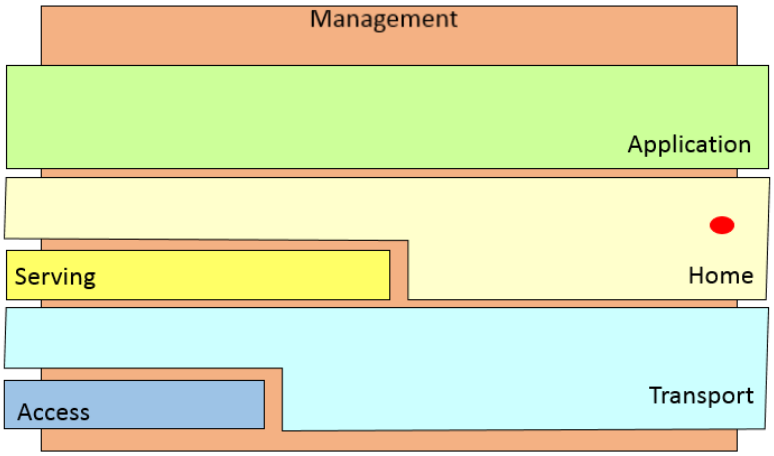


Figure 2-18

vGBA

vGBA enabler relates to the following security objectives.

O6.1 5G security protocols must scale to support massive IoT and must not have negative security impact on non-IoT services

O6.2 5G security must support also protocols efficient enough for resource constrained devices

Figure 2-19 and **Figure 2-20** show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016].

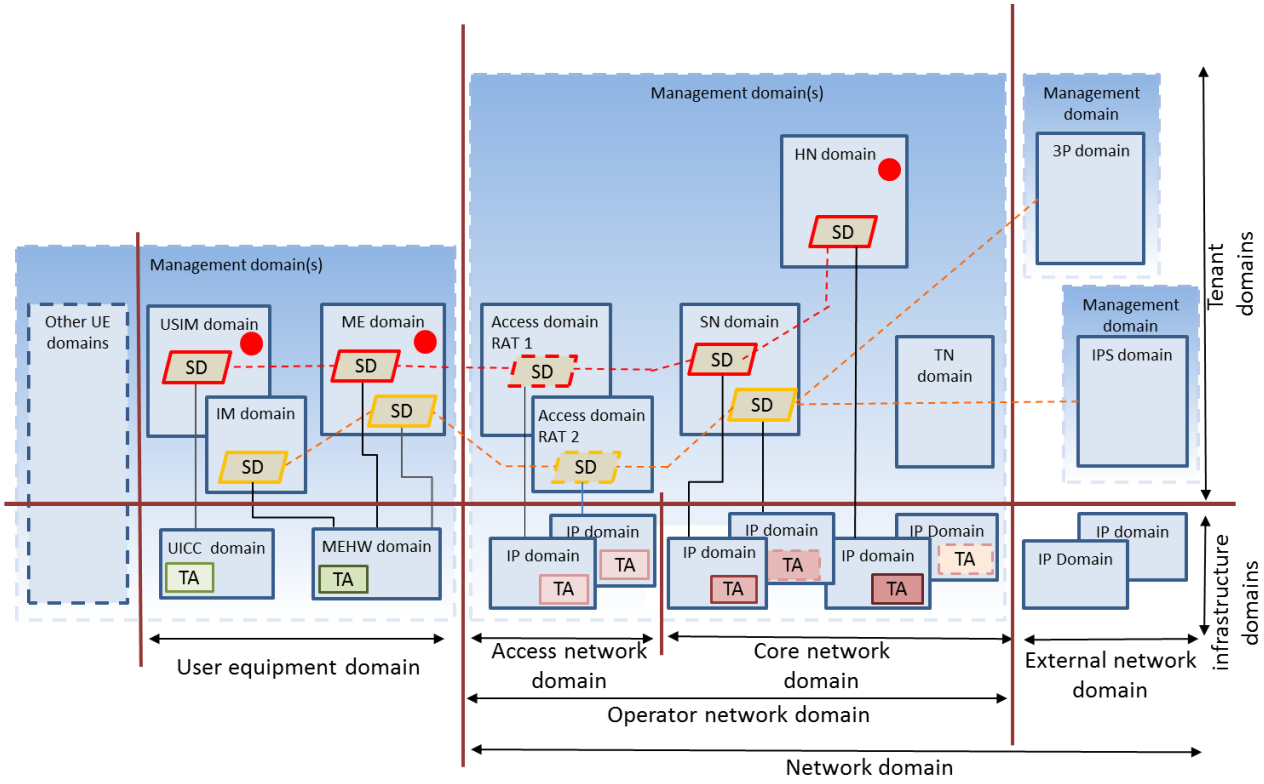


Figure 2-19: Enabler relevant domains

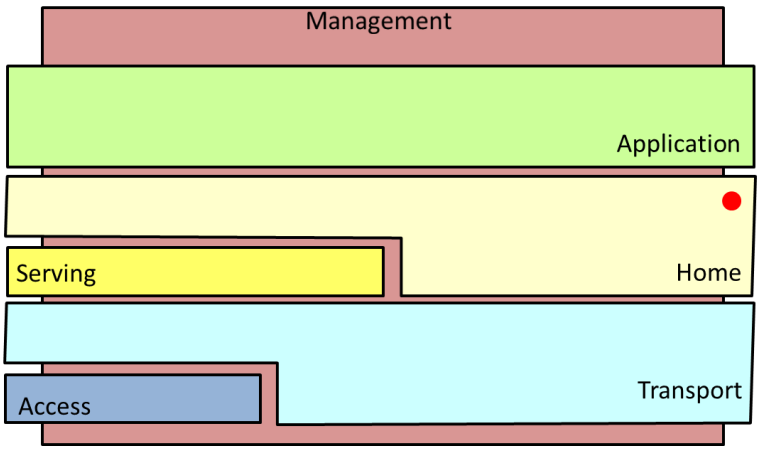


Figure 2-20: Enabler relevant strata

Bring Your Own Identity

The BYOI feature relates to the following security objectives:

O4.2 - 5G security must be dynamically scalable in order to easily and securely enable the changes required to ensure any new use cases, new trust models and new service delivery approaches.

O4.3 - 5G infrastructure components should support necessary root-of-trust functionality.

O6.1 - 5G security protocols must scale to support massive IoT and must not have negative security impact on non-IoT services.

O6.3 - 5G connected devices must be able to adequately protect critical data such as subscriber credentials against threats of unauthorized access and/or modification.

O7.2 - Management and storage of alternative credentials by an external vertical AAA must be done with a security level commensurate with both that of the vertical application as well as the operator business partner.

O7.4 - 5G systems and components must provide strong mutual authentication and authorization.

Figure 2-21 and **Figure 2-22** show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016]

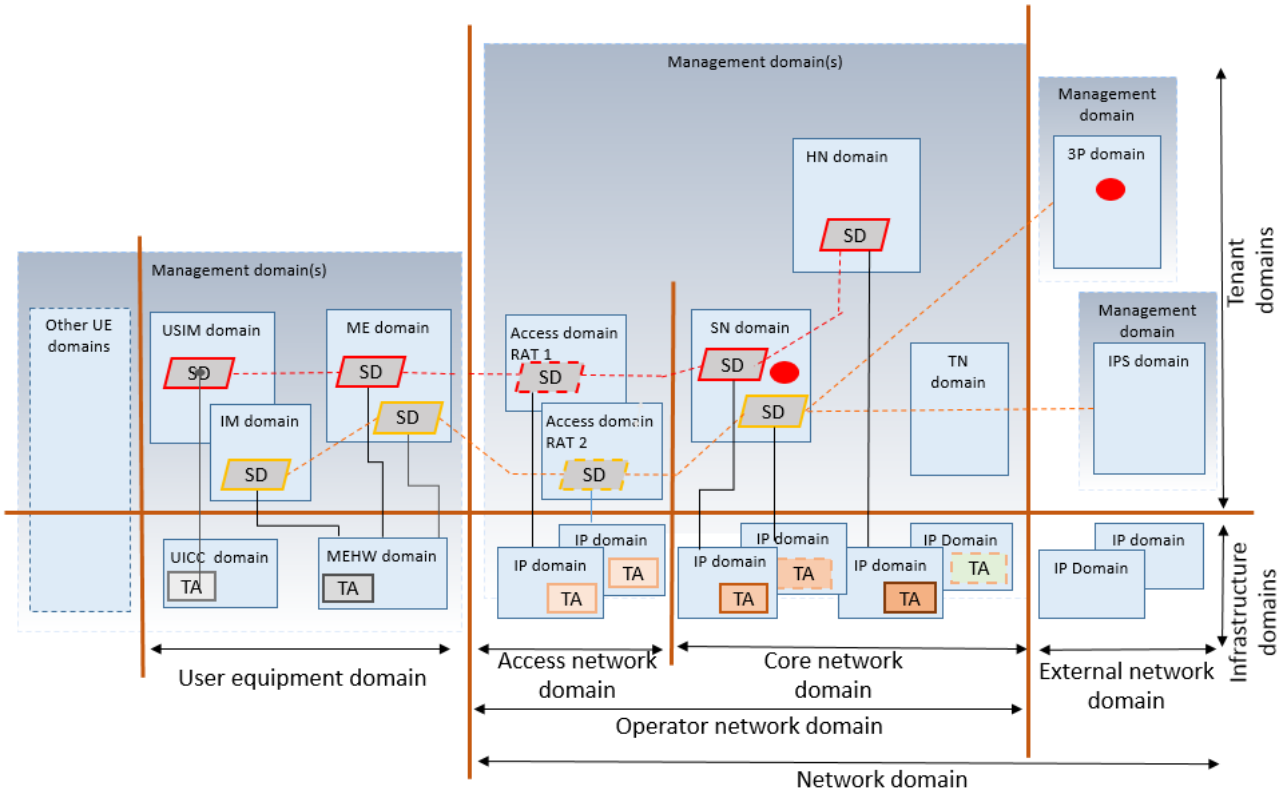


Figure 2-21 Enabler feature relevant domains

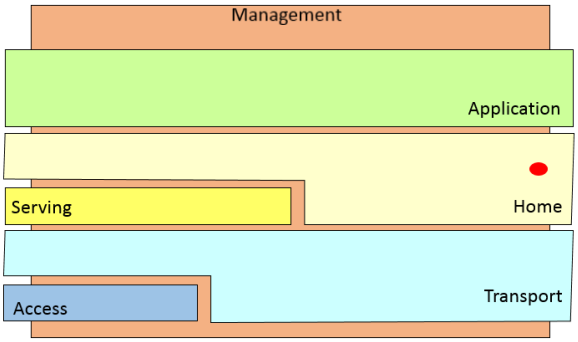


Figure 2-22 Strata of the 5G Security Architecture

Group-based authentication

Figure 2-23 and Figure 2-24 show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016] for the Group-based authentication feature.

Domains: USIM Domain, ME Domain, SN Domain, HN Domain,
Strata: Home Stratum

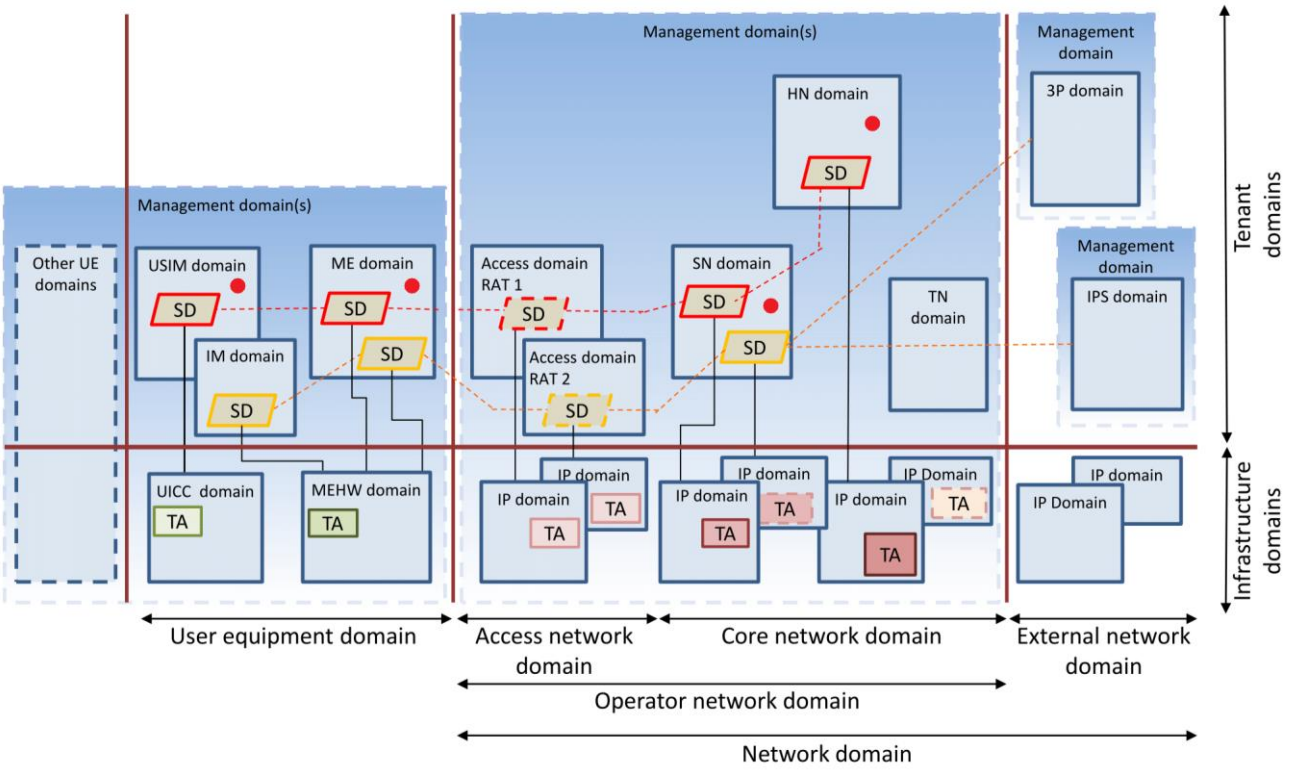


Figure 2-23: Enabler relevant domains

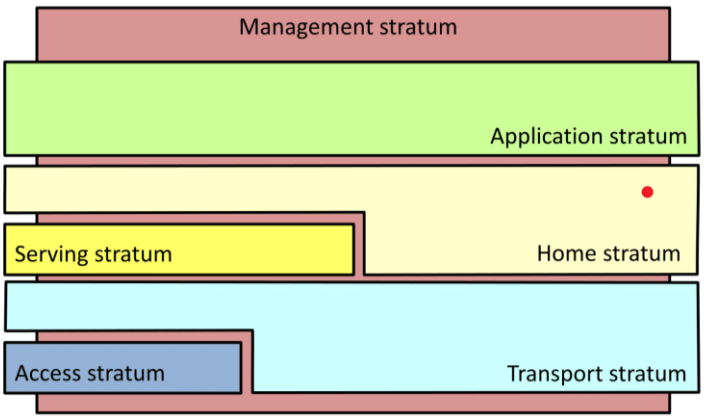


Figure 2-24: **Enabler relevant strata**

2.2.9.3 Quality attributes

A product implementing one or several of the features described in enabler should be evaluated primarily based on the following quality attributes:

Security:

- Confidentiality
- Integrity
- Non-repudiation
- Accountability
- Authenticity

Compatibility:

- Co-existence
- Interoperability

2.2.9.4 Technical constraints

Since the group is configured in the HSS, all members in a specific group must have a subscription from the same HSS. Furthermore, the key provisioning for each MTC device that should be part of a group is out of scope for this release, hence it must be preconfigured manually at this stage of research. Additionally, the API access for configuring the group(s) to the core network is out of scope.

2.2.9.5 Business constraints

No known business constraint exists.

2.2.10 Detailed specifications

2.2.10.1 Introduction

In this section, key concepts regarding the vertical GBA and the group-based authentication solutions are introduced.

vGBA

The proposed vGBA solution adds some new functionality during the AKA exchange. The changes are in the HSS, BSF and the USIM. The changes do not affect the AKA protocol exchange. However, some new logic is needed in the identified entities and a new interface between HSS and BSF is introduced.

The first change to a standard AKA exchange happens at the HSS when an authentication information requested is received from the MME. At this point the HSS redirects the request to the BSF over a new interface. The BSF requests an authentication vector (AV) based on the IMSI in the received authentication request using the standard GBA Zh interface, and receives the authentication vector and the GUSS of the subscription. Using CK/IK in the AV, and the serving network identifier found in the initial request, the BSF generates CK'/IK'. CK'/IK' are to be used by the MME for performing access authentication with the UE, while the original CK/IK are used for generating the GBA master key Ks. The reason for this is to keep the access keys separate from the application keys, especially in scenarios where the access provider and the application service provider are different entities. BSF also generates the B-TID using the RAND received in

the AV. BSF then responds to the initial authentication request with the AV, where CK/IK have been replaced by CK'/IK'. The MME will not be aware of this change.

Finally, once the USIM receives the authentication challenge, it resolves it and computes CK/IK. In addition, it computes B-TID from the received RAND, the GBA master key Ks from CK/IK and CK'/IK' in the same way as the BSF did. CK'/IK' is provided to the radio, while B-TID and Ks are provided to the GBA client. Alternatively, the USIM can provide the CK/IK to the ME, and the ME performs the further key derivations from CK/IK.

Group-based authentication

The proposed group-based extension to the AKA protocol uses mostly the same cryptographic primitives that already exist in EPS-AKA, e.g. hash and MAC functions, with the exception of new functions used for ECDH exchange support. New terms are introduced in the protocol to support this feature, see Table 2-3.

In the message exchange between the MTC device and MME, the initial message of the protocol is the *Attach Request*. Depending on whether the MME has earlier authenticated an MTC of the same group as the *Attach Request* or not, the second run of the protocol may differ. In Case A (see Figure 6), the MME contacts the HSS to retrieve the necessary information. The response from the HSS, namely the authentication vector, is returned to the MME as specified in EPS-AKA, however with additional elements for group-based authentication support. The new elements are stored in the MME.

In Case B, the MME does not need to contact the HSS to authenticate the MTC.

Term	**Description**
GID	Group identifier
PATH	The path assigned to the MTC. Each MTC is assigned with the same path in both trees.
NONCE	Random number
GKij	The key associated to the value of the node at i^{th} position and j^{th} level of the inverted hash tree GK.
CHij	The key associated to the value of the node at i^{th} position and j^{th} level of the inverted hash tree CH.
GK_{MTC}	The key associated to the MTC. It is the hash value of the leaf of the GK tree at PATH.
CH_{MTC}	The challenge key associated the MTC. It is the hash value of the leaf of the CH tree at PATH.
$O_{(MTC, N)}$	The obfuscated value that hides the key associated to the MTC
NODE DEPTH	Parameter sent to the MME from the HSS to help determine the inverted hash tree node depth of sub roots
TREE HEIGHT	Parameter sent to the MME from the HSS to help determine

Term	Description
	the inverted hash tree height
AUT_{D1}	The authentication parameter in the group authentication used in Case A
AUT_{D2}	The authentication parameter in the group authentication used in Case B
RES_{D1}	The response parameter in the group authentication used in Case A
RES_{D2}	The response parameter in the group authentication used in Case B
K_{asmeD1}	The session key generated in the group authentication used in Case A
K_{asmeD2}	The session key generated in the group authentication used in Case B
A_{ECDH}	ECDH public key sent by the MME
B_{ECDH}	ECDH public key sent by the MTC
N	A sequence number
N_{start}	A parameter indicating to the MME the start of the range of valid sequence numbers
N_{end}	A parameter indicating to the MME the end of the range of valid sequence numbers

Table 2-3: **New terms introduced by the group-based authentication protocol**

The following table (Table 2-3) specifies how the central parameters of the protocol scheme are generated. Three types of function are used: message authentication codes, key derivation functions and hash functions. Message authentication codes are denoted by f_MAC, key derivation functions by f_KDF and hash functions by H. The functions f1 and f5 are reused from EPS-AKA and the requirements for those algorithms are stated in [5].

Term	Generated by
$O_{(MTC, N)}$	H(K, H(GKmtc, N)) $\oplus$ H(CHmtc, N)
AUT_{D1}	SQN$\oplus$AK ||AMF || f1(K, SQN, RAND, AMF) || f_MAC(K_{ASME}, A_{ECDH})

AUT_{D2}	f5(H(GK_{MTC}, N),NONCE) || f_MAC(H(GK_{MTC}, N) , NONCE, H(CH_{MTC} , N), GID, SN_{ID}, PATH, A_{ECDH}).
RES_{D1}	f_MAC(RES, B_{ECDH})
RES_{D2}	f_MAC(H(GK_{MTC},N), H(CH_{MTC} , N), B_{ECDH})
K_{asmeD1}	f_KDF($K_{asme,}$ Shared ECDH secret)
K_{asmeD2}	f_KDF(H(GK_{MTC},N), NONCE, H(CH_{mtc}, N), SNid, Shared ECDH secret)

Table 2-4: Generation of authentication and key agreement parameters

2.2.10.2 Conformance

An implementation to be reported as conformant should comply with the open specifications here stated for the enabler.

2.2.10.3 API specifications

Non-USIM based AKA

The Mobile Station Classmark 3 information element as described in TS24.008 Section 10.5.1.7 shall be extended with information on EAP support, the suggested new information encoding is:

EAP support:
Bits

4 3 2 1	
0 0 0 0	EAP not supported
- - - 1	EAP-TLS supported
- - 1 -	EAP-TTLS supported
-1 - -	EAP-AKA’
1 - - -	TBD

We suggest that EAP-AKA is simplified and implemented as EPS-AKA, support of EAP-AKA is hence implicitly supported for LTE capable devices.

The SECURITY MODE COMPLETE message as defined in TS24.301 Section 8.2.21 shall be updated to be able to carry the EAP-TLS Success/Reject message according to **Table 2-5**

Table 2-5 SECURITY MODE COMPLETE message content

IEI	Information Element	Type/Reference	Presence	Format	Length
	Protocol discriminator	Protocol discriminator 9.2	M	V	1/2
	Security header type	Security header type 9.3.1	M	V	1/2
	Security mode complete message identity	Message type 9.8	M	V	1
23	IMEISV	Mobile identity 9.9.2.3	O	TLV	11
NEW	EAP Success/Reject		O		4

Table 2-6

$vGBA$

The vGBA solution will introduce functionality adjustments, modifications as well as new content in the EPS that will need specification. A high-level description of the new aspects and elements introduced in the main entities which are affected will be presented.

New functions, messages and commands

In this section, the new functionality for vGBA is introduced in the UE, BSF and HSS respectively.

UE

The USIM application is responsible for interaction between the mobile equipment (ME) and the Universal Integrated Circuit Card (UICC). The ME and USIM combined is denoted user equipment (UE). The UICC is responsible for storing security critical data.

When the USIM gets the AKA authentication challenge it responds to it in a standard way, but in addition, calculates the new parameters CK'/IK', B-TID and Ks. CK'/IK' are generated using a KDF that takes as input the keys CK/IK and the serving network ID. B-TID is generated from the RAND parameter and Ks from CK/IK per GBA specification. The response to the AKA challenge is calculated using CK'/IK' instead of CK/IK, as the latter is reserved for GBA use and the former are the ones that have been provided to the MME by the BSF.

Typically, the GBA client requests B-TID and Ks from the USIM, while in vGBA they are pushed to the GBA client, optionally with a preconfigured lifetime found on the UICC.

In the variant where the ME takes a bigger role, all further key derivations from CK/IK are performed in the ME instead of in the USIM.

Home Subscriber Server (HSS)

vGBA requires the HSS to maintain information about vGBA support for the subscriptions. When an authentication information request is received for a subscription that is marked as being vGBA enabled, the HSS needs to redirect the request to the BSF

Bootstrapping Server Function (BSF)

In vGBA, the BSF needs to support the new interface between itself and the HSS over which the authentication information request is forwarded. This is a different message type and interface than what

standard BSF supports. The BSF parses the IMSI and serving network identifier from the message, and uses the standard Zh interface for requesting an AV (and GUSS) from the HSS using the IMSI. The BSF further has to use the KDF for generating CK'/IK' from CK/IK and the serving network identifier. The response to the request over the new interface is naturally also different from standard GBA as the BSF in essence answers an authentication information request rather than a GBA bootstrapping request. The BSF needs to replace the CK/IK from the AV received from the HSS with CK'/IK' it has generated.

Group based authentication

The NAS protocols need to support the new parameters and message types used in group based AKA as described below. (Original values can be seen in 3GPP, "Universal Mobile Telecommunications System (UMTS); LTE; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3," TS 24.301 V10.5.0, January 2012)

IEI	Information Element	Type/Reference	Presence	Format	Length

IEI	Information Element	Type/Reference	Presence	Format	Length
	Protocol discriminator	Protocol discriminator 9.2	M	V	1/2
	Security header type	Security header type 9.3.1	M	V	1/2
	Attach request message identity	Message type 9.8	M	V	1
	EPS attach type	EPS attach type 9.9.3.11	M	V	1/2
	NAS key set identifier	NAS key set identifier 9.9.3.21	M	V	1/2
	EPS mobile identity	EPS mobile identity 9.9.3.12	M	LV	5-12
	UE network capability	UE network capability 9.9.3.34	M	LV	3-14
	ESM message container	ESM message container 9.9.3.15	M	LV-E	5-n
19	Old P-TMSI signature	P-TMSI signature 10.5.5.8	O	TV	4
50	Additional GUTI	EPS mobile identity 9.9.3.12	O	TLV	13
52	Last visited registered TAI	Tracking area identity 9.9.3.32	O	TV	6
5C	DRX parameter	DRX parameter 9.9.3.8	O	TV	3
31	MS network capability	MS network capability 9.9.3.20	O	TLV	4-10
13	Old location area identification	Location area identification 9.9.2.2	O	TV	6
9-	TMSI status	TMSI status 9.9.3.31	O	TV	1
11	Mobile station classmark 2	Mobile station classmark 2 9.9.2.4	O	TLV	5
20	Mobile station classmark 3	Mobile station classmark 3 9.9.2.5	O	TLV	2-34
40	Supported Codecs	Supported Codec List 9.9.2.10	O	TLV	5-n
F-	Additional update type	Additional update type 9.9.3.0B	O	TV	1
5D	Voice domain preference and UE's usage setting	Voice domain preference and UE's usage setting 9.9.3.44	O	TLV	3
D-	Device properties	Device properties 9.9.2.0A	O	TV	1
E-	Old GUTI type	GUTI type 9.9.3.45	O	TV	1
C-	MS network feature support	MS network feature support 9.9.3.20A	O	TV	1
NEW	**PATH**	**PATH**	**C**	**LV**	**2-33**
NEW	**NONCE**	**NONCE**	**C**	**V**	**16**

Table 2-7: ATTACH REQUEST message content

IEI	Information element	Type/Reference	Presence	Format	Length
	Protocol discriminator	Protocol discriminator 9.2	M	V	1/2
	Security header type	Security header type 9.3.1	M	V	1/2
	Authentication request derivable1 message type	Message type	M	V	1
	NAS key set identifier$_{ASME}$	NAS key set identifier 9.9.3.21	M	V	1/2
	Spare half octet	Spare half octet 9.9.2.9	M	V	1/2
	Authentication parameter RAND (EPS challenge)	Authentication parameter RAND 9.9.3.3	M	V	16
NEW	**ECDH public key**	**ECDH public key**	**M**	**V**	**32**
NEW	**AUT$_{D1}$**	**AUT$_{D1}$**	**M**	**V**	**24**

Table 2-8: AUTHENTICATION REQUEST DERIVABLE1 message content

IEI	Information element	Type/Reference	Presence	Format	Length
	Protocol discriminator	Protocol discriminator 9.2	M	V	1/2
	Security header type	Security header type 9.3.1	M	V	1/2
	Authentication request derivable2 message type	Message type	M	V	1
	NAS key set identifier$_{ASME}$	NAS key set identifier 9.9.3.21	M	V	1/2
	Spare half octet	Spare half octet 9.9.2.9	M	V	1/2
NEW	**ECDH public key**	**ECDH public key**	M	**V**	**32**
NEW	**CH$_{MTC}$**	**CH$_{MTC}$**	**M**	**V**	**16**
NEW	**AUT$_{D2}$**	**AUT$_{D2}$**	**M**	**V**	**14**

Table 2-9: AUTHENTICATION REQUEST DERIVABLE2 message content

IEI	Information element	Type/Reference	Presence	Format	Length
	Protocol discriminator	Protocol discriminator 9.2	M	V	1/2
	Security header type	Security header type 9.3.1	M	V	1/2
	Authentication response derivable message type	Message type	M	V	1
	Authentication response parameter	Authentication response parameter 9.9.3.4	M	LV	5-17
NEW	**ECDH public key**	**ECDH public key**	**M**	**V**	**32**

Table 2-10: AUTHENTICATION RESPONSE DERIVABLE message content

The S6a interface needs to support six new elements and new vector including the group based AKA parameters. This new vector is then appended to the Authentication-Info AVP described in [6]. The new Group-authentication-vector is as follows.

```
Group-auth-vector ::= <AVP header: >
                      [ Item-Number ]
                      { NODE DEPTH }
                      { TREE HEIGHT }
                      { GKij}
                      { CHij}
                      {User-Name}
                      {Nstart}
                      {Nend}
                      *[AVP]
```

An addition is also present in Requested- EUTRAN-Authentication-Info AVP with the addition of the PATH parameter, seen below. The GID parameter is sent using the already present User-Name AVP in the Authentication-Information-Request AVP.

```
Requested- EUTRAN-Authentication-Info ::= <AVP header: 1408 10415>
                                          [ Number-Of-Requested-Vectors]
                                          [Immediate-Response-Preferred ]
                                          [ Re-synchronization-Info ]
                                          [ PATH AVP]
                                          *[AVP]
```

2.2.11 Re-utilised Technologies/Specifications

The software prototype for group-based authentication is based upon Open Air Interface.

2.2.12 References

[1] Giustolisi, Rosario, et al. "A Secure Group-Based AKA Protocol for Machine-Type Communications." *International Conference on Information Security and Cryptology*. Springer, Cham, 2016.

[2] Page, T, “The application of hash chains and hash structures to cryptography,” Technical report Royal Holloway, University of London, 2009.

[3] Encyclopedia of Cryptography and Security. Springer US, 2011. 921-922].

[4] 5G-Ensure Consortium, *Deliverable 2.1 Use Cases,* [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-

protocol,” TS 29.272 V10.9.0, July 2014.

[5] 3GPP, “Universal Mobile Telecommunications System (UMTS); LTE; Cryptographic algorithm requirements,” TS 33.105 V10.0.0, April 2011

[6] 3GPP, “Universal Mobile Telecommunications System (UMTS); LTE; Evolved Packet System (EPS); Mobility Management Entity (MME)and Serving GPRS Support Node (SGSN) related interfaces based on Diameter

2.3 Fine-grained authorization: Open specifications

2.3.1 Preface

This section describes the open specification of the Fine-grained authorization enabler focusing on authorization for two areas, which are expected to be strongly involved in 5G.

First, in addition to the authentication focus brought by the IoT enabler, this enabler will research new methods to provide distributed authorization, suitable in resource-constrained environments. The goal of the enabler is to make 5G fully ready for Identity and Access Management (IAM) of IoT devices. We also plan to follow the ongoing IETF standardisation of security protocols for RCDs conducted at the ACE working group[4].

The second area is based on requirements from 5G satellite business needs and 5G-ENSURE use cases. The goal is to provide an integrated satellite and terrestrial approach, compared to the diverse methods existing today, to provide secure fine-grained access control to satellite resources (i.e. network element and services). For R2, it will provide better access control for the satellite resources using network information such as location, time range and operator role. This information improves the security of the access to those resources. In addition to access control, the communication changes from client-server to client-hub-server, introducing this satellite component among them.

Please, consult as well the "5G-ENSURE_D3.5 5G-PPP security enablers technical roadmap (Update)" [1]document and the website on http://www.5gensure.eu/ in order to understand the complete context of the 5G-ENSURE project.

2.3.2 Copyright

2.3.3 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

2.3.4 Terms and definitions

This section comprises a summary of terms and definitions used during the later sections.

- **ABAC (attribute-based access control):** an **access control** paradigm whereby access rights are granted according to a combination of attributes of any type (**user**, **resource**, **environment**, etc.).
- **Access:** Performing an **action**.
- **Access control:** Controlling access in accordance with a **policy** or **policy set**.

4 https://datatracker.ietf.org/group/ace

- Access token: OAuth2 term, digital object specifying or referring to an access control decision.
- **Action:** An operation on a **resource**.
- **AS:** OAuth2 term for Authorization server. Provides access tokens and token introspection.
- **Attribute:** Characteristic of a **subject**, **resource**, **action** or **environment** that may be referenced in a **predicate** or **target**.
- **Environment:** The set of **attributes** that are relevant to an **authorization decision** and are independent of a particular **subject**, **resource** or **action**.
- **PAP (policy administration point):** The system entity that creates a policy or policy set.
- **PDP (policy decision point):** component which computes **access decisions** by evaluating the applicable **access control policies**; one of the main functions of the **PDP** is to mediate or deconflict **policies**.
- **PEP (policy enforcement point):** component which enforces **policy decisions** in response to a request from a **subject** requesting **access** to a protected **resource**; the **access control** decisions are made by the **PDP**.
- **PIP (policy information point):** the system entity that acts as a source of **attribute** values (e.g. LDAP).
- **Policy:** A set of **rules**, an identifier for the **rule-combining algorithm** and (optionally) a set of **obligations** or **advice**. May be a component of a **policy set**.
- **RBAC (role-based access control):** an **access control** paradigm whereby access rights are granted according to roles and privileges.
- **RCD (resource-constrained device):** a device which has very few resources to run.
- **Resource:** Data, service or system component.
- **RS:** Resource Server, OAuth2 term for a server providing access to **resources.**
- **Subject:** An actor whose **attributes** may be referenced by a **predicate**.

For a summary of terms and definitions managed, please refer to "INCITS 359-2012" standard **Error! Reference source not found.**, the XACML v3.0 standard **Error! Reference source not found.** section 1.1.1 and the OAuth2 specification [3].

2.3.5 Overview

The enabler provides an AAA solution suitable for an environment of resource-constrained devices and satellite resources, relying on ABAC and/or RBAC policies. It consists of:

- a AAA service, based on 5G AAA credentials, able to provide a secured and standalone proof of the rights granted to a given client over a given device or satellite resource;
- a module for resource-constrained devices, able to ensure the validity and the enforcement of the proof towards a performed request, without requiring more external communication;
- a module for satellite resources, able to ensure the validity and the enforcement of the proof towards a performed request.

We assume that a trust relationship has been pre-established between the owner/provider of the enabler and the devices. The enabler scope does not include the security of the communication channels of the different components. The access control of the enabler administration API is also out of the enabler open specification scope.

In R2 it is expected to support policies for decision per user, resource and action, and access control based on dynamically changing parameters.

Also, in the satellite context, the R2 release is expected to integrate the authentication and authorization mechanism with the Satellite system. During R2 development, this integration will be done using a Hub to do the communication between the client and server.

Apart from the new communication way, the authorization process is improved using new information content broadcasted through the network, which will give action access to the right user based on policies. For example, an update of a Satellite Terminal only could be done by user with "SNO" rights, during the working hours (8:00-17:00) and from the setup location (India). The access to this task (update ST) from other user, location or outside that hours will be denied.

Finally, in the resource-constrained context, the R2 release is expected to provide the final version of PEP and PDP embedded on the RCD, and the Authentication server delivering a self-sufficient security token allowing decentralized authentication and authorization, compatible with RCDs in terms of performance.

Furthermore, SICS is working on a solution mirroring the RCD security work at the IETF, that makes use of a profile of OAuth2 **Error! Reference source not found.** to provide access control at the RCD without the need for an embedded PDP (i.e. the RCD just needs to implement the PEP component).

2.3.6 Basic concepts

2.3.6.1 OAuth2 concepts

OAuth2 is an authorization protocol which enables a third-party application to obtain limited access to a resource over HTTP, through a prior authorization from the resource owner. OAuth2 defines 4 distinct roles **Error! Reference source not found.**:

- **Resource owner**: an entity capable of granting access to a protected resource. In the enabler context, it refers to a RCD owner who can define its access rights.
- **Resource server**: the server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens. In the enabler context, it refers to the RCD itself, with the access token enforcement module.
- **Client**: an application making protected resource requests. In the enabler context, it refers to the user who performs a request toward a protected RCD.
- **Authorization server**: the server issuing authorization to the client to access protected resources. In the enabler context, it refers to the enabler AAA service.

In OAuth2, the authorization server grants rights by delivering access tokens. Access tokens are credentials, representing specific scopes of access rights and duration of access, understood by the resource server. They can have different formats, structures, methods of utilization (e.g. cryptographic properties) according to resource server requirements. The token scope is a parameter which is used to limit token access rights.

OAuth2 also specifies a few different usage flows. An applicable one in the enabler context is the *Client Credentials Grant* flow, illustrated in Figure 2-25. The client can simply request an access token using its credentials when requesting access to the protected resource. This flow is suitable for machine-to-machine communication, where manual intervention of the resource owner is not feasible. In such cases the client is already authorized to access the resource by other means, which is applicable in the enabler context through the access control policies at the Authorization Server.

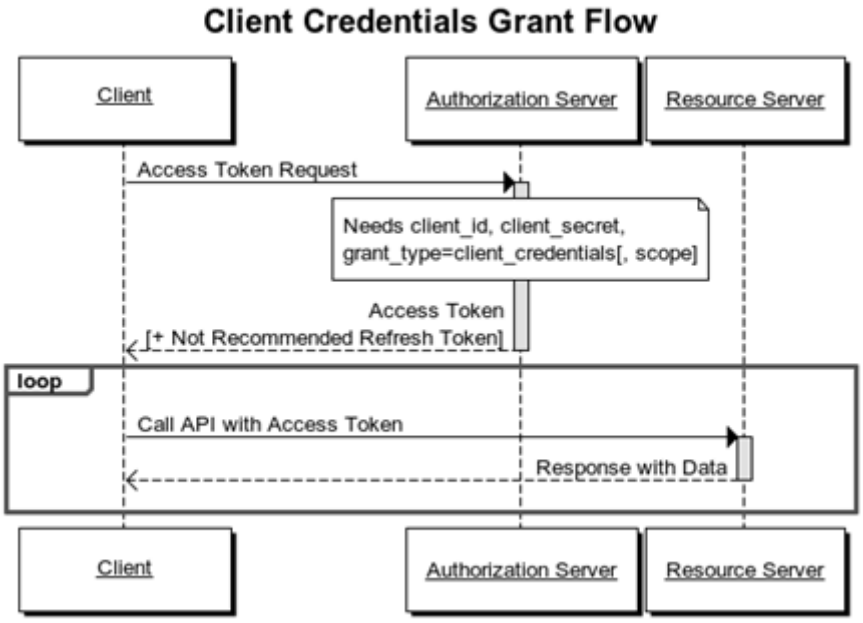


Figure 2-25: OAuth2 Client Credentials Grant flow

Another applicable flow is the *Authorization Code Grant* flow, illustrated in Figure 2-26. This flow is suitable in cases where a resource owner can explicitly grant a client an access to its protected resources. The client must get an authorization code in order to access the resource that it requests from the authorization server. The server responds with a redirection URI that the resource owner must reach to log in and then explicitly grant the requested access. The client can then exchange this authorization code against an access token usable with the resource server.

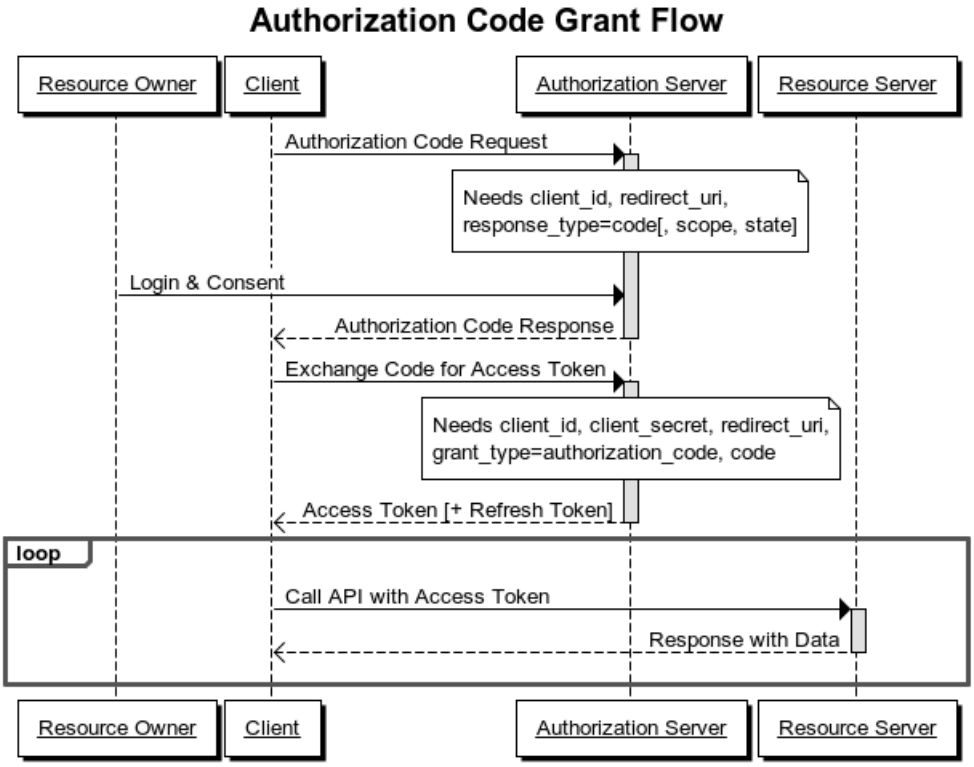


Figure 2-26: **OAuth2 Authorization Code Grant flow**

OpenID Connect [10] is an authentication standard based on OAuth2, specifying how to perform authentication through OAuth2, using Authorization Code Grant flow. In this context, the client is a service requiring a user identity; the protected resource is the user identity, obviously owned by the user, which will authenticate to the authorization server in order to provide the client a proof of his identity.

2.3.6.2 IETF standardisation on RCD security

The IETF working group ACE (Authentication and Authorization in Constrained Environments) is currently working on adapting web-security protocols to RCDs. ACE has chosen to produce a profile of the OAuth2 protocol adapted for RCDs **Error! Reference source not found.**. The two main requirements these adaptations address are the reduction of network communication (both number and size of messages), and the fact that either client or resource server (RS) may have intermittent connectivity with back-end services like the Authorization Server (AS).

To this end the following modifications to the plain OAuth protocol are defined:

- The AS provides proof-of-possession keys linked to the access token to both client and RS in case of intermittent connectivity for one party, the other is used an intermediary).

- The AS specifies which communication security profile client and RS must use.
- The RS provides a resource /authz-info that the client can send its access token to.
- All messages are encoded in CBOR **Error! Reference source not found.** instead of JSON, in order to reduce message sizes.
- COSE **Error! Reference source not found.** is used instead of JWS and JWE to protect access tokens
- CBOR Web Tokens (CWT) **Error! Reference source not found.** are used instead of JSON Web Tokens (JWT)

2.3.6.3 RBAC concepts

RBAC is an access control method for controlling user access applying roles and permissions. Roles are created for various operations. The permissions to perform certain operations are assigned to specific roles. Subjects are assigned particular roles, and through those role assignments acquire the permissions to perform particular operations.

Three primary rules are defined for RBAC:

- Role assignment: A subject can exercise permission only if the subject has selected or been assigned a role.
- Role authorization: A subject's active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.
- Permission authorization: A subject can exercise permission only if the permission is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can exercise only permissions for which they are authorized.

The use of RBAC can be problematic due to the tendency of role-explosion. This refers to the fact that over time, roles tend to accumulate in order to cater for special access needs. In the end the number of roles converges with the number of users, times the number of resources, times the number of permissions. Therefore, the access control becomes hard to manage and it becomes difficult to check who has which permissions.

2.3.6.4 ABAC concepts

ABAC**Error! Reference source not found.** is an access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions.

When an access request is made, attributes and access control rules are evaluated by the ABAC mechanism to provide an access control decision. In its basic form, the ABAC mechanism illustrated in **Error! Reference source not found.** contains both a Policy Decision Point and a Policy Enforcement Point.

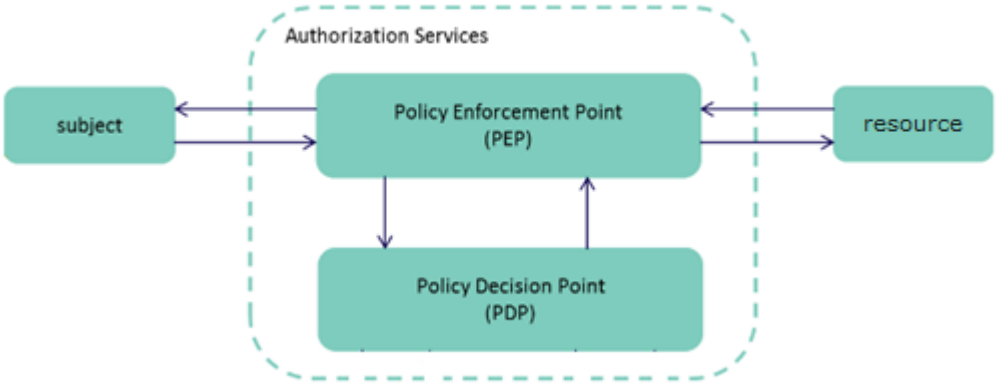


Figure 2-27

The PEP takes place in front of the protected object in order to filter subject request attempts, in accordance with the access control policy. For each incoming subject request, the PEP builds an authorization request toward the PDP, assigning different attributes according to information extractable from the subject request. The PDP is charged to evaluate the access control policy with assigned attributes and decide whether the subject request is conforming or not.

However, in the resource-constrained context, the objective is to make PEP and PDP independent and break their direct communication link. To do this, the PDP should perform a partial evaluation of the policy based on the authenticated subject information, then return a remaining attributes combination, usable as an access token scope, in order to let the PEP know about the subject access rights.

In the satellite context, the ABAC core mechanism is extended with two entities:

- The Policy Administration Point (PAP), which manage the access control policies. Basically, it is an interface to write/edit/delete policies and makes them available to the PDP.
- The Policy Information Point (PIP), which acts as a source of attribute values using LDAP. Basically, if there are missing attributes in the request which is sent by PEP, PIP would find them for the PDP to evaluate the policy.

2.3.6.5 XACML concepts

XACML (eXtensible Access Control Markup Language) is an OASIS standard **Error! Reference source not found.** specification of an ABAC language based on XML.

We can summarize it in three parts:

- **Policy language**: XACML defines a XML data model for defining authorization policies, as well as the logic to follow to evaluate them in a given access request context. *Rule*, *Policy* (set of *Rules*), and *PolicySet* (set of *Policy* elements) constitute the main elements of the model. In short, a rule consists of a condition on the access request attributes, and a decision – *Permit* or *Deny* - to apply if the condition holds true for the request. A *Policy* (resp. *PolicySet*) combines multiple *Rules* (resp. *Policies*) and therefore multiple decisions together in various ways (defined in the standard) to make the final decision.
- **Request-Response protocol**: The XACML standard also defines a XML/JSON data model for the authorization decision request (XACML *Request*) that a PEP (described later) creates with all the

necessary access request attributes and sends to the PDP API for evaluation; and the resulting response (XACML *Response*) that contains the final decision (Permit or Deny).

- **Architecture framework**: The XACML standard also defines a high-level architecture, re-using the ABAC concepts described in **Error! Reference source not found.**.

2.3.7 Main interactions

2.3.7.1 Use cases

2.3.7.1.1 Authorization in resource-constrained devices

This enabler covers the use case 4.1: Authorization in Resource-Constrained Devices Supported by 5G Network defined in D2.1 **Error! Reference source not found.**.

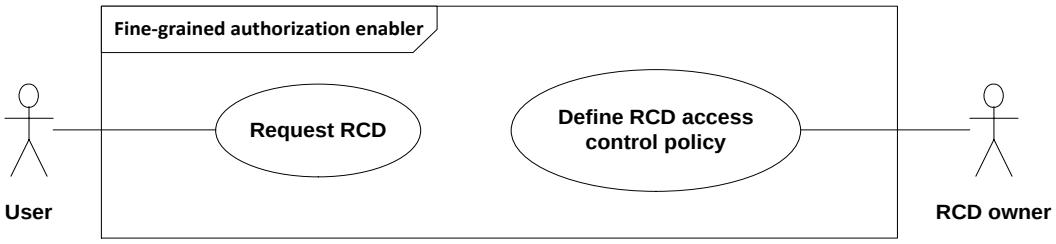


Figure 2-28: Authorization in resource-constrained devices Use cases diagram

Figure 2-28 shows a generic use cases diagram for the enabler. The user (Alice in D2.1) can perform an authenticated request towards a RCD (a sensor in D2.1). The details of the authentication and verification are not part of the use case but will be taken into account and described in the sections below. In the same way, the RCD owner (sensors' owner in D2.1) can define the security policies which will be enforced in front of its RCD and condition the user access.

2.3.7.1.2 Authorization in satellite systems Use case

This enabler covers the use case 1.3 "Satellite Identity Management for 5G Access" defined in D2.1 **Error! Reference source not found.**.

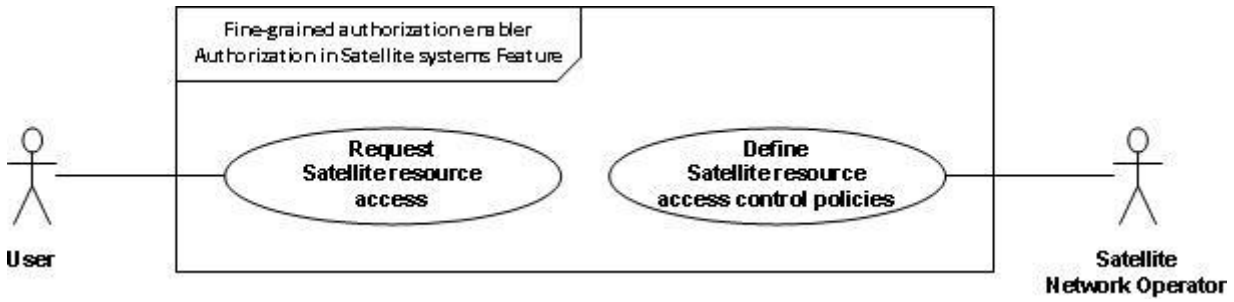


Figure 2-29: Authorization in satellite systems Use cases diagram

Figure 2-29 shows a generic use cases diagram for the enabler. The user (Bob in D2.1) can perform an access request towards a satellite resource. The details of the authentication and verification are not part of the use case but will be taken into account and described in the sections below. In the same way, the satellite resource owner (Satellite Network Operator in D2.1) can define the access control policies which will be enforced in front of the satellite resources and condition the user access. For R2, those control

policies are extended using network information such as location, timestamp and operator/user role in order to improve the grained authorization.

2.3.7.2 Components and interaction overview

2.3.7.2.1 Authorization in resource-constrained devices feature

Error! Reference source not found. shows a static view of the different components implementing the enabler and their links. The different APIs exposed by the enabler are described in section **Error! Reference source not found.**.

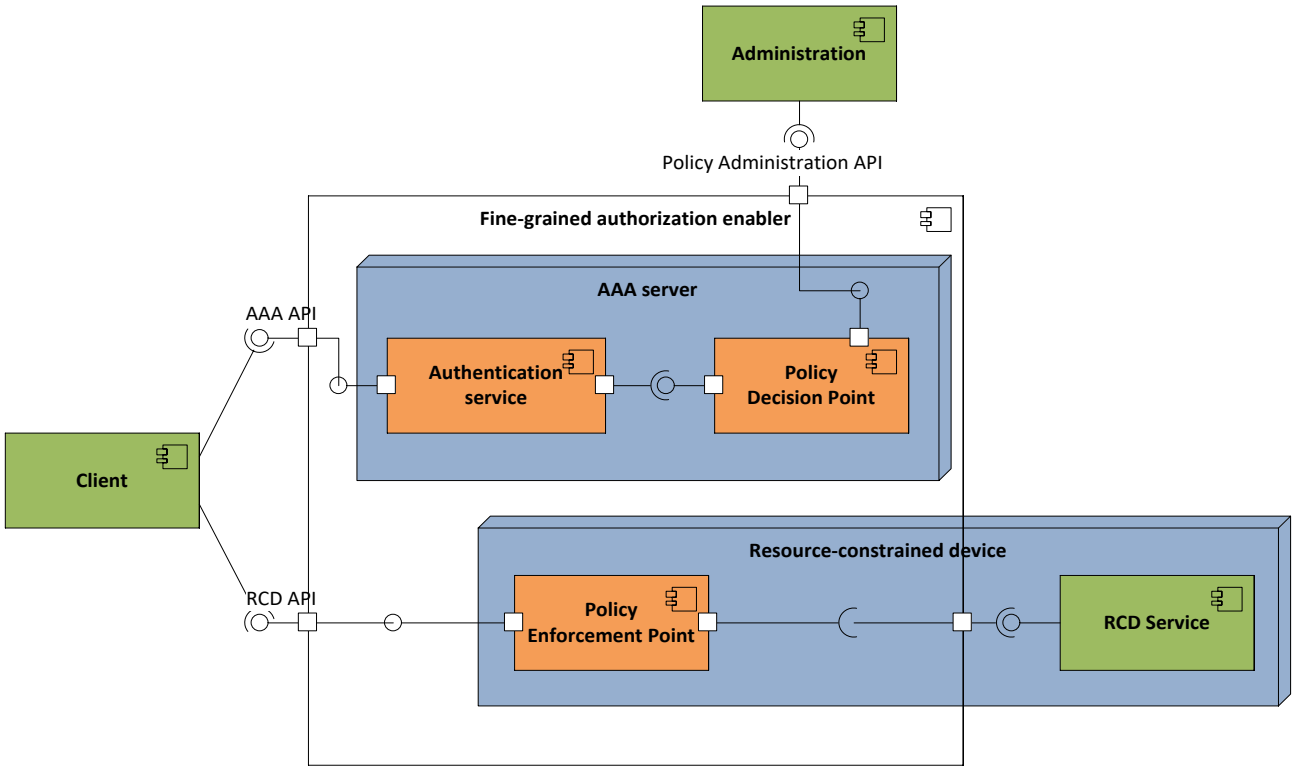


Figure 2-30: Authorization in resource-constrained devices Components diagram

A client can address the PEP in order to request a RCD, as well as an Authentication service in order to get an access token which will be used to grant the RCD request. The Authentication service is linked to the PDP in order to define a scope for the token, for evaluation by the PEP. Contrary to the common ABAC architecture described in section **Error! Reference source not found.**, the PDP and the PEP are not linked together, the information they usually need to share will be part of the token. The dynamic flow of RCD requesting use case is detailed on Figure 2-31.

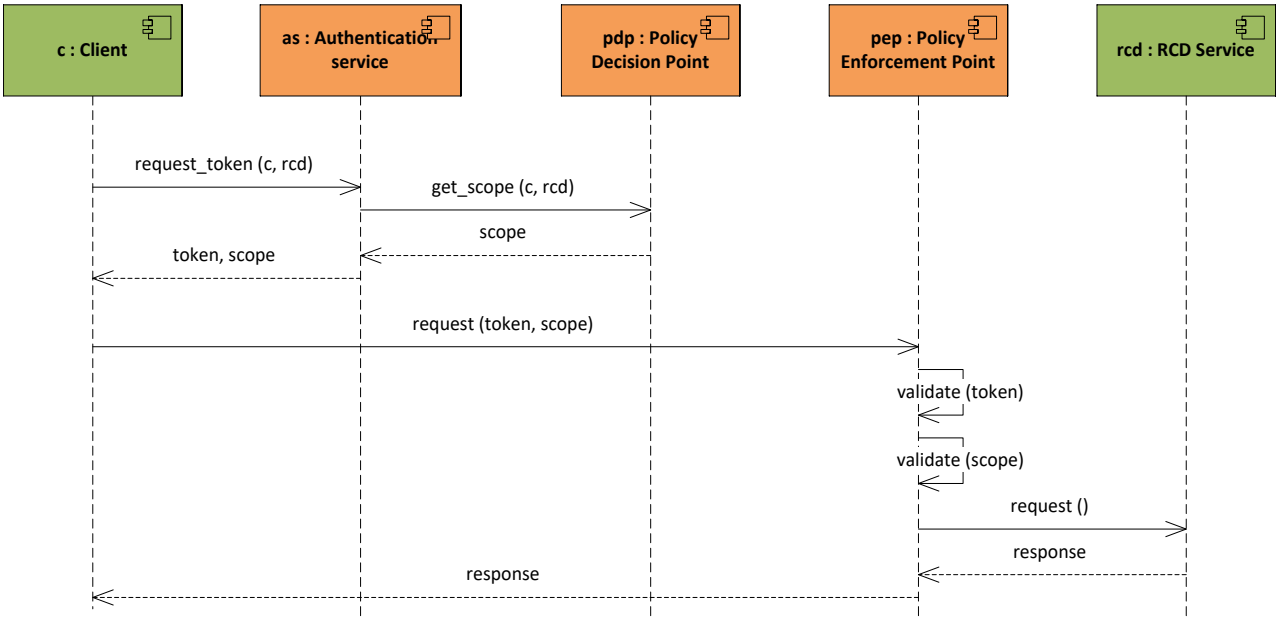


Figure 2-31: Use case "Request RCD" sequence diagram

Moreover, an administration API is exposed by the PDP component in order to let the RCD owner define his own access control policies through any administration component. The dynamic flow of the RCD access control policy definition use case is detailed on Figure 2-32.

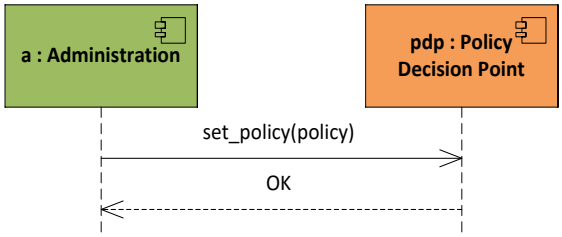


Figure 2-32: Use case "Define RCD access control policy" sequence diagram

2.3.7.2.2 Authorization in satellite systems feature

A geosynchronous satellite imposes a delay between messages: a radio signal takes approximately 0.25 of a second to reach and return from the satellite. Addressing this constraint involves optimizing the number of messages interchanged (e.g. in authorization functionality) with the satellite device. Therefore, this feature is related to satellite systems, but may also relate to other systems (e.g. RCD).

Figure 2-33 shows a static view of the different components implementing the enabler and their links. The different APIs are described in section **Error! Reference source not found.**.

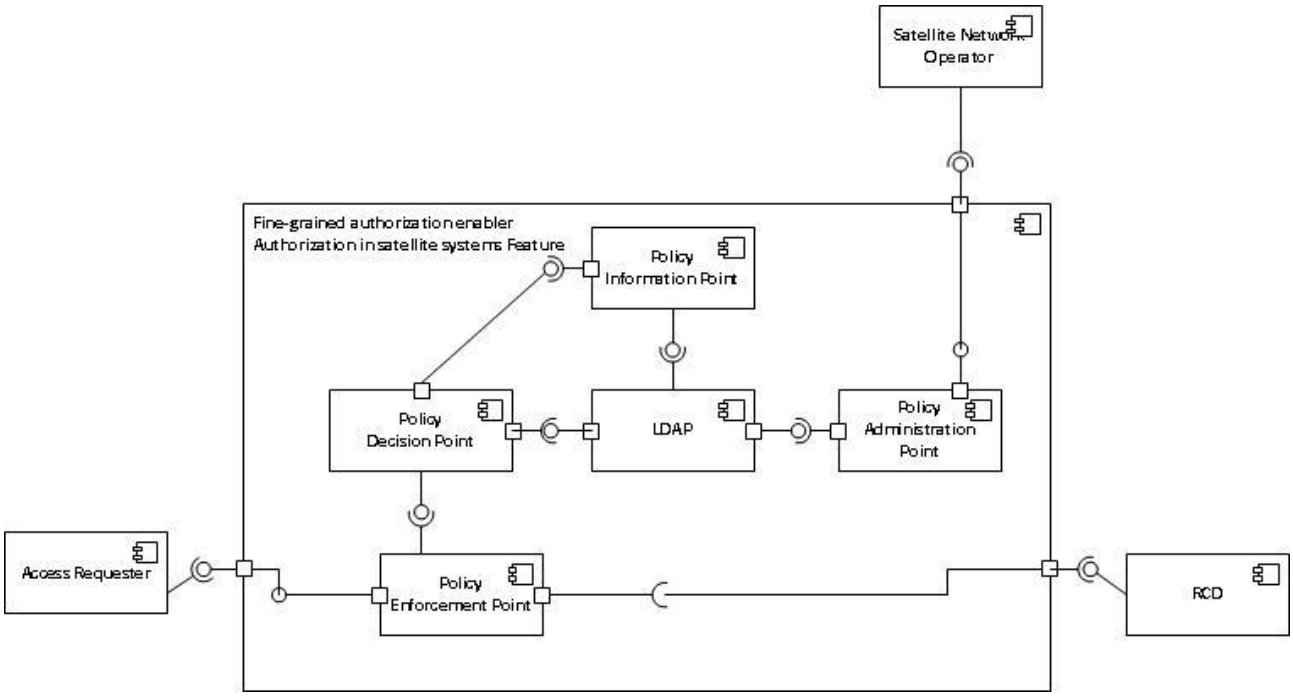


Figure 2-33: Authorization in satellite systems Components diagram

An administration API is exposed by the PAP component in order to let the satellite resource owner define his own access control policies through any administration component.

A client can address the PEP in order to request a satellite resource, as well as an Authentication service in order to get an access token which will be used to grant the satellite resource request. The Authentication service is linked to the PDP in order to define a scope for the token, for evaluation by the PEP. Unless the common ABAC architecture described in section **Error! Reference source not found.**, the PDP and the PEP are not linked together, the information they usually need to share will be part of the token. The dynamic flow of satellite resource requesting use case is detailed on Figure 2-34.

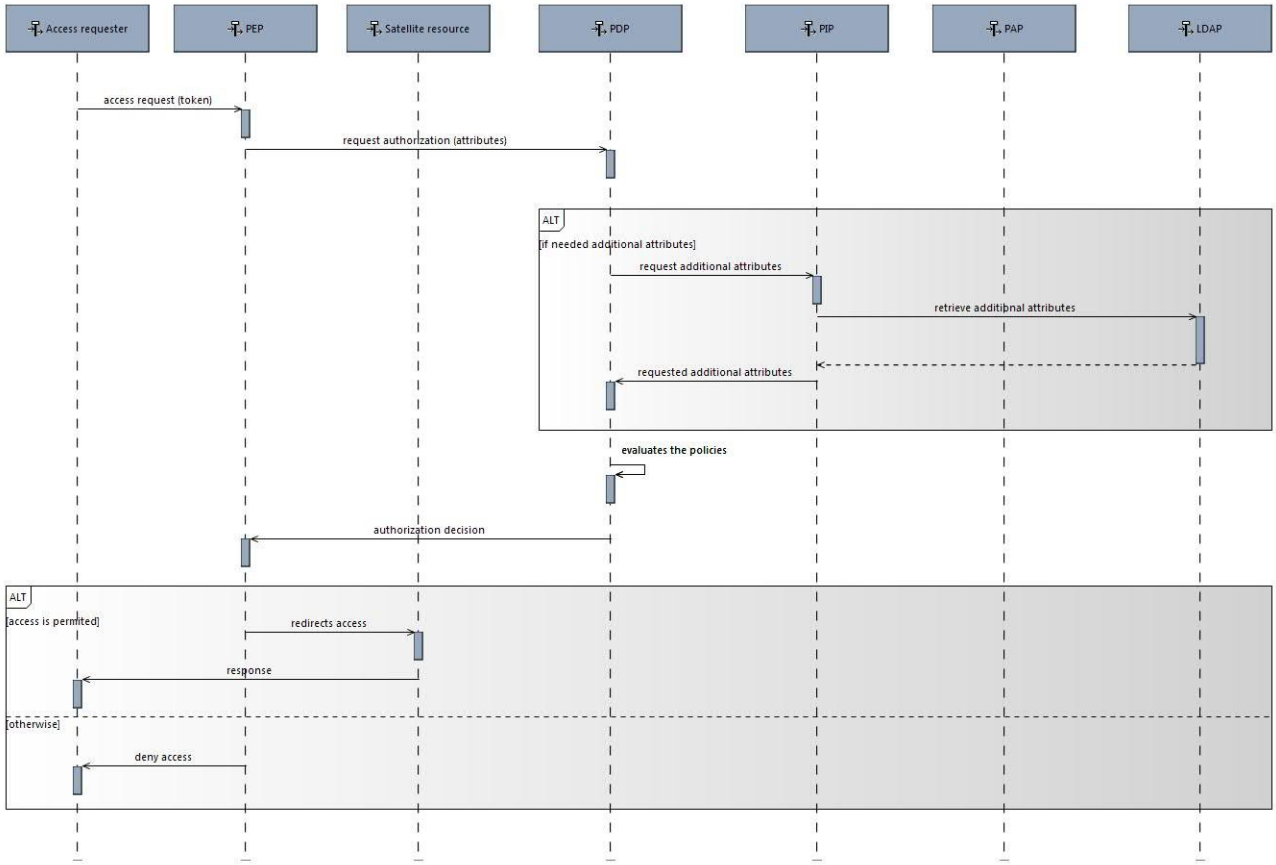


Figure 2-34: Use case "Request satellite resource access" sequence diagram

2.3.8 Architectural drivers

The enabler shall provide a method for flexible definition of the access control policies.

2.3.8.1 High-Level functional requirements

The high-level functional requirements are:

- The user shall be authenticated using 5G credentials to access a RCD or a satellite resource.
- The enabler shall deliver a token to an authenticated user, containing a proof of its access rights.
- The enabler shall ensure that any access to the RCD must be authenticated.
- The enabler shall provide a method to define access control policies.
- The enabler shall provide authorized access to the RCD or to the satellite resources according to the access control policy defined by its owner.
- The information contained in the token shall be sufficient for the PEP module to enforce a decision, in order to limit external information researches.

2.3.8.2 Link to Security Architecture

Figure 2-35 and **Figure 2-36** show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016].

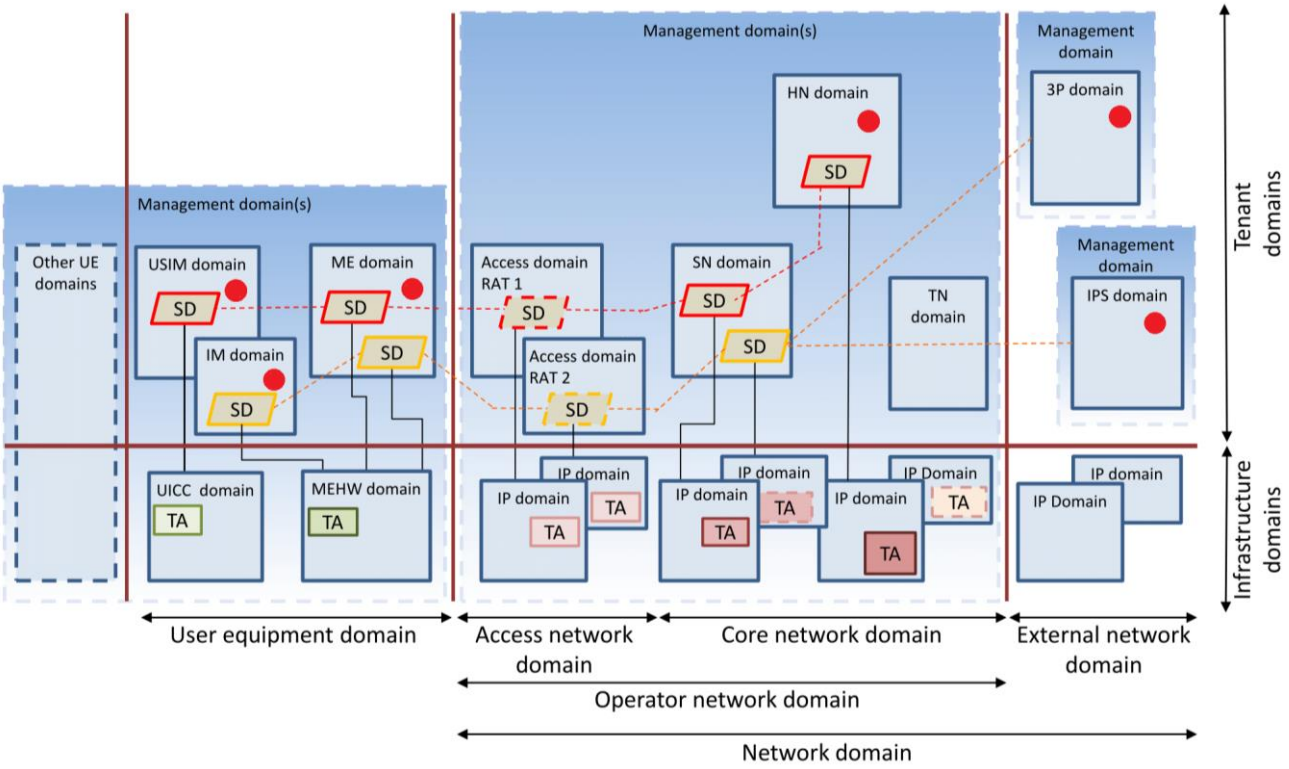


Figure 2-35 Enabler relevant domains

According to the 5G security architecture depicted in deliverable D2.4, the authentication and authorization part of the enabler can either be located in the IM domain if relying on USIM-based authentication or in 3P domain if relying on any third-party identity manager. This service delivers a token usable in 3P domain, where is then located the enforcement module of the enabler.

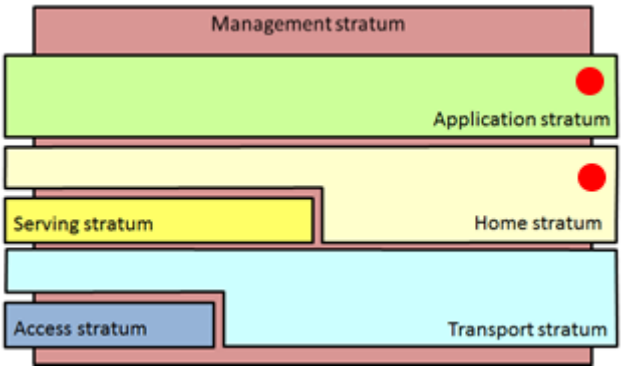


Figure 2-36 Enabler relevant strata

Regarding the strata depicted in deliverable D2.4, chap 4.3, the enabler is involved in Home stratum for USIM-based authentication aspects and in the Application stratum for third-party authentication or token usage.

2.3.8.3 Quality attributes

Compliance with standards: XACML v3.0, OAuth 2.0, JWT, CWT, upcoming IETF standards for RCD.

Fault tolerance: minimized dependency on the AAA central server; if it is temporarily unavailable, the PEP should still authorize already delivered tokens (for a limited duration)

Resource-utilization: minimized bandwidth consumption by cutting the PEP-PDP connection link, minimized CPU and memory consumption for the PEP to fit in a RCD

Integrity/Non-repudiation/Authenticity: token information must be signed by Authentication service, and are authenticatable end-to-end.

2.3.8.4 Technical constraints

For demonstration purpose, the PEP module should be deployable into an object with 900MHz CPU and 1GB RAM at the most.

2.3.8.5 Business constraints

None.

2.3.9 Detailed specifications

2.3.9.1 Introduction

This specification defines the Fine-grained authorization API, which provides fine-grained authorization access to RCD and satellite resources based on defined access control policies.

The API follows the REST design principles and complies with XACML 3.0.

Fine-grained authorization enabler is currently being developed. Below is a preliminary API specification of the enabler.

2.3.9.2 Conformance

An implementation that conforms to this open specification shall implement fully the architecture described.

All the interfaces described are mandatory and must be implemented in order to be compliant with.

The usage of LDAP as a source of attribute values is optional.

2.3.9.3 Authorization in resource-constrained devices API specifications

2.3.9.3.1 AAA API

The AAA API is compliant with OAuth 2.0 standard API. According to the Client Credentials Grant flow, the only interesting OAuth2 endpoint is the token endpoint. Since the client authentication is used as authorization grant, no additional authorization request is needed.

Method Name	*request_token*
Method Definition/Description	Request the OAuth2 token endpoint in order to get an access token. Note that the client must authenticate in any way to call this method.
Method input attributes	

Name	***Type***	***Description***
grant_type	String	Must be set to “client_credentials”.
Method output attributes		
Name	***Type***	***Description***
Token	JWT-encoded or CWT-encoded JSON	A JWT-encoded or CWT-encoded OAuth2 access token, containing the following fields: • *access_token*: an access token identifier • *token_type*: the access token type • *expires_in*: the lifetime in seconds of the access token • *scope*: the scope of the token, containing the access rights granted to the client. The scope should be defined by the PDP, according to the access control policy.

2.3.9.3.2 Policy Administration API

The Policy Administration API is a RESTful HTTP API defined as below:

Method Name	*set_policy*	
Method Definition/Description	Add a XACML PolicySet.	
Method input attributes		
Name	***Type***	***Description***
Policy	PolicySet	A XACML PolicySet
Method output attributes		
Name	***Type***	***Description***
policyUID	String	PolicySet unique identifier (PolicySet UID). This is the identifier allocated by the Service Provider for the PolicySet resource. Note that this is different from the XACML PolicySetId in at least two ways: 1. It is allocated by the Service Provider whereas the XACML PolicySetId is defined by the Service Consumer. 2. It identifies the XACML PolicySetId AND Version. In other words, you can have two different resources with the same PolicySetId but different Versions (therefore different PolicySet UIDs).

Method Name	*get_policy*

Method Definition/Description	Get a deployed XACML PolicySet.	
Method input attributes		
Name	***Type***	***Description***
policyUID	String	The PolicySet UID (see *set_policy* return)
Method output attributes		
Name	***Type***	***Description***
Policy	PolicySet	The XACML PolicySet matching the UID. None if no PolicySet match the UID.

Method Name	*delete_policy*	
Method Definition/Description	Delete a deployed XACML PolicySet	
Method input attributes		
Name	***Type***	***Description***
policyUID	String	The PolicySet UID (see *set_policy* return)
Method output attributes		
Name	***Type***	***Description***
--	--	--

2.3.9.3.3 RCD API

The RCD API exposed by the enabler is a replication of the one exposed by the RCD Service, with in addition the expectation of an access token. The way this token is expected strongly depends on the protected RCD Service API and must be specific to each one. If the RCD Service is over HTTP, the common way is using an additional HTTP header.

2.3.9.4 Examples

2.3.9.4.1 AAA API

2.3.9.4.1.1 request_token

```
POST .../token HTTP/1.1
Host: server.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
```

```
HTTP/1.1 200 OK
Content-Type: application/jwt;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhY2Nlc3NfdG9rZW4iOiIyWW90bkZaRkVqcj
F6Q3NpY01XcEFBIiwidG9rZW5fdHlwZSI6ImV4YW1wbGUiLCJleHBpcmVzX2luIjozNjAwLCJle
GFtcGxlX3BhcmFtZXRlciI6ImV4YW1wbGVfdmFsdWUifQ.5lpcvJ8zZnElXY0nsSNC_0AcuMm-
Q0WMgnSxwv8gj5A
```

2.3.9.4.2 Policy Administration API

2.3.9.4.2.1 set_policy

```
POST .../policies HTTP/1.1
Host: server.example.com
Content-Type: application/xml

<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" ... />
```

```
HTTP/1.1 200 OK
```

2.3.9.4.2.2 get_policy

```
GET.../policies/{policyUID} HTTP/1.1
Host: server.example.com
```

```
HTTP/1.1 200 OK
Content-Type: application/xml;charset=UTF-8

<PolicySet xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" ... />
```

2.3.9.4.2.3 delete_policy

```
DELETE.../policies/{policyUID} HTTP/1.1
Host: server.example.com
```

```
HTTP/1.1 200 OK
```

2.3.9.5 Authorization in satellite systems API specifications

2.3.9.5.1 Define an access control policy

Add a new XACML policy to the PAP policies repository

Request:

- Method: POST
- URI: /pap/policy
- Content-type: application/xml
- Body: XACML policy

Response:

- HTTP status code:

    - 201 when success
    - 400 when error
- Body: None

2.3.9.5.2 Retrieve an access control policy

Get a XACML policy from the PAP policies repository

Request:

- Method: GET
- URL: /pap/policies/{policyID}
    - Parameter policyID

Response:

- HTTP status code:
    - 200 when success
    - 404 when not found
- Content-type: application/xml
- Body: XACML policy

2.3.9.5.3 Request resource access

Request RCD or satellite resource access

Request:

- Method: POST
- URL: /pdp/authorize
- Authorization: {access token}
- Content-type: application/json
- Body: {"userName":<username>,"action":<action>,"resource":<resource>}

Response:

- HTTP status code:
    - 200 when allow access
    - 401 when deny access
- Redirects the request to the satellite resource

2.3.10 Re-utilised Technologies/Specifications

The Fine-grained authorization enabler is based on RESTful design principles. The technologies and specifications used in this enabler are:

- RESTful web services
- Java API for RESTful Web Services - JAX-RS 2.0
- HTTP/1.1
- JSON and/or XML data serialization formats
- The Authentication Server relies on OAuth 2.0 standard
- The AAA API can re-use an existing OAuth 2.0 standard implementation

- In order to manage self-contained tokens, the encoding of the returned token by the Authentication Server uses the standard JSON Web Token (JWT) format
- The Policy Decision Point relies on OASIS XACML v3.0 specification and re-utilises FIWARE Access Control Generic Enabler as well as the Policy Administration API re-utilises the FIWARE Authorization PDP API Specification
- LDAPJIRA issue tracking product
- SVN software versioning and revision control system
- Git software version control system
- Melody Advance system engineering modelling tool
- Thales Control continuous integration tool based on Jenkins and Sonar

2.3.11 References

[1] INCITS 359-2012 - Role Based Access Control standard by InterNational Committee for Information Technology Standards (formerly NCITS)

[2] Guide to Attribute Based Access Control (ABAC) Definition and Considerations – NIST Special Publication 800-162 http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf

[3] RFC 6749 - The OAuth 2.0 Authorization Frameworkhttps://tools.ietf.org/html/rfc6749

[4] RFC 7519 – JSON Web Token (JWT) https://tools.ietf.org/html/rfc7519

[5] eXtensible Access Control Markup Language (XACML) Version 3.0 – OASIS Standard http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf

[6] ISO 25010 – CD 25010.2, Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)Quality model

[7] Using XACML Policies as OAuth Scope – Hal Lockhart http://www.openliberty.org/wiki/uploads/3/38/XACML_as_OAuth_Scope.pdf

[8] 5G-Ensure Consortium, Deliverable 2.1 Use Cases. Available online athttp://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf, 2016.

[9] 5G-Ensure Consortium, Deliverable 3.1 5G-PPP security enablers technical roadmap. Available online at http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[10] OpenID Connect Core 1.0. OpenID foundation specification. Available online at http://openid.net/specs/openid-connect-core-1_0.html.

[11] Authentication and Authorization for Constrained Environments (ACE). IETF working group draft. Available online at https://tools.ietf.org/html/draft-ietf-ace-oauth-authz (Work in progress).

[12] CBOR Object Concise Binary Object Representation (CBOR). IETF RFC 7049. Available online at https://tools.ietf.org/html/rfc7049.

2.4 Federative authentication context usage enabler Open Specifications

2.4.1 Preface

In the context of a slice based on different infrastructures, the end user connected to the slice wants to use different services. These services, offered by these infrastructures, need to trust the authentication mechanisms used by the end user in the context of identity federation. The goal of this enabler is to collect at 5G nodes the authentication context of an end user and to provide this information to service providers allowing them to adapt dynamically their security policy using their risks evaluation before delivering the service.

2.4.2 Status

This enabler is a new enabler of release 2. As this enabler will propose mainly modifications of standards (Diameter), the more important work is on architecture side. In the context of 5G ENSURE release 2, the priority is to have a clear architecture at the end of release 2. Nevertheless, to reflect on the progress of the development, a preliminary open specification is presented here.

2.4.3 Copyright

2.4.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

2.4.5 Terms and definitions

Acronym	Signification
AAA	Authentication Authorization Accounting
HSS	Home Subscriber Server
LTE	Long Term Evolution
PIN	Personal Identification Number
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
UE	User Equipment
USIM	Universal Subscriber Identity Module

2.4.6 Overview

In 5G, the end user may enter some network/system context (e.g. a slice) through different access points. Depending on these access points, different authentication mechanisms will be used and will characterize the trust to be granted to this end user.

This enabler will propose:

- First a new way to define the level of authentication performed by a user and to store it. In a 5G environment, a user can access a service by using different way for being authenticated (e.g. PIN code of UE USIM / SIM card;) or by a login/password using an internet connection, …). This enabler will work to extend/complete the AAA authentication by adding information about the level of user's authentication and update it each time a new authentication is performed. This information shall be stored in a database, this database will offer an API to provide it. This database will be hosted by the HSS and owned by the same provider.
- Second the possibility to know how a user has been authenticated in the network in order to adapt accordingly the service offering. A typical example is for using bank payment system. In this example, the bank could request a trustworthy authentication with stronger credential than only a login/password authentication, network access through trust access points. If a node (or a service) needs this kind of authentication, it needs to have this kind of information and to be able to interact with the user in order to obtain the desired level of authentication.

