eSafety Forum Working Group on SOA

Brussels, February 24th 2010

Final report of the Service Oriented Architectures Working Group

Report and recommendations

(V0.97)

Table of contents

1 Preamble

This report is the result of the SOA Working Group’s activities over more than 2years.
The Working Group has been a purely voluntary initiative, with no contractual arrangements of any kind. The results of the Group’s endeavours as presented in this report are now in the public domain, and may be quoted, subject to specification of the source.
As Chairman of the Working Group, I want to thank all members for their support and contributions.
I would like to thank especially my co-chairs Fulvio Sansone and Natalino Curci for their contributions and engagement.

2 Partners

SOA is used by various industries as the main mechanism for flexible, open business transactions support. The list of partners active in WG work and contributing to this report reflect this wide scope of interests
For the eSafety SOA working group, there is a long list of interested parties and some core partners that share the main work load.

The work group is lead by the two chairing companies, T-Systems and Polidream S.r.l.
Two main contributors are Oracle and Nimera as former co-chair and actively supporting work group member.

Companies that support eSafety SOA working group include main industrial player from various fields:
ACEA, Brisa, Daimler Ag, Ertico, IMA Benelux, LISITT, Sap; SBD, Toll-Collect GmbH, T-Systems, Volvo, Ygomi, BMW, ISEL and others.

For a complete list of SOA WG interested and contributing partners, please refer to the annex.

3 Executive Summary

To deal with tomorrow's transportation challenges, systems allowing vehicles to communicate with other vehicles and with the infrastructure, also known as co-operative systems, are needed. A pre-condition for the successful introduction of such co-operative systems is that existing services can be extended and future services can easily be introduced on the same in-vehicle and roadside infrastructure. This calls for a common view on Service-Oriented Architectures (SOA).
In autumn 2006 the eSafety Forum Steering Group decided to launch a working group on Service Oriented Architectures (SOA) and its relevance for eSafety.
The WG had its kick-off meeting on 5th of July 2007 and will conclude in 2009.

Service-Oriented Architecture (SOA), which has taken the IT industry by storm over the past few years, is a software architecture that starts with an interface definition and builds the entire application topology as a topology of interfaces, interface implementations and interface calls. It is also a relationship of services and service consumers, with both software modules large enough to represent a complete business function.

Similar to other enterprise and consumer services, eSafety and ITS services need to have common architectures and interface specifications to help overcome possible market fragmentation. SOA concepts and related technologies are expected to play an important role in this scenario, facilitating service interoperability and cooperation among stakeholders and specialised parties. The Service-Oriented Architectures Working Group will play a key role in identifying the benefits SOA can bring to eSafety and ITS, as well as devising a roadmap for SOA implementation. Its first tasks will be to describe the state of the art, as well as identify the missing elements for deployment and the steps to facilitate market introduction.

eSafety services require involvement of different partners providing parts of an eSafety service. Public safety organisations and health systems need to co-operate with OEMs and 3rd party service providers.
The combination of partners may vary from country to country – the service provided however has to be available with a consistent quality.

This report analyses if and how SOA principles may support the special eSafety requirements that go beyond typical distributed services needs.

Due to its design features like loose coupling, SOA seems especially suited for services deployment and usage from devices only sporadically linked to the internet.

Flexibility in partnering, use of services currently available, update of services to address regional or actual requirements are just some of the features provided by the use of service oriented architectures.

4 Introduction

eSafety, by its very nature, is a field of activity that requires involvement of parties with partially diverging interests and goals.
Beside the driver as the one who benefits from the results of eSafety activities, there are usually several public and industrial partners involved even in the provision of a single service to the end user.
As of today, the required collaboration among these partners is established mostly through proprietary and domain specific interfaces (e.g. standard interfaces for road authorities). The consequences are manifold: services are usually fragmented, generally not interoperable and inflexible; partnerships set up between industrial and public partners are “point-to-point” so investments cannot be leveraged in new partnerships; local and SME service and content providers have difficulties to interface their offerings to several large providers; all this resulting in a fragmented and slow developing market.
IT technology like Service Oriented Architecture –SOA- is one way to loosely couple applications from different partners to achieve a joint service provisioning.
The promise is that thanks to applying of a SOA concepts to eSafety in few years time, most of the safety services deployed throughout Europe would be interoperable, guaranteeing users seamless coverage when travelling abroad.

In such an approach, Public Authorities and service providers would expose and consume “Services” according to certain standards and Service Level Agreements (SLA) between parties. The specific transfer of information between defined parties could happen according to the best available communications system and protocol allowing for effective interoperability between communication systems. Such a SOA-based approach would provide for the flexibility of standards-based integration and orchestration of business processes.

The use of Service Oriented Architectures would also give the possibility to Public Authorities and service providers to easily reuse the eSafety platforms deployed to support a wide range of value-added services like information and comfort services.
In addition, the SOA approach would allow for seamless integration of eSafety, and generally associated telematics services, into the back-office operations of Public Authorities and Service Providers allowing, for instance, integration with Customer Relationship Management (CRM) platforms, billing platforms for toll services, Business Activity Monitoring platforms for real-time monitoring of operations and reporting, etc.
Systems build according to the SOA best practices will also enhance and facilitate the testing, validation and certification of safety oriented ITS Systems. SOA makes it possible to integrate distributed and existing reference services and components into test bed systems thus without the need for additional developments.

This would produce relevant benefits for businesses and the society, including enhanced and widespread access to safety and rescue services, and improved sustainability of road transport, by reducing its impact on communities in terms of traffic congestion and pollution.
This paper will point out where SOA may be used, when and why it may be favoured above other solutions and what kind of prerequisites may be necessary to get the most out of a SOA architecture implementation.

5. What is SOA?

From a technical perspective, a Service Oriented Architecture is a collection of self-contained services (system functions) that communicate with each other over specified interfaces.
From a business perspective, a service-oriented architecture is a style of multi-tier computing that helps organizations share logic and data among multiple applications and usage modes.
The SOA Promise: Defining services at a "business level" enables rapid composition of end-to-end business processes, delivering on the promise of greater IT flexibility and agility: Lower Technology Costs, Smaller Business IT Gap.

5.1 Definition of "SOA":

Compared to the definition given by OASIS, the following seems more in line with our view as eSafety working group:

"SOA is a software architecture that starts with an interface definition and builds the entire application topology as a topology of interfaces, interface implementations and interface calls. SOA would be better-named "interface-oriented architecture." SOA is a relationship of services and service consumers, both software modules large enough to represent a complete business function. Services are software modules that are accessed by name via an interface, typically in a request-reply mode. Service consumers are software that embeds a service interface proxy (the client representation of the interface)." [1]

5.2 SOA Terminology

To describe SOA accurately, let us define some of its terminology:

Service	A unit of business functionality that can be invoked over the network and performs a distinct activity on its own (minimum dependencies)
Web service	A service that is called in a standard way, so anyone can use it without knowing its internals
Loosely coupled	When services are self-contained, and can be easily combined ("composed") and disassembled, they are called loosely coupled.
Service-Oriented Architecture platform	A standards-based platform that lets you model, develop, find, and combine services into flexible business

[1] Source: http://www.gartner.com/DisplayDocument?doc_cd=114358

	processes
Orchestration	Combining and assembling ("Composing") services into a coherent business process – with a central governance service (ESB) controlling the process flow: also known as *business process management*
Choreography	Combining and assembling ("Composing") services into a coherent business process without central governance
Interface	Definition of a component abstracted from its implementation. In the context of SOA the interface defines the contract between producers and consumers of data and functionality.

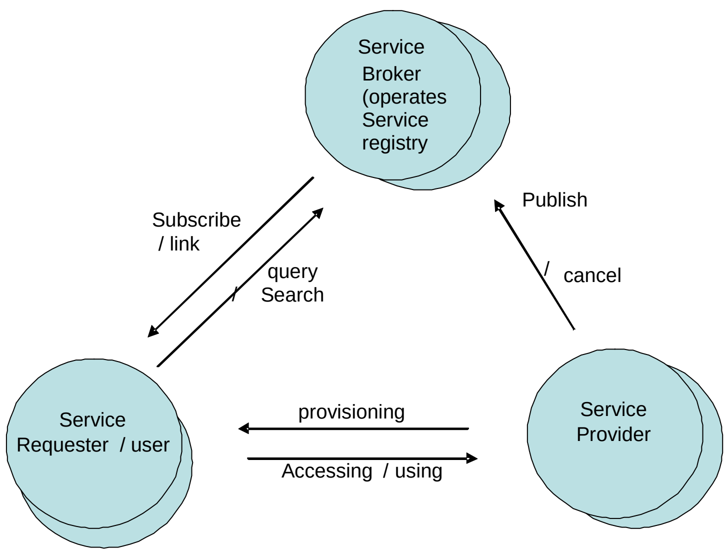


Figure 1 - Generic SOA roles

Figure 1 shows the main roles SOA defines. A service user may request a certain service (e.g. a local incident warning) by consulting a service broker or service directory. The service broker delivers always the best suitable and available provider that is available for a certain context.
This inherent flexibility may be configured by a user's profile to match certain quality or localisation criteria.

5.3 Advantages of SOA

SOA may be seen as one step forward in trying to handle increasing complexity and

enable joint processes inside companies or between different companies.
So, here in SOA two different threads join:
The evolution from a programming standpoint, started with “just writing code”, followed by structured programming, object based programming and languages modelling business relations & processes (e.g. BPEL, BPML).
All this has been done to be able to handle the increasing complexity of software development and maintenance.
For eSafety, four aspects are the most relevant:

- Loose coupling, allowing services to act independently, just synchronising via communication links when available
- SOA governance, the ability to control all aspects of a service lifecycle from development, deployment, usage and version control up to closing of a service.
- QoS: There is a specific stack in SOA dealing with this including message quality, security and transaction quality
- Business suitability: increased competition by compatible service offerings; standardized interface descriptions allow for changing services without redesign.

All described above, works independent of technical implementation, perfect for technical interoperability between different entities, even or especially with diverging technical ICT architectures.

5.4 Infrastructure / backend communication

Here, mainly the interaction of service providers is addressed. Important examples included authentication, billing or invocation of third party content. The development of standardized services available by means of web services makes it possible for more public and private organizations to construct sophisticated an innovative eSafety backend systems without the need for huge development investments. A service provider, serving enhanced traffic information to its customers is able to access services such as DATEX II nodes, Parking information systems in a B2B fashion merely by using standardized orchestration technologies such as the Business Process Execution Language (BPEL) and the Enterprise Service Bus (ESB). For the content provider it becomes easier to offer its content to the market while it becomes cost effective for even smaller SMEs to build innovative eSafety oriented ITS services. To realize this Services driven ecosystem standardization efforts such as ISO 24097 are primordial.

5.5 SOA compared to legacy technologies

The term Service Oriented Architecture did not pop-up by accident or suddenly without any precedence. As soon as computers got connected to each other they started to exchange data. For a long time however, computers remained stand-alone machine running many sessions in a sharing mode. Dumb terminals were connected to, at that time, huge computers responsible for the all applications layers. These computers received input from the end-user, stored or accessed data to or from huge storage devices and handed the result of the transaction back to the end user.

