Contract No. IST 2005-034891

HYDRA

Networked Embedded System middleware for Heterogeneous physical devices in a distributed architecture

Validation Plan for prototypes

Integrated Project
SO 2.5.3 Embedded systems

Project start date: 1st July 2006 **Duration: 48 months**

Published by the HYDRA Consortium **05.11.2007**
Coordinating Partner: C International Ltd.

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level: PUBLIC

Document file: HYDRA - D10.1 Validation Plan for prototypes

Work package: WP 10 – Validation & business modelling

Task: T10.1 – User validation

Document owner: INN

Document history:

Version	Author(s)	Date	Changes made
0.1	M. De Bona, A. Guarise	05-11-2007	Document organisation and objectives
0.2	HYDRA Consortium	07-12-2007	Structure revision at the PB meeting
0.3	A. Guarise	18-12-2007	Content assignment
1.0	A. Guarise, V. Grosso, M. De Bona	30-01-2008	Document first edition
2.0	A. Guarise, D. Thiemert, M. Ahlsén, M. De Bona	18-02-2008	Document revision based on comments from UoR and C-Net. Submitted to the Peer Reviewers
3.0	A. Guarise	14-04-2008	Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Comments
Mads Ingstrup (UAAR)	18-03-2008	See report file (inserted in the BSCW repository with the final document)
Markus Eisenhauer, Christian Prause (FhG-FIT)	18-03-2008	See report files (inserted in the BSCW repository with the final document)

Index

List of figures

List of tables

1. Introduction

1.1 Background

The HYDRA project develops middleware for networked embedded systems that allows developers to create ambient intelligence applications. System developers are thus provided with tools for easily and securely integrating heterogeneous physical devices into interoperable distributed systems.

The middleware will include support for distributed as well as centralised architectures, environment and context awareness, security and trust and will be deployable on both new and existing networks of distributed wireless and wired devices that typically are resource constrained in terms of computing power, energy and memory. HYDRA middleware will be based on a Service Oriented Architecture (SOA), to which the underlying communication layer is transparent.

The middleware is exemplified within the project in three domains: building automation, healthcare and agriculture.

1.2 Purpose and context of this deliverable

The purpose of this Deliverable is to explain the process and methods for the validation of the middleware and the prototypes within HYDRA. The report is an outcome of WP10 – T10.1 User validation dealing with the evaluation of user needs. It follows the work done in WP2 (and partly in WP3) providing the usage scenarios considered as the basis for the validation framework.

A further underlying objective of this validation plan is the collection of recommendations for improvement of the concept design.

Validation represents the testing and assessment of a system with the goal to prove the functions of the middleware (and its components) functionalities. The User Validation verifies that the system realises the benefits expected by the stakeholders, such as added value of the services, improvement of job satisfaction at the end-users, new methods of collaborative working, etc..

The HYDRA project shall validate the prototypes implemented in the project: the middleware, the software development kit (SDK), the device development kit (DDK) and the integrated development environment (IDE). Testing will be performed with developers who are directly involved in the prototypes exploitation. The evaluation shall regard also real end-user scenarios in the three domains: building automation, healthcare and agriculture. The document provides an overview of the planning including the timing, the expected results for the assessment of the prototypes and the methods for a basic evaluation of the developed applications in the three different domains. It is planned that the assessment will undergo several cycles, following the iterative approach already applied in the course of the project.

2. Executive summary

The deliverable presents a detailed validation plan for the three prototypes developed in HYDRA (SDK, DDK, IDE) with the intention to provide a comprehensive vision of the validation process.

The background is to demonstrate that the middleware and its components work satisfactory and fulfil all necessary performance criteria. This holds true as well for the validation of the SDK, DDK and IDE and its components.

After a basic technical evaluation is done, the next step is to assess the middleware, SDK, DDK and IDE with their users, that means to evaluate them with developer users and this is the first objective of the validation plan.

The last step is to evaluate the services provided by the prototypes with end-users, which are those who really make use and see the effectiveness of the HYDRA implemented applications.

This approach is followed because the general purpose of the project is to develop a middleware, which has developers as customers. The evaluation of the prototypes applied to the user application framework will be important for trying to understand if something the end-user is missing is the fault of the middleware or due to the fact that the developer was not working properly. This analysis, even if complex and limited in time, will allow to highlight important conclusions for the middleware, SDK, DDK and IDE.

The report is structured in 3 main parts.

Chapter 3 and Chapter 4 provide both the theoretical background and an introduction to validation and its application to the HYDRA framework. Validation is the testing and evaluation of a system with the goal to prove that the expected results are met. It is divided into several steps, initially focussed on technical aspect and then moving smoothly to the assessment of the quality of use of the applications. User validation is performed depending on the object to be evaluated. Technical analysis is done trough specific test to be applied to the system components, usually realised in a laboratory environment. This part is mostly done during the implementation phase and then fine tuned for the software release version. Quality assessment is performed once software development is completed, so that it is possible to have a clear idea to which extent the user requirements (i.e. user needs) are met, especially regarding the quality parameters. For this reason the validation plan is divided into two major sections.

Chapter 5 shows the organisation of the developer-users assessment. It consists of three evaluation cycles that represent the different iterations; each of them takes into consideration one of the prototypes and the middleware component, with the aim of continuously improving the software releases; the first iteration is focussed on the SDK, the second on the DDK and the third and last on the IDE validation. The objective is to gather the advices received from external developers (who didn't contribute to code writing and debugging during prototypes developments). The middleware and other components assessment is performed via the verification of relevant requirements and the collection of standard questionnaires. The section describes in details how the validation process will proceed, defining the fundamental steps to be followed and the time constraints to be respected for completing the task.

Chapter 6 regards the end-users validation plan. The object of the evaluation shall be the applications implemented from project developers to illustrate the HYDRA concepts and explain the middleware potential. For the end-user validation we underline that the HYDRA project scope is to develop tools (SDK, DDK, IDE) in order to address an intelligent cross-platforms software implementation. The applications developed are considered a mean to an end, a few good examples to understand the potential of the developed tools. So the end-users evaluation is considered an added value useful to understand how the HYDRA-implemented hardware and software could work from the point of view of the interested stakeholders. The end-users validation plan introduces the application scenarios and how to identify and select the end-users who shall be involved in the validation process (it is foreseen that a certain number of trial performers shall be found outside the

HYDRA Consortium). The applications investigation is performed via the verification of pertinent requirements and the collection of standard questionnaires. The chapter describes which steps shall be followed and the time constraints for completing the activity.

3. Introduction to User validation

Validation is the testing and assessment of a system with the goal to prove that it realises the benefits expected by the stakeholders (i.e. everyone who has an interest in the newly introduced services and applications), such as added value of the services, improvement of job satisfaction at the system users, new methods of collaborative working, etc.

The overall validation in a software implementation activity consists of three elements:

- Verification tests if the software is free of bugs.
- User validation evaluates if the services meet the expectations and requirements of its intended users.
- Usability testing is the assessment of the quality of use of the applications.

Software verification (debugging and testing) is always performed at laboratory level, given in any case the possibility that unexpected bugs are discovered after software release. So the outcome of this part is to check any eventual bug not yet faced during laboratory test. Hopefully this shall be a minor part of the User validation phase.

The second point is partly done at laboratory level, with internal technical experts analysing each software module and verifying its consistency alone and inside the overall architecture. Then the assessment of performance measurements is done with experts not participating to the implementation, so that there is an evaluation of the (stable) components and prototypes from different point of views.

The third point is at the level of the quality of use in the domain of field trials made from users, where controlled conditions are needed to assure that valid and interpretable results are obtained, useful as comparable benchmarks for customers.

3.1 User validation concepts

3.1.1 What is user validation

Validation is a key step in the development and implementation process. It is the process of verifying that an application performs as expected, often based on the assessment of results. Assessment is the process of determining the performance and/or impacts of a candidate application, usually in comparison to a reference case (existing situation or alternative applications) and usually including an experimental process based on real-life or other trials, often involving users.

3.1.2 How to make user validation

The validation plan is the basis for the validation. It defines a validation framework for test scenarios that assure that the range of the different test implementations is adequate to judge if the key aspects of the implemented system could work. To meet the specific needs of the stakeholders, a number of quality dimensions and assessment categories have to be identified.

3.1.2.1 Prepare the evaluation activities

The validation plan describes the appropriate method for user validation, meant as the identification of suitable approaches to measure different quality dimensions. The preparation of the validation plan considers as input what has to be evaluated (the objects of the evaluation), who will perform the assessment, the available resources (HW/SW tools and logging, documents and reports) and the timeframe for completing the task.

In this initial phase a considerable effort is needed for drafting the templates to be used from the evaluators, in order to guarantee a consistent reporting and collection of data.

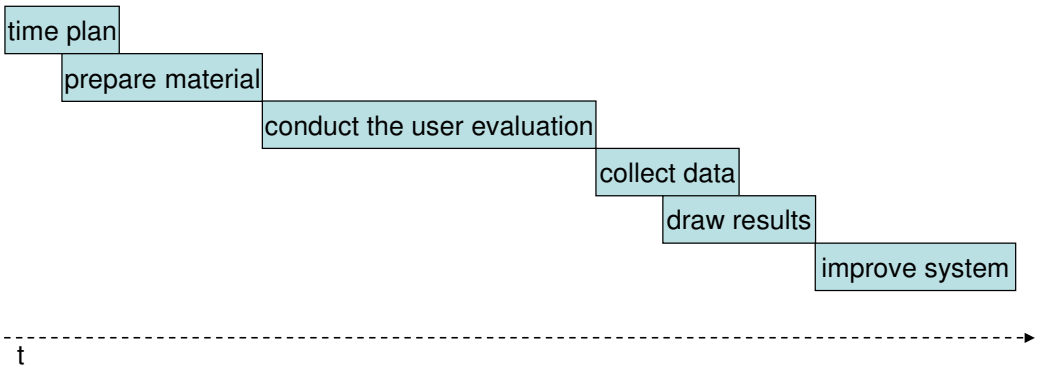

Figure 1: activity planning

3.1.2.2 Conduct the evaluation activities

The evaluation activities need precise scheduling in order to provide useful feedback into the loop at the right moment, and the validation plan gives the exact time frames and the procedures to follow at every different (new) step. The assessment is completed in several sessions. The need to assure a genuine outcome suggests proceeding with the testing in different environments and with users belonging to different kind of stakeholders. The available competencies in the resources required for user validation shall depend on the type of service.

This is the unique part where the real users are involved, because in the other tasks the work is done from project members or technicians.

3.1.2.3 Analyse data

Output derived from hardware/software tools (also logging), filled-in templates from users, comments arising during the validation activity are collected, classified and eventually filtered. This task starts after the previous activity is completed, so that the data analysis is not affected from unaligned input.

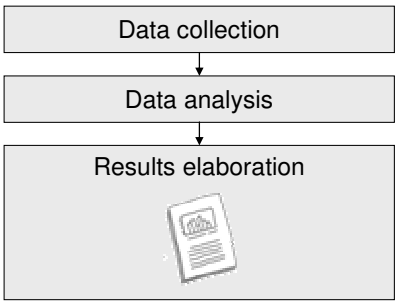

Figure 2: data analysis process

3.1.2.4 Feedback results back to the loop

User validation contributes successfully to the project development when it is integrated into the project plan and when standard project management techniques are used. For this reason an iterative approach is satisfactory meeting the constraints of a time-limited project, so that all the user feedbacks are not concentrated at the end, but they are distributed in different moments of the research activity.

It is necessary to prepare an appropriate communication scheme (also templates) to provide feedback from developer users and other stakeholders to the members of the development team. Of course it is also necessary the availability of sufficient resources to revise and adapt the system on the basis of test results (prototyping and iterative development cycles).

4. User validation in HYDRA

4.1 Object of the validation

The physical outcome of the Hydra project consists of a middleware architecture and components supported by development tools.

The middleware is based on a Service-oriented Architecture, to which the underlying communication layer is transparent. The middleware is aimed at providing interoperability of networked embedded systems: it will operate subject to variations in resource availability in terms of computer power, energy and memory usage and so it will support cost-effective and innovative embedded applications for new and existing devices. It will support distributed as well as centralised ambient intelligent architectures with reflective properties as well as including means for security and trust enabling of components.

The goal for the producers is to be able to build cost-efficient Ambient Intelligence (AmI) systems with high performance, high reliability, reduced time to market and faster deployment and still build on the assets of the installed base.

To facilitate the development, a series of development tools are available: The Hydra Software Development Kit (SDK), Device Development Kit (DDK) and IDE (Integrated Development Environment. The SDK and DDK are two different views on the middleware. The SDK will allow developers to develop the innovative software applications with embedded ambient intelligence computing using the middleware, while the DDK will allow device developers to enable their devices to participate in a Hydra network.

The SDK consists of the managers and associated tools (compilers, archives, debuggers, documentation, etc.), which are used to develop an application, together with the associated programming interface.

In contrast, the DDK consists of the managers needed to Hydra-enable a specific device. Both the SDK and the DDK offer Hydra functionality but at a low programming level.

The IDE will provide solution developers with a high-level interface for developing networked embedded AmI applications. The Hydra IDE can be integrated with existing IDE's such as Eclipse and Visual Studio.

In the user-centred approach utilised in HYDRA, the scenarios, the first basic set of requirements, the middleware implementation and the domain applications are refined in iterative steps, which follow the principles of the common standard ISO 13407. The part of assessment focused in requirements and scenario definition is performed in the design framework, in WP3 (as it appears in Deliverable 3.2 "Updated system requirements report").

In the iterative approach when a prototype is available, end-users can try it and gain personal experience with the system. Iterative cycles allow advancing from specification to implemented prototypes, from experience and evaluation to improved specifications and improved prototypes. In Hydra there are four full cycles planned to release the different prototypes throughout the project lifetime. Each cycle leads also to an application that demonstrates the developed components in a real scenario (applied to home automation, healthcare and agriculture). The first demonstrator is in the home automation domain and it is released together with the first version of the middleware. Cycle two leads to the SDK release together with applications in all domains. Then applications will be augmented in each cycle. At the same time in cycle three the DDK prototype and in the last cycle the IDE prototype will be subject of validation and evaluation. This is represented in the next Figure.

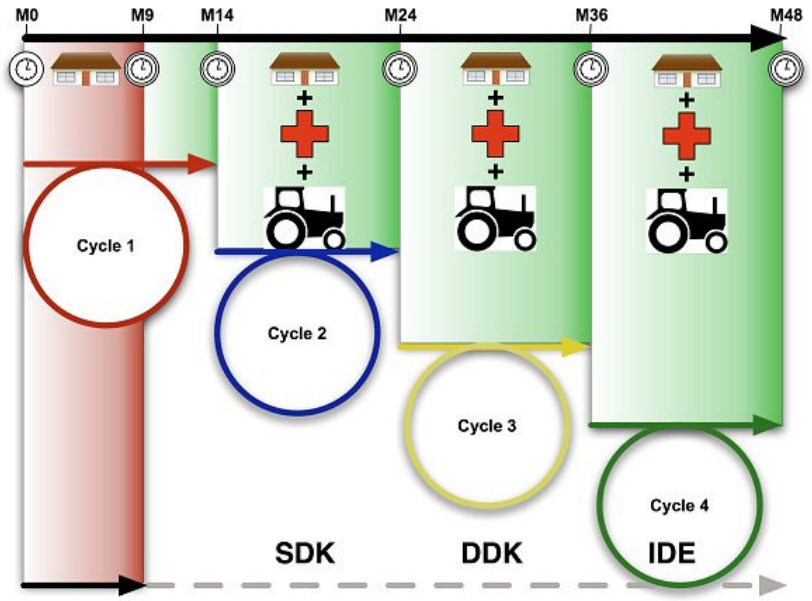

Figure 3: HYDRA prototypes timeplan

This iterative process ensuring the gradual approximation to the HYDRA middleware has a strong impact in the validation plan, because it has to be also organised in different cycles, so that simultaneous consideration of the developer needs are collected at the end of each loop.

For each of the prototypes a user validation plan is developed in this document. The plan can be updated as needed and as the project progresses. Correspondingly, for the reporting of results a similar template is proposed. As during each cycle also three applications are developed and fine-tuned, one for each of the selected domains, it is interesting to foresee also a user validation scheme to be fulfilled at cycle four that collects the user responses while interacting with each application. This is worth not just for the "look and feel" aspects, but also for a greater understanding of the functioning and usefulness of the application developed and for assessing the business benefit and added value of the new applications in a realistic application environment.

Software development project face usually common drawbacks, but HYDRA presents also the additional problem that we are trying to evaluate a middleware, which is one step more away from real users compared to a normal SW product. Traditional methods developed for evaluating software are not always applicable to highly innovative products and services which are the focus of the project. One of the objectives of the user validation approach is to use the synergy between the applications as much as possible by using common methods, and by looking for complementary results. The challenge may be met by using care in the approach, and awareness of the fact that comparison with existing applications, and the use of previous experience is not possible. The procedures followed consider the experience from other similar projects.

4.2 HYDRA target users

As HYDRA is aiming at developing a service oriented and model driven middleware for embedded systems, there are two groups of users:

- developers that will use the middleware
- end-users that will benefit from the HYDRA enabled Services created by the developers.

Therefore the term of developer-user refers to developers and end-users refer to the users of the products provided form the developers. In case that both are meant we simply speak of users. For

the type of "product" developed in HYDRA the first group is considered with greatest interest, because the developer shall be the direct and primary users of the HYDRA middleware tools. This is also a challenge because we must consider that evaluation with developer-users may or may not lead to new issues if compared to traditional user validation. The end-users are also considered because it is really useful to have a critical mass regarding the market feedback received from people who will benefit from the technology enabled with HYDRA.

Groups of users with homogeneous characteristics within a group and different characteristics between groups must be identified. The characteristics must be relevant for the development of the HYDRA middleware, especially for the demonstrators under development. The developer users will be identified among HYDRA internal resources where possible. This is done mainly because it is difficult to find the commitment from companies not directly involved in the HYDRA consortium, especially from an economical point of view (external experts who are not HYDRA partners asking for a fee shall be paid with mean of subcontracting). The selected developers will be chosen among those who were not directly involved in the HYDRA implementation, otherwise their judgement would not be unbiased. In case it is not possible to identify just resources belonging to the HYDRA environment, external experts will be contacted and eventually hired in order to perform the requested task.

