



Security And Interoperability in Next Generation PPDR
Communication Infrastructures



Project Number: 313296

Deliverable 3.2

Enterprise and System Architecture – Intermediate

Scope	Scheduled deliverable
Lead Beneficiary	FhG
Dissemination level	Public
Document creation date	10/Feb/2015
Document release date	13/Jun/2015
Contractual Date of Delivery	31/May/2015
Version	1.0
Status	Final

Abstract: Deliverable D3.2 describes the functional building blocks - also known as Services -, proposes an initial framework for their management and provides component functions of the SALUS System Architecture derived from operational nodes, information items and information exchanges captured in the SALUS Enterprise Architecture supporting operators at different levels of command in PPDR operations.

AUTHORS

Name	Organisation	Email
Frank Reinert	FhG	frank.reinert@iosb.fraunhofer.de
Wilmuth Müller	FhG	wilmuth.müller@iosb.fraunhofer.de
Hugo Fonseca	ONE	htfonseca@onesource.pt
David Jelenc	UL	david.jelenc@fri.uni-lj.si
Jérôme Brouet	ALU-I	jerome.brouet@alcatel-lucent.com
Rick Hofstede	UTWENTE	r.j.hofstede@utwente.nl
Bernd Meijerink	UTWENTE	bernd.meijerink@utwente.nl
Hugo Marques	IT	hugo.marques@av.it.pt
Luis Pereira	IT	luis.pereira@av.it.pt

QUALITY ASSURANCE TEAM

Name	Organisation	Email
Dragan Olcan	UB	olcan@etf.rs
Daniel Zerbib	CAS	daniel.zerbib@airbus.com
Olayinka Adigun	KU	o.adigun@kingston.ac.uk

EXECUTIVE SUMMARY

To solve the complex challenge of building efficient future PPDR networks that will integrate existing and new technologies and will have as major objectives to enhance operational effectiveness and improve interoperability between agencies from home and neighbouring countries, a robust methodology is required. SALUS approach is to use the Enterprise Architecture (EA) which has been successfully used in complex domain such as Defence. Indeed, the guiding principle of the EA is to support the adaptation of the “business” to benefits from new technologies and to exploit the (new) technologies to better support the “business”.

This deliverable gives a first cut at the SALUS Enterprise Architecture reflecting the Use Case scenarios developed in Work Package 2 (WP2) - City Security, Temporary Protection, Disaster Recovery - and covering the most important and challenging missions of PPDR organisations and where the use of an efficient system is paramount. As explained in Deliverable D3.1 [2], the SALUS EA is based on the Open Safety and Security Architecture Framework (OSSAF) and on NATO Architecture Framework (NAF). For the sake of simplicity, this deliverable addresses a reduced number views from the four perspectives defined by OSSAF:

- Strategic perspective: focusing only a the ‘capability planning’
- Operational perspective: developing operational node concepts and information exchange models,
- Functional model: developing systems and services (focusing on application services and technical services), their functional requirements and their connectivity model and their interface,
- Technical view: focusing on solution, standards and protocols, and device connectivity models.

Indeed, moving from one perspective to the next level gives more granularities on the way the system will operate. It is to be noted that from all the capabilities and services required for future PPDR system, this deliverable (which is the Intermediate version of the SALUS Enterprise and System Architecture) is describing the complete modelling for a sub-set of the key SALUS services only. The final version will propose detailed modelling of additional services.

The analysis of the scenarios leads to the identification of the SALUS capabilities (Strategic Perspective) and their dependencies. It shows that providing “Access to Common Information Infrastructure Services” and “Communication Connectivity” are essential capabilities to achieve the high level capabilities of protecting citizens, properties and PPDR own personnel as well as conducting missions in an integrated way.

The next level of the EA model provided in this document (Operational Perspective) is the identification and definition of the operational nodes, the information exchanges between the operational nodes and the information items details. The operational nodes reflect the different levels of PPDR organisations (strategic, tactical and operational) and different agencies (here limited to three –police, fire-brigade, emergency medical services, but could be extended to a higher number following similar principles). This step also includes the derivation of requirements related to information exchange and identification of operational activities. It is to be noted that additional efforts will be required in the description of operational workflows).

Based on the above, functional blocks are defined (Functional Perspective). At this step, it is essential to adopt a common vocabulary and a Service Oriented Architecture (SOA) approach to ease reusability, scalability and flexibility of the different services for different deployment scenarios and use cases. A service map is proposed along five domains:

- Operations domain: this include the real operational functions lists, activities and workflows and is not further developed in this document;
- Community of Interest (COI) domain: functionalities that are user and applications specific,
- Network and information infrastructure (NII) domain: services that cover information and technology aspects;
- Service management domain: cross-domain service related to system and system element management;
- Service assurance domain: cross domain service providing security functions.

A service taxonomy is proposed for each service category; this defines a finer grained definition of individual abstract services. Examples of such abstract services are 'Force Tracking Service' or 'Indoor Localization Service' for the COI category, 'Message Brokering Service' or 'TETRA2LTE Service' for NII category. A short description of each meaningful service for SALUS is provided.

Then, this deliverable develops detailed specifications and system architecture for the individual services that will be core services for SALUS first and second validation platform. These specifications include service level specifications, interfaces, state machine, information model, service composition and service functionality specifications. The services that are further detailed in this deliverable are the following:

- Message Broker Service
- Force Tracking Service
- Sensor Data Acquisition Service
- Intrusion detection Service
- Forensic Service
- Dynamic QoS Control Service.

Finally, this deliverable also includes an example of an end-to-end system architecture targeting scenario 1 and 2 (City Security and Temporary protection) with the objective to put in a 'simpler' drawing the different functions and elements and to give an example of the mapping of the services developed above in physical entities.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	5
TABLE OF FIGURES	6
TABLE OF TABLES	7
1 Introduction	8
2 Enterprise architecture development methodology	9
3 Enterprise Architecture Development – Guiding Principles	15
3.1 Architecture Scope	15
3.2 Architecture Style - Service-Oriented Architecture (SOA)	16
3.2.1 SALUS Service Definition	17
3.2.2 SOA-Bridge	18
4 SALUS Enterprise Architecture	20
4.1 SALUS related PPDR Capabilities	20
4.2 Operational Nodes, Information Exchanges and Information Items	22
4.2.1 Operational Processes (Workflows)	28
4.3 Service Map – A Framework for Functional Building Blocks –	28
4.3.1 Purpose	28
4.3.2 Service Categories and Methodology for Service Derivation	29
4.3.3 Service Specifications	38
5 Technical Solution – System Architecture	84
5.1 SALUS System Architecture – Overview	84
5.1.1 High-level System Architecture	84
5.1.2 Mapping of Services to Functional Blocks	88
5.1.3 Physical Deployment Options	94
5.2 SALUS System Architecture – Service Provisions	94
5.2.1 Service Provision: <<Sensor Data Acquisition Service>>	97
5.2.2 Service Provision: <<Intrusion Detection Service>>	99
5.2.3 Service Provision: <<Message Brokering Service>>	101
5.2.4 Service Provision: <<Force Tracking Service>>	103
Bibliography	105
Acronyms	106
Annex A – Information Items and Information Exchange requirements	108
Annex B – SALUS related capabilities of PPDR organisations	130
Annex C – Operational Nodes and Information Exchange needs	131
Annex D – Interaction required with Message broker in the direct mode and in the subscription mode	132
Annex E – Interaction required with Message Broker sending remote procedure calls	133

TABLE OF FIGURES

Figure 1: Inputs to OSSAF	9
Figure 2: The OSSAF framework	10
Figure 3: OSSAF views addressed in SALUS	12
Figure 4: NAF Model elements according to the strategic and operational scope	12
Figure 5 – SOA-Bridge – Connectivity between operational needs and IT	18
Figure 6: SALUS related capabilities of PPDR organisations	21
Figure 7: Operational Nodes and Information Exchange needs	23
Figure 8 – Overview on service categories comprising to the SALUS “Service Map”	31
Figure 9 – Message Brokering Service Interfaces – Overview	40
Figure 10 – Interaction required for a message source communicating with a message sink in the direct mode and in the subscription mode	46
Figure 11 – Interaction required for a message source communicating with a message sink sending remote procedure calls	47
Figure 12 – Force Tracking Service Interfaces – Overview	48
Figure 13 – Required interaction of force track information consumer interested in receiving the information in pro-active (notification) style	57
Figure 14 – Required interaction of force track information provider in order to deliver force track information to the service	58
Figure 15 – Required interaction of force track information consumer interested in receiving the information with client’s request pull style	59
Figure 16 – Sensor Data Acquisition Service Interfaces – Overview	60
Figure 17 – Required interaction between the source components (service consumer) to the Sensor Data Acquisition Service	65
Figure 18 – Required interaction from the source component (service consumer) to the Sensor Data Acquisition Service and the forwarding to the final components (service consumer)	66
Figure 19 – Intrusion Detection Service Interfaces – Overview	67
Figure 20 – Interaction required for an IDS consumer in order to read and set the IDS configuration parameters	70
Figure 21 – Interaction required for an IDS consumer in order to get information about possible intrusion detections either in pull mode or in a pro-active notification mode	71
Figure 22 – Forensic Service Interfaces – Overview	72
Figure 23 – Required interaction of mobile devices on order to register to the Forensic Service	78
Figure 24 – Required interaction for sending forensic data to the service and possible reactions due to detected intrusion attempts	79
Figure 25 – SALUS Dynamic QoS Controller Service Interfaces – Overview	80
Figure 26 – Required interactions for managing the flow QoS	83
Figure 27 – High level SALUS System Architecture	85
Figure 28 – Sensor Data Acquisition Service architecture	97
Figure 29 – Intrusion Detection Service architecture	99
Figure 30 – Message Brokering Service architecture	101
Figure 31 – Force Tracking Service architecture	103

TABLE OF TABLES

Table 1: Mapping of NAF views and templates to OSSAF views	11
Table 2 – Operational Nodes and Information Exchange needs as captured in the previous figure	24
Table 3 – Enterprise Architecture service taxonomy	31
Table 4 – Overview on SALUS derived services.....	34
Table 5 – SALUS common message format for PPDR force tracking service.....	56
Table 6 – Terminal types and network interfaces.....	87
Table 7 – Mapping of SALUS services on terminal / devices	89
Table 8 – Mapping of SALUS services on SALUS Gateways	90
Table 9 – Mapping of SALUS Service onto Tactical Level applications	91
Table 10 – Mapping of SALUS Services in Data Center applications servers	92
Table 11 – Annotations and their corresponding semantics used throughout system descriptions	95

1 Introduction

Public Protection and Disaster Relief (PPDR) organizations are confronted with a growing number of events affecting public safety and security, which require a close cooperation between the involved PPDR organizations in order to respond timely and adequately to these events. The need of cooperation requires aligned procedures and interoperable systems, which allows timely information sharing and synchronization of activities. This, in turn, requires that PPDR organizations adapt Enterprise Architecture methodologies to develop system architectures.

In SALUS deliverable D3.1 [2], a methodology for designing the SALUS Enterprise Architecture was described. This methodology was based on the Open Safety & Security Architecture Framework (OSSAF) and the NATO Architecture Framework (NAF).

The OSSAF provides a framework and approach to coordinate the perspectives of different types of stakeholders within a Public Safety and Security organisation. It aims at bridging the silos in the chain of command and on leveraging interoperability between PPDR organisations.

This document provides the SALUS EA and the SALUS System Architecture (SA) in their intermediate versions. The SALUS EA was defined based on the three SALUS scenarios (City Security, Temporary Protection, and Disaster Recovery) and their underlying Use Cases. The needed capabilities, information flows, and services are derived from the deliverables of WP2 (Use Cases Definition and PPDR Requirements).

The current document is structured as follows: section 2 briefly describes the methodology used for developing the SALUS EA and section 3 provides the main principles which guide the development of the EA.

The EA itself is described in section 4. Section 4.1 portrays the main SALUS related capabilities which must be provided by PPDR organisations in order to fulfil their missions. The operational nodes and information items exchanged between these nodes are described in section 4.2. Section 4.3 introduces the SALUS Service Map as a framework for the Functional Building Blocks and describes a series of services defined within SALUS.

The System Architecture as technical solution is described in section 5. The document concludes with a bibliography and a list of used acronyms.

Annex A provides a detailed list of Information Items exchanged along Information Exchange requirements between the Operational Nodes. Annex B provides a sample of SALUS related capabilities of PPDR organisations. Annex C provides a sample of information exchange needs between operational nodes. Annexes D and E provide samples of required interactions for the Message Broker in specific scenarios.

2 ENTERPRISE ARCHITECTURE DEVELOPMENT METHODOLOGY

This chapter gives a brief overview on the methodical approach to derive an Enterprise Architecture for PPDR organisations.

Looking at PPDR organisations as an “enterprise” means looking at the “enterprise” as the joint undertaking of one or more organisations with Public Safety and Security responsibilities that operate across a distributed and often complex environment. This understanding states, that we see an enterprise in this context as a non-profit-oriented organization or complex structures of organizations (inter-organizational aspect of enterprise definition) such as national PPDR organizations, for example national police or fire-fighter organizations.

To handle the task of developing an Enterprise Architecture for PPDR organizations, [2] used the approach of capability based planning. One can understand a “capability” according to [4] as:

”An ability that an organization, person, or system possesses. Capabilities are typically expressed in general and high-level terms and typically require a combination of organization, people, processes, and technology to achieve.”

Seeing capability based planning as the overarching guideline, our actual approach for the development of the SALUS EA proposes scenarios as main input. Preceding to the definition and development of scenarios, the first step in the development approach is the definition of Visions and Goals in order to depict an overall strategy including the winning of supporters for the overall architecting approach.

The methodology defined in [2] for designing the SALUS EA is based on OSSAF [5]. OSSAF incorporates concepts of several mature enterprise architecture frameworks such as the Zachman Architecture Framework (ZAF) [6], the TOGAF framework [4] and the NATO Architecture Framework (NAF) [7] (see Figure 1).

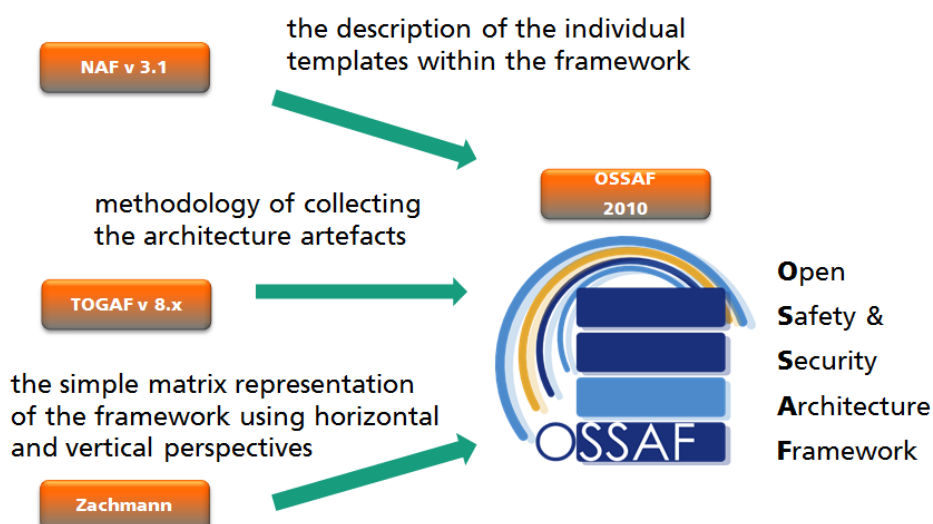


Figure 1: Inputs to OSSAF

OSSAF proposes a total of four perspectives and a total of twenty views (see Figure 2). In general it depends on the intention of the architecture under development which views are actually instantiated. In other words, the views can be tailored to the specific needs of the architecture under consideration.

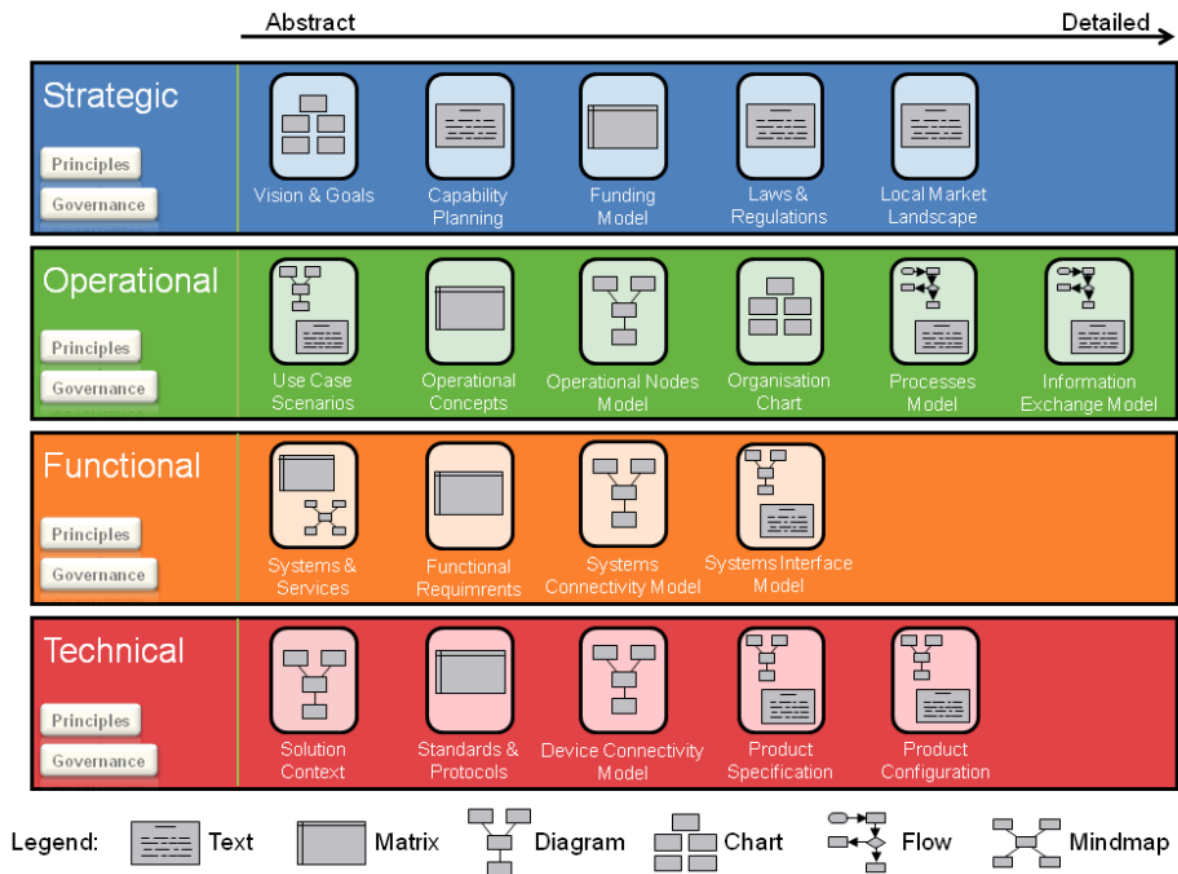


Figure 2: The OSSAF framework

The methodology proposed in [2] for the development of Enterprise Architecture of PPDR organizations, in general and for SALUS specifically, uses NATO Architecture Framework (NAF) as the modelling vocabulary for describing the OSSAF perspectives and views where suitable. The NAF views are modelled with the different elements of the Unified Modelling Language (UML). NAF provides a mature common meta-model to describe the contents of the corresponding views. Every view contains a section of the overall meta-model in order to describe view-specific contents and relations. In addition to the concepts and relationships the meta-model also defines the semantics of each of these elements [7].

The mapping between the NAF views and the OSSAF views and perspectives, i.e. which OSSAF view should be described by which NAF view, is provided in Table 1. Each column represents a perspective defined by the OSSAF framework. The rows represent the views per perspective and their representation using NAF.

Since SALUS does not address all aspects of PPDR organizations, there is a need to tailoring the Enterprise Architecture development and its artefacts to SALUS use cases. The Enterprise Architecture Components addressed in SALUS are the ones highlighted in Figure 3. The NAF views, used to describing the OSSAF views, had also to be adapted to the needs of SALUS. Figure 4 provides an extract from the overall model used for the development of the PPDR Enterprise Architecture (EA). However, for reasons of readability, not all relations, attributes, constraints, and cardinalities are shown. A detailed meta-model description as well as the description of the semantics of each concept and relationship can be found in [7].

Table 1: Mapping of NAF views and templates to OSSAF views

OSSAF Perspective							
OSSAF Views	Strategic		Operational		Functional		Technical
	Vision & Goals	NAV-1 NCV-1	Use Case Scenarios	No proper NAF view	Systems & Services	NSOV-1 NSOV-2 NSOV-3 NSOV-4 NSOV-5 NSV-12	Solution Context No proper NAF view
	Capability Planning	NCV-2 NCV-4	Operational Concepts	NCV-4 NCV-5 NCV-6 NOV-1	Functional Requirements	NSV-2d NSV-4 NSV-5 NSV-6 NSV-7 NSV-10a	Standards & Protocols NTV-1
	Funding Model	No proper NAF view	Operational Nodes Model	NOV-2	Systems Connectivity Model	NSV-1 NSV-2a NSV-2b	Device Connectivity Model NSV-2a NSV-2b NSV-2d
	Laws & Regulations	No proper NAF view	Organization Chart	NOV-4	Systems Interface Model	NSV-1 NSV-2 NSV-3	Product Specification (NTV-1)
	Local Market Landscape	No proper NAF view	Process Model	NOV-5 NOV-6a NOV-6b NOV-6c			Product Configuration NTV-3
			Information Exchange Model	NOV-3 NOV-7			

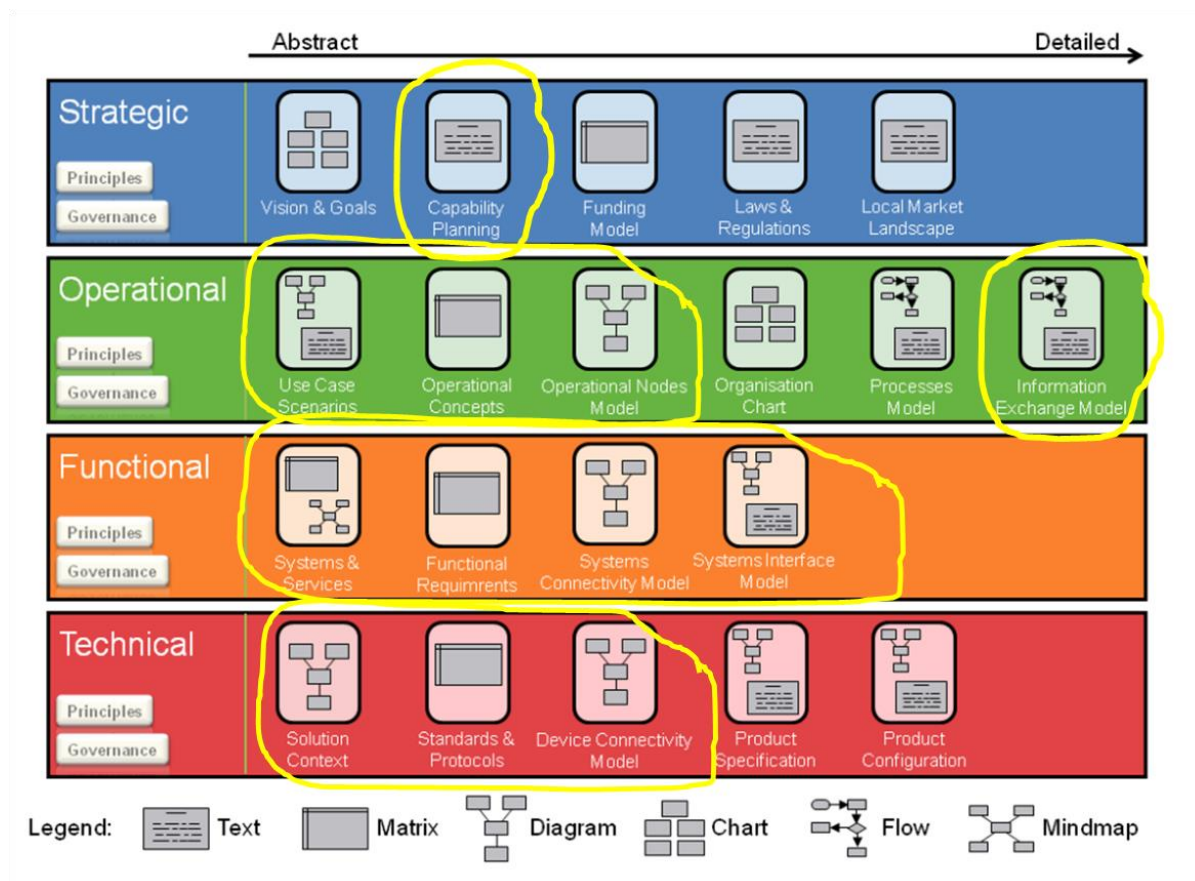


Figure 3: OSSAF views addressed in SALUS

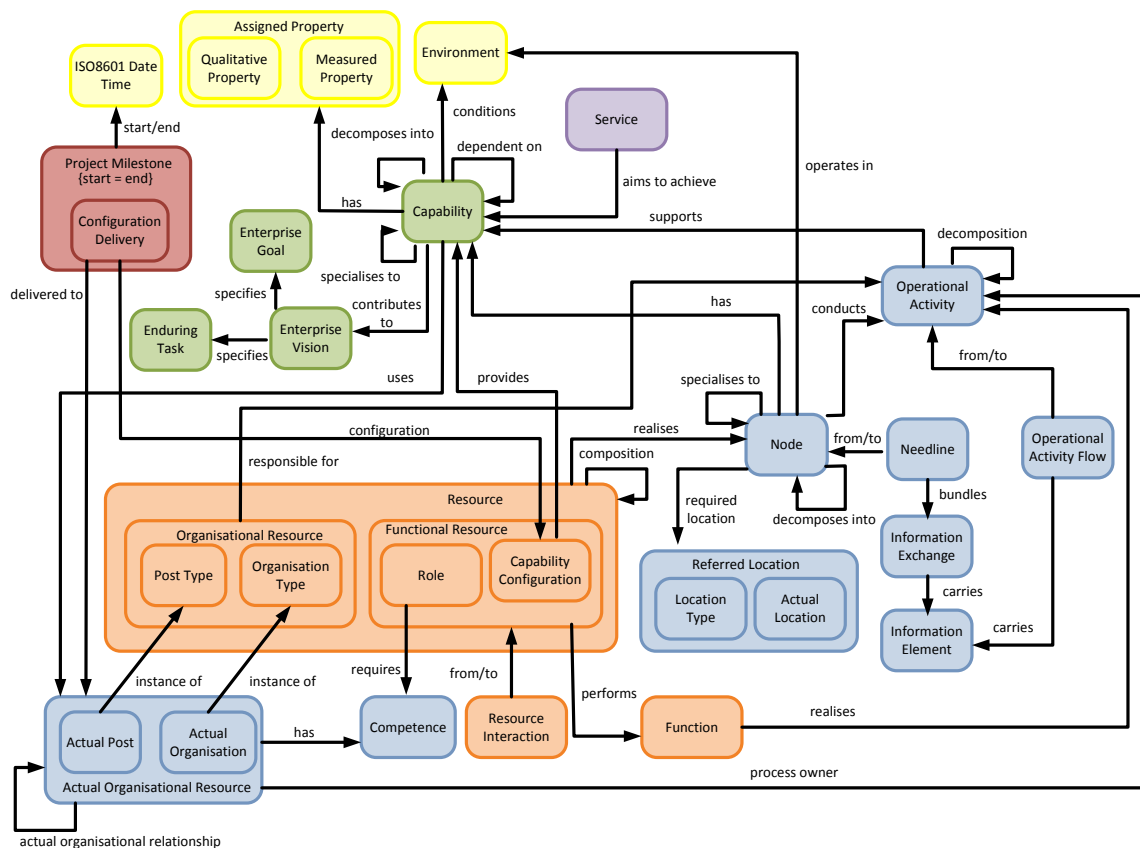


Figure 4: NAF Model elements according to the strategic and operational scope

As previously stated, our approach for the development of an Enterprise Architecture proposes scenarios as main input.

One can see the scenario development as the first concrete step towards enterprise architecture and being crucial for our proposed methodology. Therefore this step requires a very close cooperation with operational end-users. The development is an iterative and interactive process with a successive refinement of operational procedures, and requirements. Scenarios derived include as-is scenarios and also to-be scenarios or a mixture of both of them. The to-be scenarios are particularly important to identify lacks in capabilities. Since the scenarios are developed in close cooperation with end-users, it is assumed that they reflect their needs in a sufficient manner. For the SALUS PPDR Enterprise Architecture, the three scenarios developed and validated within Work Package 2 provide the foundation for the EA development. They comprise (see [3] for a detailed description):

- Scenario 1: City Security
- Scenario 2: Temporary Protection
- Scenario 3: Disaster Recovery

Starting from these scenarios, the further Enterprise Architecture development steps include:

- Define scope and principles of the architecture.
- Define or adapt a common methodology and a common modelling vocabulary (already done in [2]).
- Tailor the architecture perspectives and views (architecture artifacts) according to the project needs.
- Analyse Scenarios/Use Cases on operational Capabilities required. This is a creative process and one has to align current capabilities with further capabilities addressed in order to identify possible lacks of capabilities. As a starting point, the six top-level capabilities of OSSAF may serve for classifying further capabilities derived.
- Analyse Scenarios/Use Cases on functional and non-functional requirements.
- Derive the Capability taxonomy and dependencies (capability based planning concept).
- Identify and analyse operational Context including: Nodes conducting (operational) Activities, Players, Activities, Information flows, Processes and Constraints (e.g. operational rules).
- Identify Services (operational as well as technical services) and Systems in order to support the capabilities, operational requirements and information exchange needs. Take existing technical Standards and Service catalogues under consideration. That means, re-use existing solutions first before developing new ones, being normally not that mature.
- Design Services (including communication services) and Systems (including communication systems) as well as their interactions (logical definitions of systems and services) and derive the conceptual information models from the operational information exchange requirements.
- Define the technical implementation of Services and Systems (physical definition) and a data model corresponding to the conceptual information model.
- Define the standards that have to be considered.
- Finally, validate the architecture with operational-end users and the stakeholders. In general it shall be noticed, that the proposed method is normally conducted in several

iterations after defining the scenarios itself, although it may be necessary to refine them during further architecture development.

- After the final architecture validation, develop a potential migration plan which the decision-makers responsible for the organizational enhancements may adopt.

3 Enterprise Architecture Development – Guiding Principles

This chapter describes the basic principles guiding the Enterprise Architecture (EA) development for PPDR enhancement. An EA should foster the following goals:

- Establish the foundation for a coherent, consistent and technically meaningful decision making process.
- Serve as a basis for discussions and exchanges between different relevant stakeholders.
- Mastering complexity in supporting different viewpoints and modelling steps.
- Establish the foundation for reusability and tool support.

