HORIZON 2020 - ICT-14-2016-1

AEGIS

Advanced Big Data Value Chains for Public Safety and Personal Security

WP5 – AEGIS Data Value Chain
Early Community Demonstrators

D5.3 – Demonstrators Evaluation and Feedback – v1

Version 1.0

Due date: 30.06.2018	**Delivery Date**: 29.06.2018
Author(s): Dimitrios Miltiadou, Konstantinos Perakis, Stamatis Pitsios (UBITECH), Sotirios Koussouris, Marios Phinikettos, George Bikas, Penelope Dima (Suite5), Gianluigi Viscusi (EPFL), Alessandro Testa, Francesco Ascione (HDI), Alexander Stocker, Christian Kaiser, Andreas Festl (VIF) Elisa Rossi, Alessandro Calcagno (GFT), Dimosthenis Tsagkrasoulis (HYPERTECH), Yury Glikman (Fraunhofer)	
Editor: Dimitrios Miltiadou (UBITECH)	
Lead Beneficiary of Deliverable: UBITECH	
Dissemination level: Public	**Nature of the Deliverable:** Report
Internal Reviewers: Cinzia Rubattino (GFT), Mahmoud Ismail, Alexandru – Andrian Ormenisan (KTH)	

EXPLANATIONS FOR FRONTPAGE

Author(s): Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

Editor: Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author – if he has generated the Foreground - as well as an Editor – if he also edits the report on its own Foreground.

Lead Beneficiary of Deliverable: Only one. Identifies name of the partner that is responsible for the Deliverable according to the AEGIS DOW. The lead beneficiary partner should be listed on the frontpage as Authors and Partner. If not, that would require an explanation.

Internal Reviewers: These should be a minimum of two persons. They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically this review includes: Identifying typos, Identifying syntax & other grammatical errors, Altering content, Adding or deleting content.

AEGIS KEY FACTS

Topic:	ICT-14-2016 - Big Data PPP: cross-sectorial and cross-lingual data integration and experimentation
Type of Action:	Innovation Action
Project start:	1 January 2017
Duration:	30 months from **01.01.2017** to **30.06.2019** (Article 3 GA)
Project Coordinator:	Fraunhofer
Consortium:	10 organizations from 8 EU member states

AEGIS PARTNERS

Fraunhofer	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
GFT	GFT Italia SRL
KTH	Kungliga Tekniska högskolan
UBITECH	UBITECH Limited
VIF	Kompetenzzentrum - Das virtuelle Fahrzeug , Forschungsgesellschaft-GmbH
NTUA	National Technical University of Athens – NTUA
EPFL	École polytechnique fédérale de Lausanne
SUITE5	SUITE5 Limited
HYPERTECH	HYPERTECH (CHAIPERTEK) ANONYMOS VIOMICHANIKI EMPORIKI ETAIREIA PLIROFORIKIS KAI NEON TECHNOLOGION
HDIA	HDI Assicurazioni S.P.A

Disclaimer: AEGIS is a project co-funded by the European Commission under the Horizon 2020 Programme (H2020-ICT-2016) under Grant Agreement No. 732189 and is contributing to the BDV-PPP of the European Commission.

EXECUTIVE SUMMARY

The scope of deliverable D5.3 is to document the efforts undertaken within the context of Tasks 5.2, 5.3, 5.4, 5.5 and 5.6 of WP5. Towards this end, the deliverable builds on top of the outcome and knowledge extracted by the deliverables D5.1 and D5.2 in order to document the progress of the AEGIS demonstrators and perform the evaluation of the AEGIS platform and the AEGIS demonstrators during the first (early) phase of demonstrators' implementation.

Within the context of the deliverable D5.3 the demonstrators' operation and execution during the first (early) phase of their implementation is documented. Furthermore, the evaluation of the demonstrators' implementation and the AEGIS platform v1.00 is performed based on the AEGIS evaluation framework.

To meet its goal, D5.3 presents a comprehensive description of the evaluators involved in the AEGIS evaluation framework and the detailed results of the evaluation performed by these evaluators. The evaluators are presented focusing on their technical background and knowledge, their role in the project and their involvement in the evaluation process. As the AEGIS evaluation framework is formed with both quantitative and qualitative methods, the results of the holistic evaluation of the AEGIS platform v1.00 following both methods are documented. Additionally, the challenges encountered by the demonstrators in respect to the AEGIS platform and a series of recommendations for the enhancement of the AEGIS platform are presented. Furthermore, D5.3 provides an overview of the current implementation status of each demonstrator, describing in detail the work performed during the first (early) version of the demonstrator along with the results of the scenario(s) execution for each demonstrator. Following the scenarios execution, the results of the evaluation of each demonstrator are presented, as well as the challenges faced during the implementation of the demonstrator and a set of recommendations for the upcoming versions of the demonstrators.

In the next steps the outcomes of this deliverable will be further analysed by the AEGIS platform developers in order to perform the necessary refinements and updates in the AEGIS platform that will facilitate the implementation of the second (medium) phase of the demonstrators. The evaluation of the AEGIS platform v2.00 and the demonstrators' operation and execution during the second (medium) phase will be documented in the upcoming deliverable D5.4 entitled "Demonstrators Evaluation and Feedback – v2" which will be delivered in M24. It should be noted at this point that the demonstrator evaluation and feedback is a living process that will last till M30, when the final demonstrator evaluation and feedback (corresponding to D5.5) and the final evaluation, impact assessment and adoption guidelines (corresponding to D5.6) will be delivered.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AAL	Active and Assisted Living
API	Application Programming Interface
CO	Confidential, only for members of the Consortium (including the Commission Services)
CPU	Central Processing Unit
CSP	Care Service Provider
CSV	Comma Separated Values
D	Deliverable
DoW	Description of Work
H2020	Horizon 2020 Programme
GUI	Graphical User Interface
HVAC	Heating Ventilation and Air Conditioning
IT	Information Technology
JSON	Javascript Object Notation
KPI	Key Performance Indicator
PSPS	Public Safety and Personal Security
R	Report
RTD	Research and Development
SHAL	Smart Home and Assisted Living
VOC	Volatile Organic Compounds
XML	Extensible Markup Language
WP	Work Package
Y1	Year 1

1. INTRODUCTION

The scope of the current section is to introduce the deliverable and familiarise the user with its contents. Towards this end, the current section summarises the objective of the current deliverable, its relation to the other work packages and corresponding deliverables and analyses its structure.

1.1. Objective of the deliverable

The scope of deliverable D5.3 is to document the efforts undertaken within the context of Tasks 5.2, 5.3, 5.4, 5.5 and 5.6 of WP5. Towards this end, the scope of the deliverable is to document the demonstrators' operation and execution during the first (early) version of their implementation, as well as the first evaluation of the results of their implementation. Additionally, within the scope of the deliverable the evaluation of the AEGIS platform v1.00 is documented.

More specifically, the objectives of the deliverable D5.3 are as follows:

- Provide a comprehensive description of the evaluators included in the AEGIS evaluation framework, focusing on their knowledge and experiences, role in the project and their involvement in the evaluation process.
- Document the results of the evaluation of the AEGIS platform v1.00 during the first phase of the demonstrators' implementation.
- Present the challenges faced by the demonstrators in respect to the AEGIS platform v1.00, as well as provide a series of recommendations for the further improvement of the AEGIS platform.
- Provide a detailed overview of each demonstrator, focusing on the current implementation status of the demonstrator and the work performed during the first (early) version of the demonstrator.
- Document the results of the scenarios that were executed in the context of the first (early) version of each demonstrator as documented in deliverable D5.2. For each scenario, the results of the corresponding steps are provided in combination with the related implementation details (where applicable).
- Present the results of the evaluation of each demonstrator for their first (early) version in accordance to the AEGIS evaluation framework.
- Present the challenges faced during the demonstrators' implementation, as well as a series of recommendations for the further improvement in the upcoming versions.

According to the AEGIS evaluation framework, the evaluation of the AEGIS platform and the AEGIS demonstrators, is performed following both quantitative and qualitative methods. For the evaluation of the AEGIS platform both methods were performed. However, it should be noted that concerning the evaluation of each demonstrator the consortium decided to proceed only with the quantitative method at this stage since the demonstrators' implementation is not currently at the level of maturity that would provide valuable results through the qualitative method. In the upcoming versions of the demonstrators both methods will be performed.

1.2. Insights from other tasks and deliverables

The deliverable builds on top of the work reported in WP5. In particular, the work performed in WP5, as reported in D5.1 and D5.2, provided the AEGIS evaluation framework, as well as the methodology on how to implement the framework during the implementation phase of the three demonstrators of the AEGIS project. The outcomes of the deliverable D5.2 served as guidance on how the evaluation of both the AEGIS platform and the AEGIS demonstrators will be performed. The AEGIS platform evaluation plan, as well as the scenarios defined for each demonstrator and the documented evaluation plan for each demonstrator had driven the assessment performed during the first (early) version of the demonstrators.

1.3. Structure

Deliverable D5.3 is organised in seven main sections as indicated in the table of contents:

- The first section introduces the deliverable. It documents the scope of the deliverable and briefly describes how the document is structured. It also documents the relation of the current deliverable with the other deliverables, and how the outcomes of other deliverables are received as input to the current deliverable.
- Following the introductory section, the second section describes the evaluators involved in the AEGIS evaluation framework, providing information of their technical background and knowledge, how are they related to the project and their involvement in the evaluation.
- The third section documents the results of the evaluation of the AEGIS platform v1.00. In this section both the results of the quantitative and the qualitative evaluation are documented. Additionally, in this section the challenges faced related to the AEGIS platform during the first (early) version of the demonstrators are documented and a series of recommendations for the enhancement of the platform are presented.
- Following the second section, the upcoming sections are presenting the current status and the evaluation of each one of the three AEGIS demonstrators, section four for the Automotive demonstrator, section five for the Smart Home & Assisted Living demonstrator, and section six for the Insurance demonstrator. At first, for each demonstrator the overview of the demonstrator and the current status is documented. Following the current status description, the results of the scenario(s) execution within the context of the first (early) version of the demonstrator are documented. Following the scenarios execution, the demonstrator evaluation is presented, providing the results of the quantitative evaluation for each demonstrator. At last, the challenges faced during the implementation of the demonstrator are presented along with a list of recommendations for the upcoming versions of the demonstrator.
- Section 7 concludes the deliverable. It outlines the main findings of the deliverable, which will guide the future research and technological efforts of the consortium.

2. AEGIS EVALUATORS

The AEGIS evaluation framework, as documented in D5.1 and D5.2, is aiming at evaluating the AEGIS platform in terms of how it addresses the requirements and expectations of the AEGIS PSPS stakeholders of the three demonstrators and beyond. The AEGIS evaluation framework is also aiming at performing a holistic evaluation of the platform. As such, the evaluators included in the framework are the main direct beneficiaries of the AEGIS platform, which are the PSPS data scientists from the three different sectors that will develop the data-driven AEGIS PSPS services and the PSPS end users for the corresponding sectors that will consume the developed services. Additionally, the AEGIS platform developers are also involved in the evaluation framework in order to assist in the evaluation focusing mainly on the technical evaluation of the platform.

In the forthcoming subsections, the AEGIS evaluators are described focusing on their profile, role in the project and their involvement in the evaluation.

2.1. PSPS Data Scientists

In general, a data scientist is a professional that collects, analyses and extracts information from large amount of data using a variety of big data analytic tools, in order to extract useful information and insights that will help a business improve operations and gain competitive advantage over rivals or provide the necessary results and advancements in a research program. Data scientists have the experience and proper skills to use advanced analytics technologies that include machine learning and predictive modelling techniques, in order to unveil useful insights beyond statistical analysis. Nowadays, with the data explosion from the voluminous amounts of data produced and collected from various heterogeneous sources the role of the data scientists has become very crucial for the enterprises and organisations towards the aim of maximising the results of the data processing.

In terms of evaluation, the data scientist is one of the most important stakeholders for a Big Data analytics ecosystem. The mix of experience and analytics skills makes the data scientist the most appropriate person to perform the evaluation and identify the deficiencies of the processes and tools of the platform. Through their extended experience and usage skills with a variety of tools and technologies related to big data analysis, data scientists are able to evaluate and propose the best practises in the data collection, data management, data processing and data analysis processes.

In the context of the AEGIS project PSPS data scientists from three different sectors will be involved in the development of the AEGIS PSPS services:

- PSPS data scientists from the automotive sector. In the context of the AEGIS Automotive demonstrator, the PSPS data scientists from VIF will implement the data-driven automotive services that will create data-driven workflows on the AEGIS platform by exploring the relevant collected vehicle data and other automotive-related sources.
- PSPS data scientists from the smart home and assisted living sector. In the context of the AEGIS Smart Home and Assisted Living demonstrator, data scientists from Hypertech, UBITECH and Suite5 will implement a series of data-driven services that

will create data-driven workflows on the AEGIS platform by exploring the collected data from smart home devices, wearables and other assisted living devices.

- PSPS data scientists from the insurance sector. In the context of the AEGIS insurance demonstrator, PSPS data scientists from HDI will implement the data-driven Insurance services that will create data-driven workflows on the AEGIS platform by analysing the events detected by the AEGIS tools after exploring data for weather, news and crime open data.

In accordance with the AEGIS Evaluation Framework that is documented in deliverable D5.1, the PSPS data scientists from the three different sectors mentioned above will be involved in the evaluation of the AEGIS platform. More specifically, the AEGIS platform will be evaluated in terms of functionalities offered by the platform and required by the PSPS data scientists in order to implement the PSPS data-driven services through the AEGIS demonstrator-specific evaluation cases. Additionally, the PSPS data scientists will participate in guided interviews that will also evaluate the perceived usefulness and usability of the service design process.

2.2. PSPS End Users

The PSPS end users are experienced professionals from different sectors with different technical and theoretical background. As such, the PSPS end users have different requirements and expectations from the AEGIS platform from the rest of the stakeholders. The PSPS end users are the main consumers of the AEGIS PSPS services that will be developed from the PSPS data scientists. Their expectations vary depending on the requirements of their corresponding sectors but in general, the main goal of the PSPS end user is to exploit the platform and the developed services with the aim of enhancing a process, a product or a service and in some cases introducing new ones.

In terms of evaluation, the end user is an important stakeholder for a Big Data analytics ecosystem. The end user is the most appropriate person to perform the evaluation and identify the deficiencies of the usefulness of the platform in order to fulfil their tasks and gain valuable insights, as well as of the usability, ease of use and quality of the platform. As the consumers of the PSPS data-driven services, the end users are able to evaluate the features of the platform, as well as the added value offered by the services developed by the PSPS data scientists.

In the context of the AEGIS project PSPS end users from three different sectors will consume the AEGIS PSPS services developed by the involved PSPS data scientists:

- PSPS end users from the automotive sector. In the context of the AEGIS Automotive demonstrator, the PSPS end users from VIF will consume the AEGIS PSPS services created by the corresponding PSPS data scientists within the AEGIS Automotive demonstrator. The end users will evaluate the services in terms of gaining valuable insights for safer driving and safer roads by the analysis of the driving styles and driving behaviour.
- PSPS end user from the smart home and assisted living sector. In the context of the AEGIS Smart Home and Assisted Living demonstrator, PSPS end users from Hypertech, UBITECH and Suite5 will consume the AEGIS PSPS services created by the corresponding PSPS data scientists within the AEGIS Smart Home and Assisted Living demonstrator. The end users will evaluate the services in terms of efficiency and

effectiveness in monitoring at-risk individuals and patients, in case the of care providers or doctors, and in terms of usefulness, usability and non-intrusive behaviour in assisting their everyday lives in the case of at-risk individuals and patients.

- PSPS end users from the insurance sector. In the context of the AEGIS insurance demonstrator PSPS end users from HDI will consume the AEGIS PSPS services created by the corresponding PSPS data scientists within the AEGIS insurance demonstrator. The end users will evaluate the services in terms of benefits achieved by the developed services, the legibility of the produced reports and the customer's feedback depending on their role in the company.

In accordance with the AEGIS Evaluation Framework that is documented in deliverable D5.1, the PSPS end users from the three different sectors mentioned above will be also involved in the evaluation of the AEGIS platform. More specifically, the experiences and satisfaction of the PSPS end users in regards to usefulness, usability and business relevance of the developed PSPS data-driven demonstrator services will be leveraged and will be included in the holistic evaluation of the AEGIS platform with the use of guided interviews.

