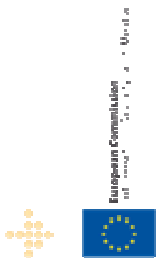


<b>D.DEPN.3.1</b>	<b>Design principles for a privacy protective, secure, safe and fault tolerant CVIS design</b>
-------------------	--

Sub-project No.	SP 1.3	Sub-project Title	DEPN
Workpackage No.	Topic 3	Workpackage Title	Security, safety and fault tolerance
Author(s)	Paul van Koningsbruggen, Nol Venema, Dave Marples		
Dissemination level PU/PP/RE/CO	PU		
File Name	DEL_DEPN_3.1_SafetySecurity&FailureModeAnalysis_V2.0		
Due date	28 February 2010		
Delivery date	15 July 2010		

<b>Abstract</b>	Safe, secure, and fault-tolerant design is one of the deployment enablers of cooperative vehicle – infrastructure systems (CVIS). To support deployment of CVIS a design principles are drafted for ensuring whole-system privacy protection, security, safety and fault tolerance. With these design principles the CVIS architecture can be matured.
-----------------	--

	<p>Project supported by European Union DG INFSO</p> <p>IST-2004-2.4.12 eSafety – Cooperative systems for road transport</p>
Project reference	FP6-2004-IST-4-027293-IP
IP Manager	Paul Kompfner, ERTICO – ITS Europe Tel: +32 2 400 0700, E-mail: <a href="mailto:cvis@mail.ertico.com">cvis@mail.ertico.com</a>

## Control sheet

Version history			
Version	Date	Main author	Summary of changes
1.0	07/04/2009	Paul van Koningsbruggen, Nol Venema	First version
1.1	04/02/2010	Technolution	Review comments coming from Annual Review EC have been incorporated.
Working documents	February - May 2010	Paul van Koningsbruggen, Dave Marples, Nol Venema	
1.2	28 June 2010	Paul van Koningsbruggen, Dave Marples	Full draft
1.3	29 June 2010	Nol Venema	Strengthened the privacy principles design
	Name		Date
Prepared	Paul van Koningsbruggen, Dave Marples, Nol Venema		09/07/2010
Reviewed	Jean-Francois Gaillet (YGOMI)		13/07/2010
Authorized	Lina Konstantinopoulou		14/07/2010
Circulation			
Recipient		Date of submission	
European Commission		14/07/2010	
Project partners		14/07/2010	

## Table of Contents

<b>TABLE OF CONTENTS</b> .....	<b>3</b>
<b>ABBREVIATIONS AND DEFINITIONS</b> .....	<b>5</b>
<b>EXECUTIVE SUMMARY</b> .....	<b>6</b>
<b>1. INTRODUCTION</b> .....	<b>7</b>
1.1. DESIGN PRINCIPLES .....	7
1.2. HIGH LEVEL OBJECTIVES FOR PRIVACY PROTECTIVE, SECURE, SAFE AND FAULT TOLERANT DESIGN .....	7
1.3. VIEWPOINTS ON COOPERATIVE SYSTEMS .....	8
1.4. DOCUMENT OUTLINE.....	9
<b>2. DESIGN RULES FROM ORGANISATIONAL VIEWPOINT</b> .....	<b>10</b>
2.1. INTRODUCTION ON THE ORGANISATIONAL VIEWPOINT .....	10
2.2. ADDITIONAL RESPONSIBILITIES OF THE ROLES.....	11
2.3. PROCESSES COMING WITH LIFE CYCLE MANAGEMENT .....	24
2.3.1. <i>Definition of ‘grades’ CVIS needs to guarantee</i> .....	24
2.3.2. <i>Processes for life cycle management</i> .....	26
2.3.3. <i>CVIS Process – Subscription to Service Aggregator</i> .....	29
2.3.4. <i>CVIS Process – Subscription to a service</i> .....	31
2.3.5. <i>CVIS Process – Service Deployment</i> .....	32
2.3.6. <i>CVIS Process – Service Provisioning</i> .....	34
2.3.7. <i>CVIS Process – Service Context Data Issuing</i> .....	36
2.3.8. <i>CVIS Process – Service Consumption</i> .....	38
2.3.9. <i>CVIS Process – Service Usage Data Generation &amp; Submission</i> .....	40
2.3.10. <i>CVIS Process – Service Payment</i> .....	43
2.3.11. <i>CVIS Process – Deregister from Service</i> .....	46
2.3.12. <i>CVIS Process – Deregister from Service Aggregator</i> .....	47
<b>3. CVIS FROM A FUNCTIONAL PERSPECTIVE</b> .....	<b>49</b>
3.1. PLATFORM, BASIC FUNCTIONS AND FUNCTIONS .....	49
3.2. PROTECTION PROFILE .....	51
3.2.1. <i>Protection by the Mobile Unit</i> .....	51
3.2.2. <i>Protection by the operating system and runtime environment</i> .....	51

3.2.3.	<i>Protection by the Service Primitives</i> .....	51
3.3.	CAPABILITY PROFILE .....	56
3.3.1.	<i>Safety Instrumented System</i> .....	56
3.3.2.	<i>Resource management</i> .....	57
3.4.	SAFETY VERSUS INTEROPERABILITY .....	60
3.5.	SECURITY .....	61
3.6.	WORLD MODEL.....	64
<b>4.</b>	<b>CVIS FROM AN INFORMATION PERSPECTIVE</b> .....	<b>65</b>
4.1.	CRITICALITY OF INFORMATION .....	65
4.2.	STATES AND STATE TRANSITIONS OF OBJECTS .....	65
4.3.	COOPERATION BETWEEN OBJECTS .....	66
4.3.1.	<i>Syntax and semantics</i> .....	66
4.3.2.	<i>Sources, Sinks and Transits</i> .....	66
4.3.3.	<i>Privacy in cooperation</i> .....	67
4.4.	TUNING OF SERVICES TO THEIR CURRENT CONTEXT (SERVICE CONTEXT DATA) .....	68
4.5.	INFORMATION STORAGE AND HANDLING .....	70
4.5.1.	<i>Dynamics Characteristics</i> .....	70
4.5.2.	<i>Data Types</i> .....	71
4.5.3.	<i>Information classification</i> .....	71
<b>5.</b>	<b>CVIS FROM AN ENGINEERING PERSPECTIVE</b> .....	<b>73</b>
5.1.	MINIMISE RISK ON INTERFERENCE BETWEEN SERVICE APPLICATIONS .....	73
<b>6.</b>	<b>CONCLUSIONS</b> .....	<b>75</b>
	<b>REFERENCES</b> .....	<b>78</b>

## Abbreviations and Definitions

Abbreviation	Definition
ADAS	Advanced Driver Assistance Systems
CALM	Communications, Air-interface, Long and Medium range
CF&F	Cooperative Freight and Fleet Management
CINT	Cooperative Interurban traffic
COMO	Cooperative Monitoring
COMM	Communications
CURB	Cooperative Urban traffic
CVIS	Cooperative Vehicle Infrastructure Systems
DSRC	Dedicated Short Range Communication
EU	European Union
FOAM	Framework for Open Application
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
GSM	Global System for Mobile
HMC	Host Management Centre
MU	Mobile Unit
PII	Personally Identifiable Information
POMA	Positioning, Mapping and Location Reference
RM-ODP	Reference Model for Open Distributed Processing
SA	Service Aggregator
SC	Service Centre
SU	Service User

## Executive Summary

A privacy protective, secure, safe, and fault-tolerant design is one of the deployment enablers of cooperative vehicle – infrastructure systems (CVIS). To support deployment of CVIS a statement of principles for ensuring whole-system privacy, security, safety and fault tolerance, as input to designers of core technologies and applications within the CVIS context, helps to enhance the CVIS design so far as captured in the architecture (and interface specifications) documents.

The first step towards such a statement of principles is to assess the completeness of the CVIS definition (so far) from safety, security and fault tolerance perspective. The overall threats within cooperative systems must be understood, as well as the possible counter measures to tackle these threats. This is the focus of this first DEPN Topic 3 deliverable (D.DEPN.3.1).

Using the discussed and accentuated analyses results in this second DEPN Topic 3 deliverable (D.DEPN.3.2) the proposed countermeasures are detailed out into design principles for ensuring whole-system privacy, security, safety and fault tolerance.

## 1. Introduction

### *1.1. Design principles*

A privacy protective, safe, secure, and fault-tolerant design is one of the deployment enablers of cooperative vehicle – infrastructure systems (CVIS). To support deployment of CVIS a set of design principles for ensuring whole-system privacy protection, safety, security and fault tolerance, as input to designers of core technologies and applications within the CVIS context, is needed.

The first step towards such a set of design principles is to assess the completeness of the CVIS definition (so far) from privacy, safety, security and fault tolerance perspective. The overall threats within cooperative systems must be understood, as well as the possible counter measures to tackle these threats. This is the focus of the first DEPN Topic 3 deliverable (D.DEPN.3.1). The analysis is based on the work package 2 and 3 deliverable of the CVIS core technologies and applications subprojects.

The second step is to derive design principles from this analysis. This is the focus of this second DEPN Topic 3 deliverable (D.DEPN.3.2).

### *1.2. High Level objectives for privacy protective, secure, safe and fault tolerant design*

Coming from the analyses depicted in deliverable D.DEPN.3.1 the design of CVIS should include the functionality to:

- From privacy perspective:
  - operate a service without using personal data if possible
  - personal data must be collected for explicit and legitimate purposes and used accordingly;
  - personal data must be relevant and not excessive in relation to the purpose for which they are processed;
  - process personal data fairly and lawfully;
  - safeguard that personal data is accurate and kept up to date;
  - safeguard that personal data that identifies individuals is not be kept longer than necessary;
- From security perspective
  - set up a circle of trust between cooperating actors and their units or centres;
  - protect data;

- provide end-to-end security in communications;
- From safety perspective:
  - unambiguously demarcate the field of application of a service;
  - guarantee interoperability of service operations
  - tune the service operations to the penetration rate of service users
- From fault tolerance perspective:
  - follow predefined structuring rules in the co operations (collaborations);
  - manage the latency in service operations;
  - manage an appropriate usage of the available resources in the unites and centres;
  - safeguard the integrity of data;
  - safeguard the validity ('up-to-date) of data;
  - Manage the availability of the communication bearers.

### ***1.3. Viewpoints on Cooperative Systems***

CVIS has adopted the Reference Model for Open Distributed Processing (RM-ODP) as a structure for its architecture. In line with the RM-ODP approach the design recommendations from privacy, security, safety and fault-tolerance point of view are structured along the RM-ODP 'viewpoints' [12, 13, 14, and 15].

A viewpoint (on a system) is an abstraction that yields a specification of the whole system related to a particular set of concerns. The five viewpoints covering all the domains of architectural design are:

- the organisational viewpoint, which is concerned with the purpose, scope and policies governing the service convergence and the roles;
- the functional viewpoint, which is concerned with the functional decomposition of the system into a set of objects that interact at interfaces – enabling system distribution;
- the information viewpoint, which is concerned with the kinds of information handled by the system and constraints on the use and interpretation of that information;
- the engineering viewpoint, which is concerned with the infrastructure required to support system distribution;

- The technology viewpoint, which is concerned with the choice of technology to support system distribution.

The design rules focus on the first four viewpoints with a glance on the fourth viewpoint. The technology viewpoint is considered to be the ultimate responsibility of the manufacturers.

#### ***1.4. Document Outline***

The document outline follows the viewpoints as set out in paragraph 1.3:

- design rules from organisational viewpoint in chapter 2;
- design rules from functional viewpoint in chapter 3;
- design rules from information viewpoint in chapter 4;
- Design rules from engineering viewpoint in chapter 5.

The document is concluded in chapter 6 with an overview of the high level objectives for privacy protective, secure, safe and fault tolerant design, and the derived design rules to fulfil these objectives.

## 2. Design Rules from Organisational Viewpoint

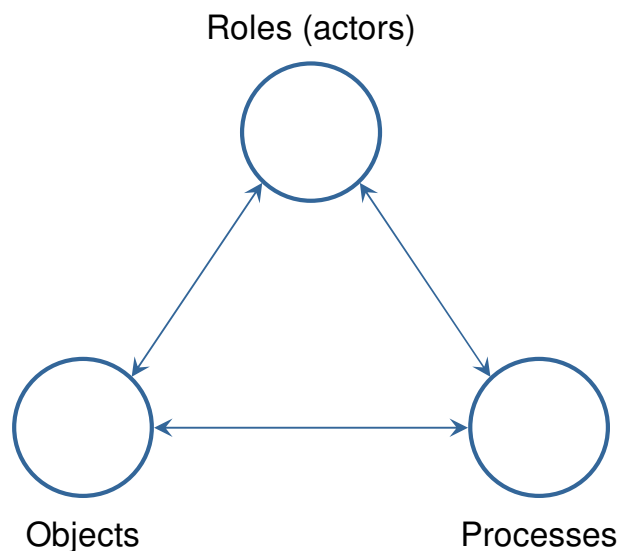
### 2.1. Introduction on the organisational viewpoint

The aim of the organisational specification is to express the objectives and policy constraints of the community needed to create cooperative vehicle – infrastructure systems (CVIS). In order to do this, the community is represented by (Figure 1): (i) roles together forming the community, (ii) the processes the roles are involved in and (iii) the objects, that are the resources for the roles.

CVIS comes with a particular kind of community that is a federation. A federation is a coming together of a number of roles answering to different authorities in order that they may jointly cooperate to achieve the objective of cooperativeness. One of the key ideas in the organisational specification is that of a contract, linking the performers of the various roles in a community and expressing their mutual obligations. This contract expresses the common goals and responsibilities which distinguish roles in a community and is expressed in terms of: processes, structuring rules and the reference points.

To meet with this idea the additional roles and / or responsibilities and the processes are defined that:

- are relevant from the perspective of privacy, security, safety and fault tolerance;
- Should be added to the CVIS architecture that focuses quite strongly on the objects.

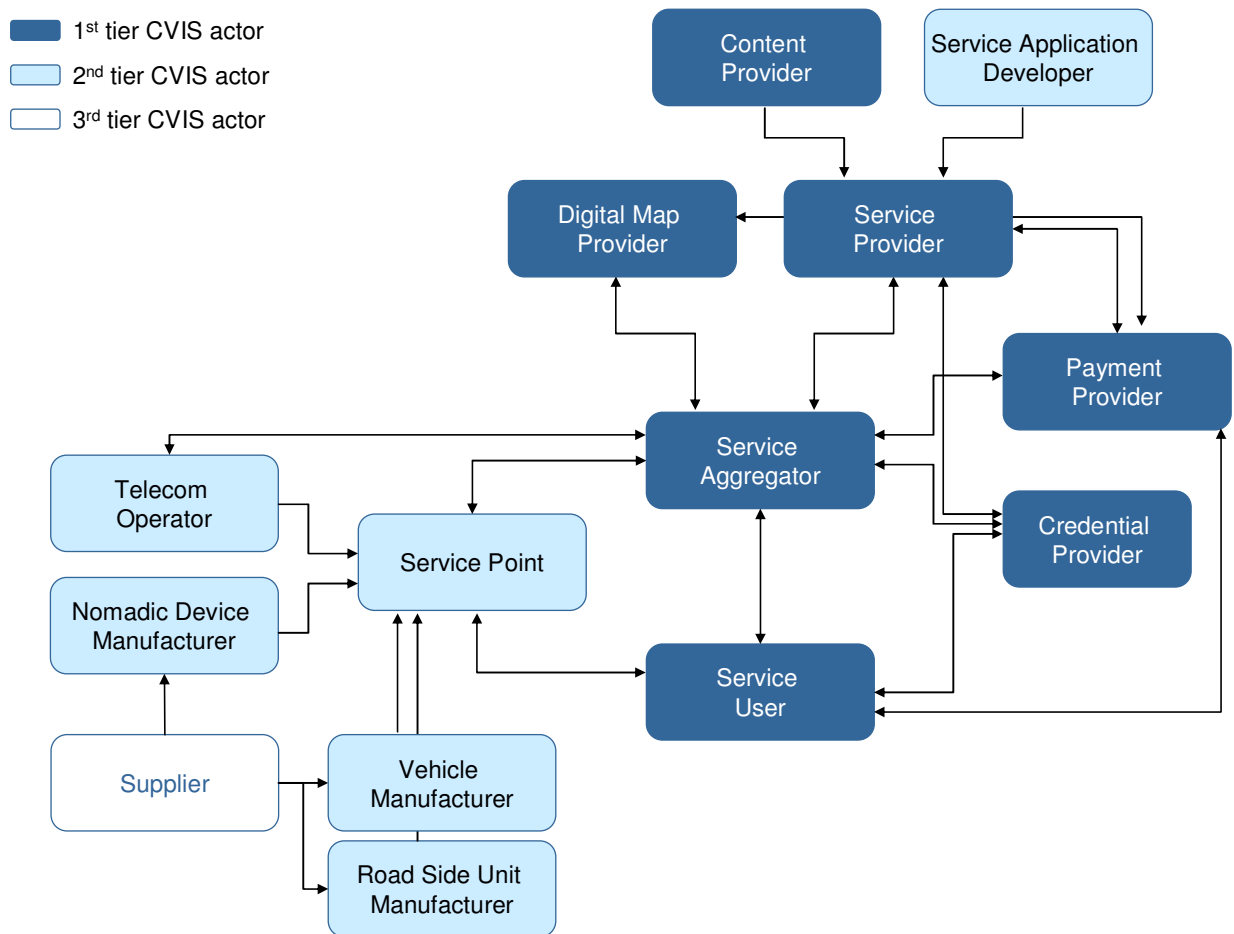


**Figure 1. Direct link between roles, objects and processes from organisational viewpoint**

## 2.2. Additional responsibilities of the roles

Design rule 1: define roles and their responsibilities, including the responsibilities with respect to safeguarding privacy, security, safety and fault tolerance.