Overall the approach foreseen would be as follow

- Nodes will collect authentication context provided by the AAA server managing the end users.

- Depending on this context, these nodes could allow or not the access.
    - If not, the entity in charge of the end user has to strengthen the end user authentication by a new authentication request for example.
        - With this new authentication, the trust level authentication of the end user is increased and the 5G node can permit access.
        - If the end user is not able to provide a more trusted authentication (due to a limitation of his/her UE for example), the access is denied.

2.4.7 Basic Concept

In the context of this enabler, two main aspects are considered to trust a user using IP services, the kind of authentication and the access point used to enter the network. Regarding the access point, 3GPP has defined an architecture [2] for allowing 3G PP access, 2G/3G access and non 3GPP access. Regarding the kind of authentication, we rely on EAP AKA [9] since defined. This obviously without presuming of other ways to come to authenticate users in 5G.

The figure below displays the 3GPP architecture for the 3 access ways we have considered and the main components involved. For the 2G/3G and 4G LTE access, the way to enter in the network is the same and starts at the level of MME and the access is allowed by the HSS.

There is a difference for non 3GPP IP access with 2 specific cases depending if this access is trusted or untrusted. In this specific case, a 3GPP AAA server is used to perform the authentication operations to enter the network. The 3GPP AAA server interacts with the HSS to manage these accesses.

These 3 access ways are associated with3 level of authentication. This enabler proposes evolution at HSS level and its interfaces which are used to manage the authentication of the different users. As it is not realistic to propose a new way to authenticate users completely separate from the 3GPP architecture, this enabler aims at proposing only minimal evolution regarding the existing HSS interfaces mainly focusing:

S6a :

The interface S6a lies between the HSS (Home Subscriber Server) and the MME (Mobility Management Entity) for authentication and authorization. This interface has the following properties:

1. Transport of subscriber related data.
2. Transport of location information.
3. Authorizing a user to grant access to EPS.
4. Transport of authentication information

S6d :

The interface S6d lies between the HSS and the SGSN (Serving GPRS Support Node) used to retrieve and store mobility-related parameters. This interface shares the same properties as those of S6a.

SWx :

The interface SWx lies between the HSS (Home Subscriber Server) and the 3G AAA for authentication and authorization in the context of an external 3GPP IP network access. This interface has the following properties:

1. Authentication Information Retrieval

2. UE Registration Notification
3. PDN GW Identity Notification
4. User Profile Update (HSS-initiated)
5. Provide User Profile (AAA-initiated)
6. UE De-registration notification (AAA-initiated and HSS-initiated)

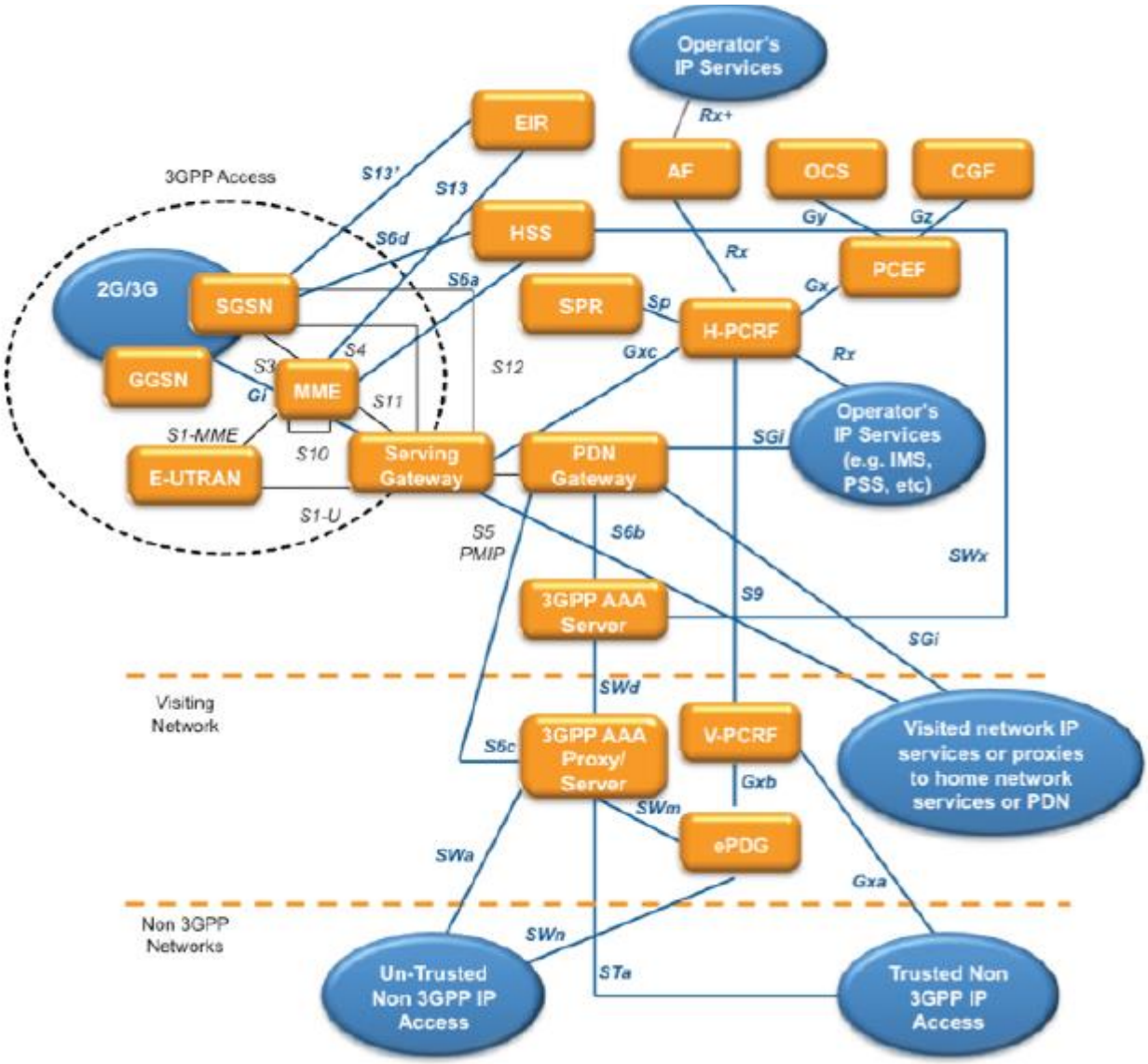


Figure 2-37: Interaction between 3GPP access, 2G/3G access and non 3GPP Networks [5]

Sh :

The Sh interface allows the communication between the Application Server (AS) function and the Home Subscriber Server (HSS), the Sh interface offers two basic ways:

1. To query or update a user's data stored on the HSS
2. To subscribe to and receive notifications when a user's data changes on the HSS

[5] http://go.radisys.com/rs/radisys/images/paper-lte-diameter-eps.pdf

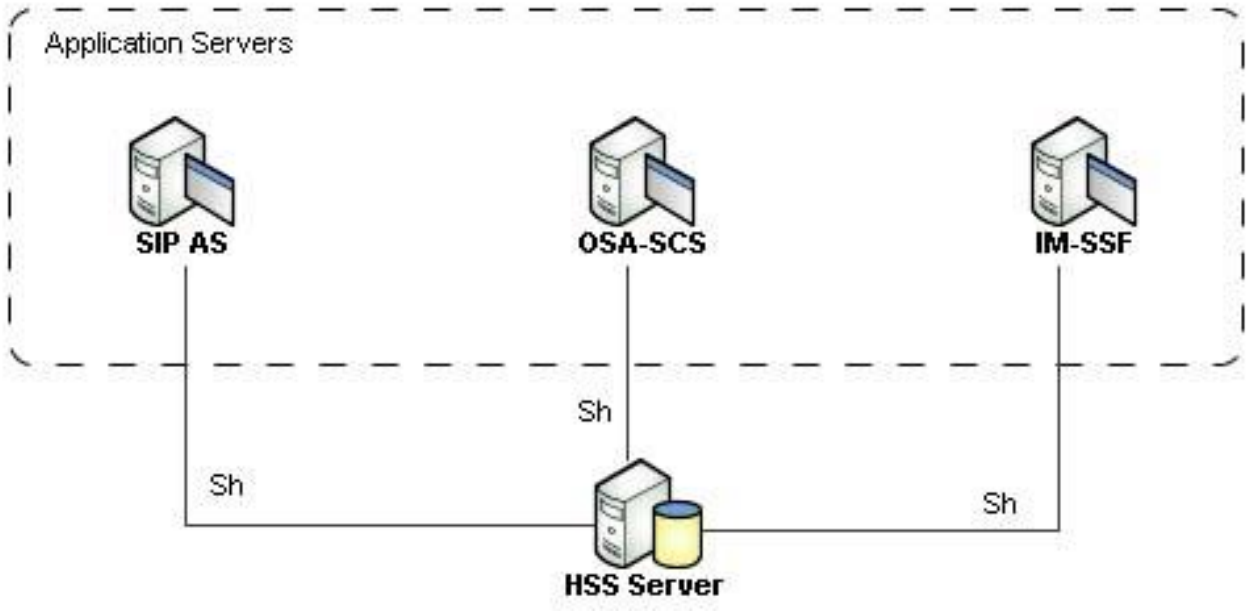


Figure 2-38: Interaction between HSS and the different AS servers of IMS[6]

2.4.8 Main interactions

2.4.8.1 Use Cases

A typical use case could be:

1. An end user is connected through an untrusted 3GPP IP access, the information is detected in the HSS through the usage of SWx interface. The HSS stores this information.
2. This end user wants to use a videoconferencing service managed by the IMS and that is not free.
3. This IMS videoconferencing service checks if the authentication of the user is at level requested. As this user was authenticated and comes from an un-trusted non 3GPP IP network, this authentication level is rejected by the IMS service.

2.4.8.2 Components and interaction overview

The figure below displays all HSS interfaces and the specific ones involved in this enabler. This enabler won't provide a new component of 3GPP architecture but only proposes minor modifications on interfaces and on HSS.

[6] https://svn.java.net/svn/jdiameter~svn/trunk/www/extension-api.html)

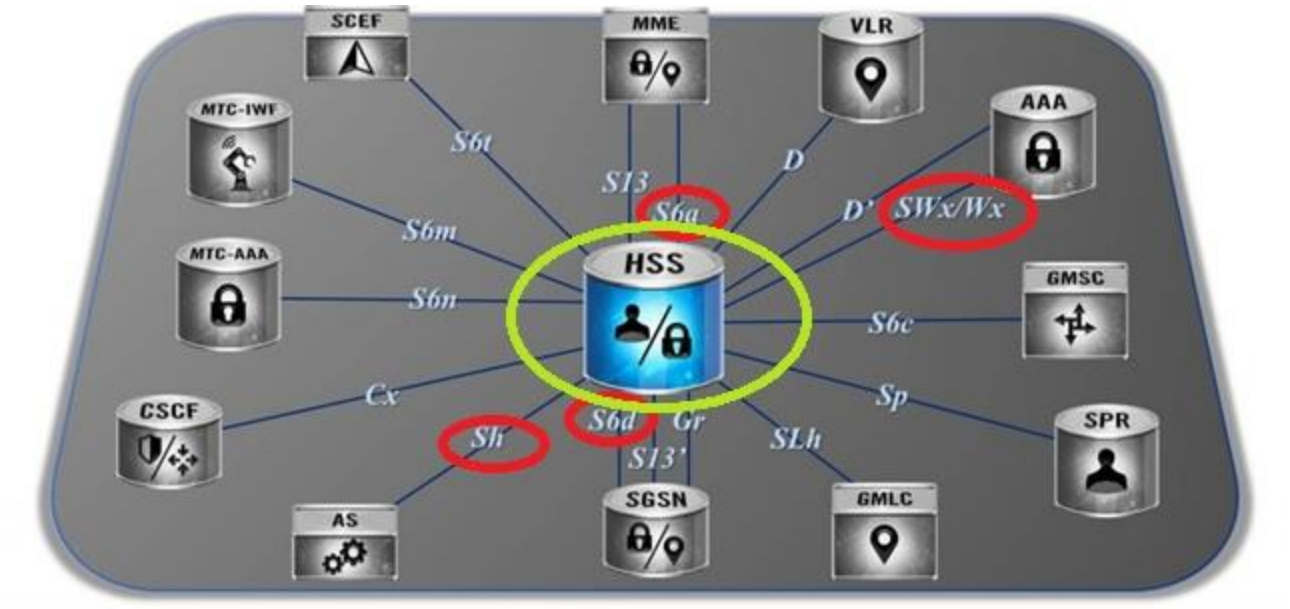


Figure 2-39: HSS interfaces [7]

2.4.9 Architectural drivers

2.4.9.1 High-Level functional requirements

The high level requirements can be summarized as follow:

1. Each time a user is authenticated, the HSS shall store the kind of authentication. The three types of authentication are:
    a. Strong: if the access point is a 2G/3G or 4G LTE access point.
    b. Medium: if the access point is a trusted or untrusted non 3GPP IP access point
    c. Light: Not used in this enabler but could be used for future other kinds of access.
2. An external service could request the type of user authentication.

2.4.9.2 Quality attributes

A product implementing the specifications described below should be evaluated primarily based on the following quality attributes:

*Security:

- Confidentiality
- Integrity
- Non-repudiation
- Accountability
- Authenticity
- Compatibility
    - Co-existence
    - Interoperability

[7] https://www.developingsolutions.com/wp-content/uploads/2012/02/HSS-Emulator_2-1024x481.jpg

2.4.9.3 Technical constraints

As this enabler proposes different evolutions of Diameter interfaces but also adds functionality to HSS, the main technical constraints are to be compliant with the existing HSS interfaces by building on top of existing specifications [3][4][5].

2.4.9.4 Business constraints

No known business constraint.

2.4.9.5 Link to Architecture

2.4.9.5.1 Security objectives fulfilled by the enabler

The different Security objectives defined in D2.4, covered by this enabler, are detailed.

Objective ID	Objective description	Enabler contribution
O2.1	5G must provide a security and privacy level higher or at least equal to the security and privacy level in 4G	This enabler is built on the top of 4G architecture and completes it, offering more security.
O2.4	5G must enable seamless interworking of different network technologies, mobile, fixed as well as satellite without exposing the security level of each of these technologies to new threats	This enabler provides a way to know the new access points and to allow the component to adapt their security policy depending on that.
O7.1	5G must support traditional UICC/USIM based management of subscriber credentials	No modification of subscriber credentials
O7.2	Management and storage of alternative credentials by an external vertical AAA must be done with a security level commensurate with both that of the vertical application as well as the operator business partner	This enabler is compliant with that objective, assuming the interface SWx is used to interact with the HSS.
O8.2	5G systems must allow secure interworking with external systems, e.g. AAA provided by a vertical or external management systems	This enabler is compliant with that objective, assuming the interface SWx is used to interact with the HSS.

2.4.9.5.2 Strata location of the enabler

Regarding the strata depicted in deliverable D2.4, chap 4.3, the enabler is involved in Home stratum for all authentication aspects (stored inside the HSS) and in the Application stratum for its usage.

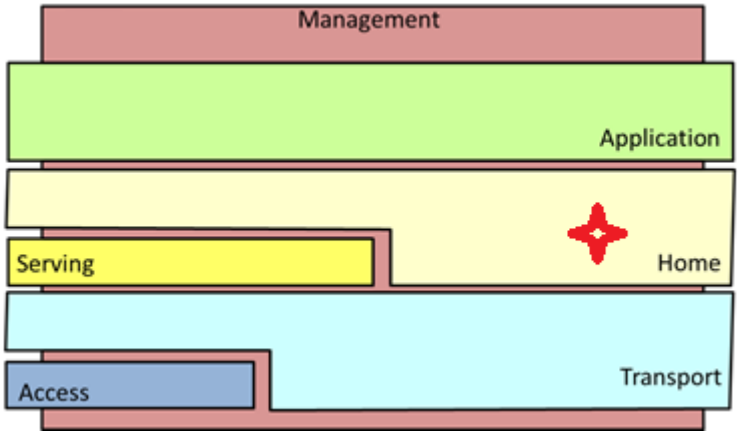


Figure 2-40

2.4.9.5.3 Enabler mapping to architecture domains

The enabler is located in the AN and HN Domains of the 5G security architecture depicted in deliverable D2.4. The enabler offers a service usable in 3P Domain and in IP Service Domain.

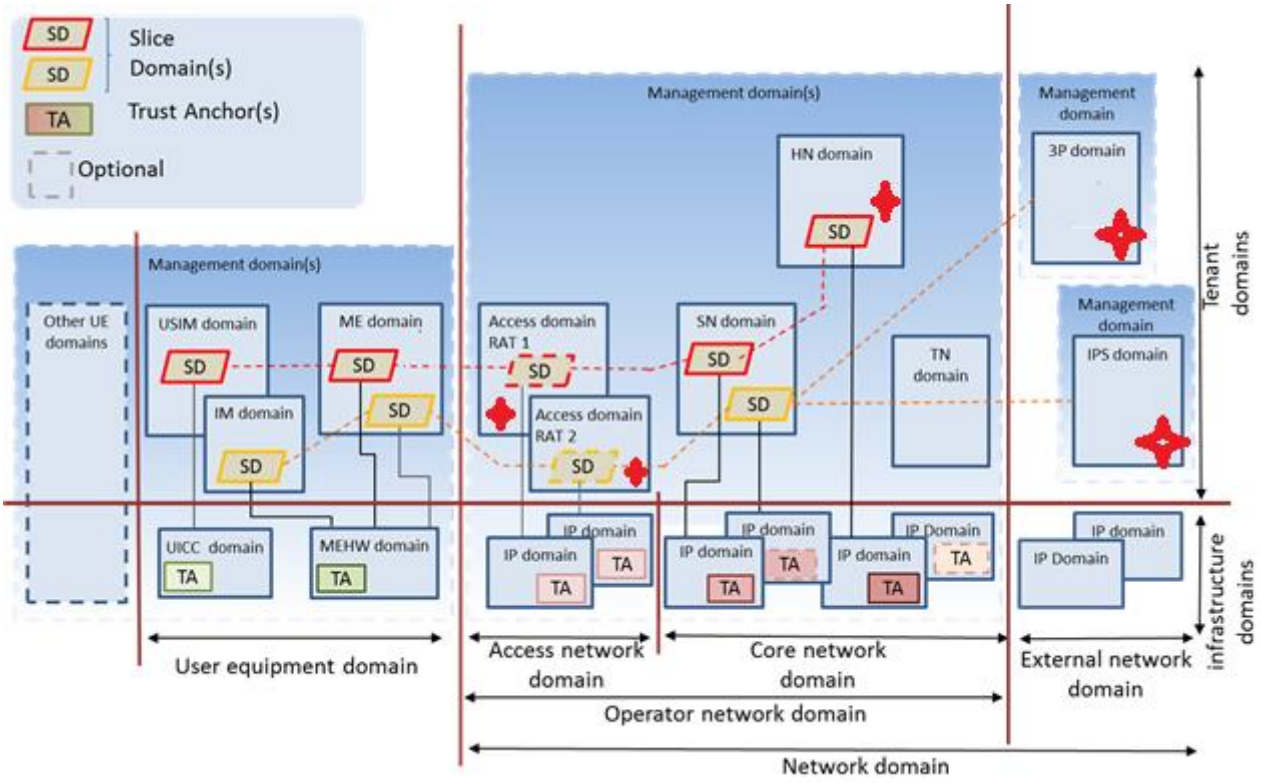


Figure 2-41

2.4.10 Detailed specifications

2.4.10.1 Introduction

This specification defines the “Federative authentication context usage” enabler, which proposes new evolutions of the Diameter protocol and new features to integrate in the HSS information regarding the authentication level.

This enabler is mainly for exploring new capabilities regarding how to use authentication information in a network. Depending on the architecture progress, some part could be prototyped but not fully developed. This specification is the first high level architecture of the solution.

2.4.10.2 Conformance

An implementation to be reported as conformant should comply with the open specifications here stated for the enabler.

2.4.10.3 Enabler Component Specifications

In this paragraph, we described the evolutions of the interfaces and the HSS we need to store the kind of authentication and to allow external components to retrieve this information.

Storage of authentication level

The existing diameter interfaces S6a, S6d and SWx are used to perform the authentication of the users in the network and to take into account the subscriber's information. The different 3GPPP working groups have specified how to perform the user's authentications for different access (2G/3G and 4G LTE) but limited to them. That is why, for this enabler, we propose to re-use the existing authentication procedures and to store inside the HSS this kind of information as the "kind of authentication".

The authentication procedures:

In [3, table 5.2.3.1.1/1], the authentication procedure contains the 2 following requests depending on what the MME/SGSN request for the authentication performs through S6a and S6d:

Information element name	Mapping to Diameter AVP	Cat.	Description
Requested EUTRAN Authentication Info	Requested EUTRAN Authentication Info	C	This information element shall contain the information related to authentication requests for E-UTRAN.
Requested UTRAN/GERAN Authentication Info	Requested UTRAN GERAN Authentication Info	C	This information element shall contain the information related to authentication requests for UTRAN or GERAN.

Through SWx, in [4, table 8.1.2.1.1], the authentication procedure (for normal one and after a failure) from 3GPP AAA to HSS is:

Information element name	Mapping to Diameter AVP	Cat.	Description
Authentication Method	SIP Authentication Scheme	M	This information element indicates the authentication method. It shall contain one of the values EAP-AKA or EAP-AKA'. EAP-AKA' is specified in IETF RFC 5448

For each of these requests, the HSS answers with authentication info. At this stage, it knows the authentication protocols used and generates the authentication vectors (EAP AKA). Different other ways of authentication could be used in 5G, not restricted to AKA, but we could anticipate that, in all cases, the information would be communicated with the following AVP:

- Requested EUTRAN Authentication Info
- Requested UTRAN GERAN Authentication Info
- SIP Authentication Scheme

The evolution is that the HSS stores the kind of authentication as follow:

If HSS receives a request S6a or S6d with a message for "Authentication Information Request" including a "Requested EUTRAN Authentication Info" or "Requested UTRAN GERAN Authentication Info" for a user, it stores "Strong authentication" for this user

If HSS receives a SWx request with a message for Authentication request" including a "SIP Authentication Scheme" for a user, it stores "Medium authentication" for this user

Usage of authentication level

The existing interfaces Sh is used by AS (e.g. IMS architecture) to exchange user's data with the HSS. This interface could be reused for providing authentication information for any applications needing them. The document [5] describes this detailed interface. The different commands are:

Command-Name	Abbreviation	Code	Section
User-Data-Request	UDR	306	6.1.1
User-Data-Answer	UDA	306	6.1.2
Profile-Update-Request	PUR	307	6.1.3
Profile-Update-Answer	PUA	307	6.1.4
Subscribe-Notifications-Request	SNR	308	6.1.5
Subscribe-Notifications-Answer	SNA	308	6.1.6
Push-Notification-Request	PNR	309	6.1.7
Push-Notification-Answer	PNA	309	6.1.8

Table 2-11: Name of Sh commands [5]

The **User-Data** AVP stored in the HSS contains different information regarding the user. This AVP is sent by the HSS to the IMS AS in different commands which are UDA/SNA depending on the **Data-Reference** AVP provides in the request.

The existing **Data-Reference** AVP already defined are described in the following table.

The **Data-Reference** AVP is of type Enumerated, and indicates the type of the requested user data in the operation UDR and SNR. Its exact values and meaning is defined in [8]. The following values are defined:

Data Ref.	XML tag
0	RepositoryData
10	IMSPublicIdentity
11	IMSUserState
12	S-CSCFName
13	InitialFilterCriteria
14	LocationInformation
15	UserState
16	Charging information
17	MSISDN or MSISDN +ExtendedMSISDN

18	PSIActivation
19	DSAI
20	Reserved
21	ServiceLevelTraceInfo
22	IP Address Secure Binding Information
23	Service Priority Level
24	SMSRegistrationInfo
25	UE reachability for IP
26	T-ADS Information
27	STN-SR
28	UE-SRVCC- Capability
29	ExtendedPriority
30	CSRN
31	Reference Location Information
32	IMSI
33	IMSPrivateUserIdentity

Table 2-12

And more detailed in [8, chap7.6].

For the "Federative authentication context usage" enabler, the proposition is to add a specific item:

Data Ref.	XML tag	Access key
34	AuthenticationLevel	Data Reference + (IMS Public User Identity OR Public Service Identity)

Table 2-13

Its possible values are:

- STRONG
- MEDIUM
- LIGHT

2.4.10.4 API specifications

There are no specific APIs to be described because the different Diameter interfaces and the AVPs take in charge this aspect.

2.4.10.5 Library Specification

none

2.4.11 References

[1] 3GPP, *TR 21.905* Vocabulary for 3GPP Specifications.

[2]3GPP, *TS 33.401 ;* 3GPP System Architecture Evolution (SAE);Security architecture; *v13.2.0*

[3] 3GPP, *TS 29.272;* Evolved Packet System (EPS); Mobility Management Entity (MME); Serving GPRS Support Node(SGSN) related interaces based on Diameter protocol*; v13.5.1.*

[4] 3GPP, *TS 29.273;* Technical Specification Group Core Network and Terminals; Evolved Packet System (EPS);3GPP EPS AAA interfaces*; V9.2.0.*

[5] 3GPP, *TS 29.329;* Technical Specification Group Core Network and Terminals; Sh Interface based on the Diameter protocol; Protocol details*; V14.2.0.*

[6] 3GPP, *TS 29.336;* Technical Specification Group Core Network and Terminals; Home Subscriber Server (HSS) diameter interfaces for interworking with packet data networks and applications*; V14.1.0.*

[7] IETF RFC 6733: "Diameter Base Protocol".

[8] 3GPP, *TS 29.328;* Technical Specification Group Core Network and Terminals; IP Multimedia (IM) Subsystem Sh interface; Signalling flows and message contents Group Core Network and Terminals; Home Subscriber Server (HSS) diameter interfaces for interworking with packet data networks and applications*; V14.2.0.*

[9] IETF RFC 4187: "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)".

2.4.12 Acknowledgements

None

3 Privacy Enablers open specifications

This section contains the open specifications of the four Privacy enablers due at Release 2: the IMSI Pseudonyms and Home-Network Centric components from the enabler Privacy Enhanced Identity Protection, the enabler Device Identifier(s) Privacy, the enabler Device-Based Anonymization and the enabler Privacy Policy Analysis.

3.1 Privacy Enhanced Identity Protection Enabler Open specifications

3.1.1 Preface

This enabler aims to provide protection against subscriber's identity disclosure in 5G networks to unauthorized parties. The main goal is to offer stronger protection of user identity than in current 3G and 4G networks. The fundamental idea behind this enabler can be summarized in several simple concepts: 5G true user identities shall not be transferred in clear text over the network; unique dynamic (pseudo) random pseudonyms should be used during all normal operations. If a true identity (e.g., the IMSI) has to be sent from the UE to the network, it should be sent encrypted.

The present open specification relates only to the R2 version of the enabler Privacy Enhanced Identity Protection Enabler and the features planned for that version and enumerated in the roadmap D3.5 [2], namely: Home Network centric IMSI protection and IMSI pseudonymization.

3.1.2 Status

Initial versions (1.0) of these two new components is currently under development.

3.1.3 Copyright

3.1.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

3.1.5 Terms and definitions

SN	Serving Network - The serving network provides the user with access to the services of their home environment. When roaming, the serving network is referred to as the visited network.
HN	Home Network – Is the network where the mobile users have the subscription.
IMSI (MCC, MNC, MSIN)	International Mobile Subscriber Identifier (Mobile Country Code, Mobile Network Code, Mobile Subscription Identification Number)
GUTI	Globally Unique Temporary UE Identity
TMSI	Temporary Mobile Subscriber Identity
M-TMSI	M-TMSI inside the GUTI uniquely identifies the UE within the MME
UE	User Equipment – the mobile device used to connect to the network.
RTMSI	RTMSI – Random Temporary Mobile Subscriber Identity
ECIES	Elliptic Curve Integrated Encryption Scheme
FPE	Format Preserving Encryption

KDF	Key Derivation Function (e.g., HMAC-SHA-256 in LTE)

Table 3-1

3.1.6 Overview

The main end user privacy issues in 5G networks may arise because of the exposure of long term identifiers. The main authentication methods used at present (i.e., AKA) rely on symmetric key encryption, and there are unavoidable situations where they use the IMSI in order to correctly identify the subscriber requesting network connectivity. Therefore, in situations where no shared keys are yet available, long time identifiers are exchanged in clear text and are exposed to the whole range of IMSI sniffing attacks, and, subsequently to user tracking attacks. Active IMSI catchers can defeat subscriber identity protection even after shared keys have been established. For increased user privacy, 5G networks should offer security mechanisms able to defeat or totally avoid such exposure. Such solutions can be based on the use of encryption or random pseudonyms.

3.1.6.1 Home Network centric IMSI protection

In this solution, as illustrated in **Error! Reference source not found.**, a public-key scheme is implemented, in which the UE directly or indirectly uses the public key of its home network to encrypt parts of the IMSI. The key can be stored on the UEs in advance (e.g. on the USIM card), given that it is static, so there is no need of deploying additional infrastructure for key management, such as a PKI, except possibly a revocation/update mechanism in case of key compromise, but that can be managed on per-operator basis.

The home network will be responsible for performing the decryption and sharing afterwards the clear-text IMSI to the rest of the network elements on the system that may need it. An example can be a visited network (e.g., the MME), which must maintain a copy of all IMSIs from users attached to the network due to e.g. Lawful Interception obligation. Such a transmission of IMSIs should be done over a secure channel.

In order to allow a visited network to route the identifier to the correct home network, both the Mobile Country Code (MCC) and the Mobile Network Code (MNC) of the IMSI are sent in clear-text, while the Mobile Subscription Identification Number (MSIN) of the IMSI will be sent encrypted.

The use of Elliptic Curve Cryptography is desirable given its computational efficiency, message size and key length compared to traditional schemes such as RSA or ElGamal over finite fields. An example of an encryption scheme based on Elliptic Curve can be Elliptic Curve Integrated Encryption Scheme, ECIES [40] which the enables also makes use of. The main idea of this scheme is to use public key cryptography for key agreement, i.e., to agree on a common secret key, which will be used for symmetric encryption of the message. As will be described later, the solution proposes to use the ECIES scheme without a Message Authentication Code (MAC). We call this variant of the ECIES scheme as ECIES* here onwards.

Using the ECIES* scheme, the UE will generate an ephemeral key pair for the key agreement, the private key of which, combined with the home network's public key, will derive a secret key. This secret key together with the MSIN will generate a cipher-text. The UE will send the cipher-text and its ephemeral public key in the attach request (identity response, so that the home network can decrypt the cipher-text and thus obtain the MSIN in clear text.

Due to the ephemeral key generation, the resultant encrypted identifier will be different every time it is created, so that it is infeasible for a third party (like an IMSI catcher) to link a given encrypted IMSI with another encrypted copy of the same IMSI or the true identifier or to guess it.

An encrypted identifier will be used every time a UE needs to send its identity over an insecure channel, such as in the Attach Request or Identity Response messages.

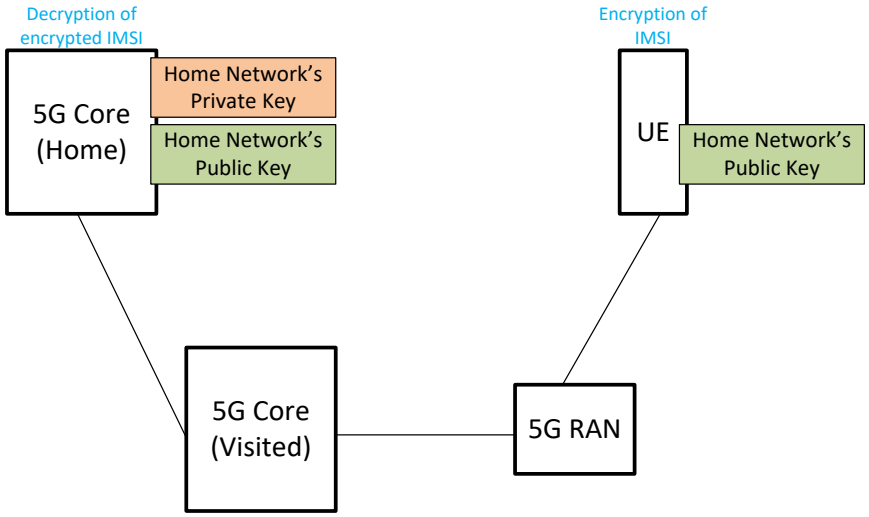


Figure 3-1: Architecture overview.

3.1.6.2 IMSI pseudonymization

This feature complements the R1 "Encryption of Long Term Identifiers" feature in order to avoid exposing the user permanent or long term identities on (at least) the air interface (i.e., in Attach Requests with GUTI, Identity Responses, Paging Requests). The feature's goal is to use one-time unpredictable temporary identifiers in each of the 5G equivalent procedures that mandate the use of GUTI in nowadays LTE systems.

Therefore, the feature proposed the use of pseudorandom dynamic pseudonyms, herein referred as RTMSI (Random TMSI) to be employed in dynamic one time GUTIs (dGUTI). These can be generated by the dedicated 5G network element (the 5G equivalent of the MME in LTE) and communicated to the UE, or can be generated both by the network and UE, by using a (standardized) pseudonym-derivation algorithm with a shared secret key (see **Figure 3-2**).

These RTMSI/dGUTIs are always used instead of real permanent or long term identities (IMSIs) on the air interface, in Attach Requests, in response to Identity Requests, in Paging Requests, etc., and are consumed by usage (as mentioned they follow a "one-time" scheme). The RTMSI/dGUTI generation mechanisms must guarantee collisions avoidance over a Tracking Area (over the pool of UEs connected to the same MME – or its 5G equivalent).

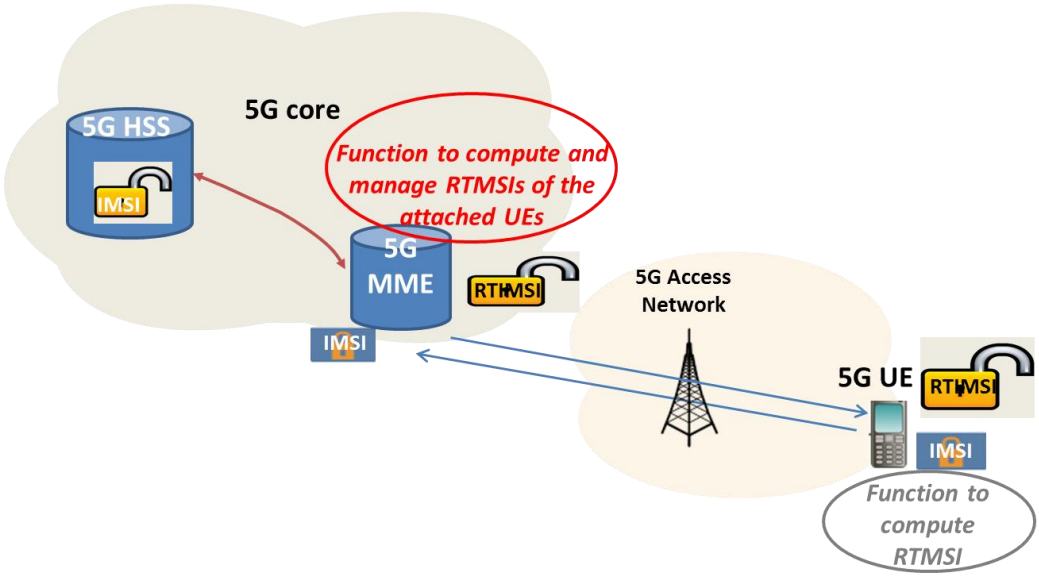


Figure 3-2: IMSI pseudonymization - generic example architecture.

3.1.7 Basic concepts

This section presents the basic concepts of the R2 features Home Network-centric IMSI protection and IMSI pseudonymization.

3.1.7.1 Home Network centric IMSI protection

It was discussed in an earlier section that a variant of ECIES, which we called ECIES*, is used by the solution. It was also mentioned that the ECIES* is basically the same as the original ECIES but without the MAC. Now, this section briefly describes the basic concepts of the ECIES* and explains why the MAC is skipped for this solution.

The ECIES* is basically an asymmetric encryption scheme based on Elliptic Curve Cryptography (ECC). It should be noted that the actual encryption/decryption of the message is symmetric and therefore a symmetric key is used. The asymmetric operation comes into play for derivation of the said symmetric key on the both sides, i.e., for key agreement. The ECIES* is constituent of the following functions (see **Figure 3-3**):

- **Key Generation (KG)**: It generates a public/private key pair according to the elliptic curve parameters. The home network does it ideally only once, and the resulting public key of the home network is provisioned at the UEs. The UEs, however, do KG for every attach and therefore the public/private key pair of the UEs are ephemeral.
- **Key Agreement (KA)**: It returns a point on the elliptic curve (i.e., shared key) resulting from the combination of the public key of one of the parties in the communication with the private key of the other party. The Elliptic Curve Diffie-Hellman (ECDH) primitive (which is a variant of the well-known Diffie-Hellman key agreement protocol using ECC) is used to generate an ephemeral shared key from the ephemeral private key of the UE and the static public key of the home network.
- **Key Derivation (KD)**: It is responsible of generating the actual ephemeral encryption/decryption key from the ephemeral shared key generated in the previous step, i.e., the KA. The hash function SHA-2 is used for KD.

- **Symmetric encryption/decryption**: It performs the actual encryption/decryption of the message (i.e. MSIN in our case) taking the derived ephemeral key from the previous step, i.e., the KD. Since the encryption/decryption key is ephemeral and used only once, the XOR function is sufficient for encryption/decryption. Other alternatives could be for example, the well-known AES function.
- **Message Authentication Code (MAC)**: As mentioned previously, the MAC is not used in our solution. It is being discussed only for the sake of completeness. The purpose of the MAC is to verify that the encrypted message is not altered during transmission by means of an integrity tag. But, in our case, when the UE sends encrypted IMSI, the UE's ephemeral keys are not related or bound to the UE's IMSI. Therefore, any integrity tags will not serve any security purpose and hence not used.

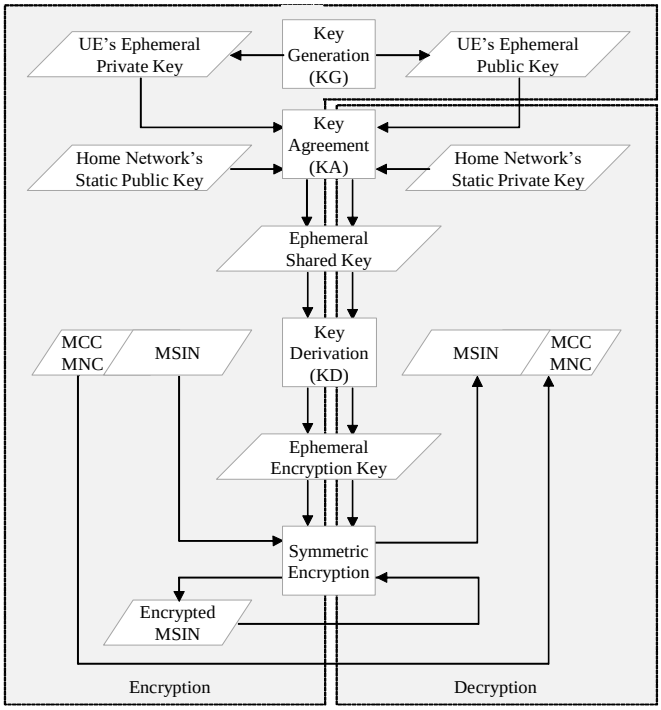


Figure 3-3 Using ECIES* for IMSI encryption/decryption.

The solution uses the combination of the OpenSSL implementation of elliptic curve named Curve25519 together with XOR as symmetric cipher. The curve operations for OpenSSL (e.g., KG and KA) were done according to the EVP_ functions for C language.

3.1.7.2 IMSI pseudonymization

The IMSI pseudonymization feature provides:

- A generation function for temporary or short term identifiers having the following properties:
  - It ensures that the probability of collisions between RTMSI allocated to different UEs over the entire Tracking Area is sufficiently small.
  - It ensures that the RTMSI are univocal random numbers over the entire Tracking Area to mitigate the chance of poor implementations for generating temporary subscription identifiers.

- A mechanism for triggering the RTMSI refresh and its periodic update. Two different approaches are possible:
    - The “push” model: in devices which are not able to host a pseudonym generation algorithm, the network generates the RTMSIs for the entire Tracking Area and maintains the state for each attached UE. The permanent or long term identity (IMSI) is communicated only once to the network in the Initial Attach Request by using one of the encryption mechanisms specified in the other features of this enabler. The RTMSI refers to the TMSI part of the GUTI. The one-time pseudonym is updated by the network after each usage (i.e., after being used in a message for UE identification).
    - The “synchronized” model: in devices which can host a pseudonym generation algorithm, the network and UE generate the RTMSIs, and the network maintains the state for each attached UE for the entire Tracking Area. The permanent or long term identity (IMSI) is communicated only once to the network in the Initial Attach Request by using one of the encryption mechanisms specified in the other features of this enabler. The one-time pseudonym is updated by both the UE and the network (in a synchronized way) after each usage (i.e., after being used in a message for UE identification).

3.1.8 Main interactions

This section presents the use cases and main interactions of the R2 features Home Network-centric IMSI protection and IMSI pseudonymization.

3.1.8.1 Use cases

3.1.8.1.1 Home Network centric IMSI protection

The use case chosen for the implementation of the feature refers to an Initial Network attach with IMSI encrypted with the public key of the Home Network. A simple use case architecture is shown in **Figure 3-4**. Similarly, a simple use case diagram is provided in **Figure 3-5**.

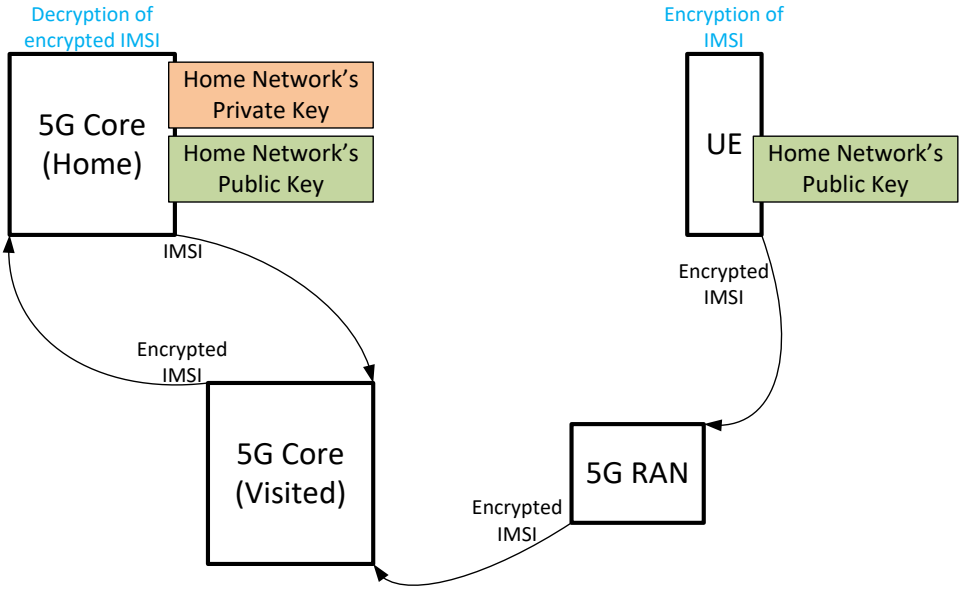


Figure 3-4: Use case architecture.

The main actors which use the system are the UE, the serving network and the home network. For both the figures, the descriptions given in the overview section apply, and further description can be found in components and interactions overview section.

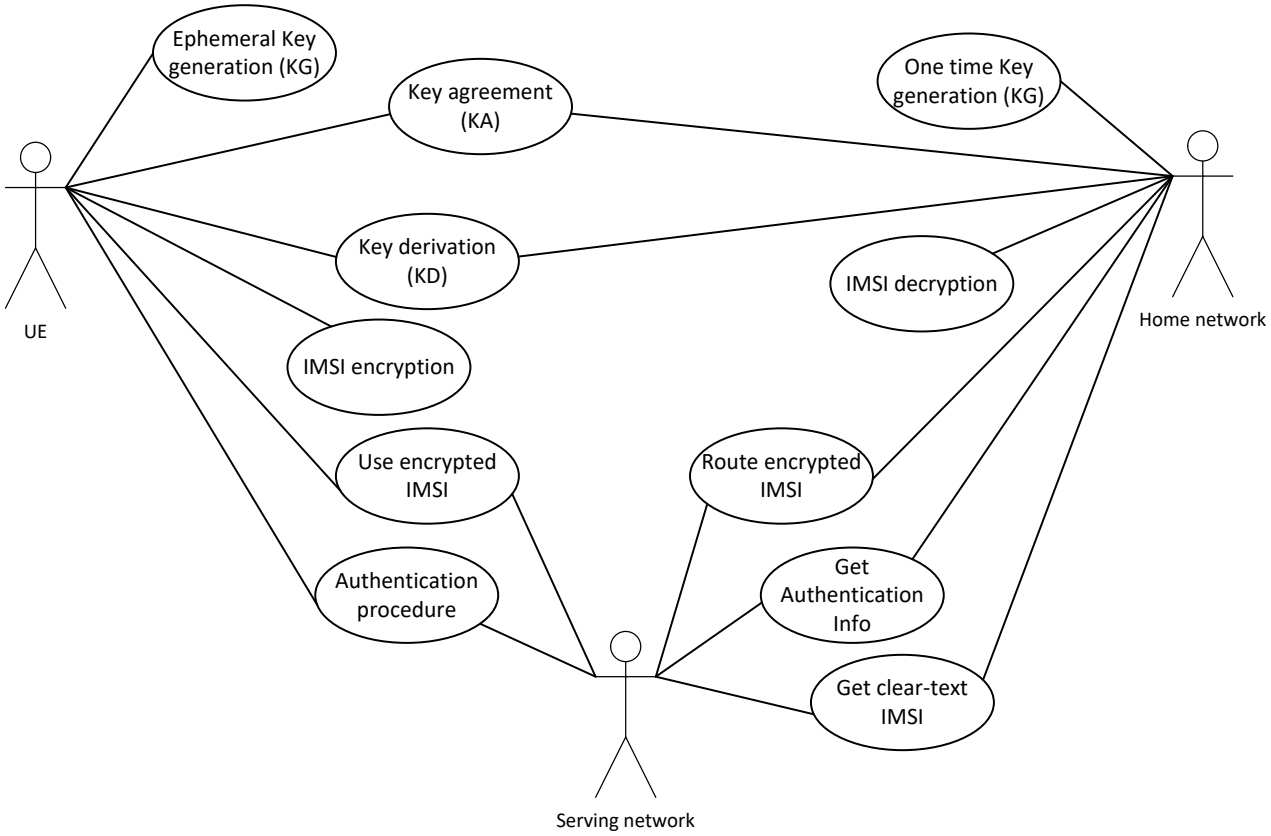


Figure 3-5: Use case diagram.

3.1.8.1.2 IMSI pseudonymization

The use case chosen for the implementation of the IMSI pseudonymization feature simulates the Attach procedures performed after an Initial Attach with encrypted IMSI. Identity Request and Responses can also be simulated. The use case architecture and interactions diagram is illustrated in **Figure 3-5**.

The Initial Attach can be performed with an encrypted IMSI by using one of the IMSI protection features of the present enabler, i.e., Encryption of Long Term Identifiers (from the previous release R1) or Home Network centric IMSI protection (R2). Upon a successful Initial Attach the 5G MME element and the UE use the IMSI pseudonymization feature's functions to generate the random one time RTMSI which will be used in subsequent Attach procedures with GUTI (containing the RTMSI).

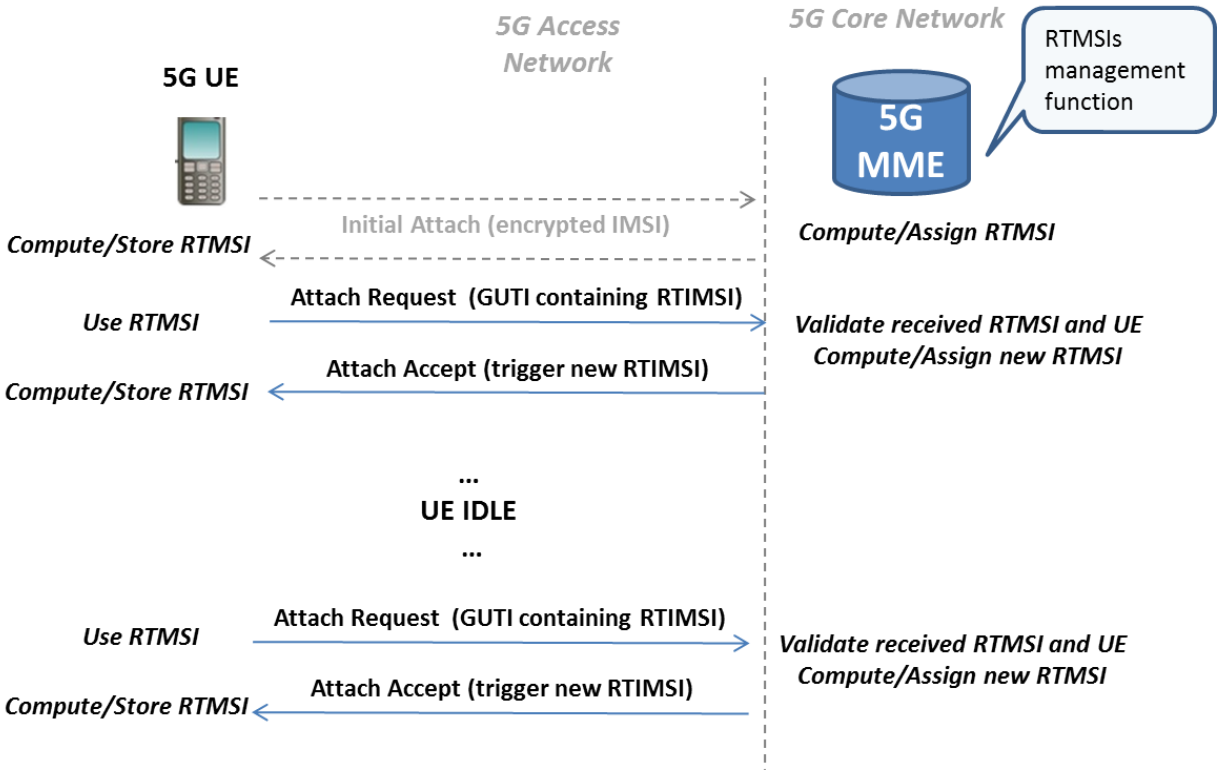


Figure 3-6: Use case architecture and interaction between components.

When the UE attaches again to the network (e.g., after an IDLE period) it uses the currently generated (or the current/new) RTMSI (which is the variable part of the GUTI used in the attach procedure).

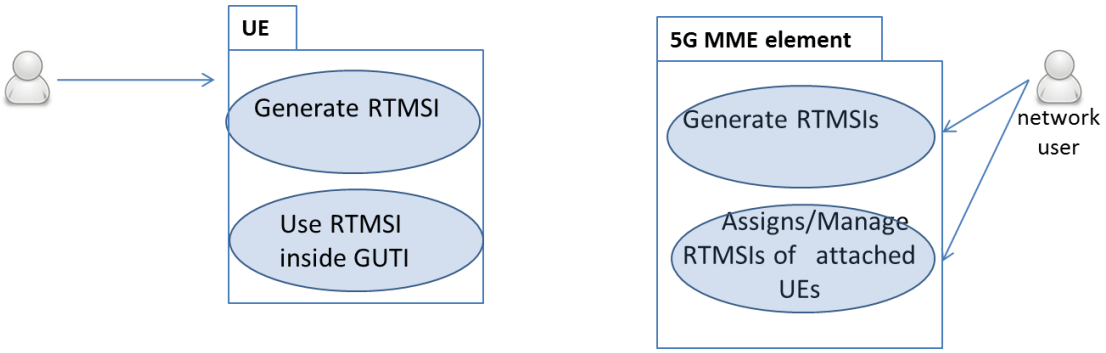


Figure 3-7: Use case diagram.

The use case diagram for IMSI pseudonymization is illustrated in **Figure 3-7**.

3.1.8.2 Components and interaction overview

This section presents the overview of the components and their interactions of the features Home Network-centric IMSI protection and IMSI pseudonymization.

3.1.8.2.1 Home Network centric IMSI protection

Interactions between various network elements are illustrated in **Figure 3-8**. The descriptions given in the overview section apply. Nevertheless, brief description of each steps follows the figure.

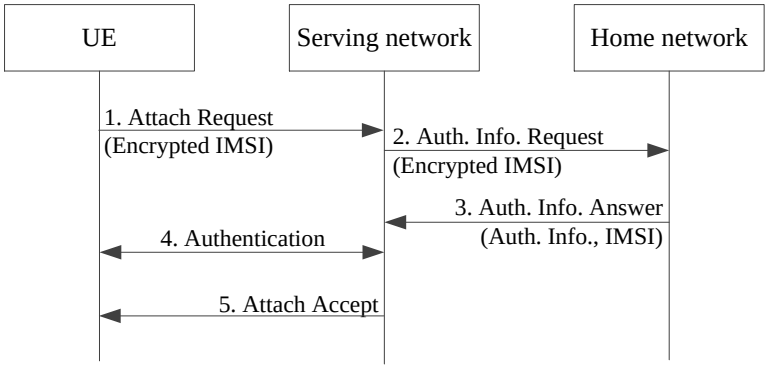


Figure 3-8: Interactions between components.

1. UE encrypts its IMSI and sends Attach Request to the serving network with the encrypted IMSI as its identifier. Note that the encrypted IMSI has MCC/MNC part in clear-text.
2. The serving network identifiers the UE's home network by using MCC/MNC part from Step 1 and request that home network for the authentication information. The encrypted IMSI from Step 1 is uses as the UE's identifier.
3. The home network decrypts the IMSI from the encrypted IMSI and fetches the corresponding authentication information. As a response of the Step 2, the home network sends the UE's authentication information along with the clear-text IMSI to the serving network.
4. The serving network performs authentication procedure with the UE.
5. If the authentication procedure in Step 4 succeeds, the serving network sends Attach Accept to the UE. Rest of the communication follows a normal.

3.1.8.2.2 IMSI pseudonymization

The architecture components are: the UE and the 5G MME element.

The feature's components are: the RTMSI generation function for the push model to be integrated inside the 5G MME or its 5G equivalent, and the RTMSI generation function for the synchronized model to be integrated inside both the 5G MME network element and UE. The 5G MME network element also has an RTMSI management function.

The interaction between the architectural components is shown in **Error! Reference source not found.**. The 5G Initial Attach with encrypted IMSI is implemented by the R1 feature Encryption of Long Term Identifiers. In the "push" model, upon a successful user identification and authentication (successful Initial Attach) the 5G MME generates the new GUTI (containing the one time RTMSI) and sends it to the UE in a protected Attach Accept message. The UE will use this GUTI (RTMSI) in the next Attach procedure (e.g., after an IDLE period).

Alternatively, the 5G MME element will only send some support information to the UE inside a protected Attach Accept message (e.g., a random value or a counter) and the UE and 5G MME element will compute the next RTMSI by their own (in a synchronized manner). At the next Attach the UE will use this computed value for identification.

3.1.9 Architectural drivers

3.1.9.1 High-Level functional requirements

The high-level requirement of the feature Home Network centric IMSI protection is the randomized encryption of the long-term identifiers, i.e., IMSIs, therefore the enabler must implement the encryption of the long-time identifier with random encryption and in such a way that only the authorized network entities can perform the decryption and obtain the identifier in clear text. The enabler should also provide means or indications on how to generate the keys used by the actors of the crypto system.
The high-level requirement of the feature IMSI Pseudonymization is the generation of univocal random pseudonyms for UEs of a given TA area (an area controlled by one logical 5G MME element) to be used in all network procedures instead of IMSIs, e.g., basically in all situation where TMSIs inside GUTIs are currently used in LTE networks.

3.1.9.2 Link to Security Architecture

This enabler is relevant to the following objectives from the draft Security Architecture D2.4:

- **O2.1** 5G must provide a security and privacy level higher or at least equal to the security and privacy level in 4G.
- **O2.3** 5G must provide solutions for security and privacy breaches identified in the previous mobile network generations such as the IMSI and IMEI unauthorized tracking or the denial of service provoked by the unsecured mobility messages (i.e., TAU messages).
- **O7.1** 5G must support traditional UICC/USIM based management of subscriber credentials
- **O7.4** 5G systems and components must provide strong mutual authentication and authorization
- **O9.1** 5G systems must comply with regulatory aspects, including those related to Lawful Intercept, user privacy, and customer notification of security breaches.
- **O9.2** If required by local regulation, 5G infrastructure operator must have means to demonstrate their provided level of security.
- **O9.3** All 5G security features must be compatible with local regulation.

The mapping onto the 5G architectural domains and strata defined in the project's deliverable D2.4 is illustrated in the two figures below (**Error! Reference source not found.**, **Figure 3-10**). Note that, on the UE side, it is not necessary that both USIM and ME will require implementation changes. Both IMSI encryption features of the enabler may be implemented either on USIM or ME.

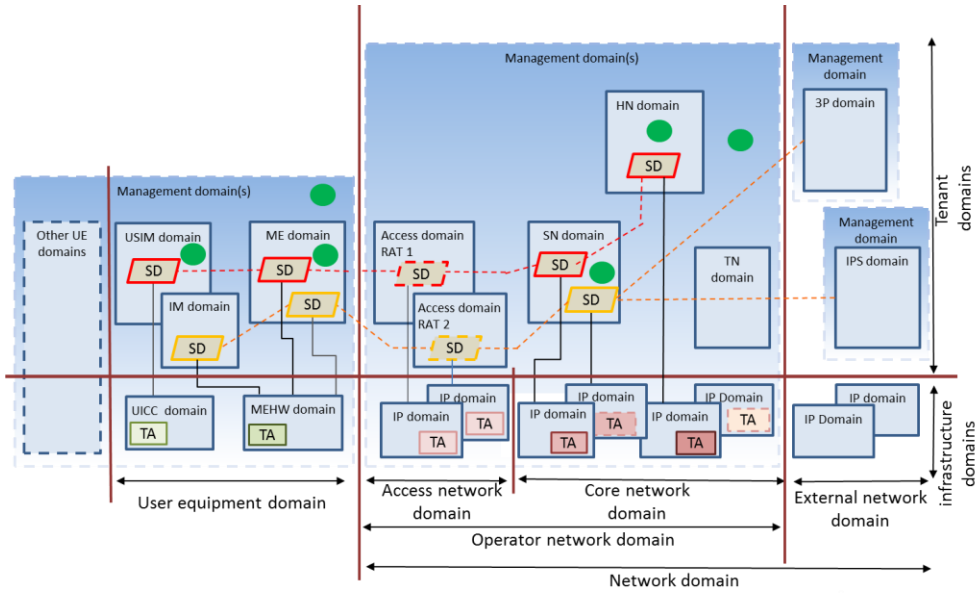


Figure 3-9 Mapping of the Enhanced Privacy enabler to 5g architectural domains.

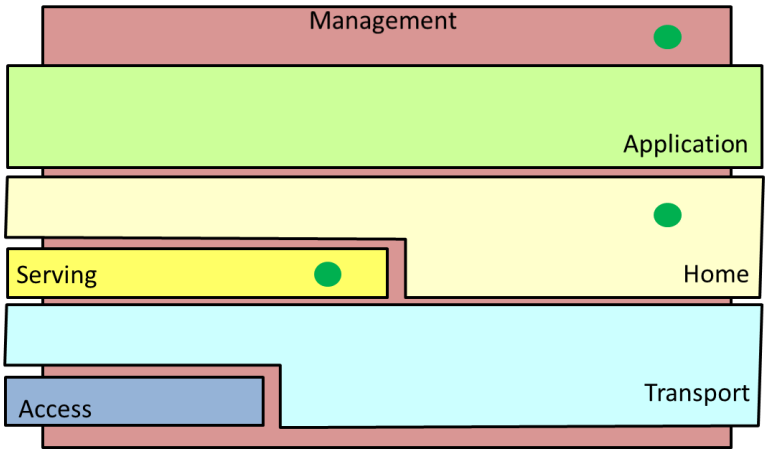


Figure 3-10 Mapping of the Enhanced privacy enabler on the architectural strata.

3.1.9.3 Quality attributes

A product implementing one or several of the features described in this enabler should be evaluated primarily based on the following quality attributes: performance efficiency, security (confidentiality), anonymity and maintainability.

3.1.9.4 Technical constraints

None.

3.1.9.5 Business constraints

None. The features can be applied in a stand-alone manner by an MNO.

3.1.10 Detailed specifications

This section contains the detailed open specifications of the features Home Network-centric IMSI protection and IMSI pseudonymization.

3.1.10.1 Home Network centric IMSI protection

3.1.10.1.1 Introduction

This enabler does not provide its own API because the OpenSSL library already provides all the implementation that are required to encrypt/decrypt the IMSI based on ECIES* scheme.

3.1.10.1.2 Conformance

Not applicable.

3.1.10.1.3 Enabler Component Specifications

Not applicable.

3.1.10.1.4 API Specifications

The enabler does not provide an API.

3.1.10.1.5 Functions, messages and commands

This section maps the functions in OpenSSL to the ECIES* functions.

The EVP_PKEY_keygen_init() and EVP_PKEY_keygen() functions are used for generating keys (ref. Key Generation function in ECIES*).

Similarly, the EVP_PKEY_derive_init(), EVP_PKEY_derive_set_peer(), EVP_PKEY_derive() functions are used for deriving a shared key (ref. Key Agreement function in ECIES*).

The SHA256() function is used for deriving an encryption/decryption key for encrypting/decrypting IMSI (ref. Key Derivation function in ECIES*).

Finally, the standard XOR provided by C language (i.e., ^) is used for encrypting/decrypting IMSI (ref. Symmetric Encryption/Decryption in ECIES*). Note that AES_set_encrypt_key()/AES_set_decrypt_key() and AES_encrypt()/AES_decrypt functions can also be used for the same.

3.1.10.1.6 Signalling

Not applicable.

3.1.10.1.7 Library Specification

The enabler does not provide a library.

3.1.10.1.8 Examples

Not applicable.

3.1.10.2 IMSI pseudonyms

3.1.10.2.1 Introduction

This section contains the detailed specification of the function(s) which can be integrated on 5G MME network components and/or the UEs of the adopting network in order to generate temporary subscriber identifiers RTMSIs by means of the chosen random pseudonym generation scheme(s).

A discussion on the 5G GUTI format and required changes is also provided.

3.1.10.2.2 Conformance

For an implementation to be conformant it should implement specifications as stated herein.

3.1.10.2.3 Enabler Component Specifications

The cryptographic component that implements the IMSI pseudonymization feature is composed of two RTMSI generation functions corresponding to the two generation models/approaches described in the previous section **Error! Reference source not found.**. The values returned by these functions are meant to fill in the M-TMSI field of the 5G GUTI.

In LTE the GUTI is constructed using the following fields (3GPP TS 23.003 [6]):

GUTI = MCC (12 bits) || MNC (12 bits) || MMEGI (16 bits) || MMEC (8 bits) || M-TMSI (32 bits).

The first most significant 48 bits uniquely identify the MME that assigned the GUTI also known as GUMMEI. Note that MCC || MNC corresponds to the serving network, and in roaming do not correspond to the MCC/MNC part of IMSI of a given UE. Furthermore, M-TMSI cannot have a value with all 32 bits equal to 1 and the 2 MSBs of the M-TMSI have special meaning. As a consequence, there are 30 bits available for any MME for assigning unique identifiers to the UEs that are served by the MME.

Therefore, in 5G only the M-TMSI part which do not have special meaning can be updated at the NGC and/or UE side, e.g., 30 bits of the M-TMSI in the current LTE system. For routing purposes, the NGC may also need to transfer one or more other parts of the identifier to the UE, e.g., MCC, MNC, MMEGI, MMEC and 2 bits of M-TMSI in the current LTE system.

The Pseudonymization feature focuses only on the generation of the 30-bits RTMSIs. Furthermore, the feature also provides the possibility to generate RTMSIs longer than 30 bits in case this becomes allowable by 5G standards.

The generation of the random temporary identifier has to ensure that the probability of collisions between short-term identifiers allocated to different UEs by the same 5G MME is sufficiently small. Should there be collisions, the NGC will not be able to tell the two colliding UEs apart and would have to apply a recovery mechanism, e.g., requesting the UE long-term identifier (IMSI).

3.1.10.2.3.1 The synchronized model

Let RTMSI denote a dynamic pseudonym to be used as a temporary subscriber identifier for a given UE by a given MME and it corresponds to the 30-bit long M-TMSI part of the LTE GUTI. Let $RTMSI_{new}$ and $RTMSI_{old}$ denote the new and old values of RTMSI for a given UE and 5G MME. KDF(K, S) denotes any secure KDF used by a given UE and 5G MME to independently compute the $RTMSI_{new}$ value from known common secret K and public parameters S.

The K parameter is the KASME key (256-bit) or any secret value derived from it. S is obtained by concatenating the GUMMEI, the IMSI, the $COUNT_{session}$ and the $RTMSI_{old}$, where $COUNT_{session}$ is the current cumulative number of signalling messages exchanged between the 5G MME and UE in a given session for a fixed KASME.

Consequently, the update function of pseudonyms is defined by:

$$RTMSI_{new} = KDF(KASME, GUMMEI \,||\, IMSI \,||\, COUNT_{session} \,||\, RTMSI_{old}),$$

Every time that a new KASME is generated for the UE, the $COUNT_{session}$ is reset and also the $RTMSI_{old}$ which can take any value shared in common by MME and UE (e.g., a globally fixed constant). When the UE changes MME, the old MME sends to the new MME the KASME together with the value of $COUNT_{session}$.

If the output bit-length of KDF is bigger than the bit-length n=|RTMSI| of RTMSI, then only n bits are extracted from the output of KDF (e.g., the most significant ones). The generated pseudonyms will be computationally unpredictable if KDF is computationally secure as a MAC, that is, if, for any unknown key K, it is infeasible to generate KDF(K, S) output, partially or as a whole, for any new string S, given the outputs for any number of known strings S.

The generated pseudonyms should avoid at any time collisions between pseudonyms for any given MME.

The probability of collisions is determined by the birthday paradox, that is, there will be at least one collision with a significant probability if the number of RTMSI's at a given time for a given 5G MME is on the order of $2^{n/2}$. If there are m such pseudonyms and $m \leq 2^{n/2}$, then the probability that they are all different is approximately $Pr\{\text{no collisions}\} \approx \exp(-m^2/2^{n+1})$.

In LTE, as n=30, the collision probability is too high if $m \approx 2^{15}$, which seems to be possible in practice. Consequently, to be on the safe side in 5G, the effective bit-length of M-TMSI should be increased from 30 to 64 or more (e.g., 80), while keeping the remaining parts of GUTI.

For unlinkability, RTMSI's should be generated and used as one-time pseudonyms, i.e., they should be updated each time a NAS message containing RTMSI is exchanged between UE and MME (e.g., inside the NAS procedures that make use of GUTI for identification purposes). Since in the synchronized generation model, RTMSI is independently computed by UE and 5G MME, it shall not be included in the LTE protocol messages used today for conveying the GUTI.

3.1.10.2.3.2 The "push" model

In the push generation approach, the RTMSIs are generated only by the network 5G MME.

Let ENC(K, S) denote a secure block cipher encryption function under secret key K and input S.

Here K is a MME-specific 256-bit secret key, denoted as KMME, necessary for 5G MME to unilaterally compute a new pseudonym $RTMSI_{new}$ for a given UE in a way that $RTMSI_{new}$ is different from all active RTMSIs for that 5G MME.

The update of pseudonyms is then defined by the counter mode of the block cipher:

$$RTMSI_{new} = ENC(KMME, COUNT_{new}),$$

where $COUNT_{new}$ is expressed in an n-bit format, n denoting the block size of ENC. Since ENC is a reversible function, all RTMSIs generated are different, as long as the counter values are not repeated. This way all 2^n different RTMSIs can be generated and used over time, thus avoiding the birthday paradox. RTMSIs can be generated in parallel in real time or pre-computed. Consequently, the technique will work without collisions as long as the number m of RTMSI's at any given time for a given MME is not bigger than 2^n. The generated pseudonyms will be computationally unpredictable, apart from the fact that they are all different, if ENC is computationally secure as a block cipher, on the condition that at most 2^n pseudonyms are generated altogether from a given KMME ($m <= 2^n$).

The push generation of pseudonyms defined as above can be generalized in order to address the two issues: the choice of the bitsize n of RTMSI and the refreshing of the secret key KMME. Namely, n should be equal to the block size of ENC which is typically 64 or 128 bits. To allow more flexibility, we need the so-

called format-preserving encryption FPE, since truncating the block would destroy reversibility and hence the collisions might occur. The update expression then becomes:

$$RTMSI_{new} = FPE(KMME, COUNT_{new}).$$

We can thus also allow n=30 as in LTE, since it is extremely unlikely to have $m > 2^{30}$. For FPE algorithm, one may use the recent NIST standard [5] in which FPE is defined as a mode of operation of block ciphers, using a Feistel-like iterative construction in which the block cipher is used in the round functions, and AES is recommended for this purpose.

To allow refreshing of KMME, e.g., periodically, in order to deal with 2^n not being sufficiently big or with possible key compromise, the same update function is used and one fixed bit of RTMSI is dedicated to flip when KMME changes. This way collisions are avoided, while the impact on subscriber privacy/anonymity is negligible.

In the push generation of dynamic pseudonyms each new pseudonym $RTMSI_{new}$ is transmitted from 5G MME to UE encrypted by using the existing NAS layer encryption key.

For un-linkability, RTMSI's are generated and used as one-time pseudonyms, i.e., they should be updated each time a message containing a RTMSI is exchanged between UE and MME. In the push generation, since each $RTMSI_{new}$ needs to be transmitted from MME to UE, to be then used by the UE in subsequent messages sent to MME, it may be necessary to use the same RTMSI in more than just one message. For example, in the Paging procedure initiated by the network, the network uses the last RTMSI it sent to the UE in the Paging request and the correspondent UE uses the same RTMSI in the attach request afterwards. However, this has minimal impact on subscriber privacy/anonymity.

3.1.10.2.4 API Specifications

N/A as a Linux shared library exposing the two functions detailed below will be provided.

3.1.10.2.5 Functions, messages and commands

The two functions provided by the IMSI Pseudonymization feature described in the feature's component specification are detailed in the following tables.

Function Name	*generate_rtmsi_synchro*	
Function Definition/Description	Generates a new rtmsi from a set of input parameters in the "synchronized" model. It should be integrated o both UE and 5G MME (or equivalent NGC element).	
Function input arguments		
Name	***Type***	***Description***
kdf	*char	Indicates the key derivation function to be used, the default value is HMAC-SHA256.
kasme	*char	The key used for rtmsi generation. This is the 5G equivalent of the KASME key of the UE, the 256-bit key derived by both UE and MME after successful authentication.
gummei	*char	The GUMMEI value globally identifying the 5G MME. Is a 48 bits globally unique identifier of

		the MME composed of MCC || MNC || MMEGI || MMEC.
Imsi_len	int	Is the length of the International Mobile Subscriber Identifier.
rtmsi_old	integer	The last rtmsi value of the given UE. In the first generation can be 30 bits of the MSIN part of the IMSI.
Function output		
Name	***Type***	***Description***
rtmsi	integer	The new rtmsi value for the given UE (30 bits).

Function Name	*generate_rtmsi_ push*	
Function Definition/Description	Generates a new rtmsi from a set of input parameters in the "push" model. It should be integrated o the 5G MME (or equivalent NGC element).	
Function input arguments		
Name	***Type***	***Description***
encf	*char	Indicates the encryption function to be used, the default value is the format preserving encryption algorithm FF1.Encrypt based on AES [5].
kmme	*char	The key used for rtmsi generation. This is a 5G MME specific 256-bit secret value.
tweak_len	int	Is the length of the tweak value required by some FPE algorithms. It may be the IMSI of the user for whom the rtmsi is currently generated. According to Appendix C of [5] in our case the tweak can be left empty (length of tweak = 0) since the plaintext to be encrypted is always different (COUNT is a global variable for the MME).
Function output		
Name	***Type***	***Description***
rtmsi	integer	The new rtmsi value for the given UE (30 bits).

3.1.10.2.6 Signalling

All functions described in the components specifications are used to produce the values of the GUTI field included in some of the 5G NAS layer's signalling messages, like the 5G NAS messages equivalent to the LTE messages Attach Request with GUTI, Identity Response, Paging Request with GUTI, GUTI Reallocation.

3.1.10.2.7 Library Specification

The two functions detailed in 3.1.10.2.5 will be provided by a Linux shared library.

3.1.10.2.8 Examples

An example on where and how the functions of the library are effectively used is given by our system depicted in **Figure 3-2** and **Figure 3-6**.

3.1.11 Re-utilised Technologies/Specifications

The re-utilised technologies are: OpenSSL [7] and Botan [8].

3.1.12 References

[1] 5G ENSURE Deliverable D3.1, "5G-PPP security enablers technical roadmap (early vision)"

[2] 5G ENSURE Deliverable D3.5, "5G-PPP security enablers technical roadmap (update)"

[3] 5G ENSURE Deliverable D2.4, "5G-PPP security architecture"

[4] 3GPP TS 33.402, "3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses", http://www.etsi.org/deliver/etsi_ts/133400_133499/133402/13.00.00_60/ts_133402v130000p.pdf

[5] NIST Special Publication 800-38G, *"Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption,"* Mar. 2016. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38G.pdf

[6] 3GPP TS 23.003, "Numbering, addressing and identification", Release 14, 15/03/2017

[7] OpenSSL, Cryptography and SSL/TLS Toolkit, https://www.openssl.org/

[8] Botan, Format Preserving Encryption, https://botan.randombit.net/manual/fpe.html

3.2 Device Identifier Privacy Open specifications

3.2.1 Preface

3.2.2 Status

The version 1.0 of this enabler has been implemented, tested, and evaluated on the testbed. Version 2.0 of this enabler is currently under development.

3.2.3 Copyright

The copyright will be attributed to existing source code authors and those authors who contribute to the development of the enabler code, which currently includes the University of Oxford ([UOXF]).

3.2.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

3.2.5 Terms and definitions

This section comprises a summary of terms and definitions used by this enabler.

- **ACD** Address Conflict Detection [5]
- **ARP** Address Resolution Protocol [4]
- **DHCP** Dynamic Host Configuration Protocol [1][2]
- **DNA** Detection of Network Attachment protocol [3]
- **MAC** Media Access Control address
- **OUI** IEEE Organizationally Unique Identifier
- **SSID** Service Set Identifier – a human readable identifier for a Wireless LAN

3.2.6 Overview

This enabler aims to utilise state-of-the art obfuscation techniques on the user's device, offering *Privacy Enhanced Attachment,* which provides protection against device identity (and also potentially subscriber identity) disclosure and unauthorized device/user tracking. The main focus is to offer improved privacy protection of the identifiers for devices, over IP-based networks, to access 5G services such as via non-3GPP access methods, or for application level interactions. Additionally, it provides enhanced device identity privacy for access to third party Internet-based resources when utilising 5G-based authentication schemes (e.g. EAP-AKA for WiFi).

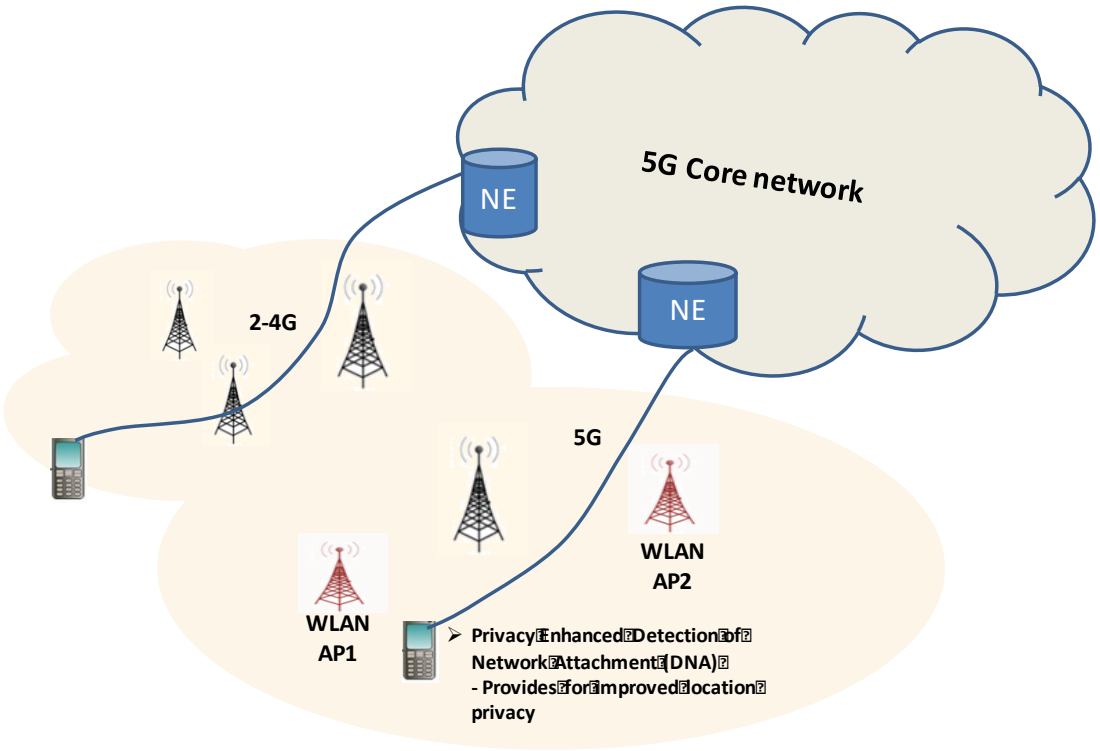


Figure 3-11: Privacy Enhanced Attachment

3.2.6.1 Features developed in R1 (Reminder)

The enabler aims to enhance location privacy as a device roams from one IP-based network to the next, primarily in the context of 5G non-3GPP access. It provides for enhanced privacy with respect to the

leakage of identifiers for previous points of attachment. Specifically, in R1, two new mechanisms were introduced to protect the privacy of the device.

- **Randomised ordering**
  - This mechanism will provide for randomisation of candidate link layer (MAC) addresses to enhance the location privacy with respect to the time-based analysis of the device's movement patterns.
- **Dummy addresses**
  - This mechanism will allow for the introduction of dummy MAC addresses into the attachment phase to enhance location privacy with respect to location identification and time-based analysis of the device.

3.2.6.2 Target features for R2

The R2 release aims to enhance the mechanisms developed in R1. Specifically, we are developing approaches which will provide for a pre-analysis phase of the address anonymity metrics which would allow for dynamically optimised choice of address randomisation and dummy addresses.

- **Pre-analysis phase of the address privacy metrics**
  - The set of existing candidate link layer (MAC) addresses for DNA will be analysed before use to ascertain their privacy metrics
- **Dynamically optimised choice of address randomisation and dummy addresses**
  - The choice of random addresses may be influenced by a range of factors including inferred ownership (e.g. from OUI), location, sensitivity, user privacy settings.

3.2.7 Basic concepts

3.2.7.1 Dynamic Host Configuration Protocol (DHCP) concepts

When a device has connected at the link-layer then it will begin the process of configuring an Internet Protocol address. If the device is set up to automatically obtain an address this is typically achieved using the Dynamic Host Configuration Protocol (DHCP) which is specified for both IPv4 [1] and IPv6 [2].

The mode of operation of DHCP depends on whether the device has determined if it has connected to a new network, in which case it enters the INIT state and attempts to obtain a new address, or if the device has determined it has reconnected to a previous network, then the DHCP client enters the INIT-REBOOT state and attempts to reuse an unexpired lease. The problem is that with mobile devices there are a limited number of cases where one can be sure whether a device is on a new or existing network. With WiFi a device can ascertain the SSID of the network which can help in determining whether it has been previously visited. Furthermore, if the device has a stored WiFi security association then it can be reasonably sure this is indeed a previously visited network. Thus, most devices will default to using the INIT state. However, as we see in the next section if a device makes use of the DNA protocol it can potentially omit the full INIT state transition and utilise an expedited INIT-REBOOT phase.

On connecting to the network, in the INIT state, a node will broadcast a DHCPDISCOVER packet which elicits a DHCPOFFER from the DHCP server, containing an offered IP address. The device then generates a DHCPREQUEST containing the offered address, which the server should acknowledge with a DHCPACK. Once the request is acknowledged the device should perform an Address Conflict Detection (ACD) [5] before it attempts to use its requested address. ACD basically consists of sending out a series of 'ARP

Probes' for its own address to check if another node is using this address. If no responses are seen then the node follows this up with a series of 'ARP Announcements' claiming the address, after which it starts using the address.

If a node determines it is on the same link as before then it will initiate the INIT-REBOOT state and attempt to reuse an existing unexpired DHCP lease by sending out a DHCPREQUEST for it. Once the server confirms it with a DHCPACK then the node can start using the address, without having to perform ACD. However most mobile devices do not employ INIT-REBOOT directly [4] as there aren't any reliable ways to ascertain whether the device has reconnected to the same network.