2.3. AEGIS Platform Developer

The AEGIS platform developer is the experienced professional involved in the development process and production of the platform. The platform developer has extended technological know-how and experience in the Big Data technologies and tools, as well as in the software design and implementation. The platform developer is involved in all the relative phases of the platform development, from the requirements elicitation and the extraction of functional requirements, the design and specification of the platform components and architecture to the implementation and integration of the AEGIS platform. The AEGIS platform developer is aiming at developing and offering novel services and applications that will allow data scientists from the PSPS-related industries to develop advanced and intuitive PSPS data-driven services that will be exploited by the PSPS end users.

In terms of evaluation, the AEGIS platform developer is the most appropriate person in order to assist in the AEGIS platform evaluation in terms of software quality characteristics. The software quality characteristics are measured with the list of technical Key Performance Indicators (KPIs) of the AEGIS platform, as defined in section 2.1 of D5.2, and is related to the quantitative evaluation of the AEGIS platform that will be executed by the AEGIS platform developers.

3. AEGIS PLATFORM EVALUATION

The AEGIS evaluation framework is aiming at performing a holistic evaluation that will capture the satisfaction of the AEGIS PSPS stakeholders with the platform and will provide valuable feedback to the AEGIS platform developers that will drive the enhancements and refinements of the AEGIS platform towards the success of the AEGIS project. In accordance to the AEGIS evaluation framework, the AEGIS platform evaluation will be performed in three different iterations, one for each of the phases of the demonstrators' implementation, providing the useful insights on the requirements and expectations of the AEGIS PSPS stakeholders on each phase.

In regards to the AEGIS platform evaluation, the consortium decided to follow a twofold approach. On the one hand, the AEGIS platform is evaluated following the quantitative method that is based on a list of Key Performance Indicators (KPIs), as defined in deliverable D5.2. On the other hand, the AEGIS platform is also evaluated following the quantitative method via guided interviews with key stakeholders. The details of this twofold approach are covered in detail in deliverable D5.2.

In the following subsections of section 3, the results of both the quantitative and the qualitative evaluation of AEGIS platform during the early version of the demonstrators' implementation are documented. Additionally, the challenges faced in relation with the AEGIS platform during the first (early) version of the demonstrators are documented along with a series of recommendations for the future versions of the platform.

3.1. Quantitative Evaluation

For every technological project in scale, the quality measurement of the developed solution has become increasingly important, especially when different people are involved in the various parts of the developed solution. In order to ensure and safeguard the quality of the solution it is crucial to continuously monitor and improve the software quality characteristics of the developed solution.

The quantitative evaluation of the AEGIS platform is focusing on the technical evaluation of the platform. For this evaluation, the software quality characteristics and the software product evaluation process model as defined by the ISO/IEC 25010:2011 were used as a guidance in order to generate the list of technical KPIs of the AEGIS platform that were documented in deliverable D5.2. The software quality characteristics that are evaluated are the following:

- Functional suitability
- Performance efficiency
- Compatibility
- Usability
- Reliability
- Security
- Maintainability
- Portability

Through the list of technical KPIs, the quantitative evaluation of the AEGIS platform is performed towards the aim of the providing quality assurance and control in all three versions of the demonstrators. The first quantitative evaluation was performed by the AEGIS platform

developers, that were involved in the development process and the production of the platform, and Table 1 presents the results of the evaluation of the AEGIS platform (first version released at M12) during the first early version of the demonstrators' implementation.

Sub-characteristics	KPIs	Calculation Type	Mandatory / Optional	Value	Comments
Functional suitability					
Functional completeness	Portion of completed User Stories	[Completed User Stories] / [Iteration Cycle of User Stories] * 100%	M	100%	All use cases planned for the early version were executed.
Functional correctness	Portion of User Stories without reported bugs	[Completed User Stories without bugs] / [Iteration Cycle of User Stories] * 100%	M	90%	A list of bugs was identified however they were successfully addressed.
Functional appropriateness	Straightforward task accomplishment	Are tasks completed without the use of unnecessary steps? [Yes/No]	O	No	Although all planned tasks were successfully accomplished, in some tasks assistance was required from the respective contact persons.
Performance efficiency					
Time behaviour	Average latency	[Total response time] / [Number of requests]	M	~1.2 sec	Average latency was measured with tools such as Chrome Dev Tools. In the next releases this should enhanced to be <= 1 sec
	Throughput	[Total Number of Kilobytes] / [Total Time of Operation]	M	~ 270 KB/sec.	Value documented while previewing files and downloading files.

Resource utilisation	Mean CPU Utilisation	[Σ[%CPU utilisation probes]] / [Number of probes]	M	<45%	Based on the resource monitoring tool of the platform
	Mean memory usage	[Σ[RAM Megabytes used in each probe]] / [Number of probes]	M	<20%	Based on the resource monitoring tool of the platform
	Maximum memory usage	Maximum % RAM Memory utilisation recorded	M	37%	Based on the resource monitoring tool of the platform
	Maximum processing power used	Maximum % CPU utilisation recorded	M	90%	As the resource management is performed by YARN the appropriate resource allocation is always performed according to the provided configuration.
Capacity	Maximum filesize upload	Total number of Kilobytes of files	M	250MB	Note: This is size of the current biggest file available.
	Maximum file system size[1]	Total number of Kilobytes of files	M	76GB	Note: This is the current size of HopsFS that can scale according to the needs of the project.
Compatibility					
Co-existence	Ability to Co-Exist (host in a single environment)	Can the AEGIS platform operate in shared environment? [Yes/No]	O	Yes	

[1] AEGIS platform utilises the distributed file system HopsFS. Thus the database size metric was modified.

Interoperability	% of APIs coverage	[Number of integrated systems exposing or consuming data through API] / [Total number of integrated systems] * 100%	M	100%	All integrated components / services are integrated through APIs
	Ability to handle different datasets	Can the AEGIS platform consume datasets from different formats (e.g. CSV, JSON, XML files)? [Yes/No]	M	Yes	No limitations on the file formats HopsFS can store. Files can be processed using the appropriate libraries by the data scientist.
		Can the AEGIS platform provide datasets in various formats (e.g. CSV, JSON, XML files)? [Yes/No]	M	Yes	No limitations on the file formats HopsFS can store and provide.
Usability					
Appropriateness recognisability	% Positive feedback on appropriateness based on the available documentation	[Number of positive response] / [Total number of responses] * 100%	O	Not applicable.	Due to the early stage of the platform implementation it lacks of documentation. This should be properly addressed within the next releases of the platform.
Technical Learnability	% Coverage of features with learning documents	[Unique number of help documents mentioning a feature] / [Total number of features	M	Not applicable.	Due to the early stage of the platform implementation it lacks of documentation. This should be

		available] * 100%			properly addressed within the next releases of the platform.
Ease of Use	Dashboard availability	Is there an available dashboard or wizard with easy navigation? [Yes/No/Partially]	O	Partially	The UI of the platform is almost straightforward, however in some cases several improvements should be made in order to facilitate the easy navigation within the platform.
User error protection	% Coverage of input fields with error protection methods	[Number of error protected fields] / [Total number of critical input fields] * 100%	M	100%	All input fields in the UI are protected.
User interface aesthetics	% Positive feedback on user interface aesthetics poll	[Number of supported screens] / [Total number of different screens] * 100%	O	80%	Since this is the early version of the platform several enhancements and refinements can be performed in the upcoming versions.
	Responsiveness	[Number of supported screens]/[Total number of different screens] * 100%	M	100%	No inaccessible or malformed screens were identified.
Technical Accessibility	WCAG 2.0	[None/ A/ AA/ AAA]	M	A	

	Conformance Level[2]				
Reliability					
Maturity	Maximum Concurrent users	Maximum number of concurrent users recorded	M	27	
	Simultaneous requests	Maximum number of simultaneous requests	M	In terms of containers allocated the maximum value recorded was 12, in terms of applications was 5 concurrent application and in terms of requests to filesystem 11 concurrent requests.	In the next versions of the platform this should be recorded via automated stress testing.
Availability	% Monthly availability	[1-[Downtime in minutes] / [Total month minutes]] * 100%	M	>95%	Most of the downtime recorded was due to infrastructure upgrade.
	Success rate	[Number of correctly completed requests] / [Total number of requests]	M	~90%	The problematic requests were successfully addressed with bug fixing.
Fault tolerance	% of identified Software problems	[Critical Software Issues] / [Total number of	M	~25%	Usage of specific AngularJS code produced issues

[2] https://www.w3.org/WAI/WCAG20/quickref/

	affecting the platform	Software faults detected] * 100%			in Glassfish. Problems have been identified and fixed.
	% of identified Hardware problems affecting the platform	[Critical Hardware Issues] / [Total number of Hardware faults detected] * 100%	M	100%	Critical hardware issues (such as system going out resources) were identified at the early stages and fixed.
Recoverability	Mean recovery time from Software problems	[Total recovering time from Software issues] / [Total number of Software issues in need of recovery]	M	~ 1 hour	
	Mean recovery time from Hardware problems	[Total recovering time from Hardware issues] / [Total number of Hardware issues in need of recovery]	M	~ 1 hour	
Security					
Confidentiality	Incidents of ownership changes and accessing prohibited data	Number of recorded incidents	M	None	
Integrity	Incidents of authentication mechanisms breaches	Number of recorded incidents	M	None	
Non-repudiation	% Activities reporting	[Number of log categories] / [Total number	M	90%	

		of system operations]			
Accountability	User actions traceability	Are usernames included in each activity log entry uniquely? [Yes/No]	M	Yes	Logging mechanisms provide all the appropriate reporting information
Authenticity	Level of User authenticity	Can you identify that a subject is the one it claims to be? [Yes/ No/ Partially]	M	Yes	
Maintainability					
Modularity	% of modularity	[Number of components that can operated individually] / [Total number of components] * 100%	M	100%	
Reusability	% of reusable assets	[Number of assets that can or are reused] / [Total number of assets] * 100%	M	100%	
Analysability	Level of analysability	Can the changes in the performance of the AEGIS platform be efficiently evaluated after each upgrade? [Yes/No]	O	Yes	The system offers monitoring tools with performance indications.
Modifiability	% of update effectiveness	[Number of updates performed without operational issues] / [Total number of updates] * 100%	M	95%	A small issue was detected and fixed during one update operation.

Testability	Level of testing	Are tests able to probe the behaviour of the AEGIS platform? [Yes/No]	M	Yes	
Portability					
Adaptability	Mean number of errors per hardware change/ upgrade	[Total number of errors recorded] / [Total number of hardware changes]	M	None	
	Mean number of errors per software change/ update	[Total number of errors recorded] / [Total number of software changes]	M	<2%	All issues were immediately identified and fixed.
Replaceability	% of software products replaceability within AEGIS platform	[Number of replaceable software components] / [Total number of used software components] * 100%	M	100%	

Table 1: AEGIS Platform quantitative evaluation

3.2. Qualitative Evaluation

In addition to the quantitative evaluation of the AEGIS platform, the qualitative evaluation approach is aimed to shed additional light on aspects including usefulness, user acceptance and satisfaction, as well as ease of use. The qualitative evaluation will provide valuable feedback that a quantitative evaluation alone would never be able to generate and will feed this knowledge back to the platform developers to further improve the AEGIS platform.

Data scientist of the demonstrator (pilot) partners are supposed to implement the demonstrator, by using the capabilities of the AEGIS platform. The evaluation was conducted together with one data scientist per demonstrator in a semi structured way. The evaluation methodology is covered in detail in D5.2. There are three major aspects, which have been in focus of the qualitative evaluation, (1) the knowledge and experiences of the data scientists in charge of demonstrator implementation, (2) the perceived overall usefulness of the AEGIS platform, and (3) the perception of the technical process of implementing the own demonstrator:

- What knowledge and experiences have you already gained in (big) data-related activities and projects?

- From what you have so far learned in the application of the AEGIS platform, how useful do you perceive the AEGIS platform in general?
- How did you perceive the technical process of developing and implementing your AEGIS demonstrator with/on the AEGIS platform?

These general questions included a set of sub-questions supporting the interviewer and ensuring that light is shed on all relevant aspects. The results of these interviews are presented in an aggregated form in the following subjections.

Knowledge and experiences of the data scientists

All data scientists in charge have already gained experiences with different tools and languages for data analysis including e.g. R, Python, Zeppelin, Jupyter, Spark, SAS, SPSS, KNIME, or Power BI. They have also already gained experiences in analysing data within their own industrial domains (e.g. insurance data, vehicle data, …). All data scientists also have gained a relevant technical education from their respective universities (mathematics, statistics, or data analytics) and are therefore well qualified to execute the implementation of the demonstrators in their domains.

Perceived usefulness of the AEGIS platform

It must be said that during the first evaluation a very early version of the AEGIS platform has been evaluated. Currently the AEGIS Platform can provide only basic features (Hadoop filesystem, Spark, Zeppelin/Jupyter notebooks) to support data scientists. Therefore, results concerning comfortability, usefulness of the platform for the job, usability of the platform, platform performance, and joy of use must be interpreted differently. If a final and more mature version of the platform would have been used for demonstrator implementation, the evaluation results would surely be different. However, the overall impression of the platform that the data scientists had, is already very good considering the basic offering of the platform as well as the vision of the platform's capabilities at the end of the project.

The following bullet points capture experiences made by the data scientists in charge of demonstrator implementation as well as their perceptions of the platform use (based on what they have learned during their experiments and implementation works), which are worth to mention and communicate to the platform development team.

Experiences made while implementing the first version of the automotive demonstrator

- The PaaS "Platform as a service concept" (Hadoop file system / Spark / Zeppelin) makes sense and saves the data scientists a lot of work (setting this environment up and maintaining it requires a lot of experience and infrastructure). However, the current version of the platform provides only basic infrastructure (in a package), limited support (with tools) in programming, and additional data sources were not available (at the time of demonstrator implementation).
- The current platform already allows adjusting many things (including Spark options), which is very useful, but at the same time very hidden in UI. Furthermore, it is very simple to install further libraries to be used by the data scientist, which is good to extend its scope.

- Currently the usability / user guidance of the platform is somewhat cumbersome, e.g. if one clicks at a dataset icon (a click on the bottom provides further information while a click on the top opens the data dataset). If one wants to write data into a dataset, the dataset has been explicitly made editable (which cannot be done during dataset creation). A new tab cannot be opened from the menu (e.g. via a right click). The autocomplete functionality of the PySpark interpreter is not satisfactory, as it does not provide the expected context-sensitive information if "shift +" is pressed.
- Version control is an issue. Zeppelin should have an integrated version control, but it does not seem to work properly. The current procedure for programming was to download the notebook from the platform whenever the data scientist wants to change something, then locally edit it, and finally save it back into a new version to the platform. This is feasible, if the data scientist works alone on a project, but not efficient if the data scientist must cooperate with more developers.
- The performance of the platform could be further improved, as the standard settings are not optimal. If the computing power is adjusted (with help of the platform operator), the performance of the platform is fine.
- With respect to the initial expectation of the platform (before project start), the quality of the platforms as well as its capabilities could be further improved. Additional data sources, easier linking of data, a workflow modeller and other features are expected. However, treating the (current version) of the platform as an infrastructural service, the quality can already be rated as very good.

Experiences made while implementing the first version smart home & assisted living demonstrator

- The AEGIS platform is a great platform for data scientists that combines a lot of things and capabilities that people working in the data science sector need. It has Hadoop, supports Spark, Kafka, notebooks (has Jupyter and Zeppelin), which are technologies that many data scientist like and will use. Finally, it uses elastic search and further tools that are very hot right now in the whole data industry. It is very innovative that AEGIS holds all these tools (in one platform).
- However, the platform currently does not integrate the R language, which is a pity. It would also be great if notebooks could be directly opened from the platform, which however also would require changes in core functionality.
- For an experienced user with sufficient knowledge of data science and engineering, the platform is already comfortable. However, for less experienced users (e.g. business users with no background in programming) right now the platform might not seem as the most appropriate tool to solve their challenges. It is hard to imagine to what extent both type of users, experienced and non-experienced, can ever be supported in a best possible way.
- Some data scientists could also opt for the option of having independent tools (e.g. standalone visualiser) and not all services integrated in one notebook (serial execution of Query Builder, Algorithm Execution Container and Visualiser).
- One drawback is that the only way to extend the platform is with notebooks, or tools that are outside the platform (e.g. offline cleaner /outside the platform). A solution approach to solve this issue would be to develop an AEGIS portal including two elements, current AEGIS features (e.g. query builder) that are based on the notebook

functionality of hops works and extended AEGIS features (e.g. data anonymizer) that can be accessed from this portal.