The general guiding principle in EA development is to support the adaptation of business to benefit from technological advancement and also to exploit this technology in order to support the business. Related to SALUS, this means that the SALUS Enterprise Architecture will enable PPDR organizations to benefit from the technology developed in SALUS and utilize the technology development in SALUS to improve the capabilities of PPDR organizations to handling crisis and disasters described in the three scenarios.

3.1 Architecture Scope

In principle the EA should answer the question; **what** functionality is required now and in the future by PPDR organizations focusing on the specific subject area of SALUS. It should logically answer **how** this functionality will be achieved and provide solutions for a technical approach in order to achieve the required functionality, answering the **with what** question [8]. As such, the EA also comprises system specifications i.e. the System Architecture.

Answers to the “what” question provide hints for:

- Medium to long term planning of capability requirements (focus > 6 years).
- Identification of high-level systems, components, standardized products and their interdependencies.

After answering the “what” question, the “how” (narrowing the universe of discourse) question should address inputs for:

- Supporting the development of capability packages.
- Reflecting high-level user requirements, concepts and processes for the derivation of individual projects.
- Focus on services, processes and functionalities in order to guide system development.
- Make strategic decisions on technologies, standards and product lines.

Answering the “how” provides the guidelines for dedicated systems which foster the specific system development and their integration into an overall system (answering the “with what” question) requiring a detailed description/specification of and a commitment to:

- Focus on specific processes, services and component functionalities.
- Detailed system descriptions.

This enables the commissioning of singular components of the overall system-of-systems that reflects the operational utilisation and role concept.

In order to address these issues, the OSSAF views highlighted in Figure 3 are selected to be instantiated describing the intended relationships to address the what, how and with what questions.

3.2 Architecture Style - Service-Oriented Architecture (SOA)

The architecture style proposed for the SALUS Enterprise Architecture is the Service-Oriented Architecture (SOA). Following this architectural style is considered as beneficial as described below:

Public Protection and Disaster Relief (PPDR) organisations are confronted with a growing number of events affecting public safety and security. Since these events may either expand from a local to a regional and to an international scale or may affect multiple countries from the beginning, the pressure on PPDR organisations to be able to cooperate in order to respond timely and adequately to such events increases as well. The need of cooperation demands for aligned procedures and interoperable systems which then allows timely information sharing and synchronization.

Also the nature of events affecting public safety and security is ever changing and every new event has its own characteristics and puts new challenges on PPDR organisations. This in turn requires PPDR organisations to be flexible and to be able to adapt their operations and structures continuously. The same hold true for the information and communication technologies supporting the PPDR operations also taking into account limited budgets and historically grown application/communication landscapes of these organisations into account.

This imposes the design objectives **effectivity, efficiency and agility** both on organisational processes and structures and also on the supporting application/communication landscape with its IT-based realisations.

Regarding the context of application/communication landscapes, operational processes are often supported more or less satisfyingly by already existing, historically grown application/communication landscapes. However, the relationships between those applications and the operational needs (processes and activities) are often unstructured. Similar operational processes or activities are often supported by different applications with redundant functions or the applications don't support new emerging functional and non-functional requirements. The same holds true for the exchange of operational data and information items. Furthermore, if new functionality is introduced, it is often not well defined how the new functionality can be accessed from the operational level. Application/communication landscapes that reveal such characteristics do not comply with the design objectives **effectivity, efficiency and agility**. There is a lack between operational needs and supporting IT.

To bridge the gap between IT and operational needs, and to ease the management and integration task SOA complement a technical oriented approach with an operational viewpoint on the application/communication landscape. The aim is to structure application/communication landscapes along operational needs. These needs are the operational activities and/or workflows and their inherent information exchanges. The latter becoming more and more important, as operations tend to shift from application centric to information centric.

In order to achieve this, SOA breaks the monolithic application systems into individual functional blocks. It introduces a systematic structuring of these blocks and tries to establish a goal-oriented assembly, being loose coupling as additional design objective in order to react to new requirements in a flexible fashion.

A new layer of abstraction between the operational level and the IT, the “SOA-Bridge”, is introduced. This layer contains the services. Services are artefacts describing how they support or partly automate operational information exchanges, operational activities and/or workflows of an organisation. In general these service-artefacts are technology independent.

The support of operational information exchanges is equally important as supporting operational activities or workflows. This is because, it is not always possible to derive the latter and they are more likely subject to change than the underpinned information exchanges.

Finally, the realisation of these service-artefacts takes place within the physical architectures and systems of the communication/application landscape.

The following two subsections provide the definition of the SOA core-concept, the service, applicable to SALUS and the concept of the SOA-Bridge along with its components and modelling elements.

3.2.1 SALUS Service Definition

The most generic view on a service is to understand it as an implementation-independent specification of a packaged element of repeatable functionality which provides a value added to the actor consuming the service. As such a service is a self-contained black-box to the consumer with a specified outcome which is executed under certain, well defined constraints up to an agreed level of quality. A service may be composed of other services.

Normally the service description comprises the specification how services are orchestrated for a certain purpose to provide advanced functionalities a single service solely cannot provide, the capabilities that services deliver and how they are going to be implemented.

In the general SOA approach, a service can be realized and provided by a mixture of different resources. Those resources comprise for example organisational resources (roles, posts and whole organisational structures), physical assets, IT-systems (software/hardware combinations) and so on. The same holds true for the service consumer. It may be a simple IT-system or a whole organisational structure.

The focus on services in the SALUS context is only on those services that can be solely supported/realized by software/hardware combinations (IT). Services, (in the general SOA approach which treats the concept of a service in a broader approach, see above) that require also organisational resources for their provision, are not part of the service considerations within SALUS. Therefore SALUS only deals with “application services” or “technical services”, realized by IT components rather than on “operational services” which involve additional organisational resources to provide the expected functionality/results to the consumer. A service (in the general sense and also in the SALUS sense) provides an external view on systems solely from the consumers’ perspective and therefore defines the possible interaction between the consumer and the service. The internal structure and comprised processes of such a service (realized by a dedicated system or system of systems) is irrelevant to the consumer. Nevertheless, a system reveals an internal structuring with a timely ordered sequence of activities in order to provide the desired outcome. These activities may also partly or completely

- A service comprises a logical definition of repeatable functionality with defined outcomes.
- A service provides its functionality and outcomes to a certain level of quality.
- A service is self-contained.
- A service is provided by a combination of software/hardware resources.
- A service acts as a “black box” to the service consumer.
- A service may be composed of other services.
- A service may have several implementations and instantiations during runtime providing the same or different Quality of Service (QoS).

3.2.2 SOA-Bridge

The diagram illustrates the SOA Bridge concept, showing the relationship between Business and Systems and IT domains. The Business domain (left) includes Business Activity, Business Service, and Business Process. The Systems and IT domain (right) includes Physical Application Architectue, System, Software, Physical Interface, and Logical Interface. The SOA Bridge (center) connects these domains, facilitating the delivery and implementation of services. Key relationships include: Business Activity uses Business Service, which is realized via Business Process. Business Service is (partially) automatized by the Application Service. The Application Service provides via Logical Interface, which requires the Physical Interface. The Physical Interface realizes the Physical Application Architectue, which exposes the Physical Interface. The Physical Application Architectue is composed of resources into the System and Software. The System hosts the Software, which is a software component. The System and Software are both composed of resources. The Application Service is grouped within the service taxonomy.

Elements of the SOA-Bridge:

- **Application Service:** Logical component, abstracts dedicated implementation (see 3.2.1 for a detailed description).
- **Software:** Executable computer code/programme, reusable across architecture(s).
- **System:** Man made physical asset (e.g. computer, hardware etc.), reusable across architecture(s).

- **Physical Application Architecture:** Forms reusable architecture and depicts structure and interactions - may be used as configurations within other physical architectures - it defines a self-contained unit within a defined environment composed of Systems and Software.
- **Logical Interface:** Implementation independent specification of service operations (organized in interfaces) that are provided to the service consumer in order to access the services functionality.
- **Physical Interface:** Implementation of the logical interface provided by a system/software combination where the service operations defined within the logical interfaces can be accessed.

The service related elements of the SOA-Bridge will be captured in several views of the architectures service viewpoint (with additional refinements) and are shown in chapter 4.3 and its sub-chapters. The description of the Systems and IT related elements is given within the system architecture part in chapter 5 and its sub-chapters. The operational elements will be described in chapter 4.2. The descriptions will focus on information exchanges underlying operational procedures and activities.

4 SALUS Enterprise Architecture

Following the capability based planning approach as the overarching guideline for developing the SALUS Enterprise Architecture, the scenarios and uses cases developed in WP 2 were used to derive the main needed capabilities related to SALUS which PPDR organisations have to provide in order to fulfil their missions.

After identifying and describing the main capabilities related to SALUS, the operational nodes and their information exchange needs were identified and described.

The Information Items exchanged along the Information Exchange flows between the different operational nodes are produced, consumed and exchanged via a set of services. The different service categories together with the service specifications are provided clustered in a taxonomy called 'The Service Map'. The services represent the functional building blocks for the SALUS System-of-Systems Architecture. Each service is realized by one or multiple individual interconnected systems working together.

The different views of the SALUS Enterprise Architecture are provided in detail in the following sections.

4.1 SALUS related PPDR Capabilities

Starting from the scenarios and use cases defined in [3], the following SALUS related main capabilities of PPDR organisations were identified (see also Figure 6):

- Public Protection - the capability to protect the public, citizens and properties
- Integrated Mission Conduction - the capability to conduct a mission in an integrated way
- Force Protection - the capability to protect the own forces

These capabilities rely on others, like assessing the development of a crisis or the capability to coordinate and cooperate, which in turn depend on the capabilities to generate and maintain the situation awareness. The latter capability depends on the capability to exchange and share information in an efficient way within an organization or agency and between organizations and agencies (Efficient Intra- and Inter Agency Information Exchange/Sharing). This capability depends again on another capability, to 'Provide Access to Common Information Infrastructure Services'. Finally, this capability depends on the capability 'Provide Communication Connectivity' (see Figure 6 and for improved readability also check Annex B – SALUS related capabilities of PPDR organisations). Providing Communication Connectivity is the main capability supported and implemented by SALUS.

The capability providing communication connectivity enables the various PPDR operational nodes to communicate with each other, exchange information and thus cooperate in order to handle a crisis and to protect the citizens.

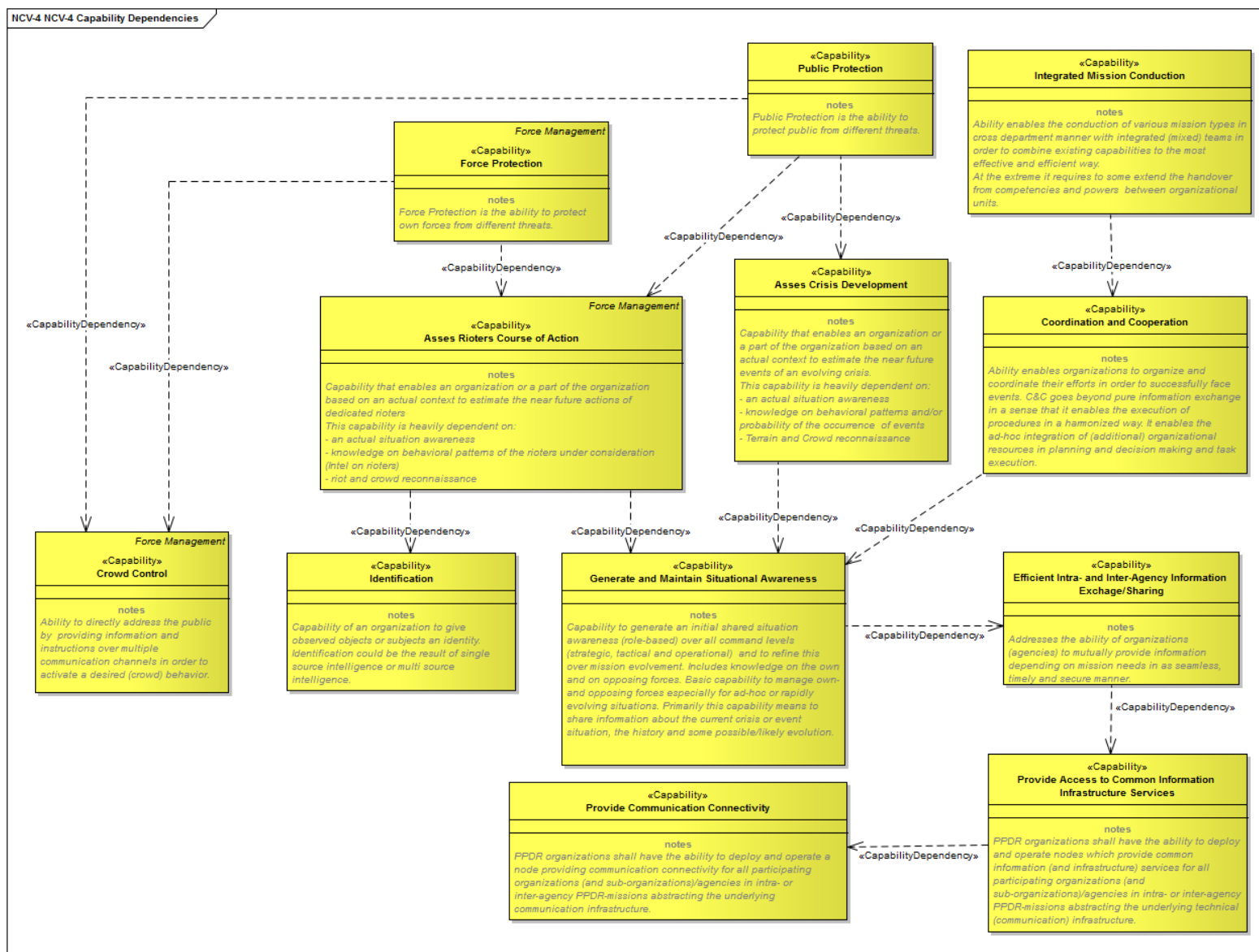


Figure 6: SALUS related capabilities of PPDR organisations

4.2 Operational Nodes, Information Exchanges and Information Items

The capability Providing Communication Connectivity enable the various PPDR operational nodes, like the command and control centres on the different command levels (strategic, tactical, and operational on the field) to communicate with each other, exchange information and thus cooperate in order to handle crisis and to protect the citizens. The PPDR operational nodes and the information exchange needs between the different operational nodes are provided in Figure 7 and for improved readability see Annex C – Operational Nodes and Information Exchange needs.

The Information Items exchanged between the different Operational Nodes along the information flows called Needlines are listed in Table 2. A detailed description of the Information Items and the Information Exchange characteristics is provided in Annex A – Information Items and Information Exchange requirements.

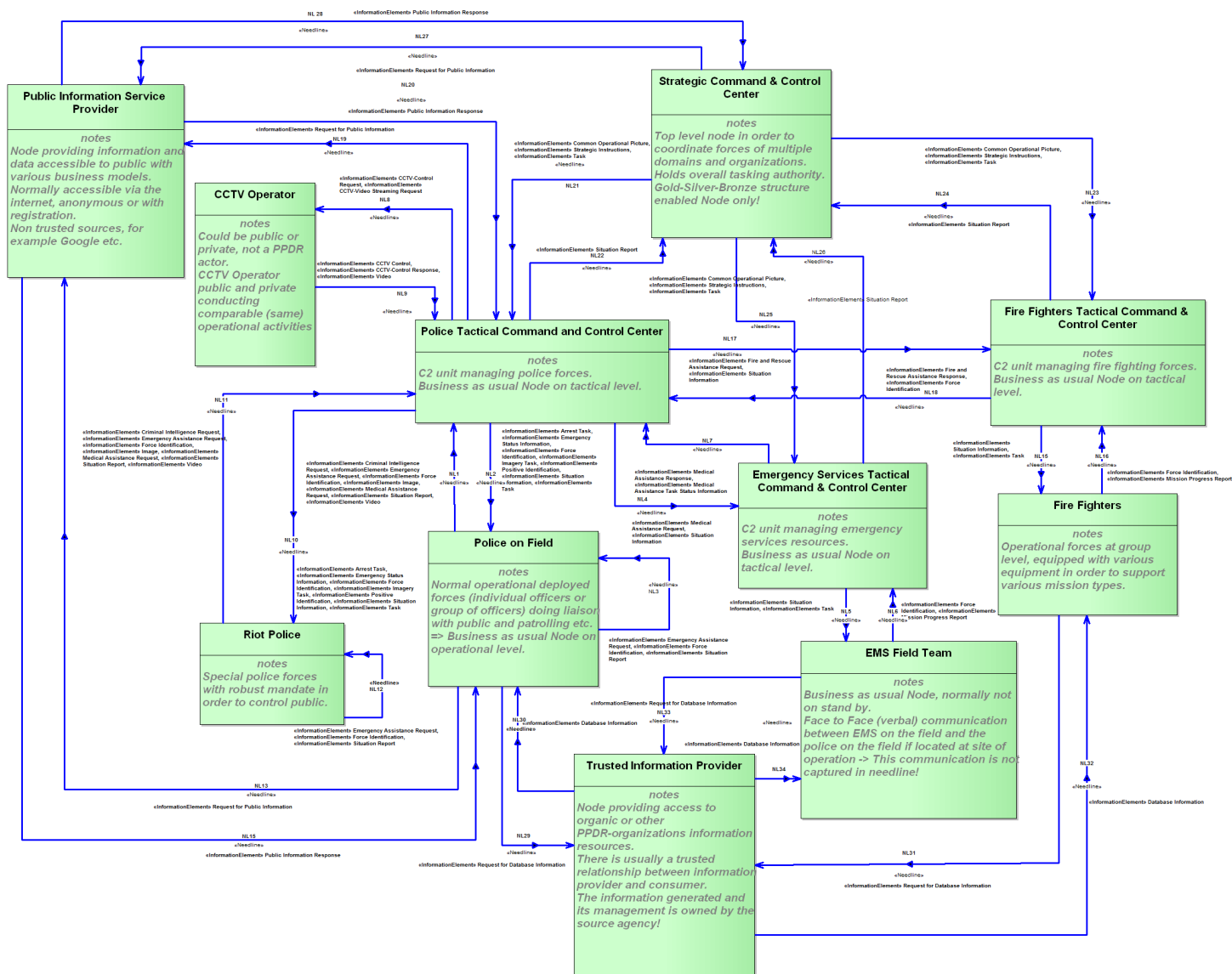


Figure 7: Operational Nodes and Information Exchange needs

Table 2 – Operational Nodes and Information Exchange needs as captured in the previous figure

Needline Id	Sending Node	Receiving Node	Information Items conveyed
NL1	Police on Field	Police Tactical CCC	Force Identification
			Situation Report
			Emergency Assistance Request
			Medical Assistance Request
			Sensor Data (Video)
			Image
			Criminal Intelligence Request
NL2	Police Tactical CCC	Police on Field	Task
			Force Identification
			Emergency Status Information
			Situation Information
			Imagery Task
			Arrest Task
			Positive Identification
NL3	Police on Field	Police on Field	Force Identification
			Situation Reports
			Emergency Assistance Request
NL4	Police Tactical CCC	EMS Control Room	Medical Assistance Request
			Situation Information
NL5	EMS Tactical CCC	EMS Field Team	Task
			Situation Information
NL6	EMS Field Team	EMS Tactical CCC	Mission Progress Report
			Force Identification

NL7	EMS Tactical CCC	Police Tactical CCC	Medical Assistance Response
			Medical Assistance Task Status Information
NL8	Police Tactical CCC	CCTV Operator	CCTV-Control Request
			CCTV-Video Streaming Request
NL9	CCTV Operator	Police Tactical CCC	CCTV-Control Response
			Sensor Data (Video)
			CCTV Control
NL10	Police Tactical CCC	Riot Police	Task
			Force Identification
			Emergency Status Information
			Situation Information
			Imagery Task
			Arrest Task
			Positive Identification
NL11	Riot Police	Police Tactical CCC	Force Identification
			Situation Report
			Emergency Assistance Request
			Medical Assistance Request
			Sensor Data (Video)
			Image
			Criminal Intelligence Request
NL12	Riot Police	Riot Police	Force Identification
			Situation Reports
			Emergency Assistance Request
NL13	Police on Field	Public Information Service Provider	Request for Public Information

NL14	Public Information Service Provider	Police on Field	Public Information Response
NL15	Fire Fighters Tactical CCC	Fire Fighters	Task
			Situation Information
NL16	Fire Fighters	Fire Fighters Tactical CCC	Mission Progress Report
			Force Identification
NL 17	Police Tactical CCC	Fire Fighters Tactical CCC	Situation Information
			Fire and Rescue Assistance Request
NL18	Fire Fighters Tactical CCC	Police Tactical CCC	Fire and Rescue Assistance Response
			Force Identification
NL19	Police Tactical CCC	Public Information Service Provider	Request for Public Information
NL20	Public Information Service Provider	Police Tactical CCC	Public Information Response
NL21	Strategic CCC	Police Tactical CCC	Strategic Instructions
			Common Operational Picture
			Task
NL22	Police Tactical CCC	Strategic CCC	Situation Report
NL23	Strategic CCC	Fire Fighters Tactical CCC	Strategic Instructions
			Common Operational Picture
			Task
NL24	Fire Fighters Tactical CCC	Strategic CCC	Situation Report
NL25	Strategic CCC	EMS Tactical CCC	Strategic Instructions
			Common Operational Picture

			Task
NL26	EMS Tactical CCC	Strategic CCC	Situation Report
NL27	Strategic CCC	Public Information Service Provider	Request for Public Information
NL 28	Public Information Service Provider	Strategic CCC	Public Information Response
NL29	Police on Field	Trusted Information Provider	Request for Database Information
NL30	Trusted Information Provider	Police on Field	Database Information
NL31	Fire Fighters	Trusted Information Provider	Request for Database Information
NL32	Trusted Information Provider	Fire Fighters	Database Information
NL33	EMS Field Team	Trusted Information Provider	Request for Database Information
NL34	Trusted Information Provider	EMS Field Team	Database Information

CCC = Command & Control Center

4.2.1 Operational Processes (Workflows)

This sub-chapter aims to describe operational workflows to be supported by the SALUS solutions. The intention was to describe the following workflows:

- Deployment of mobile communication equipment
- Determination of additional bandwidth requirements during running ops
- Establishment of Cross-Domain Talk Groups
- Request for and deployment of additional PPDR forces during running operations
- Ad-hoc collaborative planning and decision-making

This is still an ongoing process which requires additional end user input and feedback to provide meaningful results. The results achieved so far are not in a quality sufficient to be presented within this deliverable.

4.3 Service Map – A Framework for Functional Building Blocks –

4.3.1 Purpose

In order to deal with the functions required to support operations in a common fashion and to address their management issues, a framework is needed that provides structure and aligns these functional building blocks, aka services, from the planning perspective and also from the usage perspective. Within SALUS the “Service Map” is a proposal for such a framework. It is expected to support the development of common functionalities across organisations fostering interoperability.

To do so, this framework will define service taxonomy along with their respective classification criteria and a domain-based structuring. With its different levels of abstraction it will support mastering the complexity that application/communication landscapes reveal, applying the “Divide-and-Conquer” principle.

In addition, it is expected that this “Framework” will provide these additional benefits such as:

- Common capabilities that can be used by other applications, systems or services without the need to reinvent the wheel by these consumers,
- Re-use of functionality,
- Facilitated interoperability by ensuring a common baseline,
- The key to interoperability is to have well defined descriptions and interfaces,
- Provide/maintain consistency within the technical environment while defined as IT services.

In general, the Top-level category within this service taxonomy is the SALUS service. The next level of refinement structures the service landscape into domains – operations scope, community of interest (COI) scope etc. The most detailed level is the service level which represents a logical structuring of the applications into logical functional building blocks describing “what” the delivered functionality is. It does not specify “how” it is done. The latter is defined within the physical architecture providing a vendor independent, coarse grained blueprint how a specific functionality shall be realized in order to fit into an overall system and software approach (Physical Application Architecture) and enforces reusability of systems as sub-systems.

4.3.2 Service Categories and Methodology for Service Derivation

The following three sub-chapters describe the approach for characterizing, identifying and classifying services according to the concept of a “Service” outlined in the previous chapters. Its intention is to provide reproducible description towards what is subsumed under the concept of “Service Categories” resulting in a dedicated taxonomy of services providing basic toolbox for managing functionality across the enterprise.

4.3.2.1 Service Derivation Methodology

After the clarification of what one has to understand under the concept of a “Service” and having defined the criteria characterizing a service the next logical step is to define a methodology for deriving service candidates. As already stated, a service is a piece of functionality provided by a technical system or a combination of technical system and organisational resources with a dedicated responsibility for the operation providing an add-on to other organizational resources. As such, each service has to provide an added value either on strategic, tactical or operational or across all levels.

Therefore, a methodology for deriving services that has to take the operational conditions into account is the main driving force. As an additional complication, all services derived shall be generic in a sense that they support different types of operations (scenarios) adaptable to different QoS parameters and environments. Taking these requirements into account, a four-step methodology has been derived. This methodology is mainly inspired by the experiences/lessons learned gained throughout architecting solutions supporting network centric operations in the military domain. Using the NAF meta-model as a foundation for describing architectural and systemic interrelationships, as supposed by the OSSAF framework, this methodology is somewhat straightforward.

The service derivation methodology includes the following steps:

- **Step 1** - This step comprises the formulation and identification of representative scenarios. Having WP 2 as precursor work-package along with its resulting scenarios which achieved verification by end users across various PPDR-organizations and across various EU member states, WP 2 provides the required input.
- **Step 2** – Based on the scenarios identify the operational nodes and their required information exchanges in order to handle their duties. This includes the derivation of requirements bound to the information exchange such as required bandwidth, security constraints, media types, timeliness, volume etc. This step also includes the derivation and description of information items acting as possible service parameters for the services to be derived. Of special interest are the recurrent information exchanges throughout the scenarios as they provide hints for cross scenarios information exchanges and therefore for generic services.
- **Step 3** – This step comprises the identification and definition of the operational activities involved either to produce or to consume the dedicated information items. Again, the focus is on the identification of recurring activities as they provide hints for cross scenario functionalities and capabilities to be supported by dedicated services.
- **Step 4** - Identification of services supporting the execution of these operational activities and providing the required capabilities to the operational nodes. Having defined the criteria characterizing a service and a service taxonomy eases and supports this activity. This step also includes the classification, description and definition of the services, their interfaces, service functions, parameters etc. along with their QoS parameters.

All service candidates derived from the four step process are further refined and classified according to the service taxonomy outlined in chapter 4.3.2.2. Additionally a detailed description of each individual service is provided in chapter 4.3.3. As mentioned, this process is straightforward when using the overall EA approach. A more or less comparable approach, defined independently, is described in [1] supporting the validity of this approach.

4.3.2.2 Service Domains and Service Taxonomy

The SALUS “Service Map” is grouped into five domains or scopes, namely the “Operations” domain, the “Community of Interest” domain, the “Service Management” domain, the “Information Assurance” domain and last but not least the “Network and Information Infrastructure” domain as shown in Figure 8.

The introduction of domains is considered to be useful in order to manage the complexity of functional building blocks and to provide a way to assign responsibilities and to foster communication between operational users and people responsible for the communication/application infrastructure at a very high level of abstraction. The domains structure the application/communication landscape into coarse grained functional areas roughly corresponding to the capabilities defined in SALUS.

The domains are the basic foundation of the SALUS “Service Map” framework and span the boundaries for the Service Portfolio identified within SALUS. It is to say, that this portfolio is not complete and it is assumed that it is subject to change over time (evolutionary changes) since it is introduced here for the first time.

In principle, there is a functional dependency from the Operations domain to the Community of Interest domain; the latter in turn depends on functionalities from Network and Information Infrastructure domain. The **Service Management** and the **Information Assurance** domain provide cross-domain functionalities representing non-functional requirements from the user’s perspective. This issue is depicted within Figure 8 showing these two domains as vertical pillars spanning the Community of Interest and the Network and Information Infrastructure domain.

The Service Maps **Operations Scope** comprises the real operational function lists, operational activities or operational workflows. The services of this domain provide a dedicated outcome and may involve a combination of organizational-, material- and/or IT-resources and are not purely technical services. They can roughly compare to the business processes of an enterprise providing a dedicated outcome to the customer or dedicated business units. They are not part of SALUSs services but depend heavily on the functions provided by SALUS defined services.

The **Community of Interest** domain comprises services that are user specific by providing functionalities unique to the user while being more use-case specific and context sensitive than the services within the underlying Network and Information Infrastructure domain.

The **Network and Information Infrastructure** domain covers the Information and Communication Technology aspects of SALUS defined capabilities. As such, it contains the functional building blocks dealing with network and transport functions as well as basic functions for information integration and dissemination.

In addition to these domains, **service taxonomy** is defined/derived. The taxonomy, hosting the actual services, provides a more fine grained classification schema at a lower level of abstraction than the domains do. The criteria for classifying a service within this category are given in the following section while presenting the taxonomy. The taxonomy is shown in Table 3. In general services whose names are written in italic type are abstract services. That means

that these kinds of services only act as classifying elements without direct implementations. Nevertheless these services can define attributes, operations, constraints and/or QoS-parameters which will be inherited from the derived services (direct or indirect). An example may be the definition of a SALUS-wide service management interface for starting, pausing, resuming and stopping a service instance. Actually none of these features are defined on abstract service, so the inheritance feature of the taxonomy is not yet used.