3.2.7.2 Detection of Network Attachment (DNA) concepts

Although most devices employ DHCP to obtain an IP address the protocol can take up to seven seconds to complete the address assignment operation (when the protocol incurs the full range of specified delays due to conflict detection or collisions etc.). In several cases this time can be significantly reduced by employing an enhancement protocol known as Detection of Network Attachment (DNA) [5], which is widely deployed in Apple devices running iOS and OSX, and in Google devices running ChromeOS. Specifically, DNA speeds up the connection re-establishment to networks that the device has previously visited for which the device has a pre-existing DHCP lease. It does this by verifying that it has reattached to a known network for which it can quickly reuse an existing lease without the need to perform address conflict detection.

3.2.8 Main interactions

3.2.8.1 Use cases

Here we detail the use case diagram in **Figure 3-12** which provides an overview of the basic use case and entities involved in the enabler.

The steps involved are for the device to perform a privacy-enhanced IP address request and then for the system to provide an IP address. This is based upon DHCP services but the privacy services are provided at the attachment phase before a DHCP request is made.

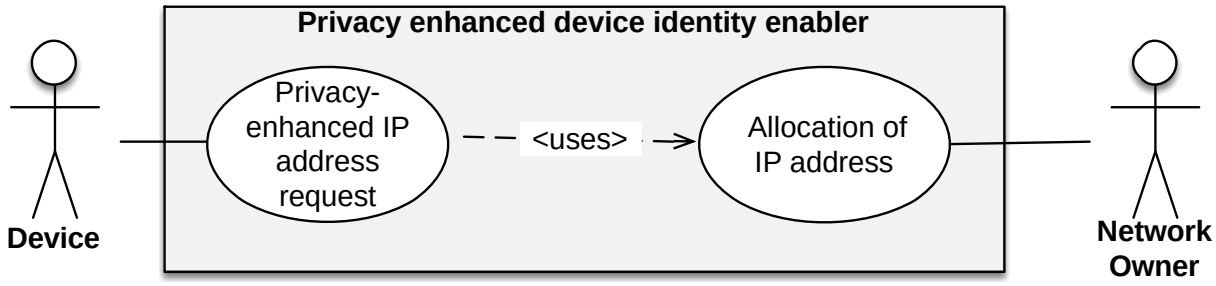


Figure 3-12: Use case diagram

3.2.8.2 Components and interaction overview

The basic components are detailed below in **Error! Reference source not found.**, with the enabler components highlighted – with the level of orange shading indicating the R2 enhanced components. The enabler operates within the system DHCP services and provides for privacy-enhanced operation.

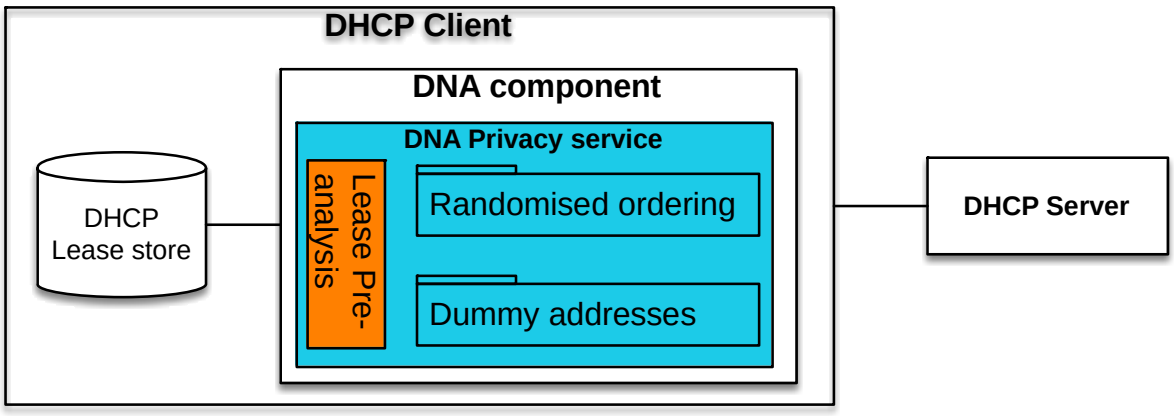


Figure 3-13: Device identifier privacy component overview

The DHCP client initiates a protocol exchange for each new connection and stores any unexpired leases in the DHCP lease store. On attachment to a new network when the client has active leases in the DHCP lease store the Detection of Network Attachment (DNA) protocol is enacted.

The sequence of events concerning the DNA privacy service are detailed by the sequence diagram in **Figure 3-14Error! Reference source not found.**. The protocol operates by utilising stored information about prior unexpired DHCP leases that have not been released back to the DHCP server. In particular, for each lease the MAC and IP addresses of the router, and the assigned IP of the client may be used to perform a 'reachability test'. This test is performed by a device once it has connected at the link-layer, provided it has a suitable set of candidate leases and is not prohibited by the configuration of authenticated DHCP. When used on WiFi networks some systems, including iOS and OSX, attempt to use the SSID of the current network to filter candidate leases.

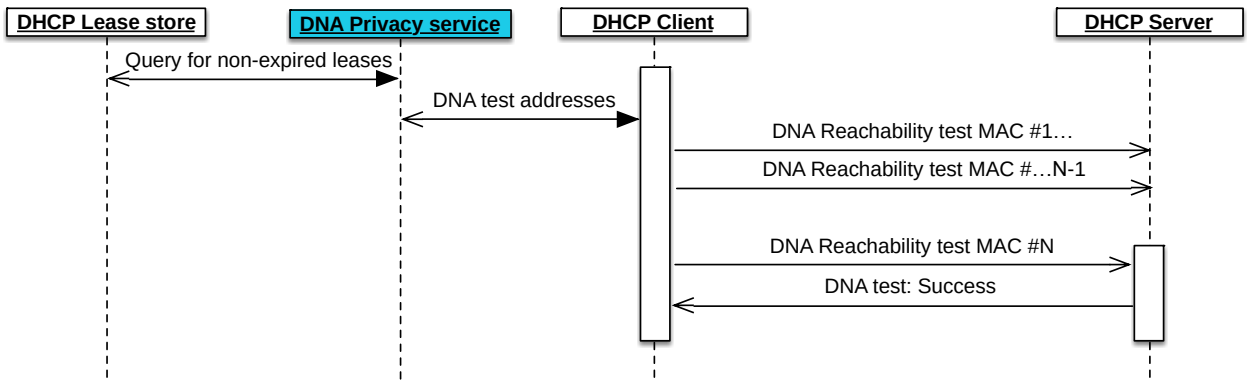


Figure 3-14: Device identifier privacy sequence diagram

The reachability test takes the form of a unicast ARP request packet, sent by the client, directed at the router's MAC address with the ARP's target protocol address set to the router's IPv4 address, and the sender protocol address field to its own candidate IPv4 address. The client includes its MAC address in the sender hardware address field and sets the target hardware address field to the null address. If a valid ARP Reply is received, the MAC address in the sender hardware address field in the ARP Reply is matched

against the target hardware address field in the ARP Request, and the IPv4 address in the sender protocol address field of the ARP Reply is matched against the target protocol address field in the ARP Request. If a match is found, then the host continues to use that IPv4 address, subject to the conditions of the matching stored DHCP lease.

3.2.9 Architectural drivers

3.2.9.1 High-Level functional requirements

The high-level functional requirements are:

- The user shall have attached to previous networks
- The user shall be attaching to IP based network
- The enabler shall provide for privacy-enhanced attachment
- The enabler shall provide for improved location privacy.

3.2.9.2 Link to Security Architecture

This enabler is relevant to the following objectives from the draft Security Architecture D2.4 [7] :

- **O2.1** 5G must provide a security and privacy level higher or at least equal to the security and privacy level in 4G.
- O2.3 5G must provide solutions for security and privacy breaches identified in the previous mobile network generations.
- **O3.3** 5G access security should allow for high efficiency in authentication and security set-up, supporting ultra-low latency services.

The enabler operates within the Mobile Equipment (ME) domain as indicated with a red dot in **Figure 3-15**.

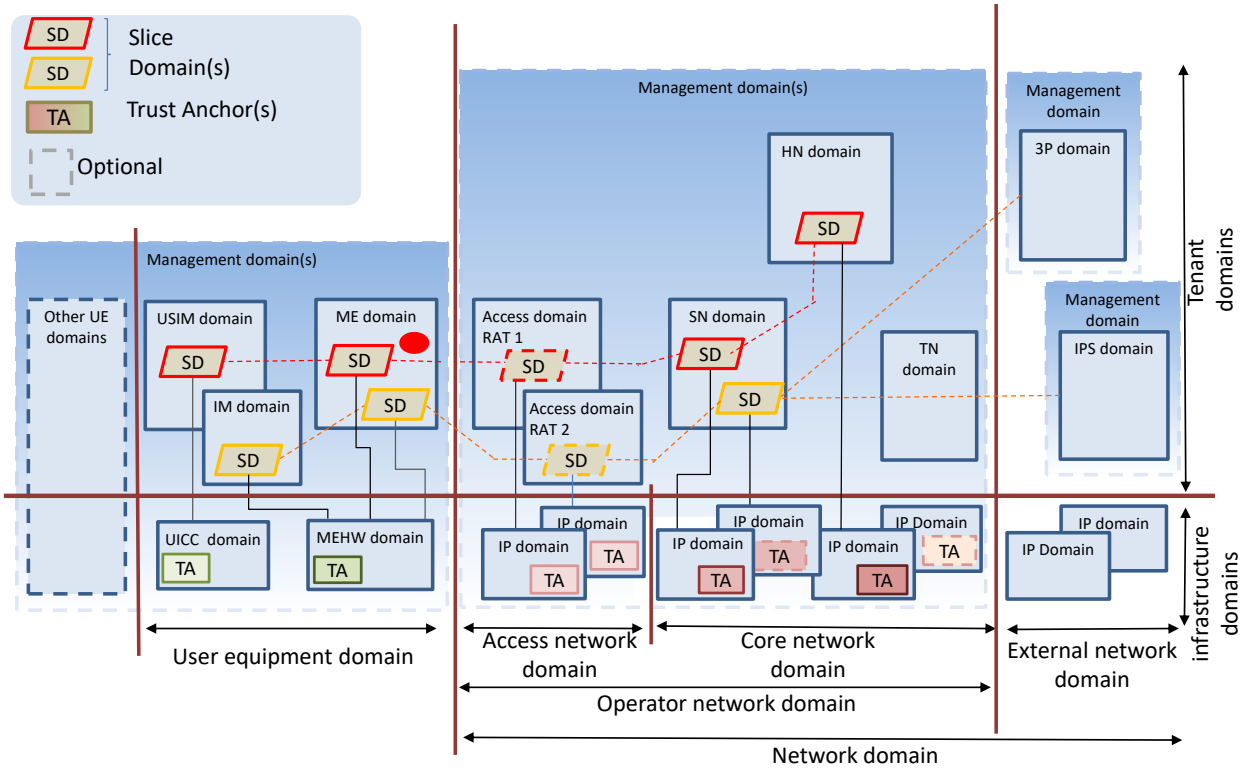


Figure 3-15 Domain placement for Device Identity Privacy enabler

The enabler functions within the transport strata as shown in **Figure 3-16.**

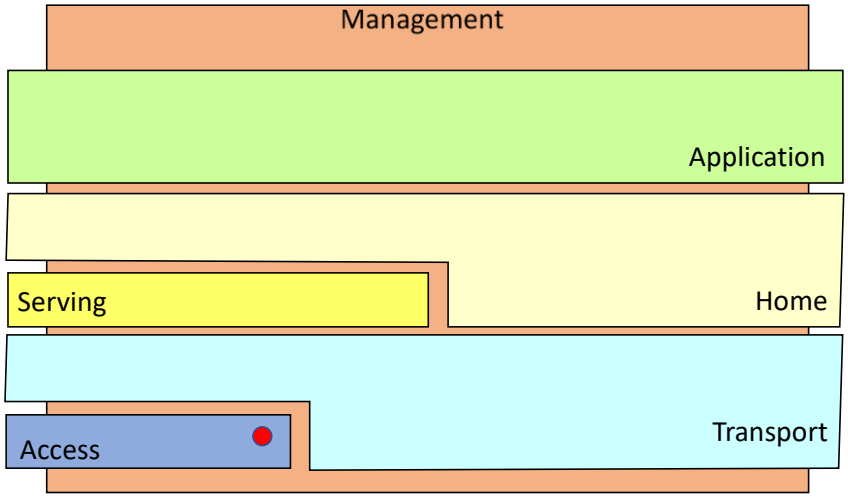


Figure 3-16 Strata placement for Device Identity Privacy enabler

3.2.9.3 Quality attributes

Compatibility: The privacy enhanced version of the DNA protocol aims to maintain compatibility with the standard version.

Security: The enabler aims to provide location privacy improvements for the device.

3.2.9.4 Technical constraints

The device needs to be capable of running POSIX compliant, preferably Linux, system which supports the use of a DHCP client.

3.2.9.5 Business constraints

None.

3.2.10 Detailed specifications

3.2.10.1 Introduction

This enabler aims to enhance location privacy as a device roams from one Internet-based network to the next, primarily in the context of non-3GPP 5G access. It provides for enhanced privacy with respect to the leakage of identifiers for previous points of attachment. Specifically, we build upon the two mechanisms developed in R1 to enhance their operation, further protecting the location privacy of the user.

- **Pre-analysis phase of the address privacy metrics**
  - The set of existing candidate link layer (MAC) addresses for DNA will be analysed before use to ascertain their privacy metrics
- **Dynamically optimised choice of address randomisation and dummy addresses**
  - The choice of random addresses may be influenced by a range of factors including inferred ownership (e.g. from OUI), location, sensitivity, user privacy settings.

3.2.10.2 Conformance

For an implementation to be conformant it should implement specifications as stated herein.

3.2.10.3 API specifications

There is no specific API for the enabler as it operates within the confines of an existing DHCP client and thus provides no external API interfaces.

3.2.10.4 Enabler component specifications

3.2.10.4.1 Pre-analysis of the candidate address privacy metrics

The set of existing candidate link layer (MAC) addresses for DNA will be analysed before use to ascertain their privacy metrics. We aim to examine the candidate addresses in terms of a range of potential metrics including their identifiability in terms of whether they may be easily geolocatable, or may be associated with privacy sensitive locations (e.g. home, medical centre, etc.), and their lease age.

This analysis will occur before the candidate addresses are utilised in the subsequent phases of the privacy enhanced DNA procedures defined in R1.

3.2.10.4.2 Dynamically optimised choice of candidate addresses

On each attachment, the choice of the selected candidate addresses may be dynamically optimised for privacy in terms of the ordering of the candidate address and in the choice of the dummy addresses.

The dummy addresses generation in R1 was based upon use of a previous candidate addresses with an added random component. In R2 we aim to improve upon the dummy address generation through the use of privacy metrics and entropy analysis. Furthermore, in terms of dummy address selection this includes

how many dummy addresses may be injected which can be optionally dependent upon how many candidate leases there are present or may be a fixed quantity.

With respect to address ordering, whilst in R1 we randomised the candidate address ordering using Fisher–Yates Durstenfeld shuffle [8], we plan to provide other potential ordering options which would include dependence upon pre-analysis privacy metrics or on the number of candidate leases. For example, we could ensure that no two consecutive leases addresses maintained their neighbours after randomisation.

These techniques aim to enhance the privacy of a user device against localised attacks on their location privacy. Their efficacy will be assessed based upon different threat models from a single adversary through to a pervasive attacker.

3.2.11 Re-utilised Technologies/Specifications

We aim to build upon existing DHCP client implementations to provide for the enabler service.

3.2.12 References

[1] R. Droms, “Dynamic Host Configuration Protocol. RFC 2131 (Draft Standard),” IETF, 1997.

[2] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins and M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6). RFC 3315 (Proposed Standard),” IETF, 2003.

[3] B. Aboba, J. Carlson and S. Cheshire, “Detecting Network Attachment in IPv4 (DNAv4). RFC 4436.,” IETF, 2006.

[4] D. Plummer, “Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware, STD 37, RFC 826,” IETF, 1982.

[5] S. Cheshire, “IPv4 Address Conflict Detection. RFC 5227 (Proposed Standard),” IETF, 2008.

[6] I. Papapanagiotou, E. M. Nahum and V. Pappas, “Configuring DHCP leases in the smartphone era,” in *ACM Conference on Internet Measurement Conference, IMC ’12*, New York, 2012.

[7] 5G ENSURE Deliverable D2.4, “Security Architecture (draft)”

[8] Durstenfeld, R. (July 1964). "*Algorithm 235: Random permutation*". Communications of the ACM. **7** (7): 420. *doi:10.1145/364520.364540*

3.2.13 Acknowledgements

We would like to acknowledge the work of Joss Wright and the UK EPSRC Being There project (grant EP/L00416X/1).

3.3 Device-based Anonymization Open specifications

3.3.1 Preface

This enabler aims to provide anonymization techniques on the user’s device, offering protection against disclosure of sensitive information stored mainly on the SIM. The privacy/anonymization configuration (or

profile) should be directly controlled by the user, who can activate different anonymization profiles stored on the device. The user's device will host a configuration tool which will enable the user to activate and configure his/her privacy profile.

3.3.2 Status

Initial versions (1.0) of these two new components is currently under development.

3.3.3 Copyright

3.3.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

3.3.5 Terms and definitions

APK	Android application package is an archive file format used to distribute and install mobile applications and middleware.
IMSI (MCC, MNC, MSIN)	The International Mobile Subscriber Identity is a unique number that identifies the user of a cellular network. An IMSI is usually 15 digits in length. The first 3 digits represent the MCC (mobile country code) and are followed by the MNC (mobile network code) which is either 2 digits (European standard) or 3 digits (North American standard). The remaining digits are the MSIN (Mobile Subscription Identification Number), that is a unique identifier within the network's customer base.
FPE	Format Preserving Encryption
FFX	FFX is a mode of operation for format-preserving encryption (FPE). FPE algorithms work in such a way that the cipher text is represented in the same format as the input. In the device-based anonymization the format of the input and of the cipher text refers to the number of digits.

3.3.6 Overview

Depending on the information to be anonymized, the device implements a specific anonymization algorithm at the lowest possible layer in the device OS stack, and offers the means to the user to activate and deactivate the anonymization. As illustrated in **Figure 3-17**, whenever a user space application requires access to SIM data protected by an active privacy profile/configuration, the request will be managed by a privacy provider, which will return an anonymized version of the sensitive data to the caller, therefore activating this specific data protection with the configured anonymization algorithm. Therefore, the requesting application will obtain the anonymized piece of data instead of the original one.

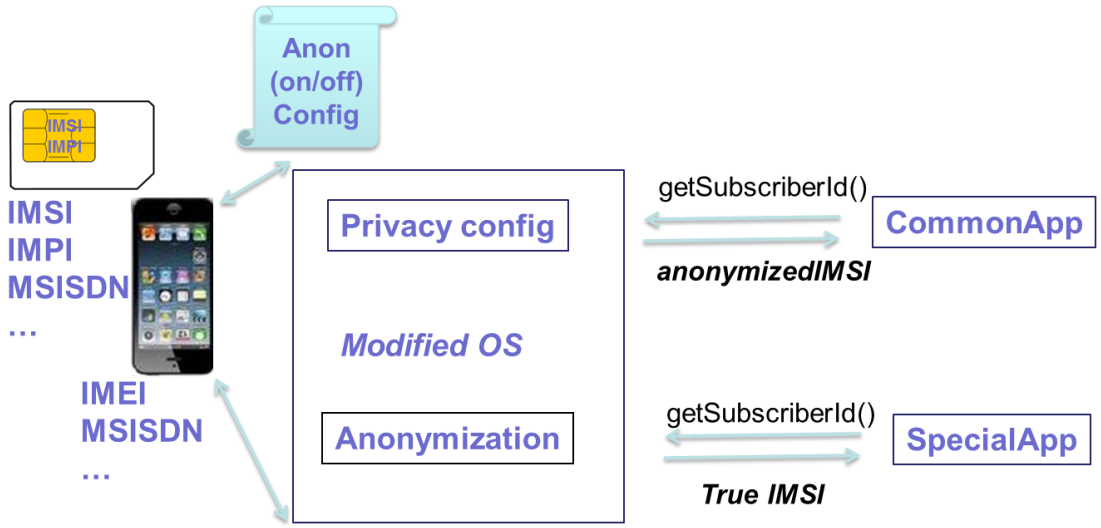


Figure 3-17 Device-based Anonymization.

The enabler aims to avoid the disclosure of sensitive information to all or only to selected user space applications. In order to do so it implements two basic components (features):

- A format preserving anonymization algorithm which provides the anonymization function for all data received in input (e.g., the IMSI), with the preservation of the input data format. The algorithm is embedded in the operating system code, as low as possible in the kernel stack.
- Privacy Configuration: the mediator between the user and the anonymizing device, which gives the user a means to enable the anonymization capabilities to protect sensitive data and avoid their disclosure to user space applications. Such anonymization capabilities can be further customized by white-listing single applications and by allowing them to access the sensitive data in clear text.

3.3.7 Basic Concept

Android applications that are granted the READ_PHONE_STATE permission have unconditional access to sensitive data about the user's device. They are allowed to read information pertaining the state of the mobile device as well the phone number of the device, current cellular network information (including the subscriber's identity IMSI), the status of any ongoing calls, and a list of any accounts registered on the device.

The current version of the enabler empowers the user to protect his own privacy by means of anonymization. The enabler is based on a custom release of the Android operating system build from the Cyanogenmod 13.1 milestone sources for the Samsung Galaxy SII device and provides a system-wide device-based data anonymization which supports customized per-application security profiles.

The method returning the IMSI to the user space applications is intercepted in the TelephonyManager system service component and the returned value is encrypted using a format-preserving algorithm. The encryption is based on a secret key generated using AES-256 that is refreshed at every device boot.

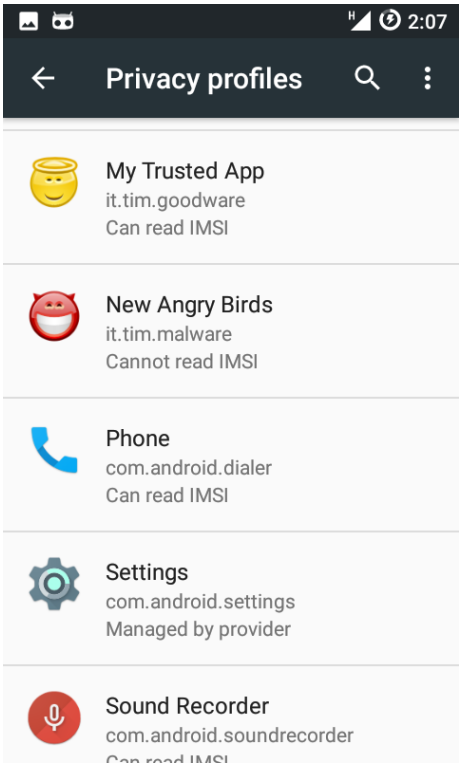


Figure 3-18 Defining privacy profiles for the installed apps.

3.3.8 Main interactions

3.3.8.1 Use Cases

The main use case chosen for the current implementation of the enabler refers to a scenario in which a Samsung Galaxy SII device has been flashed with the enhanced anonymizing Cyanogenmod based ROM image, which also includes two ad-hoc APK files (i.e., two apps pre-installed on the device). The IMSI anonymization privacy feature is enabled system-wide and the user has specified a privacy profile that allows one of the two applications (it.tim.goodware) to access the real IMSI. The privacy profile that is set for the second application (it.tim.malware) forbids obtaining the real IMSI and, therefore, this app will get an anonymized IMSI, i.e., the IMSI encrypted by means of a format preserving encryption algorithm.

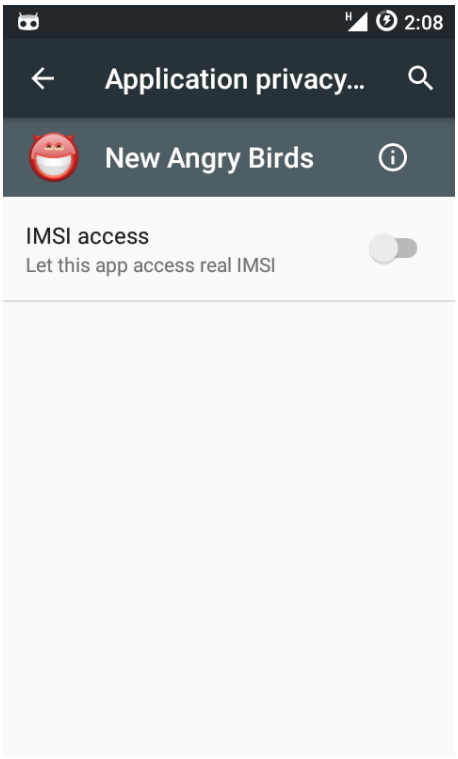


Figure 3-19 Malware app privacy profile details.

The user launches the it.tim.goodware application and tries to retrieve the IMSI by tapping the "GET IMSI" button. The operating system returns the real IMSI to the application that displays this value to the user.

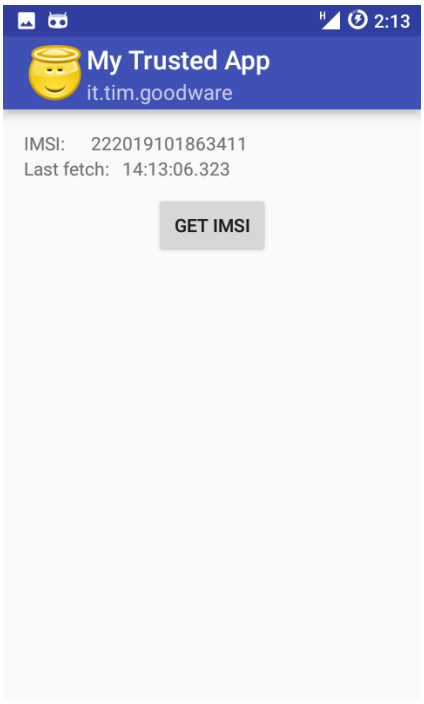


Figure 3-20 Goodware activity.

The user then executes the it.tim.malware application and taps the "GET IMSI" button. In this case the user observes that the application is not allowed to fetch the real IMSI. The displayed IMSI has in fact been

encrypted using a format-preserving encryption and this anonymized version of IMSI is returned to the app instead of the real one.

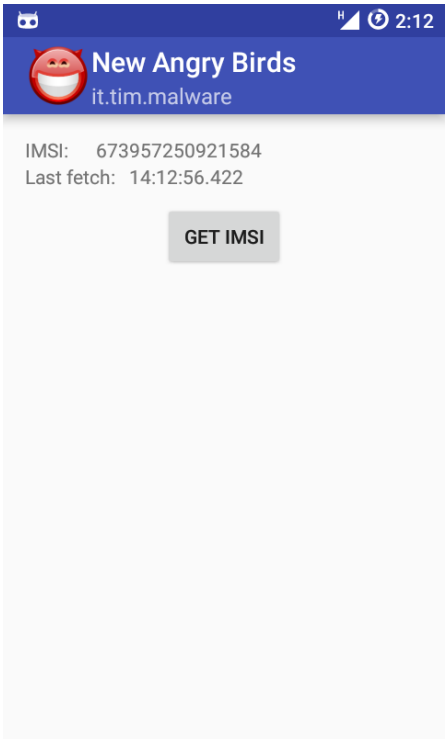


Figure 3-21 Malware activity.

3.3.8.2 Components and interaction overview

There are several actors involved in the typical usage of our enabler implementation:

- The user that is physically interacting with the device.
- The Security activity from Android's Settings.apk that contains the system-wide anonymization toggle and the list of installed applications requiring the READ_PHONE_STATE permission for which it is possible to configure a privacy profile.
- The TelephonyManager system service in the Android operating system that intercepts the call to the method that returns the IMSI value and, after evaluating several conditions, performs a format-preserving encryption using a secret generated at boot time. The encryption is selectively performed depending on the status of the system-wide anonymization toggle and the specific privacy settings for the application that is invoking the method.
- Any application that requires the READ_PHONE_STATE permission at installation time and calls the getSubscriberId() method of the TelephonyManager class.

3.3.9 Architectural drivers

3.3.9.1 High-Level functional requirements

The high-level requirements of the enabler are:

- The protection (by means of anonymization) of the sensitive subscriber's identifier (IMSI) returned by the device operating system to the user space applications. Only selected user space applications can access the IMSI in clear text.

- The configurability of the privacy profile: the device anonymizes the IMSI by default, nevertheless the user can specify (a whitelist of) applications that will be excluded from the anonymization/privacy profile.

3.3.9.2 Quality attributes

A product implementing the features described in this enabler should be evaluated primarily based on the following quality attributes: usability, anonymization (IMSI confidentiality), performance efficiency, configurability and maintainability.

3.3.9.3 Technical constraints

In order to implement the enabler access to the operating system source code is needed.

3.3.9.4 Business constraints

In order to implement the enabler in a commercial phone an alliance/special agreement with the phone OS producer is needed.

3.3.9.5 Link to Architecture

This enabler is relevant to the following objectives from the draft Security Architecture D2.4:

- **O2.1** 5G must provide a security and privacy level higher or at least equal to the security and privacy level in 4G.
- **O7.5** 5G systems and components must provide functionality to mutually assess the trustworthiness before, and during interactions
- **O6.3 5G** connected devices must be able to adequately protect critical data such as subscriber credentials against threats of unauthorized access and/or modification.

The mapping onto the 5G architectural domains and strata defined in the project's deliverable D2.4 is illustrated in the two figures below.

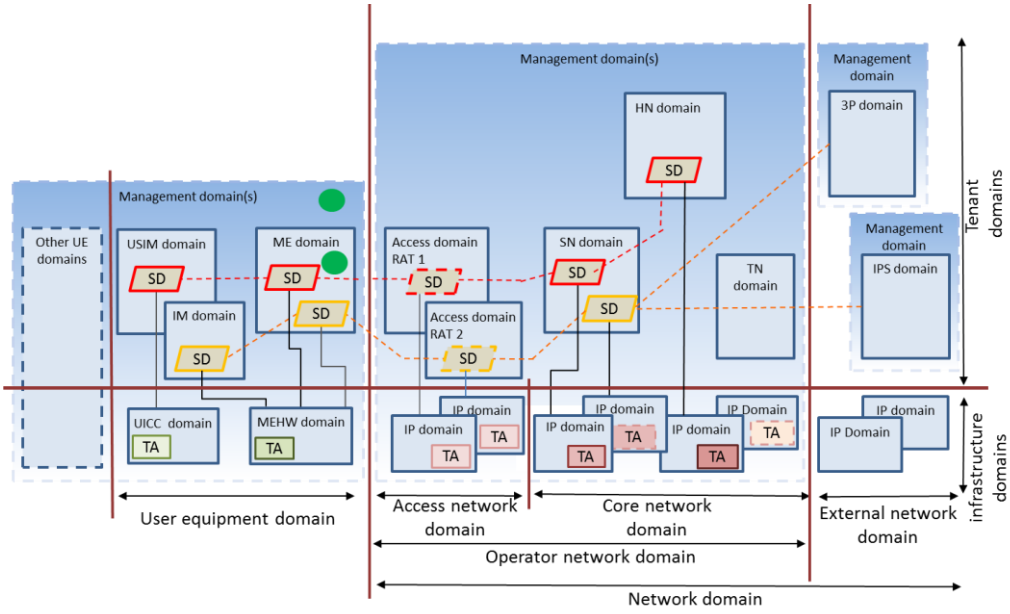


Figure 3-22 Device-based anonymizaton - domain mapping.

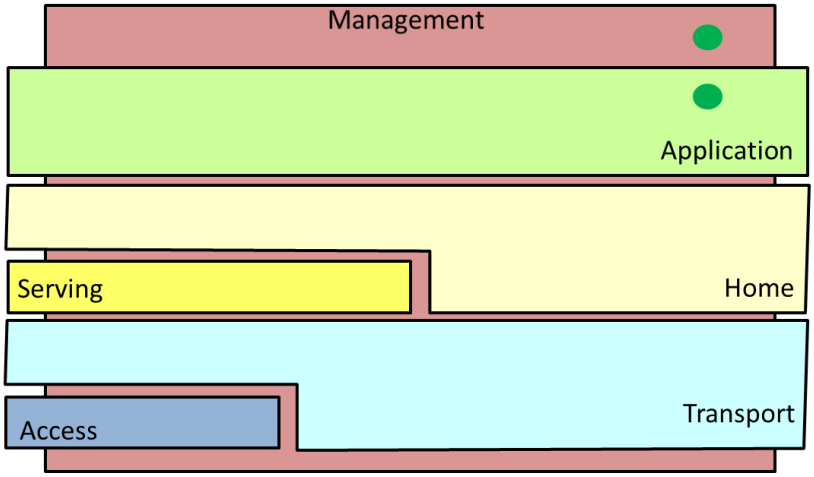


Figure 3-23 Device-based anonymization, strata mapping.

3.3.10 Detailed specifications

3.3.10.1 Introduction

This section contains the detailed specification of the modified anonymizing Android OS which can be used/flashed on Samsung Galaxy SII devices.

3.3.10.2 Conformance

For an implementation to be conformant it should implement specifications as stated herein.

3.3.10.3 Enabler Component Specifications

The enabler has two main components.

- The Security activity from Android's Settings.apk that contains the system-wide anonymization toggle and the list of installed applications requiring the READ_PHONE_STATE permission for which it is possible to configure a privacy profile.
- The modified TelephonyManager system service in the Android operating system that intercepts the call to the method that returns the IMSI value and, after evaluating several conditions, performs a format-preserving encryption using a secret generated at boot time. The encryption is selectively performed depending on the status of the system-wide anonymization toggle and the specific privacy settings for the application that is invoking the method.

The Security activity in Settings.apk has been enhanced with the addition of two custom settings. The first one is the toggle which is used to turn on the system-wide anonymization capability of the enabler for all applications that require the READ_PHONE_STATE permission. The value of the toggle persists between reboots by storing its state in the CMSettings class from the Cyanogenmod's SDK.

The second setting is related to per-application privacy profiles. Its menu entry is enabled only if device anonymization is currently active. It launches an application management activity which instantiates a list containing all installed applications that require the READ_PHONE_STATE permission. An entry in this list contains the application name, the application package name and the current app's anonymization state. Supported anonymization states are:

- "Can read IMSI": the application is allowed to get the real IMSI.

- "Cannot read IMSI": the application is not allowed to get the real IMSI and can only access the encrypted/anonymized IMSI.
- "Managed by provider": the privacy profile for this application cannot be controlled by the user. Its access level is handled by the provider.

It is possible to change the anonymization state for a given application (not managed by the provider) by tapping on its entry. An activity with a toggle to set the IMSI access level for that specific application is launched. An additional application operation type has been added to the AppOpsManager class in order to keep track of the access level for a given application.

The other main component is the TelephonyManager system service. A single TelephonyManager object is constructed at boot time. A 256-bit AES secret key, which will be needed by the encryption process, is generated in its constructor. Applications' requests to read the IMSI are intercepted in the getSubscriberId() method and, depending on the requesting application, the IMSI value can be encrypted before being returned. If the system-wide device anonymization is enabled, the encryption takes place for applications whose privacy profile does not grant access to the real IMSI. The current privacy profile state set for a specific application is fetched from the AppOpsManager object. The encryption is based on a format-preserving algorithm which uses the AES key created at instantiation time as secret and the package name of the requesting application as tweak.

3.3.10.4 API specifications

N/A, the enabler provides a modified Android OS image.

3.3.10.4.1 Functions, messages and commands

Considering the generic Android platform architecture illustrated in [3], the customization of the AOSP source of the present enabler is related to the System Apps, Java API Framework and Android Runtime layers.

At the System Apps layer, the enhancements of the Security Activity from Settings.apk involve changing some classes inside:

- Settings.java
- SecuritySettings.java
- SettingsActivity.java
- AppPrivacySettings.java (new file)
- ManageApplications.java.

The IMSI reading privileges have been implemented by adding a new application operation type in AppOpsManager.java that belongs to the Core Libraries in the Android Runtime module.

The TelephonyManager system service is part of the system Managers group in the Java API Framework. Encryption capabilities have been added by modifying its constructor TelephonyManager(), its method getSubcriberId() and by adding a new generateSecretKey() method in order to create the encryption secret.

3.3.10.4.2 Signalling

N/A.

3.3.10.5 Library Specification

The enabler does not provide a library.

3.3.10.6 Examples
The examples are provided in the Use Cases section of the present enabler.

3.3.11 Re-utilised Technologies/Specifications
The current version of the described enabler is built from the source code of CyanogenMod 13.0 [1] for the Samsung Galaxy S2 and is based on a Java implementation of the FPE algorithm [2].

CyanogenMod is an open-source operating system based on AOSP that offers a wide range of features such as unlockable bootloader, root access, layout and graphical customisation and tweaks, CPU overclocking and other performance enhancements. The CM SDK has been used to customize the Security activity in the Settings.apk and store the device anonymization toggle state.

An existing Java implementation of the FPE algorithm has been modified and added to the CyanogenMod 13.0 code base in the form of an external library in order to grant encryption functionalities in the TelephonyManager system service.

3.3.12 References
[1] Cyanogenmod 13.0, https://github.com/CyanogenMod.

[2] FPE for Java, https://github.com/robshep/JavaFPE.

[3] Android Platform Architecture, https://developer.android.com/guide/platform/index.html.

3.3.13 Acknowledgements

3.4 Privacy Policy Analysis Open specifications

3.4.1 Preface
Nowadays, users of networked services are confronted with a wide range of services and applications which may put their privacy at risk.

With this ever-increasing availability of services based on the 5G infrastructure, the potential exposition of personal data to unintended actors is a growing concern. Typically, users don't spend too much time reading services' privacy policies and have therefore little understanding of how their personal data may be accessed and used by a service, and for what purpose. The verbose textual privacy policy statements provided by services providers ensure legal compliance, but are of little help to users which usually disregard them [1].

This enabler aims to provide the user a way to analyse the privacy policies of a set of services or service providers in order to inform the users on how their privacy is handled. The analysis would usually be carried out prior to the service being used.

This enabler allows the user to specify their privacy preferences including what type of data they are willing to share, for what purpose and for what period. This allows the user to make privacy aware decisions regarding use of 5G services. The enabler may be of interest to all 5G users.

3.4.2 Status
Version 1.0 of this enabler is currently under development.

3.4.3 Copyright

3.4.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

3.4.5 Terms and definitions

VMNO	Virtual Mobile Network Operator
P3P	W3C Platform for Privacy Preferences (P3P) Project
LTS	Labelled Transition System

3.4.6 Overview

The main aim of this enabler is to create a user-centric algorithm and software framework for modelling and analysing the usage, processing and disclosure of users' private data when choosing a virtual mobile network operator. The enabler allows to have fine-grained description of services' privacy policies and how these policies ensure, or not, users' privacy preferences. For this we take a model-driven engineering approach in order to achieve:

i) **Abstraction**: the ability to abstract complex system behaviour and then machine analyse them
ii) **Reusability**: the ability to create platform independent models that can be applied to multiple concrete platform types.

This enabler shall use a data vocabulary to formally capture how services access and handle users' private information in the context of an inherently distributed and virtualised environments such as the ones envisaged by 5G systems. This data vocabulary will represent also the delegation of part of the service's implementation to third-parties and their potential access to users' private data and will provide the means to describe 5G enabled services' policies to access and make use of users' personal data.

VMNOs' privacy policies will then be formally described internally to the system, as an LTS which describes how the service will or may access different types of users' private information allowing the Privacy Policy Analysis enabler to match the changed privacy state of each private information against the user's specifications.

Input for the definition of this data vocabulary is the Platform for Privacy Preferences (P3P) W3C specification language for describing privacy policies [2].

A second standard taken into consideration in this enabler, APPEL [3], is a W3C working draft specification that complements P3P and allows users to express their privacy preferences in a set of preference-rules which allows the user agent to make automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P enabled Web sites. The preferences are encapsulated in a series of rules or "ruleset". Each rule contains conditions under which a behaviour (e.g. block, prompt) should be carried out.

The privacy enabler shall allow users to express their privacy preferences in terms of formal specifications that can then matched against privacy policies. These will allow users to define their privacy preferences, by using a system provided questionnaire, in terms of temporal logic formulas that will enable then the system to check the available services in term of:

- Properties that **must** be valid at all time during a service execution (e.g. password shall never be sent via an unencrypted channel)
- Properties that **may eventually** be valid during a service execution (e.g. grant access to a user's personal data once the user has agreed access to it)

The enabler assumes that the privacy policies of a set of services are available and allows the user to specify their privacy preferences. The enabler compares the user's preferences against the privacy policies of available services and indicates whether the respective policy is compliant with them or not.

The Privacy Policy Analysis enabler is implemented as a web application which offers to the registered users the capability to search which service, between a set of available VMNOs, is compliant to the user's privacy preferences.

The service is proposed as a standalone web-enabled application the user can use to decide which VMNO to choose. The service is in no way integrated to the process of the VMNO subscription itself.

3.4.7 Basic Concept

The Privacy Policy Analysis enabler is designed as a service that can be accessed by users wanting to choose between different VMNOs to connect with their mobiles. Depending on when the user decides to access the service we devised two main use cases:

1. The user access the service to decide which VMNO has acceptable privacy policies before subscription
2. The user needs to connect to a foreign VMNO while travelling and needs to know which of the available VMNOs has acceptable privacy policies

The use case 1 implements the bare capability and offers the service, ideally as a web application, before the user is presented with the need to subscribe with the VMNO. As a simplification, the use case 1 could be used also when roaming assuming that the user has access to a WiFi connection.

The ideal situation is represented by the use case 2, when the mobile is scanning for available networks to roam to, and it would use the user privacy preferences on the phone to choose the VMNO which provides the desired level of privacy. The mobile phone could download the policies of available VMNOs from the networks themselves and do the comparison on the phone itself prior network subscription.

For both use cases the sequence of actions would be as follows:

A) Acquire user's privacy preferences
B) Acquire VMNOs' privacy policies
C) Use user's privacy preferences and VMNOs' privacy policies to filter out the available VMNOs
D) Store preferences on the phone (potentially on the user's SIM).

3.4.8 Main interactions

3.4.8.1 Use Cases

In this section, we detail the use cases which provide an overview of the interactions the users can have with the software to use the service.

Use case name: user registration (depicted in **Error! Reference source not found.**)

- Goal: creation of a user's profile on the service to store the user's preferences.
- Description: the software will allow users to register to the service.
- Rationale: privacy based analysis of service offerings.

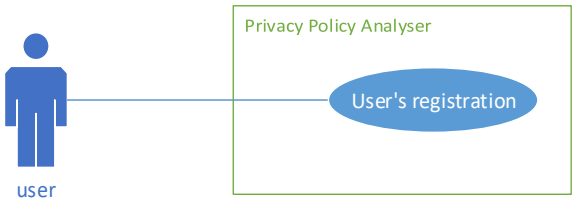


Figure 3-24 User registration

Use case name: user's privacy preferences specification (depicted in **Figure 3-24**.

- Goal: customized analysis of service privacy practices.
- Description: the software will allow users to specify their privacy preferences to be compared against the privacy policies by using a questionnaire.
- Rationale: privacy based analysis of service offerings.

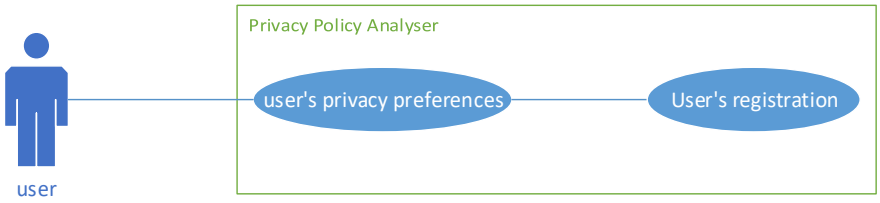


Figure 3-25 User's privacy preference specification

Use case name: specify privacy policy (depicted in **Figure 3-25**).

- Goal: load privacy policies for later user's analysis.
- Description: an administrator shall load services' privacy policies to allow users to analyze them against their privacy preferences.
- Rationale: privacy based analysis of service offerings.

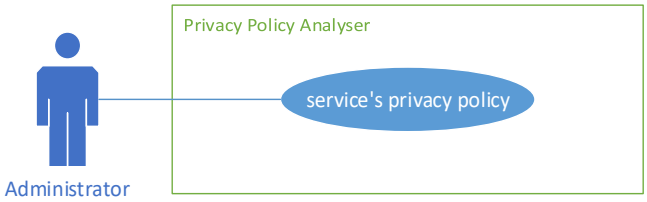


Figure 3-26 Service's privacy policy specification

Use case name: privacy aware service search (depicted in **Figure 3-26**).

- Goal: search for services based on their policies and the registered user's privacy preferences.
- Description: a registered user shall use the previously stored preferences to search for available services which comply with the user's privacy preferences.
- Rationale: privacy based analysis of service offerings.

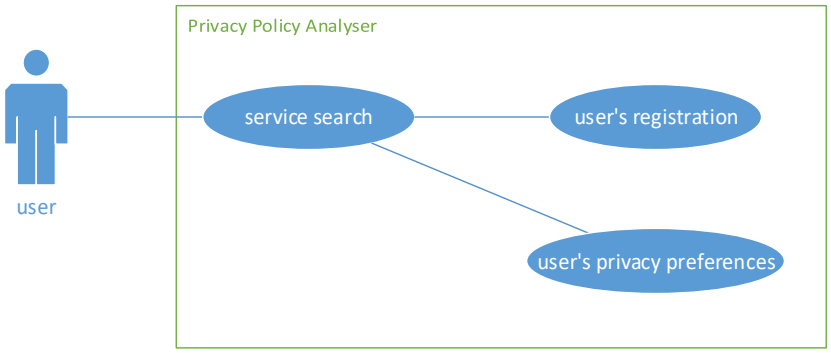


Figure 3-27 Privacy-aware service search

3.4.8.2 Components and interaction overview

Privacy Policy Analysis enabler will be a web application and REST service layer employing JavaScript libraries to implement a responsive interface. The processing of users' privacy preferences when implementing the privacy-aware service searches will be done on server side and caching will be used whenever required to increase the responsiveness of the interface.

3.4.9 Architectural drivers

3.4.9.1 High-Level functional requirements

The high-level functional requirements are covered by the use cases above

3.4.9.2 Quality attributes

The quality attributes relevant for this enabler are mainly: logical correctness, and performances in providing searches' results.

3.4.9.3 Technical constraints

The enabler shall be run on a container with access to the port 80 to allow the web application to work properly. The application shall be deployed in an Apache Tomcat servlet container using HTML5 for the client based interactions.

3.4.9.4 Business constraints

No known business constraint exists.

3.4.9.5 Link to Architecture

The enabler allows users, and local authorities, to better understand the privacy implications of new 5G-enabled services. In this regard, the enabler fulfils the security objectives defined in D2.4, section 3.8 on regulatory aspects. First of all, it allows users to assess the privacy requirements of 5G enabled services and check if they are compatible with local regulation (O9.3) or compliant (O9.1) with users' desired level of privacy.

The figures below show the relevant domains and strata (marked by a red star) of the 5G-ENSURE Security Architecture.

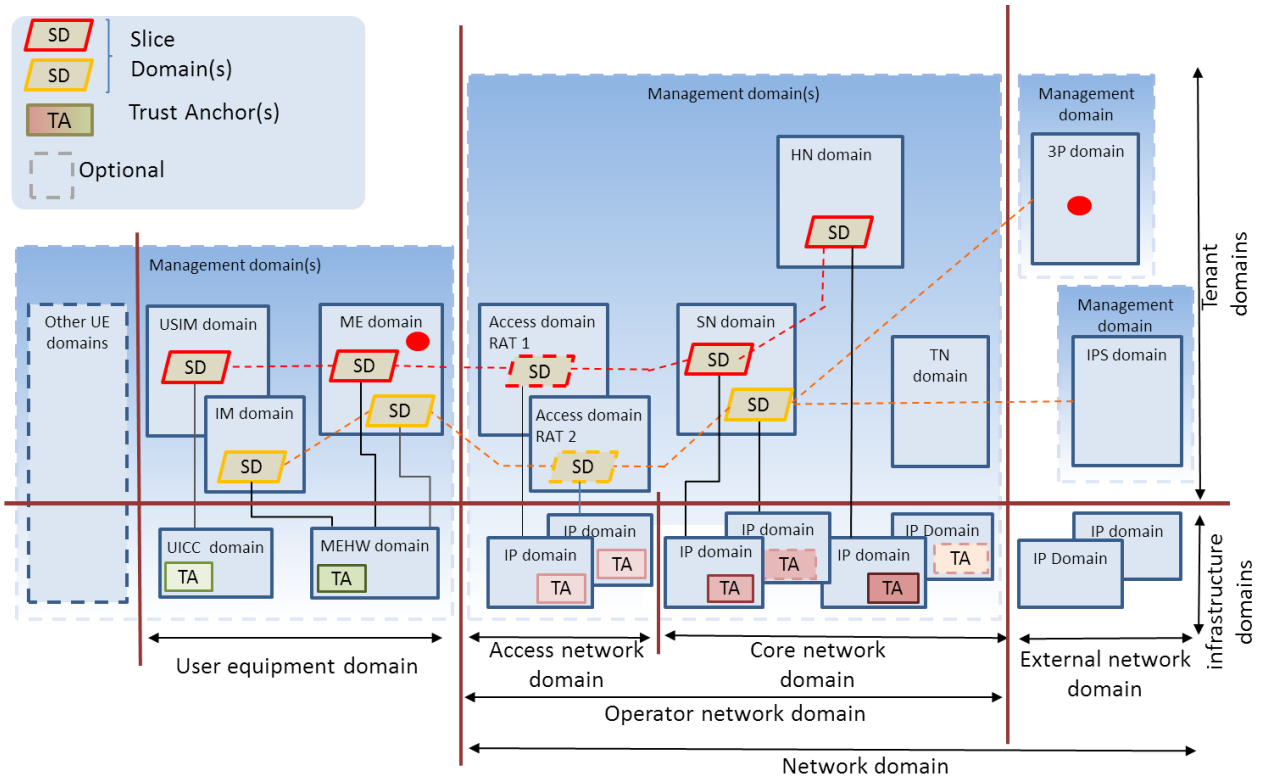


Figure 3-28

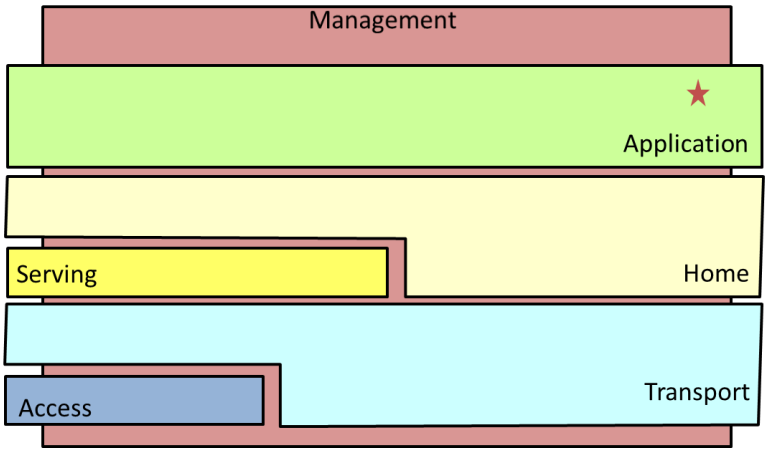


Figure 3-29 Enabler related strata.

3.4.10 Detailed specifications

3.4.10.1 Introduction

The Privacy Policy Analysis enabler provides the tools to model services' privacy policies and users' privacy preferences in order to implement a privacy aware search for services which comply with the user's preferences.

3.4.10.2 Conformance

For an implementation to be conformant it should implement specifications as stated herein.

3.4.10.3 Enabler Component Specifications

3.4.10.4 API specifications

Although the enabler provides a layer of REST APIs to support the web application, this is not the main outcome of the artefact. Nevertheless, the API specifications will be provided in swagger 2 format.

3.4.10.5 Library Specification

The Privacy Policy Analysis enabler will be developed as a three-layer application, with a presentation layer in HTML5/JavaScript, a REST service layer, and a services' implementation layer that could be exported as a library. Nevertheless, the provision of a service library is not the focus of the enabler.

3.4.10.6 Examples

After a user has registered with the application, he/she can specify his/her privacy preferences by selecting the command "specify preferences" form the user's profile page. This command will propose to the user a set of questions whose aim is to identify the preferred privacy state for each type of personal information (e.g. identity, location, financial information). Once the questionnaire is completed, the user can access the page for searching VMNO. The information provided by the questionnaire will be used jointly with the VMNO's privacy policies to filter out the services which violate some of the user's preferences.

The privacy policies for the different VMNOs are defined using an administrator interface prior to the users' registration.

3.4.11 Re-utilised Technologies/Specifications

The enabler will be implemented by using the SPRING BOOT framework and the JENA API for managing RDF and SPARQL.

3.4.12 References

[1] McDonald, A.M. and Cranor, L.F., 2008. The cost of reading privacy policies. ISJLP, 4, p.543.

[2] Reagle, Joseph, and Lorrie Faith Cranor. "The platform for privacy preferences." Communications of the ACM 42.2 (1999): 48-55.

[3] Cranor, Lorrie, Marc Langheinrich, and Massimo Marchiori. "A P3P preference exchange language 1.0 (APPEL 1.0): W3C working draft 15 April 2002." World Wide Web Consortium (W3C), URL: http://www. w3. org/TR/P3P-preferences (2002).

3.4.13 Acknowledgements

4 Trust Enablers Open specifications

This section contains the open specifications of the five trust enablers due at Release 2: the Trust Builder enabler, the Trust Metric enabler, the VNF Certification enabler, the Security Indicator enabler, and the Reputation based on Root Cause Analysis for SDN enabler.

4.1 Trust Builder Open specifications

4.1.1 Preface

The Trust Builder enabler uses innovative semantic models to provide a way to understand and control threats to a system. The Trust Builder is used primarily when specifying or designing new deployments and shares models with the Secure System State Repository enabler which is a runtime monitoring component.

4.1.2 Status

The roadmap for the development of the Trust Builder has two releases, R1 and R2, the contents of which are as follows:

- R1
    - Graphical editor v1 (software)
    - 5G asset model v1 (ontology)
- R2
    - Graphical editor v2 (software)
    - 5G asset model v2 (ontology)
    - 5G threat knowledgebase (ontology)

Following delivery of enabler in R1, work is currently ongoing towards release R2.

4.1.3 Copyright

4.1.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

4.1.5 Terms and definitions

Core Model – the core ontology, defining common vocabulary and relationships used in all higher-level models.

Generic Model – an ontology defining the typology of Assets, Threats and Controls (security measures) for a given domain (e.g. 5G networks). In release R1, this only covered Assets and some types of Controls.

Design-Time System Model – an abstract model of a particular system, described in terms of the relationships between system specific Asset classes. The design-time model can then be enriched by specifying which security Controls should be used to protect each type of system asset, and auto-generating a set of system-specific Threat classes describing potential threats to the system.

Runtime System Model – a model using instances of the Assets, Threats and Controls to describe what is known about the current state of an existing system.

4.1.6 Overview

The Core and Generic Models are both used in the Trust Builder and System Security State Repository enablers. The Core Model provides a basic structure which both tools assume will underpin any System Model they need to handle. Both tools also load the Generic Model, and use it to generate user interface elements and as a basis for semantic reasoning.

The increased complexity and dynamicity of 5G networks means that modelling and understanding the threats is even more important than in earlier relatively static and well-understood systems. For instance, with NFV, SDN and micro-segmentation there will be new virtual networks (which may be relatively short-lived) created on top of the static physical infrastructure. Design-time decisions will therefore be more frequent and more complex and decision-support tools such as the Trust Builder can play their part.

The Trust Builder enabler provides a way to model the components of 5G networks and their relationships, understand what threats are present and what controls are necessary to mitigate or annul those threats. It thereby supports creation of a Design-Time System Model describing a specific 5G network and/or application, including an automatically generated set of potential threats from which trust and security requirements can be determined.

For instance, when needing to deploy a new slice to support a specific application, a (virtual) network operator will need to define how the various network elements are orchestrated: how to create the required functionality by composing the basic elements defined by the 5G standard. These network slices which will potentially incorporate physical assets of many actors will not be one-size-fits-all solutions and so each different design will need to be analysed to determine what controls (e.g. firewall rules) are necessary to mitigate the threats specific to the design. Here the user of the enabler is the (V)MNO and potentially (depending on their business relationship) the customer of the slice.

Trust Builder will be a web application where system designers and other classes of user can create and collaborate on “design-time” system models. Based on the threats and controls encoded in the knowledgebase, Trust Builder will highlight the potential weaknesses in a model and propose controls to address the nascent threats. The Design-Time System Model can then be used as an input to the System Security State Repository to document the intended design of the system (including the necessary controls) and thus carried into the runtime operational phase.

4.1.7 Basic concepts

The Trust Builder makes use of a Core Model and a Generic Model to create a Design-Time System Model. These concepts are further described here.

4.1.7.1 The Core Model

The current Core Model is shown in **Figure 4-1**. It includes just the concepts which are common across all domains. In simple terms the model says that Assets can be threatened by Threats which cause Misbehaviour and protected by Controls. Furthermore, Assets that are related to each other using specific relationship types to form Patterns, in which each Asset takes a Role (such as client or server). Threats apply to Patterns and when active cause Misbehaviours (i.e. unacceptable behaviour) in the Assets forming the Pattern they apply to. Control Strategies, each comprising of one or more Controls located on Assets block or mitigate Threats. If all Controls within a Control Strategy are present, the Control Strategy blocks or mitigates a Threat.

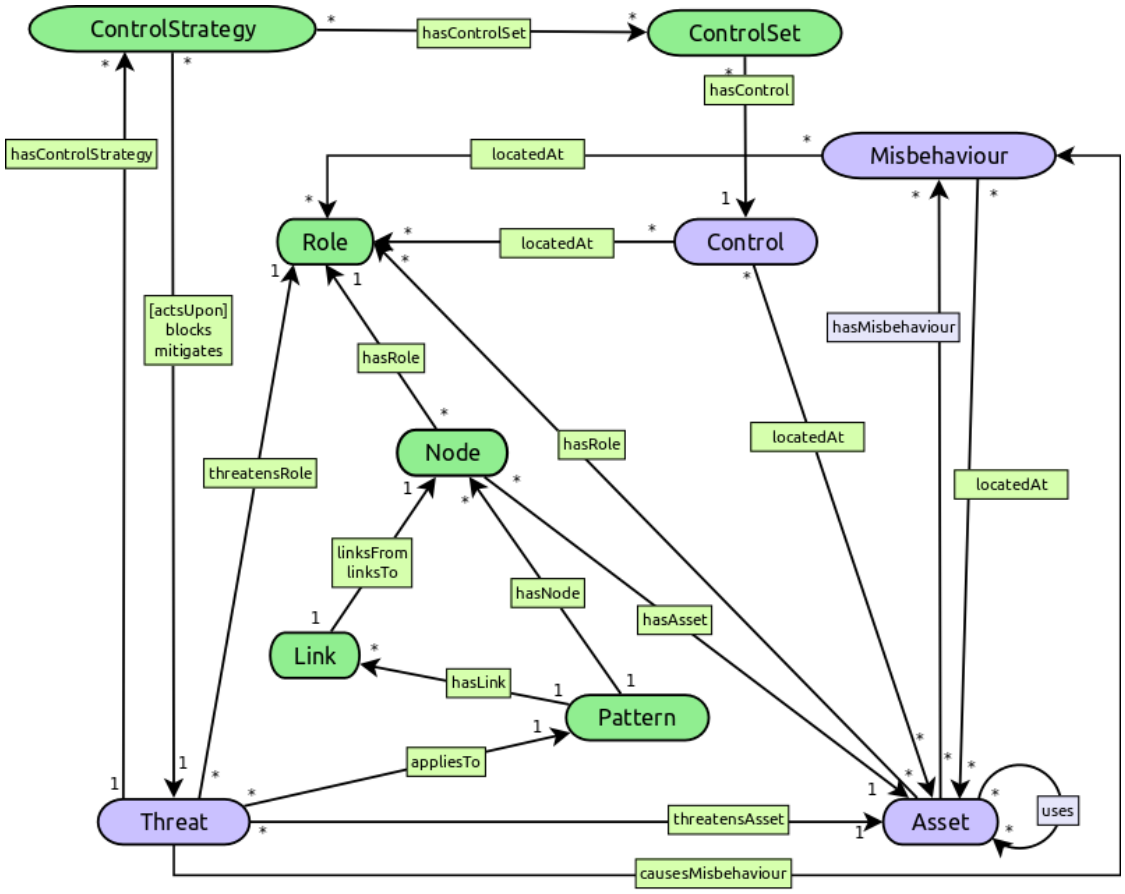


Figure 4-1: The Core Model.

4.1.7.2 The Generic Model

The Generic Model is where the domain-specific knowledge is encoded. In relation to the Trust Builder roadmap, the 5G Asset Model and the 5G Threat Knowledgebase together form the Generic Model. As the content and design of the Asset Model and Threat Knowledgebase are formed it may be necessary to adjust or extend the concepts in the Core Model itself.

The Generic Model specialises the Core Model allowing the encoding of the security knowledge. The specialisation is done via sub classing the different core model classes:

- The subclasses of core:Asset form the 5G Asset Model, representing the Assets (such as an HSS but also human stakeholders) that will be used by the system designer to express the structure of their system. Assets are categorised by type, e.g. logical asset, host, stakeholder etc.
- The 5G Threat Knowledgebase includes several parts:
    - The subclasses of core:Threat constitute the Threat knowledgebase that may compromise the generic Assets.
    - The subclasses of core:Control constitute the set of Controls that can protect Assets and thereby block Threat actions or mitigate Threat consequences. The Control Strategy elements describe groups of Controls that can counteract specific Threats.
    - The subclasses of core:Misbehaviour are the set of adverse Asset behaviours that could be induced by the Threats if they were active (i.e. if the Threat action occurs).

In addition, Patterns of Assets will also be defined and the possible Asset relationship types will be encoded.

4.1.7.3 The Design-Time System Model

The purpose of the Trust Builder is to provide a tool to create and analyse Design-Time System Models. In the use cases below, “model” refers to the instances of the Design-Time System Model.

Technically, the Design-Time System Model is created by specialising the Assets, Threats and Controls defined in the Generic Model. The user invokes this specialisation by creating system-specific asset classes and adding relationships between them. Based on these asset classes and the generic asset class they specialise, we are then able to:

- Generate system-specific Control and Misbehaviour combinations ("sets"),
- Match patterns defined in the Generic Model in the System Model and,
- Create system-specific Threats based on the matched patterns and generic Threats.

Practically, the user of the Trust Builder will create the Design-Time System Model by dragging and dropping assets from the Generic Model onto a canvas and linking them together using the relationships also defined in the Generic Model.

4.1.8 Main interactions

4.1.8.1 Use cases

The use cases are diagrammed below in **Figure 4-2**.

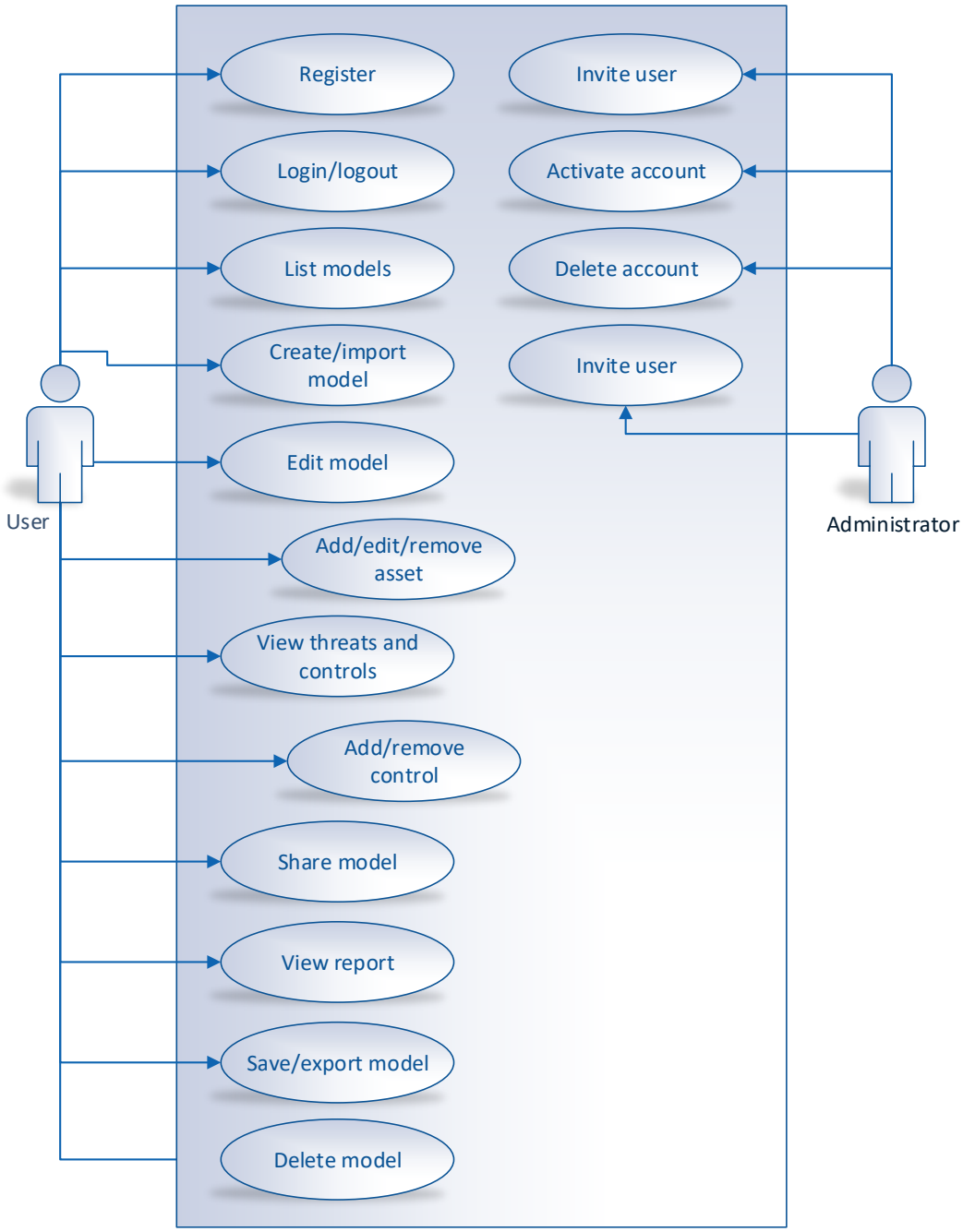


Figure 4-2: Trust Builder use cases.

4.1.8.1.1 Registration and user account setup

A new user requests an account for the service or is invited by an administrator. After activation by an administrator, the user is provided with credentials for their account. The user account will provide access to the service: an online workspace where the user can create, modify, store and share models. A user account can be deleted by an administrator at any time.

4.1.8.1.2 Login/Logout

The user logs into their account (is authenticated) using a secure connection. On logging in, the user should be informed of when they last logged in. The subsequent use-cases are only available to authenticated users. After using the service, the user should log out to prevent unauthorised access.

4.1.8.1.3 List models

The user can list all models that they have access to.

4.1.8.1.4 Share model

A user can choose to share read-only or read-write access to a model with another user. A user can also transfer ownership of a model that they own, choosing whether to retain read-only, read-write or no access rights on the model. Only exactly one user can be the owner of a model at any time. The owner of a model may revoke others' access rights on a model. A user who has access rights to a model may revoke their own rights.

Concurrent access to models will not be implemented, so if more than one user has write access, the model will need to be locked while being edited by a user.

4.1.8.1.5 Create / Import model

A model can be created by starting with a blank canvas (i.e. "empty" model). Alternatively, an existing model RDF file can be uploaded from disk. The user becomes the owner of any model they create.

4.1.8.1.6 Delete model

Models the user owns can be deleted. This will remove the model entirely.

4.1.8.1.7 Design-time model specification and validation

Application domain experts will create system models by dragging and dropping generic Assets from a curated subset of all generic (domain-specific) Asset classes onto a canvas and connecting them using predefined relationship types.

Once a system Asset model has been created, the editor should automatically:

- determine and add any implicitly created Assets and relationships as needed to completely define the relationships of system Assets needed to model Threats and Control Strategies;
- flag missing assets/relationships for some of the system-specific Asset classes, which are also needed to determine the applicability of some generic Threats;
- generate system-specific Threat classes by specialising the Generic Model classes so they apply to system-specific Asset types.

The process of carrying out these steps is known as system model 'validation'.

4.1.8.1.8 Design-time model analysis and refinement

Once a validated system model is available, the designer would be interested in finding out:

- a complete list of known potential Threats to the system (i.e. system specific Threat classes generated during the validation step) and their relationships to system specific Asset types;
- a classification of those Threat types based on whether they are secondary effects;
- a complete list of Misbehaviours that could be caused by those Threat types (i.e. which types of Misbehaviour could be caused in each Asset type);
- a list of potential Threats to (and/or involving) a selected system Asset type, taking account of their secondary effects;
- a list of Misbehaviours that may be caused in a selected system Asset type by the known system Threat types, taking account of secondary threats;
- a set of relevant Control options for blocking or mitigating a selected type of Threat;

- the set of Threats whose control options include a specific control protecting a specific Asset type.

The designer should then be able to select one or more Control options for Threats they wish to block or mitigate. By selecting a Control option, the designer is saying that the corresponding control objectives should be met by the system implementers so that type of Threat will be blocked/mitigated. At any stage the designer will be able to query the model to find out the consequences if all the control requirements so far selected by the designer are met:

- Which potential Threat types would be blocked or mitigated and which would be left untreated?
- Which Misbehaviours in which Asset types should be monitored to detect activity in untreated, unblocked or all types of Threats (i.e. which Misbehaviours could be caused by those Threats)?
- For a selected Asset type, which types of Controls should be protecting that Asset type given the selected Control options?
- For a selected Asset type, which types of Misbehaviours should be monitored to detect activity in untreated or unblocked types of Threats?

Ultimately, Trust Builder will thereby allow designers to determine which stakeholder(s) could be affected by each threat (taking account of their secondary effects) and also which stakeholder(s) are in a position to provide security measures (including other 5G-ENSURE enablers) to counteract each threat. This provides a detailed understanding of the trust assumptions (between stakeholders) underpinning the design of a scenario including its security requirements. By analysing multiple scenarios that use the 5G-ENSURE architecture, it is also possible to understand and specify common trust assumptions that underpin the architecture itself.

4.1.8.1.9 View report

After finishing the modification and refinement of a model, the user can generate a report. This will generate and display a (printable) HTML page, containing information (visual and/or text) about the model and the effects of the controls including but not limited to:

- a picture of the system model graph
- a list of all assets in the system model including information about potential controls
- a list of all threats in the system, possibly grouped by which asset they affect
- an overview of controls added to the system and the percentage of threats that would be treated with the controls in effect

4.1.8.1.10 Model import/export

This feature provides a mechanism to import/export models to/from the system. It enables users to download model files to their machine for archive or sharing purposes.

4.1.8.2 Components and interaction overview

The Trust Builder will be a web application using HTML5 and modern client-side JavaScript libraries to provide a rich responsive interface. The web client will communicate with the server side using a secure REST API. Processing of the Design-Time System Model will be performed on the server side and the Threats and potential Controls communicated back to the client for examination and selection.

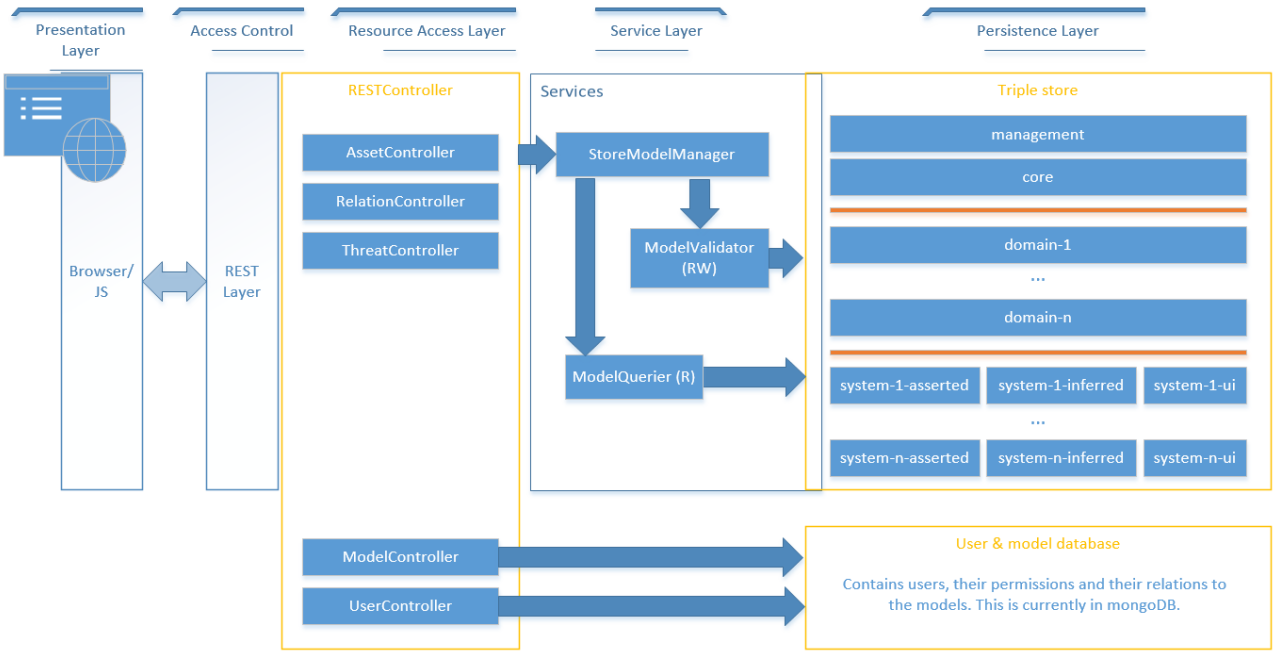


Figure 4-3: Trust Builder architecture

4.1.9 Architectural drivers

4.1.9.1 High-Level functional requirements

The high-level functional requirements are covered by the use cases above.

4.1.9.2 Link to Security Architecture

The Trust Builder enabler is primarily used at design-time when specifying or designing new deployments, and thus does not reside in any of the Domains or Strata as defined in D2.4.

The Trust Builder enabler helps deliver the 5G Security Objectives O2.2, O8.1, and O9.2. The relationship between the Trust Model and the Security Architecture is detailed in Section 6 of D2.4.

4.1.9.3 Quality attributes

Compatibility: all generated ontologies need to be valid serialized RDF files.

4.1.9.4 Technical constraints

The tool should work in recent main-stream desktop web browsers such as Chrome and Firefox.

4.1.9.5 Business constraints

No known business constraint exists.

4.1.10 Detailed specifications

4.1.10.1 Introduction

The Trust Builder enabler provides a way to model the components of 5G networks and their relationships, understand what threats are present and what controls are necessary to mitigate or annul those threats

4.1.10.2 Conformance

For an implementation to be conformant it should implement specifications as stated herein.

4.1.10.3 API specifications

Although the Trust Builder service will provide an API for the web-based client to use, the two parts are closely coupled and alternative clients using the same API are not envisaged. Therefore, the detail of the API is not relevant and does not form part of this open specification.

4.1.11 Re-utilised Technologies/Specifications

This enabler builds on the Secure System Designer software from the OPTET project [1]. Aspects of the interface design along with some libraries to be used on the server side of the Trust Builder will be re-used.

4.1.12 References

[1] OPTET Project: http://www.optet.eu/

4.2 Trust Metric Enabler Open specifications

4.2.1 Preface

This section presents 'Trust Metric Enabler'. The enabler is related to two other 5G-ENSURE security enablers, namely 'Micro Segmentation' and 'Security Monitor for 5G Micro-Segments'.

4.2.2 Status

This document provides an open specification of the framework feature that will be implemented for the second release of the enabler.

4.2.3 Copyright

4.2.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

4.2.5 Terms and definitions

AP	Access Point
eNodeB	Evolved Node B
EPC	Evolved Packet Core
ePDG	Evolved Packet Data Gateway
HSS	Home Subscriber Server
IDS	Intrusion Detection System
Micro-Segment	An isolated and secured segment inside a SDN
Micro-Segment Orchestrator	Entity that automatically arranges, coordinates and manages micro-segments
MME	Mobility Management Entity
MNO	Mobile Network Operator
PGW	Packet Data Network Gateway
RAN	Radio Access Network
SDN	Software Defined Network
SGW	Serving Gateway
UE	User Equipment
VMNO	Virtual MNO
VNF	Virtual Network Function

4.2.6 Overview

The vision of 5G is a flexible and dynamic system in terms of use cases, UEs, RAN technologies and core network services. The complexity of the system will become even more evident if new deployment models, such as third-party and multi-operator deployments, and new third-party APIs, enabling e.g. full configuration control of network functions, will be supported. The goal of Trust Metric Enabler is to reduce the complexity of operating security aware 5G services, provided by MNO or VMNO, specifically in terms of trust. In addition, it facilitates efficient use of 5G network resources by optimizing the used 5G network infrastructure and services based on the third-party service provider requirements.

The Trust Metric enabler provides an abstraction layer for utilizing trust in third-party services (also in other 5G Security Enablers). The third-party service provider may use the enabler to

- Define trust requirements of the provided service
    - 5G system will adapt to these requirements within its capabilities
    - 5G network and 5G Security Enablers may provide dynamic security controls and network management, when available
- Monitor how the trust requirements are met in the e.g. service as a whole, part of the service, specific UE or traffic flow.
- Assign trust levels for unauthenticated UEs e.g. through isolated slices with MAC authentication bypass. Trust level can be zero when UE is not-trusted

The main function of the enabler is to **produce a trust metric value which measures in real-time how the third party service provider's and the network operator's combined trust requirements are met in the 5G system**. The trust metric value is calculated based on the third party service provider's trust model, 5G network operator's trust model and 5G trust measurements. When the result of the calculation is acceptable, the service is enabled, otherwise it is disabled. The enabler has three input and one output categories:

- Inputs: Third-party service provider's trust model, network operator's trust model, trust measurement
- Outputs: Trust metric value

4.2.7 Basic concepts

Trust model

Trust model is used to present a set of security related characteristics related to a specific use case, service or system. It includes the mechanisms necessary to respond to a respective threat profile. A threat profile identifies most likely attacks and vulnerabilities that put a specific environment at risk. Trust model encompasses also trust relationships between entities. If it is possible to bind unique attributes to a unique identity (including non-verified identity) then a trust relationship can be established based on the level of confidence in the binding. The trust relationship enables a framework in which interactions between the entities takes place within predictable boundaries with some confidence. The model may include also definition of a trust domain in which a baseline trustworthiness value can be associated to entities within the domain. Trust model is used here in two main contexts:

1. Third-party service trust model defines a set of requirements that must be met in order to enable good-enough security related to a specific third-party service.

2. Network trust model defines a set of axiomatic characteristics in the current network setup and the additional security capabilities that can be applied. These additional capabilities may adapt and evolve as measured trust changes.

Trust measurement

"Trust measurement is an objective measurement that provides evidence to enable evaluation of trust from a specific point-of-view. The measurement must be aligned with the trust requirements in order to support the evaluation. For each point-of-view different measures must be used, and these measures may change when environment changes. For example, the measurement categories can include

- Application trust; level of end-to-end protection, level of platform protection, level of application protection, measurements of vulnerabilities
- Communication trust; level of transport protection, level of platform protection, QoS measurements, level of implemented security controls, measurements of vulnerabilities
- Identity trust; level of authentication mechanisms, reputation of the peers, transitive trust characteristics

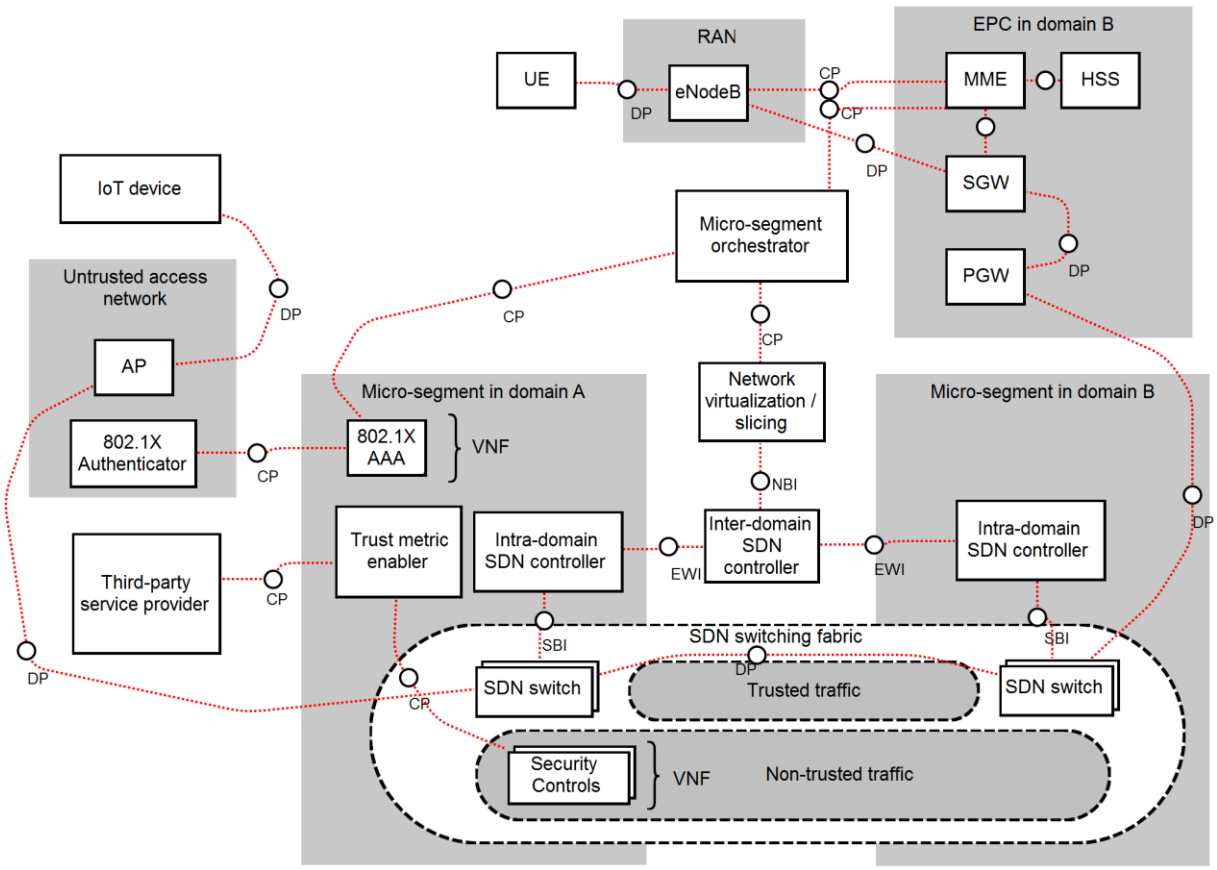
4.2.8 Main interactions

4.2.8.1 Use cases

A component diagram of an example use case is presented in the following figure. In this use case a third-party service provider provides an IoT service to end-user (presented by UE). The service's main value-added characteristics are following

- UE is able to connect directly (i.e. no intermediated servers) to IoT devices
- Untrusted access network can be utilized to connect untrusted IoT devices to the service securely

The service provider utilizes Micro-Segmentation Enabler (presented by Micro-segment orchestrator) which takes care of service isolation, multi-domain support and network management (Micro-segments and segmentation are defined in Chapter 6.4 of this document).



