Project no: 269317

nSHIELD

new embedded Systems arcHItecturE for multi-Layer Dependable solutions

Instrument type: Collaborative Project, JTI-CP-ARTEMIS

Priority name: Embedded Systems

D2.2: Preliminary System Requirements and Specifications

Due date of deliverable: M6 – 2012.02.29

Actual submission date: M14 – 2012.10.03

Start date of project: 01/09/2011 Duration: 36 months

Organisation name of lead contractor for this deliverable:

Selex Elsag, SE

Revision Final

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2012)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Authors and Approvals

Authors		Date	Signature
Name	**Company**		
Spase Drakul	THYIA		
Harry Manifavas	TUC		
Konstantinos Fysarakis	TUC		
Dimitris Geneiatakis	TUC		
Alexandros Papanikolaou	TUC		
Konstantinos Rantos	TUC		
Balázs Berkes	S-LAB		
Alfio Pappalardo	ASTS		
Francesco Cennamo	SG		
Arash Vahidi	SICS		
Christian Gehrmann	SICS		
Hans Thorsen	T2Data		
Paolo Azzoni	ETH		
Stefano Gosetti	ETH		
Antonio Abramo	UNIUD		
Mirko Loghi	UNIUD		
Andrea Morgagni	SE		
Renato Baldelli	SE		
Andrea Fiaschetti	UNIROMA		
Reviewed by			
Name	**Company**		
Elisabetta Campaiola	SE		
Approved by			
Name	**Company**		
Luigi Trono	SG		

Applicable Documents		
ID	**Document**	**Description**
[01]	TA	nSHIELD Technical Annex

Modification History		
Issue	**Date**	**Description**
Draft A	29.2.2012	First version
Draft B	26.3.2012	Draft A and review
Draft C	30.3.2012	Updated draft B
Draft D	01.10.2012	Updated according to Budapest decisions
Final	03.10.2012	Updated accoding to project partners review

Contents

Figures

Glossary

Please refer to the Glossary document, which is common for all the deliverables in nSHIELD.

This Page is intentionally left blank

1 Preface

In this document software and hardware requirements for the nSHIELD Architectural Framework for Security, Privacy and Dependability (SPD) are assembled, developed and specified. The structure is inspired but not compliant with the guidelines of the IEEE 830 SRS - Software Requirements Specification Standard (1998) (IEEE Computer Society, 2009).

The requirement specifications in this document are the outcome of Task 2.2 "Multi-technology requirements & specification" in Work Package 2 (WP2). According to the description of work, we adopted the following approach to obtain the requirements:

1. Identification of potential individual users and user groups (stakeholders).

2. Gathering of views, needs and expectations through well-known embedded systems design constraints and semi-structured risk analysis for each application scenario. Thereby, identification of functional and non-functional requirements, ranging from usability and accessibility to system security, privacy, dependability and expected performance.

3. Formal composition of the user requirements and framework specifications.

Requirements limit the range of the valid design, but do not specify any particular design. A requirement specifies any externally visible function or attribute of a system. A design describes a particular subcomponent of a system and/or its interfaces with other subcomponents. The requirement specification has to be correct, unambiguous, complete, consistent, ranked for importance and/or stability, verifiable, modifiable, and traceable (IEEE Computer Society, 2009).

If it is necessary to use the phrase “to be determined” (TBD) in the specification then it is accompanied by a description of the conditions causing the TBD so that the situation can be resolved, and description of what must be done to eliminate the TBD.

Ranking of requirements can be done by the distinction between essential, conditional, and optional requirements. In order to obtain this distinction we used the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL". In this document these key words are to be interpreted as described in RFC 2119 and below reported:

1. MUST: this word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

2. MUST NOT: this phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

3. SHOULD: this word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

4. SHOULD NOT: this phrase, or the phrase "NOT RECOMMENDED", mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

5. MAY: this word, or the adjective “OPTIONAL”, means that an item is truly optional. One designer may choose to include the item because a particular marketplace requires it or because he feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to

> interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

Imperatives of the type defined above are used with care and sparingly. In particular, they are used only where it is actually required for interoperation or to limit behaviour which has potential for causing harm (e.g., limiting retransmissions). For example, they are not used to try to impose a particular method on developers where the method is not required for interoperability. These terms are frequently used to specify behaviour with security and dependability implications. The effects on security of not implementing a MUST or SHOULD, or doing something the specification says MUST NOT or SHOULD NOT be done may be very subtle. It's important to take the time to elaborate the security implications of not following recommendations or requirements as most developers will not have had the benefit of the experience and discussion that produced the specification.

To identify the requirements we used the following instruments:

- review of the project's description of work (nSHIELD Technical Annex);
- semi-structured risk analysis for each application scenario identifying assets to be protected, threats to these assets and security objectives for nSHIELD framework;
- interviews with relevant stakeholders;
- evaluation of written feedback from dissemination activities related to pSHIELD project in order to appraise experiences made during that project;
- requirements evaluation by the project partners.

Special emphasis is given on security, privacy and dependability requirements. For this reason a review of relevant literature regarding embedded systems and their implications for security and dependability concerns was necessary to have consistent requirements.

2 Introduction

The following paragraph describes the aim and the process of requirement specification. Relevant concepts are defined to facilitate a common understanding.

2.1 Purpose

The purpose of this document is to present a detailed description of the new embedded Systems arcHItecturE for multi-Layer Dependable solutions, known as nSHIELD. The document will explain the purpose and features of the system, its interfaces, what the system will do, the constraints under which it must operate and how it will react to external stimuli. Thereby, the requirements describe functional and qualitative needs of the system but it's kept a clear distinction between the system requirements (what the system must do) contained in this document and requirements processing (how to construct the system) that is contained in system architecture design documents.

A System Requirements Specification has traditionally been viewed as a document that communicates the requirements of the potential users and/or customers to the technical community who will specify and build the system. The collection of requirements that constitutes the specification and its representation acts as the bridge between the two groups and must be understandable by both potential users and/or customers and the technical community. One of the most difficult tasks in the creation of a system is that of communicating to all of the subgroups within both groups, especially in one document.

The formalism and language used in this document have the purpose on one hand, to have a description of requirements that helps the intended potential users to express their needs and, as such, the requirements have to be understandable by system user. On the other hand the formalism and language used aim to have a specification of requirements that facilitates the development of the system by providing measurable demands. In this way it narrows down the number of possible designs. Therefore, the specification is understandable by developers and requirements have been formulated in a way that allows the assessment of the fulfilment of requirements.

This deliverable is D2.2 named Preliminary System Requirements and Specification. The final deliverable D2.6 will be Final System Requirements and Specifications which will consider feedbacks from other WPs. The document D2.6 will be refined on the basis of the results of the validation phase and on the detailed description of the application scenarios from WP7.

2.2 Scope of project

The nSHIELD project is, at the same time, a complement and significant technology breakthrough of pSHIELD, a pilot project funded in ARTEMIS Call 2009 as the first investigation towards the realization of the SHIELD Architectural Framework for Security, Privacy and Dependability (SPD). The main goal of the project is to address SPD in the context of Embedded Systems (ESs) as "built in" rather than as "add-on" functionalities.

The leading concept is to demonstrate composability of SPD technologies. Starting from current SPD solutions in ESs, the project develops new technologies and consolidates the ones already explored in pSHIELD.

The nSHIELD approaches SPD at 4 different levels: node, network, middleware and overlay. For each level, the state of the art in SPD of individual technologies and solutions is improved and integrated (hardware and communication technologies, cryptography, middleware, smart SPD applications, etc.). The SPD technologies are then enhanced with the "composability" functionality that was studied and designed in pSHIELD, in order to fit in the SHIELD architectural framework.

In order to verify these important achievements, the project will validate the nSHIELD integrated system by means of relevant scenarios: (i) Railways Security, (ii) Voice/facial recognition, (iii) Dependable Avionic Systems, and (iv) Social Mobility and Networking.

2.2.1 nSHIELD system general definitions

[nSHIELD System] - The nSHIELD system (a whole composed by several parts) is a set of interacting and/or interdependent system components forming an integrated and more complex system.

The nSHIELD system aims to guarantee the following main **taxonomy concepts**: Security, Privacy, and Dependability for itself and for the application scenarios on which it is applied. These attributes are indeed the main goals to be addressed for the new generation Embedded Systems.

The **[nSHIELD functional system]** is organized according to the following layering:

I. **Node Layer** includes the hardware components that constitute the physical part of the system.

II. **Network Layer** includes the communication technologies (specific for the selected scenarios) that allow data exchange among nSHIELD components, as well as the external world. These communication technologies, as well as the networks to which nSHIELD is interconnected, can be (and usually are) heterogeneous.

III. **Middleware Layer** includes the software functionalities that enable the discovery, composition and execution of the basic services necessary to guarantee SPD as well as to perform the tasks assigned to the system (for example, in the railway scenario, the monitoring functionality)

IV. **Overlay Layer** includes the “embedded intelligence” that drives the composition of the nSHIELD components in order to meet the SPD desired level.

[Component/Sub-system] - A component or sub-system is a smaller, self-contained part of a system. In particular for the nSHIELD system the “interacting components” are Embedded Systems.

[Embedded System/Device] - The Embedded System (or Device) is an electronic system (or device) dedicated to a specific and reduced set of functionalities. It could be an integrated circuit that has input, output and processing capabilities or more commonly it is a small programmable chip. The embedded systems are controlled by one or more main processing cores that are typically either microcontrollers or digital signal processors (DSP).

[Asset categories] - An asset can be grouped in two categories: logical and physical assets. Information, services and software are logical assets, whilst human beings, hardware or particular physical objects are physical assets.

[User] – User is any entity internal or external, human or IT that interacts with the nSHIELD system.

A more detailed nSHIELD project description can be found in nSHIELD Technical Annex.

2.3 Overall approach

The requirement specifications assembled in this document are the results of the collective effort of various project members. The requirements were developed in Work Package 2 (WP2) following the approach described below:

1. we give detailed descriptions of the different application scenarios that we consider in the project, Railroad Security (RS), Voice/Facial Recognition (VFR), Dependable Avionic Systems (DAS) and Social Mobility and Networking (SMN). It is important to take these scenarios in consideration in the early development phase of the high level requirements because we have used scenarios as a way to deduce the system requirements.

2. we analyse the different scenarios from security/dependability point of view and identify a set of high level requirements for each of the different scenarios. Shown below an overview of the steps in the system requirements development process for each application scenario. The system requirements development process interfaces with five external and internal agents: the potential users/customers, the nSHIELD Technical Annex, the semi-structured risk

analysis, the pSHIELD lesson learnt past experience and the project partners as technical community. Furthermore, existing technical ES literature was reviewed for further support of the identified requirements. Each of the agents is briefly described in the text below. Next figure shows the interactions among the various agents necessary to develop system requirements.

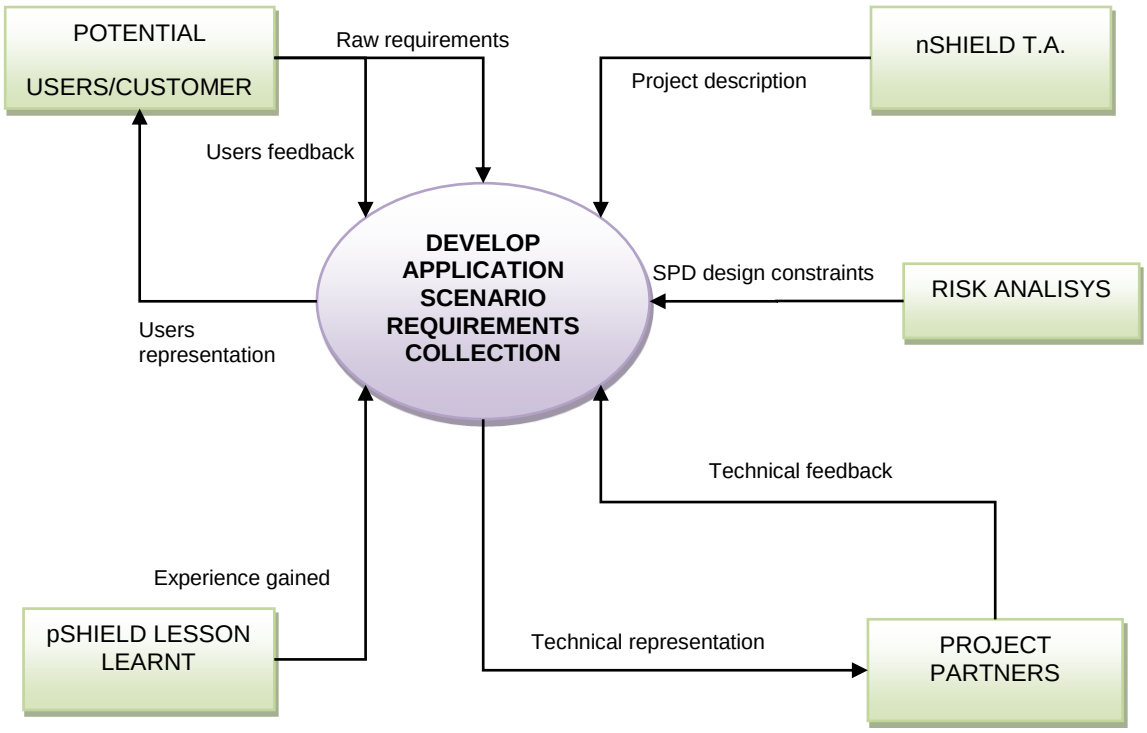


Figure 1: Context for developing application scenario requirements

Potential users/customers:

Customers are one of the keystone elements of the requirements context. They are prime system drivers providing their objectives, needs, or problems to the requirements process. The exchange between customers and requirements developers is:

a) *raw* requirements: prior to the requirements process definition the potential user/ customer has an idea for a system, for a process improvement, or for a problem to be solved. At this point, any initial concept for a system may be imprecise and unstructured. Requirements will often be intermingled with ideas and suggestions for potential designs. These raw requirements are often expressed in initiating documents.
b) *users feedback* includes:
   - information updating the users/customers objectives, problems, or needs;
   - modifying requirements concerning technical communications interchange;
   - identifying new requirements.
c) *users representation*: feedback to the users/customers includes requirements representations and technical interchange or communications clarifying and/or confirming requirements.

nSHIELD T.A.:

The technical annex is a technical document, approved by the Artemis JU, which presents in a manner as clear and concise as possible, all activities, actions and tasks which the partners are committed to undertake in order to fulfil the scientific and research objectives stipulated in the contract.

a) *project description*: technical annex is based upon the description of scientific/technological objectives that must be translated as requirements.

