Deliverable reference:	Date:	Responsible partner:
D1.2 V2.0	15 February 2015	Software AG

Project co-funded by the European Commission within
the Seventh Framework Programme
(FP7-ICT-2013-10))
ICT-2013.1.2: Software Engineering,
Services and Cloud Computing
Grant Agreement No.: 611337

www.heads-project.eu

Title:

D1.2: Validation Methodology and Consolidated Requirements

Editor(s):	Approved by:
Walter Waterfeld (Software AG) Ilias Spais (ATC)	Project Coordinator: Trine M. Seeberg Technical Manager: Franck Fleurey
	Classification: Public

Abstract / Executive summary:

Version 2.0 is enhancing the list of KPIs defined in the DoW and initial WP1 deliverables, by capturing the KPIs, both in terms of qualitative and quantitative metrics, which will steer HEADS technology to simplify the application development process in the IoT space. The updated list will be used to assess the complexity and performance of the modelling, distribution, and management components and facilitate the evaluation of the use cases in a more objective way that clearly shows the real impact of HEADS IDE. The consortium addressed this issue by a) taking a wider look at the literature of Software Engineering, Distributed Systems, and Ubiquitous Computing (for applications), b) revisiting and analysing project's objectives and c) reformulating the initial consolidated list of requirements.

Furthermore, this report advances on the initial definition of the HEADS evaluation framework both in terms of quantitative and qualitative metrics. The reformulated approach reflects the KPIs and concrete ways of validation per se are defined. Periodic assessments through appropriate experimental scenarios will be performed in order to raise the interest of experts for the HEADS IDE. Furthermore, external to the consortium experts will be engaged in the validation process, by participating in interviews, industrial focus groups, Hackathons in a major developers' event, training sessions, etc. Appropriate exploitation and dissemination activities will be defined (dissemination KPIs were defined as well).

Finally, HEADS consortium reformulated the use case scenarios to propose a more upgraded, enhanced and complete realizations of HEADS envisioned achievements. By addressing almost all HEADS objectives, use case scenarios will validate the respective KPIs and highlight why and how HEADS would achieve its objectives through them. It must be mentioned that this report provides an overview of those, while a complete description and realization will be included in D1.4 (M25).

Document URL:	ISBN:

Table of Contents

Version History

	Description	Date	Who
0.1	First draft with chapters and copy of requirements from D1.1 for tracking changes	2014-03-26	M2MZone
0.2	First input validation	2014-5-20	Walter
0.5	New outline and first input 2.4 detailed requirements validation	2014-6-20	Walter
0.6	Section 2.2	2014-6-29	Knut Eilif
0.7	Section 2.1 and 2.2	2014-7-2	Ilias
0.8	Section 1 and 2.3	2014-7-5	Walter
0.9.0	Rearranged 2.1 and 2.2	2014-7-7	Walter
0.9.1	Minor changes in chapter 2.1	2014-7-7	Knut Eilif
0.9.2	Finished for review except chapter 3	2014-7-8	Walter
0.9.3	Updated chapter 3 based on Excel document	2014-07-08	Vladimir
0.9.4	Incorporated review from Noel Plouzeau	2014-07-16	Walter
0.9.5	Incorporated review from Brice Morin	2014-07-21	Walter
0.9.6	Version for final check	2014-07-23	Walter
1.0	Final version approved by Technical Manager and Project Coordinator	2014-07-30	Walter
1.1	Content received from all partners	2015-01-30	All partners
1.2	Distribution for internal review	2015-02-04	Ilias
1.3	Enhancements and finalization of the KPIs	2015-02-06	All partners
1.4	Incorporating contributions	2015-02-09	All partners
1.5	Incorporating feedback received from internal review process	2015-02-11	Ilias
2.0	Final version approved by Technical Manager and Project Coordinator	2015-02-15	Ilias

Copyright

1 Executive Summary

Version 2.0 is enhancing the list of KPIs defined in the DoW and initial WP1 deliverables, by capturing the KPIs, both in terms of qualitative and quantitative metrics, which will steer HEADS technology to simplify the application development process in the IoT space. The updated list will be used to assess the complexity and performance of the modelling, distribution, and management components and facilitate the evaluation of the use cases in a more objective way that clearly shows the real impact of HEADS IDE. The consortium addressed this issue by a) taking a wider look at the literature of Software Engineering, Distributed Systems, and Ubiquitous Computing (for applications), b) revisiting and analysing project's objectives and c) reformulating the initial consolidated list of requirements.

Furthermore, this report advances on the initial definition of the HEADS evaluation framework both in terms of quantitative and qualitative metrics. The reformulated approach reflects the KPIs and concrete ways of validation per se are defined. Periodic assessments through appropriate experimental scenarios will be performed in order to raise the interest of experts for the HEADS IDE. Furthermore, external to the consortium experts will be engaged in the validation process, by participating in interviews, industrial focus groups, Hackathons in a major developers' event, training sessions, etc. Appropriate exploitation and dissemination activities will be defined (dissemination KPIs were defined as well).

Finally, HEADS consortium reformulated the use case scenarios to propose a more upgraded, enhanced and complete realizations of HEADS envisioned achievements. By addressing almost all HEADS objectives, use case scenarios will validate the respective KPIs and highlight why and how HEADS would achieve its objectives through them. It must be mentioned that this report provides an overview of those, while a complete description and realization will be included in D1.4 (M25).

2 Introduction

2.1 Context and objectives

This report was suggested by the reviewers during the 1st review meeting held at Brussels, 13th of November 2014. It is a re-submission of D1.2 Ver1.0, which was initially submitted at M10. HEADS accepted reviewer's suggestion and even more was grateful that bring into attention aspects that were partially missed and/or needed to be clarify with respect to the initial submission.

This deliverable presents the plan for the evaluation and validation of the HEADS results based solution with respect to i) the main objectives of the project, ii) the Key Performance Indicators (KPIs) iii) requirements that were identified in all WPs and collected in a coordinated way by T1.2, and iv) the specific objectives of the technical work packages. The evaluation strategy will be driven by the KPIs, and its main scope will be to ensure that HEADS project has met objectives and requirements while at the same time has improved the development, deployment, configuration, and troubleshooting efforts for realizing future heterogeneous HD services. The evaluation strategy is derived based on the state of the art scientific methods and experiences previously evaluated in other EU projects. Based on the selected evaluation strategy and the project road-map for delivering the tools, an evaluation plan containing goals, KPI, methods and schedule for performing evaluations has been developed.

This deliverable provides the guidelines and an activities plan that will be carried out to evaluate the HEADS achievements focusing on evaluation objectives defined in WP1.

2.2 Structure of the document

This report consist of the following sections:

Section 3, describes the reformulated evaluation methodology. KPIs definition raised the need to bring on board external to the consortium experts (mainly developers), that will validate via several ways, namely training sessions/tutorials/ Hackathons, HEADS achievements that cannot be addressed by use case providers. This task is included in HEADS revised evaluation framework.