CP = Control plane, DP = Data plane,
NBI = North-bound interface (SDN), SBI = South-bound interface (SDN), EWI = East-west interface (SDN)

Figure 4-4: Use case example component diagram

The third-party service provider has defined following network requirements, measurements and trust metric mapping in a trust model.

Network requirements

- Isolated micro-segment
  - Allow communication between IoT device and UE (data plane)
- VNFs: 802.1X AAA, IDS (Security controls)
- Minimum data plane protection
  - Transport based hop-by-hop integrity protection
- Traffic flows (data plane)
  - Trusted traffic
    - Allow trusted network access with 802.1X authentication
    - Highest QoS (no additional security controls)
  - Non-trusted traffic
    - Allow
      - untrusted network access with MAC authentication bypass
      - untrusted network access with web authentication
    - Deny end-to-end encrypted traffic (in order IDS to work)
    - Inspected by IDS

Measurements

- Validate network requirements (this measurement is included by default)
- Number of active IoT Devices and UEs (controlled by a single eNB)
- Subscribed IDS alerts
    - High severity (i.e. indicates attacks and malware)

Trust metric mapping

- Binary value (0,1)
- Value 0 (trust metric unacceptable)
    - Network requirements are not achieved or validated
    - Upper limit for active IoT devices and/or active UE per eNB is exceeded
    - IDS alert with high severity detected
- Value 1 (trust metric acceptable)
    - Otherwise

5G network (Micro-segmentation Enabler) has provided following network capabilities and measurements in a trust model.

Network capabilities

- Generic network capabilities
- Security Enablers
    - Micro-segmentation
    - Trust Metric Enabler
- VNFs (security)
    - IDS, IPS, Firewall

Measurements

- Validation of network capabilities
- Security Enablers
    - Micro-segment network and service statistics
- VNFs
    - IDS and IPS alerts
    - ePDG statistics

The following diagram depicts the activities in this use case.

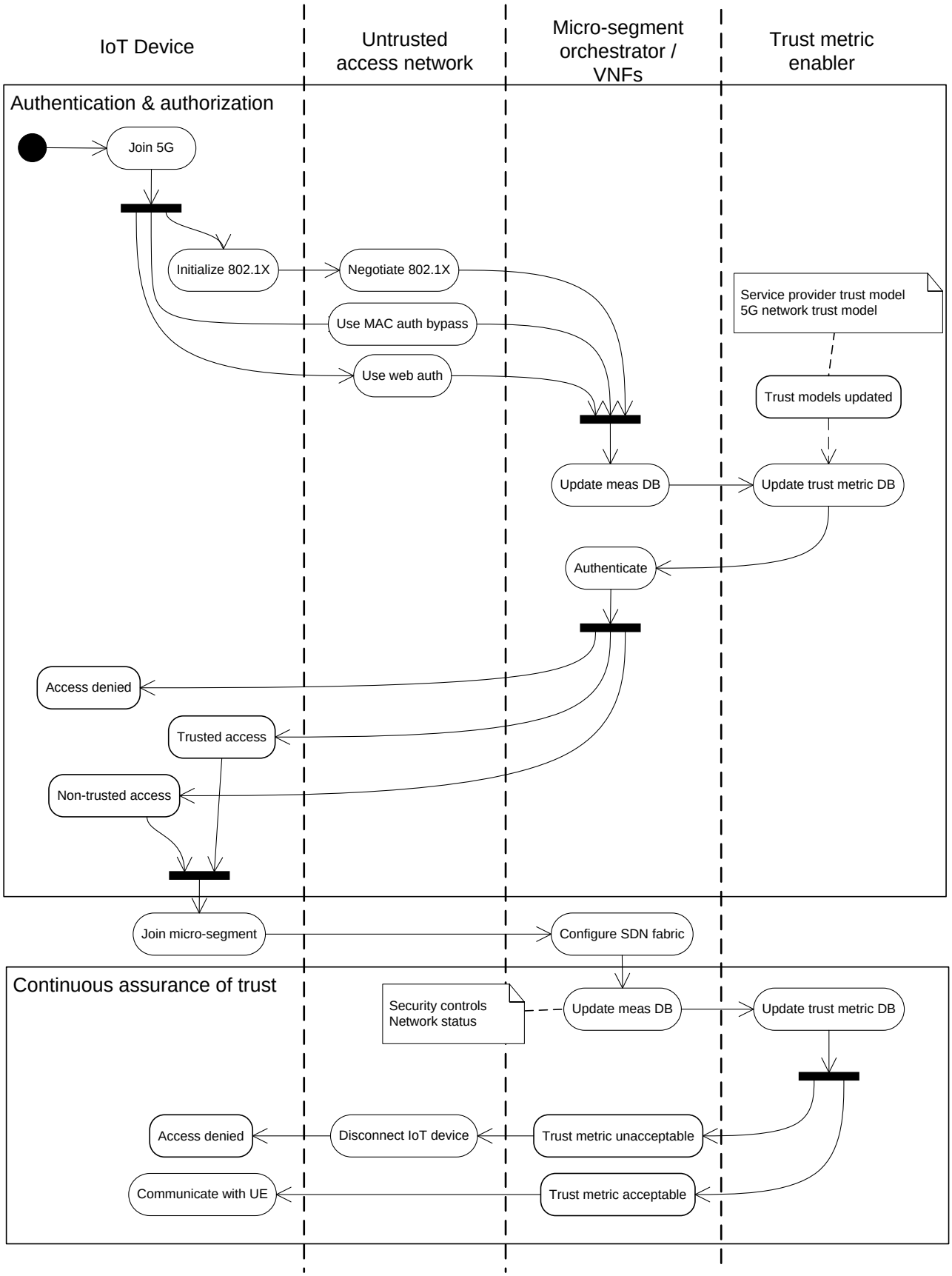


Figure 4-5: Use case example activity diagram

4.2.8.2 Components and interaction overview

The main components, APIs and interactions are presented in the following figures as discussed previously.

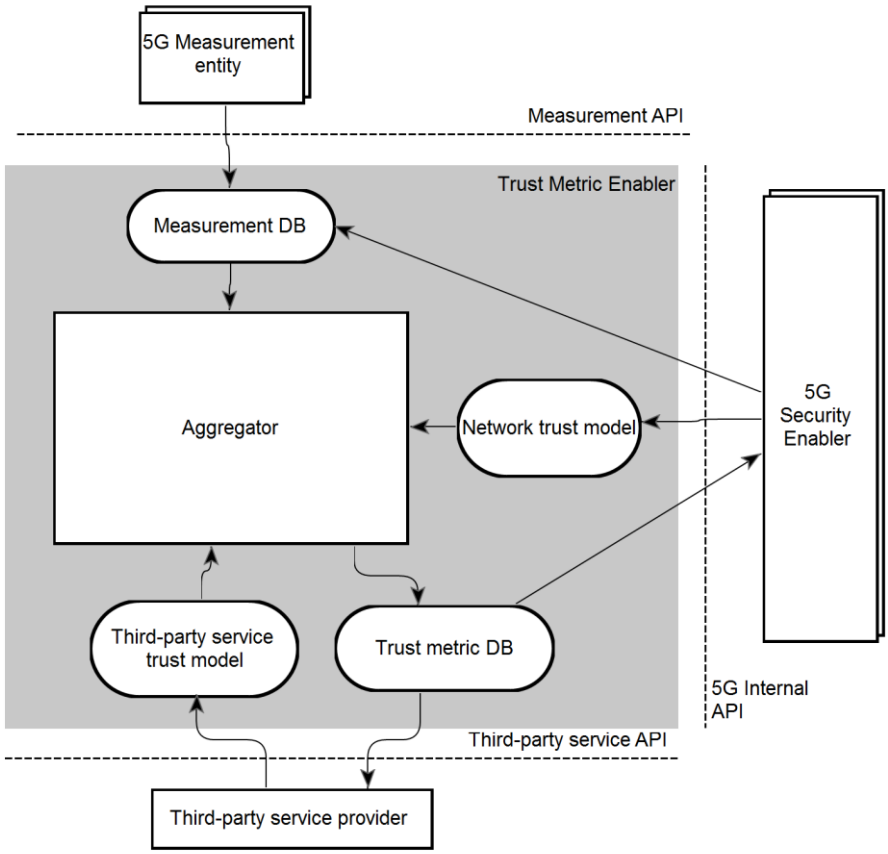


Figure 4-6: Main component diagram

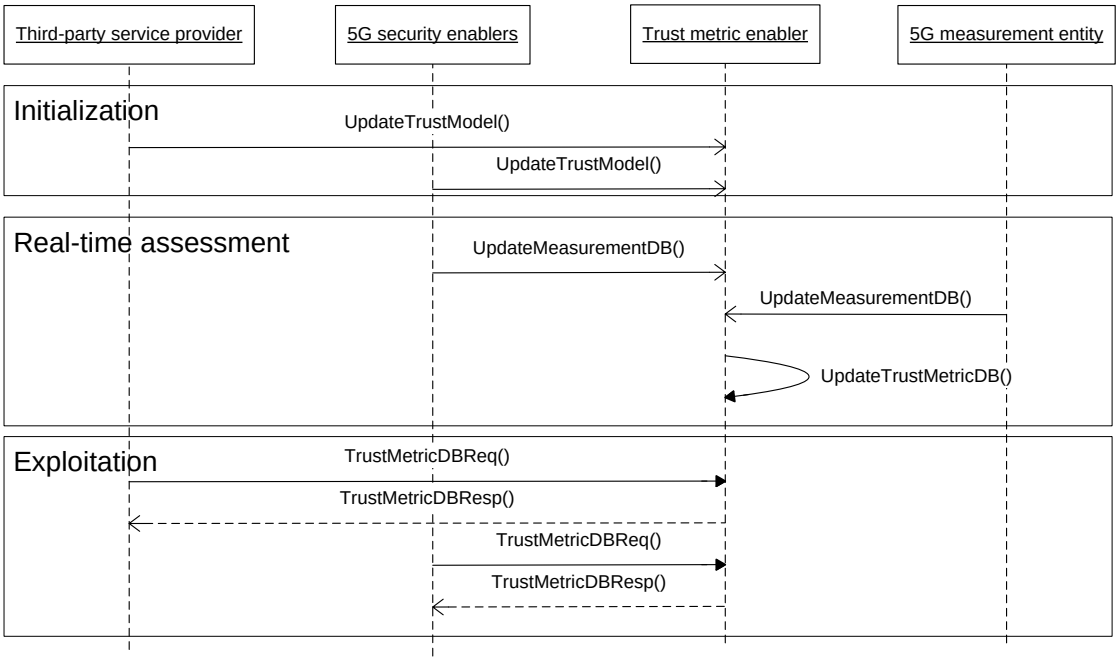


Figure 4-7: High level interaction sequence diagram

4.2.9 Architectural drivers

4.2.9.1 High-Level functional requirements

Third-party API

5G third-party (standard) API is required.

Measurement and 5G internal APIs

These APIs should be based on standard interfaces. However, it could be sufficient to rely on APIs defined for 5G security enablers such as Micro-segmentation Enabler (network capabilities) and Security Monitoring Enabler (security related measurements) in some use cases.

Producing trust metric values

Trust metric values are produced in Aggregator component based on trust models and measurements. The trust models include the high-level information to map measurements to trust values. It is essential to develop a well-defined API for this purpose. The enabler provides an abstraction layer to third-party by e.g. hiding the details of network requirement validation.

Dynamic network management

The third-party trust model could define network requirements that require dynamic network configurations, at least in the setup phase. The 5G system should facilitate dynamic network management e.g. through SDN and NFV or otherwise this functionality could be allowed only with Micro-Segment Enabler.

4.2.9.2 Quality attributes

The following list presents quality attributes are relevant when assessing the realizations of the enabler. The list also illustrates how the current framework plans support these attributes:

- Reliability – use of mature software technologies
- Operability – use of widely adopted and accepted software technologies;
- Performance efficiency – scalability through micro-segmenting to handle large amount of real-time events
- Security – advancing security through integrating trust models of the third party service provider and the network operators
- Compatibility – support for various standardized event sources and use of open interfaces for output format
- Maintainability – use of easily utilized programming languages and widely utilized technologies;
- Efficiency – capability to cover large variety of applications through one API

4.2.9.3 Technical constraints

5G network capabilities could restrict deployment of some use cases e.g. because of lack of dynamic network management. The feasibility of the Enabler is uncertain in use cases where the trust assessment of the whole end-to-end transport network cannot be ensured.

4.2.9.4 Business constraints

The business relationships between the entities need to be studied further. The basic assumption is that the third-party service provider purchases the service enabled by the Trust Metric Enabler (and other Security Enablers) from the MNO. The pricing could be based on the characteristics (e.g. network requirements and NFVs) defined in the trust model which is provided by the third-party.

4.2.9.5 Link to architecture

Trust Metric Enabler fulfils several of the security objectives defined in D2.4. First of all it supports Secure Virtualization (O4.1, O4.2) as well as its sub-objectives Slicing (O5.3) and IoT Security (O6.1, O6.2 and O6.3).

The Trust Metric Enabler should be implemented to core network's management domain and it is part of management stratum. Furthermore, this enabler belongs to Network Security Feature Group, which is also specified in D2.4.

4.2.10 Detailed specifications

Since the 'Trust Metric Enabler' is closely related to 'Micro Segmentation Enabler' (see Chapter 6.4 of this document) and 'Security Monitor for 5G Micro-Segments' (Chapter 5.3), its implementation is also close to these, and it will be based on similar technologies.

This section specifies the APIs and protocols utilized. The interfaces may be considered internal for the Trust Metric Enabler. External interfaces (including user interfaces) for controlling the behaviour of inference layer will be specified.

In testing Trust Metric Enabler in a mobile networking architecture, there needs to be a way to integrate the different functions of mobile networks (e.g. PGW, SGW, MME, PCRF) with SDN and network virtualization technologies. One possible solution could be to use the OpenEPC framework [1] that covers all the functional elements in the 3GPP Evolved Packet Core (EPC) and Systems Architecture Evolution (SAE) technologies up to 3GPP Release 12. Another alternative is the OpenAirInterface [2].

4.2.10.1 Conformance

The implementation of the enabler relies on SDN and virtualization technologies. For SDN, an SDN controller that can be chosen quite freely is needed. In this work we selected the Ryu SDN controller [3]. For network virtualization, OpenVirteX software [4] is used, but another type of network hypervisor is also plausible.

4.2.10.2 API specifications

The Release 2 version of Trust Metric Enabler takes the following inputs:

- Third party service provider's requirements for trust
- "Measurements" such as
    - Validated network characteristics
        - micro-segmentation supported
        - IDS implemented
        - MAC Authentication Bypass (MAB) allowed
    - Number of active IoT Devices
    - Number of active UEs
    - End-to-end encrypted traffic denied (only if IDS is implemented)
    - Subscribed IDS alerts

- High severity (i.e. indicates attacks and malware)

The third party's requirements for trust are always presented as a list of key-value pairs. The key is the name of the required capability and numerical value is used for quantitative requirements. In case of qualitative requirements (where the feature must or must not be provided) the value is 0 or 1 (True or False). The Trust Metric Enabler receives trust requirements as an Excel-file (as in Release 1) or as JSON delivered through Kafka broker [5]. Service provider's trust requirements may change at runtime. The changes are supported by allowing new trust models to be added to the system dynamically. The basic input data which characterizes the network's capabilities (micro-segmentation supported, IDS implemented) is obtained through the Security Monitor for 5G Micro-Segments Enabler. Furthermore, it would be possible to utilize the Component-interaction Audit Enabler, whose verdicts of policy violations in a micro-segment could set trust to zero. Just as well, the security monitoring that the Generic Collector Enabler performs could be used for trust metrics, and respectively the Trust Metric Enabler's log files could be input for the Generic Collector Enabler.

The enabler produces a binary value (0,1) output based on the input actions:

- Value 0 (trust metric unacceptable)
- Value 1 (trust metric acceptable)

4.2.10.3 Examples

Release 2 of the enabler supports dynamic exchange of trust models and run-time trust assessments. **Figure 4-8** provides an example illustrating how messages are exchanged in run-time scenario. In the scenario, an event Kafka broker is used to share both trust models (requirements), trust metrics, as well as network capability/monitoring information (measurements). The figure illustrates how Trust Metric enabler is related to Micro-Segments and Micro-Segment Monitoring. Trust Metric Enabler is in this case be used to assess trustworthiness of micro-segments for the clients (third-party service providers or other operators creating end-to-end isolated slices)

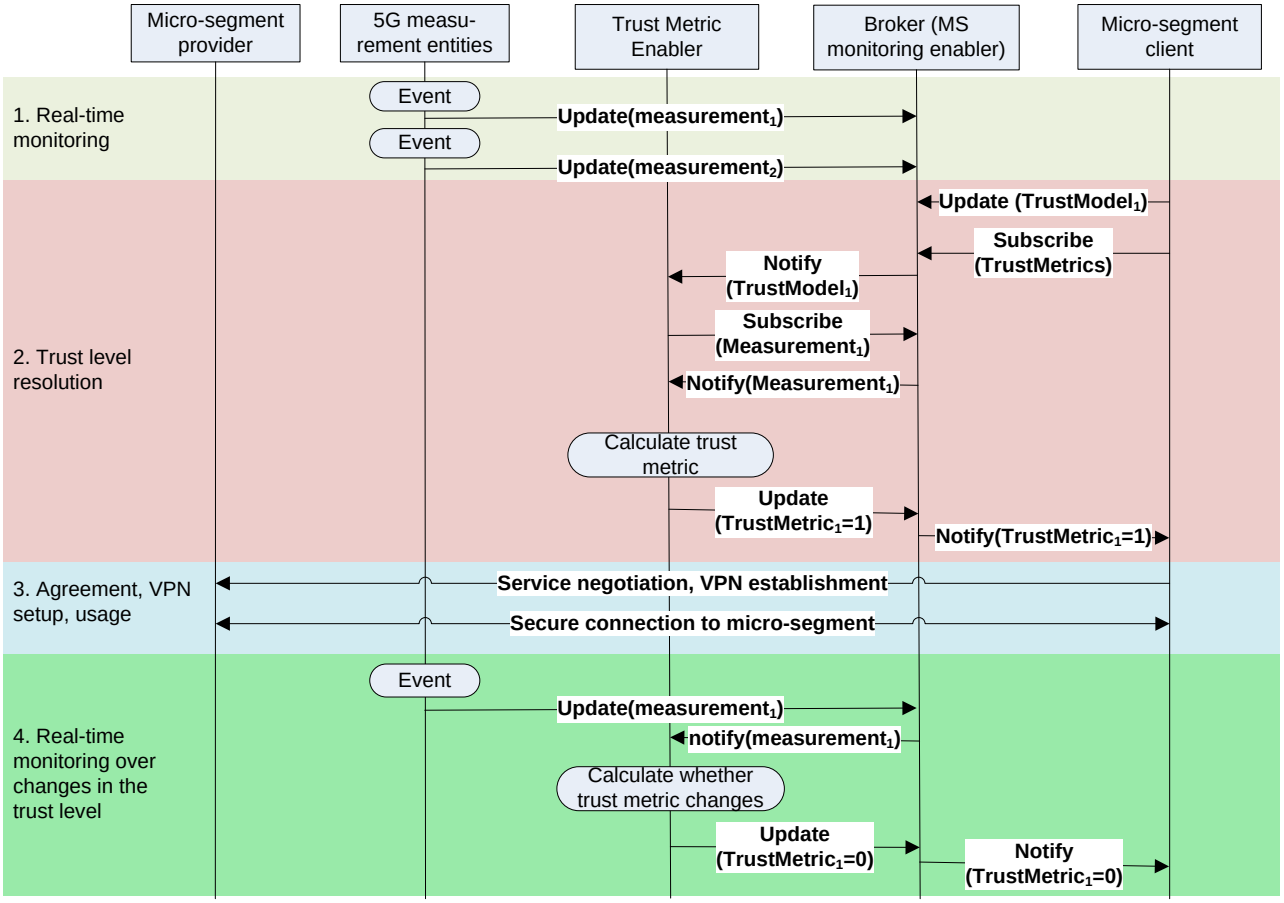


Figure 4-8. Messaging example for brokered trust level agreement and monitoring

Sharing of Trust Metrics (using the broker based architecture as presented in **Figure 4-8**) consist of the following steps:

1. At the beginning, the enabler creates Kafka topics ("event streams") for inputting Trust Models and for outputting Trust Metrics. Different information sources open also own topics for delivering monitoring information (measurements).
   An entity wishing to use micro-segment subscribes 'TrustMetrics' topic. This topic is available for third-parties. It will contain only binary information on whether Trust Model is fulfilled or not and therefore does not reveal detailed information on networks capabilities to outsiders.
2. Clients needing information on Micro-Segments trust worthiness deliver their requirements by sending their trust models in an update to the TrustModel topic. The trust model may look e.g. like this:

Kafka topic: [<micro-segment identifier>.TrustModels]
{ "ClientID": "identifierX", "IDS": "1", "MAB: "1", "maxUE": "100" }

   Trust metric enabler gets a notification and by looking on the requirements subscribes necessary measurements from the broker. After receiving measurements, the enabler looks are trust models

requirements met. If they are it will send a notification through the broker that TrustMetric for this model is 1. In JSON/Kafka the values could be presented e.g. in the following manner:

Kafk topic: [<micro-segment identifier>.TrustMetric]
{ "ClientID": "identifierX", "TrustMetric": "1" }

3. Now the client knows that the micro-segment fulfils its requirements and it can proceed to set up a secure connection and start using forwarding traffic flows to the micro-segment. The setting up of the service may require additional negotiations which are not shown in the figure.
4. The trust level of the micro-segment is monitored at run-time. Trust Metric Enabler gets notification on relevant events and each time a notification arrives it checks whether trust models requirements are still valid. In case, trust model is no longer satisfied, it will inform the client.

4.2.11 References

[1] "OpenEPC framework," [Online]. Available: http://www.openepc.com
[2] "OpenAirInterface," [Online]. Available: http://www.openairinterface.org
[3] "Ryu SDN framework," [Online]. Available: https://osrg.github.io/ryu
[4] "OpenVirteX Network Virtualization Platform," [Online]. Available: http://ovx.onlab.us/
[5] "Apache Kafka," [Online]. Available: http://kafka.apache.org/

4.3 VNF Certification enabler Open specifications

4.3.1 Preface

The goal of this enabler is to certify the trustworthy implementation of the VNF and to expose their characteristics through a Digital Trustworthiness Certificate

4.3.2 Status

The version 1.0 of this component is currently under development.

4.3.3 Copyright

4.3.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

4.3.5 Terms and definitions

VNF	Virtualized network functions
DTwC	Digital Trustworthiness Certificate
OPTET	Operational Trustworthiness Enabling Technologies
VMNO	Virtual Mobile Network Operator

4.3.6 Overview

The enabler can be used in a variety of use cases and by a variety of users, both when designing new network configurations enabled by 5G technologies and when making trust decisions during the use and operation of networks

This enabler is expected in Release 2, to provide a complete prototype for the VNF certification and for the Digital Trustworthiness Certificate.

4.3.7 Basic concepts

The goal of this enabler is to provide, through a certification process, a Digital Trustworthiness Certificate (DTwC) containing trustworthiness information (evidence with metrics values) about the VNF. Using this Certificate, the infrastructure owner could decide whether to deploy this VNF into one's infrastructure by comparing one's trustworthy infrastructure requirements against the trustworthy information contained in the DTwC of the VNF. The certification process is intended to include testing phases, so the certification is not based solely on the claims made by the VNF owner.

This certification enabler has two main objectives:

1. Help the certifier to collect evidence about the VNF and to subsequently produce a DTwC.
2. Provide a centralised service where all the DTwCs produced will be stored.

These two objectives lead to the fact that this enabler will be used differently following the life cycle of the NVF. Indeed, the certification of a VNF is more at "design time" when the VNF is finished at the developmental level and ready to be deployed. Whereas making the different DTwCs available is more at runtime, when the platform owner decides whether to deploy a new VNF into one's infrastructure.

The DTwC is the result of a certification process dedicated to the certification of VNF and based on existing processes like the ETSI NFV-SEC evaluation process[8]. It is worth noting that many of the security certification programs view security broadly, i.e., the assessment includes also organizational functions to ensure that the organization is also managed in a secure manner. This can also include assessing the software development practises. It is for further study whether such elements could be included into the DTwC process as well.

This process involves three different stakeholders:

- The VNF owner in charge of providing the initial information and the VNF. At the end of the certification process, the VNF owner has the right to decide to publish the certificate.
- The evaluator in charge of collecting evidence about the VNF and to provide the evaluation results to the certifier.
- The certifier in charge of validating and consolidating the evaluation results and producing the DTwC containing certified evidence and some VNF description. For that, the certifier could use the VNF certification enabler which provides through certification process defined a User Interface. This User Interface helps the certifier to follow the different steps of the certification process.

At the end, a DTwC is generated for a specific VNF and stored into a centralised server to be accessible by the different infrastructure owners interested in finding trustworthiness information about it, if the VNF owner authorised the publication. This DTwC is an XML file signed by the certifier and linked with the VNF involved in this certification process by the inclusion of the hash of the VNF.

The second part of this enabler is the centralised server containing all the DTwCs generated by the different VNF certifications and the certificate of the certifier used to sign the DTwC. This centralised server provides

[8]See the http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/003/01.01.01_60/gs_NFV-SEC003v010101p.pdf document

a secured interface to upload the DTwCs and, also, an interface to search for some specific DTwCs using the hash of the VNF (The DTwC is linked with the VNF using the hash of this. This Repository must be managed by a third party not involved in the certification process and independent of the VNF owner. It could be a state institution or an independent organization.

4.3.8 Main interactions

4.3.8.1 Use cases

The virtualization of network functions and network equipment enable to instantiate several of them on commodity servers, thus sharing physical resources (CPU, RAM, memory and network) with other hosted virtual machines. Nowadays, the infrastructure provider manages its own VNFs on its own infrastructure. In case a VMNO wants to use a proprietary VNF (developed by another VMNO for example), how could a VMNO actor provides trustworthiness assurance to the infrastructure provider? How the infrastructure provider could trust a VNF coming from a third party?

In a simple use case, the different actors involved into the certification process are:

- The VNF owner who provides the VNF and also some technical information about the implementation of it. He must also validate the publication of the certificate.
- The evaluator who is in charge of evaluating the VNF to extract trustworthiness evidence. The result of the evaluation will be used by the certifier to produce the DTwC. The evaluation could be realised by inspection or using automatic analysis tools to extract independently information about the VNF. For a VNF evaluation, there could be different evaluators providing different kind of evidences.
- The certifier who collects information regarding the VNF in order to fill the trustworthiness elements of the DTwC. He is responsible of the validity of the information set into the certificate. For that, he must verify the different information provided by the VNF owner and evaluators. For a VNF certification, only one certifier is involved into the certification. However, we could have different certifiers authorized to certify VNF (dependent, for instance, of their area of expertise).
- Each certifier must have a certificate (P12 certificate) delivered by a certification authority. This certificate is used to sign the DTwC and also to authenticate the certifier during the transfer of the DTwC to the DTwCs repository.
- The VNF consumer (here the infrastructure provider) who uses the DTwC to know more trustworthiness information about the VNF he wants to deploy. Using this DTwC, he could compare the information against his internal deployment policies and decide whether to deploy or not the VNF into his infrastructure. He could also verify who has performed the certification using the certificate of the certifier associated with this DTwC.

VNF owner and VNF consumer need to trust certifier to provide trustworthy services. This can take place either through contractual business relationships or through industry agreements (i.e. commonly established certification body). The main use case is the deployment of a new VNF into an NFV infrastructure where the basic flow of events is the following:

- The VNF owner wants to propose his VNF to be deployed by different VNF consumers. For that, he must, beforehand, provide the VNF element and some information about it to the certifier and to the different evaluators.

- The evaluator(s) will extract evidence from the VNF by inspection or by tests in order to provide trustworthiness evidences to the certifier

- Using the different elements provided by the VNF owner and the evaluator(s), the certifier will produce a DTwC representing the trustworthiness of the VNF.

- If the VNF owner authorizes the publication, the certifier will upload this DTwC into a centralised service containing all the certificate of all the VNFs. This service is unique and could be replicated for security and performance aspect.

- In the next step, when a VNF consumer wants to deploy a VNF, he could find the DTwC for it in the centralised service and decide whether to deploy (or not) this element into his infrastructure by comparing the trustworthiness elements provided by the DTwC with the trustworthiness requirements of his platform.

This simple use case could be integrated into the different use cases of D2.1, like 5.2 “Adding a 5G Node to a Virtualized Core Network”, 5.4 “Verification of the Virtualized Node and the Virtualization Platform“ or 9.3 “Authentication of New Network Elements “. For example, in the 5.4 “Verification of the Virtualized Node and the Virtualization Platform“ use case, the DTwC certificate could be used to validate the integrity of the VNF or know the different threats and controls around this VNF before the deployment of a new instance.

4.3.8.2 Components and interaction overview

The following diagram describes the logical architecture of the VNF Certification enabler. It covers the whole certification lifecycle.

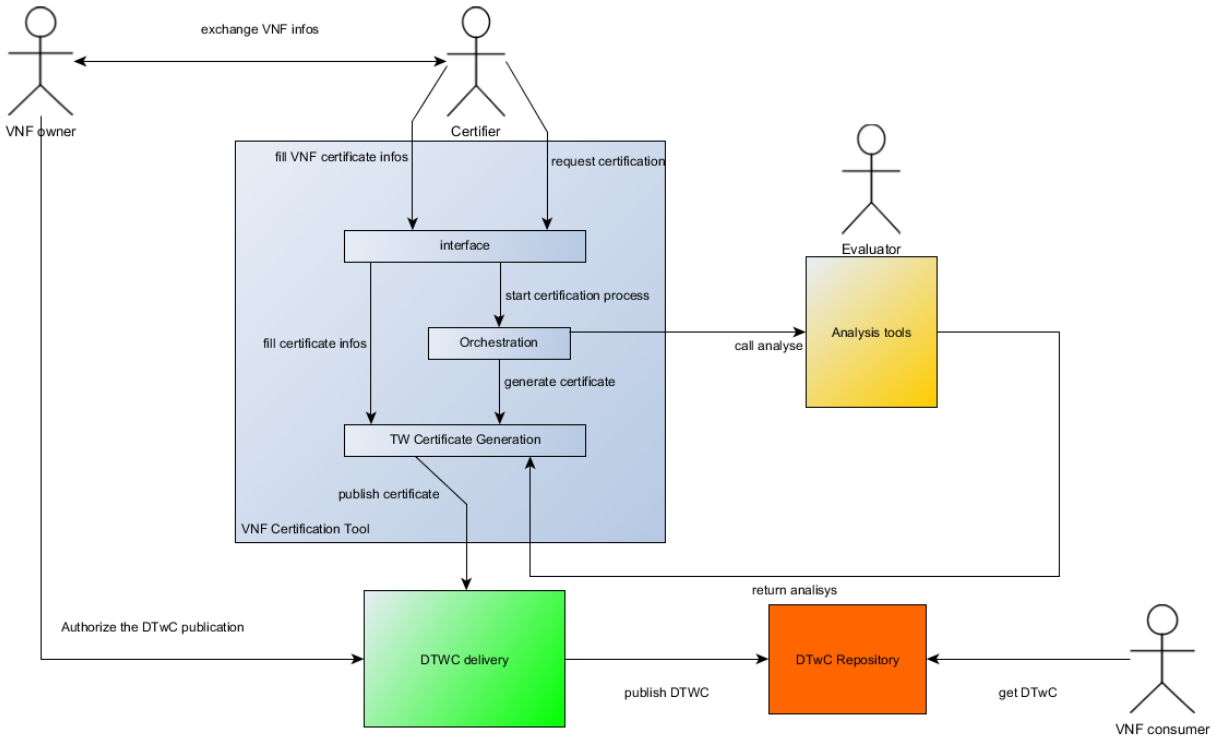


Figure 4-9: Logical architecture of the VNF Certification enabler

In this section we describe the logical components.

- The VNF Certification Tool: this component manages the trustworthiness evaluation of the VNF. It is composed by three main elements,
    - the user interface which must guide the certifier into the certification process,
    - the orchestration element which manages the different evaluations involved in the certification process
    - and the Certification generation element in charge to collect the trustworthiness information provided by the certifier and by the different evaluations to produce the DTwC.
- The Analysis tools: These elements, managed by evaluators, represent a set of external tools called by the certification to retrieve information about the VNF. Each analysis tool must return the result of its analysis to the certification tool. Then each result will be validated and inserted into the DTwC during the generation. This validation and insertion could be automatic or manually interpreted by the certifier (following his experience and the other evaluation results)
- The DTwC delivery: This element is in charge of uploading the DTwC generated to the DTwC repository in a secured way after validation of the VNF owner.
- The DTwC repository: This element stores and manages the different DTwCs generated by the different certifications and the certificates of the certifier associated to each DTwC. It also provides an interface to retrieve certificates (DTwC and the certificate of the certifier) for a specific VNF.

Regarding the different workflows, we have the generation of the DTwC and the usage of this DTwC. Sequence diagrams are given below to clarify how these components interact. The list of diagrams is not exhaustive but assists in understanding the links between actors and logical components.

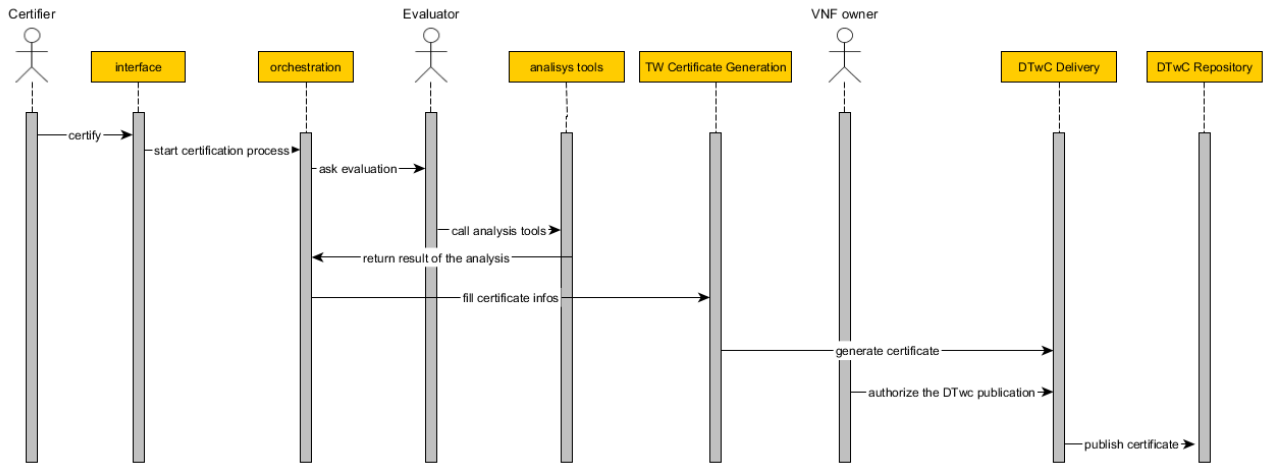


Figure 4-10: DTwC generation

This sequence diagram depicted on **Figure 4-10** focuses on the DTwC generation:

- The certifier, through the Certification platform interface, asks the evaluators to collect the evidences using analysis tools or fill manually information regarding the evaluation of the VNF

- The evaluator analyses the VNF (by inspection or by testing) and returns the result to the DTwC generator element.

- The values are verified and consolidated in order to be pushed into the DTwC

- The certifier generates the DTwC using the information collected before and inserts security information to link strongly the certificate, the VNF and the certifier (Hash and signature). If some evaluations can't be realized, the associated metric presented into the certificate will be put to undefined; otherwise, a metric value will be set to characterize the evidence

- The certifier publishes the DTwC into the centralized repository, after authorization of the VNF owner, in a secured way (the repository must authenticate the certifier in order to accept the DTwC published). He must publish also his certificate in order to give the possibility to each to know who has performed the certification.

This second workflow shows the interaction between the VNF consumer and the DTwC repository.

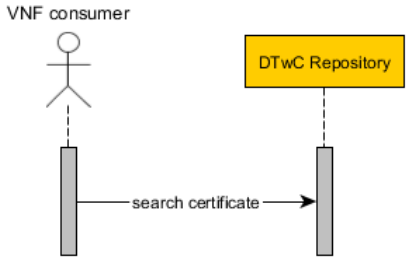


Figure 4-11: Interaction between the VNF consumer and the DTwC repository

In this workflow, the VNF consumer:

- Calls the DTwC repository to retrieve the DTwC of the VNF he wants to deploy. This search is not secured (at the authentication level but the communication must be secured using TLS) because the DTwC are considered as public and the information inside the certificate are validated and authorized to be published through the authorization of the VNF ownerArchitectural drivers

4.3.8.3 High-Level functional requirements

The main feature of this enabler is the VNF trustworthiness evaluation. This feature could be described by two functions: the certification of the VNF implementation and the exposition of their characteristics through a DTwC.

Into the overall NFV architecture, the VNF certification enabler has the responsibility to provide trustworthy information to the NFV Orchestrator in order to give it the opportunity to understand and verify the VNF it wants to deploy.

During the deployment workflow realized by the orchestrator, it selects VNFs into its VNF catalog. But before to deploy them, it could/must verify if the VNF selected is in line with its trustworthiness requirements. For that, the orchestrator must ask the DTwC repository in order to retrieve the DTwC associated with this VNF (if it exists). With this certificate, the Orchestrator could check:

- Who has performed the certification (using the certificate of the certifier associated with the DTwC)
- The trustworthy attributes characterizing this VNF like some security or reliability elements. For instance, he could validate the integrity of the VNF or check is the VNF is hardened.

If the trustworthy characteristic of the VNF is not sufficient, the orchestrator could decide to deploy another VNF element more adapted.

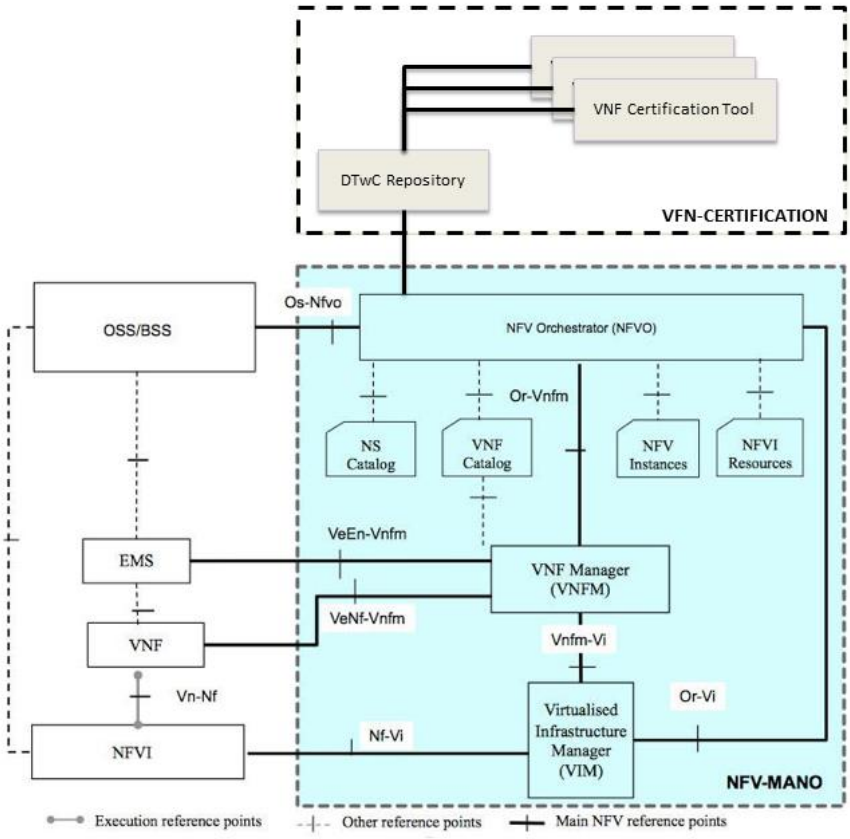


Figure 4-12: NFV context of the enabler

4.3.8.4 Quality attributes

The following list presents quality attributes which are relevant when assessing the realization of the enabler. The list also illustrates how the current framework plans to support these attributes:

- Operability- The User Interface provided to generate the certificate must be "user friendly" to guide the certifier into the certification process
- Reliability - mature software technologies must be used
- Security - The publication of DtWC is not limited to authorized persons. In addition, the security mechanisms put in place to secure the DtWC must guarantee the integrity, the accountability and the non-repudiation of it.
- Compatibility - The Trustworthiness elements used to characterize the VNF must be compatible with standards like ETSI or 3GPP.

4.3.8.5 Technical constraints

4.3.8.5.1 The evaluation:

To find evidences to characterize the trustworthiness of a VNF, the different possibilities are:

- The blackbox testing (e.g., fuzzing) and/or source code analysis if available
- The deployment of probes into the VNF
- The technical information coming from the VNF owner

The selection of methods is up to the certifier and the evaluators. The different methods have their interests and combined together, provide information to characterize the trustworthiness of an element. However, the realization of probes and tests for VNF is complicated due to the heterogeneity of the different VNFs (different functionality and implementation). More, the deployment of probes directly into the VNF could be a problem regarding deployment due to some security constraints imposed by the VNF.

The blackbox testing does not necessarily reveal all the possible vulnerabilities. If this is complemented with source code analysis, then the coverage of testing is more complete. However, source code analysis can be time consuming.

4.3.8.5.2 The certification process

Also, the different elements provided by the VNF Owner must be validated to be taken into account in the certification process. It's the responsibility of the certifier to validate all the evidences collected about the VNF before the insertion into the certificate.

Additional considerations can be given to the possibility of assessing also the development practices of the VNF owner. This could also be performed by complementing the certification process with other applicable certificates the organization possesses, e.g., ISO 27001 or PCI DSS (one ought to make sure that the scope of the certification matches the scope of VNF owner development activities).

Another consideration relates to the new versions of a specific VNF as bug or feature updates are likely to invalidate the certificate. Applying the whole process to each version of VNF might end up being too restrictive. Some alternative approaches (such as used in Mirrorlink DAP certification) include the possibility to provide a certification hierarchy, which certifies the VNF owner, who then has the possibility of issuing its own certificates. However, this makes system more complex and the assessment process ought to also ensure that VNF owner is capable of sound key management practices.

4.3.8.6 Business constraints

The business relationships between the different entities need to be further studied. Especially the potential liability aspects that might arise from misbehaving VNFs need to be solved. VNF consumer might need to make a risk assessment, whether to rely on testing done by another entity or whether to conduct its own testing. It is likely that at least VNF owner needs to have a business contract with the certifier as there likely is a cost involved (and evaluators most likely wish to be compensated).

The certifier could be a commercial entity or an industry body. One needs to study whether several such entities exist or whether there is just one. As this has regulatory implications, so it is more likely that several entities will exist. Also, evaluators could be separate entities from the certifier, thus it has to be decided what sort of requirements are set for these. One possibility is to expect them to be accredited ISO 27001 auditors or other similar accreditations.

4.3.8.7 Link with architecture

This section is about how the proposed enabler is related to the proposed security architecture defined inside the project.

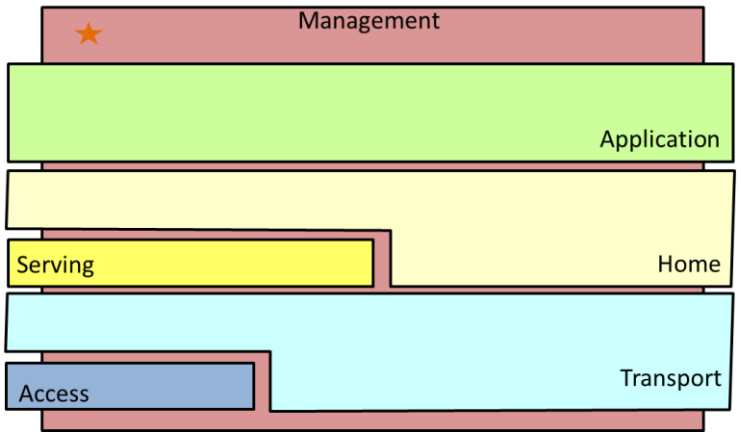
4.3.8.7.1 Security objectives fulfilled by the enabler

The enabler contributes in accomplishing the following objectives:

- O1.1: Based on NFV architecture, this enabler contributes by decoupling 5G security from specific physical deployments.
- O4.2: This enabler is compliant with a dynamic scalable deployment of the 5G Security.
- O7.5: This enabler offers a way to mutually assess the trustworthiness before, and during interactions.

4.3.8.7.2 Strata location of the enabler

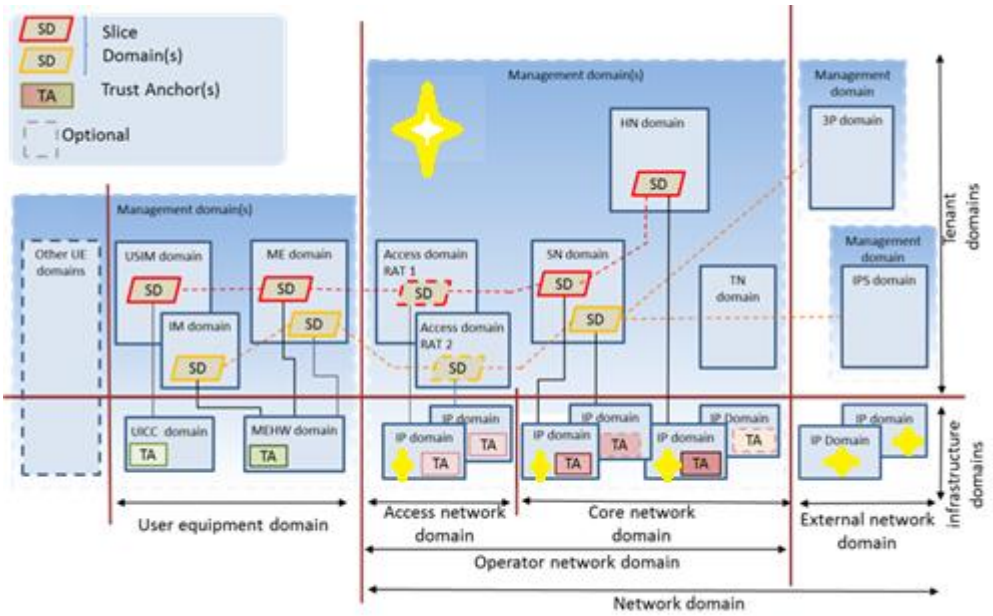
This enabler is conceived to be at the management Stratum because it manages the selection of the VNF to be deployed in the different infrastructures. The enabler may be connected to other elements that manage the virtualized infrastructure such as the SDN.



4.3.8.7.3 Enabler mapping to architecture domains

The *core network domain* is the optimal location for the enabler. This enabler can be located at the *operator network domain*, the *external network domain* or both. The figure below show the main possibilities for this enabler.

Regarding the 5G domains Security architecture enforcement, this enabler will play an important role in *Infrastructure provider domain*. Another aspect is that it helps one of the *compound domains* which is the *Administrative domain* by registering the VNF in NFV infrastructure.



4.3.9 Detailed specifications

4.3.9.1 Introduction

The implementation of this enabler imposes the API specification to be compliant with and also to use the same DTwC format specified in the next paragraph.

4.3.9.2 Conformance

All the interfaces described by this specification are mandatory and must be implemented in order to be compliant with.

The usage of DTwC model is also mandatory in order to have the same level of information for all the certifications. This DTwC concept represents the basic element for the definition of a certificate instance. Four concepts are part of its definition: SystemDescription, TWProblemDefinition, TWPropertySpecification and Evidence.

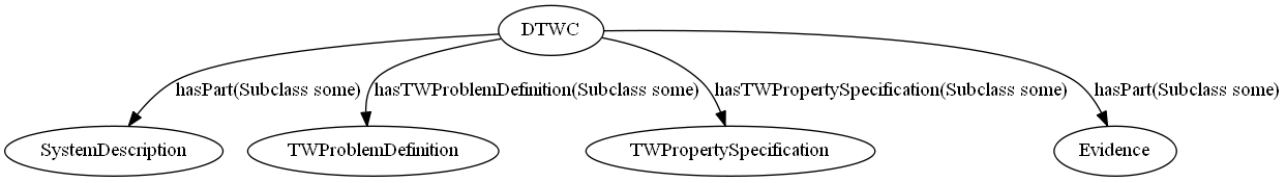


Figure 4-13: DTwC model concepts

Let us start considering SystemDescription

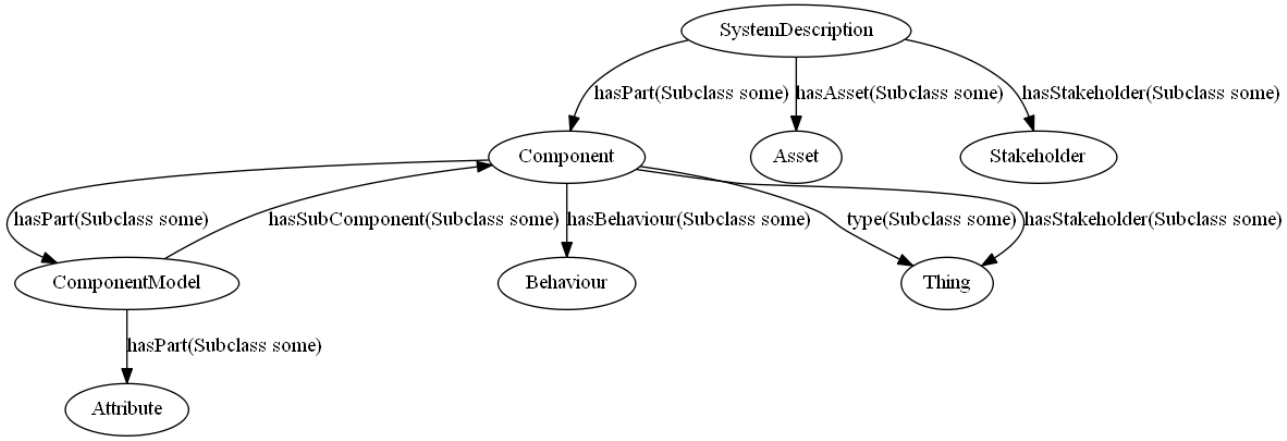


Figure 4-14: System description

The SystemDescription concept allows for detailing the NVF component and the system being certified. The three main elements of a SystemDescription are Stakeholder, Component and Asset. The Stakeholder concept can be used to indicate the stakeholders (or actors) having a role in the system being described. It is possible to detail the system's architecture through a set of Components, and through the creation of a ComponentModel instance, it is possible to illustrate all subcomponents of a Component. The Asset concept allows for declaring which Components are verified by the Certification Authority.

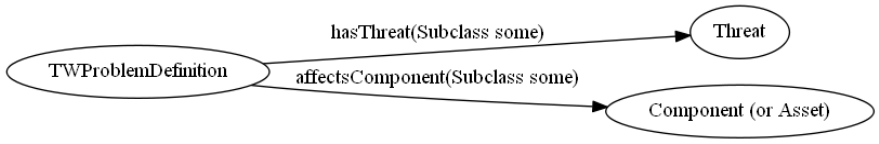


Figure 4-15: TWProblemDefintion concept

The TWProblemDefinition concept aims at illustrating how a threat affects an Asset

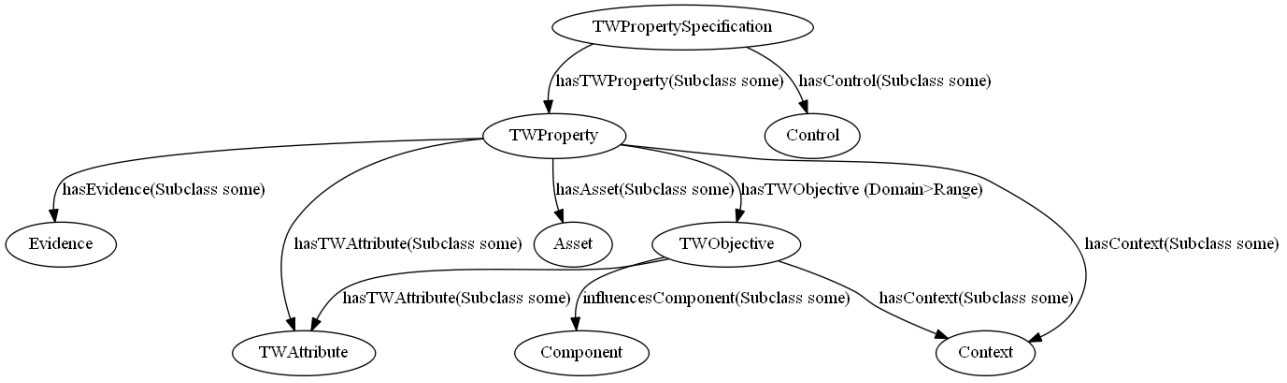


Figure 4-16: TWPropertSpecification

The TWPropertySpecification is a concept that allows for the expression of TW characteristics of Assets: its constituting elements are a TWProperty and a Control. TW Property links an Asset to a TW Attribute and a

Context that further specifies the TW Attribute.The Control concept allows for the indication of a mechanism that is able to influence or enforce the TWProperty.

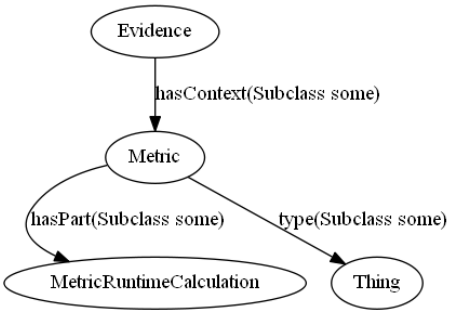


Figure 4-17: TWPropery concept

The Evidence concept captures any form of proof that is deemed sufficient by the Certification Authority. The nature and form of an Evidence can vary significantly, according for instance to the type of TWProperty it demonstrates.

4.3.9.3 API specifications

This enabler is composed of two parts, the User Interface helping the certifier to produce the DTwC, associated with evaluation tools and the DTwC repository which centrally stores all the DTwCs produced and gives the possibility to retrieve these.

- The Certification UI: This first element is a UI with no specific API. In the reference implementation, this Certification UI will be provided like an eclipse plugin
- The DTwC repository: This second element provides an API to upload the generated DTwC associated with the certificate of the certifier and to download a specific DTwC or the certifier certificate associated. In the reference implementation, this element will be provided like a REST service hosted into an application server (like Tomcat)

The different APIs are:

- Push DTwC

This method is used to push the DTwC and the certifier certificate into the repository. This needs to check the integrity of the upload element in order to ensure that no malicious content gets stored in the repository.

```
Method: POST
Path: /uploadDTwCs
Headers:
    o  Content-Type:multipart/form-data
    o  Accept: text/html,application/xhtml+xml,application/xml
```

- Get a DTwC

To search for a specific DTwC, the method is to search using the hash of the VNF

The research by hash

```
Method: GET
Path : /downloadDTwCByHash/{hash}")
```

- Get a Certificate used to sign the DTwC

To search the certificate used to sign the DTwC, , a query using the hash of the VNF must be used

The research by hash

```
Method: GET
Path : / downloadCertificateByHash/{hash}")
```

4.3.9.4 Examples

Some examples of requests for the previous requests

- Push DTwC

  An example of a push request

```
POST /CertificationRepository/rest/files/uploadDTwCs HTTP/1.1
User-Agent: curl/7.35.0
Host: localhost:8444
Accept: */*
Content-Length: 43278
Expect: 100-continue
Content-type:multipart/form-data; boundary=------------------------
7a2d881d97c6bc24
Content-Disposition: form-data; name="file";
filename="certificate_a_20170224_133500.xml"
Content-Type: application/xml

--------------------------7a2d881d97c6bc2
Content-Disposition: form-data; name="certificate"; filename="certificate.pem"
Content-Type: application/octet-stream
```

And the possible responses are:

HTTP/1.1 200 OK

Indicates that the files entry was successfully created. If an error occurs, this header will contain one of the error codes below.

Error codes

- HTTP/1.1 401 Unauthorized
- HTTP/1.1 403 Forbidden
- HTTP/1.1 404 Not Found
- HTTP/1.1 409 Conflict
- HTTP/1.1 415 Unsupported media type

- Get DtWC

An example of research request using hash

```
GET /downloadDTwCByHash/qiyh4XPJGsOZ2MEAyLkfWqeQ?type=file HTTP/1.1
```

And the possible responses are:

Returned HTTP Headers

HTTP/1.1 200 OK
Indicates that the file entry was successfully downloaded. If an error occurs, this header will contain one of the error codes below.

Error codes

- HTTP/1.1 401 Unauthorized
- HTTP/1.1 403 Forbidden
- HTTP/1.1 404 Not Found

- Get the certificate of the certifier used to sign the DTwC

An example of research request using hash

```
GET / downloadCertificateByHash/qiyh4XPJGsOZ2MEAyLkfWqeQ?type=file HTTP/1.1
```

And the possible responses are:

Returned HTTP Headers

HTTP/1.1 200 OK
Indicates that the file entry was successfully downloaded. If an error occurs, this header will contain one of the error codes below.

Error codes

- HTTP/1.1 401 Unauthorized
- HTTP/1.1 403 Forbidden
- HTTP/1.1 404 Not Found

4.3.10 Re-utilised Technologies/Specifications

This enabler specification is based on the technologies already developed in the context of the OPTET European project http://www.optet.eu/. This project provides an equipped certification process which must be adapted to the certification of VNF by adjusting the scope of the certification and the process himself.

This specification follows the “security and trust guidance” defined by the ETSI NFV-SEC. This document provides security and trust elements to characterise the trust and the security aspect at the VNF level. This list of element and methodology are also completed by some information coming from the 3GPP working group around the Security Assurance Methodology (SECAM)

4.3.11 References

[1] http://www.optet.eu/documents/

[2] http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/003/01.01.01_60/gs_NFV-SEC003v010101p.pdf

[3] http://www.3gpp.org/DynaReport/33805.htm

4.4 Security Indicator Open specification

4.4.1 Preface

This section represents 'security indicator' enabler. The enabler increases trustworthiness of 5G networks by extending security indicator capabilities of current 2G, 3G, and 4G networks.

4.4.2 Status

The version 1.0 of this component is currently under development.

4.4.3 Copyright

4.4.4 Legal Notice

The legal notice that applies to these specifications is given in Annex A.

4.4.5 Terms and definitions

A5 – Ciphering algorithm
IMSI - International Mobile Subscriber Identity
TMSI – Temporary Mobile Subscriber Identity
UE- User Equipment

4.4.6 Overview

According to 3GPP TS 22.101, ciphering indicator feature allows the UE to indicate ciphering (user plane) to the subscriber. This feature, however, may be disabled by the home network operator. Example of this feature is depicted in Figure 5.1.

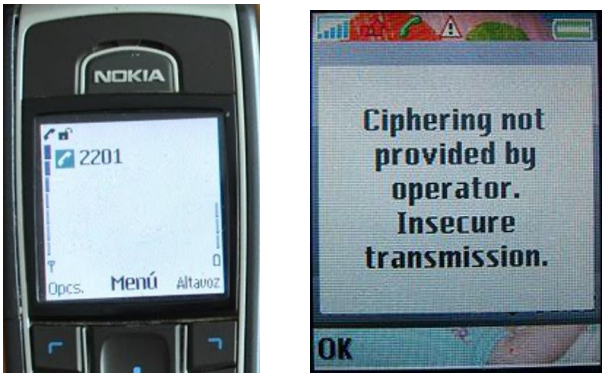


Figure 5.1: Ciphering indicator on few Nokia phones [2,3]

The ciphering indicator feature only displays status of encryption in the form of ON or OFF to be displayed on the screen. However, other security capabilities such as authentication state, temporary identities, ciphering algorithms used by the serving network are not displayed to the subscriber.

4.4.7 Basic Concepts

In this section, we provide the definition of ciphering, authentication, temporary identities in the context of 'security indicator' enabler.

Authentication: In 2G networks, the serving network authenticates UE before offering mobile services. Whereas in 3G and 4G, both serving networks and the UE mutually authenticate to each other. For every incoming or outgoing mobile services, such as call, text or data services such type of authentication process is executed by the network. However, the state of the authentication process is not displayed on the UE.

Ciphering: After authentication step, the serving network and UE establish a secure (encrypted) session to protect against eavesdropping attacks. Every network system uses different ciphering algorithms, for example in GSM A5 algorithms are used. Some variants of the A5 algorithm such as A5/2 and A5/1 are weak. Consequently, the A5/2 ciphering algorithm is no longer supported in GSM. However, the weaker A5/1 ciphering algorithm is still being used by network operators. The ciphering algorithms used in 3G and 4G networks are not vulnerable to any known attacks as of yet. However, according to [1] the UE does not indicate type of ciphering algorithm used by the serving network.

4.4.8 Main Interactions

4.4.8.1 Use cases

The UE indicates the security capabilities of the serving network such as the state of authentication, ciphering and random identities. The enabler indicates to the user if incoming or outgoing calls or texts are not ciphered. It will also notify the user if the random identities are not being updated by the serving network even after a successful authentication procedure. The further use case of such an indicator would be to assist in defining security policies for low-power or resource-constraint machine type devices.

4.4.8.2 Components and interaction overview

To demonstrate this enabler on UE, we choose Android based smartphone Samsung Galaxy S3. The reason behind selecting this particular Samsung model is that it allows for access to baseband logs to fetch serving network information. Note that the Samsung device needs to be rooted in order to get baseband logs information. The main components are the baseband logs stored on the device and an Android App. We use the Darshak framework [4] to implement the 'security indicator' enabler as an application. The figure 5.2 depicts the main logic of Darshak framework utilised in this enabler application.

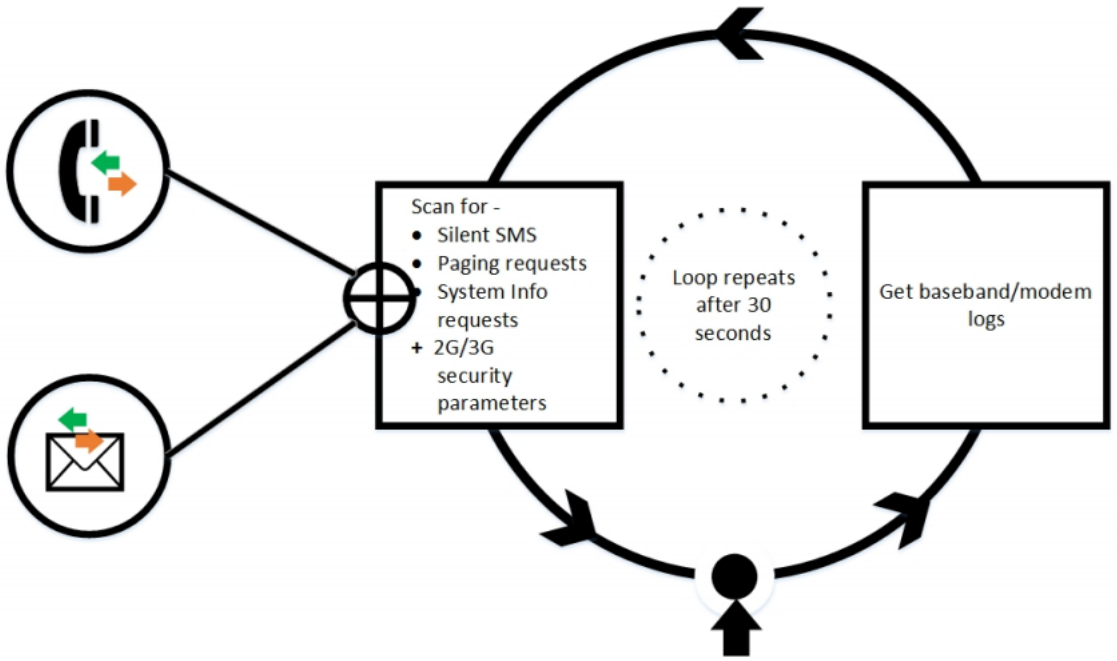


Figure 5.2: Primary logic of enabler application derived from [4]

4.4.9 Architectural drivers

4.4.9.1 High-Level functional requirements

The high-level functional requirements are:

- The enabler shall not disrupt network operations of UE.
- The enabler shall provide notifications to the subscriber.
- The enabler shall use least amount battery possible.

4.4.9.2 Quality attributes

None.

4.4.9.3 Technical constraints

Since there is no default and standardized API provided by smartphone OS providers, this enabler is installed on a Samsung Galaxy S3 model. Further, the device needs to be rooted in order to display notifications about the security capabilities of serving network.

4.4.9.4 Business constraints

None.

4.4.9.5 Link to Architecture

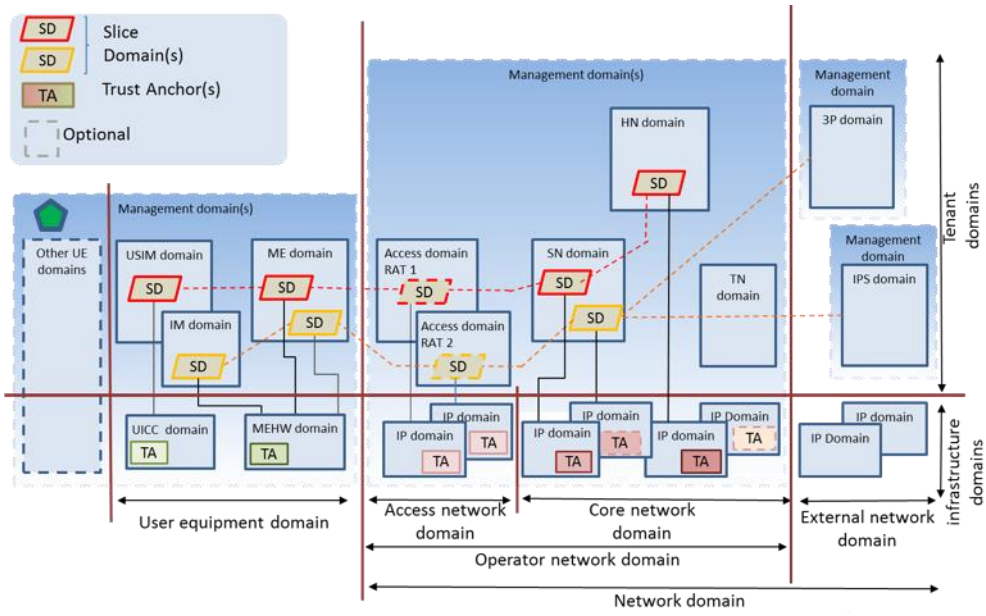
This section outlines objectives fulfilled by the enabler and mapping to the security architecture.

4.4.9.5.1 Security objectives fulfilled by the enabler

This enabler supports O2.1 to O2.4 security relation to legacy system objectives and O3.3 RAN security objective as defined in D2.4

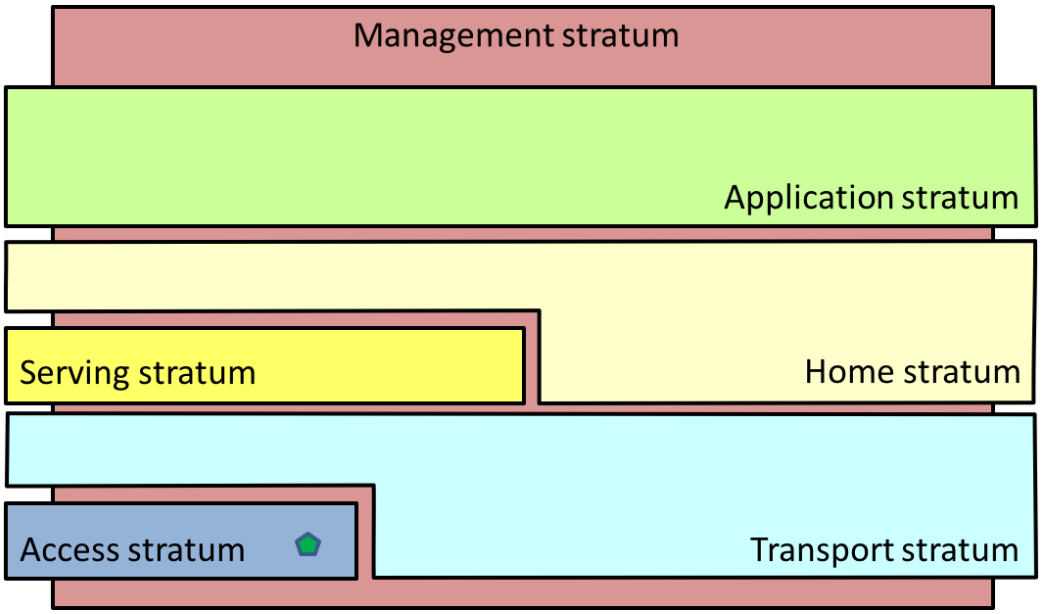
4.4.9.5.2 Enabler mapping to architecture domains

This enabler is implemented in UE domain. In particular, it will be installed on the UE as an application.



4.4.9.5.3 Strata location of the enabler

The enabler is linked to access stratum.



4.4.10 Detailed specifications

4.4.10.1 Introduction

This enabler aims to enhance trustworthiness of serving network. It provides new security indictors to be displayed on the UE. In particular, we introduce following features:

- Authentication state of serving network on particular network event (call/text)
- Ciphering details such as type of encryption algorithm
- Temporary identity (for example TMSI in GSM)

4.4.10.2 Conformance

For an implementation to be conformant it should implement specifications as stated herein.

4.4.10.3 Enabler Component Specifications

We implement this enabler as an Android app to be run on a rooted Samsung Galaxy S3 device. The specifications for this enabler component are as follows:

- The UE should notify to the user if authentication of serving network is successful or not.
- The UE should monitor the usage of temporary identity such as TMSI provided by the serving network.
- The UE should inform the user of the type of ciphering algorithm used by serving network for ongoing or recent communication events such as call or texts. In case of weaker algorithms, security alert should be notified to the subscriber.

4.4.10.4 API specifications

There is no specific API for the enabler as it operates on a real device and thus provides no external API interfaces.

4.4.11 Re-utilised Technologies/Specifications

We aim to build upon existing Darshak framework [4] implementations to provide for the enabler service.

4.4.12 References

[1] 3GPP TS 22.101, "Technical Specification Group Services and System Aspects; Service aspects; Service principles"

[2] Taddong's Security Blog, "Does my phone warn you when it is not encrypting your calls". Link - http://blog.taddong.com/2011/02/does-your-phone-warn-you-when-it-is-not.html Last visited Feb 2017.

[3] Android Ciphering Indicator. Link - https://github.com/PrivacyCollective/Android-CipheringIndicator-API Last visited Feb 2017

[4] Darshak Framework. Link- https://github.com/darshakframework/darshak Last visited Feb 2017

4.4.13 Acknowledgements

We would like to thank Darshak Framework project [4].

4.5 Reputation based on Root Cause Analysis for SDN Open specifications

4.5.1 Preface

In SDN/NFV based infrastructures, networking services will rely on a dynamic placement and migration of the virtual network functions as well as an elastic usage of the compute, storage and networking resources.

Indeed, the high network dynamicity in SDN–topological changes and rapid forwarding changes through flows–and NFV–VNF post-deployment operations such as scaling– and the centralization of the intelligence within the control plane makes the resiliency of SDN/NFV infrastructures a paramount feature.

In response to these challenges, we focus here on the root cause analysis (RCA) as a key operation to ensure the smooth functioning of networking services relying on SDN and NFV which is based on intelligent management decisions.

We propose a reputation propagation mechanism based on computational models that scores the network elements on a given infrastructure domain as a first step towards automation on the management decisions. This reputation calculation mechanism is able to store past experiences.

This reputation propagation approach is based on the output provided by the RCA algorithm. This reputation propagation mechanism presented hereafter is theoretical based on simulation results but no implementation is foreseen at the writing time of this document.

4.5.2 Status

At the time of this writing, the ***Reputation based on Root Cause Analysis for SDN*** is still under research and development. Therefore, the open specifications of this enabler are preliminary and are subject of improvement.

This enabler is not planned to be implemented or released as software for R2.

4.5.3 Copyright

Orange is currently the only contributor to this enabler.

4.5.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

4.5.5 Terms and definitions

SDN	Software-Defined Networking
NFV	Network Functions Virtualization
VNF	Virtualized Network Function
VL	Virtual Link
RCA	Root Cause Analysis

4.5.6 Overview

We propose a reputation propagation mechanism for multi-domain SDN infrastructures, where a service is delivered across different domains. In service chains delivered across two or more domains (not owned by the same operator), is necessary to establish a reputation calculation mechanism to determine which domain is responsible for a given service failure or degradation which may impact the end-users.

4.5.7 Basic concepts

Hereafter we detail the different blocks composing the reputation based RCA for SDN enabler.

In this section, we zoom in on the self-modelling based RCA used and its mission. Model-based self-modelling RCA relies on a probabilistic dependency graph which indicates how faults propagate through

the network and eventually lead to service failures. In model-based self-diagnosis approaches, a RCA module exploits this fault propagation graph to calculate the root cause.

We use an RCA module [2], [3], [4], and [5] that calculates the probability of failure for all the network elements in the infrastructure domain given any symptom of failure in the service. This RCA module is based on a self-modelling technique that generates a probabilistic graphical model containing the dependencies among the network resources in a single network domain manage by a SDN controller. This block is also in charge of updating it with topological changes.

On the other hand, the RCA exploits this generated dependency graph to compute the root cause; it means to compute the probability of each network element in the domain to be responsible for the service failure. The RCA block is based on an inference Bayesian Networks algorithm. A high-level view on the self-modelling based RCA module can be seen in the next figure, whose main steps are three:

Step 1: Transformation of the network topology into a machine-readable format containing the classified network elements in each domain.

Step 2: On-the-fly construction and continuous update of the fault propagation model from the machine-readable format and running applications. This model contains the network nodes, their internal logical and physical components such as ports or running applications to ensure a fine-granular diagnosis.

Step 3: Root cause analysis (RCA) to calculate the probability of faulty networked elements with component-level granularity by exploiting this generated fault propagation model.

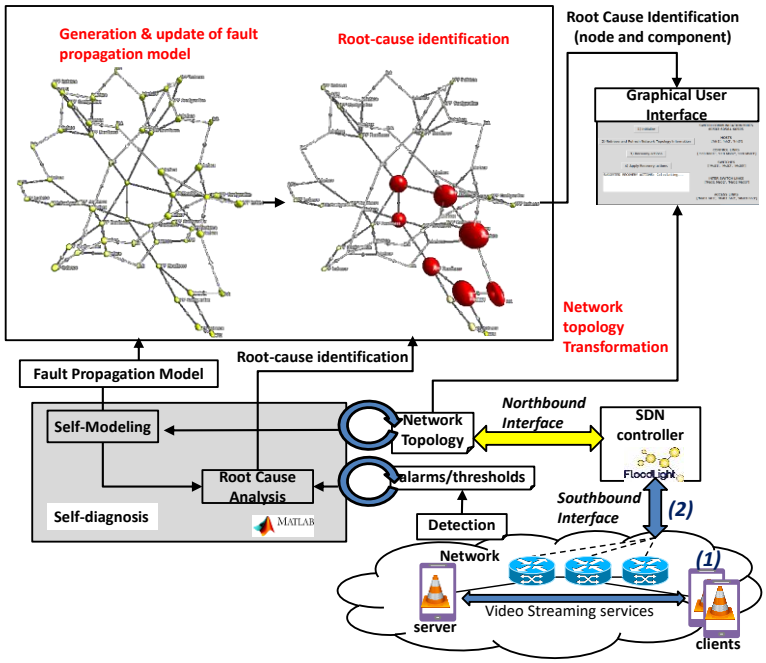


Figure 4-18. Overview of RCA for one single SDN domain, from [2]

4.5.8 Main interactions

4.5.8.1 Use cases

We propose as a use case a video streaming application between two different SDN domains, as shown in the following figure. As said before, two different SDN controllers in two different domains are involved in this service delivery.

In the event of any service delivery degradation, the enabler estimates the domain responsible for the degradation, in other words where the root cause originated. In this specific use case, the reputation calculation block continuously estimates the propagation of its domain and exposes those values to a proper entity which going to use them to settle responsibilities.

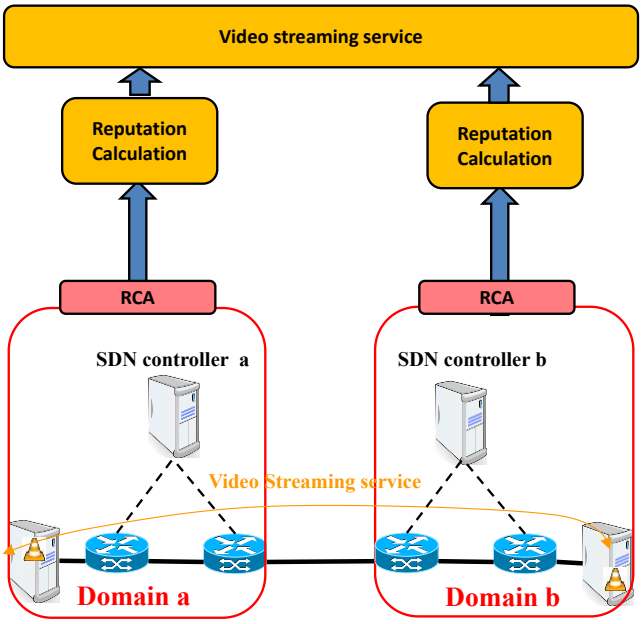


Figure 4-19. Video streaming service delivery between two domains

4.5.9 Architectural drivers

4.5.9.1 High-Level functional requirements

This enabler depends on a self-modelling RCA module that in turn depends on the network topology format given by the SDN controller. Indeed, this network topology is collected through the northbound REST API exposed by the controller in a JSON format.

4.5.9.2 Link to Security Architecture

This section is about how the proposed enabler is related to the proposed security architecture defined inside the project.

4.5.9.2.1 Security objectives fulfilled by the enabler

The enabler contributes in accomplishing the objectives O8.1 inside the new business and use cases (cf. D2.4). Indeed, the goal of this enabler is to ensure availability of the underlying infrastructure but also of the virtual resources running on top of it.

4.5.9.2.2 Strata location of the enabler

This enabler is conceived to be at the management plane, only excluding the radio access network (unless there is SDN in this network domain).

This enabler needs a direct interface (the northbound interface of the SDN controller). The enabler may be connected to other elements that manage the virtualized infrastructure such as the VIM (Virtualized Infrastructure Manager) e.g. OpenStack.

4.5.9.2.3 Enabler mapping to architecture domains

The core network domain is the optimal location for the enabler. This enabler can be located at the operator network domain, the external network domain or both. In particular, the enabler will need a transversal view, so tenant and infrastructure domains may be needed to expose their information to this enabler. The following table shows the different domains where the enabler may belong to.

