Cooperative Vehicle-Infrastructure Systems

D.DEPN.2.1 – final version	**Cooperation architecture and requirements on content interfaces for interoperability**

SubProject No.	**SP 1.3**	**SubProject Title**	**DEPN**
Workpackage No.	**Topic 2**	**Workpackage Title**	**Openness & interoperability**
Task No.	**N.A.**	**Task Title**	**N.A.**
Author(s)		**Thijs de Graaff, Marcel Konijn, Paul van Koningsbrugge, Marco Annoni, Massimo Cocozza, Mats Rosenquist, Hans-Joachim Schade, Bart Driessen**	
Dissemination level		**PU**	
File Name		**DEL_DEPN_2.1_Interoperability_v1.5**	
Due date		**28 February 2010**	
Delivery date		**02 June 2010**	

Abstract	This document is an update of the D.DEPN2.1 deliverable that has been delivered 1 Feb 2008. With respect to the original deliverable, the requirements on openness and interoperability are explicitely formulated. Besides that, the validation procedure and results related to these requirements are given.

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

Control sheet

Version history			
Version	**Date**	**Main author**	**Summary of changes**
0.1	Jan 9, 2008	Thijs de Graaff	initial version
0.2	Jan 14, 2008	Thijs de Graaff	Topic 2 members comments & additions
1.0	Jan 15, 2008	Thijs de Graaff	submitted for review
1.1	Feb 10, 2010	Bart Driessen	Updated version.
1.4	May 10, 2010	Bart Driessen	Final Draft Version
1.5	June 2, 2010	Bart Driessen	Final Version
	Name		**Date**
Prepared	Bart Driessen		10/05/2010
Reviewed			
Authorized	Lina Konstantinopoulou		02/06/2010

Circulation	
Recipient	**Date of submission**
European Commission	02/06/2010
Project consortium	02/06/2010

Table of Contents

Abbreviations and Definitions

Abbreviation	Definition
ANSI	American National Standards Institute
API	Application Programming Interface
IEEE	Institute of Electrical and Electronics Engineers
CALM	Communication Access for Land Mobiles, provides a layered solution that enables continuous (or quasi continuous) communications between vehicles or between vehicles and the infrastructure. It is a ISO TC204 Working Group 16 standard
ISO	International Standards Organization
CVIS	Cooperative Vehicle Infrastructure Systems
DEPN	DEPloyment eNabling, integrated project activity
ICT	Information and Communication Technology
ITS	Intelligent Transportation Services
ETSI	European Telecommunications Standards Institute
TC	Technical Committee
OASIS	Organization for the Advancement of Structured Information Standards
FOAM	Framework for Open Application Management (CVIS subproject)
GST	Global Systems for Telematics (EU project)
COMM	Communication and networking (CVIS subproject)
URI	universal resource identifier is a locator or a name of a resource or both. As a locator it provides means of acting upon or obtaining a representation of the resource by describing its primary access mechanism or network "location"

Executive Summary

The charter of Deployment Enablers Topic 2 is to devise a “cooperation architecture” for the exchange of information between applications included in the CVIS sub-projects. Also for all “content interfaces” requirements must be identified to ensure interoperability between the various system parts.

This document contains the opinion of the DEPN Topic 2 working group about the cooperation architecture and requirements for CVIS components with respect to interoperability.

The focus has been on deployment *after* the CVIS project, and less attention to the deployment of the CVIS example applications currently being developed, as these are likely to be replaced in the future by commercial software anyway.

The concepts ‘openness’ and ‘interoperability’ are defined like this working group feels they should be used within a CVIS context.

A few recommendations are given for development of CVIS applications and relevant standards are indicated, as well as specific requirements that hold for CVIS applications with implications for openness and interoperability. The meaning of semantics for the CVIS domain is treated briefly and consequences for design and implementation are indicated.

The cooperation architecture has to cope with various opportunities and threats that arise from trade offs that must be made. An enumeration of opportunities and threats is made that should be ticked off when taking decisions during trade offs.

The cooperation architecture for CVIS is based on various levels of interactions, comprising business models, business cases in the form of services offered, business protocols represented by contracts, service deployment and interconnection protocols.

The CVIS cooperation architecture is given as a CVIS “cooperation stack” of services and contracts together with more generic, supplementary standards and services like (open) standards, a semantic model sharing service and secure interoperability by proper identity services.

A validation plan is given for measuring to what extend the recommendation resulted in an open and interoperable system. The results of the actual validation are also supplied.

1. Introduction

CVIS relevancy; its mission revisited

CVIS, cooperative vehicle infrastructure systems –more precise: vehicle-to-vehicle and vehicle-to-infrastructure systems– will allow vehicles to cooperate directly with other nearby vehicles, and with the immediate roadside infrastructure, thus sharing information on the latest road and traffic status for greater safety, efficiency and a better environment. Each equipped vehicle will be able to connect and communicate via local ad-hoc networks of vehicles and roadside equipment in the vicinity, and also via available mobile networks to access a wide range of journey support and other services.

The aim of CVIS is to develop an open and interoperable concept for cooperative systems, based on a standardized open architecture platform and common software modules for key applications: any equipped vehicle should be able to access and run any application, anywhere in Europe where there is compatible roadside infrastructure, mobile networks or nearby vehicles.

CVIS DEPN Topic 2 charter

The goals of the horizontal Deployment Enablers activity are in general to:

- Ensure that the core technologies and applications as developed in the CVIS project are fundamentally deployable and that non-technical issues have been identified and their potential impact on deployment described along with recommendations as to how these issues could be addressed;
- Derive road maps on how to migrate from today's situation, via an intermediate phase when penetration of equipped vehicles and infrastructure grows to a critical mass, to a future with widespread take-up of operational CVIS, based on transparent deployment and cooperative business models with suitable sharing of responsibilities and liabilities;

and, more detailed, Topic 2 has the goal to:

- Devise a "cooperation architecture" for the exchange of information content between all entities involved in the set of applications included in the CVIS sub-projects;
- For all "content interfaces" between entities, identify the requirements to ensure interoperability between systems.

About this document

This document contains the opinion of the DEPN Topic 2 working group about the cooperation architecture and requirements with respect to interoperability.

Also there has been a focus on deployment *after* the CVIS project, and less attention to the deployment of the CVIS example applications, as these are likely to be replaced in the future by commercial software.

2. Openness and Interoperability

2.1. *Definitions*

Hereafter we will define the concepts 'openness' and 'interoperability' as we feel they should be used within a CVIS context. We will not give a definition straight away, but rather touch upon aspects that describe the concepts or, in contrary, that are excluded from the concepts.

2.1.1. Openness

For the meaning of "openness" we have to think about the nature of cooperative systems. Cooperative systems should be 'open' in such a way that:

inclusion	**exclusion** (i.e.: 'openness' is not)
Each actor, be he private or public, should be able to plug his service into this environment	A vendor lock in of a specific system
It should be possible to make use of each other (sub) services.	An environment which is to be developed with special tools
The services should be able to be distributed freely	A black box which can not be communicated with from outside
It should be possible to exchange information with each other for particular services. The type and nature of information exchange can differ per service. For instance the systems of the public road authorities will be open such that applications can read and write back. However it will not be completely open. There will always be security aspects to prevent systems from being hacked.	Containing indispensable interfaces and system components "hidden" within a proprietary "black-box" and at the same time requiring the use of this sub-system.
It should be relatively easy to develop new services for the CVIS environment. No extensive knowledge of the other systems is required.	A service which is solely based on a specific communication technology that might be banned by some national or local regulation
Using established public standards related to the CVIS environment and available de-facto ICT standards, for software and communication.	
It has well documented interfaces	

2.1.2. Interoperability

Also here we have to think about the nature of cooperative systems. Cooperative systems should be 'interoperable' in such a way that:

inclusion	**exclusion** (i.e.: 'interoperable' is not)
A new service build in accordance with CVIS principles and recommendations can communicate with the environment and perform as expected Can run on various hardware platforms A CVIS service will behave independently of the involved actors and sub-system suppliers. Vehicles moving on the road infrastructure are able to use ITS services from different service providers, whenever presented to the users in the consistent and uniformed way of the CVIS recommendations	A service will only perform as expected in a specific environment When CVIS enabled vehicles using the road infrastructure are not able to use available ITS services Specific ITS services behaving differently depending on the service providers and the client systems involved When vehicles roaming in different countries cannot be provided with a local service "equivalent" to the one subscribed to or provided in its home country

2.2. *Applicable standards*

Beside the general standards as issued by bodies like ANSI and IEEE also standards like emerging from the open source initiatives are relevant, the CALM ISO standard and the ETSI TC ITS.

Current open standards can be found at the website[1] from OASIS (Organization for the Advancement of Structured Information Standards), and for identity assurance through the website[2] of the Liberty Alliance.

Within CVIS the application framework offered by the FOAM subproject (Framework for Open Application Management, which amongst others heavily relies on the work done by GST project) is highly recommended as the way to develop, distribute and deploy CVIS services.

2.3. *Recommendations*

For development of CVIS applications it is recommended that one:

- adheres strictly to the CVIS recommended practices for application design (like f.i. as laid down in CVIS deliverables D.FOAM.3.1, D.COMM.3.1, and D.COMO.3.1)
- builds whenever possible on the CVIS Framework for Open Application Management
- is aware during the design phase of possible vulnerabilities to compromising the

[1] http://www.oasis-open.org

[2] http://www.projectliberty.org

systems integrity and builds in proactive counter measures (like for instance identity assurance and also critical performance issues)

- adheres to general standards for application development as available from ANSI and IEEE.

2.4. Requirements for CVIS applications

CVIS applications are meant to co-exist and to co-operate without any adverse interference. However, the context within which the applications practically will run cannot be predicted accurately. There will be a kind of local and global context to take into account. The (highly volatile) local context comprises temporary present objects in the vicinity of the applications user, whereas the (more statically in space and time) global context touches upon the global infrastructural elements of the CVIS world (e.g. back office systems, standard services, et cetera).

As the CVIS system relies heavily on resources with a limited capacity in for instance bandwidth, computing performance, availability, et cetera, it exhibits the nature of real time systems and the corresponding challenges.

Another aspect of the CVIS system is the continuous change in services offered to the end user, as well as the deployment of software upgrades and updates, maintaining equivalency of services over larger areas, compatibility of information, et cetera.

Bearing the previous in mind the following requirements hold for CVIS applications:

- It should be possible to develop and deploy applications forming services based on available system components even if the system is not complete. Then the applications forming the building blocks for a service should be possible to independently be replaced, updated and enhanced when the service is maturing.
- An application should at large be independent from others giving the possibility to independently replace it.
- The issue of "co-operation" among different concurrent applications should be taken into account during developments. Up to now in the CVIS project all applications have been conceived as stand-alone processes without taking care of any co-ordination with the other applications and with full use of all the processing and communication resources. In a real deployment this might cause the provision of conflicting or wrong information. Moreover, when the application requires an interaction with a human being, the human factor should be carefully considered in order to avoid the provision of too much information (or conflicting information) that may be a distracting factor and decrease the driver safety.
- It is highly recommendable to devise a resource sharing mechanism on a CVIS base level (in the CVIS cooperation stack; see also section 3.2) in order to budget the resource use per application. Likewise, for human interaction budgets should be defined per application given the set of applications active at any time. Applications must be budget aware in the above respects.

2.5. About the use of semantics

The use of a semantic approach is highly desirable given the diversity of the ITS world and the long time range for which the CVIS system is to be designed. The required supportive means to provide for a proper interoperability for CVIS applications is still not in place to date. The best attempt known is the initiative taken by the UK Highway Authorities running a pilot project on their ITS Metadata Registry.

At this time it can be foreseen that convergence of ITS information descriptions necessarily will take place in the future using some shared mechanism of moderation. It is wise to embed the hooks to semantic descriptions in the design of new CVIS applications. This might take the form of a placeholder for a pointer to a (not yet existing) semantic description. This pointer preferably will have the form for an URI (universal resource identifier), a compact string of characters used to identify the source of a semantic description.

3. Cooperation architecture

3.1. Problem area

The cooperation architecture has to cope with various opportunities and threats that arise from trade offs that have to be made. Hereafter an enumeration of opportunities and threats is made that should be ticked off when taking decisions during trade offs.

Opportunities	**Threats**
It allows multiple stakeholders to take part in this environment and lowers the barriers for new stakeholders entering. Private companies will have the opportunity to provide profitable services and system components and protect proprietary solutions to enable investments. Public stakeholders can fulfill their mission goals, specially if we think about large information dispatching. Public stakeholders will be able to provide same services to all users The public and private stakeholders will accomplish their goals to make the traffic more safe and efficient Services from complete different environments can plug into this world without rewriting The cost of the full deployment can be shared among different application domains and facilitate the overall return of investment	Increased risk of antagonistic system threats such as viruses, worms and Trojan horses Undesired services e.g. advertisements and spam Unsafe services Violation of privacy Data security issues such as user identification and authentication to prevent unauthorized usage Unreliable services Less robust services Lack of end-to-end responsibility of services causing quality-of-service problems and lack of ownership Open source software might get mixed into existing software, implying that proprietary software risks to be earmarked as being "open source" and hence causing loss of commercial willingness to invest Badly designed services causing performance problems of the overall system Lack of awareness during the design phase about issues related to concurrent execution of different applications (i.e. priority, sharing of resources, conflicts, etcetera)

3.2. Solution directions

For the short term, that means the period during which the CVIS project runs, the solution for openness and interoperability lays in the construction of a fixed CVIS data model and appliance of a fixed set of well defined application interfaces and the framework offered by FOAM. For the long run, however, this approach will not provide a rigorous solution to an ever changing world.

The cooperation architecture for CVIS is based on various levels of interactions, comprising business models, business cases in the form of services offered, business protocols represented by contracts, service deployment and interconnection protocols. This is schematically depicted in Figure 1, together with the components delivered by CVIS or yet to be delivered in a later stage (world models).