The raise of networked personal computers near the end of the 1980’s changed that paradigm to a more flexible and performing Client/Server architecture. Compared to the passive green screen terminals, personal computers had their own processor on

board and were able to run their own applications without depending on a centralized CPU. These personal computers were connected to centralized data servers which allowed for multiple clients to access and modify shared data. Client/Server certainly can be seen as a big step forward from the centralized processing architecture used in the dark ages of computing but still had some very important drawbacks.
First of all the Client/Server system remained rather closed and not used proprietary interfaces to shift data between client and server. The processing of the data was done on the client machine which introduced some important problems especially related to maintainability and performance. In the middle of the 1990's the internet became more common and made it possible to connect computing devices not only in a local network but move data between systems located in different geographic locations. Maintainability issues soon resulted in the introduction of a middleman role, the application server. Application servers stand between the end user and the data store. Applications servers move the processing of the information from the "intelligent" terminal back to a specialized type of server.
CORBA was one of the first application servers and is still widely in use today especially for mission critical applications. By distributing functionality over different computing systems the overall performance increases and huge applications are much easier to maintain. In general it is much easier to maintain a CORBA application on three servers serving thousands of users than having to update thousands of fat clients running on PCs. A CORBA application runs a piece of functionality which is accessed remotely from an application running on another server or a PC. Unfortunately the interfacing between entities remains on a very low level and is somewhat restricted to local area networking and due to firewall issues less appropriate for wide area networking.
With the advent of the extensible Mark-up Language and web oriented technologies such as HyperText Transport Protocol (HTTP) applications became even more distributed and presented their interfaces in a standardized XML format. The Interface Definition Language introduced by CORBA changed to an XML description of a Service end point, the Web Service Definition Language file, which made it possible to use remote application functionality on the fly even at runtime. A directory specification such as UDDI makes it possible to publish useful web services and make them available to consuming applications.
This approach clearly shows some very important advantages compared to the previous mentioned technologies. Processing is now done by specialized entities and the results made available over well defined interfaces. As a result, consuming services or endpoints have the possibility to select best of breed implementations without having to re-implement the entire or parts of the application should a producer service become unavailable.
The ability to create loosely coupled systems is without any doubt the most important result of such a "Service Oriented Architecture". Indeed, the implementation of consumers and producers are completed decoupled and abstracted from each other. A traffic information producer service can be implemented in for instance Java while the consuming application is developed on a .NET framework.
This is the true power of SOA: by selecting the appropriate implementation for each of the services in the end-to-end processing chain it is possible to create powerful and reliable systems.

5.6 Rational / Motivation for introducing SOA

SOA helps address the fragmented landscape of service provisioning in a multi-providers environment and addresses the difficulties associated with silos of different service infrastructure and applications. It enables greater flexibility through:

1. Greater Interoperability – SOA, and the industry standards underpinning it, enable existing silo'd applications to interoperate seamlessly and in an easier to maintain manner than any traditional solution.
2. Increased Re-use – Once legacy systems and applications are service-enabled, these services can be reused, which results in reduced ongoing development costs and results in reduced time to market. Further, business processes built as an orchestration of services can also be exposed as services – further increasing reuse.
3. More Agile Business Processes – SOA reduces the gap between the business process model and implementation. This enables changes to business processes already implemented as orchestrations of services to be easily captured and implemented.
4. Improved Visibility – SOA can give improved business visibility by enabling business capabilities exposed as services, and the status of in-flight business processes automated with BPM technology, to be rapidly integrated into service-enabled enterprise portals aiding business decision-making.
5. Reduced Maintenance Costs – SOA development encourages duplicated overlapping business capabilities (services) that span multiple applications and systems to be consolidated into a small number of shared services. This enables elimination of redundant services and reduces the cost of maintaining systems by providing a single point of change for application logic. Further, SOA gives the means to gradually phase out legacy systems and applications whilst minimizing disruption to the applications that are built on, or are integrated with, those using SOA principles.
6. Enhanced scalability – Services may be redesigned or moved to a dedicated machine to improve performance with influencing other processes.

5.7 SOA Standards

The SOA drive behind the use of open standards is that solutions based on standards drive businesses' total cost of ownership down and promotes successful adoption. A number of standard bodies are engaged in driving creation and refinement of standards in this area, OASIS, OAG, OMG, and W3C are some of the standard bodies where relevant pieces of standardisation work takes place.
Figure xx provides an overview of the relevant standards in the different areas related to SOA like business services, service bus, process orchestration, user interface and monitoring.
ISO has realised the importance of SOA and is working on a standard to use web services for ITS. The standard is called ISO/ICE DIS 24097-1. Some core parte are summarized below.

DRAFT INTERNATIONAL STANDARD ISO/IEC DIS 24097-1

Using Web Services (machine-machine delivery) for ITS service delivery —

Part 1: Realization of interoperable web services

1 Scope

The scope of this international standard is to provide specification for ITS sector WSs. Figure 1 shows a high level use case of WSs. The main entities are: service provider, service requester, and registry. Registry includes business information and technical information (interface and policy). Figure 1 also depicts the actions of service provider and service requester in italics.

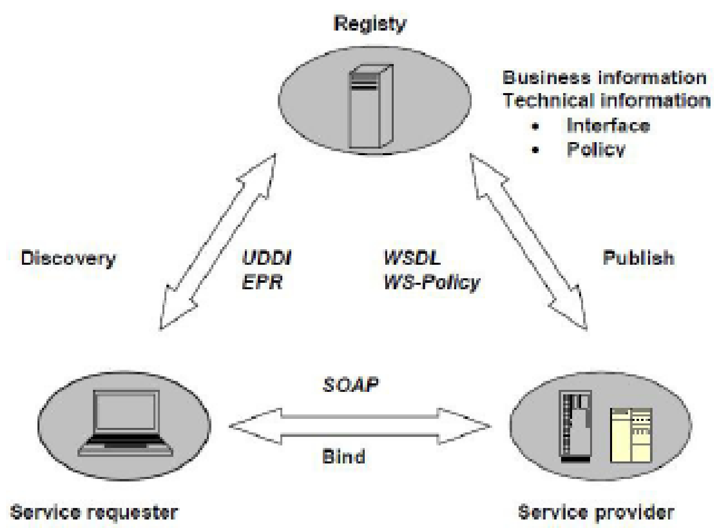


Figure 1 — WS use case

The first subject of this International standard is how to realize these use case requirements.

The second subject of this international standard is international standard WS architecture. In order to construct web services many functionalities need to be defined. To cope with this complex situation, an architecture is required. In this international standard SOA is applied. Another adopted principle is the use of metadata description of WS. Metadata are higher level (or abstract) description of WS behaviour. It enables autogeneration of both service program and consumer program. Metadata description provides a more simple and stable description of service specification and management. It also facilitates flexible evolution of ITS WS.

Figure 2 shows main components and technologies which contribute to a SOA based system

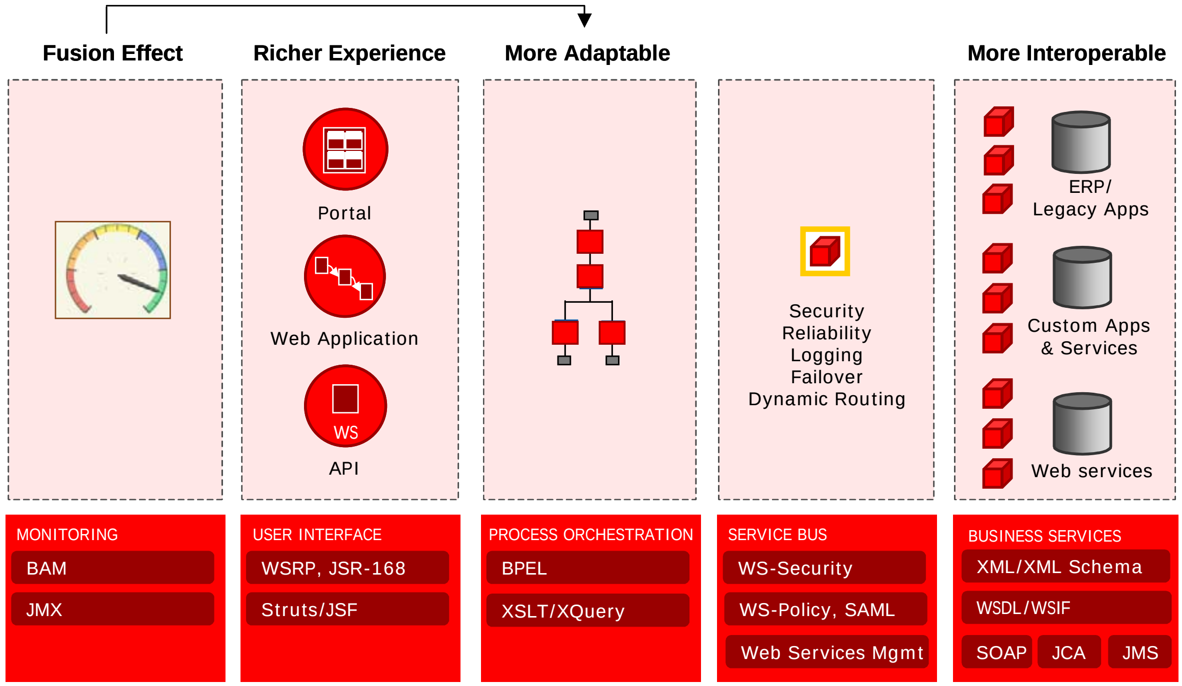


Figure 2 - SOA advantages and exemplary technologies

6 eSafety special requirements on SOA

eSafety applications require special attention with respect to availability, robustness and privacy. The status e.g. of an eCall system has to be clear at any given time to allow users to react and behave differently if services are unavailable. The following paragraphs divide the work domain of SOA in eSafety into three concrete focal points and identify the requirements for each of these focal points. For each of the focus points a number of example specifications and projects are given. By elaborating each of these areas of work the importance and use of Service Oriented Architectures to eSafety becomes clear.

6.1 Scope

It is the intention of this paragraph to look at requirements that are of sole and special interest when referring to SOA implementations and characteristics. Such requirements emerge of the very nature of SOA, namely its loose coupling, message based and distributed approach. SOA strengths like flexibility, adaptability and changing relations between service providers may turn into weaknesses if analyzed with “eSafety glasses” on.

That indicates a second main aspect: the services considered here need to be “back-end oriented”. A solely Car2Car oriented service like “lane assist” that just makes use of cars in the vicinity will not give us relevant requirements for the design an appropriate architecture.

We can focus on four aspects:

- in vehicle system
- infrastructure (or backend) communication
- Vehicle / infrastructure interaction
- Testing, validation & certification

The relation between three of these aspects is clearly shown by
Figure 3. The diagram is the derived from the architecture proposed by the GST and CVIS EC funded projects.
Testing, validation & certification is not shown as these elements are inherent part of an eSafty system. The different aspects are represented by the following entities in the diagram:

In Vehicle System – The In-Vehicle domain is represented by three main entities:

- The Client System which runs a Service Oriented Software platform .The Client System is a Multi Application platform able to run several services concurrently.
- The Vehicle Infrastructure which offers specific services to the Client System. These services could include communication and location services.
- The Nomadic Device – Nomadic devices are integrated into the vehicle by means of a gateway concept and use or offer services to the Client System.

Vehicle/Infrastructure interaction – In the diagram this domain is represented by two main entities:

- Control Centre (GST) or Host Management Centre (CVIS) – This entity remotely manages the Client System and mitigates much of the concerns related to use of SOA in the eSafety domain. Some of these concerns relate to the liability of car manufacturers regarding the use of soft- and hardware in vehicles. The Control Centre is an enforcement point so far as the management of the Client System is concerned. Towards the Service Providers it offers a standardized system for deploying services to the market. These two concerns, remote device management and deployment of services are realized by two sub-components:
    - o **The Provisioning server** – remotely manages the Client System and ensures that only validated and certified software is installed on the Client System.
    - o **The Deployment server** – Allows Service Providers to deploy their services in a well specified way to the Service Aggregator (Control Centre or Host Management Centre).

- The Service Centre – This entity realizes the actual service. For a Pay As You Drive system this is the for instance the Billing server operated by a Toll operator.