Architecture Domain
Operator Network Domain
Core Network Domain
Access Network Domain
External Network Domain

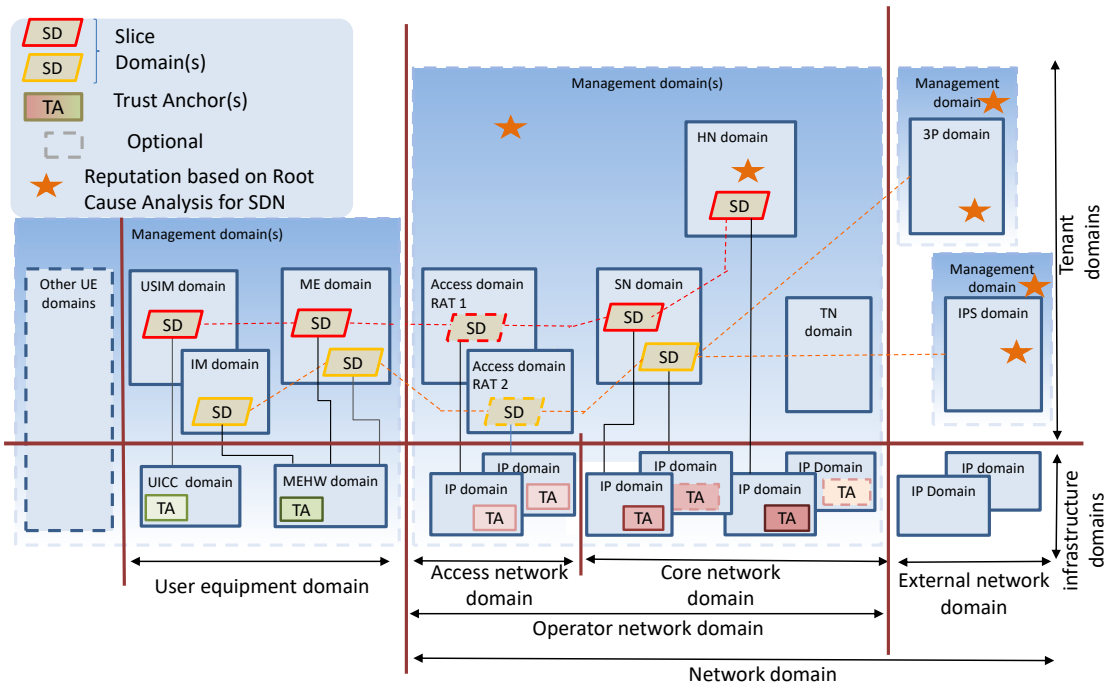


Figure 4-20. Enabler potential implementation in the 5G Security Architecture

4.5.9.3 Quality attributes

As said before, we do not plan any software realization at the writing time of the document.

Two major quality attributes are considered in this enabler: the compatibility of the enabler and its performance.

The compatibility is given by the capability of the enabler to interoperate with other SDN controllers through their exposed northbound API (REST API), as all available SDN open source controllers (Floodlight, OpenDaylight, ONOS) exposed different interfaces and different network topology formats.

The performance aspects are given by the influence of the diagnosis uncertainty and its accuracy on the reputation values assigned to the domains. Also, the time to compute the RCA and the reputation values for all the network resources in a given domain is very important, and more where the failure rate of a given infrastructure domain is rather high, which implies to update the reputation values very often.

4.5.9.4 Technical constraints

No known technical constraint exists.

4.5.9.5 Business constraints

No known business constraint exists.

4.5.10 Detailed specifications: RCA based Reputation mechanism

The reputation mechanism is based on the aforementioned self-modelling based RCA mechanism.

As it can be seen, a RCA is plugged to each domain. Each domain is composed of a single SDN controller and its underlying infrastructure. Its role is to establish the interconnection among the different hosts in that domain by installing flows on the OpenFlow switches. The infrastructure is then composed of the controller, the intermediate switches, connected to this controller and the different hosts embedding applications such as video streaming applications.

The results of the RCA will provide the probability of failure for each network element in the domain ***d*** at a given t_i to a reputation calculation block to score this RCA output. The role of the reputation calculation block is to provide with a global reputation metric of a given infrastructure domain and inform higher management entities in the event of any fault or failure. A high-level view on the reputation mechanism for three network domains can be seen here:

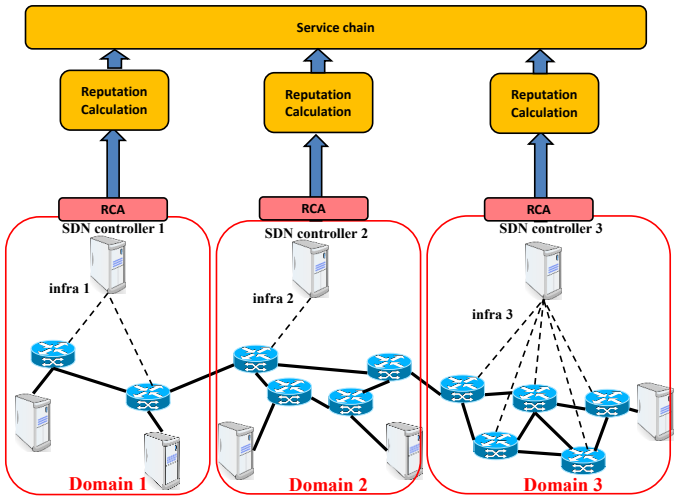


Figure 4-21. Reputation propagation mechanism

The reputation mechanism is based on three steps, which are to be described in the following subsections.

4.5.10.1 Phase 1: Creating a historical record database containing the RCA output

The first phase is to store the RCA output and the context of the failure in a historic record database. This step is necessary to identify and trace repeatable patterns of failures and faults, or trace unnoticed resource degradations that do not impact the service layer. In addition, this database can include topological changes that may explain other types of service outages due to misconfiguration. Thanks to this

database, the RCA can diagnose faster similar issues based on this historical record and it can also diagnose with more accuracy by including information on the topological transitions previous to service outages.

The joint probability distribution P(t_i,**d**) of a given domain **d** at time t_i is composed of the probabilities of failure of each network element N_k in that domain, denoted as ***prob(t_i, N_k)***. The RCA output P(t_i,**d**) is computed over on a given network topology <*topology_t_i*> in a domain **d** triggered by an alarm <*alarm_t_i*> at t_i.

The role of this block is to store the computed P(t_i,**d**) in a historical record database. The following figure shows how two different RCA outputs computed at times t_0 and t_i are stored in the historical record database. The data stored at each instant of time: timestamp <*t_i*>, <*alarm_t_i*>, <*topology_t_i*>, and P(t_i,**d**).

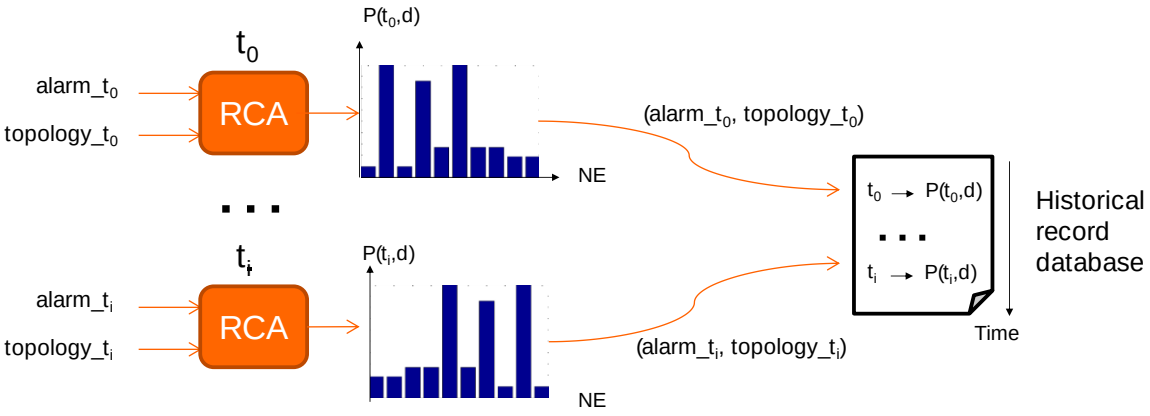


Figure 4-22. RCA output storage in historical record database

4.5.10.2 Phase 2: Scoring the RCA output and its evolution over time

This phase is to detail how the RCA output provided by the self-modelling based diagnosis module is scored. This is phase is done in two steps: firstly, the RCA output is assigned a reputation value, secondly the RCA time evolution is assigned a reputation value. In this way, the RCA output is scored by considering how it evolves over time to identify persistent vulnerabilities that may make reputation go lower due to non-suitable or fruitless recovery strategies injected in the network.

4.5.10.2.1 Scoring the RCA output

In this step, the reputation calculation block scores the RCA output P(t_i,**d**) at each instant of time t_i, without considering how the RCA output evolves over time. This calculation does not take into account past information to previous instants to t_i recorded in the historical record database, which is explained in the next subsection.

The reputation calculation block scores the RCA output P(t_i,**d**) by scoring the probability of failure ***prob(t_i, N_k)*** of its composing network elements. These probability values are transformed in reputation scores according to the figure below.

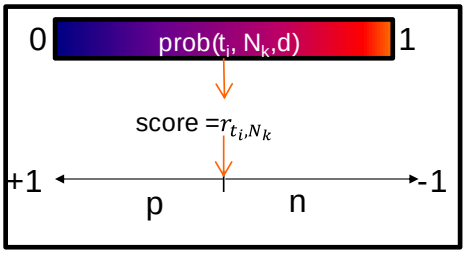


Figure 4-23. Reputation score r_{t_i,N_k} of network element N_k computed at instant t_i

The idea behind is that a high probability of failure values become negative reputation values (denoted as ***n***) and low probability of failure values become positive reputation values (denoted as ***p***). Reputation values are therefore given in the interval [-1,1], whilst the probability of failure values is evidently given in the interval [0,1]. Ideally, a probability of failure value of 0.5 is rather neutral so the reputation value in this case is 0, neither negative nor positive.

The reputation score for a network resource N_k computed at instant t_i, is denoted as r_{t_i,N_k}. The following transformation interval equation assigns reputation values r_{t_i,N_k} to the probability values given of the different network elements by the values ***prob(t_i, N_k)***:

$$r_{t_i,N_k} = 1 - 2\text{prob}(t_i, N_k)$$

For instance, the reputation of a SDN controller (***N_c***) can be computed from its probability of failure ***prob(t_1,N_c)*** for two extreme cases:

- $\text{prob}(t_0, N_c) = 0.85 \rightarrow r_{t_0,N_c} = 1 - 2\text{prob}(t_0, N_c) = -0.7$
- $\text{prob}(t_1, N_c) = 0.05 \rightarrow r_{t_1,N_c} = 1 - 2\text{prob}(t_1, N_c) = +0.9$

4.5.10.2.2 Scoring the evolution of the RCA output

In this step, the reputation calculation block scores the evolution of the RCA output value P(t_i,**d**). This step is necessary to quantify behavioral changes on the network elements. For instance, the recovery actions injected in the network resources may make things get worse instead of recovering the network.

However, one of the main limitations on the state of the art on reputation mechanisms [6] is how to represent in a compact manner the high amount of historical information extracted per network element.

, We propose to store the reputation difference with respect with the previous reputation value. We propose to compute the reputation variation term $\Delta r_{[t_i,t_{i+1}]}$ as the difference of the reputation between the current instant t_{i+1} and the previous one t_i, omitting its variation w.r.t to previous instants of time. The reputation variation term is then computed as follows: $\Delta r_{[t_i,t_{i+1}]} = r_{t_{i+1},N_k} - r_{t_i,N_k}$. This value can be positive or negative to represent positive transitions or negative transitions.

Finally, the reputation score for a given network element N_k is based on two terms, similarly to reputation schemes seen in [6]. A first term to update the reputation score based on the current reputation value r_{t_{i+1},N_k} but also a second term to take into account the historical reputation record $\Delta r_{[t_i,t_{i+1}]}$ as shown below:

$$R_{t_{i+1},N_k} = \rho_{Historical}\Delta r_{[t_i,t_{i+1}]} + \rho_{Update} r_{t_{i+1},N_k}$$

where $\rho_{Historical}$ and ρ_{Update} are weights to control the balance between the current reputation value at instant t_{i+1} and its difference with respect to the reputation value at previous instant t_i.

4.5.10.2.3 Assigning a global reputation value to an SDN domain infrastructure

The goal of the reputation mechanism is to provide with a global reputation estimation that gives a first insight on the reliability and trustworthiness of a given SDN domain a whole.

Given the reputation values R_{t_{i+1},N_k} for each network element k at a given instant of time, an average is computed among all reputation values on that domain as a high-level indicator. The global reputation metric for a given network domain ***d*** is given by this equation:

$$R_{t_{i+1}}(d) = \frac{1}{N(d)} \sum_{k=1}^{N(d)} R_{t_{i+1},N_k}$$

However, other ways to compute the global reputation values can be taken into account to avoid averaging very distant reputation values from the average.

4.5.10.3 Phase 3: Making decisions based on reputation

4.5.10.3.1 Uses of the reputation score

The main use of reputation score is for an automated decision making. As identified by Yu et. al. in [6], there are three types of decision making methods:

- Threshold-based: this method filters out the reported information from untrustworthy nodes.
- Ranking-based: this method ranks nodes according to their trustworthiness values.
- Weight-based: this method weighs the decisions according to all nodes but considering their reputation values.

Concretely, in SDN/NFV, the reputation score may be used to:

- to identify the most critical network elements to provide with additional redundancy
- to avoid the interaction with those less trustworthy nodes as long as are not recovered (survivability systems)
- to enable democratic decision making voting schemes where the reputation is an influence weight
- to enable the SDN controller to forward based on higher reputation nodes, avoiding the rest

In our understanding, this is the very first time in using a reputation propagation mechanism to assess the reputation of SDN/NFV infrastructures.

4.5.11 References

[1] Topology-Aware Self-Diagnosis framework, presented in Orange Labs Research exhibition 2015, Paris, France. Video available at: https://www.youtube.com/watch?v=xNudu48quRM

[2] J. Sanchez, I. Grida Ben Yahia, N. Crespi, "THESARD: on The road to resiliencE in SoftwAre-defined network-ing thRough self-Diagnosis" 2nd IEEE Conference on Network Softwarization, Seoul, Korea, 6-10 June 2016.

[3] J. Sanchez, I. Grida Ben Yahia, N. Crespi, "Self-Modeling Based Diagnosis of Software-Defined Networks," Workshop MISSION 2015 at 1st IEEE Conference on Network Softwarization, London, 13-17 April 2015.

[4] J. Sanchez, I. Grida Ben Yahia, et. al., "Softwarized 5G networks resiliency with self-healing," 1st Internation-al Conference on 5G for Ubiquitous Connectivity (5GU), 2014.

[5] J. Sanchez, I. Grida Ben Yahia, N. Crespi, "Self-Modeling based Diagnosis of Services over Programmable Networks," 2nd IEEE Conference on Network Softwarization, Seoul, Korea, 6-10 June 2016.

[6] Yu, H., Shen, Z., Miao, C., Leung, C. and Niyato, D., 2010. A survey of trust and reputation management systems in wireless communications. Proceedings of the IEEE, 98(10), pp.1755-1772.

5 Security Monitoring Enablers open specifications

This section contains the open specifications of the five Security Monitoring enablers due to Release-2: The Pulsar (Proactive Security Assessment and Remediation) enabler, the Generic Collector Interface enabler, the security monitor for 5G Micro-Segments, the Satellite Network Monitoring enabler and the System Security State Repository enabler.

Note: The Security Monitoring enablers of 5G-ENSURE Release-2 are enhancements of Security Monitoring enablers of 5G-ENSURE Release-1.

5.1 PulSAR Open specifications

PulSAR stands for Proactive Security Assessment and Remediation.

5.1.1 Status

A prototype of the enabler is currently developed; the specification of the enabler is an update of previous ones to cover the new features in focus of R2.

5.1.2 Copyright

5.1.3 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

5.1.4 Terms and definitions

CVE: Common Vulnerabilities and Exposures
CVS: Common Vulnerability Scoring
CVSS: Common Vulnerability Scoring System (https://nvd.nist.gov/cvss.cfm)
Datalog: a declarative logic programming language that syntactically is a subset of Prolog.
IoT: Internet of Things
MulVAL: Multi-host, Multi-stage Vulnerability Analysis Language. It is a research project at Kansas State University and a logic-based enterprise network security analyser.
NFV: Network Function Virtualization
NIST: National Institute of Standards and Technology (http://www.nist.gov)
NVD: National Vulnerability Database
PulSAR: Proactive Security Analysis and Remediation
SDN: Software Defined Network
SIEM: Security Information and Event Management
TVA: Topological Vulnerability Analysis, a framework that combines an exploit knowledge base with a remote network vulnerability scanner to analyse exploit sequences leading to attack goals
XML: eXtended Markup Language

5.1.5 Overview

Cyber Security is the first step towards understanding the actual risk exposure of a communication system and, hence, towards exploiting it with desired security behavior and detection of potential attacks or non-authorized usages. The Proactive Security Analysis and Remediation (PulSAR) enabler deals with detection of Cyber Security risks on sensitive “assets” at large, up to proposing possible remediation. It intends to

address the detection and management of cyber-security risks on different kind of environments from classical IT systems to complex 5G environments involving NFV and SDN.

PulSAR enables complex attack detection, provides a clear view on an attack by giving means to understand attack paths when a node is known as vulnerable and automatically computes possible remediation depending on the system's assets, and the system's vulnerabilities. The possible means of remediation are sorted by a cost function based on the cost of deployment of the different remediation means such as network reconfiguration, patch deployment, etc.

This enabler aims at providing means to protect against cyber-attacks. Its main features can be summarized as follows:

1. Collect the topology and the vulnerabilities, the costs of each asset, evaluate the potential threats, compute attack paths and assess risks,
2. Propose remediation solutions,
3. Deliver a visualization service centered on risk/attrition level.

Before going into the details of the enabler, the figure below shows its eco-system with possible interactions with a 5G eco-system and other enablers of this project. This eco-system includes traditional network and computer infrastructures, virtualized networks and data-centers, satellite networks, and potentially 5G-specific entities such network slices and micro-segments, Internet-Of-Things environments, etc.

For all those environments, PulSAR needs to collect several kinds of data, both to initialize the cyber risk computation and to follow it at run-time.

- Topological data: network topology, links, machines, interfaces
- Vulnerability data
- User scoring preferences, i.e. user defined scores on attack graph nodes

PulSAR then computes:

- A global attack graph showing all possible attacker progression in the network
- attack paths scored by attrition levels for a better readability by the security operator,
- possible remediation(s) for each attack path.

The results of these computations are made available through a single REST/JSON interface.

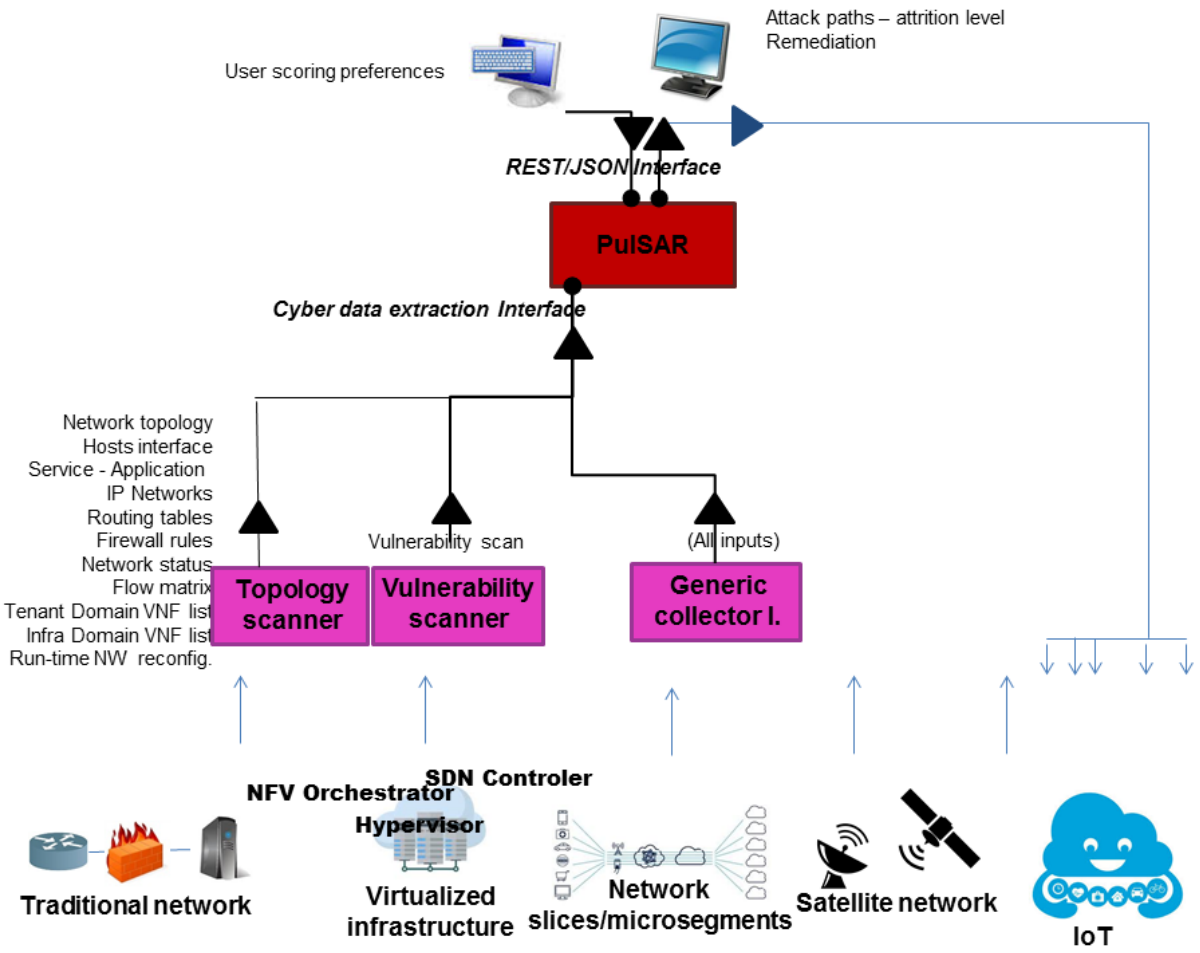


Figure 5-1- PulSAR in its eco-system

5.1.6 Basic concepts

The **Error! Reference source not found.**below provides an updated architectural sketch of the components inside PulSAR in R2 and their main interactions. This section gives a short insight for each component. The figure below is also explained in the following Architecture section.

For its basic operation, PulSAR needs mainly two types of Data as inputs, namely the network topology and vulnerabilities that apply (e.g. resulting from a vulnerability scan). These two inputs are fed into the **Cyber Data Extraction module**, which generates a single input file (known as the reference XML input file).

This input file is sent to the proactive part of the PulSAR server, and more precisely to one of its components, which is an **Attack Graph Engine**. In order to make the results of the attack graph computation more readable, all the attack paths computed are scored by attrition level. This scoring is performed thanks to the CVS scoring and manually imported user scoring preferences corresponding to his/her knowledge of the most challenging risks according to context at hands. It is done by the **Scored attack paths module**. The **remediation module** computes associated remediation such as network reconfiguration, or virtual patching according to type of remediation supported.

All the results of the PulSAR server are made available through a REST/JSON interface, that is to say the attack paths scored by attrition level, the remediation, but also the active attack paths and the countermeasures. A **visualization module** exploits all these results and makes them accessible to the Security operator. This module is also to security operator scoring preferences.

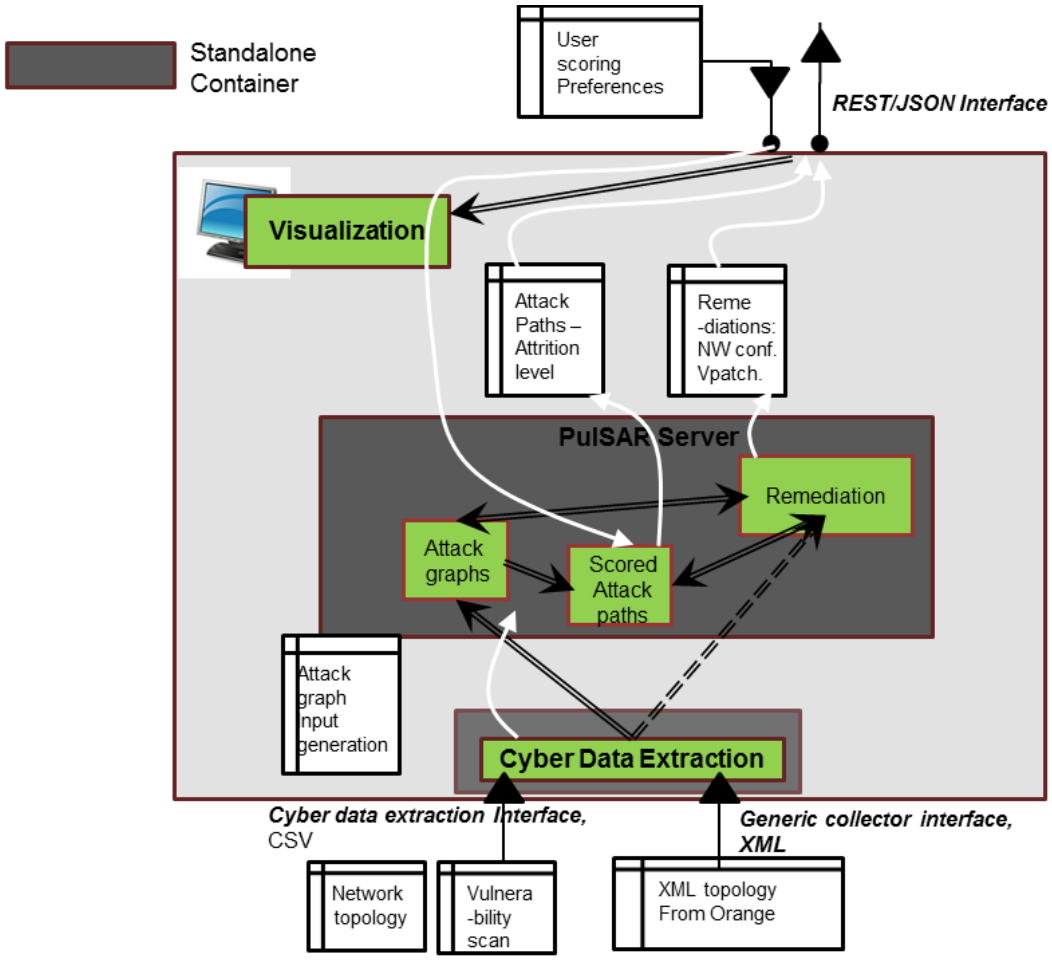


Figure 5-2: Building blocks of PulSAR

5.1.7 Main interactions

5.1.7.1 Use cases

Visualization and User scoring preferences

The use case below in **Figure 5-3** shows a Security Operator using PulSAR through the visualization interface. S/he is also able to set her/his own preferences for scoring the impact of the vulnerability on the nodes of the graph.

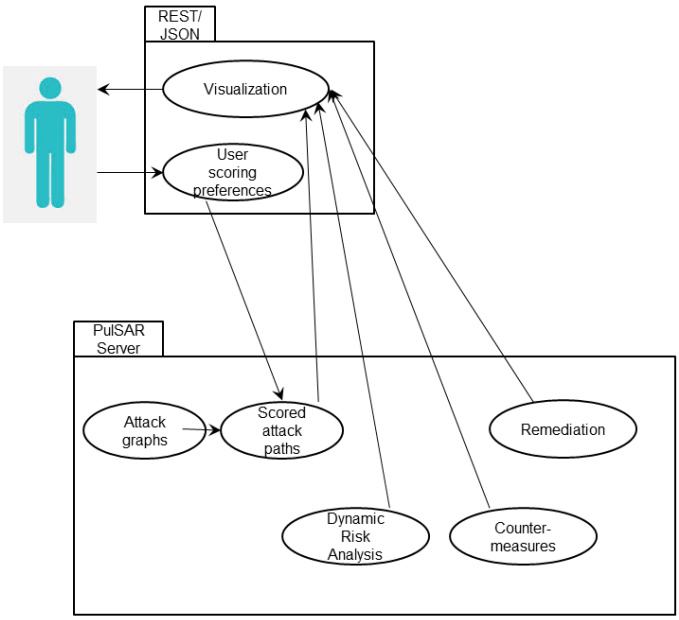


Figure 5-3: Visualization and user scoring preferences use case

5.1.7.2 Components and Interactions overview

Error! Reference source not found. above gives an overview of the internal architecture of the PulSAR enabler. As presented in the Basic concepts section, PulSAR includes the following components, which are described in detail below:

- Cyber data extraction
- Attack graphs
- Scored attack paths
- Remediation
- Visualization

Cyber Data Extraction

The Cyber Data Extraction component provides an external interface called ***Cyber Data Extraction Interface*** for Cyber data collection. This interface is specified in the API specification section. It collects network topology information and vulnerability scan reports.

In order to give access to the vulnerability and topological information to the other components internally, the Cyber Data Extraction can generate two types of inputs:

- The Attack graph input generation file which is a Datalog file used by the Attack graph engine,
- A XML file aggregating all vulnerabilities and topological information needed by the Remediation and the Countermeasure components.

Attack graph Engine

The attack graph engine relies on a MulVAL Attack graph Engine [MulVAL].

This component takes as input the Attack graph input generation file generated by the Cyber Data Extractor. The first action of this component is to generate the attack graph using MulVAL.

The attack graph output will be the basis used by the scored attack paths component. This attack graph gives the probability scores of each attack graph node, translating the probability of success of each step in the attack graph.

The Cyber Data Extraction takes as input different types of files, from several data sources:

- The original input is a collection of CSV files, containing the network topology, and a Nessus XML scan file.
- Support for other formats was added for vulnerability scans: OpenVAS and a custom JSON file, for testing purposes.
- To automate the interface with Generic Collector Interface (GCI) enabler in R2, cyber-data-extract supports an XML input containing the topology in the GCI format.

Scored Attack Paths

The scored attack paths take as input the attack graph computed by the Attack graph Engine.

This Attack graph gives the basic information for the scored attack path to be computed, that is to say both the potential attack paths and the probability scores of each attack graph node, translating the probability of success to each step in the attack graph. These probability scores are the basis for computing the risk score of an attack path.
PulSAR offers here the possibility for the Security operator to enter its own scores for each node of an attack, based on its own vision of the attack probability or of the asset criticality for the business. This input is performed through the external REST/JSON interface or by mean of the visualization component.

Therefore, the scored attack paths component uses two types of inputs:

- The attack graph from the attack graph component,
- The user scoring preferences from the external REST/JASON Interface via the visualization component.

The scored attack paths are the output available through PulSAR external ***REST/JASON Interface*** and used internally by the other components of PulSAR:

- The remediation component uses it as the basis to remediate the most critical attack paths,
- The visualization component exploits it as the basis of the visualization.

Mainly, the internal and external interfaces will operate via file input/output with a defined XML data structure for the file content.

Remediation

The remediation application needs two types of inputs:

- The scored attack paths from the Scored Attack Paths component,
- The XML file aggregating all vulnerabilities and topological information generated by the Cyber Data Extraction component.

The interactions between the Scored Attack Paths and the remediation are necessary because attack paths are the starting point of the remediation process. Security operators need to select an attack path for remediation in the list provided by the Scored Attack Paths. On the other hand, the attack path engine is

also useful to validate the remediation selected by the security operators. This feedback is necessary to compare the security state of the system before and after deploying a remediation.

The interactions between the remediation tool and the Cyber Data Extraction are necessary because the network topology (hosts, routes, deployed firewall rules …) is necessary to compute the topological related remediation (attack signature deployment and firewall rules). The remediation tool provides also a means to apply automatically some of the remediation chosen by the security operators. To do that, this tool needs to change some parameters (e.g. a firewall rule) in the Cyber Data Extraction.

Besides the internal interactions between the Remediation engine and the Cyber Data Extraction and Scored Attack Paths, the Remediation engine especially provides output through the external ***REST/JSON interface*** which is then exploited by the Visualization component to display to the user the possible remediation for the risks on his/her information system.

Visualization

The Visualization offers a visualization service that allows users to visualize data. It exploits in fact the External ***REST/JSON interface*** which makes available data such as the Scored Attack Paths, the remediation, the active attack paths or the counter-measures.

The user accesses the visualization service through a standard web browser connected to the web application server using some network connection (such as the Internet). The user will experience a single integrated application. Behind the scenes, the browser will get access to the information generated by the other components.

5.1.8 Architectural drivers

5.1.8.1 High level functional requirements

PulSAR leverages upon WP2 –T2.4 architecture introducing the 5G concepts such as NFV and orchestrators, SDN and network controllers, slices, micro-segmentation. It enables to show the consequences of an attack on orchestrators and controllers. In the eventuality of monitoring of slices and micro-segmentations, PulSAR could implement new rules to follow such new attacks.

5.1.8.2 Link to Security Architecture

5.1.8.2.1 Link to which 5G Security Objectives

PulSAR enabler relates to the following security objectives.

O4.1 5G must enable a secure, reliable, and traceable sharing of network resources (i.e., compute, storage and network) between the various services having vastly different requirements such as reliability and low latency for tactile remote surgery and availability for massive IoT services.

O5.2 Slices must provide strong isolation of (virtual) resources allocated to different slices.

5.1.8.2.2 Mapping to 5G-Security Architecture

5.1.8.2.2.1 Mapping to domains

The schema below shows the mapping of the PulSAR enabler to the 5G-Ensure domains as defined in D2.4. PulSAR is represented as a green circle with dotted lines.

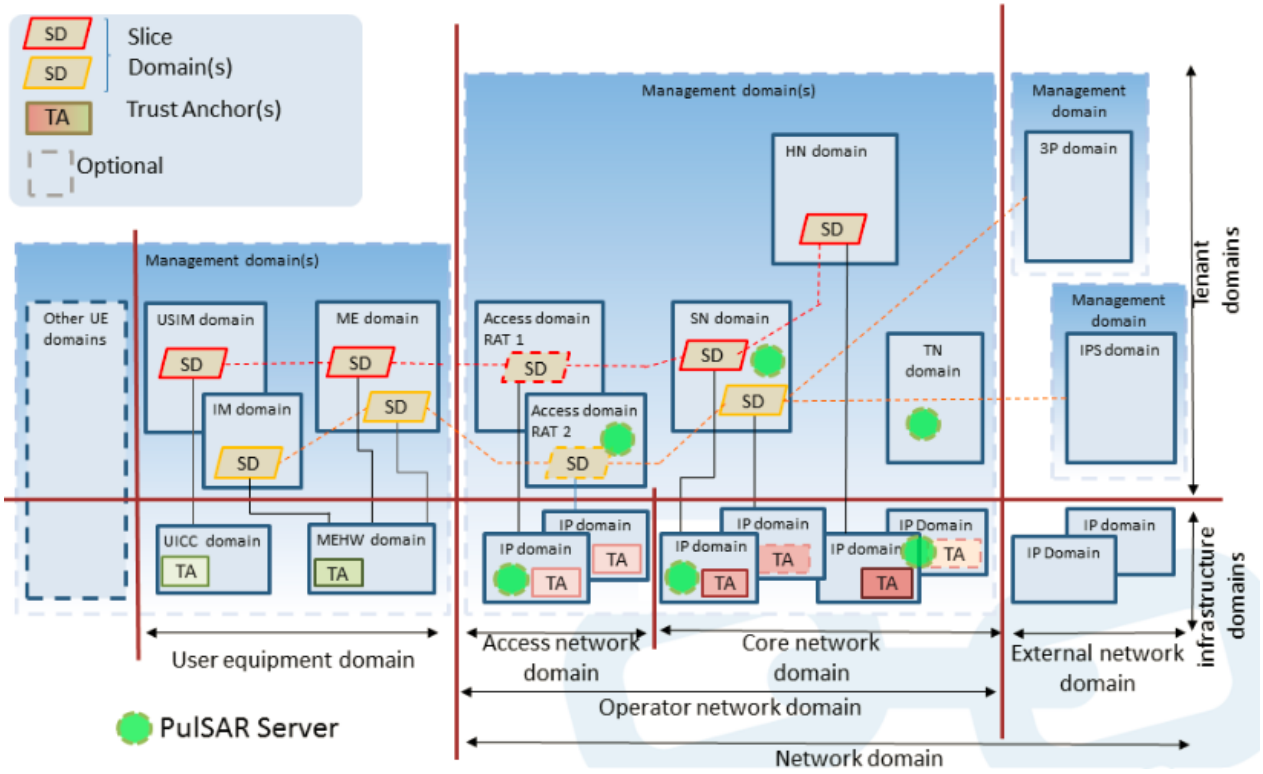


Figure 5-4: Security Monitoring mapping to 5G-Ensure Domains

Mainly, this security monitoring enabler is part of the Management domain in order to control the security of the entire network. This is particularly true for PulSAR which even targets to give orders back to the controlling instances (Orchestrator, SDN controllers, etc.). Nevertheless, PuLSAR could be installed at any level of the different network domains, which are the AN Domain and the CN Domain of the infrastructure, and the AN Domain and the SN Domain of the tenant part.

PulSAR server, looking after cyber-attacks, is valuable to be installed at all level of the network, from the infrastructure to the tenant domains.

5.1.8.2.2.2 Mapping to strata

The following schema maps the security monitoring enablers to 5G-Ensure strata. The same iconography is used.

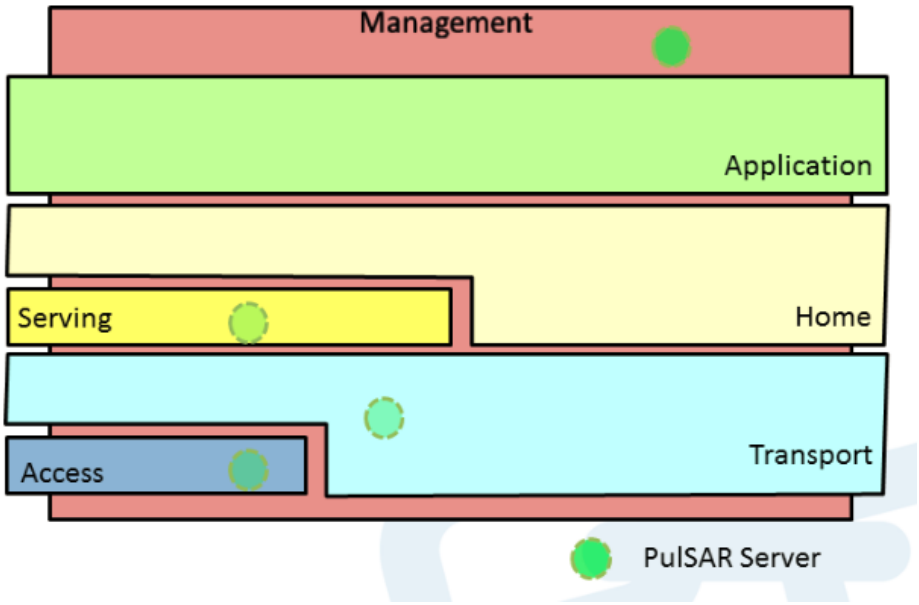


Figure 5-5: Security Monitoring mapping to 5G-Ensure Strata

The security monitoring enablers target to give an overview of the security quality of service through the network. PulSAR can be deployed through most of the strata, including the Home/Serving stratum, the Transport stratum and even the Management stratum as explained below. Nevertheless, it mostly works as part of the Management stratum.

PulSAR, based on the computation of cyber-security attack-trees, could also be deployed locally at an infrastructure provider premises which would be a sub-part of one stratum, in order to prevent cyber-attacks on its network.

5.1.8.3 Quality Attributes

Operability/Ease of use

The design of the PulSAR aims at stressing on characteristics such as usability, packaged solutions and solutions that scale.

Compatibility/Interoperability

Therefore, the following transverse design principles have been applied for the PulSAR:

- Standard data formats for exchange between the different components.
- Use of asynchronous process
- Proper error handling: to prevent any wrong analysis due to incorrect use the tools or incorrect input data (non-accurate or incomplete topology information shall be rejected).
- Flexibility and modularity: use a script programming language (such as Python) to develop the Cyber Data Extractor, to make it easy the plug new connectors into the component, if a user of the enabler wants to use another input format, while keeping the same outputs, adapted to each component. For example, if the vulnerability scanner used in a company is OpenVAS, rather than Nessus, it is easily possible, in few lines of code, to develop a connector to OpenVAS, which can be used in addition to the one of Nessus, while keeping the same outputs for MulVAL Attack Graph Engine.

5.1.8.4 Technical Constraints

Reasoning rules specify semantics of different kinds of exploits, compromise propagation, and multi-hop network access.

Datalog is used as the modelling language for the elements in the analysis (bug specification, configuration description, reasoning rules, operating-system permission and privilege model, etc.). It leverages existing vulnerability databases and scanning tools easily by expressing their output in Datalog and feeding it to the Attack Graph reasoning engine. MulVAL Attack Graph Engine adopts this modelling language [1].

Compared with a graph data structure, Datalog provides a declarative specification for the reasoning logic, making it easier to review and augment the reasoning engine when necessary.

MulVAL Attack Graph Engine uses a dependency graph to represent the pre-and post conditions for exploits. Then a graph search algorithm can combine individual exploits and find attack paths involving multiple vulnerabilities. This algorithm is adopted in Topological Vulnerability Analysis (TVA), a framework that combines an exploit knowledge base with a remote network vulnerability scanner to analyse exploit sequences leading to attack goals.

5.1.8.5 Business Constraints

The reasoning engine consists of a collection of Datalog rules that captures the operating system behaviour and the interaction of various components in the network. Thus, integrating information from the bug-reporting community and off-the-shelf scanning tools in the reasoning model is straightforward.

The interaction rules characterize general attack methodologies (such as “Trojan Horse client program”), not specific vulnerabilities. Thus, the rules do not need to be changed frequently, even if new vulnerabilities are reported frequently.

The reasoning engine scales well with the size of the network. Once all the information is collected, the analysis can be performed in seconds for networks with thousands of machines.

A key requirement is the compliance to industrial standards.

5.1.9 Detailed specifications

5.1.9.1 Introduction

PulSAR components presented above are described in detail below.

5.1.9.1.1 Cyber Data Extraction

The Cyber Data Extraction collects several types of input data described hereafter:

1. **The topology of the Information System**: The topology is extracted from CSV files describing the hosts, the services and the IP networks, the available routes from one host to another, the firewall rules, and also information regarding VNF domains, controllers and potentially other needed 5G objects.
2. **The report of a vulnerability scanner (e.g. NESSUS)**: The Cyber Data Extraction component handles the vulnerability scanner reports generated by Nessus for example. Nessus is a vulnerability scanner designed to automate the testing and discovery of known security issues. Nessus uses a client-server technology. Servers can be placed at various strategic points on a network allowing tests to be conducted from various points of view. A central client or multiple distributed clients can control all the servers. Nessus has the ability to detect the flaws of the hosts on the network

remotely but it can also detect flaws and missing patches by scanning locally if it is given the credentials to log in and run commands on the host itself. Nessus has the ability to test SSL-enabled services such as https, smtps, imaps, and more. The report generated by the Nessus scanner helps to have an overview of the status of the network and its vulnerabilities, which is the first step to prevent vulnerabilities. It detects only known vulnerabilities for which a detection plugin has been provided by Nessus. Although most of the vulnerabilities that can be exploited by a medium-level attacker have a corresponding Nessus detection plugin, very high-level attackers may know zero-days (unpublished vulnerabilities) that are not addressed by any of those plugins.

3. **The full network topology from Orange's GCI**: Alternatively, the Cyber Data Extraction module can interface to the Orange CGI to get a full network topology. This topology should be exposed in an XML file whose format definition is left to Thales and Orange, available via an API. It should contain all information on a 5G network: machines, links, interfaces, services and vulnerabilities, controllers and orchestrators. This file should be updated by GCI whenever the network topology changes. Cyber Data Extract will pull this XML file on a regular basis (e.g. every 2min) and generate the corresponding PulSAR input file, and send it to the PulSAR API to update the attack graph.

   The following schema is used between GCI and PulSAR.

```
<?xml version="1.0" encoding="UTF-8" ?>

<report>
<NFVOrchestrator>
      <ID> xx </ID>
      <NFVManagersNumber> xx </NFVManagersNumber>
      <State> xx </State>
      <VNFManagersIDs> xx </VNFManagersIDs>
      <VNFManagersStates> xx </VNFManagersStates>
      <VNFID-VLID> xx </VNFID-VLID>
</NFVOrchestrator>

<VIM>
      <ID> xx </ID>
      <NFVOrchestratorID> xx </NFVOrchestratorID>
      <VNFaddress> xx </VNFaddress>
      <VNFStorageID> xx </VNFStorageID>
      <VNFStorageName> xx </VNFStorageName>
      <VNFTypeOfStorage> xx </VNFTypeOfStorage>
      <VNFSizeOfStorage> xx </VNFSizeOfStorage>
      <VNFOwnerId> xx </VNFOwnerId>
      <VNFZoneId> xx </VNFZoneId>
      <VNFHostId> xx </VNFHostId>
      <State> xx </State>
      <VNFmetadata> v </VNFmetadata>
      <VNFstartTime> xx </VNFstartTime>
      <VNFendTime> xx </VNFendTime>
      <VNFID-VLID> xx </VNFID-VLID>
</VIM>

<VNFManager>
      <ID> xx </ID>
      <State> xx </State>
      <NFVOrchestratorID> xx </NFVOrchestratorID>
      <VNFsnumber> xx </VNFsnumber>
      <VNFsIds> xx </VNFsIds>
```

```
      <VNFsStates> xx </VNFsStates>
</VNFManager>

<MME-VNF>
      <ID> xx </ID>
      <VNFManagerId> xx </VNFManagerId>
      <VIMId> xx </VIMId>
      <MME-VNFState> xx </MME-VNFState>
      <SwImageDescriptor> xx </SwImageDescriptor>
      <VirtualStorageDescriptor> xx </VirtualStorageDescriptor>
      <More> xx </More>
</MME-VNF>
</report>
```

The Cyber Data Extraction generates two types of output:

- The first one is the Datalog file needed by the MulVAL Attack Graph Engine to generate the attack graph. This file contains both vulnerability and topological information.

- The second one is an XML file aggregating all vulnerabilities and topological information needed by the Remediation application, in order to simulate remediation in the network topology.

5.1.9.1.2 Attack graph

The attack graph engine relies on a MulVAL attack Graph Engine.
The first action of this component is to generate the attack graph using MulVAL, thanks to the input file generated by the Cyber Data Extractor.

In this process, for example, the results of the vulnerability scanner are converted into Datalog clauses like the following ones:

> **vulExists**(webServer, 'CAN-2002-0392', httpd).
> Namely, it identifies a vulnerability with CVE id CAN-2002-0392 on machine webServer. The vulnerability involved the server program httpd. However, the effect of the vulnerability — how it can be exploited and what its consequences are — is not contained directly in the results of the vulnerability scanner. The National Vulnerability Database (NDV) developed by the National Institute of Standards and Technology (NIST), provides the information about a vulnerability's effect through CVSS Impact metrics. The relevant information is converted from CVSS into Datalog clauses such as:
>
> **vulProperty**('CAN-2002-0392', remoteExploit,privilegeEscalation).
> The MulVAL Attack Graph Engine models elements in Datalog. The model elements are recorded as Datalog facts. The MulVAL Attack Graph Engine requires all Datalog facts to be defined prior to performing any analysis. Missing or incorrect facts will result in a misleading analysis of the system being modeled.

The interactions among multiple network elements must be considered in order to determine which security impact software vulnerabilities have on a particular network. The model used in the vulnerability

analysis must be able to automatically integrate formal vulnerability specifications from heterogeneous vulnerability sources. The MulVAL Attack Graph Engine is an end-to-end framework and reasoning system that conducts multi-host, multi-stage vulnerability analysis on a network. The MulVAL Attack Graph Engine adopts Datalog (a query and rule language for deductive databases) as the modeling language for the elements in the analysis (bug specification, configuration description, reasoning rules, operating-system permission and privilege model, etc.). It has leveraged existing vulnerability database and scanning tools by expressing their output in Datalog to feed the Attack Path Engine. The rules can be modeled according the security expertise and skills. In other terms, the normal users do not need to modify these rules and just need to update the database with all published vulnerabilities from NIST (http://www.nist.gov/itl/csd/stvm/nvd.cfm). The inputs to the MulVAL Attack Graph Engine's analysis are:

- Advisories: What vulnerabilities have been reported and do they exist on my machines?
- Host configuration: What software and services are running on my hosts, and how are they configured?
- Network configuration: How are my network routers and firewalls configured?
- Virtualization configuration: Which VMs run on which physical hosts?
- Orchestration configuration: Which machines (physical or virtual) are in which controllers' domains?
- Interaction: What is the model of how all these components interact?
- Policy: What accesses do I want to allow?

The Cyber Security Data Extraction allows to generate automatically such inputs, from automatically generated data (such as the vulnerability scanner) or easy to understand data (description of hosts, with their importance on a scale Negligible/Minor/Medium/Severe/Catastrophic). This allows non-expert users such as SMEs to use this tool, without high technical knowledge. The current MulVAL Attack Graph Engine data model relies on the exploit range (local or remote) and the privilege escalation consequence data that are stored in NIST NVD (National Vulnerability Database). The **Figure 5-6** below shows an extract of a logical attack graph computed by the MulVAL Attack Graph Engine. This logical graph explains how an attacker on the Internet (attackerLocated(internet)) can succeed to exec arbitrary code on a webserver (execCode(webserver,apache)) using the vulnerability CAN-2002-0392.

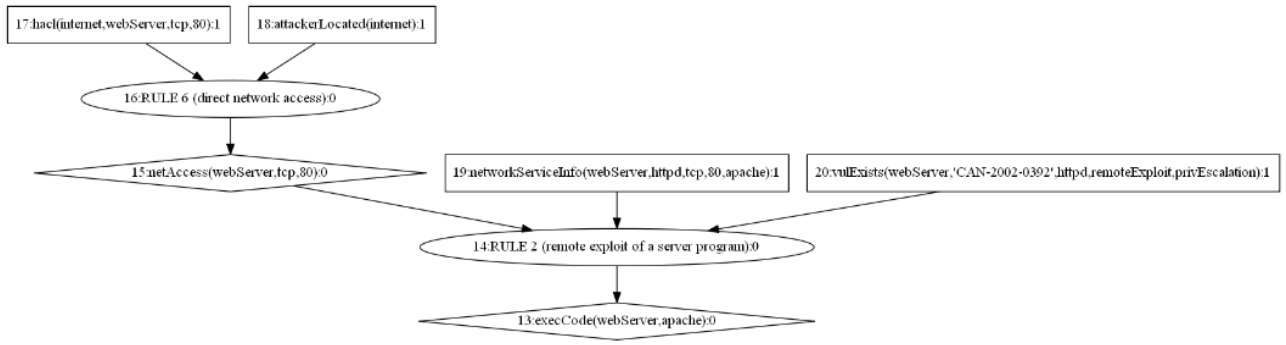


Figure 5-6: Example of logical attack graph

The MulVAL Attack Graph Engine uses Datalog (a subset of Prolog) to produce logical attack graphs. It takes as input a set of first-order logical configuration predicates and produces the corresponding attack graph. These configuration predicates include network-specific security policies, binding information and vulnerability data gathered from vulnerability databases. These are automatically generated from the NIST NVD, or from the topological configuration, thanks to the Cyber Data Extraction Component. The MulVAL Attack Graph Engine identifies possible policy violations through logical inference.

An attack graph presents a qualitative view of security discrepancies:

- It shows what attacks are possible, but does not tell you how bad the problem is.
- It captures the interactions among all attack possibilities in your system.

CVSS (Common Vulnerability Scoring System (https://nvd.nist.gov/cvss.cfm)) provides a quantitative property of individual vulnerabilities:

- It tells you how bad an individual vulnerability could be.
- But it does not tell you how bad it may be in your system.

The idea is to use CVSS to produce a component metric, i.e. a numeric measure on the conditional probability of success of an attack step. The MulVAL Attack Graph Engine aggregates the probabilities over the attack-graph structure to provide a cumulative metric, i.e. the probability of attacker success in your system. Suppose there is a "dedicated attacker" who will try all possible ways to attack your system. If one path fails, he/she will try another. The cumulative metric is the probability that he/she can succeed in at least one path.

MulVAL Attack Graph Engine rules for 5G

Specific 5G rules must be added to support 5G environment compared to previous traditional network and IT versions.

To explain shortly the extension of the Cyber-attack modelling schema, SDN and NFV bring new attack path types, due to three aspects:

- A centralized control plane
- A mutualized data plane
- 3-party interaction rule for VNF vulnerability exploitations: (NFV allows placing middle boxes between A and B, that can be targeted by the attacker)

We identified seven additional rules to cope with these threats:

- VM on host + vuln in hypervisor + vulConsequence == privEscalation => exec code in hypervisor (host compromised):
    - If a VM runs on a host, and a vulnerability exists on the Hypervisor which enables privilege escalation, then malicious code can be executed on the hypervisor which compromises the host.
- exec code on host + VM runs on host => exec code on VM
    - If code can be executed on a host and a VM runs on that host, then malicious code can be executed on the VM.
- exec code on host + VM runs on host => read VM FS
    - If code can be executed on a host and a VM runs on that host, then the File System of the VM can be read.
- exec code on orchestrator + VM in orchestrator domain => exec code on VM
    - If code can be executed on an orchestrator and a VM is in the orchestrator domain, then malicious code can be executed on the VM.
- exec code on host1 + VM on flow host1->host2 + Vuln on VM + vulConsequence == privEscalation => exec code on VM

- If code can be executed on host1, and a VM is on a flow between host1 and host2, and there is a vulnerability on the VM which enables privilege escalation, then malicious code can be executed on the VM.

A special focus is put on the description of the vulnerabilities of hypervisor, NFV orchestrators and SDN controllers and the impact of the compromising of those. For example, if a hypervisor is compromised, all the Virtual Machines (VMs) depending on it might be compromised at next step of an attack. Same type of reasoning is performed for NFV orchestrators and SDN controllers.

5.1.9.1.3 Scored Attack Paths

This Attack graph gives the basic information for the scored attack path to be computed:

- The potential attack paths;
- The probability scores of each attack graph node, translating the probability of success to each step in the attack graph. These probability scores are the basis for computing the risk score of an attack path.

Once the attack paths related to a given target is obtained, altogether with the score for each of the attack paths, the result will be output to the Remediation engine and Visualization component, in order to be employed for the objectives of the latter. Besides the internal interactions with the Attack Graph Engine and the Remediation engine, the Scored attack paths component provides one external interface to get the user preferences for attack path scoring. Scored Attack Paths represents the next step, following the metrics provided by the MulVAL Attack Graph Engine. Based on the Attack Graph provided by the MulVAL Attack Graph Engine, and the individual scores of each step, the objective is to yield the possible attack paths, along with a score associated to each one of the paths.

The considered attack paths that will be included in the list are selected based on the target node selected in the attack graph. The score of each path reflects the risk associated to the path as a whole, based on the individual scores of each step that have been previously calculated by the MulVAL Attack Graph Engine.

Additionally, to the risk score metric, the score of each path includes a second scoring component that takes into account the business impact of all IT resource(s) impacted by such attack path. Next section will elaborate on how the business impact is entered.

The main idea of scoring attack paths (see **Figure 5-7** below) is to consider paths independently from one another, as opposed to the approach of the MulVAL Attack Graph Engine, composed of individual scores, the latter being computed by taking into account all the connections existing in the attack graph.

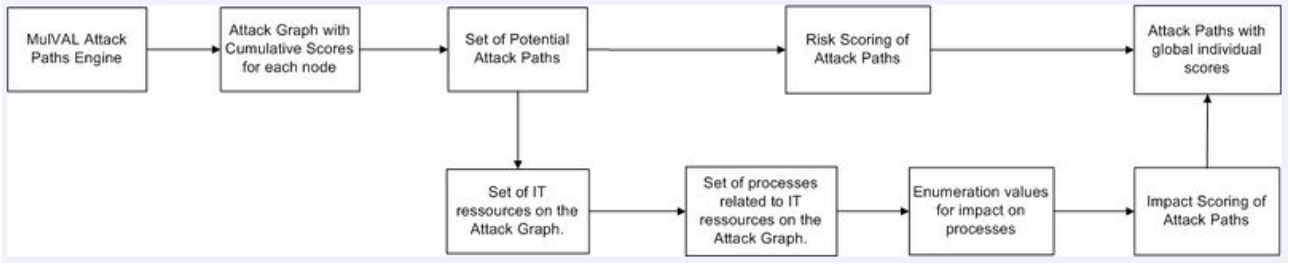


Figure 5-7: Flow diagram of scoring attack paths

5.1.9.1.4 Remediation

The Remediation application provides tools to security operators for proposing cost-sensitive remediation to attack paths. The attack paths are shown to a security operator, ordered by their scores, which allow easily understanding the severity of the consequences of the attack paths. To calculate remediation (see **Figure 5-8** below) to the chosen attack path, the tool first extracts the necessary information from the attack path to be corrected. Then, it computes several lists of remediation that could reduce / cut this attack path. Finally, it estimates the cost of each list of remediation and proposes all the lists of remediation, ordered by cost, to security operators. Operators can choose one remediation list and, thanks to the remediation validation, check whether or not the system is more secure after applying this remediation.

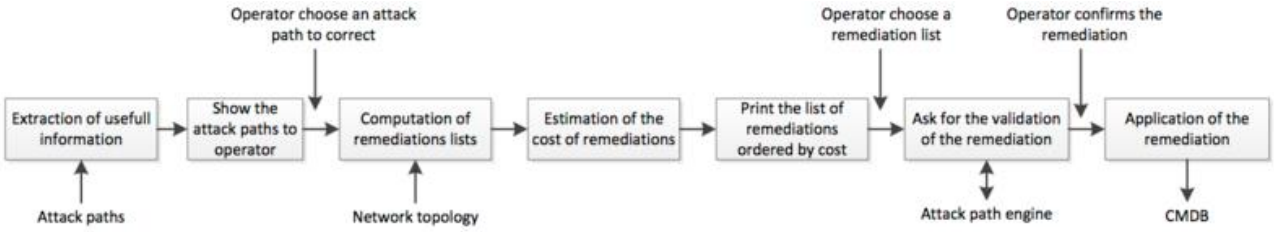


Figure 5-8: Remediation process

Remediation using a remediation database

To compute remediation, a remediation database is needed. This database makes a connection between vulnerabilities (for example thanks to a Common Vulnerabilities and Exposures identifier - CVE ID) and a possible adapted remediation. Several types of remediation could be used, for example a patch (it corrects a vulnerability) or Virtual Patching (a signature of known attacks which prevents the exploitation of a vulnerability) to quote few examples. To build the remediation database, information about patches can be extracted from publicly available sources of data (for example the National Vulnerability Database), or in Security Advisories (for example, coming from CERT-EU). Information about signatures and the related vulnerability could be extracted from the signatures database that contains the CVE ID. Python scripts are given together with the tool, in order to populate or update automatically the remediation database, from such open sources. The only action needed by the user, in order to do that, is to launch the script, which will download the files from the Internet, and populate the database.

Network remediation

The other types of remediation provided by the Remediation application cannot be stored in the remediation database, because these are network remediation, such as firewall or routing configuration change. The network configuration feature needs the simulation of the network topology obtained after modifying the network configuration, in order to confirm that it removes the vulnerability and reduces the risk to attacks. To sort the list of remediation, a cost function is applied to compute an estimate cost of each list. This cost contains two main components: operational costs and impact costs. The operational costs represent the costs caused by the deployment of the remediation (length of the deployment, maintenance costs, tests costs, …) whereas the impact costs represent the negative impact (side effects) that could happen following a remediation deployment.

Virtualization and orchestration remediation

The last remediation type, which is specific to 5G infrastructures, aims at cutting attack steps related to virtualization and orchestration. For instance, an attack step that crosses a hypervisor to compromise a VM can be cut by moving this VM to another host. Similarly, to the network remediation, it needs to be validated by a simulation of the obtained topology: we have to make sure that the new host cannot be attacked by another mean, which would lead to a compromise of the VM. Each possible placement is assessed and the ones that do not induce more attack steps are sorted according to their costs (operational and impact).

5.1.9.1.5 Visualization

The visualization is the main component of the Enabler to manage the computation of attack paths and remediation, and to display them. Systems that evaluate the security of a network, such as the Attack Graph or Scored Attack Paths Engines, can generate a large amount of data. The Visualization aims to find ways to display such big graphs, in order to present to an operator something that he/she can understand. This can for example be done thanks to the display of an attrition level associated to attack paths; therefore, the Risk Visualization is performed genuinely by attrition level.

The visualization interface also describes the different remediation that can correct attack paths, in order to help a security operator to make a choice.

5.1.9.2 Conformance

For an implement to be conformant it should implement specifications as here stated.

5.1.9.3 API specifications

The APIs are pictures on the first and second figure of this enabler section. The main external Interfaces of the PulSAR enabler are:

- Cyber data extraction Interface
- XML/JSON interface

Cyber Data extraction Interface

The Cyber Data extraction Interface collects all kind of data useful to feed the computation of the Attack Graphs. These data are mainly twofold:

- Network topology
- Vulnerability scan
- XML topology from GCI

These external interfaces will operate via file input/output with a defined CSV data structure for the file content, except for the XML topology from GCI, which will be pulled from an API to automate the process

Network topology includes elements that can be listed as follow:

- Hosts interface
- Service - Application
- VLAN
- Routing tables
- Firewall rules
- Network status
- Flow matrix

And specifically, for 5G/NFV/SDN environments

- Tenant Domain VNF list

- Infra or Party Domain VNF list
- Run-time network reconfiguration
- Slices
- Micro-segments…

Vulnerability scan is typically a report generated by a tool like Nessus for example.

XML/JSON interface

A REST API will be used to query the results: attack paths, remediation, actives paths (dynamic risk analysis) and the counter measures. The answers will be in JSON format.

5.1.10 Re-utilised Technologies/Specifications

PulSAR leverages on assets defined and developed in FI-PPP and FI-WARE/FI-Core FP7 projects, especially the Cybersecurity Monitoring Generic Enabler and its Generic Enabler Reference Implementation called CyberCAPTOR.

5.1.11 References

[1] https://www.ietf.org/rfc/rfc4765.txt
[2]https://github.com/fiware-cybercaptor/cybercaptor-dataextraction/blob/master/doc/topology-file-specifications.md
[3] François Reynaud, François-Xavier Aguessy, Olivier Bettan, Mathieu Bouet and Vania Conan: Attacks against Network Functions Virtualization and Software-Defined Networking: State-of-the-art, in SecVirtNet 2016

5.1.12 Acknowledgements

Acknowledgements are dedicated to the FI-PPP FI-WARE & FI-Core projects' participants who have been working on security monitoring and cyber security monitoring assets development on which the PulSAR enabler will leverage to address 5G security requirements on the field.

5.2 Generic Collector Interface Open specifications

In this section, we present the updates of the generic log format proposed by "Generic Collector interface" (This enabler mainly aims at monitoring the 5G core network elements / components.) in R1 and recommendations to the other monitoring enablers to integrate it. This section also includes content from Generic Collector Interface open specification R1.

5.2.1 Status

At the writing time of this document, the Generic Collector interface is still under research and development. Therefore, the open specifications of Generic Collector Interface are preliminary and are subject of improvements. In the scope of Release-2, we plan to assist other enablers integration of GCI.

5.2.2 Copyright

Orange is currently the only contributor to this enabler.

5.2.3 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

5.2.4 Terms and definitions

MME	Mobility Management Entity
HSS	Home Subscriber Server
P-GW	Packet Data Network GateWay
S-GW	Serving GateWay
NFV	Network Function Virtualization
NFVO	NFV Orchestrator
VNFM	Virtualized network Function Manager
VIM	Virtualized Infrastructure Manager
SDN	Software Defined Network

Table 5-1: Acronyms list

5.2.5 Overview

(This is a reminder from the open specification in R1)

The Generic Collector Enabler, presented in Figure 5-9, aims at collecting logs and events basically generated by security monitoring enablers gathering data and logs from the 5G core network components (e.g., MME, HSS, P-GW, S-GW…). To this end, the Generic Collector Enabler consists of two engines: Server engine and Client engine.

- The Server engine is the software managing the monitoring operations. It is responsible of
    - Triggering and ending the real-time monitoring (basic function)
    - Trigger specific commands, for instance, to get data specific to a component and / or a given Network layer and /or a particular protocol (cf §5.2.8.1 Use Cases).
    - Data processing: based on the retrieved data and some other information (e.g., security patterns), the server engine can identify abnormal behaviors and incidents.
    - Providing /getting inputs to/from other monitoring enablers.
- The Client engine is the software installed in the 5G core network components. This software is responsible of
    - Reporting data about its state, for instance, the type of the component, Component ID, active interfaces, etc.
    - Reporting signalling data (i.e., control plane) that transit by it.
    - Answering any specific request originating from the Server engine.

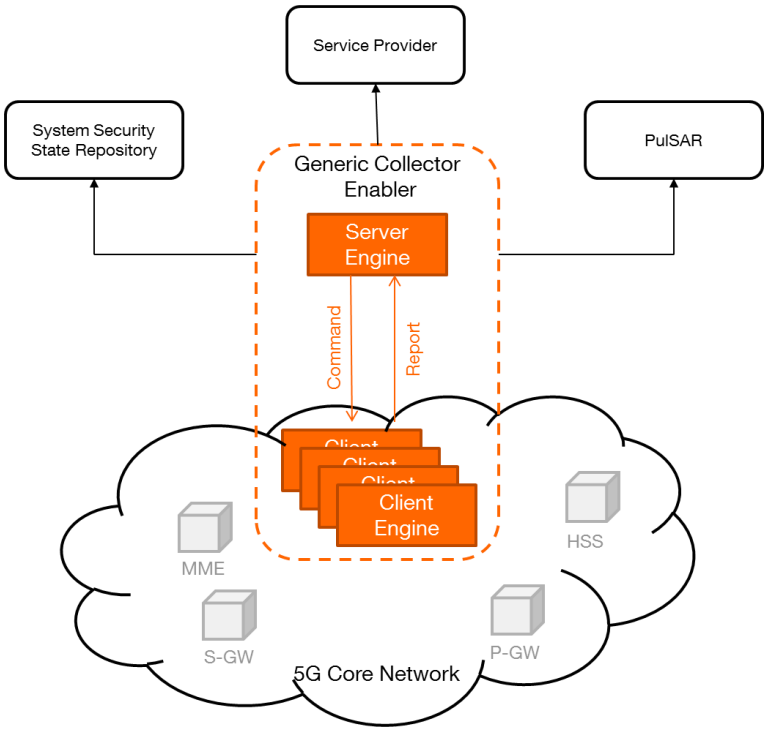


Figure 5-9: Generic Collector Enabler Overview

In Figure 5-9 the Generic Collector enabler provides inputs to PulSAR enabler as well as System Security State Repository enabler and any other Service Provider that has an agreement with the MNO to access to some network data. PulSAR enabler receives the report of the Generic Collector Interface, hence, infers the network architecture and extracts the set of identified incidents and vulnerabilities in the network. Regarding System Security State Repository enabler, it also receives all the collected events extracted from logs. Similarly, the service provider receives the logs with respect to the generic log format. These two enablers and the service provider may also have other inputs in addition to the data provided by the Generic Collector enabler. In this release, the interactions between these enablers and service provider will be formalized. Generic Collector enabler can also interact with other enablers namely other monitoring enablers to collect data from the network such as the security monitor of 5G micro-segments. In this case, the security monitor of 5G micro-segments should send / receive data to / from the Generic Collector enabler with respect to the generic log format.

5.2.6 Basic concepts

(This is a reminder from the open specification in R1)

In this section, we provide the definition of the Core Network main components / functions.

Mobility Management Entity (MME) [TS 23.401, TS 23.402, TS 36.300**]:** it is a control plane entity that handles the signalling related to mobility and security such as Non-Access Stratum signalling and security, inter Core Network node signalling for mobility between 3GPP access networks, tracking Area list management, packet Data Network GateWay and Serving GateWay selection, roaming, authentication, and lawful Interception of signalling traffic.

Home Subscriber Server (HSS) [TS 23002**]:** it is a database that contains subscription related information namely:

- User Identification, Numbering and addressing information;
- User Security information: Network access control information for authentication and authorization;
- User Location information at inter-system level: the HSS supports the user registration, and stores inter-system location information, etc.;
- User profile information.

It also provides support functions in mobility management, call and session setup, user authentication and access authorization.

Packet Data Network GateWay (P-GW) [TS 23.002, TS 23.401,TS 23.402**]:** it is the point of interconnect between the EPC and the Packet Data Network (i.e., external IP networks). It is a user plane entity. Among the functions of P-GW:

- Per-user based packet filtering (by e.g. deep packet inspection);
- Lawful Interception;
- User equipment IP address allocation;
- Transport level packet marking in the uplink and downlink
- UL and DL service level charging, gating control, rate enforcement;
- UL and DL rate enforcement based on APN-AMBR;
- DL rate enforcement based on the accumulated MBRs of the aggregate of SDFs with the same GBR QCI (e.g. by rate policing/shaping);
- DHCPv4 (server and client) and DHCPv6 (client and server) functions;

Serving GateWay (S-GW) [TS 23.401, TS 23.402**]:** it is the point of interconnect between the Core Network and the radio. It is a user plane entity. It provides a set of functions such as:

- the local Mobility Anchor point for inter-eNodeB handover;
- Mobility anchoring for inter-3GPP mobility;
- Lawful Interception;
- Packet routeing and forwarding;
- Transport level packet marking in the uplink and the downlink;
- Accounting on user and QCI granularity for inter-operator charging;
- Event reporting (change of RAT, etc.) to the Policy and Charging Rules Function (PCRF);

Network Function Virtualization (NFV) [ETSI, NFV]**:** it is a networking initiative led by Telcos, to embed current network functions (e.g., ciphering, firewall, load balancing, TCP accelerators, concentrators, DNS, QoE Management, video optimizers, etc.) in commodity hardware (high-volume standard servers, storage and switches). Network functions become then VNFs (Virtualized Network Functions). The NFV reference architecture defined by ETSI NFV group is composed of:

- **NFV Orchestrator** manages a networking service;
- **VNF Manager** ensures the life-cycle of the VNFs
- **VIM** maps the VNFs and virtual links on to the physical infrastructure (i.e., computing, memory and network)

Software Defined network (SDN) [ONF]**:** it is a novel network architecture based on abstraction, open interfaces, and control plane-data plane separation. The SDN concept is centred on the abstraction and

network programmability. Open Networking Foundation (ONF) defines SDN as “The physical separation of the network control plane from the forwarding plane, where the control plane controls several devices”. A controller SDN dictates the forwarding rules to the network elements (SDN resources) by means of flows through the southbound protocol (de facto standard OpenFlow).

Authentication, Authorization and Accounting (AAA): this is a type of applications and protocols enabling the authentication of users, the authorization of some services to these users, and collecting data about the usage of resources for billing. Diameter is one of the current AAA protocols.

5.2.7 Main interactions

5.2.7.1 Use cases

In this section, we describe an example of network architecture to which we provide later the corresponding report in Section **Error! Reference source not found.**. We consider, as shown in **Figure 5-10**, that we have four Virtualized Network Functions namely “P-GW_1”, “S-GW_1”, “MME_1” and “HSS_1”. As the names suggest, “P-GW_1” is the VNF providing the functions of a Packet Data Network GateWay; “S-GW_1” is the VNF providing the functions of a Serving GateWay; “MME_1” is the VNF providing the functions of a Mobility Management Entity and, “HSS_1” is the VNF providing the functions of a Home Subscriber Server. We also consider three hosts: h1, h2 and h3. Host h1 runs “HSS_1”. Host h2 hosts “MME_1”. “S-GW_1” and “P-GW_1” are running on host h3. In **Figure 5-10**, we also draw the connections (signalling and data exchanges) between the virtualized functions. “l1” is the virtual link between “HSS_1” and “MME_1”. “MME_1” and “S-GW_1” are connected through “l2”. “l3” is linking “S-GW_1” and “P-GW_1”. “l0” is the link between the core network and the IP-based services.

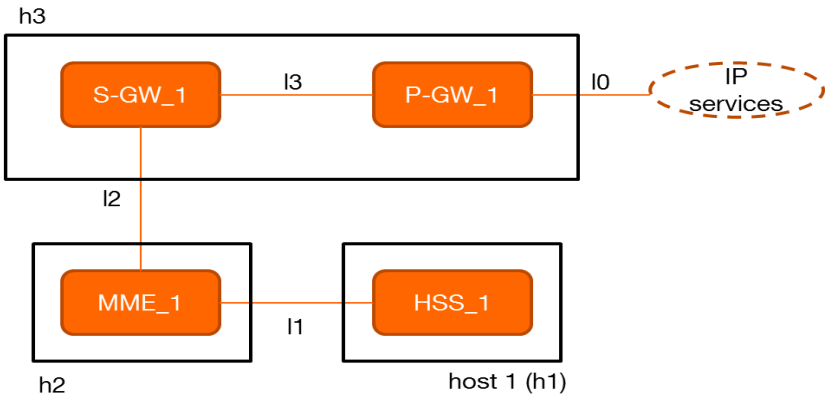


Figure 5-10: Network Architecture example

5.2.7.2 Components and interaction overview

(This is a reminder from the open specification in R1)

We distinguish two main building blocks of the Generic Collector Enabler. First, the Client Engines are the collect points. Client Engines receive Commands and return back reports containing its state and the collected data. Second, Server Engines are the analysis and stance points which can be physical as well as functional / logical. Server Engines send Commands to Client engines and other Server engines, and receive reports about the network state. Given that in 5G, new layers will arise such as Virtualization and virtualized functions layers, the security monitoring of the 5G core network will not be limited to monitoring the infrastructure. Therefore, we propose to have Client and Server engines dedicated to every

layer (i.e., Access layer, Infrastructure layer, Virtualization layer, Virtualization function layer and Applications services layer). Then, we organize Server engines into a hierarchically distributed architecture, as shown in **Figure 5-11** There are three categories / levels of Server engines: Layer server engines, Slice server engine and Operator server engine.

A Layer server engine is a server engine dedicated to a given layer (e.g., Infrastructure Server engine). For instance, Client engine implemented in the Infrastructure nodes communicates only with the Layer Server engine of this layer. Thus, it only accepts commands coming from this given layer Server engine and make reports to it. Similarly, a slice server engine (that can be physical as well as functional / logical) is a server engine dedicated to a given slice. Hence, Layer server engines located in a slice S will communicate only with the slice server engine of S. Finally, Operator server engine is the root. Only Slice server engines and some layer server engines can communicate with it.

Communications between two categories of server engines respect the same protocol (Command / Report) and the same message format. Indeed, we propose a 3GPP compliant format, i.e. an xml file that describes the requested data or reports data. The format that we propose is an enhancement of the 3GPP proposal [TS 32.421, TS 32.422, TS 32.423]. We give further details about the format in section **Error! Reference source not found.**.

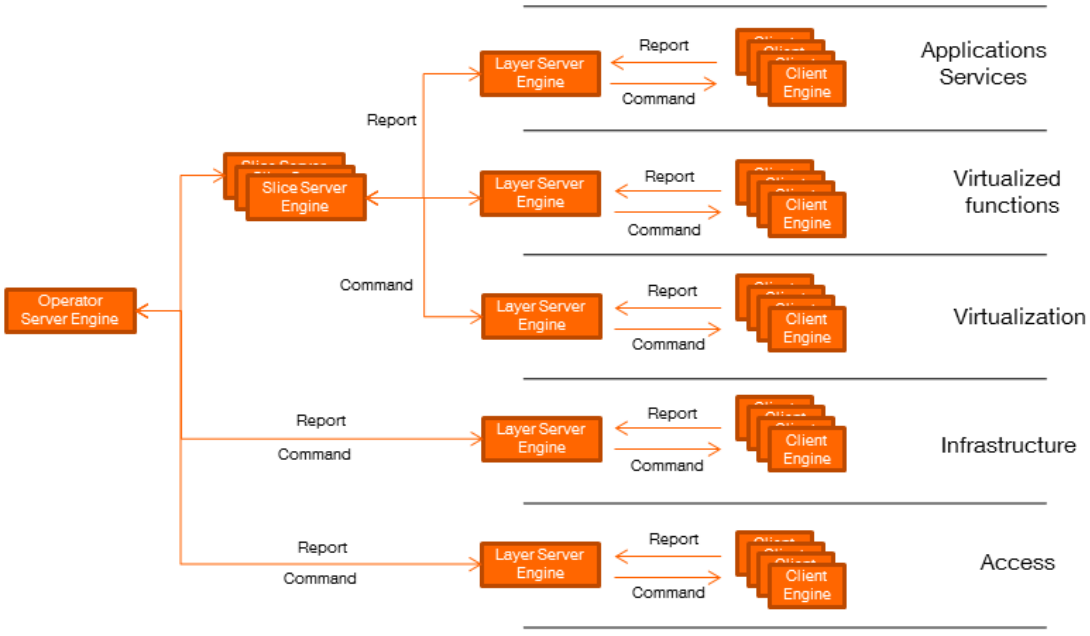


Figure 5-11: Generic Collector Enabler in layers

5.2.8 Architectural drivers

5.2.8.1 High-Level functional requirements

The Generic Collector enabler should ensure two main functional requirements. First, the monitoring of the network should not result in an overload. Interactions between the network equipment and security monitoring enablers, and the Generic Collector enabler may imply additional load in the network. This may alter the statistics, hence; impact the following processes (e.g., incident identification and remediation). Second, the Generic Collector enabler aims to gather events, vulnerabilities and incidents (with respect to the generic log format proposed in R1) from the network through some security monitoring enablers and transmit it to a set of enablers to perform real-time operations (in R2).

5.2.8.2 Link to Security Architecture

The Generic Collector Interface enabler contributes in accomplishing the objective "O7.3" of the Security Management family (cf. D2.4). This objective consists in: "5G systems must support security monitoring capable of detecting advanced cyber security threats and support coordinated monitoring between different domains and systems (e.g. mobile and satellite)".

The Generic Collector Interface has two main components: Client engine and Server engine. The Client engine should be implemented in the network components and the other monitoring enablers such as "Security Monitor for 5G Micro-Segments enabler" and "Satellite Network Monitoring enabler" to ensure the collection of logs respects the generic log format. The Server engine should be implemented in a trusted server either inside or outside the Mobile Network. The Server engine should also interact with services and enablers that will use the collected data (with respect to the log format) for other purposes such as "PulSAR" and "System Security State Repository". As mentioned in **Figure 5-12** the Generic Collector Interface should be deployed in almost all the domains defined in the Core network. This is because it helps ensuring the monitoring of the network. The exchanges of the Generic Collector Interface will be mainly located in the management strata, as shown in **Figure 5-13.**

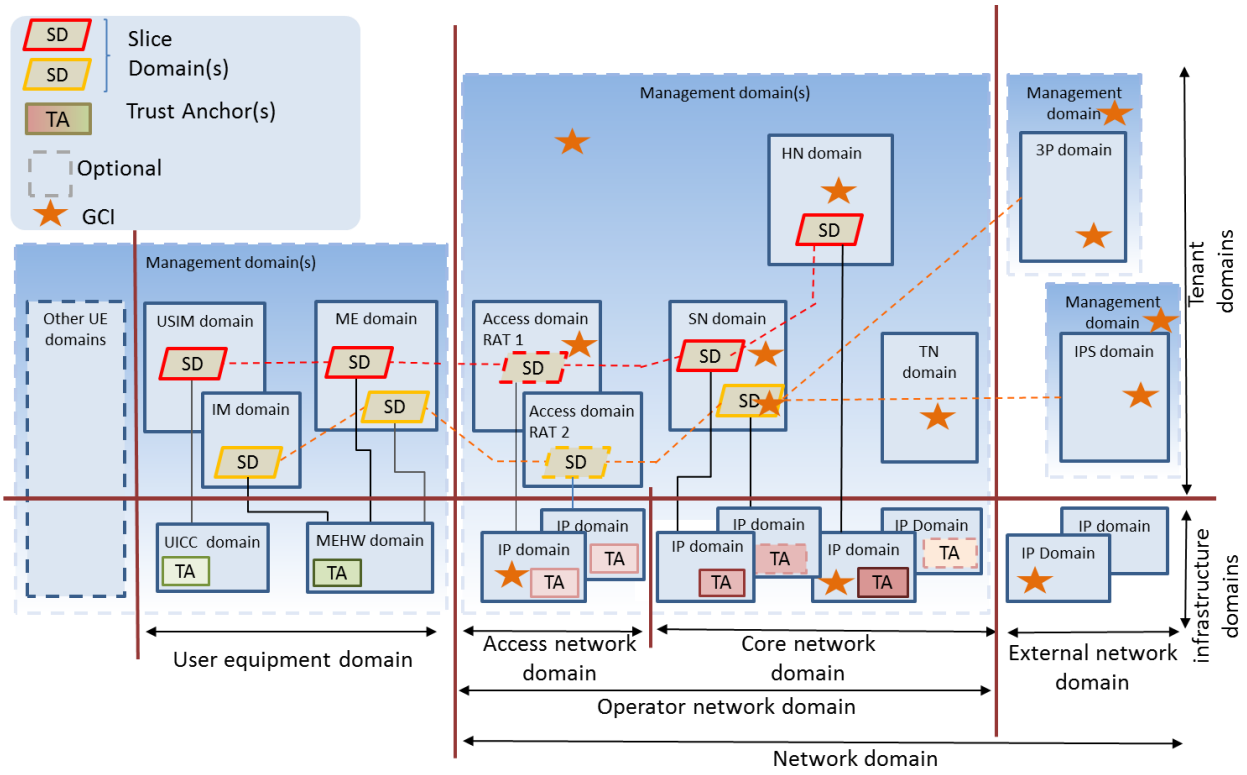


Figure 5-12: GCI potential implementation in the 5G Security Architecture

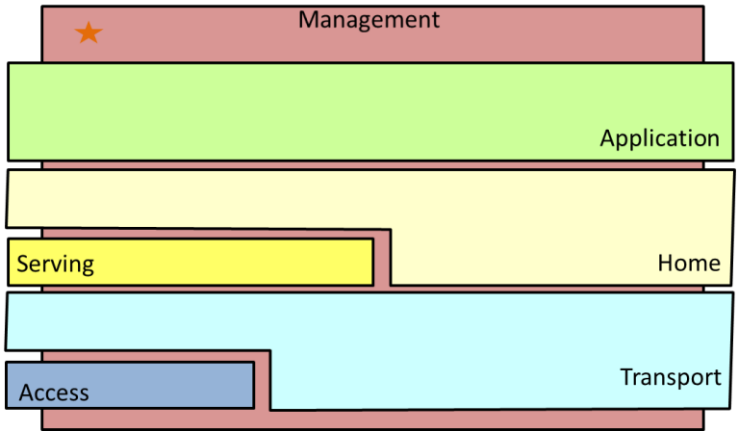


Figure 5-13: GCI exchanges

5.2.8.3 Quality attributes

For R2, we do not plan a software realization. We will assist other enablers to integrate the generic log format proposed in R1.