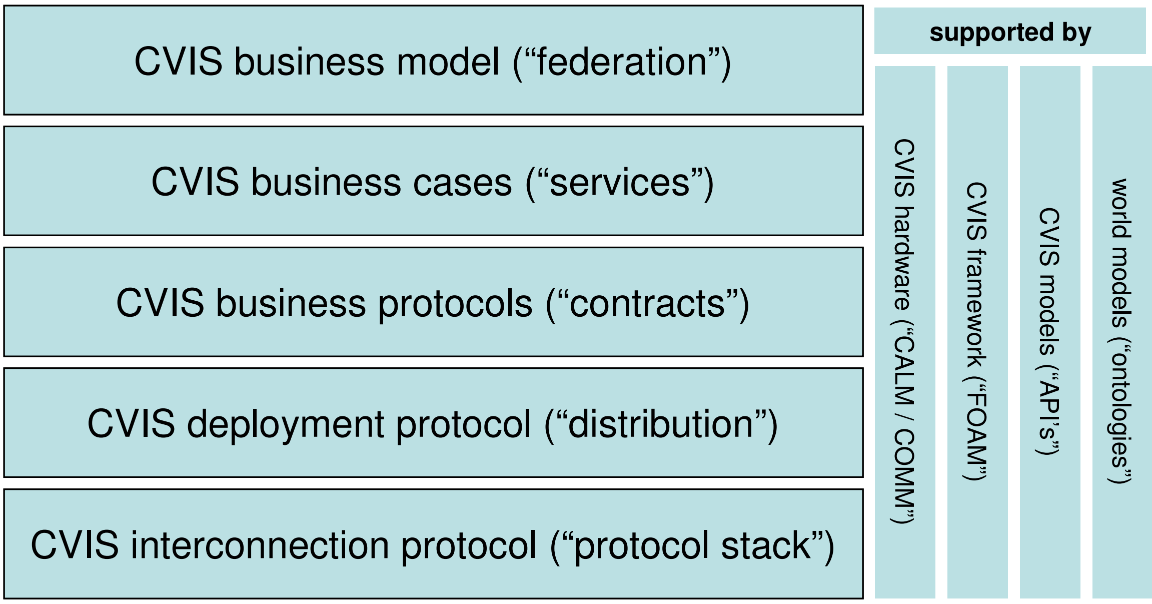


Figure 1, The CVIS cooperation stack and supportive components

3.3. Meta data registration

The set of well defined application interfaces and the framework offered by FOAM, together with the work done in the COMM subproject and based on CALM establish the base part of the cooperation stack. On top of this a mechanism must be set up to cope with the ever changing world. New business models will appear and old ones become obsolete. New concepts will evolve from old ones, the old ones becoming less frequently used and eventually obsolete. Naming will change, inconsistencies arise, etcetera. Coping with this imminent chaos necessitates maintenance on a more abstract level. This has been foreseen in the past, and already an initiative of the Highway Authorities in the UK has established an ITS Metadata Registry[3] to foster harmonization across different systems and avoid re-invention and duplication of effort. The essence consists of a central registration and a voluntary adherence to a widely supported model of the ITS world. At the same time there is room for existence of a federation of ITS businesses with a certain degree of proprietary view

[3] http://www.itsregistry.org.uk

upon the world, which is reflected in a granted compliancy level via a certification program.

The cooperation stack, in fact representing a series of contracts on top of each other, is within CVIS FOAM implemented as a tripartite interaction between the roles “service center”, “control center” and “service endpoints” (like a CVIS vehicle, a piece of roadside equipment or a nomadic device), see also Figure 2.

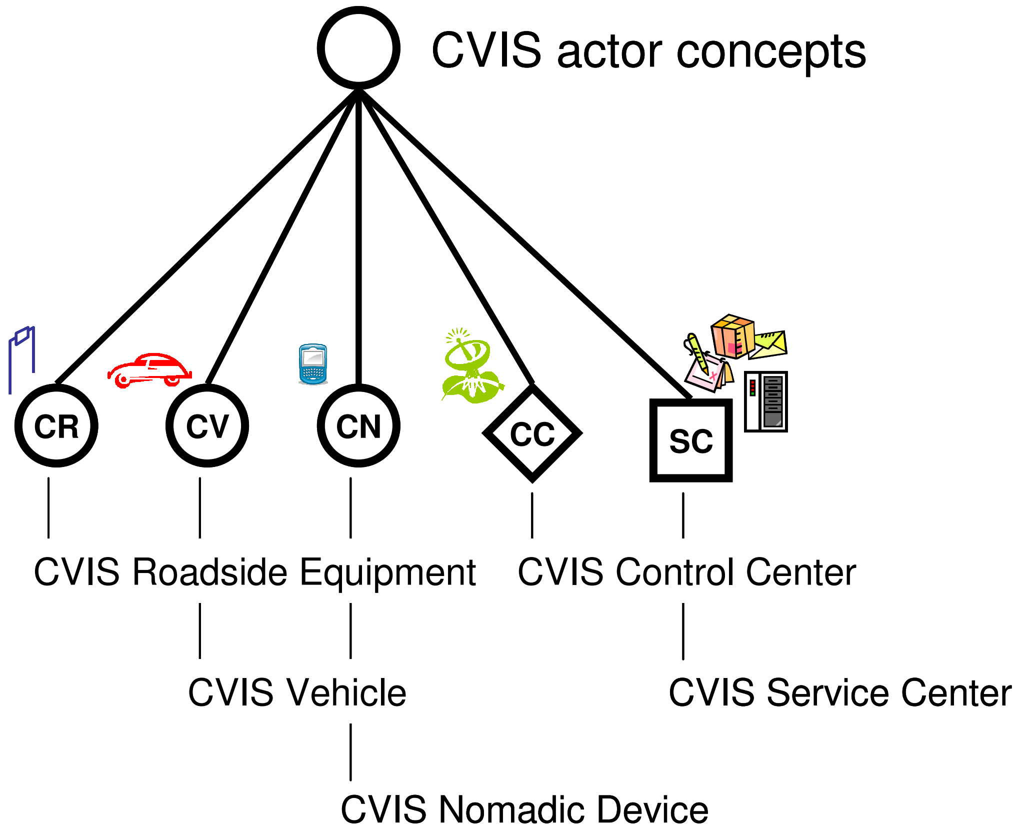


Figure 2, Various CVIS actor concepts and roles

The various roles and contracts are arranged into a service deployment role model. An example is depicted in Figure 3. From this example it can clearly be seen that the way businesses bind their customers is determining their market share (visualized by the colored dashed lines). He who knows how to bind the proper control centers gets the optimal market share. A company offering services for nomadic devices will have other interests than a business interested in services for specific roadside equipment.

Business will need to lean on (non-CVIS) services in order to streamline and secure their applications. This may vary from identity assurance in transactions to services like currently offered by the pilot project (see also section 2.2) implemented by Mott MacDonald on behalf of the Highways Agency (running until June 2008).

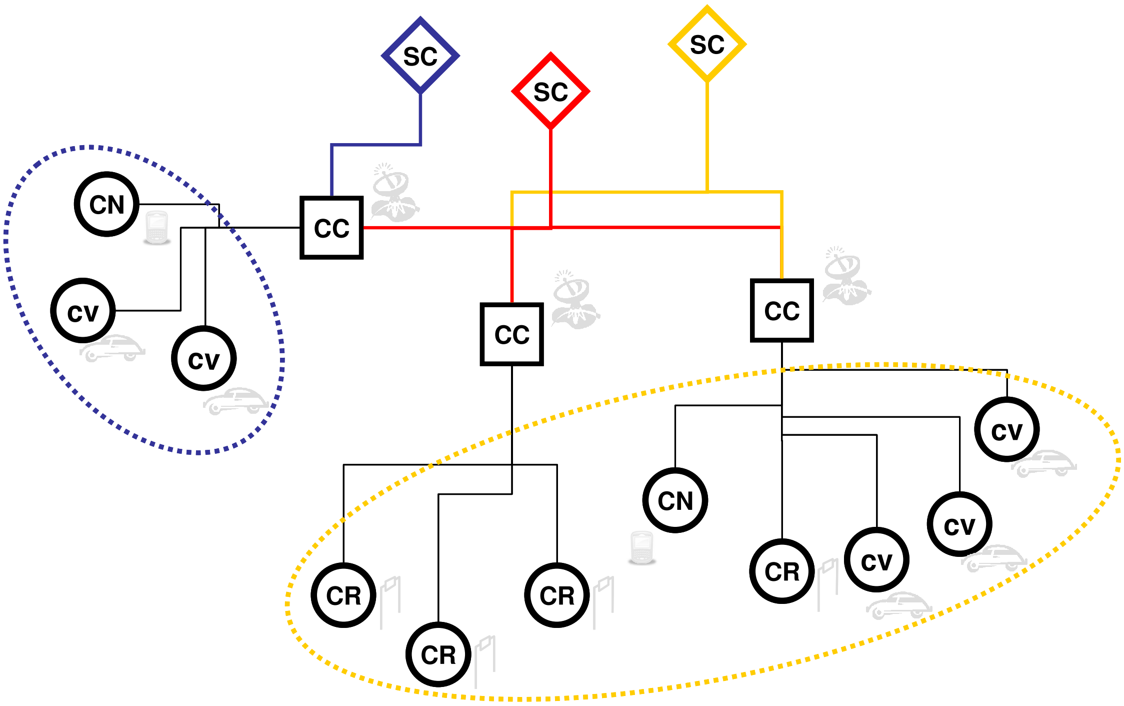


Figure 3, A CVIS service deployment role model example

Summarizing, the CVIS cooperation architecture will consist of the CVIS cooperation stack together with more generic, supplementary standards and services like (open) standards, a semantic model sharing service and secure interoperability by proper identity services (see also Figure 4).

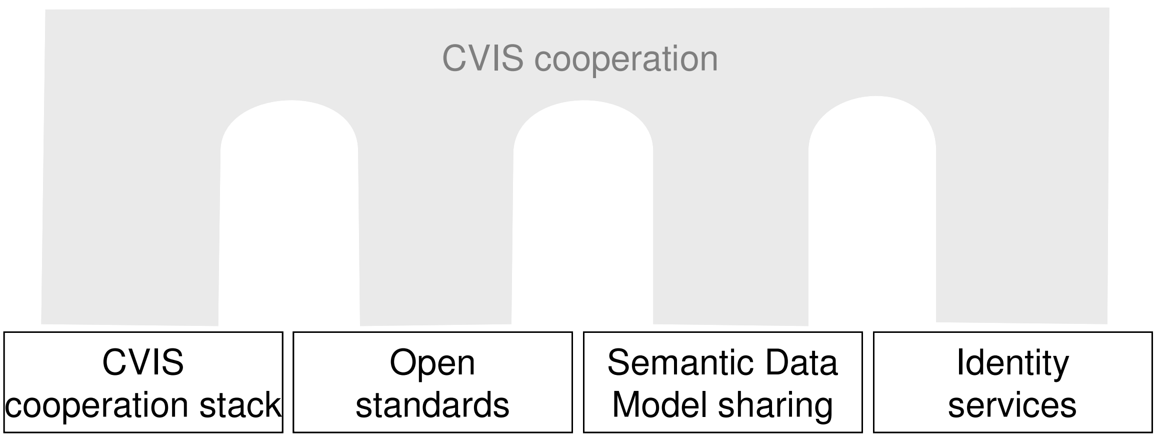


Figure 4, The CVIS cooperation architecture built on internal and external resources

4. Requirements and validation

Chapter 2 lists high level requirements on the area of openness and interoperability. This list of requirements served as the basis of setting up the non technical validation elements for DEPN topic 2. As mentioned in the deliverable D.CVIS.6.1 Non-technical validation elements, aims to specify a list of recommendations for all DEPN topics that will define enablers for smooth deployment (non-technical validation elements: NTVE). In the non-technical validation process, it will be identified how accurate these enablers were followed during the progress of the project.

4.1. Requirements

For topic 2 the following non technical validation elements (requirements) have been identified:

The high and low level architectural design of CVIS must be such that:

a. An end user can trust that CVIS will appear to be a performing system over the years, functioning as an extendable package of solutions for his ITS needs.

b. An end user can expect CVIS to provide a service environment that allows the incorporation of extra services from any organisation that supports CVIS.

c. An end user can use and access simultaneously multiple relevant services.

d. An end user can add new services to the environment without disturbing the existing services.

f. CVIS should be interoperable with legacy systems, safeguarding long-term investments.

g. The overall CVIS infrastructure leaves an open and flexible way for information to be exchanged between stakeholders (typically being 'hosts')

k. There are open and published standards that enable oem and suppliers to enter CVIS supply chain

m. An OEM will be capable of offering custom-made functions in addition to a CVIS compliant function.

o. When developing CVIS components, an OEM will be free to use its own trusted field of development and the language it uses in that domain.

p. When an OEM obtains CVIS components from other vendors, these will work directly (plug-and-play).

In next paragraph these non technical validation elements are refined and a validation procedure is added.

4.2. Validation procedure

The validation procedure followed is defined in the validation plan (D.CVIS.6.2). Following this plan the following validation tests are identified.

ID	Name	Description
NTVE_T2.a	Reliability	An end user can trust that CVIS will appear to be a performing system over the years, functioning as an extendable package of solutions for his ITS needs.
Validation manner	N.A.	Since this requirement can only be validated after the project has ended, it will not be validated during the CVIS project.

ID	Name	Description
NTVE_T2.b	Openness for organizations.	An end user can expect CVIS to provide a service environment that allows the incorporation of extra services from any organisation that supports CVIS.
Validation manner	○ Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM-OB3.
Test description	Service submission contest" for new CVIS applications in months 36. Application developers with best ideas will receive up to 25.000 funding from CVIS to develop within 3 months CVIS applications.	
What to measure	This application will be tested at least at one CVIS test sites using the CVIS reference platform.	
Expected outcome	Various innovative CVIS applications will be developed. Validation of FOAM SDK by application developers.	

ID	Name	Description
NTVE_T2.c	Interoperability for applications..	An end user can use and access simultaneously multiple relevant services.
Validation manner	○ Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM-OB1.1_t.
Test description	CVIS applications and services can be developed, deployed and provisioned on an open end-to-end framework and run-time environment.	
What to measure	Test of different applications (e.g. access control, dangerous goods in CF&F) at one test site (e.g. Gothenburg). Tests of same application (access control) in different test sites (e.g. Gothenburg and Lyon) using core FOAM functionality.	
Expected outcome	Deployment and provisioning of different application at one test site and of same applications in different test sites.	

ID	Name	Description
NTVE_T2.d	Openness	An end user can add new services to the environment without disturbing the existing services.
Validation manner	○ Test	Implement two applications, and assure that both applications are running in parallel.
Test description	A scenario will be developed where more than one application will run in parallel. The scenario will activate two major CINT applications (CTA and EDA).	
What to measure	The validation tests as described for CTA and EDA in parallel.	
Expected outcome	The outcome of the validation tests is identical with the outcome of the validation tests when run separately.	

ID	Name	Description
NTVE_T2.f	Interoperability with legacy systems.	CVIS should be interoperable with legacy systems, safeguarding long-term investments.
Validation manner	○ Test	Outside the scope of this project.