Within the context of CVIS the roles should be distinguished as depicted in Figure 2.



**Figure 2. Roles within the context of CVIS relevant for privacy, security, safety and fault tolerance**

For every role the responsibilities should be defined following from the High Level Objectives for privacy protective, secure, safe and fault tolerant design. In the rest of this paragraph the initial impetus is given.

Role	Content Provider
Processes involved	<ul style="list-style-type: none"> <li>Service Consumption</li> </ul>
Generic responsibilities	<p>A Content Provider collects, verifies, processes and fuses data into information relevant for the services provided by Service Operators.</p> <p>A special Content Provider is the competent authority who provides authorised data to be used in services, e.g. speed limits, road construction works or routes for hazardous goods vehicles.</p>
Responsibilities from privacy perspective	<ul style="list-style-type: none"> <li>Guarantee that provided data cannot be traced back to individuals</li> </ul>
Responsibilities from security perspective	<ul style="list-style-type: none"> <li>Guarantee the integrity of the data</li> <li>Encrypt data when manipulation by an intruder can have serious consequences for safety or (more generic) the nature of the service using the data</li> </ul>
Responsibilities from safety perspective	<ul style="list-style-type: none"> <li>Provide an unambiguous data context definition (what is the field of application of the data, what is the reliability of the data, what is the validity date of the data)</li> <li>Provide reliable and correct data</li> </ul>
Responsibilities from fault tolerance perspective	<ul style="list-style-type: none"> <li>Guarantee that the required data (content) is available</li> <li>Guarantee that the required data (content) is valid</li> </ul>

Role	Digital Map Provider
Processes involved	<ul style="list-style-type: none"> <li>Service Consumption</li> </ul>
Generic responsibilities	<p>A Digital Map Provider collects, verifies, processes and fuses data into a digital map.</p>
Responsibilities from privacy perspective	<ul style="list-style-type: none"> <li>Guarantee that digital maps are free of personal data that can be traced back to individuals</li> </ul>

<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• Guarantee the integrity of the data forming the digital map</li> <li>• Encrypt data forming the digital map when manipulation by an intruder can have serious consequences for safety or (more generic) the nature of the service using the data</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>• Provide an unambiguous definition of the digital map:             <ul style="list-style-type: none"> <li>– what is the field of application of the map: advanced driver assistance, navigation, road user charging, etc.;</li> <li>– what is the reliability of the digital map;</li> <li>– what is the validity date of the digital map;</li> </ul> </li> <li>• Provide reliable and correct data forming the digital map.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>• Guarantee that the required digital maps are available;</li> <li>• Guarantee that the required digital map the Digital Map Provider side are valid;</li> <li>• Provide dynamic map updates to guarantee the validity at the Service User side of both the map data and the road safety related attributes (like speed limits, road construction works or routes for hazardous goods vehicles).</li> </ul>

Role	Service Provider
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• Service Context Data Issuing</li> <li>• Subscription</li> <li>• Service Provisioning</li> <li>• Service Consumption</li> <li>• Service usage data generation &amp; submission</li> <li>• Service Payment</li> </ul>
<b>Generic responsibilities</b>	<p>A Service Provider generates, provides and (optionally) bills for consumption of one or more services, which can be (from a CVIS perspective):</p> <ul style="list-style-type: none"> <li>• infrastructure related services (road user charging, paid parking, access control, etc);</li> </ul>

	<ul style="list-style-type: none"> <li>• traffic related information services (extended floating car data, enhanced driver awareness, etc);</li> <li>• transport related information services (tracking &amp; tracing, fine tuning time of arrival, etc);</li> <li>• mobility related services (traveller assistance, navigation, monitoring of driving &amp; rest times, etc);</li> <li>• Usage charging related services (pay-as-you drive insurance, etc).</li> </ul>
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>• operate a service using as less personal data as possible</li> <li>• offer an anonymous service besides a personalised service</li> <li>• process personal data fairly and lawfully;</li> <li>• safeguard that personal data is accurate and kept up to date;</li> <li>• safeguard that personal data that identifies individuals is not be kept longer than necessary;</li> <li>• Do not build a business model on privacy related data.</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• define identification, authentication and authorisation levels coming with the service development, deployment and operations;</li> <li>• set up a circle of trust with: <ul style="list-style-type: none"> <li>– Service Aggregator for deploying services</li> <li>– Service User for consumption of services</li> </ul> </li> <li>• Protect data coming with service operations.</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>• unambiguously demarcate the field of application of a service via Service Context Data;</li> <li>• guarantee interoperability of service operations</li> <li>• tune the service operations to the penetration rate of service users</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>• safeguard the availability of the service centre</li> <li>• manage an appropriate usage of the available resources in the service centres;</li> </ul>

	<ul style="list-style-type: none"> <li>• manage experienced latency in service operations;</li> <li>• safeguard the integrity of data used in service definition and operations;</li> <li>• safeguard the validity of data used in service definition and operations;</li> <li>• data processing should be in time so data can be communicated with on-coming vehicles (in case Mobile Unit is placed behind wind screen)</li> </ul>
--	--

Role	Service Aggregator
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• Initialisation &amp; Activation</li> <li>• Service Context Data Issuing</li> <li>• Service Deployment</li> <li>• Subscription</li> <li>• Service Provisioning</li> <li>• Service usage data generation &amp; submission</li> <li>• Service Payment</li> </ul>
<b>Generic responsibilities</b>	<p>The Service Aggregator is the enabler on behalf of Service Providers for:</p> <ul style="list-style-type: none"> <li>• service deployment (Service Provider is responsible for the service context data, only);</li> <li>• service provisioning (Service Provider is responsible for the service application and the service context data);</li> <li>• Billing.</li> </ul>
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>• process personal data of Service User fairly and lawfully;</li> <li>• safeguard that personal data of Service User is accurate and kept up to date;</li> <li>• safeguard that personal data that identifies individuals (Service User) is not be kept longer than necessary;</li> <li>• Do not build a business model on privacy related data.</li> </ul>

<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• define identification, authentication and authorisation levels coming with the service deployment and provisioning;</li> <li>• set up a circle of trust with:             <ul style="list-style-type: none"> <li>– Service Provider for deploying services</li> <li>– Service User for provisioning of services</li> </ul> </li> <li>• Protect data coming with service aggregation operations.</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>• guarantee validity of services (service applications) provisioned to the Service User;</li> <li>• guarantee validity of service context data at Service User for provisioned services;</li> <li>• Guarantee validity of digital map provisioned to the Service User.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>• guarantee the availability of the Host Management Centre;</li> <li>• guarantee the correct configuration of the Host (see Figure 3);</li> <li>• Guarantee the match between Host capabilities and requirements of the (to be) provisioned service applications.</li> </ul>

Role	Service User
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• Initialisation &amp; Activation</li> <li>• Subscription</li> <li>• Service Provisioning</li> <li>• Service Consumption</li> <li>• Service usage data generation &amp; submission</li> <li>• Service Payment</li> </ul>
<b>Generic responsibilities</b>	<p>The Service User is the client, who:</p> <ul style="list-style-type: none"> <li>• wants to consume a service (provided by a Service Operator);</li> <li>• Pays for this service consumption.</li> </ul>
<b>Responsibilities from</b>	<ul style="list-style-type: none"> <li>• be reticent with releasing personal data;</li> </ul>

<b>privacy perspective</b>	<ul style="list-style-type: none"> <li>• Protect security codes.</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• do not share identification, authentication and authorisation levels with third parties;</li> <li>• set up a circle of trust with:             <ul style="list-style-type: none"> <li>– Service Provider for consuming services;</li> <li>– Service Aggregator for selection and provisioning of services;</li> </ul> </li> <li>• Protect personal data coming with service operations.</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>• guard validity of subscription to Service Aggregator;</li> <li>• guard validity of subscription(s) to Service Provider(s) for safety related services;</li> <li>• Consume services in a safe way.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>• guard the correct operations of the Mobile Unit;</li> <li>• Guard the correct operations of the provisioned services.</li> </ul>

<b>Role</b>	<b>Payment Service Provider</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• Service Payment</li> </ul>
<b>Generic responsibilities</b>	A Payment Service Provider offers connections to various payment methods for e-payments transactions.
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>• process personal data fairly and lawfully;</li> <li>• safeguard that personal data is accurate and kept up to date;</li> <li>• safeguard that personal data that identifies individuals is not be kept longer than necessary;</li> <li>• Do not build a business model on privacy related data.</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• define identification, authentication and authorisation levels coming with the payment for service subscription and service usage</li> <li>• set up a circle of trust to settle payments with:</li> </ul>

	<ul style="list-style-type: none"> <li>– Service Provider</li> <li>– Service Aggregator</li> <li>– Service User</li> </ul> <ul style="list-style-type: none"> <li>• Protect data coming with service payments.</li> </ul>
<b>Responsibilities from safety perspective</b>	<external to CVIS context>
<b>Responsibilities from fault tolerance perspective</b>	<external to CVIS context>

Role	Certification Authority
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• Initialisation &amp; Activation</li> <li>• Subscription</li> <li>• Service Deployment</li> <li>• Service Provisioning</li> <li>• Service Consumption</li> <li>• Service Payment</li> </ul>
<b>Generic responsibilities</b>	The Certification Authority is a special authority, which provides authentication tokens (e.g., public-key certificates) and authentication verification services (e.g., certificate revocation lists, online certificate status protocol responders, etc.).
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>• operate a service without using personal data if possible</li> <li>• offer an anonymous service besides a personalised service</li> <li>• process personal data fairly and lawfully;</li> <li>• safeguard that personal data is accurate and kept up to date;</li> <li>• safeguard that personal data that identifies individuals is not be kept longer than necessary;</li> <li>• Do not build a business model on privacy related data.</li> </ul>
<b>Responsibilities from</b>	<ul style="list-style-type: none"> <li>• define identification, authentication and authorisation levels</li> </ul>

<b>security perspective</b>	<p>coming with credential provisioning ;</p> <ul style="list-style-type: none"> <li>enable actors (roles) within CVIS context to set up a circle of trust: Service Provider, Service Aggregator, Service User and Payment Centre</li> <li>Protect data coming with credential provisioning.</li> </ul>
<b>Responsibilities from safety perspective</b>	<external to CVIS context>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>safeguard the availability of the credential centre;</li> <li>manage an appropriate usage of the available resources in the credential centres;</li> <li>manage experienced latency in credential provisioning;</li> <li>Safeguard the integrity and correctness of the provisioned credentials.</li> <li>Safeguard the validity of the provisioned credentials.</li> </ul>

<b>Role</b>	<b>Service Point</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>Initialisation &amp; Activation</li> <li>Deactivation</li> </ul>
<b>Generic responsibilities</b>	<p>The Service Point is responsible for :</p> <ul style="list-style-type: none"> <li>retrofit installation and activation of the Mobile Unit;</li> <li>checking state &amp; status Mobile Unit;</li> <li>initialising and updating Contract Data;</li> <li>Initialising and updating Vehicle Data.</li> </ul>
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>process personal data fairly and lawfully;</li> <li>safeguard that personal data is accurate and kept up to date;</li> <li>safeguard that personal data that identifies individuals is not be kept longer than necessary;</li> <li>Do not build a business model on privacy related data.</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>define identification, authentication and authorisation levels coming with installing, initialising and activating of mobile</li> </ul>

	<ul style="list-style-type: none"> <li>units in vehicles;</li> <li>set up a circle of trust with Service Aggregator for initialising and activating mobile unit;</li> <li>Protect personal data.</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>guarantee a correct installation of nomadic mobile units in vehicles;</li> <li>Guarantee a correct installation initialising and activating of mobile units in vehicles.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>guarantee a correct installation of nomadic mobile units in vehicles;</li> <li>Guarantee a correct installation initialising and activating of mobile units in vehicles.</li> </ul>

<b>Role</b>	<b>Service Application Developer</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Generic responsibilities</b>	The Service Application Developer is responsible for developing the service applications on behalf of the Service Provider.
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>The developer is responsible for applying PETs (Privacy Enhancing Technology) and follow “Privacy By Design” design rules, thereby [19]: <ul style="list-style-type: none"> <li>guarantee that service application does not request for personal data that it does not need for correct operations;</li> <li>guarantee that service application processes personal data fairly and lawfully;</li> <li>safeguard that personal data that identifies individuals are not stored when not necessary for correct operation of service application;</li> </ul> </li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>verify identification, authentication and authorisation levels before operating the service;</li> <li>guarantee that service applications sets up a circle of trust before it start cooperating with other service applications;</li> <li>Protect personal data.</li> </ul>

<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>• Guarantee correct operations of service application.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>• embed functionality to monitor own operations;</li> <li>• embed functionality for graceful degradation of service application in case of fatal error;</li> <li>• Embed functionality for autonomous restart after service application finds itself in a deadlock.</li> </ul>

<b>Role</b>	<b>Vehicle Manufacturer</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Generic responsibilities</b>	<p>The Vehicle Manufacturer is responsible for:</p> <ul style="list-style-type: none"> <li>• the in-vehicle Mobile Unit itself;</li> <li>• In-line installation of the Mobile Unit (in-line or retrofit).</li> </ul>
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• Provide access to vehicle bus and use proper firewalling on it.</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>• Guarantee a correct production of mobile units for vehicles.</li> <li>• Guarantee a correct installation of mobile units in vehicles.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>• Guarantee a correct production of mobile units for vehicles.</li> <li>• Guarantee a correct installation of mobile units in vehicles.</li> </ul>

<b>Role</b>	<b>Nomadic Device Manufacturer</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Generic responsibilities</b>	<p>The Nomadic Device Manufacturer is responsible for the nomadic Mobile Unit itself.</p>
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>• none</li> </ul>

<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>Guarantee a correct production of nomadic mobile units.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>Guarantee a correct production of nomadic mobile units.</li> </ul>

<b>Role</b>	<b>Road Side Unit Manufacturer</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Generic responsibilities</b>	The Road Side Unit Manufacturer is responsible for the Road Side Unit itself.
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>Provide access to road side unit bus and use proper firewalling on it.</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>Guarantee a correct production of Road Side Units.</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>Guarantee a correct production of Road Side Units.</li> </ul>

<b>Role</b>	<b>Supplier – Software Components</b>
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Generic responsibilities</b>	The Supplier of Software Components is responsible for producing and delivering correct software components.
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>guarantee correct operations of ‘middleware’</li> <li>meet the requirements as depicted in the protection profile</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>guarantee correct operations of ‘middleware’</li> <li>meet the requirements as depicted in the protection profile</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>embed functionality to monitor and communicate own operations;</li> <li>embed functionality for graceful degradation of service application in case of fatal error;</li> </ul>

	<ul style="list-style-type: none"> <li>Embed functionality for autonomous restart after service application finds itself in a deadlock.</li> </ul>
--	--

Role	Supplier – Hardware Components
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Generic responsibilities</b>	The Supplier of Hardware Components is responsible for producing and delivering correct hardware components.
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>guarantee correct operations of the hardware</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>guarantee correct operations of the hardware</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>embed functionality to monitor and communicate own operations and resource usage;</li> <li>embed functionality to enable graceful degradation of service application in case of fatal error;</li> <li>Embed functionality for autonomous restart after the component finds itself in a deadlock.</li> </ul>

Role	Supplier – Communication Components
<b>Processes involved</b>	<ul style="list-style-type: none"> <li>none</li> </ul>
<b>Generic responsibilities</b>	The Supplier of communication components is responsible for delivering correct hardware components
<b>Responsibilities from privacy perspective</b>	<ul style="list-style-type: none"> <li>prevent unauthorised usage of MACs</li> </ul>
<b>Responsibilities from security perspective</b>	<ul style="list-style-type: none"> <li>guarantee correct operations of the communication components</li> </ul>
<b>Responsibilities from safety perspective</b>	<ul style="list-style-type: none"> <li>guarantee correct operations of the communication components</li> </ul>
<b>Responsibilities from fault tolerance perspective</b>	<ul style="list-style-type: none"> <li>embed functionality to monitor and communicate own operations and resource usage;</li> <li>embed functionality to enable graceful degradation of service</li> </ul>

	<p>application in case of fatal error;</p> <ul style="list-style-type: none"> <li>• Embed functionality for autonomous restart after the component finds itself in a deadlock.</li> </ul>
--	---

### 2.3. Processes coming with life cycle management

CVIS recognises a series of objects amongst which unit that are operational in the field: mobile units (in-vehicle and nomadic) and road side units.