(This is a reminder from the open specification in R1)

The quality attributes are the following ones:

- **Performance**: capacity to handle large number of real-time events
- **Security**: the enabler should authenticate the event sources and the user of the enabler.

5.2.8.4 Technical constraints

In R2, the major technical constraint is to satisfy the requirements of enablers expecting inputs from the "Generic Collector Interface". Enablers expecting inputs from the "Generic Collector Interface" will receive data with respect to the generic and unique log format. The ideal situation is when the output of the "Generic Collector Interface" provides, in a straightforward way, the required data. In some case, the required information from enablers side will need some processing on the data sent by the "Generic Collector Interface". In the following, using an example, we will assist PulSAR enabler to get its required inputs from the log reported by the "Generic Collector Interface".

5.2.8.5 Business constraints

No business constraints.

5.2.8.6 Detailed specifications

5.2.8.7 API specifications

In this section, we provide the new version of the log format that has been proposed in R1 (i.e., XML file "report") and describe the algorithm that should be implemented by PulSAR to get the required information for its input from the received data. Similarly, the enabler "System Security State Repository" should develop an algorithm to extract information from the generic logs sent by GCI. Regarding the enabler "Security Monitor for 5G Micro-Segments", it should adapt its output / inputs dedicated to GCI to fit the generic format of GCI.

Similarly, to R1, we mainly focus on four mobile core network entities, i.e., MME, HSS, P-GW and S-GW. Therefore, our specification is strongly inspired by the 3GPP proposal in [TS 33.421, TS 33.422, TS 33.423].

Indeed, we consider MME, HSS, P-GW and S-GW as virtualized network functions (VNFs). Hence, we improve the 3GPP proposal with VNF monitoring information. In **Table 5-2**, we provide details about the format of a Report file (i.e., fields). **Table 5-2** is organized as follows: Interface / Entity Name - IE name – Notes. **Table 5-3** and **Table 5-4** give further details about two fields (i.e., "SwImageDescriptor" and "VirtualStorageDescriptor"). In **Table 5-5**, we detail incident fields.

Interface name	IE name	Notes
NFV Orchestrator		
	ID	This is a unique identifier of a NFV Orchestrator.
	NFV Managers number	We cannot directly retrieve this value. We need to perform some computation to have it.
	State	Ideally, this value will be "0" if the NFV Orchestrator is functional and "1" if the NFV Orchestrator is defective. But as there is no implementation (to the best of our knowledge), we do not know if we can get such an information.
	VNF Managers IDs	These are a set of the identifiers of VNF Managers managed by a given NFV Orchestrator
	VNF Managers states	Ideally, for every NFV Manager, this value will be "0" if the NFV Manager is functional and "1" if the node is defective. But as there is no implementation (to the best of our knowledge), we do not know if we can get such an information.
	VNF ID – VL ID	Ideally, a NFV Orchestrator should have the list of VNFs that it is managing and its associated virtual links (VL).
VIM		
	ID	This is the identifier of VIM
	NFV Orchestrator ID	This is the ID of the NFV Orchestrator that manages the VIM.
	VNF address	For every VNF, this is the physical address of the VNF
	VNF-StorageID	For every VNF, this is the Identifier of the virtualised storage resource.
	VNF-storageName	For every VNF, this is the name of the virtualised storage resource.
	VNF-typeOfStorage	For every VNF, this is the type of virtualised storage resource (e.g. volume, object).
	VNF-sizeOfStorage	For every VNF, this is the size of virtualised storage resource (e.g. size of volume, in GB).
	VNF-ownerId	For every VNF, this is the identifier of the virtualised resource that owns and uses such a virtualised storage resource. The value can be NULL if the virtualised storage is not attached yet to any other resource (e.g., a virtual machine).
	VNF-zoneId	For every VNF, If present, it identifies the Resource Zone where the virtual storage resources have been allocated.
	VNF-hostId	For every VNF, this is the identifier of the host where the virtualised storage resource is allocated. A cardinality of 0 refers to distributed storage solutions.

	State	For every VNF, this is the state of the resource allocated to the VNF.
	VNF-metadata	List of metadata key-value pairs used by the consumer to associate meaningful metadata to the related virtualised resource.
	VNF-startTime	Timestamp to start the consumption of the VNF resources. If the time value is 0, resources are reserved for immediate use.
	VNF-endTime	Timestamp indicating the end time of the reservation (when it is expected that the resources will no longer be needed) and used by the VIM to schedule the reservation. If not present, resources are reserved to the VNF for unlimited usage time.
	VNF ID – VL ID	For every VNF, the VIM should have its associated virtual links (VL).
VNF Manager		
	ID	This is a unique identifier of the VNF Manager.
	State	Ideally, this value will be “0” if the VNF Manager is functional and “1” if the VNF Manager is defective. But as there is no implementation (to the best of our knowledge), we do not know if we can get such an information.
	NFV Orchestrator ID	This is the ID of the NFV Orchestrator that manages the VNF Manager
	VNFs number	Perhaps, we cannot directly retrieve this value. We need to perform some computation to have it.
	VNFs IDs	The identifiers of VNFs that are managed by a given VNF Manager.
	VNFs states	Ideally, for every VNF, this value will be “0” if the VNF is functional and “1” if the VNF is defective.
MME-VNF		
	ID	This is the identifier of the MME-VNF
	VNF Manager ID	This is the ID of the VNF Manager that manages the MEE-VNF
	VIM ID	This is the ID of the VIM that manages the MEE-VNF
	MME-VNF state	Ideally, this value will be “0” if the VNF is functional and “1” if the VNF is defective.
	SwImageDescriptor	This describes the loaded software. We give further details about this IE in **Table 5-3**.
	VirtualStorageDescriptor	This describes the storage parameter of a virtual storage associated to a VNF. We give further details about this IE in **Table 5-4**.
	The remaining fields are the same as in 3GPP [TS 32 423]	
HSS-VNF		
	ID	This is the identifier of the HSS -VNF
	VNF Manager ID	This is the ID of the VNF Manager that manages the HSS -VNF
	VIM ID	This is the ID of the VIM that manages the HSS -VNF
	HSS-VNF state	Ideally, this value will be “0” if the VNF is functional and “1” if the VNF is defective.
	SwImageDescriptor	This describes the loaded software. We give further details about this IE in **Table 5-3**.

	VirtualStorageDescriptor	This describes the storage parameter of a virtual storage associated to a VNF. We give further details about this IE in **Table 5-4**.
	The remaining fields are the same as in 3GPP [TS 32 423]	
P-GW-VNF		
	ID	This is the identifier of the P-GW-VNF
	VNF Manager ID	This is the ID of the VNF Manager that manages the P-GW-VNF
	VIM ID	This is the ID of the VIM that manages the P-GW-VNF
	P-GW-VNF state	Ideally, this value will be "0" if the VNF is functional and "1" if the VNF is defective.
	SwImageDescriptor	This describes the loaded software. We give further details about this IE in **Table 5-3**
	VirtualStorageDescriptor	This describes the storage parameter of a virtual storage associated to a VNF. We give further details about this IE in **Table 5-4**.
	The remaining fields are the same as in 3GPP [TS 32 423]	
S-GW-VNF		
	ID	This is the identifier of the S-GW-VNF
	VNF Manager ID	This is the ID of the VNF Manager that manages the S-GW-VNF
	VIM ID	This is the ID of the VIM that manages the S-GW-VNF
	S-GW-VNF state	Ideally, this value will be "0" if the VNF is functional and "1" if the VNF is defective.
	SwImageDescriptor	This describes the loaded software. We give further details about this IE in **Table 5-3**.
	VirtualStorageDescriptor	This describes the storage parameter of a virtual storage associated to a VNF. We give further details about this IE in **Table 5-4**
	The remaining fields are the same as in 3GPP [TS 32 423]	
SDN Controller		
southband interface	**nodes number**	We cannot directly retrieve this value. We need to perform some computation to have it.
	nodes IDs	The node IDs are called Data Path ID (DPID). It consists in 64 bits such that the lower 48 bits are intended for the switch MAC address, while the top 16 bits are up to the implementer.
	links	A link connects two nodes. Therefore, it is identified as a couple of DPID. (DPID - DPID)
	links number	We cannot directly retrieve this value. We need to perform some computation to have it.
	nodes states	Ideally, this value will be "0" if the node is functional and "1" if the node is defective. But current implementations do not allow it.
	hosts number	We cannot directly retrieve this value. We need to perform some computation to have it.
	Hosts IDs	A host ID can be an IP address or / and a MAC address.

	N_packetsPerRule	This is the number of packets assigned to a rule
	N_packetsPerPort	This is the number of packets that passed a port
	N_packetsWRule	This is the number of packets without rules
More		This field will contain any additional data that do not fit within the previous elements
	Type	This is the type of the data
	Value	This is the data.

Table 5-2: Report file details

IE name	IE Parameters	Notes
SwImageDescriptor		
	ID	This is the identifier of the SwImageDescriptor
	Type	This indicates the type of virtualised storage resource (e.g. volume, object).
	Size	This indicates the size of virtualised storage resource (e.g. size of volume, in GB).
	swImageDescriptorPointer	This is a pointer to Software image of a VNF to be loaded on the VirtualStorage Resource based on this VirtualStorageDescriptor.
	swImage	This is a reference to the actual software image. The reference can be relative to the root of the VNF package or can be a URL.
	Version	This is the version of the software image.
	operatingSystem	This identifies the operating system used in the software image. This may also identify if a 32bit or 64 bit software image is used.
	supportedVirtualisationEnvironment	This identifies the virtualisation environments (e.g. hypervisor) compatible with this software image.

Table 5-3: The details of "SwImageDescriptor"

IE name	IE Parameters	Notes
VirtualStorageDescriptor		
	ID	This is the identifier of the VirtualStorageDescriptor
	Checksum	
	containerFormat	The container format describes the container file format in which the software image is provided.
	diskFormat	The disk format of a software image is the format of the underlying disk image
	minDisk	The minimal Disk for this software image.
	minRam	The minimal RAM for this software image.
	Size	This is the size of the software image

Table 5-4: The details of "VirtualStorageDescriptor"

IE name	Attributes	Notes
HostID	MAC Address and / or IP address	The identifier of the host that detected / reported an incident.
VNFdescriptor	- ID - VNF Manager ID - VIM ID - VNF state - SwImageDescriptor - VirtualStorageDescriptor	The VNF descriptor contains all the information associated to the VNF that generated / identified the incident.
IncidentDescriptor	- ID - Session ID - Protocol ID	The Incident descriptor contains all the information associated to incident that occurred.

Table 5-5: Incident format details

While studying the integration of the generic log format and PulSAR, we noted that the log format can be enhanced. Indeed, PulSAR requires for its internal functioning to run some vulnerably scan in the network elements. In **Table 5-6**, we provide details about vulnerability attributes. You will notice that there are some similarities with incident attributes. This is due to the fact that vulnerability may cause an incident.

IE name	Attributes	Notes
ID		This is the identifier of the vulnerability
ID-VNF		This is the identifier of the VNF where the vulnerability has been detected.
Service	ID	This is the ID of the service where the vulnerability has been detected.
	Version	This is the version of the service where the vulnerability has been detected.
	Protocol	This is protocol associated to the detected vulnerability.
	Port	This is the port associated to the detected vulnerability.
	Descriptor	This is a description of the detected vulnerability.

Table 5-6: Vulnerability format details

5.2.8.8 Examples

In what follows, we provide the report that corresponds to the network architecture described in Section **Error! Reference source not found.**.

```
<?xml version="1.0" encoding="UTF-8"?>

<report>

    <NFVOrchestrator>

        <ID> ochestrateur_global </ID>

        <NFVManagersNumber> 01 </NFVManagersNumber>

        <State> 0 </State>
```

```
        <VNFManagersIDs> VNFM_1 </VNFManagersIDs>
        <VNFManagersStates> 0 </VNFManagersStates>
        <VNFID-VLID> mme_1;l1 </VNFID-VLID>
        <VNFID-VLID> mme_1;l2 </VNFID-VLID>
        <VNFID-VLID> hss_1;l1 </VNFID-VLID>
        <VNFID-VLID> sgw_1;l2 </VNFID-VLID>
        <VNFID-VLID> sgw_1;l3 </VNFID-VLID>
        <VNFID-VLID> pgw_1;l3 </VNFID-VLID>
        <VNFID-VLID> pgw_1;l0 </VNFID-VLID>
</NFVOrchestrator>
<VIM>
        <ID> vim_1 </ID>
        <NFVOrchestratorID> ochestrateur_global </NFVOrchestratorID>
        <VNF>
                <ID> mme_1 </ID>
                <VNFaddress> xx </VNFaddress>
                <VNFStorageID> xx </VNFStorageID>
                <VNFStorageName> xx </VNFStorageName>
                <VNFTypeOfStorage> xx </VNFTypeOfStorage>
                <VNFSizeOfStorage> xx </VNFSizeOfStorage>
                <VNFOwnerId> xx </VNFOwnerId>
                <VNFZoneId> xx </VNFZoneId>
                <VNFHostId> h2 </VNFHostId>
                <State> xx </State>
                <VNFmetadata> xxx </VNFmetadata>
                <VNFstartTime> xx </VNFstartTime>
                <VNFendTime> xx </VNFendTime>
```

<VNFID-VLID> mme_1;l1 </VNFID-VLID>

<VNFID-VLID> mme_1;l2 </VNFID-VLID>

</VNF>

<VNF>

<ID> hss_1 </ID>

<VNFaddress> xx </VNFaddress>

<VNFStorageID> xx </VNFStorageID>

<VNFStorageName> xx </VNFStorageName>

<VNFTypeOfStorage> xx </VNFTypeOfStorage>

<VNFSizeOfStorage> xx </VNFSizeOfStorage>

<VNFOwnerId> xx </VNFOwnerId>

<VNFZoneId> xx </VNFZoneId>

<VNFHostId> h1 </VNFHostId>

<State> xx </State>

<VNFmetadata> xxx </VNFmetadata>

<VNFstartTime> xx </VNFstartTime>

<VNFendTime> xx </VNFendTime>

<VNFID-VLID> hss_1;l1 </VNFID-VLID>

</VNF>

<VNF>

<ID> sgw_1 </ID>

<VNFaddress> xx </VNFaddress>

<VNFStorageID> xx </VNFStorageID>

<VNFStorageName> xx </VNFStorageName>

<VNFTypeOfStorage> xx </VNFTypeOfStorage>

<VNFSizeOfStorage> xx </VNFSizeOfStorage>

<VNFOwnerId> xx </VNFOwnerId>

```
            <VNFZoneId> xx </VNFZoneId>
            <VNFHostId> h3 </VNFHostId>
            <State> xx </State>
            <VNFmetadata> xxx </VNFmetadata>
            <VNFstartTime> xx </VNFstartTime>
            <VNFendTime> xx </VNFendTime>
            <VNFID-VLID> sgw_1;l2 </VNFID-VLID>
            <VNFID-VLID> sgw_1;l3 </VNFID-VLID>
        </VNF>
        <VNF>
            <ID> pgw_1 </ID>
            <VNFaddress> xx </VNFaddress>
            <VNFStorageID> xx </VNFStorageID>
            <VNFStorageName> xx </VNFStorageName>
            <VNFTypeOfStorage> xx </VNFTypeOfStorage>
            <VNFSizeOfStorage> xx </VNFSizeOfStorage>
            <VNFOwnerId> xx </VNFOwnerId>
            <VNFZoneId> xx </VNFZoneId>
            <VNFHostId> h3 </VNFHostId>
            <State> xx </State>
            <VNFmetadata> xxx </VNFmetadata>
            <VNFstartTime> xx </VNFstartTime>
            <VNFendTime> xx </VNFendTime>
            <VNFID-VLID> pgw_1;l3 </VNFID-VLID>
            <VNFID-VLID> pgw_1;l0 </VNFID-VLID>
        </VNF>
    </VIM>
```

```
<VNFManager>
        <ID> vnfm_1 </ID>
        <State> 0 </State>
        <NFVOrchestratorID> ochestrateur_global </NFVOrchestratorID>
        <VNFsnumber> 4 </VNFsnumber>
        <VNFsIds> mme_1; hss_1; sgw_1; pgw_1 </VNFsIds>
        <VNFsStates> 0; 0; 0; 0 </VNFsStates>
</VNFManager>
<MME-VNF>
        <ID> mme_1 </ID>
        <VNFManagerId> vnfm_1 </VNFManagerId>
        <VIMId> xx </VIMId>
        <MME-VNFState> 0 </MME-VNFState>
        <SwImageDescriptor> xx </SwImageDescriptor>
        <VirtualStorageDescriptor> xx </VirtualStorageDescriptor>
        <More> xx </More>
</MME-VNF>
<HSS-VNF>
        <ID> hss_1 </ID>
        <VNFManagerId> vnfm_1 </VNFManagerId>
        <VIMId> xx </VIMId>
        <HSS-VNFState> 0 </HSS-VNFState>
        <SwImageDescriptor> xx </SwImageDescriptor>
        <VirtualStorageDescriptor> xx </VirtualStorageDescriptor>
        <More> xx </More>
</HSS-VNF>
<S-GW-VNF>
```

```
        <ID> sgw_1 </ID>
        <VNFManagerId> vnfm_1 </VNFManagerId>
        <VIMId> xx </VIMId>
        <S-GW-VNFState> 0 </S-GW-VNFState>
        <SwImageDescriptor> xx </SwImageDescriptor>
        <VirtualStorageDescriptor> xx </VirtualStorageDescriptor>
        <More> xx </More>
</S-GW-VNF>
<P-GW-VNF>
        <ID> pgw_1 </ID>
        <VNFManagerId> vnfm_1 </VNFManagerId>
        <VIMId> xx </VIMId>
        <P-GW-VNFState> 0 </P-GW-VNFState>
        <SwImageDescriptor> xx </SwImageDescriptor>
        <VirtualStorageDescriptor> xx </VirtualStorageDescriptor>
        <More> xx </More>
</P-GW-VNF>
<vul>
        <ID> vul_1 </ID>
        <ID-VNF> mme_1 </ID-VNF>
        <Service>
                <ID>apache2</ID>
                <Version> 2.2.4-rc10 </Version>
                <Protocol> TCP </Protocol>
                <Port> 443 </Port>
                <Descriptor> CVE-2013-2249 </Descriptor>
        </Service>
```

```
    </vul>
    <vul>
        <ID> vul_2 </ID>
        <ID-VNF> sgw_1 </ID-VNF>
        <Service>
            <ID>postgreSQL</ID>
            <Version> 42 </Version>
            <Protocol> TCP </Protocol>
            <Port> 5432 </Port>
            <Descriptor> CVE-2013-1903 </Descriptor>
        </Service>
    </vul>
    <vul>
        <ID> vul_3 </ID>
        <ID-VNF> pgw_1 </ID-VNF>
        <Service>
            <ID>apache2</ID>
            <Version> unknown </Version>
            <Protocol> TCP </Protocol>
            <Port> 80 </Port>
            <Descriptor> CVE-2011-3192 </Descriptor>
        </Service>
    </vul>
</report>
```

PulSAR and GCI integration:

PulSAR inputs consist of 3 CSV files and a file containing a vulnerability scan. These files are used by PulSAR to build the network topology and generate graph attacks. In the following, we explain how the inputs files of PulSAR can be generated based on logs sent by GCI.

1. The file “hosts-interfaces.csv »: is the file containing the link between hosts and VNFs.

This information can be found in a straightforward way. Indeed, in the tag <NFV Orchestrator>, we can find all the tags « VNF ID - VL ID » that provide the VNFs IDs and its corresponding links.

2. The file “controllers-hosts.csv” includes the mapping between VMs and orchestrators.

This information needs some processing. First, find the tags corresponding the VNFs namely “HSS-VNF”, “MME-VNF”, “S-GW-VNF” and “P-GW-VNF”. The first information that can be found is the identifier of the VNF manager managing the VNF. Then, Using the VNF manager identifier, find its associated tag, in the VNF manager tag, you can extract the NFV Orchestrator identifier.

3. The file « hosts-vms.csv » encloses the mapping between VMs and physical machines.

This information can be easily deduced. Find the tag “<VIM>”. In this tag, for every VNF, you have the tag “<VNF-hostId> » that indicates the physical machine running the VNF.

5.2.9 Re-utilised Technologies/Specifications

(This is a reminder from the open specification in R1)

We would like to enhance the following 3GPP specifications [TS 32.421, TS 32.422, TS 32.423]. These specifications provide details about subscriber and equipment trace namely trace data definition and management, trace concept and requirements, and trace control and configuration management. Indeed, these specifications focus on the call level (User plane) in UMTS to enable monitoring and optimization operations.

Compared to UMTS, 5G network will encompass new layers. In addition to the physical and application layers, a virtualization and slicing layers will probably arise. Along with the new layers, new network components / entities will show up such us the NFV Orchestrator, VIM, VNF Manager or SDN controller. Because of this new context, we need to enhance the current security monitoring approaches in order to face the new security challenges of 5G.

In these circumstances, the Generic Collector enabler will perform security monitoring. However, we pay further attention on the control plane (contrarily to [TS 32.421, TS 32.422, TS 32.423]). In the scope of R1, we provided a common format of collected events and logs.

In the scope of R2, we enhance the generic log format and support other enablers to integrate GCI.

5.2.10 References

[ETSI] ETSI NFV Group Specification: “Network Functions Virtualisation (NFV); Management and Orchestration”, Dec. 2014

[NFV] R. Guerzoni, “Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action. Introductory white paper,” in SDN and OpenFlow World Congress, June 2012.

[ONF] Open Networking Foundation, ‘SDN Architecture Overview’. December 2013. Available at: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf

[TS 32.421] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Subscriber and equipment trace; Trace concepts and requirements (Release 13)

[TS 32.422] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Subscriber and equipment trace; Trace control and configuration management (Release 13)

[TS 32.423] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication management; Subscriber and equipment trace; Trace data definition and management (Release 13)

5.3 Security Monitor for 5G Micro-Segments -Open specifications

5.3.1 Preface

This section presents 'Security Monitor for 5G Micro-Segments' enabler.

5.3.2 Status

This document provides an open specification of the monitoring framework and features that will be implemented for the second release of the enabler.

5.3.3 Copyright

5.3.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

5.3.5 Terms and definitions

AAA	Authentication, Authorization and Accounting
CEP	Complex Event Processing
DoS	Denial of Service
eNodeB	Evolved Node B
EPC	Evolved Packet Core
JSON	JavaScript Object Notation
IDS	Intrusion Detection System
IPS	Intrusion Protection System
KPI	Key Performance Indicator
NFV	Network Function Virtualization
SDN	Software Defined Network
SIEM	Security Information and Event Management

5.3.6 Overview

5.3.6.1 5G Micro-Segments

Micro-segments are isolated parts of 5G network that have been dedicated e.g. for applications or organizations. For instance, a micro-segment may be dedicated for IoT communication of a company or for vehicular communication. Micro-segments can be created using Software Defined Networking (SDN) and Network Function Virtualization (NFV) concepts and deployed to different parts in 5G architecture, to access, serving or transport network. Micro-segments may also be chained over administrative domains to create end-to-end isolation for applications. Micro-segments, if correctly implemented, minimize the attack surface (as an attack in one micro-segment can't spread to other micro-segments). They also ease configuration and customization of security as each micro-segment may have its own security functions.

5.3.6.2 Security Monitor

The 'security monitoring enabler' collects information on security related events in micro-segments and infers knowledge by processing and combining these events. The goal of is to enable gaining of real-time awareness of the security situation in micro-segments and detection of ongoing security incidents.

Micro-segmentation based monitoring provides:

- ***Customization*** - Security monitoring can be customized for different 5G applications or organizations using 5G. For instance, some micro-segments can be deeply monitored to detect stealth incidents (e.g. advanced persistent threats) and some micro-segments can be monitored only in lightweight-manner (e.g. to detect DoS attacks). The light monitoring could, for instance, inspection of communication statistics while the heavier monitoring could include inspection of communication in the different levels (cross-layer monitoring), inspection of encrypted traffic (decrypting payload for monitoring). The monitored aspects may also vary from segment to segment (e.g. searching different known threat patterns against availability and versus detection of anomalies caused by sophisticated attacks).
- ***Dynamic adaptation*** - The monitoring solution may be used to make application (micro-segment) specific 5G security solutions more autonomous. The intensity or focus of monitoring may change dynamically based e.g. on changes on micro-segment's topology or (monitored) risk levels. For instance, detected suspicious traffic may trigger more intensified monitoring. Also, other security functions may be adapted according to knowledge inferred by monitoring (tightening AAA policies, quarantining nodes etc.).
- ***Scalability*** – as heavy monitoring solutions do not have to be resourced for whole 5G network but resources may be targeted only for micro-segmented part of the network.
- ***Accuracy*** – as micro-segments have fewer nodes, the system can monitor more parameters and features. The monitoring enabler can combine and correlate information from different layers of 5G networks (including 5G specific layers, like access and serving networks, as well as network virtualization layers). This correlation could enable detection of incidents that would otherwise be undetected. Further, in some cases the micro-segments may contain more homogeneous traffic - coherent traffic patterns - enabling easier detection of anomalies.

5.3.6.3 Distributed Complex Event Processing Framework – Release 1 Feature

The first phase of development – Release 1 – provided a 'distributed complex event processing framework'. This framework provides a *tool chain* that can be used when constructing different monitoring setups. It provides a mechanism to collect and share events from various sources and to distribute them to security inherence components.

The framework increases **scalability** and **flexibility** of 5G security monitoring by:

- enabling new heterogeneous event sources to be easily added and reusable components to be used for event stream to be processing (e.g. merging, aggregating, inferring)
- enabling different 'inherence components' - such as pattern detectors, machine learners, correlation analyzers… - to be integrated to the system when a need arises in different micro-segments (the solution provides efficiency as events are provided to those components that are interested on them and only for those)

- deploying *state-of-the-art analytics 'big data' technologies* (in particular Apache Kafka[9] and Spark[10]) that are able to handle large amounts of event streams in near real-time

A downside of the approach is increased complexity. Existing sources of monitoring information ('probes'), Security Information and Event Management (SIEM), Intrusion Detection or Protection Systems (IDS/IPS) etc. must be adapted to connect through the framework. However, the hypothesis is that the scalability and flexibility benefits exceed the adaptation costs.

5.3.6.4 Extended coverage, control analytics and cooperation support – Release 2 Features

The security features developed for the second release, will extend the monitoring framework by:

- **extending data gathering features** - The monitoring enabler will gather information from the micro-segmentation enabler (collecting SDN specific configuration and network traffic information) as well as from different 5G access / serving network specific information sources.
- **providing risk-based adaptation functionality for micro-segments** - The enabler will provide machine-learning (*anomaly detection*) based control functionality that can be used adapting micro-segments' defences. The monitor will serve as an information collector for Trust Metric Enabler. The monitoring enabler's control features will then utilize output from Trust Metric Enabler to make control decisions.
- **enabling cross-domain information exchange** - The enabler may be utilized in multi-domain multi-operator scenarios natively by allowing different domains to connect to the framework in order to exchange monitoring information, trust measurements or incident notifications. In the second release, the enabler will also support exporting of data to other domains following the definitions of Generic Collector Interface (CGI). CGI support is only one directional. Other domains may import data to the micro-segment monitor through monitor's native interfaces.

5.3.7 Basic concepts

Security event is an occurrence in a system that is relevant to the security of the system. For instance, changes in network topologies and new communication flows between devices are events.

Security Monitoring is a process of collecting, analysing, and inferring of security event information in order to gain awareness of system's security state and to detect and respond to security incidents.

Complex Event Processing (CEP) is a process of analysing streams of information about events from multiple sources in order to extract meaningful knowledge. Typical CEP functions are, for instance, event filtering, aggregation and transformation, as well as pattern and correlation detection. When combined with history information there may also be functions for anomaly detection. CEP functions are typically chained to derive new knowledge from the event streams as illustrated in the simple example in **Figure 5-14**.

[9] Apache ***Kafka*** is publish-and-subscribe system that was developed originally by LinkedIn. It is designed to be fast, scalable, and durable. Kafka brokers can be clustered to provide more resources elastically and transparently. A broker keeps messages on disk and replicates them within a cluster to prevent data losses. Each broker should be able to store terabytes of messages and handle megabytes of reads and writes per second from thousands of clients.

[10] Apache ***Spark*** is a general engine for cluster-based data processing originating from UC Berkeley. It provides specialized data processing libraries (including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming) which may be combined to create own parallelized applications. Spark is designed to be fast and by supporting mainstream languages (Java, Scala, Python and R languages) it is easily accessible for developers. It can be run as a stand-alone mode or e.g. within Hadoop - which is framework for distributed storage (using HDFS) and processing (using MapReduce) of large sets of (history) data.

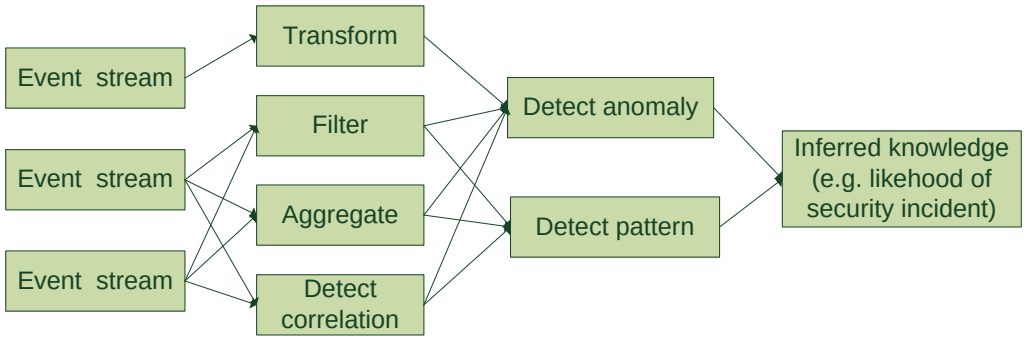


Figure 5-14: Example illustrating typical CEP functions which are chained to infer knowledge from events.

To ease development of CEP applications, different software engines have emerged, enabling developers to easily specify CEP rules. Processing of large event streams (big data) can be based on frameworks which enable parallelization of the computation. Notable existing stream processing tools are e.g. Apache Storm [Storm] and Apache Spark Streaming [Spark].

Publish-and-subscribe is a communication paradigm (illustrated in **Figure 5-15**) where information on a particular topic (event) is published through a broker that forwards this information to those parties that have subscribed the topic related updates.

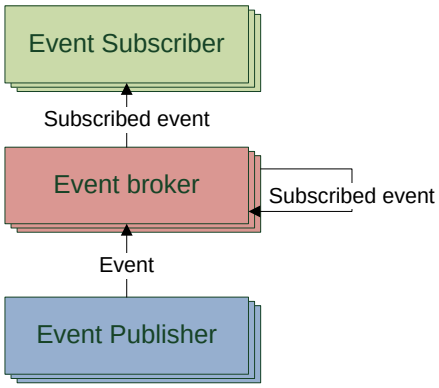


Figure 5-15: Publish-and-subscribe communication paradigm for event distribution.

The publish-and-subscribe paradigm provides greater scalability and flexibility as network topology can be dynamic (new publishers and subscribers can be added and removed). To increase scalability, the event brokers can be chained so that particular information is forwarded from a broker to another. Examples of open source brokers include e.g. Apache Kafka [Kafka].

Anomaly detection is a process of identifying events that do not conform to an expected event patterns and which might indicate an ongoing security incident. Different machine learning algorithms may be applied for anomaly detection. Typically, an anomaly detection system first learns what is normal behaviour, e.g. by clustering events, and later identifies anomalies as deviations to this behaviour, e.g. events not fitting to learned clusters. Anomaly detection enables capturing of previously unknown attacks. However, application / risk specific symptoms must be known in advance to enable feasible selection of monitored features.

Streaming K-means clustering [Ail09] is an unsupervised machine-learning algorithm used for anomaly detection. It is an adaptation of k-means algorithm [MacQ67, LLo82] for dynamic environments and particularly for scenarios where environment is frequently changing (even in normal non-attack situations).

The k-means algorithm partitions a set of data points (feature vectors) into k clusters. Streaming K-means provides support for forgetfulness; the algorithm can learn over time how the clustered data changes.

5.3.8 Main interactions

5.3.8.1 Use cases

The framework enables deployment of different security monitoring applications for different use cases. First releases will focus on monitoring of network communication in micro-segments. Here we will highlight one simple scenario.

Actors/roles in the use case are the following:

- Micro-segment provider – an entity who controls switches and SDN controller and who is able collect event information e.g. network statistics. This actor is typically the (virtual) operator. The micro-segment provider opens its APIs for inference provider to acquire security relevant information from the network (publishes events).
- Micro-segment subscriber – an organization or company requiring isolated 5G network services (and monitoring) for a particular application. A subscriber may also be another operator that is creating an end-to-end isolated service for its customers (cannot produce all services itself e.g. due to geographical restrictions).
- Security analytics provider – an entity inferring new security knowledge from monitored event data. The purpose of the analytics provider is to gain security awareness that can then be used to fast reactions to security incidents by controlling micro-segment's behaviour. The role is its own but the actor may be the micro-segment provider, a third-party, or the micro-segment subscriber. The analytics services may be tailored for each customer or customer segment.

In the use case, a micro-segment has been subscribed by a company deploying critical IoT devices. The company needs high assurances on the authenticity, availability, and confidentiality of the communication between the devices and the cloud service. There is a risk that the devices or switches become compromised as devices are deployed to untrusted physical locations. The risk cannot be completely prevented with proactive perimeter defences - such as authentication and authorization. However, some ongoing incidents may be reactively detected by security monitoring. The detectable incidents are such which have symptoms that are difficult to hide – for example, delays (a symptom of man-in-the-middle attack), excessive traffic (a symptom of denial-of-service attack), and messages with 'odd' target or source addresses (a symptom of reconnaissance or control channels to infected devices).

The micro-segment subscriber decides to use a security monitoring service that the security analytics provider is 'selling' as an extra assurance for homogeneous IoT traffic. The switches in the micro-segment collect (and publish) data on who is in communication with whom and when. The anomaly detector components of the monitoring solution (subscribe and) use this information to produce estimates on the probability of on-going risks. The operator providing micro-segment may also deploy functions for quarantining components. The monitoring system is automatically able to request the micro-segment to quarantine nodes when the probability of node's compromise is considered large enough. If micro-segment cannot be secured by quarantining, the micro-segment subscriber may be given a notification that the trust level does not match subscriber's requirements.

5.3.8.2 Components and interaction overview

The architecture for security event distribution is shown in **Figure 5-16**. The architecture separates (network and infrastructure specific) security event sources from (application and resource specific) security inference components by adding an event distribution and event processing layer (red in the figure) between them. The figure illustrates also planned event publishers (blue in the figure) and event consumers (i.e. inference components – green in the figure). The figure also encompasses some event sources planned for future releases as well as connections to other domains by exporting data through Generic Collector Interface (light blue).

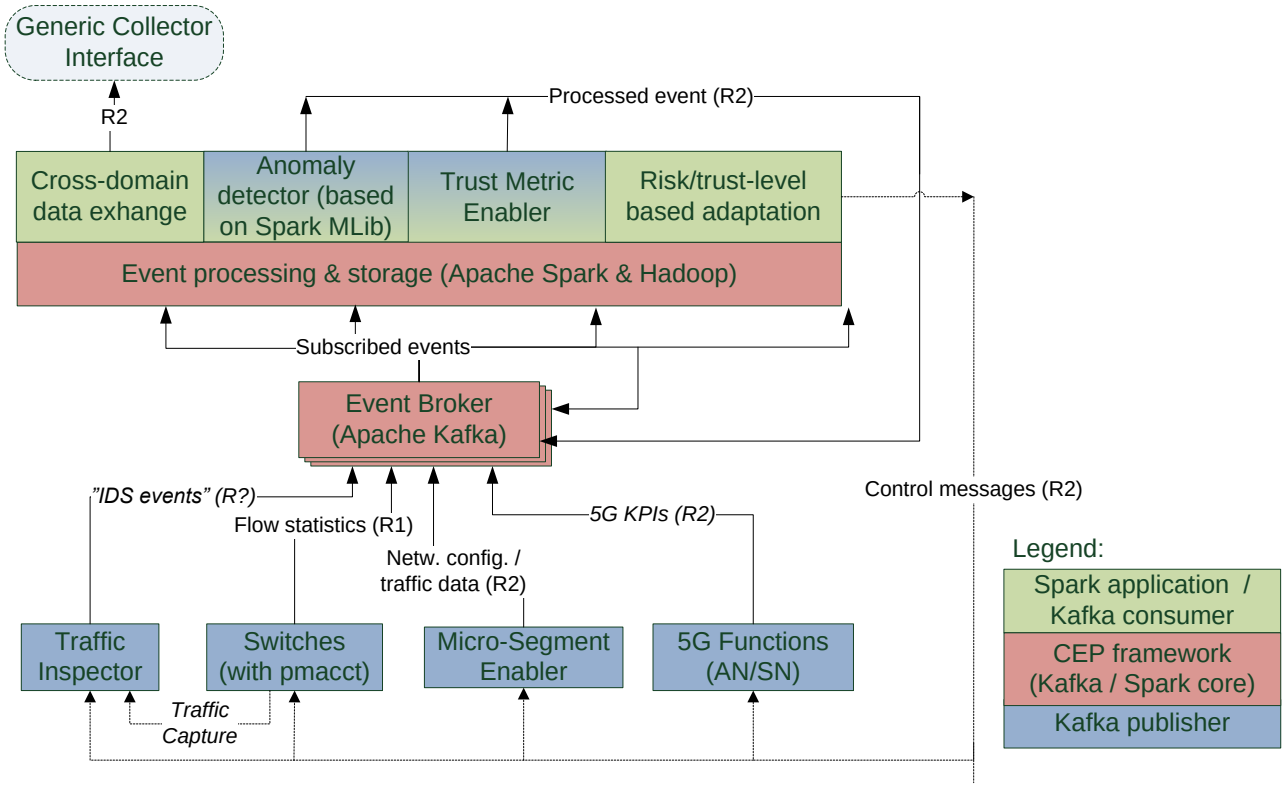


Figure 5-16: Architecture for security event distribution

The security monitor collects information from the micro-segment and (micro-segmented) 5G functions. In particular, we aim to support capturing of:

1) traffic statistics and micro-segment configuration, including e.g. topology, from Micro-Segmentation Enabler (which is essentially an SDN controller application).
2) KPIs from 5G functions with access network (eNodeB) and within serving networks (EPC).

Event information is distributed to event subscribers using the event broker. The broker can carry information on different events and from various sources. New event publishers can be added to the framework according to the availability of event information as well as the need of event subscribers. Existing data sources - log files, traffic analysers – need to be adapted to publish events using the brokering protocol.

Event subscribers provide micro-segment specific security inference, including anomaly detection and pattern detection. The subscribers may utilize common engine for processing the event streams. The subscribers may act also in publishers' role: processed events (inferred, anomaly, or combined events) from one publisher can be brokered to other subscribers. Subscribers at the end of the chain are then expected

to infer knowledge on micro-segment's security situation e.g. provide Trust Metric Enabler or initiate some security responses. Existing security inferencing components may be integrated to the system by providing an adapter that transforms brokered events into the format understood by the component.

The framework is targeted for real-time or near-real-time processing of monitored data streams. However, in security inference, also history data is needed - e.g. short term history for correlation analysis and longer history for machine learning. Smaller amounts of history data can be stored by the broker and for longer term history data the inference components must be integrated with a database for storing relevant events.

Brokers may be clustered and chained to facilitate scalability and to support cross-domain cooperation. Event information from a broker in one micro-segment may be subscribed and delivered to other brokers in order to further distribute decision making (to enable further scalability) and e.g. to enable micro-segments to share information and detect cross-segment attacks.

The system aims to make micro-segments and monitoring solutions more self-adaptive. Components that implement security response actions may, after being triggered by inferred event, request functions in the micro-segment (e.g. the micro-segment enabler, network controller, or switches) to change their behaviour. For instance, a micro-segment may remove some nodes from its topology or remove access permissions from a particular end-user. The brokering architecture itself enables event subscribers to adapt by subscribing more or less event streams. Additionally, there could be control mechanisms enabling event subscribers to request event publishers to intensify monitoring i.e. to publish more event information.

5.3.9 Architectural drivers

5.3.9.1 High-level functional requirements

Increasing coverage of monitoring by connecting various correlating event sources - Adapting different event streams so that event information from these streams can be published through the common brokering mechanism. The more event sources (publishing events) there are, the richer security monitoring applications can be build. In particularly, the larger amount of event sources enables more accurate correlation analyses.

Providing flexibility to add (easily and dynamically) application specific security inferencing components e.g. pattern or anomaly detectors tailored for particular threats. Building a solution that can be easily customized for different micro-segments that are dynamic and may have different sizes and types of communication.

5.3.9.2 Link to Security Architecture

The security monitoring enabler can be mapped to the domain and strata models of the architecture as described in the following figure.

Monitoring analytics (inferencing functionality) can be considered to locate in Management Domain (in the figure yellow dot). The monitoring enabler collects data from various domains belonging to single administrative domain (red dots). Essentially, information is collected from Slice Domains, as well as optionally from Access Network and Serving Network domains. The enabler protects slice domains (greed dots) by controlling micro-segment enablers defences. Monitoring information can be shared to external network domains. Hence, the enabler can also help to protect 3rd party and internet service provider domains (green dots).

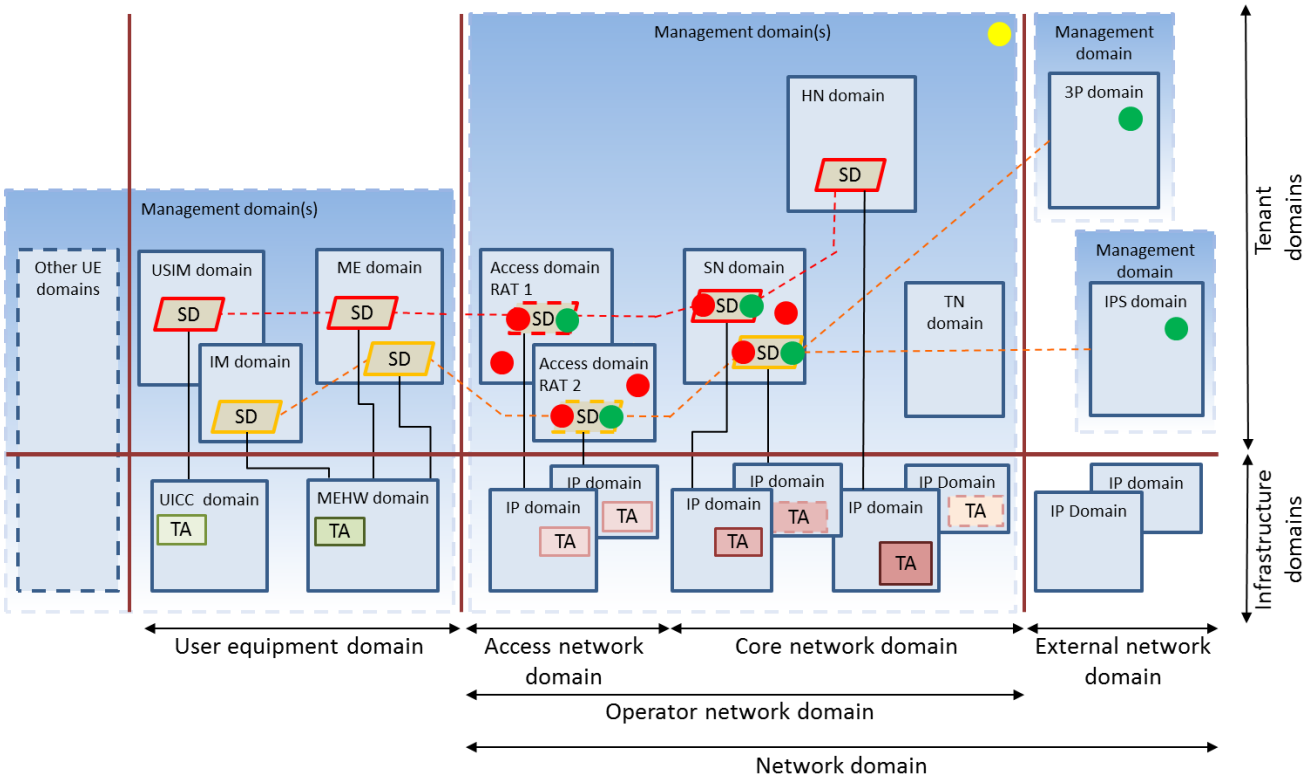


Figure 5-17: Enabler relevant domains

The enabler provides information analytics and information sharing mechanism for the *Management Stratum (yellow dot)*. Additionally, the enabler is related to serving and access stratum as it monitors those strata. The enabler is also related to serving and access stratum as it can protect them by adjusting microsegments in them (green dots). Similarly, the monitoring can be customized for applications and hence the enabler monitors and protects also application stratum (red and green dots).

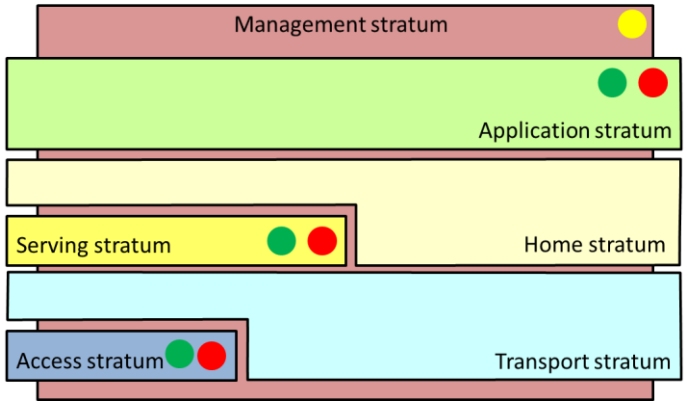


Figure 5-18: Enabler relevant strata

The security monitoring enablers links to the following security objectives listed in Deliverable 2.4 - Security Architecture Draft:

- 5G security must be dynamically scalable in order to easily and securely enable the changes required to ensure any new use cases, new trust models and new service delivery approaches (Objective O4.2). - *Essentially, we are trying to address scalability of monitoring.*
- 5G systems must support security monitoring capable of detecting advanced cyber security threats and support coordinated monitoring between different domains and systems (e.g. mobile and satellite) (Objective O7.3). - *The enabler is a monitoring enabler that also addresses cross-domain issues.*

The enabler can be mapped to the domain, strata and security features of the architecture in the following way:

- **Domain** - analytics (inferencing functionality) can be considered to locate in M*anagement Domain*. However, the monitoring enabler collects data from various domains belonging to single administrative domain. Essentially, information is collected from S*lice Domains*, as well as optionally from *Access Network, Serving Network and Transport Network domains*.
- **Strata** - the enabler provides information sharing mechanism for the *Management Stratum*.
- **Security Feature Group** - The monitoring enabler can be classified to *Security Management* Feature Group.

5.3.9.3 Quality attributes

The following list presents quality attributes are relevant when assessing the realizations of the enabler. The list also illustrates how the enabler plans to support these attributes:

- Functional suitability – accuracy of detecting security incidents is increased by focusing homogeneous micro-segment which enables analysis to use more (correlated) parameters
- Reliability – use of mature big data software technologies
- Operability – operability is eased by use of widely adopted and accepted big data software technologies; operability in different setting is enabled with customization (each micro-segment may be provided own monitoring functions)
- Performance efficiency – scalability to handle large amount of real-time events supported by micro-segmentation concept and selected technologies
    - Horizontal scalability – the system utilizes event brokering (publish-and-subscribe architecture) to enable new processing nodes and information sources to be added.
    - Vertical scalability – the system should be scalable to micro-segments of different sizes and to *near-realtime* processing of event streams for different inference applications. Vertical scalability (adding more resources for single node) is enabled by software technology selection (i.e. by using big data components supporting parallel and clustered computation as well as cloud-based deployments).
- Security – Authentication, reliability and confidentiality of event streams – Event information coming from some sources may be unreliable, untrustworthy, or malicious. Consequently, the broker should authenticate the event sources. Trustworthiness and quality of the event data may also be considered when inferring knowledge and determining probabilities of security incident. In cases where information is shared between different actors also confidentiality and access control policies may be required to determine who can access event streams.
- Compatibility – support for various standardized event sources / monitoring interfaces and use of open interfaces for output format (supporting “natively” open JSON-based specification and exporting data using GCI).

- Maintainability – use of easily utilized programming languages and widely utilized big data technologies; extendability and modularity is enabled by brokered (publish-subscribe) architecture
- Effectiveness – accuracy of detecting security incidents increases as the enabler can focus monitoring more homogenous micro-segments
- Efficiency – capability to cover larger amount of security threats/attacks is similarly increased as the volume of traffic is smaller and more homogeneous in micro-segments (when compared to heterogeneous non-micro-segmented traffic consisting different types of content streams including voice, web, IoT).

5.3.9.4 Technical constraints

To achieve interoperability with legacy event sources (logs etc.), monitored data must be transformed into event information that is understood by the inference components. We do adaptation and parsing in two phases. First the event sources (e.g. switches, SDN controller, traffic inspectors, or log collectors) encapsulate monitored data into events which can be published through the broker. Second, the inference components transform the content of events to extract needed information and derive knowledge.

5.3.9.5 Business constraints

The open framework is based on proven and widely adopted open protocols and open source software, which are scalable for large data amounts. Openness and scalability increases acceptability of the solution and eases integration as there are more directly compatible implementations and examples available.

5.3.10 Detailed specifications

5.3.10.1 Introduction

This section specifies the APIs and protocols utilized by the enabler.

5.3.10.2 Conformance

For an implementation to be conformant it should implement specifications as here stated.

5.3.10.3 API specifications

5.3.10.3.1 Functions, messages and commands

5.3.10.3.1.1 Event distribution layer

Events are presented as {“topic”: “content”} tuples. The structure of the content depends on the event publisher. The primary format for content is JSON. In the following subsection, we will specify some essential content formats.

The naming of “topics” follows the following format: [<microsegment_identifier>.<event_identifier>.<optional event parameters>], where event identifiers should be unique for the micro-segment.

5.3.10.3.1.2 Micro-segment events

SDN related configuration and network traffic information can be collected from Micro-Segmentation Enabler. The monitoring enabler supports data as specified by Micro-segmentation Enabler. The monitoring probe in Micro-Segmentation Enabler will publish the information in JSON format.

Micro-Segmentation Enabler publishes meta-data on the segment, particularly topology and node configuration data.[11] Broker topics for published metadata events are the following: [<micro-segment identifier>.<information_type>] where information types include e.g. “controller configuration”, “topology”, and (for switch specific configuration) <switch_id>.”config”. Each time a micro-segment changes, the relevant event / configuration information is published. These change events are triggered e.g. by “create”, “modify topology”, and “modify settings” commands, which are made by micro-segment subscriber. Information requests may also trigger additional information to be published.

topic: [<micro-segment identifier>.<switch_id>.config]
{ "mfr_desc": "manufacturer", "hw_desc": "hardware description", "sw_desc": "software description", "serial_num": "serial number", "desc": "Human readable description” }

Network statistics includes data on how much traffic has flown trough particular paths in during the monitoring period (since last statistic event was published).

topic: [<micro-segment identifier>.flowstats]
{“src”: “ip_address”, “dst”: “ip_address”, “packet” : “count”, “bytes” : “bytes transferred”}

Information on particular network events may be published when they occur. For instance, in intensified monitoring scenarios, it is possible to publish event information on every new TCP session that is detected in particular flow (between specific ip addresses).

topic: [<micro-segment identifier>.<flow_id>.tcp_syn]

{
“src” : “ip_address”,
“dst” : “ip_address”,
“packet” : “tcp_syn”
}

5.3.10.3.1.3 5G events

The monitoring enabler may also collect information (KPIs, event notifications) from 5G functions within access network (essentially base station / eNodeB) or from the serving network (EPC).

[11] Here we specify only some essential events. Micro-Segmentation Enabler is based on OpenVirteX [openvirtex_API] and Ryu SDN framework [Ryu_API] and monitoring data available from those interfaces can be made available through Micro-Segmentation Enabler.

Broker topics for published events are named using event source identifier and the method / message: [<micro-segment identifier>.>5G_function_id>]. The name of the KPI or event as well as KPI value/event data is given in JSON values.

topic: [<micro-segment identifier>.<5G_function_id>]]

{
“KPI” : ”kpi/event name1”
“value” : “value1”
}

5.3.10.3.1.4 Processed events

Inferred, anomaly and combined events – the ‘output’ from inference components – are published using the event broker. The events are distributed as topic-content pairs (see event distribution layer format). The content structured using JSON. The exact format for content of processed event depends on the component publishing the event.

In the release 2, at least an anomaly detector component will be implemented. It will publish the feature vectors that indicate the anomaly as well as a value indicating certainty of anomaly (derived from vectors distance from previously learned cluster centerpoints).

topic: [<micro-segment identifier>. <detector_id>.anomaly]

{
"feature_vectors" : "name1,name2…",
"feature_values" : "value1,value2,… ",
"certainty" : "severity estimation"
}

5.3.10.3.1.5 Data export

Event information can be exported to other domains using two alternative approaches.

1. Native brokered publish-subscribe model - Other domains may be given access to event streams in the broker (they may be given access permissions to subscribe particular topics from - micro-segment specific - broker or the monitor publish events for an external broker).
2. Generic Collector Interfaces (GCI) - Data can be exported using GCI client-server model. The micro-segment monitor will implement GCI client for exporting data. The exported data follows the GCI definitions.

5.3.10.3.2 Signalling

All real-time signalling between monitoring components occurs through the broker interface described in the previous section.

Signalling between monitoring interference components and external applications / end-users, is inference component specific. Such signalling will not be implemented for the release 2.

5.3.10.4 Examples

Each time the topology has been changed a list of all switches in the topology and configuration of added / removed switch will be published through a broker. The published event information is available from the micro-segmentation enabler from where it can be collected and published though the broker. For example:

topic: [<micro-segment identifier>.switch_id]
{ "mfr_desc": "Manufacturer, Inc.", "hw_desc": "Open vSwitch", "sw_desc": "1.2.3", "serial_num": "123456789", "desc": "Human readable description" }

Flow statistic events, collected from switches, can follow published using a topic [<micro-segment_identifier>.<switch_id>.<if_id>.<params>]. Collected data can contain e.g. statistics on flows between particular destinations, like:

topic: [<micro-segment identifier>.flowstats]
{"src": "195.144.1.22", "dst": "10.2.15.254", "packet" : "234", "bytes" : "23452"}

Monitored information is processed in Spark-based components, which subscribe events through Kafka broker (see [Spark_streaming_guide] and [Spark&Kafka_guide]). The *simplified* Python examples below illustrate some essential functions that may be useful for monitoring:

1) attaching Kafka events to spark's stream abstraction (called DStream) using Spark's KafkaUtils:

```
kafkaStream = KafkaUtils.createDirectStream(ssc, [topic],
{"metadata.broker.list": brokers})
```

2) transforming incoming json events to vector stream:

```
parsedStream =
kafkaStream.map(lambda(key,value):json.loads(value)).map(parsejson)
```

3) learning a model of normal behaviour (using Spark MLlib's Streaming-K-means clustering algorithm) and then finding distance of new data point to the closest cluster (a long distance indicating an anomaly):

```
model = StreamingKMeans(k=2,decayFactor=1.0).setRandomCenters(2, 1.0, 0)
model.trainOn(parsedStream)
result = model.predictOn(parsedStream)
```

4) publishing anomaly event using Kafka producer API [Kafka_API]:

```
Producer<String, String> producer = new KafkaProducer<>(properties);
producer.send(new ProducerRecord<String,String>
("anomaly_detector.anomaly1","anomaly", "eventdetails");
producer.close();
```

5.3.11 Re-utilised Technologies/Specifications

The reference implementation of the framework will be based on Apache Kafka [Kafka] for event distribution. The event distribution (brokering) follows Apache Kafka API specifications, which are described at [Kafka_API]. The "topic" can be directly mapped to Kafka topic.

Apache Spark [Spark] is used for event processing. For short-term history data, Kafka's capability to store short term history data may be utilized. Hadoop which integrates with Spark may be used as a database system to store longer term history data. Anomaly detection is based on streaming-k-means implementation [Fre15] from Spark MLlib.

Existing security monitoring software components - SIEM, IDS, log analysers, traffic analysers (like Snort), anomaly detectors etc. - may be integrated to the framework.

5.3.12 References

[JSON] The JSON specification. http://www.json.org/

[Kafka] Apache Kafka project. http://kafka.apache.org/

[Kafka_API] http://kafka.apache.org/documentation.html#api

[RFC5424] R. Gerhards. The Syslog Protocol. IETF specification. 2009. https://tools.ietf.org/html/rfc5424

[Spark] Apache Spark project. http://spark.apache.org/

[Spark_streaming_guide] Spark Streaming Programming Guide.
http://spark.apache.org/docs/latest/streaming-programming-guide.html

[Spark&Kafka_guide] Spark Streaming + Kafka Integration Guide.
http://spark.apache.org/docs/latest/streaming-kafka-integration.html

[Spark_streaming-k-means] Spark Docs: Clustering - RDD-based API. Streaming-k-means.
http://spark.apache.org/docs/latest/mllib-clustering.html#streaming-k-means

[openvirtex_API] OpenVirteX API documentation. http://ovx.onlab.us/documentation/api/

[Ryu_API] Ryu API Reference. http://ryu.readthedocs.io/en/latest/api_ref.html;
https://media.readthedocs.org/pdf/ryu/latest/ryu.pdf

[MacQ67] MacQueen, James. "Some methods for classification and analysis of multivariate observations." *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. No. 14. 1967.

[Llo82] Lloyd, Stuart. "Least squares quantization in PCM." IEEE transactions on information theory 28.2 (1982): 129-137.

[Ail09] Ailon, N., Jaiswal, R., & Monteleoni, C. Streaming k-means approximation. In Advances in Neural Information Processing Systems, 2009.

[Fre15] Freeman, Jeremy. Introducing streaming k-means in Apache Spark 1.2. Databricks, Engineering Blog, 2015. https://databricks.com/blog/2015/01/28/introducing-streaming-k-means-in-spark-1-2.html

[MulVAL] Multi-host, Multi-stage Vulnerability Analysis Language.
It is a research tool for security practitioners and system administrators to better manage the configuration of an enterprise network such that the security risks are appropriately controlled.
http://people.cs.ksu.edu/~xou/mulval/

5.4 Satellite Network Monitoring Open specifications

5.4.1 Preface

This section describes the open specification of the "Satellite Network Monitoring" enabler which focuses on providing pseudo real-time monitoring of logs and alarms on integrated satellite and terrestrial networks and threat detection in these systems.

The aim of this security monitoring enabler is to protect against internal and external threats coming from the heterogeneous 5G networks, to meet security requirements from the 5G-ENSURE trust model.

5.4.2 Status

"Satellite Network Monitoring" enabler features are in various states of research at this moment and there is no planned software delivery for all the features. Nevertheless, some client-side features are currently being developed in a R1 prototype as a preliminary open specification and as a proof of concept.

5.4.3 Copyright

5.4.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

5.4.5 Terms and definitions

This section comprises a summary of terms and definitions used during the later sections.

eNB (E-UTRAN Node B, Evolved Node B or eNodeB)	Hardware connected to the mobile network that communicates directly wirelessly with UEs
HAPS (High Altitude Pseudo-satellites)	Long-endurance aerostatic or aerodyne platforms in the lower stratosphere, above commercial aviation airspace

Indicators	Information periodically collected from the network components (hardware status, alarms…) and counters from the specific business logic (transfer rate, number of requests…). These indicators can be classified in three categories:
	• Health status:
	  ○ Intrusion detection
	  ○ Alarms scanned by network devices
	  ○ Excessive load
	• Configuration state:
	  ○ Network status (e.g. logged status, SW version…)
	  ○ Network configuration
	  ○ Credential status
	• Counters (performance indicators):
	  ○ Volume counters
	  ○ Efficiency counters

SatNO (Satellite Network Operator)	Owns and is responsible for maintaining, managing, deploying and operating the Satellite Network, i.e. leasing satellite transponders and providing the associated ground segment equipment

Smart antenna	eNB antenna with self-pointing capabilities

5.4.6 Overview

The introduction of Broadband services through the satellite is increasing significantly over the past years and is supposed to continue in that direction over the years to come. The "Satellite Network Monitoring" enabler is related to broadband telecommunication systems or telecommunication ground user segments, but may also relate to other systems.

Such as a security monitoring enabler is important because there are several 5G use cases that can only be served by satellites or for which satellites provide a more efficient solution.

Current Satellite and Terrestrial communication networks can be complemented by new HAPS or drone based services in the short future.

5.4.6.1 Provided inR1 (Reminder)

The enabler provides a security monitoring solution suitable for an environment of integrated satellite and terrestrial networks. It consists in:

- A client-side feature, able to collect indicators from the network component and send them to the server-side feature.
- A server-side feature, able to configure the client-side features and provide pseudo real-time monitoring of consumed indicators.

The "Satellite Network Monitoring" enabler consists of a client-side feature deployed on each network component. This client-side feature able to collect indicators from the network component and

Network components that are subject to security analysis will be identified.

5.4.6.2 Targeted in R2

The following functionalities will be described and analysed in detail in R2:

- Determine security metrics, counter measures and the mitigation level they provide.
- Server-side security analysis to detect attacks and malicious behavior: use of data analytics to response to the identified threats (e.g. notify the operator, network topology reconfiguration, …). The security level shall be configurable. Some of the threats currently identified are:
    - Attack on network components: RF interference, power or communications lines…
    - Attack on the SNM: intruding the system by hijacking, blackmailing, placing or impersonating the operator, to obtain credentials or/and gain control of the system, …
    - Denial of service: flood the network with dummy indicators to make the network unusable, preventing any useful communications with the network management system.
- Network components topology management. 4G backhaul networks are fixed topologies; therefore, the network barely manages accidental/deliberate congestion or link failures. In addition to the security monitoring focus, this enabler will research a dynamic solution, based on satellite links and smart antennas, enabling topology reconfiguration according to traffic demands (ultra-reliable Use Case 8.1 from D2.1).

This security monitoring enabler should improve the security of operators/users, while maintaining or increasing the level of productivity. One of the challenges is the definition of the KPIs that demonstrate such improvements.

5.4.7 Basic concepts

The infrastructure for building the Satellite Access & Transport Networks comprises the following components (see **Figure 5-19**):

- Satellite Hub: satellite earth station connected to the 5G network.
- eNBs and Satellite-capable eNBs: (traditional eNB improved with a satellite link).
- Different UEs:
  - Satellite Terminals (Ka band): satellite terminal with a Ka band antenna.
  - Satellite Modems: end-user satellite terminal connected to a satellite antenna using a communications satellite as a relay.
- 5G devices.

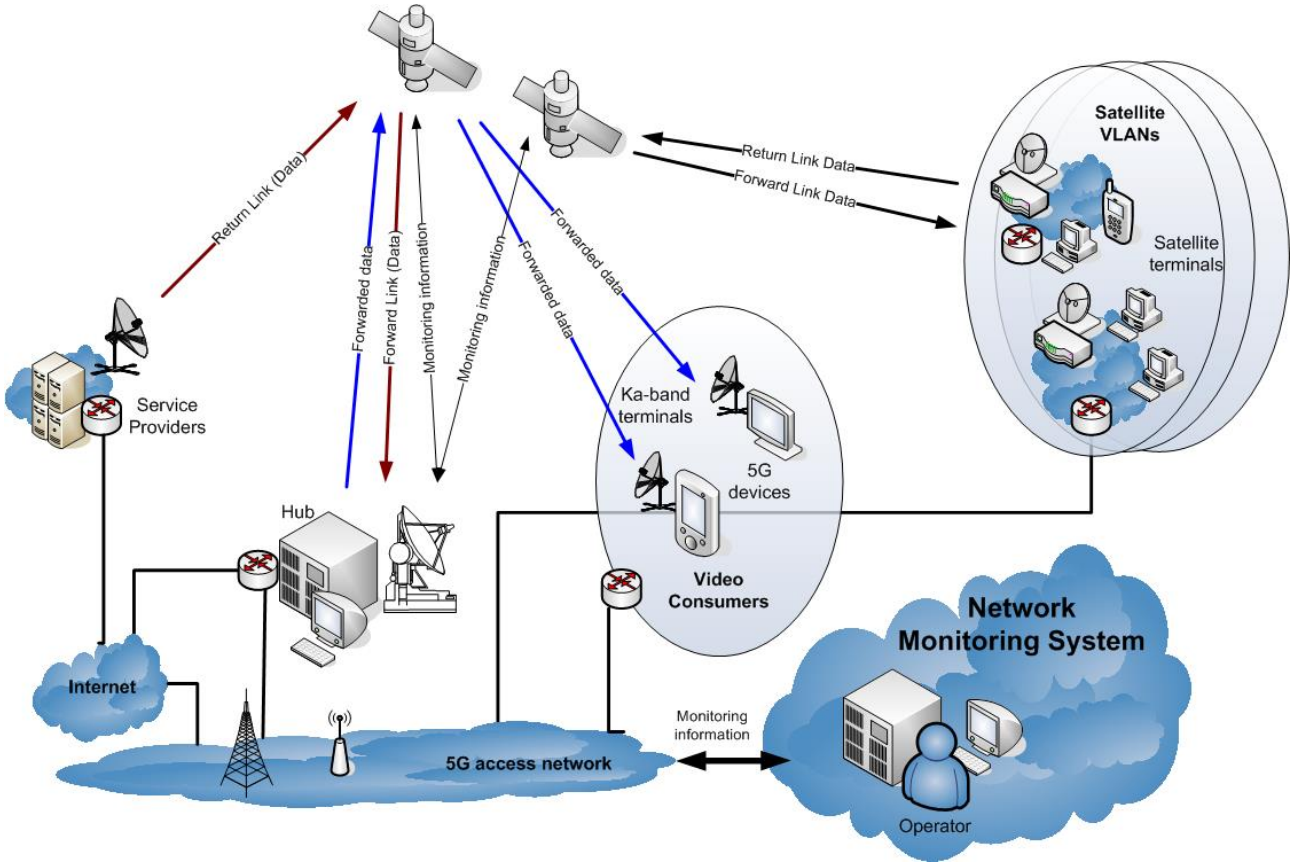


Figure 5-19: Satellite Access & Transport Network

These components are distributed in a wide-area. Satellite support ensures high network availability and service reliability with a 100% geographic coverage.

5.4.8 Main interactions

5.4.8.1 Use cases

This enabler covers the use case 5.6 “Integrated Satellite and Terrestrial Systems Monitor” and use case 8.1 “Satellite-Capable eNB” defined in D2.1.

Actors in the use cases are the following:

- Network components: satellite terminals, Hub, eNBs…
- Satellite Network operator.

5.4.8.1.1 Collect indicators (client-side feature)

The client-side feature periodically collects security and status indicators from the network components and from the specific business logic and produces a message.

5.4.8.1.2 Network Topology Management (server-side feature)

The server-side feature monitors the indicators collected. In the event of a link failure or congestion, the topology algorithm produces the optimal topology and reconfigure the network components. This use case is provided as an example of data analytics and server-side response to security events in the network.

This use case focuses on evolving the Transport Network Architecture (TNA) by combining both satellite and terrestrial transport architectures. The main goal is the ability to offer resilience to cases of link failure and to provide offloading capability via satellite to the backhaul network in case of congestion.

5.4.8.2 Components and interaction overview

The "Satellite Network Monitoring" enabler consists of two main features:

- Security Network Agent (SNA): periodically collects indicators from the network components and from the specific business logic. The "Generic Collector Interface" enabler will be analysed in order to be used as part of this building block.
- Security Network Manager (SNM): consumes such indicators to monitor the system status (e.g. fault management, performance monitoring) and is in charge of carrying out security analysis to detect attacks and malicious behaviour.

B/OSS (Business and Operational Support Systems) is an external system that receives such indicators and is in charge of service provisioning, network configuration and billing.

5.4.8.2.1 Collect indicators (client-side feature)

NOTE: The network component shall be registered in the server and their credentials shall be periodically updated. The "Fine-grained authorization" enabler will be analysed in order to be used as part of this building block.

1. The SatNO configure the indicators to be collected in the network component.
2. The SNA collects the configured indicators.
3. The SNA produces a message with this information.

5.4.8.2.2 Network Topology Failure / Congestion

1. The SNM detects a link failure / congestion.
2. The Network Topology model is updated.
3. The Topology algorithm produces the optimal topology to guarantee QoS, especially in ultra-reliable services:
    a. Some dynamic beans may be switch on.
    b. Some satellite links may be switch on with the EPC.
4. The new topology is forwarded to the network components.

5.4.8.2.3 Security analysis (R2 server-side feature)

1. SNM performs security analysis on the indicators received:
    a. to detect attacks
    b. to detect malicious behaviour

2. SNM automatically responds based on the configured counter measures or require a response from the SatNO.

5.4.9 Architectural drivers

The enabler shall provide a method for flexible definition of the access control policies.

5.4.9.1 High-Level functional requirements

The high-level functional requirements are:

- The SNA shall be identified
- The SNA shall be registered in the SNM
- The SNA credentials shall be periodically updated
- The SNM shall configure the indicators to be collected by the SNA
- The SNA shall collect the configured indicators
- The SNA shall send the collected indicators to the configured SNM
- The SNM shall consume the indicators
- The indicators shall be used in the SNM to monitor the component status
- Network components subject to security analysis shall be identified

5.4.9.2 Link to Security Architecture

The satellite network monitoring enabler is oriented to accomplish the following 5G security goals:

- O2.4 5G must enable seamless interworking of different network technologies, mobile, fixed as well as satellite [bt] without exposing the security level of each of these technologies to new threats.
- O7.3 5G systems must support security monitoring capable of detecting advanced cyber security threats and support coordinated monitoring between different domains and systems (e.g. mobile and satellite)

Next figures show the relevant domains and strata (marked by a yellow dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016].

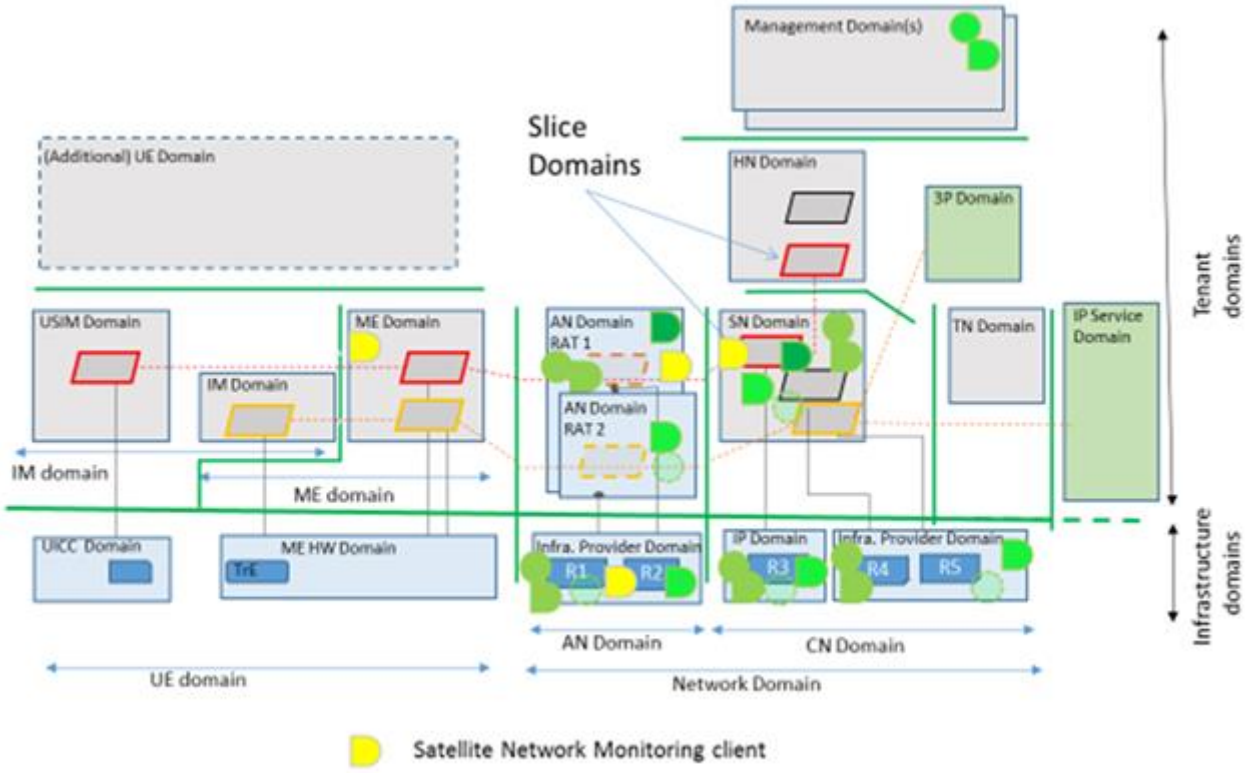


Figure 5-20 Satellite Network Monitoring Mapping to 5G Domains

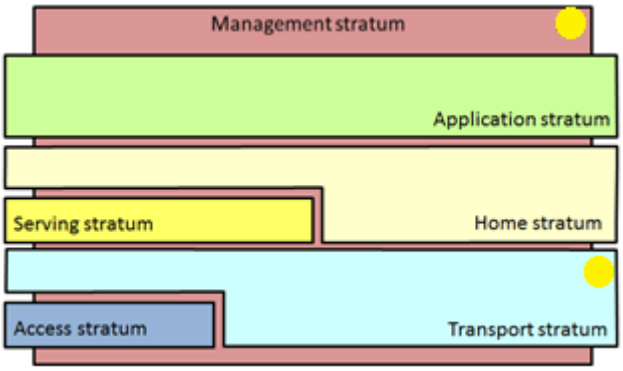


Figure 5-21 Satellite Network Monitoring Mapping to 5G Strata

5.4.9.3 Quality attributes

Compliance with standards: JAX-RS and JMS /MQTT / Kafka Wire.

Horizontal scalability: the enabler utilizes a message broker (observer pattern) to enable new collecting agents and consumer managers to be added.

Performance in terms of high-throughput messages.

5.4.9.4 Technical constraints

Main features are scheduled for R2 and are currently in a research phase, thus there are no main technical constraints at this stage.

The amount of data that needs to be analysed at pseudo real-time may be large and heterogeneous. Two complementary approaches will be analysed:

- Partitioning the network into virtual private networks might be an efficient solution, so that each segment is managed separately and appropriate solutions are tailored to each partition.
- Use a scalable and high-throughput distributed messaging system (e.g. Apache Kafka).

Nevertheless, some client-side features (see section **Error! Reference source not found.**) are currently being developed in a R1 prototype as a preliminary open specification:

- The client device needs to be capable of running an application server compliant with JAX-RS.
- The message broker shall be compliant with JMS / MQTT / Kafka Wire

5.4.9.5 Business constraints

No known business constraint exists.

5.4.10 Detailed specifications

5.4.10.1 Introduction

This specification defines the "Satellite Security" API, which provides pseudo real-time security monitoring on integrated satellite and terrestrial networks.

The client-side API follows the RESTful and messaging system design principles.

5.4.10.2 Conformance

An implementation that conforms to this open specification shall implement fully the architecture described.

All the interfaces described are mandatory and must be implemented in order to be compliant with.

5.4.10.3 API specifications

"Satellite Network Monitoring" enabler is under research. Client-side features have been specified in R1 and updated/ reminded here. The "Satellite Network Monitoring" enabler server-side technologies/specifications in scope of R2 are detailed hereafter.

Below is an update of API specification of the enabler encompassing features in scope of R2

5.4.10.3.1 Indicators configuration

Configure the indicators to be collected in the network element

Request:

- Method: POST
- URI: /sna/resource/indicators
- Content-type: application/json
- Body: Indicators to be collected [indicatorType]

Response:

- HTTP status code:
  - 201 when success
  - 400 when error

- Body: None

5.4.10.3.2 Network configuration

Configure the network element

Request:

- Method: POST
- URI: /sna/resource/topology
- Content-type: application/json
- Body: Topology to be used [{interfaceName,interfaceStatus:[on/off]}]

Response:

- HTTP status code:
  - 201 when success
  - 400 when error
- Body: None

5.4.10.3.3 Produce a message

Send an indicator to the message broker

- Method: JMS/ MQTT / Kafka Wire
- topic: “resource-indicators”
- Content-type: application/json
- Body: Collected indicator {indicatorType,[indicatorKey,indicatorValue]}

5.4.11 Re-utilised Technologies/Specifications

The “Satellite Network Monitoring” enabler client-side is based on RESTful and messaging system design principles. The technologies and specifications used in this enabler are:

- HTTP/1.1
- Java API for RESTful Web Services - JAX-RS 2.0
- JSON and/or XML data serialization formats
- Apache Active MQ/Kafka as message broker (JMS 1.1 / MQTT 3.1 / Kafka Wire)
- JIRA issue tracking product
- SVN software versioning and revision control system
- Git software version control system
- Melody Advance system engineering modelling tool
- Thales Control continuous integration tool based on Jenkins and Sonar

5.4.12 References

[D2.1] 5G-Ensure Consortium, Deliverable 2.1 Use Cases. Available online at http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[D3.1] 5G-Ensure Consortium, Deliverable 3.1 5G-PPP security enablers technical roadmap. Available online at http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf.

[TR 101 984]Standard: ETSI - TR 101 984 “SatelliteEarth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Services and Architectures”

[TR 22.891] 3GPP TR 22.891, “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Feasibility Study on New Services and Markets Technology Enablers; Stage 1 (Release 14)”, v1.0.0, September 2015.

[5G Vision] 5G vision: the next generation of communication networks and services, The 5G Infrastructure Public Private Partnership, February 2015, www.5g-ppp.eu.

[NGMN WP] "NGMN 5G WHITE PAPER", a Deliverable by NGMN Alliance, 17th of February 2015.

[SatCOM] "The role of satellites in 5G", a white paper from the NetWorld2020’s – SatCom WG, published 31st July 2014.

5.5 System Security State Repository

5.5.1 Preface

This enabler consumes monitoring events from the Generic Collector Interface enabler to provide security information about a runtime system. It uses the same technologies and models as the Trust Builder enabler described above.

5.5.2 Status

The software is currently being developed.

5.5.3 Copyright

5.5.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

5.5.5 Terms and definitions

Core Model – the core ontology, defining common vocabulary and relationships used in all higher-level models.

Generic Model – an ontology defining the typology of Assets, Threats and Controls (security measures) for a given domain (e.g. 5G networks). In release R1, this will cover only Assets and some types of Controls.

Design-Time System Model – an abstract model of a particular system, described in terms of the relationships between system specific Asset classes. The design time model can then be enriched by specifying which security Controls should be used to protect each type of system asset, and auto-generating a set of system-specific Threat classes describing potential threats to the system.

5.5.6 Overview

The purpose of the System Security State Repository is to capture the state of an instantiated 5G network (or portion thereof) and compare it with a Design-Time System Model created in the Trust Builder enabler. The comparison reveals whether assets in the instantiated network conform to the Design-Time System Model in terms of missing controls and potential threats.

For release one, the technical roadmap specifies that the “deployment model ontology” will be created. This is equivalent to the development of the Generic Model and any updates to the Core Model already

described in Trust Builder sections **Error! Reference source not found.** and **Error! Reference source not found.** respectively.

For release two, the Secure System State Repository will compare system monitoring information with the Design-Time System Model, to identify real-time assets that are non-compliant with the Design-Time System Model.

5.5.7 Basic concepts

The Secure System State Repository will capture the state of the instantiated network and issue events about assets not compliant with the Design-Time System Model based on monitoring data. The Repository will also provide a visualisation of the current state of the system, specifically which run-time assets in each asset class specified in the Design-Time System Model are known to be present. The figure below represents an outline of the Repository architecture for R2.