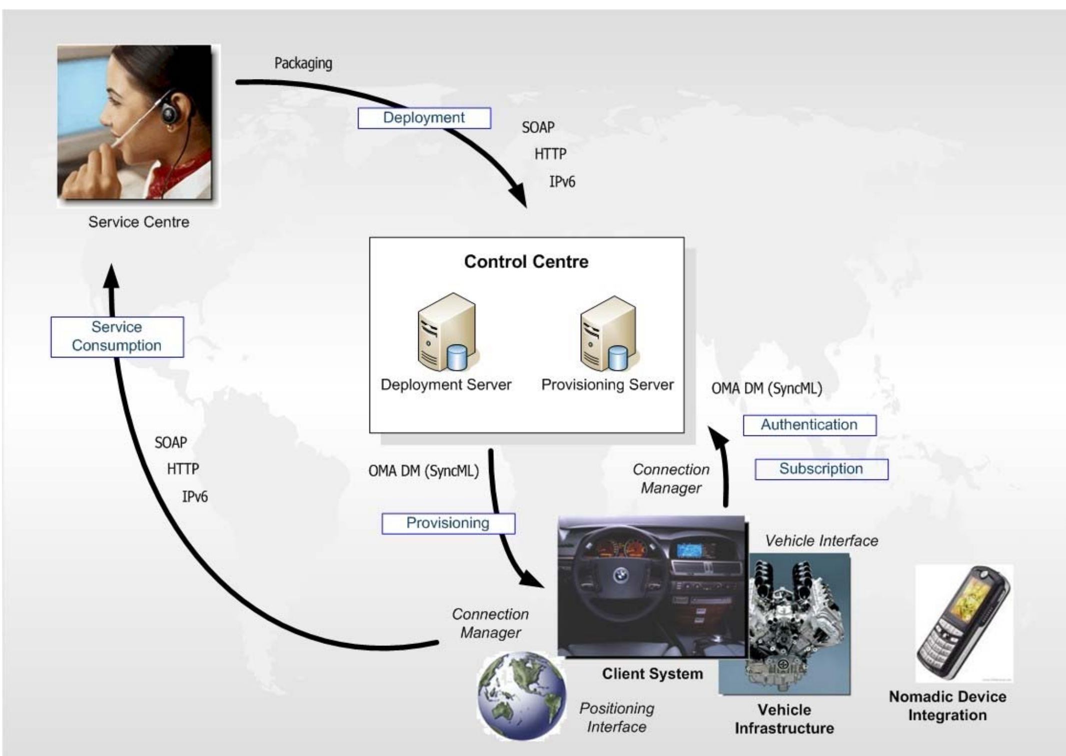


Figure 3 – Example on GST eSafety Entity relations

In vehicle systems

The On board Unit consists of components that are relevant for service usage like operating system, middleware for updates, security layer and so on:
SOA aspects include: services recovery, invocation and usage, inter-service communication. Research of Service Oriented Architectures in this domain is relatively new and is only made feasible thanks to the advances in processor capacity, memory sizes relative to the power consumption and physical size of the hardware platforms. So far as implementation of SOA is concerned on this level, the definition of “Loosely Coupled” is filled-in somewhat differently. In many respects these platforms do not support a complete abstraction of the service implementation but depend on a specific implementation technology. However, the ultimate goal is still to make components interchangeable and to support a multi-vendor environment. It must still be possible to insert “best of breed” components into a system. Some examples may illustrate this.

OSGI may be considered as a SOA solution with respect to many of the definitions of a Service Oriented Architecture. The OSGi framework consists of an OSGi framework or “runtime engine” which is able to install, start, stop and uninstall components brought-in from external media or processes. The runtime engine keeps a list of services, described by their interface description and allows other components to get access to these services by means of their interface names. It is possible to interchange existing services with new versions or even install complete new implementations of a service. Some examples of these low-level services are:

- Positioning

  In-vehicle services in a whole lot of cases need the correct geographic positioning of the vehicle. It therefore makes sense to implement a specific component which allows other components running in the framework to access these geographic coordinates.
- Communication

  Communication to the outside world is of course one of the cornerstones of a state-of-the-art ITS in-vehicle system. A communication component allowing other components to access the back-end infrastructure makes therefore perfect sense and avoids the implementation of communication component on a per service base.

Some other components which are of interest to in-vehicle services are:

- Logging
- Access to the in-vehicle infrastructure via CAN, MOST busses or other means.
- Access to the in-vehicle HMI
- Interface to Nomadic Devices

In general, the domain of embedded or in-vehicle SOA systems still needs a lot of research. Translating SOA concepts from the big-iron, internet world to the world of embedded devices still needs a lot of research. This research should not only encompass the technical realization of SOA but should also define positive business models and development methodologies. Some of the issues which still need investigation are:

- Openness of a SOA system – a loosely coupled interaction must be possible not only between components running on the same software platform but should also include interaction between several hardware components inside

of the vehicle. An example of this could be the implementation of an automatic eCall system which needs to accept events generate from front and side crash sensors.
- Certification and validation mechanisms – as stated above, SOA integrated into the vehicle, to become worthwhile, must become a multi-vendor/multi-component environment. A good certification and validation process remains therefore a primordial asset.
- Hardware and software innovation – Further research remains necessary to increase the performance, power consumption and capabilities of embedded systems.
- Positive business cases for all stakeholders involved – To realize eSafety in-vehicle components, positive business cases need to be found to make this a worthwhile investment for all stakeholders.

Infrastructure / backend communication

Here, mainly the interaction of service providers is addressed. Important examples included authentication, billing or invocation of third party content.

Vehicle / infrastructure communication

In-vehicle eSafety systems such as eCall, eLane, advanced traffic information and guidance systems need a reliable communication infrastructure. The importance of Over The Air connectivity can not be minimized related to services which may endanger lives when not working in a reliable fashion. Depending on the type of interface between the in-vehicle client system and the back-end system a number of side-effect need to be taken into consideration. The variable quality of service experienced when using over the air data communication need to taken into account when defining communication protocols and message formats. As an example, a full blow SOAP web service call over HTTP might introduce too much overhead when only a 2.5G point-to-point connection is available. For eSafety systems the OTA transmission can be divided into two main categories:

- Point-to-Point communication – Client Systems and Service Providers are connected in a one-to-one relationship. This might be an typical network connection over a 2.5G or 3G mobile network but it could also any other one to one communication such as SMS or MMS.
- Broadcasting – Service Providers transmit data in a one-to-many relation. An example of broadcast data communication is TMC broadcasted over the traditional FM band.

Each of these technologies has their own constraints which directly influence the interface descriptions between mobile and fixed system components.

Access to backend services / server / client applications, service downloads as well as updates over the air (e.g. OTA / FOTA) are aspects to be considered here. OSGI offers a good functionality, GTP (Global Telematics Protocol) as well.

6.2 Requirements

Main requirements for eSafety applications and services identified are similar to those defined for telematics applications in general.
eSafety however has more demanding requirements as malfunctions may result in possibly dangerous situations.
A user must always be sure if a service is available and trustworthy or receive a clear feedback if a service in unavailable or not up to date.

Service level agreements (SLAs)
Service level agreements and compliancy are core aspects of a trustworthy safety service. For all stakeholders, it must be traceable if all services involved are in a good shape, up to date, reachable and performing. Non-compliance should be detectable.

Availability
It needs to be clear in advance, before invoking a service, if it fulfils the required SLA criteria with respect to actuality, and validity.

Robustness
A service or application should clearly state if it is available and be able to handle lost connections or other unexpected status transparently.

Roundtrip time
As service provisioning may be distributed amongst a number of partners, timing requirements still stay as they are (see eCall example).
Security
The common requirements on security are even more relevant for eSafety applications and services: authentication, authorization, data protection, and non-repudiation

Health status
SOA services are often “orchestrated”; build of parts of a services chain operated by different providers. The status of the resulting service shall summarize all subsequent states and give clear information on its fitness for use.

6.2 Testing, Validation and Certification

In general eSafety oriented systems are often critical applications which, when failing, might prevent even the saving of lives and most certainly will not come up to their promises. System such as eCall, intelligent speed alerts, high priority traffic messages and so forth, must therefore thoroughly tested, validated and in most cases certified before they can be deployed in the large. European wide testing, validation and certification needs European wide testing facilities. Today many of these test sites exist or are planned. Most of these test sites focus on specific aspects of eSafety and ITS applications, mostly depending on the regional or national interest or on the commercial interests of the funding organizations. These test sites will operation field operational test on a limited or large scale and therefore produce services and infrastructure to conduct these tests. It would be favourable if test sites could share these services and infrastructure. In many cases, as it happens today, components are re-developed over and over again simply because existing components are not made available or are developed as closed and proprietary systems impossible to integrate.

7 Use cases overview

As SOA design and architecture is always related to certain industrial fields or business processes, it is important to describe the user activities supported by the SOA system.
To give an overview on scenarios and use cases prepares the ground for a more in-depth analysis of underplaying business processes.

In general, a variety a services and functions have been proposed to increase driver safety, they are summarised under “Smart Car” or “Intelligent Car” technologies:

The following list shows a wide range of ICT-based stand-alone or co-operative systems. Some are already in use (ABS, ESC), others are still under development or being introduced into the market (e.g. most eSafety services)

- Anti-lock Braking System (ABS)
- Adaptive Cruise Control (ACC)
- Adaptive Headlights
- Lane Change Assistant / Blind Spot Detection
- Driver Drowsiness Monitoring and Warning
- Dynamic Traffic Management
- eCall
- Electronic Brake Assist System
- Electronic Stability Control (ESC)
- Extended Environment Information
- Gear Shift Indicator
- Intersection Assistant Lane Departure Warning
- Local Danger Warning
- Night Vision
- Obstacle and Collision Warning
- Pedestrian/ Vulnerable Road User Protection
- Speed Alert
- Tyre Pressure Monitoring System (TPMS)
- Wireless Local Danger Warning

For SOA implementation, only the co-operative, communication cantered applications are of relevance (see “8.1 Scope” for details).

The uses cases drafted in the next chapter further describe such use cases / applications.

7.1 GST Service Oriented Architecture Use Cases

7.1.1 Introduction

The Global System for Telematics, or GST in short, project identified an Open Telematics framework, allowing to run multiple services on an in-vehicle or mobile computing system. The framework itself consists of a set of entities, which interact with each other over well-defined and agreed-on interfaces. These definitions form the basis for a specification effort starting from industry available standards extended with the requirements as identified by GST. To understand the discussion of those GST Use Cases valuable for a Service Oriented Architecture it is worthwhile to briefly discuss the GST concept, or design pattern as it is sometimes referred to, and add some definitions to the different entities, actors and interfaces which contribute to this design pattern.

However before jumping into the GST architecture some care should be taken about the terminology used in this chapter, especially in relation with the definition of a Service Oriented Architecture itself. Depending on the context the word “Service” indeed has a different meaning. In the Context of GST a service is the “product” offered by a Service provider. A service provider could for instance make traffic information available or in the case of a rent car, tabulate the driven distance and billing the car user accordingly. A service in the SOA acronym as defined by the SOA definition, given elsewhere in this document, is a piece of software that is accessed by name via an interface. From a high-level view this difference might sound like a small nuance simply because, how else are GST Services accessed by an end-user? Indeed, by means of a well-defined interface as is meant by the SOA definition applied by this workgroup. For the clarity of this text however we will prefix the word service with GST if a high level GST service is meant and SOA if the software module is referred to.

7.1.2 About GST

GST is an EU-funded Integrated Project that created an open and standardized end-to-end architecture for automotive telematics services.

