

NEXOF-RA

NESSI Open Framework – Reference Architecture

IST- FP7-216446



Deliverable D10.1 Requirements Report

Consorzio Interuniversitario Nazionale per l'Informatica (leader),
British Telecommunications, Logica, MoMa – Modelli matematici ed
applicazioni, Siemens, Thales, TIE

Due date of deliverable: 31/03/2009

Actual submission date: 31/03/2009

This work is licensed under the Creative Commons Attribution 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

This work is partially funded by EU under the grant of IST-FP7-216446.

Document Information

IST Project Number	FP7 – 216446	Acronym	NEXOF-RA
Full title	NESSI Open Framework – Reference Architecture		
Project URL	http://www.nexof-ra.eu		
EU Project officer	Arian Zwegers		

Deliverable	Number	D10.1	Title	Requirements Report
Work package	Number	10	Title	Requirements and Assessment Criteria

Date of delivery	Contractual	31/03/2009	Actual	31/03/2009
Status	Version 2, dated 31/03/2009		final <input checked="" type="checkbox"/>	
Nature	Report <input checked="" type="checkbox"/> Demonstrator <input type="checkbox"/> Other <input type="checkbox"/>			
Abstract (for dissemination)	The document describes the scenarios and the related requirements for the Reference Architecture.			
Keywords	Scenarios, requirements			

Internal reviewers	Reviewer1 (Organization)		
	Reviewer2 (Organization)		
Authors (Partner)	CINI (leader), BT, Logica, Moma, Siemens, Thales, TIE		
Responsible Author	Alberto Sillitti		Email asillitti@unibz.it
	Partner	CINI	Phone

EXECUTIVE SUMMARY

In general, a requirement describes a condition or capability to which a system must conform, either derived directly from user needs, or stated in a contract, standard, specification, or an otherwise obligatory document.

The objectives of this work package are to align the project to the requirements of the market. Therefore, the purpose of this document is to collect and formalize the requirements concerning the delivery of services to users from a service-centric (potentially Web based though not exclusively web based) perspective.

This deliverable describes 17 scenarios collected by the project partners and a list of requirements derived and prioritized from them in order to provide the necessary guidance to the work performed by the “Reference Architecture: Specifications” track with respect to the design of the reference architecture of the Open Service Framework.

This work has thus been in close cooperation with the “Reference Architecture: Specifications” track in order to subsequently ease the mapping of the collected requirements to its promoted layered (functional) architectural view.

The definition of requirements on base of scenarios was based on the approach promoted by the long-term research project ‘CREWS’ – Cooperative Requirements Engineering With Scenarios (European Commission ESPRIT 21903) [6].

The considered state-of-the-art includes the SeCSE Project [1], the IEEE Standard 1228-1994 [2], the Serenity Project [3], the Enterprise Grid Alliance [10], the Open Grid Services Architecture [11], and the C-Cubo framework [4].

The so obtained requirements were prioritized and classified according the concerns identified by the Reference Architecture: Specifications” track.

The NEXOF Reference Architecture must not be a static specification and must allow for the integration of changes and evolutions which result from research or changes in the state of practice; this will be taken into account through the Open Requirements Process also defined in this document.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	3
TABLE OF CONTENTS	4
1 INTRODUCTION.....	7
2 STATE OF THE ART REQUIREMENTS.....	10
2.1 Service publication	10
2.2 Service dependability	11
2.2.1 Availability requirements	12
2.2.2 Reliability requirements.....	12
2.2.3 Safety requirements.....	13
2.2.4 Security and trustworthiness requirements	14
2.2.5 Service performance requirements	15
2.3 Service monitoring.....	15
3 SCENARIOS	18
3.1 Service procurement (S1)	24
3.2 Service lifecycle support (S2)	27
3.3 Management services for grid and service platforms (S3)	31
3.4 PhiMas: personal health information monitor and alert service (S4) ...	34
3.5 Collaborative e-learning scenario (S5)	37
3.6 Deployment and configuration of a generic platform (e.g. an e-Learning one) (S6).....	42
3.7 e-Health: complex diagnostic workflow (S7).....	46
3.8 e-Health: assisted living (S8).....	49
3.9 Traffic management: large scale emergency handling (S9)	51
3.10 Crisis management system of systems (S10).....	54
3.11 Effective and efficient collaborative decision making (S11)	59
3.12 e-Commerce information sharing (S12).....	61
3.13 Mobile office for an owner of a micro-enterprise (S13)	69
3.14 Safety at work in the construction sector (S14)	75
3.15 e-Government online application submission service (S15)	78
3.16 e-Government online fee visualization and payment Service (S16) ..	82
3.17 Assisted Industrial Maintenance (using 3D virtual environments) (S17)	86

4 REQUIREMENTS COLLECTED BY PARTNERS	89
4.1 SLA processing (R1)	95
4.2 Uniform service representation (R2).....	96
4.3 Service discovery mechanisms (R3)	96
4.4 Decentralised architecture (R4).....	97
4.5 Service description (R5)	97
4.6 Service deployment (R6).....	98
4.7 Service decommissioning (R7).....	98
4.8 Interoperability and flexible communication standards (R8)	99
4.9 Federated (temporal) identity management (R9).....	99
4.10 Location based routing (R10)	100
4.11 Services integration by semantic mash-up (R11)	100
4.12 Harmonization of several heterogeneous information sources (R12)	101
4.13 Unified communication (R13)	101
4.14 Integration of an application with legacy applications, ERP, etc. critical and foremost for the client. (R14)	102
4.15 Adaptive deployment (R15)	102
4.16 Workflow management and integration (R16)	103
4.17 Aided configuration (R17)	104
4.18 Modelling capabilities (R18)	104
4.19 Technical interoperability (R19)	105
4.20 Device integration / vertical integration (R20).....	105
4.21 Distributed workflow (R21).....	106
4.22 Stateful, device adaptive service transfer (R22)	106
4.23 Adaptability (R23)	107
4.24 Rapid reconfiguration (R24)	107
4.25 Integrity (self-diagnosing and self-healing) (R25)	108
4.26 Dependability for device integration (R26).....	108
4.27 Compliance to privacy, and security policies s (R27)	109
4.28 Collaborative business process acquisition, modelling and effective management (R28).....	110
4.29 Distributed architecture (R29).....	111
4.30 Integration of services (R30).....	111

4.31 Monitoring and reliability (R31)	112
4.32 Orchestration (R32)	112
4.33 Trust and confidence (R33)	113
4.34 Service discovery (R34)	114
4.35 Information as a service (R35).....	114
4.36 Execution of human-based process steps (R36)	114
4.37 Information integration (R37)	115
4.38 Distributed transaction support (R38)	115
4.39 Non repudiability of data transfer (R39)	116
4.40 Cross-certification (R40)	116
4.41 Resilience & Continuity of service (R41)	117
4.42 3D virtual environment services support and integration (R42)	117
5 REQUIREMENT CLASSIFICATION	119
5.1 Services value stack.....	119
5.2 Classification of requirements	124
6 REQUIREMENT PRIORITIZATION	128
7 OPEN REQUIREMENTS PROCESS.....	131
8 INPUT FROM THE INDUSTRIAL ADVISORY BOARD	132
8.1 Wider Issues.....	132
8.2 Specific issues.....	132
9 CONCLUSION.....	134
10 REFERENCES.....	135
A1 GUIDELINES FOR DERIVING REQUIREMENTS FROM SCENARIOS.....	136
A2 TEMPLATES FOR QUESTIONNAIRES	138
A2.1 Template for collecting scenarios	138
A2.2 Template for collecting requirements	139

1 INTRODUCTION

In general, a requirement describes a condition or capability to which a system must conform, either derived directly from user needs, or stated in a contract, standard, specification, or an otherwise obligatory document.

The purpose of this document is to collect and formalize requirements concerning the delivery of services to users from a service-centric (potentially Web based though not exclusively web based) perspective.

The objectives of the “Requirements and Assessment Criteria” track are to align the project to the actual needs of the market, and thus, the “Requirements and Assessment Criteria” track considers requirements not only from the state of the art but also through the analysis of scenarios stated by the project partners.

The consideration of requirements from the state of the art allows us to establish a base line of requirements characterizing SOA based architectures.

The analysis based on scenarios allows us to focus not only on the final output – the reference architecture – but also on the environment in which it has to be used (i.e. the user, the system, the context, the activities) [8]. Additionally, this scenario-based approach will allow a better validation and verification of the derived requirements within our work on D10.2 (Assessment criteria).

This deliverable describes scenarios collected and a list of requirements derived and prioritized from them in order to provide the necessary guidance to the work performed by the “Reference Architecture: Specifications” track with respect to the design of the reference architecture of the Open Service Framework.

This work been done in close cooperation with the “Reference Architecture: Specifications” track in order to subsequently ease the mapping of requirements collected to layered (functional) architectural view promoted.

This work package contacts potential NEXOF stakeholders and those putting forward the specific cases to determine what they feel to be important and what might be the priorities (i.e. must, should, may have). The result of this activity is a list of scenarios from the different stakeholders. Those scenarios will in some cases be vague and will include different standpoints, interpretations, opinions, and needs. The scenarios and requirements are those considered important by the NEXOF-RA partners, who are involved in different domains, and that invest in the project.

Scenarios and requirements are provided as input to the architecture and not as a basis for the proof of concept.

Beyond the scenarios and requirements collected in this document, the Open Requirements Process (see section 7) will allow collecting more scenarios and requirements to obtain a more complete picture of the demands of the market.

The definition of requirements on base of scenarios is based on the approach promoted by the long-term research project 'CREWS' – Cooperative Requirements Engineering With Scenarios (European Commission ESPRIT 21903) [6].

The rationale of having our NEXOF-RA requirements collection based on CREWS project's approach is that it provides effective methods and tools particularly well suited to the **cooperative scenario-based elicitation, collection** and validation of requirements coming from **multiple stakeholders**.

The CREWS project points out the “lack of both formal product models and guidelines to support the process of developing scenarios” [7]. It further states that “users request more explicit methodological guidance and more adequate tool support” [7]. This **methodological guidance** is given by [6] where requirements generation through the elicitation of scenarios is described considering four aspects:

- **Purpose:** “why is a specific scenario used?”
- **Contents:** “what is the knowledge expressed in a scenario?”
- **Form:** “which is the best form to express a scenario?”
- **Lifecycle:** “how is a scenario manipulated throughout its life cycle?”

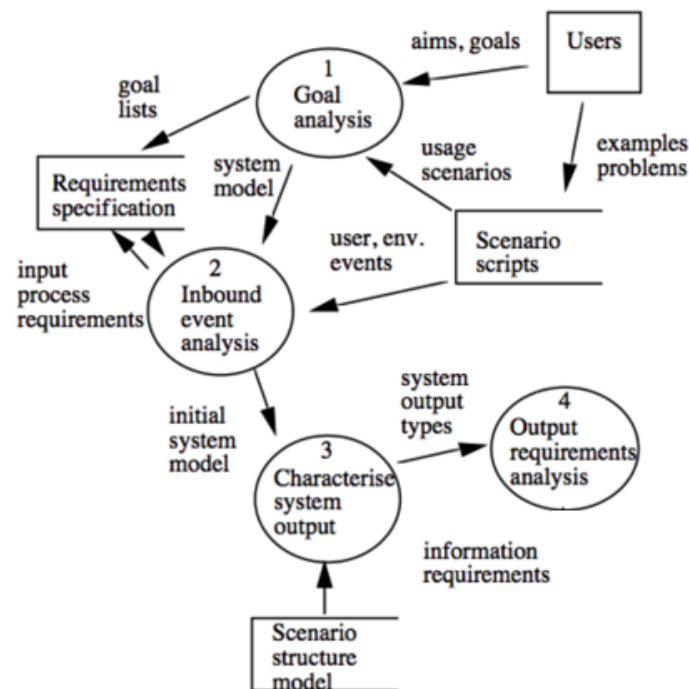


Figure 1: Outline of the method to derive requirements from scenarios (adapted from [6])

For our purposes, it was decided to use a simplified version of the process described in [6] (see figure 1) to extract the requirements from the scenarios focusing on the ones with direct impact (through demands of functionalities expressed) regarding our work for what concerns the design of Reference Model and Reference Architecture specifications on an Open Service

Framework that NEXOF-RA Project is all about. This simplified version omits the analysis of stakeholder benefits (i.e. if the stakeholders will be able to fulfil their requirements using the proposed reference architecture). This, because the definition of adequate assessment criteria will be part of the deliverable D10.2 (Assessment criteria) and the analysis and demonstration of the feasibility of the proposed reference architecture will be addressed in the “Proof-of-concept “ track.

Within paper [6] already mentioned above, scenarios are defined as “facts describing an existing system and its environment including the behaviours of agents and sufficient context information to allow discovery and validation of system requirements”. Two types of scenarios are used: “scenario scripts” that describe system usage; and “scenario structure models” that contain facts about the system environment.

Initially the user’s goals are analyzed (step 1) to check whether they are supported the requirements obtained by the state of the art. This creates a first version of the requirements specification (coming from the analysis of scenarios) describing the high-level system processes.

The requirements specification is refined analyzing the dependencies between the inbound events (step 2) described in the scenarios and the requirements functions. This identifies the requirements that have to deal with inbound events coming from users and different types of system environments as well as unexpected events.

Inbound events are defined in scenarios, which define the interaction between users and the proposed system. This analysis will guide decisions about the extent of the automation and the user-system boundary.

System output will be motivated by the analysis of the user goals in step 1 and described in terms of process and outline content in the requirements specification. This enables acceptability and impact of system output on users to be assessed, first by specifying the output in more detail in step 3 and then analyzing the requirements to support user tasks in step 4.

These steps are followed by identifying, which stakeholders receive what system output and the use they make out of it.

2 STATE OF THE ART REQUIREMENTS

This section summarizes the requirements collected from the state-of-the-art. Thereby, the requirements provided by the SeCSE Project [1] were taken as a starting point; in a second step, the requirements gathered by the IEEE Standard 1228-1994 [2], the Serenity Project [3], the Enterprise Grid Alliance [10], the Open Grid Services Architecture [11], and with the C-Cubo framework [4] were added and integrated.

Requirements that were obtained from state of the art requirements for existing implementations, were abstracted to be used as a requirement for NEXOF-RA.

The state of the art requirements were organized as follows:

- Service publication
- Service dependability
- Service monitoring

2.1 Service publication

As stated in the NEXOF-RA glossary¹, service publication is defined as any action to expose the service description. Moreover, a service description is defined as a set of documents that describe the interface, the accessibility and the capability of a service.

Table 1: Requirements for service publication

ID	Short name	Description
L1.1	Data model	The data model used to register information about services must be rich enough (and possibly extensible) to comprise all of the information needed to enable effective service publication (and discovery).
L1.2	Publication APIs	The APIs offered by the reference architecture must offer all the needed facilities for service publication.
L1.3	Registry architectures	Different registry architectures (centralized, federated and decentralized) should be supported as well as being able to publish/search for metadata directly without publication in formal registries
L1.4	Push mode publication mechanisms	Push mode publication (and discovery) methods should be supported.

¹ See NEXOF-RA deliverable 6.1: "Reference Architecture Model"

ID	Short name	Description
L1.5	Pull mode publication mechanisms	Pull mode publication (and discovery) methods should be supported.
L1.6	Categorization and selective publication	The data model used to describe services must allow for the categorization and the specification of publication criteria (e.g., visibility, scope, constraints). This information should be available in formalised (e.g. semantic ontologies) as well as non formal formats for human readers
L1.7	Up-to-dateness of published information	The data model used to store information about services should properly handle the published data in order to avoid or limit the amount of outdated or unreliable information.
L1.8	Scalability	The implementation that adheres to an instance of the reference architecture should scale well
L1.9	Applicability	The reference architecture should be applicable both in local and in federated contexts.
L1.10	Mobile users	Attention to the needs and requirements of mobile users should be paid (device and channel, etc.).
L1.11	Extensibility and evolution	The reference architecture should enable the seamless integration of different architectures and technologies. The reference architecture must be extensible and capable to evolve.
L1.12	Support to legacy systems	It should be possible for legacy systems to become part of a system built using the reference architecture without excessive impediments.

2.2 Service dependability

Dependability describes the trustworthiness of a computing system that allows reliance to be justifiably placed on the service it delivers. It is also used as the collective term gathering the following attributes or non-functional requirements: availability, reliability, maintainability, safety, and security.

In the following subsections, requirements from the state of the art of these types are considered:

- Availability requirements
- Reliability requirements
- Safety requirements
- Security and trustworthiness requirements
- Service performance requirements

2.2.1 Availability requirements

Availability describes the proportion of the time in which the system is operational, and satisfies its specification. It is quantified as the uptime divided the uptime plus downtime.

Table 2: Requirements for service dependability: availability requirements

ID	Short name	Description
L2.1	Registry availability	This is a high level requirement regarding the accessibility of a service. For a service to be available it necessarily has to be accessible through the means of an addressing mechanism.
L2.2	Registry correctness	Since registry availability is a prerequisite of service availability, registries are often replicated. In such cases, updates of service information must be performed correctly and spread timely across all replicas.
L2.3	Monitoring information	There is a requirement for tight availability monitoring of services (dependent upon the nature of the service). This monitoring information must be easily accessible by the reference architecture and also delivered in a timely fashion.
L2.4	Service availability	Services may be replicated to increase availability; the reference architecture may need to make an efficient use of these replicas and as such must support different mechanisms for replicating services.
L2.5	Workflows	The availability of services (i.e., the use of replicated services) may be included in workflows; the reference architecture should allow an extensible approach to workflow handling to allow as many workflow mechanisms as possible to interact with the reference architecture.
L2.6	Service addressing	Services are currently addressed via a URI; however this may not always be the case. To future-proof the reference architecture the addressability of services and registries may need to be taken into account.

2.2.2 Reliability requirements

Reliability is the ability of a service to perform a required function under stated conditions for a specified period of time.

Table 3: Requirements for service dependability: reliability requirements

ID	Short name	Description
----	------------	-------------

ID	Short name	Description
L3.1	Atomic transaction notifications	The reference architecture must be able to tell when atomic transactions have either successfully completed or failed, or if there is to be a delay in the transaction.
L3.2	Long running transaction notifications	The reference architecture must be able to tell how long a long running transaction may take (i.e. when a timeout should occur)
L3.3	Co-ordination semantics	The reference architecture may have to be aware of the status and performance of the co-ordination of participating services within a composition.
L3.4	Awareness of unreliable services	The reference architecture may need to be aware of unreliable services, or tell when a service becomes unreliable or when it fails to deliver what is expected. This may entail a feedback mechanism that handles receipts of service feedback from clients.

2.2.3 Safety requirements

Safety describes the condition of a system of being protected against physical, social, spiritual, financial, political, emotional, occupational, psychological, educational or other types or consequences of failure, damage, error, accidents, harm or any other event which could be considered non-desirable.

Table 4: Requirements for service dependability: safety requirements

ID	Short name	Description
L4.1	Testability	For a system implemented using the reference architecture it should be possible to verify that safety requirements have been correctly implemented and the software functions safely within its specified environment.
L4.2	Risk determination	The reference architecture should assure support to determine the risk, i.e. "the likelihood that a system hazard will cause an accident and the severity of that accident"
L4.3	Determination of failure consequences	The reference architecture should assure support to determine consequences of a failure of the system, e.g. to understand which part of the service will not be available due to the failure of a subpart.

2.2.4 Security and trustworthiness requirements

Security covers aspects of:

- **Authentication:** the process of verifying that a potential partner in a conversation is capable of representing a person or organization
- **Authorisation:** the process of determining, by evaluating applicable access control information, whether a subject is allowed to have the specified types of to a particular resource. Usually, authorization is in the context of authentication. Once a subject is authenticated, it may be authorized to perform different types of access
- **Confidentiality:** ensuring, that information is accessible only to those authorized to have access
- **Integrity:** assuring information will not be accidentally or maliciously altered or destroyed
- **Non-repudiation:** a method by which the sender of data is provided with proof of delivery and the recipient is assured of the sender's identity, so that neither can later deny having processed the data
- **Denial-of-service:** protection against attacks
- **Privacy:** protection of personal infosphere

Table 5: Requirements for service dependability, security and trustworthiness requirements

ID	Short name	Description
L5.1	Basic security features	The reference architecture should assure a basic security support. Security services such as encryption, authentication, authorization and non-repudiation should be guaranteed for the publication process.
L5.2	Interoperability between different security domains and infrastructures	The reference architecture should support the creation of secure communication channels between cross-organizational services, in order to overcome possible heterogeneity aspects concerning the security domains (i.e., a context defined by a security architecture to include a set of system entities that have access to resources) and the security infrastructures (i.e., mechanism and technologies used for managing participants and resources).
L5.3	Support to different kinds of Certificate Authorities (CA)	The architecture should support different kinds of CAs, each one potentially capable of issuing one or more kind of credentials. A credential is a statement (for instance, an authentication statement, an authorization statement, or a statement of any other kind) about a subject issued by a trusted third party (that is, the CA).

ID	Short name	Description
L5.4	Use of ratings for evaluating trustworthiness	A service consumer may publish ratings about a service provider or a particular service on the basis of their own experience. The data model used to register information about services should take into account also the various ratings received by a provider in order to establish its reputation.
L5.5	Privacy by contract	A user may wish to define by contract usage of his/her personal data as well as obsolescence of these data and be confident in their storage on the service provider side

2.2.5 Service performance requirements

Performance gives a quantification of the goodness of the service provided by a system. Performance involves metrics such as response time, throughput, reliability, etc.

Table 6: Requirements for service dependability, service performance requirements

ID	Short name	Description
L6.1	Limitation of communication overhead	The reference architecture needs to add as little overhead as possible to communications between clients and services. There may even be a need for caching some information being transacted and even the recognition of useful information such as traffic conditions so repeat service calls can be avoided, however payment and service usage information will still need to be transferred to services so payments and usage information can be carried out and monitored.
L6.2	On-the-fly service switching	This is the ability of the reference architecture to switch between services, either replicated or similar services (i.e., services that are differently constructed yet provide the same output).
L6.3	Scalable performance and throughput	The service reference architecture should allow high performance and throughput.

2.3 Service monitoring

Service monitoring is an activity that provides an awareness of the state of a service.

Table 7: Requirements for service monitoring

ID	Short name	Description
L7.1	Functional and non-functional qualities	It should be possible to define and monitor both functional and non-functional qualities.
L7.2	Event based monitoring	Non-intrusive event based monitoring should be supported.
L7.3	History based monitoring	History based monitoring should provide information on multiple executions of services over a reasonable time span. This information should provide a valuable input to service providers for their analysis of existing services in order to determine what services could be proposed in the future.
L7.4	Assertion based monitoring	The reference architecture should allow assertion based monitoring.
L7.5	Environmental Monitoring	Monitoring should notice changes in the set of available services. If new services appear and they provide better functional and/or non-functional qualities, monitoring could incite, following the respective policies, a switch to the new services.
L7.6	Dynamic regulation of monitoring activities	It should be possible to dynamically regulate the quantity and types of monitoring activities that are performed at runtime. Different kinds of monitoring activities should be modular in order to form a personalized monitoring strategy.
L7.7	Service Data Collection	The reference architecture should provide access to dynamic information regarding the execution of service centric systems to all the monitoring components of it at run-time.
L7.8	Environmental Data Collection	Collection of environmental data is necessary for assertions regarding values that are not obtainable from the service execution.
L7.9	Dynamic renegotiation of constraints	It should be possible to dynamically renegotiate monitoring constraints to better tailor context adaptivity. Human intervention could be necessary.
L7.10	Extensible framework for data collectors, data analyzers and constraint metrics	The reference architecture should permit the definition of new metrics for defining constraints, of appropriate data collectors and of the appropriate data analyzers.

ID	Short name	Description
L7.11	Process instrumentation through appropriate development time tools	Process instrumentation through graphical user interface for monitoring purposes should be provided.
L7.12	Multi level constraints definition	Constraints should be definable at different abstraction levels. End users should be able to work at an abstraction level adequate to their level of know-how. Designers should be able to work at a more specific level. Development time tools should provide for such flexibility in defining monitoring constraints.
L7.13	Adaptability	The reference architecture should ensure a high level of adaptability with regards to the number and the diversity of already existing and future SOA architectures. In particular, it (in its' entirety or as in some of its' subparts) should be easy to integrate into the various execution environments.
L7.14	Monitoring process	To ensure service level and quality, it is necessary to follow an IT process aligned to business goals that provides guidance through the entire service lifecycle.
L7.15	Compliance-based service monitoring	Monitoring of Security and Privacy from relevant perspectives (e.g. Regulations, Standards, Contracts)

3 SCENARIOS

The purpose of this section 3 is to collect and formalize requirements analyzing user scenarios. For the purposes of this document, scenarios are defined as “facts describing an existing system and its environment including the behaviour of agents and sufficient context information to allow discovery and validation of system requirements” [6].

The scenarios (and the connected requirements) have been collected by experts of their field. The contributions are listed in the table below.

Table 8: Contributions by the project partners

ID	Scenario	Partner
S1	Service procurement	British Telecommunications
S2	Service lifecycle support	
S3	Management services for grid and service platforms	Logica
S4	PhiMas: personal health information monitor and alert service	
S5	Collaborative e-learning scenario	MoMa – Modelli matematici ed applicazioni
S6	Deployment and configuration of a generic platform	
S7	e-Health: complex diagnostic workflow	Siemens/Thales
S8	e-Health: assisted living	Siemens/Thales
S9	Traffic management: large scale emergency handling	Siemens
S10	Crisis management system of systems	Thales
S11	Effective and efficient collaborative decision making	Thales
S12	e-Commerce information sharing	TIE
S13	Mobile office for an owner of a micro-enterprise	TIE
S14	Safety at work in the construction sector	TIE
S15	e-Government online application submission service	Consorzio Interuniversitario Nazionale per l'Informatica
S16	e-Government online fee visualization and payment Service	
S17	Assisted Industrial Maintenance	Siemens

The parts of the template to collect scenarios are the following:

- **Short name:** a short name for the scenario
- **Detailed, step-by-step scenario description:** a textual description of the scenario. Additionally, the domain, the sub domain or the main objective as well as contextual information about the system environment should have been specified.
It has strongly been suggested to add UML diagrams such as a use case diagram or a sequence diagram to clarify textual descriptions.
- **Rationale:** describes the reasoning and justification for the scenario - that is some important background for why the scenario is what it is. This will be important to help those less familiar with the domains to work with the scenarios.
- **Service consumer:** for each user and location (the location indicates where the consumer stays, e.g., customer-home; customer-office; technician-office, etc) the following has to be specified:
 - Primary requests, problems to solve or needs
 - Required performances or needs
 - How and when the user prefers to obtain the service etc. (subscription, pay for use.) and if it is useful or necessary
- **Service provider:** a possible description of who provides the requested service.
- **Service integrator/developer:** a possible description of who integrates or develops the requested service.
- **Problems and challenges:** the specific problems that each scenario addresses or that consumers and providers face.
- **Architecture and constraints:** all involved devices (PC, PDA, etc.), hardware, software, and possible integration with existing applications, how communication is accomplished (GPS, GPRS, Bluetooth, infrared, etc.)
- **Source:** describe how this scenario was produced, from whom it was elicited and possibly it's role within the organization.
- **Motivation:** describe why this scenario has been selected.

The following scenarios will be described below:

- **Service procurement (S1):** describes a service procurement scenario, in which a potential consumer can query a search engine for services corresponding to his needs.
- **Service lifecycle support (S2):** describes a facility that supports the entire life cycle of a service. This starts from defining a service concept and then goes right through to ultimate withdrawal and decommissioning.
- **Management services for grid and service platforms (S3):** describes services to manage and maintain grid and service networks. As a concrete example, the Large Hadron Collider computing Grid in CERN is discussed.

- **PhiMas: personal health information monitor and alert service (S4):** describes the services involved with PhiMas, a personal monitor and alert system for elderly people; people who need personal care while recovering from injuries or people that are disabled.
- **Collaborative e-learning scenario (S5):** describes the services used within a collaborative learning approach based on the definition and execution of a learning experience.
- **Deployment and configuration of a generic platform (S6):** describes the services used for the deployment of a platform.
- **e-Health: complex diagnostic workflow (S7):** describes services useful in the workflow of determining the patient's complete health status.
- **e-Health: assisted living (S8):** describes services used in assisted living systems, systems that support persons with a chronic illness or those with a need for constant medical surveillance. This scenario refers to devices such as, for instance, blood pressure units or blood glucose meters that are operated either by the patient, by a nurse, by a paramedic or by another member of an ambulance crew.
- **Traffic management: large scale emergency handling (S9):** this scenario describes services used in the traffic management domain to handle emergency cases, like the direction of rescue forces to the location of the accident as well as the deviation of traffic through places not intended for heavy traffic.
- **Crisis management system of systems (S10):** describes services used for the necessary interactions between different organizations in order to mitigate the crisis and reduce the impact of a crisis situation occurring at for example an airport.
- **Effective and efficient collaborative decision making (S11):** describes services used in collaborative decision making applications, i.e. the sharing of business operation (e.g. flight plan operation) information between multiple and different systems, and implementation of new collaborative components and services to improve the efficiency of the operations based on a "system-wide" approach.
- **e-Commerce information sharing (S12):** describes the services used in the commercialization of Software-As-A-Service (SAAS) solutions.
- **Mobile office for an owner of a micro-enterprise (S13):** describes the services used in a business involving the placement and maintenance of vending machines in different locations, all supported by a national or international franchise network.
- **Safety at work in the construction sector (S14):** describes the services used for a wireless enabled device moving around a work site in order to verify that all the safety requirements are fulfilled.
- **e-Government online application submission service (S15):** describes the services used in an e-Government service portal to submit applications and receive replies online.
- **e-Government online fee visualization and payment Service (S16):** describes services used in an e-Government portal to review and to pay taxes and fees online.

- **Assisted Industrial Maintenance (S17):** describes the scenario of improving the speed and quality of repairs of technical repair staff by giving access (via an ordinary display rendering the 3D virtual world and an avatar or via a dedicated virtual reality display) to expert knowledge, audio and visual information, as well as 3D repair instructions.