Security requirements identified through technical annex are inserted directly in nSHIELD high level requirements defined in the next step.

Risk Analysis:

The nSHIELD project aims at addressing Security, Privacy and Dependability (SPD) issues in the context of Embedded Systems. Critical systems community and the security critical systems community have independently developed similar techniques for generating dependability requirements (hazard analysis and vulnerability analysis). This focus on what problems might arise and what can be done to avoid, tolerate or reduce the impact of these problems. The techniques can be generalised as risk - based requirements analysis to assess risks that threaten the availability, performance, etc. of the system (A taxonomy of these attributed is introduced in the next chapter).

a) *SPD design constrains:* risks analysis lets us understand the root causes of risks that may affect the system and its environment, driving the process of deriving security, privacy and dependability requirements:
   - requirements that specify how threats to system assets are avoided or mitigated, detected and handled;
   - requirements that specify design constraints that support fault tolerance and recovery;
   - requirements that specify diagnostic and error messages to be generated in the event of a system failure.

Project Partners:

Project partners acting as technical community of the project comprise all those partners involved in the activities of system design, implementation, integration, test, manufacturing, deployment, operations, and maintenance. All elements of the technical community are involved in the requirements development process as early as possible. Early inclusion of the technical community provides a mechanism for the requirements developers to reduce the possibility that new requirements or changes to the original requirements may be discovered later in the system life cycle.

a) *technical representation*: representations of the requirement collection, prepared for the technical community, may include technical interchange or communications that clarify and/or confirm requirements.
b) *technical feedback*: the technical community provides feedback during the project lasting that can cause modification, additions, and/or deletions to the requirements collection. Requirements are refined as necessary to support subsequent life cycle phases of the system. For example, following the requirement phase, a system test plan is developed where individual requirements are allocated to specific tests. This process can reveal requirements that are non-testable, resulting in modification of the requirements to ensure testability. Other feedback from the technical community may provide the system with the most recent features, upcoming technologies, and insight into advanced implementation methods. Although it is desirable to freeze a set of requirements permanently, it is rarely possible to do it. Requirements that are likely to evolve are identified and communicated to all project partners. The core subset of requirements is frozen early. The impact of proposed new requirements is always evaluated to ensure that the initial intent of the requirements baseline is maintained or that changes to the intent are understood and accepted by all partners.

pSHIELD lesson learnt:

The nSHIELD basic concepts were under investigation in the pilot project pSHIELD (contract n° 100204), started on 1st June 2010 and ended on 2011, coordinated by SESM (Finmeccanica Group).

pSHIELD was a reduced R&D project addressing the basic concepts of nSHIELD framework and participated by the core/key partners of the nSHIELD consortium (about 75% of the partners are in common). Its successful conclusion allowed guaranteeing liaisons and input/output exchange.

a) *experience gained*: pSHIELD was a pilot project that wanted to investigate and validate a reduced but still consistent and coherent set of innovative concepts behind the nSHIELD philosophy. For this reason pSHIELD project allowed to nSHIELD partners to gain experience in order to collect and rewrite requirements, to harmonize them with other requirement specifications and to enhance developers' understanding.

1. The high level requirements identified as we described in step 2 are compared and reviewed by the developers of the requirement specifications to ensure comprehensibility and appropriateness, and collected in a list of the complete set of high level requirements for nSHIELD system. Each scenario appears like narrative descriptions of envisioned usage episodes. These scenarios can be called use-case scenarios. In the embedded systems area, use-case scenarios were used in both hardware and software design and in nSHIELD project are used as an input for definition of high level nSHIELD system requirements which are oriented for all possible nSHIELD application scenarios not only for those analysed in this work. All the identified requirements are described in a homogenous structure in order to facilitate design and implementation of the framework with a brief motivation of their inclusion. The next figure shows the adopted philosophy approach.

Figure 2: nSHIELD requirements collecting

2. The list of high level requirements identified in step 3 are broken down into more detailed requirement on the different nSHIELD components defined in the nSHIELD architecture, i.e. node, network layer, middleware and overlay.

3 SPD Concepts taxonomy

3.1 SPD concepts integration

The following taxonomy, derived by relevant technical literature, is introduced to have a common understanding in order to help communication and cooperation among project partners and potential users/customers.

The dependability concept in technical literature was introduced as a general concept including the attributes of reliability, availability, safety, integrity, maintainability, and so on. With ever increasing malicious attacks, the need to incorporate security issues has arisen. Effort has been made to provide basic concepts and taxonomy to reflect this convergence, this because, traditionally there are two different communities separately working on the issues of dependability and security. One is the community of dependability that is more concerned with non-malicious faults. The other is the security community that is more concerned with malicious attacks or faults.

The analysis of a design concerning dependability usually focuses on the accidental or random faults. Modelling these attributes can be performed easily; hence, the ability of optimizing the design can be achieved. There is another fault cause that must be addressed, namely, the malicious or intentional ones. These faults are mainly associated with the security concerns. The root causes of system failure in dependability context (e.g., random accidental failures) are fundamentally different from the root causes of security failure and privacy loss (e.g., intentional attacks).

In order to have a unified approach we focused on threats.

Threats are identified mainly through a risk analysis relevant to each nSHIELD application scenario and are characterized in terms of a threat agent, a presumed attack method, any cause for the foundation for the attack, and identification of the asset under attack. An assessment of risks to security, privacy and dependability qualify each threat with an assessment of the likelihood of such a threat developing into an actual attack, the likelihood of such an attack proving successful, and the consequences of any damage that may result.

To be more precise we consider threats addressed against SPD attributes of assets requiring protection, and they can cause the system flow of fault, error, failure from a dependability perspective or exploit vulnerabilities from a security perspective or cause disclosure from a privacy perspective. All these events lead to a system state that we can define as a system failure, that is, very briefly, what we want to avoid in designing and developing the nSHIELD system.

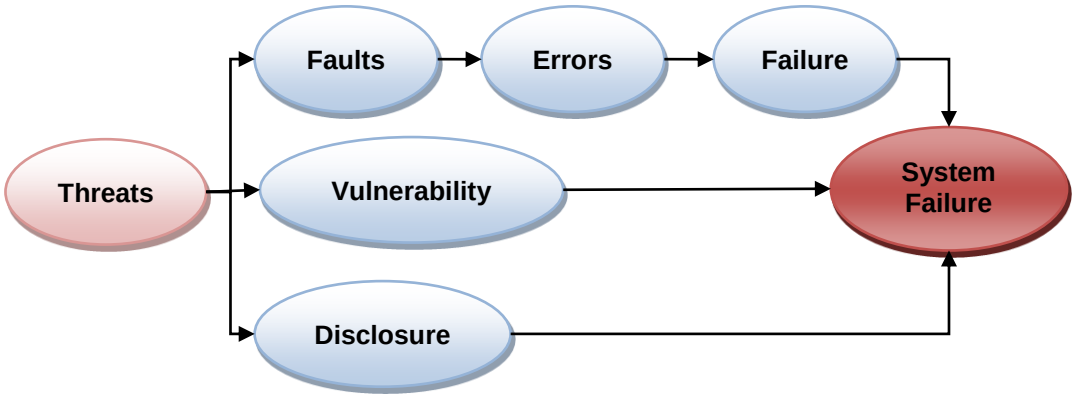


Figure 3: Threats definition

After defining threats we can establish nSHIELD SPD objective and from these derive the required SPD function, in other words the means to face the threats. When we are aware which kind of SPD function we need in our operational environment, we have a valid tool to contribute in the requirements definition useful to meet the security objectives to solve the SPD concerns.

In the following the categorization of SPD attributes and the more common means for their maintenance:

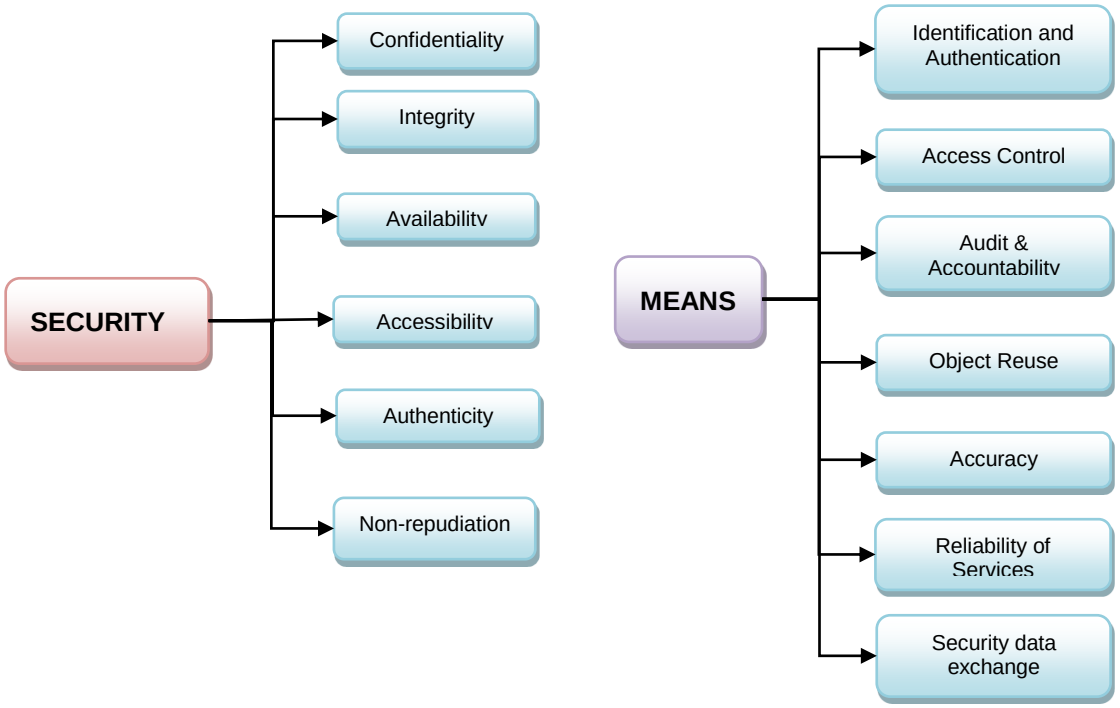


Figure 4: Security/Means definition

We have considered not only the traditional Confidentiality, Integrity and Availability (CIA) but even others attribute to have more granularities in the definition.

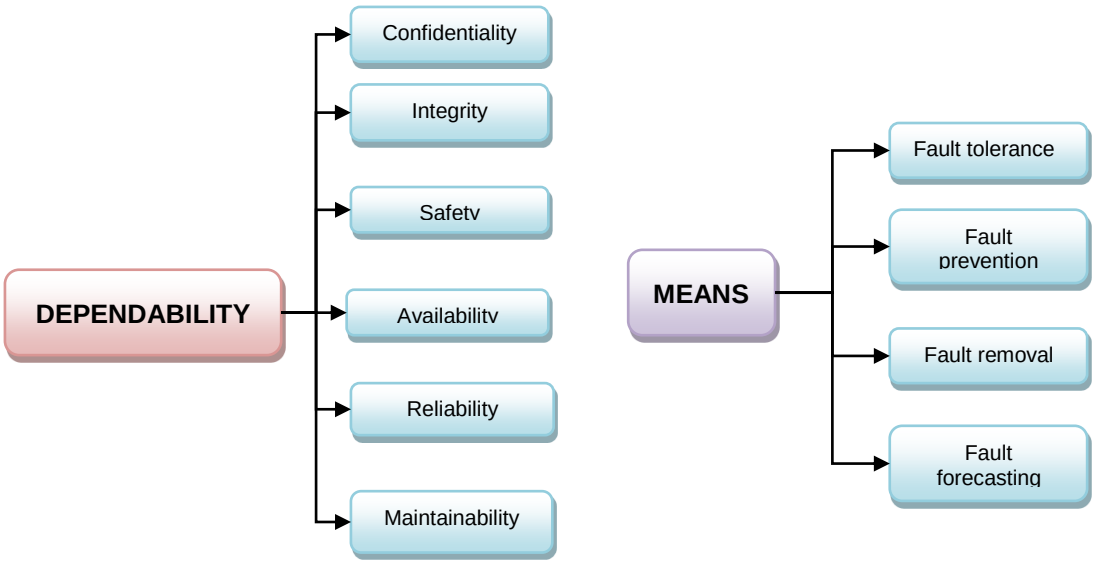


Figure 5: Dependability/Means definition

We have kept the traditional dependability taxonomy.

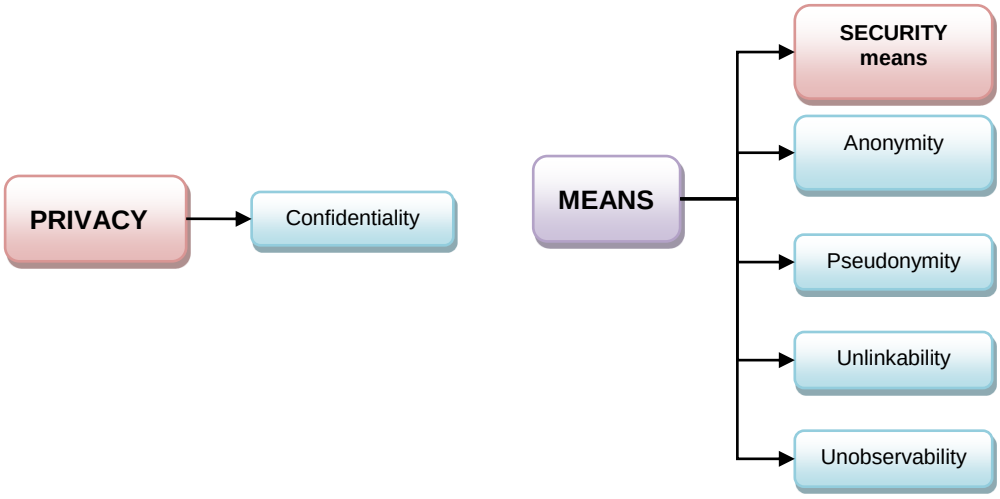


Figure 6: Privacy/Means definition

We have intended privacy as a reason for security rather than a kind of security. For example, a system that stores personal data needs to protect the data to prevent harm, embarrassment, inconvenience, or unfairness to any person about whom data is maintained and for that reason, the system may need to provide data confidentiality service, anyway, as indicated, we identified specific means that can be adopted to assure privacy other than all envisaged for security.