Section 4, outlines the updated list of KPIs together with ways of measuring the respective quantitative and qualitative metrics.

Section 5 provides an overview of the updated use case scenarios, while Section 6 the updated list of HEADS consolidated requirements.

3 Validation Methodology and Framework

The HEADS project will deliver different types of results consisting of technology like the HEADS IDE (Integrated Development Environment) and demonstrators for realistic use cases. **Therefore, we use different types of validation for evaluating the HEADS results.**

- **Overall assessment of the complete project:** the overall assessment is a high-level simple evaluation of the current project status.
- **Main requirements validation:** this more detailed validation will be done from the perspective of the use cases and will evaluate each of the specified main requirements. For the requirements we define Key Performance Indicators (KPIs) to measure to what degree the requirement is achieved.
- **Detailed requirements validation:** in line with the character of the detailed requirements this evaluation is more from the perspective of specific technical components, which are needed within the use case realisation.

3.1 Rationale and Evaluation strategy

In our context we are interested in evaluating a software based solution (the software based methods, tools and techniques of the HEADS approach).

Our evaluation strategy is summarized by the following questions that are the guidance for our evaluation plan:

1. Who is the evaluation for?
2. Why do they want the evaluation?
3. What are we evaluating?
4. When do we evaluate our artefacts?
5. How do we evaluate?

There are basically three main target audiences for our evaluation:

1. **Ourselves** (the HEADS project). A main purpose of our evaluation is to provide feedback to ourselves. In particular, the first evaluation round (End of the second period) is performed in order to ensure proper planning and achieve proper improvements of the HEADS approach when conducting our second development increment.
2. **Exploitation users.** Another important target audience of the evaluation is users that want to test and exploit HEADS assets (these may include both HEADS partners and third party/project external users). The evaluation will provide them a better understanding of the status (e.g., Technology Readiness Level, available features, strengths and weaknesses, etc.) of the provided HEADS assets. Furthermore, the evaluation report is crucial background information in order to prepare and plan for exploitation of the HEADS assets both during and after the project (again this goes both for third party users and for our own exploitation planning with respect to the HEADS assets).
3. **The Commission and reviewers.** A third target audience is the project owner, the Commission. They want to know to what extent the goals and ambitions of the project are achieved and whether it has been a good return of investment. The evaluation of the provided software based tools and techniques is a valuable input in this respect. Moreover, the reviewers will evaluate the project on the commission behalf at the annual reviews, where a main task is to rate the project status. The evaluation of the software based tools and techniques is again valuable input for the general evaluation and rating of the project.

Considering aspects like functionality, usability, performance efficiency and portability, the evaluation framework will be utilized to evaluate HEADS achievements. The main baseline to determine what characteristics and features to be evaluated are:

1. the HEADS objectives as stated in the DoW,
2. the HEADS defined KPIs, which are defined in section 4 of this document,
3. the set of requirements that are elicited in the HEADS project both related to the case studies and related to the technical work packages,
4. the specific objectives of the technical work packages.

The schedule of the evaluation includes a plan when to evaluate which deliverable. Each deliverable should provide an evaluation plan from which proper and efficient evaluation can be accomplished when the deliverable reaches a milestone. E.g. the HEADS IDE needs to be available in a stable state. According to the HEADS plan, the first increment of the HEADS IDE was provided in M11, thus, the evaluation can start at the point when this plan is ready at M16. The findings of the first round of evaluation impact the planning and focus on our second development increment of the HEADS IDE, while the final evaluation will provide an important input for the exploitation plans. Similar to this example for the HEADS IDE other WPs will provide a schedule for the WP specific evaluation.

The use case partners in the project will play the role of users evaluating the provided tools and techniques provided by the technology providers in the project. The main foundation of this evaluation is the actual development of the HEADS case studies applying the HEADS IDE as well as the execution of the case study implementations using the HEADS run time management platform.

For the HEADS project we will define KPIs to validate the main requirements. Within HEADS we define for each Key Performance Indicator (KPI) several characteristics, namely the scope with respect to the project objectives, its definition focusing on HEADS related achievements, ways of measuring, indicative metrics and threshold.

The KPIs define measurements for the delivered assets in the areas of

- Performance
- Number of platforms or classes of sensors
- Effort to develop
- Overhead in size

In each of these areas HD-services developed with HEADS methodology and tools are compared to services developed with other methodology and tools.

For the definition of the KPIs see Section 4.

3.2 External Advisory Board

The HEADS project will formulate an external advisory board (EAB). This board should guide the evaluation and comparison of HEADS methodology and tools with other methodology and tools. It will also advise on strategy directions for the project and help the consortium to understand and define in a more efficient way the added value provided by HEADS IDE. In this way it is envisioned to increase its impact in all the relevant domains.

The EAB will continuously monitor the evolvement of the project by participating in several project’s events, namely focus groups, workshops and virtual demonstrations (calls). The EAB will meet on regularly and the meetings will be prepared and chaired by the project coordinator, who will report the content of the discussions to the PMB. In order to make this activity more efficient, HEADS consortium decided to run an EAB meeting at each one of the upcoming consortium's plenary meetings.

The EAB members will receive access to project documents, testbed installations and prototype systems and will focus on validating the fulfilment of the defined KPIs. EAB members can even update the list if the proposed KPIs are complying with projects objectives and envisioned achievements.

The main criteria for selecting these members will be proven excellence in research as well as commercial potential. Each partner will suggest 2-3 candidates and the consortium as a whole will decide which of those will formulate the EAB.

3.3 Other External Evaluation

The HEADS project will engage external experts in the evaluation process. This will be done by participating in interviews and industrial focus groups. Moreover training sessions based on public tutorials and Hackathons will be organized.

4 HEADS Objectives and associated KPIs

The main objectives of the HEADS project are defined and described in the section 1.1.4 of the HEADS DoW. The DoW also includes a description of performance and research indicators (section 1.2.9) which details the expected progress of the project milestone by milestone. This section aims at complementing the DoW by providing a synthetic set of measurable success criteria to be used as key performance indicators associated to the HEADS project objectives.