The end users in charge to evaluate the applications will be selected among the group of stakeholders as emerged in the analysis carried out during the Deliverable 10.5 Business modelling concepts preparation and depending on the evolution of the demonstrators' implementation, both based on vision scenarios. In the document the different stakeholders and classes of stakeholder have been identified and explained, for each of the domain considered within HYDRA and dividing them among different level of pertinence to the HYDRA framework. When all applications will be well defined and solid, towards the end of the developments, the stakeholders will be selected and asked to participate to the validation activity. In this case it will be necessary to lean external resources able to fulfil the assessment.

Tasks are here identified in a top-down manner, from the definition of system objectives, use cases or scenarios of use, down to detailed procedures. Hereafter it is presented the decomposition of global procedures into increasingly detailed descriptions of partial tasks, divided also among the sub-sequent iterations. The decomposition stops when the appropriate level of detail is achieved. This depends on the aspects of the system which are under investigation, and may range from high-level scenarios to keystroke-level description. A detailed technical implementation is presented for each cycle of user validation (done in three steps, SDK, DDK, IDE and applications).

Type of user	Object of the evaluation	Start of the user validation (month)
Developer user	SDK + middleware vers. 1	M24
Developer user	DDK + middleware vers. 2	M36
Developer user	IDE + middleware vers. 3	M46
End user	Applications	M45

Table 1: validation plan milestones

Developer-users are more interested in requirements fulfilment, the technical aspects related to software development applied to the major object they expect: a middleware, a SDK, a DDK or a IDE. This will be investigated during the first phase of the validation (first iterations).

End-users are more focussed on the exploitation of a developed solution, in the sense that it has to be easy to understand, comfortable to use and efficient while working. This is studied at the end of the project, during the last iteration.

As a general observation, large-scale trials involving real users should only be initiated when a stable system allowing meaningful productive use is available, otherwise lack of attention towards the system shall be expected from evaluators (and this would misrepresent the results).

4.3 Quality dimensions and assessment criteria

The validation is made through the comparison between an expected impact (requirement) and how the real application works. In HYDRA the expected impact is described with mean of the user requirements, derived and collected in WP2 and WP3. The user requirements consist of a list of features and properties of the HYDRA middleware including quality criteria, which are considered relevant by the users. Deliverable 3.2 "Updated system requirements report" contains an updated overview of the requirements that shall be necessary to the HYDRA developed system as emerged in several focus groups with developer users.

Every requirement statement is composed of six fields to briefly describe it, as shown in the next example.

ID:	31
Type:	Non-functional / look and feel
Priority:	Critical
(Short) description:	An easy-to-use programming framework should be provided
Rationale:	The programming framework provided by the SDK should be easy to use in the sense that it is intuitive
Fit Criteria:	9 out of 10 developers recognise the IDE as intuitive

As quality is a relative or personal issue to be measured, a value must be attached to the cost and benefit of quality-oriented actions. Features and properties requested by stakeholders have to determine on how to implement and what the optimal investment is.

There are different frameworks analysing quality attributes, with differing vocabulary, metrics etc. that are relevant to software architecture design. Quality attributes are essential to the design of software architecture, but it is a challenge to describe quality attribute (requirements) on a common form. For this reason, together with the Volere schema for drafting user requirements, the SEI quality framework (Bass et al., 2003) and the ISO 9126 (2001) international standard have been studied. The SEI quality framework, also known as Quality Attribute Scenarios, is a well-established way of defining architectural requirements in a uniform way and introduces the concept of considering quality attribute requirements on a fixed and precise scenario form. This approach has been integrated in the context of the HYDRA project with the ISO 9126 international standard defining a comprehensive quality model for software products. Deliverable 6.1 "Quality Attribute Scenarios" gives a detailed and clear overview of the two frameworks.

A general scheme was introduced in the validation framework report, on which and how to measure different attributes in respect to a selected group of quality dimensions: performance, subjective assessment quality, learning effort, cognitive workload, added value and security, safety, privacy.

Performance (functionality, effectiveness, efficiency, reliability, etc)

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Performance – functionality	Rating by users	Global rate		Better than the average	Above average	Questionnaire, Positioning
Performance - effectiveness	Rating by users	Global rate		Better than the average	Above average	Questionnaire, Positioning

Performance - efficiency	Time to perform a given task	Minutes			Faster than traditional work / activity	Performance measurements
Performance - reliability	Working time	Rate			Less than 0.1% time not working (99.9% time correct functioning)	Performance measurements

Table 2: Quality dimension - performance

Performance of software is normally tested in a controlled environment, like in a laboratory, with mean of measurements, using log files, timestamps and other tools suitable to test the running application. As regards functionality and effectiveness, the rating given from users is of utmost importance, an immediate feedback of the usage impression from a potential customer.

Subjective assessment (affect) of the quality of an application

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Subjective assessment quality	User satisfaction	Global rate	Not satisfactory impression		Large consensus	Questionnaire

Table 3: Quality dimension - subjective assessment

The subjective assessment gives an idea of the system out from the technical scope, a sort of overall impression of the whole system, without a focus on a specific topic. For this aspect it is important the type of audience and the presentation given of the evaluated object.

Learning effort required using a system

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Learning effort	Time to coach (developer users)	Hours	More than a 5-days training period		2-days training period	Coaching time measurements
Learning effort	Time to learn (end user)	Minutes	Below average	Better than the average	Above average	Learning time measurements

Table 4: Quality dimension - learning effort

The learning effort have to be applied both to the level of the software developer, who has to learn how to make software using HYDRA middleware, and from the level of the final user who benefit from the HYDRA enabled new technology. Learning activity is highly dependent on the capabilities of the scholars, so in the first case it is possible to shift the measurement on the time necessary to complete a training on how to use the HYDRA software. This is easier than calculating the time spent for learning how to use SDK, DDK or IDE from a developer user.

On the other hand for the evaluation of the applications' example made from the final users this is difficult to pursue, then a measurement of the time spent to learn how the demonstrator works will be required.

Cognitive workload

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Cognitive workload	Time to learn	Rate	More than 50% of working time for more than one week		20% of working time for less than one week	Learning time measurements

Table 5: Quality dimension - cognitive workload

The cognitive workload is difficult to be assessed as it is for the learning effort, but the two concepts are somehow linked, so the same evaluation approach can be used for both. The time spent for learning how to use SDK, DDK or IDE (developer user) and how the demonstrator works (final users) is directly proportional to the cognitive workload, with different reference values (as reported in the Table).

Added value

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Added Value	Rating by users	Number of benefits mentioned			Above average	Questionnaire, Positioning

Table 6: Quality dimension - added value

In the validation plan the added value is analysed in the users' questionnaire. Part of the added value assessment is done also in the business framework (task 10.2), where there will be an economical estimation of the cost of ownership.

Security, safety, privacy

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Security	Rating by experts Number of vulnerabilities	Global rate of vulnerabilities number		Above average	No vulnerabilities	Questionnaire, Positioning, Conjoint Measurements
Safety	Rating by users Number of vulnerabilities	Global rate of vulnerabilities number		Above average	No vulnerabilities	Conjoint Measurements
Privacy	Rating by users and experts Number of vulnerabilities	Global rate of vulnerabilities number		Above average	No vulnerabilities	Questionnaire, Positioning

Table 7: Quality dimension - security, safety, privacy

The goal of security and privacy is to ensure that appropriate standards and regulatory procedures are integrated conforming to the governance and policing of quality of service for software development or applied to the different sectors where examples of service is deployed.

Safety issues are not pertinent to the use of the middleware, SDK, DDK or IDE. The use of software applications is harmonised by regulatory procedures that will be integrated into HYDRA platform.

Safety is considered applicable for the applications in the building automation, agriculture and especially healthcare domains.

The Volere approach allows to have a direct measure of the specific user need (and quality parameters) with mean of the fit criteria, which allows an immediate feedback on what is necessary to fulfil the requirement. So the evaluation of the system quality is measured through the correspondence between each requirement and the correspondent assessment criterion that tells how far the quality of the developed system (or component) should go in order to fulfil the requirement itself. As an example, considering the previous requirement, during the assessment period the validation method will examine (with mean of a questionnaire) if it is verified that "9 out of 10 developers recognise the IDE as intuitive"; this process will be applied to all the relevant items found in the updated system requirements report and the results obtained will be the measure of the HYDRA components successfulness.

During the validation activity there will be also an estimation of those quality issues not yet investigated with mean of the previous scheme (requirements fitness). If lack of information is identified, the validation activity will be enlarged with mean of further measurements or extended questionnaire so that all the meaningful quality aspects are properly examined.

In the last iteration it will be considered the quality of use from the viewpoint of end-users, not just from a technical point of view. In this case quality of use is an issue related to final-users (or market product), and the examples of the three developed applications will be assessed (building automation, healthcare and agriculture).

Both end-users and developer-users are also concerned about the total cost of a service. Quality seen as the cost of ownership study of the HYDRA service will be estimated in Task 10.2 on the business framework.

4.4 Selection of HYDRA user requirements

Requirements have a major sub-division into functional and non-functional (or quality) because quality is considered as 'orthogonal' to functionality in the sense that the same functionality may be supported by an open-ended amount of software architectures that to a large extent determine the quality of a constructed system (see Bass et al.).

The user validation carried out in the project with the help of the present document will focus on the quality requirements, as large part of the functional ones are considered to be tested with proper verification before the software release appear (during software debugging). Otherwise the functioning of the prototypes themselves would be undermined at the basis. Nevertheless the critical functional requirements (and those considered as particularly relevant) will be assessed during the validation phase.

The list of requirements to be evaluated is presented in the following paragraphs. The Tables report the selected requirements in normal text, while the functional (or not applicable) ones are in over-lined text: these shall be assessed during debug activity and won't be taken into account during the validation phase.

The list is not considered as definitive. During the validation tests some major requirements shall be comprised in the selection, and other will be added and updated during the project lifetime, as soon as the need for new features is identified. In particular, we will review the user requirements at the end of this first user evaluation iteration, in order to get the second improved set of user requirements.

Requirements are grouped into different clusters. One is based on their relevance to the focus of the different technical WPs:

- WP3 - Architecture Design Specification
- WP4 - Embedded AmI Architecture
- WP5 - Wireless Networks and Devices
- WP6 - SOA and MDA Middleware
- WP7 - Trust, Privacy and Security

The second cluster stands on their impact in different part of the system: architecture, middleware layer, devices, device integration, networking, communication, configurability, interfaces, service discovery, security, context, SDK, IDE.

The tables give also an indication if the single requirement is pertaining to a prototype/application, so they will be considered as input for the identification of the requirements to be tested in the following sections.

4.4.1 Middleware requirements from WP3 - Architecture Design Specification: Architecture

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~21~~	~~Constraint / assumption~~	~~Blocker~~	~~Hydra should be a Service-Oriented Architecture (SOA)~~	~~Hydra should be a SOA per the Description of Work of the project~~	~~Hydra is compatible to the SOA-definition by OASIS: http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf~~					
~~241~~	~~Constraint / requirement constraint~~	~~Blocker~~	~~Middleware should be open source.~~	~~We have stated in the DoW that we will produce open source software.~~	~~The core components of the Software are open source.~~					
25	Functional	Critical	Overwriting system decisions	Possibly dangerous outcomes of system decisions must be over writable by end-users	End-users can overwrite 90% of the application decisions	✔				
136	Non-functional / performance	Major	Dynamic architecture	An architecture of a running Hydra system can be easily modified by increasing or decreasing the degree of centralisation in order to balance utilisation of available resources.	In 95% of all cases, Hydra supports dynamic migration of components to realise centralised and decentralised systems.	✔				✔
199	Functional	Critical	Modules should be extendable	Hydra modules should be extendable in their functionality by 3rd-party solutions	80% of all Hydra modules are extendable in their functionality by integrating 3rd-party code via a standard interface or replaceable by 3rd-party modules with equivalent functionality.	✔	✔	✔	✔	
236	Functional	Critical	Middleware is extendable with additional functionality by plug-ins	The middleware provides basic services that could be enhanced and adapted by additional integration of specific plug-ins.	The middleware provides well-defined interfaces for additional plug-ins	✔				
288	Functional	Critical	Query devices for their functionality	Enable developers to get information about the offered functions of a certain device in an ad-hoc manner	All Hydra enabled devices can be queried for their supported functionality	✔				
327	Non-functional / performance	Critical	The Hydra middleware should be flexible as to allow for opt-in and opt-out on parts	Not all parts of Hydra will make sense in all situations (it will not always be beneficial to use higher layers of communication such as a service composition protocol or maybe a device may be too resource-constrained to use parts). One should be able to take the parts of the Hydra middleware that makes	Hydra is able to support the exact subset of services required by a client (user or service) in 70 % of all cases. In 20 % of all cases the middleware is able to provide a service package that includes the required service.	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				sense for a certain application. For example, it should be possible to for embedded devices with few resources (see other requirements) to take part in a Hydra application without having to install or run all Hydra components. Another example may be that one may want to use just point-to-point communication of Hydra without having to use the context-awareness part. (Werner Vogels, CTO/Amazon at JAOO 2006: "Middleware is evil!", referring to that if one chooses a certain middleware such as CORBA one makes too many decisions (not only on communication in the CORBA case but also, e.g., on transactions) that may not be appropriate for the case at hand)	In 10% of all cases Hydra is not able to provide service similar to the desired service.					
329	Non-functional / maintainability	Major	Middleware provides domain-independent services	A lot of the services needed in the apartment scenario are also needed in other scenarios (persistence, logging, visualization, ...). These should be abstracted and built and provided as part of Hydra	Large parts of the building-automation scenario can be built by reusing configurable services from across other application domains.	✔				✔
337	Non-functional / operational	Critical	UAAR: There should be a procedure/strategy for interfacing with non-Hydra devices	Not all devices will be Hydra-enabled neither in the near nor the far future. The architecture should support communication with and use of such devices to enable developers of Hydra-based applications to create rich applications	75% of non-Hydra devices can be integrated into Hydra Middleware	✔				✔
350	Functional	Critical	Data type transparency	Different devices in sensor networks use different bit sizes. Hydra must provide transparency between data types. Hydra must provide some sort of data type wrappers for the different arch and cpu types.	100% of all basic data types, 90% of less common data types can be transferred between devices with different bit sizes.	✔		✔		
17	Constraint / requirement constraint	Major	When applicable, middleware interfaces are exposed by WSA-compatible services	Web Service Architecture (WSA; http://www.w3.org/TR/ws-arch/) introduces a common definition of what a web service is and describes minimal characteristics of what is common to all web services. When web services are used in Hydra, they should comply to WSA	In min. 90% of all cases, Hydra web service interfaces are realised as WSA-compatible web services. In the remaining cases, web services use proprietary formats.	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
18	Non-functional / usability	Major	Support for different software architectural patterns	The Hydra architecture should not prescribe one way to structure applications. Thus several architectural patterns, for example MVC and PAC should be supported.	Hydra allows at least two different architectural patterns for applications.	✔	✔	✔		
~~167~~	~~Functional~~	~~Major~~	~~Distributed response composition~~	~~Service orchestration should also enable the distribution of responsibility of response composition (e.g. Multi agent collaboration).~~	~~Response composition is distributed among two entities at least.~~					
~~174~~	~~Functional~~	~~Major~~	~~Coordinated resource sharing~~	~~Resource sharing enables the exploitation of distributed collections of available resources both computational as well as other services. Scheduling computation and access to resources is essential for running several applications on the middleware concurrently.~~	~~Resources can be shared by at least two entities.~~					
~~211~~	~~Functional~~	~~Major~~	~~There are components/services in the middleware that integrate subsystems~~	~~The integration of basic systems to subsystems should ease the configuration of higher level services. Higher level services could then consist of a combination of basic systems~~	~~It should be possible to combine basic services to higher level services. At least one higher level service relying on a combination of basic services exists.~~					
217	Non-functional / performance	Major	The middleware should ensure high robustness of services	In order to ensure the service support of important components in the system, the middleware should provide a highly robust service structure.	Breakdown of crucial services of the middleware in less than 1 case per 100 hours of operation.	✔				
219	Non-functional / performance	Major	Redundant core components	To ensure high robustness, core components should be redundant.	No core component should be unique.	✔				
~~230~~	~~Functional~~	~~Major~~	~~Self-clustering of services~~	~~Cluster provides high flexibility in dynamic systems.~~	~~A new service will be clustered automatically with corresponding services.~~					
~~237~~	~~Non-functional / maintainability~~	~~Major~~	~~The guaranteed future should be ensured~~	~~The Hydra middleware should be kept adaptable and future proven.~~	~~After 10 years in the market, the middleware can still be used.~~					
323	Constraint / scope of the project	Major	Distributed Intelligence should not lead to resource-heavy systems	We have a need for "intelligence" (Semantics, reflection etc.). We have a need for supporting embedded systems. This should not conflict	Minimum hardware requirements (which must be supported by all target hardware) are defined and all hardware that meets the	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
					specifications is guaranteed to work with hydra.					
324	Non-functional / performance	Major	Systems built using Hydra should be scalable in terms of devices communicating	In large installations (such as in the apartment complex example) there will be many devices per apartment and a huge amount of embedded devices in total. Hydra should support the development of such big systems.	The Hydra middleware supports applications in which more than 100,000 devices exist.	✔				✔
329	Non-functional / maintainability	Major	Middleware provides domain-independent services	A lot of the services needed in the apartment scenario are also needed in other scenarios (persistence, logging, visualization, ...). These should be abstracted and built and provided as part of Hydra	Large parts of the building-automation scenario can be built by reusing configurable services from across other application domains.	✔				✔
~~354~~	~~Functional~~	~~Major~~	~~Support for virtual devices~~	~~In order to make each user have his own view on a device, there has to be some kind of support for virtual devices. This means that a single device may show up as multiple devices, which respectively provide a fraction of the original physical device's functionality, depending on actual user needs.~~	~~95% of all access to the Hydra middleware should be able to set up virtual devices.~~					
~~356~~	~~Functional~~	~~Major~~	~~support for both a pull and push model~~	~~By default, Hydra components should exchange messages according to the push-model. However, in some cases, a pull model should also be available.~~	~~In 90% of all cases, the system can handle push and pull commands.~~					
~~159~~	~~Non-functional / operational~~	~~Minor~~	~~Service brokers must be organized in a hierarchical way~~	~~With hierarchical brokers the system becomes more robust and scalable. Users do not want that everything acts up in case of a fire and a broker goes down. Additionally hierarchical brokers allow for having certain rules/services only within a sub-domain.~~	~~Brokers are organized hierarchically~~					
~~170~~	~~Functional~~	~~Minor~~	~~Stateful and stateless communication~~	~~Application developers should have the possibility to use stateful as well as stateless communication between components.~~	~~Hydra provides an API that allows the implementation of stateful and stateless communication protocols.~~					
~~172~~	~~Functional~~	~~Minor~~	~~Learning resource usage~~	~~Learning usage patterns of utilizing devices and computational resources, and collaboration among application components is essential for self-configuration in order to optimise usage of~~	~~Model of a resource usage can be learnt.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				~~available resources and overall application performance.~~						
~~176~~	~~Functional~~	~~Minor~~	~~Aggregation of resources~~	~~Aggregation of resources (e.g. computational) enables to outperform the limitations of a single system and to leverage available resource distributed across devices. This aggregation should be based on automatic coordination of multiple resources.~~	~~Device can distribute computation efforts among several other devices~~					
320	Non-functional / maintainability	Minor	Separate domain-oriented services and user interface services architecturally	This is a standard architectural design tactic to enhance modifiability	90% of the modules of the architecture properly separate layers for domain services and interfaces.	✔				
346	Non-functional / operational	Minor	UAAR: It should be possible to have closed subsystems	Hydra should not prescribe that a system should be completely open (and service-based) in order to be part of a Hydra application (e.g., Siemens may want Siemens heating systems to not be usable (in parts) by Philips home control systems)	A manufacturer or an application developer should be able to design Hydra components with proprietary interfaces in 100% of all cases.	✔				✔