Experiences made while implementing the first version the insurance demonstrator

- The current AEGIS platform satisfies the needs of data scientists, considering that it is the first prototype. The expectation is high as the real goal will be how well everything will be integrated, e.g. how easy will be to go from the Query Builder to the Visualiser or the algorithm container, and how the whole components will cooperate to create a good big data analytics user experience.
- The idea to make some general analytics tools available from AEGIS is a good one. However, it would be very useful if the platform would provide (more specialised) tools for geographic analysis: A flexible and useful set of tools for analysing geographical data may include e.g. tools for showing customers as markers on a map with different colours based on customer characteristics, or tools for drawing areas on a map and then calculate the number of points in it. These would be features which are so far not covered by other tools (as they are usually not very easy to implement). Practically, geographic data management and analysis requires a lot of code and experience from the data scientist.
- For a big data platform, it is important that AEGIS could enrich the company's data with open data sets from the platform that are categorized in some way to make them better accessible. This would enrich the companies' data (e.g. customer data) providing the right tool to display the geographic data linked to customer data in maps (e.g. regarding natural and social risks of countries). It would probably be more useful if AEGIS would not just offer generic analytics tools but would have some integration with geographic data management. Some of the further (envisaged) analytics tools of the platform are good, but they are also covered by other very big companies in their tools.
- The usability/comfort of the platform can be judged as being satisfactory, considering that the first prototype was used. However, the project page of the platform is sometimes not very intuitive. A user guide for the platform would be very helpful.

Perception of the technical process of implementing the own demonstrator

Experiences made while implementing the first version of the automotive demonstrator

- The relevant code was already developed offline in R, before. Hence the part of the project which was concerned with experimenting and interacting with the vehicle/driving data using the AEGIS notebooks was obsolete. Hence, no experiences could be reported how comfortable and useful it is to interact with vehicle data by using the notebooks of the AEGIS platform.
- It was somewhat challenging to find out basic features of the platform, which by the way is not the fault of the platform. One must know that Zeppelin runs PySpark as interpreter, and to use this interpreter needs a Spark context and an SQL context. Both are automatically created at start and receive a dedicated name – however, the data scientist did not know that at start.
- Loading and saving files, as well as listing files from a folder, are a challenge if the data scientist has no experience with Spark. It was hard to find a relevant documentation,

and for a non-experienced Spark user it is not always clear where Spark begins (and where the AEGIS platform ends).

- The implementation of the demonstrator on the platform has been done in about 30 hours excluding the visualisation part. However, the major part of data analysis has been done offline previously so that the data scientist exactly knew what to do on the platform. The algorithms had to be transferred from R language to Python, though. Also, the parallelisation of the trip analytics (which obviously had not been done before in R offline) had to be implemented on the platform. Both worked fine.
- The basic expectation (ideal scenario) of the data scientist before the start of the project was to implement the demonstrator without the need to create substantial lines of code. However, this was not the case.
- Beneficial to the automotive demonstrator would be a file system with more structure, e.g. in a table form like how databases are structured. The first step for the automotive demonstrator always is to bring data into a tabular form (which is possible though). However, loading, transforming, storing, etc. are always extra steps in data processing. Hence, relational databases would be of interest. There is a lot of structure in our datasets: A column is a measurement signal, and a row is a measurement. Trip measurements belong to one trip, i.e. many measurements belong to one trip that has one driver and one vehicle, which creates a 1: n hierarchy, beaglebone/driver_id/vehicle_Id/trip_Id/, which is classical for relational databases as the data scientist already has developed a well-defined data model.
- The support from the platform development team during demonstrator implementation was great and really helped a lot. However, some more introduction in how to achieve the demonstrator on the platform would have been beneficial.

Experiences made while implementing the first version smart home & assisted living demonstrator

- The data scientist has worked a lot with the platform and already digged deep into the platform's code.
- During demonstrator implementation, the platform was used for a short period of time (~ a week on/off for smart home data pre-processing). Code and algorithms were developed offline before using python. Then smart home data was uploaded to the platform, a notebook was created, and the offline-created code was copied into the notebook, which worked good from the get-go. Only minimal changes had to be done on the offline code to make it work on the platform.
- So far, the data scientist is very happy with the platform and the current development process. However, it did not stretch the possibilities of the platform. Apart from some technical difficulties (where persons of the platform development team had to be asked for support), implementing the demonstrator went smoothly.
- The data scientist had difficulties locating the created project, because the drop-down list does not show all projects (one must remember name and search for it). When the data scientist first opened the platform, she could not understand why we needed all the tabs on the left side. Hence the UI of the platform could be further improved from the perspective of a demonstrator data scientist.
- After first problems with the platform/system related to the execution of plugins and the availability of RAM (constantly crashing due to a problem with the server used) everything went fine.

Experiences made while implementing the first version the insurance demonstrator

- The focus was on implementing core requirements, not demonstrator-specific requirements. The main activities were on the event detection tool training. Thereby a python script was created to stream tweets. For a successful training, the tool needs 1000s of tweets in Italian language. However, as twitter is not so common in Italy, and floods seemed to be seldom, too, this was a challenge.
- Another script was done for the Query Builder (adding some algorithms) to make it work with an in-house HDI dataset.
- If all components are set up, the process of implementing the demonstrator is perceived satisfactorily. The satisfaction is good, even though the overall process of the HDI demonstrator is manual for some aspects (due to company-internal regulations on security and privacy).

The consortium is very collaborative and helpful which greatly supports the implementation. There is a high trust that everything will go well. However, the early demonstrator (and the platform) are a bit behind of time – from the perspective of the demonstrator data scientist.

3.3. Challenges and recommendations

One major challenge for demonstrator implementation was the fact that only the first (and very preliminary) prototype of the AEGIS platform (M12) could be used. Therefore, only basic platforms functions of AEGIS platform were practically available for the data scientists in charge, as additional powerful components (like Query Builder or Visualiser) are still under refinement and development as defined in the DoA. Therefore, many demonstrator-specific functionalities had to be hard-coded in Zeppelin/Jupyter notebooks by the data scientists (and have even been developed offline in Python or R).

Nevertheless, a series of recommendations can be derived from the platform evaluation:

- The UI of the platform could be further improved, not only with respect to data management. The overall goal should be to facilitate straightforward accomplishment of tasks through easy navigation within the platform without unnecessary steps. Especially serving the needs of two very different target groups, experts (data scientists) and non-experts (business users), will be a challenge.
- Version control with notebooks is an important issue. A guideline on how to use version control if two or more data scientists are collaborating in a project would be valuable.
- The auto complete issue in Zeppelin seems not to work as intended.
- Support for R language would be great for data scientists, as this is a common language for most of them.
- The standard settings of the platform in terms of performance are not optimal and could be further improved.
- If it is possible to bypass somehow the default mechanism of Jupyter and Zeppelin and allow notebooks to be opened directly. This would be a great enhancement of the platform in terms of comfort and usefulness.
- A user guide could ease the use of the platform and could thereby reduce (initial) efforts of trying out things. It is crucial to provide the appropriate documentation that will help the users further explore the features and the functionalities of the platform.

- The concept of the platform depends on a hierarchy of datasets. Beneficial for some data analytic activities would be a file system with more structure, like how relational databases are structured.
- The capabilities for geo data analysis and geo data management could be improved, as this would be a valuable USP for the platform.
- The capabilities of the platform to enrich company data with other (open) data (sets) from the platform could be further enhanced and would probably be another valuable USP.
- The entry page / landing page of the platform as well as the project page could be redesigned to increase its usability.

Of course, all data scientists are eagerly waiting for the advanced capabilities of the platform to be provided in the next version of the AEGIS prototype to be used for implementing the next version of their demonstrator.

4. AEGIS AUTOMOTIVE DEMONSTRATOR

4.1. Pilot overview and current status

The automotive demonstrator is developed according to three different scenarios, (1) broken road indicator, (2) safe driving indicator, and (3) regional driving safety risk estimator.

Within the first reporting pilot, version 1 of the automotive demonstrator (equalling scenario 1), broken road indicator has been implemented, providing insights into road conditions based on exploiting individual vehicle/driving data. In the following, a description of what has been achieved so far is provided. All data has been uploaded to the **Broken_Road_Indicator project** on the AEGIS platform. All files and datasets that are part of this demonstrator are prefixed with "**Demo1_**"

Data Acquisition	Data Analysis	Data Storage	Data Usage
Vehicle/driving data from multiple drivers covering multiple trips is manually uploaded to the platform. This data has been exported from a vehicle data logger before.	Vehicle/driving data is transformed in multiple steps. Thereby a series of datasets are created step by step. Finally, broken road events are identified in the transformed data. No extra functionality (e.g. Query Builder) is used.	Broken road events are stored in a separate dataset for the Visualiser. Also, the transformed data from every step is stored in separate datasets for further use.	All identified broken road (in all trips of all drivers) events are visualised on a map. Thereby the Visualiser functionality is used.

Figure 1: Broken road indicator (overview)

In a **first step** (after creating a project on the AEGIS platform), vehicle/driving data has been uploaded to the platform. Vehicle data is currently stored within a folder on one of VIF's file shares. The data includes several measurements (e.g. speed, rpm, vehicle rotation, vehicle acceleration, and vehicle position), which have been exported from the virtual vehicle data logger (a device connected to the OBD2 interface of a car and providing additional sensory) manually. The following visualisation shows that these measurements are currently stored in four different files, accelerations.csv, gyroscopes.csv, obdData.csv, and positions.csv.

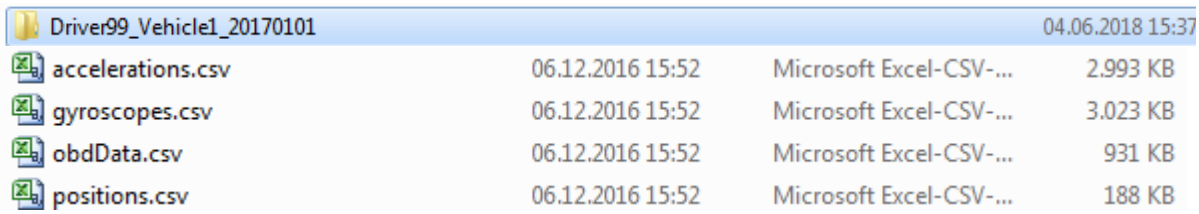


Figure 2: Vehicle measurements stored in a folder at VIF

These four CSV files exported from the data logger contain all vehicle measurements and have the following structure as shown in the visualisation below as the snapshot of each file shows.

accelerations.csv

```
"acceleration_id","trip_id","x_value","y_value","z_value","timestamp"
"1","1","0.00000","0.00000","0.00000","2017-02-14 12:44:24.058949"
"2","1","0.05078","-0.09766","0.83203","2017-02-14 12:44:24.071847"
"3","1","0.05078","-0.12500","0.89063","2017-02-14 12:44:24.084105"
"4","1","0.03906","-0.10156","0.89453","2017-02-14 12:44:24.095844"
"5","1","0.03516","-0.13281","0.88672","2017-02-14 12:44:24.107605"
```

gyroscopes.csv

```
"gyroscope_id","trip_id","x_value","y_value","z_value","timestamp"
"1","1","-0.48696","2.57391","1.39130","2017-02-14 12:44:24.058178"
"2","1","-0.83478","3.33913","1.04348","2017-02-14 12:44:24.070662"
"3","1","-0.06957","2.99130","1.18261","2017-02-14 12:44:24.082432"
"4","1","-0.06957","1.53043","1.11304","2017-02-14 12:44:24.094166"
"5","1","0.62609","1.32174","1.04348","2017-02-14 12:44:24.105902"
"6","1","0.69565","1.11304","0.97391","2017-02-14 12:44:24.117647"
```

obdData.csv

```
"obdData_id","trip_id","obdPid","data","timestamp"
"1","1","04","25","2017-02-14 12:44:24.053831"
"2","1","05","53","2017-02-14 12:44:24.071403"
"3","1","0B","108","2017-02-14 12:44:24.101636"
"4","1","0C","1717","2017-02-14 12:44:24.121312"
"5","1","0D","29","2017-02-14 12:44:24.151411"
"6","1","0F","12","2017-02-14 12:44:24.171507"
```

positions.csv

```
"pos_id","trip_id","latitude","longitude","altitude","timestamp"
"1","1","4703.5388","1527.7617","106.2","2017-02-14 12:44:24.334359"
"2","1","4703.5380","1527.7624","106.2","2017-02-14 12:44:24.534421"
"3","1","4703.5372","1527.7632","106.1","2017-02-14 12:44:24.734403"
"4","1","4703.5365","1527.7639","106.1","2017-02-14 12:44:24.934621"
"5","1","4703.5358","1527.7646","106.1","2017-02-14 12:44:25.134468"
```

Figure 3: Vehicle measurements stored in four different CSV files

The vehicle raw data (“accelerations.csv”, “positions.csv”, “obdData.csv”, “gyroscopes.csv”) is uploaded into a subfolder of the project “Demo1_rawData”. To ensure anonymization (as defined in the ethical concept of the demonstrator), the name of the subfolders containing the vehicle measurements has been done in the form “<DriverID>_<VehicleID>_<ExportDate>”. No driver names and no vehicle names are stored on the platform to ensure compliance with GDPR.

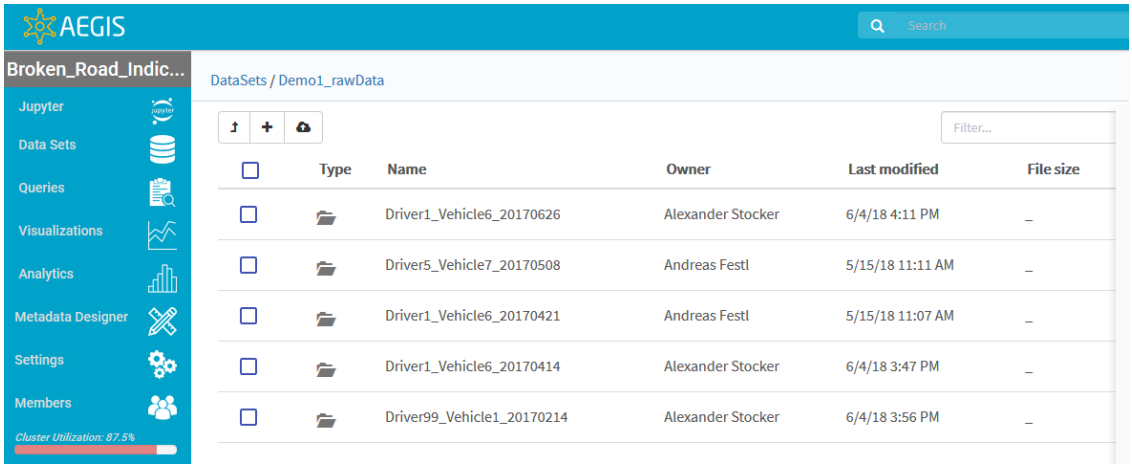


Figure 4: Demo1_rawData: Measurements uploaded to the AEGIS platform (1/2)

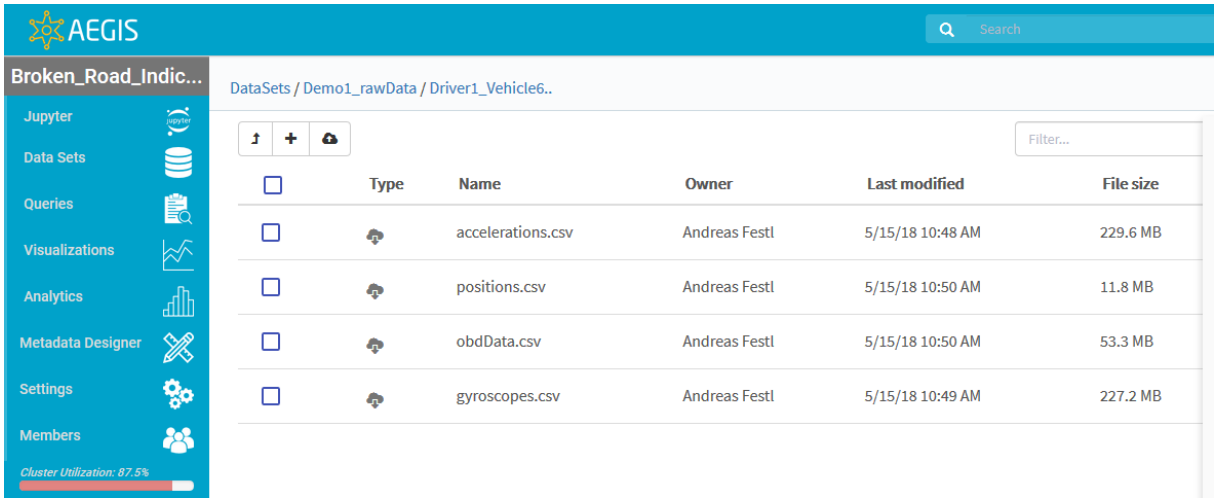


Figure 5: Demo1_rawData: Measurements uploaded to the AEGIS platform (2/2)

In a **second step** a series of data analysis tasks have been conducted. The first data transformation is concerned with opening the Zeppelin notebook "Demo1_MergeResampleTrips" and calling the function "mergeResampleRawData" on the newly uploaded data, which will join the raw data files and extract the single trips (this step is completely parallelized in spark). The second data transformation is concerned with resampling to a fixed time grid (this is done locally with pandas and numpy). Finally, the trips are written to the dataset "Demo1_TripData_raw", while the folder structure represents DriverID, VehicleID, and export date. No additional AEGIS functionality (e.g. Query Builder) has been used in data analysis.