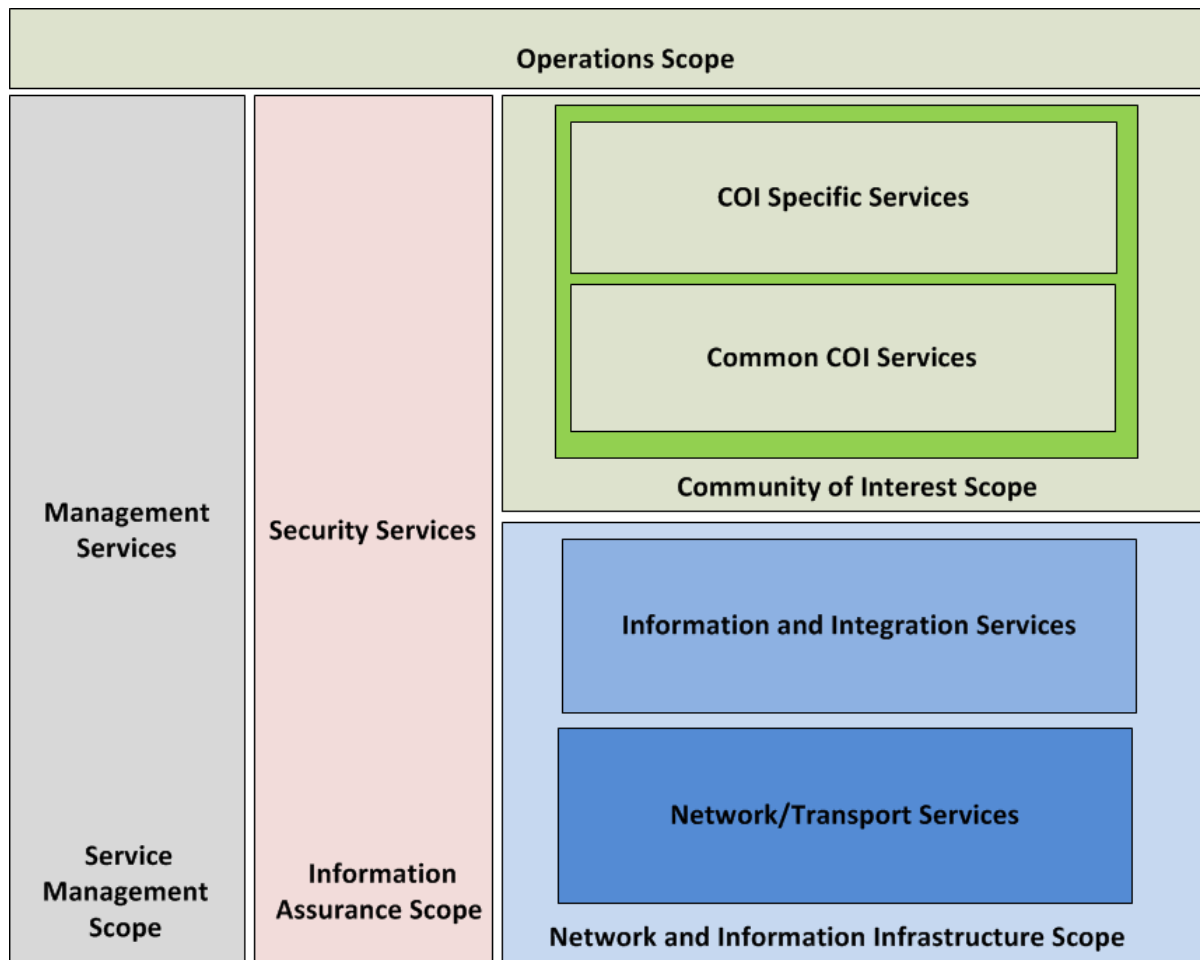
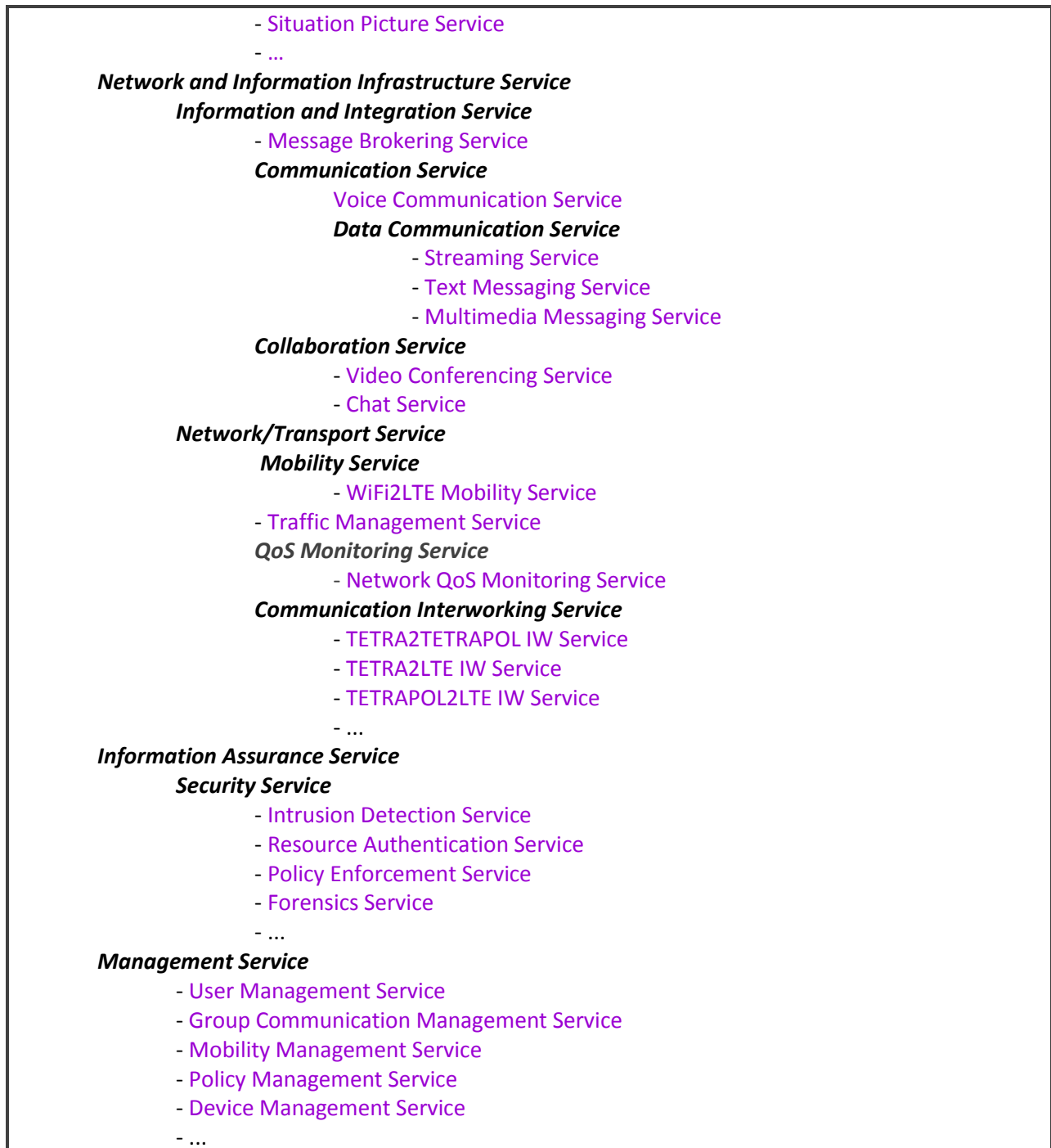


Figure 8 – Overview on service categories comprising the SALUS “Service Map”

Table 3 – Enterprise Architecture service taxonomy

SALUS Service
Community of Interest Service
COI Specific Service
- Indoor Localization Service
- Status Monitoring Service
- Number Plate Recognition Service
- ...
Common COI Service
- Sensor Data Acquisition Service
- Sensor Control Service
- Force Tracing Service
- Image Dissemination Service
- Video Dispatching Service



4.3.2.3 Service classification criteria

SALUS Service:

The top-level service category is the “**SALUS Service**” category. It just defines the anchor point for the subsidiary services stating that the latter being derived within the SALUS project. So the only criteria for a service being a SALUS Service is to be defined within the SALUS project and to be a service in the SALUS sense (refer to chapter SALUS Service Definition). No additional criterion applies to this category.

The second category level comprises the **Community of Interest (COI) services**, the **Network & Information Infrastructure services**, the **Information Assurance services** and the **Management services**.

Community of Interest Services:

These kinds of services provide unique functionalities to the user. This category is further subdivided into the **COI Specific services** category and into the **Common COI services** category.

COI Specific services provide unique functionalities that are very specific to the actual operational context. Services of this kind normally provide functions tailored to PPDR organizations and operations type needs, for example specific services for fire fighters, paramedics or police forces applied to very specific situations. The services of this category have undergone the most functional diversification.

The Common COI services are less sensitive to the operations context than the COI Specific services but provide functionalities that are more user specific than the Information and Integration services. Common COI services provide functionalities required by multiple COI across multiple operation and organization types.

Network & Information Infrastructure Services:

This category is further subdivided into the Information and Integration services category and into the Network & Transport services category. Both categories have in common that they support basic functions for communication and information exchange.

The **Information and Integration** services are services of general interest, providing a generic functionality. Generic in that sense, that they provide functionalities that do not have to be implemented by individual systems or applications. They comprise functionalities useful to other services, applications, systems and users on the operational network. They act as “building blocks” for more sophisticated functional/application services of the Community of Interest service domain. In principle they can be regarded as a framework for developing additional services, being independent of operational activities/processes and context. They will provide/maintain consistency within the technical environment while defined as core IT services.

Subcategories of the Information and Integration services category are the **Collaboration services** and the **Communication services** categories. Both provide functionalities abstracting the underlying transport layer, either air interface or wired based. The latter provide basic communication functionalities by supporting either voice-based communication or data-centric communication. Services of the Collaboration services category are dedicated services fostering a group-based multi-party collaboration and decision-making providing advanced functionalities of multimedia content sharing, application sharing and a dedicated session management and multi-user capabilities.

The **Network & Transport services** categories comprises basic services for connecting different network technologies by providing particular technology transitions, extending existing networks and provide monitoring support for the transport quality of service assurance. **Mobility services** extend networking capability by providing ad-hoc extensions of preexisting, either-wire or air-interface based, network infrastructure. The **Communication Interworking services** category comprises services for the connectivity of actually existing communication systems in use between themselves and/or with new systems. They mediate functionality of existing services and their providing technologies; either the full functions stack or partially if a mapping between functions is possible and required. Services within this category may exist mainly

during the transition phase while different existing technologies that have to be bridged. They disappear in the final phase when functionality and technology have been harmonized towards the one dedicated technology that spans PPDR communication requirements and a migration to a common technology has been accomplished.

Network & Transport services category may require additional definitions of mediation and decoding services.

The Information **Assurance services** category and the **Management services** category are directly hooked beyond the top-level category. They are agnostic to the previous service categories so that they cover integral functionality incorporated within every service domain from the Network and Information domain up to the Community of Interest domain. As such these kinds of services have no direct outcome to the user. They have an impact on the other “application services” and therefore provide indirect output to the user.

Information Assurance Services:

Services of this category contribute to the security of other services and the overall system in two ways. First, they provide functions separated from the individual service implementations in order to provide reusable functionality that has not to be implemented by each individual service of other categories. For example, the latter can delegate authentication issues to the Resource Authentication Service instead. Secondly, services of the Information Assurance category can also be self-contained. As such they provide security mechanisms to the overall system and may indirectly support the security of the other service instances within their dedicated runtime environment. An example for such a self-contained service is the Intrusion Detection service (IDS). IDS instances can be operated stand-alone and provide a certain level of security to the communication network they are responsible for.

Management Services

This service category comprises services for the management and controlling of services within the other service categories. The functionalities provided may be used during a system setup and configuration phase but also during run-time for the management and configuration of services in order to react to changing mission needs. They allow for a centralized approach for the management and configuration of a distributed system organized into services.

The following Table 4 summarizes the services derived in SALUS. Only for those services with explicit statement in the last column, a detailed specification will be given in chapter 4.3.3. These descriptions shall act as a blueprint for specifying additional services within the PPDR realm.

Table 4 – Overview on SALUS derived services

Service name	Service description	Detailed specification available in this deliverable
Indoor Localization Service	Service for localization of resources (PPDR forces) in space and time; the forces operating inside enclosed spaces with no GPS signal available. Supports push and pull mode.	No
Status Monitoring Service	Service aggregates and provides information on the state of PPDR force individuals based on several sensor data	No

	inputs in near real time.	
Number Plate Recognition Service	Service that retrieves information of the vehicles registered keeper based on an image taken from the vehicle showing the number plate (in reasonable resolution) or at least the number plate itself. Input data comprise individual still images or video clips.	No
Sensor Data Acquisition Service	The service offers the functionalities of exchanging messages from the end user devices with sensor information to the command and control centre and the processing of the information with a data base collecting and alerts notification. Relies on a generic data format for sensor data acquisition.	Yes
Sensor Control Service	Enables the control of sensors by sending dedicated parametrised control messages. Message type and structure along with possible parameters must be known prior to invocation. Service provides no means of gathering possible sensor parameters during runtime but it provides feasibility study functionality in order to check if a dedicated parametrized control message will be accepted and executed by the corresponding sensor.	No
Force Tracing Service	The service provides a way to exchange and track information on PPDR operational resources/operational PPDR forces down to the field level. Supports push and pull mode; relies on a specific message format with dedicated semantic defined.	Yes
Image Dissemination Service	Service for dispatching selected still images to (multiple) dedicated consumers in push mode. May be automatically reducing image resolution/data volume according to network characteristics not less than a minimum required image resolution/quality; abstracting from underlying network technologies.	No
Video Dispatching Service	Service for dispatching selected video clips (no real time) to (multiple) dedicated consumers in push mode. May automatically reduce video resolution/data volume according to network characteristics, abstracting from underlying network technologies.	No
Situation Picture	Service for maintaining and sharing (non-	No

Service	real time to near real time) of a common recognized picture. Relies on a an situation object model with dedicated semantics in order to describe objects, situations and their activities comprising to the overall situation picture. Works on and exchanges self-contained situation objects with guaranteed message delivery.	
Message Brokering Service	Message Brokering Service provides message exchange (non-real time to near real time) capabilities for various actors within the PPDR realm. Several message senders and receivers can interact with the service in parallel. Supports one-to-one, one-to-many and many-to-one as well as many-to-many message exchange patterns.	Yes
Voice Communication Service	Basic service for voice based communication. Provides access to the following functions. - Push-to-Talk function - Group Call function - 1-to-1 Call function - Emergency Call function - Priority Call function - Late entry - Broadcast call - Ambience/Discrete Listening function - Dynamic Group Management (within and across PPDR organizations) - ...	No
Streaming Service	Content agnostic service for network transparent transport between a data stream source and a data stream sink. May include the capability to (semi-) automatically adopt the data rate according to network current utilization or policies. Point-to-Point transport or point-to-multipoint, push mode, near to real time transmission support.	No
Text Messaging Service	Service for basic exchange and notification of text based messages supporting one-to-one and one-to-many message exchange patterns. In addition to SMS protocol support may also support Email (POP, IMAP, and SMTP) protocols.	No
Multimedia Messaging Service	Service for basic exchange and notification of multimedia based messages (files –e.g. maps-, pictures, videos) supporting one-to-one and one-to-many message exchange patterns.	No

Video Conferencing Service	Advanced multi-party real time collaboration service supporting the synchronized streaming of video, audio and shared workspaces. May only provide a subset of functionalities in disabled environments like mobile networks May be capable of policy based data rate adjustment.	No
Chat Service	Service for a real time transmission of short text messages from sender to receiver(s) supporting presence indication via dedicated protocols for example the Extensible Messaging and Presence Protocol (XMPP).	No
WiFi2LTE Mobility Service	Service for the mediation of functionality of existing services between Wi-Fi and LTE networks. Mapping of functionalities not yet defined.	No
Network QoS Monitoring Service	Service for monitoring and ensuring that the services and applications deployed have appropriate parameters (including priority). QoS of applications can be dynamically configured and changed per (group of) user(s) according to mission objectives and constraints.	Yes
TETRA2TETRAPOL IW Service	Service for the mediation of functionality of existing services between TETRA and TETRAPOL networks. This mainly includes the capability to enable voice communications and text messaging service between TETRA and TETRAPOL.	No
TETRA2LTE IW Service	Service for the mediation of functionality of existing services between TETRA and LTE networks. This mainly includes the capability to enable voice communications and text messaging service between TETRA and LTE Optionally, this may include the capability to include in voice only mode TETRA users in a video-conferencing services.	No
TETRAPOL2LTE IW Service	Service for the mediation of functionality of existing services between TETRAPOL and LTE networks. This mainly includes the capability to enable voice communications and text messaging service between TETRA and LTE Optionally, this may include the capability to include in voice only mode TETRAPOL users in a video-conferencing services.	No
Intrusion Detection	The Intrusion Detection Service analyses	Yes

Service	network traffic of dedicated networks or network segments on network scans, brute-force attacks against SSH daemons and compromised SSH daemons.	
Forensics Service	Service for supervising mobile devices on intrusion attempts, detection, alerting and subsequent management of these devices in near real time.	Yes
User Management Service	This service provides all the required information to be provisioned in the system for each user (credentials, authorisations, capabilities...) to enable each user to conduct his mission.	No
Group Communication Management Service	This service provides all the required information to be provisioned in the system for each user (credentials, authorisations, capabilities...) to enable each user to conduct his mission.	No
Device Management Service	This service ensures the correct and coherent configuration (OS version, applications, peripherals...) of all the devices in used in the system. It also provides remote configuration (including remote activation) and eventually remote disabling / wipe of memory in case of lost/stolen terminal.	No

4.3.3 Service Specifications

From a functional perspective, SALUS provides a series of services needed for the capabilities to generate and maintain situation awareness, to provide and access common information infrastructure and to provide communication connectivity. These services are clustered and grouped within the service taxonomy as shown in Table 3 (written in non-italic type). Some of them will be described within this chapter providing a blueprint character. The refinement is done for a couple of services comprising to the SALUS first and second prototype platform.

The detailed service specification comprises an initial service description (purpose description) which provides a high level general definition of the service functionality. This description is further detailed via service interfaces (provided and required) and service operations definitions along with required interaction patterns. Additionally service attributes (and service quality attributes – QoS Parameters) as well as interface operations performance requirements and additional constraints on service attributes are specified.

The service definition is done in an implementation agnostic style according to the service concept. Specific implementation details will be specified within the system architecture part of the document. The description is an initial approach and will be further extended in the final deliverable. Additional service description refinements comprise:

- Service level specification
- Service State Machine specification (describing service states if any),

- Service Information Model specification (Information exchanged via service interface),
- Service composition specification,
- Service functionality specification (providing service internals on how specific service operations shall be realized).

The service descriptions within this document provide an aggregated view on a service. The full linkage across different architectural views and modeling concepts is not shown here, but provided within the modeling environment along with additional details.

<<Message Brokering Service>>

Purpose:

The Message Brokering Service provides message exchange capabilities for various actors within the PPDR system. Several message senders and receivers (service consumers) can interact with the service via a dedicated interface in parallel.

The message broker service runs on a well-known network address and represents a central point to which dedicated service consumers connect: all messages are therefore first sent to the service and from that point routed to final recipients. This allows service consumers connected to the service to asynchronously exchange messages without having to know the network address of recipients; some clients, terminals in particular, may often switch networks, so their network addresses may not always be known in advance.

The service offers two types of message exchanges: a direct one (direct mode for exchanging control messages) and a subscription based (subscription mode for exchanging data messages). In the direct mode, the sender explicitly denotes the recipient, sends the message to the service, which then forwards it to the recipient. In the subscription based message exchange, the sender tags the message with a topic, sends it to the service, which then forwards the message to all recipients that have previously subscribed to that topic. In this case, the Service performs the application layer multicast. This type of message exchange is also known as Publish/Subscribe pattern and allows for an information push capability. In addition, a message exchange with remote procedure call (RPC) semantics is also provided, allowing for “brokered” RPC calls on dedicated resources remotely. The methods invoked on remote resources have to be known beforehand; the service provides no functionality to discover remote resources interfaces (during the run-time).

The following summarizes the interfaces that the service provides or requires in order to interact with its consumers. During runtime an API-based communication between the service and the service consumers takes place, supported by programming language specific software libraries.

Service Summary:

In general the Message Brokering Service handles six interfaces (IF) as shown in Figure 9. The “Inbound Control IF”, the “Inbound Data IF” and the “RPC Response IF” are provided by the service and allow service consumers to access functionality of the service. The “Outbound Control IF”, the “Outbound Data IF” and the “RPC Result IF” are interfaces required by the service (and provided in turn by the service consumers) in order to respond to consumers that have invoked operations on the services provided interfaces.



Figure 9 – Message Brokering Service Interfaces – Overview

Service Name	<<Message-Brokering-Service>>
Interface Description Standard	None
Communication Protocol supported	ZMTP over TCP
Communication Method	Using dedicated API
Message Format	ZMTP messages

Service operations (methods of accessing the service) organized in provided or used (P/U) interfaces:

Interface Name	IF Description	Provided/ Used (P/U)	IF Operation Name	Communi- cation style SYNC/ ASYN	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
Inbound control interface	This interface allows service consumers to send control messages to the service. The service then forwards these messages to intended recipients.	P	connect(address, port, identity)	ASYN	Connects the service consumer to the message broker service.	{String} address (IN)	The network address of the message broker service.
						{INT} port (IN)	The port of the message broker service.
						{String} identity (IN)	The identity of the consumer.
			disconnect()	ASYN	Disconnects the service consumer from the message broker service. All existing subscriptions are automatically cancelled.		
			sendControlMessage (identity, payload)	ASYN	Sends a control message to message broker service that forwards the message to the recipient. Control messages correspond to direct message exchange.	{String} recipient (IN)	The identity of the recipient to whom the message is sent.
						{ByteArray} payload (IN)	The contents of the message. The contents can be of any type as long as it can be converted to an array of bytes.
			sendRPC(recipient, method, parameters, timeout)	ASYN	Invokes a remote procedure call on intended recipient. The call is sent to the message broker service and from then on forwarded to intended recipient.	{String} recipient (IN)	The identity of the consumer to which the RPC is sent.
						{String} method (IN)	The method that will be called on recipient.
						{Dictionary[Key, Value]} parameters (IN)	The list of named parameters for the given method. The parameters are provided in a dictionary where dictionary keys

Interface Name	IF Description	Provided/Used (P/U)	IF Operation Name	Communication style SYNC/ASYNC	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
							correspond to parameter names and dictionary values to parameter values.
						{INT} timeout (IN)	The amount of time after which the RPC is considered to expire. After timeout, the originator of the RPC receives a timeout message.
Outbound control interface	This interface allows service consumers to receive messages that are sent to them by the service; the service calls these methods on the consumers.	U	onControlMessageReceived(sender, payload)	ASYNC	A method that is invoked when the consumer receives a control message.	{String} sender (IN)	The identity of the sender that sent the control message
						{ByteArray} payload (IN)	The contents of the control message.
			onRPCReceived(method, parameters)	ASYNC	A method that is invoked when the consumer receives an incoming RPC request.	{String} method (IN)	The name of the method to invoke.
						{Dictionary[Key, Value]} parameters (IN)	The dictionary of parameters for the method.
Inbound data interface	This interface allows service consumers to send topic tagged data messages to the service. The service then forwards these messages to recipients that have previously	P	sendDataMessage(topic, payload)	ASYNC	Sends a data message tagged with a given topic to the message broker service that then forwards it to all consumers that have previously subscribed to that topic. Data messages correspond to subscription based message exchange.	{String} topic (IN)	The topic of the message that is sent out.
						{ByteArray} payload (IN)	The contents of the message that is sent out. The contents can be of any type as long as it can be converted to an array of bytes.
			subscribe(topic)	ASYNC	Subscribes the consumer to receive data messages	{ByteArray} topic	The topic to which consumer wishes to subscribe. If the caller

Interface Name	IF Description	Provided/ Used (P/U)	IF Operation Name	Communication style SYNC/ ASYNC	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
	subscribed to those topics.				that are tagged with given topic.	(IN)	is not unauthorized to subscribe to given topic, the message broker service sends back a notification as a control message.
			unsubscribe(topic)	ASYNC	Unsubscribes itself from receiving data messages tagged with given topic	{ByteArray} topic (IN)	To topic from which the consumer wishes to unsubscribe itself.
Outbound data interface	This interface allows service to forward data messages to consumers.	U	onDataMessageReceived(topic, payload)	ASYNC	The method is called when a data message tagged under given topic is received. The precondition is that the consumer has previously subscribed to that topic.	{ByteArray} topic (IN)	The topic under which this message was published.
						{ByteArray} payload (IN)	The contents of the message.
RPC response interface	This interface allows consumers that have received incoming RPC requests to send out RPC responses	P	respondWithResult(result)	ASYNC	With this method, the service consumer, which received an RPC request, signals to the RPC initiator that the RPC request was successfully processed and returns a result.	{Object} result (IN)	The result of the RPC call.
			respondWithError(error, message)	ASYNC	With this method, the service consumer, which received an RPC request, signals to the RPC initiator that the RPC request was unsuccessfully processed and returns an error with a message.	{INT} errorCode (IN)	The type of error that occurred.
						{String} message (IN)	A short description of the error.
RPC result interface	This interface allows service consumers,	U	onResult(result)	ASYNC	This method is invoked when a previously sent RPC has completed	{Object} result (IN)	The result of the RPC as was returned by the recipient.

Interface Name	IF Description	Provided/Used (P/U)	IF Operation Name	Communication style SYNC/ASYNC	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
	which have sent out RPC requests, to receive RPC responses.				successfully.		
			onError(error, message, recipient, parameters, timeout)	ASYNC	This method is invoked when a previously sent RPC has completed with an error.	{INT} errorCode (IN)	The type of error that has occurred.
						{String} message (IN)	The description of the error.
						{String} recipient (IN)	The recipient to which the RPC was sent.
						{String} method (IN)	The name of the method that was invoked.
						{Dictionary[Key, Value]} parameters (IN)	The dictionary of parameters that were sent.
						{INT} timeout (IN)	The timeout that was set.
			onTimeout(recipient, parameters, timeout)	ASYNC	An RPC call-back method that is invoked when a previously sent out RPC has not been responded to within given time.	{String} recipient (IN)	The recipient to which the RPC was sent.
						{String} method (IN)	The name of the method that was invoked.
						{Dictionary[Key, Value]} parameters (IN)	The dictionary of parameters that were sent.
						{INT} timeout (IN)	The timeout that was set.

Interface Operations Performance Requirements:
None specified!

Service attributes (performance parameters), functional and non-functional parameters characterizing a service:

Service Attribute Name	Service Attribute Description	Quantitative [YES/NO]	Measure (if quantitative =TRUE)	Measurement Method
Availability	The ability to perform over a stated period of time	YES	The uptime percentage	$\frac{t_{up}}{t_{up} + t_{down}}$
Local caching	Client's ability to cache the number of sent messages until they are successfully delivered to the message broker service	YES	Number of cached messages	N

QoS-Parameters: specific Service Attributes comprising to the overall Service QoS – refer to “Service Attributes” –

Service Attribute Name
Availability
Local caching

Service Interaction Pattern:

To be described if a service is stateful or requires a sequence of interactions: - only communication with external agents aka “service consumers” should be taken under consideration -

The following figures (Fig. 10-11) illustrate the interactions required for a message source communicating with a message sink in the direct and in the subscription mode (Figure 10) and also the interactions required for a message source communicating with a message sink sending remote procedure calls (Figure 11). For improved readability see Annex D – Interaction required with Message broker in the direct mode and in the subscription mode and Annex E – Interaction required with Message Broker sending remote procedure calls.

Note that the illustration uses ordinary UML sequence diagrams for time-ordered representation of interactions. The message source and the message sink are both consumers of the Message Brokering Service.

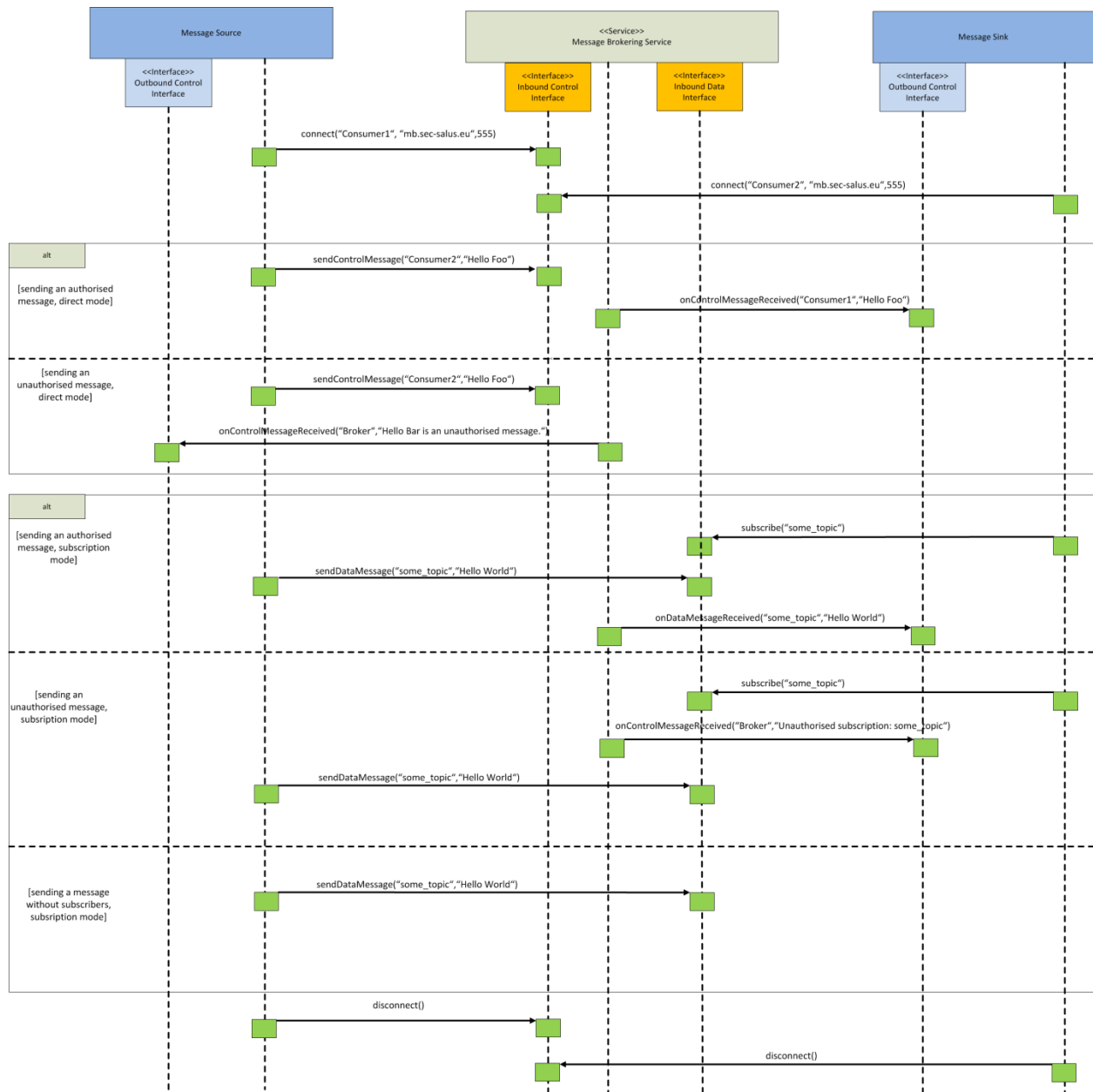


Figure 10 – Interaction required for a message source communicating with a message sink in the direct mode and in the subscription mode

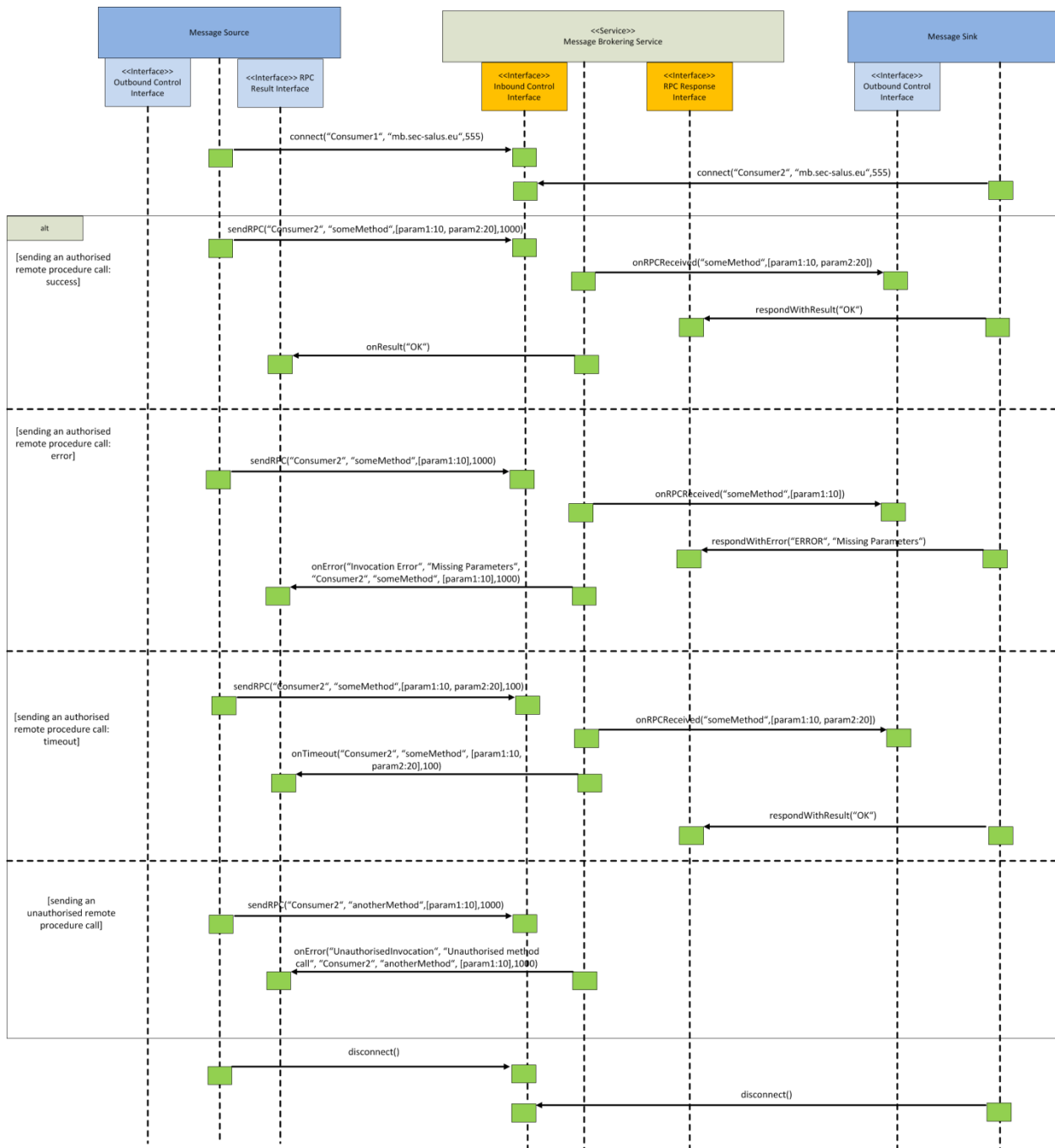


Figure 11 – Interaction required for a message source communicating with a message sink sending remote procedure calls

<<Force Tracking Service>>

Purpose:

The Force Tracking Service provides a way to exchange and track information on PPDR operational resources/operational PPDR forces down to the field level (e.g. individual firefighters or police officers).