ID	Name	Description
NTVE_T2.g	Interoperability between stakeholders/hosts.	The overall CVIS infrastructure leaves an open and flexible way for information to be exchanged between stakeholders (typically being 'hosts')
Validation manner	○ Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM OB1.2t.
Test description	The DDS (Distributed Directory Service) provides a kind of yellow pages for the applications and services available in a geographic area. In an interurban scenario: a vehicle drives from Antwerp to Rotterdam. The DDS information will be provided from a gantry or on a rest area. In an urban scenario: A vehicle approaches an intersection, in four main phases: a) retrieval of services b) submission/registration to the service, c) notification of the service availability d) start of the service itself.	
What to measure	Time for set-up connection, time for information exchange, data rate needed and other technical parameters.	
Expected outcome	CVIS infrastructure leaves an open and flexible way for information to be exchanged between stakeholders	

ID	Name	Description
NTVE_T2.k	Business accessibility.	There are open and published standards that enable oem and suppliers to enter CVIS supply chain.
Validation manner	○ Inspection ○ Deliverable	This test will be implemented in FOAM technical validation, and is covered by test D.CVIS3.2..
Reference document	D.CVIS3.2. (High Level Architecture)	
What to look for	A chapter containing mentioned standards.	

ID	Name	Description
NTVE_T2.m	Openness to custom made functions.	An OEM will be capable of offering custom-made functions in addition to a CVIS compliant function.
Validation manner	○ Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM-OB2.
Test description	Service submission contest" for new CVIS applications in months 36. Application developers with best ideas will receive up to 25.000 funding from CVIS to develop within 3 months CVIS applications.	
What to measure	This application will be tested at least at one CVIS test sites using the CVIS reference platform.	
Expected outcome	Various innovative CVIS applications will be developed. Validation of FOAM SDK by application developers.	

ID	Name	Description
NTVE_T2.o	Openness to development environment.	When developing CVIS components, an OEM will be free to use its own trusted field of development and the language it uses in that domain.
Validation manner	o Inspection o Deliverable	
Reference document	D.CVIS.3.3. Architecture and system specification.	
What to look for	Are interfaces properly specified, and is it specified how to use these interfaces for different languages.	

ID	Name	Description
NTVE_T2.p	Openness to hardware vendors	When an OEM obtains CVIS components from other vendors, these will work directly (plug-and-play).
Validation manner	o Test	Not in scope of the CVIS project.

4.3. Interoperability questionnaires

Since one of the most important CVIS objectives relates to interoperability, special emphasize is placed on inspecting to which extend during the project interoperability has been realised between:

- CVIS subprojects (e.g. interoperability between CINT, COMO, POMA).
- CVIS and SAFESPOT (e.g. share hardware between CVIS and SAFESPOT)
- CVIS and COOPERS (e.g. use data collected via COOPERS).

In order to assess the level of interoperability, a set of questionnaires has been developed, focusing on different aspects of interoperability.

1. SP leader questionnaire, aiming to assess to what extend subproject interoperability has been accomplished. Target audience: SP leaders. The questionnaire can be found in appendix 1.
2. A special questionnaire for assessing the interoperability between CVIS and SAFESPOT. Target audience: chief architect and organisers of major showcase events where CVIS and SAFESPOT were demonstrated in parallel. This questionnaire can be found in appendix 2.
3. A special questionnaire for assessing the interoperability between CVIS and COOPERS. Target audience: chief architect and organisers of major showcase events where CVIS and Coopers were demonstrated in parallel. This questionnaire can be found in appendix 3.

4.4. *Open source questionnaires*

An important objective of the CVIS project is that the vast majority of the developed software modules and interfaces are using an open source license model. CVIS high level objective 2 states that "*80% of drivers, operating system elements and application software are under open-source licence*".

This requirement is far from straightforward to measure: is it counting program lines, counting modules, or even something else. How to weigh one large module with respect to several smaller ones. The rationale behind the requirement obviously is that after the project it should be possible to continue improving the core technologies and the applications after the CVIS lifespan, without having to close contracts with several parties. Therefore it is even more important that the core-technology subprojects provide open source code and interfaces. Applications developed during the CVIS lifespan demonstrated the principle of cooperative systems and showed the possibilities and usefulness of the core-technologies. To what extend the algorithms and software will be reused and improved after the project is uncertain at this moment. However what is important is that applications developers will have the possibility to use the CVIS core-technologies and improve this software where needed. Therefore, the open source requirement of the core-technologies is weighed as more significant compared to the open source requirement of the application subprojects.

Nevertheless, it remains complicated to quantify this weighing, or the 80% criterion. The approach followed for validating this high level objective is as follows.

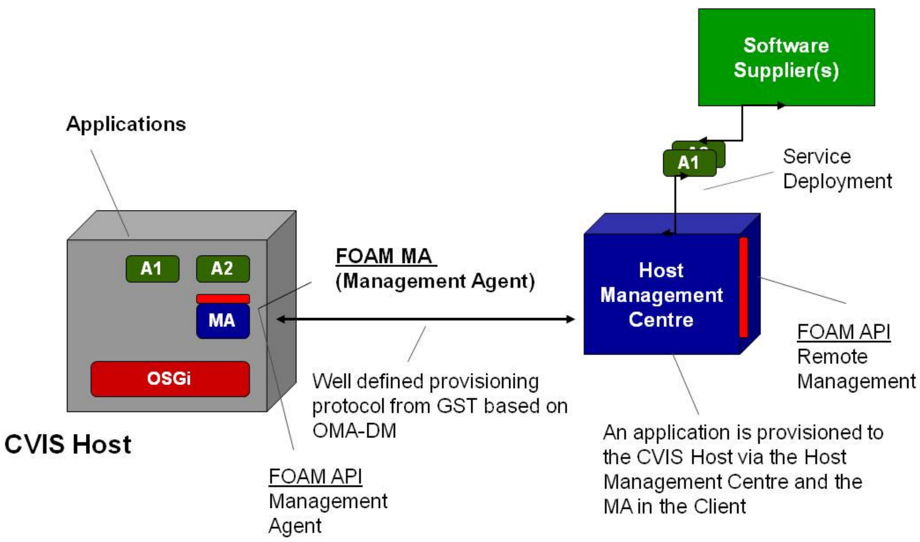
1. SP leaders are asked to list all the software modules that were developed during the CVIS project, or that are essential for deploying a working system.
2. For each module the SP leaders are asked to specify the license that is applicable to this software modeule.
3. Each SP leader is asked to do a self assessement to measure to what extend the 80% open source criterion has been reached in ther subproject.
4. Based on all inputs received a conclusion is drawn for the entire CVIS project.

5. Validation results

5.1. Results non technical validation elements

ID	Name	Description
NTVE_T2.a	Reliability	An end user can trust that CVIS will appear to be a performing system over the years, functioning as an extendable package of solutions for his ITS needs.
Validation manner	N.A.	Since this requirement can only be validated after the project has ended, it will not be validated during the CVIS project.

ID	Name	Description
NTVE_T2.b	Openness for organizations.	An end user can expect CVIS to provide a service environment that allows the incorporation of extra services from any organisation that supports CVIS.
Validation manner	o Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM-OB2.
Test description	Service submission contest" for new CVIS applications in months 36. Application developers with best ideas will receive up to 25.000 funding from CVIS to develop within 3 months CVIS applications.	
What to measure	This application will be tested at least at one CVIS test sites using the CVIS reference platform.	
Expected outcome	Various innovative CVIS applications will be developed. Validation of FOAM SDK by application developers.	

ID	AO_FOAM_OB2_t
Actual test setup	The CVIS projects initiated an application innovation contest 2009. The aim was to stimulate innovation by developers both within and external to the project to develop CVIS-compliant services and to validated that external providers can deploy CVIS applications and services using the CVIS software. 22 high-quality concepts were submitted in the first phase. Based on the documentation the jury of CVIS experts from Daimler, Swedish Road Administration and Telecom Italia made the assessment of the following criteria: 1. The potential impact of the application (on efficiency and/or safety) 2. The innovative aspects of the application 3. Compliance of the developed application with the CVIS specification. The application forms and the assessment forms are listed in the D.FOAM.4.5 – Service Submission Contest 2009. Out of 22 high-quality concepts the 11 most promising concepts were invited to continue with the software development and implementation using the FOAM SDK 1.4 (Software development kit). The FOAM SKD 1.4 was made available to the applicants via the FOAM wiki. Application developers had only 6 weeks time to develop the software that was provided to the FOAM wiki. Software developers had to the sign the software license agreement with Makewave which took some time. Makewave provided on-line training session and helpdesk support. Applications had access to the FOAM documentation. 9 out of 11 applicant developers provided software to Makewave for testing and inspectioin. The focus of testing was to verify that the application runs on the Knoflerfish OSGi framework and to assess if they used CALM communications. 

	The figure above provides the test set up. Software developers provided the software to the Host Management Centre of Makewave. From there, the software was provided to the CVIS Host. No communication could be tested due to time constraints and due to the fact that no CALM emulator was made available.
Results	3 out of 9 applications showed poor software development and/or a low level of implementation and therefore were excluded from the gold, silver, bronze awards. 3 applications from companies external to CVIS and 3 companies from the CVIS consortium provided applications with a high level of integration. 4 applications were invited by the CVIS jury to the ITS world congress in Stockholm.  3 out of 9 applications showed poor software development and/or a low level of implementation and therefore were excluded from the gold, silver, bronze awards. 3 applications from companies external to CVIS and 3 companies from the CVIS consortium provided applications with a high level of integration. 4 applications were invited by the CVIS jury to the ITS world congress in Stockholm.

The picture below is a screen dump of the winning application from Halmstad University. Halmstad University's pedestrian crossing warning could substantially improve safety for vulnerable road users such as pedestrians and cyclists.

The pictures below show the price award ceremony during ITS Stockholm and the 4 finalists that received gold, silver, bronze prizes by Juhani Jaaskelainen.

	
Explanations/ practicalities	The application development was limited to FOAM software. COMM software was not tested. POMA and COMO software were not used as the software licenses were not available at that point in time. Make available software licenses agreements took more time than anticipated. This lead to some delay. Some application developers requested access to software from POMA and COMM which was not available at that time.
Conclusions	4 application developers demonstrated in an impressive way that external companies can deploy new applications and services on CVIS infrastructure using the CVIS software. Based on the success at the ITS World Congress, Lodgon announced in October 2009 the DaliReporter as a commercial solution based on the VSN application and the FOAM middleware. Wim de Munck from Lodgon said: *Our visitors have supported us with their votes, awarding us with the Silver prize for our application. As result of the positive feedback during the conference, we have decided to transform this project into an official LodgON product:* ***DaliReporter****.* The winners of the innovation contest were invited to participate at the final showcase at Intertraffic 2010 and Lodgon will present their application.
Verdict	☒ FOAM_-OB2 - Deployment of new applications and services - has adequately been addressed in the project. ☐ FOAM – OB2 has partially been addressed in the project.

	☐ FOAM – OB2 has insufficiently been addressed in the project.

ID	Name	Description
NTVE_T2.c	Interoperability for applications..	An end user can use and access simultaneously multiple relevant services.
Validation manner	o Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM-OB1.1_t.
Test description	CVIS applications and services can be developed, deployed and provisioned on an open end-to-end framework and run-time environment.	
What to measure	Test of different applications (e.g. access control, dangerous goods in CF&F) at one test site (e.g. Gothenburg). Tests of same application (access control) in different test sites (e.g. Gothenburg and Lyon) using core FOAM functionality.	
Expected outcome	Deployment and provisioning of different application at one test site and of same applications in different test sites.	

Validation by testing

ID	AO_FOAM_OB1_t
Actual test setup	The interoperability of the CF&F access control application was tested and validated in Lyon and Turin. The access control application was installed in a test vehicle in Lyon. The car went from Lyon to Turin. The test vehicle should connect to a service centre via 3G communications. The CF&F application for Turin should be downloaded and installed. The vehicle should use the application in a given area in Turin. During 16-18 of February 2010 interoperability test between France and Italian Test site have been performed. The test had the objective to validate the interoperability of application in different contexts. Two types of tests were performed: • Scenario with real RSUs (configuration in Italian TS). • Scenario with one RSU at the entrance of the big bubble then,

<table>
<tr><td></td><td>geofence process (same setup as in French TS)

Test vehicle set up:
• Used the HA from Telecom Bretagne
• Used the FOAM Application Manager Framework as HMI
• POMA for positioning (with EGNOS correction from Thales, and Navteq map-matching)
• FOAM Framework SDK 1.8
• A webcam was present in the vehicle and video was recorded using RTMaps</td></tr>
<tr><td>Results</td><td><u>Test 1 (real RSUs)</u>

Different ways to connect the OBU and RSU were tested: FAST, socket with calm, socket without calm.

A single physical RSU was used, but its configuration was updated at every crossing of the vehicle to play the role of different RSUs (entrance of big bubble, entrance of approach area, entrance of sensitive area and corresponding departures).

The communication exchange hasn't been correctly completed. Possible causes are the presence of other RSUs (Speed Profile) configured on other networks.

In the lab:

The complete process has been validated using FAST. But using sockets, the rules are not downloaded in the approaching phase. Other messages are properly exchanged. Investigating the logs will provide more information.

In the field:

With the vehicle static, using FAST, socket with or without CALM, the exchange was not working completely. By now we suppose this was due to the presence of the other RSUs (Speed Profile) transmitting advertisement messages.

In the OBU logs, we can see that the vehicle tries sometime to connect to the wrong RSU. When the vehicle is moving, the same kind of problem happened.

<u>Test 2 (geofence)</u>

The first phase of the process is the downloading from the RSU of the URL of the Access Control service centre. This has been done correctly. Then the</td></tr>
</table>

	GPS position of the vehicle triggers connections to the service centre over 3G. The MYSQL database of the service centre will be investigated to verify the presence of the registration of the trials. Communication via 3G between the RSUs and the vehicle was possible. The vehicle could ping over IPv6 Mizar network and vice versa. Communication via M5 between the RSU and the vehicle was not possible. The communication bundles shall store logging information to track FAST communication anomalies. The CF&F application was not running as expected mainly for technical reasons but the Application Manager HMI showed information to the driver. The vehicle could ping over IPv6 Mizar network and vice versa. Position recording was performed, but communication was not stable in the field. Recording of RTMAP was available and was used for video log. The fact that France Vehicle is connected to Home Agent from Telecom Bretagne in France was not a problem. More information can be found in the validation report D.CFF.6.2 and in the test site report.
Explanations/ practicalities	• At the France TS the R7 was used. • At the Italian Test site the R8 was used. • Therefore, the France vehicle had to be upgraded from R7 to R8 on during on the first day. Networking aspects were fixed • During the test, the Access Control RSU was configured to not advertise an IPv6 prefix over M5. This is because previous tests have encountered problem when both M5 and 3G IPv6 tunnels were active and one of the two connections goes down. • It was not possible to ping from Mizar or from the vehicle to the RSU host, only the IPv6 tunnel on RSU router was reachable with the “ping” command. The pinging to the RSU host was working at the beginning of the test.
Conclusions	It was the first interoperability test of the access control application between two test sites. As expected there are still a lot of technical problems which

	need to be analysed and addressed in future projects. The usage of different software releases R7/R8 caused most of the problems, as well as the usage CALM M5 between the vehicle and a foreign RSU. The service advertisement of CALM FAST could not be handled by the access control application. Some improvements need to be made in the access control communication layer and the filter of the service advertisement from foreign RSUs / applications needs to be improved.
Verdict	☐ Passed ☒ Partially passed ☐ Failed

Validation by inspection

For validation by inspection, the validation results should be collected as follows.