3.2 Taxonomy definition

[Access control] – It is the process of mediating every request of access to nSHIELD assets determining whether the request should be granted or denied according to the security policies established.

[Accessibility] - Ability to limit, to control, and determine the level of access that entities have to a system and how much information they can receive.

[Accountability] - The ability to track or audit what an individual or entity is doing on a system.

[Anonymity] - The function that ensures that a user may use a resource or service without disclosing the user's identity.

[Authentication] - Authentication is any process by which a system verifies the identity of a user who wishes to access it.

[Authenticity] - The property of being able to verify the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.

[Authorization] - Authorization is the process of giving someone permission to access a resource or some information.

[Availability] – The readiness for correct service. It is the automatic invocation of a Web service by a computer program or agent, given only a declarative description of that service, as opposed to when the agent has been pre-programmed to be able to call that particular service.

[Composability] – Is the possibility to compose different (possibly heterogeneous) SPD functionalities (also referred to as SPD components) aiming at achieving in the considered system of Embedded System Devices a target SPD level which satisfies the requirements of the considered scenario.

[Confidentiality] - Property that data or information are not made available to unauthorized persons or processes.

[Configuration] – Configuration is the translation of logical configuration into a physical configuration.

[Dependability] - There are a lot of definitions for dependability. Here are some of them:

- Dependability is the ability of a system to deliver the required specific services that can "justifiably be trusted".
- Dependability is the ability of a system to avoid failures that are more frequent or more severe than is acceptable to the users.
- Dependability is a system property that prevents a system from failing in an unexpected or catastrophic way.

[Error] - Deviation of system behaviour/output from the desired trajectory.

[Failure] - An event that occurs when system output deviates from the desired service and is beyond the error tolerance boundary.

[Fault] - Normally the hypothesized cause of an error is called fault. It can be internal or external to a system. An error is defined as the part of the total state of a system that may lead to subsequent service failure. Observing that many errors do not reach a system's external state and cause a failure, Avizienis et al. have defined active faults that lead to error and dormant faults that are not manifested externally.

[Fault Avoidance (Prevention)] - A technique used in an attempt to prevent the occurrence of faults.

[Fault Containment (Isolation)] - The process of isolating a fault and preventing its effect from propagating.

[Fault Detection] - The process of recognizing that a fault has occurred.

[Fault Forecasting (Prediction)] - The means used to estimate the present number, the future incidence, and the likely consequence of faults.

[Fault Location] - The process of determining where a fault has occurred so a recovery can be used.

[Fault Masking] - The process of preventing faults in a system from introducing errors into the informational structure of that system.

[Fault Removal] - The means used to reduce the number and severity of faults.

[Fault Restoration (Recovery)] - The process of remaining operation or gaining operational status via reconfiguration event in the presence of faults.

[Fault Tolerance] - Ability to continue the performance of its tasks in the presence of faults.

[Graceful Degradation] - The ability of a system to automatically decrease its level of performance to compensate for hardware or software faults.

[Hardware failures] - Hardware Failures include transient faults due to radiation, overheating or power spikes.

[Identification] - Determining the identity of users.

[Information] – Information is measured data, real-time streams from audio/video-surveillance devices, smart-sensors, alarms, etc.

[Integrity] - Absence of malicious external disturbance that makes a system output off its desired service.

[Maintainability] - Ability to undergo modifications and repairs.

[Mean] - All the mechanisms that break the chains of errors and thereby increase the dependability of a system.

[Non-repudiation] - Non-repudiation refers the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.

[nSHIELD Asset] - The nSHIELD assets are information and services.

[Performability] - The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage. It is also defined as a measure of the likelihood that some subset of the functions is performed correctly.

[Privacy] - The right of an entity (normally a person), acting in its own behalf, to determine the degree to which it will interact with its environment, including the degree to which the entity is willing to share information about itself with others.

[Pseudonymity] - The function that ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use.

[Reliability] - Reliability is continuity of correct service even under a disturbance. It is removal (fault) mechanism that permits to the system to record failures and remove them via a maintenance cycle.

[Safety] - The property that a system does not fail in a manner that causes catastrophic damage during a specified period of time.

[Audit] - SPD auditing involves recognizing, recording, storing, and analysing information related to SPD relevant activities. The resulting audit records can be examined to determine which SPD relevant activities took place.

[Software failures] – Software failures include crashes, incompatibilities, computation errors, etc.

[Survivability] - Ability to fulfil its mission in a timely manner in the presence of attacks, failures, or accidents.

[Testability] - The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

[Traceability] - The ability to verify the history, location, or application of an item by means of documented recorded identification

[Unlinkability] - The function that ensures that a user may make multiple uses of resources or services without others being able to link these uses together.

[Unobservability] - The function that ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used.

4 nSHIELD Scenarios

In this chapter a brief description of each scenario will be defined and for each of them will be defined the table containing the definition of the assets to protect and threats correlated to each of them. This matrix will be useful in the last part of the document to justify each defined requirement.

4.1 Railroad Security Scenario

Rail-based mass transit systems are vulnerable to many criminal acts, ranging from vandalism to terrorism. Therefore, physical security systems for infrastructure protection comprises all railway assets as for tunnel, train on board, platform and public areas, external Areas, technical control room, depots, electrical substations and etc…

The objectives are to detect and prevent critical threats as: aggressions and abnormal behaviours, sabotage and terrorism, vandalism and Graffitism, thefts and pickpocketing.

A modern smart-surveillance system suitable for the protection of urban or regional railways is made up by the following subsystems:

1. Intrusion detection and access control:
   - volumetric sensors for motion detection;
   - magnetic contacts to detect illicit doors opening;
   - glass break detectors;
   - microphonic cables for fence/grill vibration detection;
   - active infrared barriers for detecting intrusions inside the tunnels;
2. Intelligent video-surveillance and Intelligent sound detection:
   - advanced cameras with special features;
   - digital video processing and recording, using efficient data compression protocols;
   - video-analytics of the scenes, using computer vision algorithms;
   - Microphones.
3. Dedicated communication network
4. Integrated management system

Distributed smart-sensors are installed along the railway line both in fixed (e.g. bridges, tunnels, stations, etc.) and mobile (passenger trains, freight cars, etc.) locations (Figure 7).

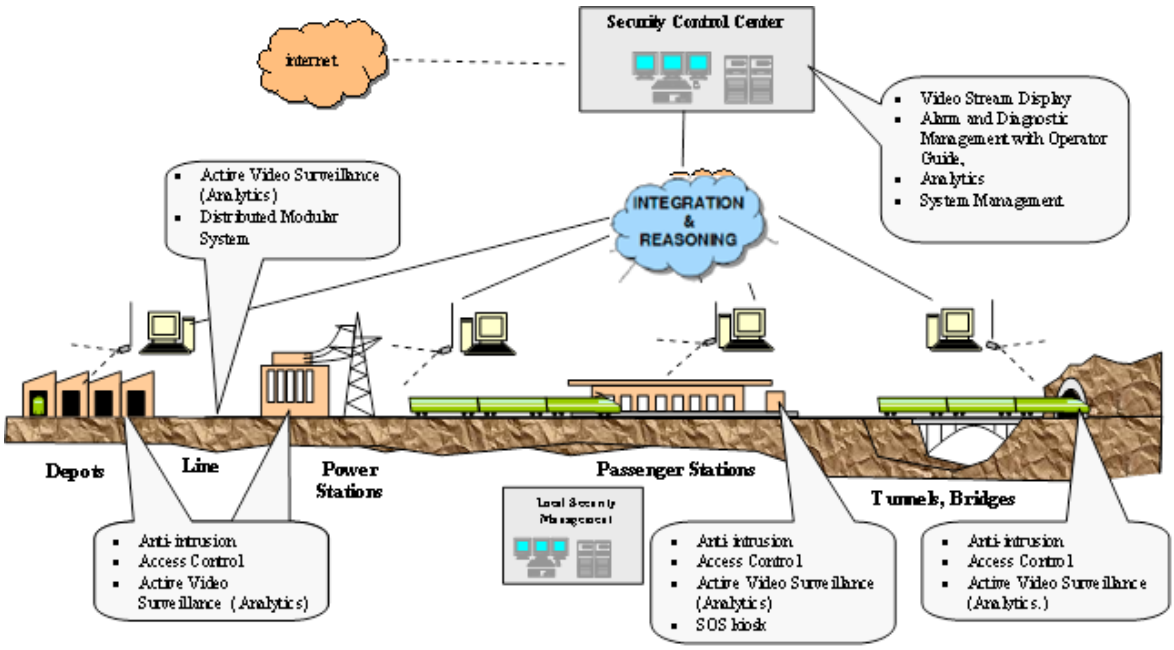


Figure 7: Architecture

They are integrated locally using local wireless infrastructures (e.g. Wi-Fi, ZigBee, etc.) and then data is collected by WSN gateway nodes and transmitted remotely by means of WAN (Wide Area Network). Low/average bandwidth networks are strictly required to transmit alarms to the control centre, which are often already available (like GSM-R for railways) or easy to deploy (like satellite) and provide an extensive coverage of the infrastructure. However, if high-quality video streams from cameras need to be shown to the operators in order to verify the alarm and/or supervise the situation; higher bandwidth is required which can be possible achieved by multiple low bandwidth connections.

This system has already been designed by Ansaldo STS for metro railways, where heterogeneous intrusion detection, access control, intelligent video-surveillance and abnormal sound detection devices are integrated in a cohesive Security Management System (SMS), Figure 8.

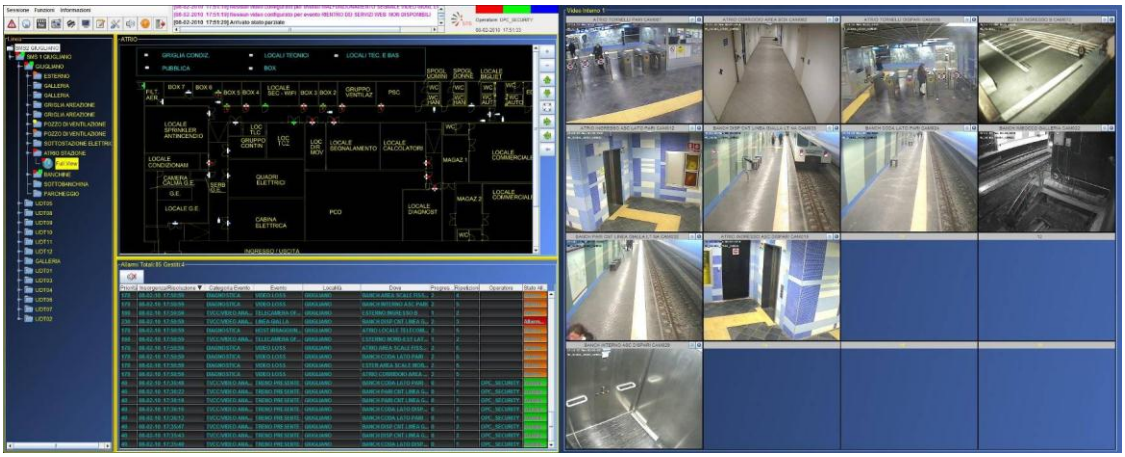


Figure 8: SMS-Security Management System

The core of the SMS consists of a web-based software application featuring a graphical user interface. System architecture is distributed and hierarchical, with both local and central control rooms collecting alarms according to different scopes and responsibilities. In case of emergencies, the procedural actions required to the operators involved are orchestrated by the SMS. Redundancy both in sensor dislocation and hardware apparels (e.g. by local or geographical clustering) improve detection reliability, through alarm correlation, and overall system resiliency against both random and malicious

threats. Video-analytics is essential, since a small number of operators would be unable to visually control the large number of cameras which are needed to extensively cover all the areas needing to be protected. Therefore, the visualization of video streams is activated automatically when an alarm is generated by smart-cameras or other sensors, following an event-driven approach. Very high resolution cameras installed close to the turnstiles are used to automatically detect and store the faces of passengers, whose database can be accessed for post-event investigations. Real-time communication between the on-board and the ground is allowed by a wide-band wireless network.

Currently, the security system described above is highly heterogeneous in terms not only of detection technologies (which will remain such) but also of embedded computing power and communication facilities. In other words, sensors differ in their inner hardware-software architecture and thus in the capacity of providing information security and dependability. This causes several problems:

- Information security must be provided according to different mechanisms and on some links - which are not “open” but still vulnerable to attacks - information is not protected by cryptographic nor vitality-checking protocols;
- Whenever any new sensor needs to be integrated into the system, a new protocol and/or driver must be developed and there is no possibility of directly evaluating the impact of such integration on the overall system dependability;
- New dedicated and completely segregated network links often need to be employed in order not to make the sensor network exposed to information related threats;
- The holistic assurance and evaluation of dependability parameters (e.g. for assessment/certification purposes) would be a very difficult task.

In particular both natural and malicious faults can impact on system availability and indirectly on safety, since the SMS is adopted in critical infrastructure surveillance applications.

The problems mentioned above can be solved by adopting the nSHIELD architecture. Cohesion will be assured by wrapping sensors of any nature with homogeneous embedded hardware and software providing information security, by e.g.:

- Cryptographic protocols
- Vitality checking (heartbeat/watchdog timers based on sequence numbers and time-stamping)

The mechanisms provided by nSHIELD would mitigate the effects on the system of the following logical threats:

- Repetition (a message is received more than once)
- Deletion (a message is removed from a message stream)
- Insertion (a new message is implanted in the message stream)
- Re-sequencing (messages are received in an unexpected sequence)
- Corruption (the information contained in a message is changed, casually or not)
- Delay (messages are received at a time later than intended)
- Masquerade (a non-authentic message is designed thus to appear to be authentic)

Some sensing devices will be converted into smart-sensors by integrating the sensor unit with the nSHIELD processing units (both hardware and software) at the node level. The sensor networks will be integrated by the nSHIELD middleware before data is collected by the SMS and used at the presentation level (integration and reasoning).