The scenarios cover the following domains:

- **Networked IT service provision:** Service procurement (S1), Service lifecycle support (S2)
- **Grid- and service networks:** Management services for grid and service platforms (S3)
- **Health care:** PhiMas: personal health information monitor and alert service (S4), e-Health: complex diagnostic workflow (S7), e-Health: assisted living (S8)
- **e-Learning:** Collaborative e-learning scenario (S5), Deployment and configuration of a generic platform (S6)
- **Crisis management:** Traffic management: large scale emergency handling (S9), Crisis management system of systems (S10), Effective and efficient collaborative decision making (S11)
- **e-Commerce:** e-Commerce information sharing (S12)
- **Ubiquitous computing:** Mobile office for an owner of a micro-enterprise (S13)
- **Construction sector:** Safety at work in the construction sector (S14)
- **e-Government:** e-Government online application submission service (S15), e-Government online fee visualization and payment Service (S16)
- **Assisted Maintenance:** Assisted Industrial Maintenance (S17)

The following table gives an overview of coverage of the scenarios considering the European industrial sectors [8], the public administration, and the education sector:

Table 9: Scenario coverage in respect to industrial sectors, the public administration, and the education sector

Sector/Scenario	<p>Service procurement (S1)</p> <p>Service life cycle support (S2)</p> <p>Management services for grid and service platforms (S3)</p> <p>PhiMas: personal health information monitor and alert service (S4)</p> <p>Collaborative e-learning scenario (S5)</p> <p>e-Health: complex diagnostic workflow (S7)</p> <p>Deployment and configuration of a generic platform (S8)</p> <p>e-Health: assisted living (S8)</p> <p>Traffic management: large scale emergency handling (S9)</p> <p>Crisis management system of systems (S10)</p> <p>Effective and efficient collaborative decision making (S11)</p> <p>e-Commerce information sharing (S12)</p> <p>Mobile office for an owner of a micro-enterprise (S13)</p> <p>Safety at work in the construction sector (S14)</p> <p>e-Government online application submission service (S15)</p> <p>e-Government online fee visualization and payment service (S16)</p> <p>Assisted Industrial Maintenance (S17)</p>
Aerospace	
Automotive Industry	● ● ●
Biotechnology	
Cement and Lime Industry	
Ceramics	
Chemicals, Rubber and Plastics	
Construction	●
Cosmetics	
Defence Industries	
Electrical Engineering	●
Food Industries	
Footwear	

Furniture	
Glass	
ICT Industries	● ●
Leather	
Mechanical Engineering	
Medical Devices	● ● ●
Non Energy Extractive Industry	
Non Ferrous Metals	
Pharmaceuticals	
Printing and Publishing	
Pulp, Paper and Paper Products	
Ships and Boats, Building and Repairing of Ships and Boats	● ● ●
Steel	
Textiles	
Wood Products	
Public administration	● ●
Education	● ●

The following table summarizes the granularity level (high, medium or low) as well as the expected time frame when this scenario will become relevant, i.e., “future”, “now” (cutting edge technology), or “stable” (technology is already in place), of each scenario:

Table 10: Detail levels of scenarios

Scenario	Detail level	Timeframe
Service procurement (S1)	High	Now
Service life cycle support (S2)	High	Now
Management services for grid and service platforms (S3)	Medium	Now
PhiMas: personal health information monitor and alert service (S4)	Medium	Now
Collaborative e-learning scenario (S5)	High	Now
Deployment and configuration of a generic platform (S6)	High	Now

e-Health: complex diagnostic workflow (S7)	Medium	Now
e-Health: assisted living (S8)	Medium	Now
Traffic management: large scale emergency handling (S9)	Medium	Future
Crisis management system of systems (S10)	Medium	Future
Effective and efficient collaborative decision making (S11)	Medium	Future
e-Commerce information sharing (S12)	High	Stable
Mobile office for an owner of a micro-enterprise (S13)	High	Stable
Safety at work in the construction sector (S14)	Medium	Stable
e-Government online application submission service (S15)	Medium	Stable
e-Government online fee visualization and payment service (S16)	Medium	Stable
Assisted Industrial Maintenance (S17)	Medium	Now

3.1 Service procurement (S1)

In a future service ecosystem there will exist providers and consumers for services at all level, ranging from complete business processes at the top through to simple hardware resources at the bottom. A consumer will identify services within the structure of their business processes, that after careful consideration of the transaction costs and other factors such a security, reliability etc, they decide to purchase from an external supplier. The economist Ronald Coase initially proposed the concept of transaction costs in 1937 in his seminal paper “The nature of the Firm”, they are widely used to assess when an external service is more cost effective for a business than developing the same functionality internally.

Having decided on the need to purchase a service externally, the potential consumer has to formulate a description of the service they require. This description needs to include both functional and non-functional requirements that the customer has identified for the service and be compatible with a search technology.

The requirements are input into a search engine that may include some semantic intelligence to identify equivalent services, a list of suitable suppliers and then their service offerings are returned to the consumer.

The consumer is then able to choose a suitable service and supplier based on both the offered functional properties of the service and the context for the service as described by the non-functional properties that would include details such as price, confidence in supplier etc.

Having chosen a supplier, the consumer then requests details of the available quality levels that the service provider guarantees for the service

and will complete some form of SLA template with the quality levels they think most closely match their requirements.

The completed SLA template is passed to the supplier to obtain agreement for the use of the service and after confirming the start of the agreement the service provider will send the consumer a service end point reference so that the consumer can begin to use the service.

During consumption of the service both service consumer and service provider will collect and analyse performance information. This is to ensure that the service both meets the customers' requirements, and that the quality guarantees embodied in the SLA are being maintained.

Either during use of the service, or at the end of the period of use, the consumer may have to make payment for their use of the service. The size of the payment and payment mechanism will have been defined in the SLA agreed between consumer and provider before use of the service began.

The **rationale** for this scenario is the following: in a modern, flexible, service oriented world companies are free to focus on their core business and increasingly are using specialist providers to provide key parts of their business processes. The advantages of such an approach includes flexibility, reduced costs (particularly in pay per use environments) and the ability to switch between suppliers as the need arises, to benefit from either better pricing or better service.

To support a market of this type there is a requirement for some consistency in service description to support discovery, comparison and potential switching between suppliers.

This scenario **applies** to

- Any consumer of any service
- Any provider of services: the service provider needs to advertise the service they are offering, define the service levels available and provide SLA templates to potential purchasers of the service
- Any service integrator/developer: the service and any metrics used to assess its performance need to be described in a consistent and meaningful fashion

There are a number of **challenges** that need to be addressed in order to fully realize the scenario described:

- The scenario relies upon common understandings between the service consumer and service provider of details of the services functional and non-functional characteristics. This may require the development of standardized vocabularies for such descriptions.
- The process of describing and then searching for a service may require the assistance of semantically enriched search and discovery tools so that all appropriate candidate services are correctly identified.
- When the service consumer requests SLA templates they must have an understanding of the terms embodied in the SLA, and relationships

between them, leading to a requirement for standardized SLA representations.

- Should the SLA be ambiguous or not exactly what the service consumer requires there may be a requirement for a negotiation protocol for the agreement of SLAs. This is particularly complex because the service provider must have an adequate management structure in place to accommodate any changes in the base SLA.

Both parties in the agreement should be able to access, if required, meaningful and consistent performance data for the service, to assist in management of the SLA.

There are a number of **architectural constraints** highlighted by the described scenario:

- A uniform representation is required for services of different types. These services will be offered depending on customer demand and not on any arbitrary definition of functional layers in the architecture. Many services will span a number of traditional technology layers.
- There is a requirement for a decentralized architecture, but that will support federation, for directories, SLA repositories etc. each of which may have different policies governing access and use, to support public and private service offerings.
- Service discovery mechanisms need to address both functional and non-functional characteristics of offered services. They also need to accommodate selective control of access and visibility of services.

There needs to be standard representations of SLAs, and standard protocols for their negotiation and monitoring.

This scenario was produced in BT Innovate based on experience in the area, and from a large number of discussions over years with potential stakeholders in a future service marketplace. The scenario draws on information from both within the organization and many external contacts.

The top three reasons for selecting this scenario are:

- This is one of the fundamental real world scenarios for a future service based infrastructure where services are bought and sold. Successful demonstration of this scenario will be a key milestone in convincing potential suppliers and consumers of services of the utility of the approach.
- Although this scenario is closely related to the real world common experience of “shopping” it requires a number of technology developments in order to facilitate high speed, low overhead, automated approaches to service selection in a heterogeneous marketplace. Demonstration of the ability to describe, discover and use services with guaranteed levels of performance across different application domains will demonstrate the advantage of SOA to stakeholders. Cost effectiveness of the approach will obviously be a key concern.
- The generality of the scenario and its focus on a very basic function in a service oriented marketplace will make it applicable to the widest range of

potential users both service providers and service consumers. Successful demonstration of the scenario will highlight the applicability of the NEXOF architecture to the widest possible constituency.

Although some parts of this scenario have been demonstrated, some key areas of the technology require appropriate standards to be defined and other areas require development into more generalized realizations.

The **inputs** for this scenario will be services with service descriptions, SLAs and management facilities. The **outputs** will be services executing for end users at agreed prices and with guaranteed levels of performance.

The **market** of this scenario is the entire market for SOA based systems, independent of type of service, industry sector, or customer segmentation.

This scenario is **relevant for all stakeholders** in a SOA based system, but might be considered especially relevant for service providers and consumers.

3.2 Service lifecycle support (S2)

NEXOF aims to ensure low barriers to entry so that service innovation can come from a very wide range of parties, companies, communities or even individuals. As a consequence, no assumptions are made about the nature of the service or service provider. Some of the steps described in this scenario may not be explicit in all cases but it is believed it represents a generally applicable pattern.

This scenario begins when a potential service provider identifies an opportunity for a new service. This is the result of activities that may include observing some unmet customer requirements, carrying out market research, determining that providing the service is within the capabilities of the provider (given available resources, including services already accessible that could be included) and that service provision is likely to be economically sustainable for its anticipated lifetime.

A service developer (or integrator) then designs the system required to deliver the service, according to the needs of the service provider. This includes identifying existing components and designing any additional software required. It is essential at this stage that the system is functionally correct (i.e. it does what the provider intends it to do) and also that facilities for monitoring and managing a running service are incorporated. (Typically this will require support for the usual FCAPS aspects – Fault, Configuration, Accounting, Performance and Security). This design may be, at least in-part, abstract, particularly where external third-party provided services are concerned. It is anticipated that these will be specified in terms of required properties, which are matched to real services at deployment time rather than having specific service instances built into the new service.

With a system design in place and novel components developed, the provider then determines resource requirements to support anticipated demand patterns. This also needs to include strategies for deployment and

scaling of the service. Resources owned and controlled by the provider are out of scope for NEXOF, so the main concern here is to define dependencies on external functionality, including infrastructure resources (computing, storage, network etc) as well as application-level functionality.

The next step is for the provider to consider in detail how to expose the functionality of the service to his customers. This involves developing templates for Service Level Agreements (SLAs) and service descriptions. The SLA templates must clearly define the parameters by which service behaviour is measured and the provider needs to understand feasible limits on these parameters. This defines what levels of service can be offered.

In parallel with addressing service exposure, and still in the phase before launching the service, the provider defines operational management policies. These should use the management and monitoring features of the service implementation to define the desired behaviour in response to any anticipated service conditions. As far as possible these policies should be amenable to automation.

At this point, the service is ready to be offered to customers. It has a clear description; relevant functional and non-functional properties have been defined. Policies for operational management are in place and deployment procedures are established.

Customers must be made aware of the service, in terms of what it can do for them, how it can be used and what to expect for a given price. The provider achieves this by advertising the service in appropriate locations. This may include the use of public or restricted directories. From an architectural point of view, information models and protocols should be independent of any policy decisions made by the provider with regard to service visibility or accessibility.

On discovering a new service, interested customers obtain SLA templates and make appropriate selections within the parametric boundaries defined by the provider. The resulting bid (completed template) is sent to the provider who then assesses the ability to deliver against the proposed SLA, taking into account the resources available, all existing commitments for the same service (or others contending for the same resources) and any other business policy decisions. Following this assessment, the provider decides whether or not to accept the bid. If accepted, the SLA is concluded and enters into force. The provider communicates access information (e.g. endpoint details) to the customer and then waits to receive service requests from the customer.

Either in response to or in anticipation of service requests from the new customer, the provider may provision the service. This may involve, for example, new or increased resource deployment (or alternatively no change).

There is then a phase in which the service runs to support the demands of its customers. During this phase, the provider monitors the behaviour of the service, takes management actions to maintain it at levels aligned to

whatever set of SLAs is current, and carries out appropriate accounting, billing and customer support actions.

While the service is running, the provider may assess usage patterns, performance levels achieved in practice and maybe improvements (in quality or cost) that could be obtained by replacing some third-party components with others. As a result, SLA templates may be modified for future customers, typically offering more control or higher levels of assurance to the customer.

When the service becomes obsolete or no longer in the interests of the provider to offer, there will be a need to withdraw and decommission it. It is straightforward to stop advertising a service and publishing SLA templates to new customers. However, existing customers and users must be dealt with.

In an open service environment, a particular service may be involved in complex relationships with others so the impact of withdrawing it may not be obvious. This needs to be recognised in the architecture. In simple cases the provider may offer a migration path to affected customers. In general the provider does not know in detail how the service is used so it may be necessary to announce its withdrawal and leave the service consumers to make their own decisions. There must be clear recognition by service consumers that services cannot necessarily be relied on in the long term. This should influence the design of NEXOF applications and services.

The **rationale** for this scenario is the following: the concerns of a service provider are not solely associated with running an existing service. A full service lifecycle view must be taken, and supported by the NESSI Open Framework. This starts from defining a service concept and then goes right through to ultimate withdrawal and decommissioning. As services in the ecosystem envisaged by NESSI may have complex relationships with each other, which are not under the sole control of a single party, it is important that the Reference Architecture ensures that services are manageable throughout their whole life.

This scenario **applies** to

- Any service consumer: Many service consumers will not be satisfied with best-effort services. Consumers need to have an awareness of the existence of services and providers through advertising.
Needs a clear understanding of the characteristics of the services he uses including full lifecycle aspects, not just properties of the running phase.
- Any service provider: It is assumed that a provider may provide a range of different services to many different customers.
Needs to be able to introduce new services into a complex service ecosystem in a systematic way.
Needs to integrate with generic support systems and services (e.g. management, accounting/billing, CRM)
Needs to make sure SLAs are based on realistic capabilities and customer needs (and price accordingly).
Needs to have automated management for operational efficiency. Not all requests for service will be accepted.

Needs an understanding of the service ecosystem and all relevant relationships to facilitate deployment and decommissioning.

- Any service integrator/developer. (Service integrator/developer role may be combined with the service provider role in practice). Every service must meet functional requirements and provide adequate functionality for monitoring and management. In the environment of the NESSI Open Framework, dependencies should be expressed in abstract form so that specific supporting components can be mapped in at deployment time (and replaced later as necessary).

The **problems or challenges** of this scenario are:

- Understanding in detail the requirements on a service to make it manageable
- Representation of low-level resource requirements in service terms (Infrastructure as a Service)
- Uniform deployment strategies applicable to a shared infrastructure. Different approaches to scaling will be appropriate for different services. It is important that those required to make management decisions know what to do
- Management of service (including infrastructure resource) dependencies

The **architectural constraints** of this scenario include:

- Consistent and uniform approach to monitoring and control
- Service description including functional and non-functional characteristics
- Commonly understood (between provider and consumers of a service) representation and semantics for SLAs
- Flexible and dynamic deployment of services into the service ecosystem
- Ability to decommission and remove services with predictable consequences for all actors

This scenario was produced in BT Innovate based on experience in the area and from a large number of discussions over years with potential stakeholders in a future service marketplace. The scenario draws on information from both within the organisation and many external contacts.

The **motivation** for this scenario is that this scenario is representative of the activities of service providers. The portfolio of services offered changes over time with new services being introduced and obsolete services being withdrawn.

Standard processes for managing all stages of the service lifecycle become increasingly important as the market becomes more dynamic and customer expectations for novel or customized services increase. Service management and control are essential issues for a service provider and these must not be confined to the operational management of existing services.

The scenario identifies the major concerns and stakeholders at different stages in the service lifecycle. As well as providing a framework for understanding the business processes of a single service provider, it

illustrates some of the relationships between service providers and other stakeholders in a complex service ecosystem.

Aspects of this scenario are already in place – it is based on a generalization of current practices. As an open market for networked services develops, service lifecycle management is expected to become increasingly reproducible and eventually significant parts of it will be automated.

The **inputs** for this scenario are ideas for new services or software components to be used in the provision of services, and descriptions of the infrastructure and management systems available to support them.

The **output** of the scenario is a reproducible and controlled process for the introduction, operation and decommissioning of a service.

The scenario has general scope in the area of networked IT services, particularly where elements of useful functionality are exposed by a service provider with the intention of being incorporated into business processes, whether by the service provider, the customer or a third party.

Several **stakeholders** in a service-oriented architecture are expected to find this scenario relevant.

- Consumers of services that are used repeatedly (e.g. within a business process), such that persistence and dependability are important factors in service selection
- Service providers with a dynamic portfolio of services which need to be managed in a consistent way, or the need to offer customization for a range of customers
- Service integrators or developers producing software that is brought to market as services using delivery platforms operated by an independent provider

3.3 Management services for grid and service platforms (S3)

The emergence of Service networks in scientific as well as industrial areas go hand in hand with IT trends such as SAAS, Web services, SOA and Grid-computing. (See also the NESSI Strategic Roadmap documentation.) In such contexts, computer systems, hardware as well as software, are part of large IT networks, enabling and providing the business of the (NESSI) service networks.

The management and maintenance of such networks themselves can be regarded by means of services. This scenario is about such services.

In this scenario we describe the situation of maintaining a grid platform, i.e. the business and technical components that are needed to keep a particular grid network up and running.

As the components of the operational service network are owned by multiple organizations, the management and maintenance of it involves more than

just technology. Organizational aspects, such as trust and collaboration have to be taken into account as well.

A concrete example that one can think of in this scenario is managing a large Grid computing environment such as the LHC Grid.

The LHC Computing Grid (LCG) is the grid environment that supports the particle physics experiments using the Large Hadron Collider (LHC) at CERN. The LHC, currently being built at CERN near Geneva, is the largest scientific instrument on the planet. The data is captured and analyzed by more than 140 participating institutes in 35 countries. The mission of the LCG is to build and maintain data storage and analysis infrastructure for the entire high energy physics community that will use the LHC. One of the great endeavours is to manage a huge data stream of 600 megabytes per second. LCG depends on two major grid infrastructures: Enabling Grids for e-science (EGEE) and US Open Science Grid (USG).

Roughly, we consider three kinds of services:

- **Adaptive Information Services:** These services monitor the resources of the service grid. They cooperate with existing (local) monitoring applications, and provide an overview of the actual status of the grid. They are adaptive to their dynamic environment, which means that in their cooperation they are flexible in terms of their interface and representation. Also, they have to take into account the level of detail of the information, a common constraint when dealing with cross-domain information, often based on trust or privacy.
- **Collaboration Support Services:** this kind of service supports the organizational aspects and collaboration of people that are involved, while belonging to different organizations. Examples are administrative support services, or services that align the business processes of the involved organizations. Examples are the Virtual Organization management services developed in the European FP6 project 'Ecolead'.
- **Workflow and Maintenance support Services:** in case certain parts of the grid are 'under construction' these services support the handling of alternatives, such as re-routing and backup. They collaborate with existing workflow, schedule and routing services.

The rationale for this scenario is the following: The trends of Service Oriented Architectures (SOA) and grid computing (parallelised programs running on a whole network of computers) met in the beginning of the 21st century. A SOA supported grid is a scalable, robust, flexible computing environment for extremely high intensive computational tasks. SOA-grid-solutions are envisaged to be auto-emergent, and this implies proper information and system management.

Different kinds of organizations, different kind of business services and many technical systems are used to monitor and manage such a large, complex infrastructure.

Various grid infrastructure and application information services provide information about certain aspects of the status and working of the grid.

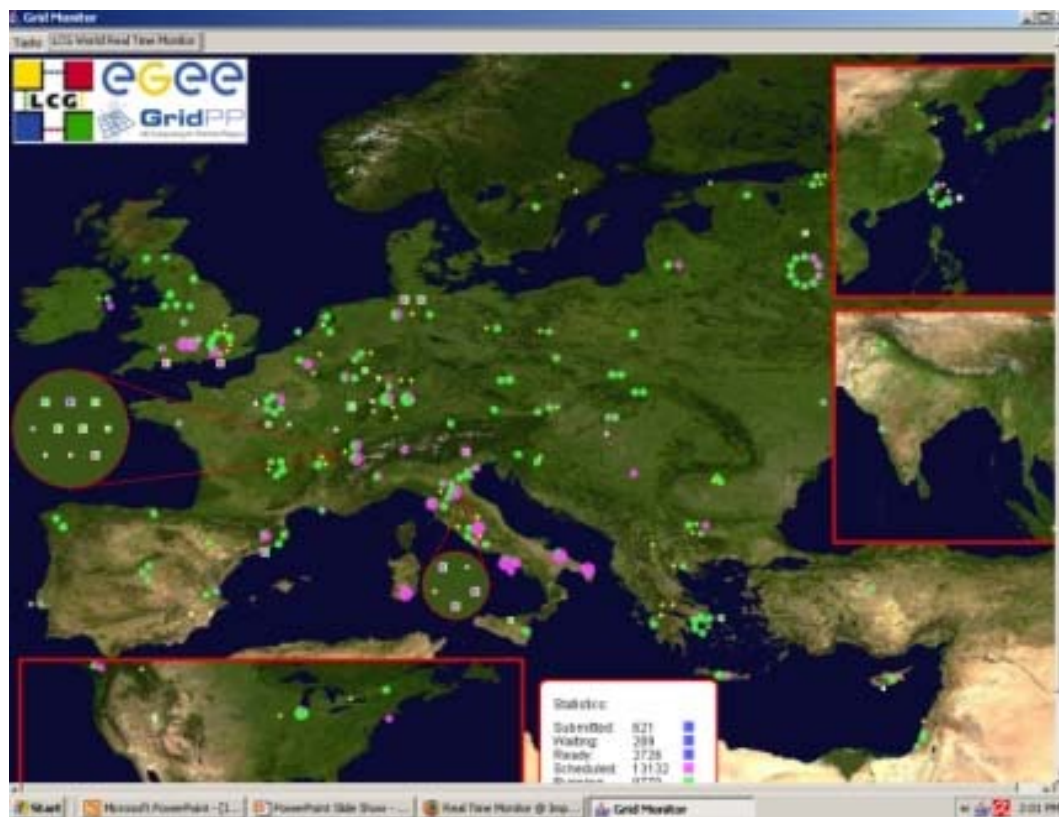


Figure 2: Participating institutes for data capturing and analysis of the LHC Computing Grid

Collaboration, in organizational terms, as well as in technical terms is a trend that changes our society dramatically over the next 10 years. Services share environments on temporal bases, or require large computing facilities on a grid-computing based approach. Due to the growth in terms of size and interaction, the involved platform will become too complex to manage by hand. Expectations are the assist of intelligent software (in the form of services) might be needed in order to control and manage these platforms.

This scenario **applies to**:

- **Service consumers**, that are system administrators of the organizations that own the components of the service networks, since the services in this scenario are intended to support the management and maintenance of the service networks. The services provide support for information provisioning, daily operation but also organizational and collaboration support to let the various kinds of maintenance being carried out by different organizations. Indirectly, the goal of these services is to optimize the seamless use of the service network by its end-users. One might think here in terms of capacity management, backup mechanisms, and self-healing and operational support.

- **Service providers**, with multiple kinds/roles services for the provisioning management services. The providers of these services can either be the organization(s) that own(s) the service platform, or it can be provided by the organization that delivers one or more components to the platform. The choice depends on the level of standardizations and future maturity of the technology of adaptive systems and semantics. Provided and used by system administrators of internal organizations (those that own operational components of the service network) or external organizations (those that do not have)
- The **open market as service integrator/developer**; the management services might be developed by any kind of organization.

The **problems and challenges** lie in the expected isolation of these management services. Many providers will intend to build what they call 'solutions', and realize services that would only solve parts of the maintenance issues. The challenge is to focus on the alignment and sharing of information, in other words the collaboration between these services.

Challenges are the issues known within the discipline of the 'semantic web' and 'multi agent systems' in general.

Apart from the technical challenges, the real challenge is to realize the right degree of collaboration of the organizations that play a role in the management of service networks.

The **architectural constraints** of this scenario include:

- **Interoperability**: In order to work with each other and with the existing services, the management services must fulfil certain basic interoperability standards
- **Heterogeneity**: The management services work on a heterogeneous platform, and might be heterogeneous themselves
- **Distributed**: The service network is a distributed platform, consisting of many domains over which information must be shared
- **Dynamic**: Often, the service network is dynamic; services come and go. The management services must be able to deal with this
- **Identity management**: Trust, security and identity have to be incorporated firmly, since it will be very likely that multiply originations will be involved. Each organization has its own information domain, and forces the system to deal with constraints about privacy

3.4 PhiMas: personal health information monitor and alert service (S4)

Elderly or disabled persons who hampered in their mobility can easily get isolated from their social environment.

PhiMas is a personal monitor and alert system for elderly people; people who need personal care while recovering from injuries or people that are disabled.

The service assist them going out of their houses, by taking away the fear that something unwanted might happen. Via a special device they can call for help or information about their status and/or situation they run into.

Information integration is one of the key aspects of PhiMas. At the moment of a call, the location and medical status is known. Depending of the situation, a specific kind of help might be needed. Redirecting requests automatically to the right place (hospital, standby teams, personal buddies) PhiMas takes care of a transparent and easy use.

PhiMas is a service that is provided by multiple, cooperation organizations. It is important that the business, as well as the underlying technology is well aligned. In PhiMas, the concepts of network organizations and service networks come together.

Using PhiMas people should feel safe enough to get of their houses.

For the user, PhiMas is used via a special, easy button device. The device is equipped with GPS and pre-programmed to dial a special telephone number.



Figure 3: PhiMas Interface

Via a web interface, PhiMas allows updates of medical information. The end users as well as their doctors can do this.

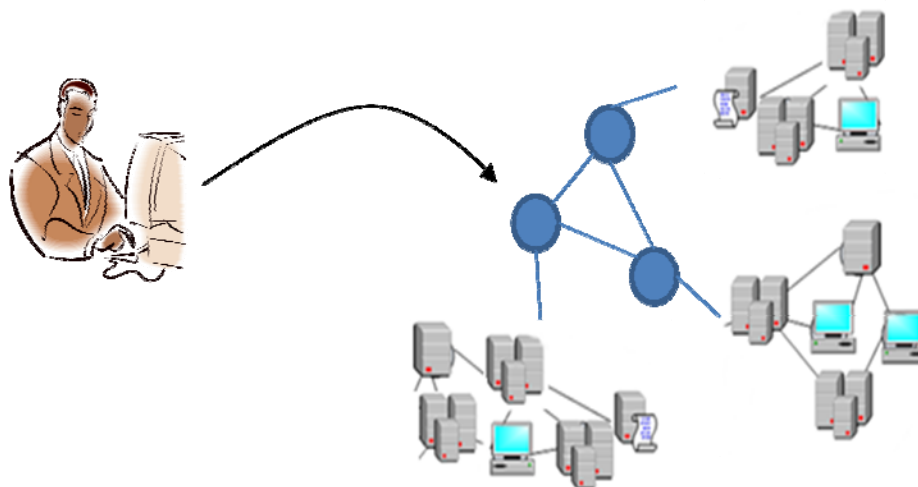


Figure 4: PhiMas access via web interface

The following services are provided:

- **Registration and enrolment services:** allow for registration of a new user, and possibly manages the billing process, e.g. for communication and messaging functionalities
- **Identity and personal security Service:** in the architecture, of course, the device should be acting on behalf of a particular user only
- **Information disclosure Service:** take care of maintaining the medical and healthcare status of the user /patient, such that whenever an alert call is made, the necessary information is available
- **Management and availability service:** the federation that allows for this backup and the technical service platform should be available 7x24. A proper SLA and backup management is crucial for the working of PhiMas

The **rationale** of this scenario is the following: the integration of ICT information systems and personal care is a contemporary trend in our society. Social isolation is an already a known result of the individualization of our society. On the other hand we see that our society is more than ever connected, but in a different way, more or less via all kinds of communication virtual networks.

Also in the context of healthcare we see such emerging network services. An example is the idea of PhiMas. Stimulating the development and use of such service allows new insights in the world of personal care in near future societies.

This scenario **applies to:**

- **Service consumers**, which are Elderly people, disabled people, people that recover from hospitals or doctors, physicians and people providing health care and personal care
- **Service providers**, which are healthcare organizations, telecom providers, insurance companies, institutes that offer 'buddies' and personal care

- The **open market as service integrator/developer**; the management services might be developed by any kind of organization

Apart from the technical challenges, the real challenge is to realize the right degree of collaboration of the organizations that play a role in the management of service networks.

The **architectural constraints** of this scenario include:

- Interoperability: In order work with each other and with the existing services, such as electronic healthcare record services
- The technical service platform must be available 7x24 and have proper backup mechanisms
- Identity management: Thrust, security and identity have to be incorporated firmly, since it will be very likely that multiply originations will be involved. Each organization has its own information domain, and forces the system to deal with constraints about privacy

3.5 Collaborative e-learning scenario (S5)

This scenario is based on the definition and execution of a learning experience constructed upon a typical collaborative learning approach. More specifically, the learning experience takes place inside a virtual classroom of students.

Usually, the collaborative learning experience, at a high level of abstraction, can be seen as a workflow made up of learning activities; each learning activity include one or more services, i.e. collaborative services, learning unit delivery service, assessment service and so on. All these services can be defined “environment-learning services” because they compose the environment, which the learner enjoys during his learning experience. So the learner will enjoy the flow of learning activities interacting with other classmates etc in order to reach the target concepts.

A learning unit (LU) can be seen as a sequence of concepts that can be expressed using knowledge representation techniques such as ontologies. In case, more than one LU can be inserted into a learning path, each learning object (LO) is stored in the repository and is linked to one or more concepts of a domain ontology. A concept can be also linked to more than one LO. Typically, that occurs when a concept is explained according to the different resource formats (i.e. text, audio, video) or different learning resource types (i.e. narrative, simulation) in each LO. The sequence of LOs linked to a specific learning path is a “bounded learning path” (BLP).

This scenario has two main sections: the definition and the execution of the learning material.