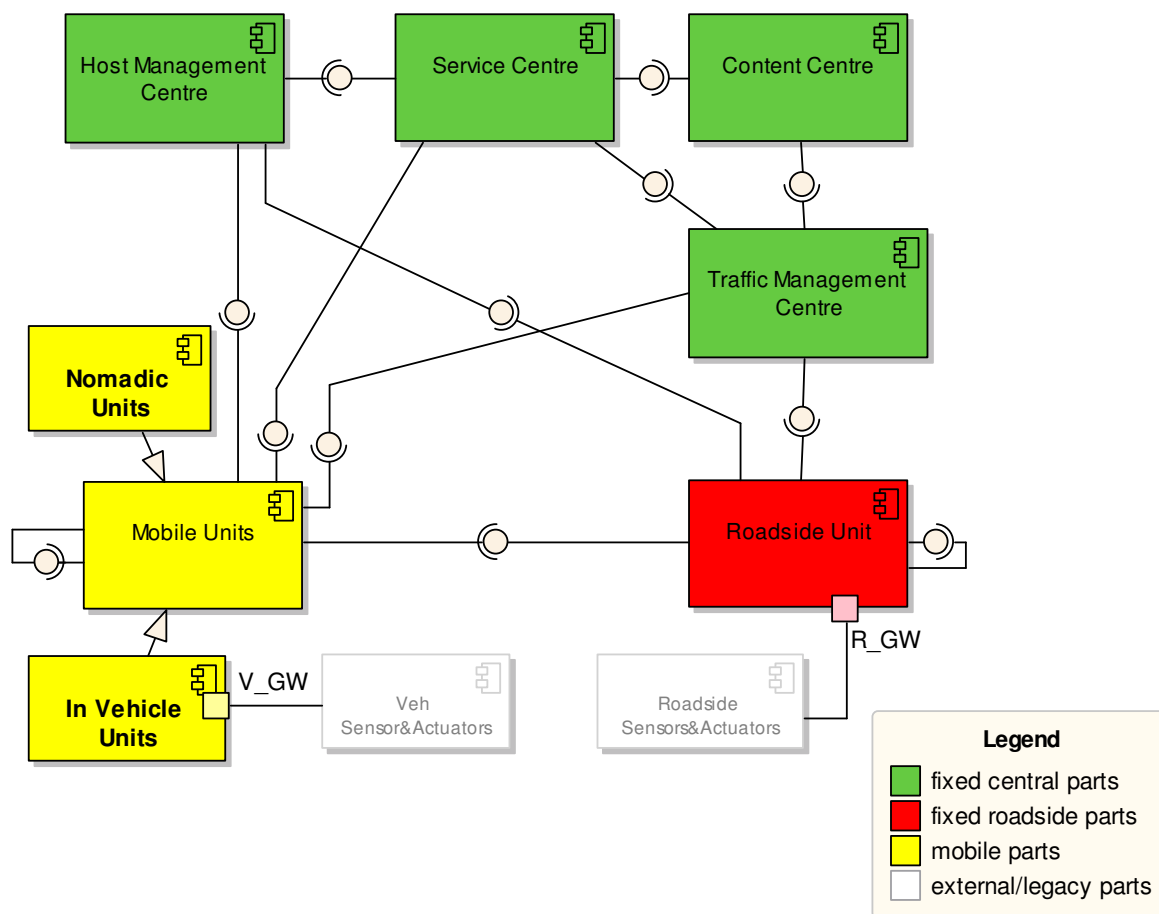


Figure 3. CVIS Objects (originally: entities) and their relations [10]

Depending on the services to be enabled by CVIS, CVIS needs too be able to guarantee certain ‘grades’. Guaranteeing ‘grades’ requires appropriate life cycle management and protection profiles. This paragraph focuses on the processes coming with life cycle management. The protection profile is dealt with in paragraph 3.2.

#### 2.3.1. Definition of ‘grades’ CVIS needs to guarantee

Design rule 2: define ‘grades’ for CVIS

- a. in the design of CVIS a unambiguous set of ‘grades’ need to be defined;
- b. the minimum set of grades for CVIS should be defined;
- c. Every CVIS component brought to the market should state which grades it guarantees.

Depend on the grades CVIS wants to offer, the mobile, roadside and central units need to guarantee a set of capabilities coming with specific grades. Figure 4 gives a first set-up of grades to be offered and capability to be guaranteed to assure these grades.

Main question for CVIS is: “what is the minimum set of ‘grades’ to be guaranteed by CVIS?”. With an eye on the services as designed and developed within CVIS (see Figure 4) the minimum set is: comfort grade, business grade, road safety grade. These grades put their claim on the life cycle management (see next paragraph) and the protection profile to be offered by CVIS (see paragraph 3.2).

**Figure 4. Set-up of capability guarantees coming with assuring specific grades**

Grade	Guaranteed capabilities coming with the grade	Examples of services enabled by CVIS
Comfort / efficiency of road usage grade	<ul style="list-style-type: none"> <li>• availability of the mobile unit;</li> <li>• appropriate communication bandwidth and latency in communications;</li> <li>• appropriate positioning;</li> <li>• Appropriate privacy protection.</li> </ul>	<p>CINT – cooperative traveller assistance (CTA)</p> <p>CURB – Cooperative network management</p> <p>CURB – Cooperative area routing and control</p> <p>CURB – Flexible bus lane allocation</p>
Business / commercial grade	<p>In addition:</p> <ul style="list-style-type: none"> <li>• end-to-end secure communications, secure environment for processing and storing of data, trustworthy units);</li> <li>• CPU capacity;</li> <li>• (Volatile and non-volatile) memory capacity.</li> </ul>	<p>CFF – Optimize delivery logistics and driver rest periods for transport companies</p>
Financial grade	In addition:	(payment for service

	<ul style="list-style-type: none"> <li>highly accurate and precise positioning;</li> <li>none of the services will interfere with or impact on the integrity or performance of the road safety related services (more generic: usage based charging services);</li> <li>Full privacy protection.</li> </ul>	<p>consumption)</p> <p>Optionally: usage based charging.</p>
Road safety grade	<p>In addition:</p> <ul style="list-style-type: none"> <li>bandwidth and maximum latency in communications despite the geographical location and situation;</li> <li>none of the services will interfere with or impact on the integrity or performance of the road safety related services</li> <li>none of the services will intervene with the in-vehicle systems</li> </ul>	<p>CFF – Increase the safety of dangerous goods transport</p> <p>CFF – Reduce, inside sensitive areas, vehicle breakdowns</p> <p>CINT – enhanced driver awareness (EDA)</p> <p>CURB – Cooperative local traffic control</p> <p>(CVIS does not support advanced driver assistance systems)</p>
Active safety grade	<ul style="list-style-type: none"> <li>none of the services will intervene with the in-vehicle systems in a non authorised and non-controlled way</li> </ul>	<p>Out of scope of CVIS, within the scope of SAFESPOT</p>

### 2.3.2. Processes for life cycle management

Design rule 3: define processes for life cycle management of CVIS objects

- a. define the processes and the involved roles in these processes;
- b. Define the responsibilities within these processes.

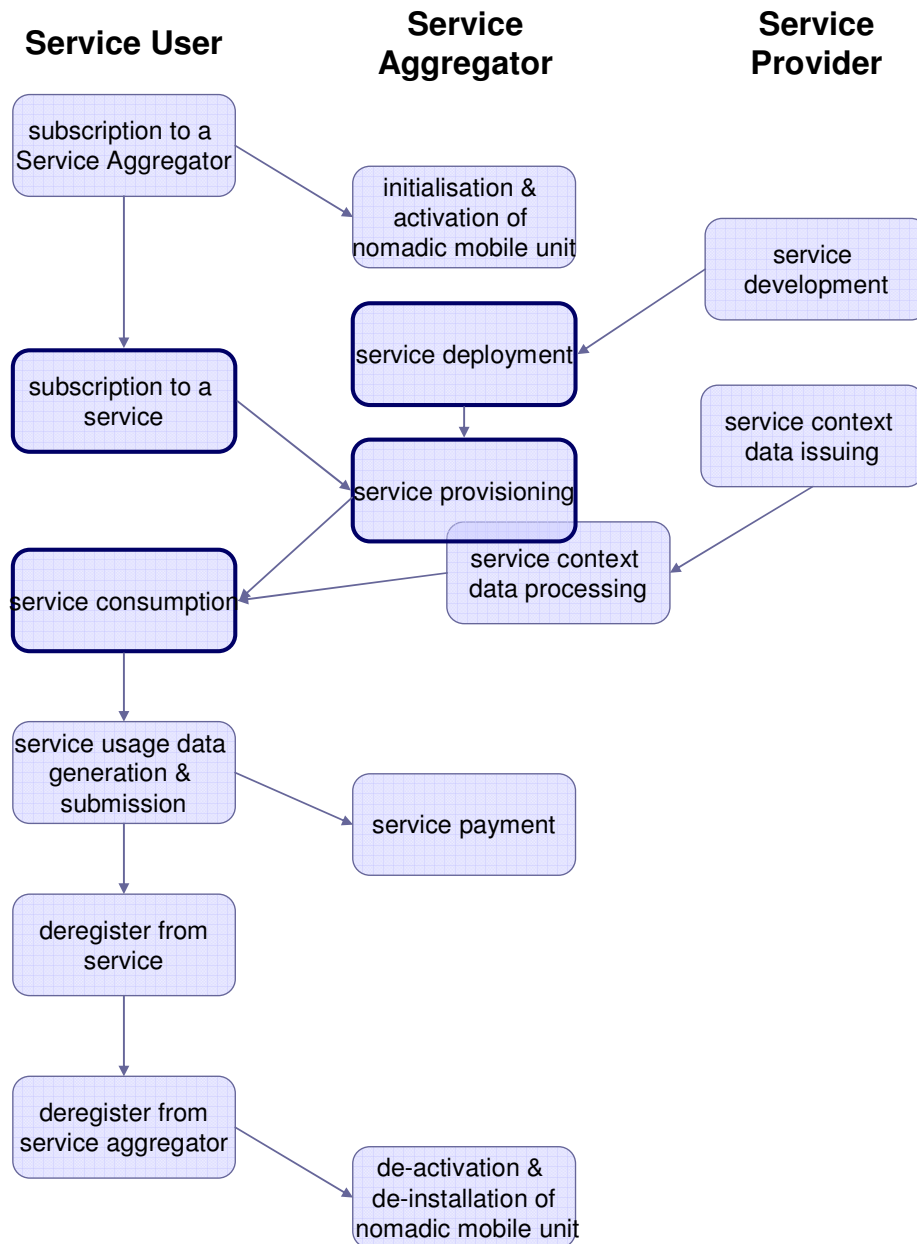
Business / commercial (, financial) and road safety grades require end-to-end security. One of the issues in guaranteeing end-to-end security is that the objects involved should be

trustworthy. For the mobile and roadside unit ‘in the field’ the implication is that the life cycle of these units needs to be controlled. Within the framework of open application management (FOAM) the emphasis is on the life cycle of service applications. For end-to-end security this life cycle needs to be stretched to the units themselves.

The full set of processes within life cycle management include at any rate:

1. Initialisation & Activation;
2. Subscription;
3. Service Deployment;
4. Service Provisioning;
5. Service Context Data Issuing;
6. Service Consumption;
7. Service Usage Data Generation & Submission;
8. Service Payment;
9. Compliance Check;
10. Deregister (cancel subscription);
11. Deactivation and de-installation.

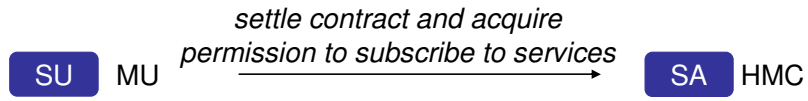
Figure 5 gives an illustration of the life cycle management of a nomadic mobile unit and service applications (and corresponding service context data) within CVIS. The consequence of the ‘grades’ of CVIS and the corresponding life cycle management is that there need to be clearly defined responsibilities regarding the processes within life cycle management. In the CVIS architecture the focus is on a subset of the processes only: service deployment, service provisioning, service subscription and service consumption. Furthermore references are made to GST’s service payment.



**Figure 5. Illustration of life cycle management of a nomadic mobile unit and service applications (and corresponding service context data) within CVIS**

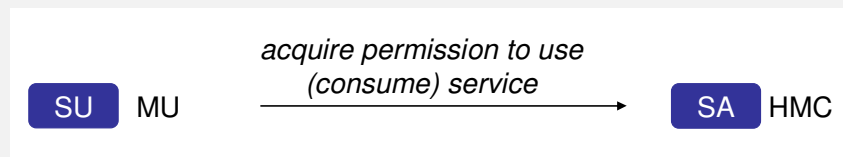
In the oncoming tables a sett-up is given for the processes within life cycle management and the responsibilities from privacy, security, safety and fault-tolerance perspective.

### 2.3.3. CVIS Process – Subscription to Service Aggregator

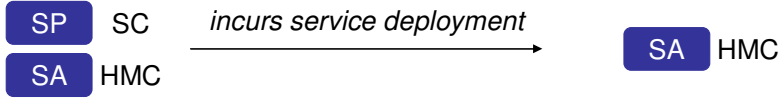
Process	Subscription to Service Aggregator
Objective	Service User gets ready to subscribe to services he or she wants to use (consume)
Functional Description	Subscribe to an Service Aggregator and initialize contract
Involved Roles	Service user, Service Aggregator
Involved Objects	Mobile Unit, Host Management Centre
Structuring rules	 <pre> graph LR     SU[SU] -- "settle contract and acquire permission to subscribe to services" --&gt; SA[SA]     MU[MU] --- SU     HMC[HMC] --- SA             </pre>
Initiating Event/Trigger	
Pre conditions	Potential Service User has no contract with Service Aggregator and has no possibilities to subscribe to services issued by Service Aggregator.
Post conditions	Service User has a contract with Service Aggregator and can now subscribe to services issued by Service Aggregator.
Sequence Steps	<ol style="list-style-type: none"> <li>1. Service Aggregator publishes the support it can give to potential Service Users, typical services it provisions and its contractual conditions.</li> <li>2. Service User subscribes to Service Aggregator by signing a contract.</li> <li>3. Equipping the Service user with a Mobile Unit: <ol style="list-style-type: none"> <li>i. Option 1: Service Aggregator organises that Service User gets a Mobile Unit fitted in its vehicle, initialised and activated;</li> <li>ii. Option 2: Service User already has a Mobile Unit; Service Aggregator organises that Service User gets its Mobile Unit verified, initialised and activated.</li> </ol> </li> </ol>
Related functions in CVIS	none
Responsibilities from privacy, security, safety and fault-tolerance perspective	Privacy: safeguard Personally Identifiable Information (PII) Security: set up ‘circle of trust’ before executing Service User’s Subscription to Service Aggregator Safety: nab.