4.2 Voice/Facial Recognition Scenario

In the last ten years SPD application scenarios are increasingly introducing the detection and tracking of devices, cars, goods, etc. One of the most important objective of this trend is to increase the intrinsic security, privacy and dependability of the scenario and have more and more services to

improve our life (automatic tolling payment, navigation, traceability, logistics…e.g.).Very frequently, these services and functionalities are based on the identification of a device while we are using it, and today there are many solutions to connect and exchange data from machine to machine (RFID, Wi-Fi, 3G connection and others wired and wireless connections through authentication by code or digital signature…).

In the next future, a similar application scenario is trying to introduce similar services performing the recognition, monitoring and traceability of people.

This scenario is oriented to develop new techniques to analyse physical quantities such as the face image and the voice sound that will be used as a “real-time” person profile that, compared with the one stored in a archive, allows the recognition, monitoring and tracking of that person. From a technical point of view, the requirements of this application scenario introduce new challenges derived from the use of embedded systems to provide recognition, monitoring and tracking services. nSHIELD project, with its SPD hardware infrastructure and software layers, represents the correct answer to these important challenges.

Face Recognition

Face images, which are commonly known as displayed portraits, have been used for many decades for manually verifying the identity of individuals. The recognition procedure required a human operator, being completely manual or semi-automatic, and was based on standard computing systems. More recently, digital face images have been used in a variety of applications ranging from analysis of human actions to computer-assisted face recognition and, the design of new algorithms in conjunction with the evolution of hardware capabilities, allowed the possibility to adopt embedded systems to provide the recognition service. Although photographic formats have been roughly standardized,(consider i.ie. application context like the issuing of passports and driver licenses) there is an ever-increasing need of defining a standard data format for digital face images that would allow interoperability among vendors, service providers and device manufacturers.

The International Standard ISO/IEC 19794-5 (ICAO) is aimed at providing a face image format for recognition applications, which enables the exchange of face image data. Typical applications are:

- human examination of facial images with sufficient resolution to allow a human examiner to ascertain small features such as moles and scars that might be used for verifying the identity;
- human verification of identity through a comparison of facial images;
- computer-automated face identification (one-to-many search);
- computer-automated face verification (one-to-one match).

In order to improve face recognition accuracy, the standard ISO/IEC 19794-5 does not specify only a data format, but it also provides:

- scene constraints (pose, expression etc.);
- photographic properties (lighting, positioning, camera focus etc.);
- as well as digital image attributes (image resolution, image size etc.).

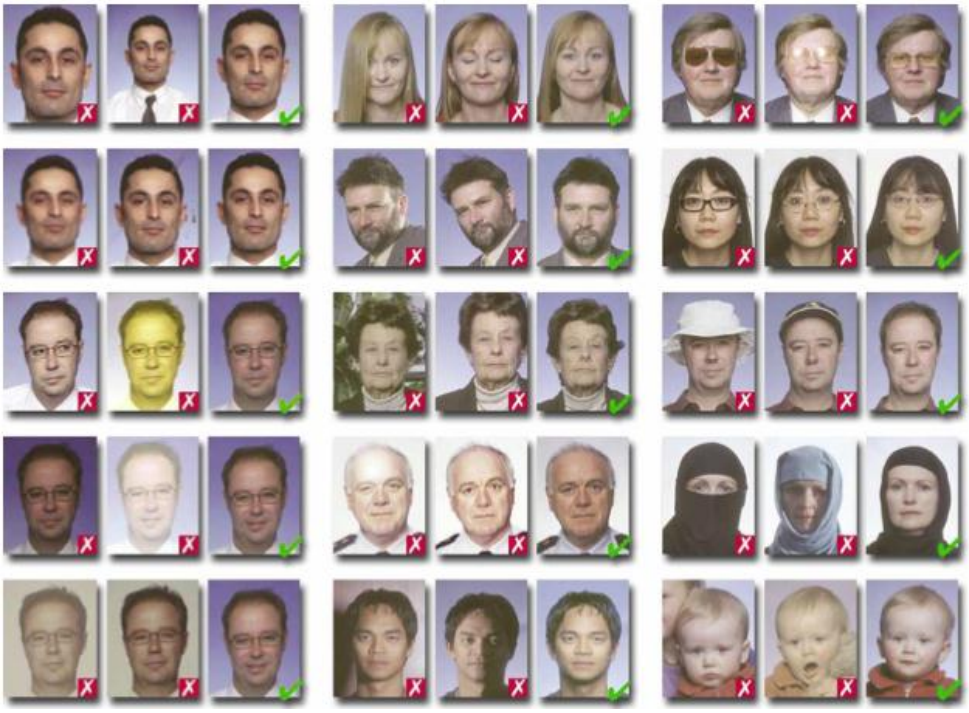


Figure 9: Examples of face recognition results.

This application scenario aims to evaluate the facial image according to the ICAO ISO/IEC 19794-5, which defines the requirements for image geometry and scenery of facial images, and returns Token Face Images and Full-Frontal Face Images that are compliant with the standard. Real-time feedback capabilities must allow capturing and automated enrolment in a faster and more efficient way.

The system will accurately find face and facial features (eyes, mouth, eyebrows etc.) in images with 8-bit grayscale or 24-bit RGB and automatically will extract the following information:

- number of faces,
- origin at upper left,
- centered image,
- pixel aspect ratio,
- resolution,
- width of head,
- length of head,
- head pose,
- position of eyes.

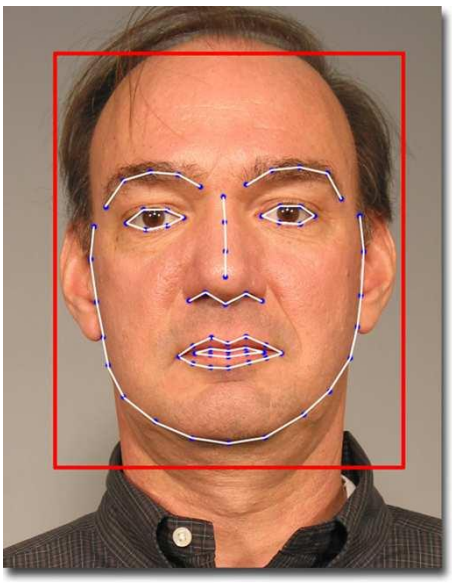


Figure 10: Part of the face recognized.

The system, according to the ICAO ISO/IEC 19794-5 standard, must test the facial area to automatically check the following quality requirements:

- Gray Scale Density,
- Color Saturation,
- Unnatural Color,
- Color Space,
- Video Interlacing,
- Radial Distortion of the Camera Lens,
- Shadows Over the Face,
- No Over or Under Exposure,
- Focus and Depth of Field,
- Eye Glasses,
- Red Eye,
- Eye Patches,
- Shadows in Eye-Sockets,
- Mouth Expression,
- Hot Spots,
- Concealment by hat,
- Background shadows,
- Uniformity of background.

Finally, the system must support (as far as both reading and writing functionalities are concerned) the CBEFF Patron Formats A-C, with ISO/IEC 19785-1 and ANSI INCITS 398-2005 interchange files, and its outputs must be fully compliant with the ISO/IEC 19794-5 standard.

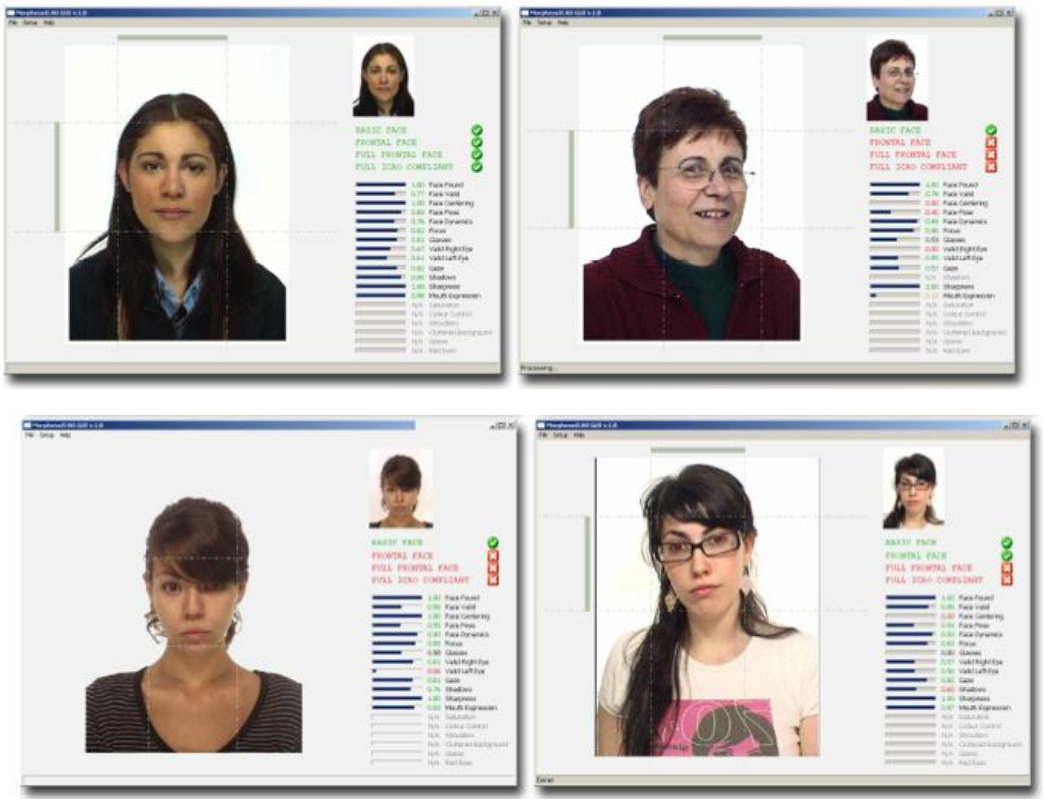


Figure 11: Example of a face recognition software.

Voice Verification

Our voices are unique for each person (including twins), and cannot be exactly replicated. Speech includes two components: a physiological component (the voice tract) and a behavioural component (the accent). It is almost impossible to imitate anyone's voice perfectly. Voice recognition systems can discriminate between two very similar voices, including twins.

The voiceprint generated upon enrolment is characterised by the vocal tract, which is a unique a physiological trait. A cold does not affect the vocal tract, so there will be no adverse effect on accuracy levels. Only extreme vocal conditions such as laryngitis will prevent the user from using the system.

During enrolment, the user is prompted to repeat a short passphrase or a sequence of numbers. Voice recognition project aims to test various audio capture devices (microphones, telephones and PC microphones). The performance of voice recognition systems may vary depending on the quality of the audio signal.

To prevent the risk of unauthorised access via tape recordings, the user must ask to repeat random phrases. This precaution increases the SPD level of the system.

Another goal of the application scenario is to improve the most important weaknesses of voice biometric systems: the high false non-matching rates. Voice verification can be used for government, healthcare, house arrest and probation-related authentication.

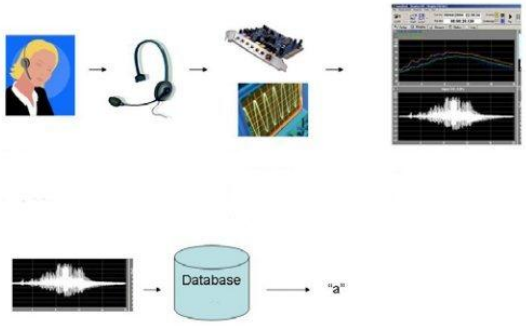


Figure 12: The voice recognition process.

Voice biometrics works by digitizing a profile of a person's speech to produce a stored model voice print, known as template. Biometric technology reduces each spoken word to segments composed of several dominant frequencies called formants. Each segment has several tones that can be captured in a digital format. The tones collectively identify the speaker's unique voice print. Voice prints are stored in databases in a manner similar to the storing of fingerprints or other biometric data.

To ensure a good-quality voice sample, a person must recite some sort of text or pass phrase, which can be either a verbal phrase or a series of numbers. The phrase may be repeated several times before the sample is analysed and accepted as a template in the database. When a person speaks the assigned pass phrase, certain words are extracted and compared with the stored template for that individual. When a user attempts to gain access to the system, his or her pass phrase is compared with the previously stored voice model. Some voice recognition systems do not rely on a fixed set of enrolled pass phrases to verify a person's identity. Instead, these systems are trained to recognize similarities between the voice patterns of individuals when the persons speak unfamiliar phrases and the stored templates.

Voice verification technology uses the different characteristics of a person's voice to discriminate between speakers. These characteristics are based on both physiological and behavioural components. The physical shape of the vocal tract is the primary physiological component. The vocal tract is made up the oral and nasal air passages that work with the movement of the mouth, jaw, tongue, pharynx and larynx to articulate and control speech production. "The physical characteristics of these airways impart measurable acoustic patters on the speech that is produced". The behavioural component is made up of movement, manner, and pronunciation.

The combination of the unique physiology and behavioural aspects of speaking enable verification of the identity of the person who is speaking. This voice verification project will work by converting a spoken phrase from analog to digital format and extracting the distinctive vocal characteristics, such as pitch, cadence, and tone, to establish a speaker model or voiceprint. A template must be generated and stored for future comparisons.

Voice verification systems can be used to verify a person's claimed identity or to identify a particular person, increasing in this way the SPD level of the application context in which they are used. These functionalities are often used where voice is the only available biometric identifier, such as over the telephone. Voice verification systems may require minimal hardware investment, as most personal computers already contain a microphone, and are perfectly suitable for embedded systems. The downside to the technology is that, although advances have been made in recognizing the human voice, ambient temperature, stress, disease, medications, and other physical changes can negatively impact on automated recognition.

Voice verification systems are different from voice recognition systems although the two are often confused. Voice recognition is used to translate the spoken word into a specific response, while voice verification verifies the vocal characteristics against those associated with the enrolled user. The goal of voice recognition systems is simply to understand the spoken word, not to establish the identity of the speaker. This application scenario, being focused on SPD aspects, considers only the voice verification.