4.1 Definition of the HEADS KPIs

Objective	HEADS achievements	KPI - Measurable success criteria	Ways to evaluate and plan
SO1: Provide service developers with new abstractions to develop HD-services	Define a set of metamodels based on the Eclipse Modelling Framework (EMF), formalizing the different modelling aspects needed to specify HD-services Develop a set of open-source tools (editors, checkers, methods) to guide developers when specifying HD-services Provide a set of comprehensive samples and tutorials illustrating the different concepts of the languages Integrate the different languages and tools and an operational methodology for developing HD-Services into the HEADS IDE.	KPI-O1-1: The HD-Service developer can use at least 5 different platforms covering the Future Internet Continuum (FIC) from end to end in a single application. KPI-O1-2: HEADS modelling languages are expressive enough to express application logic in a platform independent way KPI-O1-3: The training period for a software developer ("fluent" in either Java, JavaScript or C/C++) to get started with the HEADS modelling languages should be reasonable (around one day)	Running an experimental evaluation through experimental test cases (set of models and open-open source tools) - technical WPs Internal evaluation activities, namely focus groups interviews Evaluate though a Hackaton if the learning curve for the HEADS IDE is less that one day. External developers will run some exercises the technical team will create and based on their efforts they will evaluate the system Check that the different viewpoints for platform experts, service developers and service operators are all used in the implementation of the demos, tutorials and use cases.
SO2: Provide platform experts with efficient means to capture their knowledge	Propose and implement an open-source code generation framework with a plugin mechanism to integrate other platforms, including both popular, proprietary and legacy platforms.	KPI-O2-1: Code generation framework shall cover heterogeneous platforms. Going from microcontrollers to servers. At least 3 different languages and 2 different target platforms per languages shall be supported by the code generation framework	Use-cases involve heterogeneous platforms ranging from small microcontroller-based devices to servers. Platforms from all the target groups Tiny, Small and Large will thus be covered.

Objective	HEADS achievements	KPI - Measurable success criteria	Ways to evaluate and plan
	Develop a set of open-source transformations and generators targeting key cloud platforms, open-source middleware, mobile platforms and open-source resource constrained platforms.	KPI-O2-2: The effort required to support a new platform is correlated to the distance between this platform and already supported platforms KPI-O2-3: The effort and learning curve of writing plugins for the HEADS IDE is reasonable and on par with the effort required to write plugins for other IDEs • Number of reusable components and platforms that can be used from the by a service operator • Number of tutorials and examples available • Methodology for platform experts KPI-O2-4: Message communication framework shall handle communication in very heterogeneous networks covering the range from resource constrained devices up to cloud platforms. At least 2 different languages and 2 different target platforms and at least 2 different standard protocols per languages shall be supported KPI-O2-5: For resource constrained devices the generation framework and the communication framework shall target at least one platform in each of the platform groups Tiny, Small and Large	The level of distribution and the message communication will be evaluated through at least one use case demonstrator Running an experimental evaluation through experimental test cases (level of distribution and message communication) - technical WPs Evaluation by developers outside the consortium with Tutorials and Hackathons focussing an adding support for new platforms in the HEADS IDE.

Objective	HEADS achievements	KPI - Measurable success criteria	Ways to evaluate and plan
SO3: Make the HD-service lifecycle safe, predictable and consistent	Develop a set of tools for validating the plug-ins created by platform experts. Develop a set of tools to generate test benches and their infrastructure and provide a representative set of sample test benches. Extend the HEADS languages with a metamodel to specify test constraints and objectives, including QoS attributes.	KPI-O3-1: Automated and tool supported detection of inconsistencies between code generators targeting different platforms. HEADS IDE includes a complete test suite and a test harness for the platform expert to validate new code generator plugins. It allow: • Comparing generated behaviour with a reference generator. • Automatically detecting semantic inconsistencies. KPI-O3-2: Automated and tools supported evaluation of non-functional properties of HD-Services. HEADS provides a framework to deploy systems in a controlled environment to measure QoS properties. The framework shall include memory, bandwidth and CPU.	The plug-in validation test suite and test harness will be used and evaluated by the technical WPs for validating all plug-ins produced in the project. It will also be used for plug-ins developed for the use-cases and by external platform experts. The QoS support and evaluation tools will be validated experimentally on demos and examples by the technical WPs and will be put to the test by comparing alternative designs for the use-cases and verifying the QoS predictions on the use-case implementations.
SO4: Manage the flow of big data across the future computing continuum	Propose abstractions to capture distributed complex-event processing (CEP) tasks as part of the HEADS design language. Develop analysis and transformations to automatically distribute the execution of the complex-event processing tasks on the continuum.	KPI-O4-1: Demonstrate CEP distribution with at least 3 levels of processing nodes covering the Future Internet Continuum. KPI-O4-2: Improve performances on HD system compared to classical centralized implementation: at least 5 times higher events per second (event rate).	Evaluation performed by at least one use case demonstrator (CEP distribution and improvements related to centralized implementation). Running experimental evaluations to compare with existing centralized implementation and measure the performances differences.

Objective	HEADS achievements	KPI - Measurable success criteria	Ways to evaluate and plan
		KPI-O4-3: Distributed CEP queries with a less than 50% overhead in development effort compared to a centralized implementation.	
SO5: Provide seamless dynamic deployment and evolution of HD-services	Develop an open-source models@runtime engine able to maintain a causal connection between HD-service descriptions and artefacts running on the different platforms composing the future computing continuum.	KPI-O5-1: The HEADS IDE includes deployment plugins for platforms covering the whole Future Internet Continuum spectrum. The provided plugins should cover at least 5 different middleware and deploying components implemented in at least 3 different programming languages. KPI-O5-2: For resource constrained devices of the Future Internet Continuum, HEADS IDE shall include modules to dynamically deploy firmware binary images to at least one platform in each of the platform groups Tiny, Small and Large.	At least two use case demonstrators will be utilized to evaluate HEADS IDE. Experimental evaluation, demos examples, tutorials
SO6: Demonstrate the interest and generality and disseminate HEADS results	Provide an integrated IDE and methodologies which fits the needs of the industry partners Provide a set of techniques and tools which can be integrated in the technology providers products	KPI-O6-1: End-users perception and impact assessment: • Productivity increase by at least 30% compared to today • Performance improvements compared to centralized implementation, 10 times shorter response time, 10 times lower events per second	Use a Hackaton to evaluate the HEADS benefits. Satisfaction covering all aspects verified on at least 1 complete demonstrator

Objective	HEADS achievements	KPI - Measurable success criteria	Ways to evaluate and plan
	Publish a set of scientific paper covering the original contributions of the HEADS approach and languages. Disseminate the HEADS results to practitioner and relevant open-source communities.	(event rate, bandwidth) • Using existing legacy code without redevelopment • Easy to learn and get started. Less than 1 day training for software development • Increase of quality issues by 50% compared to today: robustness, fault-tolerance, availability, resource management and maintenance • HEADS IDE usability – reasonable overload compared to “other IDE’s” (learning curve) KPI-O6-2: Release of HEADS IDE under an open source non-viral license with a framework maintenance for at least 2 years beyond end of project. KPI-O6-3: At least 1500 downloads of the HEADS IDE and at least 100 external developer with hand-on with the HEADS IDE during at least 10 different training events (tutorial sessions, Hackaton, focus groups, surveys, demos, seminars). KPI-O6-3: Scientific dissemination with participation to software engineering, model-driven engineering and Internet of things research conferences and workshops. At least 4 journal paper and 12 conference papers (at least 50% of conference and journal papers have impact factor or ERA classification).	Implementation of 2 use case demonstrators, together validating at least 80% of tools Tutorial and experiments with external developers.