Table 8: WP3 - architecure

4.4.2 Middleware requirements for WP3 - Architecture Design Specification: middleware layer

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
25	Functional	Critical	Overwriting system decisions	Possibly dangerous outcomes of system decisions must be over writable by end-users	End-users can overwrite 90% of the application decisions	✔				
161	Functional	Critical	Middleware must implement a role concept	A role concept can significantly simplify the resolution process of contradicting instructions.	A role concept implementation is part of the middleware that can resolve contradicting instructions in 90% of all cases.	✔				
185	Non-functional / operational	Critical	Middleware provides basic services	In order to program AmI applications, the middleware must provide basic services. This makes life easier for	Middleware provides a set of basic services that at least contain basic functionality that	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				application developers. Basic services provide e.g. methods to query available devices and services or to pass messages between components	is needed by all services, like communication and a service / device registry.					
189	Functional	Critical	Plug and play support for adding devices	Devices should be accessible as soon as they are discoverable and with the need for the developer to implement this functionality. This should be something like Plug'n'Play in operating systems.	Plug and play mechanism for inclusion of newly detected devices is done by the middleware	✔		✔		✔
194	Functional	Critical	Conflict resolution mechanism	Information obtained from different sources can be conflicting or contradictory. In this situation a conflict resolution mechanism should determine on relevance, reliability, and risk related to these sources.	The Hydra middleware is able to proceed in its operation in 98% of all cases, where contradicting information or conflicting commands are received.	✔				
207	Functional	Critical	Service selection by context	In order to select an appropriate service for a specific task, contextual information, like the spatial position, must be taken into account. Hydra must provide a method to specify a desired service by contextual parameters. For example, if a certain room in a building is specified in a search request for a service, only services are returned that are relevant in the current user's location and context.	In search requests for a specific service, contextual information like a spatial position is allowed.	✔				
215	Functional	Critical	Middleware only handles communication.	The middleware should implement only the most basic service, i.e. the communication. All high level functionalities will be realized by additional services.	The middleware only handles communication. All other functionality is realised by external components.	✔				
19	Constraint / scope of the project	Major	Support of low-end devices	Hydra must support low-end devices like RFID tags. Therefore, Hydra must be compatible with at least 32-bit devices with < 512 KB RAM/FLASH or less. For smaller devices, Hydra provides proxies.	Middleware is able to be installed and run on low-end 32-bit devices with 512 KB RAM/FLASH in 90% of all cases. Proxies can be created to support more limited devices in 40% of all cases.	✔				
~~82~~	~~Functional~~	~~Major~~	~~Data Logging~~	~~For system maintenance and debugging, logging functionality is mandatory.~~	~~Hydra provides a logging component that can log system actions. Also, an API is available that enables developers to include logging~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middleware	SDK	DDK	IDE	Application
					~~in their applications.~~					
~~132~~	~~Functional~~	~~Major~~	~~Hot swap of platform components~~	~~Deployed Hydra application should enable replacement of a platform component (utilised by some middleware module(s)) without interrupting operation. It enables to reduce down time of the application.~~	~~Hot swapping a component at run time is possible in 50% of all cases.~~					
~~148~~	~~Functional~~	~~Major~~	~~Access to basic and extended functionality~~	~~The middleware should provide a device's basic functionality via standardized, common methods. The way, extended / extra functionality can be accessed, should be also standardized.~~	~~Hydra provides standardized access methods for at least 90% of all Hydra-enabled devices. Some devices can have proprietary interfaces.~~					
~~150~~	~~Functional~~	~~Major~~	~~Rules-engine not part of the middleware~~	~~A rules-engine for defining the behaviour of a network of Hydra devices must not be part of the middleware. A rules-engine is to complex to be realised inside the middleware.~~	~~The middleware does not include a rules-engine.~~					
~~162~~	~~Functional~~	~~Major~~	~~Middleware allows implementation of fault detection service.~~	~~Although fault detection as part of the middleware is not mandatory, the middleware must lay the foundation (e.g. an API) for building such services.~~	~~The middleware includes an API to implement fault detection.~~					
~~163~~	~~Functional~~	~~Major~~	~~Policy and Context are not part of the middleware~~	~~Context awareness as well as making decisions based on policy strategies can be resource intensive computing processes. Modules providing this functionality must not be part of the middleware.~~	~~The middleware does not implement context awareness and policy strategies.~~					
180	Non-functional / performance	Major	Service mediating network connections according to different qualities	There should be a service which lists different network connections depending on specified properties (connection speed, encryption). Devices can then negotiate such connections with remote devices, without the need to take care about the networking details	In 9 out of 10 cases devices should be able to automatically negotiate their networking condition.	✔				
188	Non-functional / operational	Major	Conflict prevention service	Certain combinations of multiple services' functionality can lead to contradicting instructions. A conflict prevention component should exist, that checks for agreeable combination of services.	Service combinations that lead to contradicting instructions are prohibited by a conflict prevention service.	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
216	Functional	Major	The middleware should have a graceful degradation service	Services should be organised in a cascade of services in order to allow an orchestration of services providing best possible services down to basic services automatically, according to their availability	Service orchestration is possible in a hierarchical way. An automatic selection of the best service is possible within max. 500 msec.	✔				
~~221~~	~~Functional~~	~~Major~~	~~Policy should handle the possible actions~~	~~Automatic system actions should be based on well defined policies to avoid conflicts.~~	~~All automatic actions are policy based.~~					
~~233~~	~~Functional~~	~~Major~~	~~Self-healing function of middleware~~	~~To ensure robustness and reliability, the middleware should dispose of self-healing and self-reconfiguration abilities.~~	~~A breakdown of service components should be automatically intercepted in 65% of the cases~~					
258	Functional	Major	Automatic software updates	Hydra middleware should prevent the need to manually update software	Support for automatic software updates		✔	✔	✔	
291	Non-functional / usability	Major	Quality of Service as search criteria for service selection	The selection of appropriate services for a given task requires the reflection of QoS-related search criteria such as cost, performance, etc.	QoS-criteria can be used in the selection of services in 95% of all cases	✔				
~~292~~	~~Functional~~	~~Major~~	~~Self-diagnosis of devices~~	~~To enhance the robustness of a Hydra system, devices should be able to check its own diagnostic state and report errors to an appropriate component~~	~~Hydra Devices can conduct self-diagnosis and detect / report failures in operation in 98% of all cases.~~					
293	Non-functional / maintainability	Major	Documentation of API and basic services	To enhance the developers' productivity, the API and the basic services provided by the middleware must be documented.	Documentation is available for API and basic services.	✔				
~~294~~	~~Functional~~	~~Major~~	~~Central service registry~~	~~Services announce their availability and unified a description of their functionality in a central service registry. Clients (users or other services) query that registry to find an appropriate service for their needs.~~	~~A central service registry exists. Services announce their availability and describe their functionality in a unified from. Clients can query the registry to find an appropriate service.~~					

Table 9: WP3 - middleware layer

4.4.3 Middleware requirements for WP3 - Architecture Design Specification: devices

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
33	Functional	Critical	Enable manufacturers to develop devices and applications that can be connected to Hydra	The hydra SDK should provide the manufacturers with an API to develop devices that can be connected to the hydra network.	APIs are available to develop devices that can be connected to the hydra network		✔			
151	Functional	Critical	Devices send events when their status changes	This alleviates the problem of always having to poll for a device's status, when another device is interested in that status.	10 status changes at device level result in 10 events sent	✔				
189	Functional	Critical	Plug and play support for adding devices	Devices should be accessible as soon as they are discoverable and with the need for the developer to implement this functionality. This should be something like Plug'n'Play in operating systems.	Plug and play mechanism for inclusion of newly detected devices is done by the middleware	✔				
247	Functional	Critical	Integrate non-Hydra devices with an existing Hydra environment	For Hydra to be inclusive and able to provide value beyond what developers has intentionally enabled, third parties have to be able to integrate their devices.	90% of Non-Hydra devices can be integrated in a Hydra environment	✔				✔
288	Functional	Critical	Query devices for their functionality	Enable developers to get information about the offered functions of a certain device in an ad-hoc manner	All Hydra enabled devices can be queried for their supported functionality	✔				
146	Functional	Major	Report errors in devices	Devices should be able to report errors	The API provides at least one interface for reporting all kinds of possible errors to the middleware	✔				
204	Non-functional / performance	Major	Devices have automatic error diagnostics	The devices should perform their own diagnostic test to provide their status upon request of the middleware in order to save performance and increase robustness and scalability	In nine out of ten cases a request of the middleware should result in a valid status	✔		✔		
~~226~~	~~Functional~~	~~Major~~	~~Device ontology should be available~~	~~In order to be able to integrate devices in an ad-hoc manner a device ontology must exist allowing to exchange basic information of services~~	~~In 90% of all cases, devices can be integrated in an ad-hoc manner.~~					
~~348~~	~~Functional~~	~~Major~~	~~Detect errors in devices~~	~~there should be specification language which allows the middleware to detect errors in a device~~	~~In nine out of ten cases the Middleware is able to detect errors in devices.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~153~~	~~Functional~~	~~Minor~~	~~Automatic generation of user interface~~	~~Manufacturers describe their devices in a special description language which can be used to automatically generate user interfaces for each device.~~	~~a user interface generator for all devices with standard capabilities exists~~					

Table 10: WP3 - devices

4.4.4 Middleware requirements for WP3 - Architecture Design Specification: device integration

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
14	Constraint / assumption	Critical	Automatic device discovery	In order to be able to ad-hoc enter a device into an environment	90% of devices brought into a new environment should be automatically discovered	✔				✔
160	Functional	Critical	Search masks for device/service discovery	When the developer needs a service he wants to be able to define search criteria for discovery of services	Search criteria can be specified and are respected by search services	✔				
288	Functional	Critical	Query devices for their functionality	Enable developers to get information about the offered functions of a certain device in an ad-hoc manner	All Hydra enabled devices can be queried for their supported functionality	✔		✔		

Table 11: WP3 - device integration

4.4.5 Middleware requirements for WP3 - Architecture Design Specification: communication

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
155	Non-functional / maintainability	Critical	All communication occurs through a central communication unit	Application developers need total control over a Hydra system. Decentralized communication is considered as not feasible.	Communication and coordination happens through centralized unit.	✔				
158	Functional	Critical	There should be a hook-up-service	When the developer creates a new application/device he wants to have a broker that can supply him with all available services that match certain criteria.	A request for a specific service according to specific keywords results in the provision of the corresponding service in 8 out of 10 cases	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
288	Functional	Critical	Query devices for their functionality	Enable developers to get information about the offered functions of a certain device in an ad-hoc manner	All Hydra enabled devices can be queried for their supported functionality	✔				
154	Non-functional / usability	Major	Physical details of communication are invisible to the developer	Developer is only interested in getting messages to other devices and (very often) not in how they get there	Developer can build up basic communication links between two devices without having to know what the physical transport layer looks like.	✔				
182	Non-functional / operational	Major	Middleware realises communication	The developer doesn't need to care about how to communicate between devices. The communication between the devices is handled by the middleware	Middleware handles all communication without the need of the developer to implement communication code	✔				
~~197~~	~~Functional~~	~~Major~~	~~Services define their communication needs in terms of needed QoS parameters~~	~~The services define their communication needs in terms of needed QoS parameters (needed bandwidth, needed quality...) without specifying the technical details. The middleware is free to choose the appropriate networking matching the specified needs~~	~~Every service specifies its QoS parameters~~					
291	Non-functional / usability	Major	Quality of Service as search criteria for service selection	The selection of appropriate services for a given task requires the reflection of QoS-related search criteria such as cost, performance, etc.	QoS-criteria can be used in the selection of services in 95% of all cases	✔				

Table 12: WP3 - communication

4.4.6 Middleware requirements for WP3 - Architecture Design Specification: configurability

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
25	Functional	Critical	Overwriting system decisions	Possibly dangerous outcomes of system decisions must be over writable by end-users	End-users can overwrite 90% of the application decisions	✔				
247	Functional	Critical	Integrate non-Hydra devices with an existing Hydra environment	For Hydra to be inclusive and able to provide value beyond what developers has intentionally enabled, third parties have to be able to integrate their devices.	90% of Non-Hydra devices can be integrated in a Hydra environment	✔				✔

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
26	Non-functional / usability	Major	Central configuration	In order to enhance the system's usability, all Hydra components should be manageable over a single component.	The configuration and administration of a Hydra system occurs via a central component.	✔				
27	Non-functional / usability	Major	Enable configuration for end-users	Users want to configure the system and perform changes to the application with ease	90% of the end-users are able to change the behaviour of their application	✔	✔	✔	✔	✔
~~177~~	~~Functional~~	~~Major~~	~~Dynamic scheduling of resource usage~~	~~Dynamic scheduling of resource utilisation enables for applications to tailor their behaviour dynamically so as to extract the maximum performance from the available resources and services, increases fault tolerance and cope with unexpected situations.~~	~~Application is able to re-schedule resource utilisation in 80% of single resource failure cases, if there the suitable resource(s) for substitution exists.~~					
184	Non-functional / operational	Major	Configuration with text files	In order to configure the middleware, a configuration file in text format, e.g. XML, should be used. This makes the developers' life easier, since such a configuration allows for fast changes of the behaviour of the middleware.	80% of all middleware components are configurable with a text file.	✔	✔	✔		
201	Functional	Major	Self configurable devices	Devices should be able to join (and leave) the network without any need for manual management or configuration handled by user. This feature requires the ability of devices to configure its connection and communication properties automatically.	Devices are able to join (and leave) the network without any manual user action in 80% of all cases.	✔				✔
~~216~~	~~Functional~~	~~Major~~	~~The middleware should have a graceful degradation service~~	~~Services should be organised in a cascade of services in order to allow an orchestration of services providing best possible services down to basic services automatically, according to their availability~~	~~Service orchestration is possible n a hierarchical way. An automatic selection of the best service is possible within max. 500 msec.~~					
~~226~~	~~Functional~~	~~Major~~	~~Device ontology should be available~~	~~In order to be able to integrate devices in an ad-hoc manner a device ontology must exist allowing to exchange basic information of services~~	~~In 90% of all cases, devices can be integrated in an ad-hoc manner.~~					
234	Non-functional / usability	Major	The middleware should be self descriptive	The developer should be enabled to understand all components and their interplay of the system in order to take full advantage of the Hydra Middleware	Nine out of ten developer have a clear understanding of the Hydra middleware after one week of experience	✔	✔	✔		

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
291	Non-functional / usability	Major	Quality of Service as search criteria for service selection	The selection of appropriate services for a given task requires the reflection of QoS-related search criteria such as cost, performance, etc.	QoS-criteria can be used in the selection of services in 95% of all cases	✔				
~~153~~	~~Functional~~	~~Minor~~	~~Automatic generation of user interface~~	~~Manufacturers describe their devices in a special description language which can be used to automatically generate user interfaces for each device.~~	~~a user interface generator for all devices with standard capabilities exists~~					
~~290~~	~~Functional~~	~~Minor~~	~~Share service orchestration between users~~	~~Service orchestration module should be shared between users, in order to allow a distribution of useful service orchestration to other users~~	~~Service orchestration can be shared between users~~					

Table 13: WP3 - configurability

4.4.7 Middleware requirements for WP3 - Architecture Design Specification: interfaces

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
25	Functional	Critical	Overwriting system decisions	Possibly dangerous outcomes of system decisions must be over writable by end-users	End-users can overwrite 90% of the application decisions	✔				
152	Functional	Critical	User must be able to overwrite automatism	Users dislike the idea of losing control and want to have the means to change system decisions	User can overwrite system automatisms in 90% of all cases	✔				✔
~~164~~	~~Constraint~~	~~Major~~	~~Support for Service standards~~	~~Middleware should support widely used standards for service description, discovery, orchestration and execution.~~	~~Standards defined by W3C and OASIS implemented.~~					
~~153~~	~~Functional~~	~~Minor~~	~~Automatic generation of user interface~~	~~Manufacturers describe their devices in a special description language which can be used to automatically generate user interfaces for each device.~~	~~a user interface generator for all devices with standard capabilities exists~~					