4.3 Dependable Avionic System Scenario

The driver of change in avionic system scenarios is the continuing rapid advance of electronics technology, computers, sensors, displays, data-buses etc. There are examples of avionics computers introduced less than a few years ago that are now available under half the size, weight, power consumption, but with considerably enhanced performance and functionality. Avionics components and communication standards are now being replaced increasingly by commercial ones:

- microprocessors, microcontrollers;
- data-buses (e.g. Ethernet, CANbus);
- flat panel AMLCD displays and interfaces (e.g. OpenGL graphics language).

These trends are consolidated in both the civil and military aviation markets. The rapid and continuing advance of electronic technology is constantly driving down the cost of hardware, as the number and cost of the components used falls and much more functionality is achieved with less and less hardware.

Associated with the enhanced capability afforded by the technology, the functionality of avionics systems has continued to rise

- Fly-By-Wire flight controls;
- Flight Management System (FMS),
- Full glass cockpits, large multi-function displays;
- Future Air Navigation System (FANS) capability to operate in the new air traffic management environment;
- Passenger entertainment systems and commercial/business services;
- On-Board central maintenance computers and electronic documentation.

Much effort is spend to ensure that application software can be reused on different hardware in order to avoid the high cost of new software for the application being hosted on different processors, for example, in the event that during the life-cycle of the product the processor becomes obsolete or has insufficient capability to support growth in required functionality.

Hardware independent application software requires embedded software Operating System (OS) or Executive which provides a generic interface to the application code and which translates it for the particular processor and hardware architecture being used.

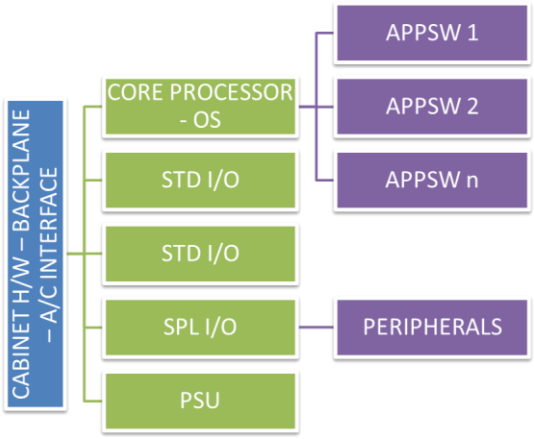
As technology has advanced so there has been a continuing trend of avionics integration.

In the past most integration has been concerned with processes or functions which were already interdependent and by so doing savings could be made in interface hardware, aircraft wiring, power supplies, etc., with little risk to the certification of the system. This is usually referred to as “vertical” integration. The integration of inertial sensors with computers to produce inertial reference systems (IRS) is an example of vertical integration.

In other cases separate processors have been co-located into a single box because they can share the same I/O hardware on which they both depend, thereby eliminating one set of I/O hardware and simplifying the aircraft wiring.

More latterly there is a trend to go one step further by the integration of largely unrelated avionics functions onto common processors. This may be referred to as “horizontal” integration. There are significantly higher risks because, whilst additional hardware resource may be saved, there are added complexities to provide “equivalent independence” or partitioning within the processing platform. Partitioning is used to ensure that malfunctions within one application (function) cannot affect the others, or that modifications made to one may be certified without the need to revalidate the others. A further complication may be that additional levels of hardware redundancy and associated monitoring

and configuration management facilities may be needed to offset the risk of failures within the shared processor causing simultaneous loss of all the otherwise unrelated application functions. The trade between savings of hardware (e.g. processor modules) on the one hand, and the escalation in cost to achieve acceptable levels of segregation and integrity on the other, needs to be carefully weighed. This is a trade that would appear to be more difficult to support as the hardware proportion of overall cost continues to fall with time.



A primary goal of Integrated Modular Avionics (IMA) is to establish the application of a "standard" set of hardware modules, directly line-replaceable, encompassing as much of the total avionics suite as possible.

The main drivers for aircraft and systems architecture are:

- reduction in installation complexity – wires, connectors;
- robustness of peripheral interfaces to noise, improved RFI/EMC immunity;
- reduced uncertainties in identifying fault location, i.e. whether fault is in the peripheral, the computer or the wiring;
- simplification in data interfaces; by incorporating special peripheral processing at the "point of action" and reducing communication data flow to higher-order parameters;
- architectural flexibility. The I/O interface is available on aircraft-level data-buses for direct use elsewhere;
- enabling the interface to be independent of the technology of the peripheral;

As electronics are used at the peripherals and serial data-buses provide the interface, the systems become "digital" from end to end.

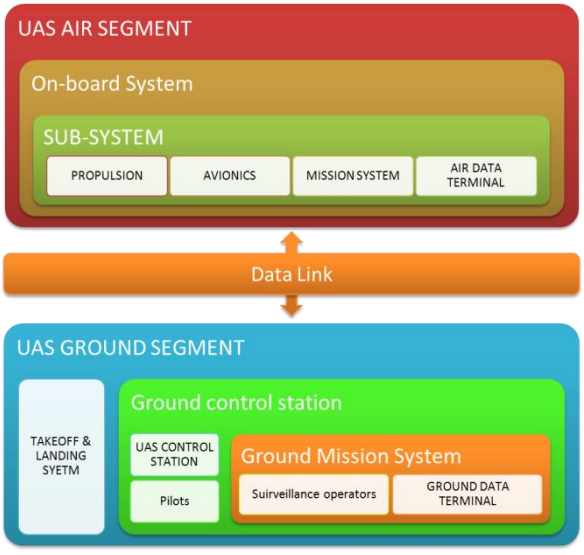
The greater use of smart peripherals and remote data concentrators will also significantly diminish the need for analogue and discrete I/O modules within IMA cabinets and computer LRUs.

The goal of standard, reusable and interchangeable modules is central to the concept of IMA. By rigorous definition and control of each module, both hardware and software, and of its interfaces.

The application scenario for the nSHIELD project is the Avionic System for a Unmanned Aircraft System.

To preserve the main functionalities represents an example application of great interest for the Avionic System. In particular, in this use case, the following requirements have to be fulfilled:

- Secure and dependable handling of on-board computing and sensor processing capabilities
- Secure and dependable Ground Operator sensor control terminals
- The nSHIELD philosophy will be applied to preserve the data for the following components
- AIR/GROUND DATA TERMINAL: the nSHIELD solution will guarantee that the data exchanged between the UAV and Ground Station will be preserved by



anomalous interface. (Security paradigm oriented)

- Avionics Unit: the nSHIELD solution will guarantee that the data acquired by sensors are protected against the possible corruption. (Dependability paradigm oriented)
- Mission System: the nSHIELD solution will define solution for easy integration of new sensors and/or replaced old version (Dependability paradigm oriented)
- Ground Control System: the nSHIELD solution will be able to manage the different access to the Unmanned System for the Mission Operators and Pilot (Security and Privacy paradigm oriented)

4.4 Social Mobility and Networking Scenario

The project nSHIELD is considering Social Mobility and Networking (SMN) as a scenario of humans that are moving from one place to other by walking, using public means of transport (train, bus, etc.) or personal one, such as a bicycle, car, etc., and they desire to communicate with other persons or things (here we are referring to the future **I**nternet of **T**hings and **H**umans, named hereafter ITH). There will be two different sub-scenarios, indoor and outdoor. Indoor sub-scenario is related to social mobility of people inside the houses, buildings, etc. Outdoor sub-scenarios is related to social mobility of people on the streets in cites/towns, villages, highways, and other roads. In the scope of SMN in the nSHIELD project we are aiming to present this as a common R&D area in which SPD plays an extremely important role for a successful implementation of SMN scenario. Figure 13 illustrates how different fields will interact jointly in this scenario. There are four fundamental overlapping areas:

1. New Street and Building Lights (SBLs),
2. Public and Private Transportation Means PPTMs,
3. The future Internet of Things and Humans (ITHs), and
4. Ubiquitous and Pervasive Computing (U&PC)

that have a common one, i.e., SPD aspects that are considered in details in this project in which a common nSHIELD system architecture is composed of four layers: node, network, middleware and overlay, that play an important role in the implementation of SMN scenario. In SMN scenarios a focus will be given to intelligent systems and intelligent ICT, since intelligent street lighting is maturing and providing cost-effective approach to manage municipal street lighting. Even though several attempts that have tried to merge the two worlds could not reach the masses, experts expect that future mobile social networking systems possibly even exceed the success of their Internet bound counterparts. We believe that two key features are the user‘s permanent reachability and location awareness, which is called P3 (Peer-to-Peer-to-Place).

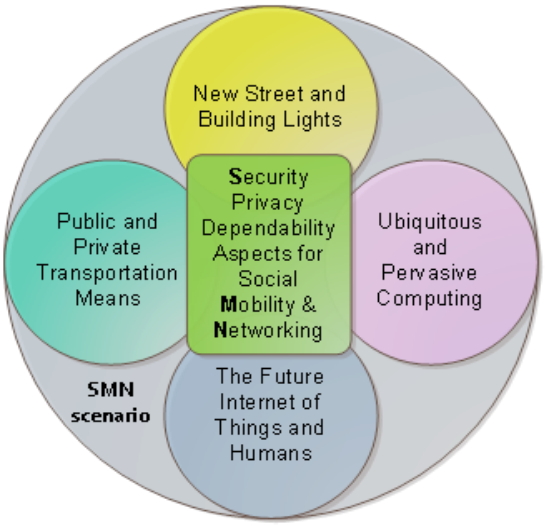


Figure 13: Social Mobility and Networking Scenario.

For that we need to integrate the old and new communication infrastructures. This lead to the creation of large and complex networks called *real-life networks* that include: electrical power grid, World Wide Web, the Internet backbone, collaboration and citation networks, and airline connection networks. Step-by-step, we are moving into a world of ubiquitous and pervasive computing (U&PC), Security, Trust, Privacy and Dependability (STPD) issues will be in focus more than ever before. Sensor networks have been used in numerous applications such as remote sensing, environmental monitoring, habitat, human activity, health monitoring, industrial appliances monitoring, medical applications, space and underwater phenomena monitoring, home and building automation and so on. Capturing sensory data from Body Area Networks (BANs), or Body Sensor Networks (BSNs) and sending it to social networks is a challenging task, because it requires a number of distributed networks to work together seamlessly. Disseminating the sensory data in real time to one's community of interest such as family, friends, family doctors, and emergency services is very crucial and critical. Therefore, SPD aspects are needed in such a complex networks and environments.

First, **PPTMs** play important role in social life of people. For example, bus, tram, trains, ships, cruisers, aircrafts, etc., are places where an individual can interact with other people and surrounding environments. The same when an individual is using his/her bicycle, motorcycle, car, camions, boats/yachts, etc. For example, the railways scenario is easily integrated in SMN scenario. Second, the future **ITH** is one of the most important areas of research in FP7. Additionally, internet based social networking services have experienced an enormous growth over the past years. Simultaneously to the social networking services' triumphant advance, mobile devices in general and smart phones in particular have rapidly penetrated the consumer market. Recent phones do not only exhibit considerable computing resources but also feature means for Internet access as well as wireless short distance communication such as Bluetooth and Wi-Fi. They seem to be the perfect platform to combine the market potential of traditional social networking services and the success story of mobile devices. Even though several attempts that have tried to merge the two worlds could not reach the masses, experts expect that future mobile social networking systems possibly even exceed the success of their Internet bound counterparts. Third, **U&PC** devices are very tiny - even invisible - devices, either mobile or embedded in almost any type of object imaginable, including cars, tools, appliances, clothing and various consumer goods - all communicating through increasingly interconnected networks. Fourth, **SBLs** will have impact on streets and building lighting, since it is a European directive acting as new European standard. The state-of-the art of the street lighting and the future expectation in the partner-countries is enormous.

5 High Level Requirements for Scenarios

5.1 Railway Scenario

{REQ_RW01 Real-time data availability
Surveillance data shall be provided in real-time.

Motivation: video surveillance video must be available in central surveillance nodes in real time.

}

{REQ_RW02 Transmitted information integrity
Integrity of data transmitted over wireless or cable networks shall be granted.

Motivation: An attacker shall not be able to change security critical information during its transmission.

}

{REQ_RW03 Transmitted information confidentiality
Confidentiality of data transmitted over wireless or cable networks shall be granted.

Motivation: An attacker shall not be able to get access to clear text critical information during its transmission.

}

{REQ_RW04 Peer authentication
All communication peers shall be securely identified.

Motivation: An attacker shall not be able to impersonate a legitimate unit in the system such as surveillance camera or sensor.

}

{REQ_RW05 User authentication
All users in the system must be authenticated.

Motivation: A non-legitimate user shall not be able to get access to any data and function in the system such as images, sensors control system etc.

}

{REQ_RW06 Authorization
All data access must be subject to access control.

Motivation: An inside attacker shall not be able to get access to data that he or she does not have the right to access to.

}

{REQ_RW07 Secure node deployment
A new node shall be identified as a trusted node prior to be accepted.

Motivation: A new node such as a sensor must be identified as a trusted node prior to that it is accepted in the system such that an attacker not will be able to insert malicious units into the system.

}

{REQ_RW08 Secure software upgrade
All system components must provide a secure and safe software upgrade process.

Motivation: A software upgrade of software component in nodes such as a video camera should not imply a risk for that hostile code is installed on the device.

}

{REQ_RW09 Denial of service
All network nodes should provide methods to prevent or reduce the impact of denial of service attacks.

Motivation: An attacker should not be able to destroy the data delivery such as sensor data by launching different type of denial of service attacks such as network flooding and “ping to death” attacks.

}

{REQ_RW10 Secure execution
A system node should provide an isolated protected execution environment that can offer secure execution for the most security sensitive computing tasks.

Motivation: If a software system such as a software component in the SMS or in a video camera is infected by hostile software (for example due to viruses in software installation packages) this should not jeopardize the most security critical functions and parameters handle by the system.

}

{REQ_RW11 Secure boot
All nodes should provide a secure boot process.

Motivation: It shall not be possible for an attacker to inject hostile code into surveillance sensors nodes by installing malicious software on non-volatile memory or by forcing boot from an external interface with malicious code.

}

{REQ_RW12 SPD level assignment
Each component or set of component in the system must have an assigned SPD level.