5 Use cases scenarios reformulation

In this section we provide a very short description (overview) of use case scenarios reformulation. A complete and detailed description will be provided in D1.4 (M25). The goal of this overview is to show that the use case cover the project objective in an acceptable way (a respective KPI has been defined). The realization of the use case scenarios, together with the respective test cases that address the defined KPIs will be reported in D1.4.

As an overview TellU use case focuses on heterogeneity (and CEP in an embedded context) while the ATC use case focuses on distribution (and CEP in a big data context).

5.1 Evaluation related to the TellU use-case

The industrial use-case from TellU to be implemented in HEADS is a smart-home system for helping elderly people to stay home as long as possible. The system will in addition to control equipment that are normally present in smart homes to make life comfortable (automatic light control, door locks, and heater control, etc.), have an additional focus on the situation for elderly people. One of the main fears for elderly people is the fear for falling and not being helped. This fear causes elderly to isolate and become less active, which in turn make them more exposed to illness. We will include a fall detection system in the smart-home based on a distributed sensor network for measurements of air pressure from both stationary and wearable sensors. In addition we will also be able to measure the activity level and indoor location of the elderly residents. This information can be used to monitor if the elderly residents have an adequate activity level during the day. This information can also be supervised by physicians, care-givers, family etc. Furthermore, several services can be automated based on measured indoor locations and controllable equipment, e.g. light control, turn of stove when leaving house, etc. Eventually, we will include sensors for measuring physiological parameters like heart-rate and skin temperature. In this way care-givers can monitor if the elderly resident is not feeling well. Fall detection and heart-rate monitoring should also be applicable outdoors to give the elderly the needed confidence that they will get assistance if they need help outside of the home. Elderly people should not isolate indoor and become inactive because of fear and anxiousness for leaving the home.

The main purpose of the use-case has, in addition to specify a system of industrial relevance, been to assess the HEADS IDE capabilities of handling heterogeneous computing nodes and distribution of services. Thus, we have focused on specifying a broad range of computing nodes spanning from small and very constrained platforms (Arduino, Raspberry PI, Intel Edison) to cloud resources with virtually unlimited capacity. In terms of distribution we have also had focus on deploying different communication technologies spanning from wired sensor communication (e.g. I2C), via wireless (e.g. ZWave, WiFi, BLE) to communication with Internet protocols (e.g. TCP/IP, HTTP, MQTT). Eventually, we have specified a set of sensors to be included in the use case. The list contains sensors quite commonly used in modern house automation which sensor outputs can be used to create interesting and useful services. Although, the quantities of deployed nodes, sensors and communication links will be relatively constrained in the use-case implementation (magnitude of tens and not thousands), we have managed to specify a mix of equipment and technologies that will be adequately challenging to evaluate the usefulness of the HEADS IDE. Through this use-case, the HEADS IDE will demonstrate how the service developer and operator can work at an abstraction level that is convenient in terms of handling differences in the underlying platforms implementations and system distribution. Thus, the development and operation process will be facilitated and speed up.

The use-case described above takes the use-case described in D1.1 as a starting point. However, it is extended with a much higher degree of heterogeneity and distribution to meet the requirements from the needs of the evaluation HEADS IDE. The complete use-case will be thoroughly described in D1.4, Consolidated use case implementation and evaluation.

5.1.1 *Use-case infrastructure*

Based on the functionality described above we describe an infrastructure of the smart-home system. The infrastructure will cover both the indoor and mobile (out of home) scenario.

- Indoor infrastructure– the house will be equipped with smart-home equipment (light control, stove control, door locks control, heater control, environmental sensors (air-pressure, temperature, humidity, light)). Further, the user will be wearing sensors that monitor movements/indoor positions and some physiological parameters of the user. Fall to ground and high/low heart-rates will be monitored. The smart-home system will consist of distributed field nodes communicating wirelessly with a home gateway that is in connection with cloud resources.
- Mobile infrastructure – In the mobile scenario, the user will be carrying a SmartPhone and sensors attached to the body. The system will monitor movements and some physiological parameters of the user. Fall to ground and high/low heart-rates will be monitored.

In order to describe the heterogeneity and distribution aspects of the use-case infrastructure, we have listed in three tables the computing nodes, sensors and communication technologies going to be applied in the use case realization.

Node	Description	Platform
Field node	Node with connected smart-home sensors. Has responsibility to wirelessly transfer collected sensor data to gateway. Stationary placement in the smart-home. Limited computing /and or low power consuming resources.	Arduino Uno Arduino Yun Intel Edison
Home-GW	Computing node that receives sensor information from several connected field nodes. Performs local decision logic and coordinates information from several field nodes	Raspberry PI
Cloud resource	Powerful computing nodes in the cloud that have responsibility to store big amounts of sensor data, coordinate decisions between distributed smart-home nodes	Linux
SmartPhone	Computing and communication node that the user wears with him/her when being out of home	Android
Body sensor	Sensor attached to the body. Transferring body sensor data to either SmartPhone (mobile scenario) or Field node (indoor scenario)	SINTEF CU (ESUMS – iSenseU)

Table 1: Computing nodes

Sensor	Description
Sound Sensor	Sensor detecting sound and measuring amplitude
Light Sensor	Sensor detecting light intensity
Area Intrusion Monitor	A PIR sensor monitoring movement in rooms
Gas/Fire/Smoke Sensor	Sensor detecting smoke and gas
Temperature/Humidity Sensor	Sensor detecting temperature and humidity
Air Pressure Sensor	Sensor measuring differential air pressure (used for fall detection)
Water Leak Sensor	Sensor detecting water on floor
Reed-switch Sensor	Sensor detecting position of windows and doors
Distance Sensor	Sensor detecting distance to object
Light Switch	Controllable light switch
Door Lock	Controllable door lock

Table 2: Sensors

Communication Technology	Description
Bluetooth Low Energy (BLE)	Communication technology used between SINTEF CU and Field Node
WiFi	Communication technology used between Home-GW and Field Node
I2C	Communication technology used between digital sensors and Field Node
Ethernet	Communication technology used between Home-GW and Internet
Mobile Communication	Communication technology used between SmartPhone and Cloud resources
Wire	“Communication technology“ used between analogue sensors and Field Node
ZWave	Communication technology used between Field Node and Door lock/light switch