Table 14: WP3 - interfaces

4.4.8 Middleware requirements for WP3 - Architecture Design Specification: service discovery

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application

ID	Type	Priority	Description	Rationale	Fit Criteria	Middleware	SDK	DDK	IDE	Application
206	Non-functional / operational	Blocker	Middleware supports service discovery	The developer needs to query the available services during runtime	Services discovery during runtime in the Middleware results in at least 95% available services	✔				
158	Functional	Critical	There should be a hook-up-service	When the developer creates a new application/device he wants to have a broker that can supply him with all available services that match certain criteria.	A request for a specific service according to specific keywords results in the provision of the corresponding service in 8 out of 10 cases	✔				
196	Functional	Critical	Basic Service Registry	Services should register at a basic service/module of the middleware in order to provide a base for service orchestration	All services should be itemised at the Basic service registry	✔				
198	Functional	Critical	A service broker is responsible to provide services according to specific keywords	Service discovery should be enhanced by a service broker module/service as basic service of the middleware that enables the search for services according to specific keywords	Requests according to specific keywords will be provided a corresponding service in 8 out of 10 cases.	✔				
179	Non-functional / performance	Major	Dynamic resource handling	Resources (computational as well as devices) should be able to join or leave the environment whenever they choose. Could e-g. be enabled by short-lived transient services	Resources are able to join/leave the runtime middleware within less than 8 sec.	✔				✔
~~197~~	~~Functional~~	~~Major~~	~~Services define their communication needs in terms of needed QoS parameters~~	~~The services define their communication needs in terms of needed QoS parameters (needed bandwidth, needed quality...) without specifying the technical details. The middleware is free to choose the appropriate networking matching the specified needs~~	~~Every service specifies its QoS parameters~~					
~~209~~	~~Functional~~	~~Major~~	~~Middleware has a service for providing information about the technical environment / infrastructure~~	~~In order for the services to query the available infrastructure the middleware should provide such a service~~	~~A services at the middleware provides information about more than 95% of the technical environment/infrastructure~~					
291	Non-functional / usability	Major	Quality of Service as search criteria for service selection	The selection of appropriate services for a given task requires the reflection of QoS-related search criteria such as cost, performance, etc.	QoS-criteria can be used in the selection of services in 95% of all cases	✔				
157	Functional	Minor	Availability of combined services	A developer wants to easily access a higher level service which is in fact a	High level services, consisting of at least two basic services,	✔				✔

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				combination of multiple services	can be handled automatically by the middleware in five out of ten cases					

Table 15: WP3 - service discovery

4.4.9 Middleware requirements for WP3 - Architecture Design Specification: security

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
25	Functional	Critical	Overwriting system decisions	Possibly dangerous outcomes of system decisions must be over writable by end-users	End-users can overwrite 90% of the application decisions	✔				
229	Functional	Critical	Services are responsible for authentication	The single service should be responsible for authentication request in order to ensure a robust and secure system	All security critical services trigger authentication requests	✔				
~~222~~	~~Functional~~	~~Major~~	~~Role management should be handled by the middleware~~	~~Conflict resolution referring to access rights should be based on a role management~~	~~Role management is implemented~~					

Table 16: WP3 - security

4.4.10 Middleware requirements for WP3 - Architecture Design Specification: context

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
335	Functional	Critical	Location awareness / positioning support	Hydra should enable developers to write applications that depend on context, especially spatial context.	A component for acquiring spatial context exists. At any time, min. 75% of all devices attached to a Hydra system can be spatially located. Also, there is a programming model for using spatial context.	✔				✔
~~379~~	~~Functional~~	~~Major~~	~~Intelligent data fusion on real-time data~~	~~In order to derive information from sensor data a semantic interpretation on incoming data needs to be performed in a semantic way on real-time data.~~	~~Data fusion on real-time data is performed in 90% of the time without dropping real-time data~~					

Table 17: WP3 - context

4.4.11 Middleware requirements for WP3 - Architecture Design Specification: SDK

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
31	Non-functional / look and feel	Critical	An easy-to-use programming framework should be provided	The programming framework provided by the SDK should be easy to use in the sense that it is intuitive.	9 out of 10 developers recognise the IDE as intuitive.		✔	✔	✔	
33	Functional	Critical	Enable manufacturers to develop devices and applications that can be connected to Hydra	The hydra SDK should provide the manufacturers with an API to develop devices that can be connected to the hydra network.	APIs are available to develop devices that can be connected to the hydra network		✔	✔		
133	Non-functional / usability	Critical	Platform independent (meta) code base	Using only one (meta) code base for an application to be deployed on several platforms reduces development cost, time to deployment, and makes maintenance easier since the developer is not bothered by writing platform specific code.	A unique code base can be used at least on two different platforms.	✔	✔	✔		
38	Functional	Major	Compiling & debugging feature	Just like any other popular IDE, the Hydra IDE must be able to compile and debug the code.	Compiling & debugging functionality is available in the IDE.		✔			
41	Functional	Major	Hydra Developer's Companion	Complete and comprehensible documentation is very important to the hydra software developer.	Complete documentation is available. It is at least considered "very helpful" by at least 8 out of 10 developers.		✔			
~~135~~	~~Functional~~	~~Major~~	~~Migration to other platforms~~	~~The IDE should support easy migration of Hydra applications between different platforms. The IDE should contain tools for the identification of platform dependent code. Tools supporting the identification and writing of platform specific code should make the development process more easy and effective.~~	~~The IDE supports application migration at least between two different platforms.~~					
147	Functional	Major	Simple interface for exploring / testing devices	There should be an unintelligent/simple user interface which allows one to explore / test the functionality of a device out of the box. This interface is not part of the device, but can connect to all different kinds of devices.	A user interface for testing / exploring the functionality of a device exists in the SDK.		✔			
187	Non-functional / maintainab	Major	standardized API for device classes	All devices of a device class should have a set of methods that will be supported by each device. This makes it easier to	A set of methods is standardized for each device class.		✔	✔		

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
	ility			implement functionality. To get a complete list of supported methods of a device the device should support querying it and responding back. This query for a complete list of methods is an example of one standardized method.						
~~39~~	~~Functional~~	~~Minor~~	~~Cross compiling on different architectures~~	~~Hydra SDK must support cross compiling on different architectures~~	~~Cross compiling features are available in the IDE.~~					
186	Non-functional / operational	Minor	GUI for configuring middleware parameters	To make the configuration of the parameters of the middleware easier for the developer	A GUI exists for configuring the middleware	✔	✔	✔		
225	Non-functional / maintainability	Minor	Interactions and consequences of changes to services on other services should be highlighted	The developer should have a tool that helps him understand the complex interactions of services and the possible consequences of changes on one middleware service to other middleware services	A service monitor that is able to show interactions with other services is implemented	✔				

Table 18: WP3 - SDK

4.4.12 Middleware requirements for WP3 - Architecture Design Specification: IDE

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
33	Functional	Critical	Enable manufacturers to develop devices and applications that can be connected to Hydra	The hydra SDK should provide the manufacturers with an API to develop devices that can be connected to the hydra network.	APIs are available to develop devices that can be connected to the hydra network		✔	✔	✔	
~~28~~	~~Functional~~	~~Major~~	~~Emulation / simulation tool is needed~~	~~Developers need to test applications under reality-like conditions. IDE integrated software modules for real-time evaluation of software components should be available.~~	~~Emulation / simulation tools exist.~~					
~~30~~	~~Functional~~	~~Major~~	~~Security Modelling to choose services~~	~~The developer should be able to choose predefined security modules he wants to use in his application. This could be done in a~~	~~The developer can include predefined software modules for security in his application.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
			~~and devices~~	~~"Drag&Drop" way.~~						
34	Non-functional / usability	Major	The IDE must be easy to use.	If the IDE is cluttered and complex, It will refrain the users from using Hydra Middleware	80% of users should find that the IDE is easy to use				✔	
38	Functional	Major	Compiling & debugging feature	Just like any other popular IDE, the Hydra IDE must be able to compile and debug the code.	Compiling & debugging functionality is available in the IDE.				✔	
40	Functional	Major	IDE must be capable of deploying software to real devices.	The IDE must support multiple interfaces with different devices, so that the developer can not only test his code on the simulation tool but also deploy it on the actual devices through the IDE. This might require the IDE to have device specific interfaces/ drivers.	Developers can deploy their application code on real devices via the IDE.				✔	
~~131~~	~~Functional~~	~~Major~~	~~Model-based rapid development environment~~	~~Development process can be speeded up by utilising formal models (structural as well as behavioural) of applications. Using the formal models, applications could be analysed, simulated, visualised, validated against requirements and documented on various levels of abstraction.~~	~~IDE enables to use abstract models.~~					
~~29~~	~~Functional~~	~~Minor~~	~~IDE provides real-time hot plugging of software modules~~	~~The developer must be able to add modules/plug-ins and remove them from the IDE in real time.~~	~~The developer can add/delete software modules in real time.~~					
36	Non-functional / look and feel	Minor	Drag & Drop components	Drag & Drop functionality makes the programming easy for the developer	User is able to drag & drop components into the project.				✔	
37	Functional	Minor	Online Help / documentation with IDE	IDE must provide a help/ documentation so that the users can directly access the help pages to know more about the working of IDE or about deploying IDE and its various features.	Users are able to open & view help pages related to creating a new project and the corresponding steps from within the IDE.				✔	
~~44~~	~~Functional~~	~~Minor~~	~~IDE must provide support for Model Driven Architecture~~	~~The developer must be able to choose the appropriate software model for his/her project and hence the IDE must provide support for model driven architecture. The user must be able to select various models while starting his project. For example MVC architecture, Client-Server Model etc.~~	~~The user is able to select MVC Architecture for his new project.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~153~~	~~Functional~~	~~Minor~~	~~Automatic generation of user interface~~	~~Manufacturers describe their devices in a special description language which can be used to automatically generate user interfaces for each device.~~	~~a user interface generator for all devices with standard capabilities exists~~					
42	Functional	Trivial	Maintaining a History	The IDE must maintain a History cache for the previous projects. It will make it easier for the developer to access the project which he/she was programming before and resume from where he/she left.	The user is able to view the history of his actions.				✔	
43	Functional	Trivial	Undo / Redo Feature	Just in the case of any other popular IDE, hydra IDE must also have Undo/redo functionality so that the developer can go back to the previous state in case of an error.	The Hydra IDE provides undo / redo functions.				✔	

Table 19: WP3 - IDE

4.4.13 Middleware requirements for WP4 - Embedded AmI Architecture: architecture

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~171~~	~~Functional~~	~~Major~~	~~Learning user behaviour patterns~~	~~Learning of basic user behaviour patterns on device level (device configuration, sensor activation) in relation to specific users and specific security and situation contexts. Adaptation of devices enables applications to offer added value (e.g. detection of unusual situations, customized default configuration).~~	~~Device knowledge model of user behaviour can be expanded with new information.~~					
~~317~~	~~Functional~~	~~Major~~	~~UUAR: Support runtime reconfiguration~~	~~To supporting monitoring leading to adaptation, the architecture should be dynamic in the sense that components/services should be connectable in new ways at runtime~~	~~Services and devices can be connected in new ways during runtime in Hydra-based applications~~					

Table 20: WP4 - architecture

4.4.14 Middleware requirements for WP4 - Embedded AmI Architecture: devices

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
312	Non-	Major	UAAR: Support	The middleware should contain services that allow	Said services available in	✔				✔

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
	functional / operational		profiling of devices' performance	monitoring and reaction on what devices are doing. This includes monitoring response time, device load (e.g., CPU), and message interchanges per second	Hydra					
366	Non-functional / performance	Major	Web services should run on embedded devices	Service-orientation is a good match for many embedded devices. Web services will provide a gateway to many applications and it would be beneficial to be able to structure all of the communication in a system using the same primitives.	Hydra supports web services on embedded device (Initial target should be Develco's DevCom 02 ZigBee module)	✔				
~~315~~	~~Functional~~	~~Trivial~~	~~UAAR: Devices should be able to contain user interface/management interface~~	~~Given a discovered device/service, it may be complex to also discover a user interface for such a service. One approach to this is to enable the devices itself to send a user interface description (e.g., in XML) so that the discoverer may render it and allow user interaction. (This is done, e.g., in Lund University's MUI system)~~	~~Format for user interfaces defined. Support for creating, publishing, and discovering said interfaces~~					

Table 21: WP4 - devices

4.4.15 Middleware requirements for WP4 - Embedded AmI Architecture: device integration

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
318	Functional	Critical	UAAR: Devices should be able to be added to the system at runtime	It should not be necessary, e.g., to shut a building complex down to add a new device to a room :-)	Devices can be installed, discovered, and used while the Hydra runtime is running	✔				✔

Table 22: WP4 - device integration

4.4.16 Middleware requirements for WP4 - Embedded AmI Architecture: communication

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~314~~	~~Functional~~	~~Major~~	~~UAAR: Faults should be intercepted by middleware, notified to interested services~~	~~To create reliable and available systems it is essential to catch faults/partial failures before they become failures / complete failures. There needs to be uniformity in how this is done; thus it should be supported by the middleware~~	~~The middleware has support (through components/services) for sending and receiving notifications for partial failures~~					

Table 23: WP4 - communication

4.4.17 Middleware requirements for WP4 - Embedded AmI Architecture: configurability

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~334~~	~~Functional~~	~~Major~~	~~UUAR: There should be support for developing auto-configuration of certain devices~~	~~A number of use scenarios calls for the ability to bring a device home, turn it on, and have it function reasonably~~	~~The middleware supports defining auto-configuration properties and using these at runtime. This is not in conflict with security~~					

Table 24: WP4 - configurability

4.4.18 Middleware requirements for WP4 - Embedded AmI Architecture: interfaces

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
193	Functional	Major	Support for natural interaction	Natural user-system interactions as ambient interfaces, multimodal interaction, innovative interaction styles and concepts should be supported in order to hide invasion into user environment and minimise user's perception of hydra based applications. Proactive interfaces, which use interactions based on perception of user's behaviour and observations of situational context should be supported.	User-system interactions wouldn't require traditional explicit input from user (e.g. keyboard, mouse) in 10% of all cases of interactions.	✔				✔

Table 25: WP4 - interfaces

4.4.19 Middleware requirements for WP4 - Embedded AmI Architecture: context

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~190~~	~~Functional~~	~~Major~~	~~Learning situational context~~	~~Knowing situational context (based on e.g. learnt knowledge on people's actions, behaviour patterns, movement patterns, intonation, registering specific events, etc.) is essential for classification of possible situations and related actions. Necessary for guessing intent of the user.~~	~~Recognition of 50 % of all situations.~~					
191	Functional	Major	Intelligent location determination	Incorporating a wide range of location sensing techniques to obtain location information from different providers enables a reasoning engine to	Always select location determination mechanism with the	✔				✔

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				determine location with a certain probability.	highest accuracy.					
~~192~~	~~Functional~~	~~Major~~	~~Context modelling~~	~~Use knowledge models in order to specify the interrelations among context entities, to ensure common, unambiguous representation of these entities, to provide an explicit semantic representation of context, and to represent current context supports reasoning about context.~~	~~Current context represented as an instance of a knowledge model.~~					

Table 26: WP4 - context

4.4.20 Middleware requirements for WP5 - Wireless Networks and Devices: architecture

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~256~~	~~Functional~~	~~Major~~	~~Peer to peer support~~	~~There should exist the possibility to share applications and services across mobile devices~~	~~Hydra supports peer-to-peer communication.~~					

Table 27: WP5 - architecture

4.4.21 Middleware requirements for WP5 - Wireless Networks and Devices: middleware layer

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~261~~	~~Functional~~	~~Major~~	~~Bridges between different technologies~~	~~Devices of different communication technologies are supported by the use of bridges.~~	~~80% of the devices with different technologies supported can communicate~~					
~~272~~	~~Functional~~	~~Major~~	~~Small devices~~	~~The system provides a mechanism to integrate small devices with lack of memory, process capacity...~~	~~90% of the small devices are integrated into the system~~					
~~284~~	~~Functional~~	~~Major~~	~~Legacy components~~	~~Legacy components/devices should have to be integrated; thus the system should allow for the support of legacy protocols implemented by devices already in use by a potential customer~~	~~80% of old design devices are supported~~					

Table 28: WP5 - middleware layer

4.4.22 Middleware requirements for WP5 - Wireless Networks and Devices: devices

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
371	Functional	Critical	Devices classes hierarchy	Devices have different capabilities in terms of memory storage or power consumption or processing, some of them can route data, others can be discovered. Thus, devices with more capabilities (Hydra-enabled devices) have to help those with major constraints to be integrated into the system and be accessible from middleware layer, by creating a hierarchy of devices belonging to different classes.	90% of non-Hydra devices are integrated into network architecture	✔				
~~260~~	~~Functional~~	~~Major~~	~~Devices Registration~~	~~New devices will initiate their integration into the system. These devices will identify themselves and provide information about their functionality and availability.~~	~~90% of the devices are registered into the system~~					
279	Non-functional / performance	Minor	Quality of Service - Power consumption	The system must work with acceptable power consumption. Hydra system is able to minimise the amount of power needed to perform the communication between devices by the use of algorithms or information caches.	The Hydra system reduces in 10% the total power consumption wasted in communication between devices in the system in comparison with other non-Hydra systems	✔				✔

Table 29: WP5 - devices

4.4.23 Middleware requirements for WP5 - Wireless Networks and Devices: device integration

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~259~~	~~Functional~~	~~Major~~	~~Devices Discovery~~	~~The system needs elements to scan the network for new devices and to integrate them into the system.~~	~~90% of the devices are discovered~~					
~~267~~	~~Functional~~	~~Major~~	~~Failure detection~~	~~The system will be scanning the devices' network to discover failures on devices or communication failures in order to take the needed maintenance actions.~~	~~90% of the devices failures are detected and solved~~					
~~270~~	~~Functional~~	~~Major~~	~~New devices~~	~~New devices can easily be added to the system.~~	~~90% of new devices can be added to the system~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~271~~	~~Functional~~	~~Major~~	~~Remove devices~~	~~The system has to be able to detect devices that have been disconnected and remove them from the system.~~	~~90% of the disconnected devices are removed~~					