Definition phase

The teacher is involved in the definition of a learning material. This task could be split in two sets of sub-activities: one related to learning path, the other related to environment in which the learner will enjoy their learning experience:

- Definition of learning path
 - The teacher defines or loads one or more ontologies related to the domain (i.e. an educational ontology, or e-learning ontology, is used in order to model specific disciplinary domains)
 - Once the teacher establishes the target learning concepts, a dedicated service is able to design the best possible learning path (i.e., a sequence of concepts). In case of personalization enabled service it will design a learning path taking into account a learner's knowledge, skills, prior performance, learning style and background. Alternatively, the teacher may define a static handmade learning path
 - The teacher provides new LOs. They are added to others already stored in the repository. For each LO the teacher defines its metadata; among these information there could be the ontology concept (or concepts) described by the LO; this, sometime boring, task could be driven by well tested and agile templates
 - The teacher arranges a specific typology of LO: assessment LOs (i.e. tests) needful to evaluate the initial and/or final learner's knowledge
 - Previous steps are preliminary to the automated composition of learning path definition (personalized or not)

- Definition of environment
 - A teacher organizes the classroom based on the personal characteristics of the students and recommendations from the collaborative learning theories and practices
 - A teacher defines types of desirable activities and didactical approach. After that, a dedicated service is activated in order to identify a subset of environmental services to be proposed to the teacher. After the choice of the most appropriate services is made they become available for the students. This approach allows dynamic adaptation of services (which services and which settings for each service) to the context. For instance, during the test fulfilling the learner could be able to use annotation (but just for writing new notes, not for reading the old ones), but not the chat service; instead during the studying phase the learner will be authorized to use the Instant Messaging, blog and forum services.

For instance, a teacher can arrange an initial set up of the learning experience. In such a case the environment may include the following services: delivery service of LUs (just for LOs related to overview), Instant Messaging and Video Conferencing in order to provide synchronous communication between the teacher and a student.

It is important to emphasize that the enjoyability of a composed learning path is heavily relied on environment, in which several services join up: instant messaging, forum, collaborative writing as well as the one that expose the contents. Thus, a composition of learning services should be

based on a very subjective parameters like usability, maturity level, fit for the target audience, etc.

Execution phase

The teacher and the students are involved in the experience with the composed learning path:

- Execution on the student side
 - Each student takes part in the presentation phase held by the teacher;
 - The student should execute a preliminary step: the “learning-style assessment” to identify the most suitable “learning resource type” (this assessment could be made up of a profiling questionnaire, structured taking into account the most representative learning style models in literature and also used in the different learning experiences)
 - The student should execute another preliminary step: the “knowledge assessment” to check the learner’s “cognitive status”
 - The student can start a standard learning experience (handmade by a teacher or automatically generated); in this case only learning-style assessment should be considered, in order to choose the most appropriate LOs
 - The student can start a personalized learning experience; in this case both the learning-style (in order to choose the most appropriate LOs) and knowledge assessments (in order to identify at first the concepts the learner doesn’t master yet) should be considered
 - During their learning experience, each student has a contextualized (not personalized) set of environment services, according to teacher’s preferences as described above
 - Depending on the availability of environmental services, each learner is able to communicate and collaborate with others (learners and teacher)
 - At the end of the learning path, each learner should execute another “knowledge assessment” to check the new “cognitive status”; depending on results of this check, the learner could start a new personalized learning path (so called remedial work) in order to fulfil the possible gaps. This process may be iterative.

- Execution on the teacher side
 - The teacher shows an overview of the learning experience by means of synchronous communication services, i.e. videoconference
 - Depending on the available environmental services, a teacher is able to communicate and collaborate with learners

During a learning course, a teacher could combine students in workgroups, within one class or across different classes. At this phase, teacher can assign specific tasks and can allow a specific subset of available environmental

services for each group. This feature is useful for evaluating the effectiveness of the overall learning process.

The **rationale** for this scenario is the following: usually, e-learning scenarios involve learning activities common for all students.

This scenario emphasizes the personalization and collaborative aspects of learning. This is addressed by the needs of:

- Communication and collaborative environment services, in order to make more the learning experience itself more effective and in order to suggest to the teacher improvements to learning strategies and objects
- Automated contextualization of environment in each phase of learning experience according a chosen didactical approach
- Flexible personalization of learning experience according to learner's "cognitive status" and "learning style".

This scenario gives chance to face **innovative architectural aspects**. The most significant are the following:

- Attempt to join the personalization and collaborative aspects, by mean of social and semantic strategies: i.e. the convergence of the Social Web and Semantic Web
- Environment services integration and coordination based on mashup server. The choice of "semantic mash-up" (ontology based) allows the management of several aspects:
 - Work environment contextualization depending on specific phase
 - Workflow management of activities (in this scenario learning activities)
 - Interaction with ontology server
 - Services metadata making
- Harmonization of several heterogeneous information sources by means of semantic technologies; in this scenario, as well as in an enterprise context, the opportunity of retrieving or organizing information regardless of its own structure or format, but focusing on its semantics, is of great importance (i.e. think to the search phase across resources represented by txt files as well as movies or images).
- Dynamic personalization of a learning experience; this feature could be useful in domains other than e-learning, thanks to the ability to adapt a path or a strategy according validated models (represented by ontologies).
- Capability to automatically identify the most suitable service to contact a called user, irrespective of the calling service.

Environment services of this scenario are focused mainly on communication and collaboration, typical features of web 2.0, involved to make a more and more effective learning experience. Collaboration and communication can be provided among heterogeneous services and sources (structured or not),

allowing extraction and classification of information for coming needs (i.e. new collaboration phase).

The devices to be involved in this scenario could be identified according to specific activity. For instance:

- During the authoring phase (teacher) the device should allow a comfortable interaction; so a Web Browser is the most suitable device
- During activities such as presentation reading or audio listening, a PDA could be enough, on the condition that communication is fast enough (i.e. Bluetooth or GSM are not recommended)

During activities such as video conferencing, usually specific plug-ins or add-ons are required; so, their availability on operating systems may restrict usable devices.

The consortium partner MoMA has provided this scenario. MoMA is involved in the production of innovative service oriented learning and collaborative platforms, exploiting the achievements of the most recent research conducted on learning technologies and pedagogical solutions.

The “Collaborative e-learning scenario” has been derived in conjunction with both research and technical teams. The scenario takes into account the analyses of the requirements related to the training and collaboration needs of several customers: large companies, SME, universities and schools. On the other side, the scenario proposes some ideas derived from pedagogical studies related to the personalization and to the collaboration during the learning experience.

This **scenario been selected because:**

- This scenario takes place in the learning context, but involves personalization and collaborative aspects, common in several contexts. These aspects don't simply refer to front-end level, they rather refer to all work environments aiming to allow the user to accommodate his preferred services, but also to drive him according the best settings for his needs. The target should be achieved by means of social and semantic strategies, based on the convergence of Social and Semantic Webs → The ability to manage and support this convergence can be considered an added value for a SOA, while SOAs able to support the mentioned aspects separately are quite common
- This scenario expects also interactions among distinct types of collaborative services. In a larger context, this aspect leads to the “Unified Communication” that is improving its importance not only in the large company → The sharing of information and the ability of communicating have a huge importance. Nowadays a SOA should guarantee these feature quickly and freely of any obstacle due to geographical or technical issues deriving from devices (mobile phone, PDA, notebook, etc.) or services (email, chat, SMS/MMS, phone call, video call, etc.)

- This scenario proposes a learning path personalized according the previous knowledge background. This feature is based on one or more ontologies mapping the most important concepts and the relationships among them → A SOA supporting this feature allows to manage a flexible workflow engine based on ontologies, also switchable on the fly

This scenario is partially satisfied by commercial solution devoted to training services. The effective challenge is to accomplish it an instance of the NEXOF framework, considering also the features (i.e. workflows and collaborative services) described in the scenario for needs different from training.

The **inputs** of this scenario are:

- Ontology mapping the most suitable collaborative and training services for each phase of the learning experience
- Typology of work (specified by the teacher): university course, research activity, company training, etc.
- Ontologies mapping the most significant concepts involved in the learning activity
- Didactical contents and their metadata

The **outputs** of this scenario are:

- Specification of collaborative and training services for each phase (Environment)
- Optimized learning path and bound resources (didactical contents) for each learner

This scenario is relevant for several types of organization of distinct size with training needs for their users, for instance universities, large companies, schools, etc. as well as companies that would like to manage ontology-based workflows and would like to provide specific services to their users during their activities (not only for training purposes).

3.6 Deployment and configuration of a generic platform (e.g. an e-Learning one) (S6)

This scenario assumes that a commercial application or solution has to be delivered at customer site, or better has to be available for the final consumer. The delivery of an e-learning platform could be chosen as reference.

The deployment involves several activities, often dependent on each other. These activities could be carried out at customer site and at supplier one.

Usually, it's reasonable a customization phase of the application, other than an ordinary configuration. This customization aims at matching the needs that are specific of each customer in order to overall improve and optimize the business processes.

In this case the following aspects should be weighed up:

- The available installation package should furnish appropriate tools for the delivery, the configuration and the integration in order to fit structures, processes, roles, policies, etc. of the hosting organization
- The application should be flexible enough to be really customizable in a suitable way

The real deployment work follows the assessment phase between the customer and the supplier at several levels.

The first level is related to the deployment approach of the customer. This aspect leads to several distinct chances. The customer could require that the application has to run on his own infrastructure or could ask for enjoying the application as an external service hosted by the supplier/producer of the application or hosted by a further provider chosen by the customer. In any case, there are the constraints resulting from the distinct deployment models: the application/platform should be able to face most of them. For instance, in case of deployment as a service the security matter is very critical because the customer enjoys remotely the application, i.e. via browser. In addition, in case the producer hosts the application, the maintenance and the configuration are made easier, but the link with customer's needs and requirements could be more critical.

The second level is related to the configuration of the application. Depending on the domain and on the size of the customer, an initial configuration could be set up. In case of the e-learning platform chosen as reference in this scenario, if the customer is a University the resulting configuration will be very different from the best configuration in case the customer is a multinational company. In the first case the final user is the student included in a classroom and following one or more courses. In the second case the final user could be a stand-alone employee following just one course in order to fill some experience lack. A further refining of the configuration could be derived from specific needs, structure and organization of the customer. For instance, some specific settings could be required in case of large company split in several independent departments instead of a centralized structure, likewise other specific settings could be required in case the employees need to enjoy the application via PDA over GSM or UMTS network.

The third level is related to the integration of the application into the policies, roles, tasks, processes of the customer. This aspect could be faced by several points of view. For instance, the customer could require interoperability with other previously existent applications, so as the customer could require that the workflow running inside the new application have to interact with other workflows of the customer. Often, a deep interaction is required between the e-learning platform and the knowledge management system of the company, but also with the personnel department applications, in order to automatically update skills of employees and their CV.

The **rationale** of this requirement is the following: the deployment of an application in the hosting environment faces several aspects related to both customer and supplier needs:

- The customer requires that the new applications should be integrated in his pre-existing infrastructure and synchronized and working cooperatively with already running, unwavering and reliable processes and applications. Of course, new incoming applications should match customer's requirements and should meet customer's needs, according his specific business, structure, roles, policies, size and arrangement.
- The supplier aims for delivering the provided application in the most effective and speedy way, fulfilling customer's expectations and requirements. This target is easy to be achieved in case the application provides agile and easy tools for deployment and configuration. On the other hand, aided deployment and configuration could be useful to manage changing versions and settings during time at manifold clients. Of course, the application should be flexible enough to be adaptable without spending too much time and too many resources.

The overall need, by both customer and supplier points of view, is related to capability of the application to be flexible and adaptable in order to fit the company model of the customer and to reduce at the latest the effort for the implementation of the required customization.

This **scenario applies to:**

- **Service consumers** that are typically the organizations that need to provide training to their employees, students, collaborators, etc. These organizations could be classified according to their own typology: large company, university, SME, etc. For each typology of organization there are many distinguishing aspects linked to specific business, to organizational structure, to technologies, etc. Usually these consumers require new incoming applications or platforms to be integrated to already existent ICT infrastructures, to be synchronized to stable workflows and to other processes, to demand just a few changes in the hosting environment.
- **Service providers** (i.e. the supplier of the application or platform) that have internally several skills to arrange in advance features required to satisfy organizational and technological constraints, to make faster and easier the integration at workflow, process, software and infrastructure levels.
- **Service integrators/developers** with the characteristics described as follows: the supplier of the application or platform is the developer of the service. On the other hand, the integration of service (the main target of this scenario) requires the joined involvement of both supplier and client organization.

The client organization has to provide information about technical environment and has to arrange infrastructure in order to host the new service/application/platform, but first of all the client organization has to provide the information related to its processes, structure, policies than can affect the new application.

The supplier has to install infrastructure components, has to deploy the applications according approaches of client organization, and has to

integrate the overall provided service into the organizations processes and structure.

Problems and challenges in this scenario are focused on quick, easy and reliable deploy of the incoming application. These aspects imply:

- Integration of application with legacy applications, ERP, etc. critical and foremost for the client
- Ability to deploy the application according to the model of software deployment of the client, i.e. Software as a Service or Stand Alone
- Ability to easily configure the application according to the specific domain of the Client (University, SME, telecommunications company, automotive industry, research centres web, etc.)
- Interaction with most relevant processes of the client. For instance, by technical point of view, this aspect could involve the workflow area, but by a larger perspective, this aspect implies and requires the capability to model the company organization by means of architectural facilities

The **architectural constraints** of this scenario include:

- The deployment and configuration phases have to take into account both server and client environments
- The server environment, that provides the service, should carry out the integration at several levels already mentioned
- The manifold client environments (desktop PC, PDA or mobile phone with respective operating system, browser, etc.) influence the final configuration of application; in order to preserve the properly working with all expected client devices

The consortium partner MoMA has provided the scenario. MoMA is involved in the production of innovative service oriented learning and collaborative platforms. The deployment and configuration of these platforms are usual activities at customer locations.

The scenario takes into account the analyses of the requirements, needs and problems, encountered during these activities with MoMA's customers: large companies, SME, universities, schools, multi-department stores, etc. However, the scenario applies to common situations in which a new application is installed in a hosting environment, facing integration issues at technical, business, organizational, etc. levels.

This scenario has been selected because:

- This scenario faces quite common needs during the deployment of an application at the customer site or at the provider one → The deployment itself (as well as the configuration and the customization tasks) can be very quick and easy only if the architecture of the new application has specific features, mainly because the hosting environment can have several deployment approaches (i.e. SaaS), distinct maintenance and security policies, etc.

- The scenario points out also the adaptability issues also in non-technical areas. For instance, the architecture underlying the application should allow to configure it according the needs of several contexts: from large company to SME, from large university to a medium research centre → The framework, the application is based on, should make easy this adaptability, in order to minimize the effort requested for it.
- The scenario emphasizes the integration of the new application with the overall hosting environment at several levels → The architecture of the application should allow an easy integration at technical level, but mainly at business and organizational levels. These last two aspects allow to minimize the impact on the preexisting processes (and related software) and maximize the benefits from the use of the new application.

This scenario faces a very common need. Of course, it is well addressed in case of integrated solutions (i.e. the most popular ERPs), but it quite challenging in other cases in which the integration often means to force the components to work together and, on the other hand, means to be unable to fully exploit their capabilities.

The **inputs** of this scenario are:

- Specification of the deployment model of the customer (SaaS, virtualization, physical hosting, etc.)
- Specifications of the business processes interacting with the new application
- Specification of the organizational model of the customer
- Technical specification of the hosting environment

The **outputs** of this scenario are:

- An application configured according the customer model
- An application integrated in the business processes of the customer
- An application integrated in the technical infrastructure of the hosting environment

This scenario is **relevant** for:

- Providers of applications and platforms
- System integrators
- Customers strict in demanding the most effective and profitable integration of the products/services they buy

3.7 e-Health: complex diagnostic workflow (S7)

During a consultation in a hospital/care centre or at a local doctor, typically activities as sketched below are carried out, when the doctor examines the patient. Thereby, focus is either on determining the patient's complete health status which enables the doctor to recommend further actions, or on integrating further useful services in the workflow once the complete health status is determined and the doctor is about to take measures. The actions typically to be taken are the following:

- Accessing previously collected health data of the patient
During the medical examination, the medical staff (“the doctor” for short) needs access to the patient’s previously recorded and now archived health data (that is, blood test results, X-ray images, etc.) which were either recorded in the same location or at a different place, for instance at a different hospital (which possibly belongs to a different hospital chain) or in the ordination of a local doctor.
- Accessing the patient’s present health data
The doctor also needs access to the data recorded online during the consultation by either the doctor himself or his assistants. He may, in addition, need data that was recorded shortly before the consultation, or that was collected in the hospital or at home during a long-term monitoring with a mobile diagnostic device like, for instance, an ambulatory blood pressure unit or something similar. It is even conceivable that the doctor would use diagnostic data received from nanobots (that is, agent-like devices of nanometre-size brought into a human body for diagnosis or even for therapy).
In addition, whatever kind of data he is using, the doctor should be supported in his analysis by expert systems and databases.
- Device independent and mobile access of patient health data during examinations
To reach a diagnosis during a complex examination, the doctor may need to use several devices in several locations. Typically, they are all in the same hospital, but also an external examination at the patient’s home is conceivable.
While the doctor is changing devices, they are still included in the same workflow. Its status should be stored when the examination with one device is finished and it should be retrieved when it is continued with another device. The devices could be a general-purpose handheld computer or a specific integrated device for medical diagnostics, for instance, an X-ray device. Depending on the capability of the device including its communication facilities (if, for example, only a wireless connection is available), the amount of transmitted data and the visualization has to be configured accordingly.
- Placing a phone call
The doctor might need to call a colleague in for consultation or to evaluate a specific result. To this end, the doctor has access to directories and can place a phone call by one mouse-click from just the computer he uses at that moment. This feature may be taken a step further to collaborative environments and expert call centres.
- Making a reservation for a particular medical examination, surgery or treatment

If the doctor decides as a result of the medical examination that the patient needs additional treatment, he could easily reserve the necessary medical device or make the respective appointment (by, for instance, just clicking a button). There is no need to switch over to a reservation application or to call a responsible person.

Owing to the demographic change and to increasing costs, healthcare has to become more effective while using its resources more efficiently. Therefore, IT support is a critical factor in hospital workflows and diagnostic workflows. e-Health seeks to provide new kind of services and a better integration of new and existing services, thus supporting the work of healthcare staff. This scenario takes the viewpoint of medical staff. It does not address administrative hospital workflows like patient admission, accounting and the like (though integration would be very reasonable).

The implementation of this scenario will allow achieving the following:

- Improved quality of healthcare through complete access to patient data
- Reduced duration of healthcare activities through ubiquitous and immediate access to patient data, facilitated access to expert consultancy, and easier planning of further examinations and therapies
- Side effects are cost reduction through supported workflows and improved usage rates through automatic scheduling of expensive medical devices; a further benefit is that complete access to patient data surely avoids redundant examinations

This **scenario applies to:**

- **Service consumers:** even if the patients benefit largely from the proposed scenario, the real service consumers are the doctors and the medical staff. Their expectations comprise the following features:
 - The doctor needs access to intranet and Internet services from within the hospital or his office using a high-performance network as well as wireless access. For some devices, remote or wireless control is necessary.
 - The doctor potentially has to access a large amount of archived data and also creates huge quantity of data during the examination. He requires a scalable access to this data when using different kind of devices. Scalability affects the amount of data that can be transferred but also their visualization on small screens.

In this scenario, the usage of most services is either pre-planned or necessary from a medical point of view; sometimes, it even may be critical. Thus, charging for the services should be included in the overall medical fees. Only access to external patient data, expert databases, or placing certain analysis tasks to external labs (for instance, complex image analysis) could possibly result in an additional charge. Selecting those services, however, is today usually done by the hospital administration and not on a case-by-case basis by the doctor or in an automated way, taking only into account the currently published service charges.

- **Service providers:** the usual hospital services, like patient record service, measuring services, mobility service, reservation service and telephony, are provided by the hospital. Today, telephony services are already outsourced in some hospitals. Access to patient records may also require an external service

(collaboration with other hospitals or patient record archives). All in all, however, the share for external service providers is limited in this scenario.

- **Service integrators/developers:** The system is developed by IT system provider(s).
Basically, the doctor is not expected to create new service compositions, only to configure them, for instance, for devices since devices may change or there may be new devices to be integrated on demand. Thus, the doctor may be asked to integrate them as new services. A wizard, however, must guide this.
Also, it might be more efficient not to have all services for all the existing devices already integrated from the beginning. Instead, they may be integrated only when needed. Again, the doctor merely selects the devices in a catalogue and a wizard carries out the service composition.

Problems and challenges in this scenario include:

- Legal and technical issues with distributed and shared patient records
- Privacy and security compliance with respect to healthcare IT providers and medical staff as, for instance, required by the US-Congress-enacted “Health Insurance Portability and Accountability Act” (HIPAA)
- On demand integration of data from various devices
- Storing working sessions and allowing to move sessions between devices
- Integration of external applications (telephony, reservation, external patient records)
- Integration of distributed workflows, distributed transactions, federated identities
- Integration across domains
- Horizontal (enterprise information systems) and vertical integration (devices)
- Platform heterogeneity, interoperability
- Dependability, performance, security, and trust

As **architectural constraint**, the devices involved in this scenario have to be manifold: desktop PCs, handheld computers, diagnostic devices, etc. They are connected either peer-to-peer, through the hospital intranet or via Internet, including wireless communication.

Existing prototypes for electronic health records are based on Web Services.

3.8 e-Health: assisted living (S8)

Assisted living systems are supposed to support persons with a chronic illness or those with a need for constant medical surveillance (as, for instance, elderly people) in a way that they can live at home. Therefore, this scenario refers to devices such as, for instance, blood pressure units or blood glucose meters that are operated either by the patient, by a nurse, by a paramedic or by another member of an ambulance crew. A doctor may even

operate those devices remotely. Also appliances continuously surveying essential vital functions of the patient may be utilised.

Each device transmits its measurement results to a PC that provides instant analysis but also long term monitoring. Via the Internet, the PC can also access environmental information.

From the data gathered, a diagnosis can be derived either in the home of the patient or remotely in the doctor's practice or in the hospital; nurses or physiotherapists may also use the data to, for instance, provide remotely recommendations for physical exercise or to give detailed dietary advice based on the day's health condition of the patient – and without contacting the patient personally every day.

The data is also made available to a doctor during a regular check up. The development of the patient's health condition is stored in an electronic patient record that will also be accessible during a possibly future stay at a hospital or in case of an emergency. If the vital functions degrade in a dramatic way, the assisted living system can trigger an alarm; the arriving ambulance crew can, of course, access the patient's healthcare record as well.

The **rationale** of this scenario is the following: to face the demographic change, society has to improve healthcare for elderly people. Healthcare at home will see major changes especially in the field of prevention and monitoring, as these are drivers for cost reduction in the health system while at the same time they help improving the quality of life for elderly people. The term "assisted living" has been coined for IT-based solutions that allow the elderly to stay independent while receiving optimal healthcare.

This **scenario applies to:**

- **Service consumers:** there are various user roles:
 - The assisted person (patient) who may operate some of the services by him- or herself
 - The medical personnel (nurses, doctors, diet advisors, etc.)
 - An ambulance crew which may use the services of the assisted living system in case of an emergency

The access to the services and data of the assisted living system is restricted by the respective user role.

- **Service providers:** most services run on devices, some run in a hospital or care centre involved in monitoring the assisted person, some external services may be integrated, for instance a weather forecast with information about the pollen count (if the patient is allergic and needs this kind of information).
- **Service integrators/developers:** a healthcare centre offering an "Assisted Living Service" may be confronted with the need to configure selected services. So, for instance, services providing pollen count may differ according to the geographic region. It may also be the case that different monitoring services of different hospitals have to be used. Or the

services related to patient health records are different in different regions depending on, for instance, governmental regulations.

The **challenges** of this scenario are:

- On a technological level: Integration of devices with enterprise software while providing availability
- On a business process level: Integration of the different workflows of the involved stakeholders (that is, doctors, nurses, physiotherapists, ambulance crews, etc.); furthermore, provision of a secure access to a potentially distributed patient record
- On an administrative level: Collaboration of different public and private institutions with different responsibilities across boundaries; in particular public institutions belong to different administrative areas (local, regional, national and European) with usually different regulations
- On the compliance level: Compliance with privacy and security rules as, for instance, posed by the US-Congress-enacted Health Insurance Portability and Accountability Act (HIPAA), which, in the context of this scenario, particularly addresses healthcare providers

The configuration of such a complex system as presented in this scenario is a challenge on all levels because of its inherent dynamicity, which results from the need to configure the involved devices in a personalized way according to the patients' health status and to adapt them dynamically if changes have occurred. Further dynamic adaptations may become necessary in the case of emergencies, where some workflows may be assigned higher priority than others to ensure the unrestricted availability of critical system functions. In addition, regulatory changes need to be incorporated.

The **architectural considerations and constraints** of this scenario include:

- Heterogeneous platforms
- Device integration

3.9 Traffic management: large scale emergency handling (S9)

This scenario assumes a large area with loosely coupled traffic management systems, operated by local and regional authorities. The scenario distinguishes the "normal situation" and the "emergency case":

- **Initial (normal) situation:** In a certain area, various municipal and regional traffic systems are available and they guide (long distance) highways as well as village, township and city traffic. They are tailored to the respective local conditions, depending on the time of the day. This means, that, according to the known and expectable traffic situations, they control and direct the traffic control devices (traffic lights, directions of multi-line highway lanes, lane and street closures, etc.), optimising probably throughput or other set parameters (as for instance air pollution or noise reduction). The local traffic control systems integrate the needs

of pedestrians, bikers, car driver, etc. and take into account school hours or other factors, which alter, traffic demand.

There is a host of distributed devices providing input on traffic density, car speed, air pollution, traffic light status, congestion indicators, etc. The devices used in the various traffic systems are rather diverse, even if they perform the same or similar functions (this is due to the devices' longevity and that they are provided by various firms during the life time of a traffic system; their diversity is also largely caused by the independence and the lacking co-operation of the political institutions involved in the acquisition and maintenance of the respective traffic systems).

- **Emergency case:** Due to a serious accident, a highway has to be closed. This means that (a) rescue forces have to be directed to the location of the accident and their arrival has to be facilitated; (b) the traffic has to be deviated through places not intended for heavy traffic.

Consequently, conflicting and unforeseen situations will occur. The control has to be reconfigured according to unplanned control patterns, which perhaps have been already used at other locations where they are offered for re-use. Those available or retrievable control patterns have to be configured to optimise the local situation, taking into account the changed conditions and the still remaining local settings (the local theatre, for instance, will – notwithstanding the emergency situation – still close at the announced time). Deviation roads have to be selected accordingly. Signalling patterns (traffic light phases) have to be adapted in order to reduce stop & go traffic (that should help to keep air pollution low). Decisions have to be integrated in order to keep “feature interactions” with undesirable results at bay (that is, two actions, each being strictly goal directed and effective itself, may result in a disaster if applied together).

Return to the “normal situation”: There is probably a slow evolution back to the “normal”. If measures have to be taken immediately in the emergency case, to revert to the normal situation may require a slow process that has to be carefully reacting on the various inputs from the traffic sensors.

The **rationale** of this scenario is the following: in large urban agglomerations, an appropriate traffic management is imperative in order to ensure the quality of life for the residents. Since traffic does not stop at administrative boundaries, traffic management systems have to integrate across cities, townships and rural districts (at least in those nearby to urban areas). There are many different strategies adapted to the respective local situations. In case of emergencies, when the traffic pattern changes drastically, these optimal local strategies may become problematic and be probably even a reason for congestions themselves. In particular, if ecological objectives are also pursued (for instance, to keep air pollution low), these strategies have to be adapted swiftly.

Traffic management is a complex task. It has to cope with a large amount of different sensors and their data, a large number of rules, many algorithms and various mechanisms. The need for federation in inhomogeneous settings

makes things even more complex. The scenario deals with unexpected change in such an environment.

This **scenario applies to:**

- **Service consumers:** the consumers of the outcome of the traffic management system in the scenario are the traffic participants, that is, primarily the citizens, the drivers of cars, motor bikes and bicycles, but also the drivers and users of public transport, and of course, the rescue forces which need to get to the site of the accident.
The problems to be solved are manifold: any traffic congestion should be avoided or at least minimized; additional pollution of the environment (due to a high traffic volume) should be reduced; access of the emergency area should be facilitated for emergency vehicles (ambulances, fire brigade, police).
The traffic management system has to react rather quickly to emergencies. Other regulations may be performed at a lower pace.
Managing traffic is an administrative task that has to be performed by the local authorities. Thus, the consumers (that is, the citizens) support it with their tax money.
- **Service providers:** basically, the local authorities can be assumed to be the provider of a traffic management system. However, provision may be outsourced to a private firm, which acts on behalf of the local authorities.
- **Service integrators/developers:** Local authorities cannot be expected to develop a traffic management system themselves. There will be an IT system provider; actually, it can be assumed that there are several providers with different systems. Also, the sensors that enable the traffic management system to identify certain situations will presumably be provided by different enterprises.
However, local authorities have to decide about the traffic management strategies, which mirror the political goals (for instance with respect to ecological ambitions of a certain city council, etc.).

Problems and challenges in this scenario:

- Integration of several traffic management systems with different strategies (cities, rural areas)
- Integration of many heterogeneous devices (traffic control devices like traffic lights and variable direction signs, traffic monitoring devices, etc.)
- Acquisition of adequate new strategies (control algorithms) in order to react upon an unexpected situation
- Constant evolution
- Dependability, performance (emergency reactions), security and trust (in particular with respect to the acquisition of new strategies)

As **architectural constraint**, the devices involved in this scenario have to be manifold: desktop PCs, handheld computers, diagnostic devices, etc. They are connected either peer-to-peer, through the hospital intranet or via Internet, including wireless communication.

3.10 Crisis management system of systems (S10)

An airport can face numerous situations where major crises can arise. Some of them require responses with the highest degree of coordination, communication, system interoperability and system availability.

A Crisis Management system allow for shared situational awareness from the field up to the command centres, reliable information from fused sources and the shaping of a full, seamless communications link. When a crisis occurs, clear communications are essential for successfully controlling the situation. Swift and effective measures require reliable information and its effective dissemination to all involved organization and actors. Effective services trough software and hardware solutions are required for the direct support of all crisis management activities: everything from incident reporting to logistics workflow support. In case of a major crisis, the interoperability with military information systems will be required.