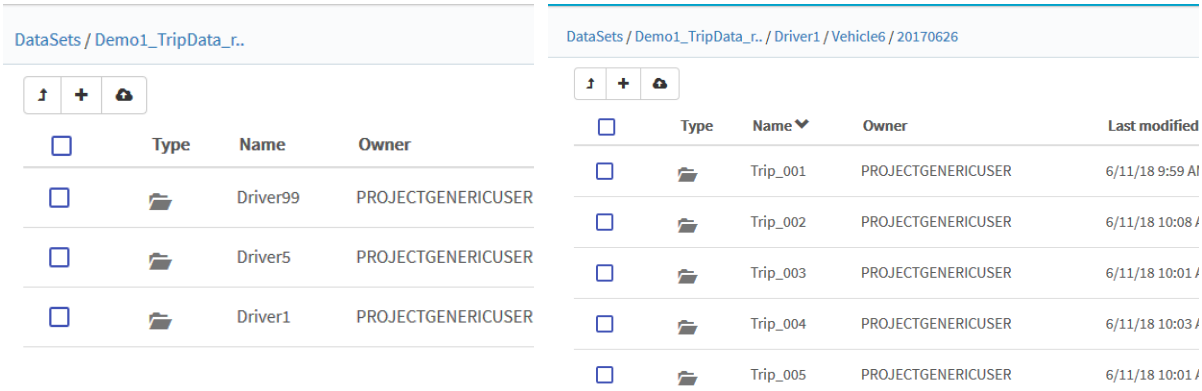


Figure 6: Demo1_TripData_raw: Vehicle drivers, vehicles, and trips

In a **third step** the notebook "Demo1_prepareTrips" must be opened and all cells have to be executed. This will load the trips from the step before (all trips are loaded/recalculated automatically). For each trip, the coordinate system of the sensor is aligned with the vehicle and the data is filtered and transformed where necessary (currently this runs locally in pandas). The results are then saved in dataset "Demo1_TripData_prepared".

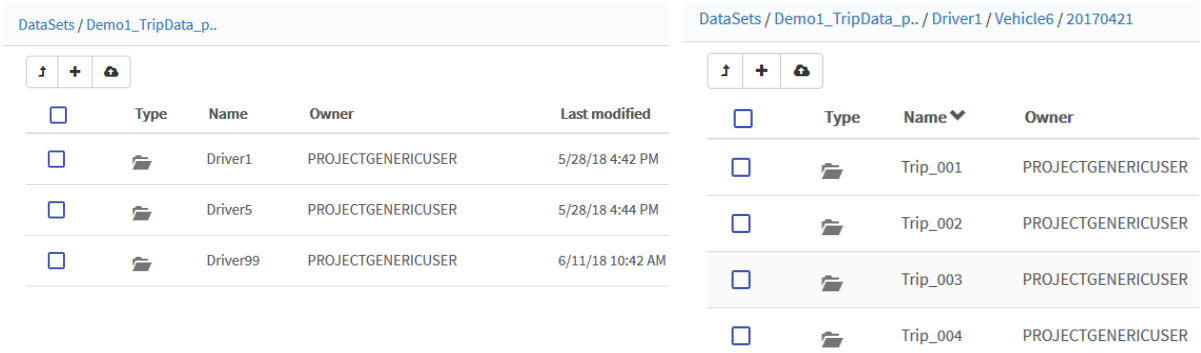


Figure 7: Demo1_TripData_prepared: Vehicle drivers, vehicles, and trips (finalized)

In a **fourth step the** notebook "Demo1_CalculateEvents" must be opened and all cells must be executed. This will load the trips from step 3 (all trips are loaded / recalculated automatically). For each trip the broken road events are calculated (currently this runs locally in pandas). The broken road events are then saved in the dataset "Demo1_BrokenRoadEvents" or "Demo1_BrokenRoadEventsCSV" (there is a row for each event, the only columns are latitude and longitude). Finally, a dataset "Demo1_BrokenRoadEventsCSVComplete" including the broken road events of all trips in a big file for the Visualiser has been created, too. This also includes besides latitude and longitude of an event, driver, vehicle, export date, and trip name.

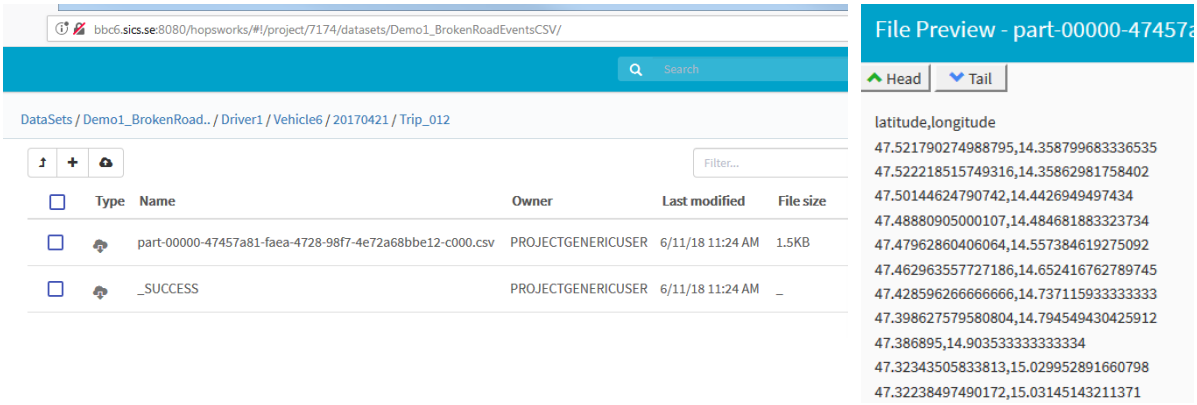


Figure 8: Demo1_BrokenRoadEvents: Broken road events of one trip

In a **fifth step**, the broken road events contained in the dataset "Demo1_BrokenRoadEventsCSVComplete" can be visualised on a geographic map using the Visualiser functionality in a Jupyter notebook. The following screenshots shows a map of Greater Graz area including all detected broken road events, while the numbers on the map indicate the number of broken road events identified in the area. Thereby the data scientist can click on one of the circles and further zoom in.

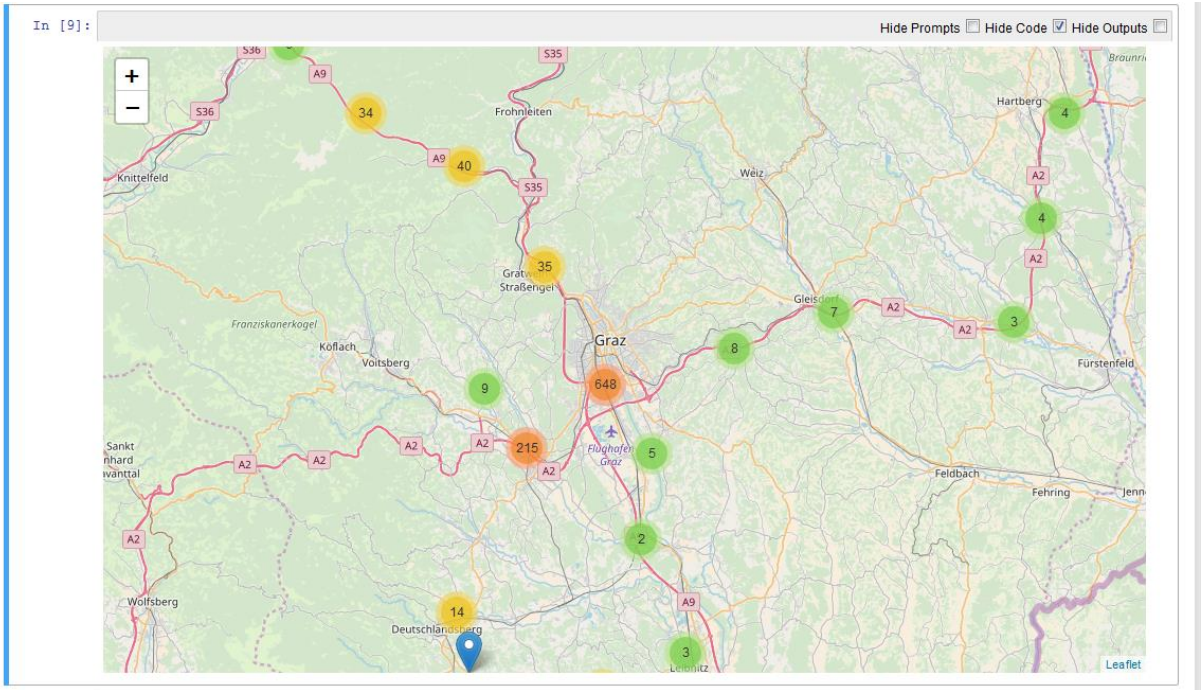


Figure 9: Visualising "Demo1_BrokenRoadEventsCSVComplete"

4.2. Scenario execution

The **first scenario** "broken road indicator" include executing five different scenarios, (1) create project, (2) upload vehicle/driving data (in bulk), (3) transform driving data and save results, (4) identify road damage and save results, and (5) provide visualisation of road damage.

4.2.1. Test Case 1 for "Create project"

Actors:	PSPS data scientist from VIF
Pre-conditions:	1. A data scientist wants to create a broken road identification service for smart city planners. 2. The service should be based on vehicle data and identify areas of road surface damage.
Post-conditions:	1. User account is created 2. Project is created
Normal Flow	1. The data scientist opens the AEGIS platform in the Web browser and creates a user account. 2. He/she then logs into the platform and creates a new project named "broken road indicator".
Pass Metrics	1. The data scientist can create a user account 2. The data scientist can create a new project with the desired name

Fail Metrics	1. No user or no project is/are created.
Notes and Issues:	-
Execution Results	Both the creation of users accounts as well as the creation of a project on the platform went smoothly. A project named "Broken_Road_Indicator" was created on the platform.

4.2.2. Test Case 2 for "Upload driving data (in bulk)"

Actors:	PSPS data scientist from VIF
Pre-conditions:	1. User must be registered with email address and password. 2. Project must have been created. 3. User must have logged in.
Post-conditions:	1. Dataset is uploaded.
Normal Flow	1. The data scientist creates a new dataset named "raw vehicle data". 2. He/she selects the "upload data" functionality and uploads a bulk of vehicle data (csv-files) from his/her local computer to the platform.
Pass Metrics	1. The data scientist can upload a new dataset on his/her project
Fail Metrics	1. The data scientist is not able to upload a new dataset on the platform 2. The uploaded dataset is not available on the project
Notes and Issues:	Each folder contains the data from one logging period stored in four csv files, one for each sensor (OBD.csv, acceleration.csv, gyro.csv, gps.csv).
Execution Results	Vehicle measurements from three different drivers covering multiple trips were uploaded to the created project. Upload in bulk worked smoothly. The data is available in the dataset "Demo1_TripData_raw".

4.2.3. Test Case 3 for "Transform driving data and save results"

Actors:	PSPS data scientist from VIF
Pre-conditions:	1. Required "raw vehicle data "dataset is available on the platform

Post-conditions:	1. A new dataset named "trip data" is created on the platform. It includes one file per trip with a rising number (tripXY.csv).
Normal Flow	The data scientist transforms the data (either by executing R code or by using the Query Builder). In either case he/she performs the following steps. 1. Rename columns in multiple files 2. Transform OBD data from "tall" to "wide" format (see: https://en.wikipedia.org/wiki/Wide_and_narrow_data): Each obd_Pid is replaced by the associated signal's name and is then promoted to a new column. The value in that column is the value of "data". In each row there is only one column != NULL. 3. Join the 4 tables (FULL OUTER JOIN) 4. Split the result into the single trips as identified by the column trip_id. 5. Interpolate the measurements onto a regularly spaced time grid (e.g. 10Hz). 6. Rotation of acceleration and gyroscope data: The data needs to be aligned with the coordinate system of the vehicle. (Example: acc_x should always point in the same direction of the vehicle, regardless of the sensor position) 7. Save the transformed data one file per trip with a rising number (tripXY.csv) in a new dataset (prepared trip data).
Pass Metrics	1. The dataset has been successfully transformed and is available in a new dataset for the project.
Fail Metrics	1. The transformation has not been successful and an error message is provided.
Notes and Issues:	-
Execution Results	To execute this test case the R code developed offline has been transferred to Python/Spark. Additionally, a functionality for parallelizing the trip analysis has been implemented. The query builder was not available to the data scientist and was therefore not used. All transformation steps went fine. A series of sub datasets have been created to store the data after each data analysis step. The raw data from before is used as input. The Zeppelin notebook "Demo1_MergeResampleTrips" is the starting point. It joins the raw data files and extract the single trips (a step that is completely parallelized in spark). Then it resamples the trip to a fixed time grid (this is done locally with pandas and numpy). Finally, the trips will be written to the dataset "Demo1_TripData_raw".

	The Zeppelin notebook "Demo1_prepareTrips" continues the analysis. It loads the trips from the previous steps. For each trip, the coordinate system of the sensor is aligned with the vehicle and the data is filtered and transformed where necessary (currently this runs locally in pandas). The results are saved in dataset "Demo1_TripData_prepared".

4.2.4. Test Case 4 for "Identify road damage and save results"

Actors:	PSPS data scientist from VIF
Pre-conditions:	1. Transformed vehicle data is available on the platform
Post-conditions:	1. Road damage data is computed. 2. A new dataset named "road damage data" is created on the platform.
Normal Flow	1. The data scientist queries all trip.csv files generated in the previous step for trips in a certain time range by using R coder or the Query Builder (trip-selector). 2. The data scientist executes R-code on the platform using the result of the query as input to detect events related to road damage (e.g. potholes). 3. The detected events are saved on the platform in a new dataset (road damage data).
Pass Metrics	2. Road damage data dataset is available for the project.
Fail Metrics	3. The event detection has not been successful, and an error message is provided.
Notes and Issues:	-
Execution Results	The note "Demo1_CalculateEvents" loads the trips from the step before and calculates the broken road events for each trip (currently this runs locally in pandas). The events are saved in dataset "Demo1_BrokenRoadEvents" (there is a row for each event, the only columns are latitude and longitude) and in "Demo1_BrokenRoadEvents" (one parquet file per trip), "Demo1_BrokenRoadEventsCSV" (one CSV file per trip), and "Demo1_BrokenRoadEventsCSVComplete" (all events from all trips are stored in a single CSV file).

4.2.5. Test Case 5 for "Provide visualisation of road damage"

Actors:	PSPS data scientist from VIF
Pre-conditions:	1. "Road damage data" dataset is available.
Post-conditions:	2. Road damage events are visualised on a map.
Normal Flow	1. The data scientist selects the visualisation tool from the platform and the road damage dataset generated previously to visualise the detected events on a map. 2. The visualisation is intended for a smart city planner, who wants to access it without having a user account on the platform. 3. The visualisation is made available as a web page for externals.
Pass Metrics	1. A proper visualisation is shown to the user.
Fail Metrics	2. The visualisation is not computed, and an error message is provided to the user.
Notes and Issues:	-
Execution Results	The Visualiser capabilities of the platform are used to visualise the broken road events detected in the step before on a geographic map. The visualisation worked fine as shown in Figure 10.