ID	AO_FOAM_OB1_t
Input documents	Test Site documents - functionality status overview (internal status document) per test site - test site reports D.TS.5.2 = D.CVIS.7.7b –Cooperative Mobility Showcase 2010 - D.CVIS.7.5a/b - Planning and description of ITS World Congress 2009 demonstrations.
Results	All CVIS applications used the JAVA runtime environment and OSGi/Knopflerfish software provided by FOAM. Most applications used additional software components provides by FOAM such as the HMC. CINT used and tested the DDS, CURB and COMO used and tested the Location API, CF&F used and tested the secure communications and the LDT. All 13 CVIS test locations and test sites used the FOAM software. 3 test sites (Sweden, Netherlands, Germany) set up and installed the host management centre software. 3 test sites (Italy, France, UK) used the central host management centre provided by Makewave for service deployment and provisioning. A set of 13 applications has been integrated and tested for public road tours in Stockholm, including 6 CVIS applications from the subprojects CF&F,

	CURB, POMA and COMM. A set of 21 applications has been integrated and tested for public road tours in Amsterdam, including 9 CVIS applications from the subprojects CF&F, CINT, CURB, COMO and applications to show the performance of POMA, FOAM and COMM. CF&F transferred the access control application developed in Turin first to Helmond, then to Stockholm and Amsterdam. The CF&F application parking booking that was developed for the test site London was successfully tranferred to the test sites Helmond, Stockholm and Amsterdam. CINT provided the ghost driver application to Helmond, Stockholm and Amsterdam and the CTA application speed profile was implemented in Amsterdam. CURB showed the priority and micro routing application in Helmond, Dortmund and Amsterdam. CURB successfully transferred the information application from Rotterdam to Amsterdam using two different traffic management centres and different data gateways. COMO provides traffic state calculation (based on loop data and vehicle information) and monitoring. This application was installed at the test site Hessen and the same functionality was transferred (traffic state calculation and monitoring) to Amsterdam. COMO also provided the COMO radar view application to Hessen and Amsterdam to show the location of RSUs and vehicles on a map. In addition, new applications from the application innovation contest (Lodgon, Logica) and native applications (eCall, road charging) were successfully integrated in the public road tour demonstration in Amsterdam. The presentation of the applications used the HMI developed in FOAM (MyService). 10 equipped vehicles showed the CVIS applications to more than 800 visitors in Amsterdam. Meanwhile the first integration of various CVIS application in Helmond took a huge amount of adaptation and software updates, the integration in Stockholm and Amsterdam could focus on the optimisation of the robustness and the overall performance. 21 CVIS applications and 3 SAFESPOT applications run in parallel on the CVIS host. Therefore, the performance of some applications decreased. The application software for the Cooperative Mobility Showcase was

	deployed by the host management centre set up at the Logica premises in Rijswijk. Service centres with legacy data interfaces in Karlsruhe, Turin, Amsterdam and other sites were connected. In the Public Road Tour demonstrations in Helmond and Amsterdam, CVIS also integrated 3 SAFESPOT applications (IRIS, SMAV). CVIS and SAFESPOT use the same router but to avoid interference a separate host was used. While the CVIS platform uses the FOAM software, SAFESPOT is using a different middleware on the host. Interoperability with COOPERS was not achieved as the COOPERS platform didn't allow the installation of external applications and COOPERS was not willing to adopt the COOPERS applications to the CVIS OSGi framework
Contemplation	Integrating 30+ applications from more than 10 suppliers with a connection to 7 different legacy data interfaced proved to be a big challenge. Although, most applications behalved as expected there were also some technical problems due to the following points: • Some applications were developed for a very limited scope, i.e. to test or demonstrate a single concept, and it was never envisioned that the application was going to be used in a larger scenario. • Robustness of the application in relation to transient failures of the i.e. 3G communication link to backend services. In some cases application would allocate important system resources and then do time consuming tasks. If there were unexpected communications latencies this would lock the system for up to 30 seconds. • Some application was developed vertically without taking full advantage of core and domain services. This would duplicate computational work and unnecessary increase CPU and memory requirements. • Some application would consume a lot of memory and CPU resources but still work very well alone on a system, but when you put 3 or 4 of these apps in the same environment it would not work. • Some applications relied on core services (like SQL database system) that were never design in properly, again vertical thinking of application development would in some cases introduce multiple different database systems into the environment. • Some applications relied on 3rd party packages that were not ever meant to be run in a OSGi v4 environment (missing bundle manifest). • Some applications was delivered very, very late in the process (the week before the event), months after the official deadline.

Conclusions	The CVIS software provided by FOAM allows application interoperability between different test sites.
Verdict	☒ FOAM_-OB1 – Application interoperability - has adequately been addressed in the project. ☐ FOAM – OB1 has partially been addressed in the project. ☐ FOAM – OB1 has insufficiently been addressed in the project.

ID	**Name**	**Description**
NTVE_T2.d	Openness	An end user can add new services to the environment without disturbing the existing services.
Validation manner	o Test	Implement two applications, and assure that both applications are running in parallel.
Test description	A scenario will be developed where more than one application will run in parallel. The scenario will activate two major CINT applications (CTA and EDA).	
What to measure	The validation tests as described for CTA and EDA in parallel.	
Expected outcome	The outcome of the validation tests is identical with the outcome of the validation tests when run separately.	

The validation of NTVE_T2.d is fully covered by the results listed in NTVE_T2.b and NTVE_T2.c, and will not be repeated here.

ID	Name	Description
NTVE_T2.f	Interoperability with legacy systems.	CVIS should be interoperable with legacy systems, safeguarding long-term investments.
Validation manner	○ Test	Outside the scope of this project.

ID	Name	Description
NTVE_T2.g	Interoperability between stakeholders/hosts.	The overall CVIS infrastructure leaves an open and flexible way for information to be exchanged between stakeholders (typically being 'hosts')
Validation manner	○ Test	This test will be implemented in FOAM technical validation, and is covered by test FOAM OB1.12b.
Test description	The DDS (Distributed Directory Service) provides a kind of yellow pages for the applications and services available in a geographic area. In an interurban scenario: a vehicle drives from Antwerp to Rotterdam. The DDS information will be provided from a gantry or on a rest area. In an urban scenario: A vehicle approaches an intersection, in four main phases: a) retrieval of services b) submission/registration to the service, c) notification of the service availability d) start of the service itself.	
What to measure	Time for set-up connection, time for information exchange, data rate needed and other technical parameters.	
Expected outcome	CVIS infrastructure leaves an open and flexible way for information to be exchanged between stakeholders	

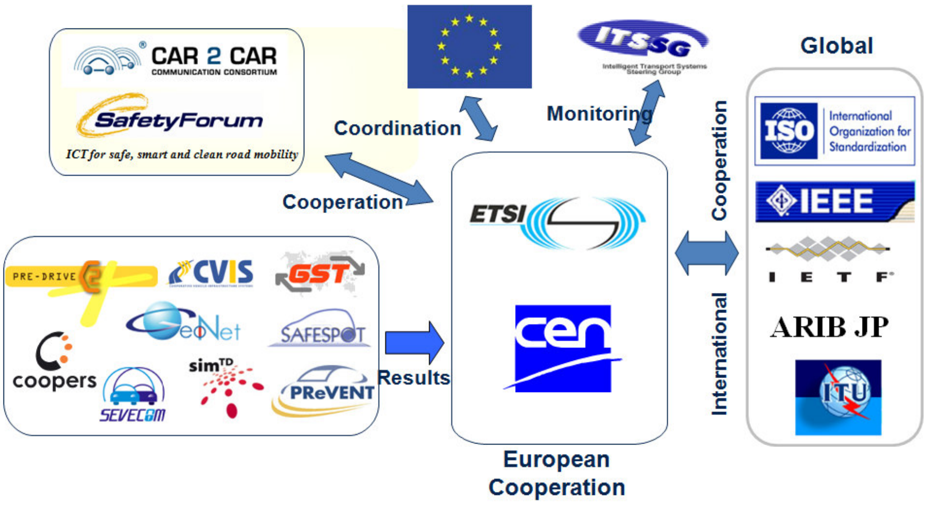
ID	AO_FOAM_OB1.12_b
Testsite	Helmond, Netherlands/Pisa - Firenze, Italy
Date	December 2009, February 2010
Actual test setup	The DDS is part of the FOAM infrastructure that CTA is running on. CTA consists of several components working together to provide the CTA functionality to the individual travelers. The following components use the DDS: • TripStore; registers itself at the DDS to provide a way for other CVIS components to find it; • VehicleTripManager; uses the DDS to locate the TrafficEventService and Tripstore and registers and updates the vehicle relavant attributes in the DDS (destination, location, user and route); • Traffic Event Service; the traffic event service uses the DDS to find the cars that might be interested in a traffic events and registers itself to provide a way for other CVIS components to find it.
Data format (WP6)	not applicable
Results (WP4/WP5)	In the first version of the CTA components dynamic lookup of components via the DDS was not implemented, but the URL's to the necessary services (Tripstore and Traffic Event Service) were hard coded. In later versions, this lookup was implemented to use DDS and it was controllable via configuration parameters whether the DDS should be used to retrieve the services or not. This was also true for the selection of relevant cars by the Traffic Event Service.
Explanations/ Practicalities (WP4/WP5)	In the NL/BE testsite it showed that the services registered properly. The configuration used during CTA validation didn't make use of the DDS lookup, but used the direct URL. At the Italian testsite the CTA in car component was configured to use the DDS that was running at Logica in the Netherlands. The complete CTA

	scenario has successfully been done for several different routes.
Conclusions (WP 6)	The DDS was not completely used for the purpose it was setup in the NL/BE testsite. In the Italian testsite the DDS was used as intended, All tests passed succesfully.
Verdict (WP6)	☐ Passed ☒ Partially passed ☐ Failed
Other remarks	

ID	Name	Description
NTVE_T2.k	Business accessibility.	There are open and published standards that enable oem and suppliers to enter CVIS supply chain.
Validation manner	○ Inspection ○ Deliverable	This test will be implemented in FOAM technical validation, and is covered by test D.CVIS3.2..
Reference document	D.CVIS3.2. (High Level Architecture)	
What to look for	A chapter containing mentioned standards.	

ID	**NTVE_T2.k**
Input documents	D.CVIS.3.2 High level architecture
Results	A specific paragraph listing applicable standards could not be found in the high level architecture document. However, it is verfified that CVIS consortium actively participates in joint CEN/ETSI standardisation activities related to cooperative systems. Since these discussion take much longer than the lifespan of the CVIS project, it cannot be expected to have a list of applicable standards available at this moment. See also contemplation text listed below.
Contemplation	Standardisation is a priority area for the European Commission in the ITS Action Plan in order to achieve European and global ITS co-operation and coordination. Standardisation for Co-operative ITS systems has already been initiated both by ETSI and ISO as well as within other international standards organisations. European standardisation activities to provide standardised solutions for Co-operative ITS services are therefore closely related to the world wide standardisation activities. A draft standardisation mandate on co-operative systems has already been drafted in order to prepare a coherent set of standards, specifications and guidelines to support European Community wide implementation and deployment of Co-operative ITS systems. The Mandate supports the development of technical standards and specifications for Intelligent Transport Systems (ITS) within the European Standards Organisations in order to ensure the deployment and interoperability of Co-operative systems, in particular those operating in the 5 GHz frequency band, within the European Community. CVIS is unique in its capability to interact with the ongoing communications

<table>
<tr><td></td><td>standardizations work. The reason is that the project includes the convenors and editors of all relevant work related to protocols, architecture as well as the core media of GSM, MM, IR and M5 (mobile WIFI). Moreover, they are taking part to all standardization organisations & for and they are contributing actively to the communications standardizations work. These persons are assigned central roles in CVIS that should guarantee good information flow and transparency in both directions. CVIS is also adopting protocols defined by the IETF (Internet Engineering Task Force) from the TCP/IP family (IPv6 protocol suite, NEMO (RFC 3963). CVIS acquired hands-on experience on IPv6 will help improving the standards from the IPv6 family. New features developed by CVIS are being discussed at the IETF through the ISO and ETSI channels.


</td></tr>
<tr><td>Conclusions</td><td>CVIS has given maximum support to standardisation organisations for generating a (minum) set of requirements.</td></tr>
<tr><td>Verdict</td><td>☒ NTVE_T2K has bee properly addressed in the CVIS project.
☐ NTVE_T2K has partially been addressed in the project.
☐ NTVE_T2K has insufficiently been addressed in the project.</td></tr>
</table>

ID	**Name**	**Description**
NTVE_T2.m	Openness to custom made functions.	An OEM will be capable of offering custom-made functions in addition to a CVIS compliant function.
Validation manner	○ Test	This test will be implemented in FOAM technical validation, and is covered by

		test FOAM-OB2.
Test description	Service submission contest" for new CVIS applications in months 36. Application developers with best ideas will receive up to 25.000 funding from CVIS to develop within 3 months CVIS applications.	
What to measure	This application will be tested at least at one CVIS test sites using the CVIS reference platform.	
Expected outcome	Various innovative CVIS applications will be developed. Validation of FOAM SDK by application developers.	

Results are already described above in test NTVE_T2.b (Interoperability) and will not be repeated her.

ID	Name	Description
NTVE_T2.o	Openness to development environment.	When developing CVIS components, an OEM will be free to use its own trusted field of development and the language it uses in that domain.
Validation manner	○ Inspection ○ Deliverable	
Reference document	D.CVIS.3.3. Architecture and system specification.	
What to look for	Are interfaces properly specified, and is it specified how to use these interfaces for different languages.	