Fault Tolerance: nab

### 2.3.4. CVIS Process – Subscription to a service

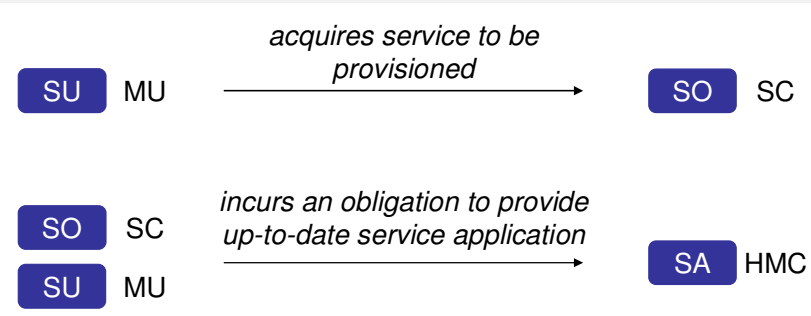
Process	Subscription to a service
Objective	Service User gets ready to use (consume) a specific service.
Functional Description	Subscribe to a service that can be provided by the Service Aggregator (the Service User has a contract with)
Involved Roles	Service user, Service Aggregator
Involved Objects	Mobile Unit, Host Management Centre
Structuring rules	 <pre> graph LR     SU[SU] --- MU[MU]     SA[SA] --- HMC[HMC]     MU -- "acquire permission to use (consume) service" --&gt; SA             </pre>
Initiating Event/Trigger	Service User selects a service he or she wants to subscribe to.
Pre conditions	Service user has no subscription to this specific service and is not in the position to use the service.
Post conditions	Service Aggregator starts the process of Service Provisioning.
Sequence Steps	<ol style="list-style-type: none"> <li>1. Service Aggregator issues a customised list of applicable services to all of its Service Users and the contractual conditions coming with each service.</li> <li>2. Service User selects one or more services and subscribes to this (these) service(s)</li> </ol>
Related functions in CVIS	CVIS – FOAM – service subscription
Responsibilities from privacy, security, safety and fault-tolerance perspective	<p>Privacy: safeguard Personally Identifiable Information (PII).</p> <p>Security: set up ‘circle of trust’ before executing Service User’s Subscription to a service.</p> <p>Safety: safety related services with no valid service subscription, should be taken out gracefully, that is with timely warnings, possibilities for the service user to extend the subscription period and never during service consumption.</p> <p>Fault Tolerance: monitor validity of the service subscription.</p>

### 2.3.5. CVIS Process – Service Deployment

Process	Service Deployment
Objective	Make the service available at the Service Aggregator to be provisioned to Service Users.
Functional Description	Deployment is the process of making a Service Application available at a Host Management Centre. This includes the packaging and transport of the application and all its components from the Service Centre to the Host Management Centre.
Involved Roles	Service Aggregator, Service Operator
Involved Objects	Host Management Centre, Service Centre
Structuring rules	 <pre> graph LR     SP[SP] --- SC[SC]     SA[SA] --- HMC[HMC]     SC -- "incurs service deployment" --&gt; HMC     </pre>
Initiating Event/Trigger	Service Deployment can be done by both the Service Operator and the Service Aggregator, depending on the business rules on who is responsible for the software applications.
Pre conditions	Service Aggregator is ready to get services deployed.
Post conditions	Service is deployed, accepted and stored by the Service Aggregator. The Service Aggregator might choose for a process of certification of the service application before accepting the application.
Sequence Steps	<ol style="list-style-type: none"> <li>1. Option 1: <ol style="list-style-type: none"> <li>i. Service Operator develops services applications needed to make a service operational;</li> <li>ii. Service Operator contacts Service Aggregator and set up an agreement to deploy a service</li> <li>iii. Service Operator deploys the service to the Service Aggregator</li> </ol> </li> <li>2. Option 2: <ol style="list-style-type: none"> <li>i. Service Operator provides Service Context Data to Service Aggregator;</li> <li>ii. Service Aggregator develops services applications needed to operationalise a service according to the</li> </ol> </li> </ol>

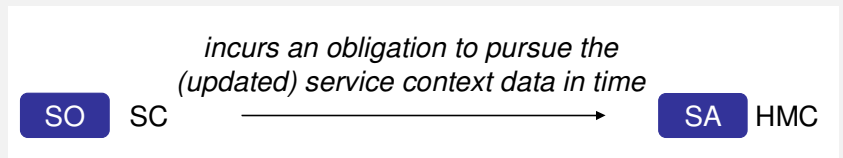
	<p>Service Context Data;</p> <p>iii. Service Aggregator deploys the service to itself.</p>
<p>Related functions in GST, CVIS, RCI</p>	<p>Option 1: CVIS – FOAM – service deployment</p> <p>Option 2: none (method comes from interoperable road user charging)</p>
<p>Responsibilities from privacy, security, safety and fault-tolerance perspective</p>	<p>Privacy: n.a.</p> <p>Security: set up ‘circle of trust’ before deploying services.</p> <p>Safety: keep deployed service up-to-date.</p> <p>Fault Tolerance: keep deployed service up-to-date.</p>

### 2.3.6. CVIS Process – Service Provisioning

Process	Service Provisioning
Objective	Provisioning is the process of enabling a Service Application for use on a Mobile Unit. This includes packaging, transport of the application and all of its components and activation of the application on the Mobile Unit.
Functional Description	
Involved Roles	Service User, Service Aggregator
Involved Objects	
Structuring rules	
Initiating Event/Trigger	Service User has selected and subscribed to a specific service in the process of ‘Subscription to a service’.
Pre conditions	Service is deployed by the Service Operator to the Service Aggregator Service User has subscribed to the Service Aggregator
Post conditions	Service User can start to use (consume) the provisioned service.
Sequence Steps	<ol style="list-style-type: none"> <li>1. Service User selects service;</li> <li>2. Service Aggregator confirms selections and start provisioning the corresponding service application.</li> </ol>
Related functions in CVIS	CVIS – FOAM – service provisioning CVIS – FOAM – reference to GST: Service payment (Purchasing)
Responsibilities from privacy, security, safety and fault-tolerance perspective	<p>Privacy: safeguard Personally Identifiable Information (PII).</p> <p>Security: set up ‘circle of trust’ before executing Service User’s Subscription to a service.</p> <p>Safety: safety related services with no valid service subscription, should be taken out gracefully, that is with timely warnings, possibilities for the service user to extend the subscription period and never during service consumption.</p>

Fault Tolerance: monitor validity of the service subscription.

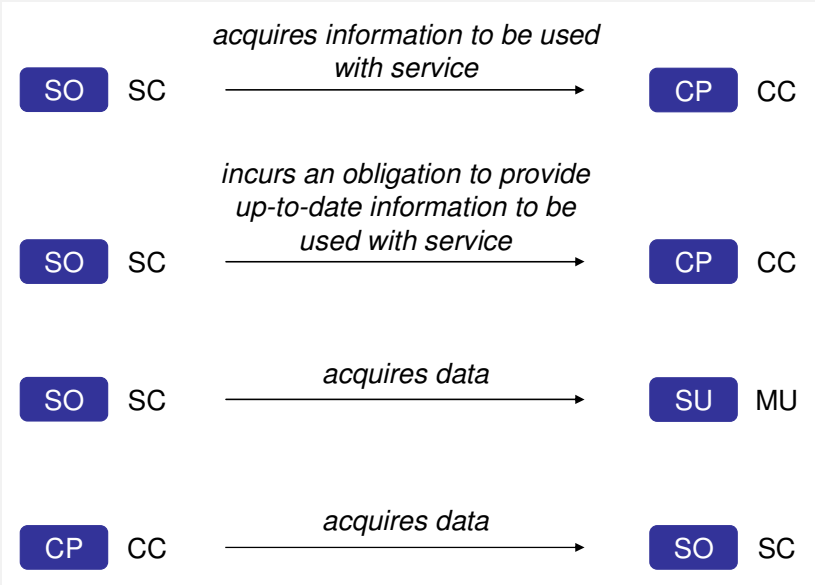
### 2.3.7. CVIS Process – Service Context Data Issuing

Process	5. Service Context Data Issuing
Objective	A specific service uses the actual Service Context Data
Functional Description	<p>Service can work with an operational scheme that is captured in Service Usage Data. The Service Usage Data comes with the provisioned service.</p> <p>In the life time of the service this Service Usage Data might change following changes in the operational scheme of the Service. These changes need to be pursued to the running Service Usage Data generations.</p>
Involved Roles	Service Operator, Service Aggregator
Involved Objects	Service Centre, Host Management Centre, Mobile Unit
Structuring rules	
Initiating Event/Trigger	Service User has subscribed to service and has got service provisioned by Service Aggregator.
Pre conditions	<p>Successful Service Provisioning</p> <p>A recent, up-to-date set of service context data is available</p>
Post conditions	Process of ‘Service Usage Data Generation & Submission’ work with actual, up-to-date set of Service Context Data
Sequence steps	<ol style="list-style-type: none"> <li>1. Service Operator issues its service context data a minimum period of time (to be determined) in advance of applicability.</li> <li>2. Service Aggregator ensures that it works with the actual service context data and changes the service context data when needed.</li> </ol>
Related functions in CVIS	none (method comes from interoperable road user charging)
Responsibilities from privacy, security, safety and fault-tolerance perspective	<p>Privacy: protect the relationship between Personally Identifiable Information (PII) and the actual location of the Mobile Unit, in case the actual position is used for timely updating of the Service Context Data in the Mobile Unit.</p> <p>Security: set up ‘circle of trust’ before issuing Service Context Data.</p> <p>Safety: keep issued service context data up-to-date; safety related services with no valid service context data, should be</p>

taken out gracefully, that is with timely warnings and before service consumption.

Fault Tolerance: keep deployed service context data up-to-date.

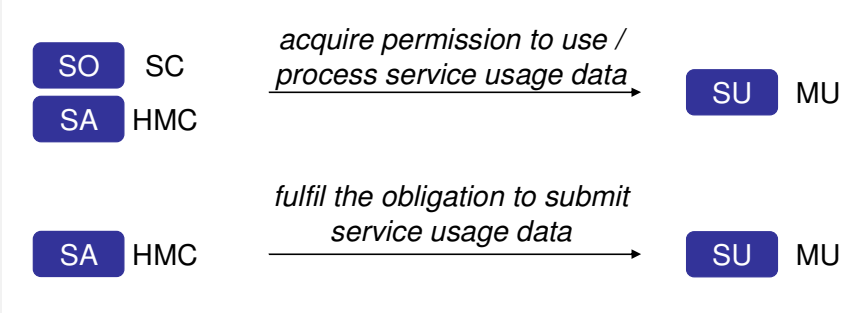
### 2.3.8. CVIS Process – Service Consumption

Process	6. Service Consumption
Objective	Service User uses the services it has subscribed to (and has paid / pays / will pay for).
Functional Description	Service User uses the services and exchanges information and data with the Service Operator.
Involved Roles	Service User, Service Operator, Content Provider
Involved Objects	Mobile Unit, Service Centre, Content Centre
Structuring rules	 <pre> sequenceDiagram     participant SC as SC     participant CC as CC     participant SU as SU     participant MU as MU     SC-&gt;&gt;CC: acquires information to be used with service     SC-&gt;&gt;CC: incurs an obligation to provide up-to-date information to be used with service     SC-&gt;&gt;MU: acquires data     CC-&gt;&gt;SC: acquires data     </pre>
Initiating Event/Trigger	Service User starts using the considered service.
Pre conditions	Service is provisioned as well as the Service Context Data.
Post conditions	Service User is ready with using the considered service
Sequence Steps	<ol style="list-style-type: none"> <li>Content Provider collects, verifies, processes and fuses data into relevant information;</li> <li>Service Operator derives information from Content Provider and feeds the Service User with all the information needed for service usage;</li> <li>Service User uses (consumes) the service and might (depending on the service) feed the Service Operator with data (e.g. floating car data);</li> <li>Service Operator forwards the received data to the Content Provider.</li> </ol>

Related functions in CVIS	CVIS – FOAM – service consumption
<p>Responsibilities from privacy, security, safety and fault-tolerance perspective</p>	<p>Privacy: protect the relationship between Personally Identifiable Information (PII) and the actual location of the Mobile Unit, in case the actual position is used for tuning the service consumption</p> <p>Security: verify both the source (origin) of the content used in service consumption and the transits used to bring the data from source to sink (see paragraph 4.3.2, information viewpoint).</p> <p>Safety: tune the control method within the service to the penetration degree, i.e. the number of service users on the road consuming the service.</p> <p>Fault Tolerance: monitor the process, verify the authorised consumption of the service and block unauthorised service consumption.</p>

### 2.3.9. CVIS Process – Service Usage Data Generation & Submission

Process	Service Usage Data Generation & Submission
Objective	Generate service usage data and submit this data to the role responsible for the billing (Service Operator or Service Aggregator).
Functional Description	<p>Generation of service usage data follows the detection of usage events, which can be based on:</p> <ul style="list-style-type: none"> <li>• Location, e.g.: <ul style="list-style-type: none"> <li>– entry of or exit from a zone or cordon;</li> <li>– passage of a specific point</li> <li>– usage of a specific road segment (e.g. a road section, bridge, tunnel or entrance to a city);</li> </ul> </li> <li>• Duration in Time: <ul style="list-style-type: none"> <li>– duration in time, e.g.: <ul style="list-style-type: none"> <li>▪ parking of a vehicle, driving within a zone,</li> <li>▪ consumption of driving and rest hours, or</li> <li>▪ using of a digital service (like real-time traffic information)</li> </ul> </li> <li>– physical duration, e.g.: <ul style="list-style-type: none"> <li>▪ distance driven</li> </ul> </li> </ul> </li> <li>• Download, e.g.: <ul style="list-style-type: none"> <li>– Provisioning of a service application</li> </ul> </li> <li>• Execution, e.g.: <ul style="list-style-type: none"> <li>– Each time the application is executed</li> </ul> </li> </ul>
Involved Roles	<p>Service User: has to support the detection of service usage events</p> <p>Service Aggregator:</p> <ul style="list-style-type: none"> <li>• (service payment option a) responsible for Service Usage Data generation;</li> <li>• (service payment option a &amp; b) responsible for Service Usage Data generation and submission to Service</li> </ul>

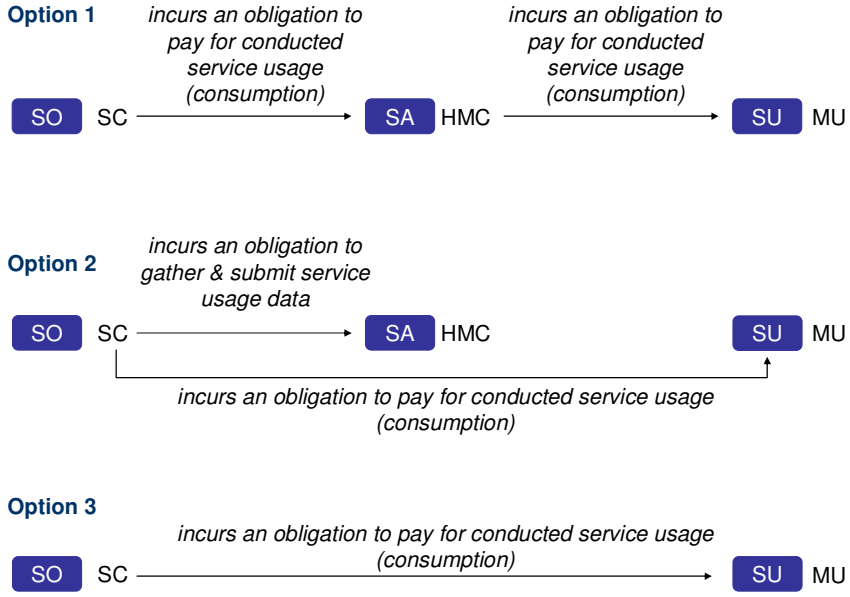
	<p>Operator.</p> <p>Service Operator:</p> <ul style="list-style-type: none"> <li>(service payment option c) responsible for Service Usage Data generation.</li> </ul>
Involved Objects	Mobile Unit, Host Management Centre, Service Centre
Structuring rules	
Initiating Event/Trigger	Service provisioning
Pre conditions	Service Context Data has been issued and processed by the Service Aggregator
Post conditions	All Service Usage Data has been submitted
Sequence steps	<ol style="list-style-type: none"> <li>Start process of Service Usage Data Generation &amp; Submission.</li> <li>Keep up to date the dynamic vehicle data which are needed for calculation of charge. <ol style="list-style-type: none"> <li>continuously determine actual vehicle state &amp; status relevant for the charging, depending on the requirements coming from the provisioned services.</li> </ol> </li> <li>Localise the Service User using the GNSS Position (via the service primitive, see paragraph <b>Error! Reference source not found.</b>).</li> <li>Option 1: on-the spot generation &amp; submission of Service Usage Data <ol style="list-style-type: none"> <li>Keep pre-defined set of Service Usage Data up-to-date given vehicle state&amp;status and actual location (for electronic fee collection: EG 11 data set, LSVa data set);</li> <li>Respond to requests from an RSE through pre-defined transactions (for electronic fee collection: EG 11 transaction, LSVa Transaction).</li> </ol> </li> <li>Option 2: on-line generation &amp; submission of Service usage data</li> </ol>