Table 30: WP5 - device integration

4.4.24 Middleware requirements for WP5 - Wireless Networks and Devices: networking

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
371	Functional	Critical	Devices classes hierarchy	Devices have different capabilities in terms of memory storage or power consumption or processing, some of them can route data, others can be discovered. Thus, devices with more capabilities (Hydra-enabled devices) have to help those with major constraints to be integrated into the system and be accessible from middleware layer, by creating a hierarchy of devices belonging to different classes.	90% of non-Hydra devices are integrated into network architecture	✔				
~~262~~	~~Functional~~	~~Major~~	~~Store information about the attached devices in a central element~~	~~A central element will store the information related to the devices integrated in the system. These information will deal with unique identifiers, functionality, availability ...~~	~~All the devices connected are registered into the central element~~					
~~269~~	~~Functional~~	~~Major~~	~~Back off protocol~~	~~the network should have a back off protocol to detect malfunction devices~~	~~90% of the malfunction devices are detected~~					
~~276~~	~~Functional~~	~~Major~~	~~New communication technologies~~	~~New communication technologies have to be easily added to the system, so that Hydra should provide means to facilitate this inclusion~~	~~80% of new technologies are supported~~					
~~349~~	~~Functional~~	~~Major~~	~~support for "global time"~~	~~In some system a link to a "global" network time service has to present to support data sync. of data. Global == for a set of nodes. It does not mean the total installation.~~	~~99% of the times a node should be able to get a global time~~					
~~352~~	~~Functional~~	~~Major~~	~~support for ad-hoc network~~	~~Getting data from one node placed out of range to the gateway must be able to route its data via other nodes in its neighbour list.~~	~~power efficient routing by using the wake time of the neighbour nodes to route the data, by only increase~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
					~~the power consumption of max 2%~~					
~~353~~	~~Functional~~	~~Major~~	~~All-ip (tcp and udp) usage~~	~~Using well know protocol will help the devices to speed up programming devices and backbone. About ipv6 or IPv4, small clusters of devices can work on a private subnet so using IPv4 could be the way but it lacks the security of IPv6. IPv6 will be used.~~	~~99% of all devices should be supported on a all ip~~					
~~355~~	~~Functional~~	~~Major~~	~~naming service~~	~~Naming and location goes hand in hand. we need a simple naming service to be able to find and name objects in the network~~	~~Less than 1% naming clash occurs in a Hydra system.~~					
~~380~~	~~Functional~~	~~Major~~	~~Ability to send a broadcast message to wake up sleeping devices~~	~~For device confidentiality some devices will remain silent until owner authorizes them to communicate; e.g. RFID tags, WiFi in private mode. Also for energy consumption reasons.~~	~~A specific message can be broadcast to the network that wakes up the device. If this message is not sent, the device remains dormant.~~					
381	Non-functional /	Major	Secure Communications	Hydra communication protocols should be secure enough in order to support privacy and protection of data	90% of the critical links are secure	✔				
382	Non-functional /	Major	Protected Communications	In some cases it may be essential that the communication is hidden as otherwise personal privacy may be in danger	90% of critical communication is protected	✔				
383	Non-functional /	Major	Communication Integrity	Data received must be the same than data sent	90% of data communication support an integrity control mechanism	✔				✔
~~386~~	~~Functional~~	~~Major~~	~~Fault tolerance network~~	~~Sensor networks have to be fault tolerant, meaning that if one node gets offline it should be replaced seamlessly.~~	~~In case of a node failure, it must be replaced by another node to provide the same service if available, within 5sec.~~					

Table 31: WP5 - networking

4.4.25 Middleware requirements for WP5 - Wireless Networks and Devices: communication

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
371	Functional	Critical	Devices classes hierarchy	Devices have different capabilities in terms of memory storage or power consumption or processing, some of them can route data, others can be discovered. Thus, devices with more capabilities (Hydra-enabled devices) have to help those with major constraints to be integrated into the system and be accessible from middleware layer, by creating a hierarchy of devices belonging to different classes.	90% of non-Hydra devices are integrated into network architecture	✔				
~~257~~	~~Functional~~	~~Major~~	~~MANET support~~	~~The developers should have the need to develop applications supporting the integration of MANETs~~	~~Middleware support mobile ad-hoc infrastructure less networks~~					
~~263~~	~~Functional~~	~~Major~~	~~Exchange information between devices~~	~~Different device types connected to the TAC system will be able to exchange information among them via messaging protocols.~~	~~90% of the messages are received by the devices~~					
~~264~~	~~Functional~~	~~Major~~	~~Common message protocol~~	~~The devices communicate with a common message protocol. Messaging protocols format will take into account to be machine readable and easy to implement by device manufacturers, among other issues~~	~~All devices registered into the system use the common message protocol~~					
265	Functional	Major	Remote access	While the tenants are on the move, they are interested to access the system's services with their mobile phones or their office PCs.	90% of the tenants can access the system remotely.					✔
275	Functional	Major	Direct communication between same technology	Devices of the same technology can directly communicate and exchange information and orders without the necessity of bridges.	90% of the devices with same technology can directly communicate					✔
278	Non-functional / operational	Major	Quality of Service - Interference between technologies	The system has to consider that some wireless devices works in the same frequency and could have interferences between them.	90% of the devices not interfere seriously with others	✔				
280	Non-functional / performance	Major	Quality of Service - Bandwidth network	The system must have sufficient bandwidth to support the communication between devices	Hydra systems must have sufficient network bandwidth to support QoS.	✔				
~~282~~	~~Functional~~	~~Major~~	~~Seamless hand-off~~	~~The user should be able to remain connected while moving through different locations~~	~~Support seamless hand-off between network technologies~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~367~~	~~Functional~~	~~Major~~	~~All Effort for QoS – Bandwidth Network~~	~~Some applications need to deliver every information package they sent~~	~~99% of data packages goes from source to sink~~					
~~368~~	~~Functional~~	~~Major~~	~~Best Effort for QoS – Bandwidth Network~~	~~In some type of applications is not needed to deliver every package sent.~~	~~Best effort QoS is supported by Hydra.~~					
~~326~~	~~Functional~~	~~Minor~~	~~UAAR: Support (residential) gateway devices at least at physical level~~	~~There are a large number of physical protocols (wired, wireless, power line, wifi) at play in a Hydra based system. It should be possible to build and reuse devices that bridge between these. Also such bridges should be able to do "semantic translations" (such as adding identity/security to the data from a simple, embedded device that does not support that)~~	~~Hydra supports gateways for bridging between different link layer communication protocols.~~					
~~330~~	~~Functional~~	~~Minor~~	~~UAAR: Communication protocols should support QoS in/between protocols~~	~~A number of applications of Hydra depend on knowing which quality of service to expect when doing a certain communication. This is true within protocols (such as when communicating via WiFi between devices) and between protocols (such as when communicating from a WiFi enabled device to a Bluetooth enabled device using a gateway/bridge)~~	~~85% of the times protocol QoS should be known~~					
381	Non-functional /	Major	Secure Communications	Hydra communication protocols should be secure enough in order to support privacy and protection of data	90% of the critical links are secure	✔				
382	Non-functional /	Major	Protected Communications	In some cases it may be essential that the communication is hidden as otherwise personal privacy may be in danger	90% of critical communication is protected	✔				
383	Non-functional /	Major	Communication Integrity	Data received must be the same than data sent	90% of data communication support an integrity control mechanism	✔				✔

Table 32: WP5 - communication

4.4.26 Middleware requirements for WP5 - Wireless Networks and Devices: interfaces

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~266~~	~~Functional~~	~~Major~~	~~Device public services~~	~~Each device will offer public interfaces to be invoked by the rest of the elements in the system.~~	~~90% of the devices have public services~~					

Table 33: WP5 - interfaces

4.4.27 Middleware requirements for WP5 - Wireless Networks and Devices: service discovery

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~336~~	~~Functional~~	~~Major~~	~~UAAR: Discovery protocol should support multiple networks~~	~~There is a need for discovery of services across multiple physical networks (e.g., Ethernet vs. GSM). Hydra should enable developers to create applications that have discoverable services on different types of networks~~	~~The service discovery protocol of Hydra able to work over multiple physical networks, finding at least 90% of the services available~~					

Table 34: WP5 - service discovery

4.4.28 Middleware requirements for WP5 - Wireless Networks and Devices: context

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
283	Functional	Minor	Location services	The Hydra system should support functionalities allowing to detect the position of people and assets	75% of devices are geo-located by the system					✔

Table 35: WP5 - context

4.4.29 Middleware requirements for WP5 - Wireless Networks and Devices: security

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
381	Non-functional /	Major	Secure Communications	Hydra communication protocols should be secure enough in order to support privacy and protection of data	90% of the critical links are secure	✔				
382	Non-	Major	Protected	In some cases it may be essential that the	90% of critical	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
	functional /		Communications	communication is hidden as otherwise personal privacy may be in danger	communication is protected					
383	Non-functional /	Major	Communication Integrity	Data received must be the same than data sent	90% of data communication support an integrity control mechanism	✔				✔

Table 36: WP5 - security

4.4.30 Middleware requirements for WP6 - SOA and MDA Middleware: architecture

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~108~~	~~Functional~~	~~Major~~	~~Device discovery~~	~~Middleware should be able to detect new device that enters the network~~	~~7 of 10 devices are discovered~~					
~~117~~	~~Functional~~	~~Major~~	~~Hydra component ontology~~	~~In order to support and ease the management of the Hydra middleware, the Hydra middleware components should be described and mapped to a corresponding Hydra middleware software component ontology.~~	~~All Hydra components can be identified through a software component ontology~~					
~~119~~	~~Functional~~	~~Major~~	~~Domain modelling support~~	~~The middleware and IDE should be able to host or interface with application domain frameworks representing core concepts and functions of specific application domains. These could in the most basic form be represented by UML Profiles, or domain ontologies.~~	~~The Hydra IDE supports at min 2 defined domain modelling frameworks.~~					
123	Non-functional / usability	Major	Support updates at run-time	The middleware should be dynamically updatable at run-time due to critical systems updates (security updates, component upgrades, etc.).	Deployed middleware should execute 70% of the dynamic updates without failure and restart	✔				
~~210~~	~~Functional~~	~~Major~~	~~Middleware should support different architectural styles~~	~~It must be possible to build systems with different architectures such as fully decentralised vs. centralised. De/centralization can pertain to: - data/knowledge - control - computation~~	~~Supports at least two different architecture styles~~					

Table 37: WP6 - architecture

4.4.31 Middleware requirements for WP6 - SOA and MDA Middleware: middleware layer

ID	Type	Priority	Description	Rationale	Fit Criteria	Middleware	SDK	DDK	IDE	Application
~~93~~	~~Functional~~	~~Major~~	~~Re-playable event logging~~	~~The Hydra system should maintain a re-playable event log of all events and tasks relevant for a specific application and its set of related devices. It should be possible to parameterize the logging functionality regarding event types and time.~~	~~History list and event logging is automatically available after the application is deployed.~~					
~~96~~	~~Functional~~	~~Major~~	~~Detect deadlocks~~	~~The middleware must have functionalities for detecting deadlocks between devices, for instance two devices that are waiting for each other to take an action.~~	~~Detects deadlocks in 7 out 10 cases~~					
~~97~~	~~Functional~~	~~Major~~	~~Detect livelocks~~	~~The middleware must be able to detect livelocks between two or more devices, i.e. devices that are constantly changing each others state back and forth.~~	~~Detects livelocks in 7 out of 10 cases~~					
~~98~~	~~Functional~~	~~Major~~	~~Detection of device failures~~	~~The system should be able to detect malfunctioning devices in order to be robust.~~	~~Malfunctioning devices are detected in 8 out of 10 cases.~~					
~~104~~	~~Functional~~	~~Major~~	~~Automatic Discovery of Services~~	~~It should be possible to configure the middleware to discover available services that meets defined criteria.~~	~~8 of 10 services are automatically discovered.~~					
~~108~~	~~Functional~~	~~Major~~	~~Device discovery~~	~~Middleware should be able to detect new device that enters the network~~	~~7 of 10 devices are discovered~~					
~~110~~	~~Functional~~	~~Major~~	~~Device Categorisation~~	~~Middleware should after discovery of device be able to categories a device based on device ontology information.~~	~~7 of 10 devices are correctly categorised and described.~~					
~~111~~	~~Functional~~	~~Major~~	~~Dynamic Web Service Binding~~	~~Middleware should be able to after device discovery and categorisation expose a new device as a web service that can be called without re-compilation.~~	~~New devices are callable and controllable in 7 out of 10 cases.~~					
~~114~~	~~Functional~~	~~Major~~	~~Semantic enabling of device web services~~	~~Middleware should be able to attach semantic descriptions to device web services based on device ontology.~~	~~7 of 10 devices are semantically enabled.~~					
115	Non-functional / operational	Major	Decomposable middleware	Middleware must consist of decomposable components to allow different deployments depending on available performance restrictions.	It is possible to deploy middleware on at least 3 different platforms.	✔	✔	✔		
~~117~~	~~Functional~~	~~Major~~	~~Hydra component~~	~~In order to support and ease the management of the Hydra middleware, the~~	~~All Hydra components can be identified through a~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
			~~ontology~~	~~Hydra middleware components should be described and mapped to a corresponding Hydra middleware software component ontology.~~	~~software component ontology~~					
118	Non-functional / operational	Major	Considering interaction device capabilities	The device should be able to collect data about the environment regarding other hydra devices in its proximity. Additionally, the system should be able to use this knowledge in adapting information sent to the interaction devices.	Interaction devices receive information that is tailored to its capabilities	✔				✔
~~120~~	~~Functional~~	~~Major~~	~~Multiple Device Virtualisations~~	~~It should be possible to have several different views/virtualisations of a device depending on context and applications.~~	~~At least 2 different virtualisations are provided~~					
122	Non-functional / usability	Major	Configurable and easy to install middleware	The middleware should be configurable and easy to install/deploy.	The average installation time is less than 1 hour.	✔	✔	✔	✔	
125	Non-functional / usability	Major	Transactional updates	It should be possible to rollback and recover from an unsuccessful update.	Rollback works in 7 out of 10 scenarios.	✔	✔	✔	✔	
127	Functional	Major	Spatial information management	In order to be able to deal with the location of devices and other actors Hydra needs to manage spatial information.	The system can refer to in 90% of all cases "where" something is with an accuracy of 80%.					✔
~~129~~	~~Functional~~	~~Major~~	~~Support for Semantic Web Standards for Device Communication~~	~~Middleware should support different semantic web standards, including OWL S, WSMO, and selected parts of WS *~~	~~Support for at least OWL S and WSMO~~					
~~210~~	~~Functional~~	~~Major~~	~~Middleware should support different architectural styles~~	~~It must be possible to build systems with different architectures such as fully decentralised vs. centralised. De/centralization can pertain to: - data/knowledge - control - computation~~	~~Supports at least two different architecture styles~~					

Table 38: WP6 - middleware layer

4.4.32 Middleware requirements for WP6 - SOA and MDA Middleware: devices

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~91~~	~~Functional~~	~~Major~~	~~Any Hydra device should have an associated description~~	~~For management, search and discovery purposes, all Hydra enabled devices should be described (classified) according to the Hydra device ontology.~~	~~Any device associated to a Hydra application is also included in the Hydra device ontology, and its description can be retrieved.~~					
218	Functional	Major	Support interaction devices	Interaction devices provide users with different forms of output (display) capabilities. This could include simple displays, tablets or more advanced units.	Interaction devices (displays) are included in the Hydra device ontology and can be mapped to the end-user interface of an application.					✔
~~325~~	~~Functional~~	~~Minor~~	~~UAAR: Support aggregation and separation of devices and services~~	~~Devices and services may exist in a separate application where they should not be influenced by nearby (wireless) devices such as in the case of an apartment. Thus it should be possible to view a set of services/devices as an aggregate that is separated and isolated from other sets of services/devices~~	~~Check support for aggregation and separation of devices/services~~					

Table 39: WP6 - devices

4.4.33 Middleware requirements for WP6 - SOA and MDA Middleware: device integration

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~91~~	~~Functional~~	~~Major~~	~~Any Hydra device should have an associated description~~	~~For management, search and discovery purposes, all Hydra enabled devices should be described (classified) according to the Hydra device ontology.~~	~~Any device associated to a Hydra application is also included in the Hydra device ontology, and its description can be retrieved.~~					
~~101~~	~~Functional~~	~~Major~~	~~Manual device ontology definition~~	~~The developer should be able to define and extend device ontologies. The IDE is required to provide descriptors for devices and device classes~~	~~The Hydra IDE supports the manual editing of devices in the framework of a device ontology.~~					
~~108~~	~~Functional~~	~~Major~~	~~Device discovery~~	~~Middleware should be able to detect new device that enters the network~~	~~7 of 10 devices are discovered~~					
~~110~~	~~Functional~~	~~Major~~	~~Device~~	~~Middleware should after discovery of device~~	~~7 of 10 devices are~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
			~~Categorisation~~	~~be able to categories a device based on device ontology information.~~	~~correctly categorised and described.~~					
118	Non-functional / operational	Major	Considering interaction device capabilities	The device should be able to collect data about the environment regarding other hydra devices in its proximity. Additionally, the system should be able to use this knowledge in adapting information sent to the interaction devices.	Interaction devices receive information that is tailored to its capabilities	✔				
218	Functional	Major	Support interaction devices	Interaction devices provide users with different forms of output (display) capabilities. This could include simple displays, tablets or more advanced units.	Interaction devices (displays) are included in the Hydra device ontology and can be mapped to the end-user interface of an application.					✔
~~359~~	~~Functional~~	~~Major~~	~~Device ontology versioning~~	~~The device ontology should be able to handle different versions of a device.~~	~~The device ontology can maintain at minimum 2 versions of any single device~~					

Table 40: WP6 - device integration

4.4.34 Middleware requirements for WP6 - SOA and MDA Middleware: configurability

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~93~~	~~Functional~~	~~Major~~	~~Re-playable event logging~~	~~The Hydra system should maintain a re-playable event log of all events and tasks relevant for a specific application and its set of related devices. It should be possible to parameterize the logging functionality regarding event types and time.~~	~~History list and event logging is automatically available after the application is deployed.~~					
123	Non-functional / usability	Major	Support updates at run-time	The middleware should be dynamically updatable at run-time due to critical systems updates (security updates, component upgrades, etc.).	Deployed middleware should execute 70% of the dynamic updates without failure and restart	✔				