ID	**NTVE_T2.o**
Input documents	D.CVIS.3.2 High level architecture
Results	The aim with CVIS is to produce an architecture and specification that is implementation independent, i.e. allow implementation for various client and back-end server technologies. However, for the reference execution environment, FOAM has created a so-called binding to specific technologies in order to create a fully functional system. The binding of choice for a client binding in CVIS is Java / OSGi on top of a Linux operating system.
Contemplation	Anyone implementing a CVIS system is completely free to create new binding to other technologies to implement a CVIS client system, e.g. Linux, .Net or other technologies.
Conclusions	When developing CVIS components, an OEM will be free to use its own trusted field of development and the language it uses in that domain.
Verdict	☒ NTVE_T2K has bee properly addressed in the CVIS project.

	☐ NTVE_T2K has partially been addressed in the project. ☐ NTVE_T2K has insufficiently been addressed in the project.

ID	Name	Description
NTVE_T2.p	Openness to hardware vendors	When an OEM obtains CVIS components from other vendors, these will work directly (plug-and-play).
Validation manner	○ Test	Not in scope of the CVIS project.

5.2. Results interoperability questionnaire SP leaders.

This sections shows the aggregated results and conclusions form the interoperability quesionnaires.

Appendix 5 shows the aggregated results of the questionnaires that are sent to the SP leaders in order to investigate to what extend the individual subprojects did indeed use the results that were developed other subprojects, and what difficulties were experienced during integrating the different services.

The most important findings of the received questionnaire are given below.

1. **FOAM** is based on results that were developed in in EU funded GST project (reuse of secure mechanism as well as the payment mechanism. FOAM extended the GST opens systems service platform.
2. FOAM uses intensively the services/modules developed by COMM. The direct dependability caused some problems in the project flow due to delays in developing some COMM components.
3. Technical integration of COMM and FOAM did not cause major problems.
4. The LDM could not be tested in the project lifetime of FOAM.
5. **POMA** did not fill in the questionnaire.
6. **COMO** has direct links with COMM, FOAM, POMA and SAFESPOT (LDM).
7. COMO experienced significant problems setting up de IPv6 comunication.
8. Delays in the LDM development (SAFESPOT) caused delays in COMO project.
9. COMO and POMA did not share common interfaces for map matching and positioning.
10. COMO reported poor documentation and bad testing of external software components that were required for COMO services. This led to an ‘enormous

amount of integration work'.

11. **CINT** reported direct links with POMA, FOAM and COMM.
12. With respect to COMM, it was identified that the (POMA) AGORAC encoding/decoding mechanism did not produce full interoperable systems from Navteq and Tele Atlas. Integrating these modules in CINT caused major problems.
13. In line with the remarks 10 (COMO), CINT reported: "Developing an application when developing core technology simultaneously has not worked very well in CVIS. Some components of the CORE technology were still developing when the applications should already have been finished."
14. The item listed above also resulted in multiple CINT-EDA implementations.
15. CINT did not use COMO modules (although originally intended to do so).
16. Integrating FOAM with CINT did not cause major problems.
17. **CURB** reported direct links with COMM, FOAM, POMA, COMO.
18. The COMO-LDM did not offer the required functionality for CURB. Therefore, CURB developed their own software bundle.
19. POMA RT-MAPS was considered too complicated and sometimes unstable. However one application (strategic routing) reported a very smooth integration process (note however, that this application was developed by people that were deeply involved in COMO and POMA subprojects).
20. Different applications reported totally different about COMM CALM Fast:
    - CURB priority application: "CALM FAST has shown to be effective and easy to use without much configuration."
    - CURB micro routing: "Usage of the CALM/FAST bundles and the serviceadvertisements is really hard. There is very little documentation available on how it should work and it doesn't allow to be used in a well-structured program."

    This can be explained by the communication complexity of the applications. For CURB priority application only a single context awareness message (CAM) needs to be transmitted, whilst micro routing has a more complicated communication need.
21. IPv6 caused severe problems, especially in Italy.
22. **CF&F** reported direct links with COMM, FOAM, POMA and COMO.
23. CF&F reported difficulties using FOAM authentication and authorization bundles.
24. In Italy, significant problems with M5 communications were experienced.
25. POMA positioning and geofencing modules were easy to use for CF&F.

Overal conclusion on SP level interoperability:

In the CVIS project, technologies have been researched and developed, and an attempts has been made to make these technologies available at an early stage, so subprojects can optimally profit from each others development effort.

It can be concluded that CVIS managed to deliver a basic set of software and hardware modules that can be used for developing cooperative applications.

However, from the questionnaire it is becomes clear that although several software / hardware modules have been made available by the different subprojects, the modules are in most the cases still in the stage of 'proof-of-concept', and not mature enough for being used in commercial applications.

For achieving interoperability at SP level it is crucial that software modules are well tested, documented and stable. Several subprojects reported that the approach where core technologies and applications were being developed in parallel did not work. It resulted in prototypes that were not properly tested and poorly documented, but made available under the pressure of the needs of the higher level subprojects. In other words, application SPs depend directly on the availability of the core technologies SPs modules, and therefore put much pressure on the core technology SPs to deliver their modules as early as possible.

As a consequence, some subprojects now have multiple implementations of (more or less) identical functionalities (CINT-EDA), or re-implemented functions that originally should have been delivered by core technology subprojects (CURB for COMO LDM).

With respect to COMM, it becomes clear that despite the fact that CALM FAST appears to offer the proper communication mechanisms, for complex applications it was not straightforward using it.

With respect to FOAM the application subprojects reported generally positively.

A clear conclusion about the interoperability of POMA functionality is difficult to supply since totally differing responses have been received.

The modules of COMO were insufficiently used by application subprojects (resons: late availability and poor stability).

5.3. Results interoperability questionnaires EU projects.

The results of the interoperability questionnaire that has been sent to key persons involved in organising major events (i.e. Helmond event, ITS Stockholm, and the cooperative mobility showcase event) are shown in appendix 6.

The following conclusions for interoperability between CVIS and SAFESPOT can be drawn based on the inputs received:

- CVIS and SAFESPOT can run on the same hardware platform.
- However different communication architectures are developed for CVIS and SAFESPOT, requiring special tools for exchanging data.
- CVIS and SAFESPOT use the same CALM networking stack.
- CVIS and SAFESPOT application can in principle run in parallel on one hardware platform.
- It has been demonstrated that CVIS and SAFESPOT can share routers.

- It has been demonstrated that by means of a cooperative awareness message (CAM), beaconing of CVIS and SAFESPOT can be integrated.

The following conclusions for interoperability between CVIS and COOPERS can be drawn based on the inputs received:

- Interoperability between CVIS and COOPERS has hardly been realised.

5.4. Results openness questionnaires

Appendix 7 contains the modules the SP leaders reported for the subproject including the associated licenses.

The self assessement is aggregated into one table and shown below.

SP	Self assessement on open source requirement.	Conclusion
COMM	"23 software packages were created or modified in COMM during the project, of which 3 are partially or fully closed source. Therefore COMM fully met the 80% open source requirement. "	Passed
FOAM	As the core CVIS FOAM execution environment (Knopflerfish OSGi) is open source, at least 80% of the FOAM software is "open source" (as defined at http://www.opensource.org/docs/definition.php). For the implementation of the specific CVIS FOAM API:s, FOAM partners have provided both open source implementations e.g. CALM, as well as binary implementations, e.g. DDS, HMC. The binary implementations, i.e. no source code, are free to use within the the scope of the CVIS project for all CVIS partners as well as external parties. The CVIS FOAM test suite, used in the Beta stage testing, is provided open source under a BSD license agreement.	Partially passed.
POMA	No input received.	---
COMO	No self assessement received.	---

CINT	We cannot state that the CINT resources are 80% open source. It is a matter of counting lines. If you add Linux (open source) and OSGI/Java plus the bundles in CINT , which are open source, we will reach at least 60% open source.	Failed.
CURB	A large number of components are open source using the BSD license agreement. However, also a significant number of proprietary components exist (due to high number of applications build as extension on existing proprietary systems). Conclusion: CURB resources are for almost 50% open source.	Failed
CF&F	The reason for the low extent of open source software generated by the SP itself is that the applications built, are either based on pre-existing proprietary software or built for a specific proprietary platform. However the data exchange protocols and the interfaces between the different systems has been set as open source.	Failed

Looking to the table above, as well as appendix 7, we can only draw the following conclusions:

- CVIS is build on Ubuntu Linux, which is an open source operating system.
- FOAM is using the Knopflerfish service platform, that is as well open source.
- Above mentioned components were not developed in the CVIS project.
- For the components that were developed during the CVIS lifespan, only COMM is considered to have passed the 80% criterion. From this subprojets the vast majority of the components are open source.
- All other subprojets did, individually, not fulfil the open source requirement on itself.
- Taking into consideration the operating system and the Knopflerfish service platform, the percentage of open source software significantly increases. Whether is has reached the 80% criterion cannot be measured. However, it can be concluded that when these components are included in the measurement, the vast majority of the CVIS software is under open source license.

Therefore the final conclusion about CVIS high level objective 2 is: PARTIALLY PASSED.

6. Conclusions

Openness and interoperability are major issues in the CVIS project. Early in the project a first version of a document addressing definitions and requirements related to openness and interoperability document has derived. Based on these definitions and requirements, a non-technical validation list has been generated, containing items that should have been addressed during the CVIS project.

The non-technical validation elements that are identified conclude to:

ID	Description	Conclusion
a	An end user can trust that CVIS will appear to be a performing system over the years, functioning as an extendable package of solutions for his ITS needs.	N.A. (can only be tested after the projects).
b	An end user can expect CVIS to provide a service environment that allows the incorporation of extra services from any organisation that supports CVIS.	Fully reached.
c	An end user can use and access simultaneously multiple relevant services.	Fully reached.
d	An end user can add new services to the environment without disturbing the existing services.	Fully reached.
f	CVIS should be interoperable with legacy systems, safeguarding long-term investments.	N.A.
g	The overall CVIS infrastructure leaves an open and flexible way for information to be exchanged between stakeholders (typically being 'hosts')	Partially passed.
k	There are open and published standards that enable oem and suppliers to enter CVIS supply chain	Not reached, but sufficiently addressed.
m	An OEM will be capable of offering custom-made functions in addition to a CVIS compliant function.	Fully reached.
o	When developing CVIS components, an OEM will be free to use its own trusted field of development and the language it uses in that domain.	Fully reached.
p	When an OEM obtains CVIS components from other vendors, these will work directly (plug-and-play).	Outside scope of this project.

Furthermore it can be concluded that:

CVIS managed to deliver a basic set of software and hardware modules that can be used for developing cooperative applications.

Several software / hardware modules that have been made available by the different subprojects, are in many cases still in the stage of ‘proof-of-concept’.

The situation where all subprojects developed technologies in parallel (core-technologies as well as applications) limited the level to which interoperability has been reached.

And for interoperability with other EU projects:

- CVIS and SAFESPOT can share hardware and software resources to a certain extend.
- CVIS and COOPERS are hardly interoperable.

On the open source requirement that has been formulated in CVIS-OB2, it can be concluded that:

- Taking into consideration the operating system and the Knopflerfish service platform, it can be concluded that, the vast majority of the CVIS software is under open source license. However, many of the modules that were developed during the project lifespan are NOT open source.

Appendix 1: SP leader interoperability questionnaire

SP name	e.g. COMM
SP leader	e.g. Erik Olssen

At the final implementation of you applications, you probably used software components that were developed in other subprojects. In the table below, please indicate for each application that you developed whether you used software from other SPs. **Copy as many tables as you have applications.**

<table>
<tr><td>Subproject</td><td colspan="4">e.g. CURB*)</td></tr>
<tr><td>Application</td><td colspan="4">e.g. Flexible bus lane</td></tr>
<tr><td colspan="5">This application uses components that are part of:</td></tr>
<tr><td>SP</td><td>Component/ Module</td><td>Purpose</td><td>Remarks</td><td>Ease of integration**)</td></tr>
<tr><td>e.g.POMA (1)</td><td>WLAN positioning</td><td>Measuring position of vehicles.</td><td>Alternative for GPS.</td><td>5 (easy, no problems)</td></tr>
<tr><td>COMM</td><td>IR</td><td>Connect Bus to RSU.</td><td>…</td><td>…</td></tr>
<tr><td>FOAM</td><td>SDK XYZ</td><td>Application management</td><td>…</td><td>…</td></tr>
<tr><td>FOAM</td><td>HMC</td><td>Installing application</td><td>…</td><td>…</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>

*) The example is just a theoretical one, that is not reflecting the real situation

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Below you have the possibilities to specify lessons learned related to software integration and interoperability. Please really spend some time on this. It is important for future projects, since it may define the area where future projects should focus on.

Lessons learned (use as much space as you need).

Appendix 2: Interoperability questionnaire CVIS - SAFESPOT

Were CVIS and SAFESPOT applications sharing the same demonstration area at your event?

YES	☐	NO	☐

If yes, what applications were demonstrated in this shared area? (add rows if required)

Project	Appliction	Short description
CVIS	App1	…
	App2	…
	App3	…
SAFESPOT	Appx	…
	Appy	…
	Appz	…
	…	…

Were (some of) these application demonstrated at the same time?

YES	☐	NO	☐

Were CVIS and SAFESPOT sharing hardware and/or software resources of the RSU? Please elaborate. Use as much space as you need.

Were CVIS and SAFESPOT sharing hardware and/or software resources of the OBU? Please elaborate.

Did you experience difficulties when demonstrating CVIS in parallel of SAFESPOT? Please elaborate.

How would you conclude about the interoperability between CVIS and SAFESPOT?

Appendix 3: Interoperability questionnaire CVIS - Coopers

Were CVIS and Coopers applications sharing the same demonstration area at your event?

YES ☐	NO ☐

If yes, what applications were demonstrated in this shared area? (add rows if required)

Project	Application	Short description
CVIS	App1	…
	App2	…
	App3	…
Coopers	Appx	…
	Appy	…
	Appz	…
	…	…

Were (some of) these application demonstrated at the same time?

YES ☐	NO ☐

Were CVIS and Coopers sharing hardware and/or software resources of the RSU? Please elaborate. Use as much space as you need.

Were CVIS and Coopers sharing hardware and/or software resources of the OBU? Please elaborate.

Did you experience difficulties when demonstrating CVIS in parallel of Coopers? Please elaborate.

How would you conclude about the interoperability between CVIS and Coopers?

Appendix 5: Aggregated results interoperability questionnaire SP leaders.

COMM

COMM supplies the lowest level of the CVIS architecture. It supplies services that will (via FOAM) be used by other SPs. The SP level interoperability is towards COMM is contained in the outcomes of the questionnaire for the other SPs.

FOAM

SP name	FOAM
SP leader	Peter Christ

Interopability with earlier EU funded projects.