	<ul style="list-style-type: none"> <li>i. Service Usage Data Generation: <ul style="list-style-type: none"> <li>- detection of charge events;</li> <li>- the charging events may depend on additional conditions like moment in time (e.g. time of the day, day of the week), location state (e.g. being inside a zone), vehicle state (e.g. class, weight, number of axles);</li> <li>- generate the usage data that belongs to the detected charge event.</li> </ul> </li> <li>ii. Manage Counters of Service Usage Data <ul style="list-style-type: none"> <li>- add / sum the generated usage data to the corresponding counter;</li> <li>- manage the counters coming with the different services that are consumed by the Service User.</li> </ul> </li> <li>iii. Submit counted Service Usage Data <ul style="list-style-type: none"> <li>- submit the counted Service Usage Data;</li> <li>- submission may be triggered by: counter condition (the value of a counter being within a defined range), time (e.g. at fixed times, maximum time period), type of the last recorded charging event.</li> </ul> </li> </ul> <p>6. End of service usage:</p> <ul style="list-style-type: none"> <li>i. recognise when a Service User stopped using a specific Service;</li> <li>ii. stop process of Service Usage Data Generation &amp; Submission for this specific service.</li> </ul>
<p>Related functions in CVIS</p>	<p>CVIS – FOAM – reference to GST: Service Payment (Billing)</p>
<p>Responsibilities from privacy, security, safety and fault-tolerance perspective</p>	<p>Privacy: protect the relationship between Personally Identifiable Information (PII) and both the service consumed and the locations of the Mobile Unit of service consumption.</p> <p>Security: verify both the source (origin) of generated Service Usage Data and the transits used to bring the data from source to sink (see paragraph 4.3.2, information viewpoint).</p> <p>Safety: settle the rules for Service Usage Data Generation &amp; Submission in the service subscription contract and do not allow</p>

interference of the Service user with the process while driving.  
 Fault Tolerance: monitor the process while operational, monitor the process, verify the authorised Service Usage Data Generation & Submission and block unauthorised Service Usage Data Generation & Submission.


### 2.3.10. CVIS Process – Service Payment

Process	Service Payment
Objective	Based on the submitted Service Usage data the corresponding Charge Data need to be calculated and the billing process can be triggered. The billing should lead to payment for service usage by the Service user.
Functional Description	<p>Three billing options are distinguished:</p> <ul style="list-style-type: none"> <li>• Option a: Service Operator uses billing service of Service Aggregator (EETS approach). Service Aggregator is responsible for collecting service usage data, settles the payments with the Service Operator and invoices its Service User;</li> <li>• Option b: Service Aggregator is responsible for collecting service usage data, submits this data to the Service Operator, who invoices its Service Users;</li> <li>• Option c: Service Operator is responsible for collecting service usage data and invoices Service User itself.</li> </ul>
Involved Roles	Service user, Service Operator, Service Aggregator
Involved Objects	Host Management Centre, Service Centre, Payment Centre

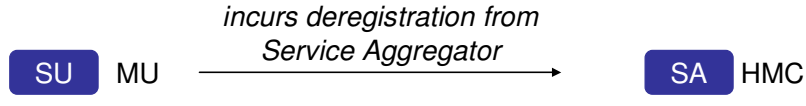
<p>Structuring rules</p>	 <p><b>Option 1</b> <i>incurs an obligation to pay for conducted service usage (consumption)</i></p> <p><b>Option 2</b> <i>incurs an obligation to gather &amp; submit service usage data</i></p> <p><b>Option 3</b> <i>incurs an obligation to pay for conducted service usage (consumption)</i></p>
<p>Initiating Event/Trigger</p>	<p>Service Usage data received.</p>
<p>Pre conditions</p>	<p>Service User consumes a provisioned service and Service Usage Data is gathered and submitted.</p>
<p>Post conditions</p>	<p>Service User has paid for its service usage (consumption).</p>
<p>Sequence Steps</p>	<ol style="list-style-type: none"> <li>1. Option 1             <ol style="list-style-type: none"> <li>i. Service Operator provides tariff information to Service Aggregator;</li> <li>ii. (Service Aggregator gathers the service Usage Data)</li> <li>iii. Service Aggregator calculates Charge Data (using Service Usage Data and tariff information);</li> <li>iv. Service Operator verifies calculated Charge Data;</li> <li>v. Service Aggregator pays for verified Charge Data;</li> <li>vi. Service Aggregator invoices its Service Users for the verified Charge Data (+ administration costs).</li> </ol> </li> <li>2. Option 2:             <ol style="list-style-type: none"> <li>i. (Service Aggregator gathers the service Usage Data and submits this data to Service Operator)</li> <li>ii. Service Operator calculates Charge Data (using Service Usage Data and tariff information);</li> </ol> </li> </ol>

	<p>iii. Service Operator invoices the Service Users for the verified Charge Data (+ administration costs).</p> <p>3. Option 3:</p> <p>i. (Service Operator gathers the service Usage Data)</p> <p>ii. Service Operator calculates Charge Data (using Service Usage Data and tariff information);</p> <p>iii. Service Operator invoices the Service Users for the verified Charge Data (+ administration costs).</p>
<p>Related functions in GST, CVIS, RCI</p>	<p>CVIS – FOAM – reference to GST: Service Payment (Pricing and Payment Transaction)</p>
<p>Responsibilities from privacy, security, safety and fault-tolerance perspective</p>	<p>Privacy: follow the privacy rules coming with financial transactions.</p> <p>Security: follow the security rules coming with financial transactions.</p> <p>Safety:</p> <ul style="list-style-type: none"> <li>• settle the rules for Service Payment in the service subscription contract and do not allow interference of the Service user with the process while driving;</li> <li>• safety related services with failed service payments, should be taken out gracefully, that is with timely warnings with opportunities of the service user to correct the failed payments and before service consumption.</li> </ul> <p>Fault Tolerance: follow the rules coming with financial transactions.</p>

### 2.3.11. CVIS Process – Deregister from Service

Process	Deregister from Service
Objective	<p>Service User deregisters from a service it has itself subscribed to previously.</p> <p>Service Aggregator deregisters Service User after proven fraud or arrears of payment.</p>
Functional Description	<p>Service User deregisters itself, after which it can not make use of the service anymore.</p> <p>Service Aggregator deregisters Service User, after which the Service User can not make use of the service anymore.</p>
Involved Roles	Service User, Service Aggregator, Service Operator
Involved Objects	Mobile Unit, Host Management Centre, Service Centre
Structuring rules	 <pre> graph LR     SU[ SU ] --- MU[ MU ]     MU -- "incurs deregistration from service" --&gt; SA[ SA ]     SA --- HMC[ HMC ]             </pre>
Initiating Event/Trigger	Service User wants to deregister from a service
Pre conditions	Service has been provisioned
Post conditions	Service is not provisioned any longer, subscription to service is cancelled.
Sequence Steps	<ol style="list-style-type: none"> <li>1. Service User deregisters from a service it has itself subscribed to previously;</li> <li>2. Service Aggregator confirms deregistration, annuls the provisioned service and informs the Service Operator</li> <li>3. Service processes the deregistration by the Service User.</li> </ol>
Related functions in CVIS	None
Responsibilities from privacy, security, safety and fault-tolerance perspective	<p>Privacy: safeguard Personally Identifiable Information (PII).</p> <p>Security: set up ‘circle of trust’ before executing Service User’s deregistration from a service.</p> <p>Safety: n.a.</p> <p>Fault Tolerance: n.a.</p>

### 2.3.12. CVIS Process – Deregister from Service Aggregator

Process	Deregister from Service Aggregator
Objective	Service User deregisters from a Service Aggregator it has itself subscribed to previously.
Functional Description	Service User deregisters itself, after which it can not make use of the service anymore.
Involved Roles	Service User, Service Aggregator
Involved Objects	Mobile Unit, Host Management Centre
Structuring rules	 <pre> graph LR     SU[SU] --- MU[MU]     MU -- "incurs deregistration from Service Aggregator" --&gt; SA[SA]     SA --- HMC[HMC] </pre>
Initiating Event/Trigger	Service User wants to deregister from a service
Pre conditions	Service User has subscribed to Service Aggregator
Post conditions	Subscription to Service Aggregator is cancelled, Service User can not subscribe to services provided by the Service Aggregator anymore.
Sequence Steps	<ol style="list-style-type: none"> <li>1. Service User deregisters from a Service Aggregator it has itself subscribed to previously;</li> <li>2. Service Aggregator confirms deregistration, annuls the contract with the Service User.</li> <li>3. Neutralising Mobile Unit: <ol style="list-style-type: none"> <li>i. Option 1. Service Aggregator organises that Mobile Unit will be de-installed form Service User’s vehicle;</li> <li>ii. Option 2. Service Aggregator erases all contractual data from Mobile Unit and deactivates Mobile Unit.</li> </ol> </li> </ol>
Related functions in CVIS	none
Responsibilities from privacy, security, safety and fault-tolerance perspective	<p>Privacy: safeguard Personally Identifiable Information (PII).</p> <p>Security: set up ‘circle of trust’ before executing Service User’s deregistration from a Service Aggregator.</p> <p>Safety: n.a.</p> <p>Fault Tolerance: n.a.</p>



### 3. CVIS from a functional perspective

#### 3.1. Platform, basic functions and functions

Design rule 4: define the basic function for the CVIS objects

CVIS objects come with a hardware platform, basic functions and functions (running on the platform and using the basis functions). Within the CVIS architecture the emphases is on the Host (Figure 6). From security, safety and fault tolerance perspective the unit itself (the hardware) needs to be incorporated too. The basic functions for both the unit and the host are depicted in Figure 7 and Figure 8.

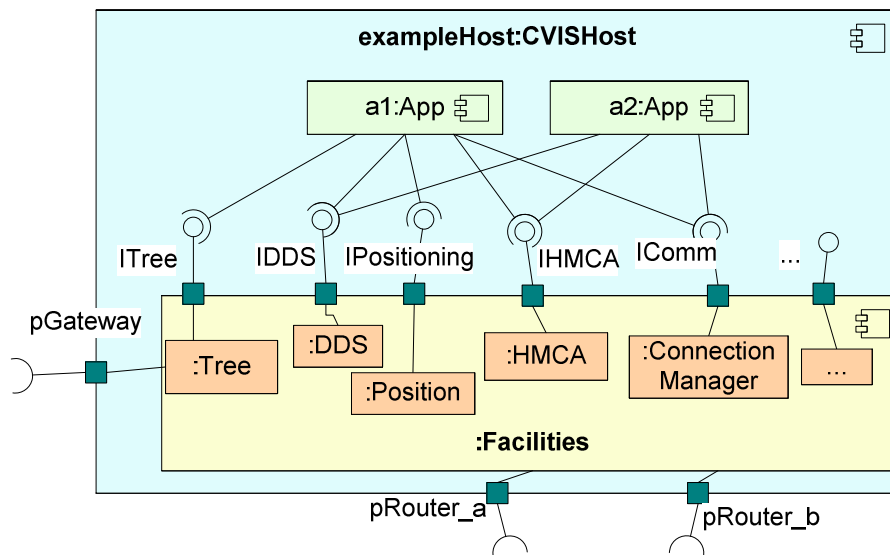


Figure 6. Illustration of the CVIS Host [11]

Figure 7. Service primitives coming with a CVIS Unit (mobile, roadside or central)

Basic Function	Basic sub-function (= service primitive)	Meaning
<none>	volatile data storage	Provide volatile data storage for executed service applications and data (especially dynamic data, see paragraph 4.5.1) used in service consumption.
	non volatile data storage	Provide non volatile data storage for provisioned service applications and

		data (especially invariant and static data, see paragraph 4.5.1) used in service consumption.
	vehicle status & actuation	Monitor the vehicle status & actuation and generate a signal that gives expression to this status & actuation.
	resource management	Monitor resource consumption and management the dynamic usage of the resources.

The basic functions of CVIS can be detailed in a set of required Service Primitives.

**Figure 8. Service primitives coming with a CVIS Host**

Basic Function	Basic sub-function (= service primitive)	Meaning
Framework for Open Application Management (FOAM)	service life cycle management	Provide the functionality to provision service applications from the HMC to the hosts and keep these service applications up-to-date.  Provide the functionality to provision the elements from the Service Context Data from the HMC to the hosts that should be available on the hosts and keep this data set up-to-date.
	inter-service communication	Provide the functionality to service applications to: <ul style="list-style-type: none"> <li>• subscribe themselves to the processed data of another service applications;</li> <li>• Provide subscribed service applications with processed data.</li> </ul>
	I/O to other devices connected to the platform	Provide a local device tree via which service applications can subscribe themselves to other devices (sensors and actuators), retrieve and interpret data from sensors and forward data to actuators.
Positioning, Mapping and Location Reference (POMA)	positioning	Provide continuously a position-time-velocity (PTV) fix, including confidence interval.

	map handling	Provide the functionality to map the PTV to the digital map following the requirements of a service application.
Communications Service compliant with CALM (COMM)	off-platform communication	Provide seamless off-platform communication to service applications.
Cooperative Monitoring (COMO)	verify signals	Provide the functionality to verify the source of signals, interrogations and data, as well as the transits via which and signals, interrogations and data are received (see subparagraph 4.3.2, information viewpoint).
	world model	Provide a ‘word model’ to verify the validity of the received signals and data (see paragraph 3.6).

### 3.2. Protection Profile

Design rule 5: define the protection profile for CVIS

- a. following the defined grades for CVIS (see design rule 2);
- b. Define the protection profile per basic function.

A Protection Profile provides an (implementation independent) specification of the grades as guaranteed / offered by CVIS. In the following subparagraphs a set-up is given for the protection profile of CVIS.

#### 3.2.1. Protection by the Mobile Unit

The mobile unit comes with physical protection for non-authorized opening of the unit by an intruder.

#### 3.2.2. Protection by the operating system and runtime environment

The Mobile unit comes with a runtime environment (e.g. a Java runtime environment) that provides built-in security features such as platform security, cryptography, authentication and authorization.

#### 3.2.3. Protection by the Service Primitives

For every basic sub function / service primitive the protection offered to the services is

defined in the tables below.