Motivation: All software and hardware components that constitute a system node such a railway sensor must have gone through a SPD evaluation and been assigned a SPD level. This must be done such that controlling units in the system can make access decisions based on the security quality of the sensor component and avoid that low security quality nodes getting access to sensitive system functions while still allow them to function for less security critical purposes.

}

{REQ_RW13 SPD level identification
It must be possible to securely verify the assign SPD level of all security critical components in a node.

Motivation: In order for a system controlling unit to make access decisions based on security level, it must be possible to securely verify the security level of the components of a node such a surveillance sensor.

}

{REQ_RW14 Software failure mitigation
The system should provide automatic means to recover from a situation when a node fails due to software failure.

Motivation: If a node stops functioning, this should automatically be detected by the SMS and a secure process for bringing the node back into a functional state should be in place.

}

{REQ_RW15 Hardware failure mitigation
The system should be robust against hardware faults.

Motivation: It shall be possible by the SMS to detect hardware fault and to replace non-functional hardware units in order to make nodes such as sensors to function again. This is only possible as long as the communication with the unit.

}

{REQ_RW16 Data backup
The system shall provide data backup for most sensitive data.

Motivation: Video and sensor evidence of crime must be kept securely not to have the risk of being lost.

}

5.2 Voice/Facial recognition Scenario

{REQ_VF01 Secure identification though face recognition
The system should allow secure identification of end users through facial recognition.

Motivation: Due to user convenient reasons, it should be possible to use face recognition as a mean to securely identify end-users in the system

}

{REQ_VF02 Secure identification though voice recognition
The system should allow secure identification of end users through voice recognition.

Motivation: Due to user convenient reasons, it should be possible to use face recognition as a mean to securely identify end-users in the system

}

{REQ_VF03 Protection of biometric data and metadata storage
Confidentiality of biometric data and metadata storage should, due to security and privacy reasons, be protected.

Motivation: It shall not be possible for an intruder to break the secure face of voice identification scheme by extracting stored biometric data from the system.

}

{REQ_VF04 Biometric data privacy
The original voice or image signal must be deleted after analysis and cannot be transferred in any way outside the embedded device used for the acquisition and recognition.

Motivation: Illegal usage or spread of biometric data shall be avoided due to privacy reasons.

}

{REQ_VF05 Biometric data and metadata confidentiality during transmission
Biometric data and metadata shall be encrypted in transit.

Motivation: In order to prevent an attacker to gain access to original biometric data, the date signals from the video or audio sensors shall be encrypted in when sent to the analysing unit.

}

{REQ_VF06 Face recognition quality
Secure identification based on face recognition shall of such high quality that the risk of false identification is very low.

Motivation: If there is a high probability of false detection, so is the risk that an attacker manage to break the secure identification scheme.

}

{REQ_VF07 Biometric data privacy
The original voice or image signal must be deleted after analysis and cannot be transferred in any way outside the embedded device used for the acquisition and recognition.

Motivation: in order to ensure privacy, any photo or voice record of people must be deleted immediately after being processed by the recognition algorithms and must not leave in any way the embedded device in which it is temporarily stored.

}

{REQ_VF08 Biometric metadata privacy
The metadata obtained from the analysis of voices and images must be used respecting privacy.

Motivation: although the metadata obtained from voices and images are less sensible than their analog sources, they always represent important information, especially when contextualized, and for this reason must be treated respecting privacy.

}

{REQ_VF09 Collaborative process
In the case the recognition system is not completely autonomous; the user may collaborate to the recognition result.

Motivation: in certain cases the recognition system is not capable to operate in a fully autonomous way, but requires some condition to be respected in order to work properly. The user is requested to collaborate in order to facilitate the occurrence of these conditions: i.e. the user must stand in front of the camera in a particular position, he cannot wear a hat, etc..

}

{REQ_VF10 System fault tolerance
The components of the recognition system shall use periodic keep alive messages, independent from the acquisition process, in order to inform each other of any fault occurred in the system.

Motivation: i.e., in order to ensure the dependability of the overall recognition system, the embedded device shall send period keep alive message to the other components of the recognition system.

}

{REQ_VF11 Recognition process determinism
The recognition process could not be deterministic, but the overall recognition system must provide a reliable and secure answer in terms of recognition and identification. This can be obtained using specific thresholds that define the boundaries between a positive and a negative answer.

Motivation: the recognition process is self-learning and requires a training procedure. For its intrinsic nature it doesn't provide a binary result in terms of recognition but only a matching score

and could be non-deterministic. In order to guarantee the reliability of the recognition process and the overall system, it must be "forced" to be deterministic.

}

{REQ_VF12 Onsite recognition
The recognition (not the identification) of human faces and voices in the input stream must be entirely performed by the embedded system on site. On the contrary, the user identification shall be performed onsite or remotely. The analog sources of voices and images cannot leave in any way the embedded system: only the results of the recognition process (metadata) can be used by the other components of the recognition system.

Motivation: if the features detection and recognition of face and voice is performed entirely onsite there is no need to send these sensible data to another computing unit over the network, and this increase security and privacy.

}

{REQ_VF13 Transmitted Biometric information integrity
Metadata transmitted over wireless or cable networks shall be integrity protected.

Motivation: it should not be possible to change biometric metadata that contain security critical information used for user identification during any kind of data transmission.

}

{REQ_VF14 Secure execution
The embedded system in charge of recognition should provide an isolated and protected execution environment that can guarantee the security during the recognition and identification process.

Motivation: it should not be possible to interfere in any way with system during the recognition and identification process.

}

5.3 Avionics Scenario

{REQ_AV01 Real-time data availability
Critical data shall be provided in real-time.

Motivation: Critical Data (Avionic control data, Raw Radar and Video data, sensor information) must be provided according to strict real-time constraint. If not, there is a risk that air crafts can be lost or severely damaged or system operation safety can be compromised.

}

{REQ_AV02 Transmitted information integrity
Data transmitted over wireless or cable networks shall be integrity protected.

Motivation: An attacker shall not be able to change security critical information such as video surveillance data or audio data during transmission. In addition, due to safety reasons, data integrity must be preserved as transmission.

}

{REQ_AV03 Transmitted information confidentiality
Data transmitted over wireless or cable networks shall be confidentiality protected.

Motivation: An attacker shall not be able to get access to clear text information such avionic sensor or control data.

}

{REQ_AV04 Peer authentication
All communication peers shall be securely identified.

Motivation: An attacker shall not be able to impersonate a legitimate such as the UAS ground segment or the UAS air segment.

}

{REQ_AV05 User authentication
All users in the system must be authenticated.

Motivation: A non-legitimate user shall not be able to get access to any data and function in the system such as UAS air control.

}

{REQ_AV06 Authorization
All data access must be subject to access control.

Motivation: An inside attacker shall not be able to get access to data that he or she does not have the right to access to. For instance unauthorized UAS air control.

}

{REQ_AV07 Secure node deployment
A new node shall be identified as a trusted node prior to be accepted.

Motivation: A new node such as a sensor must be identified as a trusted node prior to that it is accepted in the system such that an attacker not will be able to insert malicious units into the system.

}

{REQ_AV08 Secure software upgrade
All system components must provide a secure and safe software upgrade process.

Motivation: A software upgrade of software component in nodes such as sensors or control units should not imply a risk for that hostile code is installed on the device.

}

{REQ_AV09 Secure execution
A system node should provide an isolated protected execution environment that can offer secure execution for the most security sensitive computing tasks.

Motivation: Integration of multiple functions into the same node shall be possible without having the risk that less security critical tasks compromise more security critical.

}

{REQ_AV10 Secure boot
All nodes should provide a secure boot process.

Motivation: It shall not be possible for an attacker to inject hostile code into air control nodes by installing malicious software on non-volatile memory or by forcing boot from an external interface with malicious code.

}

{REQ_AV11 SPD level assignment
Each component or set of component in the system must have an assigned SPD level.

Motivation: All software and hardware components that constitute a system node such aircraft sensors must have gone through a SPD evaluation and been assigned a SPD level. This must be done such that controlling units in the system can make access decisions based on the security quality of the sensor component and avoid that low security quality nodes getting access to sensitive system functions while still allow them to function for less security critical purposes.

}

{REQ_AV12 SPD level identification
It must be possible to securely verify the assign security level of all security critical components in a node.

Motivation: In order for a system controlling unit to make access decisions based on security level, it must be possible to securely verify the security level of for instance on-board computing and sensor processing capabilities.

}

{REQ_AV13 Software failure mitigation
The system should provide automatic means to recover from a situation when a node fails due to software failure.

Motivation: If a node such as an aircraft sensor stops to function, this should automatically be detected by the system and a secure process for bringing the node back into a functional state should be in place.

}

{REQ_AV14 Hardware failure mitigation
The system should be robust against hardware faults.

Motivation: It shall be possible by the system to detect hardware fault and to automatically switch from non-functional hardware to functional ones in real-time.

}

{REQ_AV15 Data backup
The system shall provide data backup for most sensitive data.

Motivation: Due to air safety reasons, data essential for the system to function must be backed-up.

}

{REQ_AV16 Data storage redundancy
Critical system data shall be secured through redundant storage

Motivation: Due to air safety reasons, data essential for the system must be stored in multiple copies and in multiple storage units.

}

{REQ_AV17 Data storage integrity
Safety and/or security critical data shall be integrity protected at storage

Motivation: Storage corruption shall not make critical air system to fail and it shall not be possible for an attacker to destroy critical system functions through replacing or changing security/safety critical data.

}

{REQ_AV18 Data storage confidentiality
Confidentiality of critical data storage shall be protected.

Motivation: It shall not be possible for an attacker with physical access to storage units to get access to classified information.

}

{REQ_AV19 Data Acquisition Integrity

The integrity of each acquired Data from sensors and/or equipments shall be guaranteed through a defined acquisition check.

Motivation: In order to not compromise the mission, all the data exchanged among the avionic equipments must be error free and trusted.

}

{REQ_AV20 Information confidentiality
Mission Captured Data shall be accessed only by authorized users.

Motivation: Based on the mission type, the data captured by an UAV, can have a different level of accessibility. An authentication mechanism is required.

}

{REQ_AV21 Data Masking
The inbound and outbound system data shall be masked / unmasked through the adoption of a dedicated procedure.

Motivation: Based on the context, scenario or threat, the UAV must be capable to mask/unmask the data exchanged between air segment and ground segment. The amount of masking should be adapted to the mission context (e.g.: search and rescue or battlefield usage).

}

{REQ_AV22 Data access control
Mission Captured Data shall be accessed only by authorized users.

Motivation: Based on the mission type, the data captured by an UAV, can have a different level of accessibility. An authentication mechanism is required.

}

{REQ_AV23 ESs integration/expansion
Any Avionic ES shall be integrated into the Avionic System with minimal engineering effort.

Motivation: In case of any components replacement or sub-system's design modification, it is mandatory to re-qualify the whole system. Based on that, and, considered that such process is extremely expensive a new solution that could simplify and reduce such effort is needed.}

}

{REQ_AV24 Procedure against SW failure
Procedure against the ES software failures shall be provided.

Motivation: The UAV is a complex system, composed by different types of electro mechanical objects, mechanical parts, electronics, software and firmware. All of them interact each other in a complex and almost unpredictable way. The software components must be compliant to DO-254-a standard. A procedure for software fault identification, isolation and recovery must be considered.

}

{REQ_AV25 Procedure against HW failure
Procedure against the ES hardware failures shall be provided.

Motivation: The UAV is a complex system, composed by different types of electro mechanical objects, mechanical parts, electronics, software and firmware. The Hardware/Firmware components must be compliant to DO-178-B standard. A procedure for HW/firmware fault identification, isolation and recovery must be considered.

}

{REQ_AV26 Dynamic adaptively to the available resources
The ESs should be able to dynamically adapt the quality of the transmitted data and/or their SPD level according to the available resources and performance constraints (e.g. available bandwidth, CPU load, memory occupation, threat level).

Motivation: The UAV system must be capable to adapt the security privacy and dependability levels based on threats, scenario, performance and critical condition. i.e. Depending on the threats in the scenarios the UAV must properly adapt the SPD levels.

}

{REQ_AV27 Maintainability
In case of permanent failures, failed components should be replaceable with a limited effort, and with no SPD impact or with predictable SPD impact of the repair intervention.

Motivation: The UAV maintenance operations are extremely expensive, in terms of cost, time, obsolescence, and resources. A mechanism that can positively impact such aspects is strongly needed. In any case, the SPD level should be kept across the component replacement operation.

}

5.4 Social Mobility and Networking Scenario

{REQ_SM01 Means of use
The system shall provide a means to allow users to use the system services anonymously.

Motivation: Users want to protect their private data

}

{REQ_SM02 Personal data control
Users shall control how personal data are exposed to other users

Motivation: Users want to decide what kind of personal data to share with other users.

}

{REQ_SM03 Critical messaging
The system shall support time critical message handling and delivery.

Motivation: In case of emergency the system has to send or receive time critical messages.

}

{REQ_SM04 Historical information
The system shall provide historical information about the physical entity.

Motivation: a method for clarification whether goods transportation policy has been violated or not is required. To be able to do this, the continuous context information (e.g., temperature) of the things needs to be collected. This is for example of major importance to avoid any damage to the goods like pharmaceutics during the transport and storage process.

}

{REQ_SM05 Mobility
The system shall support mobility of the human user

Motivation: users want to access all areas covered by the system.

}

{REQ_SM06 Transmitted information integrity
Integrity of data transmitted over wireless or cable networks shall be granted.

Motivation: An attacker shall not be able to change security critical information during its transmission.

}

{REQ_SM07 Transmitted information confidentiality
Confidentiality of data transmitted over wireless or cable networks shall be granted.

Motivation: An attacker shall not be able to get access to clear text critical information during its transmission.

}

{REQ_SM08 User authentication
All users in the system must be authenticated.

Motivation: A non-legitimate user shall not be able to get access to any data and function in the system such as images, sensors control system etc.

}

{REQ_SM09 Authorization
All data access must be subject to access control.

Motivation: An inside attacker shall not be able to get access to data that he or she does not have the right to access to.

}

6 nSHIELD high level requirements

Below we describe the complete list of high level requirements we have derived from the four different nSHIELD scenarios. These requirements are derived as the “union” of all the identified different requirements from respective scenario or common to all scenarios. For traceability reasons, each of the requirements are marked with reference to the scenario requirements they origin from or with a motivation if they are common to all scenarios.