The purpose of GST was to create an environment in which innovative telematics services can be developed and delivered cost-effectively and hence to increase the range of economic telematics services available to manufacturers and consumers. 50 partners from industry and research worked together to achieve its results.

As a comment, I would like to state that GST finished in 2006. I would state that GST concept have further taken on by CVIS and further developed in this project.

7.1.3 The GST Architecture

Figure 1 - GST Entities and Entity Relations provides a high level view on GST.

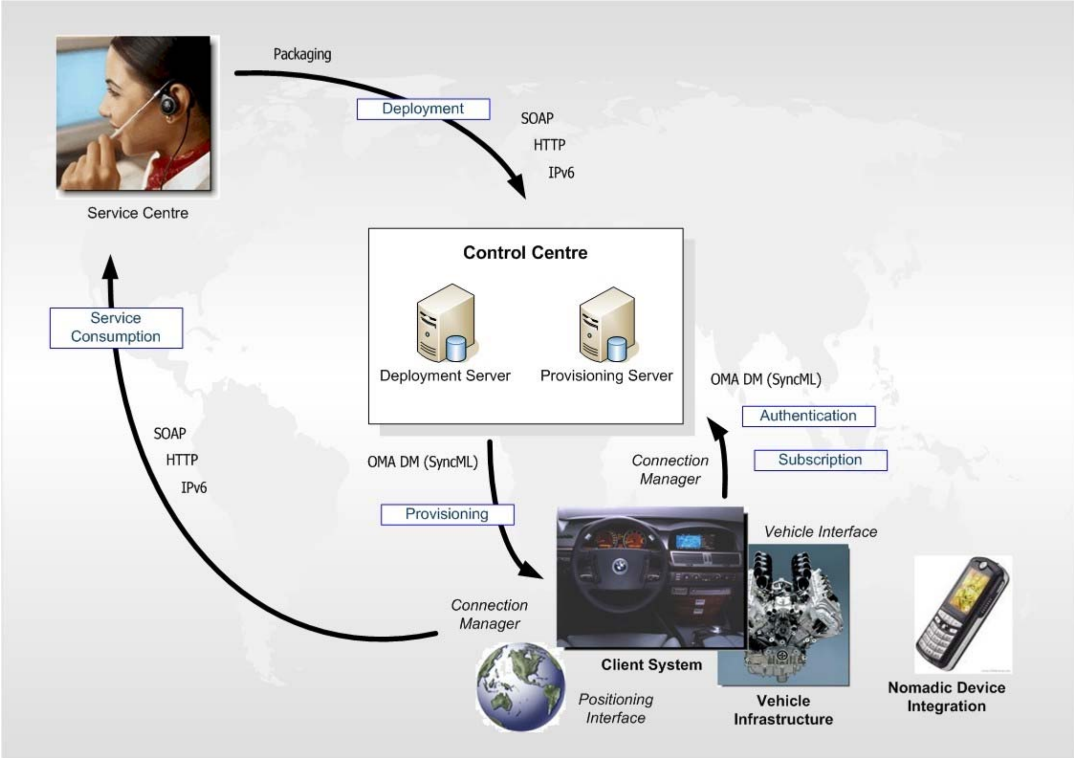


Figure 1 - GST Entities and Entity Relations

The following table summarizes the different Entities. On the figure these entities are printed in **Bold**.

Entity	Role
Service Centre	Provides a “GST Service” to the end-user. Example of services are: Traffic Information, eCall, Diagnostics, Pay as you drive insurance, Point of Interest, Speed control alerts etc.
Control Centre	Acts as the GST Service Aggregator. In general a Control Centre offers a selection of GST Services Application to the End-User and assures the correct delivery of the software necessary to consume the service to the in-vehicle client system.
Client System	Set of devices which, combined, realize a multi-application in-vehicle computing unit. Typically a Client System contains Input/Output interfaces (Screens, steering buttons, voice control, Text to Speech etc.) and a computing unit also referred to as the Telematics Control unit or TCU
Vehicle Infrastructure	Represents the data properties made available by the vehicle. These data properties can be read-only, write-only or read-write. In general the vehicle

	infrastructure offers information to the outside world about the state of the vehicle (diagnostics, eCall), the state of the vehicle environment (road conditions, occupants) and, depending on the type of services allowed to operate in the vehicle, the parameterization of a vehicle (ECU flashing to localize and tune a vehicle).
Nomadic Device	A nomadic device can be anything that could benefit from an interaction with the vehicle. Examples are music players, smart phones, Personal Navigation Devices (PND), Personal Digital Assistants (PDA), Diagnostic Systems etc.
Content Centre	Aggregates data either provided by the Vehicle Infrastructure (Enhanced Floating Vehicle Data) or external sources such as the government, traffic information providers etc.

These entities interact with each other over a set of well-defined and openly specified interfaces. These interfaces form the core discussion of this document and essential turn the GST design pattern into a Service Oriented Architecture. The following list briefly summarizes these interfaces:

Interface	Description
Deployment Interface	Describes the “deployment” of GST Service Applications from a Service Provider to the GST Service Aggregator or Control Centre. GST defines for this interface protocols such as SOAP, HTTP, IPv6. Where the SOA Services are implemented as Web Services or Web Applications. The interface furthermore specifies a deployment descriptor and packaging format.
Provisioning Interface	Defines the interaction between a Client System and the Control Centre. This interface is driven by Use Cases such as: GST Service Discovery GST Service Subscription GST Service Download GST Service Install
Vehicle Interface	Enables the interaction between a Client System and the Vehicle Infrastructure
Nomadic Device Gateway	Vehicle Infrastructure SOA Service allowing the integration of a Nomadic

	Device with the Vehicle Infrastructure and the outside world via the GST Service Consumption interface or Provisioning Interface
Nomadic Device Interface	Interface allowing a Nomadic Device to access the Client System. From the description it becomes clear that the Nomadic Device Gateway and Nomadic Device Interface are two SOA services which collaborate with each other and combined realize the Nomadic Device integration.
Service Consumption	Describes the interaction between the Service Provider and the Client System. This part of the GST Design pattern realizes the consumption of the service

7.1.4 Main Use Cases

The following table summarizes the main use cases identified by the GST IP project. For each of the Use Cases you will find a short description, a reference to the original GST deliverable and a short step by step description of the Use Case. This list is by far not complete but contains only those Use Cases relevant to the work of the eSafety SOA workgroup, so only safety relevant use cases are retained. Furthermore, the use cases are categorized according to their function in the interfaces table.

7.1.5 Making "eSafety" services available to the end-user

This set of Use Cases relate to the deployment and provisioning of GST Service Applications. Literature references are represented by numbers, which are specified at the end of this chapter.

Use Case	Literature reference	Short description
Upgrade of a service application	[1], UC-OS-0006, page 44	The Control Centre notifies the End User that a newer version of a specific Service Application is available. The End User chooses to download the upgraded version of the Service Application The Service Application is downloaded on the Client System using the Communication Infrastructure
Automatic Upgrade of a Service Application	[1], UC-OS-0007, Page 47	The Control Centre sends an upgrade request to the Client

		System. The Client System authenticates the Control Centre as per UC-GST-0023. This step is necessary to prevent malicious downloads by defector Control Centres. If the driving conditions permit and various other conditions are met (e.g. the user has requested auto updates, the Client System is compatible with the new version etc.), the Client System acknowledges or refuses the upgrade request by responding to the Control Centre. If acknowledged by the Client System, the Control Centre pushes the upgraded version of the Service Application on the Client System using the available Communication Infrastructure

7.1.6 Making information available to SOA data consumers

This set of a Use Cases describe content provisioning scenarios either at the origin of the value chain where content is unprocessed data made available by a vehicle or further down the chain where content is processed by a content centre and made available in the form of information such as high priority messages and traffic data.

Use Case	Literature reference	Short description
Access to the vehicle infrastructure	[1], UC-OS-0008, Page 48	The Service Centre wants to obtain in-vehicle data by means of the Service Application or the End-User wants to obtain in-vehicle data. A Service Application reads the Vehicle data.
Service Provisioning and application download	[1], UC-OS-0020, Page 58	The Control Centre determines the need for Service Provisioning. This may be the result of either a Control Centre initiative

		or an End-User initiative such as an initial log-on and so forth. The CC checks authorization and dependencies to other applications The Control Centre checks for local copies of the application to be provisioned The CC matches the Client System capabilities to the Service Application requirements The Control Centre updates the needed tables in the database to prepare the system for application download to the Client System. The Control Centre downloads the Service Application. The Client System checks the consistency of the downloaded file, the origin of the application and the permissions. The Client System installs the application and confirms the successful installation. The Control Centre continues the Service Fulfilment
End User Initiated Service Provisioning	[1], UC-OS-0021, Page 60	Identical to the previous Use Case but initiated by the end-user
Control Centre Initiated Service Provisioning	[1], UC-OS-0022, Page 64	Identical to UC-OS-0020 but initiated by the Control Centre
Service Consumption	[1], UC-OS-0023, Page 67	The End-User or Client System triggers the start of a Service Application. At this stage no connection to a back-end system, Control Centre or Service Centre is necessary. In other words Service Applications do not need

		any remote connectivity to run correctly. The Service Application communicates with the Client System or with a remote Control Centre or Service Centre. If necessary the Service Application sends an accounting log message to the Control Centre for billing purposes The Service Application is closed by the End-User or the Client System.
Service Deployment	[1], UC-OS-0025, Page 72	The Service Centre User physically deploys the package on the Control Centre according to the chosen business model. The Control Centre unpacks the Service Application package and allocates the different parts in its repository. The Service Centre User applies all necessary configuration changes in the Control Centre

7.2 *Additional eSafety Use Cases*

7.2.1 Introduction

To get a good understanding of the scope of an eSafety oriented Service Oriented Architecture a few practical and real-life use cases might be very helpful. This chapter discusses three use cases with a focus on the different boundaries between the entities of a complex ITS system. These entities collaborate with each other by means of well-defined and standardized interfaces. Exactly these interfaces are the subject of the eSafety SOA working group. The following picture gives an overview of a typical ITS system, in this case a traffic information application.

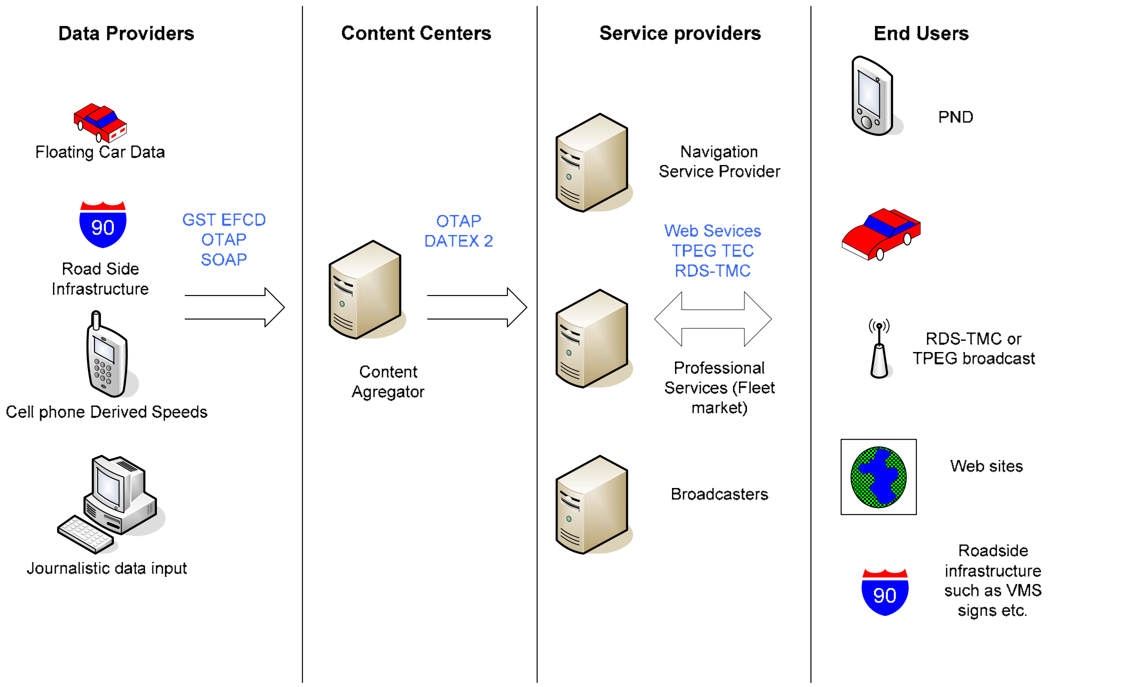


Figure 4 - Traffic Information system

From left to right the different layers of the system are bound tighter by means of interfaces specified by standards which, in many cases, are the result of EC funded ITS projects. The following scenarios are discussed:

1. Traffic Information Content aggregation and service provisioning
2. Diagnostic Service scenario
3. eCall Service scenario

For each of these scenarios we will have a look at the different entities and interfaces between these entities. A summary table gives an idea on the different specifications and standards involved. For each of the specifications and standards a pointer to the originating project and/or organization body is given.