**Figure 9. Protection on volatile data storage**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
host code and data	host the running service applications	resource manager controls the dynamic memory allocation
	host the specific Service Context Data for usage by running service application(s)	
	host the specific parts of the digital map data for usage by running service application(s)	
	host the recently registered service usage data before storage	

**Figure 10. Protection on non volatile data storage**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Store Secure	store Service Context Data available for usage by service application	Service Context Data cannot be manipulated by an intruder
	store service usage data	service usage data cannot be manipulated by an intruder
	store service applications	service applications cannot be manipulated by an intruder

**Figure 11. Protection on positioning**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Import	optionally: Import EGNOS data from IP-based server (EDAS)	transfer of EGNOS data is done via a secured communication link (thin client)
	odometer signal (via I/O to other devices connected to the platform)	Service Point takes care that Mobile unit is installed in vehicle

		correctly and wheel rotations are received correctly by Mobile Unit
	Gyro or Accelerometer data	gyro or accelerometer is intrinsic component of Mobile Unit
Actions	calculate current PVT (position, velocity, time)	PVT is calculated using input data meeting the accuracy requirements for EETS and advanced driver assistance systems (ADAS)
	hash calculated PVT	PVT is hashed to protect it from manipulation by third party
Output	provide current PVT to listening service applications	Deliver actual NMEA string
Register	register hashed PVTs	(see service primitive 'Non volatile data storage')
Store&Secure	store and secure hashed PVTs	(see service primitive 'Non volatile data storage')

**Figure 12. Protection on off-platform communication**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Import	request for communication from Service Application	
Actions	provide communication interface by connecting Service Application to communication channel	take the available QoS into account when providing a communication interface to a Service Application
	Connect to a Service Centre by means of a standardized Connection interface.	support priority management in communication
		provide secure end-to-end communications

**Figure 13. Protection on I/O to other devices connected to the platform**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Import	Request for communication from	

	Service Application	
Actions	Provide local device tree interface for an application that will allow to access device related status information like odometersignal.	Service Point takes care that Mobile unit is installed in vehicle correctly and data from in-vehicle sensors is hard real-time written in local device tree

**Figure 14. Protection on mapping**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Import	receive set of calculated PVT, with a minimum of one PVT	transfer of PTVs from positioning to mapping service primitive is done in a in secure environment (thick client), or via a secured communication link (thin client)
Actions	map match calculated PVT and determine current road segment	map data is correct and up to date
Output	provide current road segment to listening service applications	map matching is accurate meeting the accuracy requirements for EETS and ADAS
Register	register road segments	(see service primitive ‘Non volatile data storage’)
Store&Secure	store and secure road segments	(see service primitive ‘Non volatile data storage’)

**Figure 15. Protection on vehicle status & actuation**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Actions	receive signal that vehicle is active	Service Point takes care that Mobile unit is installed in vehicle correctly and receives ignition signal correctly
Output	activation trigger for service applications to start up once vehicle has become active	

**Figure 16. Protection on service life cycle management**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Actions	provide remote management of the configuration of the Mobile Unit (setting configuration data)	Remote Manager takes care of the service life cycle management remotely in a secure manner safeguarding the privacy of the Service User and commercial interests of the Service Provider
	Provide service application life-cycle management (installing, starting, updating, stopping and uninstalling service applications).	
	Provide security management, by setting the permissions for bundles and handling Service User data on the Mobile Unit (Host).	
	Provide fault management by running diagnostics and correcting problems.	
	enable service provisioning	
	Enable accounting management, i.e. collecting accounting information and preparing them for financial transactions	

**Figure 17. Protection on financial transactions**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Actions	grant (or deny) the Service Platform the ability to perform a specific service for a specific Service User	Billing Agent takes care of financial transactions in a secure manner safeguarding the privacy of the Service User and commercial interests of the Service Provider
	store and/or forward charging events coming from the Service	
	Invoice the Service User	
	settles bills with the Service Provider	

**Figure 18. Protection on inter-service communication**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Actions	enable service applications to register as listener for receiving events from another service application running on the Host	FOAMs White board model prevents direct interaction between service applications
	Enable service applications to send events to other service applications registered as listener	

**Figure 19. Protection on HMI**

Issue	To do	CVIS Quality statement to ensure privacy, security, safety and fault tolerance
Actions	prioritise service applications	HMI manager (broker) controls the dynamic access to the display of the mobile unit
	prioritise signals, input request and data coming from service applications	

### 3.3. Capability profile

#### 3.3.1. Safety Instrumented System

Design rule 6: design the Safety instrumented system that should be embedded within the CVIS design.

So far the emphasis in the CVIS designs is on the functionality of cooperative services and systems [EN/IEC 61511]. However, CVIS should be looked upon as a ‘safety critical system’, whose failure or malfunction may result in: death or serious injury to people, or loss or severe damage to vehicles or environmental harm. A ‘safety critical system’ distinguishes two type of process control: (i) regular control focussed on the functionality of the system and (ii) safety control focussed on the correct state and status of the system. Safety control monitors the actual state and status of the processes and performs specified functions to achieve or maintain a safe state of the process when unacceptable or dangerous process conditions are detected.

Safety control implies that the CVIS object should be instrumented with applications and even components that are separate and independent from the regular control applications and components.

The specified functions, or safety instrumented functions (SIF) are implemented as part of an overall risk reduction strategy which is intended to reduce the likelihood of identified hazardous events involving a catastrophic release. The safe state is a state of the process operation where the hazardous event cannot occur. The safe state should be achieved within one-half of the process safety time. Most SIF are focused on preventing catastrophic incidents.

Safety control requires:

- sensors capable of detecting abnormal operating conditions, such as temperatures that are too high or low, for a proper functioning of the CVIS objects;
- monitoring functions to monitor the factual usage of the resources (basic function / service primitives) of the CVIS objects;
- A logic solver to receive the sensor and monitoring input signal(s), make appropriate decisions based on the nature of the signal(s), and change its outputs according to user-defined logic.
- Actuators that can take action on the process, based on the logic solver output(s) results, to bring the concerned CVIS object to a safe state.

These additional sensors, functions and actuators should be embedded in the CVIS designs required integrity and reliability.

### 3.3.2. Resource management

Design rule 7: define a method for resource management within the CVIS objects:

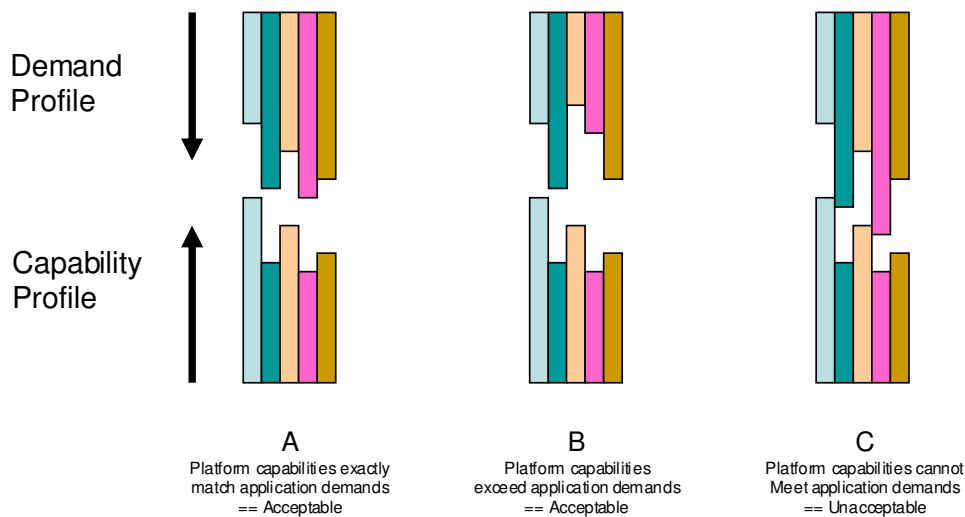
- a. define the capability profile;
- b. define a method (including matching criteria) to fit services to the capability profile(s);
- c. Define the decision rules to be followed when the selected service scan not is fit to the actual capability profile of an object.

Resource management is an example of a function of the Safety Instrumented System for CVIS. Resource management can be based on capability profiles. In this subparagraph a set-up is given for working with a capability profile within CVIS.

### Fitting Services to Capability Profiles

Simply defining the functionality and interfaces of Service Primitives is not sufficient as Derived Services need to make assumptions about non-functional aspects of those service primitives in order to be appropriately robust for their application domain. For example, it may be appropriate that positioning is only available 98% of the time for a SatNav application, but that would be unacceptably low for an automated steering application.

Individual services can only operate safely and appropriately on a given platform if the non-functional capabilities of that platform (abstracted via the Service Primitives) meet certain invariants that the Derived Services demand. For example, it may be essential that a service has access to a communication channel which is encrypted and mutually authenticated, or perhaps it demands a certain amount of computational resource. Whatever a services' Demand Profile (DP) these capabilities have to be met by the underlying platform, which has a Capability Profile (CP). This is shown Figure 8.



**Figure 20. Matching Application demands and Platform Capabilities [14]**

Different services may be presented with different CPs by the underlying platform depending upon their class and other services that have already consumed available resources from it. It is assumed that if a service has established a supply contract then that contract will be honoured no matter what additional services are hosted by the supplying entity.

Note that these concepts of demand and capability profiles are not limited to the relationship between the service and the supporting platform, but also apply to the non-functional aspects of the relationship between individual services (since these are largely indistinguishable from the platform itself anyway). This means that, for the introduction of a service into a platform which needs to interwork with several other services (e.g. to obtain positioning information, or for payment support) the next service will have several Partial Demand Profiles which together form the overall demand profile for the service...a platform cannot advertise the fact that it has 200 ARM7MIPS of processing power available if it's already hosting three applications needing 30 ARM7MIPS each, for example!

**Definition of matching criteria**

For the matching of service related requirements (demand) and platform capabilities, four classes can be used [14]:

- **Nominal:** Also known as categorical. For this class there is no relationship between the distinct values in the series. Nominal measures offer labels for certain characteristics and are mutually exclusive.
- **Ordinal:** The numbers assigned impose a rank order, but differences have no meaning – For example a pain scale 1-10. A score of 8 implies more pain than a score of 5, but it may not be an additional 3’s work of pain. A better example might be positions in a race.
- **Interval:** A difference in the level of an attribute has meaning, but the ratio does not. An example would be temperature measured on a Celsius scale; 20°C is not twice as hot at 10°C.
- **Ratio:** All of the properties of an Interval measure, with a clear definition of zero such that multiplication is valid. An example would be temperature measured on a Kelvin scale.

Variables of interest in Demand Profiles can be defined in terms of these four classes. The classification directly affects the comparisons that can be made when assessing suitability of a service.

Variable type	Short form	Applicable Comparisons
Nominal	N	==, !=
Ordinal	O	<, >, <=, >=, ==, !=, ranked preference
Interval	I	<, >, <=, >=, ==, !=, ranked preference, linear difference
Ratio	R	<, >, <=, >=, ==, !=, ranked preference, linear difference

The type of variable becomes important when matching Demand and Capability profiles. A few examples include:

- Nominal variables can only be tested for equality and inequality; does this platform support EBDIC?
- Ordinal values can be checked for ‘better than’ conditions; is the link at least encrypted, and preferably mutually authenticated?
- Interval and Ratio values can be used to ensure ‘best fit’ across a set of services; Service X needs 23 MIPS and Service Y needs 12 MIPS – what is the best way to deploy them to retain maximum headroom on the platforms?

Variables may be specified in a Demand Profile as Preferred or Mandatory. These define the

level of fit required. If it is not specified in the profile then it is a don't care case. If a variable is specified in a Demand Profile and is not specified in the Capability Profile then that is an error condition and the service cannot be hosted on the platform (although different platforms may chose another method to resolve the error).

A single variable may be specified more than once in a Demand Profile. For example a service may demand at least 30 ARM7MIPS of CPU resource, but prefer to have 50 ARM7MIPS available.

### *Variables in Demand and Capability Profiles*

The set of variables to be used for matching, and their types, is for definition during the design phase and for discussion amongst the partners. A few suggestions include:

Variable Name	Type	Description
Availability	R	Percentage of platform operation time for which the service is available.
commsSecurity	O	Level of security on the communications link. An enum of {none=0, obfuscate, encrypt}
ARM7Mips	R	Number of ARM 7 MIPS of processing power available (note that other processors may be used, but this gives a common measure).
CALM M5	O	Is a CALM M5 link available?
UIType	N	User Interface type. An enum of {none=0, LED, NumericLCD, AlphaLCD, MonoGraphic, ColourGraphic}

The set of variables, their definitions, units and scaling factors will all be defined during the design process. Standardised variables will, as for Service Primitives, be identified as such.

### **3.4. Safety versus Interoperability**

Design rule 8: support road safety by enabling interoperability via Service Context Data

Road safety is supported by providing relevant services all over Europe, for instance the (CINT-EDA) enhanced river assistance service. Therefore, CVIS must be able to enable services on a European level, i.e. not restricted to the borders of specific service providers. Since cooperative services, however, by nature require cooperation with the local service provider. Such cooperation might require tuning of the service application to the local context.

Such tuning can be done via Service Context Data. The nature of Service Context Data is described in paragraph 4.4 (information viewpoint).

On a functional level, tuning of the service application to the local context using the Service Context Data should be part of the service life cycle management. Service life cycle management consists of:

- service deployment;
- service provisioning;
- Service Context Data issuing:
  - issuing of the Service Context Data by the Service Provider;
  - Tuning of the service application by the Service Aggregator using the issued Service Context Data.

The related processes are described in paragraph 2.3.

### **3.5. Security**

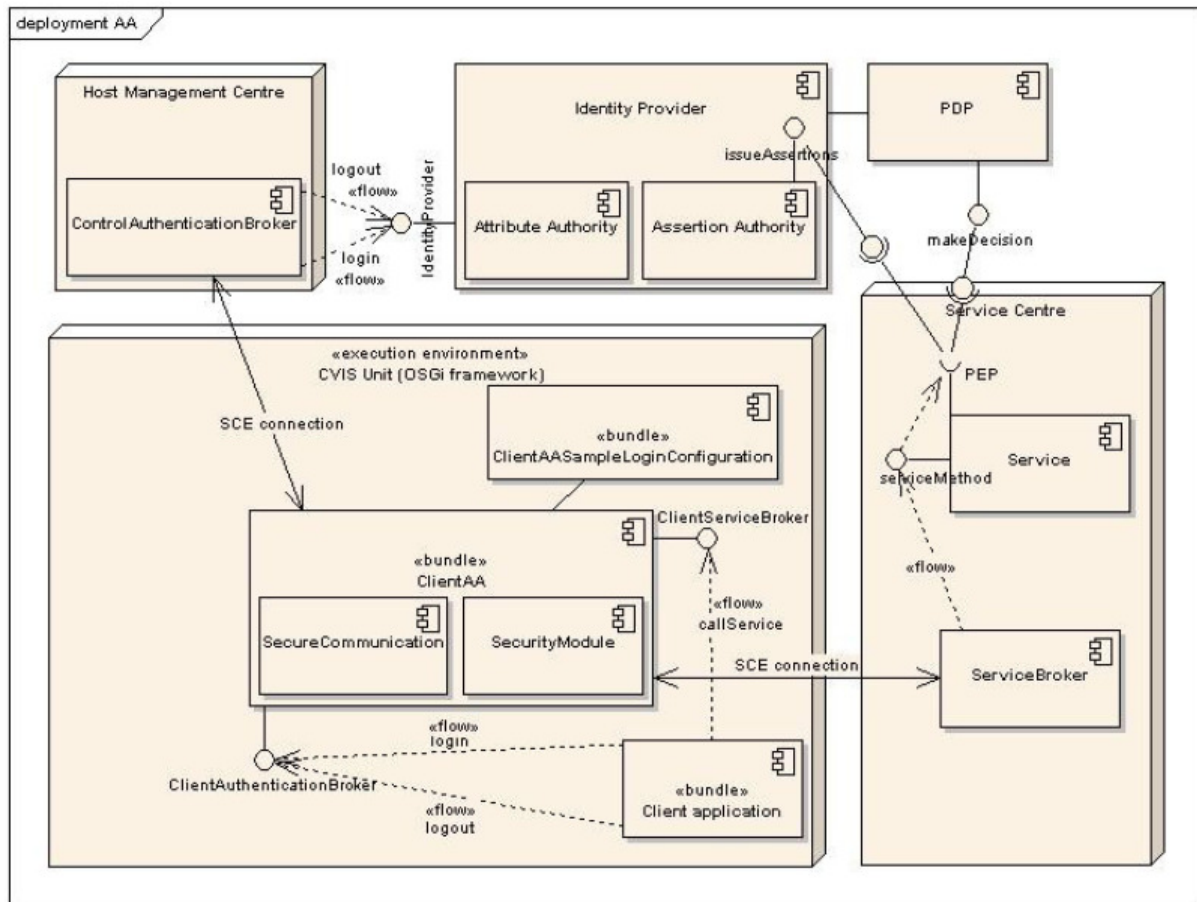
Design rule 9: Embed the relevant security elements in the design of CVIS

Security within CVIS requires identification of tasks and responsibilities for security roles in the system.