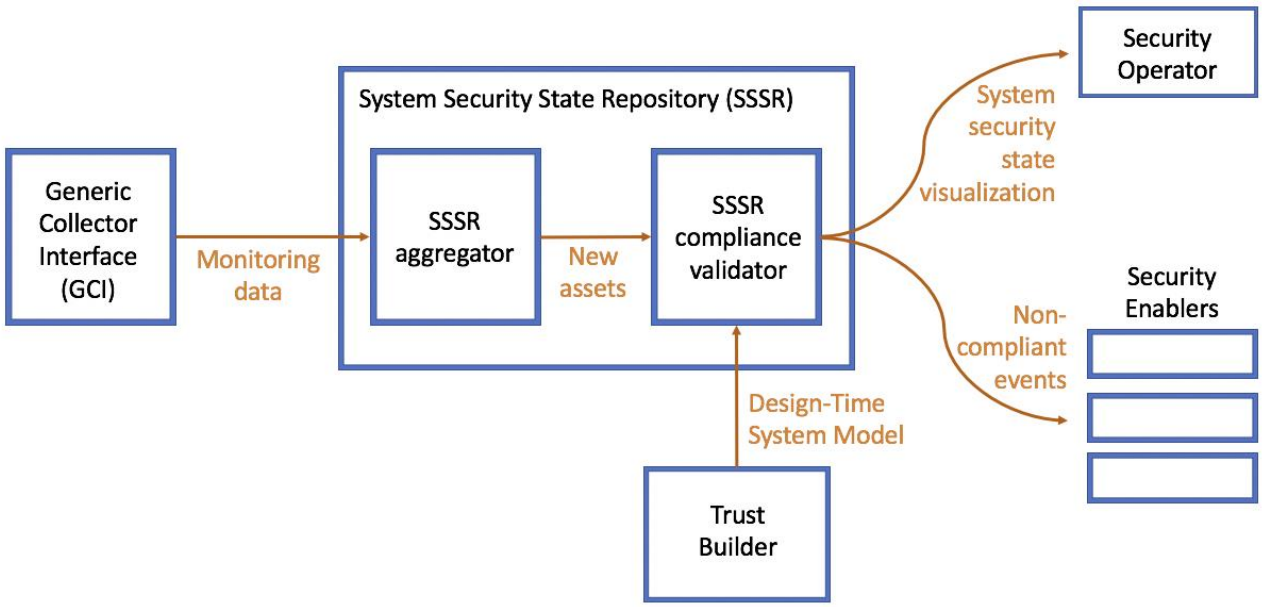


Figure 5-22 Basic concepts of the System Security State Repository

Main components of the architecture are described in sub sections below.

5.5.7.1 SSSR aggregator

The System Security State Repository (SSSR) aggregator component will perform the following functions:

1. Fetch monitoring reports from Generic Collector Interface (GCI).
2. Store and parse the reports extracting the following:
    a. Assets (new and updated)
    b. Asset classification with respect to the Design-Time System Model asset types
    c. Controls applied to assets
3. Submit parsed reports to the SSSR compliance validator component.

Depending on the information provided by the GCI, it may not be possible to determine all the security controls present for a given asset. Priority will be given to detecting the presence (or absence) of 5G-ENSURE security enablers in that case.

5.5.7.2 SSSR compliance validator

The System Security State Repository compliance validator component will compare assets submitted by the aggregator with the Design-Time System Model received from the Trust Builder. The comparison will determine if any assets lack security controls specified in the Design-Time System Model, and allow them to be flagged as non-compliant.

Information about non-compliant assets will be published as non-compliant asset events made available to other security enablers: GCI and PulSAR. GCI will receive feedback on published monitoring reports, allowing it to enrich reported data. Non-compliance events can be used by PulSAR as vulnerability data input.

Note that the SSSR will collect and use information about assets in different domains. However, it will not generate non-compliance alerts about assets in remote domains where there is insufficient monitoring data to determine whether or not those assets have the necessary security protections.

5.5.7.3 System state visualisation

The System Security State Repository will provide a dashboard-style visualisation of the current security state to allow Security Operator to monitor non-compliant assets and the system overall.

5.5.8 Main interactions

5.5.8.1 Use cases

New or updated asset event

SSSR aggregator polls GCI for new reports, receives a new report. The aggregator compares data in the report to the current saved state of the system and identifies a new or an updated asset on the system. The aggregator maps the new or updated asset to the Design-Time System Model representation and submits to the validator for validation. The validator compares the representation with the model and issues a non-compliance event in case of non-compliance. The state of the system is updated in the repository.

System state visualisation

Security Operator can view the latest state of the system in the dashboard-style UI. Overall view of the system and the view of just non-compliant assets will be available.

5.5.8.2 Components and interaction overview

The main SSSR components are outlined in the figure below:

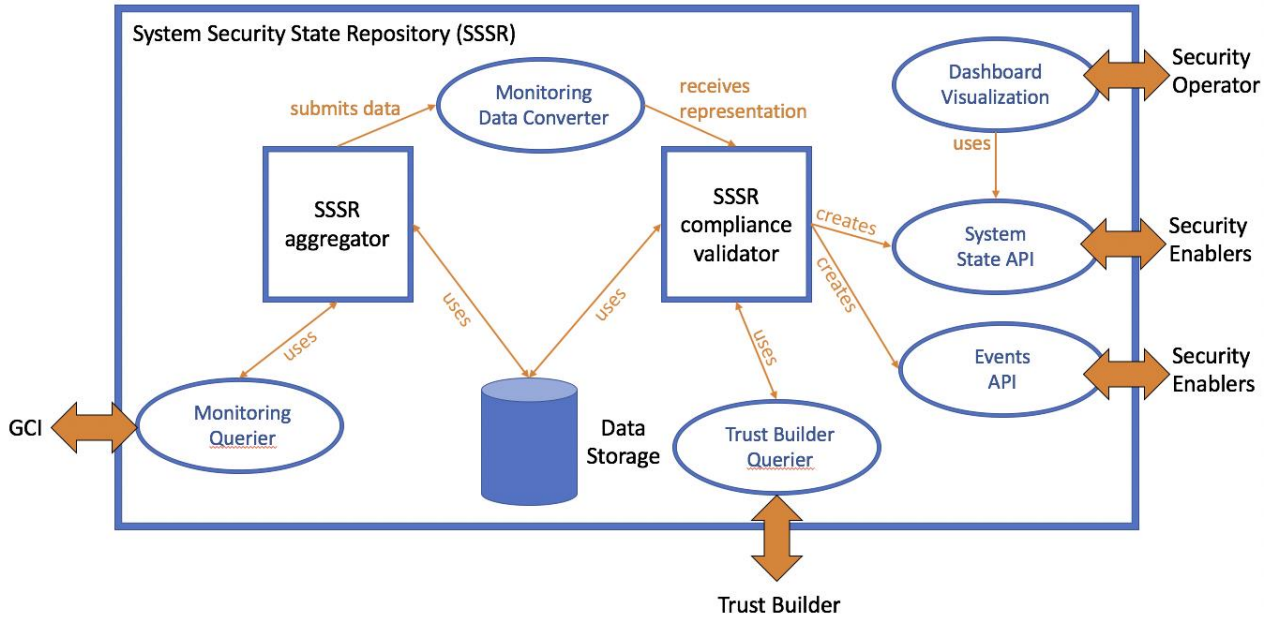


Figure 5-23 Main SSSR components and their interactions

SSSR Aggregator makes use of Monitoring Querier to receive monitoring data from GCI. Monitoring data is passed through Monitoring Data Converter to be compared to the Design-Time model in the validator. Current security state of the system is stored in the Data Storage. Validated assets are made available to the Security Operator and Security Enablers via the dashboard and events endpoints. Validator uses a special querier to communicate with the Trust Builder enabler.

5.5.9 Architectural drivers

5.5.9.1 High-Level functional requirements

The System Security State Repository will store the security state of the system and issue events about new or updated assets' non-compliance with the Domain System Model. Other security enablers can subscribe to those events. Security Operator can visually monitor the state and non-compliant assets.

5.5.9.2 Link to Security Architecture

The System Security State Repository links to the following security objectives listed in Deliverable 2.4 – Security Architecture Draft:

- **O2.3** 5G must provide solutions for security and privacy breaches identified in the previous mobile network generations such as the IMSI and IMEI unauthorized tracking or the denial of service provoked by the unsecured mobility messages (i.e., TAU messages).
- **O4.3** 5G infrastructure components should support necessary root-of-trust functionality
- **O7.3** 5G systems must support security monitoring capable of detecting advanced cyber security threats and support coordinated monitoring between different domains and systems (e.g. mobile and satellite)

Figures below show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [5G-ENSURE Architecture, 20170328].

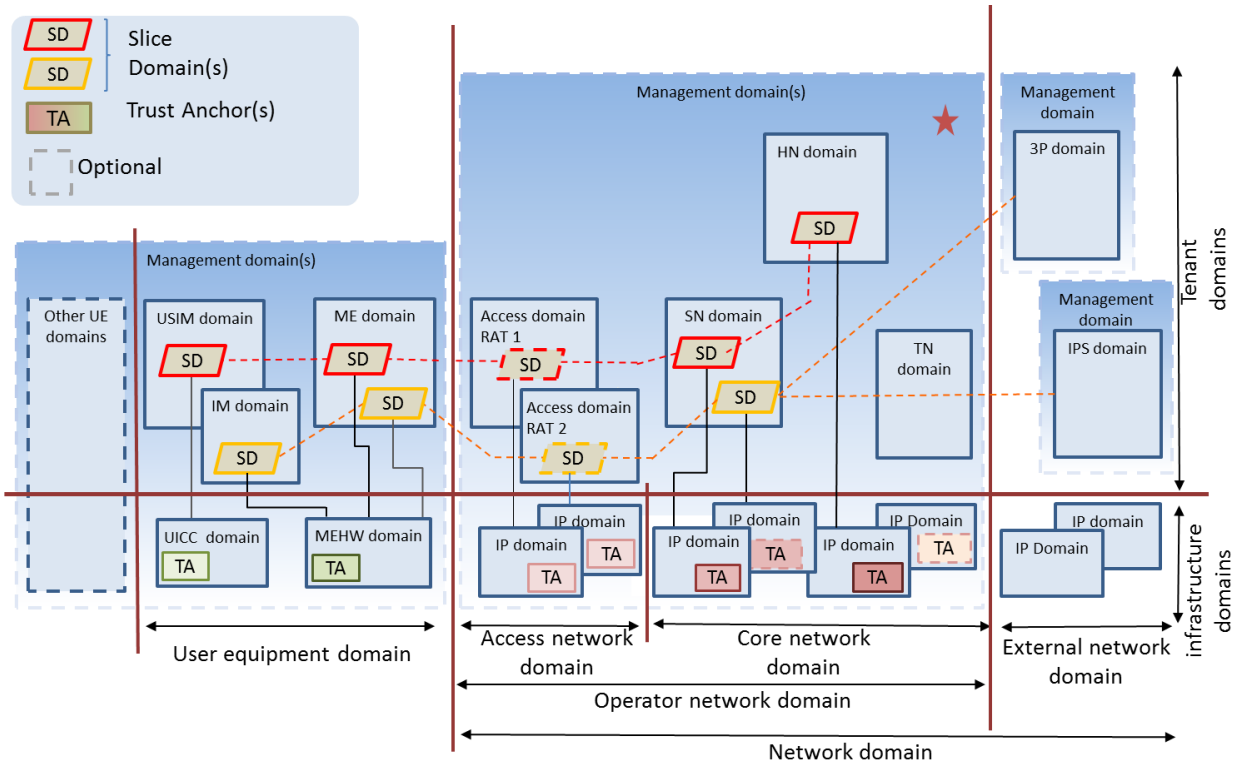


Figure 5-24 Enabler relevant domains

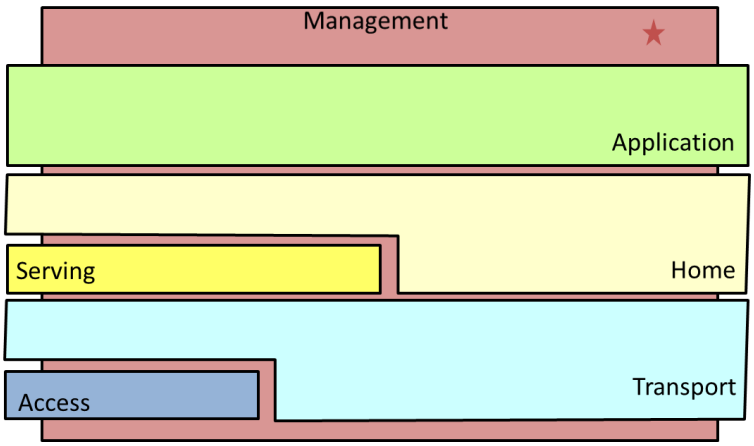


Figure 5-25 Enabler relevant strata

5.5.9.3 Quality attributes

Following standard formats in all data exchanges. Ease of use by the Security Operator.

5.5.9.4 Technical constraints

Relying on GCI to provide required data on assets and implemented security controls in the same domain, and in different domains.

5.5.9.5 Business constraints

No business constraints.

5.5.10 Detailed specifications

5.5.10.1 API specifications

Release two of this enabler will provide an API to subscribe to the non-compliance events and a REST API to access the security state of the system (also used by the Dashboard visualisation), as outlined below.

5.5.10.1.1 Non-compliance events

SSSR will send reports about non-compliant assets to GCI. Vulnerability entries' format will be used with minor adjustments.

5.5.10.1.2 Security state API

The API consists of REST CRUD methods for:

- Model – selected design time model
- Assets – assets in the selected model
- Relations – relations between assets in the selected model
- Reports – received GCI reports
- Non-compliance events – events generated after a report was received

All responses to be returned in JSON.

Example of events API:

- GET /events – return IDs of all events generated for the most recent report as JSON
- GET /events/{eventId} – return details of an event with a particular ID as JSON

The API Is still under development and will be presented in more detail in D3.7.

5.5.11 Re-utilised Technologies/Specifications

This enabler builds on the Secure System Designer software from the OPTET project [1].

5.5.12 References

[1] OPTET Project: www.optet.eu

6 Network management and virtualisation isolation enablers open specifications

6.1 Anti-fingerprinting enabler: open specification

6.1.1 Preface

The separation of the network planes (e.g., the data plane and control plane as in SDN) opens the doors for a remote adversary to fingerprint the network. For instance, in an SDN network, whenever packet forwarding is performed in hardware, then packets at the data plane are processed several orders of magnitude faster than at the software-based control plane. This discrepancy acts as a distinguisher for a remote adversary to learn whether a given probe packet is handled just at the data plane or triggers an interaction between the data plane and the control plane. An interaction provides evidence that the probe packet does not have any matching flow rule stored at the switch's flow table (or it requires special attention from the controller). This knowledge empowers an adversary with a better understanding of the network's packet-forwarding logic and it even might reveal some information about the network's topology. A network operator wants or is even required to prevent the leakage of such kind of information, since it exposes the network to a number of threats. In particular, with this additional knowledge it is possible to launch more powerful denial-of-service (DoS) attacks.

6.1.2 Status

A prototype of the enabler has been developed and evaluated. Details are given in [Cui et al., 2016]. The specification of the enabler is a preliminary one.

6.1.3 Copyright

The enabler is owned by NEC. NEC is currently the only contributor to this enabler.

6.1.4 Legal notice

The Legal Notice that applies to this specification is given in Annex A.

6.1.5 Terms and definitions

DoS	Denial of Service
ONF	Open Networking Foundation
OpenFlow	Protocol standardized by the Open Networking Foundation (ONF) for controller-switch communication.
SDN	Software-Define Networking

6.1.6 Overview

This security enabler prevents fingerprinting attacks in networks with separated planes like in an SDN network. More concretely, certain packets of a network flow are delayed at a switch before the switch forwards them. Such a delay mimics an interaction between components at different network planes. In an SDN network, this would be the interaction between the switch and the network controller. With this enabler in place, a remote attacker (active or passive) cannot distinguish anymore whether a real interaction took place or an artificial delay. Note that the impact on the network performance is insignificant, since the enabler only delays a few packets of a network flow. Experiments have shown that this is already effective against fingerprinting attacks.

6.1.7 Basic concept

SDN separates the control and data planes by defining a switch's programming interface and a protocol to access such interface, i.e., the OpenFlow protocol. The controller leverages the OpenFlow protocol to access the switch's programming interface and configure the forwarding behaviour of the switch's data plane. The communication between the controller and switches is established using an out-of-band control channel.

The core entities exposed by the OpenFlow switch's programming interface are flow tables and flow rules. A flow table of a switch is just a container for its flow rules, which define the switch's forwarding behaviour. The controller can add, delete, or modify flow rules of a switch's flow table by sending an OFPT_FLOW_MOD OpenFlow message to the switch. The parameters of an OFPT_FLOW_MOD message specify how the flow table of the switch should be modified. A flow rule, for instance, provides a semantic like "if a network packet's IP destination address is 1.2.3.4, then forward the packet to port 2." In general, a flow rule contains a match set that defines the network packets to which the rule applies. It further contains an action set that defines the actions that should be applied to such packets, for example, forward to port 2. Whenever a packet is received by a switch, the packet's header is used as a search key to retrieve the rule that applies to the packet, by performing a lookup in the flow table. The lookup operation compares the packet's header with the rules' match set to find the rule that matches the packet. Rules are prioritized in case multiple rules match. For the cases in which the controller needs to inspect a network packet, before performing a forwarding decision and installing the corresponding forwarding rules, OpenFlow defines a special "forward to controller" action. When this action is applied to a packet, the switch generates an OFPT_PACKET_IN message that is sent to the controller. This message contains the original packet and some additional information, such as the switch and the port ID onto which the packet was received.

The OFPT_PACKET_IN feature is used in basic network control logic implementations, such as the one of an Ethernet learning switch. It is also used in more complex dynamic control plane implementations. In both cases, the network operates as follows: a packet received by the switch generates an OFPT_PACKET_IN message; the controller receives and analyses the message to take a forwarding decision; the decision is finally implemented by sending OFPT_FLOW_MOD messages, which install rules at the relevant switches. This ensures that all similar packets, i.e., those that belong to the same network flow, are forwarded directly by the switches with no further interactions with the controller.

The main objective of this enabler is to study the ability of a remote adversary to identify whether an interaction between the controller and the switches (and a subsequent rule installation) has been triggered by a given packet. The absence of a controller-switch interaction typically provides evidence that the flow rules that handle the received packet are already installed at the switches. Otherwise, if a communication between the controller and the switches is triggered, then this suggests that the received packet requires further examination by the controller, e.g., since it does not have any matching entry stored at the switch's flow table, or because the controller requires additional information before installing a forwarding decision at the switches.

6.1.8 Main interactions

6.1.8.1 Use cases

The enabler prevents information leakage about how network packets are processed in an SDN network. In particular, it prevents the leakage of the information about which packets trigger a controller-switch

interaction. Having such information at hand makes an SDN network vulnerable to different kinds of attacks. Cf. the uses case 5.3 from [Deliverable 2.1, 2016].

Rule scanning. By fingerprinting the SDN network, an adversary can infer whether a flow rule has been already installed by the controller to handle a specific type of traffic or route towards a given destination. For example, the adversary can craft probe packets whose headers match the traffic type and/or destination address and infer by measuring the timing of the packets whether these packets triggered the installation of a rule. This provides a strong evidence for the adversary that communication with the given destination address has recently occurred. Depending on the underlying rule, the adversary might also be able to infer the used network protocol, and the destination port address. By doing so, the adversary obtains additional information about the occurrence of a particular communication event. For example, the adversary can infer whether the destination address has recently established an SSL session to perform an e-banking transaction. Note that this leakage is only particular to SDN networks, and does not apply to traditional networks. Also, note that the adversary can send the probe packets from a remote destination. However, additional knowledge about the network or the network slices reduces the adversary's space of crafted probe packets.

The fingerprinting of rules enables the adversary to better understand the logic adopted by the controller in managing the SDN network. This includes inferring the timeouts set for the expiry of specific rules, whether the controller aims at fine-grained or coarse-grained control in the network, etc. Similar to existing port and traffic scanners, this knowledge can empower the adversary with the necessary means to compromise the SDN network. Even worse, the adversary can leverage this knowledge to attack other networks which implement a similar rule installation logic. For instance, in a geographically dispersed datacentre, different subdomains typically implement the same policies. The adversary can train using one subdomain and leverage the acquired knowledge in order to compromise another subdomain.

Denial of service. The rule space is a scarce resource in existing hardware switches. Namely, state-of-the-art OpenFlow hardware switches can only accommodate few tens of thousands of rules, and only support a limited number of flow-table updates per second. While these limitations can be circumvented by means of a careful design of the rule installation logic, an adversary that knows which packets cause an interaction with the controller can abuse this knowledge to launch tailored DoS attacks. For instance, an adversary might simply try to overload the controller with processing OFPT_PACKET_IN OpenFlow messages. Instead of blindly guessing for which packets a switch sends an OFPT_PACKET_IN OpenFlow message, the adversary first fingerprints the SDN network, i.e., it gains knowledge for which packets a switch interacts with the controller. This can be done passively by observing the network traffic. The adversary then exploits this knowledge by sending dedicated packets, where each of them most likely triggers a controller-switch interaction.

Another kind of DoS attack is to fill up the switches' flow tables. An analogy to this is when a computer runs out of memory and starts swapping. Usually, the computer becomes unusable. Similarly, the network performance is severely harmed when the flow tables are full (or even almost full). First, installing flow rules in an almost full table is more costly than in an almost empty flow table. Second, in case the flow table is full, either new network flows cannot be established, which would already be a denial of service, or some installed flow rules need to be deleted. However, in general, it is not obvious which rules should be deleted to make room for new rules; this needs to be coordinated by the controller and is a complex operation, which can quickly overload the controller and the switches. For example, the deletion of a rule

of an ongoing network flow might entail the rule's immediate reinstallation. This can escalate and the controller will have to constantly delete and reinstall rules.

6.1.8.2 Components and interaction overview

The enabler comprises a unit in an OpenFlow switch to mimic controller switch-interactions.

6.1.9 Architectural drivers

6.1.9.1 High-level functional requirements

Mimic a controller-switch interaction.

6.1.9.2 Quality attributes

Fingerprinting attacks on the controller-switch interactions should become infeasible. That means, an attacker, should not be able to learn (with a high probability) which packets trigger the switch to contact the controller. Furthermore, protecting the network from fingerprinting attacks should not substantially harm network performance.

6.1.9.3 Technical constraints

Switches must be extended by a unit for mimicking the controller-switch interactions.

6.1.9.4 Business constraints

There are currently no business constraints.

6.1.9.5 Link to security architecture

Figure 6-1 and **Figure 6-2** show the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture [Deliverable 2.4, 2016].

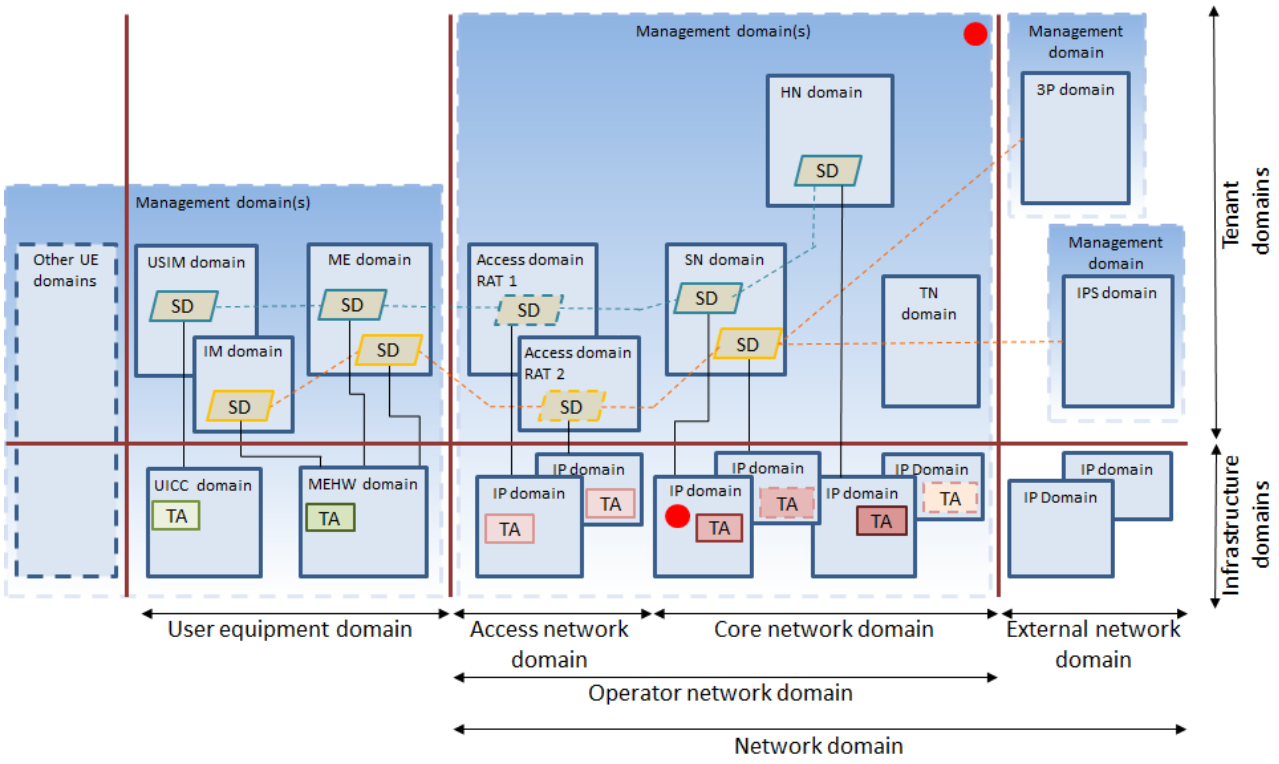


Figure 6-1 Enabler relevant domains

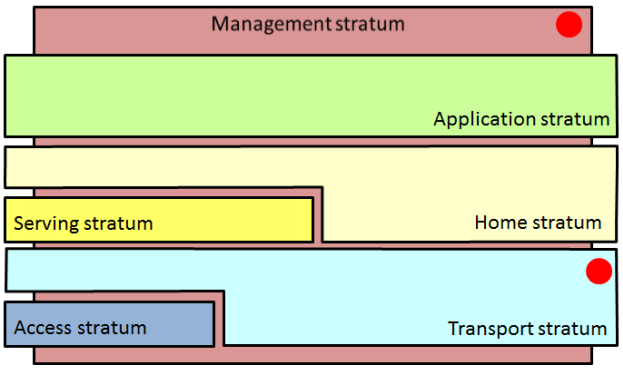


Figure 6-2 Enabler relevant strata

6.1.10 Detailed specification

6.1.10.1 Introduction

This enabler is "SDN generic," i.e., the switch unit that mimics the interaction with the controller could be used in any SDN network.

6.1.10.2 Conformance

An implementation to be reported as conformant should comply with the open specification here stated for the enabler.

6.1.10.3 Enabler component specification

Our proposal does not concern the handling of new flows, but focuses on processing packets which pertain to existing flows. Namely, for a packet of an existing flow, we leverage the group table and the internal timer maintained by a switch to identify whether this flow has recently appeared. The group tables are used in OpenFlow switches to describe per-packet forwarding actions. They allow one to realize forwarding strategies, such as ECMP [Thaler and Hopps, 2000], which could not be achieved using the flow table abstraction that describes only per-flow forwarding actions. A group table contains one or more buckets, which in turn contain an action set, similar to the one contained in the flow rules. A group table is further associated with a bucket selection logic, which is related to the group table type. For example, a group table of type "fast failover" implements a selection logic that associates each bucket to a switch's port. Then, the logic selects the first bucket in the table whose associated switch's port status is live. The action group in a flow rule's action set enables one at selecting which packets should be processed by which group.

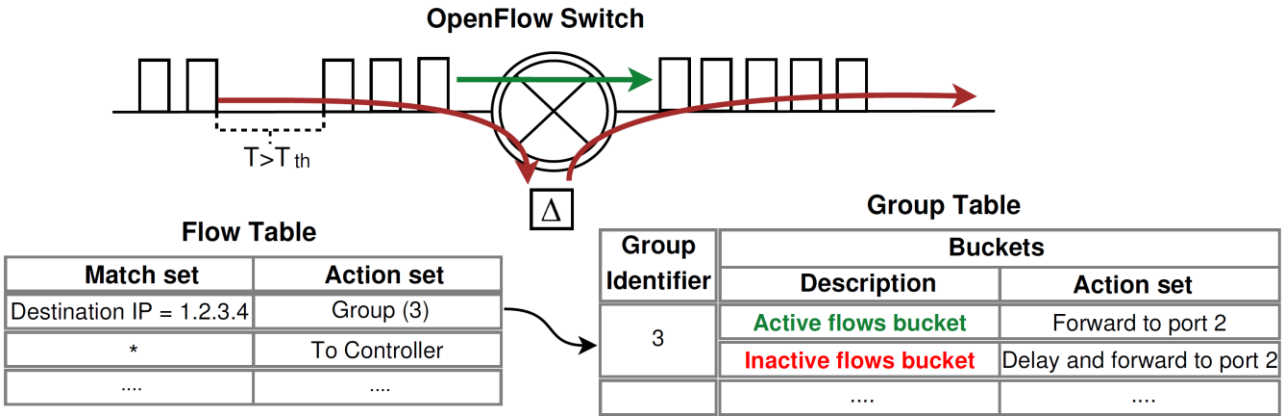


Figure 6-3 Sketch of our countermeasure. The packets with destination IP 1.2.3.4 are processed by the group table 3, which implements the bucket selection logic specified by our countermeasure. If no packets for this network flow are processed by the switch for a time T>threshold, then the next few packets of the flows are delayed by an appropriate amount.

Our proposed countermeasure (cf. **Figure 6-3**) defines a new bucket selection logic for the group table, such that packets of active flows are immediately forwarded, while packets of inactive flows are forwarded onto a special port that connects the switch to a network delay element. Our selection logic considers a flow to be inactive if no packets for such a flow were received by the switch in a threshold amount of time, threshold, which is measured (in seconds) by the switch's internal timer.

The first received packet of an inactive flow is delayed by a time close to a previously measured value (delta-RTT), which gives the adversary little advantage in identifying whether the additional delay measured by the RTT feature is caused by a controller-switch interaction or is artificially introduced by our countermeasure. Moreover, all packets of the same flow received within a short time window W are also delayed by a small delta; this procedure prevents fingerprinting attempts that leverage the dispersion feature. As shown in [Cui et al., 2016] delta and delta-RTT can be fitted to predetermined distributions, depending on the network size and the number of hops on the communication path. Alternatively, the controller can estimate the distributions corresponding to delta and delta-RTT through a feedback loop.

Notice that our countermeasure is unlikely to deteriorate network performance, since only few packets per flow are delayed by few milliseconds. We further remark that our proposal requires minor modifications—which are supported to a large extent already in the OpenFlow v1.3 specification—by the switches' manufacturers. As such, we argue that our proposal can be efficiently implemented (in hardware) within the switches.

6.1.11 References

[1] 5G-Ensure Consortium, Deliverable 2.1 Use Cases, [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf, 2016.

[2] 5G-Ensure Consortium, Deliverable 2.4 Security Architecture (draft), [Online]. Available: hhttp://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.4-SecurityArchitectureDraft.pdf, 2016.

[3] H. Cui, G. O. Karame, F. Klaedtke and R. Bifulco, "On the fingerprinting of software-defined networks," *IEEE Transactions on Information Forensics and Security,* vol. 11, no. 10, pp. 2160-2173, 2016.

[4] Open Networking Foundation (ONF). OpenFlow switch specification – version 1.3.0 (wire protocol 0x04). 2012.

[5] D. Thaler and C. Hopps. Multipath issues in unicast and multicast next-hop selection, 2000. https://tools.ietf.org/html/rfc2991.

6.2 Access control mechanisms enabler: open specification

6.2.1 Preface

In 5G, a much stronger adoption of software-defined networking (SDN) is expected than in current mobile networks. It is also expected that various network applications will run at the network's control plane on top of the controller. These applications will manage the network's data plane and offer a wide range of network services. Examples of such applications are routing applications, load balancer, and monitoring and analysis tools for network traffic. The diversity of network applications and their large-scale deployment actually applies to SDN in general. The network applications, however, might not be trusted by the network operator. Reasons for this are: (1) they might be from different network tenants or service providers, (2) they might be developed by third parties, or (3) they might contain bugs—as any complex software—and the control plane is therefore vulnerable to various kinds of attacks. Note that even if the network applications run in separate virtualized networks or network slices, they still indirectly access the same physical network resources.

It is also expected that a 5G network will comprise several service providers, each providing network functions that run in virtualized environments of a data center. These virtualized network functions (VNFs) will be managed by an orchestrator, which is, e.g., responsible for starting, terminating, and mitigating containers for these VNFs. Similar to SDN network applications, the access to network resources of the processes that run in these containers should be controlled. Analogously, these containers themselves should have only the permissions that are needed for their network tasks.

We remark that although we focus in the following on access control mechanisms on SDN networks, our comments and observations carry over to the context of VNFs and 5G networks.

6.2.2 Status

A prototype of the enabler is currently developed; the specification of the enabler is a preliminary one.

6.2.3 Copyright

The enabler is owned by NEC. NEC is currently the main contributor to this enabler. SICS and VTT are also contributor to the enabler.

6.2.4 Legal notice

The Legal Notice that applies to this specification is given in Annex A.

6.2.5 Terms and definitions

ONF	Open Networking Foundation
OpenFlow	Protocol standardized by the Open Networking Foundation (ONF) for controller-switch communication.
REST	Representational State Transfer
SBI	Southbound Interface
SDN	Software-Define Networking
NBI	Northbound Interface
VNF	Virtualized Network Function

6.2.6 Overview

Current SDN controllers fall short in restricting the access of network applications to network resources. The security enabler of this section applies the principle of least privilege to the network applications, that is, the enabler enforces that each network application must be able to only access the information and resources that are necessary for performing its tasks. To this end, a reference monitor is added to the network's control plane as a component of an SDN controller. The reference monitor permits and denies access to network resources according to a given access control policy. In particular, the reference monitor targets the controller's southbound interface, i.e., it restricts the OpenFlow messages a network application can send to and receive from the components at the network's data plane. The enabler also targets resources used by VNFs.

6.2.7 Basic concepts

SDN separates the control plane from the data plane. The control plane comprises a (logically centralized) controller. It interacts via its southbound interface (SBI) with the data plane components (e.g., switches). The most widely used SBI is the OpenFlow protocol [McKeown et al., 2008]. Network applications run at the network's control plane. Various network application deployment forms are possible, e.g., directly integrated within the controller or as separate entities that communicate through a so-called north-bound interface (NBI) like a REST API. Examples of such applications are network traffic routing applications and network monitoring applications. A network application indirectly accesses the data plane components by interacting with a controller's northbound interface.

An access control policy stipulates which subjects (e.g., the network applications) can access which objects (e.g., the switches) and how. A reference monitor enforces a given access control policy, i.e., it permits or denies an access request. For example, a reference monitor denies a network monitoring application to modify the flow rules of a switch, while reading the counters of the flow rules are permitted.

6.2.8 Main interactions

6.2.8.1 Use cases

Widely used APIs and the resulting deployment of various third-party applications pose new security threats in SDN. Indeed, malicious or misbehaving network applications can leverage such an API and infiltrate the network. These threats become even more evident when the network owner "leases" network slices, which are administrated by the leasing tenants, which in turn might install their own, possibly third-party, network applications that interact with the network owner's controller. Since the leasing tenants might have competing objectives, access to the shared network resources must be appropriately handled and secured. Cf. the use case 5.2 of [Deliverable 2.1, 2016].

Note that by using network virtualization the network owner can provide an isolated view on the network to each leasing tenant. However, the different tenants still share the data plane's network resources, which they indirectly access whenever they access one of their virtual network components. Furthermore, accessing a virtual component might result in accessing several data plane components, e.g., when a virtual component comprises multiple components of the data plane as in the "one-big-switch" abstraction. The access translation can be non-trivial and the controller's compiler implementing the translation might be incorrect because of software bugs or controller misconfigurations. This in turn can lead to incorrect updates of the switches' forwarding logic and such vulnerabilities of the controller can be exploited to launch attacks to the network.

One way to reduce the network's attack surface and protect network resources is by enhancing the network owner's controller (i.e., the controller that directly interacts with the network components at the physical data plane) with an access control mechanism. Such a mechanism could control and restrict the access of the network users (i.e., the network owner and the leasing tenants) and the applications to the network components at the data plane. This enabler adds a reference monitor—similar as in an operating system [Anderson, 1973]—as an additional controller component.

6.2.8.2 Components and interaction overview

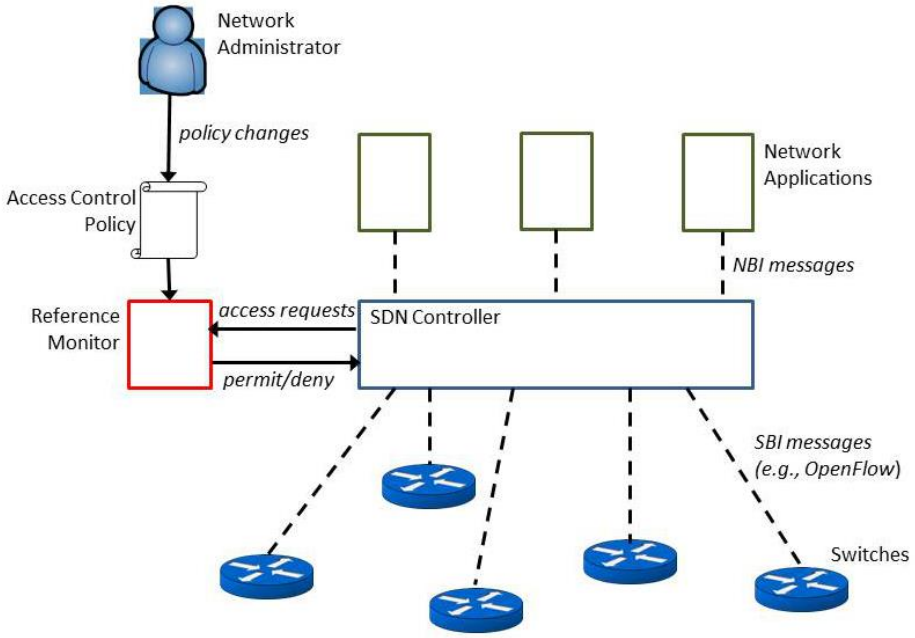


Figure 6-4 Components and their interactions

The main component of the enabler is the reference monitor. Its inputs are an access control policy and access requests. For each access request, it outputs a permit or deny. See **Figure 6-4** for illustration.

The access requests are initiated by network events. For example, a network application requests to install new flow rules in a switch, or a switch receives a network packet that does not match any of its flow rules and is thus forwarded to the controller. In the first case, the controller requests to send OpenFlow messages of the type OFPT_FLOW_MOD to a switch, which, e.g., originate from a NBI message sent from a network application to the controller. In the second case, the controller receives an OpenFlow message of the type OFPT_PACKET_IN from a switch and requests to forward the message to a network application.

The decisions of the reference monitor are according to the given access control policy. The network administrator can make changes to this policy. The network applications can also make policy changes. However, their changes are limited and only concern the objects that they own. For example, a network application A can allow another network application B to read the counters of a flow rule that A owns (i.e., created).

6.2.9 Architectural drivers

6.2.9.1 High-level functional requirements

The reference monitor must check for each access request whether the access is compliant according to a given access control policy. That is, it must either permit or deny an access. The decision must be according to the given access control policy.

6.2.9.2 Link to security architecture

The enabler's main security objective with respect to the 5G-ENSURE security architecture [Deliverable 2.4, 2016] is to secure the management and (re-)configuration of network devices. **Figure 6-5** and **Figure 6-6** shows the relevant domains and strata (marked by a red dot) of the 5G-ENSURE security architecture.

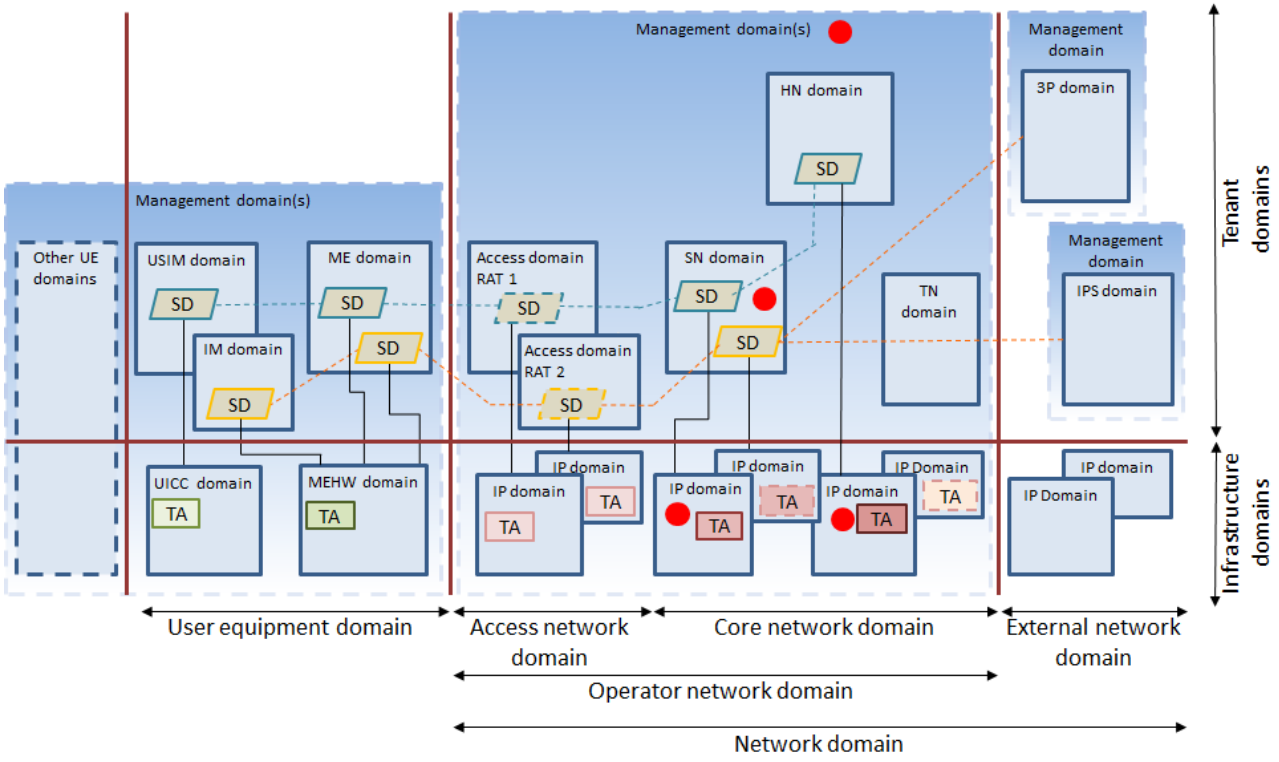


Figure 6-5 Enabler relevant domains

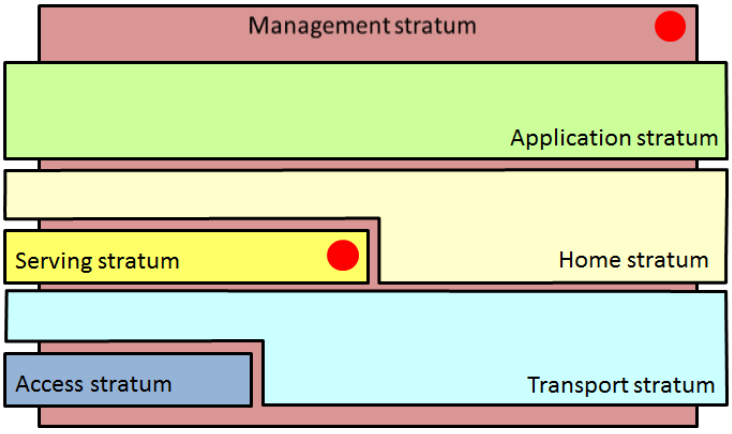


Figure 6-6 Enabler relevant strata

6.2.9.3 Quality attributes

The overhead of checking whether a resource can be accessed must be low. In particular, the reference monitor must not significantly decrease the performance of an SDN controller. This means, the reference monitor must efficiently decide whether a requested access is permitted or denied with respect to the given access control policy. Furthermore, the reference monitor should be tamper proofed and verifiable.

6.2.9.4 Technical constraints

The reference monitor must be integrated as a separate component to the SDN controller. The reference monitor must be invoked by the controller on every OpenFlow message that is sent to the control plane and that is received from the control plane. Analogously for VNFs, the usage of resources needs to be checked against the given policy and granted.

6.2.9.5 Business constraints

There are no business constraints for this enabler.

6.2.10 Detailed specification

6.2.10.1 Introduction

The enabler is "SDN generic," i.e., the reference monitor is a component of an SDN controller. We opted for access control mechanisms (access control policies and reference monitor) that are simple, focused, and close to the SBI of the controller, which interfaces directly with the switches using the OpenFlow protocol. The rationale behind this design decision is as follows. First, it supports one to build a tamperproof and verifiable reference monitor. This is based on the scheme's simplicity and since it is focused. Furthermore, since the controller only communicates via OpenFlow messages with the switches, we obtain complete mediation by permitting or denying OpenFlow messages by the reference monitor before they are sent. These are essential principles for a reference monitor [Anderson, 1973].

Rendering the reference monitor tamperproof from the applications running on top of the controller can be achieved by simple OS-level techniques (process or user isolation, container isolation, etc.). Note that the reference monitor is a separated component. Increased isolation can be achieved by using containers or VM separation. However, the increased isolation might not justify the performance degradation due to additional context switches, system calls, and so on.

We expect that future NBIs in SDN will support multiple different abstractions of the network at the control plane. Any such interface will be built on top of the interface provided by OpenFlow, which directly interacts with the network components. Access control at higher layers will utilize our access control scheme and complement it.

In the following, we specify the reference monitor's input in more detail.

6.2.10.2 Conformance

An implementation to be reported as conformant should comply with the open specification here stated for the enabler.

6.2.10.3 API specification

Methods. In order for the reference monitor to check that the SBI traffic conforms to the policy, it needs to be hooked on all calls delivering messages to or from the switches.

When a message arrives from the switch, the following method should be called:

Boolean RequestReceiving(Switch src, Application dst, OpenFlowMessage msg)

where src indicates the source switch from which the message originates from, dst is the application to which the controller intends to distribute the message, and msg is the OpenFlow message itself. As is the case for other information coming from the controller's NIB, it is assumed that the controller already authenticated the information. For example, in the case of switch source, the controller guarantees the authenticity of the switches and TLS can be used for the OpenFlow communication between the controller and the switches.

To avoid bypassing the reference monitor, the controller has to cover all possible delivery scenarios. This might include direct relay to a specific application for a certain message type (e.g., link discovery messages go to an LLDP application), a pipeline in which all applications read the message sequentially or a store in which subscribed applications are notified. These and any other means of delivering messages to the application have to be instrumented such that the RequestReceiving hook is called.

After evaluating the message, source and destination against the policy, the method returns a Boolean value, where true means that receiving the message is permitted and false means that receiving the message is denied. To address in the future more complex scenarios, a binary permit/deny decision might not be enough. In particular, in multitenant networks where multiple policies have to be composed, the decision of a reference monitor could also be "don't know," a third truth value. Furthermore, in addition to the reference monitor's decision, the reference monitor could return information explaining its decision. In particular, when denying the request, such information can be helpful for the source. However, such extensions are currently not considered for the enabler.

In the other direction, before sending messages to the switches the following method should be first called:

Boolean RequestSending(Application src, Switch dst, OpenFlowMessage msg)

where src is the originating application, dst is the affected switch, and msg is the OpenFlow message.

Similar to delivering messages to applications, sending messages to switches must cover all check points in the controller to avoid bypassing the reference monitor.

As for the method RequestReceiving, RequestSending's return value specifies whether the sending request is policy compliant (true) or noncompliant (false).

In order to support dynamic policy changes, the following method allows the administrator to modify or replace the policy:

Boolean UpdatePolicy(Policy pol)

where pol is either an entirely new policy replacing the old one or a fragment that is incorporated in the existing policy to extend, restrict or otherwise modify the behaviour. If the method's return value is true, the update was successfully applied and new requests will be processed according to the new policy. Otherwise, there was an error applying the new policy.

A separate method might be used by applications to refine the policy concerning resources that they own:

Boolean UpdateAppPolicy(Application app, Policy pol)

where app is the application wishing to extend or restrict how other applications might interact with resources it owns (i.e.: give read only access to counters of flows that it installed).

Messages. The messages that are permitted or denied to transit the SBI correspond to OpenFlow messages. Depending on the controller, they might be decorated with additional fields. The kinds of fields likely to be used in a policy include but are not limited to:

- DPID – switch identifier (Data Path ID),
- ingress port – the port on which the message was received,
- source – e.g., MAC or IP address of the source,
- destination – e.g., MAC or IP address of the destination, and
- flow entry – the entry to modify in case of a flow mod.

The available fields depend on the message type (e.g., OFPT_PACKET_IN for a message received by an application and OFPT_FLOW_MOD for a message sent by an application). Apart from the information about encapsulated layers (e.g., IP address or TCP port) that the controller extracts when it deserializes an OpenFlow message, other network information can be used inside policies. Controllers regularly collect facts about links, switches, and other network resources that they store in a network information base (NIB). The NIB can be augmented with an *ownership tag* that can be used by applications to delegate permissions (see UpdateAppPolicy method). Depending on the expressiveness of the policy, other tags could be associated with resources stored in the NIB. When communicating with the reference monitor, this extra information could be part of the OpenFlow message object or it could be passed as an additional context parameter.

6.2.11 Re-utilised technologies/specifications

The reference monitor will be integrated into the ONOS controller [Berde et al., 2014]. Some of the concepts of this enabler have been described by Klaedtke et al. [2014] and [2015].

6.2.12 References

[6] 5G-Ensure Consortium, Deliverable 2.1 Use Cases, [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf, 2016.

[7] 5G-Ensure Consortium, Deliverable 2.4 Security Architecture (draft), [Online]. Available: hhttp://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.4-SecurityArchitectureDraft.pdf, 2016.

[8] J. Anderson. Computer Security Technology Planning Study. Technical Report ESD-TR-73-51. US Air Force Electronic System Division, 1973.

[9] P. Berde, M. Geralo, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Conner, P. Radoslavov, W. Snow, and G.M. Parulkar. ONOS: towards an open, distributed SDN OS. In Proceedings of the 3rd SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN). ACM Press, 2014.

[10] F. Klaedtke, G. O. Karame, R. Bifulco and H. Cui. Access control for SDN controllers. In *Proceedings of the 3rd SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, 2014.

[11] F. Klaedtke, G. O. Karame, R. Bifulco and H. Cui. Towards an access control scheme for accessing flows in SDN. In *Proceedings of the 1st IEEE Confernce on Network Softwarization (NetSoft)*, 2015.

[12] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling innovation in campus networks. SIGCOMM Computer Communication Review, 38(2):69–74, 2008.

[13] ONOS - a new carrier-grade SDN network operating system designed for high availability, performance, scale-out. [Online]. Available: http://onosproject.org/.

Open Networking Foundation (ONF). OpenFlow switch specification – version 1.3.0 (wire protocol 0x04). 2012.

6.3 Component-interaction audits enabler: open specification

6.3.1 Preface

Networks comprise multiple components, e.g., endhosts and switches, and a controller in case of an SDN network. Policies specify how these components must and must not behave. There is a wide spectrum of policies, targeting various aspects of a network like correctness, performance, quality of service, reliability, and security. Detecting noncompliant behaviour of components with respect to a given policy is an important task to ensure the correct and safe operation of a network. In particular, in a network in which physical and virtual components are managed by different tenants (e.g., a network with multiple network or service providers), the detection of noncompliant behaviour of a component is a major concern for the network operator. It helps the operator to protect the network, e.g., against misbehaving components and misconfigurations.

6.3.2 Status

A prototype of the enabler is currently developed; the specification of the enabler is a preliminary one.

6.3.3 Copyright

The enabler is owned by NEC. NEC is currently the main contributor to this enabler. SICS and VTT are also contributor to this enabler.

6.3.4 Legal notice

The Legal Notice that applies to this specification is given in Annex A.

6.3.5 Terms and definitions

NTP	Network Time Protocol
ONF	Open Networking Foundation
OpenFlow	Protocol standardized by the Open Networking Foundation (ONF) for controller-switch communication.
SDN	Software-Define Networking
VNF	Virtualized Network Function

6.3.6 Overview

The scope of the enabler is to check the interactions between network components against a given policy, which specifies how components must and must not interact with each other. The checking can be done online, i.e., while the components are running, or offline during an audit. The enabler supports a rich formally defined specification language for expressing a large variety of policies. It is not in the scope of the enabler to perform corrective actions in case the network components interactions are not policy compliant.

6.3.7 Basic concepts

A *policy* is an informal or formal document consisting of rules that stipulate, for instance, when performing an action is allowed and who is allowed to perform the action. A (*policy) violation* is a behaviour that

violates one of the policy rules. A behaviour is *(policy) compliant* if no rule of the given policy is violated by the behaviour.

A *(network) component* is an entity of a system (network) that performs actions. For example, a switch processes network packets and forwards packets to ports. Components usually *interact* with each other. The interaction is according to a given protocol and usually asynchronous. For example, the controller of an SDN network interacts with the switches by sending and receiving OpenFlow messages [McKeown et al., 2008]. Note that a controller can be composed out of several smaller components, which again interact with each other. It depends on the given policy on which abstraction level the system is considered.

A *compliance checker* monitors the components, in particular, their performed actions. With respect to a given policy, it reports compliant and noncompliant behaviour by outputting verdicts, e.g., explaining when the policy was violated and which action caused the violation.

6.3.8 Main interactions

6.3.8.1 Use cases

One use case of this enabler is to increase the resilience of the network. For this, the network administrator deploys the compliance checker into the network. Furthermore, the administrator instruments some of the network components such that they send messages describing their performed actions to the compliance monitor. A policy specifies how these components should behave, e.g., how they must and must not react to certain network events. Whenever the policy is violated, the network administrator is notified by the compliance checker and can take countermeasures against this noncompliant behaviour. Policy violations can for example be caused by a wrongly configured component or a malicious component. Cf. the use case 5.2 of [Deliverable 2.1, 2016].

Another use case is where the network administrator deploys a new component to the network. Similar to the first use case, the performed actions of this new component are checked against a policy and noncompliant behaviour is detected and reported to the network administrator. For instance, a new component might behave not as intended because of a software bug or a misconfiguration, which are then detected by the compliance checker. Note that buggy software and misconfigurations might make the network vulnerable to attacks. Cf. the use case 5.4 of [Deliverable 2.1, 2016].

6.3.8.2 Components and interaction overview

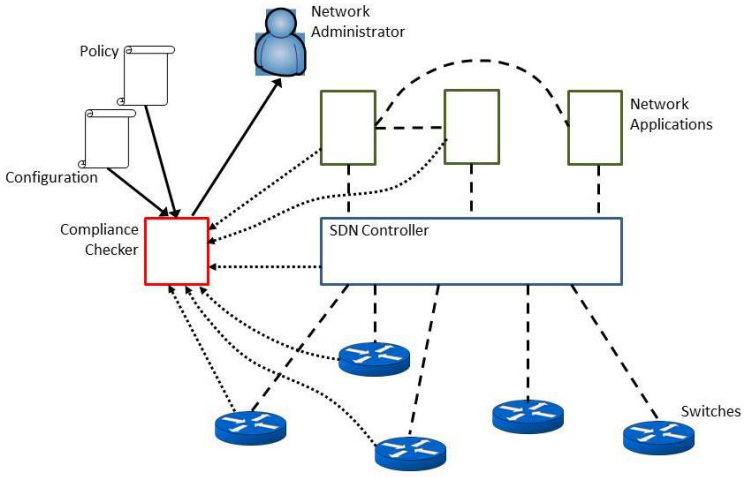


Figure 6-7 SDN network with its components and their interactions

Figure 6-7 provides an overview of the involved components and illustrates with whom the enabler interacts in an SDN network and how. Recall that switches, controllers, and network applications are network components. Note that this list is not exclusive. Furthermore, note that the network components interact with each other. For example, in an SDN network, the controller sends OpenFlow messages to the switches. An OpenFlow message can, e.g., be initiated by a network application. The main component of the enabler is the compliance checker. The interactions between the network components are illustrated in **Figure 6-7** by the dashed lines.

In the remainder of the enabler specification, we focus on the enabler's feature of targeting SDN networks as illustrated in **Figure 6-7**. However, the enabler's second feature, which targets the checking of policies about VNFs, running in Docker containers [Docker] and their reconfigurations in networks the components interact analogously with the compliance checker. Namely, the compliance checker receives the actions performed that concern the reconfiguration of VNFs (e.g., the VNF orchestrator and the virtualization environment). In particular, the message format describing the performed actions is identical. Analogously to **Figure 6-7** the compliance checker processes the received messages (either online or offline) and checks whether these actions are compliant with respect to given policies or workflows, provided by the network operator. In case of a violation, the compliance checker outputs a warning, e.g., it informs the network operator. Furthermore, since the interface for specifying the policies is similar to the first feature, we focus on the first feature in the following.

In the following, we explain the compliance checker's input and output in more detail. The inputs to the compliance checker are of two kinds. In **Figure 6-7**, they are illustrated as solid and dotted ingoing arrows.

- The first kind of input (the solid ingoing arrows in **Figure 6-7**) is static and provided by the start of the compliance checker. These inputs are the policy and the configuration. The configuration describes the network components that should be monitored and the format of the messages that describe the components performed actions. The policy specifies the interactions between the network components that are allowed.
- The second kind of input (the dotted ingoing arrows in **Figure 6-7**) are the messages from the network components that describe their performed actions. When using the compliance checker

online, the monitored networks components send messages describing their performed actions to the compliance checker. Note that when using the compliance checker offline, these messages are logged (either directly by the component or by a log server) and the obtained logs are, e.g., merged and processed by the compliance checker during an audit.

The output of the compliance checker are verdicts. They are sent to the network administrator, or some other entity, which may want to take corrective actions when the given policy is violated. These corrective actions, which are not in the enabler's scope, may depend on the frequency a policy is violated. The enabler's focus is on detecting policy violations. Note that verdicts are sent while the compliance checker processes the messages from the network components.

We remark that the network components are distributed, the components act concurrently, and the communication between the components is asynchronous. When using the compliance checker online, it can be seen as yet another network component, which receives and processes messages from other components. The messages that are sent to the compliance checker can arrive at any order. Furthermore, components can crash, including the compliance checker, and recover later.

Furthermore, we remark that the current version of the enabler assumes that: (1) messages are not tampered with, and (2) components correctly report their actions and do not send bogus messages. The first assumption can be easily discharged by adding information to each message such as a cryptographic hash value, which is checked by the compliance checker when receiving the message. To discharge the second assumption additional mechanisms need to be deployed. For example, a message contains a timestamp when the action is performed. To ensure the correctness of the timestamp a trustworthy clock must be used that signs the timestamp. The compliance checker can then check the validity of a timestamp by checking its signature. For ensuring that a reported action was performed or a non-reported action did not take place, one could deploy additional monitors or "traps" that check this. Such monitors or "traps" are, however, action and component dependent and not in the scope of the current version of the enabler.

6.3.9 Architectural drivers

6.3.9.1 High-level functional requirements

Policies are formally specified as formulas of a temporal logic. The main functionality of this enabler is to monitor network components, in particular, their performed actions, and detect and report all the performed actions that are noncompliant with respect to the given policy. The enabler's output, i.e., the verdicts it outputs should be sound and complete. Soundness means here that the enabler only outputs verdicts that correspond to policy violations. Completeness means here that the enabler outputs a verdict for every policy violation.

Note that the enabler might not have complete knowledge about all the performed actions. For example, the performed action has not yet been received by the monitor due to message loss. In this case, the enabler should not output any verdict.

6.3.9.2 Link to security architecture

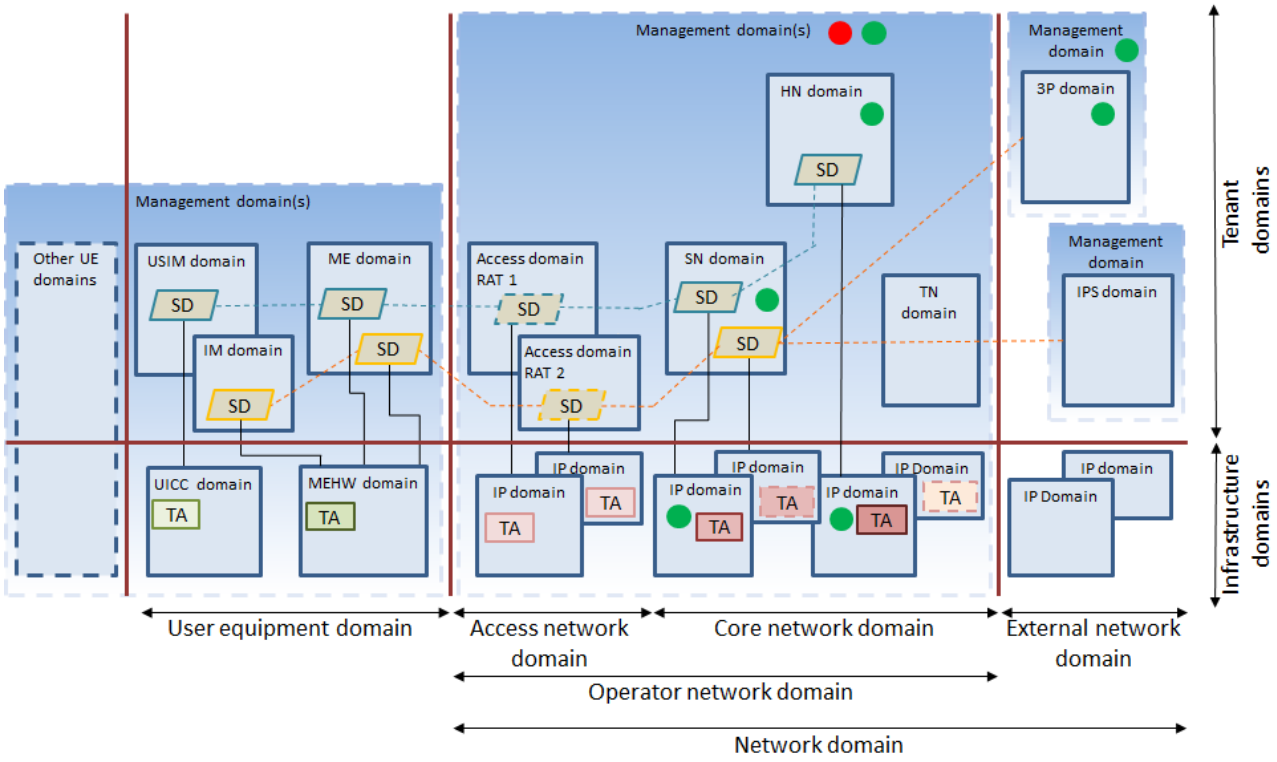


Figure 6-8 Enabler relevant domains

Figure 6-8 shows the most relevant domains of the 5G-ENSURE security architecture [Deliverable 2.4, 2016] that are relevant for this enabler. The domains for which the enabler contributes to their security are marked by a red dot. Since the enabler targets to check policy compliance of management tasks of the core network, only the management domains of core network domain are marked by a red dot. However, in principle, the enabler can be used to check compliance of network tasks in other domains. Furthermore, note that depending on the given policy, the enabler may receive information about actions performed by network components located in other domains. **Figure 6-8** marks the common domains from which information must be sent to the enabler by a green dot. Analogously, **Figure 6-9** marks the relevant strata of the 5G-ENSURE security architecture.

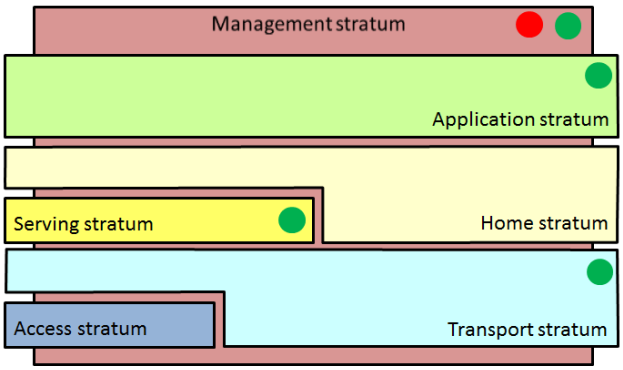


Figure 6-9 Enabler relevant strata

6.3.9.3 Quality attributes

When using the compliance checker online, it should be capable of processing the received messages at the speed of the network that performs these actions. That is, the compliance checker's throughput of processing messages should be most of the time has high as the number of the (policy-relevant) actions that are performed by the network components. Otherwise, the compliance checker would lag behind and output verdicts late. Note that the compliance checker can store unprocessed messages in a queue in case it receives many messages in a short time. In this case, it might lag behind of outputting verdicts for some time. However, this should only rarely happen and the compliance checker should quickly "catch up". Even in an offline use, logs should be processed efficiently.

6.3.9.4 Technical constraints

The network components must be instrumented such that they inform the compliance checker about their actions that are policy relevant actions. For this, we require that the actions must only be performed via trustworthy subcomponents.

The format describing these actions must be standardized. The compliance checker needs to be configured appropriately. The first feature of the compliance checker focuses on the network components that interact via OpenFlow messages, i.e., the interaction between switches and SDN controllers. The second feature focuses on interactions between Docker containers, the VNFs running in them, and the orchestrator that manages the VNFs.

Furthermore, the network components need to timestamp their actions. The timestamps are used by the compliance checker to relate the received actions timewise. Because of clock drifts, the network components must synchronize their clocks regularly, e.g., by the Network Time Protocol (NTP) [Mills, 1995]. The verdicts of the compliance checker are only correct with respect to the provided timestamps.

6.3.9.5 Business constraints

There are currently no business constraints.

6.3.10 Detailed specification

6.3.10.1 Introduction

The enabler is "SDN generic," i.e., the compliance checker is a network component at the control plane of an SDN network. It runs separately to the other control plane components (e.g., controller and network applications). The interaction with the other components is "one way." In particular, in its online use, it only receives over a socket messages from the other network components, both at the data plane and control plane. The compliance checker does not directly impact the network traffic and the network configuration. Nevertheless, it might indirectly impact the network configuration, since it outputs verdicts, on which a network administrator, e.g., can take corrective actions. In the following, we specify the compliance checker's input, and also its output, in more detail.

6.3.10.2 Conformance

An implementation to be reported as conformant should comply with the open specification here stated for the enabler.

6.3.10.3 API specification

Messages. The messages that are sent to the compliance checker and describe the monitored components performed actions must be timestamped. The timestamp of a message describes when the action was

carried out. The enabler assumes the timestamp as Unix time (with integral part and a possibly fractional part). Furthermore, the message must specify the sender and a sequence number. The sequence number is the number of messages that the sender has sent so far to the compliance checker. With the sequence numbers the compliance checker can determine whether it has received all messages up to a given time. Finally, the message must describe the performed action. Overall, the fields of a message are as follows.

timestamp	sender identifier	sequence number	description of performed action

For interpreting the received messages, a configuration file must be provided to the compliance checker. This configuration file consists of rules with regular-expression matching to identify the performed action and to extract relevant data values. The first matching rule determines the interpretation of the message. One can think of these rules as an if-then-else program.

As an example, consider the rule

```
[event matches “OFPT_FLOW_MOD”, event matches “ \(xid=0x(?P<xid>[0-9abcdef]+)\)”]
=>
 flow_mod(<xid>)
```

for an SDN controller like Ryu. It matches a message that contains the string OFPT_FLOW_MOD and a string of the form (xid=0x*hexnumber*). The interpretation is that the controllers performed action is the sending of an OpenFlow message of the type OFPT_FLOW_MOD with the xid *hexnumber*. The specified policy, which we describe next, can refer to predicates like flow_mod. See also Section **Error! Reference source not found.**.

Policy specification language. The enabler uses a temporal logic to express and formalize policies. We refer to Alur and Henzinger [1992], Baier and Katoen [2008], and Basin et al. [2015] for background. The core grammar of the enabler’s policy specification language is given by the following grammar.

spec ::= TRUE
| p(x1, …, xn)
| (NOT *spec*)
| (*spec* OR *spec*)
| (*spec* SINCE[a,b] *spec*)
| (*spec* UNTIL[a,b] *spec*)
| (FREEZE x1[r1], …, xn[rn]. *spec*)

Here, p is a predicate symbol, x1, …, xn range over variables, a and b are nonnegative integers, and r1, …, rn range over data registers.

- TRUE specifies the trivial policy that any behaviour satisfies.
- p(x1, …, xn) specifies the policy that a behaviour satisfies at time t whenever in (x1, …, xn) are in the relation p at time t.
- (NOT spec) specifies the policy that a behaviour satisfies whenever it does not satisfy spec.
- (spec1 OR spec2) specifies the policy that a behaviour satisfies whenever it satisfies spec1 or spec2.

- (spec1 SINCE[a,b] spec2) specifies the policy that a behaviour satisfies at time t whenever the behaviour satisfies at some time s<=t, with a<=t-s<=b, the policy spec2, and since s, the behaviour satisfies the policy spec2.
- (spec1 UNTIL[a,b] spec2) specifies the policy that a behaviour satisfies at time t whenever the behaviour satisfies at some time s>=t, with a<=s-t<=b, the policy spec2, and until s, the behaviour satisfies the policy spec2.
- (FREEZE x1[r1], …, xn[rn]. spec) specifies the policy that a behaviour satisfies at time t whenever the behaviour satisfies the policy spec, where x1 to xn are assigned to the register values x1 to xn at time t.

Various additional syntactic sugar can be defined. For example, (spec1 AND spec2) abbreviates (NOT ((NOT spec1) OR (NOT spec2))). As expected, (spec1 AND spec2) specifies the policy that a behaviour satisfies whenever the behaviour satisfies the policy spec1 and the policy spec2. (spec1 IMPLIES spec2) is syntactic sugar for ((NOT spec1) OR spec2); its meaning is as expected. Another example is (ONCE[a,b] spec) that abbreviates (TRUE SINCE[a,b] spec). A behaviour satisfies the policy (ONCE[a,b] spec) at time t whenever the behaviour satisfies spec at some time s, with a<=t-s<=b.

The connectives are assigned to different binding strengths, which allow one to omit parenthesis. We use the standard conventions here. For example, NOT binds stronger than OR. By this convention, NOT spec1 OR spec2 abbreviates ((NOT spec1) OR spec2). Finally, we also allow open intervals and half-open intervals as metric temporal constrains like (a,b) and [a,b), for integers s a and b with 0<=a<b. In these two cases b could also be infinity to specify an unbounded interval. A time unit (e.g., ms, s, and m) can be attached to the numbers. If no time unit is given the numbers are interpreted as seconds.

We refer to Alur and Henzinger [1992] for details, in particular, for the underlying temporal model and the semantics of the specification language. Examples for expressing policies are given in Section **Error! Reference source not found.**.

Verdicts. The verdicts that the enabler's output specify the time when the policy is violated. Note that this time corresponds to a message that the compliance checker has received previously. Additional information can be included to a verdict, e.g., the message that caused the policy violation. The verdicts are either written to a log file or they are sent over a socket to another program.

6.3.10.4 Examples

For illustration of the use of the compliance checker in general and the policy specification language in particular, consider a setting in which an SDN network has multiple controllers. One of them is the master controller and the others are the slave controllers. The master controller is responsible for tracking the liveness of the links between adjacent switches. However, the master controller must be elected among the controllers. Such an election takes place, for example, after a reboot of a controller (e.g., because it crashed previously). The election protocol of the ONOS controller contained a flaw, which could result in the state that no controller is tracking the liveness of the network links. See Scott et al. [2014].

By monitoring the relevant actions of the two monitors for the election process, the compliance checker could detect the harmful situation when no controller would track the liveness of the network links. The following formula expresses that whenever there is a controller election then, within 1 second, one of the controllers is the master controllers.

```
ALWAYS ControllerElection() IMPLIES
   EVENTUALLY[0,1s] FREEZE c[election_outcome]. MasterController(c)
```

Note that EVENTUALLY[a,b] spec is syntactic sugar for TRUE UNTIL[a,b] spec, and ALWAYS spec is syntactic sugar for NOT EVENTUALLY[0,infinity] NOT spec.

Furthermore, note that the above formula however does not account for the case in which more than one controller is elected as the master controller, which is also an unwanted state, which could be expressed by a second formula. For example, the following formula expresses that whenever c is elected as the master controller then there are no other controllers elected as the master controller (at most 1 second before c's election).

```
ALWAYS FREEZE c[election_outcome]. MasterController(c) IMPLIES
            NOT (ONCE(0,1s] FREEZE d[election_outcome]. MasterController(d) AND d /= c)
```

The network controllers must be instrumented to send messages about their policy-relevant actions. Here, the relevant actions are (a) when a controller initiated an election of the master controller, and (b) when a controller starts acting as the master controller. In the latter case, a controller would send a message of the form

```
1461144315.2435@[ONOS:199.37.70.30] (346): start acting as master
```

where the first decimal number is the timestamp of the performed action, the text in square brackets specifies the sender, the number in parentheses is the sequence number of the sender, and the text after the double column describes the action. Such a message is interpreted by the compliance checker according to the given configuration file. For example, with the rule

```
[component matches "ONOS:(?P<election_outcome>[0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)",
 event matches "start acting as master"]
=>
{MasterController(<election_outcome>)}
```

the compliance checker extracts the information that the predicate MasterController(ONOS:199.37.70.30) is true at (Unix) time 1461144315.2435 from the above message.

6.3.11 Re-utilized technologies/specifications

The compliance checker is based on an extension of the monitoring approach described by Basin et al. [2015]. The extension comprises an online algorithm that handles a richer policy specification language as described in Section **Error! Reference source not found.**, and will be described and analysed in detail in a forthcoming paper [Basin et al., 2017].

6.3.12 References

[1] 5G-Ensure Consortium, Deliverable 2.1 Use Cases, [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.1-UseCases.pdf, 2016.

[2] 5G-Ensure Consortium, Deliverable 2.4 Security Architecture (draft), [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.4-SecurityArchitectureDraft.pdf, 2016.

[3] R. Alur and T. A. Henzinger. Logics and models of real time: A survey. In Proceedings of the 1991 REX Workshop on Real Time: Theory in Practice, volume 600 of Lect. Notes Comput. Sci., pages 74–106. Springer, 1992.
[4] C. Baier and J.-P. Katoen. Principles of model checking. MIT Press, 2008.
[5] D. Basin, F. Klaedtke, and E. Zalinescu. Monitoring metric first-order temporal properties. J. ACM, 62(2): 15, 2015.
[6] D. Basin, F. Klaedtke, and E. Zalinescu. Failure-aware runtime verification of distributed systems. In Proceedings of the 35th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science. Leibniz International Proceedings in Informatics (LIPIcs), volume 45, pages 590—603, 2015.
[7] D. Basin, F. Klaedtke, and E. Zalinescu. Runtime verification of temporal properties over out-of-order data streams. Manuscript. Submitted for publication, 2017.
[8] Docker. Build, Ship, and Run Any App, Anywhere. https://www.docker.com/.
[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling innovation in campus networks. SIGCOMM Computer Communication Review, 38(2):69–74, 2008.
[10] D. L. Mills. Improved algorithms for synchronizing computer network clocks. IEEE/ACM Trans. Netw., 3(3):245–254, 1995.
[11] Network Time Protocol (NTP). http://www.ntp.org/.
[12] Open Networking Foundation (ONF). OpenFlow switch specification – version 1.3.0 (wire protocol 0x04). 2012.
[13] C. Scott, A. Wundsam, B. Raghavan, A. Panda, A. Or, J. Lai, E. Huang, Z. Liu, A. El-Hassany, S. Whitlock, H.B. Acharya, K. Zarifis, and S. Shenker. Troubleshooting blackbox SDN control software with minimal causal sequences. In Proceedings of the ACM SIGCOMM 2014 Conference. ACM Press, 2014.

6.4 Micro-segmentation enabler: open specification

6.4.1 Preface

The upcoming 5G networks are currently being designed and the general vision is that 5G will be software-defined and virtual. There will be a new mobile ecosystem that consists of heterogeneous services, applications, networks, users and devices.

The security of 5G will be critical, as multiple different players will be included in the 5G ecosystem, such as Massive Machine-Type Communications (mMTC), Machine to Machine (M2M) or Industrial Internet based companies. In the case of a cyber-attack, the consequences could be dramatic.

Consequently, the role of isolation, virtualization and network management is going to be important. Applications or services requiring high level of security need to be clearly isolated and secure from the rest of the network. Network slicing has been introduced before to provide network isolation and this work presents the concept of micro-segmentation into 5G network security.

This section presents ‘Micro-segmentation’ enabler. The enabler is related to two other 5G-ENSURE security enablers, namely ‘Trust Metric’ and ‘Security Monitor for 5G Micro-Segments’.

6.4.2 Status

These are preliminary open specifications of Micro-segmentation Enabler focusing on enabler in release two (R2).

6.4.3 Copyright

6.4.4 Legal notice

The Legal notice that applies to these specifications is given in Annex A.

6.4.5 Terms and definitions

AAA	Authentication, Authorization, Accounting
IoT	Internet of Things
LTE	Long Term Evolution
MME	Mobile Management Entity
NFV	Network Function Virtualization
PCRF	Policy Control Resource Function
PGW	Packet Data Network Gateway
R2	Release two
SGW	Serving Gateway
SLA	Service Level Agreement
VMNO	Virtual Mobile Network Operator

6.4.6 Overview

The micro-segmentation enabler describes how network virtualization can be used to secure the network, users, and their traffic effectively against cyber-attacks. The enabler allows creating secure segments into the 5G network in which granular access controls and strict security policies can be enforced.

Network slicing [1, 2, 3, 4] is said to be a key component of 5G networks. Network slices provide a dedicated, logical network for a specific application, such as IoT. In other words, they provide all necessary functionality for a service and basic isolation. All other functionality is thus avoided in order to minimize the internal complexity of the slice. In general, it can be seen that network slices provide basic network security.

Micro-segments[12] can be viewed as a further sub-partition of slices into parts with more specific security properties. They are meant to complement network slicing by providing customizable security settings that can be modified to fit the purpose of the used service or application. In other words, they are an extra option for network slices. Micro-segmentation can be a cost-effective solution for the mobile network security control as methods such as next generation firewalls and reactive counter-measures based on complex event processing may not be practical in large scope. By the use of micro-segmentation, the threat landscape can be thus minimized.

Generally, there are three entities regarding the micro-segmentation concept: micro-segment provider, micro-segment subscriber, and security inference provider. Micro-segment provider is an entity who controls the switches and SDN controllers in micro-segments and collects event information, e.g. network statistics. This actor is typically the Mobile Network Virtual Operator (MVNO).

Security inference provider is an entity that acquires security information from micro-segments through the APIs of the micro-segment provider. This information is used for analytics and controlling the micro-segment. The actor can be a micro-segment provider, a third-party, or the micro-segment subscriber.

[12] Another term that could be used is a sub-slice (i.e. a small network slice) or a secure network slice. In this document we use the term micro-segments, used previously in data centres, to put more emphasis on network security.

Micro-segment subscriber is an organization, company requiring isolated 5G network services (and monitoring) for a particular application.

Figure 6-10 shows an example of the micro-segmentation approach in a single domain (single operator) that could be built on top of existing 4G architecture. Network slices and micro-segments are created by the use of virtualization. For example, there could be one general network slice for "IoT", but two micro-segments for "smart metering" and "personal health". The subscriber of a micro-segment is typically an organization, service provider or a Virtual Mobile Network Operator (VMNO). The overall control of the micro-segments would be by (virtual) operators. The organizations and service providers that use the micro-segments may also have some control, especially related to the security functionalities within the micro-segment. Individual end-users would not have control over a micro-segment. Within a single domain, the segments should typically lay within a single network slice.

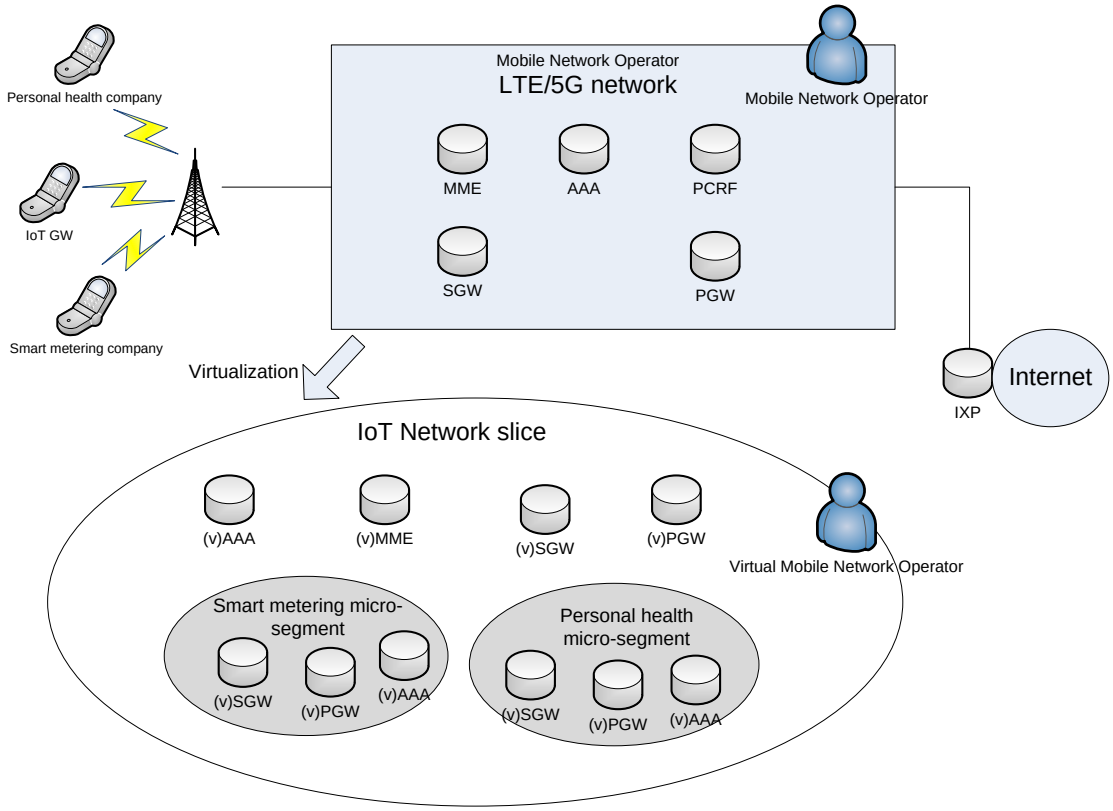


Figure 6-10 Micro-segmentation in a single domain network

6.4.7 Basic concepts

The specific components that are included in a network slice or micro-segment depend essentially on the used application or service. What kind of functionality from the mobile core network is needed to run the service or application? By discarding unneeded functionality, the complexity of a micro-segment can be reduced.