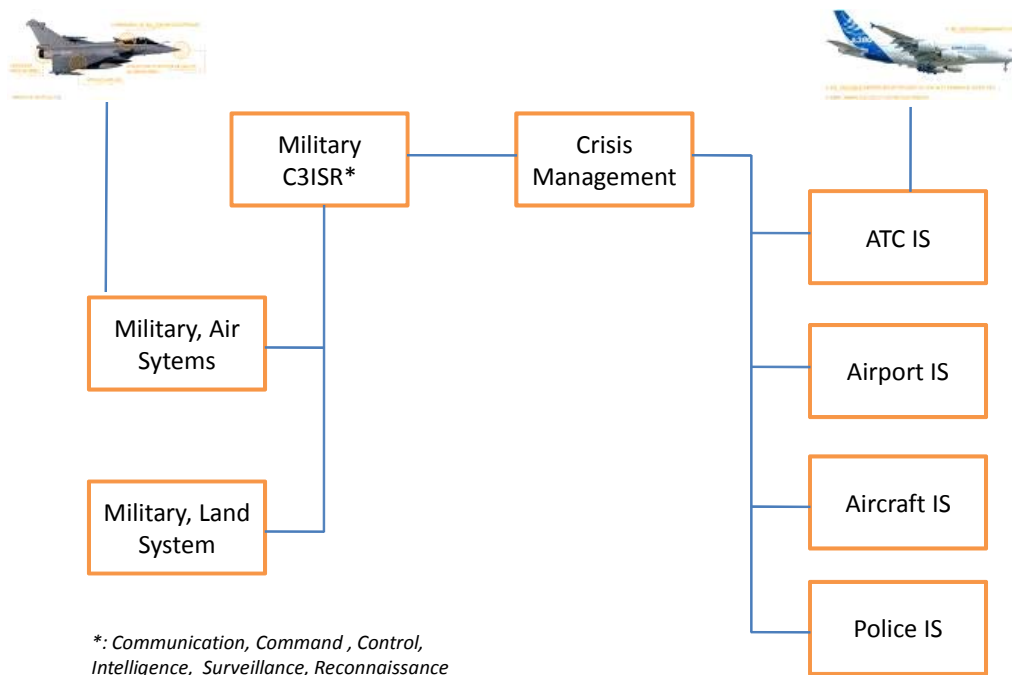


Figure 5: Systems Involved in the Airport crisis scenario

This scenario focuses on the necessary interactions between these different organizations in order to manage and reduce the impact of a crisis situation occurring at an airport. These organizations can be described as follows:

- The Crisis management cell depending of the Ministry of the interior witch is responsible for internal security and the protection of the constitutional order, for civil protection against disasters and terrorism.
- The Defence Headquarter and the Military Communication, Command and Control systems: Command and control functions are performed through an arrangement of personnel, equipment, communications,

facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission.

- The Airport Operator is responsible for the management of the day-to-day operations of the respective Airport and to ensure the airport is keeping up with all specific performance criteria as prescribed by international standards. This includes the safety management of operations, opening / closing runways (and movement areas as appropriate) etc.
- The ATC organization is responsible for services provided by ground-based controllers who direct aircraft on the ground and in the air. The primary purpose of ATC systems worldwide is to separate aircraft to prevent collisions, to organize and expedite the flow of traffic, and to provide information and other support for pilots when able.
- The Aircraft Operator is the organization that is responsible for the use of one or more aircrafts (as owner, lessee, or otherwise), for the purpose of air navigation. Aircraft: This role refers to the plane and its crew belonging to an airport operator.
- The governmental Policy and more particularly the information databases about criminals and terrorists.

Although all these roles need to work and collaborate together they are completely independent of each other.

The structure of the scenario of our case can be defined as follows:

- A major disaster had occurred in Brussels Airport. A crisis management cell is created under the interior ministry responsibility. This event could be the first act of a terrorist operation.
- The crisis management system is interconnected to the military Communication, Command and Control information systems. Information, events and reports can be in real time shared and exchanged. Video conference is established between the crisis team and the defence command headquarters.
- A plane flying to Brussels, disappear on the civilian Traffic Control screens (secondary radar), and the communication with the cockpit are interrupted. The civilian Air Traffic Control teams required information from military primary radar to know if the transponder has been disconnected.
- *The military radar confirms that the plane is still flying to Brussels; do terrorist have taken the control?* The crisis management cell asks the Air Force to prepare an intervention. To define the flying plan, military Air Forces need information from civilian ATC and will ask changes in the traffic.
- Information about passengers is transmitted to the different teams to verify identities and submit a crisis communication plan to the interior minister.
- The Aerodrome ATC will adapt the airport configuration taking into account the constraints defined by the airport authorities.

- The crisis management team required to the military headquarter a ground protection of the Airport and of the logistics roads accessing to the airport.
- Once the airport capacity and configuration are known, the arrival and departure flows can be updated to match the actual constraints. ATC, ATFCM and airport authorities have to collaborate together in order to take the appropriate decisions.

The scenario involves the definition, development and simulation of a crisis management super-system based on pre-defined, existing systems (either real or emulated) and newly defined systems.

Considering **Errore. L'argomento parametro è sconosciuto.**, the link between the military C3ISR and the Crisis Management although depicted by a simple link DOES NOT imply a symmetric information sharing paradigm. In fact the only information that would be shared is information that is in the common domain of the information model but owned by the military (e.g. sector plan specific to military controlled airspace can be shared with an aircraft which is in a crisis and need to find the shortest path for landing).

Furthermore Thales has experienced in the UK where UAVs can enter and share the same landing space than aircrafts (e.g. Watchkeeper Programme in which Thales is involved)

Of course in all scenarios security and integrity of information would be guaranteed by commonly agreed and implemented mechanisms.

The Airport Crisis Management system proposed in this case exhibits a range of complexity characteristics:

- **Size:** a Crisis Management System of Systems involves the coordination and integration of several large-scale distributed systems involving very large legacy code bases.
- **Heterogeneity:** several different classes of system will need to be integrated (real-time, information, communication, security, military, etc.). In addition, each will have a specific and unsynchronized development lifecycles inferring huge challenges in terms of evolutionary maintenance and technical interoperability.
- **Distribution and Dynamicity:** Crisis Management is inherently a domain where the ability to dynamically discover and connect distributed components and systems represents a strong added-value in terms of responsiveness and adaptability.

The scenario is representative of various complex systems challenges. In particular it highlights the complexity dimensions that come into play with the interactions of data distribution systems in different domains, with enterprise information systems, mission critical safety systems amongst others.

The **rationale** of this scenario is the following: the emergence of a number of key technologies and other converging factors is creating new opportunities for more integration and communication between ICT systems:

- Network and computing ubiquity

- Convergence of IT and Telecommunications
- Global standardization and convergence around the Internet protocols stack within industries where proprietary protocols used to be the norm; for example defence and air traffic systems
- Interoperability standards : industry, civilian-military, EU, NATO
- Resilience in system of system environment,
- System management and monitoring, configuration and reconfiguration capabilities
- Availability of more contextual and behavioural information through the ambient intelligent environments
- Designing a conceptual model of a complex system (critical infrastructures with inter-dependence) to insert and detect essential and critical markers.

These emerging or maturing trends are enabling new classes of systems amongst which the concept of system of systems is spreading both in the military and civil domains. In this context a system of systems is a global system based on the cooperation of autonomous systems. Systems of systems can be characterized by the following: operational independence of the elements, managerial independence of the elements, evolutionary development, emergent behaviour, and geographic distribution.

At the same time, the ICT market is characterized by a constant need to bring to the clients new innovative solutions and services in order to deal with increased complexity and huge amount of functionality at ever diminishing prices. Within this context, the effect of the constant increasing technology complexity & capabilities as described by Moore's law² is out-pacing the current productivity improvements of the solution providers. This leads to an ever-widening expectation productivity gap that must be addressed.

Principally, the following gaps have been identified:

- Tools to address system complexity and to share understanding between the different stakeholders. These stakeholders are customers, operational users, co-contractors or sub-contractors, analysts, architects, integrators or validators each having their specific view of the system of systems.
- Robust concepts and languages that allow the description of these systems architectures, properties and business capabilities they support.
- Processes, methods and techniques for architecting, developing, integrating, validating and managing the evolution of these systems.

It can be concluded therefore, that the definition, development, deployment and maintenance of this class of complex system is challenging. This targeted scenario clearly supports the NESSI initiative regarding the development of technology for producing ambient intelligence and open systems that enhance the safety and security of citizens.

² Moore's law: Based on empirical observation, Gordon Moore projected in 1965 that the complexity of electronic circuits will double every 18 months. Up and until now this has proven to be true, and it is expected to continue for some decades.

This **scenario applies to:**

- **Service consumers:** the consumers of this kind of systems are the scenario participants, that is, primarily the different organizations involved in the crisis management and operations, citizens who are travellers or end users of the aviation industry – airports, aircrafts, Airport Operators, Air Traffic Control.
- **Service providers:** as can be expected from a complex system of systems scenario, there are numerous, multi-modal, interdependent service providers who are also at times the service consumers. An example of a candidate crisis is the one of a bomb alert and terrorist threat, with crisis management relying on the cooperation of several systems including Airport Security Systems, Departure Control Systems, Police Forces Systems, Flight Management Systems, Intelligence Tools, Geographical Information Systems, Rescue and Operations Management Systems, etc., all of which need to work in concert with each other to execute the scenario.
- **Service integrators/developers:** the various organizations involved are focused on executing their own roles and not on developing the IT systems to support their objectives. Nor is it reasonable to expect them to actually achieve the complex independent interaction to make their systems work together in concert effectively in a crisis.

The main **challenge** is to better support the controlled, efficient and evolutionary definition, development, configuration and deployment of this category of complex system. This involves developing more adequate meta models and representations of the different engineering viewpoints based on standards:

- Modelling the collaboration grid between systems
- The involved ontologies
- The new super system-level operational capabilities and services
- The static and dynamic configuration options

These modelling techniques need to be integrated in an architectural framework supporting verification (i.e. tests and nominal functioning) and validation (i.e. conformity to specifications), either through analysis, model-based testing or mock-up synthesis and simulation. This framework needs to enable the synthesis of implementation artefacts, with a focus on integration and interoperability management. This means there should be the capability to allow framing reference models and architectures of existing systems, business processes, etc that are not domain specific as they need to allow different types and classes of systems to work in concert.

Other challenges include:

- Interoperability standards adoption for required services particularly between military and civilian systems.
- Quality of service aspects, especially security and performance that are key engineering concerns that need to be addressed.

- Legacy evolution and integration must also be considered, along with heterogeneity of the system components.
- Exploratory features of the case system include a level of dynamic monitoring and re-configurability; the synergistic usage of models for design and for exploitation will be explored as a solution component.
- Verifying and validating the expected system of systems behaviour through simulation.

The **architecture** will involve multiple systems of different categories, connected by a variety of means and which will need to manage the constraints of allowing for a unified model driven system of systems interaction utilizing gamut of architectural styles relevant to the specific system contexts, such as DDS, REST, EA and SOA.

3.11 Effective and efficient collaborative decision making (S11)

This scenario is focusing on Collaborative Decision Making (CDM) applications. Although illustrated on air traffic management (ATM) domain this scenario is applicable (can be declined) to a broad range of various applications domains where effective and efficient CDM is at stake (e.g. Transport, Crisis Management domain, etc.).

By essence a CDM application (e.g. Air Transport CDM) covers a wide range of strongly interacting processes and it is recognized that the various systems currently operated by the various stakeholders (such as in the context ATM-CDM: Airlines, Air Traffic Service Providers and the Airports) have for historic reasons little inherent capability for interoperability. In addition to that, interoperability is mostly regarded as a connectivity and data communication problem, but in fact must be raised to the level of the communication of meaning (semantic and contextual interoperability).

The foundation of a Collaborative Decision Making (CDM) application is thus the sharing of business operation (e.g. flight plan operation) information between multiple and different systems, and implementation of new collaborative components and services to improve the efficiency of the operations based on a “system-wide” approach. The services should be focused on the optimization of resources usage by the various stakeholders (for ATM-CDM airlines, airports and Air Traffic Services providers) and lead the key decision makers having the most pertinent information to ensure best decision making.

The **rationale** of this scenario is the following: the current decision-making processes and procedures provides high level of safety but regularly leads to situations in which there is lack of tight real-time synchronization between all the stakeholders due to the absence of robust common situational awareness and collaborative infrastructure. Indeed operation (e.g. operation planning in CDM-ATM, that is, the time schedule and estimates for key events like aircraft departure, take-off and arrival at the gate) are not established in a collaborative way but each stakeholder establishes its own

individual scheduling based mostly on “one to one” co-ordination and decision-making process.

This, in turn, generates major inefficiencies in the prediction and use of the available capacity and resource by the various CDM application stakeholders (such as for CDM-ATM: Airlines, Airports and Air Traffic Service Providers) and leads to a number of problems with direct Business, Societal and/or Economic consequences (e.g. for CDM-ATM: loss of connections between flights, non-optimal use of the airport capacity, disruption of the gates allocation and passenger transfer and loss of air traffic capacity due to low predictive departure slots).

Therefore, the rationale behind this scenario is to investigate how a next generation SOA system, through innovative concepts and technology advancement which might result, could meet the operational demand of moving from individual & concurrent Decision-Making to Distributed & Collaborative Decision making.

The scenario aims at achieving and demonstrating the following:

- Improvement of the overall efficiency/performance of the whole global CDM application (effective and efficient Collaborative Decision-Making),
- “Semantic” Collaborative Information Backbone,
- Increased degree of satisfaction of various actors ranging from stakeholders to end-users (passengers).
- Enhanced Situational Awareness,

This **scenario applies to:**

- **Service consumers:** service consumer can be any representative of the various stakeholders’ categories (e.g. Airlines for CDM-ATM). The service consumer consumes services locally and globally (coming from service providers) for operations of concerns on a (conceptual) object (e.g. Flight plan for CDM-ATM) shared with others, for the sake of business interaction continuity some (since mobile) need to get access to the same information on mobile devices.
- **Service providers:** a service consumer can in turn become service provider and vice versa.
- **Service integrators/developers:** an IT system provider develops the software.

The **problems and challenges** in this scenario include:

- Socio-technical design of CDM system
- Enhanced/Shared Situational Awareness/Shared Situational Understanding)
- Naturalistic collaborative decision-making
- Generation of trust and confidence in decision-making
- UI and USI independent of terminals/devices while staying user task and profile consistent
- Semantic interoperability
- Harmonization object-operation data exchange between stakeholders

- Understand the social nature of the interaction among various stakeholders this in order to comprehend it, regulate it and make it collaborative (“activity logic” of the actors,

3.12 e-Commerce information sharing (S12)

Within the last years, more and more companies have switched to use instruments such as outsourcing and Software-As-A-Service (SAAS) solutions to liberate a company from creating and maintaining all parts of their e-Commerce solutions manually. This has been the case for large and medium companies for many years but recently, even small and micro-sized companies have been attracted by these approaches.

This situation has increased the number of companies that are participating in typical e-Commerce scenario significantly and e-Commerce today usually involves many different partners. In an ideal implementation, many of these partners interact in the background without being visible for end-users/customers.

The following example will describe a scenario that is not uncommon in e-Commerce. It deals with purchasing a software product at an Internet website. Although it looks very simplistic from a user perspective, it is highly complex and challenging from the corresponding e-Commerce vendor perspective.

Outsourcing and Software-As-A-Service (SAAS) solutions have increased the average number of partners involved in e-Commerce scenarios. The reason for using these instruments and for involving additional partners in the value chain is usually two-fold:

- Companies may usually save money when handing some processes over to more specialized companies.
- Companies may use the knowledge and expertise of partner companies. For example, small companies would normally not be able to provide a multi-lingual web-shop on their own. But outsourcing to a specialized web-shop-provider will usually allow them to include many different languages in their shopping interface without problems.

Companies thus need to ensure that there is a seamless communication between all participating companies. This basically means that a high interoperability is necessary in order to allow companies to exchange data and to use the data that has been exchanged in the right way. This loosely coupled process makes it important to track information, to monitor system functionality and to keep awareness of security and reliability issues. The scenario described above is therefore a well-suited example for a complex scenario involving many different partners, service and data structures that would benefit from NEXOF as a base for their cooperation.

This **scenario applies to:**

- **Service consumers:** The user view, described in this scenario consists of a group of users that are interested in purchasing software from an Internet Software Vendor (ISV).
 1. They have read about a software product on a download portal in the Internet and they have followed a link to visit the website of the ISV. They use the product specific information of the ISV to get detailed knowledge about the product.
 2. They are allowed to download a trial version and after a test period they decide to purchase the full software product from the web-shop of the ISV. They therefore visit the ISV website again and are clicking on the 'buy now' button on the website of the ISV. They see a list of product versions and delivery methods (e.g. delivery on CD) and they see a button next to each entry allowing them to order. This button leads them to a checkout page allowing them to select their language, preferred currency and the way of payment (e.g. credit card or PayPal).
 3. When clicking on the next button they receive a dialog window asking them if they want to upgrade their purchase to a software bundle giving them a discount of 20% if they add a second product to their purchase. They choose not to accept this and to continue with their purchase. After typing in all customer information, they receive a confirmation page and are requested to check their email inbox for a delivery code. Back at their email program, they check their email, where they find an invoice from the ISV and a secure way of downloading the software product that they have purchased about 5 minutes ago.
 4. About five days later, they find a friendly email in their inbox asking them if they are satisfied with their product and rate the purchase by different criteria in order to allow the ISV to optimize the order process.
 5. About 2 months later, they receive a newsletter of their ISV informing them of an update of their software product, which is immediately available for download.
- **Service providers:** In order to identify the different service providers that are involved in this scenario, the table below will show an overview about possible services that are outsourced to external partners in typical e-Commerce scenarios today.
The table refers to the scenario described in the section "Service consumer" above and breaks this scenario down into different steps that may be outsourced by the e-commerce Vendor to external service providers as described in step 1 and 2 of section "Service integrator (e-Commerce Vendor)".

Table 11: Service providers involved in the scenario

Scenario element of section "Service consumer"	Service provider involvement	Example provider/s

<p>Users have read about a software product on a download portal in the internet and they have followed a link to visit the website of the ISV.</p>	<p>Business Intelligence or Analysis</p> <p>Analysis providers allow e-Commerce Vendors to better track any analyze the behaviour of their customers. This allows them to find out where users came from and how they have reacted to specific changes in the order process.</p>	<p>Google Analysis</p>
<p>They use the product specific information available from the ISV to get detailed knowledge about the product.</p>	<p>Digital Channel Providers</p> <p>In those cases where the e-Commerce provider itself is a reseller and not the original manufacturer, it is possible to integrate the content coming from the original manufacturing company(ies) into the website of the e-Commerce vendor – the so-called ‘Digital Channel’ – through either structured product catalogues or structured marketing information. This allows an exchange of information along the whole supply chain of products.</p>	<p>TIE Digital Channel, TIE Kinetix</p>
<p>Users are allowed to download a trial version of the software.</p>	<p>Data Storage Providers</p> <p>Data Storage providers allow e-Commerce vendors to store data and provide it to their customers again. The benefit is that the data storage providers ensure that the data is stored in a highly available and high scalable environment.</p>	<p>Amazon Simple Storage Service (Amazon S3)</p>
<p>After a test period they decide to purchase the full software product from the web-shop of the ISV.</p>	<p>Web-shop Providers</p> <p>Web-shop providers provide a basic web-shop installation that is highly customizable by e-Commerce Vendors. They care about maintenance and updates and usually provide the corresponding software in a ‘Software-as-a-Service’ manner (SAAS).</p>	<p>1&1, Hosted Web-shops</p>

<p>Users visit the ISV website again and are clicking on the 'buy now' button on the website of the ISV. They see a list of product versions and delivery methods (e.g. delivery on CD) and they see a button next to reach entry allowing them to order.</p>	<p>Product and Stock Management</p> <p>Allowing the real-time check of the current situation is necessary in order to show customers the current availability of goods. While this is unimportant for virtual goods that are delivered electronically, it is important for e.g. delivering products on CD or with a bundled hardware product. Stock management providers allow to check and plan the stock situation (e.g. out of stock, numbers left etc) and they may even interact with end manufacturers in case of having a complex supply chain.</p>	<p>TIE Kinetix, VMI</p>
<p>This button leads them to a checkout page allowing them to select their language, preferred currency and the way of payment (e.g. credit card or PayPal). When clicking on the next button they receive a little window asking them if they want to upgrade their purchase to a software bundle giving them a discount of 20% if they add a second product to their purchase.</p>	<p>Order Processing</p> <p>Providers assisting the e-Commerce Vendor in taking and processing orders are allowing the e-Commerce provider to enable a multi-lingual order process and to accept different currencies. They also allow the definition of discount coupons and price configurations.</p> <p>Currency Conversion</p> <p>A service provider may be involved for fetching currency exchange rates on a daily base.</p> <p>Payment Processing</p> <p>In many cases, the payment process is handled by the order processing service. However, it is also possible to handle payment by integrating different payment providers such as PayPal, Google Checkout or Visa. There are several service providers that allow a holistic handling of credit card payments such as WorldPay.</p>	<p>Digital River, Cleverbridge, Avangate</p> <p>XE.com; webserviceX</p> <p>PayPal, Google Checkout, iPayment (1&1), WorldPay</p>

	<p>Messaging</p> <p>When involving an external party for carrying out the physical delivery, they may automatically be informed about an order and receive the delivery details electronically. In the use case of section “ Service consumer”, an external provider might deliver a CD to the customer.</p>	<p>TIE Messaging Portal</p>
<p>After typing in all customer information, they get a confirmation page and are requested to check their email inbox for a delivery code. Back at their email program, they check their email, finding an invoice from the ISV and a secure way of downloading the software product that they have purchased about 5 minutes ago.</p> <p>About five days later, they find a friendly email in their inbox asking them if they are satisfied with their product and rate the purchase by various criteria in order to allow the ISV to optimize the order process.</p>	<p>Mash Up Providers</p> <p>The Web 2.0 hype has lead to a large number of distinct components that may seamlessly be integrated into websites. This allows e-Commerce providers to integrate services such as Ratings, Surveys or other gadgets into their websites</p>	<p>Google Friends Connect, Yahoo Pipes</p>
<p>About 2 months later, users eventually receive a newsletter of their ISV informing them about an update of their software product, being</p>	<p>Customer Relationship Management</p> <p>Keeping contact with customers is a most important instrument to increase sells and to keep users interested in new products and updates. Marketing providers allow</p>	<p>Salesforce, domeus, TIE Digital Channel</p>

available for download now.	e-Commerce vendors to keep in touch with their customers.	
-----------------------------	---	--

- **Service integrators/developers:** The process described in the last section is a rather typical example that is found everywhere on the Internet today. In many cases, the ISV is providing all functionality on its own, thus sometimes making it hard to fulfil all user requirements such as security issues and handling multiple currencies and payment methods. Alternatively, the e-Commerce Vendor may use a 'one size fits all' service providing all functionality but with little flexibility. However, in the last years the possibility to outsource this process either completely or at least partially has increased. e-Commerce Vendors are therefore able to realize some parts on their own by outsourcing other parts to third parties. This has lead to the situation that the main task of the e-Commerce vendor is no longer the creation and maintenance of all components but rather the integration of third party services with the overall system. The main challenge is therefore turning into a question of interoperability between all systems. This means that the possibility to exchange and understand information gains more and more importance. For the e-Commerce Vendor the overall process when realizing the example above therefore consists of the following main steps:
 - Analysis and breakdown of his overall web-shop functionality into separate components
 - Identification of possible third party providers
 - For each partner
 - Identification of services
 - Customization of services
 - Integration of services (graphically and logical)
 - Setup and maintenance of an "overall" core system connecting all parts.

The main **challenges** outlined by the example above may be summarized as follows:

- **Service Discovery:** finding Business partners that offer a specific service is a major challenge. Of course web search engines such as Google or Yahoo provide a way of searching and finding business partners but what is missing is an easy to use way for finding services based on specific criteria.
- **Service Agreement:** agreeing on collaboration is a necessary step before using e-Commerce services. For example, protocols need to be defined, agreements need to be signed and service fees need to be agreed. Several approaches have been defined in the last years such as ebXML CA/CP for performing a structural collaboration agreement (CA) and a collaboration profile (CP). Most of the existing solutions are focusing technical agreements such as automatic protocol selection.

However, in most real-world scenarios this is still performed in a manual process.

- **Graphical Integration:** integrating services in a graphical way is important in order to give the user the impression of a holistic system with a coherent look and feel. Technologies such as CSS allow the adoption of colours and images and may be used to implement a graphical adaptation of external components to the own system. In addition to this, many services allow a menu-based configuration of their look and feel. For example, many Web 2.0 widgets may be configured using a configuration UI. However, what is missing is a common and service-independent way of describing and defining graphical constraints, which might be used by external providers automatically in order to adopt their services without any manual refinement and being able to change these easily as company marketing imagery changes.
- **Logical Integration:** integrating a component is a manual process today. Depending on the type of the component, it might be performed by adding a couple of lines into the HTML code of an e-Commerce website or it might require a major change in the core systems of the e-Commerce Vendor. The main problem is that the variety of protocols and formats is very high without allowing a standardized access to third party services.
- **Communication:** In real-world use cases, the data exchange is usually performed by either exchanging messages via an RPC interface such as Web Services or by using HTTP-Calls or Emails as the main way of exchanging information, depending on the type of services. Using RPC is normally applied when integrating components in a Web 2.0 manner. Email communication is a typical way for exchanging payment confirmations or order messages and sending it to the ISV.
- **Formats:** independently from the communication protocol, the data itself may be described in multiple formats. For exchanging order information, the most common formats found in real world examples are XML, Text, EDI and CSV.
- **Monitoring and Reliability:** monitoring services is important in order to ensure a continuous operation of systems. However, in most cases, a monitoring and alerting system for informing a vendor of system failures does not exist. In a typical scenario, vendors receive a message from customers whenever a problem with their web-shop appears.
- **Security and Trust:** security is a major issue in e-Commerce today. In the scenario that has been described, all critical information such as credit card details is usually only exchanged using secure channels such as HTTPS. However, much information is also exchanged via unsecured email today, including customer addresses and purchase details.
- **Problem tracking and analysis:** most e-Commerce vendors perform analysis on their customer behaviour allowing them to optimize their websites and to increase their sales. Although this is a rather easy task for a single website, it turns out to be more complex when involving third party components since user behaviour is normally not exchanged between all participating systems.

- **Orchestration:** integrating all third party components into one holistic system requires a careful orchestration of all involved components. It is required to describe the process in a structural way. Although business process description languages are available today, they are often not used for orchestrating the collaboration between the different services because they have focused on complex rather than more often utilized light scenarios
- **Core Platform:** managing users and orders and performing a basic integration of services is still a task that needs to be performed by the e-Commerce Vendor. What is required is therefore a core system that provides base functionality and allows him to quickly access all other services in one holistic management interface and to change or add new services.

As **architectural constraint**, in order to realize a scenario as described above, it is required to allow partners to exchange information in a distributed way. A suitable architecture therefore has to support the message exchange between remotely located applications and the possibility to add and replace services at any time.

The consortium partner TIE has provided the scenario. TIE's foundation is as a business-to-business (B2B) interoperability company and this currently encompasses business integration and supply-chain, master data management (MDM), content syndication, and digital commerce.

Enveloping these functionalities is the concept TIE Kinetix that can be summed up as the environment, which enables the complete interworking, through pick-and-mix selection, of all individual functionality in the TIE portfolio. The ambition of TIE is to perform all of this in a services orientated environment and with delivery as both as pure product and pure SaaS with of course (elements of) NEXOF forming some future basis.

The actual scenario itself has been derived from Stuart Campbell, TIE Chief Technical Officer, in conjunction with other TIE technical and commercial staff.

The top three reasons for selecting this scenario are:

- This scenario represents an important and real scenario for one of the project partners with an ambition to implement (parts of) NEXOF. In addition the scenario, or at least aspects of it, is relatively common and a foundation for many service and user organisation.
→ NEXOF can be assessed to see its fit with a common scenario and a very real scenario at partner level.
- The scenario is composed of many real world elements some of which are perceived as 'common services' today (e.g. payment) but for which the service is typically not exposed through common paradigms. These are then 'networked' into overall functionality composed of many functions from many applications.
→ This is a very typical and non invented scenario with a reliance on existing services which need to be able to fit into the architecture of

NEXOF – this support of existing application and the difficulty of adoption can be measured.

- The specific scenario can be largely product and SaaS based.
→ To ensure that product based services and SaaS based services are fully considered.

This scenario is already in existence and common in ICT but driven by bespoke interfaces. The challenge is to reinvent the scenario efficiently using NEXOF work.

The **inputs** of this scenario are:

- Specification of the existing applications and sub-services
- Specifications of the scenarios/processes
- Existing application environment

The **outputs** of this scenario are:

- Large scale and popular process
- SaaS and Product definition

This scenario is relevant for multiple eCommerce environments as well as existing users thinking of migrating to NEXOF.

3.13 Mobile office for an owner of a micro-enterprise (S13)

The current boost in the field of wireless and ubiquitous computing has great impact on many aspects of every day life and business. It becomes a formidable challenge for information systems to cope with rising complexity being caused by ubiquitous computing and increasing amount of electronic interactions between partners. The following example will demonstrate the increasing amount of complexity caused by human-human interaction, human-machine interaction and the pervasive and embedded nature of where such services may be required.

Imagine a one-person micro enterprise that runs a part-time vending business involving the placement and maintenance of vending machines in different locations, all supported by a national or international franchise network. The machines support a changing range of products and provide advanced functionality for performing specific activities.

The machines in this scenario are capable of storing and managing stock and price information as well as specific event related actions, such as saying 'Hello!' when a proximity sensor triggers. Some machines are even GSM enabled where this communication channel is used for providing update/diagnostic information straight to the servicing enterprise, and for authentication of payments performed via debit cards.

It is assumed that the owner of those machines is performing this job as a part-time business along with their normal day-to-day job. Since this is a part-time occupation the person still needs to be in touch with the main job which

revolves around remote IT and they are dealing with e-mails and requests from multiple partners.

These activities can be broken down into 5 main clusters:

- Servicing, maintaining and reconfiguration
- Managing the financial purchasing cycles
- Performing payments of commissions
- Synchronizing with other parties, dealing with emails
- Booking tickets, hotels, cars for business trips for the main activity whilst on the road

Those activities will now be described in more detail.

- **Servicing, maintaining and reconfiguring of these machines:** operations include the servicing and maintenance of these machines, and their reconfiguration. Each machine's embedded computer would be enabled with specific services through the hardware/software solution. This would also include that the machines would broadcast which services are available, for example stock information, cash information, diagnostic information etc. Obviously each machine type may provide a different range of services. On their service round the service operators would use a hand held devices which can communicate wirelessly with the machines identifying service possibilities and the wireless device would act accordingly. It is worth to mention that vending machines and handheld devices need some form of authorization such as electronic certificates. All the information concerning described actions and their prices should be immediately reported to the business owner who can be anywhere and this will allow tracking of the operator.

- **Managing the financial cycle for the purchasing of stock and machine partners:** the wireless device can also communicate to other devices via the GSM network to, for example, update in real time the local/franchisor databases or to resolve any stock disparities. Once validated, stock movement and status information is extracted to the wireless device, which then synchronizes with the spreadsheet application (for instance MS Excel) via GSM. Once updated this should then trigger both information to the franchisee and franchisor systems autonomously and also potential trigger any stock purchases (once volumes have been met) due to low stock as well as stimulating the payment process.