7.2.2 Scenario 1: Traffic Information Content aggregation and service provisioning

Safety on the road: Better informed drivers can make better judgment about their time of leave and itinerary. As a consequence they help in reducing traffic congestions and improve air quality by doing so. In general we could distinguish about a number of information types including:

1. Congestion and incident information with a normal priority
2. Short notice, high priority information such as obstacles, ghost drivers etc.
3. Public transport information
4. Delays and drive times
5. Etc.

Figure 1 gives a good view on the different steps involved in this scenario. The story starts by gathering the basic content for the different traffic information services. In this stage many interfaces take part in the interaction between the originator of the information and the content aggregators. The following are only a few of the possible interfaces:

Geo-positioning data transmitted from trucks, cars etc. (Floating Vehicle Data) – In general this information is received from fleet operators, leasing and renting companies, taxi operators, busses etc. In general each of the geo-positioning data points is forwarded on regular intervals to a fleet management application of some sort. An enhanced version of such an interfacing system has been researched by the GST Enhanced Floating Vehicle sub-project. In general what becomes important for this to work in an open and standardized way is:

1. A clear specified message format or at least a clear message element specification
2. An open specified interface to the vehicle environment

An enhanced Floating Vehicle system would allow not only providing information about location, speed and heading, but also about other measured information such as meteorological information etc. Standardized web services accessed over an HTTP/SOAP transport layer are a way forward to provide open implementations for this interface.

Raw data provided by road infrastructure – This data is forwarded by road-side infrastructure such as loops, infrared detectors, speed cameras etc. To be able to integrate this data, Content Aggregators need standardized message formats and/or message elements. Only standardizing this interface guarantees an open and efficient integration of a wide range of data originating equipment. As for the floating vehicle data providers, web services could offer a good set of specifications for an open interface.

Journalistic Information – While measuring systems such as floating vehicle data providers and detection loops indicate road conditions and report information on traffic congestions and travel times, journalistic information adds specifics on the circumstances of a traffic event. Additionally, these kinds of information sources could also add information on the availability of public transport, parking lot location and availability etc. Specifications such as OTAP and DATEX2 are found on this boundary.

The raw data provided by these different content sources as such is not very useful. The content aggregator tier adapts the raw data into information useful for service providers to package in valuable end-user services. Content Aggregators in general bundle the raw input data into cause and result information and provide this information as an OTAP or DATEX 2 feed to the different Service Providers. Depending on the Service the aggregation level is less or more detailed. As an example, navigation systems might need detailed congestion lengths and travel times while broadcasted traffic information relies more on journalistic details. Here again OTAP and DATEX 2 are good candidates for interfacing the Content Aggregator entity with the different Service Providers.

Service Providers create useful services from the aggregated traffic information.

Here I think we should introduce considerations about interoperability of information services that could be achieved by using SOA and Web Services for interconnecting different Content Aggregators and Service Providers. Without the need of specifying raw data formats, being FVD, road-side data or journalistic information, CA and SP may expose Web Services for fellow CA and SP. The possibility to consume such Web Services may be conditional to commercial agreements and fulfilment of specified SLAs, which may also be described and endorsed in M2M environment. The consumer SP would integrate the consumed service within its own service to the end user. For instance a SP covering a border region (e.g. Italy-Austria) has access to raw data from Trentino Alto Adige, but may not get access to Sud Tirolen raw data from neighbouring region. In this case he may strike a deal with a SP from Sud Tirolen who has access to these raw data. The Italian SP would calculate travel times on its own covered area and would complement travel times for users crossing border with travel times obtained through Web Service from the fellow Austrian SP. The Italian SP would not need raw data about the Austrian network since he would use the travel times computed by the Austrian SP.

8 Implementation examples

8.1 A SOA based eCall implementation

This implementation example is based on a proposal from Oracle on uses Oracle back-end components as a core element. The description given here focuses mainly on the SOA aspects. For a full description of the scenario, please refer to “The fully networked car” presentations.

Overall description

The eCall is an emergency call, which can be performed either manually by activation of a special SOS button in the car or automatically via in-vehicle sensors, indicating that an accident happened. Automatic triggering is generated by the airbag control module and/or other sensor data like rear sensors and transmitted over the bus system (CAN, MOST or others). After being activated, the in-vehicle eCall system establishes a 112-voice connection to the nearest, relevant PSAP (Public Safety Answering Point), which is a public authorized body. At the same time, a minimum set of data (MSD) is sent to the PSAP operator receiving the voice call. Because the minimum set of data contains key information concerning the accident such as time, location and vehicle-specific information the emergency call can be processed rapidly.

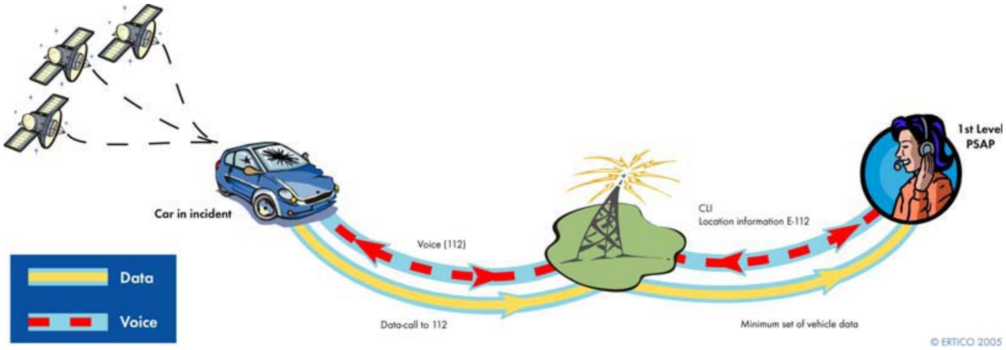
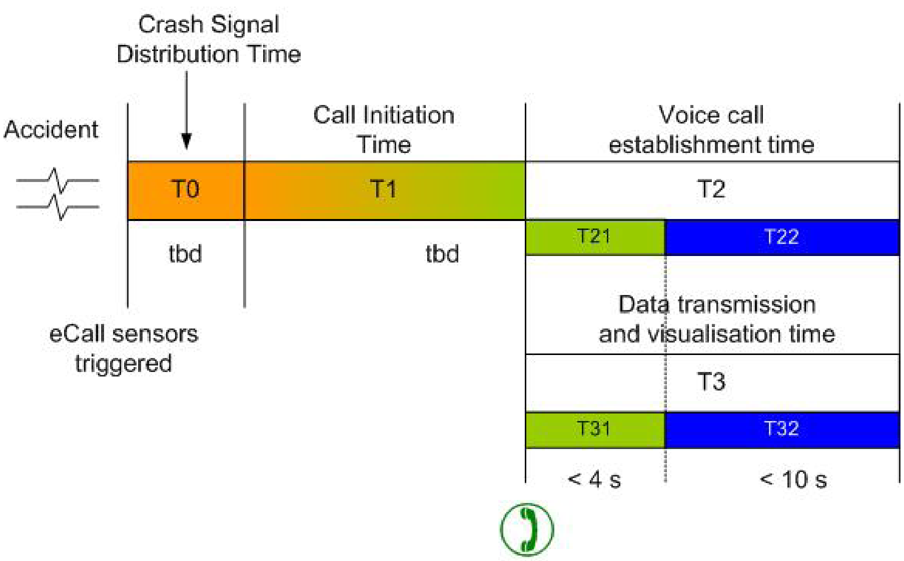


Figure 5 - eCall Voice / Data flow

Timing

The figure below indicates time restrictions, divided into different fields of responsibility/implementation. The time between the crash detection and the call initiation is not yet defined but it should not exceed 20 seconds. Thus the overall eCall service must not exceed 34 seconds until the PSAP is reached.
As SOA uses a loosely coupled paradigm, such requirements have to be closely monitored and ensured.



The SOA Approach

The proposed approach to realize the eCall architecture is the implementation of a Service Oriented Architecture (SOA). In such an approach PSAPs and Service Providers would expose and consume services according to certain standards and Service Level Agreements (SLA). This would consist in pushing data to an emergency web site, which would be accessible to all authorized parties involved.

Such architecture could look like it is shown in the next figure:

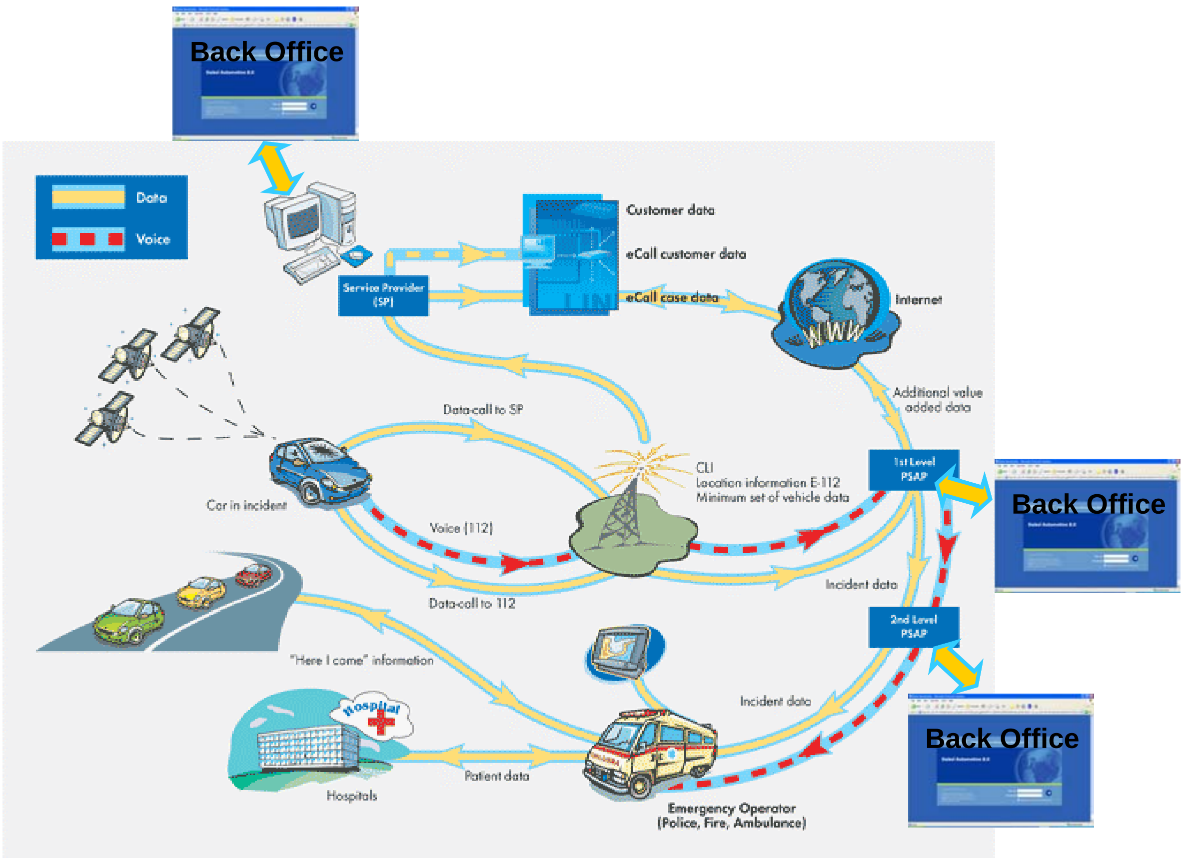


Figure 6 - SOA based eCall

Benefits

Such a SOA-based approach would allow an easy integration of the eCall with additional value added services that Service Providers may want to deploy in the future. In addition, it enables seamless integration of the eCall and generally associated Telematics services into the back-office operations of PSAP and Service Providers allowing, for instance, integration with Customer Relationship Management (CRM) platforms, billing platforms for toll services, Business Activity Monitoring platforms for real-time monitoring of operations and reporting, etc.
The web site used for joint eCall processing could be deployed at a national, regional or European level without further efforts. The content and access rights could be agreed with the service customers to ensure that privacy issues are properly handled within the different user groups defined.