In [5]: Hide Prompts ☐ Hide Code ☑ Hide Outputs ☐

	latitude	longitude	driver	vehicle	exportdate	trip
0	47.052606	15.457662	Driver1	Vehicle6	20170414	Trip_002
1	47.017372	15.479394	Driver1	Vehicle6	20170414	Trip_002
2	47.015889	15.473054	Driver1	Vehicle6	20170414	Trip_002
3	46.999163	15.411247	Driver1	Vehicle6	20170414	Trip_002
4	46.996718	15.407297	Driver1	Vehicle6	20170414	Trip_002
5	46.981172	15.394334	Driver1	Vehicle6	20170414	Trip_002
6	46.973353	15.337209	Driver1	Vehicle6	20170414	Trip_003
7	46.974417	15.335439	Driver1	Vehicle6	20170414	Trip_003
8	46.975667	15.330349	Driver1	Vehicle6	20170414	Trip_003
9	46.990004	15.400119	Driver1	Vehicle6	20170414	Trip_004
10	46.991717	15.401617	Driver1	Vehicle6	20170414	Trip_004
11	47.002718	15.419681	Driver1	Vehicle6	20170414	Trip_004
12	47.006965	15.432834	Driver1	Vehicle6	20170414	Trip_004
13	47.007404	15.434250	Driver1	Vehicle6	20170414	Trip_004
14	47.010193	15.443203	Driver1	Vehicle6	20170414	Trip_004

In [6]: Hide Prompts ☐ Hide Code ☑ Hide Outputs ☐

Select a visualization type

- ○ Scatter Plot
- ○ Pie Chart
- ○ Bar Chart
- ○ Line Chart
- ○ Box Plot
- ○ Histogram
- ○ Time Series
- ○ Heatmap
- ○ Bubble Chart
- ◉ Map

Apply

In [8]: Hide Prompts ☐ Hide Code ☑ Hide Outputs ☐

Latitude Variable: latitude
Longitude Variable: longitude
Display Type: Markers
Position: 1
Visualize

In [9]: Hide Prompts ☐ Hide Code ☑ Hide Outputs ☐

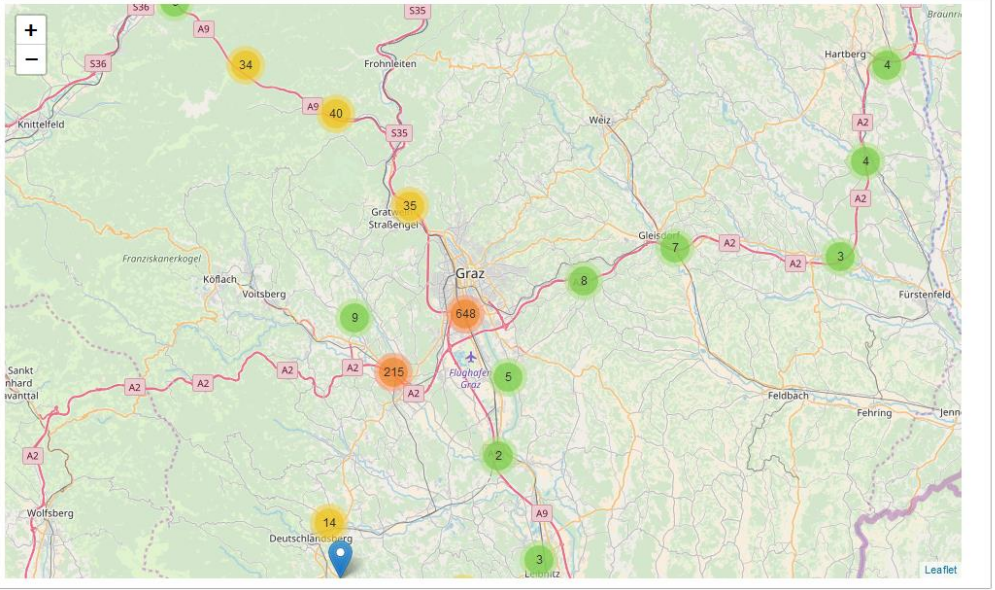


Figure 10: Visualisation of road damage

4.3. Demonstrator Evaluation

4.3.1. Quantitative Evaluation

The demonstrator-specific quantitative evaluation action focuses on the completeness on the test cases as well as on some demonstrator-related KPIs.

Sub-characteristics	KPIs	Calculation Type	Mandatory / Optional	Means to Verify	Value
Broken road indicator functional completeness	Number of successfully completed Test Cases	[Successfully Completed test cases] / [No of cases] * 100%	M	Calculation (0-100%)	100%
Include multiple trips in driving data	Number of different Trips	Sum	O	Counting numbers	120
Include multiple drivers in driving data	Number of different drivers	Sum	O	Counting numbers	3
Demonstrate event detection	Number of detected events in total	Sum	O	Counting numbers	1375
Demonstrate road damage detection	Number of detected road damage	Sum	O	Counting numbers	1375

Table 2: Automotive demonstrator quantitative evaluation

4.4. Challenges and recommendations

For the implementation of the first demonstrator, the first prototype of the AEGIS platform has been used which was available at M12 (V1). Hence, only basic functionality of the AEGIS platform could be used. Therefore, all the data analysis and transformation steps had to be done manually using notebooks. The only enhanced functionality of the platform used so far was the Visualiser. During the implementation of the demonstrator, a few challenges have arisen, which could be easily solved through the strong support of KTH (AEGIS Platform in general) and UBITECH (Visualiser component).

Recommendations concerning the perspective of the automotive demonstrator are included in the section on the qualitative platform evaluation. The data scientists are looking forward to using the extended capabilities of the second AEGIS platform prototype to implement the second version of their demonstrator.

5. AEGIS SMART HOME AND ASSISTED LIVING DEMONSTRATOR

5.1. Pilot overview and current status

The objective of the Smart Home and Assisted Living (SHAL) Demonstrator is to illustrate the benefits of a big data platform through the implementation and offering of a services bundle towards advanced holistic monitoring and assisted living management, which aims to aid the everyday wellbeing of people belonging to vulnerable groups. In summary, the case study is the following: a social care service provider, for example a care centre for elderly individuals or a nursing home, desires to exploit big data-driven insights, in order to provide added value services to vulnerable individuals. The services pertain proactive and reactive security and protection through smart notifications and personalised recommendations, as well as indoor comfort and quality preservation. Proactivity and reactivity of the aspired services aim at prolonging self-sufficiency and independence of the at-risk individuals, boosting safety, and facilitating informed decision making, either by the individuals themselves, or by their (in)formal carers. The demonstrator is developed in Athens by Hypertech, UBITECH and Suite5, all Information and Technology (IT) companies which adopt the AEGIS roles of the service developers and data scientists.

Overall, the SHAL demonstrator will implement two main services that can be offered by a care service provider to at-risk individuals and/or their (in)formal carers:

a) Monitoring and analysis of an individual's well-being conditions, physical activity, positioning and wearable information and external environment data (e.g. weather, crime, news, social media), towards provision of a service for personalised notification and recommendation system for at-risk individuals, including notifications for carers.
b) Additional service pertaining monitoring and analysis of weather, indoor environmental conditions, energy and operational device data towards the provision of a smart home application, which can be offered by care providers to at-risk people for increased indoor comfort and welfare.

These services were broken down to six implementation scenarios, three for each service pertaining the early, medium and advanced version of the demonstrator, following the time plan for the circulation of versions 1, 2 and 3 of the AEGIS platform. In this deliverable, we report the evaluation of the scenarios of the early-version demonstrator through the predefined test cases. Table 3 provides a summary of the two early-version scenarios.

ID	Scenario	Functionalities	Demonstrator Version
1	(At-risk) Individuals data fetching, processing and classification	Registration of personal devices (for data collection and aggregation of activity and health data), registration of external data streams (events, weather, etc.), data processing, creation of personas based on data classification, definition of generic rules per persona, definition of outlier detection	Early

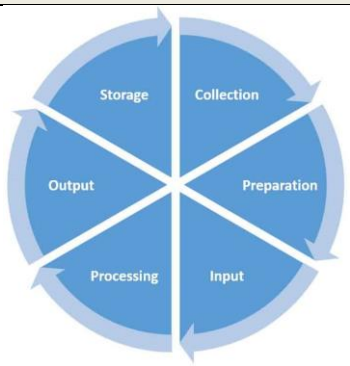
		algorithms, generation of simple alerts to (at-risk) individuals.	
2	Smart home data monitoring and processing	Registration of smart home data streams, Smart Home Data Monitoring, Pre-processing and normalization.	Early

Table 3: Smart Home and Assisted Living demonstrator scenarios

Regarding the current status of the demonstrator, all required steps for the early version demonstrator have been successfully implemented. It is worth mentioning that during the early version demonstrator, activities associated with all steps in the data life cycle of the demonstrator (see Table 4) have been performed to some extent.

Following the evaluation of this stage, development will continue towards the realization of the subsequent scenarios 3: Notifications and alerts for (at-risk) individuals, and 4: Smart home comfort profiling and notifications. Further details on the implementation scenarios and associated test cases can be found on Deliverable 5.2.

In the following subsections, we first present the work performed within the scope of each scenario, in order to implement the related test cases, and provide the results of this process. Subsequently, the quantitative evaluation is performed, followed by a summary of encountered challenges and recommendations for improvement of the platform.

Data Life Cycle of the SHAL demonstrator	
	
Collection	The data sources used in the demonstrator are the following: Motion data, Luminance, Indoor Air Quality, Indoor temperature and humidity, Control actions over lighting and HVAC, HVAC Energy Consumption, Wearable Sensor Data (Fitbit and/or Apple watch), Smartphone Sensors (Accelerometer/GPS), Expert Rules Data.
Preparation	The collected data are to be subjected to an online homogenization algorithm and then to an offline pre-processing workflow, with the primary aim of data to become initially homogenous (due to the possible inclusion of multiple sensors from different vendors),

	and then to preserve privacy through anonymization and dissociation with any sensitive information.
Input	The pre-processed data are to be uploaded to the AEGIS platform for further processing/cleaning that would be necessary in order to prepare data (in terms of quality assessment, outlier detection, fill missing values, visualisation) for consequent analysis.
Processing	The processing step entails the training of regression and clustering models, so as to develop the different persona classifications that will be used by the demonstrator.
Output	The outcome of the processing results to clusters of patients, which, alongside with the employment of the third-party event detection service creates groups of individuals which are candidates for receiving notifications.
Storage	The anonymised datasets are stored as private datasets in the AEGIS platform for as long as it is needed in order to perform the requested actions and can be accessed only by the SHAL demonstrator data scientists. Additionally, the data are securely stored in the local server of the demonstrator.

Table 4: Data Life Cycle of the SHAL demonstrator.

5.2. Scenario execution

5.2.1. Scenario 1 – (At-risk) Individuals data fetching, processing and classification

Scenario 1 includes the steps that have to be performed in order to onboard an individual and a carer to the system, which results in the acquisition of personal data that are used to classify (at-risk) individuals into specific profiles ("personas"). The accumulated data is then used (in an anonymous manner) by the CSPs, in conjunction with external data, for further analysis and for issuing simple notifications. In parallel, cares are registered to the platform and are connected to the (at-risk) individuals.

5.2.1.1. Test Case 1.1 – (At-risk) Individuals Profile Building and Devices Registration

This test case focuses on the registration of individual users (at-risk individuals) to the platform and to the registration of their devices for (semi)-automatic data retrieval.

In this context, the necessary registration mechanisms were implemented in the SHAL Web App, while a simple profile page has been created where each user would be able to specify some personal information, as well as some information regarding his health and his conditions, to allow the system to perform simple queries and categorisation functions.

As a next step, the individual is offered the option to register his devices to the platform, to populate his profile with data from diverse sources. At this point, the early version demonstrator offers the ability to link data coming from Apple Devices through Apple Healthkit, data coming from FitBit as well as Smarthome data coming from Hypertech's UELO implementation.

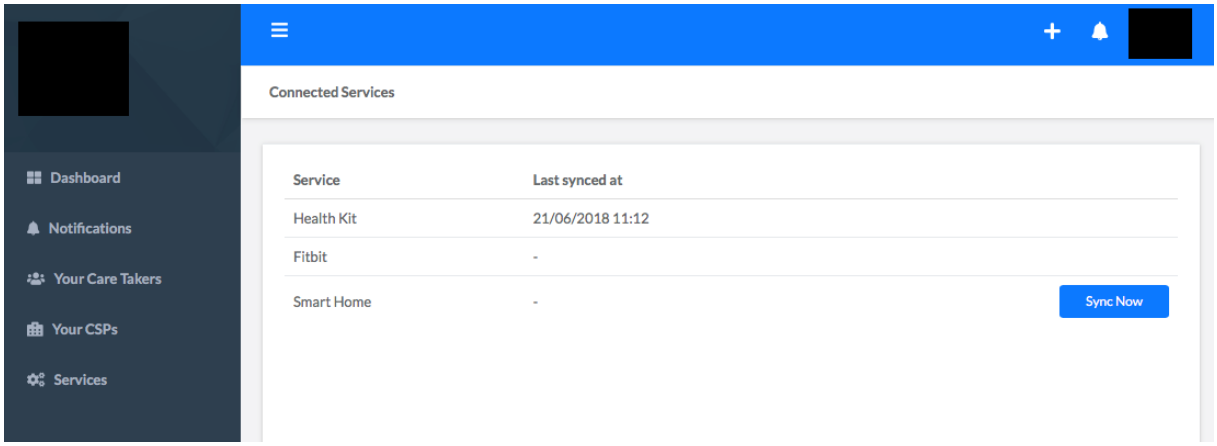


Figure 11: At-risk individuals device registration and sync view

In the case of the iOS App, data from Apple Healthkit is automatically retrieved from the user's device (upon user acceptance) and then the data is transferred to the SHAL WebApp, where it is transformed at the right format. In the case of the Android App, data from FitBit is automatically retrieved and as with the iOS App, the data is appropriately transformed and transferred to the WebApp

As far as it regards the registration of the smart home sensors to the platform, the individual is able to provide to the WebApp the id of each sensor he/she possesses, and the WebApp is responsible for periodically retrieving data from the smart home gateway.

Actors:	(At risk) individual
Pre-conditions:	Mobile app
Post-conditions:	1. Private (at-risk) individual profiles with registered devices attached
Normal Flow	1. The individual registers to the platform as a "at-risk individual" 2. The individual fills in his/her profile, including demographic data 3. The individual registers his/her devices/data streams to the web app 4. The individual provides his/her consent for the system to use his/her data in an anonymous manner and to receive notifications
Pass Metrics	1. (At-risk) individuals' profiles 2. Activity data stored periodically for each individual
Fail Metrics	1. No evidence of updated activity data
Notes and Issues:	-

Execution Results	The necessary infrastructure for allowing registration of individuals at risk was implemented and the profile building mechanisms was developed. Device registration has been implemented, focusing on 3 different devices, Apple Healthkit, FitBit and UELO. The test case was successfully completed.

5.2.1.2. Test Case 1.2 – Carers Profile Building

This test case focuses on the registration of care takers to the platform. In this context, the necessary registration mechanisms were implemented in the SHAL Web App, while a simple profile page has been created where each care taker would be able to specify some personal information.

Actors:	Carer
Pre-conditions:	N/A
Post-conditions:	1. Carers profiles
Normal Flow	1. A Carer registers to the platform as a "carer" 2. The carer fills in his/her profile, including demographic data
Pass Metrics	1. Carers profiles
Fail Metrics	1. No Carers profiles
Notes and Issues:	-
Execution Results	The necessary infrastructure for allowing registration of caretakers and their profile building mechanisms was developed. The test case was successfully completed.

5.2.1.3. Test Case 1.3 – Linking (at-risk) Individuals with Carers

Individuals are provided with the feature to see care takers who are available in the platform and link these care takers to their profiles. Furthermore, individuals are able to revoke links with care takers.