As services become enabled from the franchisor systems, for example an update to the firmware of machines, this can also become part of the process, for example in this instance the firmware update would be downloaded updated automatically when an appropriate machine is detected.

- **Performing payments of commissions:** the payment of commissions to the venue and the franchisor needs to be performed regularly. This produces a financial cycle for purchasing goods from partners, entering billing details into the local application of the business owner (e.g. self made database in MS Access) and the franchisors web based application

of stock and commission information. When the personal service of the business owner receives notifications about new payments, bills, etc. it triggers several actions and communicate with an appropriate web services for generating reports that lately become a part of the aggregated report sent to the tax office, for entering data into local database and for sending data to the franchisor system.

- **Dealing with emails, synchronizing agenda and schedule with partners:** being always online is vital for the business owner who communicates with various partners around the Globe. The services should be able to escalate the most important issues to the owner when needed and delegate tasks that can be performed without human close eye.
- **Booking tickets, hotels, and cars for business trips:** the owner travels a lot in order to keep the business up and running. They have to be flexible and fast reacting. So, it is crucial for her to be able to make reservations rapidly and without spending much time while busy with other tasks that require more intelligence. In an ideal situation, a service that knows their agenda should notify the booking service when a new entry is added or removed. Afterwards the booking service should ideally ask the preferences service about the owners favourite hotels, transport means and prices. Armed with this important information it can perform the travel arrangements automatically send a status notification to the owner.

Those activities can be handled in parallel whenever possible. It is physically impossible for one person to do all those things at the same time. It means, that it is highly desirable if some of the tasks can be delegated and automated, e.g. via web services.

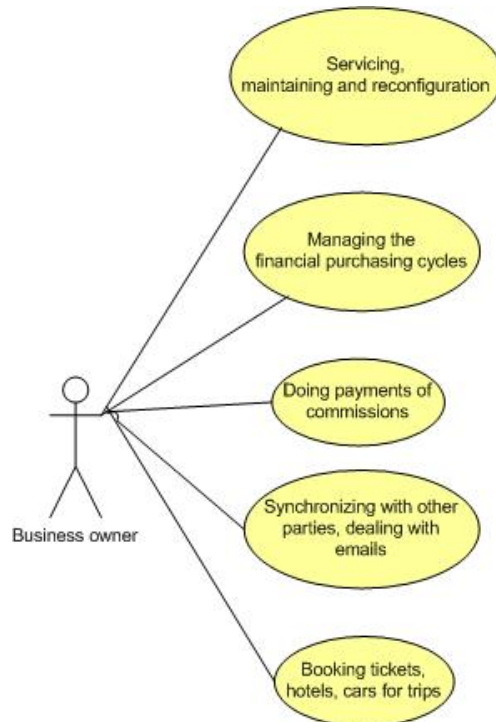


Figure 6: Use cases for the business owner

The **rationale** for this scenario is the following: users employ various applications and devices while performing multiple tasks. Simultaneously, user applications grow both in their number and in their complexity, in an attempt to meet users' requirements. These diverse types of systems support work of their users with particularly isolated domains and thus must be included in any SOA model. To improve service quality and make the user workflow seamless and coherent in distributed and multi-task environments it is necessary to establish reciprocal alliances among various systems.

For example, the current situation for our business owner seriously differs from the desired scenario. When performing the service rounds, the owner currently records the information on paper since systems are not in place to extract this information from the machines. Then they enter it back into the two databases: their own local one and the global one provided by franchisor. Currently all systems involved in those processes are not synchronized and manual input takes place on a monthly basis and is an onerous activity due to the manual processing.

They prepare quarterly commission forms and yearly accounts information. During the year they pay wages and have various interactions with the tax office through telephone or in person. They have to filter out emails for the vending business from their normal emails and react on them. All bookings for trips are performed manually and needs careful synchronization with other partners.

Unfortunately, present information systems typically are not compatible with each other, which cause problems for their collaboration. Although there are growing numbers of quick-witted devices and applications, they do not know

about each other and do not interoperate. Therefore, they do not support user activities as a sequence of a single workflow.

The scenario described in the previous section emphasizes a close cooperation among different systems based on the shared/interchanged information. The scenario suggested also is not geared around the internet itself as the primary mechanism for exchange, of course this is involved but it illustrates a scenario with a multitude of operating environments from near-embedded through to web applications on the internet which will be common for most non desk-jobs.

This **scenario applies to:**

- **Service consumers:** the group of the persons who can reap the benefits of implementation of such a scenario is very broad. It can include but not exceeds owners of micro business, as it was described in the scenario above.

In general, the audience consists of persons who have multiple business activities running at the same time at different locations, who are mobile and who need to be in touch with their clients, partners and employees all around a clock and all over the world.

- **Service providers:** based on the developed scenario the following groups of service providers can be identified: franchisor, stock suppliers, logistic companies, travel agencies, financial organizations like accounting firms and governmental institutions like tax office. In turn other customers can become providers when share their information with or expose personal services to our customer.

Nevertheless, any IT company can come up with a specialized service that can attract users due to the quality, price policy and/or uniqueness. Service-based solutions provide an opportunity for small and medium companies to compete with bigger players.

- **Service integrators/developers:** as a rule, a micro business owner knows exactly what they need and what the workflow is. Also they can react fast and accordantly to the changing situation. Whilst in many cases the end user might be non technical and thus require plug-and-play intuitive construction, often they are so close to their business they want to be involved in every aspects so some may require flexibility configurable by them. Thus, ideally the end user also should be able to discover required services by themselves and compose them into meaningful sequences.

Although, it is envisioned realistic direction would be to expect that the role of integrator can be played by any specialized IT company of any size, which can facilitate the user in this task. The company then should give an entry point for a monitoring service for a customer.

The most interesting **challenges** discovered in the above scenario are as following:

- **Dynamic discovery of pertinent services:** ability to add new services on the fly, such as firmware updates for our scenario, is crucial. Also to be

able to discover and interact with existing common small business applications, e.g. MS Excel, and make use of it. It means a user should be able to publish an interface to this application as a web service.

Another feature that is foreseen in connection with discovering is an availability of local repositories that can contain services available for a certain physical range. For instance, vending machines can have a local service repository and individuals with handheld devices would be able to dynamically discover them when close enough.

- **Semantic composition of web service chains:** it is a complicated and intelligence-led process of juggling the metadata web services have and combining them into meaningful workflows. For micro-enterprise the process should be user friendly, fast and cheap.
- **Security and single entry point for authentication:** the importance of security is hardly possible to underestimate. We want to tackle it from a slightly different perspective, which is not typical for modern service-oriented architectures. Portals provide, besides many other things, one feature particularly beloved by the users. It is a central point of entry and one time authorization for the whole set of applications. In the service-oriented world the role of portal can be played by a user's personal service that can represent its owner within a virtual realm.
- **Ambient intelligence implemented via services:** services can go into life not only through Internet. Currently many devices have embedded wireless interfaces from routers to vending machines. Protocols vary from IRQ to Wi-Fi, and to GSM. All those devices have to be enabled in the NEXOF-RA architecture.
- **Personalization:** the ability of services to take into account users preferences will enormously enforce power of the services and their usability. Personalization in distributed environments introduce many problems from searching for scattered pieces of information about user, dealing with privacy, resolving conflicts and trust in information providers. Architecture should facilitate in resolving those problems.

The **architecture** that can handle the situation described above should:

- Allow for systems of various categories to work together
- Allow (semi-) automatic services discovery and composition
- Allow wired and wireless devices working with different protocols co-exist and collaborate among each other
- Allow monitoring of services and
- Allow their reconfiguration on run time
- Allow end-users to expose their application as services in a friendly way
- Allow end-users to compose services in workflows in a friendly way

At the same time it cannot rely on broad bandwidth of the network connections and devices.

3.14 Safety at work in the construction sector (S14)

This scenario presents the person responsible for safety who is equipped with wireless enabled device moving around a work site in order to verify that all the safety requirements are fulfilled. This type of scenario will be adopted increasingly in the construction sectors as health and safety requirements ever increase.

The Safety Responsible (SR) person enters the work place with a Personal Digital Assistant (PDA). At this moment the system downloads from the remote server the information related to the current phase of construction. According to the status, necessary safety elements can vary.

The inspection is started by checking the identity and training skills of the workers. Safety Responsible identifies each worker by means of the Radio Frequency Identification (RFID) reader connected to the PDA and obviously each worker is equipped with an RFID tag. The system provides identity information about the workers including a photograph for visual identification. RFID tag also contains information about the necessary training that the Safety Responsible can match with the type of work they are performing on a construction site.

Any additional information about a worker can be retrieved from a remote server at the request of SR. Every worker must be equipped with the corresponding safety equipment. Some elements of this equipment depend on the tasks performing by workers. The system will show SR the required equipment to be worn for each worker. SR should visually check the completeness of required equipment. During this phase of the inspection SR interacts directly with the PDA using a pen or oral commands via a headset. The latter being necessary due the nature of typical construction sites and their 'outdoor' environment.

The second phase of the inspection corresponds to the identification of safety elements in the work place. In this phase, SR is again equipped with the PDA, a headset to record results and the mentioned previous inspection status. In addition they have positioning system to follow SR position, a camera for precision positioning and elements detection by image processing, an interaction wristband for the user interaction with the PDA (gestures recognition), and a Head Mounted Display (HMD) for visual information.

SR will walk around the work place; the system will automatically detect safety elements in the environment and ask SR for the localization in the correct place. SR can confirm the position or ask for help and the system will show in the HMD a map with the correct position of the element. Missing elements will be asked to visually inspect by SR.

Results and reports of the inspection will be recorded in electronic format (documents and audio notes) and transferred to the server. The interaction with the system will be performed using the wristband.

The **rationale** for this scenario is the following: safety at work is one of the main issues within the construction sector according to the number of accidents and their consequences. The construction environment is very different from other industrial environments, mainly because it is an ever changing, fast moving and also an uncontrolled environment.

Ambient intelligence based on the combination of distributed hardware and software solutions can bring a necessary level of control to such an environment through a real time monitoring of people, equipment and processes presented on a construction site.

This **scenario applies to:**

- **Service consumers:** construction workers and companies are the first among the consumers of such a solution. For workers any problem with equipment, unskilled partners or insufficient process can make a difference between life and death.
Construction companies that can take advantage of safety related to their resources help them reduce non-productive expenses. For them, it is important that safety measurements can be controlled quickly and in a non-intrusive way.
In general, it is expected that personnel that work in hostile and dangerous environments along with their employers can benefit from bringing this scenario to life.
- **Service providers:** service providers have to be certified and to be a subject of strict government regulations since a market cannot guarantee human safety.
- **Service integrators/developers:** any trusted and certified company can play a role of a developer or integrator.

The following **problems** are identified:

- Wireless communication between different devices and remote servers in a noisy and heterogeneous environment
- Real cooperation between humans and machines, e.g. when a user can act as an agent on a certain steps of a process and pass the result of activity back to the automated workflow
- Integration and controlling of scattered information from diverse sources
- Visualization of results for the user. Results can be represented by a huge amount of data of a different nature

The **architecture** that allows the above scenario to work should:

- Allow different protocols to co-exist within a platform including protocols for communication with humans, i.e. speech recognition, gestures, etc.
- Allow execution of human-based process steps in combination with automated steps and work flows
- Allow a representation of information as a service, e.g. RFID offers access to the information about workers, equipment, etc.

The scenario has been provided by the EU FP7 PRESTO project whose aim is to “support[s] Small and Medium-sized Enterprises Associations (SMEsAs) to define their research priorities [within the Construction sector] through a series of analyses, activities and support measures, all articulated with a sustainable dialogue with European Technology Platforms”. This scenario has then been elaborated by TIE in terms of the technology details in order to insert it in to NEXOF-RA. It should be noted that NEXOF (NESSI and also NESSI2010 Support Action) and PRESTO made contact mutually and this is an initial and good cooperation activity. The scenario was provided by Nikolaus Sennhauser European Project Executive of Targeting Innovation Ltd who is one of PRESTOS partners and is in liaison with other active members of PRESTO. It was made via a face-face interview where other elements of NEXOF were also discussed.

The top three reasons for selecting this scenario are:

- This scenario represents a scenario based on output from a user orientated ETP with partners which are also relatively low tech.
→ It can thus help determine how applicable NEXOF is to their of non-hi-tech cases
- The construction sector is perhaps perceived by many to be one of the least high-tech (bricks, mortar etc) but actually at the other end of the spectrum is extremely high tech (Advanced CAD/CAM, Intelligent building etc). Thus by exploring with this sector way to help them can help influence also their thinking on NEXOF/SOA and explore a pathway to adoption
→ Interviewed parties will be contacted re the NEXOF-RA roadmap / exploitation / dissemination activities to determine how to best shape material
- The specific scenario represents individuals of low technical awareness who have a focus on ‘just doing their job’ and also a low tech environment. So this relates to non functional aspects that maybe present – for example tools to support interaction need to be supported by the architecture or a robustness of the communication (e.g. service requests)
→ To derive pointers for user interaction and non functional aspects

These inputs of this scenario are:

- Specification RFID/Service interactions/formats
- Specifications of the scenarios/processes
- Lower skilled users without patience to configure technology (self configuring and adaptation)
- Differing resources (devices)

The **outputs** of this scenario are:

- Interworking of different devices
- Man-Machine interworking

This scenario is relevant for low tech environments – individuals and resource as well as multiple device environments.

3.15 e-Government online application submission service (S15)

One goal of e-Government is the state-wide provision of online services for the citizen, companies, government agencies and other authorities to carry out processes in a more effective and faster way. One of these services is the possibility to submit applications and receive replies online. These service portals have the advantage to be available 24 hours per day and that they do not require the citizen to go to a specific office – which is particularly useful in rural areas where the next office is not easily reachable.

This scenario describes the submission of applications to obtain subsidies from the province of Bolzano, Italy.

The steps carried out in the e-Government application submission are:

1. A citizen creates a login name and assigns a password using their social insurance number and identification card.
2. A citizen starts the compilation of a form either from scratch or decides to use a previous version (typically the application of the year before) to compile the form.
3. The submitted data is validated and combined with a PDF template to create the official document that the citizen is about to send to the public administration.
4. The created document is signed (if needed), encrypted and sent to the e-Government application.
5. If needed, the submission of the document is also accompanied by a payment through the e-Payment service.
6. The e-Government application receives the document in a virtual post-box and submits the document to the virtual post-box of the competent administrative office.
7. Together with the submission, the e-Government application performs the necessary signature and certificate checks and creates a transmission log, which is made available to the sender and receiver as a proof of delivery.
8. The competent administrative office can open the documents present in the virtual post-box anytime for further processing.
The submission of a document *from* the public administration *to* the citizen occurs in the same way.

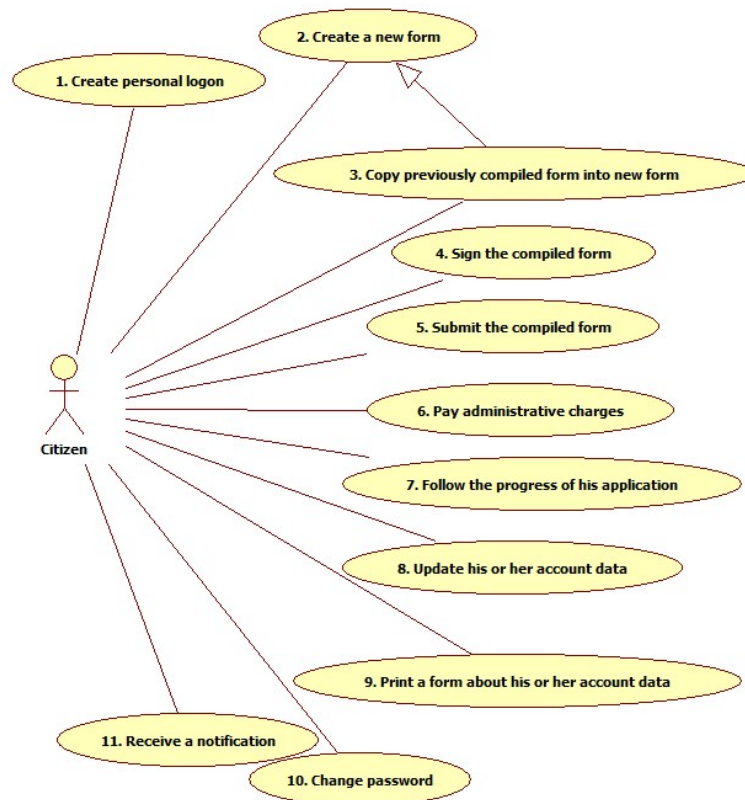


Figure 7: Use case diagram showing use cases from the point of view of the citizen

The use case diagram in **Errore. L'argomento parametro è sconosciuto.** above shows the possibilities for the citizen to interact with the system. The submission of a form is then shown in the sequence diagram in the next figure.

From the perspective of the administrative office, the use cases are shown in **Errore. L'argomento parametro è sconosciuto..**

The **rationale** of this scenario is the following: traditionally, the submission of applications to the administrative offices of the province of Bolzano follows the schema of **Errore. L'argomento parametro è sconosciuto..**

The problems with this approach were:

- Lengthy and complex process
- Missing transparency and user friendliness during the compilation of the application
- Office loses time with incomplete applications;
- Higher personnel costs due to the manual entry and check of the applications
- Missing advice for the citizen on how to fill out the application
- Offered service depends on the opening hours of the competent office
- Low collaboration between different administration offices

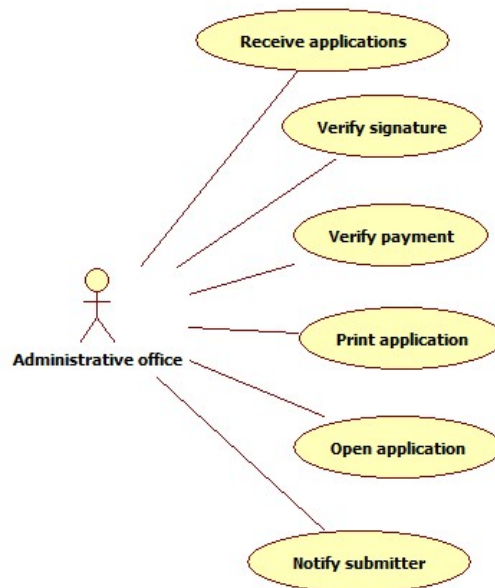


Figure 8: Use case diagram for the administrative office

The creation of a way to submit applications online had the goals of:

- Creation of a reliable, secure, and scalable IT infrastructure to handle online applications
- Secure communication channels between all stakeholders
- Integration of the existing IT systems
- Uniformly regulated privacy protection and security
- Ensuring the necessary acceptance by the citizen through intuitive user interfaces
- No installation of additional devices or special software

This **scenario applies to:**

- **Service consumers:** The service consumers are the citizens in this scenario. In this case the citizen wants to submit the online application from his home (as citizen-home) or from his working place (as citizen-work). To do this, they need to understand the required information that has to be attached to the application, wants to be able to submit and trace the progress of the application till its outcome.

The required performances depend on the expected number of users of the system – in our specific example the system has around 12.000 registered users with around 1.000 submissions per year. Similar systems in the province of Bolzano are used with the following frequency:

- Online application of student stipends: 4600 per year
- Online application of excellence scholarships: 900 per year
- Online application for child-care allowance: 35000 per year
- **Service providers:** the requested service is provided by the public administration.

- **Service integrators/developers:** in general, the service is developed by the public administration, which for specific aspects (e.g., e-Payment) will delegate parts of the development to third parties.

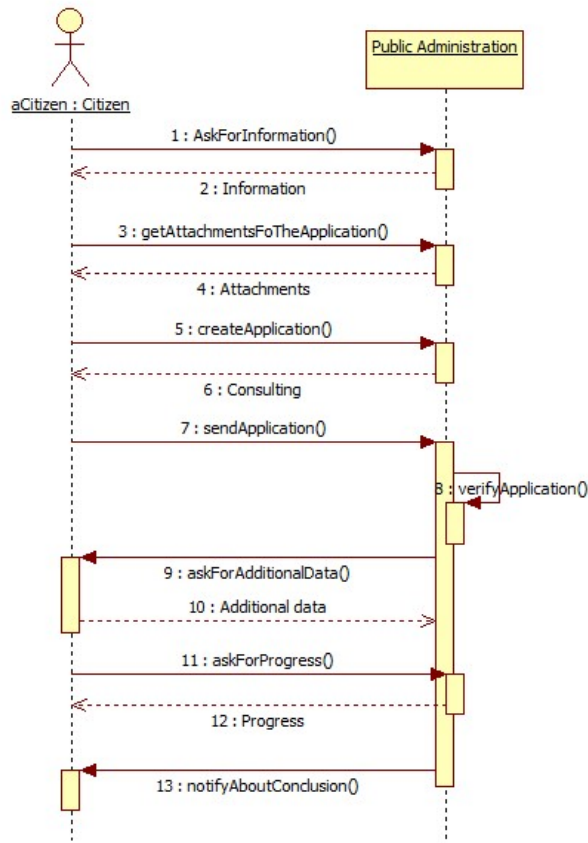


Figure 9: Traditional scenario when submitting an application to the public administration

From the perspective of privacy protection the following **challenges** were identified; the e-Government platform has to:

- Guarantee confidentiality through the use of encryption mechanisms
- Guarantee that the data is used only for the specified purpose, the data should be therefore encrypted only for the competent administrative office
- Guarantee the integrity and authenticity of individual-related data through electronic signatures
- Guarantee the non-reputability of the transfer of the data through a secure logging of the entrance of the message (like having a registered letter with the advice of receipt that proves that the letter has been sent and that it arrived)

A **possible architecture** of the here-described system consists of:

- A web server cluster to publish html pages and to guarantee a reliable access of the citizen to the offered service
- A database system to store the necessary data

The communication between different subsystems occurs through web services.

This scenario was produced in conjunction with the South Tyrol Municipal Consortium (the townships and district communities of the Province of Bolzano). The scenario was elicited from organization contacts and knowledge of their business.

This scenario was selected because it represents a typical application within the e-Government domain, which is of increasing importance within the European Union. E-Government aims to ease the contact between the citizen and the public administration. This is particularly important for the European Union with its 495 million citizens within 27 member states. NEXOF-RA can leverage the potential of e-Government since it facilitates the interoperability between institutions, organizations, provinces, regions, and countries.

This scenario is already in place within single public offices, but the missing interoperability between different public bodies on one side, and the proliferation of different communication protocols and solutions on the other side hinder the development of an implementation on a broad level.

The **inputs** of this scenario are:

- The user credentials of a citizen
- The type of document that has to be modified/submitted/signed
- The payment information of a citizen

The **outputs** of this scenario are:

- The archived, traceable interaction of a citizen with a public administration (including payment information)

This scenario is relevant for every public administration that intends to use service oriented technologies to implement e-Government solutions.

3.16 e-Government online fee visualization and payment Service (S16)

One goal of e-Government is the state-wide provision of online services for the citizen, companies, government agencies and other authorities to carry out processes in a more effective and faster way.

One of these services is to review and to pay taxes and fees using a Web interface. Within the Province of Bolzano an ongoing project is the implementation of a Web interface to allow citizens to pay the taxes for owning non mobile goods (houses, castles, etc.), fees for the trash collecting service, and the fees for consumed electricity.

Both applications are very similar, we will explain now how the citizen can review and pay taxes for non-mobile goods (houses). The second and third applications are built very similar just that instead of non mobile goods, trash cans are managed together with the fees for the emptying service or counters together with the fees for the consumed electricity.

This application includes the following use cases:

- 1) List all owned non-mobile goods: Lists all buildings owned by the citizen. The citizen can click on one building to see the past and due future payments.
- 2) List all past payments together with due date: lists all past payments and if their were made on time.
- 3) List all due payments together with due date: lists all future, open payments.
- 4) Print all past and due future payments together with due date: Prints past and future payments. It is possible to specify a date range.
- 5) Calculate amount to pay for payment that was not paid on due date: for overdue payments it is possible to calculate the due amount for the current day (the normal payment + the additional fees for delayed payment)
- 6) Payment of open amounts: it is possible to pay online using a credit card.
- 7) Print payment: prints the payment done in the current session for documentation purposes.

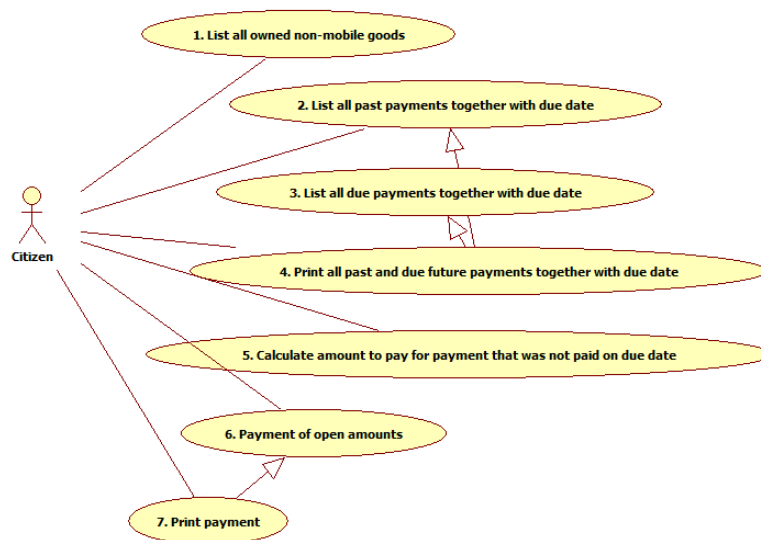


Figure 10: Use cases from the point of view of the citizen

The **rationale** of this scenario is the following: traditionally, the payment of the taxes for non mobile goods as well as the fees for services such as the trash collecting services had to be done going to the public offices from 8am to 12am (to get the necessary forms and to calculate the due amount in case of overdue payment) and then by paying at any post office the due amount using the specific forms.

This process is cumbersome for everyone that is working during regular office hours. The possibility to pay taxes and fees online makes it easier for the citizen to pay and requires fewer personnel by the public administration.

This **scenario applies to:**

- **Service consumers:** The service consumers are the citizens in this scenario. In this case the citizen wants to pay taxes and fees from his home (as citizen-home) or from his working place (as citizen-work). The response time should not cause timeouts when using the application. The user accesses the service through the web site provided by the public administration.
- **Service providers:** the requested service is provided by the public administration.
- **Service integrators/developers:** in general, the service is developed by the public administration, which for specific aspects (e.g., e-Payment) will delegate parts of the development to third parties.

The screenshot shows the 'ICI online' web application interface. At the top, there are logos for 'Consorzio dei Comuni della Provincia di Bolzano Coop.s.r.l.', 'ICI online', and 'AltoAdige'. The user is identified as 'Sara Tumati' on 'lunedì 16 giugno 2008'. The main content area displays 'Imposta Comunale sugli immobili' for the year 2008. It lists the user's property in 'APPIANO SULLA STRADA DEL VINO'. Below this, there is a table titled 'Scadenze e versamenti già effettuati 2008 APPIANO SULLA STRADA DEL VINO' with columns for 'Scadenza', 'Tipo', 'Dovuto', 'Versato', 'Rimborso', 'Ammanco', and 'Versamento'. The table shows two entries: one for '16.06.2008' (Acconto) and one for '16.12.2008' (Saldo). The total amount due is listed as '€ 1.149,00'. At the bottom, there is another table titled 'Elenco degli immobili 2008 APPIANO SULLA STRADA DEL VINO' with columns for 'Immobile', 'Indirizzo', 'Tipo Particella', 'Pertinenza', 'Perc. di possesso', 'Detrazione spettante', and 'Imposta dovuta'. One property is listed with address 'S. MICHELE VIA DELLE' and a tax amount of '€ 341,64'.

Figure 11: Main page, showing list of buildings owned by the citizen and his past payments

The problems and challenges include:

- **Access of information stored in different databases:** the data about the consumed electricity is stored on a different location, by a different organization that e.g., the data about the taxes for non-mobile goods.
- **Different public offices:** the payment of the three different taxes/fees have to be done on different bank accounts, for the moment a complete integration so that the citizen gets one bill with all what he or she has to pay (and to pay all in one step) is not possible.
- **Privacy law constraints:** privacy law imposes that if one public office wants to use a service like 'getAllOverdueTaxes(citizenID)' or 'getAllOwnedHouses(citizenID)' the citizen has to agree that his data is shared. This means that the use of a service is bound to a legal requirement, the permission by the citizen.

A **possible architecture** of the here-described system consists of:

- A web server cluster to publish html pages and to guarantee a reliable access of the citizen to the offered service;
- A database system to store the necessary data

The communication between different subsystems occurs through web services.

Immobile	Indirizzo	Tipo Particella	Pertinenza	Perc. di possesso	Detrazione spettante	Imposta dovuta																																				
1894	S. MICHELE VIA DELLE	Fabbricato	-	100 %	€ 0,00	€ 341,64																																				
<p>Periodo: 01.01.2008 - 01.01.2009</p> <p>Dati catastali</p> <table border="1"> <tr> <td>Comune catastale</td> <td>A332</td> <td>Valore dell'immobile ai fini ICI</td> <td>€ 75.919,20</td> </tr> <tr> <td>Numero e denominatore particella</td> <td>GLOBAL</td> <td>Perc. di possesso</td> <td>100 %</td> </tr> <tr> <td>Subalterno</td> <td>3</td> <td>Aliquota</td> <td>4,5 %</td> </tr> <tr> <td>Foglio Catastale</td> <td>951</td> <td>Codice detrazione</td> <td>/</td> </tr> <tr> <td>Consistenza</td> <td>7 vani</td> <td>Perc. detrazione spettante</td> <td>0 %</td> </tr> <tr> <td>Categoria</td> <td>A</td> <td>Detrazione spettante</td> <td>€ 0,00</td> </tr> <tr> <td>Classe</td> <td>1</td> <td>Imposta dovuta</td> <td>€ 341,64</td> </tr> <tr> <td>Rendita</td> <td>€ 723,04</td> <td></td> <td></td> </tr> <tr> <td>Immobile storico</td> <td>No</td> <td></td> <td></td> </tr> </table> <p>Calcolo imposta su questo immobile</p>							Comune catastale	A332	Valore dell'immobile ai fini ICI	€ 75.919,20	Numero e denominatore particella	GLOBAL	Perc. di possesso	100 %	Subalterno	3	Aliquota	4,5 %	Foglio Catastale	951	Codice detrazione	/	Consistenza	7 vani	Perc. detrazione spettante	0 %	Categoria	A	Detrazione spettante	€ 0,00	Classe	1	Imposta dovuta	€ 341,64	Rendita	€ 723,04			Immobile storico	No		
Comune catastale	A332	Valore dell'immobile ai fini ICI	€ 75.919,20																																							
Numero e denominatore particella	GLOBAL	Perc. di possesso	100 %																																							
Subalterno	3	Aliquota	4,5 %																																							
Foglio Catastale	951	Codice detrazione	/																																							
Consistenza	7 vani	Perc. detrazione spettante	0 %																																							
Categoria	A	Detrazione spettante	€ 0,00																																							
Classe	1	Imposta dovuta	€ 341,64																																							
Rendita	€ 723,04																																									
Immobile storico	No																																									
1895		Fabbricato	-	100 %	€ 0,00	€ 344,07																																				
1879		Fabbricato	-	100 %	€ 0,00	€ 258,06																																				
1893		Fabbricato	-	100 %	€ 0,00	€ 29,87																																				
1896		Fabbricato	-	100 %	€ 0,00	€ 39,83																																				
1897		Fabbricato	-	100 %	€ 0,00	€ 70,87																																				

Figure 12: Second part of main page, showing the summary

Data	Motivazione	In
17.06.2008	Ritardato pagamento entro 30 giorni dalla scadenza	Acconto

Dettaglio calcolo		
Importo dovuto		€ 574,00
Sanzione amministrativa	7 %	€ 40,18
Interessi	Periodo: 16.06.2008 - 17.06.2008 Interessi 8% (giorni 1)	€ 0,13
Totale		€ 614,31

Eseguito da	Codice fiscale	Ubicazione Immobili
Demo User		APPIANO SULLA STRADA DEL VINO

Importi riferiti a:	
Terroni agricoli	€ 0,00
Aree fabbricabili	€ 0,00
Abitazione principale	€ 0,00
Altri fabbricati	€ 574,37
Detrazione spettante	€ 0,00

Figure 13: Shows the calculation of the new payment after the past due date was missed

This scenario was produced in conjunction with the South Tyrol Municipal Consortium (the townships and district communities of the Province of Bolzano). The scenario was elicited from organization contacts and knowledge of their business.