The force tracking information is a highly aggregated type of information providing information on PPDR resources regarding their state and their position in space and time along with the

PPDR resources history. This kind of information is derived from variable input data sources, for example sensor readings, but it is also possible to infer this kind of information semi-automatically with a human in the loop.

The Force Tracking Service relies on this aggregated information as input and provides a management capability for this kind of information across different information sources and consumers accessing these. The service supports two kinds of interaction patterns. A publish/subscribe based message exchange and a pull-based message exchange where the service consumer pulls the service in order to receive force tracking information.

The following summarizes the interfaces that the service provides or requires in order to interact with its consumers.

Service Summary:

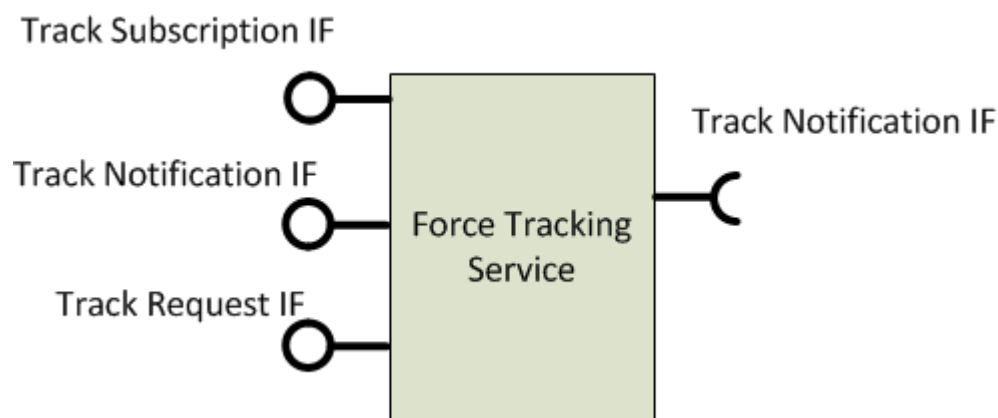


Figure 12 – Force Tracking Service Interfaces – Overview

Service Name	<<Force Tracking Service>>
Interface Description Standard	WSDL
Communication Protocol supported	HTTP
Communication Method	SOAP
Message Format	XML

Service operations (methods of accessing the service) organized in provided or used (P/U) interfaces:

Interface Name	IF Description	Provided/ Used (P/U)	IF Operation Name	Communicati on style (SYNC/ASY NC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
Track Subscription IF	Interface for Force track Information providers and consumers to register their interest in Force Tracking information	P	registerTrackInform ationSource	SYNC	Operation to register a ppdr resource providing their force track information. Only messages from previously registered sources will be recognized	{Literal} resourceID (IN)	Unique ID identifying the ppdr resource. Format must be defined and agreed according to operations needs
						{URL} endpointAddress (IN)	Parameter indicating the endpoint address to which the messages shall be sent via notification mechanism from the information source to the service listening on this interface
						{Exception} SourceRegistration Exception (OUT)	Indicating whether a source was not accepted and successfully registered to the service. No further justification will be given, null otherwise
			registerToTrackInfo rmation	SYNC	Operation to register a sink for getting force tracking information	{Literal} subscriptionID (OUT)	Parameters globally identifying the information sink registered within the service context. Defined and agreed according to operation's needs.
						{Exception} RegistrationExcepti on (OUT)	Exception to be thrown, if the message sink interested in force information was not successfully registered. In case of denial no

							justification will be given
						{Literal} subscriptionID (IN)	Id identifying the message sink, previously achieved via successful registration
						{URL} endpointAddress (IN)	Parameter indication the endpoint address to which the messages shall be sent via notification mechanism and the information sink provides the notification interface
						{Array} resources[{Literal} resourceID] Optional(IN)	Array indicating the resources for which the subscriber will receive track information- optional, if not specified (parameter has to be set to "null") Force Track information will be received from all registered sources.
						{Array} resourcesServed[{L iteral} resourceID] (OUT)	Array indicating the resources for which the subscriber will receive track information. May be a subset of the resources specified within the optional parameter "resources"
						{Exception} SubscriptionExcepti on (OUT)	Exception to be thrown, if the message sink interested in force information was not successfully registered. In case of denial no justification will be given
			subscribeToTrackIn formation	SYNC	Method for registering consumer (endpoint) in order to receive information pro- active via notification interface. Throws exception, if the sink could not be registered		

			setNotificationInterval	SYNC	Set the time interval within you want to receive force track information, if the time interval expires, no further track information will be sent to the subscriber	{Array} Interval[start, end] {DateTime} start,end (IN)	Time interval indicating for which the consumer is interested in receiving force track information, start = interval begin, end = interval end, after expiring no further track information will be sent to the consumer
						{Boolean} success (OUT)	Indicating whether the time interval was accepted or not. In case of denial no justification will be given
			renewSubscription	SYNC	Operation for renewing a subscription while a valid subscription is still active for this track information consumer	{Literal} subscriptionID (IN)	Parameters globally identifying the information sink registered within the service context. Defined and agreed according to operation's needs. Achieved during previous successful registration.
						{Array} Interval[start, end] {DateTime} start,end (IN)	Time interval indicating for which the consumer is interested in receiving force track information, start = interval begin, end = interval end, after expiring no further track information will be sent to the consumer
						{Array} resources[{Literal} resourceID] (IN)	Array indicating the resources for which the subscriber will receive track information
						{Array} resourcesServed[{Literal} resourceID] (OUT)	Array indicating the resources for which the subscriber will receive track information. May be a subset of the resources specified within the optional

							parameter "resources"
						{Exception} SubscriptionException (OUT)	Exception to be thrown, if the message sink interested in force information was not successfully registered. In case of denial no justification will be given
			unsubscribeToTrackInformation	SYNC	Method indicating to the service that a force track information consumer is no longer interested in receiving information via notification, also de-registers the message sink. After calling this operation a new registration is necessary in order to access track force information again.	{Literal} subscriptionID (IN)	Id identifying the message sink, previously achieved via successful registration
						{Boolean} success (OUT)	Indicating whether a message sink was successfully removed from the receivers list
			deRegisterTrackInformationSource	SYNC	Method for de-registering the track information sources. Clients polling or subscribed to this resource will achieve no more force tracking information. They will not be notified for de-registration of the information source.	{Literal} resourceID (IN)	Unique ID identifying the ppdr resource. Format must be defined and agreed according to operations needs
						{Boolean} success (OUT)	Indicating whether a message source was successfully de-registered.
						{Exception} DeRegistrationException (OUT)	Exception to be thrown, if the message source to be de-registered is not yet or no longer registered within the service context

			deRegister ToTrackInformation	SYNC	Method de- registers the message sink. After calling this operation a new registration is necessary in order to access track force information	{Literal} subscriptionID (IN)	Id identifying the message sink, previously achieved via successful registration
						{Boolean} success (OUT)	Indicating whether a message sink was successfully de-registered.
						{Exception} DeRegistrationExce ption (OUT)	Exception to be thrown, if the message sink to be de- registered is not yet or no longer registered within the service context
Track Notification IF	Interface for pro-active force track message delivery	P/U	notifyForceTracking Information	SYNC	Method for transmitting force track information messages to interested receivers	{Message} forceTrackMessage (OUT)	Message containing track items. Message defined in: ForceTrackMessageDefiniti on.xsd
						{Boolean} success (OUT)	Indicating whether the information sink accepted the force track message
Track Request IF	Interface for pull-based force track message delivery	P	getResourceTrackI nformation(resource eID)	SYNC	Operation provides the last known track information of the resource specified with its ID or null if there exists no location report for that resource. Throws exception, if the resource specified via the resourceID does not exist within the context of the service (not registered or already de-	{Literal} resourceID (IN)	Unique ID identifying the ppdr resource. Format must be defined and agreed according to operations needs
						{Message} forceTrackMessage (OUT)	Message containing only a single track item or null. Message defined in: ForceTrackMessageDefiniti on.xsd

					registered). To get the resource location a client has to pull the service.	{Exception} NoSuchResource Exception (OUT)	Exception to be thrown, if the resource specified via the resourceID does not exist within the context of the service (not registered)
--	--	--	--	--	---	---	--

Interface (IF) Operations Performance Requirements:

IF	IF Operation Name	Performance Requirement	Metric	Measure	Reference (Description from where this constraint originates)
Track Request IF	getResourceTrackInformation(resourceID)	Operation response time	msec	< 300	empiric

Service attributes (performance parameters), functional and non-functional parameters characterizing a service:

Service Attribute Name	Service Attribute Description	Quantitative [YES/NO]	Measure (if quantitative = TRUE)	Measurement Method
minimum location co-ordinate accuracy	Attribute describing the location accuracy a PPDR's location can be estimated and contained within the force tracking information	YES	meters	empiric
availability	Umbrella term to also include serviceability, resilience, reliability and maintainability. The ability of a IT service (under combined aspects of its reliability, maintainability and maintenance support) to perform its required function at a stated instant or over a stated period of time	YES	%	$\text{availability} = \frac{\text{Uptime}}{(\text{Uptime} + \text{Downtime})} \times 100$
Delivery retries	Attribute describing the attempts a service executes in order to deliver a force track information message to s subscribed message sink	YES	Number of delivery retries	<i>N</i>

QoS-Parameters: specific Service Attributes comprising to the overall Service QoS – refer to “Service Attributes” –

Service Attribute Name
Minimum co-ordinate accuracy
availability

Message Format:

The dissemination of force tracking information relies on a common format in order to achieve interoperability on syntactic and also on a semantic level. The main message elements of the message format used throughout the force tracking service is shown in Table 5. The overall message format is technically defined using the Extensible Markup Language (XML) schema language (out of scope of this document).

Table 5 – SALUS common message format for PPDR force tracking service

Message Element	Semantics
Heartbeat	If true this Item represents a Heartbeat signal. Normally if there are no data collected in a predefined time interval (defined as an operational rule as the Minimum delay in seconds between two consecutive updates) the track message is not be sent at the interval expiration. If the heartbeat is "true" the data provider always to sends a message (also if empty) at the interval expiration time (again this requires an operational agreement)
Resource Position	Element contains a set of elements which all together constitute positional data of a resource in space and time. Every element type is defined separately.
Resource Identification	Element contains a set of elements which all together constitute to the identification of the resource that is associated with data in the Resource Position element. Usually needed for a standard situation display. Every element type is defined separately
Resource Operational Status	Element provides information on operational status of the PPDR resource associated with data in Resource Position section. Various pre-defined state definitions are available.

Service Interaction Pattern:

To be described if a service is stateful or requires a sequence of interactions: - only communication with external agents aka "service consumers" should be taken under consideration -

The following figures (Fig. 13-15) illustrate the interactions required for a service consumer in order to receive information from the service in a pro-active (Figure 13) fashion, for a service consumer to feed the service with force track information (Figure 14) and for a service consumer to get force track information from the service in a pull style (Figure 15).

Note that the illustration uses ordinary UML sequence diagrams for time-ordered representation of interactions. The source and the sink are both consumers of the Force Tracking Service.

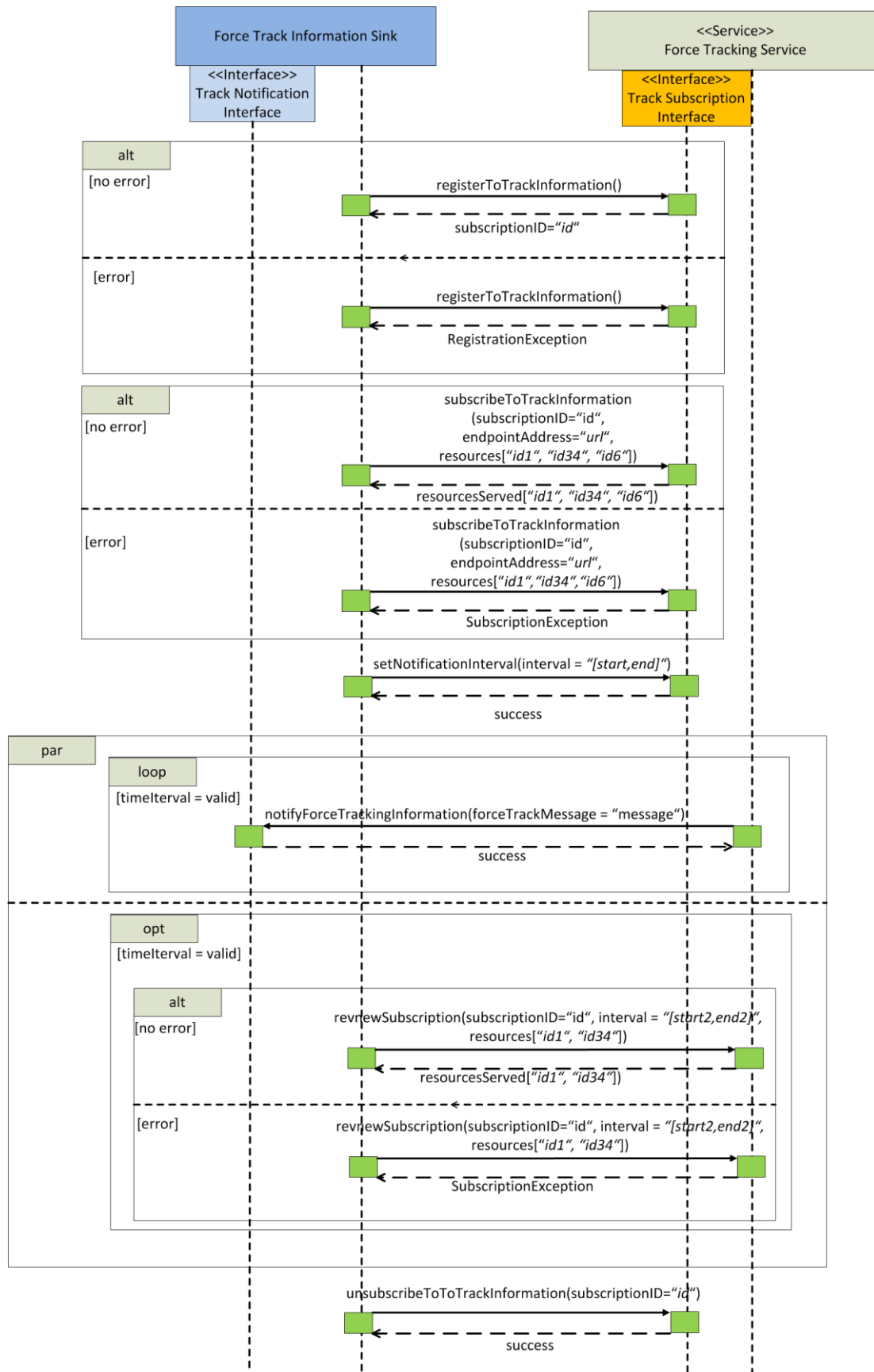


Figure 13 – Required interaction of force track information consumer interested in receiving the information in pro-active (notification) style

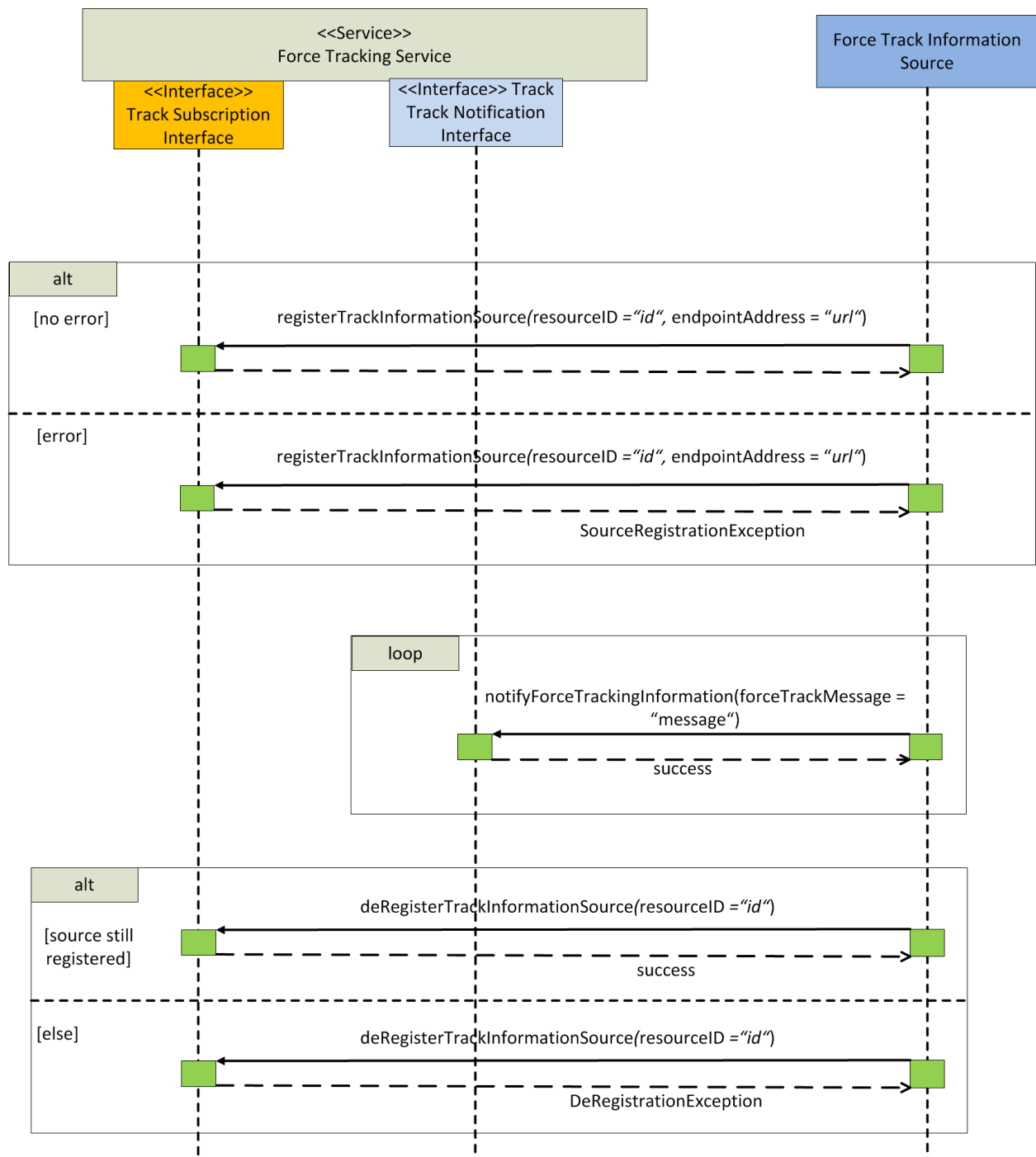


Figure 14 – Required interaction of force track information provider in order to deliver force track information to the service

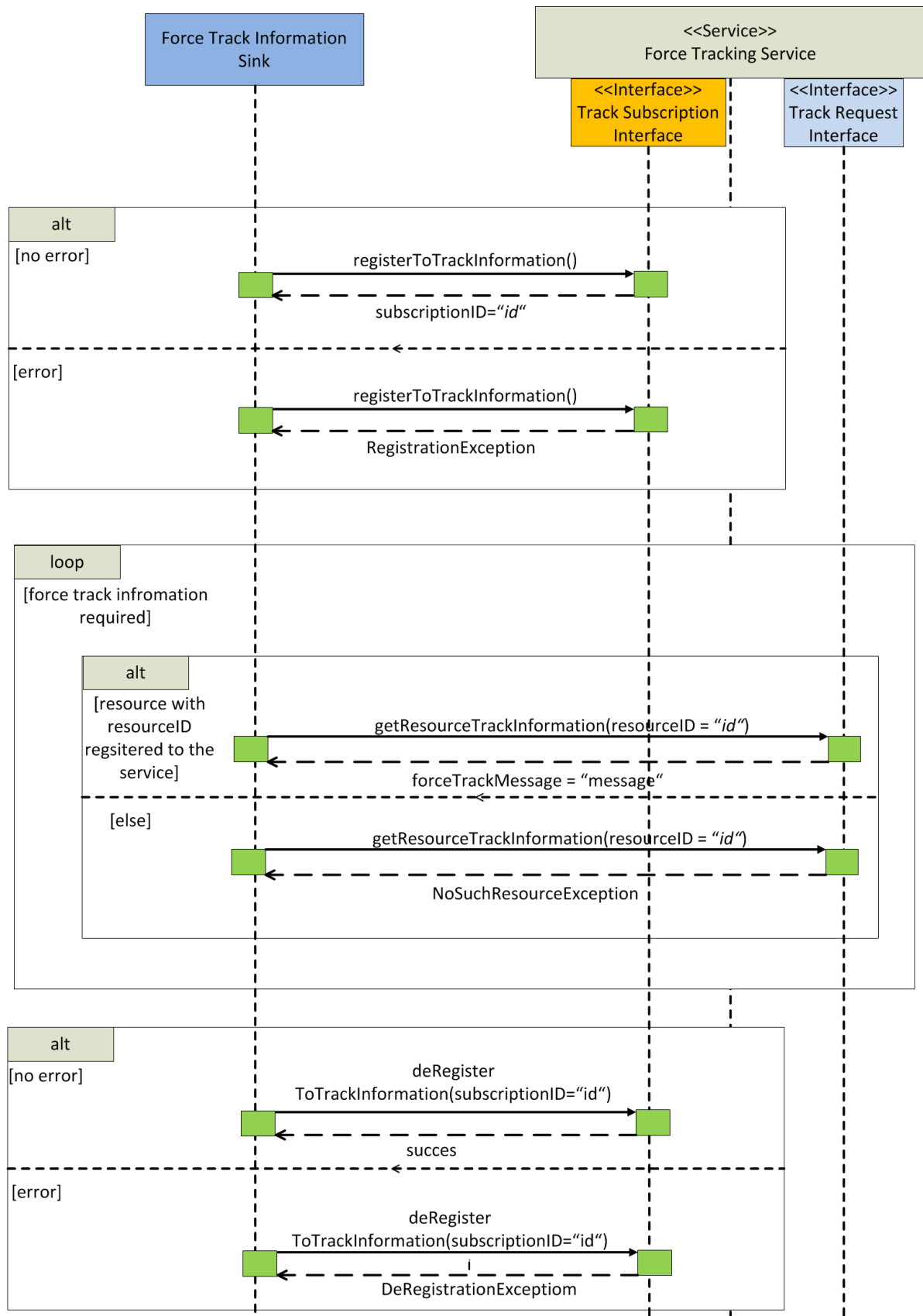


Figure 15 – Required interaction of force track information consumer interested in receiving the information with client's request pull style

<<Sensor Data Acquisition Service>>

Purpose:

The Sensor Data Acquisition Service provides processing and exchange capabilities for various actors within the PPDR system. Multiple devices which gather information from sensors and from other devices and user specific applications. For example a dedicated control command centre application, can interact with the service through a dedicated interface in parallel.

The service offers two types of functionalities: the exchange of messages from the end user devices with sensor information to the command and control centre; and the processing of the information with the data base collecting and alerts notification.

The Sensor Data Acquisition Service runs on a well-known network address and represents a central point to which dedicated gather information devices and control command centre connect. All the messages are therefore first sent to the Service and from then routed to final recipients. This allows the components to connected to the service, asynchronously exchange messages without having to know the network address of recipients.

The following summarizes the interface that the service provides to its consumers. During runtime an API-based communication between the service and the service consumers takes place. This run-time communication is supported by programming language specific software libraries.

Service Summary:

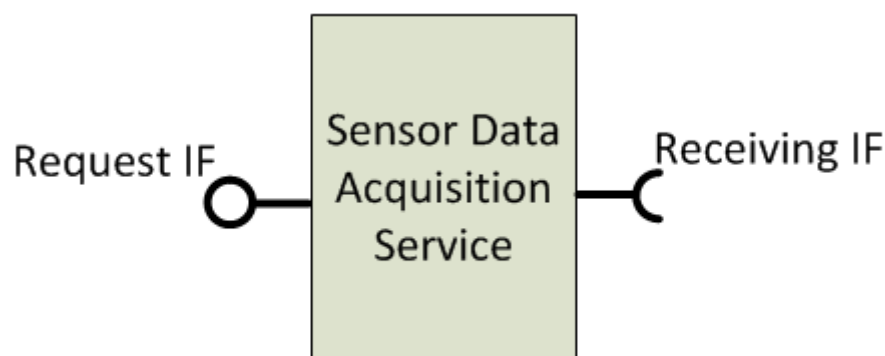


Figure 16 – Sensor Data Acquisition Service Interfaces – Overview

Service Name	<< Sensor-Data-Acquisition-Service >>
Interface Description Standard	None
Communication Protocol supported	TCP
Communication Method	Using dedicated API
Message Format	JSON

Service operations (methods of accessing the service) organized in provided or used (P/U) interfaces:

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYN)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
Sending interface	This interface allows sensor data sources sending messages to sensor data acquisition service.	P	authentication (identity, password)	ASYN	Sends an authenticate message from a device to the sensor data acquisition service.	{String} identity (IN)	The identification of the device.
						{String} password (IN)	The password of the device.
			sendConfiguration (identity, parameters, persons)	ASYN	Sends a configuration message, after the authentication of a device. Configuration messages correspond to the configuration and information of the users for the components.	{String} identity (IN)	The identification of the device.
						{Dictionary[Key, Value]} parameters (IN)	The list of parameters for the device.
						{List[PersonDetails]} persons (IN)	A list of person details, with the respective sensors information.
			sendSensorInfo (identity, values)	ASYN	Sends a message from a device with sensors to the sensor data acquisition service. The message contains a list with the sensor values and the corresponding timestamp.	{String} identity (IN)	The identification of the device.
						{List[SensorValues]} values (IN)	A list of sensor values information containing the sensor identification, value and a timestamp.
			sendDeviceInfo (identity, parameters)	ASYN	Sends a message from a device with his information to the	{String} identity (IN)	The identification of the device.

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYNC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
					sensor data acquisition service. The message contains the details of the device.	{{Dictionary[Key, Value]} parameters (IN)	The list of parameters with details of the device.
			sendPersonValue (identity, parameters)	ASYNC	Sends a message to a device with last sensor values from a user. The message contains a list with the sensor value.	{String} identity (IN)	The identification of the device.
						{{Dictionary[Key, Value]} parameters (IN)	A list of sensor values information containing the sensor identification and the corresponding value.
			sendCommand (identity, parameters)	ASYNC	Sends a message from a device to the sensor data acquisition service.	{String} identity (IN)	The identification of the device.
					The message contains a command to be process, from the sensor data acquisition service; or pass to other device.	{{Dictionary[Key, Value]} parameters (IN)	The list of parameters for the command message.
Receiving interface	This interface allows the handle the delivery messages from the sensor data acquisition service.	U	onSensorInfo (sender, values)	ASYNC	A method that is invoked when the sensor data acquisition service receives a sensor values.	{String} identity (IN)	The identification of the device.
						{List[SensorValues]} values (IN)	A list of sensor values information containing the sensor identification, value

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYNC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
							and a timestamp.
			onCommand (sender, parameters)	ASYNC	A method that is invoked when the sensor data acquisition service receives a command message.	{String} sender (IN)	The identity of the sender that sent the control message
						{Dictionary[Key, Value]} parameters (IN)	The list of parameters for the command message.
			onAuthentication (sender, response)	ASYNC	A method that is invoked when the sensor data acquisition service receives an authentication message.	{String} sender (IN)	The identity of the sender that sent the authentication message
						{String} response (IN)	The response to the sender that sent the authentication message.

Interface Operations Performance Requirements:
None specified!

Service attributes (performance parameters), functional and non-functional parameters characterizing a service:

Service Attribute Name	Service Attribute Description	Quantitative [YES/NO]	Measure (if quantitative =TRUE)	Measurement Method
Availability	The ability to perform over a stated period of time	YES	The uptime percentage	$\frac{t_{up}}{t_{up} + t_{down}}$
Message Throughput	Sensor data acquisition service ability to process the number of sent messages.	YES	Number of messages per second	N

QoS-Parameters: specific Service Attributes comprising to the overall Service QoS – refer to “Service Attributes” –

Service Attribute Name
Availability
Message Throughput

Service Constraints: constraints governing a service – refer to “Service Attributes” –

Service Attribute Name	Metric	Measure	Reference (Description from where this constraint originates)
Availability	99	%	empiric
Message Throughput	500	messages	empiric

Message Format:

At the time of writing of this document the message format is actually under discussion.