8.2 Examples of SOA applications in Logistics

The following paragraph gives a first idea on how SOA is used today for logistics scenarios:

ARKTRANS is a Multimodal ITS Framework Architecture and thereby a template for intelligent transport systems (ITS). Standard functionality and common information elements are defined as well as the interfaces that arrange for interoperability between ITS. ARKTRANS was established in 2001 by the Norwegian Ministry of Transport and Communications, together with the Norwegian transport authorities for sea, road, rail and air.

FREIGHTWISE is an integrated project within the EU's 6th Framework Programme. FreightWise's overall objective is to support the modal shift of cargo flows from road to inter-modal transport using road in combination with short sea shipping, inland waterways and rail. It will achieve this objective by means of improved management and facilitation of information access and exchange between large and small public and private stakeholders across all business sectors and transport modes. Project also promotes EU-policies encouraging the development of open and interoperable systems, which meet the requirements of cargo owners, transport operators and inter-modal freight integrating services. The aim is to support the Commission in formulating future legislation and in developing initiatives that can provide a platform on which the industry can develop management solutions thus helping to increase the competitiveness of inter-modal transport. Agile Freight will liaise with and re-use the deliverables of FreightWise to inform the road map for co-modal eFreight

MarNIS (Maritime Navigation and Information Services) is an Integrated Research Project in the 6th FP, bringing together 44 partners and 12 sub-partners, to develop maritime navigation and information services on a pan-European basis. The main goals of MarNIS is to accommodate main elements in the European Transport Policy 2010 – “Time to Decide”, and specified objectives further developed in the Sustainable Surface Transport Work Programme 2002 – 2006. It is in this relation one of the challenges to turn the vision of “One Stop Shopping” into reality on a pan-European and global basis. The development of a mandatory systematic use of modern localisation and communication systems will be key elements in this process. The role of sea transport in an intermodal transport chain will be focused throughout the project. The results are expected to give a solid technological and scientific basis to the Commission and the Member States administrations to study, substantiate, test and formulate possible legislation on Safety, Security and Efficiency in Shipping.

EURIDICE (EURopean Inter-Disciplinary research on Intelligent Cargo for Efficient, safe and environment friendly logistics) is an Integrated Project funded by EU’s Seventh Framework Programme ICT for Transport Area. EURIDICE aims to create the necessary concepts, technological solutions and business models to establish the most advanced information services for freight transportation in Europe. The project is built on the Intelligent Cargo concept. EURIDICE will allow to address simultaneously the logistics, business and public policy aspects of freight transportation, by dynamically combining services at increasing levels of extension: in the immediate proximity of a cargo item for services directly interacting with the item itself; with supply chain services, for interaction with the actors responsible of shipping, carrying and handling the goods, as well as of the goods themselves; with freight corridor services, managed by authority and infrastructure operators who are not directly involved in supply chain business processes, but are in charge of infrastructures efficient operation, security and safety control. Such a service infrastructure is realized by adapting current SOA standards to the specific needs of moving goods and cargo communities.

8.3 SOA in ITS test environment

The goal of the project was to develop a re-usable and extensible framework for conducting ITS Test Cases

To be re-usable was a major design criterion for all elements like:

- Architecture
- Procedure and organisation
- Tools and services

The extensibility was mainly important for community development of reference components and services

After carefully analysing all requirements the decision was taken to use a SOA based approach; this allows the integration of legacy and new services in a well-specified and distributed fashion
The resulting system supports preparation for local and European wide Field Operational Tests.

The project team consisted of
ITS Nationals:
ITS Belgium (Project Management), ITS France, ITS Norway, ITS Netherlands (Connekt)
Research Organizations:
TINC, SINTEF, DLR, INRETS, TNO, IBBT
And SME's:
NXP, Technolution, Q-Free, TC-Matix

SOA principles were especially used to allow
- Re-use Europe wide Test Site infrastructure
- Setting up a Test Site infrastructure =
  - Combine existing services or new developed services to conduct a test process
  - Combining services = orchestration
  - Heavy use of Web service oriented technology
- One core component is the Test Aggregation Service Centre:
  - Data management
  - Data storage
  - Data retrieval
- Distributed (re-)use of well tested and proven services

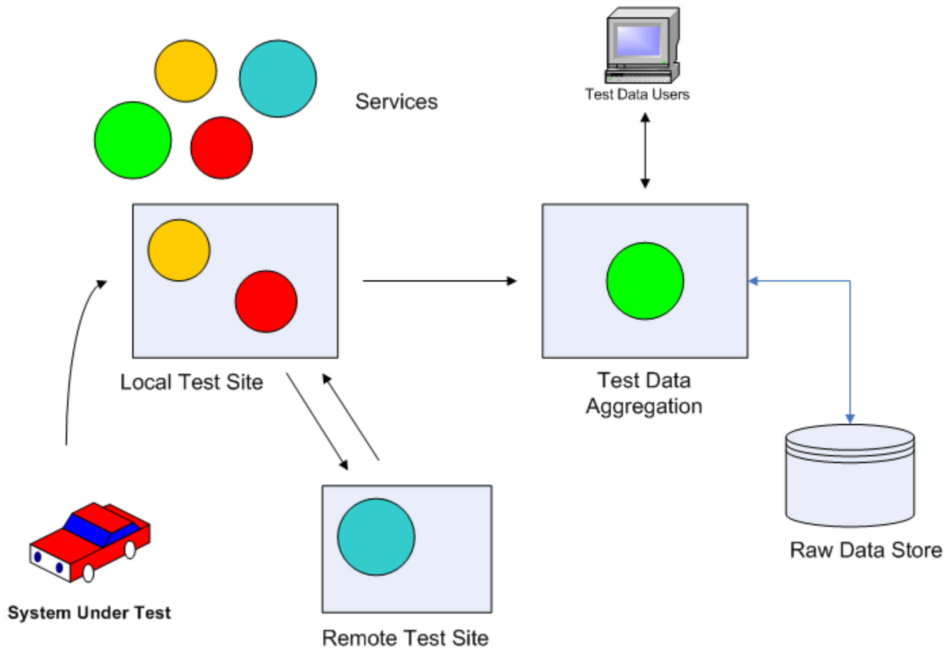


Figure 7 - SOA based ITS test environment

Figure 7 shows a basic architecture overview of the ITS Test Beds “system” with a focus on the integration of SOA concepts in the architecture. The Local Test Site runs a test case which includes the integration of own locally running services but also services made available by Remote Test Sites. Systems Under Test (SUT’s) send test messages to the local test site which after executing processes such as filtering, validation and certification forwards any relevant result data to the Test Aggregation Centre (TASC) which in its turn stores the data in the Raw Data Store.

Test messages in the domain of the ITS Test Beds project have a well defined structure with an identification section and a payload section which contains the actual test message. This message is transported through the local ESB and after some processing, depending on the nature of the test case, the test message is finally delivered to the Test Aggregation Service Centre where the message is sequenced into the test case cycle and stored in the Raw Data Store.
To assure the multi-purpose aspect of the system the ITS Test Beds project selected a number of typical ITS Use Cases covering a broad aspect of ITS Applications. Most of them have a very close relation to the activities of the eSafety forum. The eSafety oriented Use Cases are:

Use Case	Short Description
Automatic eCall	Automatic initiation and processing of an E112 call from a vehicle
eLane	Obtain lane position and update information with existing maps and negotiate driving trajectories with other vehicles
Incident Hazard Warning	Incident Hazard Warning based on an online "road incident database" maintained by Norwegian Radio P4
Speed Alert & Speed Database	Speed Alert is a system warning for speed limits when the speed limit is exceeded. It communicates with roadside systems (VMS, TMC) for variable limits and recommended speeds. The goal is to reduce the number of accidents due to speeding.

These Use Cases serve as benchmarks for the ITS Test Beds prototype

implementation to be constructed and are the subjects for the different test sites which are involved in the ITS Test Beds project.
The overall concept of ITS Test Beds is developed according to the FESTA handbook and is intended to deliver a technical and organizational framework for executing field operational tests in Europe.

The system components are shown in Figure 8.

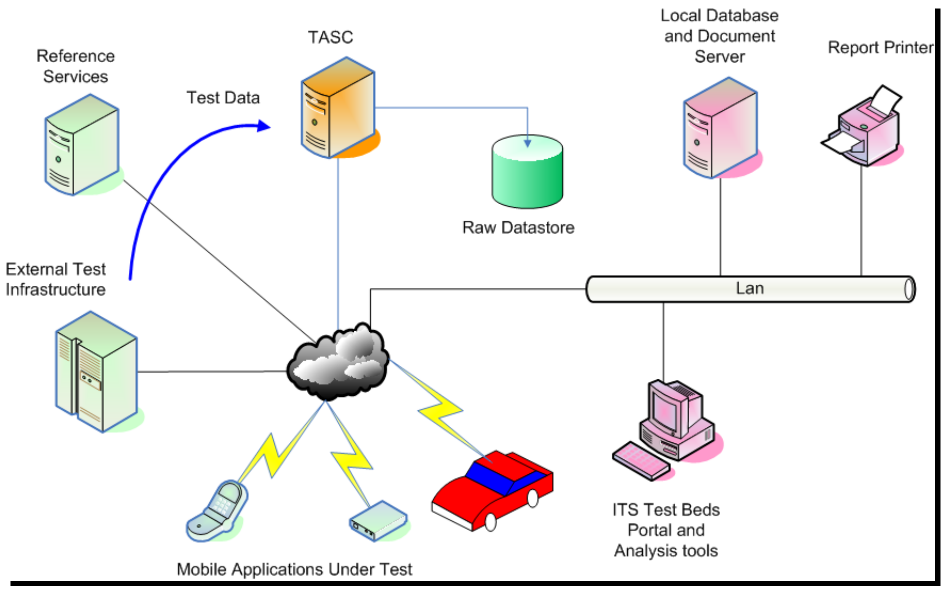


Figure 8 - SOA based ITS test environment: system overview

SOA like, loosely coupled services perform tasks like
- Validate a test message
- Enhance data content
- Write data to the core database

The system is designed in a flexible way and allows as well some services to either run locally or on a remote server.
BPEL and an ESB server are used to orchestrate messages in the system.