Table 3: Communication technologies

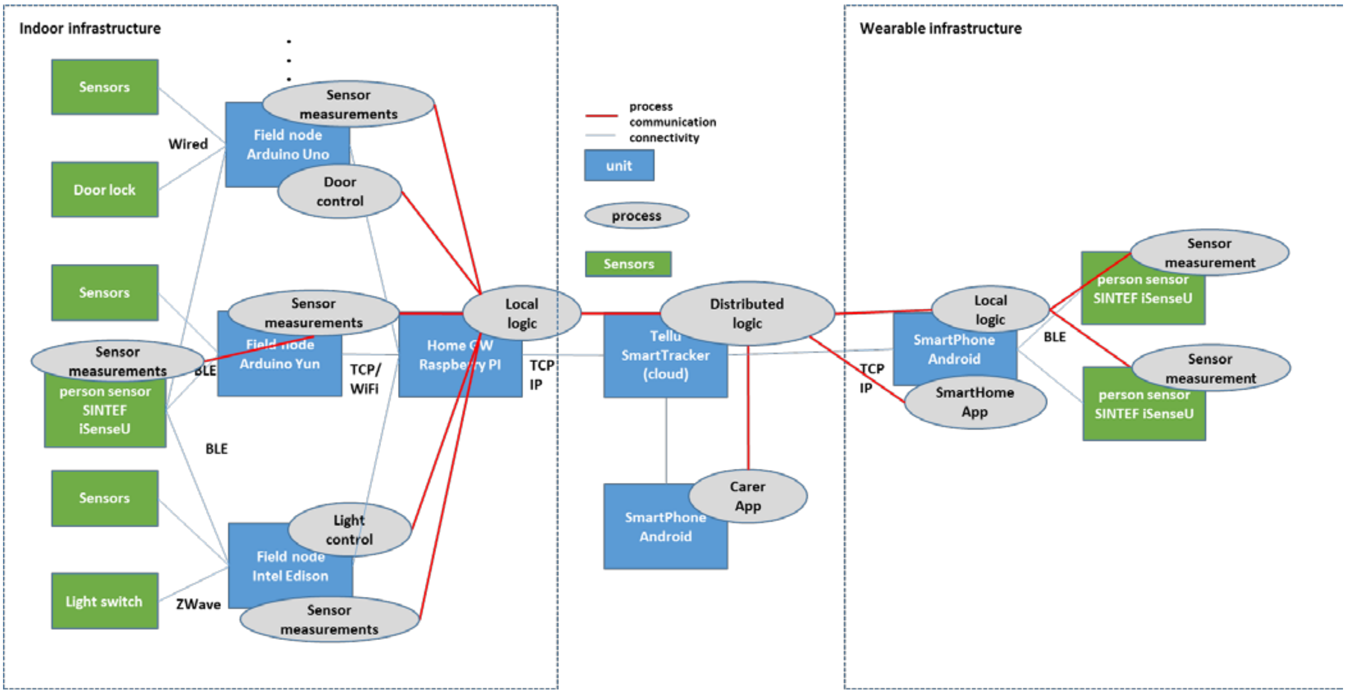

Figure 1: Infrastructure and process deployment

5.2 Evaluation related to the ATC use-case

NewsAsset suite constitutes an innovative management solution for handling large volumes of information offering a complete and secure electronic environment for storage, management and delivery of sensitive information in the news production environment. The developed solution is able to aggregate the requirements and expressed needs of the most demanding enterprises in the areas of digital news content production, management and delivery. Times change fast, and news agencies are already feeling the impact of the web and the on-line world in their operations. At the same time there is a rapid increase of demand for cross-media news publishing tools and ways to make management of any type of files an easy and risk free process. News agencies are looking for additional ways to create and distribute their content, whether it is text, images, graphics, video or other digital data. At the same time, the future computing continuum, composed of a wide set of heterogeneous platforms, promises to be an environment that will definitely affect the media industry in terms of creating, managing and exploiting trustworthy media items.

Nowadays, interesting media info is out there in the digital world and can be possessed and exploited by utilizing a variation of nodes and software engineering practices like network nodes, gateways, smart-phones, social networks, cloud computing and service-oriented methods, etc. In the context of the “Internet of Things” (IoT), the number of sensors in the environment will grow ever more quickly. More and more sensors, connected to the Internet, will feed real-time data about a variety of things: average traffic speed on roads; slipperiness index for roads receiving rain or snowfall; air pollution levels by location; etc. from government entities. As more sensors enter our world, journalists will be able to access data from more and more of them, aiding not only in disaster coverage (disaster news coverage will be much better in a few years), but being used in all manner of news stories. Furthermore, when a disaster is happening somewhere in the world there is a lot that is going on, so with power and Internet access available, people can watch the news eco-system to try to learn what damage is where, and what conditions exist the specific time. Many eyewitnesses will snap a disaster photo and post it to Instagram, Twitter, Facebook, etc., explaining what's going on or the location. Sometimes, a smartphone-toting eyewitness may have location-tracking turned on and physical coordinates of the image are attached when the photo is published to a social-media site. As smartphone photo apps are set to include location by default, more and more people discover the value of location-tagging the photos they shoot. As that trend plays out, photos and video from social-media sites are becoming increasingly useful for disaster news coverage. In addition to, if all those images spotted on Twitter and Instagram include location metadata, they can be incredibly useful and a detailed interactive map of the disaster could be created.

News organizations can benefit from this future trend and utilize the IoT and social-media content to provide much better and more detailed coverage of a major disaster. Twitter itself, one of the primary conduits of eyewitness images and reports during disasters, has come to understand that their service provides a valuable news service to the public. The company easily could evolve to become (also) a news operation, utilizing its own tools and analysis to provide the fastest and most-detailed information of a breaking disaster. The challenge for NewsAsset is to catch up with this evolution and provide services that can handle the developing new situation in the media industry.

5.2.1 Use-case infrastructure

HEADS innovations are envisioned to offer the opportunity to NewsAsset to leverage on environments spanning from cloud computers and social networks to handheld smartphones and sensor platforms situated in the field. These platforms are very different when it comes to capabilities like power consumption, computing resources, connectivity and communication channels. HEADS tools and methodologies will have to address the device and platform heterogeneity and support systems especially adapted to environments with different capabilities. More specifically in the context of NewsAsset journalism scenario, HEADS will have to address the following challenges:

- Support the collection of high volumes of data from various heterogeneous sources.
- Handle streams of data fed from intelligent devices and platforms. The architecture for this type of real time stream processing must deal with ingest, processing, storage and analysis of a large number of events per hour.
- Store the data in a highly scalable distributed storage system.
- Analyze real time streaming data (for instance filtering, pre-processing and validation) and provide insights. Merge the outcome with a batch processing analysis using historical data.
- Scale out to serve a wide range of workloads and use cases, while providing low-latency access to the stored information. The system should be linearly scalable, and it should scale out rather than up, meaning that throwing more machines at the problem will do the job.
- Support the need for a robust system that is fault-tolerant, simple and comprehensive.
- Manage end points from the platform. This should include the ability to register / discover a device.
- Support extensibility so that nodes can be added easily.
- Provide users with the ability to customize and build applications for their specific scenario.