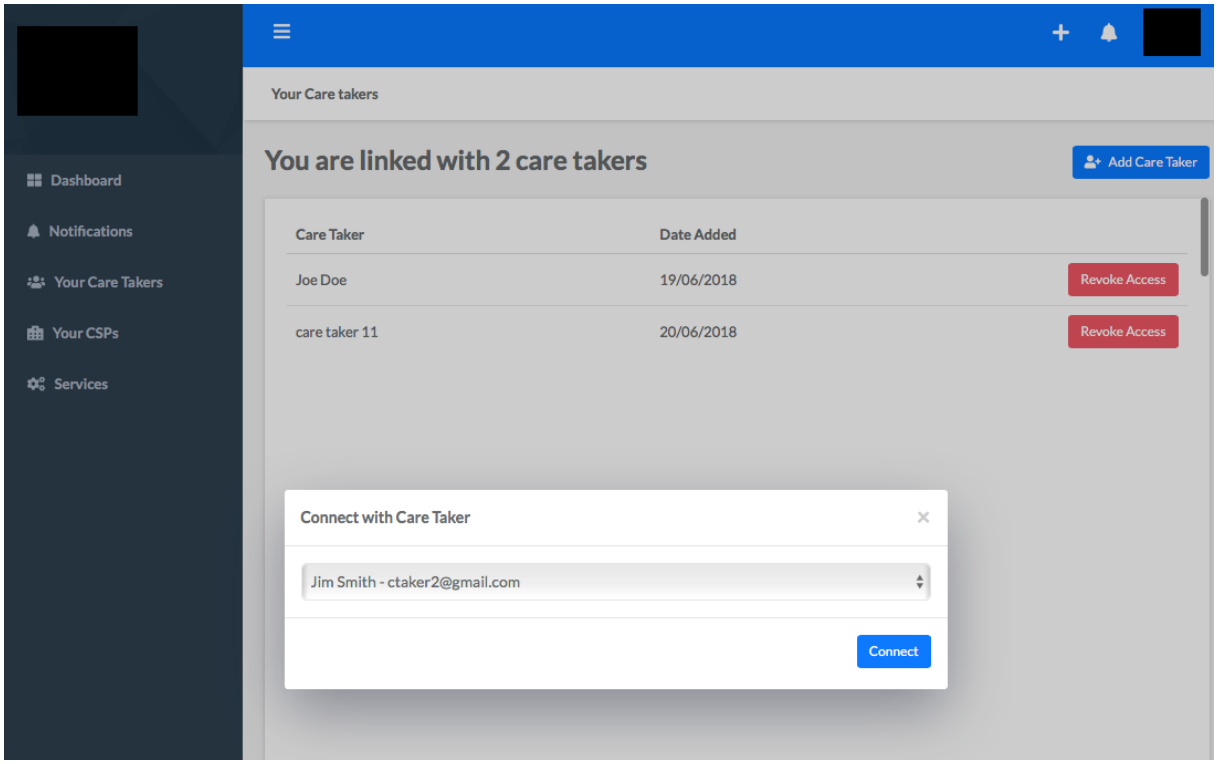


Figure 12: Linking Individuals with Care Taker

Actors:	(At-risk) Individual, Carer
Pre-conditions:	N/A
Post-conditions:	1. (At-risk) individual profiles 2. Carers profiles
Normal Flow	1. The individual searchers for a carer profile in the platform 2. The individual selects a carer profile and sends a connection request 3. The carer gets a notification in the mobile app for a new (at-risk) individual connection request 4. The carer acknowledges/rejects the connection request 5. The individual is notified about the acceptance/rejection of his/her request.
Pass Metrics	1. Connection link established between (at-risk) individuals with carers
Fail Metrics	1. No connection links shown between individuals and carers
Notes and Issues:	No personal information (other than name/contact details) are shared by the (at-risk) individual to the carer

Execution Results	The necessary features have been implemented for the individual to search for and link his profile with a care taker, as well as to revoke the link in case he wants to. The test case has been successfully completed.

5.2.1.4. Test Case 1.4– CSP services-relevant External Data Sources registration

This test case refers to data pulling from the AEGIS platform, capitalising on third party data sources (static or streaming) which are registered in the AEGIS platform and that utilise the main AEGIS platform as a data storage and brokerage service. The test case for the demonstrator will be tested by pulling data which is relevant to weather conditions, as well as data that are generated by the event detection module of Suite5.

Actors:	CSP
Pre-conditions:	N/A
Post-conditions:	1. Data pulled form external sources profiles
Normal Flow	1. The CSP selects the type of services to be registered to the web app out of a predefined list. 2. The CSP provides some descriptions regarding the service to be registered 3. The CPS registers the external service via its API in the platform
Pass Metrics	1. Data Samples visible in the web app
Fail Metrics	1. No data samples visible as preview
Notes and Issues:	Assumption that third party data source come from AEGIS, in the correct format needed for the rule-based system of the web app
Execution Results	The test case was implemented by successfully linking the demonstrator WebApp to these third parties directly (weather data and event detection output), as integration of those data sources is not yet available on the AEGIS platform. Registration of those sources to the WebApp and data pulling from the AEGIS directly will be available at the next version of the demonstrator, once these third-party sources are available through AEGIS.

5.2.1.5. Test Case 1.5 – "Personas" Building and Clustering

This test case focuses on the cluster analysis of the at-risk individuals into specific profiles ("Personas"). More specifically, the profile data of the at-risk individuals are anonymised by removing any sensitive data, thus ensuring that the individual records or subjects of the data cannot be re-identified and then exported from the SHAL platform. The assigned CSP data scientist creates a new project in the AEGIS platform called "Smart_Home_Demonstrator Personas_Building" and uploads the anonymised profile data in the form of a csv file as a private dataset of the project.

In order to perform the cluster analysis of the anonymised (at-risk) individuals' profiles, the Algorithm Execution Container was utilised by the data scientist. Initially some pre-processing and data preparation is conducted. In the next step, Principal Component Analysis (PCA) is performed for gaining visual insights of meaningful clusters in the data. Afterwards, k-means clustering analysis is performed with different values for the number of clusters in order to find the optimal number of clusters. Once the optimal number is found the results are visualised focusing on the characteristics of the clusters. Finally, the results of the analysis indicating the "Personas" as a set of clusters and the members of each cluster are saved as a new dataset (containing only the cluster ids and the ids of the patients corresponding to each cluster) in the form of a csv file, called "personas.csv". The assigned CSP data scientist imports the newly created "Personas" manually into the SHAL Web app.

Actors:	CSP
Pre-conditions:	1. Existing profiles of (at-risk) individuals
Post-conditions:	1. Clusters of "Personas" including patients
Normal Flow	1. The CSP triggers the anonymization services and retrieves pseudo-anonymised profiles of (at-risk) individuals. 2. The CSP inserts the anonymised profiles in the AEGIS notebook and performs a clustering analysis. 3. The outputs of the analysis are imported into the web app (saved as one set of clusters)
Pass Metrics	1. Clusters available on the web app
Fail Metrics	1. No anonymised data retrieved 2. No clusters generated on the notebook 3. No clusters stored on the web app
Notes and Issues:	-

Execution Results	The test was successfully executed by implementing the various steps described in the test case. At first, the anonymised profile data are exported from the SHAL platform utilising the AEGIS Anonymisation Tool. Then, a new project is created in the AEGIS platform and the exported data set is imported to the project. The clustering analysis was performed using the Algorithm Execution container of AEGIS. Once the analysis is conducted, the results containing the "Personas" and the members of the "Personas" are imported manually in the SHAL platform.

5.2.1.6. Test Case 1.6 – Identification of Critical External Conditions/Events

As a third-party data source, events that are of interest for the CSPs and relate to their patients have to be identified so that the CSP Analyst can decide on issuing notifications to his monitored individuals. The event detection toolkit has to be customised setting the keywords that are required for identifying an event, and once it is running, it is able to capture and display events that are relevant to the input provided by the CSP analyst.

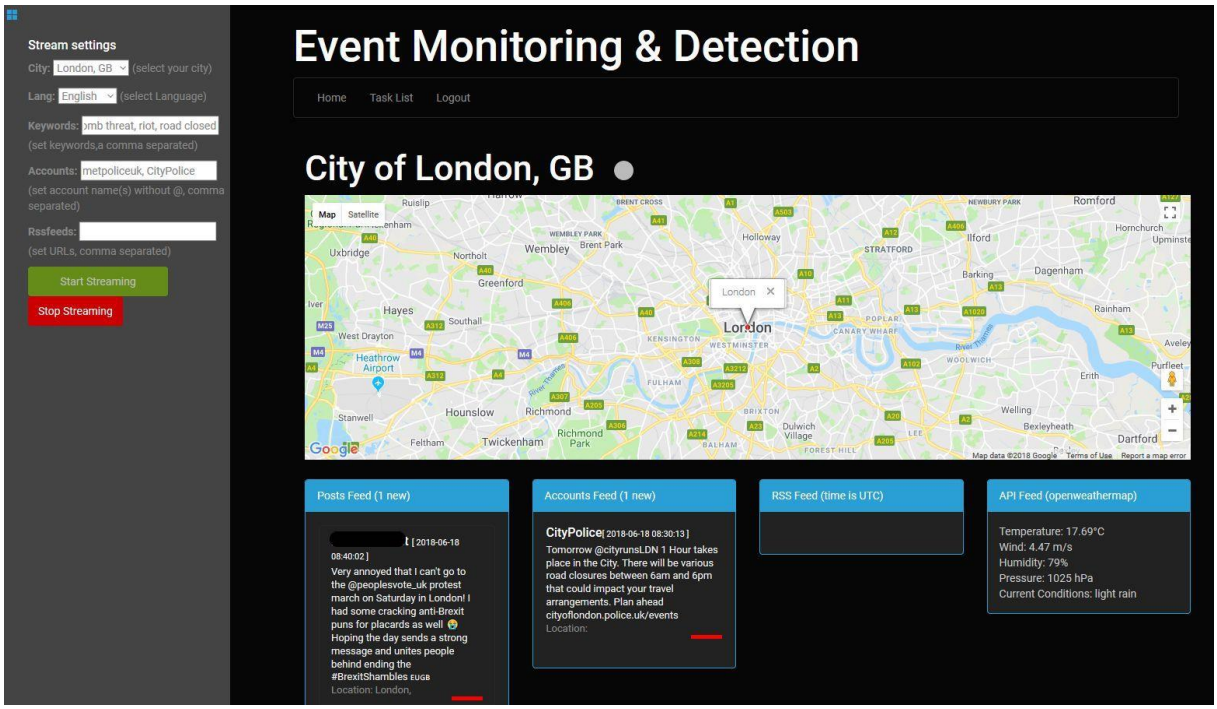


Figure 13: Event Detection Output

Actors:	CSPs
Pre-conditions:	Externals Services Registered to the platform
Post-conditions:	1. List of critical conditions/events

Normal Flow	1. The CSP sets certain thresholds on the external data sources registered on the web app 2. The CSP is notified about a critical external condition/event
Pass Metrics	1. Critical conditions/events identified
Fail Metrics	1. No existence of critical events
Notes and Issues:	-
Execution Results	Test case successfully executed. The event detection module has been customised to identify regional events that are relevant to the demonstrator scenario (keywords that are related to Heat, Humidity and CO2 have been utilised)

5.2.1.7. Test Case 1.7 – Simple Notification Issuing

A notification mechanism reaching both individuals and also care-takers (not included in the original test case) has been implemented, using FireBase[3] which allows the real-time provision of notification to users of mobile devices.

Notifications are issued by CSP analysts and are sent to person groups, where the CSP analyst only knows the user-id of people belonging to a persona, and not their real personal data. Once a notification is sent, a copy of it is sent also to the care taker who is linked to an individual.

[3] https://firebase.google.com/

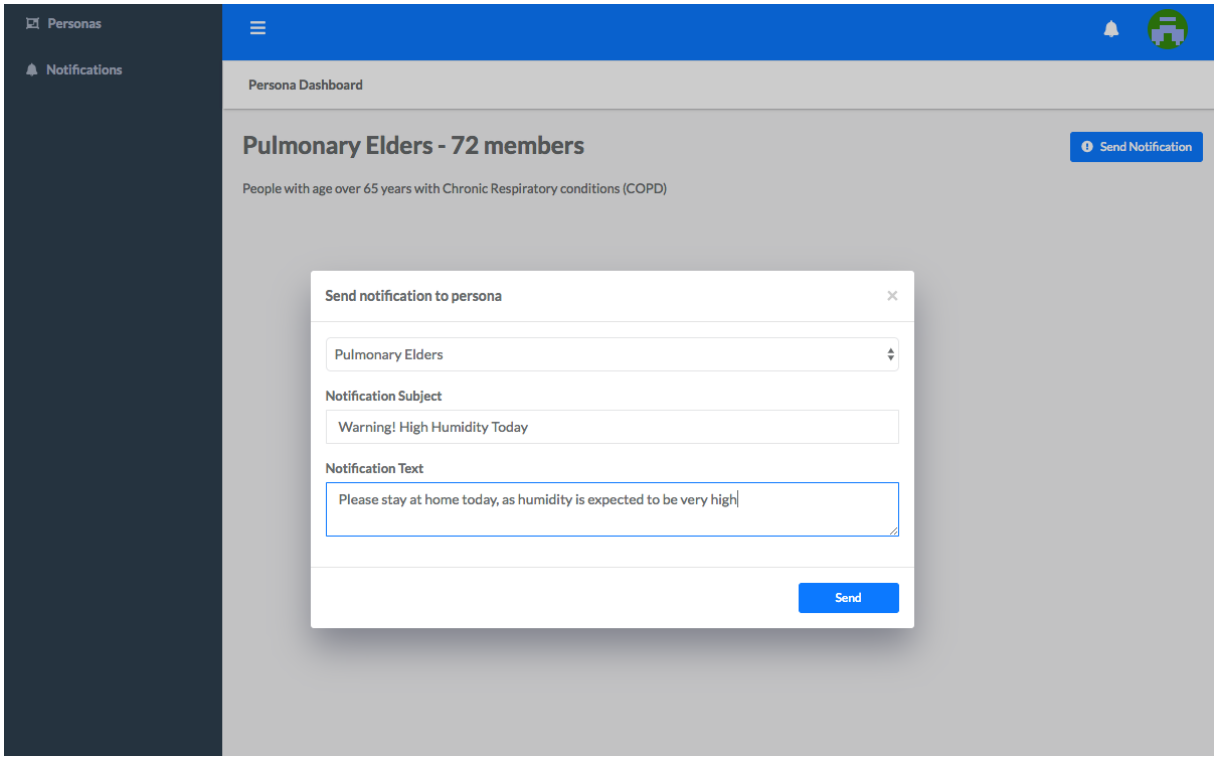


Figure 14: Notification Issuing

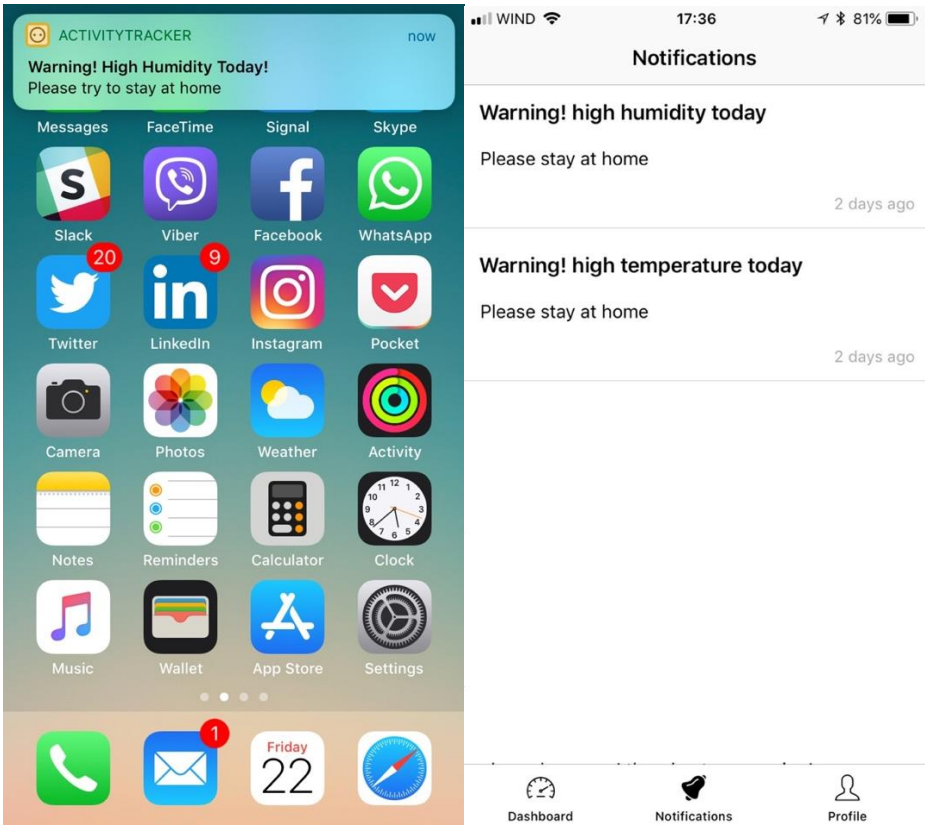


Figure 15: Notification Screen on iOS

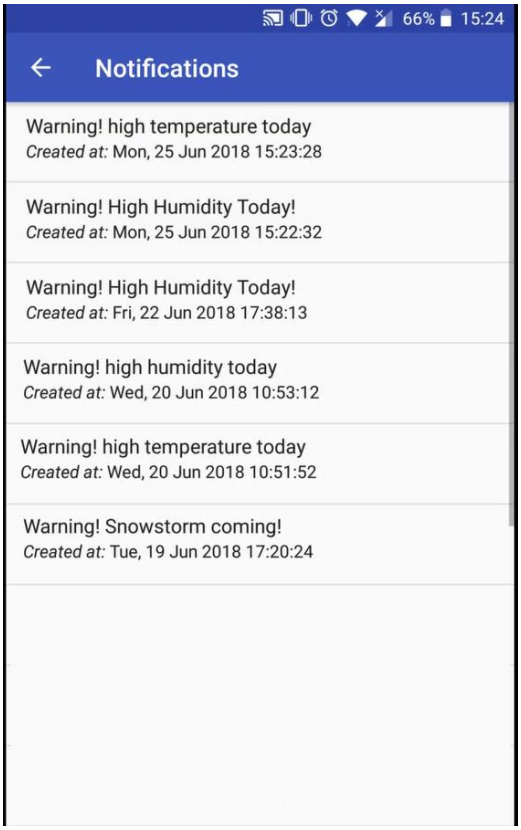


Figure 16: Notification Screen on Android

Notifications are also stored in the database so that they are available to CSPs, care takers and at-risk individuals.