Table 41: WP6 - configurability

4.4.35 Middleware requirements for WP6 - SOA and MDA Middleware: interfaces

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
218	Functional	Major	Support interaction devices	Interaction devices provide users with different forms of output (display) capabilities. This could include simple displays, tablets or more advanced units.	Interaction devices (displays) are included in the Hydra device ontology and can be mapped to the end-user interface of an application.					✔

Table 42: WP6 - interfaces

4.4.36 Middleware requirements for WP6 - SOA and MDA Middleware: service discovery

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
118	Non-functional / operational	Major	Considering interaction device capabilities	The device should be able to collect data about the environment regarding other hydra devices in its proximity. Additionally, the system should be able to use this knowledge in adapting information sent to the interaction devices.	Interaction devices receive information that is tailored to its capabilities	✔				

Table 43: WP6 - service discovery

4.4.37 Middleware requirements for WP6 - SOA and MDA Middleware: context

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
118	Non-functional / operational	Major	Considering interaction device capabilities	The device should be able to collect data about the environment regarding other hydra devices in its proximity. Additionally, the system should be able to use this knowledge in adapting information sent to the interaction devices.	Interaction devices receive information that is tailored to its capabilities	✔				
~~119~~	~~Functional~~	~~Major~~	~~Domain modelling support~~	~~The middleware and IDE should be able to host or interface with application domain frameworks representing core concepts and functions of specific application domains. These could in the most basic form be represented by UML Profiles, or domain ontologies.~~	~~The Hydra IDE supports at min 2 defined domain modelling frameworks.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
127	Functional	Major	Spatial information management	In order to be able to deal with the location of devices and other actors Hydra needs to manage spatial information.	The system can refer to in 90% of all cases "where" something is with an accuracy of 80%.					✔

Table 44: WP6 - context

4.4.38 Middleware requirements for WP6 - SOA and MDA Middleware: IDE

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~93~~	~~Functional~~	~~Major~~	~~Re-playable event logging~~	~~The Hydra system should maintain a re-playable event log of all events and tasks relevant for a specific application and its set of related devices. It should be possible to parameterize the logging functionality regarding event types and time.~~	~~History list and event logging is automatically available after the application is deployed.~~					
~~94~~	~~Functional~~	~~Major~~	~~Simulation environment~~	~~Use of a simulation environment is important for validating the rules/software interaction with devices. It can also be used for replaying the event log in order to examine unwanted system behaviour.~~	~~Simulation environment is available~~					
~~101~~	~~Functional~~	~~Major~~	~~Manual device ontology definition~~	~~The developer should be able to define and extend device ontologies. The IDE is required to provide descriptors for devices and device classes~~	~~The Hydra IDE supports the manual editing of devices in the framework of a device ontology.~~					
~~103~~	~~Functional~~	~~Major~~	~~Automatic device ontology construction~~	~~The IDE should facilitate the construction of a device ontology should be facilitated through finding and parsing product or device descriptions to annotate and produce ontology entries. The component should handle different input formats like Word, PDF, HTML, databases.~~	~~7 of 10 device descriptions can be successfully processed~~					
~~117~~	~~Functional~~	~~Major~~	~~Hydra component ontology~~	~~In order to support and ease the management of the Hydra middleware, the Hydra middleware components should be described and mapped to a corresponding Hydra middleware software component ontology.~~	~~All Hydra components can be identified through a software component ontology~~					
~~119~~	~~Functional~~	~~Major~~	~~Domain modelling support~~	~~The middleware and IDE should be able to host or interface with application domain frameworks representing core concepts and~~	~~The Hydra IDE supports at min 2 defined domain modelling frameworks.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				~~functions of specific application domains. These could in the most basic form be represented by UML Profiles, or domain ontologies.~~						
~~126~~	~~Functional~~	~~Major~~	~~Automatic Device ontology updates~~	~~The device ontology should automatically update its device descriptions.~~	~~The device ontology can detect device updates and handle that in 7 of 10 cases.~~					
127	Functional	Major	Spatial information management	In order to be able to deal with the location of devices and other actors Hydra needs to manage spatial information.	The system can refer to in 90% of all cases "where" something is with an accuracy of 80%.					✔
~~210~~	~~Functional~~	~~Major~~	~~Middleware should support different architectural styles~~	~~It must be possible to build systems with different architectures such as fully decentralised vs. centralised. De/centralization can pertain to: - data/knowledge - control - computation~~	~~Supports at least two different architecture styles~~					

Table 45: WP6 - IDE

4.4.39 Middleware requirements for WP7 - Trust, Privacy and Security: architecture

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
347	Constraint / requirement constraint	Critical	Authentication and authorisation must be resolved semantically	A Hydra enabled device cannot be assumed to know another device simply through hardware identification but needs to know WHAT security properties and related credentials this other new and unknown device adhere to. Hydra has to be able to resolve dynamic and unpredicted connections.	Virtualisation of devices is important to protect devices from attacks and balance stakeholders' security risks and ensure damage control. Therefore, both devices and users can operate with virtual identities. Two devices with respective end-users CAN connect based on semantic security description according to the security meta-model.	✔				
~~50~~	~~Functional~~	~~Major~~	~~An identity management must be provided~~	~~Hydra middleware has to provide highly sophisticated mechanisms for identity management in order to ensure that in systems featuring Hydra only authorised access to data, applications and devices is possible.~~	~~Identity management mechanisms are provided at all levels and to all stakeholders.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
55	Functional	Major	Integrity should be provided	The receiver should be able to determine whether the data was manipulated or not. This is especially necessary in eBilling.	Hydra middleware provides specific encryption mechanisms and protocols in order to improve integrity standards.					
253	Functional	Major	Hydra has to be open for user centric Identity Management	In order to build trust & security, Hydra middleware has to be open to many identity management principles, of which User Empowerment is core to overcome basic trust challenges. Hydra cannot be assumed as a "trusted party" having complete user profiles as this turn Hydra into a Single point of trust failure.	A Hydra End User has at least two non linkable identities within the same system.					
358	Functional	Major	Developer must be able to semantically define security requirements	If developers are to make devices that can co-operate through other protocols and security mechanisms, they have to be able to describe the inherent security requirements in a semantic interoperable language. It is not enough just to use a specific protocol's security as this does NOT tell WHY he uses it and WHAT he really needs for the application to proceed.	On the one hand Hydra supports the semantic description of security requirements and provides mechanisms to translate those requirements into device specific protocols automatically. On the other hand Hydra provides means in order to analyse (prospectively) existing device specific proprietary security protocols. Hydra can detect incompatibilities of different protocols' security mechanisms.					

Table 46: WP7 - architecture

4.4.40 Middleware requirements for WP7 - Trust, Privacy and Security: middleware layer

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
48	Functional	Major	Support for multilateral communication involving several security protocols.	The Hydra security system should support multilateral communication involving several security protocols.	The Hydra security framework supports mechanisms (e.g. as plug in extension) to support multilateral communication between today's and future security protocols.					

Table 47: WP7 – middleware layer

4.4.41 Middleware requirements for WP7 - Trust, Privacy and Security: devices

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
357	Constraint / requirement constraint	Critical	Hydra must support device authentication based on context and semantics	In strong security implementation, virtualisation and context isolation depend on isolation. As such Hydra has to be able to support devices that authenticate indirectly through recognition of pre-shared keys or using credentials (such as Direct Anonymous Attestation plus additional credentials) instead of through assumed identification of the physical device (such as MAC). The Security & Communication meta-model must not assume mandatory identification.	Device authentication is supported without device identification.	✔				
297	Functional	Major	Secure data erasing on Hydra enabled devices	In order to protect personal data it should be possible to securely erase data from Hydra devices even after they have been disconnected - intended and non-intended.	Hydra provides certain configuration mechanisms to easily erase personal data right on the device and - on end-user demand - even remotely.					✔

Table 48: WP7 - devices

4.4.42 Middleware requirements for WP7 - Trust, Privacy and Security: device integration

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~166~~	~~Functional~~	~~Major~~	~~Trust based orchestration~~	~~Orchestration of services and functional composition of response should be based on trust relations. Response should be composed only from responses of services that are trusted.~~	~~Functional composition of responses only from trusted services. In 0 out of 10 cases the response from an untrusted service is used.~~					

Table 49: WP7 - device integration

4.4.43 Middleware requirements for WP7 - Trust, Privacy and Security: networking

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~55~~	~~Functional~~	~~Major~~	~~Integrity should be provided~~	~~The receiver should be able to determine whether the data was manipulated or not.~~	~~Hydra middleware provides specific encryption~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				~~This is especially necessary in eBilling.~~	~~mechanisms and protocols in order to improve integrity standards.~~					
~~363~~	~~Functional~~	~~Major~~	~~The security model should support revocable keys~~	~~To prevent misuse of keys in more than one case, the security model/system should support revocable keys.~~	~~The Hydra security model supports at least one revocable key scheme.~~					
~~308~~	~~Functional~~	~~Minor~~	~~The Security Level of an existing network should be determinable~~	~~For a device entering an existing network it can be useful to determine the security level of that network. Depending on the provided security level the device can decide to enter the network or not.~~	~~Hydra middleware provides at least one mechanism enabling devices to determine the security level of an existing network.~~					
~~302~~	~~Functional~~	~~Trivial~~	~~Security Support for IPv6~~	~~Since IPv6 is going to be the future routing protocol, the Security Model must also have provisions for supporting IPv6.~~	~~The Security Model supports secure communication using IPv6.~~					

Table 50: WP7 - networking

4.4.44 Middleware requirements for WP7 - Trust, Privacy and Security: communication

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~48~~	~~Functional~~	~~Major~~	~~Support for multilateral communication involving several security protocols.~~	~~The Hydra security system should support multilateral communication involving several security protocols.~~	~~The Hydra security framework supports mechanisms (e.g. as plug-in extension) to support multilateral communication between today's and future security protocols.~~					
~~49~~	~~Functional~~	~~Major~~	~~The authenticity of a communication partner has to be ensured~~	~~For critical communication, such as ePayment, the authenticity of the communication partners has to be ensured.~~	~~Mechanisms enabling mutual authentication have to be provided. Especially Hydra enabled devices have to support authentication mechanisms.~~					
~~51~~	~~Functional~~	~~Major~~	~~Private communication must be particularly secured~~	~~Any private communication must not be monitored by any unauthorised third party. One consequence of this requirement is that we have to be able to distinguish the character of communication either by e.g. default settings, configuration or context~~	~~Hydra middleware has to provide particular mechanisms to protect communication indicated as 'private'.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				reasoning.						
55	Functional	Major	Integrity should be provided	The receiver should be able to determine whether the data was manipulated or not. This is especially necessary in eBilling.	Hydra middleware provides specific encryption mechanisms and protocols in order to improve integrity standards.					
56	Functional	Major	The system should provide different types of authentication.	To gain access to data/applications/devices, the entity should provide appropriate authentication. This can have different forms of strength. This can involve authentication by being (biometrics), knowing (shared secrets) or owning (smart cards).	At least one type of authentication, which is available to the whole Hydra system, should be provided.					
249	Functional	Major	Hydra should be open to indirect authentication	Instead of identifying the user or device, a session may be authenticated through indirect authentication where the user or device authenticates towards another entity which then authenticate towards the node. That means, instead of providing security credentials itself, a device shall be able to refer to an online third party that can verify device authenticity after interaction with the device. The third party can be a user, a server or any other entity such as another device with established credentials. Example: A pure indirect authentication could be Mobile IP in any token-based system where the device validate towards a server which then raise some level of clearance such as opening an access to the outer world in a Smart Home.	Hydra allows at least one indirect authentication scheme.					
363	Functional	Major	The security model should support revocable keys	To prevent misuse of keys in more than one case, the security model/system should support revocable keys.	The Hydra security model supports at least one revocable key scheme.					
364	Functional	Major	Hydra should be open to credential based authentication	Instead of identifying the user or device, a session may be authenticated through credentials recognised by the application such as blinded certificates, direct anonymous attestation, previously agreed tickets, reuse of previous accepted keys (e.g., PGP keys). That means the network can operate with authentication schemes using credentials without having to identify the device and/or user. The point is that identification of people or devices MUST NOT be MANDATORY. Alternative mechanisms such as credential based	Hydra allows at least one credential based authentication scheme.					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				~~authentication MUST be ALLOWED. Example: In Smart Home when a Service Agent of a Service Provider needs access to the home - instead of a door identifying the person or the device from the service agent, the Home Owner/Home System provide the Service Provider with a one-time-only token that the Service Provider is accountable for. The Service Provider can then forward this to the Service Agent who presents the token to the Home Access Control System. The Home Access Control System can accept the token as is or in real time contact the Home Owner and/or Service Provider System when the Service Agent is at the door. The System doesn't need to create the risk of identity theft by identifying the Service Agent person or device. He can use a device that create a random handle and communicate without further security requirements even though the system only has a credential proving traceability to the Service Provider.~~						
~~302~~	~~Functional~~	~~Trivial~~	~~Security Support for IPv6~~	~~Since IPv6 is going to be the future routing protocol, the Security Model must also have provisions for supporting IPv6.~~	~~The Security Model supports secure communication using IPv6.~~					

Table 51: WP7 - communication

4.4.45 Middleware requirements for WP7 - Trust, Privacy and Security: configurability

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~51~~	~~Functional~~	~~Major~~	~~Private communication must be particularly secured~~	~~Any private communication must not be monitored by any unauthorised third party. One consequence of this requirement is that we have to be able to distinguish the character of communication either by e.g. default settings, configuration or context reasoning.~~	~~Hydra middleware has to provide particular mechanisms to protect communication indicated as 'private'.~~					
297	Functional	Major	Secure data erasing on Hydra enabled devices	In order to protect personal data it should be possible to securely erase data from Hydra devices even after they have been disconnected - intended and non-intended.	Hydra provides certain configuration mechanisms to easily erase personal data right on the device and - on end-user demand - even					✔

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
					remotely.					
~~253~~	~~Functional~~	~~Major~~	~~Hydra has to be open for user centric Identity Management~~	~~In order to build trust & security, Hydra middleware has to be open to many identity management principles, of which User Empowerment is core to overcome basic trust challenges. Hydra cannot be assumed as a "trusted party" having complete user profiles as this turn Hydra into a Single point of trust failure.~~	~~A Hydra End-User has at least two non-linkable identities within the same system.~~					
~~361~~	~~Functional~~	~~Major~~	~~Protection of System Integrity~~	~~In order to prevent an inexperienced user to cause malfunctions by changing system configurations, the middleware should monitor, analyse and, if necessary, prevent or give notifications about faulty changes.~~	~~Hydra middleware provides specific encryption mechanisms and protocols in order to improve system integrity standards.~~					

Table 52: WP7 - configurability

4.4.46 Middleware requirements for WP7 - Trust, Privacy and Security: security

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
347	Constraint / requirement constraint	Critical	Authentication and authorisation must be resolved semantically	A Hydra enabled device cannot be assumed to know another device simply through hardware identification but needs to know WHAT security properties and related credentials this other new and unknown device adhere to. Hydra has to be able to resolve dynamic and unpredicted connections.	Virtualisation of devices is important to protect devices from attacks and balance stakeholders' security risks and ensure damage control. Therefore, both devices and users can operate with virtual identities. Two devices with respective end-users CAN connect based on semantic security description according to the security meta-model.	✔				
357	Constraint / requirement constraint	Critical	Hydra must support device authentication based on context and semantics	In strong security implementation, virtualisation and context isolation depend on isolation. As such Hydra has to be able to support devices that authenticate indirectly through recognition of pre-shared keys or using credentials (such as Direct Anonymous Attestation plus additional credentials) instead of through assumed identification of the physical device (such as MAC). The Security & Communication meta-	Device authentication is supported without device identification.	✔				