Based on the previous analysis, HEADS IDE should focus on several heterogeneity and distribution aspects for the realization of the NewsAsset scenario. In terms of heterogeneity, different type of platforms/data sources will be utilized to produce and collect the data and demonstrate the writing of scenario specific code, while in order to process very large amounts of data, highly robust and scalable computations must run in parallel across a cluster of machines. Thus, the code of the scenario has to be implemented and executed in a distributed way.

An indicative example of the NewsAsset scenario is to distribute the business logic of receiver and sender components, currently implemented with a centralized architecture. The receiver is the component responsible for receiving news assets from various sources and inserting them into the NewsAsset architecture. On the other hand, the server forwards produced or collected news assets to subscribers or clients of a service. Both components are implemented as Windows services and run in one or more servers. They implement business logic related to access rights, assets tagging and categorization and a prioritization mechanism. The idea is that a part or all of this logic is migrated to smart phone devices that have registered to the system. The smart phone devices will automatically annotate the assets with additional information (geo-location information, author name, time) and auto tag the items. They will be able to automatically publish an item, bypassing the normal workflow (which will be executed in parallel for archiving the asset). Another channel for producing data is data streams from social networks (Twitter). These will be monitored by a distributed set of servers, which will follow the same principle of automatically annotating and distributing an item.

From a bird's eye view, the NewsAsset HEADS scenario will be realized by the implementation and deployment of four layers, namely the nodes, the network, the operation/management analysis and the application/services layer. The following figures presents all layers and the respective components, while the following tables describe the computing nodes, sensors and communication technologies that are going to be applied.

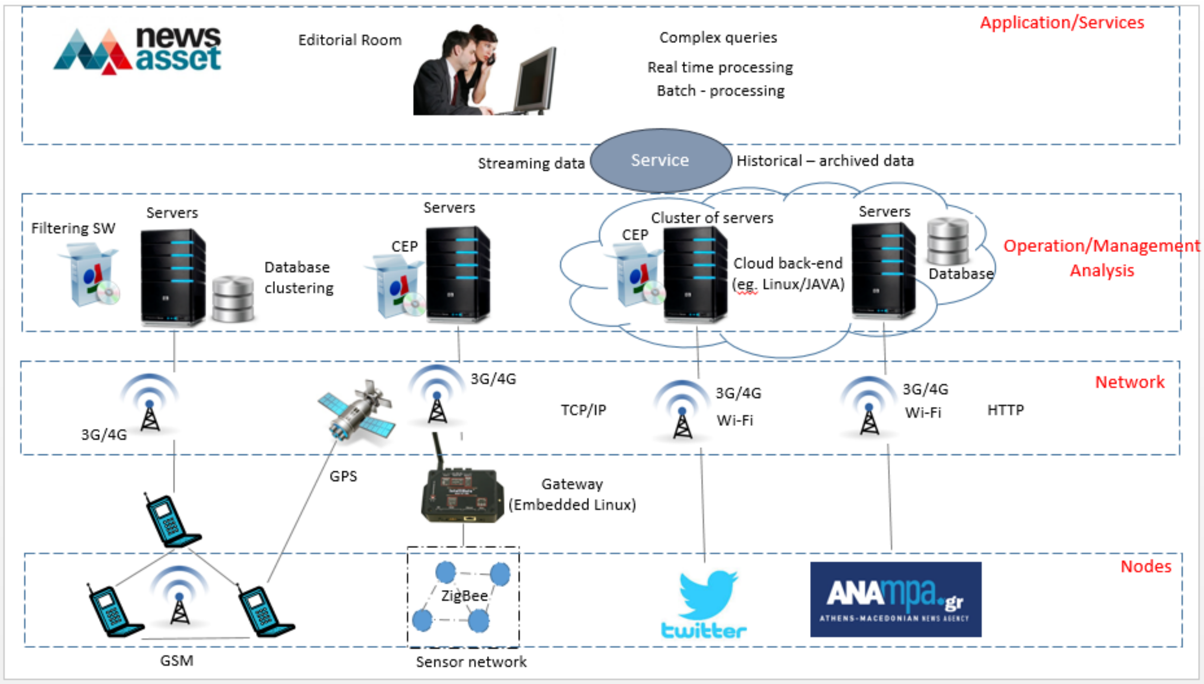

Figure 2: The NewsAsset scenario: high level architecture (deployment layers)

Node	Description	Platform
SmartPhone	Computing and communication node that the user carries to take a picture, annotate it with metadata and automatically distribute it.	Android and iOS
Server machine	Hosts the operation management system of the mobile functionality. The system filters pre-processes and validates data captured by the SmartPhone. Hosted on ATC premises	Requires Microsoft Visual Studio 2013, MSSQL Database Server 2008+, Microsoft Windows Server 2012 and .NET Framework 4.5
NewsAsset infrastructure	Aggregating and storing media data – master database. Hosted on ATC premises	Windows machine: Windows Server 2003 with NewsAsset server. Linux machine: Oracle Express for hosting the NewsAsset SQL Database and the File Repository (file server for storing images, audio and videos attached to news items)
Database clustering	Distributed database – database clustering can be used to store an immutable, constantly growing master dataset, and support the computation of arbitrary functions on that dataset. Hosted on ATC premises or in a Cloud platform	Indicative example is Oracle Clusterware, a portable cluster software solution, allows the clustering of independent servers so that they cooperate as a single system
Cluster of servers	Distributed compute engine – multiple instances for processing raw streaming data. Hosted on ATC premises or in a Cloud platform	Indicative example: Storm cluster
Cluster of servers	Distributed compute engine – capability to run parallelize large scale batch computations on very large amounts of data (historical/archived data). Create batch views – predefined queries/filters. Need to adopt low latency computational results. Hosted on ATC premises or in a Cloud platform	Indicative example: Hadoop framework
Cloud resources	Powerful computing nodes in the cloud that have responsibility to store big amounts of media data, coordinate decisions between distributed nodes and operate parallel large scale computations	Linux/Java OpenStack/Amazon

Table 4: Computing nodes

Communication Technology	Description
WiFi	Community Technology used between smart phone devices and access point (Internet).
Mobile Communication (3G/4G/GSM)	Community Technology used between smart phone devices and internet providers.
Ethernet	Communication technology used by social networks data streams processing servers.

Table 5: Communication technologies

6 HEADS Main Requirements Specification

This section describes the consolidated requirements for HEADS project. A first version of them has been published in deliverable D1.1,"Use case scenario definitions and initial requirements". Based on the result of the technical work packages and a further refinement of the use cases a new consolidated version has been developed.

Description of the priorities:

Priority	Description
MUST	This word means that the definition is an absolute requirement of the specification.
SHOULD	This word means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

Descriptions according to RFC 2119 (see http://tools.ietf.org/html/rfc2119).