Actors:	CSP
Pre-conditions:	1. (at-risk) individuals' profiles
Post-conditions:	1. Notification at the side of the (at-risk) individuals 2. Notification at the side of the carers
Normal Flow	3. The CSP selects a persona group of (at-risk) individuals 4. The CSP inputs the notification to be sent to the persona 5. The notification is sent to the persona
Pass Metrics	1. Notifications at the side of the (at-risk) individuals
Fail Metrics	1. No notifications pushed
Notes and Issues:	-

Execution Results	The notification service has been implemented successfully and CSPs are able to send notifications to at-risk individuals. The feature has been extended also to care takers looking after individuals. The test case has been successfully implemented.

5.2.2. Scenario 2 - Smart home data monitoring and processing

Scenario 2 constitutes the early version demonstrator with relation to the added value service of smart home automation. The scenario includes the following user stories and associated test cases, from the perspectives of the CSP and at-risk individuals respectively.

- CSP: The first step towards the implementation of the smart home offering, pertains the establishment, by the data scientist working for the CSP, of data flows regarding the at-risk individual's indoor conditions, and associated external weather measurements, as well as the required pre-processing and normalization that will allow, in a subsequent step, to train the profiling mechanism and estimate the personal preferences of the individuals.
- At-risk individual: The individual, after registration to the SHAL service, is supplied with a mobile application. The Smart Home monitoring UI allows real-time information on temperature, humidity, VOC concentration and HVAC status to be visualised for informative reasons.

In the following we give the implementation and evaluation details for the scenario 2 test cases.

5.2.2.1. Test Case 2.1 – Establishment of smart home monitoring infrastructure and data collection

Test case 2.1 pertained the establishment of the necessary hardware and software infrastructure for the collection of the defined smart home data streams. To achieve this target, a number of sensing and actuation instruments was installed.

In particular, a proprietary multi-sensor and gateway combo device, developed and manufactured by Hypertech was installed. The combo device is equipped with luminance, temperature, humidity and VOC sensors, as well as the gateway controller, which accumulates the signals and transmits them to a server through a rest interface, using Wi-Fi and TCP/IP communication protocols. The combo device was coupled two other actuation instruments. First, an air condition actuator module, which allows monitoring and control of the HVAC device, specifically: ON/OFF operational status, "heat", "cool", "auto", "dry", "fan" operational mode, and set point of the HVAC unit Second, dimmable ballasts with wireless controllers for measuring and setting the dimming level on lighting devices (For the purposes of the early version demonstrator only monitoring of the operational details was required). The multisensory/gateway device can be seen in Figure 17, while the HVAC and light actuators are shown in Figure 18 and Figure 19 respectively.

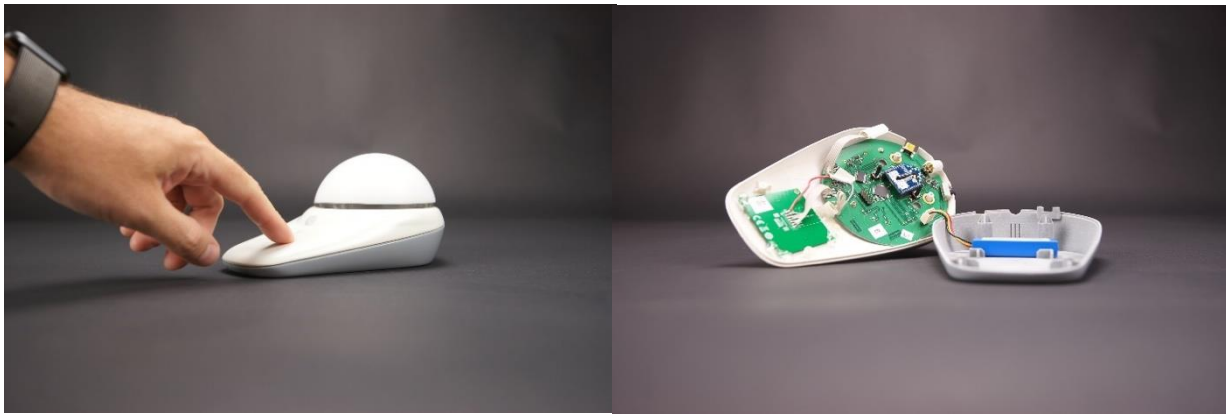


Figure 17: Hypertech's multisensory/gateway combo used in the SHAL demonstrator.

Figure 18: HVAC Actuator used in the SHAL demonstrator.

Figure 19: Dimmable Ballast & Actuator used in the SHAL demonstrator.

The monitored data are sent directly to the SHAL backbone server through a secure Rest interface. No personal data are transmitted, apart from the asset id, whose details were provided by the user upon successful registration to the demonstrator's app.

Ideally, the envisioned final solution will allow for automatic uploading of this data to the AEGIS platform for processing. Nevertheless, since such functionality is not yet well defined, the data were accumulated into a batch of csv files and uploaded to the AEGIS platform manually. External weather data for the same time period were also acquired manually. In order to upload the data, the assigned data scientist logged in to the AEGIS platform and created the project "Smart_Home_Early_Evaluation_Test_Project". Within the project, a dataset folder was created with the name "smart_home_dataset", and each stream of data was saved as a csv file with two columns, the timestamp of the measurement and its value. See Figure 20 and

Figure 21 for example screenshots of the dataset in the AEGIS platform. With the successful uploading of the data, the test case 2.1 was implemented.

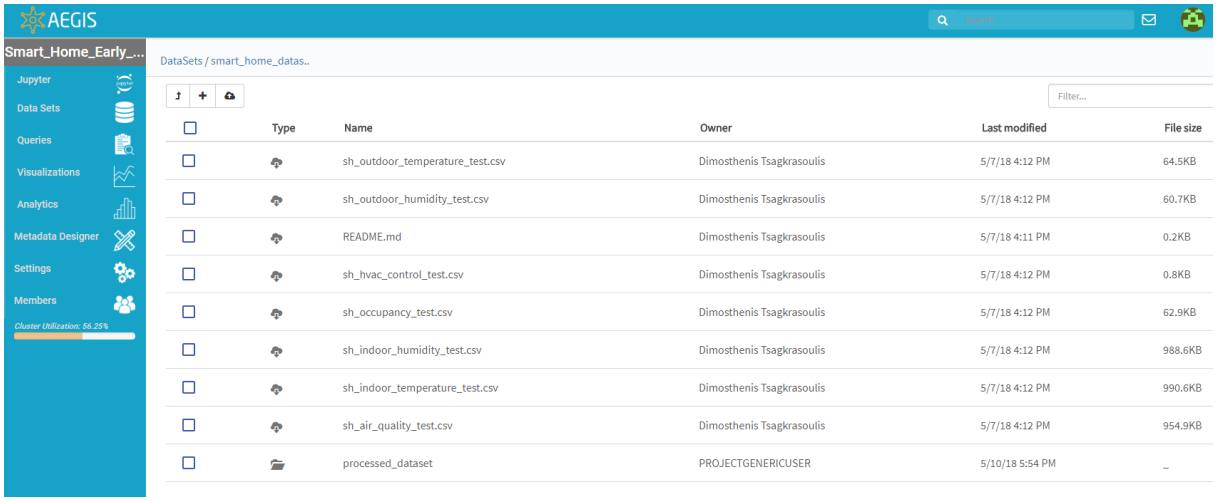


Figure 20: Smart home dataset files uploaded on the AEGIS platform.

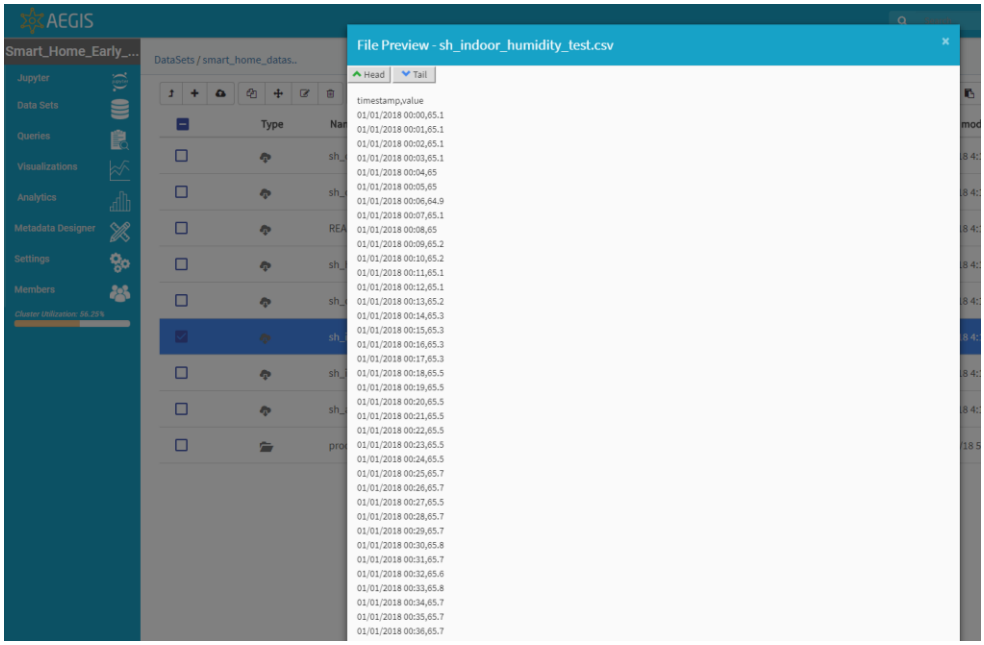


Figure 21: Example timeseries smart home.

Actors:	CSP
Pre-conditions:	1. Installation of HVAC and lighting actuators, communication gateway and sensors at the at-risk individual's premises.

	2. Establishment of external data source/API to retrieve external environmental conditions.
Post-conditions:	1. Batches of data measurements are ready to be processed in the AEGIS platform
Normal Flow	1. Smart home sensors and the gateway are registered to the SHAL backbone app. 2. The gateway communicates temperature, humidity, luminance, VOC concentration, PIR motion events, HVAC status and lighting status to the backbone server. 3. External temperature and humidity data, for the same spatiotemporal variables, are also established in the server, from a publicly available weather service. 4. In the backbone server, any sensitive information (name, address, etc.) is removed (de-identification). 5. Streamed data are organized into batches of predefined size, according to the date/time acquired, and uploaded to the AEGIS platform 6. Following the acquisition of data to the backbone server, the data scientist logs in to the AEGIS platform, creates a project and a respective dataset. 7. He/she arranges a schedule for retrieval of data from the server, through a specified API, along with their metadata information. 8. Data are uploaded to the AEGIS platform through this API.
Pass Metrics	1. Each measured time series object is available within the respective project dataset in a tabular format.
Fail Metrics	1. Problematic communication between the backbone server and the AEGIS platform, leading to loss of data. 2. Inability to retrieve external environmental conditions.
Notes and Issues:	-
Execution Results	The required infrastructure for acquisition of smart home data was installed. Subsequently, data were transformed to csv files and uploaded to the AEGIS platform, under the respective smart home project. The test case was successfully completed.

5.2.2.2. Test Case 2.2 – Smart home data processing and normalization

The Test Case 2.2 concentrated on the evaluation of the data processing capabilities of the AEGIS platform. In particular, smart home data are often dirty, with missing values and outliers, due to unexpected behaviour of sensors, communication infrastructure of even the end user, and not standardized in terms of sampling frequency. As such, processing of the data is of paramount importance for the subsequent analytical steps, which require clean data.

To perform the required steps, as these are detailed in the test case (see below), a custom zeppelin notebook was created by the data scientist within the "Smart_Home_Early_Evaluation_Test_Project". All processing commands was programmed in python and executed with the pyspark interpreter. The steps were the following:

1. The required libraries were imported, the dataset path was set along with the data filenames, and subsequently the contents were read from the files.

2. All datetime entries were transformed to a standard format, and all data points were sorted according to their measurement time.

3. Missing values for all measurement variables, apart from actuation events were filled in. The approach adopted was to fill any missing entries with the corresponding values resulted from linear interpolation of the closest non-missing prior and posterior measurements.

4. The minimum and maximum datetimes are identified and a timestep constant is provided by the user. This information is used in order to interpolate all measurements to a constant time frame.

5. Motion data from the PIR sensors are processed using a custom approach, in order to extract binary occupancy data.

6. Finally, the processed data matrix is saved back to the dataset.

In the following figures, the output log of the aforementioned processing steps is provided, as well as a preview snapshot of the processed dataset. The raw dataset used corresponded to a period on one month, between 01/01/18 and 31/01/18.