Service Interaction Pattern:

To be described if a service is stateful or requires a sequence of interactions: - only communication with external agents aka “service consumers” should be taken under consideration -

The following figures (Fig. 17-18) illustrate the interactions required for the communication from the source component (service consumer) to the Sensor Data Acquisition Service (Figure 17) and the forwarding to the final components (service consumer), respectively (Figure 18).

Note that the illustration uses ordinary UML sequence diagrams for time-ordered representation of interactions.

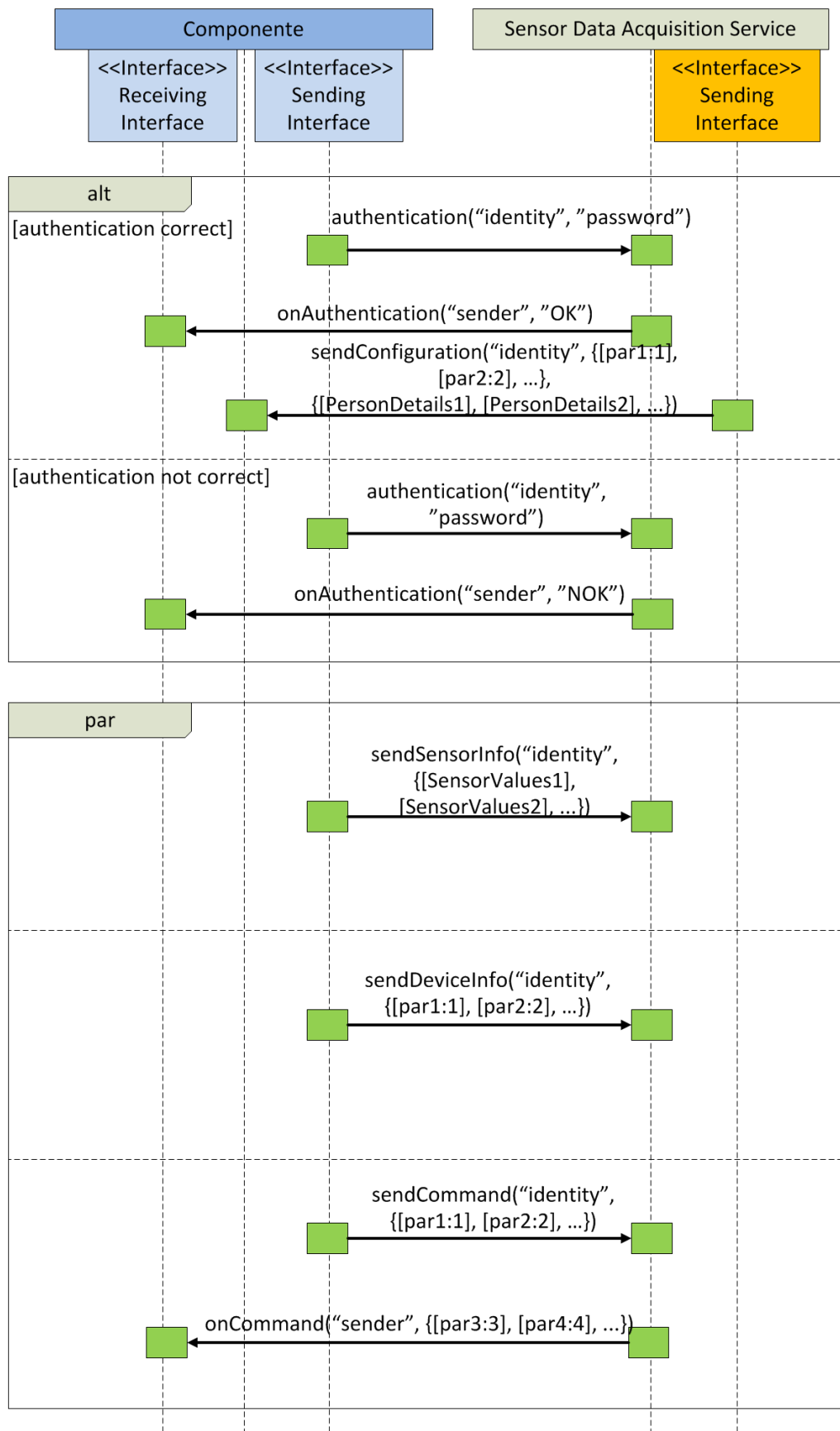


Figure 17 – Required interaction between the source components (service consumer) to the Sensor Data Acquisition Service

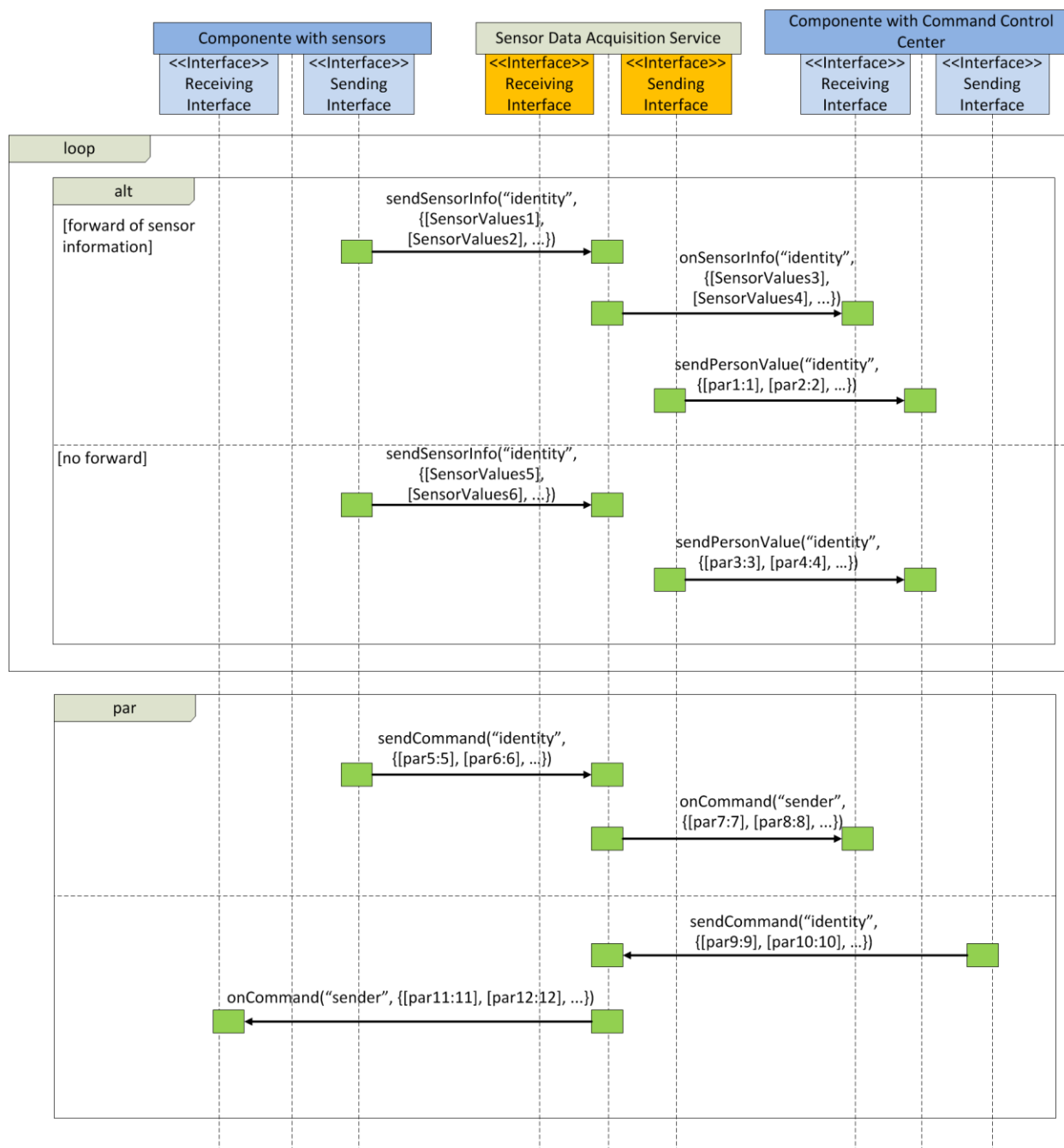


Figure 18 – Required interaction from the source component (service consumer) to the Sensor Data Acquisition Service and the forwarding to the final components (service consumer)

<<Intrusion Detection Service>>

Purpose:

The Intrusion Detection Service (IDS) analyses network traffic - flow data (exported using NetFlow/IPFIX) - of dedicated networks or network segments and reports on network scans (on port 22), brute-force attacks against SSH daemons (on port 22) and compromised SSH daemons.

For service consumers that want to be informed about intrusion detected within dedicated networks the service supports two kinds of interaction patterns. A pro-active notification based message exchange and a pull-based message exchange where the service consumer pulls the service in order to receive intrusion detection information if available.

The following summarizes the interfaces that the service provides or requires in order to interact with its consumers.

Service Summary:

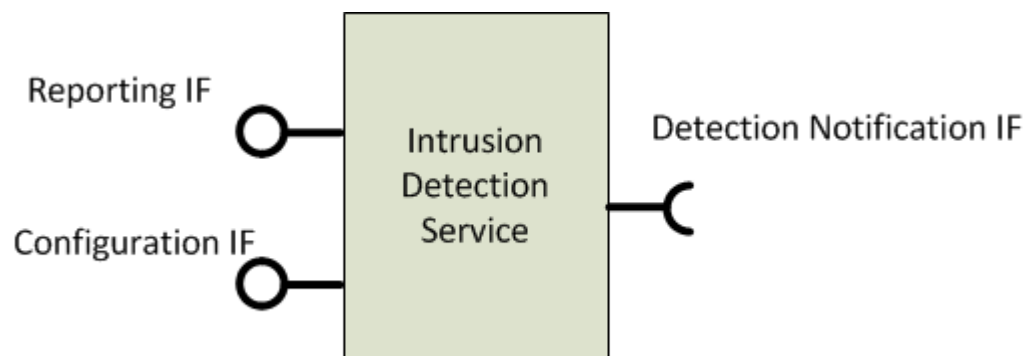


Figure 19 – Intrusion Detection Service Interfaces – Overview

Service Name	<<Intrusion Detection Service>>
Interface Description Standard	None
Communication Protocol supported	HTTP, SMTP
Communication Method	Using dedicated API
Message Format	HTML, plain text

Service operations (methods of accessing the service) organized in provided or used (P/U) interfaces:

Interface Name	IF Description	Provided/Used (P/U)	IF Operation Name	Communication style (SYNC/ASYNC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
Reporting IF	Interface for reporting on detections.	P	getDetections	SYNC	Operation to retrieve all detections within a certain time frame, in a pull mode	{Dates} Time period {IN}	Time period over which detection data should be retrieved
						{Array} Detections {OUT}	List of detections in time period
			registerForDetectionNotifications	ASYNC	Operation to configure parameters of the 'notifyDetection' operation	{Array} Notification settings {IN}	List of key-value pairs
Configuration IF	Interface for configuring the IDS	P	getConfiguration	SYNC	Operation for retrieving one or more setting values	{Array} Configuration keys {IN}	List of keys for which the values are to be retrieved
						{Array} Configuration value {OUT}	List of keys and their values
			setConfiguration	SYNC	Operation for setting one or more setting values	{Array} Configuration keys and values {IN}	Configuration keys and their new values
						{Exception} Message {OUT}	Error message in case of failure
Detection Notification IF	This interface allows clients to handle the delivery of messages.	U	notifyDetection	ASYNC	Operation to notify consumers on detections.	{Array} Detections {OUT}	List of detections in time period

Interface Operations Performance Requirements:
None specified!

Service attributes (performance parameters), functional and non-functional parameters characterizing a service:

Service Attribute Name	Service Attribute Description	Quantitative [YES/NO]	Measure (if quantitative = TRUE)	Measurement Method
Detection rate	Detection rate is a direct indication of the accuracy of IDS. Ultimately, the rate should be 100%.	YES	%	The detection rate is defined as the number of intrusion instances detected by the service (True Positive) divided by the total number of intrusion instances present in the test set
Detection delay	Time between a certain event and the event being reported to the systems/persons in charge	YES	minutes	In case of an intrusion, it is the time between the intrusion taking place in the test set and the service consumer being notified about the intrusion (pro-active notification)
Availability	Indirectly indicates the service's resilience, reliability and maintainability.	YES	The uptime percentage %	$\frac{t_{up}}{t_{up} + t_{down}}$

QoS-Parameters: specific Service Attributes comprising to the overall Service QoS – refer to “Service Attributes” –

Service Attribute Name
Detection rate
Availability

Service Constraints: constraints governing a service – refer to “Service Attributes” –

Service Attribute Name	Metric	Measure	Reference (Description from where this constraint originates)
Detection rate	90	%	empiric
Detection delay	5	minutes	empiric
Availability	95	%	empiric

Service Interaction Pattern:

To be described if a service is stateful or requires a sequence of interactions: - only communication with external agents aka “service consumers” should be taken under consideration -

The following figures (Fig. 20-21) illustrate the interactions required for an IDS consumer in order to read and set the IDS configuration parameters (Figure 20) and in order to get information about possible intrusion detections either in pull mode or in a pro-active notification mode (Figure 21).

Note that the illustration uses ordinary UML sequence diagrams for time-ordered representation of interactions.

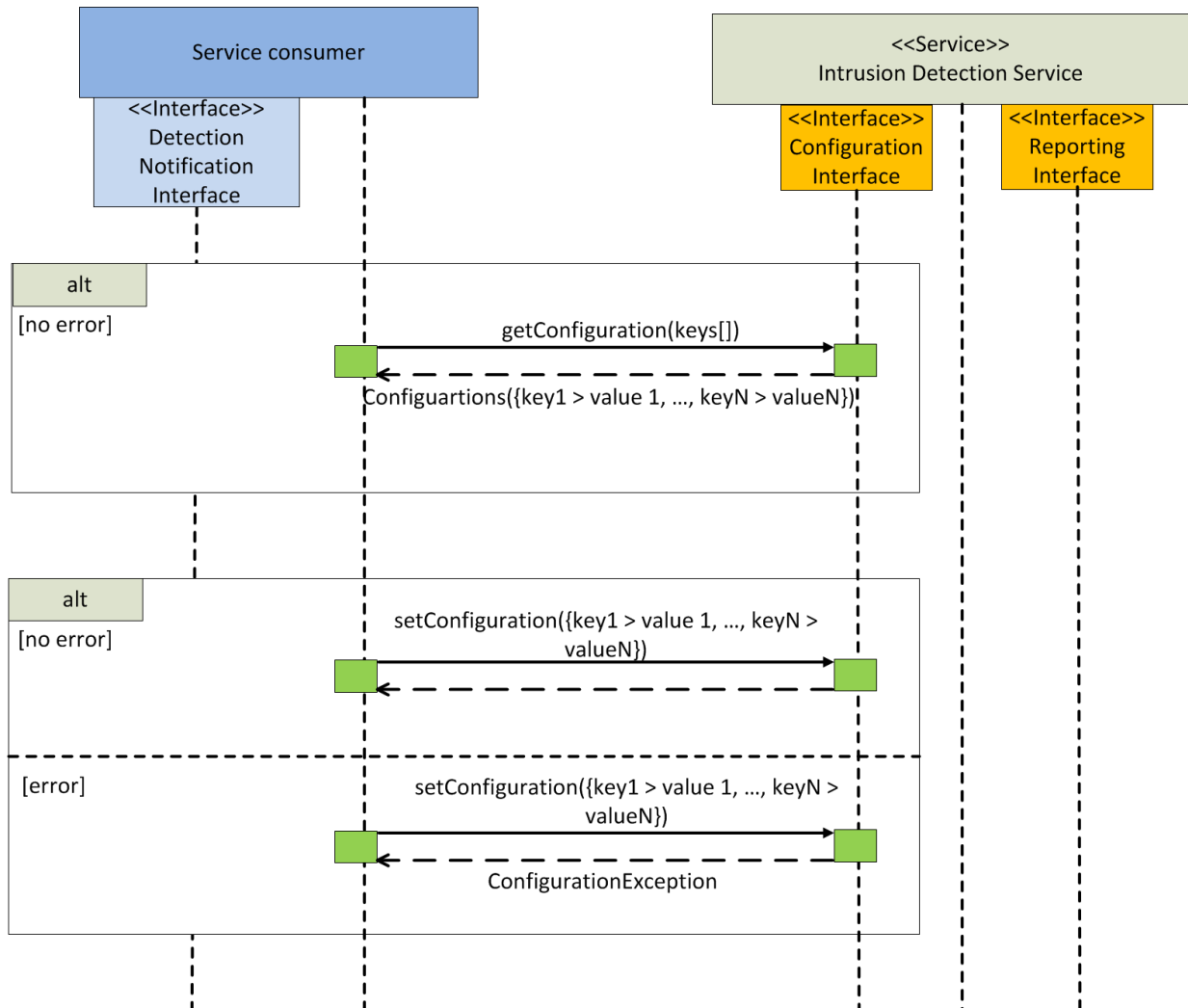


Figure 20 – Interaction required for an IDS consumer in order to read and set the IDS configuration parameters

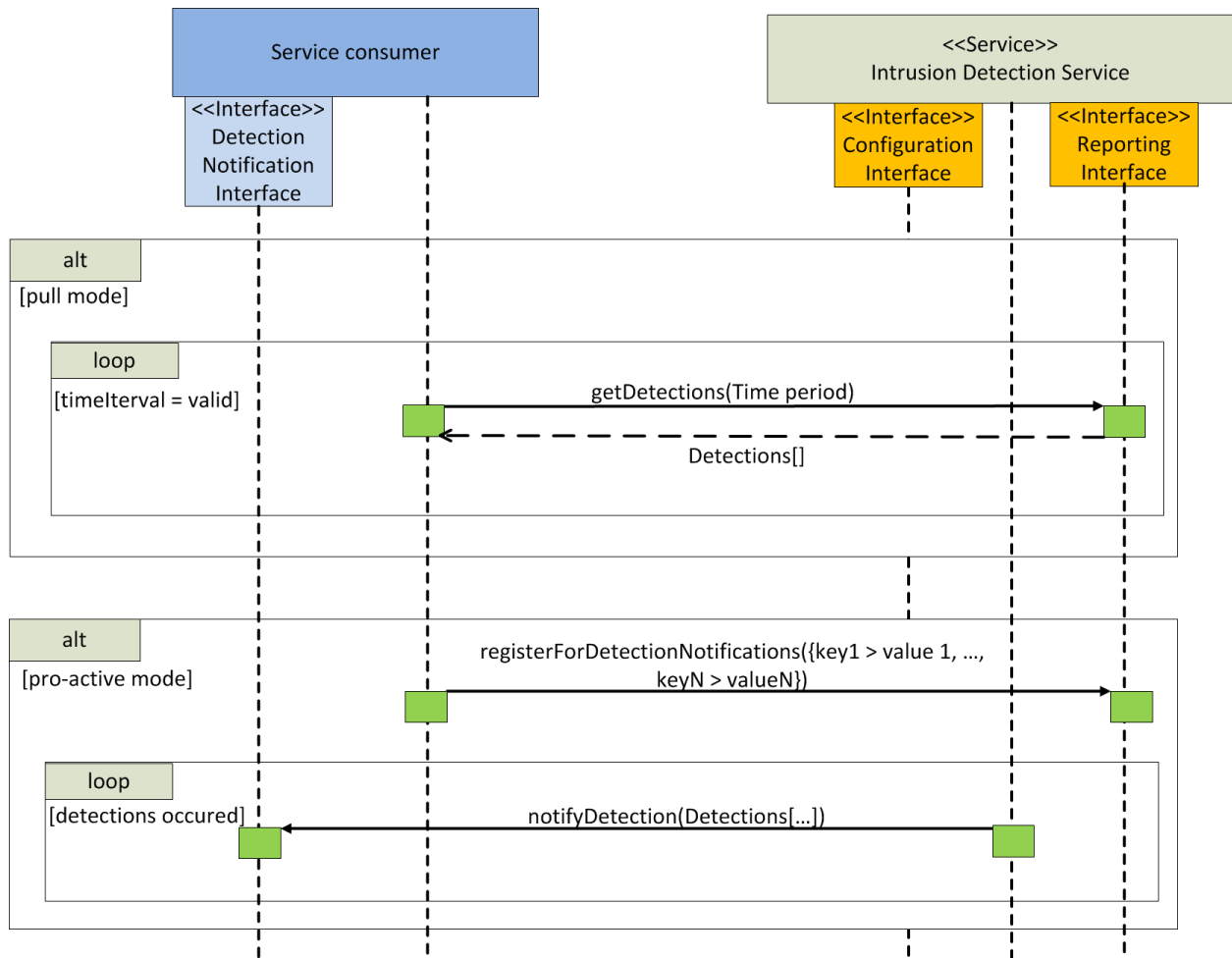


Figure 21 – Interaction required for an IDS consumer in order to get information about possible intrusion detections either in pull mode or in a pro-active notification mode

<<Forensic Service>>

Purpose:

The Forensic Service allows for supervising mobile devices on intrusion attempts, detection, alerting and subsequent management of these devices. As such it provides acquisition, examination and (asynchronous) data exchange capabilities for various actors within the PPDR system.

The service is composed of four components: the acquisition client, the Mobile Device Management (MDM), the aggregation server and the database server. The acquisition client runs on the (LTE) mobile terminals, gathers information and sends them to the forensic service. It also manages the device so it is possible to interact with the device via a dedicated interface in parallel. The core component of the service is the MDM. The MDM is responsible for managing the registered devices and forwarding the messages received from them to the aggregation server. The latter being responsible for processing/analysing the received data from the mobile devices, eventually generate and notify alerts, and finally persist the gathered information using the database server.

The following summarizes the interfaces that the service provides to or requires from its consumers. During runtime, an API-based communication between the service and the service consumers takes place, supported by programming language specific software libraries.

Service Summary:

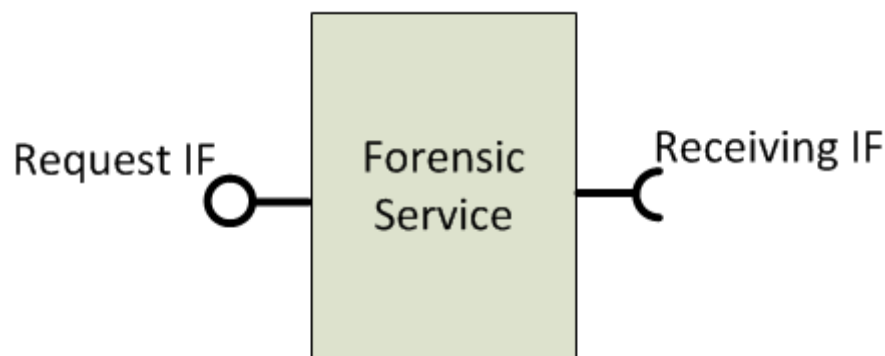


Figure 22 – Forensic Service Interfaces – Overview

Service Name	<< Forensic-Service >>
Interface Description Standard	None
Communication Protocol supported	TCP
Communication Method	Using dedicated API and IDMEF bus
Message Format	XML

Service operations (methods of accessing the service) organized in provided or used (P/U) interfaces:

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYN)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
Request interface	This interface allows to send requests to the forensic data acquisition service	P	Registration (identity, password)	ASYN	Sends a registration request from a device to the forensic service control entity.	{String} identity (IN)	The identification of the device.
						{String} password (IN)	The password of the device.
			Authentication (identity, password)	ASYN	Sends an authenticate message from a device to the forensic service control entity.	{String} identity (IN)	The identification of the device.
						{String} password (IN)	The password of the device.
			vpnAuthentication(identity, password, domain)	ASYN	Sends an authentication token similar to the one in the authentication operation, with the domain field added	{String} identity (IN)	The identification of the device.
						{String} password (IN)	The password of the device.
						{String} domain (IN)	The domain of the VPN server.
			sendAcquiredData(identity, category, size, payload)	ASYN	Send acquired forensic data to the aggregated instance	{String} identity (IN)	The identification of the device.
						{Boolean} Category (IN)	Values concerning the module where the data derives from. 0: Post-mortem Unit 1: Live Unit
						{Float} Size (IN)	A number indicating the data size.

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYNC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
						{ByteArray} payload (IN)	A XML document containing the collected data with the appropriate structure, containing other details, such as associations to databases, timestamps, etc.
Receiving interface	This interface allows the handle the delivery of requests from the forensics service.	U	registrationState(identity, state, code)	ASYNC	Successful or unsuccessful outcome of the registration procedure	{String} identity (IN)	The target device user's username.
						{Boolean} state (IN)	0: Unsuccessful 1: Successful
						{String} code (IN)	Null for successful registration. Otherwise, the error reason is presented (e.g. duplicate credentials)
			authenticationToken(request_id, token)	ASYNC	A token that is returned to the mobile device when it is accepted or rejected from the MDM authorization mechanisms	{String} request_id (IN)	The request identifier from the device.
						{Boolean} token (IN)	A Boolean variable for defining the outcome of the request. Values: 1== accept 0==reject
			sendConfiguration(identity, parameters)	ASYNC	A method that is invoked when a device successfully connects to the MDM.	{String} identity (IN)	The identity of the sender device
						{Dictionary[Key, Value]} parameters (IN)	The list of parameters for the established connection.

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYNC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
			serviceBlock(identity, MAC_address, type, duration)	ASYNC	A command sent when a device has to be blocked.	{String} identity (IN)	The identity of the target device
						{String} MAC_address (IN)	The MAC address of the target device
						{String} type (IN)	Type of block (temporary or permanent)
						{Integer} Duration (IN)	Duration of the block period
			applyCountermeasure(identity, countermeasure_id, group)	ASYNC	A command sent when a specific set of countermeasures has to be applied to the system.	{String} identity (IN)	The identity of the target device
						{String} Countermeasure_id (IN)	Identity of the equivalent countermeasure stored in the system database.
						{Boolean} group (IN)	0: no group 1: applied within a group of devices
			acquisitionState(identity, type, handler)	ASYNC	In which state the acquisition taking place is.	{String} identity (IN)	The identity of the target device
						{String} type (IN)	Live or post-mortem acquisition
						{String} handler (IN)	A value or state signifying if the acquisition procedure is started, finished or paused
			wipeDevice(identity, state)	ASYNC	Command sent from the MDM which leads to the remote	{String} identity (IN)	The identity of the target device

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/ASYN)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
					wipe of the device	{Boolean} state (IN)	0: Non-wiped 1: Wiped

Interface Operations Performance Requirements:
None specified!

Service attributes (performance parameters), functional and non-functional parameters characterizing a service:

Service Attribute Name	Service Attribute Description	Quantitative [YES/NO]	Measure (if quantitative = TRUE)	Measurement Method
Availability	The ability to perform over a stated period of time	YES	The uptime percentage	$\frac{t_{up}}{t_{up} + t_{down}}$
MDM Connection Timeout	The time a device is allowed to remain connected to the MDM	YES	Seconds	Sec.

QoS-Parameters: specific Service Attributes comprising to the overall Service QoS – refer to “Service Attributes” –

Service Attribute Name
Availability
MDM Connection Timeout

Service Constraints: constraints governing a service – refer to “Service Attributes” –

Service Attribute Name	Metric	Measure	Reference (Description from where this constraint originates)
Availability	99	%	empiric
MDM Connection Timeout	3600	seconds	empiric

Service Interaction Pattern:

To be described if a service is stateful or requires a sequence of interactions: - only communication with external agents aka “service consumers” should be taken under consideration -

The following figures (Fig. 23-24) illustrate the interactions required for the communication from the source component(s) – mobile devices - (Figure 23) to the Forensic service and the respective responses (Figure 24).

Note that the illustration uses ordinary UML sequence diagrams for time-ordered representation of interactions.