The resulting system show clearly advantages of the use of SOA for a robust and yet flexible system design that is still in wide areas “technology agnostic” and integrates well with other systems.

9 Domain model

Domains in SOA are used to cluster services and interfaces that have a common theme.

For eSafety, a look at the GST project shows the domains used at the Munich test site (and in the entire project as well):

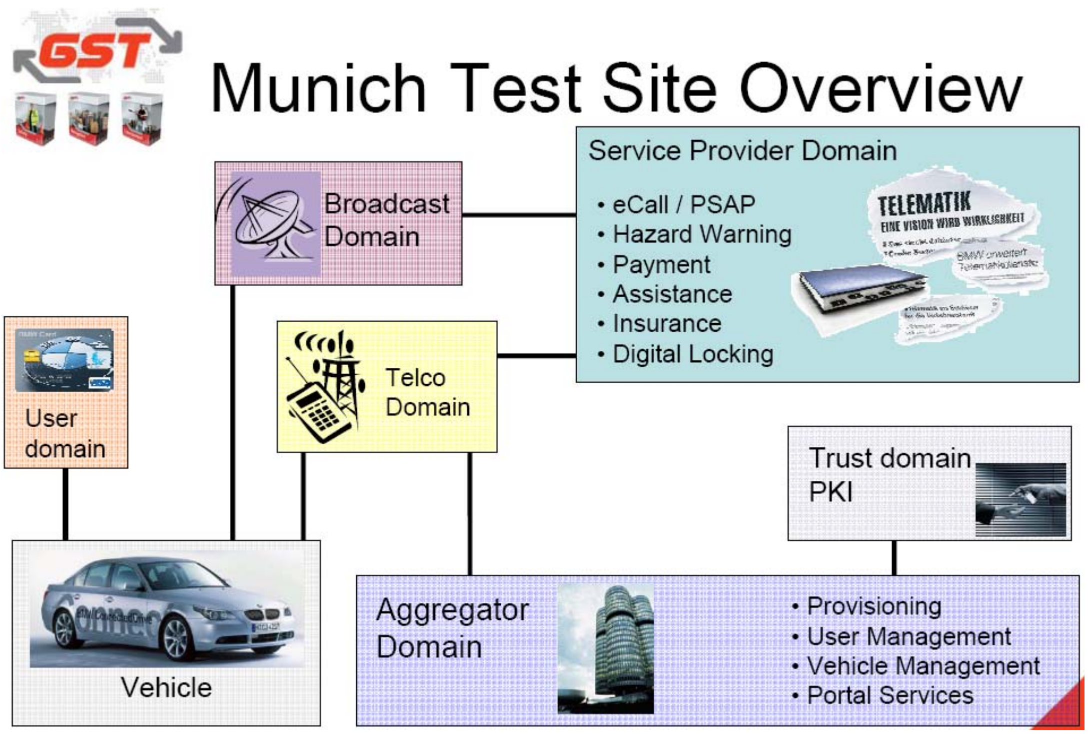


Figure 9 - GST domain model (Munich example)

Each of these domains may be company internal or shared between partners. It is a good design rule to define domains in a way that bundles interfaces and functions or services that have a similar purpose or common theme.
Domains shall have minimal number of interfaces between them but a lot of "interaction" internally. A second criterion is that domains shall contain at least a couple of interfaces / functions /services. If just some functions are important, they are better placed in an existing domain.

A generic, complete eSafety domain model may be too complex for this WG to introduce, but some core elements may be identified:

- in addition to the GST domain model, there will be several service provider and aggregator domains involved.

10 End2End SOA for eSafety

Typically, eSafety services may be accessed from different countries by a brought variety of users with heterogeneous interests based on their actual context.

This fact is illustrated in Figure 10 where a user is travelling through 3 countries, using for example a service on regional hazard warning.

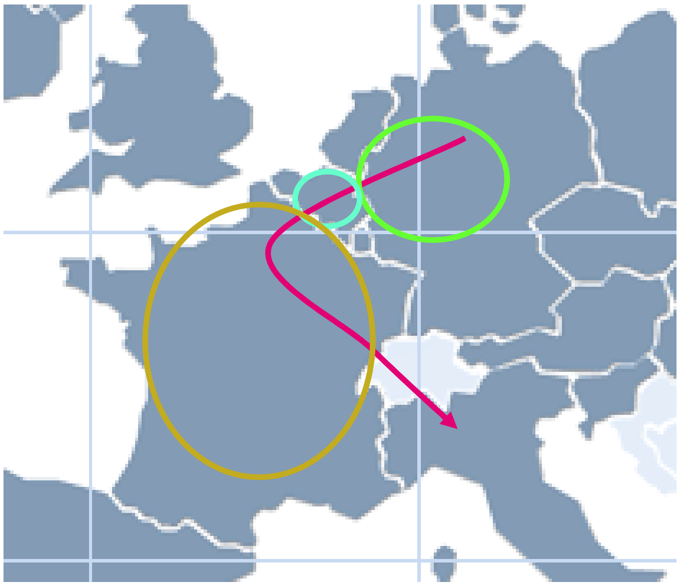


Figure 10 - Accessing services while roaming

Taking this into account and considering requirements identified in previous chapters, An End2End services oriented architecture is best suited to fulfil actual and future requirements and act as blueprint for a .Europe-wide services platform.

Due to mechanisms explained, “platform” may be the wrong term: This architecture defines how a service may be identified, invoked and finally used.

All components shown in Figure 11 are services themself.
As an enhancement of SOA principles mainly used within companies to perform economic transactions, services are used not only in the back-office / at the service provider but as well for communication, delivery, and even on the client.
The entire value chain of eSafety is entirely mapped to services that may be provided by changing stakeholders using published service interfaces.

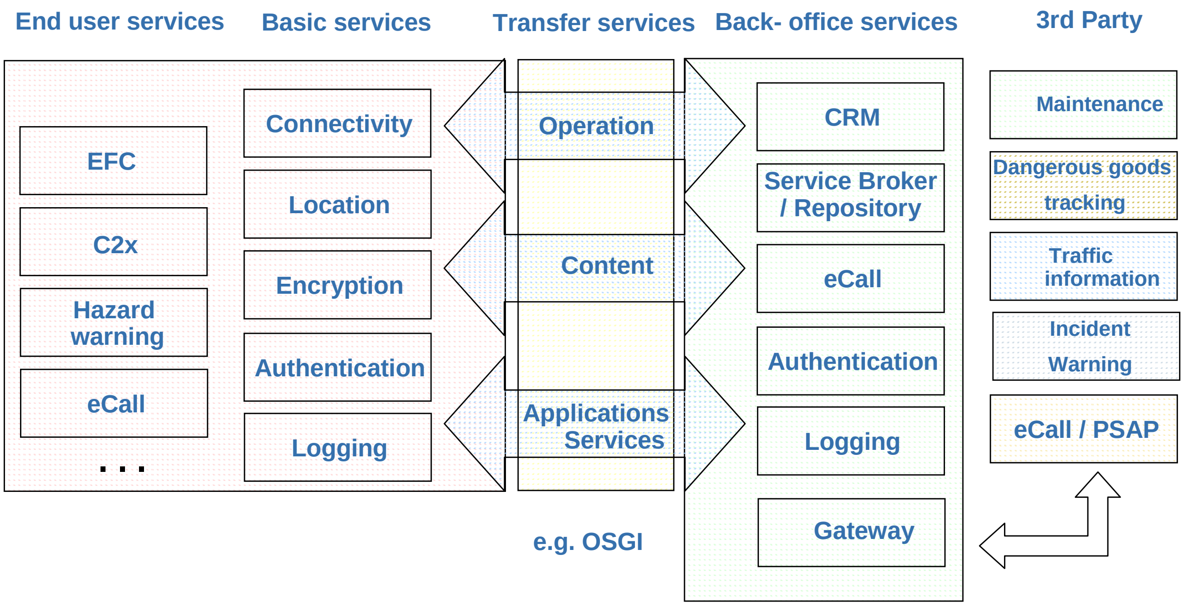


Figure 11 - Seamless End2End services architecture

This architecture model may be applied as well for C2C scenarios. The services running on the OBE need to be deployed, authorised, and managed.

C2C scenarios may be seen as part of more complex scenarios where client / server interaction is necessary.

11 "Importance of SOA for future eSafety processes"

The challenge for eSafety in the future will be multinational, multi-company driven. SOA supports easily instances of one service supporting different languages or service provider.
Most important hereby is the existence of directory services and public repositories. Domain and service type classification offer the ability to choose "best fit" services depending on individual needs.
Using such mechanisms, e.g. a vehicle maintenance service provider may therefore be changed automatically depending on regional settings.

Some important aspects of SOA for future eSafety processes include:

- Co-existence, interoperability and competition of services (or service elements),
- Faster introduction of new services (by re-use of existing service components)
- Higher quality and robustness
- Clear responsibilities and traceability of service availability and usage
- SLA agreements contracts possible
- Requirements on real-time / near time or a certain latency

12 Identify SOA activity fields for Stakeholders

If we look at SOA for eSafety, we have as well to look at possible value chains within that domain.
This analysis is closely linked to an analysis of what role or activity a stakeholder in this process may have.

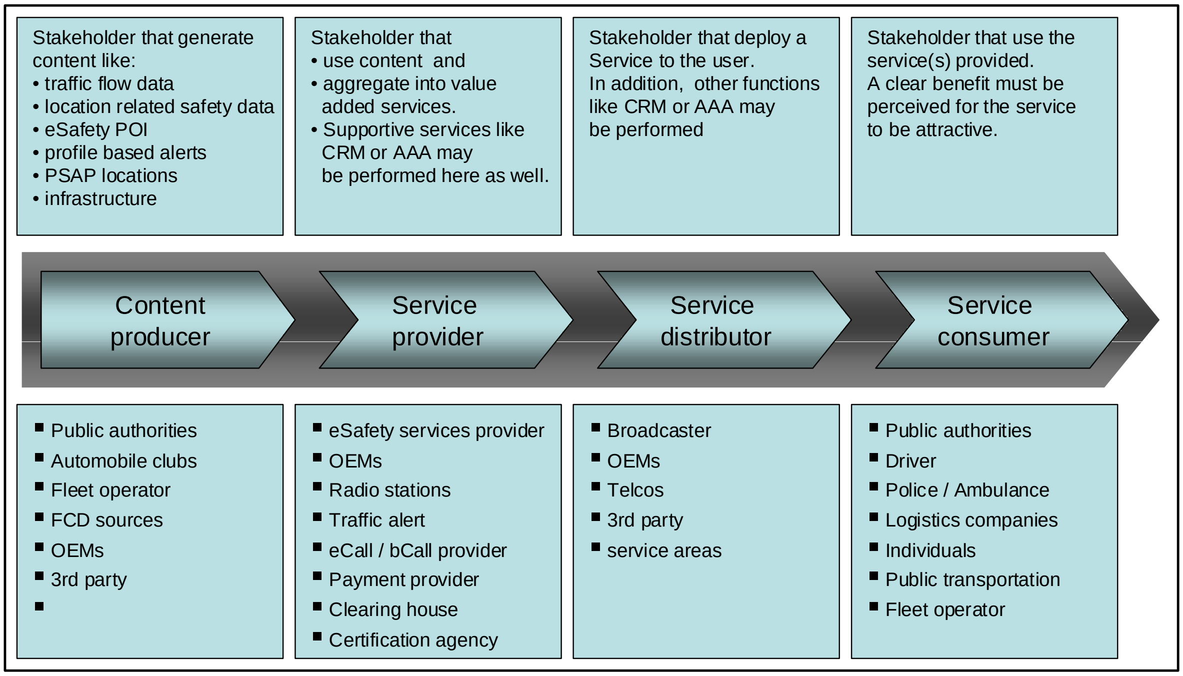


Figure 12 - The eSafety value chain

The figure above shows an ideal value chain for eSafety.
It may consist of many partners, some of them not even known to others when the business process for deployment of an eSafety service is set up.