Project	GST			
Application	FOAM reused the secure mechanism and the payment mechanism from GST. he Service Platform provided by FOAM is an extension of the GST Open Systems service platform [GST.OS.3.1].			
This application uses components that are part of GST:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
GST	SEC	The security mechanisms provided by FOAM is an extension of the GST Security mechanisms [GST.SEC.3.1].	The specification worked fine, but it came out the GST didn't made a full implementation	3

			of the he security component	
GST	PAY	The payment mechanisms provided by FOAM shall be based on the GST Service Payment specification [GST.S-PAY.3.1]	Not implemented	
GST	OS	The concept of the LDT was developed as GST vehicle tree, improved and tested for cooperative systems in FOAM.		4
GST	OS	GST defined the service deployment and provisioning [GST.OS.3.1]. The concept includes the OMA Device Management for the usage in vehicles. FOAM extended the concept to cooperative systems.	The host management centre provides more functionalities the GST	3
GST	OS	FOAM reused parts of the gateway subsystem, an integral part of the GST OS Telematic Control Unit, for the implementation of the host management centre (HMC)		3

Application interoperability

Subproject	COMM/COMO
Application	The hardware and operating system of the host subsystem was developed by COMM. The communication (or router) subsystem was developed by COMM.
This application uses components that are part of GST:	

SP	Component/ Module	Purpose	Remarks	Ease of integration
COMM	OS	COMM decided to use a Linux OS (Ubuntu) and a standard car PC (Plug-in) for the host and router respectively.	Most FOAM components could be tested on a standard PC.	4
COMM	CALM connection	The COMM subproject provided the software for the communication connector	Several delays due to late implementation in COMM.	3
COMM	CALM FAST	Broadcast manager was developed to support applications with the possibility to publish data through a broadcast mechanism, and subscribe to the reception of these data. The broadcast mechanism was developed in COMM based on CALM FAST.	Several delays due to late implementation in COMM.	3
COMO	LDM interface	The LDM is a SAFESPOT concept, representing the traffic situation on the road network in the vicinity of the CVIS Unit. The interface to the Local Dynamic Map developed in SAFESPOT was finally implemented by the COMO SP.	Not implemented and tested together with FOAM	

Lessons learnt:

The implementation of the communication facilities was done by the COMM subproject. COMM experienced several delays in hardware and software development which had an impact on the FOAM tests. Meanwhile, FOAM components could be implemented on the host in the lab already in late 2008, the final testing of the communications components using M5 and IR could only take place in January 2010.

IR was not tested but the CF&F applications that used FOAM as the middleware was successfully tested at the London test site.

The LDM component was developed in SAFESPOT which caused delays mainly for the COMO subproject and additional work to provide a work around. Therefore, the final integration and testing of the LDM together with FOAM was not possible within the project lifetime.

POMA

No input received.

COMO

SP name	COMO
SP leader	Matthias Mann

Subproject	COMO			
Application	COMO API			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
Safespot	LDM	Interoperability with Safespot by use of the SAFESPOT – Local Dynamic Map (LDM)	LDM was delayed, APIs provided by Navteq and Tele Atlas are inconsistent	4

FOAM	SDK 1.8	Application integration	Integration of COMO functionality into CVIS SDK	2

Subproject	COMO			
Application	e/xFCD			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
POMA	Positioning	Receive accurate vehicle positions.	No common interfaces (e.g. JSR 179)	3
POMA	Map Matching	Map matching of vehicle positions to LDM	No common interfaces (e.g. JSR 179)	3
COMM	3G	Transmission of FCD Events to Service Centre	Main difficulties were caused by IPv6 Issues and integration of supported 3G modems	4
FOAM	SDK 1.8	Application integration	Integration of COMO functionality into CVIS SDK	2

Subproject	COMO			
Application	Vehicle Beaconing			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
POMA	Positioning	Receive accurate vehicle positions.	No common interfaces (e.g. JSR 179)	3
POMA	Map Matching	Map matching of vehicle positions to LDM	No common interfaces (e.g. JSR 179)	3
COMM	CALM Fast	Transmission of vehicle beaconing		3
COMM	3G	Transmission of vehicle beaconing (3G beacon) to Service Centre	Main difficulties were caused by IPv6 Issues and integration of supported 3G modems	4
FOAM	SDK 1.8	Application integration	Integration of COMO functionality into CVIS SDK	2

Subproject	COMO
Application	Map Matching

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration
COMM	CALM Fast	Reception of vehicle beaconing		3
FOAM	SDK 1.8	Application integration	Integration of COMO functionality into CVIS SDK	2

Subproject	COMO
Application	Monitoring (infrastructure and center traffic state calculation)

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration
COMM	CALM Fast	Reception of vehicle beaconing		3
COMM	3G	Reception of FCD Events in Service Centre	Main difficulties were caused by IPv6 Issues and integration of supported 3G modems	4

Subproject	COMO			
Application	Driver Awareness (bring TMC info to driver)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
COMM	CALM Fast	Transmission of VMS information from RSU to vehicle		3
COMM	3G	Transmission of VMS information from centre to RSU	Main difficulties were caused by IPv6 Issues and integration of supported 3G modems	4

Subproject	COMO			
Application	Radar View (map display of co-operative vehicles within CVIS vehicle)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**

				**)
COMM	CALM Fast	Reception of vehicle beaconing		3
CURB	PTV Routing Client	Basic functionality for moving map and display of additional layer		3

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Lessons learned (use as much space as you need).

Especially in the beginning a lot of software was delivered with poor documentation and not well tested. This resulted in an enormous amount of integration work (a lot of loops were necessary in order to make things run). The co-operation with SAFESPOT caused delays as the LDM development was delayed. Therefore a mitigation plan had to be designed and was implemented. Lack of IPv6 expertise did defer the proper testing of the applications as it took much more time than expected to setup a running IPv6 environment.

CINT

SP name	CINT
SP leader	Marcel Konijn

Subproject	CINT
Applicatio n	Cooperative Travel Assistance

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration **)
POMA	Agora-C encoder/decoder library Navteq	Unified way of transferring traffic events using a standard and agreed way between map providers	No full interoperability between Tele Atlas and Navteq	2
POMA	Agora-C encoding/decoding	Unified way of transferring traffic events using a standard and agreed way between map providers	No full interoperability between Tele Atlas and Navteq	2
POMA	Hybrid position SW			4
POMA	EGNOS SW	Accurate positioning (lane level matching)	Not really a CINT application	4
POMA	Map-Cache for POMA map-matching SW	Positioning based on GPS		4
POMA	RTMaps SW			4
FOAM	Knopflerfish OSGI Framework	Open service environment		3
FOAM	Management Agent Bundle	Open service environment		
FOAM	CALM Service Implementation	Open service environment		
FOAM	Distributed Directory Service (DDS) Client implementation		Not really used	NA
COMM	CVIS OS kernel	Basic communication modules		
COMM	Mobile IPv6 daemon	Basic communication modules		
COMM	CALM Management Daemon	Basic communication modules		

CINT	WebPortal –1.2.2	Component of CINT - CTA		NA
CINT	TripStore – 1.2.3	„		NA
CINT	Dynamic Routing Service – 1.2.3	„		NA
CINT	AgoraC library	„		NA
CINT	TripPlanManager - 1.0.0	„		NA
CINT	VehicleTripManager -	„		NA
CINT	Navigator -	„		NA
CINT	MapMatcher -	„		NA
CINT	TrafficEvent – 1.0.0	„		NA
CINT	TravelTimeManager 1.0.2		TS-NL Specific	NA
CINT	TrafficIntegrator – 1.0.4		TS-NL Specific	NA
CINT	Traffic Event Server – 1.00		TS-NL Specific	NA
CINT	OnTripTrafficState-Updater		TS-Italy Specific	NA
CINT	TrafficIntegrator		TS-Italy Specific	NA

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Lessons learned (use as much space as you need).

Developing an application when developing core technology simultaneously has not worked very well in CVIS. Some components of the CORE technology were still developing when the applications should already have been finished. New releases of the core platform were made in the last stage of the project. Moreover some developments in COMM are still ongoing when the CINT subproject already closed its development activities. Next time it would be better to fix the base and start building application on this base.

Another lesson: there should be a much stricter directive to use the same software bundles

and architecture. At present many applications chose their own implementation. This is obvious, because there was no standard. COMO was supposed to be used, but has not been delivered in time to CINT .

Subproject	CINT			
Application	Enhanced Driver Assistance (EDA)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration **)**
POMA	Agora-C encoder/decoder library Navteq	Unified way of transferring traffic events using a standard and agreed way between map providers	No full interoperability between Tele Atlas and Navteq	2
POMA	Agora-C encoding/decoding	Unified way of transferring traffic events using a standard and agreed way between map providers	No full interoperability between Tele Atlas and Navteq	2
POMA	Hybrid position SW			4
POMA	EGNOS SW	Accurate positioning (lane level matching)	Not really a CINT application	4
POMA	Map-Cache for POMA map-matching SW	Positioning based on GPS		4
POMA	RTMaps SW			4
FOAM	Knopflerfish OSGI Framework	Open service environment		3
FOAM	Management Agent Bundle	Open service environment		(3)
FOAM	CALM Service Implementation	Open service environment		(3)

COMM	CVIS OS kernel	Basic communication modules		
COMM	Mobile IPv6 daemon	Basic communication modules		
COMM	CALM Management Daemon	Basic communication modules		

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Lessons learned (use as much space as you need).

EDA has started to be developed in two different locations , namely France and Sweden. Originally it was planned to create a single core of applications which could be adapted to the local situation in France and Sweden. However due to the delays in developing core technologies and due to the lack of available tools at the time of developing the applications, EDA has focused on local implementations. The implementation in the local environments in Sweden and France drove the technical developments.

EDA was the first application which was showed in CVIS. Already during the technical review of 2008, EDA was demonstrated in Gothenburg live. It was using a Road side unit which communicated over M5 with the vehicle and a traffic control centre at the Lindholm Science Park.

By the time that the CORE technology was available, it was too late to integrate the French and Swedish developments into a single applications. However the applications itself were successfully demonstrated during the ITS world conference in Stockholm. The EDA Ghost driver applications was also demonstrated in Helmond 2009 and during the final Showcase event in Amsterdam.

The lessons learnt from EDA development:

The ideas of the applications are very viable. Often a dynamic speed limit is mentioned as the most desired and logical cooperative application on the inter-urban network. The business case for the application is easily made. A dynamic speed limit application saves a huge investment in existing road side equipment. The only thing which will be need in future is a bi-directional communication unit over M5.

The most work is <u>not</u> in building the application, but in adopting the environment (legacy systems) to run the application in the correct way. If the engineers are familiar with the CVIS environment, including all underlying technology, it proves to be relatively easy to create a new application. The work which will be going into modifying the legacy systems, is very specific and veries per country or even region.

Developing an application when developing core technology simultaneously has not worked

very well in EDA. Some technology (like COMO) has been developed, but has not been used in the application groups. Next time we should be based on working technology when starting to build the application. A common architecture and strict guidance in using the correct Java bundles would also help the success of EDA.

CURB

Subproject	CURB			
Application	Priority Application (Siemens)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
POMA	WLAN positioning	Measuring position of vehicles		3
COMM	WLAN / beaconing	Connecting car to RSU		4
FOAM	SDK (OSGI framework)	Application environment		5
FOAM	HMC	Controlling / Visualization of application		4
COMO	LDM	Gathering vehicle positions		3
	DDM	Get vehicle position		4
	Map matching	Process vehicle position		4

Below you have the possibilities to specify lessons learned related to software integration and interoperability.

Lessons learned (use as much space as you need).

Subproject	CURB			
Application	Priority Application (Peek)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
COMM	**Linux OS**	**Basic Operating System and drivers**		
COMM	**CALM**	**Communication between RSU and OBU**	**FAST only**	
FOAM	**SDK**	**Knopflerfish runtime environment**		
POMA	**RTmaps**	**Collect position from GPS receiver**	**Vehicle**	
COMO	**Beaconing**	**Vehicle beaconing**	**Vehicle**	
COMO	**DDM**	**Inter-bundle communication**	**RSU**	
COMO	**Mapmatching**	**Map-matching of incoming beacons**	**RSU**	
COMO	**LDM**	**Local Dynamic map**	**RSU, Navteq**	
FOAM	**LDT**	**Local Device Tree**	**Vehicle**	

Below you have the possibilities to specify lessons learned related to software integration and interoperability.

Lessons learned (use as much space as you need).

Even though the LDM is used we experienced that it does not suit the application's needs very well. Therefore a bundle has been created that supports the use of long 'reference tracks' with associated objects like stop lines. See the ITS world Congress 2009 paper 3485, "Map based applications The reference track approach". For a vehicle which just must send out GPS based beacons the use of RTmaps is experienced to be too complicated and sometimes unstable. Map matching incoming beacons onto the LDM has given a lot of problems during demonstrations; for practical use the process involved is too complex and error-prone. The firmware used for the M5 radio cards is closed source, which withstands the use outside the CVIS project. CALM FAST has shown to be effective and easy to use without much configuration. The M5 radio range has shown to be at least 400 meters, which is very good.

Subproject	CURB
Application	Speed Profile Application (Mizar, CRF)

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration
FOAM	SDK1.8	Application Hosting		
FOAM	CalmConnectionFactory, CalmConnector	Communication (Utp, Tcp, Fast)		
COMM	M5 Card	Communication via ipv6 over M5 and FAST		
COMM	3g driver	communication		Use of USB 3G Modem
COMM	HA	For ipv6 configuration		
external	OpenVN	For ipv6 tunnel		

Lessons learned (use as much space as you need).

Ipv6 over Calm

Major problems during test for handover between RSUs. Multiple version of the FOAM has been required and provided.

Physical Antenna Placement.

Antennas location is critical. Presence of vehicle in between can affect heavily the communication range.

RSU tailoring of application HMI (Application manager and Desktop) has been removed to have the ability to reboot from remote Connection with UTC Maintenance activities of UTC affected on field test. During real test on field the presence of various version of UTC hw and firmware . Time Synchronization Time synchronization solution had to be implemented specifically. General Framework lack of the function, which is fundamental for the application. Debugging of application Debugging of application is not easy. PCI board connector Solution Weak connector, better fast plug connectors. Unique Hw on the Vehicle It would be better to have a unique HW in the vehicle, as implemented for the RSU. Serial connection on Java Weak serial support in Java has created some issue on the integration of the application inside the vehicle 3g Card PCMCIA and Onda Driver PCMCIA (Option) did not start or fail after some time. Onda USB 3g modem, sometime required to start manually (in some hw configuration it has to be started always manually).

Subproject	CURB

Application	Traffic Control Assessment (Mizar)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**

Lessons learned (use as much space as you need).

Analysis of the data

The application has been implemented in the centre and on the RSUs. The information is collected in the Central DB and information is accessible on the DB. An HMI to display the data would be needed to allow user to make analysis.