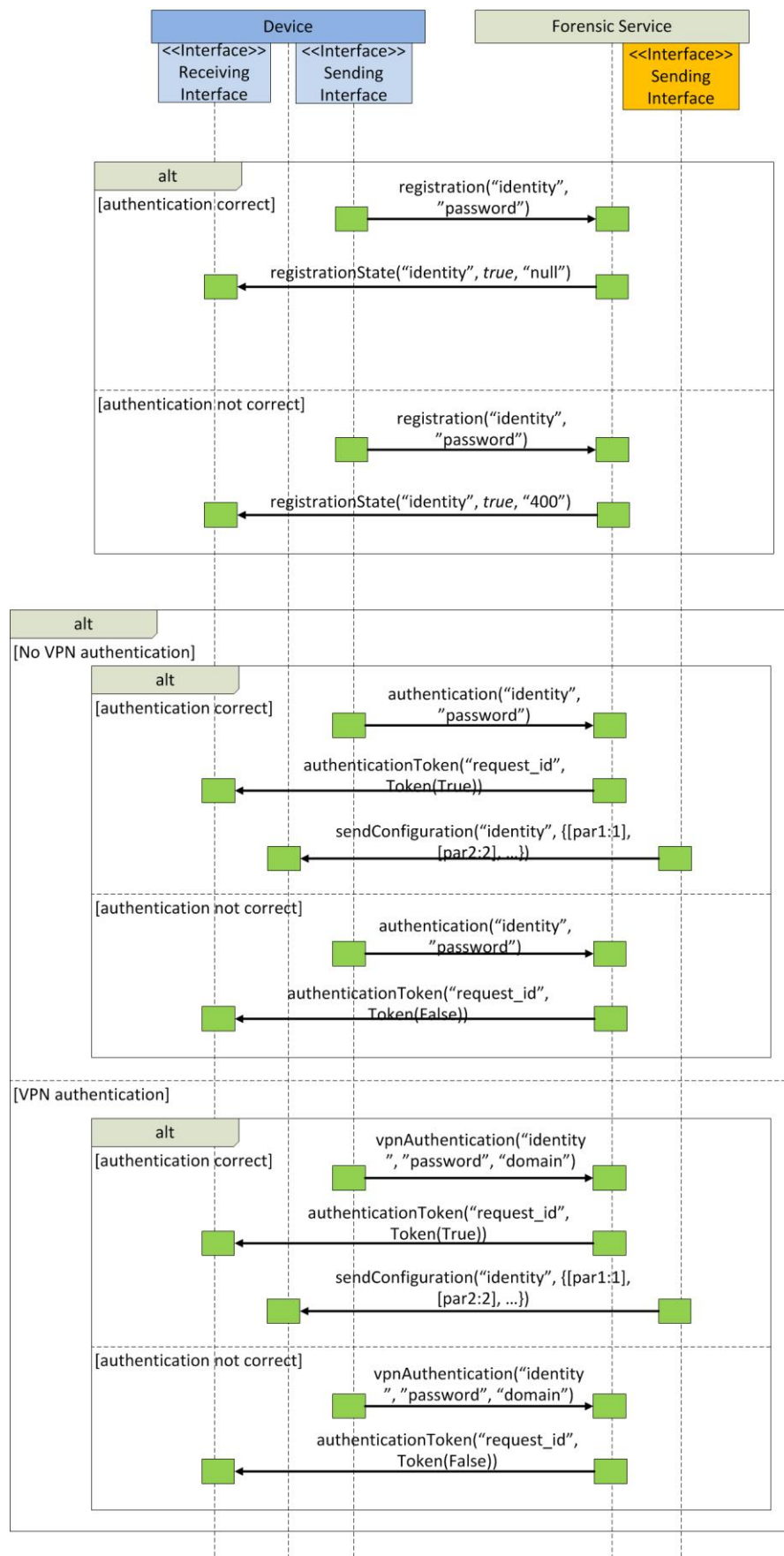


Figure 23 – Required interaction of mobile devices on order to register to the Forensic Service

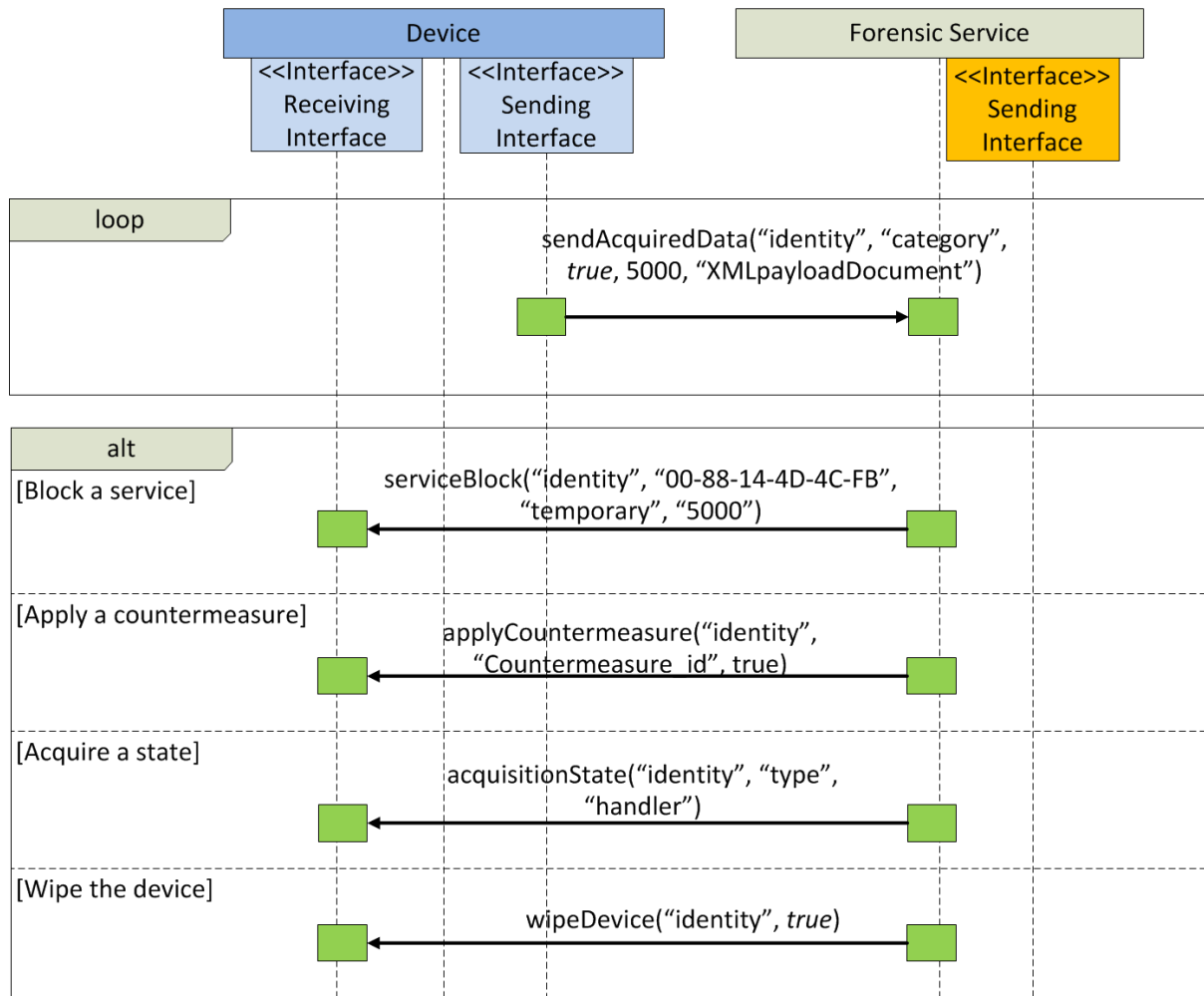


Figure 24 – Required interaction for sending forensic data to the service and possible reactions due to detected intrusion attempts

<<SALUS Dynamic QoS Controller>>

Purpose:

The SALUS Dynamic QoS Controller (DQC) provides an easy way to create and/or modify service data flow policies and charging rules via HTTP/HTTPS protocol. This service will run on the SALUS CCC and is to be administered by a communication's responsible officer. The service allows to create and/or modify service data flow policies and charging rules for each network user (UE) or group of UEs in real-time and for each type of service. The SALUS DQC is the client side of the Alcatel 5780DSC Module located at the LTE EPC. Communication between these entities is done through the RESTful Web Services.

SALUS DQC communicates with the Alcatel 5780DSC Module through QoS flow requests in order to create, modify, query or remove QoS flow sessions. The QoS flow session mode allows the use of a set of tags to set up the required QoS for a data flow. In the context of the guaranteed QoS for Application Establishment over a RESTful API, the requests are called QoS flow requests. A QoS flow request should specify the subscriber, bandwidth and QoS level, server ID, and duration (optional) for the QoS flow API (running in the Alcatel 5780DSC Module).

The following summarizes the interfaces that the service provides or requires in order to interact with its consumers.

Service Summary:

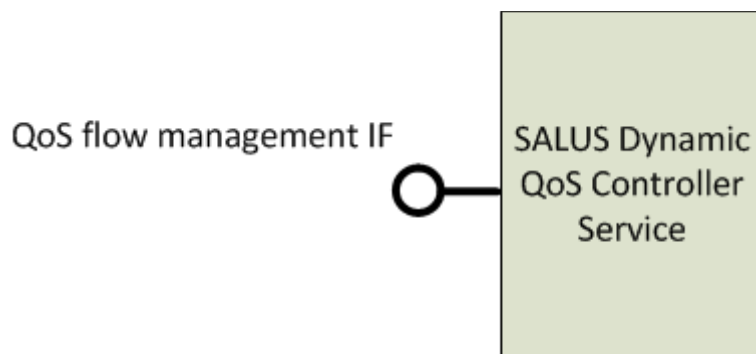


Figure 25 – SALUS Dynamic QoS Controller Service Interfaces – Overview

Service Name	<< SALUS Dynamic QoS Controller >>
Interface Description Standard	RSDL
Communication Protocol supported	HTTP/HTTPS
Communication Method	RESTful
Message Format	XML

Service operations (methods of accessing the service) organized in provided or used (P/U) interfaces:

Interface Name	IF Description	Provided Used (P/U)	IF Operation Name	Communication style (SYNC/AS YNC)	IF Operation Semantics	IF Operation Parameter (IN/OUT)	IF Operation Parameter Semantics
QoS flow management interface	This IF provides an easy way to create and/or modify service data flow policies and charging rules via common internet protocol http/https	U	Create flow session	AS YNC	Provides the QoS parameters needed to create a QoS flow	Subscriber, bandwidth and QoS level, server ID, and duration (OUT)	QoS flow request
			Modify flow session	AS YNC	Provides updated QoS parameters needed to change a QoS flow	Subscriber, bandwidth and QoS level, server ID, and duration (OUT)	QoS flow request
			Query flow session	AS YNC	Queries the QoS parameters in a specific QoS flow	Subscriber, bandwidth and QoS level, server ID, and duration (IN)	QoS flow request
			Delete flow session	AS YNC	Deletes a specific QoS flow	Subscriber, bandwidth and QoS level, server ID, and duration (OUT)	QoS flow request

Interface Operations Performance Requirements:

None specified!

Service attributes (performance parameters), functional and non-functional parameters characterizing a service:

The SALUS DQC service performance parameters are dependent on the service level agreement (SLA) established between the service provider and the PPDR organization. Considering this service is running for mission-critical operations, it is supposed the SLA to be compliant with any legal regulations on this field.

QoS-Parameters: specific Service Attributes comprising to the overall Service QoS – refer to “Service Attributes” –

None specified!

Service Constraints (Service Policy):

The **SALUS DQC** service constraints are dependent on the SLA established between the service provider and the PPDR organization. Considering this service is running for mission-critical operations, it is supposed the SLA to be compliant with any legal regulations on this field.

Service Interaction Pattern:

To be described if a service is stateful or requires a sequence of interactions: - only communication with external agents aka “service consumers” should be taken under consideration -

Figure 26 illustrates the interactions required for managing the flow QoS.

Note that the illustration uses ordinary UML sequence diagrams for time-ordered representation of interactions.

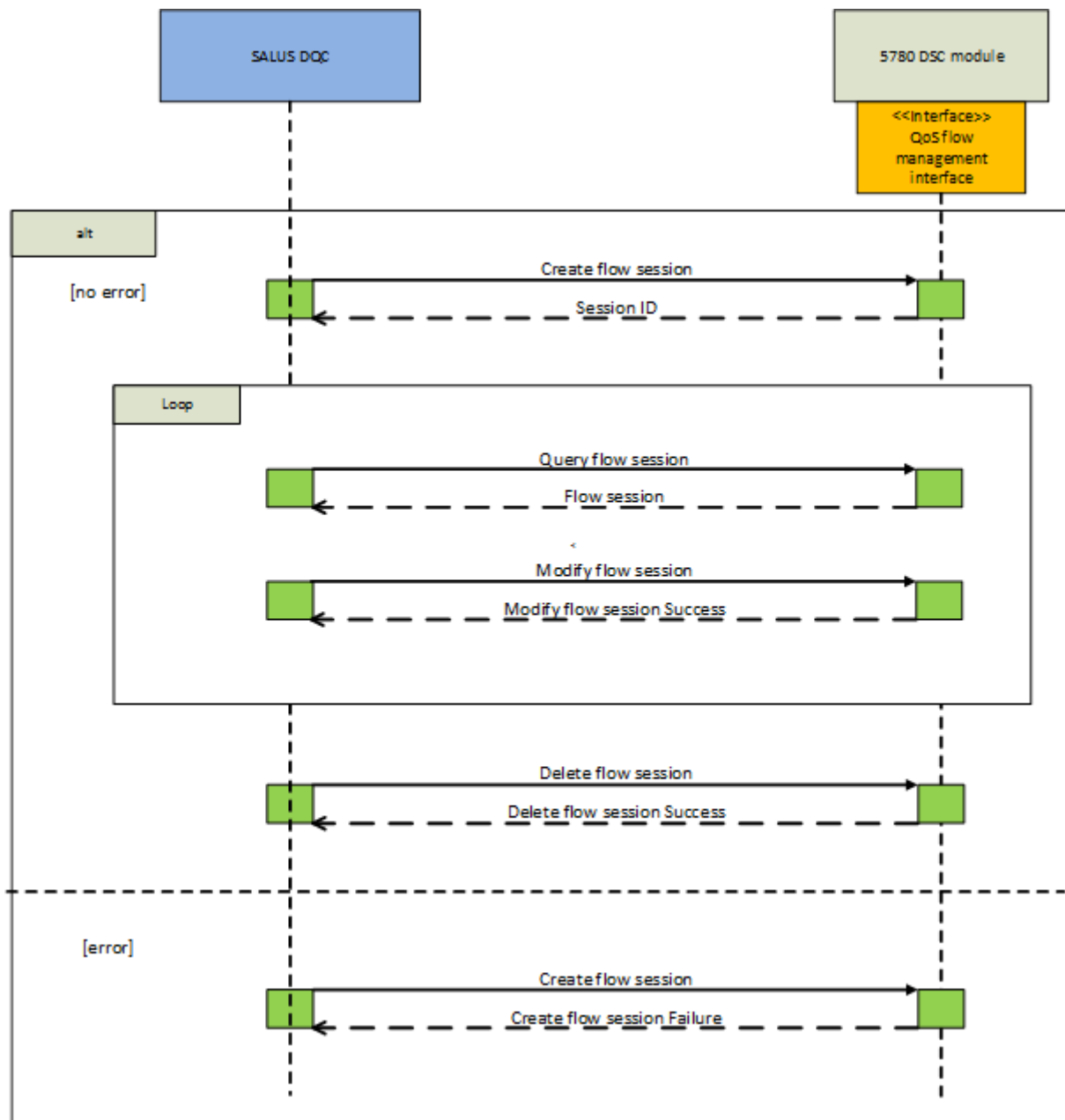


Figure 26 – Required interactions for managing the flow QoS

5 Technical Solution – System Architecture

5.1 SALUS System Architecture – Overview

This section provides a high level view of the end-to-end SALUS System Architecture. It actually gives a simplified physical and logical view of the mapping of the services defined earlier for a multi-organisation / interoperable PPDR system.

The example provided hereafter is describing system architecture for scenarios 1 and 2 (City protection, Temporary protection) which are more mature in terms of solutions that can be put in place. The third scenario (Disaster Recovery) will be covered in the final version of this deliverable.

In this section an example of mapping of services to functional blocks is provided as well as discussion on the physical deployment options of the different functional blocks.

5.1.1 High-level System Architecture

As described in the operational scenarios developed in WP2 [3], PPDR agencies are usually organised in three organisational levels:

- The operational level – PPDR officers in the field conducting routine or specific missions. The operational teams are organised per agency (e.g. police, swat team, customs; fire-brigades, emergency medical services...), per geographical areas and per mission.
- The tactical level – PPDR officers, sitting in command and control rooms coordinating the activities of the operational level and reporting situation to the upper level. The strategic level are usually organised by PPDR agency and geographical areas. In case of multi-agency missions, one of the agencies (usually the police) takes over the role of coordinating the activities of the operational level of selected officers from other agencies and synchronise with the other tactical levels involved.
- The strategic level – a multi-agency top level organisation maintaining a common operational picture of the situation and giving strategic orders to the tactical level.

In the high level architecture (see Figure 27), three PPDR agencies with a single tactical level per agency is considered (Police, Fire-brigade, EMS). Communication flows between these different stakeholders are represented by brown dotted lines. However, the organisational architecture can include more agencies / tactical levels.

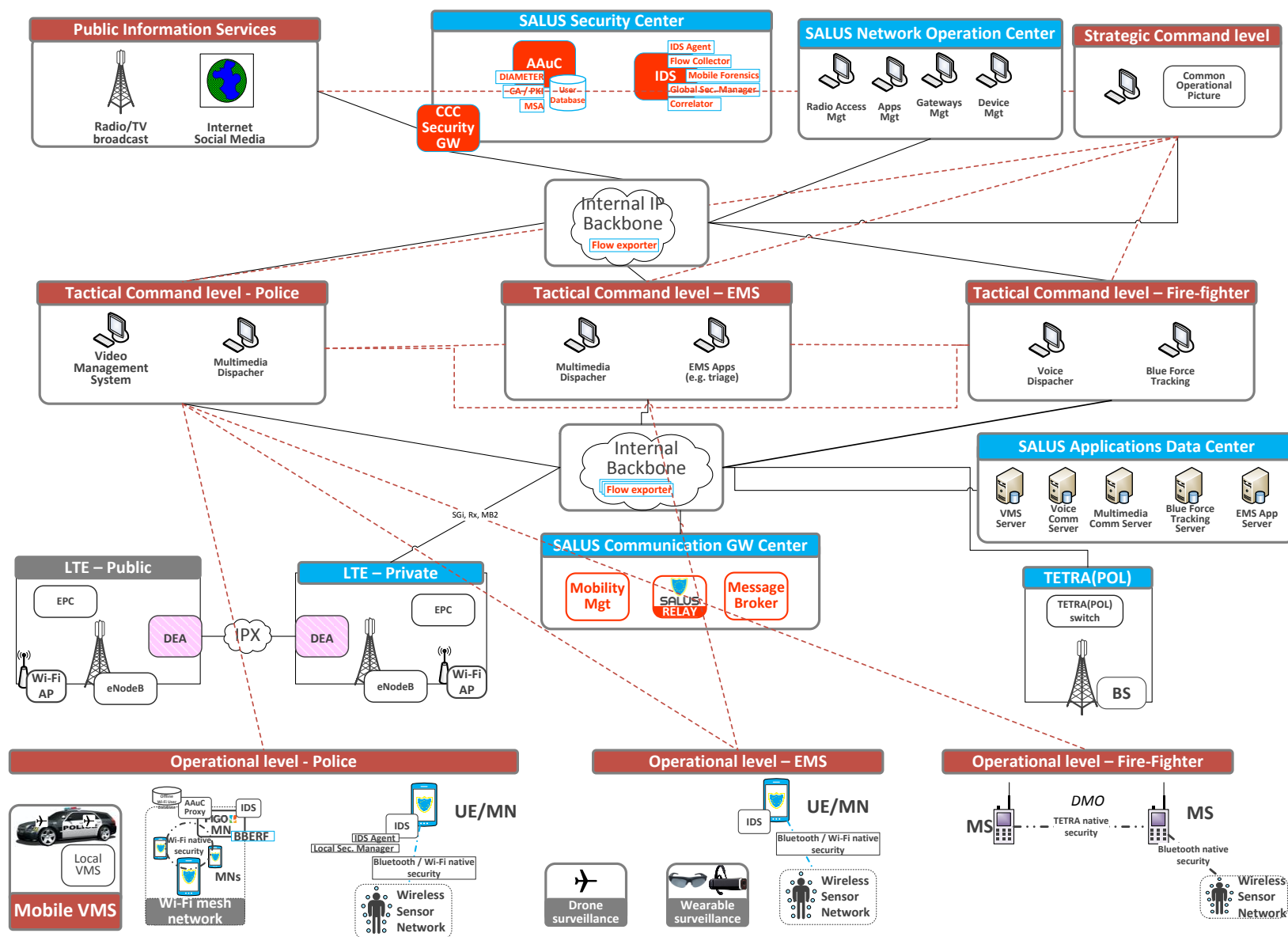


Figure 27 – High level SALUS System Architecture

The different levels and organisations are exchanging information through secure communication systems. Between the strategic level and each PPDR organisation's tactical level and between tactical levels of different PPDR agencies, the communication system is a fixed and secure network capable of transporting any type of communication / information (voice, data, video...). This network is usually based on IP transport technology. However, since not all information have the same characteristics in terms of bandwidth, bandwidth variations, delay, jitter, priority, availability, resiliency..., it is recommended to use IP/Multiprotocol Label Switching (MPLS). With such approach, it is possible to have differentiated and guaranteed delivery of information flows. Besides, in case of congestions and/or any failure in the network, pre-defined rules (traffic engineering) will quickly restore an alternative path and will ensure that in any case, the flows with higher priority will get through the system without any impacts. In addition IP/MPLS can allow creating different Virtual Private Networks at layer-2 and/or layer3 to interconnect the different sites / organisations and to segregate different traffic flows if needed. This fixed communication infrastructure needs to be secured using e.g. IPSec associated with the appropriate certificates for encrypting the information and using intrusion detection systems (IDS) for detecting and solving potential external cyber attacks to the network (see WP5 deliverables for more details [9]). It is to be noted (although if it is not described in the WP5 deliverables) that additional layer 1 encryption can also be performed in case microwave and/or optic are used as a transmission medium to increase protection against eavesdropping.

Between the operational and tactical levels, the communications are usually exchanged using a wireless system. This wireless system can be:

- A specialized legacy narrowband digital system such as TETRA or TETRAPOL – in that case, the information between the tactical and operational levels are mainly limited to voice services; text messaging (status messages and/or SMS like) and low data rate exchange (e.g. location data, biometric, ...) of few kbps maximum.
- A dedicated broadband LTE system (LTE private) – this allows for exchanging broadband information (video, data, pictures, voice...) over dedicated infrastructure and spectrum (using unicast and broadcast transport bearers),
- A public broadband LTE system - this allows for exchanging broadband information (video, data, pictures, voice...) over one (or multiple) public commercial network(s). In order to get some level of control on such networks, a full-MVNO approach needs to be considered (see WP4 deliverable D4.3 for more details [10]).
- A Wi-Fi infrastructure (trusted or untrusted) owned or not by the PPDR operator.

When the broadband access connection is based on a commercial LTE network, it is advised to deploy a Diameter Edge Agent (DEA) to interconnect the commercial LTE network. This functions simplifies the connection to multiple commercial network (if this is the case) but more importantly provides required security level for the diameter messages exchanged between the PPDR infrastructure and the commercial networks (LTE interfaces S6 and S9 between the HSS and PCRF of both networks).

Since operational level PPDR users of different organisations may use different wireless access networks and may need to communicate between themselves, communication gateways need to be deployed for two purposes:

- To enable interworking between communications (at least voice and messaging) between users on different radio access technologies (LTE or Wi-Fi, TETRA and TETRAPOL),
- To enable seamless roaming between broadband wireless systems.

In case the wireless infrastructure is not available or when coverage is not available at a given location, operational level officers may still need to communicate locally. In TETRA or TETRAPOL this is done by using Direct Mode Operation (DMO). For broadband, the solution

can rely on a Wi-Fi based mesh infrastructure (see WP6 deliverable [11]). In a latter phase, device to device communications over LTE will be available after 3GPP standard activities on Release 13 are finalized (planned mid 2016 for specification completion).

In the high level architecture depicted here after, it is assumed that:

- Both the police and emergency medical services have access to a broadband wireless access network through a dedicated LTE system, or a commercial LTE system or a Wi-Fi system,
- Fire-fighters have only access to a narrowband system (TETRA or TETRAPOL).

Each officer (and each vehicle) is equipped with a terminal able to communicate with the relevant access system. The following table (Table 6) gives the interfaces capabilities of the terminals considered in SALUS connecting to the infrastructure side (WAN/LAN interface –Wide Area Network / Local Area Network-) and to the PAN (Personal Area Network). In the Intermediate version of the report, dual mode terminals (e.g. TETRA + LTE) are not considered although we may expect that such devices will exist in the future.

Table 6 – Terminal types and network interfaces

Terminal	WAN / LAN interface	PAN interface	Comment
Narrowband	TETRA	Bluetooth and PEI	DMO available
Narrowband	TETRAPOL	Bluetooth	DMO available
Broadband	LTE / Wi-Fi	Wi-Fi / Bluetooth	LTE terminals can also support 2G/3G No DMO now

In SALUS, security concepts are also extended to terminals connected over broadband wireless systems. In particular, it is proposed to run IDS / Forensic systems on the broadband terminals to detect potential malware / viruses that would be present in the terminal [9].

Tactical levels are exchanging information with operational level using specific applications. The ones included in the exemplary architecture are:

- Video management system (VMS) – a system that is used to record / monitor different video feeds from video sensors (CCTV cameras –fixed, in-vehicles...-, drones, smartphones...),
- Multimedia dispatchers systems – a system to exchange multimedia information (voice, video, pictures, messages, location...) with officers using group communications or individual communications with advanced PPDR features such as late entry, ambience listening, dynamic group assignment, and emergency/distress calls.(see deliverable D6.2 for more details [12]).
- Voice dispatcher system – a system to coordinate officer through voice and textual information exchange in group communications or individual communications with advanced features such as late entry, ambience listening, dynamic group assignment, and emergency/distress calls (see deliverable D6.2 for more details [12]).
- Blue Force tracking system – a system to have in real time access to the position and status of each officer and vehicle in the field. The application can also share the same information to authorised users in the field or in other command and control rooms.
- Specific applications – for instance, EMS may use triage application to prioritise assistance to injured people during mass casualty events.

These applications are in most cases IP based (except for voice dispatcher for TETRA / TETRAPOL that use circuit switched protocols over TDM or eventually IP) and usually rely on RTP/RTCP/SIP/HTTP/FTP and UDP or TCP. Also, most of these applications are client server based. There is usually a server (or multiple in different locations servers for redundancy purposes) located in a secured environment (e.g. a private data center) and the applications running in the tactical command and control rooms are specific clients of this server.

It is to be noted that all tactical level applications can be located in fixed command and control rooms or can be installed in mobile command and control vehicle. In that case, the connection to the other components is performed through a wireless network (preferably a broadband system but can also be done using TETRA/TETRAPOL of limited to voice and low data rate applications).

In SALUS, there is an emphasis on exchanging information data using a common format for different types of (sensor) data (location, man-down, biometrics...). For that purpose a message broker has been specified and developed [9].

In addition, management platforms are necessary to configure, supervise and troubleshoot all the elements of the solutions. Besides, it is a common practice in complex communication systems to deploy an umbrella management system to have a consistent view of the network status and be able to quickly assess the origin of the source of a failure through alarms/events correlation and put in place workflows for solving problems in an efficient way. Also, recently, there has been more and more emphasis on device management; For PPDR organisations, this is indeed of utmost importance since when moving to LTE, each device will have access to a large number of applications and may have large amount of massive information in its memory. In order to have a consistent and controlled fleet of terminals and applications, tools are required to deploy / verify validated versions of operating systems and applications, to configure over the air the terminals (e.g. remotely enabling / locking some peripherals such as Wi-Fi or Bluetooth interface, a camera or GPS) or to lock and eventually wipe a terminal that has been reported lost or stolen.

5.1.2 Mapping of Services to Functional Blocks

In this section, a mapping of the services described in section 4 to the high level architecture components is proposed. It is to be noted that SALUS services category in this section are identified with their acronym:

- COI – Community of Interest Service,
- COISS – COI Specific Service
- CCOIS – Common COI Service
- NIIS – Network and Information Infrastructure Service
- IIS - Information and Integration Service
- NTS – Network/Transport service

As already mentioned, some of the services are transversal. This is in particular the case of the “Information Assurance Service”. In the SALUS architecture it is for example proposed that all elements use a common Authentication system [9]. Hence all equipment would include the Resource Authentication Service. Similarly, all IP equipment will include Intrusion Detection Service client and Forensic Service clients.

Table 7 gives the preliminary mapping of the SALUS services for the terminals/devices. Indeed, it is possible to only include a subset of the services in every single terminal according to the capabilities required. The terminals considered are terminals able to communicate to wireless

radio access system (such as a TETRA terminal or a LTE terminal) but also integrated devices (such as a drone) and wearable devices.

Table 7 – Mapping of SALUS services on terminal / devices

Physical node	SALUS Service category	SALUS Service	Type	Comment
TETRA(POL) Terminal	NNIS/IIS	Voice Communication Service	Client	Standard voice TETRA(POL) services
		Text Messaging Services	Client	Standard TETRA(POL) text service
		Multimedia Messaging services	Client	Standard TETRA(POL) data service (limited to few kbps)
	CCOIS	Sensor Data Acquisition Service	Client	e.g. for location (GPS coordinates)
LTE/Wi-Fi terminal	NNIS/IIS	Voice Communication Service	Client	CVDP protocol
		Text Messaging Services	Client	CVDP protocol
		Multimedia Messaging services	Client	CVDP protocol
		Video Conferencing Service	Client	
		Chat Service	Client	
		Streaming service	Client	To send / receive video streams
		Message Brokering Service	Client	To exchange data in predefined format
	NIIS/NTS	WiFi2LTEMobility Service	Client	
	COISS	Indoor location service	Client	
		Status Monitoring Service	Client	
		Number plate recognition	Client	In case it is done locally
	CCOIS	Sensor Data Acquisition Service	Client	Integrated or sensors connected in a PAN
		Force Tracking Service	Client	To see location of other users
		Situation Picture Service	Client	To visualise / capture images
Drone	NIIS/IIS	Data communication Service	Client	To exchange flight control information with ground station
		Streaming Service		To stream video to ground station
	NNIS/NTS	WiFi2LTEMobility Service	Client	In case of drone with multiple radio modems
	CCOIS	Situation Picture Service	Client	To capture / send image to ground station
Wearable device	CCOIS	Sensor Data Acquisition Service	Client	Location, vital sign, position
		Situation Picture Service	Client	To send picture
	NIIS/IIS	Streaming Service	Client	To stream video from a wearable camera or to

				display a video on a wearable device (e.g. smart watches / smart glasses)
		Text Messaging Service	Client	To read / send a text on a wearable device (e.g. smart watches / glasses)
		Multimedia Messaging Service	Client	To read / send a multimedia message on a wearable device (e.g. smart watches / glasses)

Table 8 gives the preliminary mapping of the SALUS services for the SALUS Communication GWs. These gateways include the Mobility management GW (to ensure mobility between LTE and Wi-Fi), the SALUS Relay GW (for communications interworking between broadband and narrowband systems) and the Message Broker.

Table 8 – Mapping of SALUS services on SALUS Gateways

Physical Node	SALUS Service Category	SALUS Service	Type	Comment
Mobility Management GW	NIIS/NTS	WiFi2LTE Mobility Services	Server	
SALUS Relay GW	NIIS/NTS	TETRA2TETRAPOL IW Service	Server	Voice (including advanced features) and messaging interworking
		TETRA2LTE IW Service	Server	Voice (including advanced features) and messaging interworking
		LTE2TETRA IW Service (new)	Server	Multimedia communications interworking (e.g. granting a voice only access to a TETRA user connecting to a visio-conference running on LTE)
		TETRAPOL2LTE Services	Server	Voice (including advanced features) and messaging interworking
		LTE2TETRAPOL services (new)	Server	Multimedia communications interworking (e.g. granting a voice only access to a TETRAPOL user connecting to a visio-conference running on LTE)
	CCOIS	Image dissemination service	Server	Reformatting image to fit radio access / device capabilities
		Situation Picture Service	Server	Reformatting situation information to fit radio access / device capabilities
Message Broker GW	NIIS/IIS	Message Brokering service	Server	To collect data in a pre-defined format from sensors and present it to application servers through an API

Table 9 gives the preliminary mapping of the SALUS services for the SALUS Tactical applications. It is to be noted that the list of applications is not exhaustive but may represent the minimum set of applications to deploy in future PPDR systems.