This scenario was selected because it represents a typical application within the e-Government domain, which is of increasing importance within the European Union. E-Government aims to ease the contact between the citizen and the public administration. This is particularly important for the European Union with its 495 million citizens within 27 member states. NEXOF-RA can leverage the potential of e-Government since it facilitates the interoperability between institutions, organizations, provinces, regions, and countries.

This scenario is already in place within single public offices, but the missing interoperability between different public bodies on one side, and the proliferation of different communication protocols and solutions on the other side hinder the development of an implementation on a broad level.

The inputs of this scenario are:

- The user credentials of a citizen
- The payment information of a citizen
- The tax information of a citizen

The outputs of this scenario are:

- The archived, traceable payment of taxes by a citizen

This scenario is relevant for every public administration that intends to use service oriented technologies to implement e-Government solutions.

3.17 Assisted Industrial Maintenance (using 3D virtual environments) (S17)

An industrial plant or its devices are maintained on site by a service technician with support from an engineer not on site. The engineer has expertise knowledge but is located far away from the plant in question. The engineer is observing the work of the technician and gives instructions to the technician.

The engineer has access to a 3D virtual world, either via an ordinary display rendering the 3D virtual world and an avatar or via a dedicated virtual reality display. The engineer can also interact with the 3D virtual world, either via an ordinary keyboard and mouse or via a dedicated device or motion sensor mechanisms.

In the 3D virtual world, the engineer sees a so called virtual shadow the device. A virtual shadow is a representation of the device that depicts the relevant features and a real-time status of the device. The engineer can use voice as well as 3D objects to guide the technician through the repair process.

The repair technician is wearing a HUD (Head up display), which provides the possibility to show 3D overlay pictures. He also sees the shadow of the device. He can interact with the device either in real world or directly via the 3D virtual world.

The maintenance may take place on a regular basis or may be the result of an emergency case. In both cases, down times are costly (in many respects)

and it can be especially costly in an emergency case. Hence, the described 3D visualization and the corresponding data transmission have very high dependability requirements. In particular, 3D virtual environments are data processing intensive applications. They must be assigned enough computational and communication resources to allow operating. In the emergency case, they must be given priority over other computations and communications that take place at the same time on a regular basis but have not a critical dependability.

The **rationale** for this scenario is the following: today it is not always possible to gain access to expert knowledge of technical devices, especially when such devices are installed in very remote locations of the world. On the other side it is much easier to have technical repair personnel on site, which has reasonable basic knowledge, but lacks the necessary details. By allowing the repair staff gaining access to expert knowledge and to not only receive audio but also visual information, 3D repair instructions can significantly improve the speed and quality of repairs. Wide spread availability of 3D technology allows us to utilize its advantages like rich communication via 3D content.

The scenario is based on remote maintenance but takes it a step further. It overcomes the problem that remote maintenance is not applied when it comes to very expensive devices such as a huge gas turbine. Here the risk is simply too high.

Maintenance is not the only scenario benefitting from a distributed integration of 3D and reality. Similarly, plant development and product life cycle management can benefit.

This **scenario applies to:**

- **Service consumers:** the user will be the technician using a device like a HUD (Head Up Display).
- **Service providers:** the services will either be integrated into or provide interfaces to 3D worlds like Second Life.
- **Service integrators/developers:** services may be provided/implemented by the same organization that constructed or maintains the technical device. There might already be customized 3D objects, which can be used as give instructions to the technicians.

The **problems and challenges** include:

- The post Second Life era has spawned several approaches of 3D virtual worlds. Current research includes for instance interoperability between different 3D platforms. This is also the challenge for NEXOF to allow hosting and integration of these kinds of applications.
- Providing virtual shadows for devices is another research topic. Devices and controls need to be enhanced to provide the necessary data for virtual shadows, moreover, without affecting their dependable functioning. Standards need to be developed that allow integrating enhanced devices and controls into 3D virtual environments. Similarly, devices and controls need to be enhanced to process input coming from 3D virtual environments.

The **architecture** that allows the above scenario to has to consider:

- Heterogeneous platforms that support high dependability, in particular for 3D virtual environments
- Device integration
- Device shadow standards

4 REQUIREMENTS COLLECTED BY PARTNERS

The requirements were identified analyzing the scenarios by the partners of this workpackage using the methodology described in the section “Approach”.

The parts of the template for requirements (used by the partners to submit requirements derived from the scenarios) are the following³:

- Short name:** a short name for this requirement.
- Requirement type:** one of the following
 - Functional requirement: what the product has to do or what processing actions it is to take
 - Quality attributes: properties that the functions must have, such as performance and usability
 - Project constraints: restrictions due to the budget or the available time
 - Design constraints: restrictions on how the reference architecture must be designed
 - External constraints: restrictions because of business or law restrictions
- Related to:** the scenario ID to which this requirement is related
- Description:** the intention of the requirement
- Rationale:** a justification of the requirement
- Domain:** the domain and sub-domain of this requirement.
- Target:**
 - Domain-independent requirement
 - Cross-domain requirement
 - Domain-specific requirement
- Originator type**
 - Service consumer
 - Service provider
 - Service integrator/developer
- Fit criterion:** a measurement of the requirement such that it is possible to test if the solution matches the original requirement.
- Comments:** additional comments if needed.
- Conflicts:** requirements that cannot be implemented if this one is.
- Supporting materials:** a pointer to documents that illustrate and explain this requirement.
- Priority of accomplishment within the NEXOF-RA project**
 - Must have
 - Should have
 - Could have

The following requirements were collected by the partners and will be described below:

³ The “Template for the description of requirements” uses material from the Volere Requirements Specification Template, copyright (c) 1995-2006 the Atlantic Systems Guild Limited [5].

- **SLA processing (R1):** describes the requirements to define and manage consumer provider interaction in a service-oriented ecosystem.
- **Uniform service representation (R2):** discusses properties necessary to allow potential consumers of services be able to discover, compare and choose between services.
- **Service discovery mechanisms (R3):** discusses mechanisms need to accommodate searches for services based on both functional and non-functional properties
- **Decentralised architecture (R4):** discusses the requirement of a decentralized architecture.
- **Service description (R5):** discusses the need for a consistent approach among service providers to the description of functional and non-functional characteristics of services
- **Service deployment (R6):** discusses the requirement to introduce new services without reference to a central authority.
- **Service decommissioning (R7):** discusses the need for a service provider to cease offering a particular service.
- **Interoperability and flexible communication requirement (R8):** discusses the need that information and management services are able to communicate with existing services and with each other.
- **Federated identity management (R9):** discusses the need of trust and identity management when information is shared across multiple organisations.
- **Location based routing (R10):** discusses the routing of a service request depending on the location of the end user.
- **Services integration by semantic mash-up (R11):** discusses the integration of services based on a mash-up server to manage integration, setting, choice and coordination of available services resting on environment ontologies.
- **Harmonization of several heterogeneous information sources (R12):** discusses the harmonization of several heterogeneous information sources by means of semantic technologies.
- **Unified communication (R13):** discusses collaboration and communication services to allow interaction among heterogeneous services and sources.
- **Integration of an application with legacy applications, ERP, etc. critical and foremost for the client (R14):** discusses the integration of services with legacy applications.
- **Adaptive deployment (R15):** discusses the ability to deploy the application according to the model of software deployment of the client.
- **Workflow management and integration (R16):** discusses the integration of new services in the business processes of the client.
- **Aided configuration (R17):** discusses the capability of the architecture and an application, which is based on it, to support the configuration task during the deployment phase and during maintenance.

- **Modelling capabilities (R18):** discusses the need to allow modelling the company whose business will benefit of the new application/platform based on the architecture itself
- **Technical interoperability (R19):** discusses the need of the interoperability between services regarding distributed workflows and individual service interaction.
- **Device integration / vertical integration (R20):** discusses the requirement of vertical integration of devices in service-oriented architectures.
- **Distributed workflow (R21):** discusses the integration of workflows from different stakeholders.
- **Stateful, device adaptive service transfer (R22):** discusses the need of statefulness during service interaction.
- **Adaptability (R23):** discusses the adaptability of deployed services considering autonomously reacting systems.
- **Rapid reconfiguration (R24):** discusses the reconfiguration of a system in case of an emergency by selecting a new strategy and the propagation of the new rules to the system.
- **Integrity (self-diagnosing and self-healing) (R25):** discusses the need that no one can override the decisions of a critical system (such as a traffic management system) from outside by either manipulating the sensors and the control devices or by introducing harmful strategies.
- **Dependability for device integration (R26):** discusses the need of performance, reliability and availability when integrating embedded devices with services.
- **Compliance to privacy, and security policies (R27):** discusses compliance to privacy, and security requirements of services.
- **Collaborative business process acquisition, modelling and effective management (R28):** discusses the capacity to acquire/form, model and manage effectively and efficiently collaborative business processes for what concerns CDM applications.
- **Distributed architecture (R29):** discusses the exchange of information in a distributed way.
- **Integration of services (R30):** discusses the integration of services considering the user interface.
- **Monitoring and reliability (R31):** describes the need of monitoring in order to ensure a continuous operation of systems.
- **Orchestration (R32):** describes the need of a careful orchestration of all involved components.
- **Trust and confidence (R33):** describes the need of trust and confidence in applications such as eCommerce Service discovery (R34): describes the need of service discovery.
- **Information as a service (R35):** describes the need of providing information in a coherent and trusted way.
- **Execution of human-based process steps (R36):** describes the integration of services with human interactions and manual decisions.

- **Information integration (R37):** describes the requirement for a service architecture to provide support for information integration.
- **Distributed transaction support (R38):** describes the need to support distributed transactions.
- **Single sign on (R39):** describes the need to support single sign on.
- **Non-repudiability of data transfer (R39):** describes the need to prove that the recipient really received the message and that the sender really sent the message
- **Cross-certification (R40):** describes the need arising from interfacing multiple organisations having their own but different PKI certification policy.
- **Resilience & Continuity of service (R41):** describes the requirement of resiliency, i.e. the ability to avoid, minimize, withstand, and recover from the effects of adversity.

Fields that were not filled out are omitted in the requirement descriptions. The priorities of the requirements are summarized in section 6.

Although the requirements have been extracted analyzing the scenarios, a single requirement can also be valid in several scenarios. Which requirements apply to which scenario is shown in the table below:

Table 12: Requirement applicability

Requirement /Scenario	<p>Service procurement (S1)</p> <p>Service life cycle support (S2)</p> <p>Management services for grid and service platforms (S3)</p> <p>PhiMas: personal health information monitor and alert service (S4)</p> <p>Collaborative e-learning scenario (S5)</p> <p>e-Health: complex diagnostic workflow (S7)</p> <p>Deployment and configuration of a generic platform (S8)</p> <p>e-Health: assisted living (S8)</p> <p>Traffic management: large scale emergency handling (S9)</p> <p>Crisis management system of systems (S10)</p> <p>Effective and efficient collaborative decision making (S11)</p> <p>e-Commerce information sharing (S12)</p> <p>Mobile office for an owner of a micro-enterprise (S13)</p> <p>Safety at work in the construction sector (S14)</p> <p>e-Government online application submission service (S15)</p> <p>e-Government online fee visualization and payment service (S16)</p> <p>Assisted Industrial Maintenance (S17)</p>
SLA processing (R1)	●
Uniform service representation (R2)	●
Service discovery mechanisms (R3)	●
Decentralised architecture (R4)	●
Service description (R5)	●
Service deployment (R6)	●
Service decommissioning (R7)	●
Int. and fl. comm. standards (R8)	●

Federated (temporal) identity management (R9)	●	● ●
Location based routing (R10)	●	
Services integration by semantic mash-up (R11)	●	
Harmonization of het. inf. sources (R12)	●	
Unified communication (R13)	●	
Integration with legacy applications (R14)	●	
Adaptive deployment (R15)	●	
Workflow management and integration (R16)	●	
Aided configuration (R17)	●	
Modelling capabilities (R18)	●	
Technical interoperability (R19)	●	
Device integration / vertical integration (R20)	●	
Distributed workflow (R21)	●	
Stateful, device adaptive service transfer (R22)	●	
Adaptability (R23)		●
Rapid reconfiguration (R24)		●
Integrity (self-diagnosing and self-healing) (R25)		●
Dependability for device integration	●	

(R26)

Compliance to privacy,
and security policies

• •

(R27)

Collaborative bus.
proc. acq., mod. and
eff. M. (R28)

•

Distributed architecture
(R29)

•

Integration of services
(R30)

•

Monitoring and
reliability (R31)

•

Orchestration (R32)

•

Trust and confidence
(R33)

•

Service discovery
(R34)

•

Information as a
service (R35)

•

Execution of human-
based process steps
(R36)

•

Information integration
(R37)

•

Distributed transaction
support (R38)

•

Non repudiability of
data transfer (R39)

•

Cross-certification
(R40)

•

•

•

•

•

•

•

•

•

Resilience & Continuity
of service (R41)

•

3D virtual env. services
supp. and int. (R42)

•

4.1 SLA processing (R1)

The use of service level agreements (SLAs) to define and manage consumer provider interaction in a service-oriented ecosystem raises a number of requirements. There needs to be a common representation, a negotiation

protocol and a strategy for monitoring to ensure compliance. The common representation allows effective comparisons between offerings from different suppliers, the negotiation protocol allows modification of SLA terms and monitoring provides the sound financial basis for the commercial transaction.

The **rationale** for this requirement is that the expected benefit of the service-based ecosystem relies on commercial provision and consumption of services; SLAs set the context for these commercial relationships and manages obligations for both parties.

Related to: Scenario S1

Requirement type: Functional

Target: Domain independent requirement; will be required for all services, even if the SLA is only implicit.

Originator type: Service Provider

4.2 Uniform service representation (R2)

In a flexible dynamic service based ecosystem it is essential that potential consumers of services be able to discover, compare and choose between services offered by competing service providers. A uniform service representation will assist in the formulation of customer queries.

The **rationale** for this requirement is the following: the expected benefit of the service-based ecosystem relies on commercial provision and consumption of services. The services offered will range in complexity from business processes down to simple provision of hardware resources, but their exact nature will be determined by commercial requirements, not any arbitrary layer structure imposed on the architecture. For consumers to be able to search for discover, choose and use services there needs to be some commonality in the representation to facilitate automation of many of the routine tasks.

Related to: Scenario S1

Requirement type: Functional requirement

Target: Domain independent requirement

Originator type: Service consumers, service providers

4.3 Service discovery mechanisms (R3)

There are a number of requirements for service discovery mechanisms in flexible service oriented architectures. The mechanisms need to accommodate searches for services based on both functional and non-functional properties. The mechanisms should also support selective discovery and access mechanisms to support the fact that many companies will have internal service directories that are not accessible to the public and that use of other services may have access restrictions based on other criteria, such as compliance with varying national legislations. An end user must be able to specify any combination of functional and non-functional

parameters to identify services of interest, and the services returned by any search must be available to the consumer should they decide to use them.

The **rationale** for this requirement is the following: the expected benefit of the service-based ecosystem relies on commercial provision and consumption of services. Customers that want to purchase services have to discover and select them. Services may be advertised both publicly and privately so selective visibility and use of services will be required.

Related to: Scenario S1

Requirement type: Functional requirement

Target: Domain independent requirement

Originator type: Service provider, service consumer

4.4 Decentralised architecture (R4)

In future large-scale distributed service architecture there will be large numbers of repositories and data sources covering topics such as SLAs service advertisements etc. These will be split between public and private offerings with different access policies depending on user characteristics. Additionally there may be performance constraints associated with attempts to provide any form of centralised directory structure for service discovery

The **rationale** for this requirement is the following: in a future large scale service ecosystem there will be a large number of suppliers and customers many of whom will have mixtures of public and private service offerings. The architecture will need to support federation of public directories while at the same time supporting access controlled private directories for private services.

Related to: Scenario S1

Requirement type: Functional requirement

Target: Domain independent requirement

4.5 Service description (R5)

There needs to be a consistent approach among service providers to the description of functional and non-functional characteristics of services.

The **rationale** for this requirement is the following: it is fundamental to an effective open service market that consumers of services can compare offerings from different providers. This must include both what the service can do for them and what quality of experience they can expect if they choose to use the service.

Related to: Scenario S2

Requirement type: Functional requirement

Domain: Service discovery

Target: Domain-independent

Originator type: Service consumer, -provider, -integrator

Fit criterion: If this requirement is fulfilled, automated matching of customer requirements to provider offerings will be possible.

Comments: A single universal ontology is not expected to be feasible or required. Agreement on basic principles, terminology and representation will improve the dependability of semantic or other publish/search tools.

4.6 Service deployment (R6)

It must be possible to introduce new services without reference to a central authority.

Management systems must be extensible (e.g. in terms of new monitored parameters and control behaviour) so that new services can be introduced dynamically.

The **rationale** for this requirement is the following: there must be support for a provider to deploy a new service into an existing service ecosystem autonomously if the system is to be open. Services must be able to implement manageability features with high flexibility. Generic management support services must be sufficiently open to handle this.

Related to: Scenario S2

Requirement type: Functional requirement

Domain: Service deployment

Target: Cross-domain requirement

Originator type: Service provider

Fit criterion: If this requirement is fulfilled, service providers will be able to introduce new services whenever they want and will be able to use management functionality provided by the NESSI Open Framework.

4.7 Service decommissioning (R7)

It must be possible for a service provider to cease offering a particular service. The platform should have facilities to allow this to be performed with minimal disruption to those with direct and indirect dependencies on the service. These facilities could include ensuring that service registries are up-to-date, providing notification mechanisms to users of withdrawn services, providing support for tracing service dependencies.

The **rationale** for this requirement is the following: when a service becomes obsolete or no longer in the interests of the provider to offer, there will be a need to withdraw and decommission it. It should be possible to minimise the impact on those with a dependency on the service.

Related to: Scenario S2

Requirement type: Functional requirement

Domain: Service decommissioning

Target: Cross-domain

Originator type: Service provider, Service consumer

Fit criterion: If this requirement is fulfilled, it will be possible for a service provider to withdraw a service gracefully. Service consumers will have increased confidence that critical services will not disappear without warning.

Comments: Well-behaved service providers will take appropriate actions to mitigate the effects on their customers. However, this may not be possible in all cases (e.g. if a provider goes out of business). In addition, while a provider may know his direct customers, some of these may incorporate the service as a component of a composite. Dependencies can rapidly become complex and hard to track. Architectural support for service decommissioning is therefore required.

4.8 Interoperability and flexible communication standards (R8)

The information and management services must be able to communicate with existing services and with each other. Since the architecture has to deal with a heterogeneous dynamic system (services come and go) the interoperability requirement includes semantics and use of ontologies.

The **rationale** for this requirement is the following: it is expected that the management services will be developed by multiple organisations, serving temporary and local parts of the service grid. This should be taken into account from the beginning of the design of the interoperability aspects.

Related to: Scenario S3

Requirement type: Functional requirement

Target: Cross-domain

Originator type: End user, Service provider

4.9 Federated (temporal) identity management (R9)

Trust and identity management is crucial when information is shared amongst multiple organisations.

Furthermore, since organisations might be involved with management and maintenance tasks only for a particular amount of time, the identity management should be able to deal with temporality. Another issue is scalability, the service networks are distributed. Federated Identity is a prime component of any service network architecture.

Possible solutions might be based on proxy-based certificate solutions, as developed within the fields of grid-middleware.

The **rationale** for this requirement is the following: collaboration implies to deal with trust. This holds for scientific service environments, but even more for industrial service environments.

Related to: Scenario S3, Scenario 10, Scenario 11

Requirement type: Functional requirement

Target: Cross-domain

Originator type: End user, Service provider

4.10 Location based routing (R10)

The routing of a service request depends on the location of the end user. GPS data in combination with service location information must lead to an optimal routing of this request.

For the **rationale** for this requirement see the rationale in the scenario description.

Related to: Scenario S4

Requirement type: Functional requirement

Target: Cross-domain

Originator type: End user, Service provider

Fit criterion: The end user should have his request routed to most locally perceived service (human or machine)

4.11 Services integration by semantic mash-up (R11)

The integration of “environment services” (i.e. forum, blog, annotation, videoconference, etc.) could be provided and supported by means of ontologies. The “semantic mash-up” satisfies this need. In other words, a mash-up server should manage integration, setting, choice and coordination of available services resting on environment ontologies.

These features involve other sub-requirements, such as:

- The mash-up server should hold the role of workflow engine at macro-activity level
- The mash-up server should perform the contextualization (and customization) of services during each phase of work
- The mash-up server needs interaction with an ontology server
- Available services should be effectively metadated (not only the contents, but the services themselves, too).

Currently, mash up approach is commonly adopted for presentation of several services provided by heterogeneous source; what should be pointed out is deep integration among them based on semantic approach, in order to coordinate and synchronize each other, in case sharing and linking (e.g. RSS) their data.

The **rationale** for this requirement is the following: mash up approaches is going to be followed in enterprise environment, even more in critical application, too. In addition, enterprise environments require more and more

integration of several heterogeneous sources, mainly by semantic approaches.

Related to: Scenario S5

Requirement type: Functional requirement

Domain: Any domain that expects user interaction with services (collaboration, communications, etc.), for instance that provided by means of gadgets.

Target: Domain-independent

Originator type: Service integrator/developer

Fit criterion: A mash up integration service should really combine simplicity, flexibility, coordination and deep integration among services. In addition, a semantic mash-up must go as far as to connect services at a semantic level.

Supporting materials: A. Jhingran, "Enterprise Information Mashups: Integrating Information, Simply", Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB '06), Seoul, Korea, Sep. 2006, pp.3-4.

4.12 Harmonization of several heterogeneous information sources (R12)

In an enterprise context, the opportunity of retrieving or organizing information regardless of their own structure, their own semantics, in short regardless of their specific source, has a strategic importance. The harmonization of several heterogeneous information sources by means of semantic technologies seems to be the most suitable strategy to face the increasing of different typologies of sources.

Related to: Scenario S5

Requirement type: Functional requirement

Domain: Any domain that involves several source types of information, collaboration and communication services, as well as multimedia providers.

Target: Domain-independent

Originator type: Service integrator/developer

Fit criterion: The harmonization of several heterogeneous information sources (i.e. by means of semantic technologies) results in friendly searching, classifying, analyzing, etc. of information themselves.

4.13 Unified communication (R13)

Collaboration and communication services should be provided keeping interaction among heterogeneous services and sources (structured or not), allowing extraction and classification of information for several needs (i.e. new collaboration phase, abstract of discussions for report or analysis). The integration leads to the so-called "unified communication".

The **rationale** for this requirement is the following: in enterprise environments as well as in other social contexts, improving collaboration and communication (not only for training purposes) should drive to immediate and effective collaboration work. In addition, extraction and classification of contents allows information otherwise lost to be preserved.

Related to: Scenario S5

Requirement type: Functional requirement

Domain: Any domain that involves interaction among more than one user

Target: Cross-domain

Originator type: Service consumer, Service integrator/developer

4.14 Integration of an application with legacy applications, ERP, etc. critical and foremost for the client. (R14)

In an enterprise, as well as in a university context, the deployment of a new application or platform could require a deep integration with other systems or application. In case the previously existing applications are prearranged to interactions, this task could be sensible (i.e. SOA), otherwise (i.e. legacy applications) the integration could require a lot of effort. In order to minimize the impact for this task, some additional ability of integration could be required. For instance, a simple, but often not performing way, to interact with closed systems is via text files.

Related to: Scenario S6

Requirement type: Functional requirement

Domain: Interoperability

Target: Domain-independent

Originator type: Service integrator/developer

Fit criterion: At first, the capability of interaction with already running applications could be considered the mandatory target. Afterwards the easiness in the setup phase and the performances during execution phase measure the effectiveness of solution in order to satisfy the requirement.

4.15 Adaptive deployment (R15)

Each client has one or more than one models of software deployment and, usually, the client requires that new incoming applications should be deployed according this model. So, the ability to deploy the application according to the model of software deployment of the client could be required to the application.

Of course, this ability implies to face some fundamental topics in different manners depending on the deployment adopted. For instance, the security and the availability have to be guaranteed in any case, but these features will be implemented in different way in case of “Software as a Service”, rather than in case of “Stand Alone” deployment on the intranet of the client.

On the other hand, depending on the architecture adopted, this capability may impact all components, rather than just components working as interface with external world.

The **rationale** for this requirement is the following: of course, the client prefers applications and platforms that have minimal impacts on his environment and processes. So, the capability to adapt the software deployment to the model required by the client could pull the client forward to the applications that satisfy this requirement.

Related to: Scenario S6

Requirement type: Quality attributes

Domain: Deployment

Target: Domain-independent

Originator type: Service consumer, service provider

Fit criterion: This requirement could be measured by the capability of the adopted architecture to cover several models of deployment, without (or with reduced) users' intervention. In any case, basic features should be preserved (cf. security and availability example).

4.16 Workflow management and integration (R16)

Often new application must be integrated in the business processes of the client. The workflows are a typical example. The workflow management inside an application is quite usual; on another level of complexity there is the integration of a new workflow (owned by a specific application) with other workflows already participating to the business processes.

This requirement involves several aspects. One of these aspects is related to interoperability at functional level and at technical level. For instance, one challenge could be the integration of flows of each workflow with the flows of others workflows by functional point of view. In this case, the main task is to identify the connection points among the flows and the rules governing the interaction among them. By another point of view, the technical aspect is involved in case of workflow engine provided by different vendors.

The **rationale** for this requirement is the following: it's quite unusual that an application could provide an effective added value to the client business without integration with the main processes of the organization. This requirement stresses the aspects related to the integration of a specific kind of process (in this case, a technical one): the workflow.

Related to: Scenario S6

Requirement type: Functional requirement

Domain: Integration/Interoperability

Target: Domain-independent

Originator type: Service consumer

4.17 Aided configuration (R17)

This requirement aims to stress the capability of the architecture an application is based on, to support the configuration task during the deployment phase and during maintenance (cf. versioning and customization for each client). For instance, an application could be provided to several clients, belonging to different domains and with different size (University, SME, telecommunications company, automotive industry, research centres web, etc.). An aided configuration phase could be very useful in term of effort and effectiveness first of all from provider side. Of course, besides technological aspects, this requirement involves the configuration models that should cover as many cases as possible. Again, an aided configuration allows storing and managing configurations of several clients during the time, being able to replicate or to go back to previous versions of the configuration.

The **rationale** for this requirement is the following: this requirement should allow a speedy configuration during the initial deployment phase and should support the maintenance tasks after.

Related to: Scenario S6

Requirement type: Quality attributes

Domain: Deployment and maintenance

Target: Domain-independent

Originator type: Service provider

4.18 Modelling capabilities (R18)

The overall architecture should allow modelling the company whose business will benefit of the new application/platform based on the architecture itself. Of course, the adaptability and the flexibility of an application should be inherited from the architecture in order to reach natively two main goals. The first is the reduction of effort required for the customization, configuration, integration, and the second one is the capacity to actually fit the client's needs and to actually reflect the client structure, processes, policies, roles, etc.

The **rationale** for this requirement is the following: the modelling capabilities of the architecture allow fitting the company organizational model with a reduced effort.

Related to: Scenario S6

Requirement type: Design constraints

Domain: Integration/Interoperability/Deploying

Target: Domain-independent

Originator type: Service consumer, Service integrator/developer

4.19 Technical interoperability (R19)

Interoperability between services regarding distributed workflows and individual service interaction.

The **rationale** for this requirement is that without this requirement, integration can only be performed on a case-by-case basis.

Related to: Scenario S7

Requirement type: Functional requirement

Target: Domain-independent requirement

Originator type: Service integrator/developer

4.20 Device integration / vertical integration (R20)

It is needed to integrate devices in service-oriented architectures, also known as vertical integration. For different kind of devices different embedded SOAs have to be developed including respective standards.

In practice, there exist already domain specific standards or best practices for device handling. For instance, in the healthcare domain, there exist frameworks like the Microsoft CHF (Connected Health Framework) or the Eclipse OpenHealthFramework.

Such standards are of great importance to the developers of applications for devices. These standards often contain domain specific information models and/or protocols and hence substantially facilitate the application development and interoperability.

Domains with elaborated standards and best practices will stick to them and continue to use them instead of some generic technologies. Other domains will develop their own standards and best practices. Therefore, architectures for device integration have to develop integration mechanisms for the different domain specific standards.

The **rationale** for this requirement is the following: without this requirement, integration can only be done on a case-by-case basis.

Because of the practical relevance and the apparent indispensability of the domain standards for information models or protocols related to embedded devices these standards have to be taken into account by NEXOF-RA.

This is not only a requirement from the e-Health domain. Note that in industrial domains there exist standards like BACnet (Building Automation and Control Networks) or OPC UA (Object Linking and Embedding for Process Control Unified Architecture). This underlines the importance and the universality of this requirement.

There are also emerging standards like SODA (Service Oriented Device Architecture) that have not yet gained such a wide acceptance as the domain specific standards.