{REQ_SH01 Real-time data availability
Specific data (according to particular scenario) shall be provided in real-time.

Sources: RW01, AV1

}

{REQ_SH02 Information transmission integrity
Integrity of data transmitted over wireless or cable networks shall be granted.

Sources: RW02, AV02, SM06

}

{REQ_SH03 Transmitted information confidentiality
Data transmitted over wireless or cable networks shall be confidentiality protected.

Sources: RW03, AV03, SM07

}

{REQ_SH04 Peer authentication
All communication peers shall be securely identified.

Sources: RW04, AV04

}

{REQ_SH05 User authentication
All users in the system must be authenticated.

Sources: RW05, AV05, SM08

}

{REQ_SH06 Authorization
All data access must be subject to access control.

Sources: RW06, AV06, SM09

}

{REQ_SH07 Secure node deployment
A new node shall be identified as a trusted node prior to be accepted in the system.

Sources: RW07, AV07

}

{REQ_SH08 Secure software upgrade
All system components must provide a secure and safe software upgrade process.

Sources: RW8, AV8

}

{REQ_SH09 Denial of service
All network nodes should provide methods to prevent or reduce the impact of denial of service attacks.

Sources: RW09

}

{REQ_SH10 Secure execution
A system node should provide an isolated protected execution environment that can offer secure execution for the most security sensitive computing tasks.

Sources: RW10, AV09

}

{REQ_SH11 Secure boot
All nodes should provide a secure boot process.

Sources: RW11, AV10

}

{REQ_SH12 SPD level assignment
Each component or set of component in the system must have an assigned SPDy level.

Sources: RW12, AV11

}

{REQ_SH13 SPD level identification
It must be possible to securely verify the assign SPD level of all security critical components in a node.

Sources: RW13, AV12

}

{REQ_SH14 Software failure mitigation
The system should provide automatic means to recover from a situation when a node fails due to software failure.

Sources: RW14, AV13

}

{REQ_SH15 Hardware failure mitigation
The system should be robust against hardware faults such than when non communication critical hardware stop to function, it should be detected by the system.

Sources: RW15, AV14

}

{REQ_SH16 Data backup
The system shall provide data backup for most sensitive data.

Sources: RW16, AV15

}

{REQ_SH17 Data storage redundancy
Critical system data shall be secured through redundant storage

Sources: AV16

}

{REQ_SH18 Data storage integrity
Safety and/or security critical data shall be integrity protected at storage

Sources: AV17

}

{REQ_SH19 Data storage confidentiality
Security critical data shall be confidentiality protected at storage.

Sources: AV18

}

{REQ_SH20 Secure identification though face recognition
The system should allow secure identification of end users through facial recognition.

Sources: VF01

}

{REQ_SH21 Secure identification though voice recognition
The system should allow secure identification of end users through voice recognition.

Sources: VF02

}

{REQ_SH22 Protection of biometric data and metadata storage
Confidentiality of biometric data and metadata storage should, due to security and privacy reasons, be protected.

Sources: VF03

}

{REQ_SH23 Biometric data privacy
The original voice or image signal must be deleted after used in the system for secure identification.

Sources: VF04

}

{REQ_SH24 Biometric data and metadata confidentiality during transmission
Biometric data and metadata shall be encrypted in transit.

Sources: VF05

}

{REQ_SH25 Face recognition quality
Secure identification based on face recognition shall of such high quality that the risk of false identification is very low.

Sources: VF06

}

{REQ_SH26 Means of use
The system shall provide a means to allow users to use the system services anonymously.

Sources: SM01

}
{REQ_SH27 Personal data control
Users shall control how personal data are exposed to other users

Sources: SM02

}

{REQ_SH28 Critical messaging
The system shall support time critical message handling and delivery.

Sources: SM03

}
{REQ_SH29 Historical information
The system shall provide historical information about the physical entity.

Sources: SM04

}
{REQ_SH30 Mobility
The system shall support mobility of the human user

Sources: SM05

}

{REQ_SH31 Configuration management
Units/components/devices developed for nSHIELD project SHALL have version identifiers unique per each release of unit/component/device. These version identifiers SHALL be used to identify versions and trace version changes.

Motivation: nSHIELD Project tasks, e.g. **T6.3 Lifecycle SPD support** would necessitate methods to identify and track versions uniquely as developed over time or as research alternatives.

}

{REQ_SH32 Deployment manual
nSHIELD partners developing system components SHOULD provide description on how to deploy those system components, and how to operate them safely

Motivation: nSHIELD Project tasks, e.g. **T6.1 Multi-Technology System Integration** and **T8.2 Standardization** would necessitate specification of methods to deploy systems consistently, securely, and safely.

}

{REQ_SH33 Automated testing tools
Assigned nSHIELD project partner(s) SHOULD create automated testing tool(s) to generate arbitrary (even potentially faulty or malicious) input to exercise and evaluate input/output and internal interfaces of the critical components. Documentation, especially guidance documents (see {REQ_SH34}) MAY be used as a basis to generate relevant test excitation.

Motivation: nSHIELD Project tasks, e.g. **T6.2 - Multi-Technology Validation & Verification** would gain from support of tools defined herein.

}

{REQ_SH34 Guidance documents (AGD)
nSHIELD partners developing system components SHALL provide user documentation on how to use the system securely and safely. The guidance shall include warnings about actions that that can cause errors and lead to faults or failures in the secure environment.

Motivation: nSHIELD Project tasks, e.g. **T6.2 - Multi-Technology Validation & Verification** and **T8.2 Standardization** would necessitate specification of methods to deploy safe and secure systems.

}

{REQ_SH35 Vulnerability Assessment (AVA)
Using results from test, validation, and verification tools, assigned nSHIELD project partner(s) SHALL carry out security analysis on the system components for all scenarios in order to find potential threats that could leave the nSHIELD system vulnerable.

Motivation: nSHIELD Project tasks, e.g. **T6.2 - Multi-Technology Validation & Verification** would necessitate specification of methods to carry out validation and verification activities.

}

{REQ_SH36 Remote management
In the nSHIELD system the operating system or the middleware shall implement routines to send requests to neighbours, to force remote nodes to change their working point, in order to react to node failures, or to accept portion of a distributed computation.

Motivation: To enforce dependability a node must be able to cooperate with its neighbours, sending information on its possible failure and reacting on failures of other nodes.}

7 nSHIELD layers requirements

7.1 Node requirements

{REQ_ND01 Code execution
An nSHIELD node should verify that only authorized code (booting, kernel and application) runs on the system.

Motivation: only authorized code runs on system in order to avoid a malicious attack on information managed by node through unauthorized code execution.

}

{REQ_ND02 Data Freshness
An nSHIELD node should include data freshness checks.

Motivation: To avoid replay attacks.

}

{REQ_ND03 Digital Signatures
An nSHIELD node should be able to verify digital signatures even in cases where a trusted third party is not available.

Motivation: To allow for flexible secure operation in highly mobile scenarios.

}

{REQ_ND04 Policy updates
An nSHIELD node shall not accept security policy updates without authorized access.

Motivation: To prevent bypassing of security features by a malicious attacker injecting/replaying policy update messages.

}

{REQ_ND05 TPM Low Power mode
An nSHIELD node's TPM shall be able to enter into a low power state without compromising its security.

Motivation: Resource-constraints may lead to a device running out of power. In such a scenario the system must ensure it won't compromise the network's security or the security of stored data on the node itself.

}

{REQ_ND06 Third-party key management
An nSHIELD node may offer support for third-party key management services.

Motivation: To compensate for the shortage of ES computational power in constrained environments.

}

{REQ_ND07 Situational-aware and context-aware SPD
An nSHIELD node shall be able to provide situational-aware and context-aware SPD services.

Motivation: Offers appropriate levels of security, taking the resources/security trade-offs into consideration.

}

{REQ_ND08 TPM Remote attestation
An nSHIELD node should utilize the TPM remote attestation functionality to ensure the integrity of a node prior to resource allocation.

Motivation: To ensure resources are not allocated and sensitive data are not sent to a compromised device.

}

{REQ_ND09 Virtualization
An nSHIELD power node should employ virtualization techniques to allow concurrent virtual nodes to run independently onto the system.

Motivation: To offer a virtualized hardware redundancy mechanism. This is especially important in applications where dependability is a key parameter (e.g. avionics).

}

{REQ_ND10 Dependable authentic key distribution mechanisms
An nSHIELD node shall support secure and dependable low-cost key distribution mechanisms for initialisation or re-keying.

Motivation: Key distribution mechanisms are of great importance because of the distributed nature of the nSHIELD system. Resource-constraints mandate that this mechanism is not only secure and dependable but also lightweight in nature.

}

{REQ_ND11 Node – Physical/tamper resilience
An nSHIELD node shall be designed not to compromise the privacy of the contained information in the case of a malicious user gaining physical possession of the device. The device shall be resilient to tampering, micro-probing and reverse-engineering.

Motivation: Nodes often deployed in hostile and/or not monitored environments, thus owner's physical control over the device is not always an option (i.e. malicious users might have access to the actual node without being detected).

}

{REQ_ND12 Low computation authentication schemes
An nSHIELD node should include an optimized hardware implementation for an ECC-based public-key authentication algorithm.

Motivation: ECC will be one of the main lightweight cryptographic primitives utilized by nSHIELD nodes, both for encryption and digital signatures

}

{REQ_ND13 Location Privacy
An nSHIELD node shall feature privacy-aware management of location, utilizing secure storage and sanitization of such information prior to transmission.

Motivation: Part of the content and context awareness functionality. Also important for scenarios where access to location-based services is required. Regulatory requirement for each scenarios.

}

{REQ_ND14 Storage of private information
An nSHIELD node shall incorporate provisions that ensure the long-term storage of private information, not allowing the confidentiality of that information to be compromised even under fault conditions.

Motivation: To ensure that even if a malicious entity gains physical access to an nSHIELD node, no sensitive information will be disclosed.

}

{REQ_ND15 µNode / Wearable personal node – Anonymity & Location Privacy
An nSHIELD wearable personal node must feature privacy-aware management of location and other sensitive personal information, utilizing secure storage and sanitization mechanisms to be applied to such information prior to transmission.

Motivation: Regulatory requirement for each scenarios dealing with personal private sensitive information.

}

{REQ_ND16 µNode / Wearable personal node – Storage of private information
An nSHIELD node must incorporate provisions that ensure the long-term storage of personal sensitive information, not allowing the confidentiality of that information to be compromised even under fault conditions.

Motivation: Regulatory requirement for certain scenarios dealing with personal private sensitive information.

}

{REQ_ND17 Privacy in different trust domains
An nSHIELD node shall feature the necessary mechanisms for security token exchange to enable the issuance and dissemination of credentials within different trust domains.

Motivation: To allow the interoperability of nSHIELD nodes between trust domains. Especially important in high-mobility scenarios.

}

{REQ_ND18 eNetwork / Hybrid network compatibility
An nSHIELD node should be designed to allow switching between infrastructure-centric and ad-hoc networks on demand, in order to adapt continually to changes in the physical environment.

Motivation: To allow the flexible operation of nSHIELD nodes in changing environments and across trust domains, without compromising the secure operation. Especially important in high-mobility scenarios.

}

{REQ_ND19 Node – Flexible key distribution mechanisms
An nSHIELD node should support a flexible and secure low-cost key distribution mechanism for initialisation or re-keying, allowing the distribution of authentic public keys via insecure channels, either according to a pre-defined schedule or ad-hoc, while adhering to policy requirements as well as requirements participating parties impose.

Motivation: To allow the flexible operation of nSHIELD nodes in changing environments and across trust domains, without compromising secure operation or wasting valuable resources in re-keying. Especially important in high-mobility scenarios.

}

{REQ_ND20 Node – Conflict resolution between policy domains
In case nSHIELD nodes in different policy domains need to communicate, there should be mechanisms in place to facilitate the communication and resolve this conflict with the minimum processing and communication overhead.

Motivation: To allow the flexible operation of nSHIELD nodes in changing environments and across trust domains, without wasting valuable resources. Especially important in high-mobility scenarios.

}

{REQ_ND21 Dynamic security behaviour
An nSHIELD node shall be able to change its security behaviour based on the dynamic change of policy requirements without requiring reprogramming or shutting down the node.

Motivation: Seamless adaptation to policy changes, without requiring physical access to neither the device nor any downtime.

}

{REQ_ND22 Lightweight embedded operating system
nSHIELD nodes' operating system shall have low resource requirements.

Motivation: To satisfy resource constraints.

}

{REQ_ND23 Hardware/Software co-design
An nSHIELD node should incorporate hardware-software co-design techniques.

Motivation: To substantially increase application performance (e.g. public-key cryptography) with minimal device surface area and cost increase

}

{REQ_ND24 Situational-aware and context-aware SPD
An nSHIELD node should be able to provide situational-aware and context-aware SPD services.

Motivation: To offer optimization of the available system resources and maximization of performance..

}

{REQ_ND25 SCA protection based on EM emissions
An nSHIELD node should include interfaces to monitor its own EM emissions and modify its functionality accordingly.

Motivation: To protect its assets against SCA.

}

{REQ_ND26 Location awareness
An nSHIELD node should include the necessary interfaces that will enable location-based functionality.

Motivation: To offer access to location-based services but also allow its advanced features (e.g. context awareness) to operate.

}

{REQ_ND27 Accommodations for future energy sources
An nSHIELD node should have provisions for future alternative power sources including super-capacitors and wireless power schemes.

Motivation: Utilization of advances in the field of energy sources which would help alleviate some of the resource-constraints or offer fail-safe alternatives.

}

{REQ_ND28 Power management
In the nSHIELD system the hardware shall provide frequency dividers and variable supply voltage, in order to meet the most appropriate energy/performance tradeoff.

Motivation: This requirement manages any system power supply risk, which might affect to the node behaviour. It is useful at this level a continuous power supply source, without any cut in time neither in the power, voltage or current levels, to correctly bias the devices.

}

{REQ_ND29 Power management interface
In the nSHIELD system the hardware shall provide an appropriate interface, such as I/O ports or memory mapped registers, in order to allow the software to drive the frequency dividers and the supply multiplexers.