6.1 Main Requirements: TellU

Identifier	TellU – 01	**Type**	Functional
Short name	Personal Security System		
Objectives	Primary: Objective 6, Secondary: Objective 1		
Description	Using HEADS system to provide solution for TellU eHealth use-case. For detailed description of the scenario check chapter 3 of D1.1: "Use case scenario definitions and initial requirements"		
Rationale	Verification of main project objectives based on real scenario		
Related scenario	A		
Contributors	Primary: WP1		
Stakeholder	HD-Service developer		
Priority	MUST		
Satisfaction criteria	Detailed satisfaction criteria are presented in chapter 3 of D1.1: "Use case scenario definitions and initial requirements"		

6.2 Main Requirements: ATC

Identifier	ATC-01	Type	Functional
Short name	Media System		
Objectives	Primary: Objective 6, Secondary: Objective 1		
Description	Using HEADS system provides solution for Media System scenario. For detailed description of the scenario check chapter 4 of D1.1: "Use case scenario definitions and initial requirements"		
Rationale	Verification of main project objectives based on real scenario		
Related scenario	B		
Contributors	Primary: WP1		
Stakeholder	HD-Service developer		
Priority	MUST		
Satisfaction criteria	Detailed satisfaction criteria are presented in chapter 4 of D1.1: "Use case scenario definitions and initial requirements"		

6.3 Main Requirements: Common

Identifier	COMMON-01	Type	Functional
Short name	Integration with legacy and third-party code		
Objectives	Primary: Objective 1, Secondary: Objective 2		
Description	HEADS modelling languages and transformations should provide mechanisms to integrate with pre-existing legacy or third-party code, with no need to re-model it. In particular, the code produced from the HEADS modelling languages should provide clear APIs (e.g. plain Java interfaces or plain C headers) and rely on as less dependencies as possible, to facilitate reuse, integration and maintenance within or outside the HEADS IDE. The generated code should comply with state-of-the-art practices in software engineering.		
Rationale	The adoption of the HEADS technologies is facilitated by a step-wise approach where some parts of the system can be developed with HEADS, while some other parts can be developed without the HEADS technologies.		
Related scenario	A, B		
Contributors	Primary: WP2, Secondary: WP5		
Priority	MUST		

Identifier	COMMON-01	Type	Functional
Stakeholder	Platform expert, HD-Service developer		
Satisfaction criteria	- Use case partners can integrate artefacts developed using HEADS technologies with their existing code - HEADS will provide a standard library of artefacts, some of them wrapping existing libraries for communication and serialization (e.g. MQTT, HTTP/REST, WebSocket). The cost of wrapping those libraries should be proportional to the size/complexity of the public APIs of those libraries, not to the size/complexity of their implementations.		

Identifier	COMMON-02	Type	Functional
Short name	Platform Independent Component on Multiple Platforms		
Objectives	Primary: , Secondary:		
Description	Platform independent components modelled with the HEADS modelling language should be able to run on at least 3 different platforms. Platform-independent components should indeed be reusable across.		
Rationale	Reduce development time by promoting reuse of platform independent components across platforms		
Related scenario	A		
Contributors	Primary: WP2		
Priority	MUST		
Stakeholder	HD-Service developer		
Satisfaction criteria	- A standard library of Platform-Independent Models (PIMs) is available in the HEADS repository - Use case partners have several PIMs that they reuse across platforms or across use cases.		

Identifier	COMMON-03	Type	Functional
Short name	Heterogeneous Data Sets		
Objectives	Primary: Objective 4, Secondary: Objective 3		
Description	HEADS system must incorporate heterogeneous data sets and provide support to easily extend the system by adding new data sets. The description of data sets must include connectivity information, data structure and data types. It should be possible for HD-Services to easily access and filter various data sets and process the data.		

Identifier	COMMON-03	**Type**	Functional
Rationale	HEADS services should be able to access data sets relevant to the selected scenarios. Heterogeneous infrastructure is valid only if it exposes heterogeneous data sets.		
Related scenario	A, B, C		
Contributors	Primary: WP4, Secondary: WP5		
Priority	MUST		
Stakeholder	HD-Service developer		
Satisfaction criteria	Sensor and social networks offer for further analysis open data sets Heterogeneous data sets are described and utilized		

Identifier	COMMON-04	**Type**	Functional
Short name	Target Heterogeneous platforms		
Objectives	Primary: Objective 2, Secondary: Objective 1		
Description	HEADS system must provide practical means to target new platforms that have not been integrated as part of the "standard HEADS library". In particular, the effort should be limited when a new platform to be integrated is fairly similar to another platform that has already been integrated. Only the parts that vary should be re-implement, all the rest should be reused.		
Rationale	Sustainability of the HEADS library is ensured.		
Related scenario	A		
Contributors	Primary: WP2, Secondary: WP5		
Priority	MUST		
Stakeholder	Platform expert		
Satisfaction criteria	- A modular transformation framework is provided, allowing platform experts to reuse/extend existing parts of existing transformation - At least two "variations" of the core platforms supported by HEADS will be provided - The effort of targeting a variation of a given platform should be proportional to the number/size of differences among those platforms, not to the complexity of the new platform		

Identifier	COMMON-05	**Type**	Functional
Short name	Capabilities and Requirements		

Identifier	COMMON-05	Type	Functional
Objectives	Primary: Objective 5, Secondary: Objective 3		
Description	HEADS should provide means to document and formalize the capabilities of the different platforms (e.g. in terms of CPU/RAM resources, I/O throughput, etc.) as well as define qualitive requirements and guarantees (possibly in an automated way e.g. by statistical analysis) about the individual components composing the HD-Services.		
Rationale	This should provide guidance to drive the deployment of the HD-services.		
Related scenario	A		
Contributors	Primary: WP4, Secondary: WP2		
Priority	MUST		
Stakeholder	Service Operator, Platform Expert		
Satisfaction criteria	Different platforms are efficiently documented from the HEADS offered means		

Identifier	COMMON-06	Type	Functional
Short name	HEADS IDE		
Objectives	Primary Objective 1, Secondary Objective 2		
Description	Designers should be able to offer services that pack and present results in powerful visualizations that assist interpretation, and support user collaboration		
Rationale	Systems with a rich palette of visualizations become important in conveying to the users the results of the queries in a way that is best understood in the particular domain		
Related scenario	C		
Contributors	Primary: WP4 Secondary: WP2, WP3, WP5		
Priority	SHOULD		
Satisfaction criteria	Rich palette of visualization for the analysed data is used		

Identifier	COMMON-07	Type	Non-Functional
Short name	HEADS Repository		
Objectives	Primary Objective 1, Secondary Objective 2		
Description	A repository with all HEADS components must be provided. HEADS IDE should provide easy access to this repository.		