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				model must not assume mandatory identification.						
~~45~~	~~Functional~~	~~Major~~	~~Stored/private data must be protected~~	~~Any stored data must be protected from unauthorised access. This can be done by access control, encryption, context isolation or a combination.~~	~~Hydra provides developers at least one mechanism to protect any stored data from unauthorised access.~~					
~~48~~	~~Functional~~	~~Major~~	~~Support for multilateral communication involving several security protocols.~~	~~The Hydra security system should support multilateral communication involving several security protocols.~~	~~The Hydra security framework supports mechanisms (e.g. as plug-in extension) to support multilateral communication between today's and future security protocols.~~					
~~49~~	~~Functional~~	~~Major~~	~~The authenticity of a communication partner has to be ensured~~	~~For critical communication, such as ePayment, the authenticity of the communication partners has to be ensured.~~	~~Mechanisms enabling mutual authentication have to be provided. Especially Hydra enabled devices have to support authentication mechanisms.~~					
~~50~~	~~Functional~~	~~Major~~	~~An identity management must be provided~~	~~Hydra middleware has to provide highly sophisticated mechanisms for identity management in order to ensure that in systems featuring Hydra only authorised access to data, applications and devices is possible.~~	~~Identity management mechanisms are provided at all levels and to all stakeholders.~~					
~~51~~	~~Functional~~	~~Major~~	~~Private communication must be particularly secured~~	~~Any private communication must not be monitored by any unauthorised third party. One consequence of this requirement is that we have to be able to distinguish the character of communication either by e.g. default settings, configuration or context reasoning.~~	~~Hydra middleware has to provide particular mechanisms to protect communication indicated as 'private'.~~					
~~55~~	~~Functional~~	~~Major~~	~~Integrity should be provided~~	~~The receiver should be able to determine whether the data was manipulated or not. This is especially necessary in eBilling.~~	~~Hydra middleware provides specific encryption mechanisms and protocols in order to improve integrity standards.~~					
~~56~~	~~Functional~~	~~Major~~	~~Hydra should allow different types of authentication~~	~~To gain access to data/applications/devices, the entity should provide appropriate authentication. This can have different forms of strength. This can involve authentication by being (biometrics), knowing (shared secrets) or owning (e.g., smart cards). However, authentication by being should NOT be MANDATORY due to the lack of graceful degradation or fallback.~~	~~At least two types of authentication, which are available to the whole Hydra system, should be allowed.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
~~65~~	~~Functional~~	~~Major~~	~~Correctness of proposed cryptographic algorithms~~	~~To ensure that the whole security system is not obsolete, it has to be guaranteed that the used cryptographic algorithms are correct. Thus, the Hydra IDE/SDK should only propose Security Model(s) based on correct cryptographic algorithms.~~	~~Any Security Model proposed by Hydra IDE/SDK is based on correct cryptographic algorithms.~~					
~~79~~	~~Functional~~	~~Major~~	~~Secure cryptographic key management~~	~~As a large variety of keys will be used, for authentication, encryption, access control etc., a secure key management is needed.~~	~~A secure cryptographic key management is provided.~~					
228	Non-functional / usability	Major	User controlled trust levels	The user should always be in control of trust levels.	No automatic trust management without user approval. Trust management should be - user approved only - in 10 out of 10 cases.	✔				✔
~~249~~	~~Functional~~	~~Major~~	~~Hydra should be open to indirect authentication~~	~~Instead of identifying the user or device, a session may be authenticated through indirect authentication where the user or device authenticates towards another entity which then authenticate towards the node. That means, instead of providing security credentials itself, a device shall be able to refer to an online third party that can verify device authenticity after interaction with the device. The third party can be a user, a server or any other entity such as another device with established credentials. Example: A pure indirect authentication could be Mobile IP in any token-based system where the device validate towards a server which then raise some level of clearance such as opening an access to the outer world in a Smart Home.~~	~~Hydra allows at least one indirect authentication scheme.~~					
~~253~~	~~Functional~~	~~Major~~	~~Hydra has to be open for user centric Identity Management~~	~~In order to build trust & security, Hydra middleware has to be open to many identity management principles, of which User Empowerment is core to overcome basic trust challenges. Hydra cannot be assumed as a "trusted party" having complete user profiles as this turn Hydra into a Single point of trust failure.~~	~~A Hydra End-User has at least two non-linkable identities within the same system.~~					
~~296~~	~~Functional~~	~~Major~~	~~Adaptability of Security Model with regard to existing~~	~~In the case of already existing security systems, Hydra Security Model should be able to interoperate with them.~~	~~The Hydra Security Model can operate with already existing security systems in 9 of 10 cases.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
			~~security system(s)~~							
297	Functional	Major	Secure data erasing on Hydra enabled devices	In order to protect personal data it should be possible to securely erase data from Hydra devices even after they have been disconnected - intended and non-intended.	Hydra provides certain configuration mechanisms to easily erase personal data right on the device and - on end-user demand - even remotely.					✔
~~358~~	~~Functional~~	~~Major~~	~~Developer must be able to semantically define security requirements~~	~~If developers are to make devices that can co-operate through other protocols and security mechanisms, they have to be able to describe the inherent security requirements in a semantic interoperable language. It is not enough just to use a specific protocol's security as this does NOT tell WHY he uses it and WHAT he really needs for the application to proceed.~~	~~On the one hand Hydra supports the semantic description of security requirements and provides mechanisms to translate those requirements into device specific protocols automatically. On the other hand Hydra provides means in order to analyse (prospectively) existing device specific proprietary security protocols. Hydra can detect incompatibilities of different protocols' security mechanisms.~~					
~~361~~	~~Functional~~	~~Major~~	~~Protection of System Integrity~~	~~In order to prevent an inexperienced user to cause malfunctions by changing system configurations, the middleware should monitor, analyse and, if necessary, prevent or give notifications about faulty changes.~~	~~Hydra middleware provides specific encryption mechanisms and protocols in order to improve system integrity standards.~~					
~~363~~	~~Functional~~	~~Major~~	~~The security model should support revocable keys~~	~~To prevent misuse of keys in more than one case, the security model/system should support revocable keys.~~	~~The Hydra security model supports at least one revocable key scheme.~~					
~~364~~	~~Functional~~	~~Major~~	~~Hydra should be open to credential based authentication~~	~~Instead of identifying the user or device, a session may be authenticated through credentials recognised by the application such as blinded certificates, direct anonymous attestation, previously agreed tickets, reuse of previous accepted keys (e.g., PGP keys). That means the network can operate with authentication schemes using credentials without having to identify the device and/or user. The point is that identification of people or devices MUST NOT be MANDATORY.~~	~~Hydra allows at least one credential based authentication scheme.~~					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				~~Alternative mechanisms such as credential based authentication MUST be ALLOWED. Example: In Smart Home when a Service Agent of a Service Provider needs access to the home - instead of a door identifying the person or the device from the service agent, the Home Owner/Home System provide the Service Provider with a one-time-only token that the Service Provider is accountable for. The Service Provider can then forward this to the Service Agent who presents the token to the Home Access Control System. The Home Access Control System can accept the token as is or in real time contact the Home Owner and/or Service Provider System when the Service Agent is at the door. The System doesn't need to create the risk of identity theft by identifying the Service Agent person or device. He can use a device that create a random handle and communicate without further security requirements even though the system only has a credential proving traceability to the Service Provider.~~						
~~301~~	~~Functional~~	~~Minor~~	~~Most appropriate Security Model(s) should be proposed to developer~~	~~It could be useful during the development stage, that the developer gets support to find the best security model/systems for his device/software. Thus, the Hydra IDE/SDK should propose the most appropriate Security Model(s) to be applied on the respective application. However, the developer is not compelled to accept any proposals.~~	~~Hydra IDE/SDK always proposes at least one appropriate Security Model to the developer - "No Security Model" allowed.~~					
~~308~~	~~Functional~~	~~Minor~~	~~The Security Level of an existing network should be determinable~~	~~For a device entering an existing network it can be useful to determine the security level of that network. Depending on the provided security level the device can decide to enter the network or not.~~	~~Hydra middleware provides at least one mechanism enabling devices to determine the security level of an existing network.~~					
~~302~~	~~Functional~~	~~Trivial~~	~~Security Support for IPv6~~	~~Since IPv6 is going to be the future routing protocol, the Security Model must also have provisions for supporting IPv6.~~	~~The Security Model supports secure communication using IPv6.~~					

Table 53: WP7 - security

4.4.47 Middleware requirements for WP7 - Trust, Privacy and Security: context

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
51	Functional	Major	Private communication must be particularly secured	Any private communication must not be monitored by any unauthorised third party. One consequence of this requirement is that we have to be able to distinguish the character of communication either by e.g. default settings, configuration or context reasoning.	Hydra middleware has to provide particular mechanisms to protect communication indicated as 'private'.					

Table 54: WP7 - context

4.4.48 Middleware requirements for WP7 - Trust, Privacy and Security: DK

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
65	Functional	Major	Correctness of proposed cryptographic algorithms	To ensure that the whole security system is not obsolete, it has to be guaranteed that the used cryptographic algorithms are correct. Thus, the Hydra IDE/SDK should only propose Security Model(s) based on correct cryptographic algorithms.	Any Security Model proposed by Hydra IDE/SDK is based on correct cryptographic algorithms.					
301	Functional	Minor	Most appropriate Security Model(s) should be proposed to developer	It could be useful during the development stage, that the developer gets support to find the best security model/systems for his device/software. Thus, the Hydra IDE/SDK should propose the most appropriate Security Model(s) to be applied on the respective application. However, the developer is not compelled to accept any proposals.	Hydra IDE/SDK always proposes at least one appropriate Security Model to the developer - "No Security Model" allowed.					

Table 55: WP7 - DK

4.4.49 Middleware requirements for WP7 - Trust, Privacy and Security: IDE

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
65	Functional	Major	Correctness of proposed cryptographic algorithms	To ensure that the whole security system is not obsolete, it has to be guaranteed that the used cryptographic algorithms are correct. Thus, the Hydra IDE/SDK should only propose Security	Any Security Model proposed by Hydra IDE/SDK is based on correct cryptographic					

ID	Type	Priority	Description	Rationale	Fit Criteria	Middle ware	SDK	DDK	IDE	Application
				~~Model(s) based on correct cryptographic algorithms.~~	~~algorithms.~~					
~~301~~	~~Functional~~	~~Minor~~	~~Most appropriate Security Model(s) should be proposed to developer~~	~~It could be useful during the development stage, that the developer gets support to find the best security model/systems for his device/software. Thus, the Hydra IDE/SDK should propose the most appropriate Security Model(s) to be applied on the respective application. However, the developer is not compelled to accept any proposals.~~	~~Hydra IDE/SDK always proposes at least one appropriate Security Model to the developer - "No Security Model" allowed.~~					

Table 56: WP7 - IDE

5. Developer users validation plan

5.1 Middleware and SDK validation (first iteration)

5.1.1 Middleware (I)

Experience shows that the more immature an implementation is, the faster defects will be found. Users who are confronted with incomplete and faulty software become frustrated and can not provide much constructive feedback. So it is decided to proceed with the first middleware evaluation at an advanced stage, when the implementation of software has already reached certain robustness. As the middleware is recursively improved, the part regarding the middleware assessment is repeated in all iterations. The collected feedback allows having a constant improvement of the implemented system.

The middleware assessment is performed via the verification of the fit value fulfilment in each relevant requirement (as found in the tables of paragraph 4.4) and with mean of a questionnaire to be completed from developer users who exploited the HYDRA middleware.

First there will be a collection of data as a result of laboratory test by considering each requirement referring to the middleware. This will be the case for those quality dimensions that need a specific measurement (for example, an efficiency performance test).

On the other hand requirements that need a special evaluation, not feasible with a simple measurement, will be assessed through a set of question to be inserted in the questionnaire and then proposed to different developer users.

5.1.2 SDK

The SDK assessment will be performed in same way as it is done for the middleware, but considering only the requirements related to the SDK (paragraph 4.4). The assessment will use again both laboratory measurements and a questionnaire to be completed from developer users who exploited the HYDRA SDK.

5.2 Middleware and DDK validation (second iteration)

5.2.1 Middleware (II)

The second middleware evaluation shall arrive when the implementation of software has already been improved.

The middleware assessment is repeated as it was done in the previous cycle and it is performed via the verification of the fit value fulfilment in each relevant requirement (as found in the tables of paragraph 4.4) and with mean of a questionnaire to be completed from developer users who exploited the HYDRA middleware.

First there will be the collection of data as a result of laboratory test by considering each requirement referring to the middleware. This will be the case for those quality dimensions that need a specific measurement. Other requirements will be assessed through the questionnaire proposed to developer users.

5.2.2 DDK evaluation

The DDK assessment will be performed considering its relevant requirements (paragraph 4.4). The assessment will use again both laboratory measurements and a questionnaire to be completed from developer users who exploited the HYDRA DDK.

5.3 Middleware and IDE validation (third iteration)

5.3.1 Middleware (III)

The third (and last) middleware evaluation will be performed towards the end of software implementation. At this stage the software shall be in the release version and at a consistent level, being improved during previous cycles.

The middleware assessment is repeated again via verification of the fit value against relevant requirement and with mean of a questionnaire to be completed from developer users. First there will be the collection of data as a result of laboratory test. Then issues that need a special evaluation will be assessed through a set of question to be inserted in the questionnaire.

In this case the outcomes won't enter again the loop, as the project iterations are completed. The results and suggestions will be inserted in a list of recommendations that will sketch the most appropriate next steps to be followed in order to allow a complete and successful exploitation of the HYDRA middleware.

5.3.2 IDE evaluation

The IDE assessment will be performed not exactly at the end of the last iteration, at M48, as this represents the deadline of the project. A reliable version of the IDE is expected at the end of M46, so that during the last two months of project activity it will be possible to complete the last HYDRA prototype evaluation.

The software testing will be done considering requirements relevant for the IDE indicated in paragraph 4.4. The assessment will use again both laboratory measurements and a questionnaire to be completed from developer users who exploited the HYDRA IDE.

The results will be inserted in the list of recommendations for a successful HYDRA exploitation.

5.4 Validation process

The process followed is similar in all validation cycles and foresees fixed steps to pursue with exactitude: an initial preparation part, a validation activity done in laboratory and with developer user experts, the collection and analysis of the outcomes and a final part for feed back into the loop the results for addressing the next steps of the project.

5.4.1 Steps

5.4.1.1 Prepare the validation activities

The first part of the task defines and briefly describes the subject of the validation, with insight of the technical platform and components to be validated. The validation templates, to be prepared before the evaluation activities take place, identify the actors, i.e. the test persons, and eventually recruit them (if they do not belong to the project consortium). It is useful also to draft the corresponding user scenarios that the actors need to go through as part of the validation. This allows to customise the validation procedure, selecting from already existing methods that are considered appropriate. The scenario definition should be used also for defining eventual augmentation, so that the validation doesn't exceed certain fixed boundaries.

In case the fit criterion has to be measured with mean of laboratory test, the validation template, prepared from the HYDRA technical partners (depending on their expertise), has to clearly indicate information such as testing method, statistical processes to be applied, number of trials for

considering trustworthy the result, boundary conditions and any other data necessary to conduct a reliable experiment.

In case the fit criterion refers to a quality dimension which is not related to a numerical measurement (as an example, user satisfaction or user acceptance) it will be prepared a questionnaire, drafted from all task participants, aimed at measuring the specific argument. For this second part of the evaluation there will be the possibility to investigate if questionnaire templates (or part of them) already exist which are considered suitable for the intended trial, and especially for assessing the quality of use in software. A tailor-made questionnaire for the user group has to be prepared, where provided functionality, added value and other related topics are investigated. This kind of questionnaires will give other information for depicting a quality space through performance, productivity and added value dimensions from a user point-of-view.

5.4.1.2 Conduct the validation activities

Once the validation template and questionnaire are completed, the test person has to follow the indications given and to perform the validation, which can be a laboratory test or a trial of the middleware/SDK before answering the questions. Different expert evaluators do not find the same defects, and not in the same order. It is therefore advisable to use at least two or three experts (even more if available). In later development stages longer test sessions should be foreseen.

The user has to be assisted from the working group who prepared the evaluation activities, in case something is not clear or misleading. The conduction of the validation itself from developer users should be linear if the planning is done carefully and the validation templates are prepared with sufficient attention.

5.4.1.3 Analyse data

After completing the test trials and questionnaire submission, it is time to analyse the results and formulate the conclusions. This part is strongly dependent on the testing method applied, decided during the first phase.

Data analysis will be performed with different approaches for the laboratory measurements figures and the questionnaire responses. While the first will hopefully result in immediate numbers, the examination of questionnaires will be made with both quantitative (statistical calculations on multiple choice questions) and qualitative analysis (comments and observations emerging from open questions).

5.4.1.4 Feedback results back to the loop

The validation results will contributes to the project success just if it is the project plan foresees that all the user feedbacks are given back to the developers of the system, with an iterative approach. The data emerged in the previous analysis will be distributed to the HYDRA consortium, and they will be mean for refining the user requirements and improving the system characteristics.

This is the case also in the last iteration, where the assessment result won't enter again the project but will be inserted in a list of further recommendations.

5.4.2 Planning

The planning of the first validation cycle, composed from the phases just described, is depicted in the following Figures, one for each of the assessment iterations. The timetable indicated is referring to the HYDRA GANTT and overall project (work package and task) planning. The timing is elaborated in relation to Figure 3, depicting the HYDRA prototypes timeplan, to which the validation plan is adapted.

5.4.2.1 First iteration

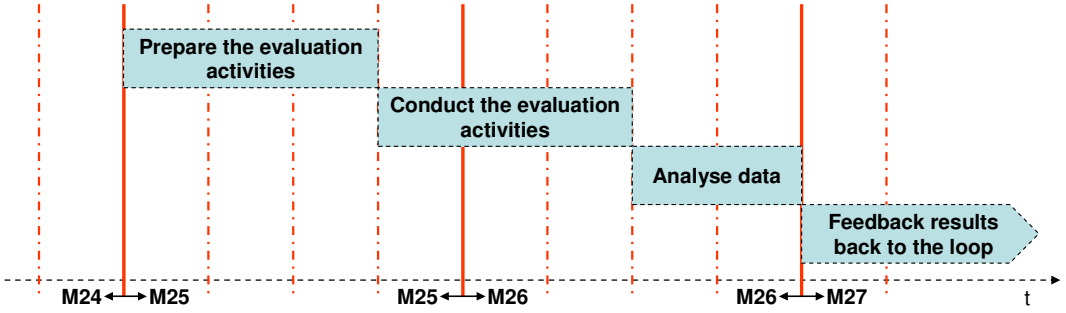

Figure 4: user validation first iteration - time plan

The first iteration will take place from the end of M24 till the end of M26. First and second iteration will last for a period of two months, while the third iteration shall last for three months.

5.4.2.2 Second iteration

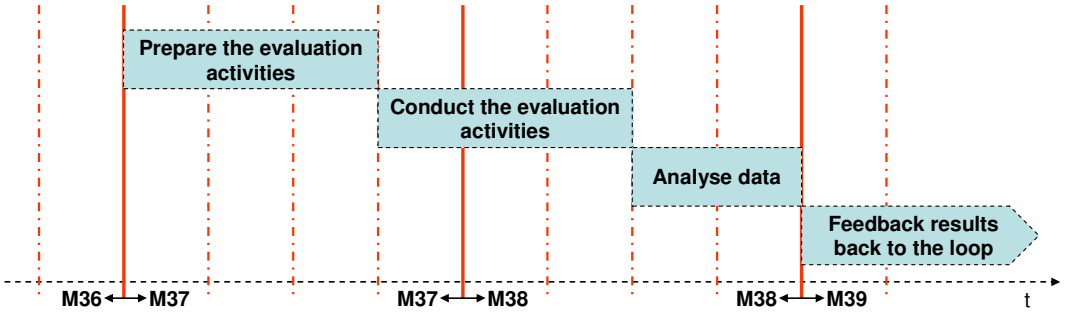

Figure 5: user validation second iteration - time plan

The second iteration will take place from the end of M36 till the end of M38.

5.4.2.3 Third iteration

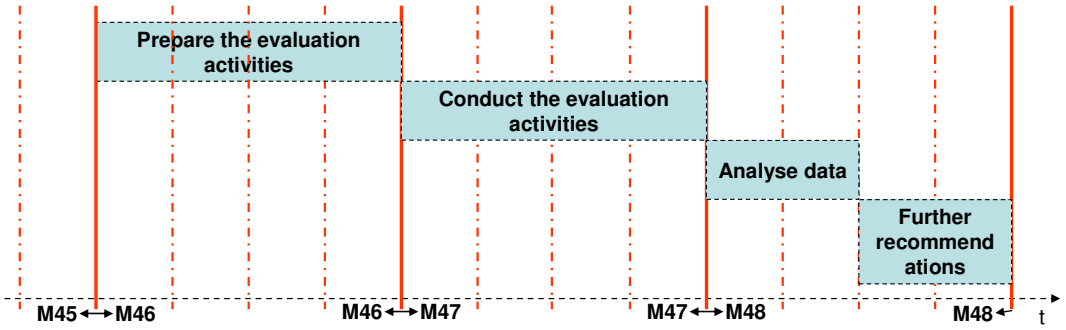

Figure 6: user validation last iteration - time plan (developer users)

The third and last iteration will take place from the beginning of M46 till the end of M48.