Subproject	CURB			
Application	Flexible Bus lane Application (Thetis)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
COMM	3G	Connection among CVIS vehicles and RSUs Connection among Public Transport	3G module had to be tipped and tricked because of no native module integration.	2) Hard to integrate with sw module

		vehicles and public transport management service (PTMS) Connection among RSU and PTMS	Communications with HA instable	
COMM	CALM-FAST	Connection among CVIS vehicles and RSU	A little tricky to setup in Osgi environment, very poor initial specifications	4) broadcast not always spread
FOAM	HMC	Application installer	-	5) easy
FOAM	SDK	Application management	-	5) easy
CURB	Strategic Routing Application	Routing and positioning of the CVIS vehicles with list of bus lanes	Very good support by PTV	4) some hidden configurations but good PTV support

Lessons learned (use as much space as you need).

In detail we point the following learned lesson within this application.

Before to start an application with such this integration among other applications and modules and technologies, the base modules and technologies should be well specified, strong enough and stable to avoid wasting time trying to make them work.

Subproject	CURB
Application	Information Application (Vialis)

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration
COMM	M5	Connect Car to RSU.	None	Good
FOAM	SDK 1.9 beta 1	Application management	None	Good

Lessons learned (use as much space as you need).

Steep learning curve.

Little support from other CVIS partners.

Good use of standards.

Hardware could be optimized / standardized.

Software distribution unnecessary complex.

Subproject	CURB
Application	Routing Application (Mizar)

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration
FOAM	SDK1.7	Application Hosting		
FOAM	HTTP/AXIS	Publication of service		Some conflict between use of SOAP services and provision of SOAP services

Lessons learned (use as much space as you need).

Subproject	CURB
Application	Strategic Routing Application (PTV)
This application uses components that are part of:	

SP	Component/ Module	Purpose	Remarks	Ease of integration
POMA	RT Maps	Measuring position of vehicles.	GPS	5 (easy, no problems) (2)

COMM	3G	For routing request to routing service centre	It was very difficult to achieve compatible 3G hardware	1
FOAM	SDK 1.7	CURB Client software was integrated into SDK		3
FOAM	SDK 1.7	Application management	CURB client was integrated in environment provided by application management	3
COMO	EFCD	Traffic information for routing service centre (vehicle providing traffic information)		4
COMO	Monitoring	Traffic information for routing service centre	Based on available incident information strategies were activated or deactivated in routing service centre	**3**

Below you have the possibilities to specify lessons learned related to software integration and interoperability.

Lessons learned (use as much space as you need).

Most of the time was spend for POMA integration. In the beginning interfaces were not well defined, examples (how to) were missing, configuration (including 3g communication) was quite a challenge.

Subproject	CURB			
Application	Micro-Routing Application (Peek)			
This application uses components that are part of:				
SP	**Component/ Module**	**Purpose**	**Remarks**	**Ease of integration**
Safespot	**Org.safespot.trail**	**Determine distance to stopline**		**easy**
Java	**Javax.microedition.io**	**Datagram connection for FAST**		**easy**
Calm	**Calmconnector_api**	**Communications/service advertisement**		**Very hard**
Como	**VehicleBeaconingData**	**Extract beacons from vehicles to supply to the trail package**		**easy**

Lessons learned (use as much space as you need).

Usage of the CALM/FAST bundles and the serviceadvertisements is really hard. There is very little documentation available on how it should work and it doesn't allow to be used in a well-structured program. Everything has to be stuffed in the activator class and it's unclear how to trigger the rest of the applications code when something happens in the communication. Normal Java sockets work much easier and it shouldn't have been so hard to make something structured in a similar intuitive way.

POMA was also hard to use and therefore the Safespot trail was used instead to determine vehicles positions.

CF&F

SP name	CF&F
SP leader	Niclas Nygren

Subproject	CF&F
Application	Dangerous Goods

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration **)
FOAM	SDK 1.7	Application management		5
COMM	CALM-M5	Connect truck to RSU over FAST		5
COMM	CALM-3G	Connect truck/RSU to backoffice systems		5
POMA	Positioning	Measuring position of vehicles.		5
POMA	Geofencing	Entrance / Departure detection		5
COMO	PTV CVIS-SDK	navigation services		3
FOAM	HMC	Installing application		4
FOAM	A&A modules	For secure communication to the TMC		2

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Subproject	CF&F
Application	Parking Zone
This application uses components that are part of:	

SP	Component/ Module	Purpose	Remarks	Ease of integration **)
POMA	Positioning	Measuring position of vehicles.		5
POMA	Geofencing	Entrance / Departure detection		5
COMM	CALM-IR	Connect truck to RSU over FAST.	For London TS	3
COMM	CALM-M5	Connect truck to RSU over FAST	TS France & NL	4
COMM	CALM-3G	Connect truck/RSU to backoffice systems		3
FOAM	SDK 1.7	Application management		5
COMO	Routing component	ETA calculation		4
External component	IRID	Image recognition system to detect illegal vehicles	For TS London	2
External component	Variable Message Sign	Display information to driver at parking entrance	TS France & NL	5

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Subproject	CF&F

Application	Access Control

This application uses components that are part of:

SP	Component/ Module	Purpose	Remarks	Ease of integration **)
POMA	Positioning	Measuring position of vehicles.		5
POMA	Geofencing	Entrance / Departure detection		5
FOAM	LDT	for vehicle data access		
FOAM	SDK 1.7	Application management		5
COMM	CALM-3G	Connect vehicle/RSU to backoffice systems		1-CRF
COMM	CALM-M5	For communication with road side units.	For the France, Netherlands/ Belgium and Italian test sites.	1-CRF

**) Please indicate how complicated it was to integrate this module 1 = very complicated; 5 = easy.

Below you have the possibilities to specify lessons learned related to software integration and interoperability. Please really spend some time on this. It is important for future projects, since it may define the area where future projects should focus on.

Lessons learned (use as much space as you need).

3G:

The support for the different 3G cards available at TS was poor at the beginning but increased drastically. Still an auto-reconnection script was missing, which was the source of some problems during testing. CVIS R8 introduced such a script.

CALM FAST:

Some problems have been encountered with FAST and threads. This was solved by FOAM SDK 1.7

IRID system for Parking Zones in London:

In real situation, the IRID system fails to detect some vehicle entrances / departures which caused the application to enter in unexpected states for the detection of illegal vehicles. The application has been corrected to support better IRID's unreliability.

Integration:

Before proceeding with the integration of the application SW, communication HW and SW should be delivered totally debugged and integrated in the framework. Hw configuration should be done by expert people before deliver the items

Access Control

3G:

TI asked explicitly support for 3G USB cards (ONDA 505UP), it has been included in the list of supported cards. Still an auto-reconnection script was missing in CVIS R7, which was the source of some problems during testing. CVIS R8 introduced such a script but it proved not working for the ONDA card. A manual procedure must be executed to make the connection available.

CALM:

Some problems have been encountered with CALM and threads. Not all have been solved by FOAM SDK 1.7 and CVIS R8.

In particular, when useing the http protocol, the CalmConnector produces several exceptions with both the open and the close methods. It seems that, even if not required, the CalmConnector uses the QoS parameters. This issue is under investigation by TI and it has been communicated to the CVIS helpdesk.

IPv4:

With Adoption of the S.O. CVIS R8 the IPv4 connectivity in the OBU Router is no more available. No explanation has been provided so far.

Integration:

Until now the integration test have not produced satisfactory results.

Mainly issues were due to CALM communication aspects and to mismatch in the Sw versions adopted (S.O., FOAM) between the infrastructure and vehicles.

Concerning the RSU in TI laboratory we used native IPv6 connection. Concerning the OBU, formerly, with the help of LOGICA team we have configured our devices to use their Home Agent, … services. Now we are also able to switch to the equivalent services provided by the Italian Test Site manager (Mizar).

Moreover, the City Operator web application is up and running in a TI server and is reachable over the internet by both IPv6 and IPv4 connections.

Appendix 6 : Interoperability questionnaire for EU projects.

CVIS / Safespot:

To what extend can CVIS and SAFESPOT be implemented on the same hardware platform?

HELMOND

Although started out with different visions and requirements the event in Helmond has shown that CVIS and Safespot applications can run on the same platform. However the architectures of CVIS and Safespot in communication and the supplied tools are very different. The SAFESPOT router reference hardware platform (as defined in "Router Hardware Specification v6.5.doc") only supports one radio channel, any CVIS service/application using this router is limited to the service channel.

This makes is necessary to create special tools on the platform to exchange messages and data.

For the Showcase in Amsterdam the integration of CVIS and Safespot will be demonstrated in the same way. There will be a single Linux / Ubuntu platform on which all applications (CVS & Safespot) will run. In the same way as for Helmond special tools will be installed to hand over the data and messages through the communication channels.

ITS Stockholm

Vehicles: In Vehicles, common router HW, HOST PC/HMI PC HW and Vehicle Gateway HW is possible. These three have been implemented in the Swedish Test Site Vehicles.

RSU:s: For RSU:s the Router HW and HOST/Main PC HW can be used in common.

Infrastructure: Depending on Infrastructure Gateway the Infrastructure can be in common. In Sweden the infrastructure is shared. (Application servers, Data logging HW, Visualization HW, Back-bone network etc.)

Cooperative Mobility Showcase Event, Amsterdam

The public road tour in Amsterdam demonstrated applications of CVIS, SAFESPOT and COOPERS. The vehicles were equipped with two platforms, The COOPERS platform and the CVIS/SAFESPOT platform, during the tour the presenter switched between the two platforms. No COOPERS applications were demonstrated on CVIS/SAFEPSOT platform or visa-versa. Several SAFESPOT applications (2xIRIS, SMAEV) based on V2I and V2V communication are demonstrated on the CVIS platform.

Interoperability of CVIS and SAFESPOT beaconing messages are shown by applications like: radar view and bird eye view. Which show the position of equipped vehicles and RSU in the surround if a vehicle. Both SAFESPOT vehicles and RSUs and CVIS vehicles and RSU are all visualized in the same overview, demonstrating the exchange of position, speed etc between the projects.

To what extend can CVIS and SAFESPOT run in parallel on one hardware platform?

HELMOND:

CVIS and Safespot have been installed in parallel on the hardware platform. They do not share a single architecture, but are able to communicate with each other to some extend.

ITS Stockholm

The systems can run completely on one HW platform. If defining the platform as a systems of HW components. In Sweden, the Safespot and CVIS systems work in parallel in Vehicles, RSU:S and infrastructure. No reboot or switching of HW components is required for running the different project applications SW.

Cooperative Mobility Showcase Event, Amsterdam

Similar to Helmond

To what extend can CVIS and safespot share hardware/software resources?

HELMOND

CVIS and SAFESPOT use the same hardware:

- The router software was integrated in such a way that both hosts could share one router. In practice this meant one instead of two routers and 2 two instead of four antennas.

CVIS and SAFESPOT hosts also shared the interface to the traffic light controller (the roadside gateway) to extract data on signal states and detection from the controller.

Some software components are shared:

- The vehicle display (the HMI) was shared between the two platforms. Both CVIS and SAFESPOT messages were collected at the CVIS platform and shown on the display according to a priority scheme.
- Implementation of the Cooperative Awareness Message (CAM) enabled a unified way of communication that allowed receiving and understanding beacons from both CVIS and SAFESPOT vehicles.
- CVIS and SAFESPOT uses the same implementation of the CALM networking stack for V2V/V2I communication.

ITS Stockholm

Core components such as LDM, communication capabilities, Gateway SW, HMI manager can be shared in parallel.

Cooperative Mobility Showcase Event, Amsterdam

Similar to Helmond

How would you conclude about the interoperability between CVIS and Safespot?

HELMOND

The router integration and the CAM are important milestones concerning interoperability between CVIS and SAFESPOT. Interoperability on host level is also possible, but here complexity increases dramatically. It is obvious that CVIS and Safespot started out with their own architectural ideas and visions. This was based on the fact that Safespot was targeted for safety application, thus a higher accuracy needed. CVIS was designed for efficiency applications. In Helmond it was managed to create some a form of single environment, but it is not the longer term solution.

The most ideal situation for the future development of cooperative systems is to look back and redevelop the single architecture for cooperative systems in which the best components of Safespot and CVIS will be taken into account. Note that some components developed in

CVIS could perfectly be used in Safety application and vice-versa.

ITS Stockholm

Very Good

Cooperative Mobility Showcase Event, Amsterdam

Similar to Helmond

CVIS / Coopers:

To what extend can CVIS and COOPERS be implemented on the same hardware platform?

HELMOND

CVIS and COOPERS use different wireless media for communication (CVIS uses 3G, M5 and IR, while COOPERS does not use M5). The CVIS platform assumes a host – router separation in the vehicle with IPv6 applications running only on the host for the CVIS IPv6 policy routing implementation to work.

CVIS, nevertheless, is mostly medium-agnostic, and can use any communication medium as long as it is adapted to the CALM framework.

ITS Stockholm

Currently the CVIS and Coopers HW platforms are not compatible. There has been no attempts to harmonize the two HW platforms.

Cooperative Mobility Showcase Event, Amsterdam

No interoperability between CVIS and COOPERS have been demonstrated in Amsterdam

To what extend can CVIS and COOPERS run in parallel on one hardware platform?

HELMOND

none

ITS Stockholm

No research has been done within the context of the test site to validate if this is possible.

Cooperative Mobility Showcase Event, Amsterdam

Not researched.

To what extend can CVIS and COOPERS share software resources?

HELMOND

none

ITS Stockholm

Some interoperability between the projects was done during the joint session of ITS Stockholm 2009. Two different applications were implemented and run on all 3 projects with shared software resources. The first of these applications was the speed limit application. The dynamic speed limit of different road links was provided by software running on the CVIS platform. The interface of this application was a web service. The coopers RSU polled the web service for up-to-date speed limits.

In general a translator layer between the two projects platforms was needed to un-marshal information from one platform and reinterpret and marshal it in the other platform's information model. In the example given above a common web service was use to represent the CVIS internal software resources.

The interoperability between software resources in the CVIS and COOPERS project is very limited.

Cooperative Mobility Showcase Event, Amsterdam

No interoperability has been demonstrated in the event

How would you conclude about the interoperability between CVIS and COOPERS?

HELMOND

There is no interoperability. So the conclusion is to designs the single architecture for CVIS , Safespot and Coopers. A mix between the Safespot and CVIS platform is the best way forward. This can serve safety applications and efficiency. It can also handle the Coopers applications.

ITS Stockholm

Poor interoperability, as the projects have been developed separately. To improve, the same work that has been done between Safespot and CVIS projects will have to be done, both HW and SW interoperability. Also the projects can share SW resources as has been demonstrated at the joint sessions at WC2009.