#### ***Authorisation and authentication broker***

Authorisation and authentication broker is responsible for authentication of users and service providers in the system. Authentication broker authenticates users and entities of the system before servicing. Authorisation broker authorises all service requests.

Authorisation and authentication broker is part of the Service Operator, Service Usage Data Collector, Billing Centre, Host Management and Mobile Unit.



**Figure 21. Positioning of the authorisation and authentication broker in the FOAM architecture [3]**

### *Secure Execution environment*

The secure execution environment controls integrity of application and provides isolated dedicated software environment for each application or application provider. From security point of view the main responsibility of execution environment is to protect applications against undesirable interactions and control applications integrity. Isolation between applications is assured by providing dedicated execution environment with limited and controlled resources and capabilities. Application integrity protection is to avoid unauthorised application modifications or system integrity loss.

The secure execution environment is part of the Mobile Unit.

### *Secure communication engine*

Secure communication engine guarantees secure communication between system entities.

The secure execution environment is part of the Service Operator, Service Usage Data Collector, Billing Centre, Host Management and Mobile Unit.

### *Service application*

Service application is responsible for processing information accordingly to agreement

committed with end user, service aggregator and law. Application before deployment on production environment must be verified for compliance with regulations and system requirements.

***Key management***

The Credential Provider (Identity Provider) is responsible for key and certificate management. Good protection of encryption key is paramount for forceful data protection. It is suggested to use hardware modules for key storage.

***Secure Module***

The Security Module is used by the Secure Communication, and the Authentication and Authorization subsystems. The two main uses of the Security Module are:

- to provide secure persistent storage of keys, certificates and data;
- To provide cryptographic functions (like signing and verification, encryption and decryption, message digests etc.).

Its function is to store and retrieve keys, certificates and data in a secure and persistent way.

<b>Name</b>	<b>Description</b>
Store keys	Stores different types of keys (public, private etc.) in a secure and persistent way.
Store certificates	Stores certificates in a secure and persistent way.
Store data	Stores arbitrary data in a secure and persistent way.

Another function is to provide cryptographic functions

<b>Name</b>	<b>Description</b>
Sign and verify data	Signing and verification are necessary if data needs authentication. Private keys are used for signing, public keys for verifying data.
Encrypt and decrypt data	Encryption is required in the case of data classified as confidential.
Create message digest	Produces a hash
Generate MAC	A message authentication code is a code produced with a secret key to protect the integrity and authenticity of a message.

The Security Module can be decomposed into the following structurally significant components and their interfaces:

- Security Module;
- Secure Storage;
- Crypto Engine;
- SecurityModuleFactory;

- Provider.

The Security Module interface is at the core of the Security Module’s architecture. It provides methods for secure storage and retrieval of data, and cryptographic functions. Therefore, the Security Module interface is derived from the Secure Storage and the Crypto Engine interfaces. Secure Storage is the interface that defines the methods needed for the secure storage and retrieval of keys, certificates and data. Crypto Engine’s methods are for signing, verification, encryption etc.

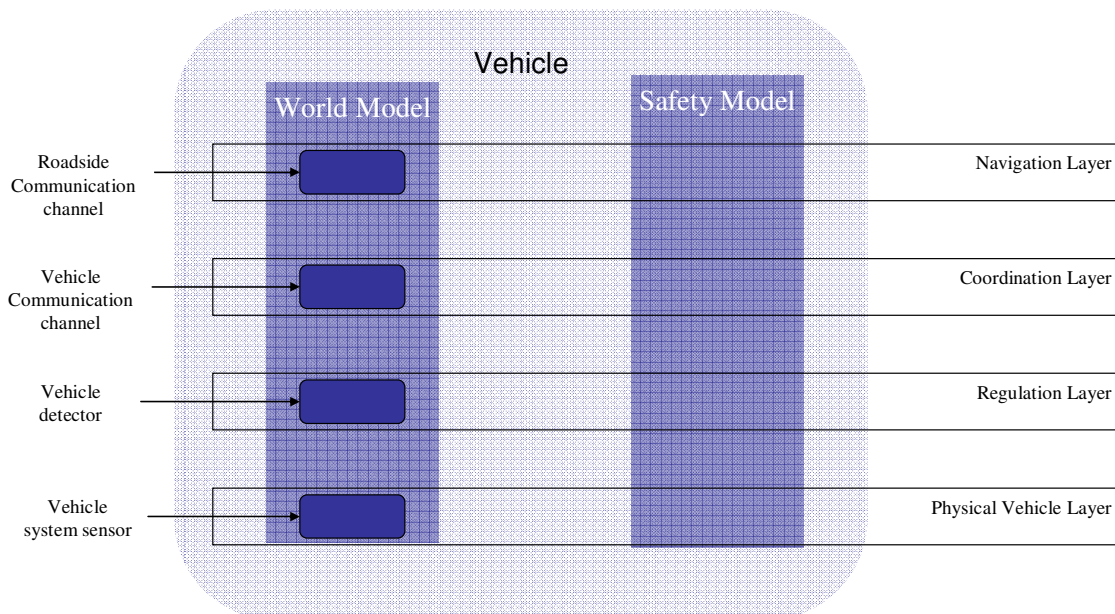
The SecurityModuleFactory interface is the factory used for creating Security Module instances.

The Provider interface should be implemented by service providers who want to create their own secure storage and crypto engine implementations.

### 3.6. World model

Design rule 10: Verification of the integrity and correctness of received data in the cooperation on continuous basis

Verification of the integrity and correctness of received data can be done in a ‘world model’. Such a world model should be build within the mobile unit (vehicle), road side unit and centre, where all the incoming data on the different levels are verified and fused into one consistent model of the outside (and inside) world (see illustration in Figure 22).



**Figure 22. World model to come to a coherent dataset for location awareness**

## 4. CVIS from an information perspective

### 4.1. Criticality of information

Cooperation in the end comes down to communication and exchanging information. The criticality of information handling for correct operations of CVIS follows from:

- the states and state transitions of objects;
- the cooperation between objects;
- the tuning of services to their current context;
- the storage and handling of information;

### 4.2. States and State Transitions of objects

Design rule 11: Define the states of the objects and the conditions under which a state transition takes places.

Data provisioned to or cleared from Mobile or Road Side Unit incurs a state transition. In Figure 9 the state transitions for a Mobile Unit are outlined.

**Table 1. State transitions of the Mobile Unit**

Process	Data set	State Transition
Initialisation & Activation	Service Aggregator Contract Data Vehicle	From non-active Mobile Unit becomes active
Subscription	Service Operator Contract Data	Mobile Unit is ready to get service application provisioned
Service Context Data Issuing	Service Context Data	Mobile Unit is ready to execute service application and have Service User consume the corresponding service
Service Consumption	Content (data)	Service consumption takes place
Service Usage Data Generation & Submission	Service Usage Data	Service consumption takes place
Compliance Check	Control data	Mobile Unit is compliant → service consumption continues Mobile Unit is non-compliant → service consumption is blocked
Deregister (cancel subscription)	Service Operator Contract Data	Mobile Unit can not execute service application any longer. Service provisioning and consumption is blocked.

Deactivation	Service Aggregator Contract Data Vehicle data	From active Mobile Unit becomes non-active
--------------	--	--

### 4.3. Cooperation between objects

Cooperation between objects comes with:

- an appropriate understanding of the syntax and semantics of the information;
- overview over the sources, sinks and transits of the information;
- privacy in cooperation;

For these issues design rules are derived and explained.

#### 4.3.1. Syntax and semantics

Design rule 12: Shared information should be enriched with meta-information that supports service applications in a correct understanding of the retrieved information.

Within a cooperative environment like CVIS information should be understandable by service application running on hosts. These service applications perform most of the tedious work involved in retrieving, verifying, combining, and acting upon information from the CVIS context.

The implication is that services within the CVIS must have a common understanding of the information they must share while interacting (or they and thus the overall system will not behave as expected). Shared information should be enriched with meta-information that supports service applications in a correct understanding of the retrieved information.

New technologies like the semantic web can be of help here.

#### 4.3.2. Sources, Sinks and Transits

Design rule 13: trace the source that sends information and the transits that are used to forward the information to the sink.

The location where data are created is referred to as the Source. The location(s) where it is consumed are the Sinks. Services and/or locations where it is transported reliably without modification (apart from delay) are known as Transits.

For a given data element there is;

- Only a single source (there may be multiple streams carrying the same data from multiple sources, but these are separate and distinct streams).
- Zero or more transits carrying data from the Source to Sink(s).

- Zero or more Sinks.

### 4.3.3. Privacy in cooperation

Design rule 14: prevent unauthorised tracking & tracing of Service users.

From privacy point of view location information is a very sensitive issue and needs special attention. To provide Service Users the ability to control their own space the design of CVIS should be such that:

- Eavesdroppers can not trace the acting of service users within the CVIS community by monitoring their co operations and communications. A weak spot here is that today on communication level fixed addresses are used to steer the communication, such as IP-addresses and MAC addresses in communication bearers. Fixed addresses bring the risk that eavesdroppers will use them (e.g. ‘measuring’ travel times using the MAC addresses of Bluetooth) and implicitly will have the possibility to build patterns. For trustful communications and thus a CVIS an innovation is needed here;
- Work with changing IP-addresses. The implication of working with pseudo-identities and changing IP-addresses is that service users are hard to find or the service provider and service aggregator and have to take the initiative themselves:
  - service users take the initiative towards service aggregators themselves and have to give explicit permission to their service aggregator to track & trace them;
  - service users use certified pseudo-identities;
  - service users use changing IP-addresses while consuming services;

Design rule 15: define the appropriate levels of identification when participating in a cooperative service

To provide Service Users the possibilities to determine for themselves when, how, and to what extent personal information about them is communicated to others the CVIS design should be such that:

- Service Users should be able to hide their identity within non-personalised services by working with certified pseudo identities;
- collected data can not be disclosed without the consent of the data subject;
- The purposes for which personal data are collected shall be determined before the time of the collection of the data and shall be explicit and legitimate at the time of collection of the data and use of the data limited to the fulfilment of those purposes.
- Exchanged data is protected (encryption of data and authorised access of the encrypted data) so is hard for eavesdroppers to build “trails” and “flags” from the intercepted data.
- Even in those cases where the cooperation is build on trust between the actors, identity

and detailed personal data should be hidden as long as there is no reason to mistrust a specific actor. Only in case of argument mistrust authorised employees of a service provider or aggregator can unfold the detailed adapt and identity of the mistrusted person as far as necessary to proof or refute fraud. Before getting access to the data and/or identity these employees have to identify, authenticate them and prove their authorisation. Examples of such trust based cooperation are subscriptions and user based charging services.

**Design rule 16: define retention policy whenever personal data is stored**

From privacy perspective it is preferred: (i) to use as less personal data as possible and (ii) not to store personal data. For the personal data that needs to be stored, a clear retention policy is needed to safeguard that:

- Data has finite storage duration and is not stored longer then strictly necessary for the corresponding applications and subscriptions. In other words, personal data should not be stored longer than the retention period that is defined for it.
- identification of data subjects is used for no longer than is necessary for the purposes for which the data were collected;
- before getting access to personal data employees have to identify, authenticate themselves and prove their authorisation, meaning that access to data is restricted to those who have a demonstrable 'need to know';
- Under no circumstances authorisation is granted to process the collected data in a way incompatible with the purposes for which it was originally collected.

**Design rule 17: Respect the rights of Service Users to: inspect, adjust and block the usage of their personal data**

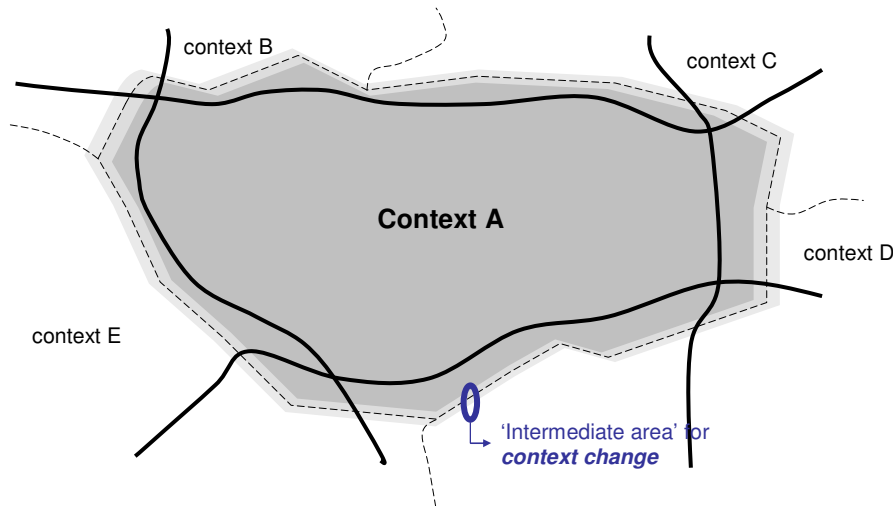
To back-up Service Users “Privacy in their right to be known by their actual identity service providers and service aggregators should guarantee that identity related data is accurate, kept up to date and is not used without permission of the Service User. Service Users should have the possibility to inspect, update / adjust and block the usage of their personal data. Service Providers and Service Aggregators should guarantee that only the updated data is used in all relevant transactions if and only if permission is granted by the Service User.

#### ***4.4. Tuning of services to their current context (Service Context Data)***

**Design rule 18: Define the Service Context Data that will be used to tune a service to a specific context.**

Service Context Data is a set of information which describes the expected behaviour of a service (service application) within a specific context. With each change of the context a new set of data has to be sent or should be already present in the Mobile Unit. The Service

Providers are responsible for distributing the data to the Service Aggregator and the Service Aggregator is responsible to distribute (parts of the) information to the Mobile Unit, possibly after some pre-processing to adapt them to the software used in the Mobile Unit (without changing the meaning of its basic contents).



**Figure 23. Illustration of working with contexts**

To establish an open market and to support interoperability for CVIS a unified context data exchange interface is a requirement. Such an interface might incorporate, for example:

- ContextID
  - ID
- ContextVersion
  - VersionNumber
- ContextSecurityProperties (optional)
  - Signature (optional)
  - SecurityToken (optional)
  - KeyVersion (optional)
- ContextValidityProperties
  - renewalDate (optional)
  - expiryDate
- ContextBasicProperties
  - ServiceProviderID
  - ServiceProviderAddress (optional)
  - TargetServiceAddress (optional)
  - TargetServiceID
  - ServicePriority

- ContextDynamicProperties
  - ContextData (Container) giving expression
    - the geographical border of the context
    - the involved infrastructure
    - the rules and regulations within the context the service should comply to (e.g. the speed limits for a Speed Alert service, vehicle dimensions and weight for Access Control service, classes of hazardous goods for Tracking & Tracing of hazardous goods transports and so on)
- ContextControlProperties
  - Timestamp (date and time of transmission)
- Action
  - actionID
  - actionParameter

A good example comes from the Road Charging Interoperability (RCI) project, where such an interface has been defined for GNSS based road user charging [18]. The interface is adopted and matured by the CEN TC 278.

#### 4.5. Information storage and handling

Design rule 19: define the nature of information (characteristics, types and classification) to enable safe & secure information storage and handling.

Any object within CVIS, whether it is a centre, mobile unit or road side unit, should organise the information – the service application are working with – correctly taking into account the nature of the information. The privacy aspects of storing personal data have already been discussed in par 4.3.3.

##### 4.5.1. Dynamics Characteristics

Information in the CVIS context is categorised into three distinct types depending on the frequency with which the data values change;

- *Invariant Data*: Data elements that will never change over the lifetime of applications using the system. Typically this data will include vehicle provisioning data, service user characteristics and physical constants. Invariant Data can only be changed when all references to that data in the system have been removed such that there is no possibility that different parts of the system may take a different value for a datum. If this restriction cannot be met, then the data should be classified as Static Data. Note that updates to Invariant Data may be accommodated by versioning.
- *Static Data*: Data elements which have slow changing characteristics. Static data are not expected to change during regular operation but may change over time such that temporally indexed local references need to be kept of static data values so that the appropriate values can be selected when post-processing etc. Services need to be able to

cope with changes to static values, but the mechanisms by which these changes are accommodated can be quite ‘expensive’ (e.g. service restart) as they are not frequent. If this restriction cannot be met then the data should be classified as Dynamic Data.