In a 4G LTE network, the minimum requirements could be to include virtualized instances of both the Serving Gateway (SGW) and the Policy Control Resource Function (PCRF) in a network slice or micro-segment. For applications or services requiring Internet access, the network slice or micro-segment should include also the Mobile Management Entity (MME) and the Packet Data Network Gateway (PGW). Each network slice and micro-segment could also have its own AAA entity, but the AAA functionality of a micro-

segment should not be too heavy to avoid the complexity. All these components would be virtualized resources or functions, i.e., Network Function Virtualization (NFV) would be used.

A micro-segment instance is, however, not necessarily required to form a complete logical network. Typically, some parts of the functionalities of network slices can be inherited by micro-segments and a single micro-segment instance includes only a part of the subscribers of a network slice.

For instance, a massive Internet of Things (IoT) service may not necessarily require features such as handover or location update, which are being used with mobile devices. This is because the IoT service connects with a large number of immobile sensors that measure different parameters, such as humidity, precipitation, etc. and mobility is thus not considered. However, security for the service is critical.

Micro-segmentation needs to take into account different trust models for different micro-segments. Some micro-segments may require a Zero Trust model, which states that in the network there are no trusted zones, but instead all network traffic should be monitored and logged. In this model, all nodes are considered as potential threats. There is no default trust level for any entity or object in the network. In an SDN network, the Zero Trust model could be achieved by having the SDN controller inspect each packet that arrives into the network.

The Zero Trust model, however, brings more computational complexity to security monitoring. Consequently, such a trust model can be, e.g., provided to micro-segments with critical services. Such a case could be an authority network in a crisis situation, in which trust would not be self-evident and the micro-segment should be highly secure. Other possible use case includes networks related to critical infrastructures, such as electricity distribution.

6.4.8 Main interactions

6.4.8.1 Use cases

Potential customers for the micro-segmentation approach in 5G networks include hospitals, factories, Industrial Internet and IoT based companies. An IoT company that is using the 5G network needs to gather data from different sensors reliably and securely. The delay may not need to be small but security is of high concern. Hospitals would also benefit from a dedicated micro-segment from the 5G network that is highly secure since e.g. gathering patient information from various sensors needs to be extremely private. Another possible use case is IoT based video monitoring and remote surgery.

6.4.8.2 Components and interaction overview

In release 1 (R1), the enabler was functioning independently without a connection to the Trust Metric and Security Monitor for 5G Micro-Segments enablers. This means that the enabler was able to create micro-segments, delete micro-segments, add nodes to micro-segment, and delete micro-segments without control from the Security Monitor for 5G Micro-Segments enabler.

In release 2 (R2), the enabler is integrated with the Trust Metric and Security Monitor for 5G Micro-Segments enablers. **Figure 6-11** shows the architectural diagram of the micro-segmentation approach inside a single administrative domain. In the figure, the micro-segmentation enabler is used for creating and deleting micro-segments dynamically. The monitoring framework gathers information and metrics from various sources, such as mobile network functions (MME, P-GW, S-GW, etc.), IoT gateway and eNodeB. This information is fed to the security inferencing (i.e., Security Monitor for 5G Micro-Segments) and Trust

Metric enabler modules. Based on this information, the security inferencing module will generate actions for the micro-segmentation enabler to execute.

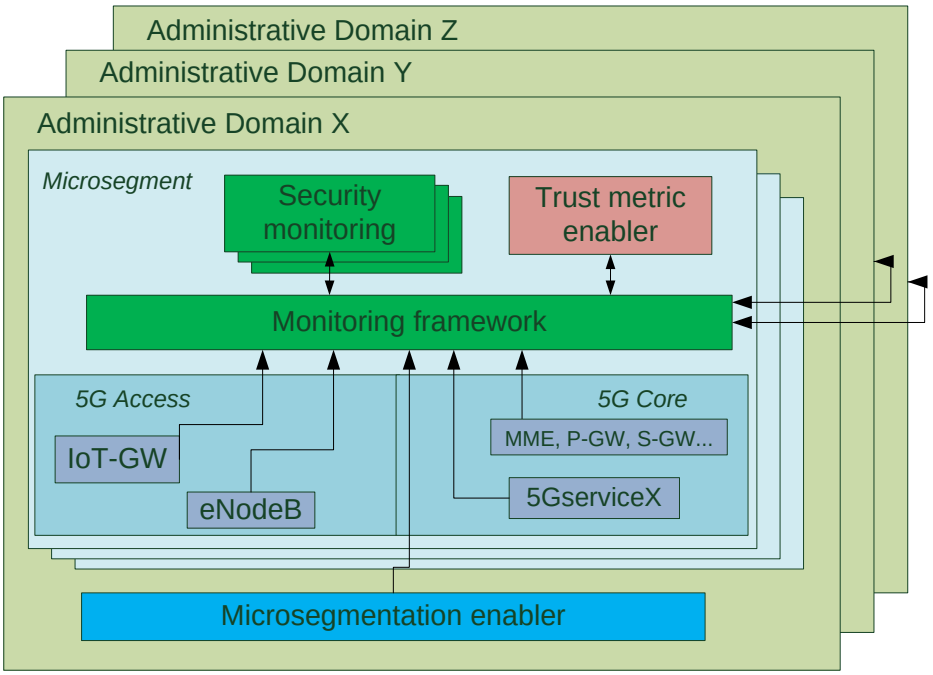


Figure 6-11 Architectural diagram of micro-segmentation approach inside a single administrative domain

Figure 6-12 shows the architectural diagram of the micro-segmentation enabler. As can be seen from the figure, the enabler will be connected to a Micro-segment Apache Kafka feeder module that retrieves network topology and network traffic information via the northbound interface of the enabler. This information is sent in WebSocket format. Subsequently, this information is sent to the Event broker (Apache Kafka) in JSON format. The event broker distributes the information to the Trust Metric and Security Monitoring enablers. Based on this information, the Security Monitoring enabler will generate control actions to the micro-segmentation enabler in REST format. The trust metric enabler also needs information about micro-segments to calculate a trust metric value for a micro-segment.

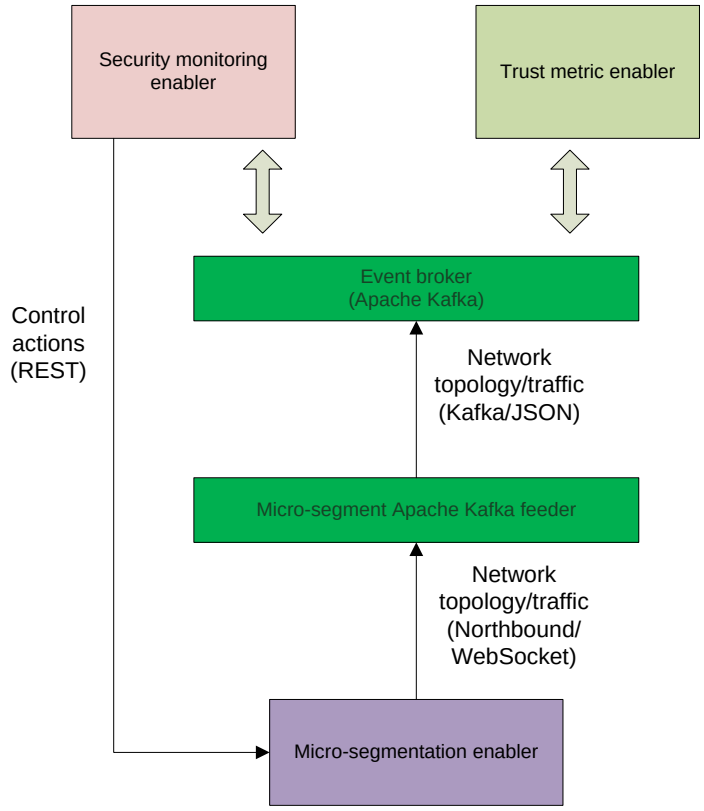


Figure 6-12 Architectural diagram of the micro-segmentation enabler

Figure 6-13 depicts an example how the micro-segmentation concept would realize in an SDN network. The physical topology consists of two micro-segment subscribers, i.e. organizations or companies, connected to SDN switches and a SDN virtualization controller. The micro-segments are virtual networks that consist of a virtual SDN controller and several virtual SDN switches. The physical users are connected to one micro-segment while the physical SDN switches may be connected to both micro-segments. The virtual SDN controller is responsible for controlling the traffic inside the micro-segment and the SDN virtualization controller used for creating the two micro-segments A and B by the use of OpenVirteX [5] virtualization software. The virtual SDN controllers may hold applications for AAA and Security monitoring. Each micro-segment thus can have its own AAA entity for authenticating and authorizing users and accounting.

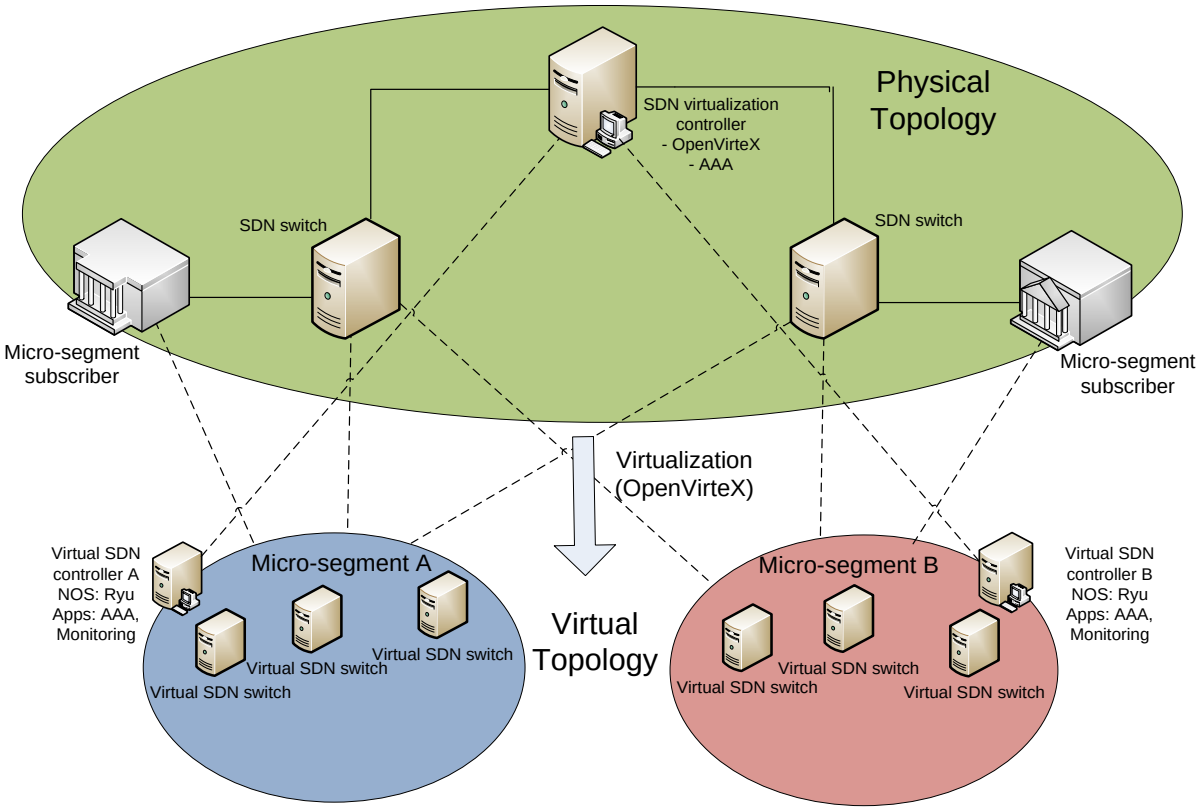


Figure 6-13 Micro-segmentation in an SDN network

6.4.9 Architectural drivers

6.4.9.1 High-level functional requirements

The enabler will in general create and delete micro-segments, which are isolated parts of the 5G network. In other words, micro-segments are virtualized, logical instantiations of the network that are highly isolated and secure.

Micro-segmentation will in effect provide a more homogeneous and smaller environment for security monitoring in order to respond better to abnormal behavior or attacks. Consequently, the threat landscape becomes smaller.

6.4.9.2 Quality attributes

When using the micro-segmentation enabler, it should be able to provide up-to-date information to the Security monitoring and Trust Metric enablers in order to do appropriate decisions. Also, when deployed in the 5G network, the enabler should not degrade the performance of the running applications or services in the micro-segment, i.e., the Service Level Agreements (SLAs) should be guaranteed.

6.4.9.3 Technical constraints

One of the challenges is how the micro-segmentation approach could be integrated into the 5G mobile network architecture and this may be partly addressed in Task 2.4: Architecture of the 5G-Ensure project. Notable questions include, for example, what network functions of the 5G architecture can be included in a micro-segment? Are there any limitations in virtualizing mobile network functions? Should micro-segments have common network functions? Essentially, some of the questions may be addressed in this project but some might be left for future work.

6.4.9.4 Business constraints

The enabler is built on existing open source software that can be quite easily integrated together. This facilitates the applicability of the solution to other environments.

6.4.9.5 Link to security architecture

Figure 6-14 and **Figure 6-15** show the relevant domains and strata (marked by red and green dots) of the 5G-ENSURE security architecture. In **Figure 6-14**, the enabler provides protection and security to the serving network domain (marked by a red dot). Also, the enabler has an impact on the IP and management domains (marked by green dots). This means that, for example, switches need to be configured and micro-segments need to be set up.

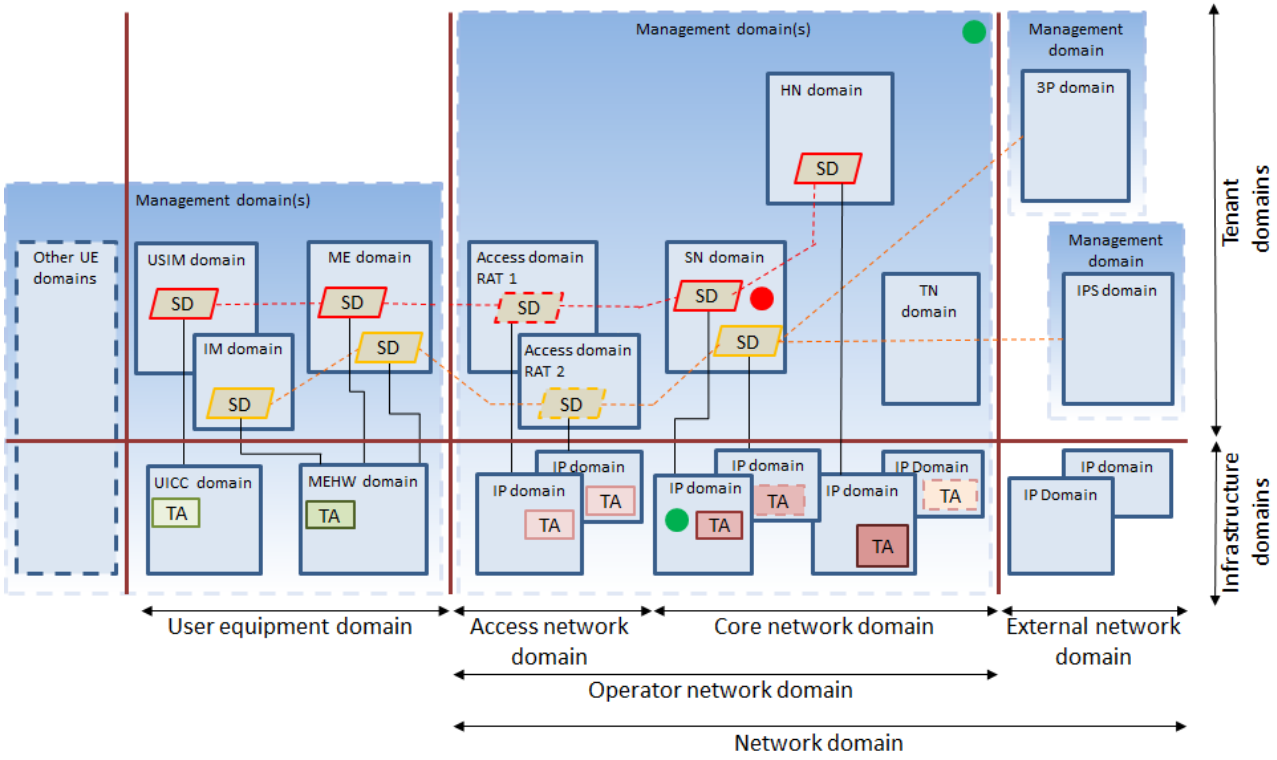


Figure 6-14 Enabler relevant domains

In **Figure 6-15** a red dot is shown in the Serving stratum. This means that the enabler protects user communications that are involved in the stratum. The enabler also has an impact on the Management and Application strata (marked by green dots). In the management stratum, communication mechanisms, i.e., micro-segments are set up. In the application stratum, a user can request for the creation of micro-segments.

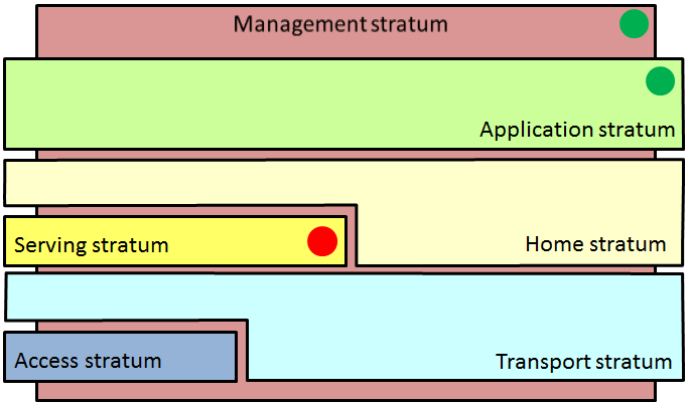


Figure 6-15 Enabler relevant strata

6.4.10 Detailed specification

6.4.10.1 Introduction

In order to implement micro-segmentation, SDN and virtualization technologies are needed. **Figure 6-13** showed the micro-segmentation concept in an SDN network that showed how micro-segments are in general created. There needs to be an SDN virtualization controller that is able to create virtual networks that are isolated from the physical network. In our scenario, the virtualization is done by OpenVirteX software [5], but other network virtualization hypervisors are possible. A detailed review of different hypervisors can be found here [6]. The SDN virtualization controller may also have its own AAA application.

For controlling the traffic inside the micro-segment, a virtual SDN controller is needed. In our case, we are using the Ryu SDN controller [7], but other type of SDN controller is plausible. The virtual SDN controller needs to have SDN applications for AAA and security monitoring. The AAA application will be used for authenticating and authorizing users and monitoring user statistics, i.e., accounting. In the open specification, the authentication is based on Extensible Authentication Protocol over LAN (EAPoL). Other authentication methods are naturally plausible.

Micro-segments may support and require particular authentication mechanisms. For example, if the mobile device has been authenticated strongly to the mobile network by the use of USIM, it can be authorized to use the micro-segment. However, if lighter authentication methods have been used in the mobile network, the device needs to be authenticated using stronger or second-factor mechanism to authorize use of the micro-segment. The security monitoring application will be used for monitoring traffic inside the micro-segment and responding to threats, attacks, and anomaly behaviour. The monitoring enabler is a separate entity from the SDN controller and it tries to gather information from a wide perspective in addition to the network traffic information that can be obtained through the SDN controller.

In order to test micro-segmentation with mobile networking architecture, there needs to be a way how to integrate the different functions of mobile networks (e.g. PGW, SGW, MME, PCRF) with SDN and network virtualization technologies. One possible solution could be to use the OpenEPC framework [8] that covers all the functional elements in the 3GPP Evolved Packet Core (EPC) and Systems Architecture Evolution (SAE) technologies up to 3GPP Release 12. Another alternative is the OpenAirInterface [9].

6.4.10.2 Conformance

The enabler needs SDN and virtualization technologies for implementation and these two are the building blocks of the enabler. For SDN, an SDN controller is needed that can be chosen quite freely. In this work, we

are currently using the Ryu SDN controller [7]. For network virtualization, OpenVirteX software [5] is used, but another type of network hypervisor is plausible.

For AAA methods, the selection depends on the used application. If a high level of security is needed, then perhaps a strong authentication method is in place. The AAA methods can be therefore chosen quite freely.

6.4.10.3 API specification

Figure 6-12 showed the architectural diagram of the micro-segmentation enabler, which could receive the following control actions from the Security monitoring enabler:

- ***Modify micro-segment topology***. Modification of the micro-segment can be done by adding and deleting nodes from the micro-segment. Micro-segment can also be deleted. It could be also possible to export micro-segment topology information.
- ***Modify micro-segment security settings***. This includes modifying AAA, trust, monitoring and privacy settings.

The micro-segment provider can generate the following control action:

- ***Create micro-segment based on the needed security level***. The security level can depend on the following entities: AAA (e.g. how are the end users authenticated?), security monitoring (e.g. what kind of monitoring is needed?), anomaly detection (e.g. what kind of anomaly detection is needed?), intrusion detection (what kind of intrusion detection is needed?), and prevention mechanisms (e.g. what kind of prevention mechanisms are needed? E.g. Deep Packet Inspection?). Consequently, the definition of a security level depends on multiple factors and it could be worthwhile to investigate how to take advantage of the other enablers developed in the project. Another question is to what extent the security level can be defined.

The enabler can produce the following outputs to the monitoring framework:

- Information about micro-segments.
- Information about network traffic.
- Information about network topology.

The specific technical details of the API depend on the selection of the network hypervisor and the SDN controller. For example, in this work OpenVirteX is used as the network hypervisor that does the network virtualization and Ryu SDN framework is used as an SDN controller. Apache Kafka [10] could be used as an API that produces events coming from the enabler while REST API could be the way how commands are sent to the enabler. Information needed from the Ryu SDN controller can be retrieved with WebSocket protocol.

In release 1 of the micro-segmentation enabler, creation of micro-segments, deleting micro-segments, adding nodes to micro-segment, and deleting nodes from micro-segment was implemented. Release 2 includes integration with the Trust Metric and Security monitoring for 5G Micro-Segments enablers.

6.4.10.3.1 Functions, messages and commands

6.4.10.3.1.1 Event distribution layer

Events are presented as {“topic”: “content”} tuples. The structure of the content depends on the event publisher. The primary format for content that the micro-segmentation enabler produces to the Micro-segment Apache Kafka feeder (see **Figure 6-12**) is WebSocket. The feeder will consequently publish the information in JSON format to the Event broker.

In the following subsection, we will specify some essential content formats.

The naming of "topics" follows the following format: [<microsegment_identifier>.<event_identifier>.<optional event parameters>], where event identifiers are unique for the micro-segment.

6.4.10.3.1.2 Micro-segment events

The micro-segmentation enabler publishes mainly SDN related configuration and network traffic information, which is used by with the Trust Metric and Security monitoring for 5G Micro-Segments enablers.

Specifically, the Micro-Segmentation Enabler publishes meta-data on the micro-segment, particularly topology and node configuration data. Broker topics for published metadata events are the following: [<micro-segment identifier>.<information_type>] where information types include e.g. "controller configuration", "topology", and (for switch specific configuration) <switch_id>."config".

Each time a micro-segment changes, the relevant event / configuration information is published. These change events are triggered e.g. by "create", "modify topology", and "modify settings" commands, which are made by micro-segment subscriber. Information requests may also trigger additional information to be published. One goal of the Security Monitor for 5G Micro-Segments enabler (Section 5.3) is to gather information from micro-segments and export this information to other security enablers. The Generic Collector Interface Enabler (Section 5.2) could be used as the middleware to achieve this.

topic: [<micro-segment identifier>.<switch_id>.config]
{ "mfr_desc": "manufacturer", "hw_desc": "hardware description", "sw_desc": "software description", "serial_num": "serial number", "desc": "Human readable description" }

Network statistics includes data on how much traffic has flown trough particular paths in during the monitoring period (since last statistic event was published).

topic: [<micro-segment identifier>.flowstats]
{"src": "ip_address", "dst": "ip_address", "packet" : "count", "bytes" : "bytes transferred"}

6.4.10.3.2 Signalling

All real-time signalling between Micro-segmentation, Security monitoring for 5G Micro-Segments, and Trust metric enablers components occurs through the broker interface and micro-segment Apache Kafka feeder that was shown in **Error! Reference source not found.**.

6.4.10.4 Examples

Micro-segments can be provided by Mobile Virtual Network Operators (MVNOs) to subscribers, such as hospitals, factories, Industrial Internet and Internet of Things (IoT) based companies. Initially, the micro-segment subscriber informs the MVNO that a micro-segment with a certain security level is needed. The security level can depend on the used AAA methods, type of security monitoring, privacy, and trust.

Ultimately, micro-segments are customizable. Subsequently the micro-segment is created based on the subscriber input. A micro-segment created – event is sent to the security framework module to inform of a new micro-segment in the system.

The security monitoring enabler gathers information from the micro-segmentation enabler about micro-segments, possibly via Apache Kafka API. This information is gathered via the SDN controller of a micro-segment through WebSocket protocol. The controller sees all the switches and nodes in the micro-segment, micro-segment topology, and status information about amount of packet/bit network traffic.

The security monitoring enabler also provides actions to the micro-segment enabler to modify micro-segment topology or modify micro-segment security settings. This could be done via REST API. Modifying the topology can happen when nodes need to be removed or added to the micro-segment or micro-segments need to be deleted, split or merged. Deleting a micro-segment can happen when the subscribers are no longer using the micro-segment. All of these events are subsequently provided to the security monitoring and trust metric enablers to handle up to date information about micro-segments and nodes in micro-segments.

6.4.11 Re-utilised technologies/specifications

The enabler uses the OpenVirteX software for network virtualization and the Ryu SDN controller for management of micro-segments. Both of these are modified to fit the purpose of the enabler.

6.4.12 References

[1] Ericsson, “Network functions virtualization and software management,” 2014. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/network-functions-virtualization-and-software-management.pdf.

[2] Ericsson, “5G systems - Enabling Industry and Society Transformation,” 2015. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/what-is-a-5g-system.pdf.

[3] Ericsson, “5G Security - Scenarios and Solutions,” June 2015. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/wp-5g-security.pdf.

[4] NGMN Alliance, “Description of Network Slicing Concept,” 2016.

[5] “OpenVirteX Network Virtualization Platform,” [Online]. Available: http://ovx.onlab.us/.

[6] A. Blenk, A. Basta, M. Reisslein and W. Kellerer, “Survey on network virtualization hypervisors for software defined networking,” IEEE Communications Surveys & Tutorials, vol. in print, 2015.

[7] “Ryu SDN framework,” [Online]. Available: https://osrg.github.io/ryu.

[8] “OpenEPC framework,” [Online]. Available: http://www.openepc.com.

[9] “OpenAirInterface,” [Online]. Available: http://www.openairinterface.org.

[10] “Apache Kafka,” [Online]. Available: http://kafka.apache.org/.

6.4.13 Acknowledgements

Acknowledgements to Kimmo Ahola, Pekka Ruuska, Mikko Uitto, Mikko Majanen, Jani Suomalainen, Felix Klaedtke and Nicolae Paladi.

6.5 Bootstrapping trust enabler: open specification

6.5.1 Preface

The physical infrastructure supporting 5G networks will be multiplexed among multiple tenants with own environments, similar to the cloud infrastructure model. In this model, SDN allows tenants to configure complex topologies with rich network functionality, managed by a network controller [ONF, 2012]. The availability of a global view of the forwarding plane enables advanced controller capabilities – from pre-calculating optimized traffic routing to managing software applications that replace hardware middleboxes with virtual network functions (VNFs). However, these capabilities also turn the controller into a valuable attack target: once compromised, it can provide the adversary with complete control over the network [Porras 2015]. Furthermore, the global view itself is security sensitive: an adversary capable of impersonating network components – such as virtual switches or VNFs -- may distort the network controller's view of the forwarding plane or influence the network-wide routing policies. Virtual network components – such as virtual switches and VNFs – are security sensitive elements in SDN deployments. They operate on commodity operating systems (OS) and are often assigned the same trust level and privileges as physical switches or middleboxes – specialized, hardware components with compact embedded software. Commodity OS with large code bases are likely to contain multiple security flaws which can be exploited to compromise virtual network components. Such risks are accentuated by the extensive control a cloud provider has over the compute, storage and networking infrastructure of its tenants. It is therefore essential to establish trust in the virtual network components prior to enrolling them into the network infrastructure [Bursell et al, 2014].

This enabler aims to reduce the risk to the integrity of the virtual switches and VNFs enrolled in the SDN deployment, as well as to ensure the integrity and confidentiality of the communication between SDN components and the network controller.

6.5.2 Status

An enabler prototype is under development. This is a preliminary specification and is subject to changes.

6.5.3 Copyright

The enabler is owned by SICS. SICS is currently the sole contributor to this enabler.

6.5.4 Legal notice

The Legal Notice that applies to this specification is given in Annex A.

6.5.5 Terms and definitions

OpenFlow	Protocol standardized by the Open Networking Foundation (ONF) for controller-switch communication.
SDN	Software-Define Networking
SDK	Software Development Kit
API	Application Programming Interface
TEE	Trusted Execution Environment
TCB	Trusted Computing Base
TLS	Transport Layer Security
VNF	Virtual Network Function
FIB	Forwarding Information Base

6.5.6 Overview

The *Bootstrapping Trust* enabler allows tenants to securely deploy SDN components (virtual switches, VNFs) supported by TEEs and set up virtualized network topologies in a cloud environment, under the adversary model described in [Paladi et al, 2016; Paladi et al, 2015]. Furthermore, the enabler allows tenants to establish trust in the topology of the SDN deployment using remote attestation and enrolment procedures. Thus, only virtual switch and VNF instances supported by TEEs and attested by the network controller are considered enrolled in the topology of the SDN deployment. Following the enrolment, all communication between the network controller and enrolled components is confidentiality and integrity protected.

6.5.7 Basic concepts

The main principles of this enabler are presented in **Figure 6-16** and described below.

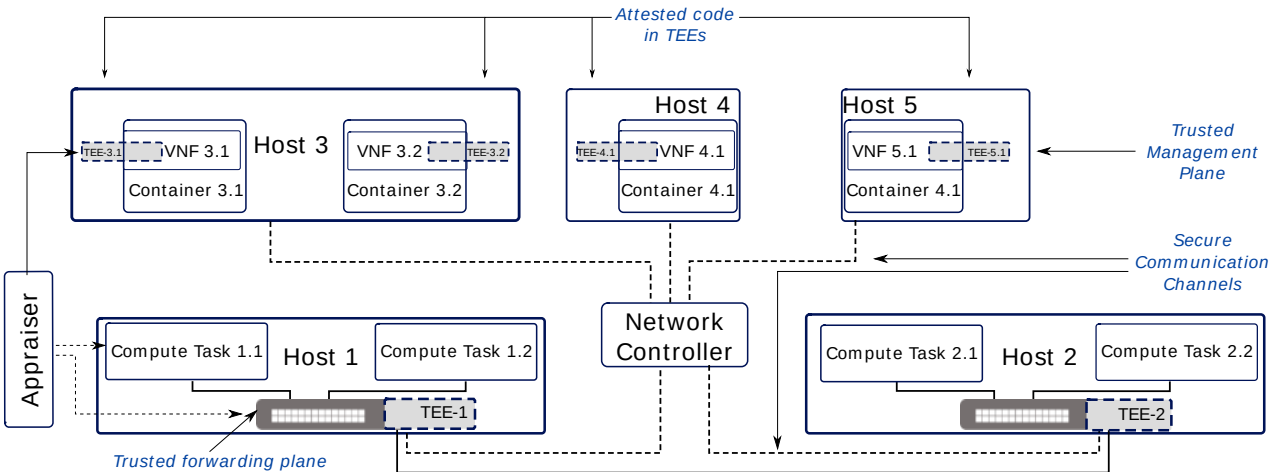


Figure 6-16 Principles of the Bootstrapping Trust enabler

Trusted Execution Environments (TEEs) guarantee isolated execution in the given adversary model, assuming correct implementation of the trusted computing base (TCB), e.g. the CPU and executed code. The TEE can be located on the same platform or on other platforms within the deployment.

Attested code and data in TEEs: Integrity of the code and data deployed in the TEEs is attested before any keys or key material is provisioned to the respective TEE. An appraiser under the control of the tenant performs the attestation of the TEE [Cocker et al, 2011].

Trusted Forwarding Plane: The integrity of the platform TCB, as well as the virtual switches is attested by an appraiser. Only verified virtual switches are allowed to communicate with the network controller. Authentication keys and other confidentiality and integrity sensitive cryptographic material is only stored in the TEE and never leaves its security perimeter.

Trusted Management Plane: The integrity of the platform TCB, as well as the candidate VNFs is attested by an appraiser. Only verified VNFs are installed on the deployment. Authentication keys and other confidentiality and integrity sensitive cryptographic material is only stored in the TEE and never leaves its security perimeter.

Secure Communication Channels: The enabler protects communication channels between compute tasks, as well as communication channels between network components and the network controller.

Communication security is ensured using confidentiality and integrity protection keys provisioned to TEEs and only used by the authenticated network components and endpoints.

6.5.8 Main interactions

6.5.8.1 Use cases

In this section, we describe two typical use cases for the *Bootstrapping trust* enabler. Use cases are described in the "fully-dressed" format [Cockburn, 2001].

ID	BT-1
Title	Attest integrity of a virtual network component in the SDN deployment
Description	Administrator obtains a quote of the virtual network component TCB, verifies its authenticity and verifies that it matches the expected values.
Primary Actor	Administrator
Preconditions	The TCB of the network component is measured in a chain of trust originating in a hardware root of trust and reliably stored in an enclave, i.e. an isolated execution environment.
Post-condition	Administrator has obtained a statement of whether the virtual network component TCB measurements match the expected values
Main success scenario	1. Administrator requests a quote of the virtual network component TCB 2. Enclave on the virtual network component platform produces the quote 3. The "Quoting Enclave" (QE) on the respective platform signs the quote and returns to the administrator. 4. Administrator verifies quote signature against the known public key of the host QE 5. Administrator matches quote against expected values for the virtual switch.
Extensions	4a1. The signature verification step shows that the quote signature is invalid. 4a2. Administrator either retries operation or excludes the virtual network component on the respective host from the list of virtual network components in the deployment.
Frequency of Use	At each deployment of the SDN infrastructure or as required for audit purposes.
Status	Implementation on-going
Owner	SICS

ID	BT-2
Title	Enrol virtual network component into SDN deployment
Description	Once the virtual network component TCB has been verified, the verification manager provisions to the virtual network component enclave the necessary key material to establish a protected communication channel with the network controller and updates the list of virtual network components in the deployment.
Primary Actor	Software Verification Manager
Preconditions	Integrity of the virtual network component has been verified and a verified enclave is present on the platform.
Postcondition	The virtual network component has established a secure communication channel to the network controller using a private key stored in a verified enclave.

Main success scenario	1. *Verification Manager* locates the virtual network component enclave identity and its public key in the list of attested virtual network component enclaves. 2. *Verification Manager* generates an "enrolment message" containing necessary certificates and a pre-shared key. 3. *Verification Manager* confidentiality and integrity protects the enrolment message using the public key of the virtual network component enclave. 4. *Verification Manager* provisions enrolment message to virtual network component enclave. 5. Virtual network component enclave uses contents of the enrolment message to establish a TLS session with the network controller.
Extensions	None
Frequency of Use	At each deployment of the virtual network component.
Status	Implementation on-going
Owner	SICS

6.5.8.2 Components and interaction overview

In this section, we describe the high-level overview of the enabler components and their interaction.

Figure 6-17 illustrates the interaction between the main actors in the *Bootstrapping Trust enabler*. The subjects of the interaction are the Administrator and the *Verification Manager*, the signature verification service and the application fingerprint store. The subjects of the interaction are the network deployment components – the virtual switches on the hosts within the deployment and the SDN network controller, as well as VNF instances (omitted from the illustration for clarity). The below description follows the numbered steps illustrated in **Figure 6-17**.

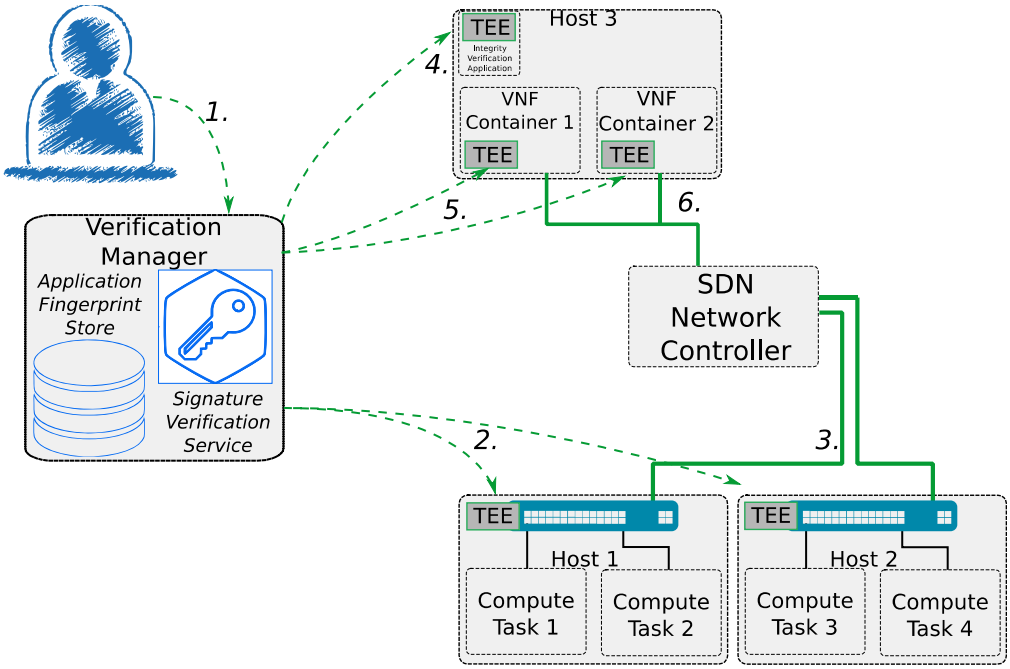


Figure 6-17 High-level components and interactions

As a prerequisite to this workflow, we assume that all necessary components on the network infrastructure (verification manager, VNFs, network controller, virtual switches and secure enclaves) are deployed.

In step (1), the *Administrator* initiates the verification of the network infrastructure components, which is performed by the *Verification Manager*. Next, the verification manager proceeds to verifying the virtual network component enclaves.

In step (2) the *Verification Manager* attests the integrity of the virtual network component enclaves on the virtual switch hosts. This involves first appraisal of the integrity of the code and data in the enclaves, as well as the appraisal of the TCB of the virtual switches (such as binaries, configuration and other ancillary data – as defined by a deployment-specific policy). The deployed enclave attests the TCB of the virtual switch as reported by the Linux Integrity Measurement Architecture. A *quoting enclave* signs the quote prior to being returning it to the *Verification Manager*. This signature is required in order to prove – while presenting the privacy of the platform – that the enclave is executing on a genuine platform capable to create a TEE (we assume in this specification that the TEE is implemented using Intel SGX [Anati et al, 2013]). The *Verification Manager* (which fulfils the role of an appraiser in this case) verifies the signature of the quote message, and then attests (matches) the quote message against an application fingerprint store.

The *Verification Manager* verifies the quote as follows: it verifies the signature of the quote message, then attests (matches) the quote message against an application fingerprint store. The attestation succeeds if the quote message matches the respective entry in the application fingerprint store.

If the attestation of the integrity of the virtual network component enclaves succeeds, the *Verification Manager* provisions the attested enclaves with a public certificate of the Network Controller as well as a signed certificate with credentials to be used by the virtual switches for authentication to the Network Controller. Note that the signed certificate with credentials contains security sensitive information and is hence provisioned over a confidentiality and integrity protected channel established using the functionality of the TEE.

In step (3), the *Verification manager* repeats the same sequence of steps in order to attest the integrity of the VNFs deployed in the infrastructure and provisions authentication credentials to the respective enclaves.

Steps (5) and (6) describe respectively the communication between virtual switches and the network controller and the communication between the VNFs and the network controller. In both cases, the communication is conducted over TLS (or other supported protocols for communication protection). The computed integrity and confidentiality protection keys are maintained ***exclusively*** in the respective enclaves and ***never*** leave their security perimeter, even in the event of an operating system compromise (assuming physical integrity and excluding side-channel attacks).

In this scenario, it is critical that the signature verification service is reliable enough to detect attempts to modify or forge the quote messages; it is also essential to ensure the integrity of the application fingerprint store. However, these aspects are out of the scope of *Bootstrapping trust* enabler.

6.5.9 Architectural drivers

6.5.9.1 High-level functional requirements

Authentication All communication between control plane components must the authenticated; a secure enrolment mechanism for management applications and forwarding plane components must be in place.

Topology integrity: The network controller must be protected from network components that attempt to distort the visible global network view. This applies both to centralized and distributed network controllers.

Component integrity: Integrity of forwarding plane components must be verified prior to enrolment; the cryptographic material required for their network access must be protected with a hardware root of trust.

Confidentiality protection of domain secrets: Network domain secrets – such as VPN session keys – should not be revealed in plaintext even if the adversary succeeds in compromising the software stack on the host.

Confidentiality and integrity of network communication: All network communication in the tenant domain must be confidentiality and integrity protected.

6.5.9.2 Link to security architecture

Figure 6-18 illustrates (marked by a red dot) the most relevant domains of the 5G-ENSURE security architecture with respect to the *Bootstrapping Trust* enabler. Considering that the goals of the *Bootstrapping Trust* enabler are to allow mutual trustworthiness assessment and facilitate auditing of components, the enabler targets elements in the management domain, namely the serving network domain, home network domain and transit network domain. Note however that the enabler functionality can also be applied for appraisal of components in other domains.

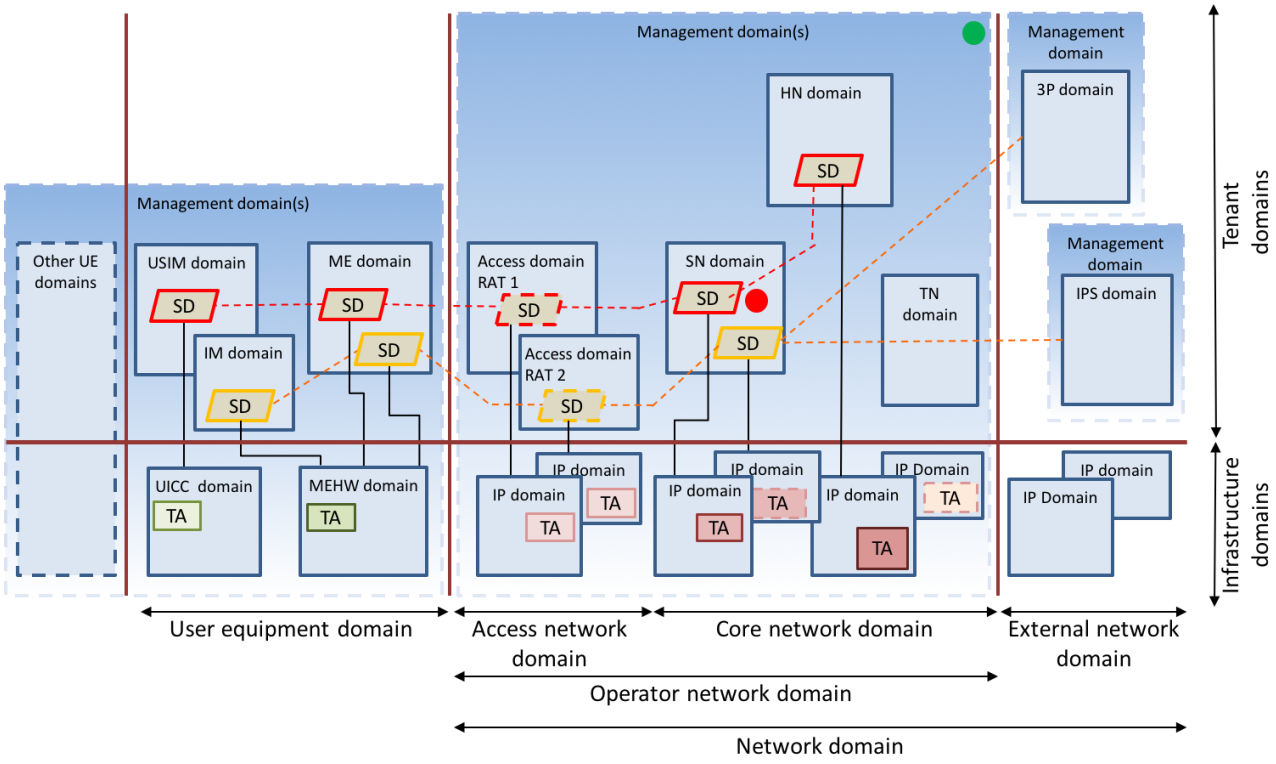


Figure 6-18 Enabler relevant domains

Figure 6-19 illustrates (marked with a red dot) the strata of the 5G security architecture relevant to the *Bootstrapping Trust* enabler.

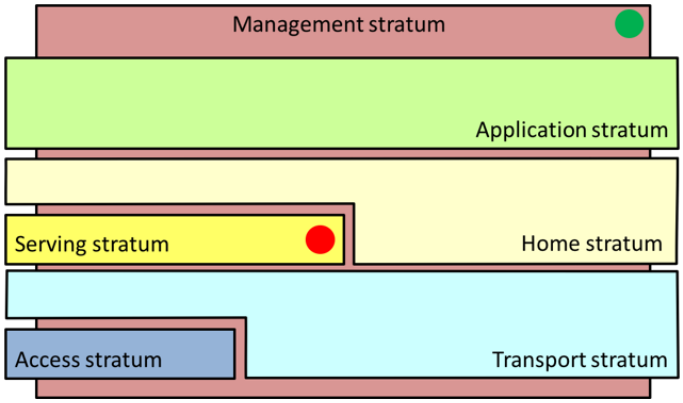


Figure 6-19 Enabler relevant strata

6.5.9.3 Quality attributes

Several *preliminary* quality attributes have been defined for the *Bootstrapping trust* enabler. The quality attributes below have been defined following the currently assumed use cases and the principles of network virtualization:

- Virtual network components deployed with TEE support using the *Bootstrapping trust* enabler should forward packets at a rate similar to virtual network components deployed without TEE support.
    - **Clarification**: Intra-host virtual switching is near instant, as packets are matched against policies and transferred in memory and use of TEEs should not induce a significant overhead. Similarly, VNFs must support a high throughput for large SDN deployments. Hence, implementation should be done with TEEs that offer native performance. Failure to keep up with the required performance would transform virtual network components deployed in TEEs into bottlenecks and would prevent their adoption.
- Virtual network components deployed in TEEs using the *Bootstrapping trust* enabler should be easily upgradable and interoperate with other commodity components and common protocols.
    - **Clarification:** Radical changes in the software architecture or operation of the virtual network components deployed with TEE support could limit their maintainability or require the use of implementation-specific protocols. This in turn would prevent adoption of such virtual network components.

As the enabler is currently in development, the specified quality attributes may change at a later stage.

6.5.9.4 Technical constraints

The *Bootstrapping Trust* enabler introduces a number of technical constraints and prerequisites, on the platform, software and protocol level.

On hardware level, the virtualization platforms must have support for Intel Software Guard Extensions (SGX) or other technology for creating TEEs with properties comparable to SGX enclaves. In the absence of such support, the API provided by Intel SGX SDK can be used. While it is helpful or even essential for development and testing, an SDK without hardware support does not provide any security guarantees expected from a TEE.

Software applications may need modification in order to add support for executing in a TEE. In most cases, this excludes closed-source software from the list of potential software applications to be used by the

enabler (unless the vendor itself adds the required support). Additionally, a verification manager must be deployed. The network orchestrator can be extended to implement the *Verification Manager* functionality.

Finally, on the protocol level, it is essential that the OpenFlow protocol [ONF 2012], the network controller, as well as VNFs feature support for transport layer security (TLS). Such support is essential for ensuring confidentiality and integrity of communication between network components.

6.5.9.5 Business constraints

No business constraints have been found at this point.

6.5.10 Detailed specification

6.5.10.1 Introduction

The purpose of the *Bootstrapping trust* enabler is to deploy and operate a trusted SDN infrastructure, given that the physical security is ensured and that the underlying roots of trust – in this case the platform processor and the code and data of the network components deployed in TEEs – have been implemented correctly.

The enabler aims to be implementation-agnostic and one should be able to implement the principles of the enabler using different technologies. However, for the sake of clarity, the specification uses – where needed – concrete technologies and software applications.

The *Bootstrapping trust* enabler aims specifically to allow tenants to securely deploy virtual network components with TEE support and set up virtualized network topologies in a cloud environment. While, this addresses some of the threats towards SDN infrastructure deployments, as described in [Paladi et al, 2015; Svensson et al, 2016], it *does not* protect the SDN deployment from exploits of vulnerabilities in the implementations of the network controller, VNFs or virtual switches. The *Bootstrapping trust* enabler is a mechanism to deploy the VNF or virtual switch code and data in immutable TEEs and attest that their integrity has been maintained in the process.

Another important aspect is that executing software in a TEE such as Intel SGX enclaves requires certain adaptations to the application binary interface available to SGX enclaves. To minimize the effort required for enabler implementation, one may choose to only place carefully selected security sensitive parts of the application in TEEs. The definition of which parts are *security sensitive* (e.g. form its TCB) is application specific and out of the scope of this specification.

Finally, it must be noted that the current API specification is likely to change both as the enabler itself matures, and as new features are added.

6.5.10.2 Conformance

The enabler requires the use of SDN and both hardware and software support for TEE. The *Bootstrapping Trust* enabler implementation uses Intel SGXs to create the required TEEs.

Once the virtual network components (namely the VNFs or virtual switches) have been deployed in TEEs, various approaches for establishing a secure communication channel can be used. This implementation of the *Bootstrapping Trust* enabler uses TLS to establish a secure communication session between the components.

In order to be conformant, an enabler should comply with all the specifications described below.

6.5.10.3 API specification

Below is a preliminary API specification of the enabler.

Each API call is described with title, the set of expected inputs, the set of expected outputs and a clarification comment. As the enabler is under development, this API is preliminary and may – with a high probability – be changed. As the enabler implementation will mature, a more detailed API will be made available.

1. ***Deploy virtual network components enclaves***
    a. **Input**
        i. List of target hosts (ip addresses/hostnames)
        ii. Url to repository with enclave-capable virtual network component application code
        iii. Url to repository with bootstrap application code
    b. Output
        i. Tuple list of <target host, enclave identity> (for hosts where deployment was successful)
        ii. Tuple list of <target host, error message> (for hosts where deployment has failed)
    c. **Comment** This API command is intended to the initial deployment of virtual network component application code on the virtualization hosts (or bare-metal hosts) within the domain of the tenant, specified by the list of target hosts. On each target host, upon receiving the command, the bootstrap application deploys the virtual network component code and interacts with the host-local SGX driver to create a support enclave. As a result, on each target host, the bootstrap application either obtains an enclave identity of the created enclave, or an error code in case of failure. These results are returned to the caller.
2. ***Quote virtual network component enclaves***
    a. **Input**
        i. Tuple list <target host, enclave identity>
    b. Output
        i. Tuple list <target host, quote message> (for cases when the signature of the quote has been successfully validated)
        ii. Tuple list <target host, error message> (for cases when the quote signature validation failed)
    c. Comment Quoting a virtual network component enclave allows to reliably obtain a snapshot of the code and data in the created enclave. The trustworthiness of the quote depends upon the properties of the TEE (in this implementation – Intel SGX enclaves) and on the security of the signing keys (the signing keys are provided by the vendor and never leave the security domain of the TEE). The signature of the *quote message* is then validated against the public key of the platform in order to ascertain its integrity and authenticity. Failure to do so invalidates any trust expectations on the respective enclave.
3. ***Attest virtual network component enclave***
    a. Input
        i. Tuple list <Enclave identity, Quote>
    b. Output
        i. Tuple list <Enclave identity, Attestation Result>
    c. Comment Attestation allows to verify whether the code and data executing in the enclave has not been modified and its fingerprint is identical to the expected values. This is a simple

matching operation with a binary answer – the fingerprint either matches or does not. If the fingerprint matches, the enclave is executing the expected code and data (assuming trust in the implementation of the platform). If the fingerprint does not match, nothing can be stated about the enclave integrity.

4. ***Enroll attested virtual network component***
    a. **Input**
        i. Tuple list <Enclave identity, Enrolment message>
    b. **Output**
        i. Updated list/view of the switches in the topology
    c. **Comment** Once the platform TCB and the enclave has been successfully attested (i.e. with a matching fingerprint), the virtual network component is included in the topology. The enrolment message sent to each virtual network component enclave is confidentiality and integrity protected (using the public key of the enclave, which was earlier communicated as part of the quote message), and contains the keying material used to establish a protected communication channel between the network controller and each virtual network component.

6.5.11 Re-utilized technologies/specifications

The *Bootstrapping Trust* enabler uses the "Open vSwitch" virtual switch implementation, the Floodlight SDN controller, and sample VNFs supported by Floodlight. The utilized components are modified where necessary for the purposes of the enabler. Furthermore, the enabler utilizes the Intel SGX SDK [Anati et al, 2013] for hardware-based TEE isolation and Docker containers for lightweight virtualization.

6.5.12 References

[1] Open Networking Foundation (ONF). OpenFlow switch specification – version 1.3.0 (wire protocol 0x04). 2012.

[2] Porras, P., Cheung, S., Fong, M., Skinner, K., Yegneswaran, V.: Securing the software-defined network control layer. In: Proceedings of the 2015 Network and Distributed System Security Symposium (NDSS), San Diego, California (2015).

[3] Paladi, N., Gehrmann, C.: Towards Secure Multi-tenant Virtualized Networks. In: Trustcom/BigDataSE/ISPA, 2015 IEEE. vol. 1, pp. 1180–1185. IEEE (2015)

[4] Svensson, Martin, Nicolae Paladi, and Rosario Giustolisi. "5G: Towards secure ubiquitous connectivity beyond 2020." (2015). Tech. report, SICS Swedish ICT http://soda.swedishict.se/5933/1/T2015_08.pdf

[5] Cockburn, Alistair. "Writing effective use cases." *Addison-Wesley, 2001.* ISBN 9780201702255

[6] Anati, I., Gueron, S., Johnson, S., Scarlata, V.: Innovative technology for CPU based attestation and sealing. In: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. p. 10 (2013)

[7] 5G-Ensure Consortium, Deliverable 2.4 Security Architecture (draft), [Online]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.4-SecurityArchitectureDraft.pdf, 2016.

[8] Paladi, N., Gehrmann, C., and Michalas, A.: "Providing User Security Guarantees in Public Infrastructure Clouds," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[9] M. Bursell, A. Dutta, H.-L. Lu, M.-P. Odini, K. Roemer, K. Sood, M. Wong, and P. Wörndle, "Network Functions Virtualisation (NFV), NFV Security, Security and Trust Guidance, v.1.1.1." http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/003/01. 01.01_60/gs_NFV-

SEC003v010101p.pdf, 2014. Accessed: 2016-11-20.

[10] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, "Principles of remote attestation," *International Journal of Information Security*, vol. 10, no. 2, pp. 63–81, 2011.

6.6 Flow control enabler: open specification

6.6.1 Preface

By adopting Network Functions Virtualization (NFV), network operators will benefit greatly from the flexible provisioning of new services, reducing at the same time both CAPEX and OPEX. NFV abstracts network functionalities from the underlying hardware used to operate them, enabling thus on-demand deployments on general purpose hardware. In the context of mobile network architecture, examples of such applications are: firewalls, load balancers, CDNs, NATs, DPI probes, VPN, IMS, and packet gateways.

When deploying Virtual Network Functions (VNFs) with a critical impact on overall network functionalities, network operators should take also into account the security threats (e.g., DDoS) that come with and that may severely affect them, given also that the deployed network applications may run over data-centres not directly owned and operated by them. For instance, for some critical network functions such as firewalling, load balancing and packet gateway, an attack may result in a cascading effect, taking down most of the network functionalities. Therefore, the rapid detection and mitigation of such threats represents one of the most impelling priorities for operators.

6.6.2 Status

A prototype of the enabler is currently developed; the specification of the enabler is a preliminary one.

6.6.3 Copyright

The enabler is owned by Thales Communications & Security (TCS). TCS is the only contributor to this enabler.

6.6.4 Legal notice

The Legal Notice that applies to this specification is given in Annex A.

6.6.5 Terms and definitions

DDoS	Distributed Denial of Service
EvS	Enhanced virtual Switch
ONF	Open Networking Foundation
OpenFlow	Protocol standardized by the Open Networking Foundation (ONF) for controller-switch communication.
NFV	Network Function Virtualisation
SDN	Software-Define Networking
VIM	Virtualized Infrastructure Manager
VNF	Virtual Network Function

6.6.6 Overview

Current approaches to secure virtualized networks from external threats propose to proactively add means to minimize network disruptions and data loss in the case of attacks. These goals can be achieved either by adding redundancy or by the setting up one or more dedicated protection schemes. The *Flow Control*

enabler described in this section applies to the latter solution, by proposing a means to detect and mitigate in-network threats targeting critical VNFs, both in the control and data planes, logically deployed in a larger core network. To this end, an *Enhanced virtual Switches (EvS)* embedding the capability to protect the virtual network interfaces is added to the network's data plane as a gateway to the targeted critical VNF. In particular, the *EvS* can be deployed and instructed at runtime, by exploiting SDN concepts – namely, through a dedicated Flow Controller. The EvS is capable of automatically detecting network-based security threats and to act appropriately to minimize their impact (e.g., applying rate limiting policies, black holing or discarding certain flows coming from suspicious hosts).

6.6.7 Basic concept

In an NFV capable network, the network manager/orchestrator maintains an up-to-date global view of the network-storage-compute resources available and interacts with the virtualized applications by the means of dedicated protocols (e.g., Open-Flow to setup the control plane of a vSwitch). For this reason, implementing a software-based threat detection and mitigation strategy on top of the network manager to protect critical network functions seems a straightforward idea.

As a result, in order to protect deployed VNFs (typically offering a network function or service application) from network-based attacks, the effects of which can be harmful for the entire network - a large scale DDoS attack targeting a critical function in the core of the network (e.g., HSS database) - three steps are typically required 1) traffic monitoring; 2) threat detection; and 3) threat mitigation. We will detail these three steps in the following:

1. In order to perform traffic monitoring, the forwarding device (e.g., the EvS in this enabler) must be able to retrieve statistics on the packets flowing through, such as host/port binding, and to report them to a controller to take the ultimate decision on the dangerousness of that traffic. Data collection is particularly tricky here, because we would like the EvS to track efficiently any flow feature with fine-grained information without overloading too much the forwarding performance and the control channel towards the controller.
2. Anomaly detection is performed on the monitored data in order to discriminate heavy hitters and suspicious activities from the legitimate traffic, and also to identify attacker and target tuples. Since this task is complex and requires correlating several flows in real-time, detection is typically done at the controller, by frequently requesting EvS for flows statistics e.g., through OpenFlow messages. Moreover, this controller-based strategy allows performing a sort of network-level detection, by receiving several lectures from logically independent EvS deployed in the network to protect different VNFs. For completeness, it should also be noted that some recent approaches propose offloading a part of the detection process directly to the switches in order to have a quicker reaction process.
3. Finally, threat mitigation should set up the most appropriate reactions on the data-plane to counter the detected attack (e.g., by black-holing certain flow or rate limiting some hosts), by issuing and installing flow rules.

6.6.8 Main interactions

6.6.8.1 Use cases

Denial of Service (DoS) attacks and in general cyber-threats gained momentum in the recent years, threatening both network and web infrastructures around the world. By using distributed spam-bots together with sophisticated reflection techniques, attackers can easily generate traffic volumes in the order

of Tbps [1], also causing catastrophic consequences, as proven by the Dyn attack in October, 2016 that led to major disruptions of internet services both in US and Europe. Different types of threats exist [2]. Their impact can be significant: they are able to generate such a huge amount of traffic causing targets to crash and the associated service(s) to become partially or totally unavailable. Detecting and mitigating such attacks is far from being straightforward [3]. First, they usually do not come from a single identified source, which makes remediation very difficult without also affecting legitimate traffic. Second, they appear either very suddenly, thus requiring fast reaction to counter their effects, or very slowly, thus making the detection even more complicated [4].

Despite to date these threats target mostly application servers located on the public internet, the emergence of NFV will certainly trigger an increased rate of attacks toward VNFs designated to assure core network functionalities. Simplicity and flexibility are some of the key features offered by SDN, making it an ideal candidate for managing network security in the context of NFV. SDN allows for fast reactions to security threats by dynamically enforcing simple forwarding rules as counter-measures.

Protecting all sort of critical network function in a virtualized environment from any kind of network threats with a mechanism that detects and mitigate threats with high accuracy and without congesting the control plane would increase the reliability of the entire network. This enabler proposes an on-demand deployable *EvS* – an enhanced version of vSwitch embedding also detection and mitigation capabilities – coupled with a *Flow Controller* to protect the critical VNF from network-based threats.

6.6.8.2 Components and interaction overview

Two main components of the enabler are presented, namely the *EvS* and the *Flow Controller* as depicted in **Figure 6-20**. The proposed enabler can be deployed either as a standalone container (embedding both the EvS and the Flow Controller functionalities) acting as a gateway to the target VNF, or as an EvS directly protecting the VNF and a Flow Controller deployed in the Virtualized Infra Manager (VIM).

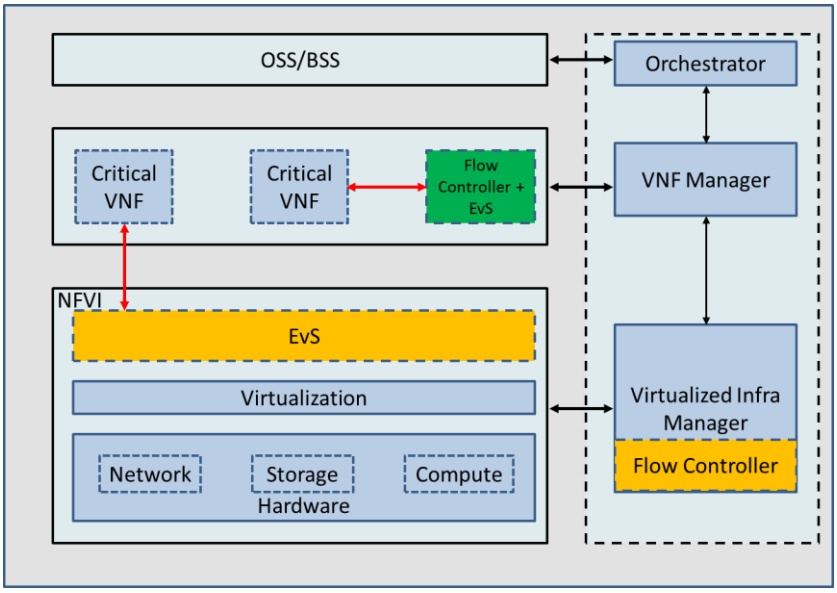


Figure 6-20 ETSI NFV architectural framework comprising the flow control enabler that can be deployed either as a standalone container (in green) or as a centralized controller + switch (in orange).

The inputs of the Flow Controller are the location of the target VNF(s), a list of monitored flow features, the detection parameters and eventually the pre-existing mitigation and forwarding policies to install on the

EvS (e.g., hosts or flows that are already in the whitelist and/or blacklist). Whenever an EvS is instantiated to protect a running VNF, the Flow Controller, regardless of where it is deployed, sends OpenFlow messages of the type OFPT_FLOW_MOD to match the list of features to track (e.g., the source address and port) and to perform actions upon packets reception (e.g., the action may be either to increment a counter of the tracked feature at the EvS, to construct a sketch for the flow, or to reroute the incoming packets towards the controller). Initially, all the flows directed to the VNF are forwarded using the rule OFPT_ACTION_OUTPUT, all other flows dumped. The controller retrieves periodically a list of the tracked features and their associated counters to perform threat detection. Eventually, detection triggers the installation of mitigation rules on the EvS by using one or more OFPT_FLOW_MOD messages.

6.6.9 Architectural drivers

6.6.9.1 High-level functional requirements

The EvS must be able to track the features of the data plane flows directed to the protected VNF as instructed by the Flow Controller (this may require to execute some computation or take into account states of execution) and to inform periodically the Flow Controller of its up-to-date counter values. The decision whether a given flow is malicious or not should be taken at the controller (eventually by correlating the information received by multiple EvS). The EvS must be able to comply with the mitigation rules as instructed by the controller.

6.6.9.2 Quality attributes

The overhead on the control channel should be maintained as low as possible, in order not to decrease the performance of the controller and to limit the overload of the control plane. The processing overhead due to traffic monitoring at the EvS should be as low as possible to not interfere with the forwarding capabilities (data plane performance reduction).

6.6.9.3 Technical constraints

The EvS must be deployed on-demand in the logical proximity (1 hop) of the protected critical VNF. All the traffic going to the VNF should transit through the EvS. Its network interfaces should need a very limited amount of configuration. The threat detection appliance may or may not be integrated in the VIM. In the standalone container scenario, the threat detection appliance must be invoked by the controller on every every OpenFlow message that is sent to the control plane and that is received from the control plane.

6.6.9.4 Business constraints

No business constraints to take into account for this enabler.

6.6.9.5 Link to security architecture

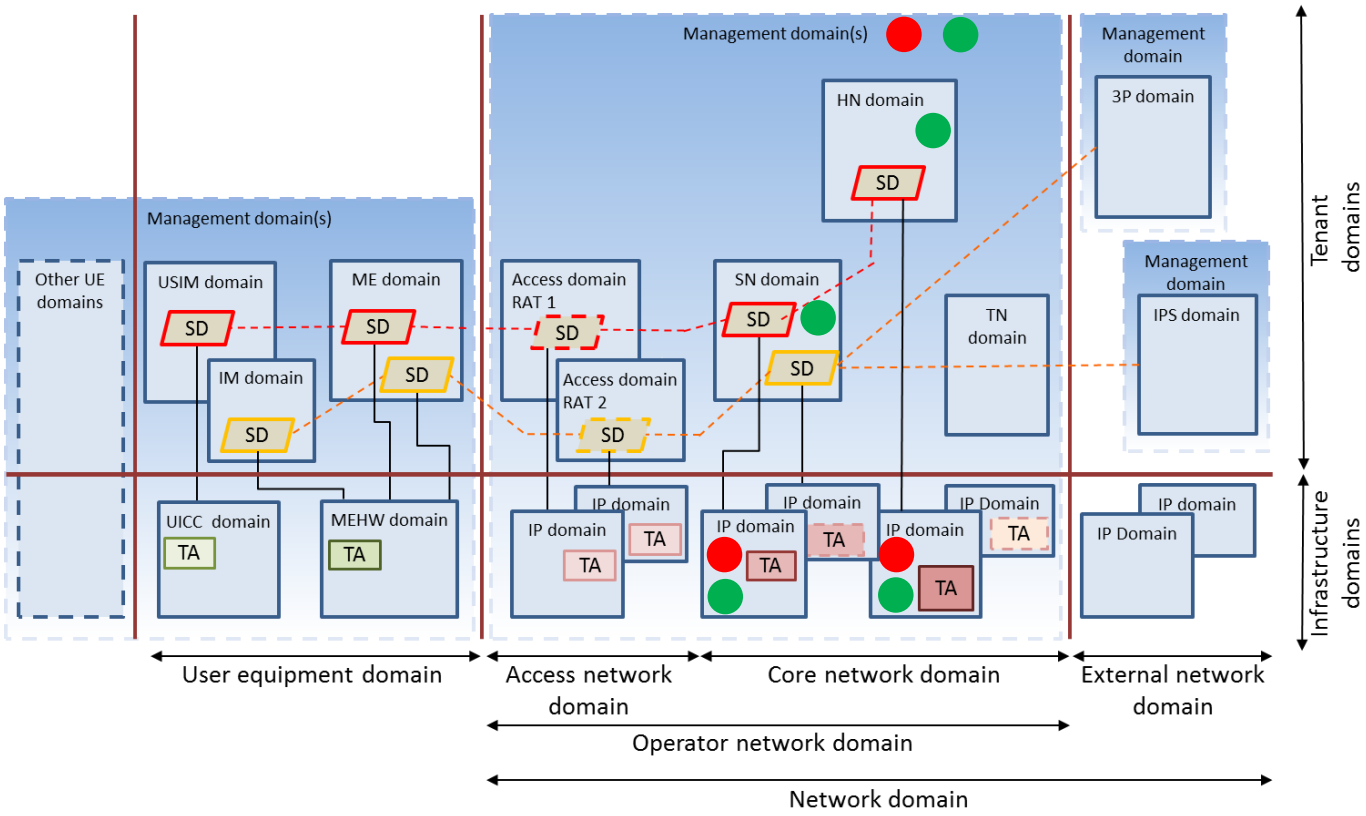


Figure 6-21 Enabler relevant domains

Figure 6-21 depicts the most relevant domains of the 5G-ENSURE security architecture [5] that are covered by this enabler. The domains for which the enabler contributes to their security are marked by a red dot. The Flow Control enabler can protect the VNFs dedicated both to management and infrastructure, mostly in the core network. **Figure 6-21** marks the common domains from which information must be sent to the Flow Control enabler by a green dot. Analogously, **Figure 6-22** marks the relevant strata of the 5G-ENSURE security architecture.

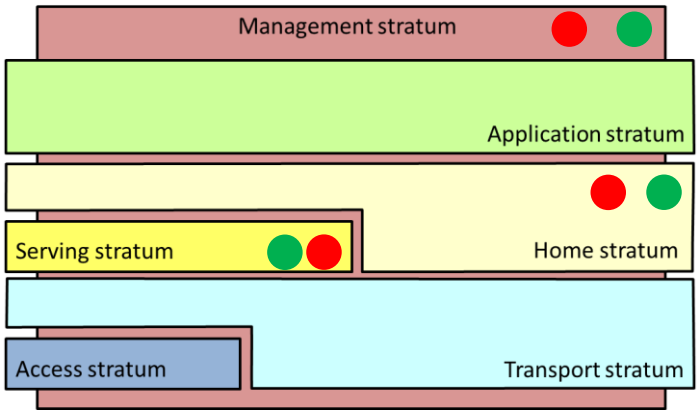


Figure 6-22 Enabler relevant strata

6.6.10 Detailed specification

6.6.10.1 Introduction

The enabler proposes a generic Flow Controller that can be deployed either inside a container together with the EvS or logically centralized within the VIM. The controller communicates with the EvS via OpenFlow protocol, eventually extended to take into account the "evolved" functionalities of EvS (i.e., efficient features counting and offloaded detection).

The rationale behind the deployment decision (distributed and standalone container vs. centralized controller coupled with EvS) stems from the type of VNF to protect. If it is expected that the VNF will sustain a huge traffic and will be a probable target of strong attacks or if the VNF is critical to the functioning of the overall core network, it may be a wise decision to deploy the threat detection controller as a container embedding also its related EvS (one-to-one relation between the controller and the EvS). On the contrary, for less critical VNFs, in order to consume less computing resources, it is possible to deploy a single Flow Controller, logically located at VIM, to handle a multitude of EvS (one-to-many relation between the controller and the EvS). This latter case can benefit also from a network-wide detection strategy, which is not possible with a completely distributed strategy, but could have difficulties in scaling with traffic.

In the following, we will define the APIs for the threat detection controller in detail.

6.6.10.2 Conformance

An implementation to be reported as conformant shall comply with the open specification here stated for the enabler.

6.6.10.3 API specifications

6.6.10.3.1 Functions, messages and commands

The API for the controller must be able to describe the desired detection and mitigation policy for any type of deployment. Namely, it must be possible to define whether the target deployment requires a centralized Flow Controller logically deployed at the Virtualized Infrastructure Manager (VIM) rather than into a standalone container embedding both the EvS and the controller. After this step, it is required to indicate the list of critical VNF(s) to be protected. In order to drive the detection step, at configuration time it is required to specify which L3 and L4 flow features to monitor (e.g., source or destination addresses and ports, the protocol type, packets having certain special flags) – by default, only the source port and source IP address of a flow are counted. Our choice here is to reuse the very flexible OpenFlow protocol to describe the features of the flows, since the interactions will be with an OpenFlow capable controller. It is then possible to decide which type of detection algorithm to use, together with its related detection parameters, like the sensitivity, or some thresholds used to decide whether a flow is malicious or not. Finally, the proposed API eventually gives the ability to define the list of flows that are already known to be in a whitelist or blacklist, defined as OpenFlow matches.

In order to setup the Flow Control detection and mitigation, the following API is offered:

```
      void SetupFlowControl(bool distributed, List<Vnf> applications,
List<OpenFlowMatch> features, List<OpenFlowMatch> whiteList ,
List<OpenFlowMatch> blackList, detectionType detection, Object
optionalParameters)
```

where `distributed` indicates whether the deployment has to be distributed or centralized, `applications` is the list of deployed VNFs to be protected, `features` represents the list of OpenFlow features to match for counting, `whitelist` and `blacklist` represents the list of flows to be included into a whitelist or a blacklist, finally, `detection` is an indicator of the type of detection to perform together with its `optionalParameters` values. Examples for defining API calls are proposed in Section **Error! Reference source not found.**.

6.6.10.4 Examples

In order to illustrate the instantiation of the Flow Control enabler, we consider a networking setup composed a set of VNFs already instantiated having two different criticality levels, named respectively *CRITICAL-VNF$_i$* and *STANDARD-VNF$_j$* with *i* in [1,MAX_CRITICAL] and *j* in [1,MAX_STANDARD].

We want to protect all the critical VNFs with dedicated means, but we can settle to protect all the standard VNFs with a single Flow Control controller that we deploy to the VIM.

For all the critical VNFs then we call:

```
SetupFlowControl(distributed=True, applications=vnf_c_list, features=matches, whitelist=None, blacklist=None, detection="entropy", deviation=3)
```

Having defined `vnf_c_list=[`*CRITICAL-VNF$_1$*`, .., `*CRITICAL-VNF$_{MAX_CRITICAL}$*`]`, and `matches = [ofproto.OXM_OF_IPV4_SRC, ofproto.OXM_OF_TCP_SRC]`.

In this case, the controller will not insert any host in the blacklist/whitelist, and will use an entropy-based detection strategy that flags as suspicious all the flows composed of the tuple `<IP source, TCP source port>` hitting more than the mean entropy plus three times its standard deviation.

Instead, for the normal priority VNFs, we can opt for a centralized approach as follow:

```
SetupFlowControl(distributed=False, applications=vnf_n_list, features=matches, whitelist=, blacklist=None, detection="counter", threshold=1000)
```

Having defined `vnf_n_list=[`*NORMAL-VNF$_1$*`, .., `*NORMAL-VNF$_{MAX_NORMAL}$*`]`, `matches = [ofproto.OXM_OF_IPV4_SRC, ofproto.OXM_OF_TCP_SRC]`, and `safe=[<`*safeIP:safePort*`>]`.

In this latter case, the controller will deploy a single Flow Controller in the VIM to handle all the normal priority VNFs. The detection stage is performed simply by flagging as malicious the tuples `<IP source, TCP source port>` that exceed the threshold of 1000 match/s. In this case, the parameter safe is used to flag whitelisted users that are always allowed to pass the switch.

6.6.11 Re-utilised technologies/specifications

The threat detection and mitigation controller is based over the Ryu SDN [6] controller but can be adapted to any existing SDN controller. The EvS is an ad hoc Open vSwitch version guaranteeing advanced in-switch processing capabilities such as feature counting.

6.6.12 References

[1] S. Khandelwal, «World's largest 1 Tbps DDoS Attack launched from 152,000 hacked Smart Devices,» [En ligne]. Available: http://thehackernews.com/2016/09/ddos-attack-iot.html. [Accès le 2016].

[2] C. Douligeris et A. Mitrokotsa, «DDoS attacks and defense mechanisms: Classification and state-of-the-art,» *Computer Networks,* vol. 44, n° %15, pp. 643-666, 2004.

[3] P. Holl, «Exploring DDoS Defense Mechanisms,» *Network Architectures,* pp. 25-32, 2015.

[4] J. Park, K. Iwai, H. Tanaka et T. Kurokawa, «Analysis of slow read DOS attack,» ISITA, 2014.

[5] 5G Ensure Consortium, «Deliverable 2.4 Security Architecture,» 2016. [En ligne]. Available: http://www.5gensure.eu/sites/default/files/Deliverables/5G-ENSURE_D2.4-SecurityArchitectureDraft.pdf.

[6] «Ryu SDN framework,» [En ligne]. Available: https://osrg.github.io/ryu/.

7 Conclusions

In this document, the open specifications of 5G-ENSURE security enablers planned to compose the second (also last) release (i.e. v2.0) as per updated Technical Roadmap reported in D3.5 have been presented and detailed. Each of these enablers either continued from R1 with new features or fully new in R2. is expected to be delivered on due time by M22 (i.e. August 2017) as part of the next deliverables to come (namely D3.7 security enablers software release (v2.0) and D3.8 security enablers documentation). Work reported here was performed in close collaboration with other technical work packages and exploited the deliverables already produced by the project, especially D2.1 on Use Cases, D3.5 on Technical Roadmap (Update) but also D2.4 defining early version of 5G security architecture. D3.6 conveying open specs of 5G ENSURE Security enablers for R2 is expected to feed also the work of other Technical WPs starting with WP2 busy with update of 5G Security Architecture to which security enablers are now linked but also WP4 in charge to evolve 5G Security Testbed to get them integrated, deployed and finally assessed

These open specifications of 5G-ENSURE Security enablers in R2 are also expected to be shared once again with 5G-PPP community at large, starting first and foremost with 5G-PPP Security community through 5G-PPP Security WG activities on-going.

A Open specification Legal Notice

General Information

"5G-ENSURE Partners" refers to parties of the 5G-ENSURE Project in accordance with the terms of the 5G-ENSURE Consortium Agreement.

Copyright Holders of the 5G-ENSURE Open Specification is/are the Party/Parties identified as the copyright owner/s of the 5G-ENSURE Open Specification that contains this Legal Notice.

Use of Specification - Terms, Conditions & Notices

The material in this specification details a 5G-ENSURE Enabler Specification (hereinafter “Specification”) and is provided in accordance with the terms, conditions and notices set forth below (“License”). This Specification does not represent a commitment to implement any portion of this Specification in any company's products. The information contained in this Specification is subject to change without notice. These terms, conditions and notices are only applicable to this Specification.

Licenses

Subject to all of the terms and conditions below, the Copyright Holders in this Specification hereby grant you, the individual or legal entity exercising permissions granted by this License, a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide, royalty free license (without the right to sublicense) under its respective copyrights incorporated in the Specification, to copy and modify this Specification and to distribute copies of the modified version, and to use this Specification, to create and distribute special purpose specifications and software that is an implementation of this Specification. Any distribution or redistribution of this Specification, with or without modification, must include these terms, conditions and notices.

Patent Information

The 5G-ENSURE Partners and/or Copyright Holders shall not be responsible for identifying patents for which a license may be required for any implementation of any 5G-ENSURE Specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. 5G-ENSURE specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

General Use Restrictions

Any unauthorized use of this Specification may violate copyright laws, trademark laws, and communications regulations and statutes. This Specification contains information which is protected by copyright. All Rights Reserved. This Specification shall not be used in any form or for any other purpose different from those herein authorized, without the permission of the respective Copyright Holders.

Neither the name(s) of the Copyright Holders nor the names of the 5G-ENSURE Partners may be used to endorse or promote products derived from this Specification without its/their specific prior written permission.

For avoidance of doubt, the rights granted are only those expressly stated in this Legal Notice herein. No other rights of any kind are granted by implication, estoppel, waiver or otherwise.

Disclaimer of Warranty

WHILE THIS SPECIFICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE 5G-ENSURE PARTNERS AND THE COPYRIGHT HOLDERS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, WARRANTY OF NON-INFRINGEMENT OF THIRD PARTY RIGHTS, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE.

YOU ARE SOLELY RESPONSIBLE FOR DETERMINING THE APPROPRIATENESS OF USING OR REDISTRIBUTING THIS SPECIFICATION AND ASSUME ANY RISKS ASSOCIATED WITH YOUR EXERCISE OF PERMISSIONS UNDER THIS LICENSE.

IN NO EVENT AND UNDER NO LEGAL THEORY SHALL THE 5G-ENSURE PARTNERS AND THE COPYRIGHT HOLDERS BE LIABLE FOR ERRORS CONTAINED IN THIS SPECIFICATION OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS SPECIFICATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF SOFTWARE DEVELOPED USING THIS SPECIFICATION IS BORNE BY YOU. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OFTHE COPYRIGHT LICENSES GRANTED TO YOU TO USE THIS SPECIFICATION.

Trademarks

You shall not use any trademark, marks or trade names (collectively, "Marks") of the 5G-ENSURE Partners or the Copyright Holders or the 5G-ENSURE Project without prior written consent, except as required for reasonable and customary use in describing the origin of this Specification and reproducing the content of this Legal Notice.

Issue Reporting

This Specification is subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Procedure described on the web page.

However, there is no obligation on the part of the Copyright Holder to provide any review, improvement, bug fixes or modifications this Specification to address any reported ambiguities, inconsistencies, or inaccuracies.