Cooperative Mobility Showcase Event, Amsterdam

The public road tour in Amsterdam demonstrated applications of CVIS, SAFESPOT and COOPERS. The mini-vans were equipped with two platforms. The COOPERS platform and the CVIS/SAFESPOT platform, during the tour the presenter switched between the two platforms. No COOPERS applications were demonstrated on CVIS/SAFEPSOT platform or visa-versa. No interoperability has been demonstrated.

Appendix 7: Opens source questionnaire.

COMM

SP	Component	Owner	License
COMM	CVIS OS kernel	RAMSYS	GPL (v2)
COMM	Mobile IPv6 daemon	RAMSYS	GPL (v2)
COMM	Collection of base networking programs	RAMSYS	GPL (v2)
COMM	User-space implementation of L2TP	RAMSYS	GPL (v2)
COMM	CALM Management Daemon	RAMSYS	GPL (v3)
COMM	Routing Optimalization Utility	RAMSYS	GPL (v3)
COMM	CALM FAST ping utility	RAMSYS	GPL (v3)
COMM	Packet filtering ruleset application	RAMSYS	GPL (v2)
COMM	CALM 3G measurement utility	RAMSYS	GPL (v3)
COMM	CALM Interface manager utility	RAMSYS	GPL (v3)
COMM	CALM M5 Sensor Board management	RAMSYS	GPL (v2)
COMM	CALM Management library	RAMSYS	LGPL (v3)
COMM	CALM Management library for OSGI	RAMSYS	LGPL (v3)
COMM	Low-Layer CALM Management (CDDF) library	RAMSYS	LGPL (v3)
COMM	CALM FAST message library	RAMSYS	LGPL (v3)
COMM	CALM FAST Forwarding Table manager library	RAMSYS	LGPL (v3)
COMM	Inter-CCK message library	RAMSYS	LGPL (v3)
COMM	Common CVIS library (misc.)	RAMSYS	LGPL (v3)
COMM	Mobile IPv6 daemon	RAMSYS	GPL (v2)
COMM	Collection of base networking programs	RAMSYS	GPL (v2)
COMM	Policy Exchange Daemon	RAMSYS	GPL (v3)
COMM	Point-to-Point Data Link Protocol Daemon	RAMSYS	Dual GPL (v2) / BSD
COMM	CALM 3G measurement utility	RAMSYS	GPL (v3)
COMM	CALM M5 device driver	Q-FREE	Dual GPL (v2) / BSD
COMM	CALM IR device driver	EFKON	GPL (v3)
COMM	CALM IR Beam Manager	EFKON	Proprietary
COMM	CALM M5 HAL	Q-FREE	Proprietary

COMM	CVIS ICM	RAMSYS	Evaluation

Conclusion / self assessment:

23 software packages were created or modified in COMM during the project,

of which 3 are partially or fully closed source. Therefore COMM fully met the 80% open source requirement.

FOAM

SP	Component	Owner	License
FOAM	API: Data broadcast	Makewave/Ramsys	BSD
FOAM	Client API distributed directory service	Technolution	BSD
FOAM	API: Secure module	Ygomi	BSD
FOAM	API: Authentication and authorization module	Ygomi	BSD
FOAM	API: Local device tree	Bosch	BSD
FOAM	API: Calm service module	Makewave/Ramsys	BSD
FOAM	API: JSR179	PTV	BSD
FOAM	API: Management Agent Bundle	Makewave	BSD
FOAM	Knopglerfish OSGI framework	Makewave	CVIS binary
FOAM	Management agent bundle.	Makewave	CVIS binary
FOAM	Communication mechanism for supporting large files.	PTV	To be defined.
FOAM	Native management agent	Technolution	CVIS binary
FOAM	CALM service implementation	Ramsyst	To be defined.
FOAM	Distributed directory service (DDS) client implementation	Technolution	CVIS binary
FOAM	Secure module	Ygomi	CVIS binary
FOAM	Authentication and authorization module	Ygomi	CVIS binary
FOAM	Local device tree admin	Bosch	CVIS binary
FOAM	Data providers core	Bosch	CVIS binary
FOAM	Vehicle utilities	PTV	CVIS binary
FOAM	JSR179	PTV	CVIS binary
FOAM	Host management centre	Makewave	CVIS binary

Conclusion / self assessment:

For the implementation most FOAM partners did not agree to the open source license agreement and provided binary code instead of source code to the FOAM Software development kit. The binary code was shared with external companies in the applications submission contest and cooperating partners of CVIS.

The biggest part of the FOAM software implantation is based on Knopflerfish, which is an open source service platform maintained and improved by Makewave. All code, resources and web pages for the Knopflerfish project are managed by a subversion repository. The repository is open for reading to everyone.

All APIs are open source using the BSD license agreement. As the Knopflerfish service platform is open source, about 80% of the software developed in FOAM is "open source" (as defined at http://www.opensource.org/docs/definition.php).

POMA

SP	Component	Owner	License agreement
POMA	Map-Matching SW	Heudiasyc UTC/CNRS	Proprietary
POMA	Map-Cache client	Heudiasyc UTC/CNRS	Proprietary
POMA	Hybrid position SW	LCPC/LIVIC	Proprietary
POMA	WLAN positioning	ISMB	Open source
POMA	EGNOS SW	TASF	Proprietary
POMA	LDM SW (including map update client)	NAVTEQ	Proprietary
POMA	Map Update	TeleAtlas	To be defined
POMA	Map Update server (to be defined)	NAVTEQ	Proprietary
POMA	Agora-C encoder/decoder library	NAVTEQ	Proprietary
POMA	Agora-C encoding/decoding	TeleAtlas	To be defined
POMA	Map-Cache for POMA map-matching SW (internal to POMA only!)	NAVTEQ	Proprietary
POMA	Native Agent SW	Logica	Open source
POMA	EMAP (enhanced map) SW	LCPC	Open source
POMA	RTMaps SW	INTEMPORA	Proprietary - Existing commercial software
POMA	SAPOS SW	Public service	Free

POMA	WSN positioning SW	Mizar	To be defined
POMA	GRL	Mizar	To be defined
POMA	Geospatial platform	Mapflow	Proprietary
POMA	OEM GW access SW	VOLVO	Proprietary

Conclusion/self assessment:

COMO

	Component	Owner	License agreement
COMO	COMO API	PTV	To be defined
COMO	LDM API	PTV	To be defined
COMO	VEH: Data Fusion	VOLVO	Open source
COMO	RSU: Data Fusion	SIEMENS	Proprietary
COMO	CEN: Data Fusion	PTV	Proprietary
COMO	Vehicle beaconing	MAT.TRAFFIC	To be defined
COMO	Map Matching	MAT.TRAFFIC	To be defined
COMO	Radar View	MAT.TRAFFIC	To be defined
COMO	Driver Awareness	PTV	Proprietary
COMO	VEH: Elaboration of local traffic state (EFCD)	VOLVO	Open source
COMO	RSU: Elaboration of local traffic state	SIEMENS, PTV	Proprietary
COMO	CEN: Elaboration of network traffic state	MAT.TRAFFIC	Proprietary

Conclusion/self assessment:

CINT

Subproject	Component	License agreement	Owner
CINT	WebPortal –1.2.2	Open Source	Logica Netherlands B.V.
CINT	TripStore – 1.2.3	Open Source	Logica Netherlands B.V.,

CINT	Dynamic Routing Service – 1.2.3	OSGi bundles: Binary licensed SOAP Interface and Data Processing: Binary licensed XML Exchange Data Format: Open Source	Mizar Automazione S.p.A.
CINT	AgoraC library	Java SW: Open Source	Mizar Automazione S.p.A.
CINT	TripPlanManager - 1.0.0	SOAP Interface: Binary licensed Java Client Library: Open Source	Mizar Automazione S.p.A.
CINT	VehicleTripManager -		Technolution B.V.,
CINT	Navigator -	binary licensed to use free for the project	Technolution B.V.,
CINT	MapMatcher -	binary licensed to use free for the project	Technolution B.V.,
CINT	TrafficEvent – 1.0.0	Open source	Technolution B.V.,
TS-NL Specific			
CINT	TravelTimeManager 1.0.2	Binary licensed	Logica Netherlands B.V.,
CINT	TrafficIntegrator – 1.0.4	Binary licensed	Logica Netherlands B.V.,
CINT	Traffic Event Server – 1.00	Binary licensed	Logica Netherlands B.V.,
TS-Italy Specific			
CINT	OnTripTrafficState-Updater	Binary	Infoblu S.P.A.
CINT	TrafficIntegrator	Binary	Infoblu S.P.A.

Subproject	Component	License agreement	Owner
CINT	Host platform 1.0.1.6	Open Source	INRETS
CINT	SDK 1.0.1.6	Open source	INRETS
CINT	SDK extended 1.0.1.6	Binaries only	INRETS
CINT	Ghost driver service 1.0.1.6	Open Source	INRETS
CINT	Dynamic Speed Limit Service 1.0.1.6	Open Source	INRETS
CINT	Providers 1.0.1.6	Open Source	INRETS
TS-Sweden			
CINT	Dynamic Speed Limit Service - 1.0.0	Binaries only	Volvo Technology
CINT	Ghost Driver Service - 1.0.0	Binaries only	Volvo Technology

Conclusion/self assessment:

We cannot state that the CINT resources are 80% open source. It is a matter of counting lines. If you add Linux (open source) and OSGI/Java plus the bundles in CINT , which are open source, we will reach at least 60% open source.

CURB

Subproject	Component	Owner	License agreement
CURB	Priority Application, Vehicle	Peek, Siemens	BSD
CURB	Priority Application, Roadside	Peek, Siemens	BSD
CURB	Roadside Gateway, generic interface	Peek, Siemens	BSD
CURB	Roadside Gateway, Peek interface	Peek	Proprietary
CURB	Roadside Gateway, Siemens interface	Siemens	Proprietary
CURB	TLC interface, Peek controller	Peek	Proprietary
CURB	TLC interface, Siemens controller	Siemens	Proprietary
CURB	Priority Application, Peek TLC	Peek	Proprietary
CURB	Priority Application, Siemens TLC	Siemens	Proprietary
CURB	Priority Application, Siemens	Siemens	Proprietary

	Central		
CURB	Speed Profile Application, Vehicle, without HMIGateway	CRF	BSD
CURB	HMIGateway of Speed Profile Application, Vehicle	CRF	Proprietary
CURB	Speed Profile Application, Communication Component	Mizar	BSD
CURB	Speed Profile Application, Legacy Interface	Mizar	Proprietary
CURB	Traffic Control Assessment, Roadside	Mizar	Proprietary
CURB	Flexible Bus Lane RSU	ATC	BSD
CURB	Flexible Bus Lane Vehicle	Thetis	BSD
CURB	PTMS & legacy systems Interface	ATC	Proprietary
CURB	Information Application, Vehicle	Vialis	BSD
CURB	Information Application, Roadside	Vialis	BSD
CURB	Roadside Gateway, Vialis Interface	Vialis	Proprietary
CURB	Routing Application, Vehicle, Navigation System	Mizar	Proprietary
CURB	Routing Application, Route management Component	Mizar	BSD
CURB	Routing Application, Centre Routing engine	Mizar	Proprietary
CURB	Waypoint Generation, Centre	PTV AG	Proprietary
CURB	Vehicle data source	PTV AG	BSD
CURB	Strategy Application, Centre	PTV AG	Proprietary
CURB	Strategy Editor, Centre	PTV AG	Proprietary
CURB	Strategy Application, Vehicle	PTV AG	BSD
CURB	Micro-Routing Application, Vehicle	Peek	BSD
CURB	Micro-Routing Application, Roadside	Peek	BSD

Conclusion/self assessment

A large number of components are open source using the BSD license agreement. However, also a significant number of proprietary components exist (due to high number of applications build as extension on existing proprietary systems).

Conclusion: CURB resources are for almost 50% open source.

CF&F

SP	Component	Owner	License agreement
CF&F – DG	eWPServer – route calculation	PTV	Existing commercial SW
CF&F – DG	Vehicle Monitoring Server	PTV	Existing commercial SW
CF&F – DG	Vehicle Monitoring Client: Tour Assistant	PTV	Existing commercial SW
CF&F – DG	Road Network Management System	PTV	Existing commercial SW
CF&F – DG	On-board route display client	PTV	Proprietary
CF&F – DG	On-board Dangerous Goods Monitoring Service	PTV	Proprietary
CF&F – DG	On-board Dangerous Goods Monitoring Service (with hand-over)	Volvo	CVIS binary
CF&F - DG	Dangerous Goods Monitoring Gate (Roadside)	Volvo	CVIS binary
CF&F – DG	Geofence Tool	Volvo	CVIS binary
CF&F – DG	Transport Order Tool	Volvo	CVIS binary
CF&F – DG	Dangerous goods data exchange protocols	Volvo	Open Source
CF&F – DG/PZ	Transport Order Client	Volvo	CVIS binary
CF&F – PZ	Parking Zone Operator System	Thetis	CVIS binary
CF&F – PZ	Parking Zone data exchange protocols	Thetis /Volvo	Open source
CF&F – PZ	Fleet Operator System	Volvo	CVIS binary

CF&F – PZ	Parking Zone RSU	Volvo	CVIS binary
CF&F – PZ	Parking Zone Vehicle Client	Volvo	CVIS binary
CF&F – AC	Access Control City Operator: Web application to manage the Critical Area access authorization policies and area definition	Telecom Italia	CVIS binary
CF&F – AC	Access Control Service Centre: Software to manage the AC operation	Telecom Italia	CVIS binary
CF&F – AC	Access Control Road Side Unit: software to manage vehicles triggering (vehicle proximity to the area) and early data exchange (short range)	Telecom Italia	CVIS binary
CF&F – AC	Access Control On-board System: software to manage AC vehicle activities and data exchange with RSU and SC (short and long range)	Telecom Italia	CVIS binary
CF&F – AC	Access Control API: Vehicle assessments and the V2I and V2V data exchange protocols	Volvo, Telecom Italia, CRF	Open source
CF&F – AC	Access Control On-board System – Volvo HMI & diagnostics algorithms	Volvo	CVIS binary
CF&F – AC	Access Control On-board System – CRF HMI & diagnostics	CRF	Existing on board HMI is Fiat Group

	algorithms		proprietary

Conclusion/self assessment

- CF&F will meet the CVIS-OB2 requirement if the whole system is considered with drivers OS etc.
- The reason for the low extent of open source software generated by the SP itself is that the applications built, are either based on pre-existing proprietary software or built for a specific proprietary platform. However the data exchange protocols and the interfaces between the different systems has been set as open source.