Related to: Scenario S7

Requirement type: Functional requirement

Domain: Embedded Devices

Target: Cross-domain

Originator type: Service integrator/developer

Comments: NEXOF-RA should be general enough to cover architectures for integrating mobile, pervasive and embedded devices.

Conflicts: Performance and footprint are important criteria.

4.21 Distributed workflow (R21)

Integrating workflows from different stakeholders raises many issues, as, for instance, federated identities, transactions, etc. Each such issue represents a complex requirement in itself. The requirement stated here is intended as a higher-level requirement and formulates that all the conceptual solutions provided by NEXOF-RA have to interrelate in order to allow a seamless integration of workflows, in particular, if the workflow combines different domains.

The **rationale** for this requirement is the following: the expected benefit of the service-based economy lies in B2B and B2C integration which is de facto based on distributed workflows.

Related to: Scenario S7

Requirement type: Functional requirement

Target: Domain-independent / Cross-domain requirement

Originator type: Service integrator/developer

4.22 Stateful, device adaptive service transfer (R22)

When interacting with a service, it must be possible to interrupt the session, save the status of it, and continue working with the same service and the previous state on a different device. Depending on the device, the amount of data and the views may be scaled.

The **rationale** for this requirement is the following: there are working environments that frequently require changing devices while continuing working in the same workflow.

Related to: Scenario S7

Requirement type: Functional requirement

Domain: The idea originated in clinical workflows (integrating diagnostic appliances during a patient's examination) but is applicable also to factory workflows such as complex assembly or maintenance tasks, etc.

Target: Domain-independent

Originator type: Service consumer

Conflicts: There are scalability issues, since not all data can be accessed on all kind of devices. That means that in some cases certain functional requirements may not be fulfilled.

4.23 Adaptability (R23)

In order to react on changed conditions in the traffic situation (that is, for instance, in the case of traffic accidents, emergencies; major road constructions or similar activities), new control and management strategies have to be executed. The traffic management system should then be able to identify the key necessities of the new situation, retrieve an appropriate realization of the needed control mechanisms and adapt to the new strategies.

The thereby required reaction pattern of the system is not specific to the traffic management domain, but inherent to autonomously – or at least partly autonomously reacting systems.

The **rationale** for this requirement is the following: the requirement results from the need to react autonomously on unexpected and unforeseen situations in a reasonable way.

Related to: Scenario S9

Requirement type: Functional requirement

Target: Domain-independent

Originator type: Service consumer

4.24 Rapid reconfiguration (R24)

In case of an emergency, the traffic management system has to react quickly and change its strategies. This includes the fast selection of the right new strategy as well as the swift propagation of the new rules to the traffic control devices (for instance, traffic lights).

However, this means not only, that the reconfiguration and adaptation process is performed in a very short time, it also includes that the reconfiguration and adaptation control process comes to a conclusive decision in a very short time and, in particular, does not fall into an “adaptation lifelock”, that is, that is does not try to perform again and again repeated adaptation steps without coming to a reasonable and stable new control structure in the requested time.

Again, the traffic management system stands here only as an example for a dynamically adapting system and thus, the requirement is not domain specific.

The **rationale** for this requirement is the following: reactions on emergencies, etc. have to be quick.

Related to: Scenario S9

Requirement type: Quality attributes

Target: Domain-independent

Originator type: Service consumer

4.25 Integrity (self-diagnosing and self-healing) (R25)

It has to be made sure that no one can jeopardize the traffic management system, that is, to override the decisions of the traffic management system from outside the system by either manipulating the sensors and the control devices or by introducing harmful strategies. Any attempt has to be detected.

The **rationale** for this requirement is the following: any attempt to manipulate the traffic management system in a wrongful manner may result in a disastrous traffic situation with all its bad consequences.

Related to: Scenario S9

Requirement type: Functional requirement

Target: Domain-independent

Originator type: Service consumer, service provider

4.26 Dependability for device integration (R26)

Embedded devices, which are typically sensors or actuators or combinations thereof and which are typically consisting of hardware and software, have per se to fulfil critical dependability requirements. For example, performance, reliability and availability requirements are important.

When these devices are integrated into other applications, in particular, in the case of distributed non-embedded applications, the integrating software has to fulfil similar dependability requirements. Hence, the dependability is not only threatened by the device itself but by hardware and software aspects of the integrating application. Such related hardware and software aspects are, for instance, communication mechanisms (like internet, RPC, etc.), storage mechanisms (like databases, etc.) and also the computational complexity.

For example, in the assisted living or the complex diagnostic workflow scenario, the devices used during a diagnosis or for monitoring a patient's health status need not only be dependable but also, after their integration into application specific workflows, the data from these devices must be accessible in a dependable way.

The **rationale** for this requirement is the following: when devices are integrated into other applications, then dependability requirements are lifted to the application level as well.

In particular, existing communication means or storage components such as messaging middleware based on the Internet or intranet often does not provide the needed guarantees.

Related to: Scenario S7

Requirement type: Functional requirement

Target: Domain-independent

Originator type: Service consumer

4.27 Compliance to privacy, and security policies (R27)

Security and privacy functions regarding treatments, services, workflows and individual services interactions, with the aim to be compliant to (for example in the Health domain the US-regulations defined within the Health Insurance Portability and Accounting Act (HIPAA) Privacy and Security rules)

The **rationale** for this requirement is the following: without this requirement, a specific security and privacy policy will have to be defined. It is better to rely on a standard recognized in world of health. This standard covers all health stakeholders: individuals including doctors, nurses, pharmacists, physical therapists and organisations including hospitals, laboratories, pharmacies, nursing facilities and more generally, all health services and clearinghouses. The privacy and security rules require safeguarding all PHI (e.g. Protected Health Information).

At the beginning, appropriate PH must be identified and then a risk analysis must be performed taking into consideration various levels of risks (present and residual risks). Then the Privacy Rules and the Security Rules will require identifying security and privacy functions related to safeguard of electronic notice and patient chart authorisation (access and integrity), separation of duties, accounting of disclosure, mitigation procedures, emergency access that are appropriate for the Health scenarios. Using the results of this analyse, appropriate privacy and security functions related to safeguard of electronic notice and patient chart authorization (access and integrity) will be developed, separation of duties, accounting of disclosure, mitigation procedures, emergency access, audit.

Related to: Scenarios S7 and S8

Requirement type: Functional requirement

Domain: e-Health

Target: Domain-dependent

Originator type: Privacy officer / security officer

Fit criterion: One key area to focus on to test if the solution matches the original requirement is to perform a same risk as in compliance analysis phases to HIPAA standard. On a longer-term, the only major differences will be that many new threats and vulnerabilities will exist.

Comments: Security and privacy policy, risk analysis, protection functions and mechanisms **Conflicts:** It has to be carefully checked that solutions for monitoring are not in conflict with safety rules (separation of duties). The same is true between the rules of privacy and security, for example concerning the audit means and emergency procedures.

In fact, it is necessary to understand where the privacy rules and security rules overlap. The privacy rule requires safeguarding all Protected Health

Information. To restrict access to PHI, it is necessary to implement security controls. The privacy rules also explicitly require security safeguards.

Safeguard measures concerning privacy and security cannot be chosen independently of each other.

Supporting materials: Some documents that illustrate and explain this requirement:

- <http://www.hipaa.org/>
- <http://www.hhs.gov/ocr/hipaa/finalreg.html>
- <http://privacyruleandresearch.nih.gov/resources.asp>
- <http://www.hipaacomply.com/>
- <http://www.ioma.org/pdf/iomahipaahelp.pdf>

4.28 Collaborative business process acquisition, modelling and effective management (R28)

This requirement aims at offering the capacity to acquire/form, model and manage very effectively and efficiently Collaborative Business Processes for what concerns CDM applications.

The **rationale** for this requirement is the following: the rationale behind the proposed scenario is to move from Business Process management to Collaborative Business Process Management for the sake of improving the overall performance of (global) CDM applications (also the degree of satisfaction of end-users).

It is also to investigate and propose the right (service) tooling to effectively and efficiently manage them once acquired/modelled

Related to: Scenario S11

Requirement type: Functional requirement

Domain: CDM application domain

Target: Domain independent

Originator type: Global CDM application integrator

Fit criterion: Tools to acquire/form Collaborative Business Processes; language (presumably XML-based) to model/represent those; reasoning tools and techniques to learn, evolve CBP to make them more effective/efficient (incl. re-engineering of existing processes); measure of the effectiveness of the approach (both objective and subjective).

Conflicts: The Collaborative Business Processes should be acquired, modelled and managed in full accordance with rules coming from either stakeholder and/or from existing legislations/regulations.

Supporting materials: Information on CDM-ATM can be found on Eurocontrol Web site:

- <http://www.euro-cdm.org/>

- http://www.eurocontrol.int/eec/public/standard_page/proj_Airport_CDM.html

4.29 Distributed architecture (R29)

In order to realize a scenario as described in the scenario S12, it is required to allow partners to exchange information in a distributed way. A suitable architecture therefore has to support the message exchange between remotely located applications and the possibility to add and replace services at any time.

Related to: Scenario S12

Requirement type: Functional requirement

Domain: Architecture

Target: Domain-independent

Originator type: Service integrator/developer

Fit criterion: If this requirement is fulfilled then the construction of applications will be possible that are based on several physically distributed services.

4.30 Integration of services (R30)

Graphical Integration: integrating services in a graphical way is important in order to give the user the impression of a holistic system with a coherent look and feel. Technologies such as CSS allow the adoption of colours and images and may be used to implement a graphical adaptation of external components to the own system. In addition to this, many services allow a menu-based configuration of their look and feel. For example, many Web 2.0 widgets may be configured using a configuration UI. However, what is missing is a common and service-independent way of describing and defining graphical constraints, which might be used by external providers automatically in order to adopt their services without any manual refinement.

Logical Integration: integrating a component is a rather manual process today. Depending on the type of the component, it might be performed by adding a couple of lines into the HTML code of an e-Commerce website or it might require a major change in the core systems of the e-Commerce Vendor. The main problem is that the variety of protocols and formats is very high without allowing a standardized access to third party services.

Communication: in real-world use cases, the data exchange is usually performed by either exchanging messages via an RPC interface such as WebServices or by using HTTP-Calls or Emails as the main way of exchanging information, depending on the type of services. Using RPC is normally applied when integrating components in a Web 2.0 manner. Email communication is a typical way for exchanging payment confirmations or order messages and sending it to the ISV.

Formats

Independently from the communication protocol, the data itself may be described in multiple formats. For exchanging order information, the most common formats found in real world examples are XML, TXT, EDI and CSV.

Related to: Scenario S12

Requirement type: Functional requirement

Domain: Interoperability

Target: Domain-independent

Originator type: Service consumer, Service integrator/developer

Fit criterion: If this requirement is fulfilled then the search integration of services will be eased by providing a precise way of integrating 3rd party services graphically and logically.

4.31 Monitoring and reliability (R31)

Monitoring services is important in order to ensure a continuous operation of systems. However, in most cases, a monitoring and alerting system for informing a vendor of system failures does not exist. In a typical scenario, vendors receive a message from customers whenever a problem with their web-shop appears.

The **rationale** for this requirement is monitoring.

Related to: Scenario S12

Requirement type: Functional requirement

Target: Domain-independent

Originator type: Service consumer

Fit criterion: If this requirement is fulfilled then service consumers will be notified if a service fails or becomes unavailable.

4.32 Orchestration (R32)

Integrating all third party components into one holistic system requires a careful orchestration of all involved components. It is required to describe the process in a structural way. Although business process description languages are available today, they are often not used for orchestrating the collaboration between the different services.

Related to: Scenario S12

Requirement type: Functional requirement

Domain: Orchestration and process management

Target: Domain-independent

Originator type: Service integrator/developer

Fit criterion: If this requirement is fulfilled then services will be combinable in an easy way allowing service integrators to easily specify and maintain a business process and to connect it to specific services

4.33 Trust and confidence (R33)

Trust and confidence is a major issue in e-Commerce today. The condition of trust is an important enabler to personal and market transactions of many kinds, even where there are sophisticated legal remedies available.

The major trust principles and mechanisms that support the promotion of online trust are identified hereafter.

- Trust depends on identity: Identity management and personal data's protection are particularly important in the design of a trusted infrastructure.
- Trust is a function of perceived risk: Trust is a belief or expectation that the word or promise by the merchant can be relied upon and the seller will not take advantage of the consumer's vulnerability.
- Trust deepens over time and with increased reciprocity: Most theorists agree that trust is intimately associated with risk and when a trustee realizes that a truster has taken considerable risk in trusting them, they tend to be motivated to behave in a trustworthy manner. We must find a fine balance between security and privacy.
- It is important in encouraging trust in e-commerce is not the specific product or service features, but the behaviour of the parties involved and whether they deliver on their promises. Consequently the focus should be on maintaining feedback on an individual and a company's behaviour during the course of the transaction and rate the dimensions that indicate that the parties have delivered on their promises to one another.
- Trust is a matter of degree: Trust is a matter of degree. There is no such thing as blanket trust. In fact trust can be defined as the degree to which the truster holds a positive attitude toward the trustee's goodwill and reliability in a risky exchange situation.

This involves that we go up from security goals to security services. The specification of the move from security goals to security services requires additional work to classify the security services according to the various levels of the security goals.

Related to: Scenario S12

Requirement type: Quality attributes

Domain: Security

Target: Domain-independent

Originator type: Service consumer, Service provider, Service integrator/developer

Fit criterion: If this requirement is fulfilled then all communication and data exchange will be performed in a secure way

4.34 Service discovery (R34)

Finding Business partners that offer a specific service is a major challenge. Of course web search engines such as Google or Yahoo provide a way of searching and finding business partners but what is required is an easy to use way for finding services based on specific criteria and semantic descriptions.

Related to: Scenario S12

Requirement type: Functional requirement

Domain: Service Discovery

Target: Domain-independent

Originator type: Service consumer

Fit criterion: If this requirement is fulfilled then the search and retrieval of services will be possible based on different criteria.

4.35 Information as a service (R35)

Providing information in a coherent and trusted way can leverage the business of a whole company.

The **rationale** for this requirement is the following: information is one of the most valuable goods in the modern society. Access to it might make a huge difference for the business, people lives and safety.

Related to: Scenario S14

Requirement type: Functional requirement

Domain: Service publication and discovery

Target: Domain independent

Originator type: Service integrator/developer

Fit criterion: If this requirement is fulfilled then the information can be exposed as a service.

4.36 Execution of human-based process steps (R36)

Humans should be asked for help when a performed task is going beyond the machine intelligent capabilities. The result of the human activities should be passed back to the system.

The **rationale** for this requirement is the following: in most scenarios, machines cannot perform all of their tasks without human interactions and manual decisions. Tight cooperation between human intelligence and machine computational power and networking is needed. In addition, some

actions require the human confirmation when a human must take a responsible for a certain circumstances of performing an action.

Related to: Scenario S14

Requirement type: Functional requirement

Domain: Service orchestration

Target: Domain independent

Originator type: Service integrator/developer

Fit criterion: If this requirement is fulfilled then the humans will be able to act as a part of hybrid human-machine system enforcing it with intelligence and responsibility.

4.37 Information integration (R37)

Gather together all the information about an interesting fact is a challenging task in a current distributed environment. Service architecture should provide an adequate support for information integration.

The **rationale** for this requirement is the following: the current boost in IT makes it possible for any single party to store information in its own location and format. Creating an unbroken picture out of the scattered pieces is an absolute requirement for the future information systems.

Related to: Scenario S14

Requirement type: Functional requirement

Domain: Distributed Transaction support

Target: Domain independent

Originator type: Service integrator/developer

Fit criterion: If this requirement is fulfilled then the system can create a coherent view on a scattered data distributed throughout the network.

4.38 Distributed transaction support (R38)

Distributed transaction support is required.

The **rationale** for this requirement is the following: e-Government applications act as an interface for data that is kept in a distributed way. This can occur because of legal needs to ensure data privacy. If data is changed on one place, distributed transaction support is needed to guarantee that the data is updated consistently.

Related to: Scenario S15

Requirement type: Functional requirement

Domain: e-Government

Target: Cross-domain

Originator type: Service provider

4.39 Non repudiability of data transfer (R39)

It is possible to prove that the recipient really received the message and that the sender really sent the message

The **rationale** for this requirement is the following: the submission of an application to a public administration has to be recorded in a way that it can be proven later that the message was sent and that it arrived. Also the sender wants to receive a receipt of the successful transmission so that they can prove that the public administration received the message

Related to: Scenario S15

Requirement type: Functional requirement

Domain: e-Government

Target: Domain-specific

Originator type: Service provider

Fit criterion: The transmission of a document is logged in a way that it proves that the sender submitted the message and that the receiver received it. Both parties get a confirmation of this

4.40 Cross-certification (R40)

In order to secure the access to some critical applications, it is relevant to implement an internal Public Key Infrastructure (PKI). PKI is a trusted framework to provide authentication and confidentiality. Generally, an authority only has a suitable relationship with a limited community or a trust domain. PKI interoperability with some trust model can propagate trust beyond local domain to satisfy needs of larger and more diverse communities, which enables organizations and corporations to communicate with one another within different trust domains.

So, in the context of complex infrastructures, PKI interoperability is usually addressed through the cross-certification service. The term cross-certification is more commonly used today to simply refer to the ability of one certification authority to certify another certification authority's public key, making it possible to build chains of trust through arbitrary paths of certification authorities in infrastructures operated by different organizations. Instead, a "chain of trust" is established from the user's trusted copy of his domain's CA's public key to the public key used to sign the foreign certificate.

So the cross-certification provides the necessary links between normally unconnected domains to establish a valid certification path. It is not required that the same trust models be used in each domain.

This method has been described and standardized by the ISO and the American National Standards Institute (ANSI).

Related to: Scenarios S1, S5, S9, S10, S11, S12, S15, S16, S17

Requirement type: Functional requirement

Domain: e-Government, e-Commerce, e-learning, Traffic management, Crisis management, collaborative decision making, and assisted industrial maintenance

Target: Cross-domain

Originator type: Certificate authority

Fit criterion: Cross-certification avoids each user having to acquire, through a trusted delivery mechanism, a foreign Certificate Authorities (CA's) public key.

4.41 Resilience & Continuity of service (R41)

Resiliency is the ability to avoid, minimize, withstand, and recover from the effects of adversity, whether natural or manmade. Resiliency applied to the nation's critical infrastructure is trustworthiness under stress and spans high availability, and continuous operations.

These operations are evolving into large systems of systems. In normal times these operations may operate satisfactorily in a loosely coupled arrangement. However, for these operations to be resilient under stress, more than a loosely coupled arrangement is needed.

A defined engineering challenge of adopting resilience throughout the nation's critical infrastructure is needed. The interoperability of information sharing and platform operations must be assured, distributed supervisory control protocols must be in place, and operation sensing and monitoring must be embedded.

These capabilities cannot be expected to evolve in a loosely coupled environment. They must be holistically specified, architected, designed, implemented, and tested if they are to operate with resilience under stress. A management, process, and engineering maturity framework is necessary to advance the assurance of services continuity, system survivability, and system of system resiliency capabilities.

Related to: Scenarios S10

Requirement type: Functional requirement

Domain: System management, monitoring, configuration reconfiguration

Target: System of System flexibility

Fit criterion: The system has fault tolerance and real time reconfiguration capabilities. A limited number of parameters governing the behaviour of large families of unit modules by type of networks within the system have been defined.

4.42 3D virtual environment services support and integration (R42)

It is needed to host and integrate 3D distributed interactive applications. They require specific services and specific support from the underlying

architectures. In case, they require high dependability, it must be possible to assign high priority to such applications.

In practice, there exist already 3D solutions and de facto standards such as Second Life. There exist also industrial standards for 3D developments of devices from the Computer Aided Design (CAD). Applications built with one standard but also built with different standards have to be interoperable and 3D data needs to be exchanged.

The **rationale** for this requirement is the following: without this requirement, hosting of 3D services needs a lot of adaptation and integration. It can only be done on a case-by-case basis. In the near future, 3D services will gain increased practical relevance. This is a requirement from the industrial domain. It applies to plant maintenance, collaborative engineering environments as well as to healthcare equipment maintenance and remote assisted surgery.

Related to: S17

Requirement type: Functional requirement

Domain: 3D virtual reality services

Target: Cross-domain

Originator type: Service producer/integrator/developer

Comments: NEXOF-RA should be general enough to cover architectures for integrating 3D services.

Conflicts: Performance and dependability is an important criterion.

5 REQUIREMENT CLASSIFICATION

This section describes the classification of requirements according to the early (provisional) reference architecture (“Services Value Stack”), which by now (at the date of this report) are proposed and promoted by the “Reference Architecture: Specifications “ track.

The requirements described in this document are considered in the creation of the model and the specification, based on the identified concerns. These two tracks (“Reference Architecture: Model” and “Reference Architecture: Specifications”) base their system requirements on the requirements obtained through the requirements obtained from the stakeholders in this document.

The classification in this section groups requirements according to the primary component within the services value stack that is responsible for their implementation.

This classification points out which component of the future reference architecture is responsible for its implementation. This will **ease the assessment of the requirements** within the deliverable “Assessment Report” (D10.3) and help to understand if the proposed reference architecture implements the requirements collected within this document or not.

5.1 Services value stack

The Services Value Stack developed by the “Reference Architecture: Specifications” track is the following:



Figure 14: Services Value Stack

For each component of the Services Value Stack, responsibilities are defined, which are formulated as questions. For example, the question “How

can a service be created?” is one among the questions assigned to the component “Service creation and execution”, which is responsible to provide the necessary support to create and execute services.

New requirements can be classified (i.e., assigned to a specific component of the Services Value Stack) by going through the questions and analyzing for which question a particular requirement gives an answer.

The “Reference Architecture: Specifications” track team developed an initial set of questions. These questions were then refined using the collected requirements. The generated questions are listed below (the requirements that fit to the respective question are added in square brackets).

The questions are hierarchically structured, this means that if a question covers a specific aspect of a more general question, than the specific question is indented and listed below the general one.

The following list shows the classification in detail, stating the concern as well as the linked requirement (in form of a question) by “Reference Architecture: Specifications” track. To obtain an overview of the links between the single concern and its linked requirement(s) see section 5.2 (“Classification of requirements”).

Service Creation & Execution

- SR 1. How can a service be created?
 - SR 1.1. How can a service be specified to be implemented?
 - SR 1.2. How can a new service be implemented?
 - SR 1.2.1. How can a service be tested to validate the implementation?
[L4.1 (Testability)]
 - SR 1.2.2. How can stateful services be implemented?
[R22 (Stateful, device adaptive service transfer)]
 - SR 1.2.3. How can services be extended to adapt to new user requirements (versioning and compatibility)?
[L1.11 (Extensibility and evolution),
L7.13 (Adaptability),
R23 (Adaptability),
R24 (Rapid reconfiguration)]
 - SR 1.3. How can a functionality provided by a legacy application be promoted to a service?
[L1.12 (Support to legacy systems),
R14 (Integration of an application with legacy applications)]
 - SR 1.4. How can a service be executed (run-time support)?
[R36 (Execution of human-based process steps)]
 - SR 1.4.1. How can service execution be stopped and resumed?
[R22 (Stateful, device adaptive service transfer)]
 - SR 1.5. How can services and tools for their execution and implementation be realized in order to participate to monitoring policy?
[L7.12 (Multi level constraints definition)]

Service Invocation

- SR 2. How can a service be invoked?
 - SR 2.1. How can services be identified and addressed?
[L2.6 (Service addressing)]
 - SR 2.2. How can a service be described to be invoked?
[R8 (Interoperability and flexible communication standards)]

- SR 2.3. How can a service requester agree with the service provider for service invocation (Service Level Agreement, Agreement Templates Negotiation)?
[L3.4 (Awareness of unreliable services),
R1 (SLA Processing)]
- SR 2.4. How can messages be sent/received to/from a service?
[R19 (Technical Interoperability),
R11 (Services integration by semantic mash-up),
R13 (Unified Communication),
R29 (Distributed architecture),
R30 (Integration of Services)]
- SR 2.4.1. How can messages be adapted between a client (requester agent) and a service (provider agent)?
[R12 (Harmonization of several heterogeneous information sources),
R37 (Information integration)]
- SR 2.4.1.1. How can one message be split into several messages?
[L6.2 (On-the-fly service switching)]
- SR 2.4.2. How can messages be controlled and validated?
- SR 2.4.3. How can a network protocol be adapted?
- SR 2.4.4. How can service location transparency be achieved (routing)?
[L6.2 (On-the-fly service switching),
R10 (Location based routing)]
- SR 2.5. How can a service be tested to validate if it satisfies consumers needs?
[L4.1 (Testability)]

Service Discovery

- SR 3. How can a service be discovered?
[L2.1 (Registry availability)]
- SR 3.1. How can a service be published to make it knowable so that it can be found and used by consumers (browsing)?
[L1.2 (Publication APIs),
L1.4 (Push mode publication mechanisms),
L1.5 (Pull mode publication mechanisms)]
- SR 3.1.1. How can a service be described in order to be found (provider entity description, etc.)?
[L1.1 (Data model),
R2 (Uniform Service representation),
R5 (Service description)]
- SR 3.1.2. How can a service that satisfies client requirements be found (searching)?
[R34 (Service Discovery),
R2 (Uniform Service representation),
R3 (Service Discovery mechanisms)]
- SR 3.1.3. How can published services be organized with different levels of visibility?
[L1.6 (Categorization and selective publication)]
- SR 3.2. How can the existence of a service be notified to potential consumers (publish/subscribe discovery, market places)?
[L1.4 (Push mode publication mechanisms),
L1.5 (Pull mode publication mechanisms)]
- SR 3.3. How can service directory (registry) be organized (centralized, distributed, federated) and updated?
[L1.3 (Registry architectures),
L1.7 (Up-to-dateness of published information),
L1.9 (Applicability),
L2.2 (Registry correctness),

R4 (Decentralised architecture)]

Management

- SR 4. How can services be managed?
- SR 4.1. How can a service/process be made ready to be executed/enacted in the operational environment (deployment)?
- [R6 (Service deployment),
R15 (Adaptive deployment),
R16 (Workflow management and integration),
R17 (Aided configuration),
R18 (Modelling capabilities)]
- SR 4.2. How can service activation (run/stop, run-time configuration) be managed?
- SR 4.3. How can services be phased out (decommissioned/retired) when they will no longer be supported or needed?
- [R7 (Service Decommissioning)]
- SR 4.4. How can management take into account service versioning?
- SR 4.5. How can the system and its environment be monitored (which information, which mechanisms)?
- [L7.2 (Event based monitoring),
L7.3 (History based monitoring),
L7.4 (Assertion based monitoring),
L7.5 (Environmental Monitoring)]
- SR 4.5.1. How can services be monitored?
- [L7.1 (Functional and non-functional qualities),
L3.3 (Co-ordination semantics),
L3.4 (Awareness of unreliable services),
R31 (Monitoring and Reliability),
R1 (SLA Processing)]
- SR 4.5.2. How can processes be monitored?
- [L7.11 (Process instrumentation through appropriate development time tools),
L3.3 (Co-ordination semantics),
R1 (SLA Processing)]
- SR 4.5.3. How can monitoring policies be configured or extended?
- [L7.6 (Dynamic regulation of monitoring activities),
L7.9 (Dynamic renegotiation of constraints),
L7.10 (Extensible framework for data collectors, data analyzers and constraint metrics),
L7.12 (Multi level constraints definition)]
- SR 4.5.4. How can information regarding service and process execution or about the system and its environment be collected for monitoring purposes?
- [L7.7 (Service Data Collection),
L7.8 (Environmental Data Collection)]
- SR 4.5.5. How can monitoring information be provided in a timely fashion?
- [L2.3 (Monitoring information)]

Process

- SR 5. How can a process be realized by composing services?
- SR 5.1. How can processes be designed in terms of the services they are composed of (Orchestration, Choreography descriptions)?
- [R32 (Orchestration)]
- SR 5.1.1. How can a process be designed to select some of the services it composes at run-time in order to complete and satisfy QoS constraints (Dynamic composition, Dynamic Binding, Constraints Satisfaction)?
- [L2.5 (Workflows)]

- SR 5.1.2. How can process be designed in order to manage interoperability matters (data mapping, message transformation)?
- SR 5.2. How can a process be implemented?
- SR 5.3. How can a process be enacted?
[R21 (Distributed Workflow)]
- SR 5.3.1. How can the process engine be aware of the execution state of process instances and participating services?
[L3.3 (Co-ordination semantics)]
- SR 5.3.2. How can the process engine enable dynamic composition (late binding)?
[L6.2 (On-the-fly service switching)]
- SR 5.4. How can a process manage transactions (both at design and run-time)?
[L3.1 (Atomic transaction notifications),
L3.2 (Long running transaction notifications),
R38 (Distributed Transaction support)]
- SR 5.5. How can a process be realized as a service?
- SR 5.6. How can services and tools for their execution and implementation be realized in order to participate to monitoring policy?
[L7.11 (Process instrumentation through appropriate development time tools),
L7.12 (Multi level constraints definition)]

Business Analysis

- SR 6. How can processes be analysed from the business perspective?
[R28 (Collaborative Business Process acquisition, modelling and effective management)]
- SR 6.1. How can a process be monitored to provide information suitable for business analysis?
[L7.12 (Multi level constraints definition),
L7.14 (Monitoring process)]
- SR 6.2. How can risks and consequences of failure coming from the automation of a business process be quantified?
[L4.2 (Risk determination),
L4.3 (Determination of failure consequences)]

User & Client Interaction

- SR 7. How can users be supported to access all the SOA-based system functionalities (Services and Processes creation, management, execution, etc.)?
[L7.11 (Process instrumentation through appropriate development time tools),
R26 (Dependability for device integration),
R30 (Integration of Services),
R36 (Execution of human-based process steps),
R42 (3D Virtual Environment Services Support and Integration)]
- SR 8. How can system functionalities be provided to external systems (API)?
- SR 8.1. How can system functionalities be provided to mobile devices?
[L1.10 (Mobile users),
R20 (Device Integration/Vertical integration)]

Security & Trust

- SR 9. How can security be managed through the whole SOA-based system?
[L5.1 (Basic security features for the service delivery platform),
L7.15 (Compliance-based service monitoring),
R39 (Non-repudiability of data transfer)]
- SR 9.1. How can authentication be managed?
- SR 9.2. How can authorization be managed?
[R9 (Federated (temporal) identity management)]
- SR 9.3. How can privacy be managed?
[R27 (Compliance to Health Privacy and Security requirements),

- SR 9.4. How can security domains be federated?
[L5.2 (Interoperability between different security domains and infrastructures),
L5.5 (Privacy by contract),
R21 (Distributed Workflow),
R40 (Cross-certification)]
- SR 9.5. How can different kind of CAs be managed?
[L5.3 (Support to different kinds of CAs)]
- SR 9.6. How can trustworthiness be evaluated?
[L5.4 (Use of ratings for evaluating trustworthiness)]

Resource Infrastructure

- SR 10. How can the infrastructure enable the execution of all the other modules of the service value stack?
[L1.8 (Scalability),
L6.1 (Limitation of communication overhead),
R25 (Self-diagnosing and self-healing),
R24 (Information as a service),
R41 (Resilience & Continuity of service),
L6.3 (Scalable performance and throughput)]
- SR 10.1. How can the infrastructure enhance service availability by replication?
[L6.2 (On-the-fly service switching),
L2.4 (Service availability)]

5.2 Classification of requirements

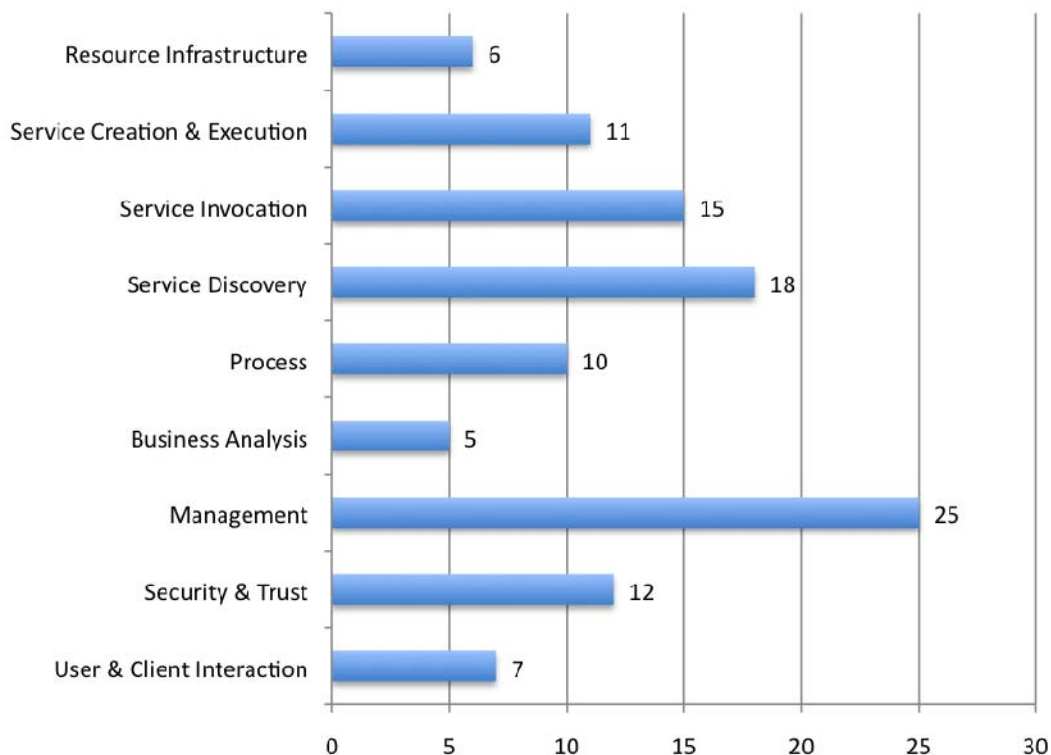


Figure 15: Number of collected requirements per category

The questions described in the previous were used to assign the State-of-the-Art requirements as well as the requirements collected by partners to the concerns of the services value stack.