Identifier	COMMON-07	Type	Non-Functional
Rationale	HEADS design tool should have a module library where service designers can search for existing modules and integrate them in design		
Related scenario	C		
Contributors	Primary: WP5 Secondary: WP6		
Priority	MUST		
Satisfaction criteria	A module library that can search for existing modules and integrate them in design is implemented		

Identifier	COMMON-08	Type	Functional
Short name	Deployment and Operation Management		
Objectives	Primary Objective 3, Secondary Objective 5		
Description	Support for deployment and operation of platforms and add-on services components must be provided. This includes provisioning, distribution, configuration, and monitoring of the components.		
Rationale	HEADS should have a life-cycle management system for handling distribution and upgrade of code		
Related scenario	C		
Contributors	Primary: WP5 Secondary: WP2, WP3, WP4		
Priority	MUST		
Stakeholder	HD-Service Operator		
Satisfaction criteria	Life-cycle management for distributed code achieved		

Identifier	COMMON-09	Type	Non-Functional
Short name	Data Security		
Objectives	Primary Objective 4, Secondary Objective 3		
Description	Information through the HEADS platform should be handled in a secure way to ensure integrity and authenticity of data.		
Rationale	In the media industry, all transactions between personnel must be secured. Moreover, all the media items must be hosted in a secure environment.		
Related scenario	A, B, C		
Contributors	Primary: WP2, WP3, WP4 Secondary: WP5		

Identifier	COMMON-09	Type	Non-Functional
Priority	MUST		
Stakeholder	HD-Service operator		
Satisfaction criteria	Secure exchange of information Concepts for adding secure communication channels where needed. Concepts are only added on a need to have basis. Devices and platforms selected for the respective scenario are offering secured transactions		

Identifier	COMMON-10	Type	Functional
Short name	Events Correlation		
Objectives	Primary Objective 4, Secondary Objective 1		
Description	HEADS should provide means to combine independent sensor streams to achieve higher precision output e.g. combine for fire fighter scenario sensors for smoke particles, temperature, and carbon monoxide centration in order to raise an alarm based on the combination of only almost dangerous changes in each sensor in the same time period. Verify the functioning of the sensor by continuously receiving events. Furthermore combine those alarms with unusual occurrence of reports about fire in social media based on text analysis		
Rationale	Achieve higher precision output and provide support for events correlation.		
Related scenario	A, B, C		
Contributors	Primary: WP4 Secondary: WP2		
Priority	MUST		
Stakeholder	HD-Service developer		
Satisfaction criteria	Several independent sensor streams are combined to achieve higher precision output		

Identifier	COMMON-11	Type	Non-Functional
Short name	Runtime Information and Monitoring		
Objectives	Primary Objective 3, Secondary Objective 5		
Description	Models should be updated with monitoring information coming from the running system (load, etc.) in order to support decision making e.g., should the system be adapted/re-deploy		
Rationale	Several metrics (like loading, fault tolerance, response time) should be monitored to ensure system's efficient behaviour		
Related scenario	A, B, C		
Contributors	Primary: WP5, Secondary: WP2, WP3, WP4		
Priority	SHOULD		
Stakeholder	HD-Service operator		
Satisfaction criteria	Models are enhanced with monitoring information from the running system		

Identifier	COMMON-12	Type	Functional
Short name	A methodology blueprint		
Objectives	Primary: Objective 1, Secondary: Objective 6		
Description	HEADS should provide a methodology blueprint consisting of: a) a graphical representation (outline) of the process integrated within the main tooling suite, and easily accessible throughout the suite, b) a cheat sheet explaining the overall process and details for each task and c) links to launch required tools for each task from the cheat sheet or the graphical outline		
Rationale	Ease the complexity of the process and increase HEADS acceptance from all stakeholders		
Related scenario	C		
Contributors	Primary: WP6 Secondary: WP5		
Priority	MUST		
Satisfaction criteria	A methodology blueprint is offered to the service designers		

Identifier	COMMON-13	Type	Functional
Short name	HD-Service Validation		
Objectives	Primary: Objective 1, Secondary: Objective 5		
Description	HEADS should provide the means (methods, tools and infrastructure) to test and subsequently validate HD-services against their intended objective and scope. By continuous monitoring of HD-services, deviations can be detected and corrective actions can be taken by the HD-Service developers in development phase and Service Operator and Administrator in production.		
Rationale	Minimize the costs and the risks of failures The system should be verified against its prior behaviour. HEADS should provide ways to conduct assessments through both functional and non-functional tests: a) both systems have the same functionalities, b) verify that the required targeted node's requirements are met HEADS should provide tools to allow the design of applications (HD-services) that can be distributed and deployed in the infrastructure in a sound way. I.e. the application should behave as expected when e.g. asynchronous channels are used.		
Related scenario	A, B, C		
Contributors	Primary: WP4, Secondary: WP1, WP2		
Priority	MUST		
Stakeholder	HD-Service developer		
Satisfaction criteria	Deployment and execution of large-scale test beds for HD-services The “new” system complies to the functionalities of its prior behaviour Consistency is ensured regardless of the infrastructure		

Identifier	COMMON-14	Type	Non-Functional
Short name	Cover the whole computing continuum		
Objectives	Primary: Objective 2, Objective 5		
Description	HEADS should provide the means (methods, tools and infrastructure) to generate and deploy code to at least 5 platforms representative of the computing continuum: - Platform group "Tiny" shall be supported with code generated image with a predefined set of tasks/channels. Tiny has not more than 2kbyte program memory and not more than 512byte data memory. - Platform group "Small" shall be supported with code generated image with dynamic task creation and dynamic channel creation on embedded target and towards host. Small has not more than 128kbyte program memory and not more than 64kbyte data memory. - Platform group "Gateway" shall be supported with code generated for a full operating system with HEADS-IDE components. Large is a restricted embedded computer running a full operating system like Linux. - Platform group "Mobile" shall be supported with code generated for a typical mobile device (smartphone, tablets) running a full operating system like Android - Platform group "Cloud" shall be supported with code generated for a typical virtual machine running a full-fledged OS, with (virtually) no resource constraint		
Rationale	To cover the resource constrained part of the continuum different approaches has to be applied.		
Related scenario	A, B, C		
Contributors	Primary: WP2, WP3		
Priority	MUST		
Stakeholder	HD-Service Designer, HD-Service Operator		
Satisfaction criteria	Code generators for platform groups "Tiny", "Small" and "Large" have to be provided. The static code (flash memory) generated must not introduce any substantial overhead. (max +25%)		

Contributing partners

SINTEF

Franck Fleurey
Franck.Fleurey@sintef.no

Brice Morin
Brice.Morin@sintef.no

Inria

Olivier Barais
barais@irisa.fr

tell.u

Knut Eilif Husa
knut.eilif.husa@tellu.no

software AG

Walter Waterfeld
walter.waterfeld@softwareag.com

ATC Athens Technology Center

Illias Spasis
I.Spais@atc.gr