The roles shown may be performed by one company at all or each by one or more.

In the future, an entire ecosystem of service providers with a variety of well defined specialised service elements will emerge.
The target solution for a certain user may then be combined "on the fly" using available services from changing providers.

13 Threats / Obstacles

This paragraph discusses some aspects that are observed as critical for SOA implementation decisions.
By giving possible solution for mitigating these threats, a more use case oriented discussion may start.

Companies have for years invested huge amounts of money of the last years in their IT infrastructure. It is well adapted to the internal business processes and supports as well information exchange with selected partners over well defined interfaces (EDI . .)
To change this infrastructure and follow the new paradigm seems not practical and expensive.
SOA may be implemented and used as well by simply adding a new interface to an existing software system. For new systems to be planned, SOA principles should be considered from the start.

A service oriented architecture is by definition distributed. Such systems require new and well suited tools for bug tracking and service validation and testing. Services are just described, their internals are not visible to the user or manager.
Typical tools to address these issues include TCP monitoring, and special SOA tools like SOAPScope or SOAtest.
If not understood and designed correctly, typical problems of distributed systems like transaction tracking (rollback) scalability and availability may occur.
Due to extensive XML message processing, and the number of parties involved, scalability and latency effects have to be considered.
These aspects have to be considered by design and resource allocation in an initial phase.
Responsibility for processes (safety related= special consideration on availability, robustness and trust).
SOA principles are useful within one company (mainly reusability and flexibility aspects) but are especially useful in co-operation of different partners.

13.1 NGTP (Next Generation Telematics Protocol)

The needs of industry and other stakeholders have lead to the development of other standards aiming in similar directions as SOA but merely focused on mobile telematics and eSafety applications.

BMW, in collaboration with telematics service providers (TSPs) Connexis and WirelessCar, has developed both a new telematics framework and a technology-neutral telematics protocol to bring greater flexibility and scalability to the industry. The result of this collaboration is a new open approach to implementing telematics services called the Next Generation Telematics Protocol (NGTP).

NGTP is a new approach for delivering over-the-air services to in-vehicle devices and hand sets alike, with the focus on open interfaces across the entire service delivery chain.

NGTP's developers set the following six objectives:

- Provide a technology-neutral protocol and consistent user interface for telematics services;
- Reduce barriers to collaboration and implementation;
- Enable adoption of new technologies as they come online;
- Support legacy systems for connectivity throughout the service life of a vehicle;
- Gain wide acceptance and encourage innovation through an open approach;
- Increase the value proposition for vehicle manufacturers, service providers, content providers, and motorists.

NGTP will enable vehicle manufacturers to use the best offerings from a variety of partners while maintaining a consistent driver experience. The new protocol will also allow TSPs and content providers to sell the same services to multiple vehicle manufacturers. Moreover, NGTP will support legacy systems, allowing older and newer vehicles alike to access new telematics offerings.

NGTP accommodates the EU's pending eCall initiative, and the protocol's open architecture will accommodate future industry trends.

Compared to the use of SOA, which is quite "universal" in its applications, NGTP is specially designed for telematics use.

NGTP offers synergies with a SOA based approach if business transactions and joint service provisioning is realised with SOA and telematics messages are exchanged using NGTP.
But: some NGTP services like the dispatcher are not stateless and their specification does not provide discoverability. Moreover, NGTP defines interfaces **between** subsystems and not just the interfaces **of** the subsystems. So NGTP does not meet all the SOA design criteria.

13.2 Different domains

Although Manufacturer and supplier use SOA now for many years as the standard means for providing commercial transaction support, SOA principles have not been commonly accepted as new paradigm by the ITS / Telematics community.
Possible synergies within companies are therefore not identified and business opportunities are lost.
This leads to a low acceptance and slow take-up of such principles.

14 Recommendations

This chapter gives some recommendations based on the findings of working group. The recommendations are focussed on eSafety support and are intended to maximise the benefit of SOA principles for this application area..

	Recommendation	Who
1	SOA should be strongly considered as a better alternative for implementing safety /ITS solutions compared to a centralised closed platform approach	EC Member states automotive industry suppliers service provider insurance companies motorway operators road safety organisations police and road authorities in Member States local authorities
2	Initiate a study which focuses on governance of the relationships among stakeholders (with respect to trust, quality, SLAs, commercial aspects, DRM, privacy)	EC Member states Local authorities Research community
3	Define a common ontology and semantic for describing eSafety services	EC Member states Local authorities Research community
4	Promote SOA to open legacy systems along the business needs of the eSafety services. Only add new interfaces where required (no complete IT change required, only partial updates)	EC automotive industry suppliers service provider
5	End to End seamless service orientation should be preferred; it offers a unified way of operation / governance.	EC Member states automotive industry suppliers service provider Research community
6	The technology to implement SOA oriented systems is available today and should be used. (e.g. BEPL4WS, XML, WSDL, Web services, SOAP, UDDI, OSGI, Java(6),…)	EC Member states automotive industry suppliers service provider Research community
7	Support a business oriented new approach to interoperability in CEN/ISO (as specified in the "Archetypical approach to the standardization" presentation made by N. Curci on behalf of UNI/UNINFO in ICTSB/ITSSG #14 held in November 2008).	EC Member states Standardisation for a Research community

8	Testing, validations s& certification are core features of eSafety SOA and should supported by stakeholders from beginning	EC Member states automotive industry suppliers service provider Research community
9	Services should be designed with suitable granularity, with well defined functionality allowing for good service orchestration and performance	automotive industry suppliers service provider Research community
10	- A good analysis of own core business (be it existing or desired) is crucial for modelling of services and interfaces	automotive industry suppliers service provider
11	Concentrate SOA interfaces on business processes and transactions support as a variety of dedicated telematics centred protocols have emerged that are well suited and specially designed to fulfil e Safety requirements	automotive industry suppliers service provider Research community
12	Support the evolution of SOA for ITS/eSAfety by supporting projects which implement a core European registry of eSafety/ITS services (UDDI, Broker, ontology . .)	EC Industry Research community
13	Promote the use of SOA test bed for validation and certification (ITS test beds project)	

15 References, Links.

"SOA for Automotive" Initiative

"SOA For Automotive", a joint activity of University of St. Gallen, BMW AG, Hella KGaA, Magna Steyr Fahrzeugtechnik, Siemens VDO, Supply On and ZF Friedrichshafen AG, currently investigates a Service-oriented approach for process integration between OEMs and suppliers. The project builds on SOA concepts and the idea of making Web Services available to external partners: The group has been leveraging the emerging VDA recommendation 4965 on Engineering Change Management for defining a "public process". This process definition was then translated into a Web Service design which allows for coupling IT systems based on SOA concepts and open internet standards.
Starting in October 2005, the group came up with their "interoperability profile" this summer - consisting of the public process, an organizational and informational model as well as the ECR (Engineering Change Request) Business Service. Piloting has started in October and the participating companies are now in the process of implementing the interoperability profile using SOA platforms and PLM systems from different vendors.

Ultimately, the group intends to demonstrate that service-based process integration will help Automotive companies to become more interoperable. The project partners expect the interoperability profile to reduce the need for bilateral agreements and to thereby minimize costs and effort of electronic B2B integration. However, this approach requires the combination of business standards (on the level of cross-organizational business processes and business semantics) with open web service standards.

http:// soa.iwi.unisg.ch
http://new.eic-community.org/index.php?option=com_content&task=view&id=144&Itemid=417

A Service Oriented Architecture Based eCall Implementation

(Presentation session by Oracle at "the fully networked car", please contact Oracle for further information)

The Volkswagen collaboration platform:
http://www.vwgroupsupply.com/b2b/vwb2b_folder/supplypublic/en/platform.html

Municipal Infrastructure Data Standard (MIDS)

Keynote Interop-ESA 05: Enterprise Interoperability and ICT, an EU perspective By Gérald Santucci, DG Information Society and Media – D5 ICT for enterprise networking.

http://www.explore-soa.de
The SOA blog of T-Systems

http://www.oracle.com/technologies/soa/index.html
A starting point towards Oracle's SOA offerings

http://www.esafetysupport.org/
Home page of eSafety support

http://www.oracle.com
Home page of Oracle (Co-chair)

http://www.t-systems.com
Home page of T-Systems (Co-chair)

http://www.ngtp.org
The NGTP homepage

http://www.soa-know-how.de
A good introduction to SOA

http://www.whatissoa.com/
Good and detailed overview of SOA concepts and components

http://www-01.ibm.com/software/solutions/soa
IBMs view on SOA

http://www.bitkom.org/de/wir_ueber_uns/18151.aspx
The German Bitkom SOA working group

http://www.sap.com/platform/soa/index.epx
A starting point towards SAP´s SOA view and offerings

16 List of figures

17 List of members of the WG-SOA and cc: recipients

Bohnert, Thomas Michael	thomas.michael.bohnert@sap.com
Calado, João	jcalado@sapo.pt
Camolino, Rui Dias	Rui.Camolino@brisa.pt
Carrotta, Alessandro	a.carrotta@esafetysupport.org
Curci, Natalino	n.curci@polidream.it
Cyran, Yolande	yolande.cyran@ec.europa.eu
Davila - Gonzalez, Emilio	Emilio.Davila-Gonzalez@ec.europa.eu
Emmanuel Bloch	Emmanuel.Bloch@t-systems.com
eSafety Support	info@esafetysupport.org
Foster, Howard	howard.foster@imperial.ac.uk
Gaillet, Jean-François	gaillet@ygomi.com
Helbig, Thomas Helbig	Thomas.Helbig@bmw.de
Jan Gacnik	Jan.Gacnik@dlr.de
Janssen, Hartmut	Hartmut.Janssen@toll-collect.de
Kenis, Eric	ericjm.kenis@mow.vlaanderen.be
Kompfner, Paul	p.kompfner@mail.ertico.com
Mayer, Dr. Joachim	joachim.mayer@daimlerchrysler.com
McClure, David	davidmcclure@sbd.co.uk
Mehl, Frédéric	Frederic.Mehl@t-systems.com
Michel, Hans-Ulrich	Hans.Michel@bmw.de
Mundin, Thomas	thomas.mundin@t-systems.com
Potters, Paul	potters@connekt.nl
Rataj, Jürgen	Juergen.Rataj@dlr.de
Reinhardt, Dr.Wolfgang A.	wr@acea.be
Rosines, Francesc	francesc.rosines@atosorigin.com / francesc.rosines@atosresearch.eu
Ruggiero, Michelangelo	m.ruggiero@cobra-at.com
Sansone , Fulvio	fulvio.sansone@oracle.com
Schild, Dr. Klaus	klaus.schild@toll-collect.de
Silva, Irina	i.silva@esafetysupport.org
Spell , Dr. Sabine	Sabine.Spell@volkswagen.de
Stenberg, Patrik	patrik.g.stenberg@volvo.com
Trenor, Tomás	Tomas.Trenor@robotica.uv.es
Valtchev, Dr. Dimitar	d.valtchev@prosyst.com
van de Kraats, Henri	h.vandekraats@imabenelux.com
Vermassen, Erwin	evermassen@nimera.net
Vierroth, Volker	volker.vierroth@t-systems.com
Vila, Marta	Marta.Vila@robotica.uv.es
Wiecker, Martin	mwiecker@ford.com