In total, the chart in **Errore. L'argomento parametro è sconosciuto.** shows the total number of requirements classified per category.

The classification of state of the art requirements is shown in figure 16. The classification of requirements collected by partners is shown in figure 17.



Figure 16: Classification of state-of-the-art requirements

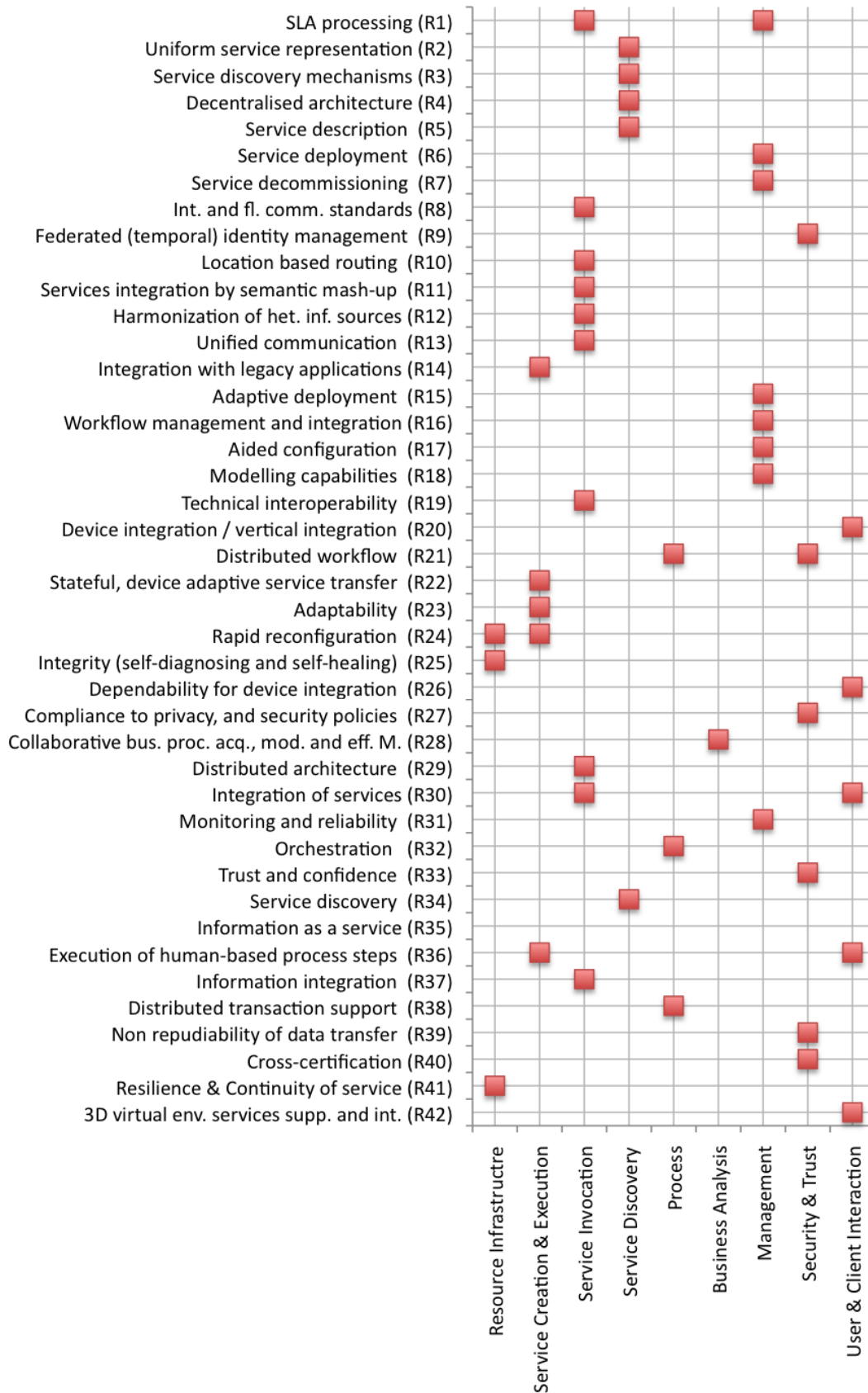


Figure 17: Classification of requirements collected by partners

6 REQUIREMENT PRIORITIZATION

The here presented requirements prioritization aims at identifying the most important requirements for the future reference architecture given by the stakeholders of the here described scenarios and requirements.

This prioritization aims to provide additional help to workpackages that take into consideration the here collected market requirements.

The prioritization was done using three categories:

- **Must have:** it is expected that every user needs the described requirement; therefore it has to be implemented.
- **Should have:** it is expected that the majority of users need the described requirement; therefore it should be implemented.
- **May have:** it is expected that some users would find this requirement useful; it would be nice to have it implemented.
- **Not applicable:** the requirement is not applicable for architecture or is out of the scope of NEXOF-RA.

The prioritization of requirements derived from the state-of-the-art is shown in figure 18; the prioritization of requirements collected by partners through the analysis of scenarios is shown in figure 19.

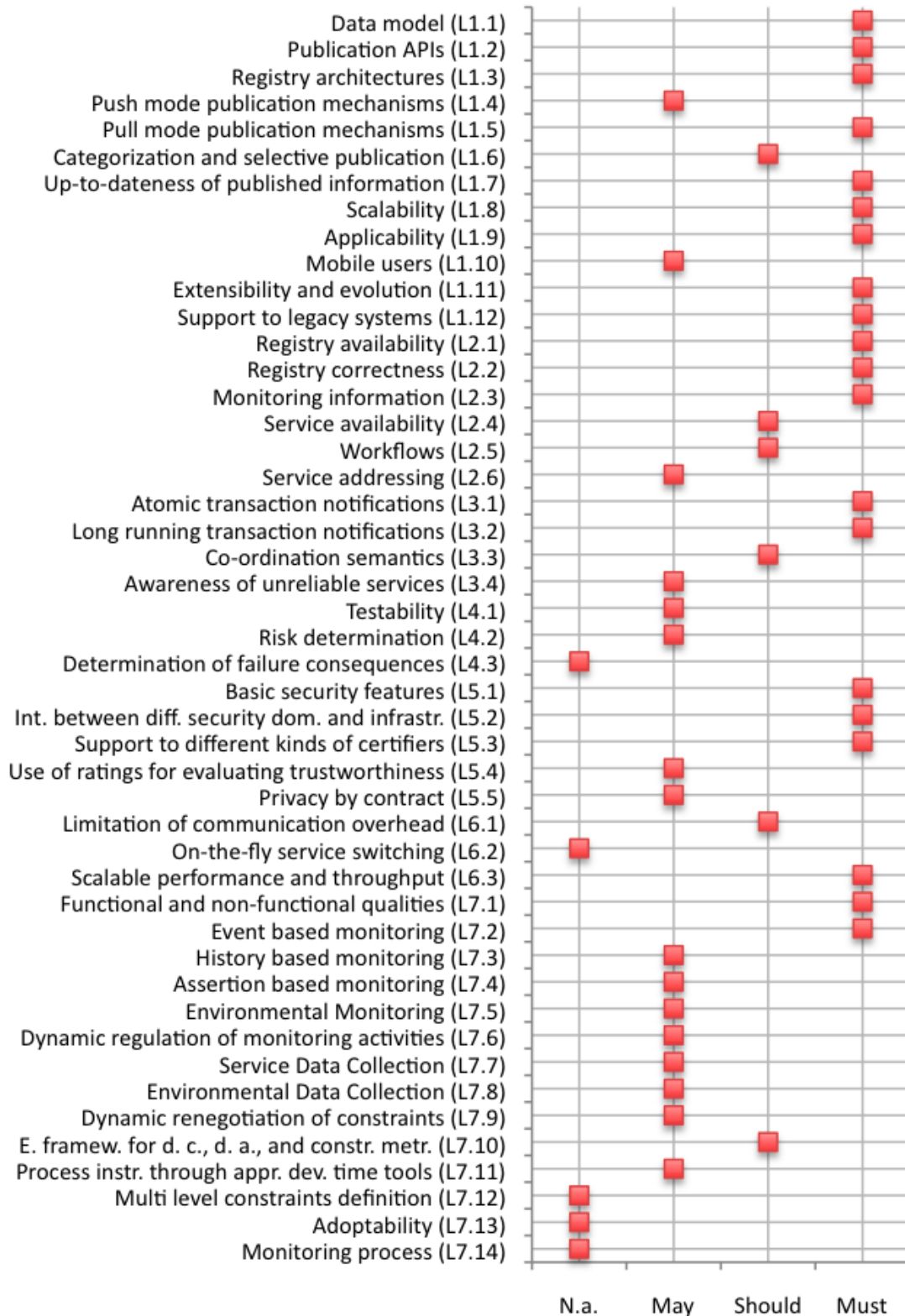


Figure 18: Prioritization of requirements derived from the state-of-the-art

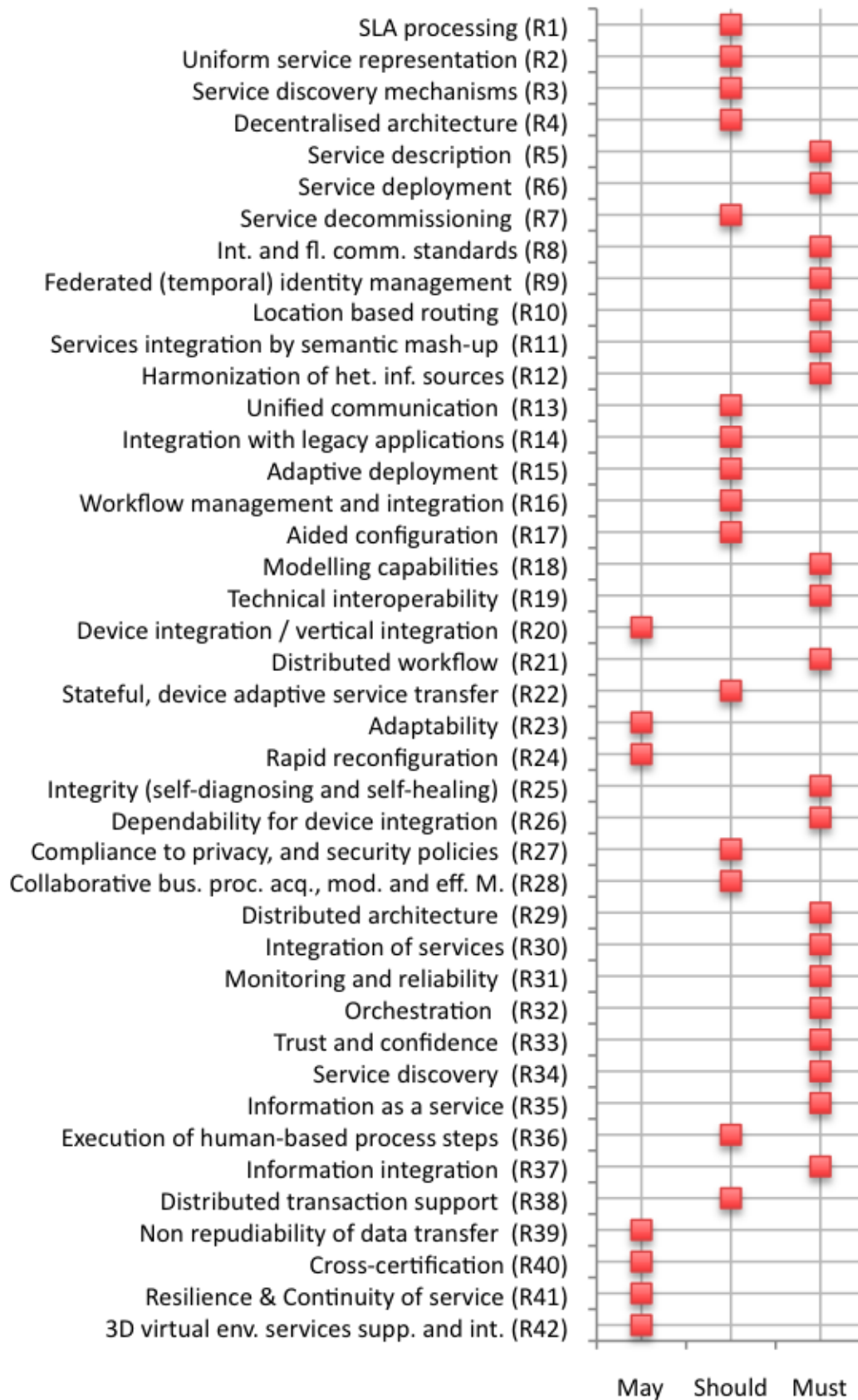


Figure 19: Prioritization of requirements derived from user scenarios

7 OPEN REQUIREMENTS PROCESS

To achieve a higher alignment of the project with the actual needs of the market an Open Requirements Process to collect additional scenarios and requirements will be carried out on a 6-months basis.

The templates given in A2 (Templates for questionnaires) will be used to collect scenarios and requirements.

The Open Requirements Process will allow to collect and to consolidate new requirements from different stakeholders (e.g., the industrial sectors of table Table 9 not yet covered or the domains suggested by the industrial advisory board) and to identify gaps in the architecture based on the new needs.

The following describes the steps carried out in the Open Requirements Process:

- Submission of the request for proposals to mailing lists, forums, publication on NEXOF-RA website
- Responses must be received within one month after the publication of the request for proposals
- One/several phone conference/s among the “Requirements and Assessment Criteria” track members will be scheduled to discuss the submitted proposals
- The accepted proposals will be integrated with the existing requirements and submitted to all work package leaders

The request for proposals will contain the following sections:

- Background information on NEXOF-RA
- NEXOF-RA stakeholders and goals
- Questionnaires for the scenario and requirement/s

The submitted proposals will be **evaluated** considering the completeness and motivation of the submitted requirement, the client references, and the potential impact on the existing architecture.

8 INPUT FROM THE INDUSTRIAL ADVISORY BOARD

This advisory board is divided in three committees (Industrial, Telecommunications, and SME), each with its own membership and operating rules. Members of this board are selected, at project start, to represent various profiles of adopters, including developers, service providers, and end-users. This board is scheduled to meet approximately every 6 months. This board reviews the current developments in the project and provide input in order to ensure the results of the project cater to the needs of the various profiles of adopters.

The considerations of the industrial advisory board are not concrete requirements as those collected in the remaining part of the document but general concerns sometimes regarding the entire NEXOF-RA project.

The considerations are grouped into:

- Wider issues
- Specific issues: Organisation, Proof of concepts, Standards, Survivability, Adoption, Domains, Security, Other

8.1 Wider Issues

To maximise the adoption of NEXOF, it should plan for

- **Guidance material:** How to 'implement' and use NEXOF
- **Assistance:** How to understand documents and clarify or change NEXOF
- **Pragmatism:** Use a pragmatic, step-by-step approach starting from core and building up
- **Concreteness:** The reference architecture should be clear and concise (even if limited scope) rather than large and vague
- **Simplicity:** NEXOF is only valuable if it can be implemented easily, quickly, and cheaply and also if it can be easily communicated

8.2 Specific issues

Organization:

- NEXOF needs to communicate the real prime deliverables of NEXOF, what will be usable by organizations at the end of the project and the conditions for using them
- It is necessary to see the clear picture of NEXOF with different layers and projects including a simple statement on what are the interfaces with other projects.

Proof of concept:

- The concrete content of the proof of concept has to be communicated to convey the advantage of this approach
- SMEs (IT providers and as users) typically use more basic and inexpensive tools. Thus any POC should also include such tools to prove

the credibility to the large SMEs client base of NESSI and the vast potential base.

Standards:

- Standards are critical for the adoption and acceptance of NEXOF, they should be more visibly addressed; they create a level playing for SMEs and they reduce the perception of risks to clients
- It is important to state the progress made with standards and standards bodies, which elements will be standardised and how, where, when
- NEXOF has to define how users will migrate from existing standards to the standards of the new NEXOF SOA environment
- Intermediary results, e.g., a comparison between SOA standards, i.e., which ones to use and which/why not the other ones, can be a very valuable output.

Survivability:

- A clear path for the maintainance of NEXOF has to be defined to encourage adoption. It is necessary to define who will take care of it, how changes will be applied, if backward compatibility will be guaranteed, if maintenance will be a truly open process, etc.

Adoption:

- NEXOF needs to clearly state how it is to be used and which resources are necessary to use it

Domains:

- NEXOF needs to communicate which domains profit from NEXOF
- Eureka.itea has identified 6 major application domains, which are important for Europe in the next years: automotive, aerospace, health, automation production, consumer electronics, and communications. NEXOF needs to consider also these domains.
- NEXOF needs to communicate the effort of moving from domain specific architectures to NEXOF
- NEXOF needs also to communicate with which technologies or standards it conflicts

Security:

- Security is an encapsulating philosophy covering many items with the functions being specific relevant aspects like privacy, authentication. NEXOF needs a simple security that can be used in practice.

Other:

- Communication of the progress of NEXOF to NESSI members is needed
- A technology to share the benefits and deficits of using NEXOF is needed

9 CONCLUSION

This deliverable describes scenarios collected and a list of requirements derived and prioritized from them in order to provide the necessary guidance to the work performed by the “Reference Architecture: Specifications” track with respect to the design of the reference architecture of the Open Service Framework.

This work been done in close cooperation with the “Reference Architecture: Specifications” track in order to subsequently ease the mapping of requirements collected to layered (functional) architectural view promoted.

The NEXOF Reference Architecture must not be a static specification and must allow for the integration of changes and evolutions which result from research or changes in the state of practice; this will be taken into account through the Open Requirements Process.

10 REFERENCES

- [1] CEFRIEL; Requirements for the service delivery platform and evaluation plan, IST European Integrated Project SeCSE, 6th Framework Programme, Deliverable A4.D2, available online at <http://secse.eng.it/wp-content/uploads/2007/08/a4d2-requirements-for-the-service-delivery-platform-and-evaluation-plan.pdf>, 2007
- [2] IEEE Std 1228-1994, IEEE Standard for Software Safety Plans, available online at <http://ieeexplore.ieee.org/iel1/3257/9808/00467427.pdf>, 1994
- [3] UNA, Security and dependability requirements for workflows, IST European Integrated Project Serenity, 6th Framework Programme, deliverable A2.D2.1, available online at http://www.serenity-forum.org/IMG/pdf/A2.D2.1_S_D_requirements_for_workflows_v3.2_Final.pdf
- [4] Canfora, G., Corte, P., De Nigro, A., Desideri, D., Di Penta, M., Esposito, R., Falanga, A., Renna, G., Scognamiglio, R., Torelli, F., Villani, M. L., and Zampognaro, P. 2005. The C-Cube framework: developing autonomic applications through web services. *SIGSOFT Softw. Eng. Notes* 30, 4 (Jul. 2005), 1-6. DOI= <http://doi.acm.org/10.1145/1082983.1083087>
- [5] The Atlantic Systems Guild Limited, Volere Template, available on <http://www.systemsguild.com/GuildSite/Robs/Template.html>
- [6] Sutcliffe, A.G. 1998. Scenario-based requirements analysis. *Requirements Engineering*, 3 (1), pp 48-65, available online: <http://sunsite.informatik.rwth-aachen.de/CREWS/reports98.htm>.
- [7] Rolland, C., Ben Achour, C., Cauvet, C., Ralyté, J., Sutcliffe, A., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois, E., Heymans, P. 1998. "A Proposal for a Scenario Classification Framework," *Requirements Engineering*, vol. 3, pp. 23-47
- [8] Carroll J.M. 1995. The scenario perspective on system development. In: J.M. Carroll (ed.) *Scenario-based design: envisioning work and technology in system development*. J. Wiley, New York.
- [9] European Commission, Enterprise and Industry Directorate-General. 2006. European Industry: A Sectoral Overview. Technical Update – 2006, available online: http://ec.europa.eu/enterprise/enterprise_policy/industry/doc/sec_overview_update06.pdf
- [10] Enterprise Grid Alliance. 2007. EGA Reference Model v.1.5, available online: https://forge.gridforum.org/sf/docman/do/listDocuments/projects.rmgw/docman.root.ega_reference_model_v1_5
- [11] Open Grid Services Architecture. 2003. Open Grid Services Architecture Use Cases, available online: https://forge.gridforum.org/sf/docman/do/listDocuments/projects.ogsa-wg/docman.root.published_documents.use_cases_1_0

A1 GUIDELINES FOR DERIVING REQUIREMENTS FROM SCENARIOS

The requirements of this work package will be based on the analysis of scenarios as described in the NEXOF-RA DoW.

An analysis based on scenarios will allow us to focus not only on the final output – the reference architecture – but also on the in which it has to be used environment (i.e. the user, the system, the context, the activities) [8]. Additionally, this will allow a better validation and verification of the derived requirements within the deliverable D10.2 (Assessment criteria).

The definition of requirements on the base of scenarios will be based on the approach from the European Commission ESPRIT 21903 'CREWS' (Cooperative Requirements Engineering With Scenarios) long-term research project [6].

The rationale of having our NEXOF-RA requirements collection based on CREWS project's approach is that it provides effective methods and tools particularly well suited to the **cooperative scenario-based elicitation, collection** and validation of requirements coming from **multiple stakeholders**.

The CREWS project points out the “lack of both formal product models and guidelines to support the process of developing scenarios” [7]. It further states that “users request more explicit methodological guidance and more adequate tool support” [7]. This **methodological guidance** is given by [6] where requirements generation through the elicitation of scenarios is described considering four aspects:

- **Purpose:** “why is a specific scenario used?”
- **Contents:** “what is the knowledge expressed in a scenario?”
- **Form:** “which is the best form to express a scenario?”
- **Lifecycle:** “how is a scenario manipulated throughout its life cycle?”

For our purposes, it was decided to use a simplified version of the process described in [6] to extract the requirements from the scenarios focusing on the ones with direct impact (through demands of functionalities expressed) regarding our work for what concerns the design of Reference Model and Reference Architecture specifications on an Open Service Framework that NEXOF-RA Project is all about. This simplified version omits the analysis of stakeholder benefits (i.e. if the stakeholders will be able to fulfil their requirements using the proposed reference architecture). This, because the definition of adequate assessment criteria will be part of the deliverable D10.2 (Assessment criteria) and the analysis and demonstration of the feasibility of the proposed reference architecture will be addressed in the “Proof-of-concept “ track.

Within paper [6] already mentioned above, scenarios are defined as “facts describing an existing system and its environment including the behaviours of agents and sufficient context information to allow discovery and validation of system requirements”. Two types of scenarios are used: “scenario scripts”

that describe system usage; and “scenario structure models” that contain facts about the system environment.

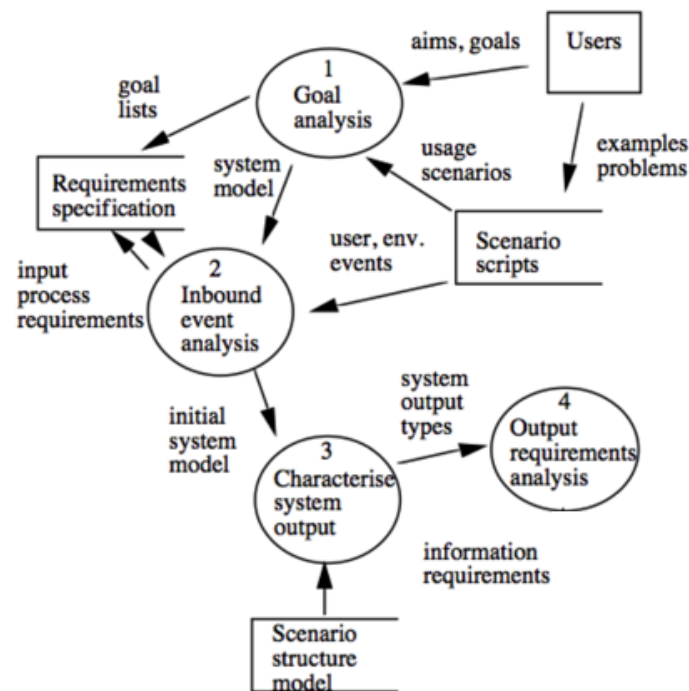


Figure 20: Outline of the method to derive requirements from scenarios (adapted from [6])

Initially the user’s goals are analyzed (step 1) to check whether they are supported the requirements obtained by the state of the art. This creates a first version of the requirements specification (coming from the analysis of scenarios) describing the high-level system processes.

The requirements specification is refined analyzing the dependencies between the inbound events (step 2) described in the scenarios and the requirements functions. This identifies the requirements that have to deal with inbound events coming from users and different types of system environments as well as unexpected events.

Inbound events are defined in scenarios, which define the interaction between users and the proposed system. This analysis will guide decisions about the extent of the automation and the user-system boundary.

System output will be motivated by the analysis of the user goals in step 1 and described in terms of process and outline content in the requirements specification. This enables acceptability and impact of system output on users to be assessed, first by specifying the output in more detail in step 3 and then analyzing the requirements to support user tasks in step 4.

These steps are followed by identifying which stakeholders receive what system output and the use they make out of it.

A2 TEMPLATES FOR QUESTIONNAIRES

A2.1 Template for collecting scenarios

Short name: a short name for the scenario

Detailed, step-by-step scenario description: a textual description of the scenario. Additionally, the domain, the sub domain or the main objective as well as contextual information about the system environment should be specified.

It is strongly suggested to add UML diagrams such as a use case diagram or a sequence diagram to clarify textual descriptions.

Rationale: describes the reasoning and justification for the scenario - that is some important background for why the scenario is what it is. This will be important to help those of us less familiar with the domains to work with the scenarios.

Service consumer: for each user and location (the location indicates where the consumer stays, e.g., customer-home; customer-office; technician-office, etc) the following has to be specified:

Primary requests, problems to solve or needs

Required performances or needs

How and when the user prefers to obtain the service etc. (subscription, pay for use.) and if it is useful or necessary

Service provider: a possible description of who provides the requested service.

Service integrator/developer: a possible description of who integrates or develops the requested service.

Problems and challenges: the specific problems that each scenario addresses or that consumers and providers face.

Architecture and constraints: all involved devices (PC, PDA, etc.), hardware, software, and possible integration with existing applications, how communication is accomplished (GPS, GPRS, Bluetooth, infrared, etc.)

Source: describe how this scenario was produced, from whom it was elicited and possibly it's role within the organization.

Motivation: describe why this scenario has been selected.

A2.2 Template for collecting requirements

Short name: a short name for this requirement.

Requirement type: on of the following

- Functional requirement: what the product has to do or what processing actions it is to take
- Quality attributes: properties that the functions must have, such as performance and usability
- Project constraints: restrictions due to the budget or the available time
- Design constraints: restrictions on how the reference architecture must be designed
- External constraints: restrictions because of business or law restrictions

Related to: the scenario ID to which this requirement is related

Description: the intention of the requirement

Rationale: a justification of the requirement

Domain: the domain and sub-domain of this requirement.

Target:

- Domain-independent requirement
- Cross-domain requirement
- Domain-specific requirement

Originator type

- Service consumer
- Service provider
- Service integrator/developer

Fit criterion: a measurement of the requirement such that it is possible to test if the solution matches the original requirement.

Comments: additional comments if needed.

Conflicts: requirements that cannot be implemented if this on is.

Supporting materials: a pointer to documents that illustrate and explain this requirement.

Priority of accomplishment within the NEXOF-RA project

- Must have
- Should have
- May have