Table 9 – Mapping of SALUS Service onto Tactical Level applications

Physical Node	SALUS Service Category	SALUS Service	Type	Comment
Video Management System	COISS	Number Plate Recognition Service	Client	Automatic identification of a car from its licensed plates through a CCTV system
	CCOIS	Video Dispatching service	Client	Interface to dispatch specific video streams to predefined groups of users
	NIIS/IIS	Streaming Service	Client	Streaming video (and optionally audio) from CCTV cameras or other cameras in the system
	NIIS/NTS	Network QoS Monitoring Service	Client	Controlling/Modifying QoS parameters on selected streams
Multimedia Dispatcher System	COISS	Indoor Localisation service	Client	Optional add-on
		Status Monitoring Services	Client	Optional add-on
	CCOIS	Sensor Data acquisition Service		Optional add-on
		Force Tracking Service	Client	Optional add-on
		Image dissemination service	Client	To send/share images to (groups of) users
		Video Dispatching Service	Client	Optional add-on
		Situation Picture Service	Client	To send/share situations to (groups of) users
	NIIS/IIS	Voice Communication Service	Client	CVDP protocol (includes ambiance/discrete listening)
		Streaming Service	Client	(may include ambiance/discrete streaming from specific users)
		Text Messaging Service	Client	CVDP Protocol
		Multimedia Messaging Service	Client	
		Video conferencing service	Client	
		Chat Service	Client	
	NIIS/NTS	Network QoS Monitoring Service		Controlling/Modifying QoS parameters of selected users/streams
EMS application (triage)	COISS	Status Monitoring Service	Client	
	CCOIS	Sensor Data Acquisition Service	Client	

Voice Dispatcher system	COISS	Indoor Localisation Service	Client	Optional Add-on
		Status Monitoring Service	Client	Optional Add-on
	CCOIS	Sensor Data Acquisition Service	Client	Optional Add-on
		Force Tracking Service	Client	Optional Add-on
	NIIS/IIS	Voice Communication Service	Client	CVDP Protocol
		Text Messaging Services		
	NIIS/NTS	Network QoS Monitoring Service	Client	Controlling/Modifying QoS parameters of selected users/streams
Blue Force Tracking System	COISS	Indoor Localisation Service	Client	
		Status Monitoring Service	Client	
	CCOIS	Sensor Data Acquisition Service	Client	
		Force Tracking Service	Client	

Table 10 gives the preliminary mapping of the SALUS services on data center application servers. It is to be noted that the list of applications is not exhaustive but may represent the minimum set of applications to deploy in future PPDR systems.

Table 10 – Mapping of SALUS Services in Data Center applications servers

Physical Node	SALUS Service Category	SALUS Service	Type	Comment
Video Management System	COISS	Number Plate Recognition Service	Server	Automatic identification of a car from its licensed plates through a CCTV system
	CCOIS	Video Dispatching service	Client	To dispatch specific video streams to predefined groups of users
	NIIS/IIS	Streaming Service	Client	Streaming video (and optionally audio) from CCTV cameras or other cameras in the system and to VMS clients in Command and control centers and/or mobile terminals. Also includes recording of all video stream
	NIIS/NTS	Network QoS Monitoring Service	Server	Interface to LTE PCRF (Rx)
Multimedia Dispatcher System	COISS	Indoor Localisation service	Server	Optional add-on
		Status Monitoring Services	Server	Optional add-on

	CCOIS	Sensor Data acquisition Service	Server	Optional add-on
		Force Tracking Service	Server	Optional add-on
		Image dissemination service	Server	To send/share images to (groups of) users
		Video Dispatching Service	Server	Optional add-on
		Situation Picture Service	Server	To send/share situations to (groups of) users
	NIIS/IIS	Voice Communication Service	Server	CVDP protocol (includes recording)
		Streaming Service	Server	Includes recording
		Text Messaging Service	Server	CVDP Protocol (includes recording)
		Multimedia Messaging Service	Server	(includes recording)
		Video conferencing service	Server	(includes recording)
		Chat Service	Server	(includes recording)
		Message Brokering service	Client	
	NIIS/NTS	Network QoS Monitoring Service	Server	Interface to LTE PCRF (Rx)
EMS application (triage)	COISS	Status Monitoring Service	Server	
	CCOIS	Sensor Data Acquisition Service	Server	
	NIIS/NTS	Message Brokering service	Client	
Voice Dispatcher system	COISS	Indoor Localisation Service	Server	Optional Add-on
		Status Monitoring Service	Server	Optional Add-on
	CCOIS	Sensor Data Acquisition Service	Server	Optional Add-on
		Force Tracking Service	Server	Optional Add-on
	NIIS/IIS	Voice Communication Service	Server	CVDP Protocol
		Text Messaging Services	Server	
	NIIS/NTS	Network QoS Monitoring Service	Server	Interface to LTE PCRF (Rx)
		Message Brokering service	Server	
Blue Force Tracking System	COISS	Indoor Localisation Service	Server	
		Status Monitoring Service	Server	
	CCOIS	Sensor Data Acquisition Service	Server	
		Force Tracking Service	Server	
	NIIS/NTS	Message Brokering service	Server	

5.1.3 Physical Deployment Options

The actual physical location of the different servers and applications should be flexible. Indeed, in order to provide maximum interoperability and interworking between different agencies, a common infrastructure with common security, gateways and applications should be recommended. However, the governance between the different agencies may imply that different instance of the same function is deployed in each agency. The functions developed in SALUS should offer this flexibility.

Besides, to improve availability and resiliency of the system, the key functional blocks of the solutions (SALUS security center, SALUS Communication gateways center, SALUS network operation center and SALUS applications data centers) shall be built in a redundant way and shall also support geo-redundancy to be resilient to major site disasters.

5.2 SALUS System Architecture – Service Provisions

This chapter provides a detailed specification of systems or system-of-systems required to provide the services specified in chapter 4.3.3. This is still an ongoing process. Not for all services, for the time of writing of this deliverable, system architectures have been derived. Only a subset is designed yet. System specifications will be provided throughout this chapter for the following services specified:

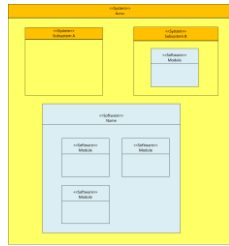
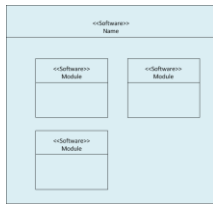
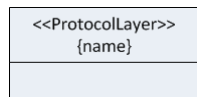
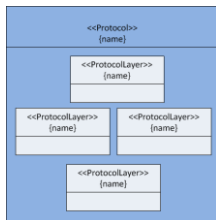
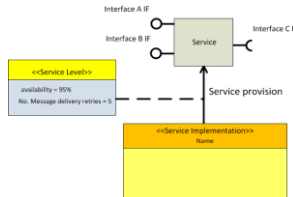
- Message Brokering Service,
- Sensor Data Acquisition Service,
- Force Tracking Service,
- Intrusion Detection Service.

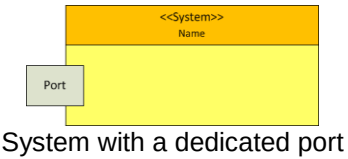
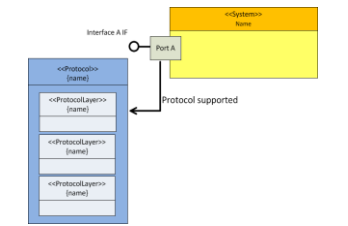
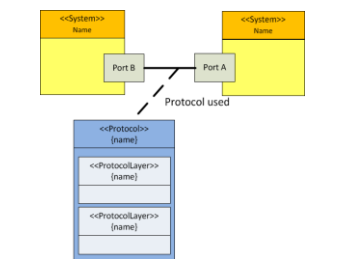
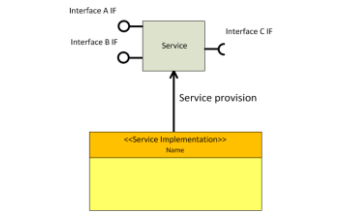
The specification follows a uniform fashion and provides an aggregated view on the SA model (using the modelling environment and its formal approach) under development in a service by service manner. The description covers the following aspects of the service provision architecture, illustrated in a dedicated diagram:

- Systems to Service Mapping,
- Interfaces and Interface interactions,
- Technical standards and protocols applied,
- System composition,
- Software composition,
- System connections.

The following elements of description with their dedicated semantics will be used throughout the diagrams describing the SA for the corresponding service (see Table 11 and subchapter 3.2.2)

Table 11 – Annotations and their corresponding semantics used throughout system descriptions

Annotation	Semantics	Diagram Element
<<System>>	Man-made physical asset (e.g. computer, hardware etc.), reusable across architecture(s). A system may be composed of sub-systems and can host software components	 System composed of Systems and hosting Software
<<Software>>	Executable computer code/programme, reusable across architecture(s). A software may itself composed of several software components	 Software composed of Software modules
<<Physical Architecture>>	Forms reusable architecture and depicts structure and interactions - may be used as configurations within other physical architectures -, it defines a self-contained unit within a defined environment. Composed of Systems and Software	same as System
<<Protocol Layer>>	A standard for communication.	 A protocol layer with a certain name
<<Protocol>>	A protocol is composed of different protocol layers. Within a protocol at the same level different layers may be supported/used interchangeable.	 A protocol with certain name, build up from certain layers; providing interchangeable layers
<<Service level>>	A value specification for a set of service attributes indicating the level to which a service implementation delivers a Service, in a particular environment. Example: Service attribute "availability" may be defined against a Service.	 Implementation realizing a Service to a certain QoS level

	A given Resource could have a corresponding Service level - e.g. "90%"	
<<Service Implementation>>	Specific set of physical assets that together deliver the service. Specifies typical configurations (templates) but does not describe actual deployed resources	Same as System or Physical Architecture
Port	An interface (logical or physical) provided by a System. Ports functionality may be implemented either by an artefact (man-made object, car, fuel etc.) or software. May be used throughout Systems, Software, Service Implementations or Physical Architectures	 System with a dedicated port
Protocol supported	Via certain ports dedicated interfaces (functionalities) may be provided/required (see Port) and can be accessed using/supporting a dedicated protocol.	 System with a dedicated port providing a certain interface supporting a certain protocol
Protocol used	Indicating, that several ports are connected and communicate via a certain protocol.	 Two Ports (and their Systems) connected using a certain protocol
Service provision	An assertion that a Resource (Service Implementation) delivers a Service to a specified Service level (see Service level)	 Implementation realizing a Service

5.2.1 Service Provision: <<Sensor Data Acquisition Service>>

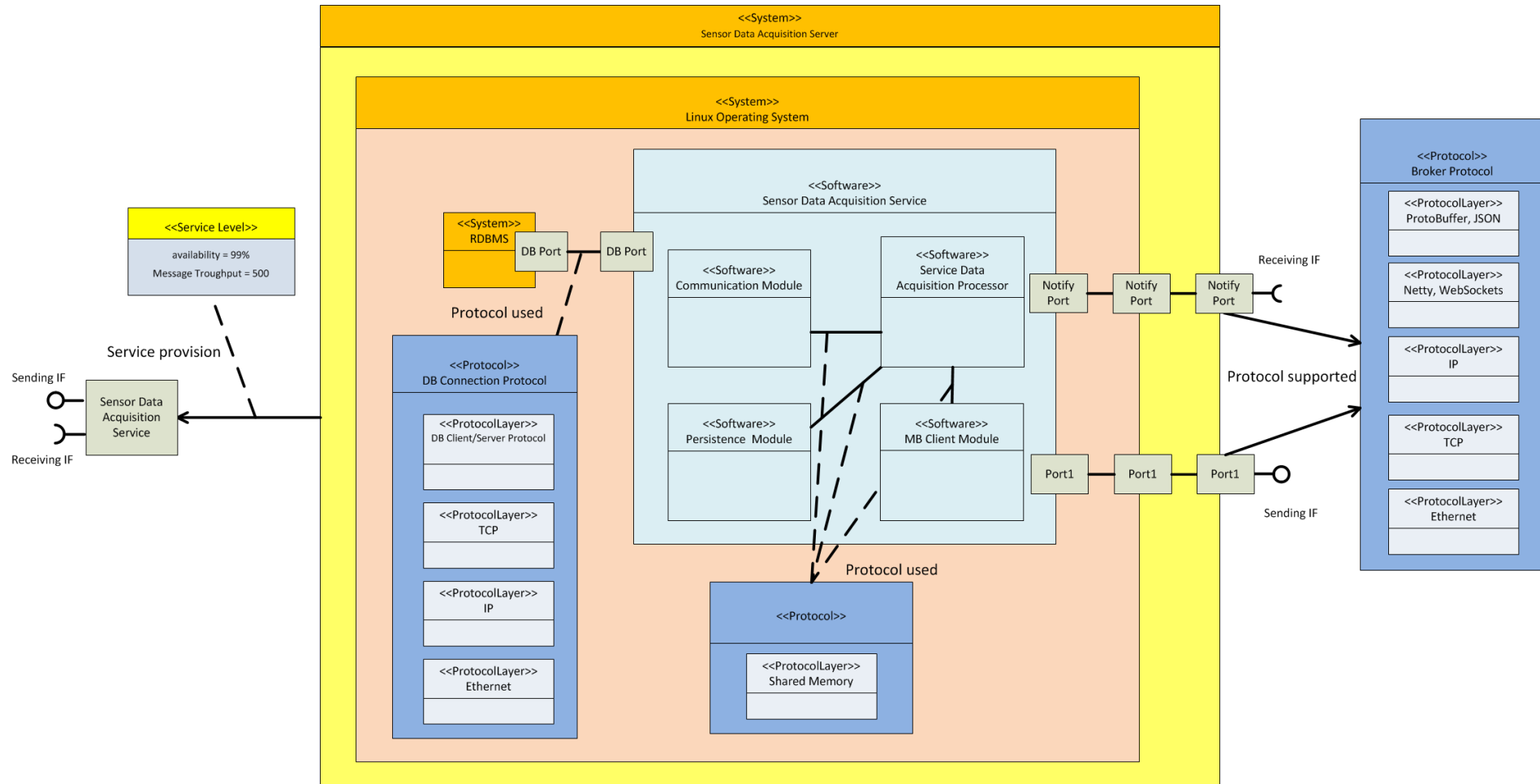


Figure 28 – Sensor Data Acquisition Service architecture

Sensor Data Acquisition Service functional elements description:

Element Name	Type	Description
Sensor Data Acquisition Server	<<System>>	The Sensor Data Acquisition Server hosts the software for the Sensor Data Acquisition Service. This software is composed of multiple software components, modules.
Sensor Data Acquisition Service	<<Software>>	Core software element implementing the service logic, composed of individual modules.
Service Data Acquisition Processor	<<Software>>	This module is responsible for the processing and coordination the sending of information.
Persistence Module	<<Software>>	The second module is the Persistence Module, which is responsible for the saving of all the information of the service, gathered or processed.
Communication Module, Message Broker (MB) Client Module	<<Software>>	The Communication Module and Message Broker (MB) Client Module are responsible for the communication between the service and the end-user applications.
RDBMS	<<System>>	Relational Database Management to store all the data that is send by the end devices, as well as the process data from the service.

5.2.2 Service Provision: <<Intrusion Detection Service>>

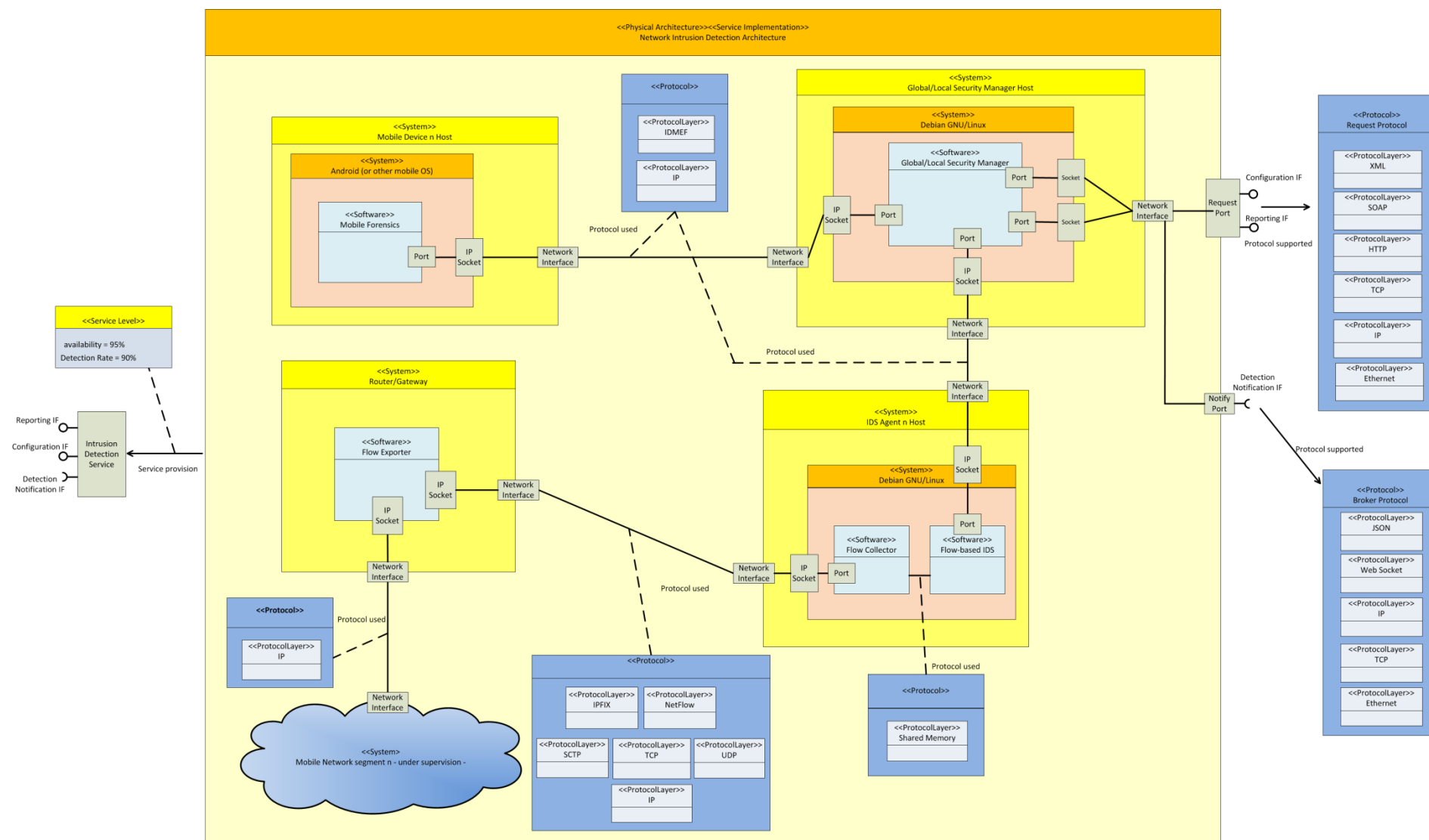


Figure 29 – Intrusion Detection Service architecture

Intrusion Detection Service functional elements description:

Element Name	Type	Description
Global/Local Security Manager Host	<<System>>	The Sensor Data Acquisition Server hosts the software for the Global/Local Security Manager.
Mobile Device n Host	<<System>>	Hardware with the mobile forensics software installed. May be a dedicated mobile/smart phone. n denotes, that there are arbitrary numbers of mobile devices under supervision.
IDS Agent n Host	<<System>>	The IDS Agent n Host hosts the software for the flow-based intrusion detection. Several software components contribute to this. n denotes, that there may be several hosts deployed in order to observe dedicated network segments independently.
Mobile Network segment n - under supervision -	<System>>	Dedicated part of a (mobile) network whose network traffic (IP-layer traffic) is analysed on potential intrusion attempts. n indicates that there may be several network segments under observation.
Global/Local Security Manager	<<Software>>	Gathers security reports from the detection systems and provides data to the service interfaces.
Flow Exporter	<<Software>>	Exports flow information gathered on a collection point (A router or separate system) to a flow collector.
Flow Controller	<<Software>>	Gathers flow data from one or more flow exporters and writes them to the file system.
Flow-based IDS	<<Software>>	Analyses flow data to detect intrusion attempts on the network
Mobile Forensics	<<Software>>	Detects intrusion attempts on a mobile device. Software running on very individual device to be supervised.

5.2.3 Service Provision: <<Message Brokering Service>>

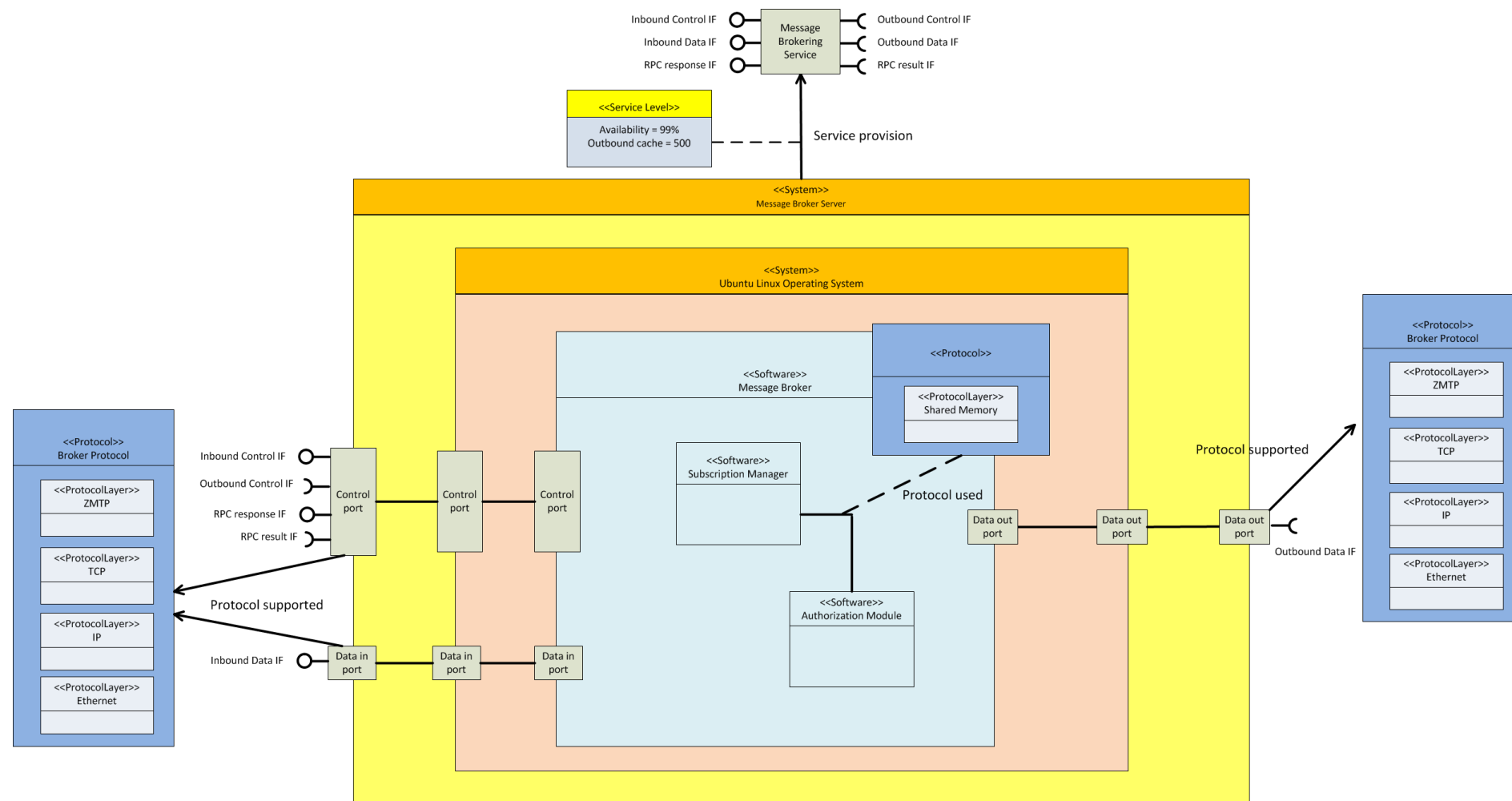


Figure 30 – Message Brokering Service architecture

Message Brokering Service functional elements description:

Element Name	Type	Description
Message Broker Server	<<System>>	The Message Broker Server hosts the software for the Message Brokering Service.
Message Broker	<<Software>>	Core software component that implements the service logic. This software is further composed of multiple software components/modules encapsulating certain aspects of the service logic.
Subscription Manager	<<Software>>	Module within the core service software responsible for managing subscriptions of service consumers in order to use the services data exchange capabilities.
Authorization Module	<<Software>>	Module within the core service software responsible for checking if service consumers accessing the functionalities provided by the service (via its operations) are authorized to do so.

5.2.4 Service Provision: <<Force Tracking Service>>

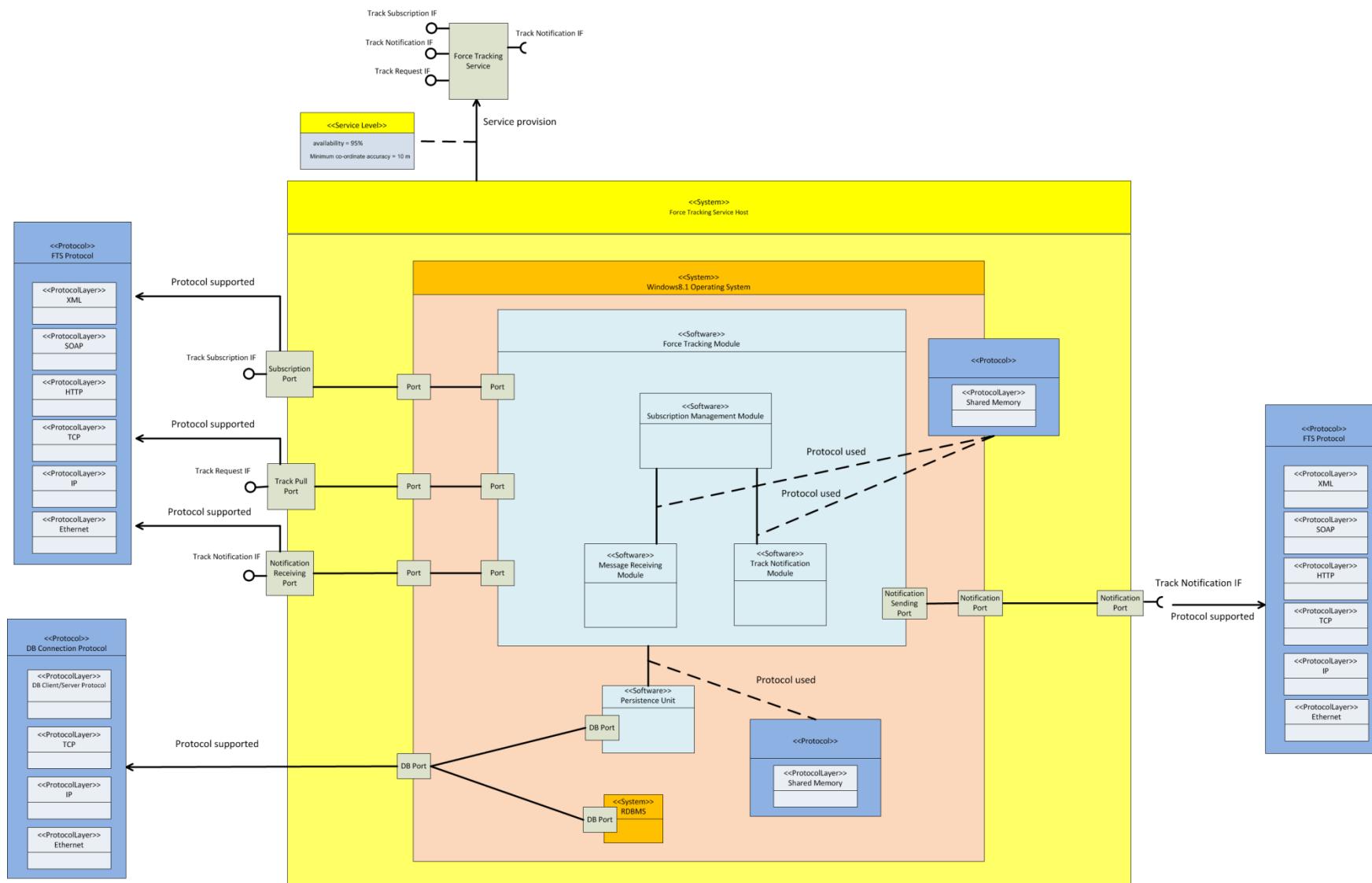


Figure 31 – Force Tracking Service architecture

Force Tracking Service functional elements description:

Element Name	Type	Description
Force Tracking Service Host	<<System>>	The Force Tracking Service Host hosts the software for the Force Tracking Service.
Force Tracking Module	<<Software>>	Core software component that implements the service logic. This software is further composed of multiple software components/modules encapsulating certain aspects of the service logic.
Subscription Management Module	<<Software>>	Module within the core service software responsible for managing subscriptions of service consumers in order to use the services data exchange capabilities.
Message Receiving Module	<<Software>>	Module within the core service software responsible for receiving incoming requests, validate messages payload and coordinates with the Subscription Management Module if requests are valid.
Track Notification Module	<<Software>>	Module within the core service software responsible for sending messages to service consumers.
Persistence Unit	<<Software>>	The Persistence Module is responsible for the saving of all the information of the service, gathered or processed.

BIBLIOGRAPHY

- [1] ETSI TS 102 181 v1.2.1 (2008-02), "Emergency Communications (EMTEL); Requirements for communication between authorities/organizations during emergencies", 2008
- [2] SALUS Deliverable 3.1 "System requirements, Enterprise Architecture and methodology", March 2014
- [3] SALUS Deliverable 2.1 "SALUS PPDR use cases – Intermediate", November 2013.
- [4] Website TOGAF, <http://www.opengroup.org/togaf/>
- [5] Open Safety & Security Architecture Framework (OSSAF), <http://www.openssaf.org/download>
- [6] Website Zachman Framework, <http://zachman.com/>
- [7] NATO Architecture Framework Version 3, ANNEX 3 TO AC/322(SC/1-WG/1)N(2007)0004.
- [8] NATO Architecture Framework Version 3, APPENDIX 1 TO ANNEX 3 TO AC/322(SC/1-WG/1)N(2007)0004.
- [9] SALUS Deliverable 5.2 "PPDR Security Architecture, end-to-end security, privacy mechanisms and intrusion detection approach, intermediate", January 2015.
- [10] SALUS Deliverable D4.3 "Business Analysis –Intermediate", March 2015.
- [11] SALUS Deliverable D6.1 "Connectivity, QoS Provisioning and Seamless Mobility for Hetnets – Intermediate", January 2015.
- [12] SALUS Deliverable D6.2 "PMR-over-LTE services definition – Intermediate", February 2015.

ACRONYMS

3GPP	Third Generation Partnership Program
API	Application Programming Interface
CCTV	Closed Circuit TV
COI	Community of Interest
DEA	Diameter Edge Agent
DMO	Direct Mode Operations
EA	Enterprise Architecture
EAI	Enterprise Application Integration
EMS	Emergency Medical Services
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
IDS	Intrusion Detection Service
IF	Interface
IMAP	Internet Message Access Protocol
IPFIX	Internet Protocol Flow Information Export
IT	Information Technology
IP	Internet Protocol
JSON	JavaScript Object Notation
LTE	Long Term Evolution
MDM	Mobile Device Management
MPLS	Multi-Protocol Label Switching
NAF	NATO Architecture Framework
NATO	North Atlantic Treaty Organization
NAV	NATO All View
NCV	NATO Capability View
NOV	NATO Operational View
NSOV	NATO Service-Oriented View
NSV	NATO System View
NTV	NATO Technical View
OSSAF	Open Safety & Security Architecture Framework
PCRF	Policy Control and Rating Function
PEI	Peripheral Equipment Interface
POP	Post Office Protocol
PPDR	Public Protection & Disaster Relief
PS&S	Public Safety & Security
QoS	Quality of Service
RDBMS	Relational Database Management system
RPC	Remote Procedure Call
SA	System Architecture
SLA	Service Level Agreement
SMTP	Simple Mail Transfer Protocol

SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TETRA	Terrestrial Trunked Radio
UE	Network User
UML	Unified Modeling Language
VMS	Video Management System
Wi-Fi	Wireless Fidelity
WP	Work Package
WSDL	Web Services Description Language
XMPP	Extensible Messaging and Presence Protocol
XML	Extensible Markup Language
ZAF	Zachman Architecture Framework
ZMQ	ZeroMQ Message Transport Protocol

ANNEX A – INFORMATION ITEMS AND INFORMATION EXCHANGE REQUIREMENTS

This annex describes the details of the Information Exchanges between operational nodes.