6. End-users validation plan

6.1 Object of the validation

A foreword underlining the background of this evaluation part is necessary for end-user validation. The HYDRA project scope is to develop tools (SDK, DDK, IDE) in order to permit an intelligent software implementation among several platforms in a straightforward, but comprehensive way. The applications developed are considered means to an end, an exemplification for each of the three selected domains to understand potentialities of developed tools. From this point of view the end-users' evaluation is not considered the central point of the HYDRA middleware assessment, but an important added value that is useful to understand how the HYDRA-implemented hardware and software could work, and what a first feedback from the interested stakeholders could be. HYDRA platform should be able to support many different applications, so it is not important if each built test-application is fully adequate for its purpose, but it is crucial to evaluate the applications to understand the sources of their shortcomings. As an example if an application doesn't work because a third-party GUI library causes bad usability it is ok; if the users cannot perform the application because the security scheme used in the HYDRA middleware is a severe performance bottleneck then it is necessary to understanding what is not working.

The object of the evaluation shall be the applications implemented from project developers to illustrate the HYDRA concepts and explain the middleware potentialities. The chosen domains are building automation, healthcare and agriculture.

At the time of writing the unique implemented demonstrator is concerned with the first domain, so actually it is not yet possible to make the complete planning for the validation activity to be brought up with final user (stakeholders) in the application. As a consequence in the next sections there is a brief introduction of the selected vision scenarios, one for each field, which will be the base to build the real demonstrators. For each we will consider a technical and an introductory evaluation based on the requirements collected in paragraph 4.4, as some of the requirements are referring also to the application and not just to the prototypes (middleware, SDK, DDK and IDE).

Before the beginning of the last iteration, where the end-user validation plan is concentrated, and there will be a better knowledge of how the applications were implemented. It is necessary to specify better the end-user validation activities, especially in terms of quality dimensions not yet analysed but which are essential to be investigated and how to proceed with the assessment.

6.1.1 User scenarios

The gathering of user scenarios in the in three domains building automation, healthcare, and agriculture was performed in WP2 and refined in WP3 with the aim of establishing a set of useful and realistic usage pictures on a possible and not too far future. Creating scenarios of end-user behaviour and interaction with platform functionality is a really useful instrument for identifying key technological and socio-economic hints to collect end-user prioritised needs.

Scenarios are snapshots of future common situations that help to understand how the HYDRA system impact would be. They provide coherent and comprehensive descriptions of plausible futures built on the imagined interaction of key trends.

In the building automation domain the "Beehive" scenario has been drawn, highlighting new management features that will be available in the future housing complexes. The field of intelligent homes comprises an enormous variety of technologies, across commercial, industrial, institutional and domestic buildings, including energy management systems and building controls. HYDRA has the potential to have a deep impact on the interaction of technologies and systems in the building automation domain.

Healthcare is the second of the tree user domains in which the project will be demonstrated and validated. In this sector the "Overload" scenario has been sketched: a patient with a diabetes type II disease shall be able maintain his normal habitudes, also at work, with mean of the HYDRA interconnected devices. The healthcare sectors are extremely complicated in terms of how healthcare services are delivered and financed. HYDRA can enable applications that can improve the delivery of healthcare services by securing higher quality of treatment, improved access to care, avoidance of unnecessary hospitalisation and more efficient delivery of healthcare services at lower costs.

The "From Farm to Fork" scenario has been chosen in the agriculture area, foreseeing an enrichment of the actual potentialities for food traceability. The most ancient economic activity of human beings has been commonly identified as a traditional activity and – unfortunately – very often considered a static sector, unable to actively enhance the economic and social development of our countries. The most important challenge for the agriculture domain is the adaptation of traditional farming method with more innovative approaches, where the use of uprising technologies is not seen as a treat, but as an opportunity.

The complete storylines may be found in HYDRA Deliverable "D2.1 Scenarios for usage of HYDRA in 3 different domains".

From the scenarios and their description a formal collection of all relevant user requirements have been derived. Functional user requirements have been taken into account in previous paragraphs, while the present assessment will involve the most important aspects of user expectations in the chosen application domains.

6.2 Selection of end-users

End users performing the evaluation will be selected among the actors and stakeholders found in the different scenarios previously described. Testing users differ depending on the domain to which the validation is applied.

6.2.1 Building automation

Stakeholders

The primary stakeholder class consists of end-users of services and applications that have been developed using the Hydra middleware. In the Beehive scenario there is a large number of primary stakeholders, from residents of apartment blocks to customers and staff attending the shopping mall and the fitness complex. Also service technicians that have to maintain the HYDRA enabled installed system are considered a primary stakeholder, together with authorised vendors, subcontractors and service organisation to access the building management system. They are in charge of service and maintaining the building management systems and have to get physical access to the buildings and web based access to the systems.

Secondary stakeholders are those actors who are directly accountable for the end-user experience. In the scenario, secondary stakeholders are facility management companies, system integrators and building automation vendors.

Tertiary stakeholders are all actors supporting the previous ones, like the Danish government, housing complex designers and constructors, facility management companies and also standardisation and certification authorities.

Stakeholder selection

The primary stakeholders are the main target users to assess the HYDRA enabled experience. They represent the first "tester" for the example applications, giving a preliminary feedback on how the new technology was able to improve their lives. The secondary stakeholders are represented within the HYDRA validation framework from the application developers, so they are considered as "bulked"

inside the HYDRA system. The validation that could be fulfilled from this level of actors is comprised in the SDK, DDK and IDE validation, so it is not considered in this section. The last level of actors is very broad and is not directly involved in the HYDRA usage, so the tertiary stakeholders won't be involved in the valuation plan.

For the selection of end users of the building automation domain it is enough to identify a group of demanding home owners who are familiar with technological devices (this is a representative user for that market sector, as a low demanding home owner would not be interested in buying advanced instruments that are more expensive than basic commodities). These users may be found also between colleagues or partners who have certain knowledge of domestic systems.

The main features to be evaluated by the *home owner stakeholder* are clustered in two major groups.

- Private users:
  - enhanced convenience and comfort resulting from Hydra's ambient intelligence features
  - enhanced security of their e-systems
  - advanced remote control and monitoring features
  - seamless intelligent networking
- Professional users:
  - time and cost saving (through remote services and remote error detection)
  - interactive support services for troubleshooting
  - proactive security management features for the access control to the building.

The listed features shall be taken into account and cross compared with the selected requirements to be assessed (tables of paragraph 4.4) and the relevant quality dimensions derived in section 4.3 (summarised in the next Table).

Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
Performance – functionality	Rating by users	Global rate		Better than the average	Above average	Questionnaire, Positioning
Performance - effectiveness	Rating by users	Global rate		Better than the average	Above average	Questionnaire, Positioning
Subjective assessment quality	User satisfaction	Global rate			Above average	Questionnaire
Learning effort	Time to learn (end user)	Minutes	Below average	Better than the average	Above average	Learning time measurements
Cognitive workload	Time to learn	Rate	More than 50% of working time for more than one week		20% of working time for less than one week	Learning time measurements
Added Value	Rating by users	Number of benefits mentioned			Above average	Questionnaire, Positioning
Safety	Rating by users Number of vulnerabilities	Global rate of vulnerabilities number		Above average	No vulnerabilities	Conjoint Measurements
Privacy	Rating by	Global rate of		Above	No vulnerabilities	Questionnaire,

	users (if applicable) Number of vulnerabilities	vulnerabilities perceived		average		Positioning

Table 57: selected quality dimensions for the user assessment

6.2.2 Healthcare

Stakeholders

The primary stakeholders identified from the scenario are patients (especially people participating in virtual communities), doctors, practitioners, nurses and other officials working in health services.

The secondary stakeholders are represented by the UK government healthcare provider and local hospitals, the organisation supporting a virtual group of patients with medical conditions (also the ICT department), the UK tax authorities and public service providers.

Tertiary stakeholders are all other actors supporting the primary and secondary stakeholders. Identified tertiary stakeholders in the scenario are a truck company, third party companies offering trusted authentication, a medical device and handset manufacturer, content providers and network providers.

Stakeholder selection

Primary stakeholders are the only users to assess the HYDRA enabled experience. They will test the example application and give a preliminary feedback on how the new technology works. The secondary and tertiary stakeholders are represented from the HYDRA application developers and sub-sequent suppliers, so they are considered before the user validation (in fact they "prepare" the object to be assessed).

For selection of end users of the healthcare domain it is necessary to identify a few practitioners/nurses and their patients who will receive a short training on how to use the HYDRA (and non-HYDRA) devices exploited in the scenario. These users shall be found among experts and technicians working in the relative sector.

The main features valued by the two relevant classes of stakeholders are:

- Medical assistant:
  - automatic and contextual feedback to patients and end-users
  - statistical tools for analysis and comparison of medical data
- Patient:
  - support in compliance, medication intake, household task
- Both:
  - access to easy-to-understand information tailored to specific needs
  - convenient and context aware services
  - security, pervasiveness, mobility, robustness of the services and user-friendly interfaces.

The features shall be cross compared with relevant requirements and the quality dimensions summarised in Table 57.

6.2.3 Agriculture

Stakeholders

Primary stakeholders in the described scenario are consumers but also farmers, the transport company, different actors in the value chain of the food industry, the wholesaler/traders and the retailers. The first stakeholder level is generally intended as the *private* and *professional end-users* featuring and using the traceability concept from HYDRA enabled devices.

Secondary stakeholders are technology integrators, software developers and manufacturers of sensors and devices, companies producing hardware/firmware components with Hydra integrated features, vendors and traders of the HYDRA enabled products. The interest of the different stakeholders may differ from category to another, but all of them are interested in exploiting the HYDRA enabled technology in order to allow the product chain traceability.

The tertiary stakeholder level includes a wide range of commodity producers that are related to the business framework, e.g. a third party certification entity. Also other relevant actors exist that are not mentioned in the scenario text description (content provider and network provider).

Stakeholder selection

Even in the agriculture domain the validation will focus on the primary stakeholder level. The end-users pertaining to this class do not know which technologies were used to let them be aware of the complete product chain, they just choose and buy the product because it is as they expect it to be. Their wish is to have a product with specific characteristics and so they are suitable to be testers for the example application. Other stakeholders are represented within the HYDRA validation framework from the application developers and the sub-sequent suppliers, so they are considered as before the user validation (in fact the "prepare" the object to be assessed).

The selection of the agriculture end users seems not as immediate as in the previous domains, because farmers and the food industry after them are quite a long chain. So in order to simplify the work it is enough to select a group of agriculture experts who are familiar with technological devices (this is also representative for that market sector, as normal farmers are less attracted from advanced instruments). These users may be found externally from the Consortium.

All the end-users are characterised from similar interests:

- statistical data for analysis and comparison of products
- seamless integration of product data (information awareness of product history and data addition) coherent with the rest of the value chain
- access to simplified and easy-to-understand information
- convenient and context aware services
- security, pervasiveness, mobility of the services and user-friendly interfaces.

The features shall be cross compared with relevant requirements and the quality dimensions summarised in Table 57.

6.3 Validation process

The process followed is the same as in all previously described validation cycles, with fixed steps to be followed: preparation part, validation activity done in a laboratory environment and with end-users, collection and analysis of the outcomes and a final part for drawing the main results and addressing recommendations and next steps.

6.3.1 Steps

6.3.1.1 Prepare the evaluation activities

End user evaluation activities are prepared (together with and) similarly to what is done for the developer user assessment, with differences on the templates to be drafted.

The first part of the task defines and briefly describes the subject of the validation, with insight of the application. Before evaluation activities take place test persons are identified and recruited. For the applications testing this is needed because building automation, healthcare and agriculture are domains whose expertise is not present in the project consortium. After the stakeholder choice is done, it is fundamental to draft user scenarios that end users need to go through. The scenario definition is needed also for giving fixed boundaries to the validation.

The quality dimensions referring to requirements where the fit criterion needs a numerical measurement shall be assessed by drafting and completing a validation template (prepared from the HYDRA technical partners) during laboratory test.

The quality dimension related to "empirical" measurement (as an example, user satisfaction or user acceptance) will be tested through a questionnaire aimed at measuring the specific argument and drafted from task participants. Each domain will have a questionnaire with several questions related to the specialised requirements for that specific domain. The tailor-made questionnaire for end users group will have to investigate all topics (i.e., quality dimensions) that were not yet consider all along the requirements list (paragraph 4.4).

6.3.1.2 Conduct the evaluation activities

The test persons have to perform the validation along the lines of the validation template (in the laboratory test for numerical measurements) and on the questionnaire while answering the enquiries. For the second part it is worthwhile to use a certain number of evaluators, to be determined by considering both the effort constraints derived from a limited (time and cost) evaluation interval and the need to obtain a statistically significant analysis. The Consortium must be able to determine how large a sample is needed to ascertain that a requirement is satisfied; in case that requirement is stated using an appropriate quantitative metric. The end user has to be assisted in case something is not clear or misleading while the validation scenario is done.

6.3.1.3 Analyse data

Data analysis will be characterised from the same procedures as seen in paragraph 5.4.1.3. Testing method, statistical processes to be applied, number of trials/questions' response and any other data will be used as parameters to study the experiment outcome.

The examination of questionnaires will be made with both quantitative (statistical calculations on multiple choice questions) and qualitative analysis (comments and observations emerging from open questions).

6.3.1.4 Recommendations and further steps

The validation results in the final loop will draw a set of recommendations for the improvement of the HYDRA system that will contribute to the project success. Hereafter the data analysis will not be used for refining the user requirements, but to sketch a roadmap for improving the system characteristics and achieving a market appealing product, that is helpful for software implementation for embedded platforms.

6.3.2 Planning

The last validation cycle, depicted in Figure 6 (and repeated here for reader's convenience, see paragraph 5.4.2.3), will be held in the last three months of the project. The timetable indicated is referring to the HYDRA GANTT and overall work package and task planning.

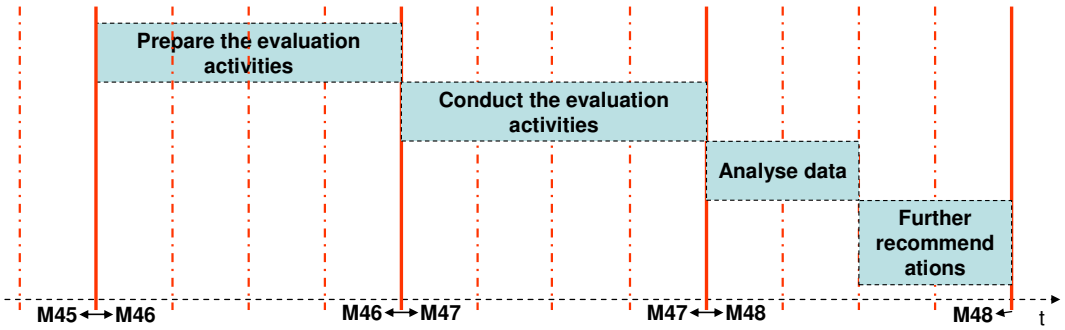

Figure 7: user validation last iteration - time plan (end users)

7. Conclusions

This document tries to select a shared approach and common ways to identify critical aspects of the HYDRA middleware with adequate means of validating project results. The presence of different types of users (developers or end-users) requires that the analysis be split into two parts: the prototypes assessment and the application examples evaluation. HYDRA validation is so divided into two tasks, whose completion ensures a full investigation of possible in and outs of project results.

The content of this report has three major aims.

1. To give a basic framework on user validation concepts.

   User validation is a well established activity that studies how quality attributes have been considered during system implementation. In essence, this part briefly introduces quality-based software engineering evaluation techniques.

2. To apply this framework to the HYDRA prototypes.

   User requirements and relative fit criteria are the principal mean to perform the validation of prototypes. The focus is towards operational quality attributes rather than development oriented ones, even if the HYDRA iterations shall provide testing for both middleware and toolkits (to be used at development time). The quality attribute requirements presented in the report are considered an initial list that is subject to modification during project cycles.

3. To apply this framework to the application examples (building automation, healthcare, agriculture).

   Part of the user requirements (and correspondent fit criteria) may also be applied to perform the applications evaluation. The focus is towards quality attributes rather than technical ones (these are better investigated trough laboratory trials). The selected list is considered a starting point subject to modification during project cycles.

Please note that the document is not validating the business models, large consumer tests or impact validation nor benefit evaluation as these issues are more of business concept nature and/or give insight in the usability of the market application, analysed in T10.2 on business modelling. Moreover the HYDRA project will not validate in-depth aspects related to the full functioning of the three domain applications, as these are considered as useful examples to understand middleware potentialities: an overall analysis is considered out of the scope of the project.

8. References

International Standards Organisation website
http://www.iso.org/

Thestrup, J. and Sørensen, T. (2007). *Scenarios for usage of HYDRA in 3 different domains*. Technical Report D2.1, HYDRA Consortium. EU Project IST 2005-034891

Foglia, T. and Costa, N. (2007). *Validation Framework*. Technical Report D2.6, HYDRA Consortium. EU Project IST 2005-034891

Kupries, M. and Hansen, K.M. (2007). *Updated Systems Requirements Report*. Technical Report D3.2, HYDRA Consortium. EU Project IST 2005-034891

Hansen, K.M. (2007). *Quality Attribute Scenarios*. Technical Report D6.1, HYDRA Consortium. EU Project IST 2005-034891

Thestrup, J., De Bona, M., Graefe, G., Guarise, A., Powell, D., Roehr, F. (2007). *Business modelling concepts*. Technical Report D10.5, HYDRA Consortium. EU Project IST 2005-034891

De Bona, M., Thestrup, J., Melchior, E.M. (2007). *User validation plan for eu-DOMAIN*. Technical Report D7.3, eu-DOMAIN Consortium. EU Project FP6-004420