Motivation: The power management policies should be realizable as software routine, to allow flexibility and upgrading

}

{REQ_ND30 Workload assessment
In the nSHIELD system the operating system shall provide information on the past and on the current workload, for example the number of processes and the expired deadlines in a given amount of time, to allow the management software to choose the most convenient level of performance.

Motivation: The application level needs low-level information to effectively choose the power strategy to adopt.

}

7.2 Network requirements

{REQ_NW01 Confidentiality
The nSHIELD network shall support encryption of the packet data.

Motivation: In order to avoid malicious data corruption and to ensure transmission of sensible information over an unprotected channel, the network should provide a mechanism to protect the data, with a configurable confidentiality level based on the given situation and treats.

}

{REQ_NW02 Integrity
The nSHIELD network shall ensure the integrity of a transmitted packet.

Motivation: The network should prevent attack during transmission and it should ensure that data has not been tampered with replayed packets. Algorithms should be used to provide data integrity assuring that information is not altered during its transmission over the network.

}

{REQ_NW03 Secure Routing

The nSHIELD network shall support secure routing of transmitted packets.

Motivation: It should not be possible to interfere with the network path selection during data transmission or its configuration. The network layer shall provide a secure protocol and routing mechanism to prevent any malicious intrusion as well as unauthorized passive or active attacks.

}

{REQ_NW04 Fault Tolerance
The nSHIELD network layer shall tolerate faults.

Motivation: The network should continue operating properly also in the case of failure of one or more components. Node should be frequently checked to determine the network outages with the aim to contain faults as well as to prevent the propagation of the failure.

}

{REQ_NW05 Self-Management and Self-Coordination
The nSHIELD network layer shall provide distributed self-management and self-coordination schemes for unmanaged and hybrid networks.

Motivation: Because of their nature, nodes can be different. The heterogeneity of components can cause the generation of hybrid networks, to ensure the dependability of different networks, both in terms of structure or components, a self-management and self-coordination methodology should be provided.

}

{REQ_NW06 Multiple Protocol Support
The nSHIELD network layer should provide support for a variety of protocols used by the heterogeneous system of composable, embedded devices.

Motivation: The embedded node that belongs to a given network are generally independent and heterogeneous. To ensure a proper communication and a fully compatibility between different kind of nodes, a protocol discovery service should be provided, as well as the possibility to support technologically different components.

}

{REQ_NW07 Availability
The nSHIELD network shall ensure the availability of the service.

Motivation: In order for the nSHIELD network to be dependable, its availability must be ensured.

}

{REQ_NW08 Network Security Cryptographic Support
The nSHIELD network layer shall support both symmetric and asymmetric cryptography.

Motivation: Confidentiality can be achieved through encryption. Symmetric ciphers are faster and more efficient, whereas asymmetric ones solve the key management problem. Hence, both kinds are required.

}

{REQ_NW09 Traceability
The nSHIELD network should provide traceability information on each transmitted packet.

Motivation: If an anomaly or malfunction is detected in the network, it is useful to be able to track the source of the problem. In addition, it may be used for locating malicious nodes in the network.

}

{REQ_NW10 Audit
The nSHIELD network shall provide the required information to be used for auditing purposes.

Motivation: Auditing data will provide input that will enable the continuous monitoring of the network and the detection of anomalies/faults in the network.

}

{REQ_NW11 Reputation-Based Secure Routing
The nSHIELD network should support reputation-based secure routing.

Motivation: Given that the nSHIELD network is of a dynamic nature, reputation metrics for each network node are a possible means for distinguishing legitimate nodes from malicious ones.

}

{REQ_NW12 Reputation-Based Intrusion Detection
The nSHIELD network should support reputation-based intrusion detection schemes.

Motivation: Since the nSHIELD network will contain reputation information, this can be exploited for intrusion detection purposes.

}

{REQ_NW13 Fault Recovery
The nSHIELD network layer shall be able to recover from faults.

Motivation: In order for the nSHIELD network to be dependable, it should be able to recover from as many faults as possible. Otherwise, the network's availability may be jeopardised, if the number of faults accumulates.

}

{REQ_NW14 Application-Based Dependable Connectivity
The nSHIELD network layer shall provide dependable connectivity for a given application scenario.

Motivation: The nSHIELD network must provide dependable connectivity also at application level, so that overlay applications can maintain their normal operation.

}

{REQ_NW15 Dependable Authentic Key Distribution Mechanisms
The nSHIELD network layer shall provide dependable authentic key distribution mechanisms for the involved nodes.

Motivation: Cryptographic keys should be distributed to the involved nodes in a secure way, ensuring their authenticity. Otherwise, man-in-the-middle attacks may occur, where the attacker will have full access to the transmitted information, while the communication will seemingly be secure.

}

{REQ_NW16 Reliable Transmission Methodologies
The nSHIELD network layer shall provide waveform-agile and reliable transmission methodologies.

Motivation: The nSHIELD network will comprise a large number of wireless nodes. In order for the network to be dependable, reliable transmission methodologies are required. For instance, limiting

interference to the least possible extent is one step towards preserving the good operation of the network.

}

{REQ_NW17 Anonymity
The nSHIELD network layer shall provide anonymity of the source.

Motivation: Source anonymity is a valuable building block for meeting other requirements, such as location privacy).

}

{REQ_NW18 Location Privacy
The nSHIELD network layer shall provide location privacy for the source.

Motivation: In some cases, privacy issues may arise by exposing an individual's position, or route that he follows, by monitoring personal, electronic devices they carry (e.g. mobile phone).

}

{REQ_NW19 Application-Based Configurability
The nSHIELD network layer should/may support different configurations, according to the primary field of application.

Motivation: The nSHIELD network is meant for supporting different application scenarios, each one featuring different demands. It is therefore necessary to be able to configure the network, so as to provide the required support.

}

{REQ_NW20 Low Network Delay
The nSHIELD network shall feature low delay.

Motivation: Certain applications demand for quick response times (e.g. biometric recognition) and the network should therefore be able to support such requirements.

}

{REQ_NW21 Information Capacity
The nSHIELD network shall provide sufficient information capacity, for conveying information from/to the nodes and thus ensuring the good operation of the applications that exploit it.

Motivation: Certain applications (e.g. the ones that convey multimedia content) have increased requirements for information capacity and the network should be able to meet them.

}

{REQ_NW22 Secure Channel Establishment Interfaces
The nSHIELD network shall provide well-defined interfaces for establishing secure channels.

Motivation: A heterogeneous, dynamic network like nSHIELD should offer well-defined interfaces for enabling the nodes to join/leave the network and/or initiate secure communications. Otherwise, exploitation of the various ambiguities for malicious purposes may be possible.

}

{REQ_NW23 Key Exchange Interfaces
The nSHIELD network shall provide well-defined interfaces for cryptographic key exchange.

Motivation: Having a not well-defined key exchange interface may lead to malicious exploitation, thus flawing the security level of the network.

}

7.3 Middleware and Overlay requirements

{REQ_MW1 Discovery

The nSHIELD middleware shall offer discovery functionalities.

Motivation: Reference to Technical Annex par. 2.2.2.

}

{REQ_MW2 Secure Discovery

It is recommended that the nSHIELD middleware discovery is performed in a secure way.

Motivation: nSHIELD could discover the available services in an unsecure way bringing nSHIELD system to work in an unsuitable SPD level.

}

{REQ_MW3 Composition

The nSHIELD middleware shall be able to compose nSHIELD components and functionalities.

Motivation: Reference to Technical Annex par. 2.2.2, prevention of malicious attacks.

}

{REQ_MW4 Secure/Trusted Composition

It is recommended that the composition of nSHIELD components and functionalities is performed in a secure/trusted way.

Motivation: Reference to Technical Annex par. 2.2.2, Reference to deliverable D5.1 par. 4.2.2.1.3

}

{REQ_MW5 Orchestration and choreography

The nSHIELD middleware shall be able to orchestrate the composition of nSHIELD components orchestration may be driven by defined policies, rules or control algorithms.

Motivation: Reference to deliverable D5.1 par. 4.2.2

}

{REQ_MW6 Information retrieving

The nSHIELD middleware shall be able to retrieve information and requests from nSHIELD components.

Motivation: Reference to deliverable D5.1 chapter 4.

}

{REQ_MW7 Information filtering for intrusion detection

The nSHIELD middleware shall be able to filter information and requests from/to nSHIELD components to prevent malicious attacks.

Motivation: Reference to deliverable D5.1 chapter 4

}

{REQ_MW8 Enforcement

The nSHIELD middleware shall be able to enforce decisions taken by the overlay into SHIELD components.

Motivation: Reference to deliverable D5.1 chapter 4.

}

{REQ_MW9 Data management

The nSHIELD middleware shall be able to manage both service description and semantic data related to the discovered nSHIELD components and to the specific application domain.

Motivation: Reference to deliverable D5.1 chapter 3.

}

{REQ_MW10 Interoperability

The nSHIELD middleware shall be able to interface with heterogeneous legacy component.

Motivation: Reference to Technical Annex par. 2.2.2.

}

{REQ_MW11 Non-repudiation of origin for secure service discovery, composition and delivery protocols

The middleware components for service discovery, composition and delivery protocols should provide a method to ensure that a subject that receives information during a data exchange is provided with evidence of the origin of the information. This evidence can then be verified by either this subject or other subjects

Motivation: [TA].2.1.3.9 nSHIELD improved technologies, [TA].SP6 – Inter-networked ES for Security and Critical Infrastructures Protection

}

{REQ_MW12 Non-repudiation of receipt for secure service discovery, composition and delivery protocols

The middleware components for service discovery, composition and delivery protocols should provide a method to ensure that a subject that transmits information during a data exchange is provided with evidence of receipt of the information. This evidence can then be verified by either this subject or other subjects.

Traceability: [TA].2.1.3.9 nSHIELD improved technologies, [TA].SP6 – Inter-networked ES for Security and Critical Infrastructures Protection

}

{REQ_MW13 Access Control Policies for middleware components

Middleware components shall have policies that identify sets of the following entities for access control functions: the subjects of access control, the objects of access control and the operations between the object and the subject covered by the policy.

Motivation: providing technical solution to high level requirement of Secure Discovery and Trusted Composition

}

{REQ_MW14 Access Control Functions for middleware components

A mechanism to perform Access Control, based on defined Access Control Policies, shall be implemented in the Middleware component(s).

Motivation: providing technical solution to high level requirement of Secure Discovery and Trusted Composition

}

{REQ_MW15 Configurations definition

The nSHIELD overlay shall be able to elaborate feasible system configurations. The elaboration shall use, as input, service descriptions and semantic information retrieved by the middleware.

Motivation: Reference to Technical Annex par. 2.2.2.

}

{REQ_MW16 Configurations quantification

The nSHIELD overlay shall be able to measure the SPD level associated to each system configuration.

Motivation: Reference to Technical Annex par. 2.2.2.

}

{REQ_MW17 Configurations selection

The nSHIELD overlay shall be able to choose, among the feasible configurations, the one that satisfies the SPD requirements in terms of SPD level and pre-defined policies

Motivation: Reference to deliverable D5.1 par. 6.1.2.

}

8 Conclusion

Different project members conducted SPD assessment for SPD requirements definitions thus perspectives are probably less biased by national characteristics.The risk assessment was analysed by several project members. The analysis followed an explorative and incremental process. The requirements were described in a standardised way to ensure a common understanding by the analysts and to facilitate later exploration and usage by developers. The resulting requirements from the different analysts were integrated in a collaborative process.

9 References

[1] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, Transactions on Dependable and Secure Computing 1 (1) (2004) 11–33. January–March, ISSN 1545-5971.

[2] C.E. Landwher, A.R. Bull, J.P. McDermott, and W.S. Choi, "A Taxonomy of Computer Program Security Flaws," ACM Computing Survey, vol. 26, no. 3, pp. 211-254, 1994.

[3] www.w3.org/XML/

[4] www.w3.org/TR/soap/

[5] www.oasis-open.org/

[6] Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components.

[7] David A. Wheeler, Secure Programming for Linux and Unix HOWTO, v3.010, 3 March 2003 - http://tldp.org/HOWTO/Secure-Programs-HOWTO/x641.html

[8] V. Drakulovski et al "System Requirements and Specification for the pSHIELD-project", http://www.pshield.eu/index.php?option=com_docman&task=doc_download&gid=238&Itemid=37

[9] M. Al-Kuwaiti, N. Kyriakopoulos, S. Hussein, A Comparative Analysis of Network Dependability, Fault-tolerance, Reliability, Security, and Survivability, IEEE Communications Surveys & Tutorials, Vol. 11, No. 2, Second Quarter 2009.

[10] James McCabe, Practical Computer Network Analysis and Design, Morgan Kaufmann Publishers, Inc., CA, 1998, pp. 1-9.

[11] pSHIELD project, Deliverables D2.1.1, "Preliminary SPD Metrics Specification for the pSHIELD project," September 2011.

[12] A. Avizienis, J.-C. Laprie, and B. Randell, "Fundamental concepts of dependability", Research Report No. 1145, LAAS-CNRS, Apr. 2001.

[13] B. Melhart, and S. White, "Issues in defining, analyzing, refining, and specifying system dependability requirements", *Proc. 7th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2000)*, Edinburgh, Scotland, UK., Apr. 3-7, 2000, pp. 334-311.

[14] P. Neumann, *Practical architectures for survivable systems and networks*, Technical report, Final Report, Phase Two, Project 1688, SRI International, Menlo Park, California, Jun. 2000.

[15] J. Park, and P. Chandramohan, "Static vs. dynamic recovery models for survivable distributed systems", *Proc. 37th Annual Hawaii International Conference on System Science*, Maui, HI, 5-8 Jan. 2004, IEEE Comp. Soc. Press, 2004, pp. 55-63.

[16] R. Ellison, D. Fischer, R. Linger, H. Lipson, T. Longstaff, and N. Mead, *Survivable network systems: an emerging discipline*, (Report CMU/SEI-2001-TN-001), Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, Mar. 2001.

[17] Hu J, et al. seamless integration of dependability and security concepts in SOA: A feedback control system based framework and taxonomy. J Network Comput Appl (2011), doi:10.1016/j.jnca.2010.11.013

[18] Jonsson E. An integrated framework for security and dependability. Proceedings of the 1998 workshop on new security paradigms. ACM Press; 1998