- *Dynamic Data*: Data elements which can change value asynchronously and at any rate. Data which cannot be classified as Invariant or Static data should be considered Dynamic. Services need to be able to accommodate changes in Dynamic Data values efficiently as changes may be rapid and continuous.

#### 4.5.2. Data Types

Data are distinguished into Primitive and Processed types;

- *Primitive Data* comes from basic functions (service primitives) or the native platform; it is never generated by Derived Services. It is not possible to re-generate Primitive Data and so it must be recorded if re-animation of a system state is later required. Examples: Time, Random Number Seed for a PRBS, GPS NMEA strings, power level etc.
- *Processed Data* may come from basic functions (service primitives), the native platforms or derived services. Processed data are created by means of actions performed on Primitive Data and can thus always be re-created if the primitive data have been recorded. There is no requirement to record processed data in order to re-animate the system apart from as an optimisation convenience. Examples: Vehicle tracks, PRBS Random Number, Current Position.

If a data element can be re-generated by means of a process performed on another data element (or set thereof) then it is Processed Data. If it can only be determined by means of an elemental sense (e.g. position, time etc.) then it is Primitive Data.

#### 4.5.3. Information classification

From security and privacy point of view, CVIS-enabled services will process high volume of different types of information.

##### *Payment data*

All data containing information about payment, payments history, credit card numbers, bank account numbers, date and time of transactions. All data sufficient to commit valid transaction must be protected against steal and fraud.

##### *Privacy sensitive data*

Privacy sensitive data contains Personally Identifiable Information (PII), information than can be used to uniquely identify single individual. As a PII can be also considered all kind of information, which with other sources of information can help uniquely identify a single individual. Identification of privacy sensitive data in the system is not obvious and is highly dependent on data required to deliver service. For CVIS-enabled services PII are at least:

- User data: name, surname, date of birth and any unique identifications like Health Insurance Number, driving licence number

- Car data: Licence plate number
- Data related to health and insurance: Information about accidents, style of driving, usage of emergency services.
- Location data: All data containing information about car location, routes chosen by the driver, car staging places.

From privacy point of view important is to understand relation between processed data and privacy of users.

Privacy sensitive data must be protected according to the European Union [20] and national regulations.

The decision what privacy protection measures will be taken is dependent on risk analysis process and the privacy policies following from this analyses [21].

#### *System data*

Any data required for proper functioning of the system, like system configuration data, application data, and service directories.

#### *Encryption keys*

Repository of encryption keys and certificates needs special protection mechanisms against tampering or unauthorised modification.

Each category of information needs suitable protection mechanism applied into the system. The decision what security measures will be taken is dependent on risk analysis process.

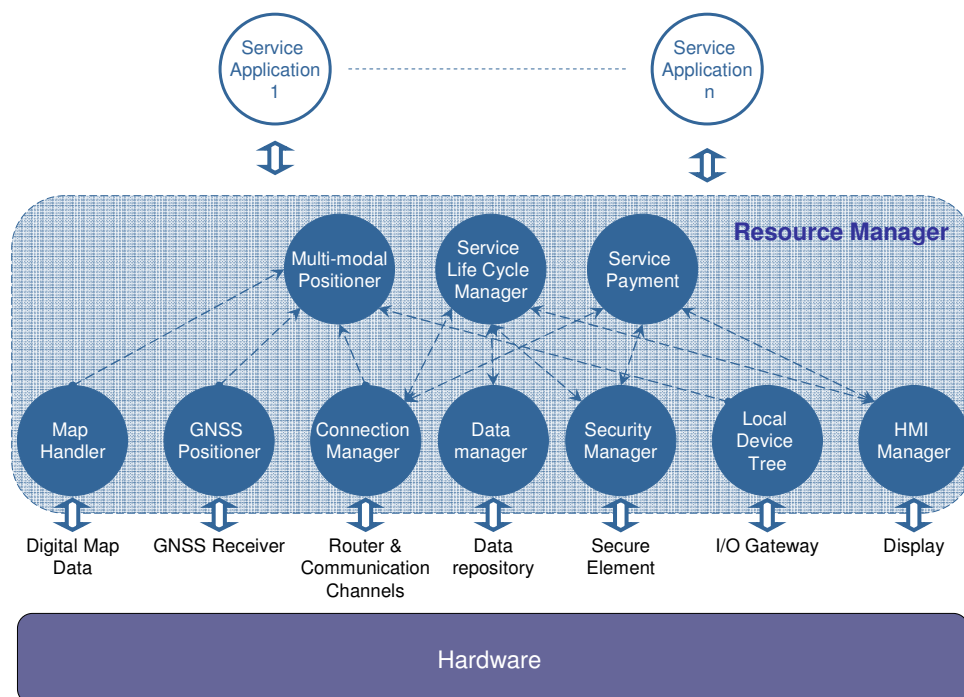
## 5. CVIS from an engineering perspective

### 5.1. Minimise risk on interference between service applications

Design rule 20: design the basic functions (service primitives) of CVIS that enable service applications such that there is no queuing of service applications waiting for the requested data or action.

To prevent redundancy in functionality and interaction with hardware components, CVIS uses basic functions (service primitives, see sub paragraph 3.1). Figure 24 contains an illustration of the corresponding software architecture (on a mobile unit). Basic functions (service primitives, see sub paragraph 3.1) can be invoked by a third party service application.

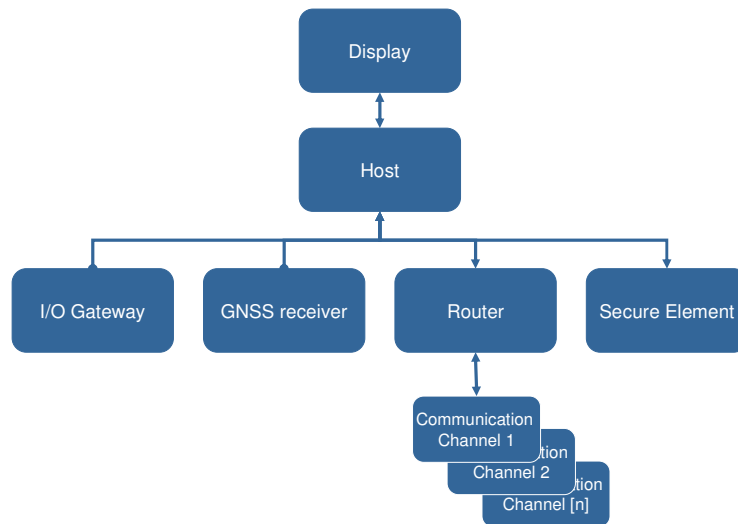
The basic functions should be designed such that service applications do not have to queue to retrieve data from or invoke an action by the basic function.



**Figure 24. Illustration of the software architecture on mobile units**

Design rule 21: design the units of CVIS that enable service applications such that there is minimal risk on interference between service applications.

On hardware level there can be interference between service application on the all the elements (Figure 25): (i) Host, (ii) HMI, (iii) I/O Gateway, (iv) Router and (v) Secure Element.



**Figure 25. Illustration of hardware components in a Mobile Unit**

Interference on I/O Gateway is solved in FOAM via the Local Device tree, where service application can subscribe to new data.

Interference on the Router is solved in COMM via the CALM Communication Manager.

Interference on the Host is still a critical issue. Here the designer of the CVIS unit might decide upon working with more than one Host. For instance, one Host for the critical service (e.g. traffic control or traffic signalling which are road safety grade services) and second Host for other CVIS services.

Interference on the HMI is also a critical issue. Here a broker can help, that: (i) prioritises the requested human-machine interactions taking into account the current work load / mental load of the Service User and (ii) grants access to the HMI following the priorities.

In a similar way interference on the secure element can be dealt with.

## 6. Conclusions

Coming from privacy, security, safety and fault tolerance the design of CVIS should be enhanced such that it includes the functionality to:

- From privacy perspective:
  - operate a service without using personal data if possible;
  - personal data must be collected for explicit and legitimate purposes and used accordingly;
  - personal data must be relevant and not excessive in relation to the purpose for which they are processed;
  - process personal data fairly and lawfully;
  - safeguard that personal data is accurate and kept up to date;
  - safeguard that personal data that identifies individuals is not be kept longer than necessary;
- From security perspective
  - set up a circle of trust between cooperating actors and their units or centres;
  - protect data;
  - provide end-to-end security in communications;
- From safety perspective:
  - unambiguously demarcate the field of application of a service;
  - guarantee interoperability of service operations
  - tune the service operations to the penetration rate of service users
- From fault tolerance perspective:
  - follow predefined structuring rules in the co operations (collaborations);
  - manage the latency in service operations;
  - manage an appropriate usage of the available resources in the unites and centres;
  - safeguard the integrity of data;

- safeguard the validity ('up-to-date') of data;
- Manage the availability of the communication bearers.

To enhance the design of CVIS the following design rules can be used:

- *design rules from organisational viewpoint*
  - design rule 1: define roles and their responsibilities, including the responsibilities with respect to safeguarding privacy, security, safety and fault tolerance
  - design rule 2: define 'grades' for CVIS
    - in the design of CVIS a unambiguous set of 'grades' need to be defined;
    - the minimum set of grades for CVIS should be defined;
    - Every CVIS component brought to the market should state which grades it guarantees.
  - design rule 3: define processes for life cycle management of CVIS objects
    - define the processes and the involved roles in these processes;
    - Define the responsibilities within these processes.
- *design rules from functional viewpoint*
  - design rule 4: define the basic function for the CVS objects
  - design rule 5: define the protection profile for CVIS
    - following the defined grades for CVIS (see design rule 2);
    - Define the protection profile per basic function.
  - Design rule 6: design the Safety instrumented system that should be embedded within the CVIS design.
  - design rule 7: define a method for resource management within the CVIS objects:
    - define the capability profile;
    - define a method (including matching criteria) to fit services to the capability profile(s);
    - define the decision rules to be followed when the selected service can not be fit to the actual capability profile of an object
- design rule 8: support road safety by enabling interoperability via Service Context Data
- design rule 9: Embed the relevant security elements in the design of CVIS
- design rule 10: Verification of the integrity and correctness of received data in the cooperation on continuous basis;

- *design rules from information viewpoint*
  - design rule 11: Define the states of the objects and the conditions under which a state transition takes places;
  - design rule 12: Shared information should be enriched with meta-information that support service applications in a correct understanding of the retrieved information;
  - design rule 13: trace the source that sends information and the transits that are used to forward the information to the sink;
  - design rule 14: prevent unauthorised tracking & tracing of Service users;
  - design rule 15: define the appropriate levels of identification when participating in a cooperative service;
  - design rule 16: define retention policy whenever personal data is stored
  - design rule 17: respect the rights of Service Users to: inspect, adjust and block the usage of their personal data
  - design rule 18: define the Service Context Data that will be used to tune a service to a specific context;
  - design rule 19: define the nature of information (characteristics, types and classification) to enable safe & secure information storage and handling;
  
- *design rules from engineering viewpoint*
  - design rule 20: design the basic functions (service primitives) of CVIS that enable service applications such that there is no queuing of service applications waiting for the requested data or action;
  - Design rule 21: design the units of CVIS that enable service applications such that there is minimal risk on interference between service applications.

## References

- [1] Baggen, Mick, Gijs Withagen, Nol Venema, Christer Larsson, Francesco Alesiani, Florian Wildschutte, Antonio Kung, *Cooperative Vehicle-Infrastructure Systems (CVIS) – Framework for Open Application Management (FOAM) – Architecture and System Specifications*, 31 July 2007, Brussels
- [2] Eichler, Stephan, Jérôme Billion, *Global System for Telematics (GST) – Security, Architecture and Interface Specifications*, 1 June 2005, Brussels
- [3] Fazekas, Imre, Gergely Boza, *Authentication and Authorization Framework - Software Architecture Documentation*, CVIS FOAM SEC A&A Software Architecture Documentation, version 1.1, Brussels, January 2010
- [4] Gustafson, Per, Sofia Doncheva, Hans Ulrich Michel, Florian Wildschutte, Hurgen Wojatchek, Volker Vieroth, Erwin Vermassen, *Global System for Telematics (GST) – Open Systems (OS), Architecture and Interface Specifications*, 1 June 2005, Brussels
- [5] ISO/IEC10746-1, First edition, 1998-12-15 Information technology — Open Distributed Processing — Reference model: Overview, Reference number ISO/IEC 10746-1:1998(E)
- [6] ISO/IEC10746-2, First edition, 1996-09-15 Information technology — Open Distributed Processing — Reference model: Foundations, Reference number ISO/IEC 10746-2:1996(E)
- [7] ISO/IEC10746-3, First edition, 1996-09-15 Information technology — Open Distributed Processing — Reference model: Architecture, Reference number ISO/IEC 10746-3:1996(E)
- [8] ISO/IEC10746-4, First edition, 1998-12-15 Information technology — Open Distributed Processing — Reference model: Architectural semantics, Reference number ISO/IEC 10746-4:1998(E)
- [9] Mühlethaler, Franz, Thomas Kallweit, Oene Kerstjens, Volker Vierroth, Ralf Grigutsch, Francois Malbrunot, Jean Marc Gautier, Walter Scheibenberger, *Road Charging Interoperability (RCI) - Security Architecture for Interoperability*, version 1.01, 28 June 2007, Brussels
- [10] Schmid, Andreas, Arnor Solberg, Christer Larsson, Erik Olsen, Frank Tuijnman, Gino Franco, Hans Haddingh, Jean-François Gaillet, Josef Kaltwasser, Knut Evensen, Marcel Konijn, Mick Baggen, Richard Bossom, Siebe Turksma, Thierry Ernst, Zeljko Jetic, *Cooperative Vehicle-Infrastructure Systems (CVIS) – Core Architecture Group – High Level Architecture*, 1 March 2007, Brussels
- [11] Arnor Solberg, Andreas Schmid, Mick Baggen, Francesco Alesiani, Hannes Stratil, Richard Bossom, Silke Forkert, Marius Schlingelhof, Axel Burkert, Sjoerd Haverkamp, Paul Mathias, *Cooperative Vehicle-Infrastructure Systems (CVIS) – Core Architecture Group – Architecture and System Specifications*, 31 July 2007, Brussels
- [12] Schlingelhof, M., J-F. Gaillet, S. Dreher, K. Demaseure, D. Betaille, Ch. Bartels, Ph. Bonnifait, M. Spirito, F. Sottile, F. Alesiani, Ph. Poire, F. Peyret, L. Harte, N. du Lac, B.

- Schokker, *Cooperative Vehicle-Infrastructure Systems (CVIS) – Positioning, Mapping and Location Reference (POMA) – Architecture and System Specifications*, 31 July 2007, Brussels
- [13] Solberg, Arnor, Hannes Stratil, Vilmos Nebehaj, Thierry Ernst, *Cooperative Vehicle-Infrastructure Systems (CVIS) – Communications Service compliant with CALM (COMM) – Architecture and System Specifications*, 31 July 2007, Brussels
- [14] Stanley Smith Stevens, *On the Theory of Scales of Measurement*, Science 103 (2684): 677–680., 1946
- [15] van Koningsbruggen, Paul, Nol Venema, *Cooperative Vehicle-Infrastructure Systems (CVIS) – Security, safety and fault tolerance*, CVIS deliverable D.DEPN.3.1, Brussels, 29 January 2010
- [16] Van Koningsbruggen, Paul, Dave Marples, Hubert Mikulicz, Thomas Stranner, *GNSS-enabled Services Convergence (GSC) – Architecture and Interface specifications*, May 2010, Brussels
- [17] Venelin Arnaudov, et al, *Global System for Telematics (GST) – Service Payment (SPAY), Architecture and Interface Specifications*, 31 May 2005, Brussels
- [18] Vierroth, V., F. Bode, C. Egeler, R. Grigutsch, T. Kallweit, O. Kerstjens, F. Mühlethaler, W. Scheibenberger, P. van Haperen, P. van Koningsbruggen, *Road Charging Interoperability (RCI) - Minimum Architecture for Interoperability*, version 1.01, 8 February 2007, Brussels
- [19] Cavoukian, Ann, <http://www.privacybydesign.ca/>
- [20] European Privacy directives: Directive 95/46/EC & Directive 2002/58/EC
- [21] PRECIOSA: <http://www.preciosa-project.org/>