Information Item Name	Information Item Description	Sending Node	Receiving Node	Required Information Exchange Attributes			
				Media (text, video, voice, data etc.)	Quality (frequency, timeliness, security, etc.).	Quantity (volume, speed etc.)	Abilities (communication, communication-pattern, processing, display etc.)
Force Identification	Information item describing the position of a PPDR resource in space and time and also the resource status and the resource type (for example which types of unit this resource represents). (EA: Location, type and state information on PPDR-forces i.e. individual humans, organizations or physical Assets over time.)	Police on Field	Police Tactical CCC	Preferably via data (location, status) but could also be provided via voice (location, status) or text (status)	* location (periodic), frequency can change according to situation from 10s of seconds to seconds * Status (occasionally during change of status -e.g. "available", "on-scene", ...) * Resource type (once, at connection)	Low volumes (10s of bytes typically) per unit	Pull-Mode (default); could also be push-mode
		Riot Police	Police Tactical CCC	Idem	Idem	idem	Pull-Mode (default); could also be push-mode
		Police Tactical CCC	Police on Field	Idem	When needed as part of the situational awareness sharing. When shared, periodic updates according to situation (see above)	Low volumes (few Kbytes typically) - aggregation of all data from all units in the group. Sent to all authorized users (from selected groups)	Pull-Mode (default); could also be push-mode

		Police Tactical CCC	Riot Police	Idem	Idem	idem	Push-Mode
		Police on Field	Police on Field	Idem	On demand, and only when authorized by CCC. Same quality as Police on field -> Police CCC information flow)	Low volumes (10s of bytes typically) per unit. Sent to all authorized units from selected groups	Push- and Pull-Mode
		Riot Police	Riot Police	Idem	On demand, and only when authorized by CCC. Same quality as Police on field -> Police CCC information flow)	idem	Push- and Pull-Mode
		EMS Field Team	EMS Tactical CCC	Idem	* location (periodic), frequency can change according to situation from 10s of seconds to seconds * Status (occasionally during change of status -e.g. "available", "on-scene", ...) * Resource type (once, at connection)	idem	Pull-Mode (default); could also be push-mode
		Fire Fighters	Fire Fighters Tactical CCC	Idem	Idem	idem	Pull-Mode (default); could also be push-mode
		Fire Fighters Tactical CCC	Police Tactical CCC	Idem	On demand	Low volumes (few Kbytes typically) - aggregation of all data from all units in the group	Pull-Mode (default); could also be push-mode

Situation Report	Description of the situation on field	Police on Field	Police Tactical CCC	Different media possible (e.g. voice, picture, video...)	On demand: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode
	In case of Observation semantics : An observation is an information element with arbitrary information content. This type of information element is of raw type, which means there has been no additional validation or processing. This type of information is transported via different types of media e.g. voice, imagery, video, textual etc.	Riot Police	Police Tactical CCC	Different media possible (e.g. voice, picture, video...)	On demand: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode
	In contrast to SITREPs which represent information instantiated according to a dedicated model. -> basic syntax and semantic check.	Police on Field	Police on Field	Different media possible (e.g. voice, picture, video...)	On demand (and eventually authorized by Police CCC) * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Video: typically from 500 kbps to	Push-Mode

						10 Mbps per video stream.	
		Riot Police	Riot Police	Different media possible (e.g. voice, picture, video...)	On demand (and eventually authorized by Police CCC) * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode
		Police Tactical CCC	Strategic CCC	Different media possible (e.g. voice, picture, video...) most probably formal reports according to defined formats	On demand: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream.	Push and Pull-Mode??

		Fire Fighters Tactical CCC	Strategic CCC	Different media possible (e.g. voice, picture, video...) most probably formal reports according to defined formats	On demand: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream.	Push and Pull-Mode??
		EMS Tactical CCC	Strategic CCC	Different media possible (e.g. voice, picture, video...) most probably formal reports according to defined formats	On demand: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...), * Picture = high quality (M pixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream.	Push and Pull-Mode??
Emergency Assistance Request	Ultra-high priority communication used in case of imminent peril or danger. May include different information items. Usually based on the push of an emergency button of the device	Police on Field	Police Tactical CCC	Different (combination of) media possible: voice, data (e.g. location, movement, vital signs...), video	Used in case of immediate peril / danger (by usually pressing an emergency button on the device). Needs to be processed with ultra-high priority (can pre-empt lower priority communications if needed)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per	Push-Mode

						video stream.	
		Riot Police	Police Tactical CCC	Different (combination of) media possible: voice, data (e.g. location, movement, vital signs...), video	Used in case of immediate peril / danger (by usually pressing an emergency button on the device). Needs to be processed with ultra-high priority (can pre-empt lower priority communications if needed)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode
		Police on Field	Police on Field	Different (combination of) media possible: voice, data (e.g. location, movement, vital signs...), video	Used in case of immediate peril / danger (by usually pressing an emergency button on the device). Needs to be processed with ultra-high priority (can pre-empt lower priority communications if needed)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per video stream. Sent to all units of the group and/or in the vicinity of the unit in distress.	Push-Mode

		Riot Police	Riot police	Different (combination of) media possible: voice, data (e.g. location, movement, vital signs...), video	Used in case of immediate peril / danger (by usually pressing an emergency button on the device). Needs to be processed with ultra-high priority (can preempt lower priority communications if needed)	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per video stream. Sent to all units of the group and/or in the vicinity of the unit in distress.	Push-Mode
Medical Assistance Request	Used to ask support of EMS units. May include additional information such as location, number of injured people, type of injuries...	Police on Field	Police Tactical CCC	Different (combination of) media possible: voice, data (e.g. location, vital signs...), picture, video	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode

			Riot police	Police Tactical CCC	Different (combination of) media possible: voice, data (e.g. location, vital signs...), picture, video	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode
			Police Tactical CCC	EMS Tactical CR	Different (combination of) media possible: voice, data (e.g. location, vital signs...), picture, video	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture: typically few MB per picture, * Data: typically few 10s of KB * Video: typically from 500 kbps to 10 Mbps per video stream.	Push-Mode
Task		General purpose Task EA: Information for PPDR organizational units triggering specific behavior. Normally initiated by	Police Tactical CCC	Police on Field	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically	Push-Mode

		roles with dedicated tasking authority for example a commanding police officer. This type of information is transported via different types of media e.g. voice, or textual.					few 10s of KB Can be sent to a particular unit of to a selected group of units	
			Police Tactical CCC	Riot Police	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode
			EMS Tactical CCC	EMS Field Team	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode

			Fire Fighters tactical CCC	Fire Fighters	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode
			Strategic CCC	Police Tactical CCC	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode
			Strategic CCC	Fire Fighters Tactical CCC	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB	Push-Mode

	Imagery task	Special Tasks with dedicated semantics	Strategic CCC	EMS Tactical CCC	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB	Push-Mode
			Police Tactical CCC	Police on Field	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode
			Police Tactical CCC	Riot Police	Typically voice and/or data (mission order, map with instructions...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode

	Arrest Task		Police Tactical CCC	Police on Field	Typically voice and/or data (e.g. picture of the persons to arrest, location of person to arrest...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture typically few MB per picture * Data: typically 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode
			Police Tactical CCC	Riot Police	Typically voice and/or data (e.g. picture of the persons to arrest, location of person to arrest...)	On demand	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Picture typically few MB per picture * Data: typically 10s of KB Can be sent to a particular unit of to a selected group of units	Push-Mode

Emergency Status Information	(EA: Represents an discrete information referencing on a specific observable situation type. Description takes place according to a semantic understating of situations (situation model). => Basic syntax and semantic check. Information is captured in discrete messages)	Police Tactical CCC	Police on Field	Typically voice and/or data (formalized message for automatic processing at device level)	On demand, real-time * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...),	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB	Push- Mode
	(EA: Represents an discrete information referencing on a specific observable situation type. Description takes place according to a semantic understating of situations (situation model). => Basic syntax and semantic check. Information is captured in discrete messages)	Police Tactical CCC	Riot Police	idem	idem	idem	Push- Mode,
Situation Information	Information item describing situation relevant to decision making and actions. May also include formal messages (specific message formats) according to specific taxonomies an domain	Police Tactical CCC	Police on Field	Typically voice and/or data (formalized message for automatic processing at device level)	On demand, real-time and non-real-time: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...),	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB	Push-Mode

models for formal description and automatic processing. (EA: In case of TSO information is represented as an discrete information referencing on a specific observable situation type. Description takes place according to a semantic understating of situations (situation model). In either case, information has been validated and most probably undergone additional processing steps. => Basic syntax and semantic check. Information is captured in discrete messages	FireFighters Tactical CCC	Fire Fighters	idem	idem	idem	Push-Mode
	Police Tactical CCC	Riot police	idem	idem	idem	Push-Mode
	Police Tactical CCC	EMSTactical CCC	Different (combination of) media possible: data (also specific message formats), imagery, video	On demand, non-real-time, high-resolution images and video possible: * Picture = high quality (Mpixel) and near real-time (few seconds), * Video = high quality (QCIF, 720p, higher)	Depends on media type: * Picture : typically few MB per picture, * Video: typically from 500 kbps to 10 Mbps per video stream, * Data: typically few 10s of KB	Push- and Pull-Mode
	Police Tactical CCC	Fire Fighters Tactical CCC	idem	idem	idem	Push- and Pull-Mode
	EMS Tactical CCC	EMS Field Team	Typically voice and/or data (formalized message for automatic processing at device level)	On demand, real-time and non-real-time: * voice = clear audio quality even in noisy environment (e.g. crowd, engine, alarms...),	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB	Push-Mode

Mission Progress Report	Reports the progress of a dedicated mission initiated via corresponding request or task		EMS Field Team	EMS Tactical CCC	Typically voice and/or data (formalized message for automatic processing at device level)	On demand or at predefined frequency or predefined events occurring during mission	Depends on media type: * voice = typically between few kbps to few 10s of kbps per voice channel; * Data: typically few 10s of KB	Push- and Pull Mode, requires interactive reporting tool at device level in case of formalized messages
			Fire Fighters	Fire Fighters CCC	idem	idem	idem	idem
Medical Assistance Response	Information item, that describes if an Medical assistance request can be served or not		EMS Tactical CCC	Police Tactical CCC	Typically data (formalized message for automatic processing at device level	On demand, near real-time (few seconds)	Data: typically few 10s of KB	Push-Mode
Medical Assistance Task Status Information	Provides information of an actual served medical assistance previously requested and served		EMS Tactical CCC	Police Tactical CCC	Typically data (formalized message for automatic processing at device level)	On demand or at predefined frequency or predefined events occurring during medical task execution	Data: typically few 10s of KB	Push-Mode
Sensor Data	Video	Video footage, real or elapse time	Police on Field	Police Tactical CCC	Video (digital)	On demand	Live Streams	Push-Mode and pull mode
			Riot Police	Police Tactical CCC	Video (digital)	On demand	Live Streams	Push-Mode and pull mode
			CCTV Operator	Police Tactical CCC	Video (digital)	On demand	Live Streams and Clips	Push-Mode and pull mode

	Health Status							
	Image	Still Image as part of a situation report	Police on Field	Police Tactical CCC	Photograph (digital)	On demand	Periodically as and when beneficial to the effort to gain public order	Push-Mode
			Riot Police	Police Tactical CCC	Photograph (digital)	On demand	Periodically as and when beneficial to the effort to gain public order	Push-Mode
Criminal Intelligence Request		Records of previous or current offences or serious crime, current wanted or arrest warrants	Police on Field	Police Tactical CCC	Voice communioniation	On demand	Periodically either when a suspect is apprehended or and an attempt to apprehend is being planned	Push-Mode
		(EA: Request to assign identity on individual objects under consideration. These objects could be individual humans, groups of humans, vehicles etc.)	Riot Police	Police Tactical CCC	Voice communioniation	On demand	Periodically either when a suspect is apprehended or and an attempt to apprehend is being planned	Push-Mode
CCTV-Control Request		Request to take control of CCTV cameras owned by council-run CCTV operators	Police Tactical CCC	CCTV-Operator	Voice communication ?	On demand	Periodically as and when beneficial to the effort to gain public order	Push-Mode

CCTV-Control Response		CCTV-Operator	Police Tactical CCC	Voice communication?	On demand	Periodically as and when beneficial to the effort to gain public order	Push-Mode
CCTV-Video Streaming Request	Indirect control of the camera (based on location information) in order to get the stream of a dedicated situation! -> live stream, request for historic data may contain additional parameters.	Police Tactical CCC	CCTV-Operator	Voice communication ?	On demand	Live Streams and Clips (is it the video stream or the request that we are recording here?)	??
CCTV Control	Provides information necessary for direct control of CCTV devices (cameras)	CCTV-Operator	Police Tactical CCC	??	On demand	Periodically as and when beneficial to the effort to gain public order	??
Positive Identification	Response (positive) due to an Criminal Intelligence Request (EA: Information on the identity of individual objects under consideration. These objects could be individual humans, groups of humans, vehicles etc.)	Police Tactical CCC	Police on Field	Voice communication if access request via a dispatcher or mobile data if access via a mobile device	On demand	Periodically as and when beneficial to the effort to gain public order	Push-Mode
		Police Tactical CCC	Riot Police	Voice communication if access request via a dispatcher or mobile data if access via a mobile device	On demand	Periodically as and when beneficial to the effort to gain public order	Push-Mode
Request for Public Information	Request public information, such as activities on social networks, updated news on journals, newspapers,	Police on Field	Public Information Service Provider	data	On demand	data request based on HTTP traffic (access to content in social media sites)	Pull-Mode

	TV channels.	Police Tactical CCC	Public Information Service Provider	data	On demand	data request based on HTTP traffic (access to content in social media sites)	Pull-Mode
		Strategic CCC	Public Information Service Provider	data	On demand	data request based on HTTP traffic (access to content in social media sites)	Pull-Mode
Public information Response	Response from public information request containing Information from open information sources (social media, internet forums, websites, news-channels, newspapers etc.) on arbitrary information requirements/requests for PI	Public Information Service Provider	Police on Field	data, picture, video (non-real-time)	On demand	Depends on media type data = few kbps picture = few Mbps video = from kbps to Mbps	Pull-Mode
		Public Information Service Provider	Police Tactical CCC	data, picture, video (non-real-time)	On demand	Depends on media type data = few kbps picture = few Mbps video = from kbps to Mbps	Pull-Mode
		Public Information Service Provider	Strategic CCC	data, picture, video (non-real-time)	On demand	Depends on media type data = few kbps picture = few Mbps video = from kbps to Mbps	Pull-Mode

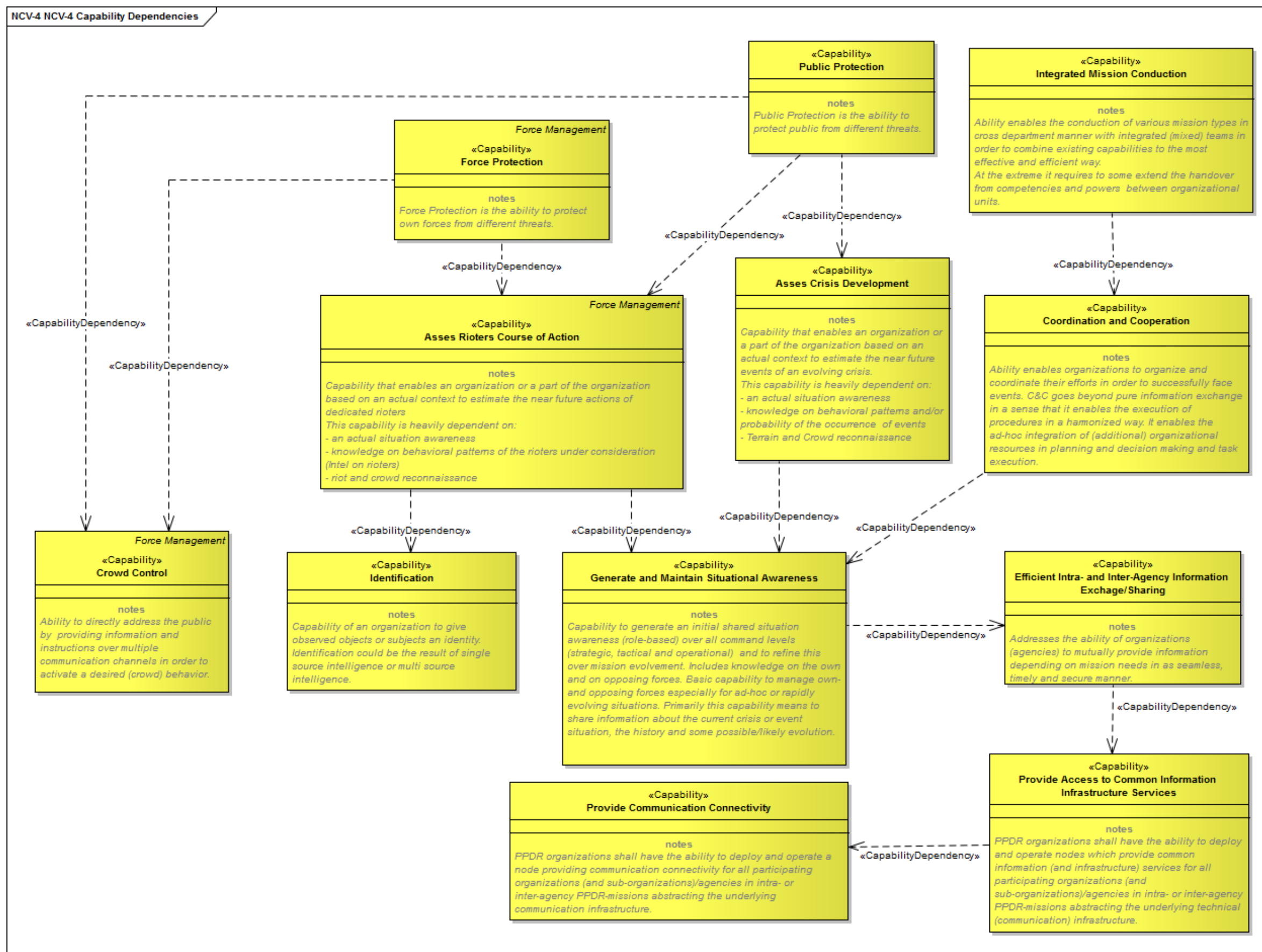
Fire and Rescue Assistance Request	Provides information regarding the request of rescue assistance resources.	Police Tactical CCC	Fire Fighters Tactical CCC	voice, video, data	On demand	Depends on media type voice = few kbps (might be long duration calls > 5minutes) video = from kbps to Mbps	Push-Mode and Pull Mode
Fire and Rescue Assistance Response	Response from request rescue assistance resources	Fire Fighters Tactical CCC	Police Tactical CCC	voice, video, data	On demand	Depends on media type voice = few kbps (might be long duration calls > 5minutes) video = from kbps to Mbps	Push-Mode and Pull Mode
Strategic Instructions	Provide information regarding strategic operations that must be followed	Strategic CCC	Police Tactical CCC	voice, video	On demand	Depends on media type voice = few kbps (might be long duration calls > 5minutes) video = from kbps to Mbps	Push-Mode and Pull Mode
		Strategic CCC	Fire Fighters Tactical CCC	voice, video	On demand	Depends on media type voice = few kbps (might be long duration calls > 5minutes) video = from kbps to Mbps	Push-Mode and Pull Mode

		Strategic CCC	EMS Tactical CCC	voice, video	On demand	Depends on media type voice = few kbps (might be long duration calls > 5minutes) video = from kbps to Mbps	Push-Mode and Pull Mode
Common Operational Picture	Relevant parts of the common recognized COP, not the full COP of the source CCC. (Remark: the recognized COP is a result of combining information of different sources via dedicated processes (including activities like processing, evaluation, assessment, validation and rating) into one common relevant operational picture. (EA: Represents individual and evaluated information on certain aspect of the commanders' information requirement. All these information elements make up the Situational Picture concerning various aspects of the commanders information needs in order to provide	Strategic CCC	Police Tactical CCC	data (formally structured with dedicated semantics e.g. JC3IEDM, imagery, video (non-real-time)	On demand or periodically as defined by sop's or special business rules		Push-Mode and Pull Mode (according to sop or rules)
		Strategic CCC	Fire Fighters Tactical CCC	data (formally structured with dedicated semantics e.g. JC3IEDM, imagery, video (non-real-time)	On demand or periodically as defined by sop's or special business rules		Push-Mode and Pull Mode (according to sop or rules)
		Strategic CCC	EMS Tactical CCC	data (formally structured with dedicated semantics e.g. JC3IEDM, imagery, video (non-real-time)	On demand or periodically as defined by sop's or special business rules		Push-Mode and Pull Mode (according to sop or rules)

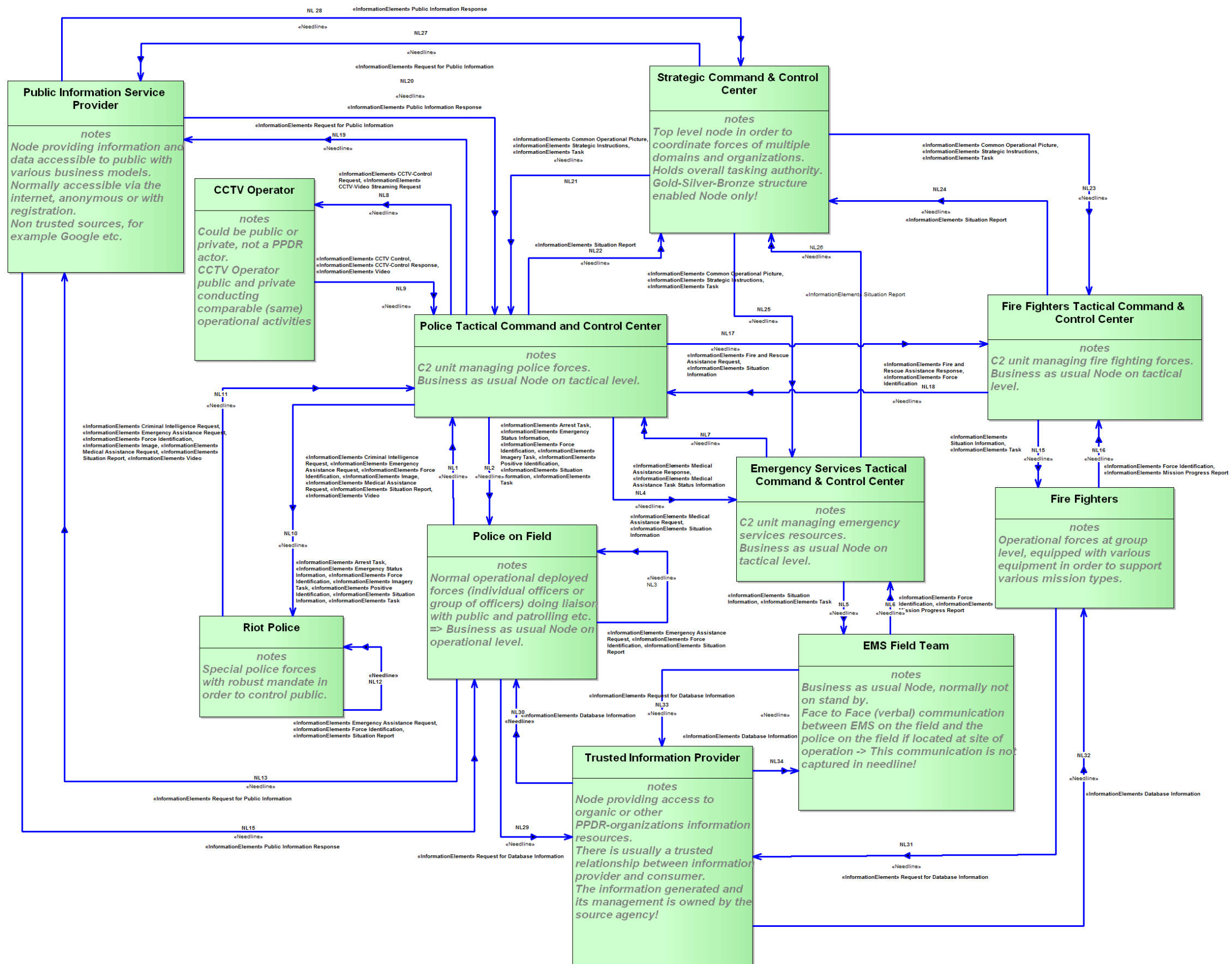
	the basis for the commanders' decision making process.)						
Request for Database Information	How is this info different from other info requests?	Police on Field	Trusted Information Provider	data	On demand, secured access	data request based on protocol above the IP-layer, most probably HTTP/HTTPS	Pull-Mode
	(EA: Information from confidential information sources (for example own organizations data- and knowledge-bases or from other trusted sources/organizations on arbitrary information requirements.)	Fire fighters	Trusted Information Provider	data	On demand, secured access	data request based on protocol above the IP-layer, most probably HTTP/HTTPS	Pull-Mode
		EMS Field Team	Trusted Information Provider	data	On demand, secured access	data request based on protocol above the IP-layer, most probably HTTP/HTTPS	Pull-Mode
Database Information	Information retrieved from a dedicated request to a specific database. Actually content agnostic. (EA: Information from	Trusted Information Provider	Police on Field	data - structured or unstructured, picture, video-clips	in response to corresponding request triggered by On demand needs, non-real-time, content protection required	Depends on media type data = few kb picture = few Mb video = from kb to Mb	Pull-Mode

	confidential information sources (for example own organizations data- and knowledge-bases or from other trusted sources/organizations on arbitrary information requirements.	Trusted Information Provider	Fire fighters	data - structured or unstructured, picture, video-clips	in response to corresponding request triggered by On demand needs, non-real-time, content protection required	Depends on media type data = few kb picture = few Mb video = from kb to Mb	Pull-Mode
		Trusted Information Provider	EMS Field Team	data - structured or unstructured, picture, video-clips	in response to corresponding request triggered by On demand needs, non-real-time, content protection required	Depends on media type data = few kb picture = few Mb video = from kbp to Mb	Pull-Mode

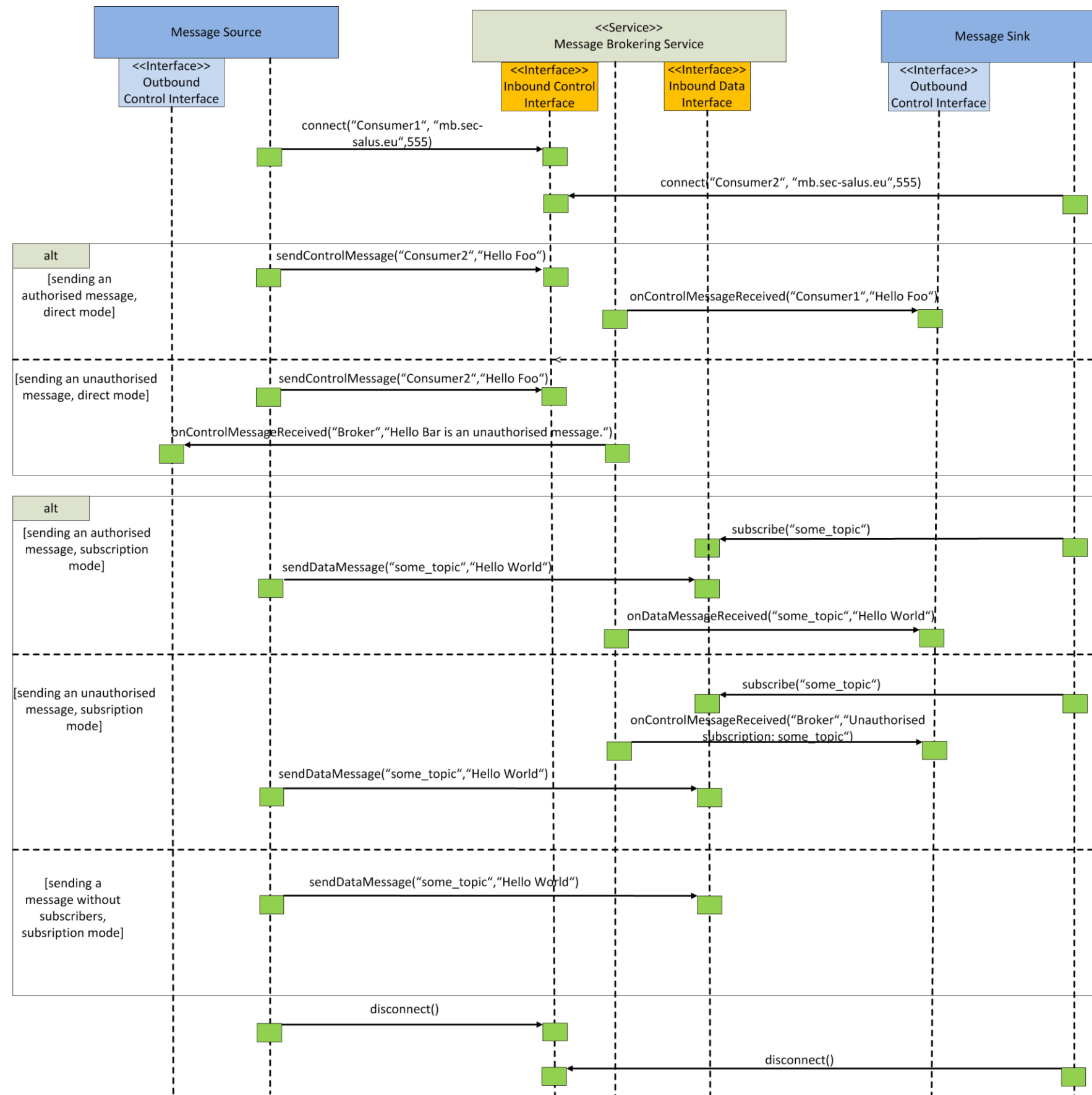
ANNEX B – SALUS RELATED CAPABILITIES OF PPDR ORGANISATIONS



ANNEX C – OPERATIONAL NODES AND INFORMATION EXCHANGE NEEDS



ANNEX D – INTERACTION REQUIRED WITH MESSAGE BROKER IN THE DIRECT MODE AND IN THE SUBSCRIPTION MODE



ANNEX E – INTERACTION REQUIRED WITH MESSAGE BROKER SENDING REMOTE PROCEDURE CALLS