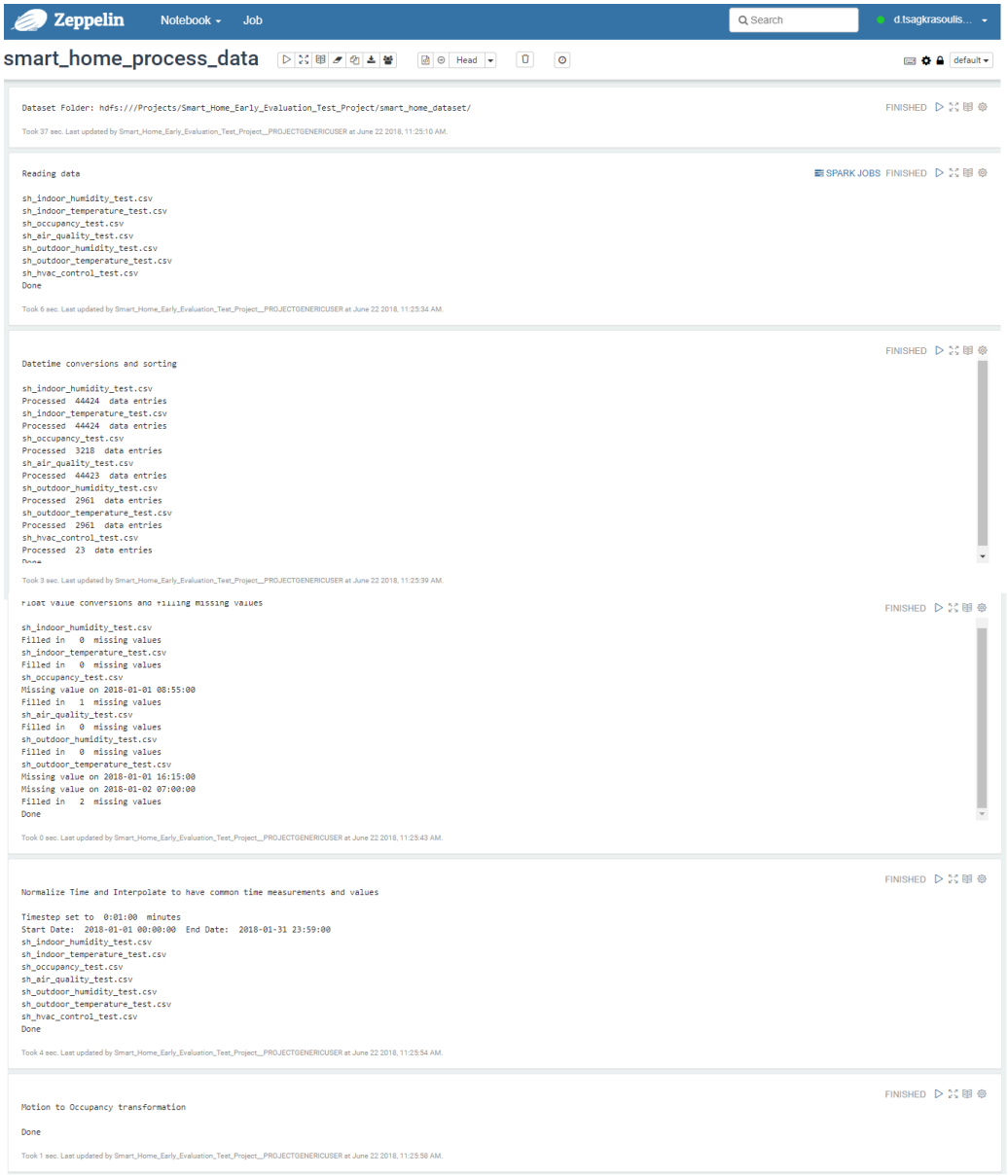


Figure 22: Zeppelin output log for the Smart Home notebook

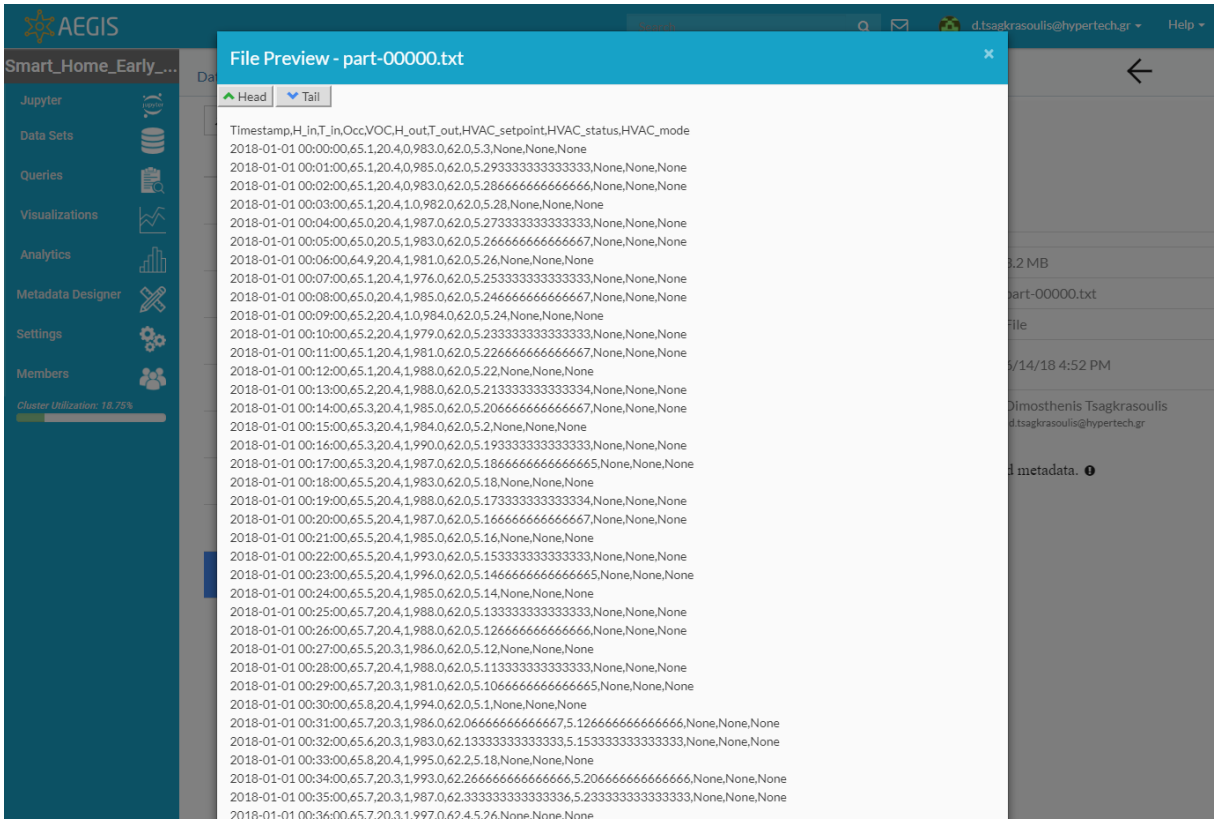


Figure 23: Snapshot of the processed smart home dataset

After processing the data, the Visualiser tool was employed in order to visualise timeseries of the smart home data, in order to inspect for any potential anomalies in the data that could have still remained. Firstly, the appropriate Jupyter notebook was imported to the project. The processed data file was read, as shown in Figure 24. Example timeseries of indoor and outdoor Temperature, as well as occupancy are shown in Figure 26 and Figure 26.

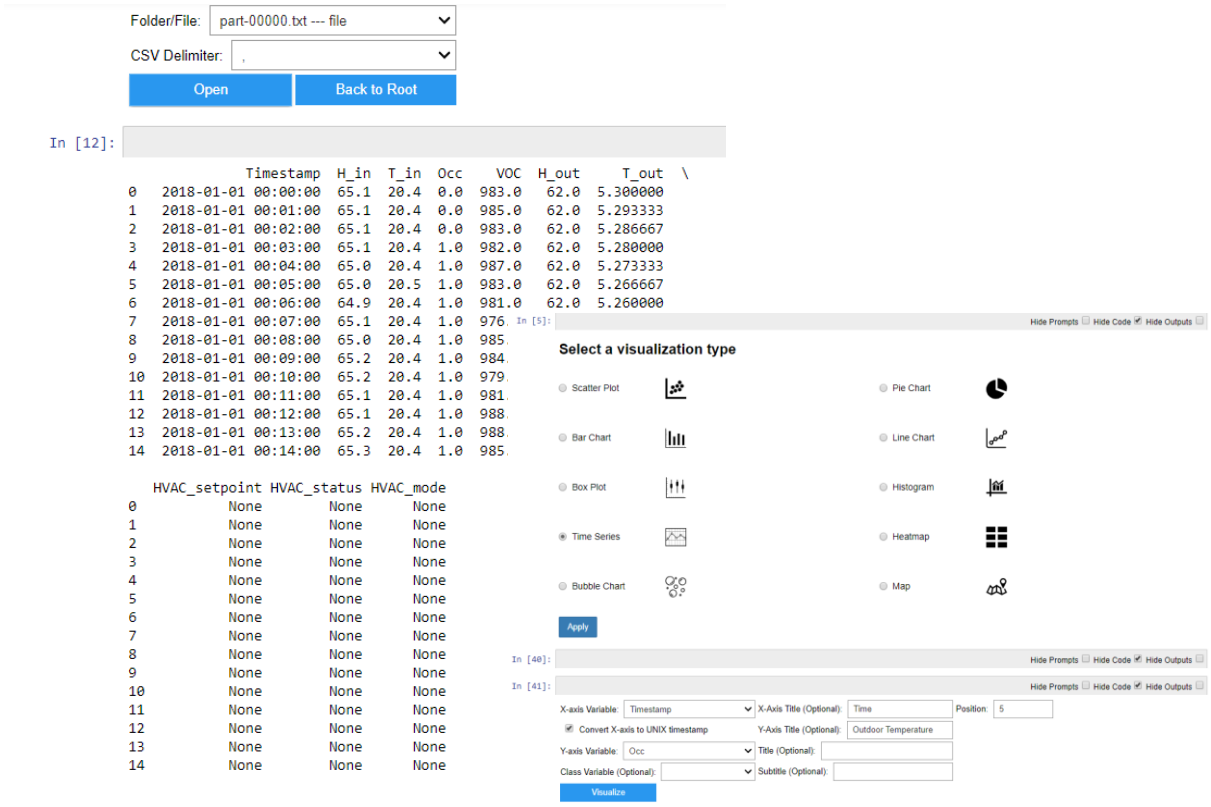


Figure 24. Imported Smart home data in the Visualiser tool.

Figure 25: Example visualisations of indoor, outdoor and occupancy data using the visualiser tool (1)

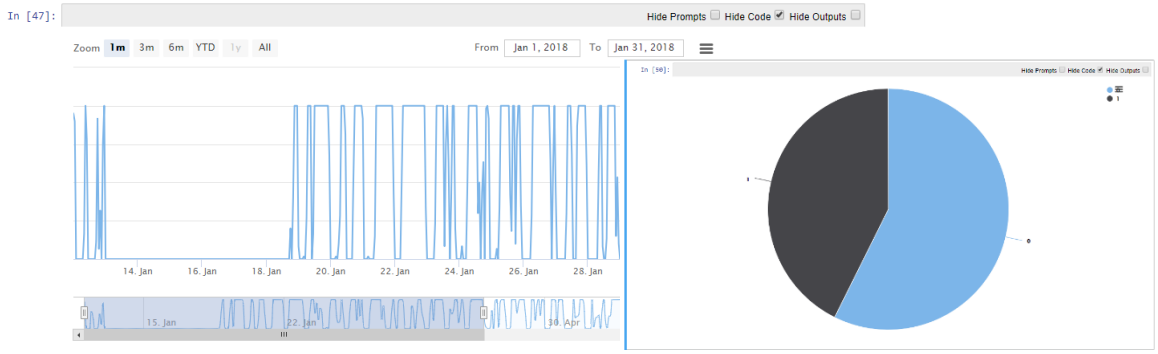


Figure 26. Example visualisations of indoor, outdoor and occupancy data using the Visualiser tool (2).

Actors:	CSP
Pre-conditions:	1. Batches of smart data time series are available as a project dataset.
Post-conditions:	1. All variables are pre-processed, cleaned and interpolated in order to have the same time granularity. 2. Processed data are sent back to the backbone server through a communication API.
Normal Flow	1. The data scientist accesses the smart home project and related dataset in the AEGIS platform. 2. He/she utilizes the Query Builder, or a custom notebook to perform the following data processing steps. 3. Fill in missing values in all data. 4. Transform motion sensor data to occupancy events through a custom algorithm. 5. Define initial and final time points, as well as time step for normalization of all variables. 6. Performs interpolation of the time series so that all variables contain the same number of data points with the same timestamps. 7. Concatenates data into a matrix. 8. Sends data back to the SHAL application backbone server. 9. Optionally, the data scientist may wish to visualise through graphs, scatterplots, histograms, the data, to gain insights regarding the data distribution and correlation between the variables.

	10. Optionally, the day scientist may wish to remove seasonality effects from the time series data through a moving average algorithm.
Pass Metrics	1. Missing values are suitable filled in/interpolated. 2. PIR motion events are transformed to occupancy events (absence/presence). 3. All variables are suitably sampled and/or interpolated, so that all time series have the same granularity and measured time points (i.e. the various time series can be concatenated into an MxN data matrix, where M the number of time points, and N the number of variables).
Fail Metrics	1. The algorithm for the motion to occupancy transformation cannot be applied. 2. Interpolation, sampling/subsampling, and concatenation functionalities are not available in the platform.
Notes and Issues:	-
Execution Results	Raw smart home data measurements were uploaded in the AEGIS platform in the form if individual timeseries in .csv format. A custom zeppelin notebook was created, with algorithmic procedures regarding: filling missing data, extracting occupancy status, interpolating variables so as to have measurements on the same timepoints for all variables, combining results to a matrix and exporting it. As an additional step, the Visualiser component was used to produce timeseries plots of the processed data. The test case was successfully completed. The optional step of removing seasonality effects could not be implemented, since the monitoring process has not yet provided enough data covering different seasons.

5.2.2.3. Test Case 2.3 – Real-time smart home data monitoring

The Test Case 2.3 prescribed a preliminary subset of the visualisation functionality with regards to the visualisation, to the end user, of smart home data, for informative purposes. To achieve this, after the user registers to the SHAL platform and connects a living area to his account, the data coming from the multisensory combo device are shown in the dashboard of his UI app. It must be mentioned that the UI design is still in developmental phase. As such, significant changes regarding the graphical presentation may take place on the later stages of the demonstrator. Additionally, the optional feature of visualising historical data is not currently in place and will be considered for subsequent versions when the UI has matured further.

Actors:	(At-risk) individual
Pre-conditions:	1. Registration to the SHAL service by the individual. 2. Installation of HVAC actuator, communication gateway and sensors at the individual's premises.
Post-conditions:	1. The individual can monitor in real-time environmental conditions through a dedicated UI.
Normal Flow	1. The user logs in to the mobile app. 2. He/she switches to the Smart Home UI. 3. Visualisations showing current temperature, humidity, luminance, and VOC concertation are shown in the GUI dashboard. 4. Optionally, the person may wish to examine past behaviour. He/she clicks on the dashboard of his/her interest, and a time series plot for that variable, containing measurements of a predefined duration in time, is plotted.
Pass Metrics	1. The individual is able to monitor in real-time his/her living conditions through intuitive visualisations.
Fail Metrics	1. Communication issues between the smart home gateway and the UI.
Notes and Issues:	-
Execution Results	Smart home data logged in the backbone server are communicated to the end user through the UI app. The test case was successfully completed. The optional step of visualising past data will be implemented later on when the UI development has progressed further and design details are finalized.

5.3. Demonstrator Evaluation

5.3.1. Quantitative Evaluation

The following table summarizes the quantitative evaluation for the early version of the SHAL demonstrator.

Sub-characteristics	KPIs	Calculation Type	Mandatory / Optional	Means to Verify	Value
Early demonstrator	Portion of successfully	[Successfully Completed	M	Calculation (0-100%)	100%

functional completeness	completed early stage Test Cases	test cases] / [No of tested cases] * 100%			
Comfort satisfaction	Average comfort satisfaction rate	[Sum of Comfort probability as learnt by the profiling model] / No of learnt profiles	O	Quantified comfort of users based on control actions.	Not Applicable during early version
Number of Medical Rules	Medical rules in the web app	No. of medical Rules	M	Count of medical rules in the web app	Not Applicable during early version
Number of Notifications per condition	Notifications / Recommendations attached to outlier	No. of notifications defined	O	Count of notifications defined	For the early version of the demonstrator 20 notifications per condition were successfully issued (manually as described in the scenario)
Outlier Detection Model Accuracy	% Outlier Detection	[Outliers Correctly Identifies]/[Outliers Detected]	O	Manual identification of outliers upon detection	Not Applicable during Early Stage

Table 5: Smart Home and Assisted Living demonstrator quantitative evaluation

5.4. Challenges and recommendations

During the evaluation of the demonstrator and the associated procedures implemented and tested on the AEGIS platform, minor difficulties were encountered, primarily associated with technical aspects of the platform. Such issues were expected, considering the fact the AEGIS platform is currently on the first prototype version, and were solved each time through communication with the technical partners of the consortium. In particular, on a few occasions, execution of notebooks would freeze and required exiting and re-entering the platform. This

issue was encountered mainly during the first phases and is now resolved. Furthermore, on one occasion, the task manager was stuck, leaving zombie processes, which was resolved through a server restart. Again, this can be classified as an expected risk, since the work is performed on the development server, which accepts regular updates and changes. It is expected that a properly maintained production server will have minimal such issues. The recommendations from the SHAL demonstrator's perspective are included in the corresponding qualitative platform evaluation section (See Section **Fehler! Verweisquelle konnte nicht gefunden werden.**). To highlight some of them, a tighter integration of the various tools comprising AEGIS will be an important improvement, which is to our knowledge already worked upon for the later versions of the platform. Furthermore, fine-tuning of the user interface will be an important improvement, since certain glitches still remain (e.g. display of all projects for each user is not intuitive and requires a search, if the project is not on the top ones).

6. AEGIS INSURANCE DEMONSTRATOR

6.1. Pilot overview and current status

The big data analytics offered by the AEGIS platform will allow the Insurance Demonstrator to improve its operational efficiency, with the analysis of growing volumes of structured and unstructured data from both internal and external data sources. The HDI Data Scientists working with the AEGIS Platform will be able to create detailed reports for various needs of the company including customer support, personalized offers, business plan, marketing and pricing strategies.

The Insurance Demonstrator is developed according to three different scenarios, (1) financial impact, customer support and services, (2) personalized early warning systems for asset protection and (3) business plan and marketing strategy. An updated overview of the three scenarios is shown in Figure 27, while Table 6 provides a summary of the Early Demonstrator.

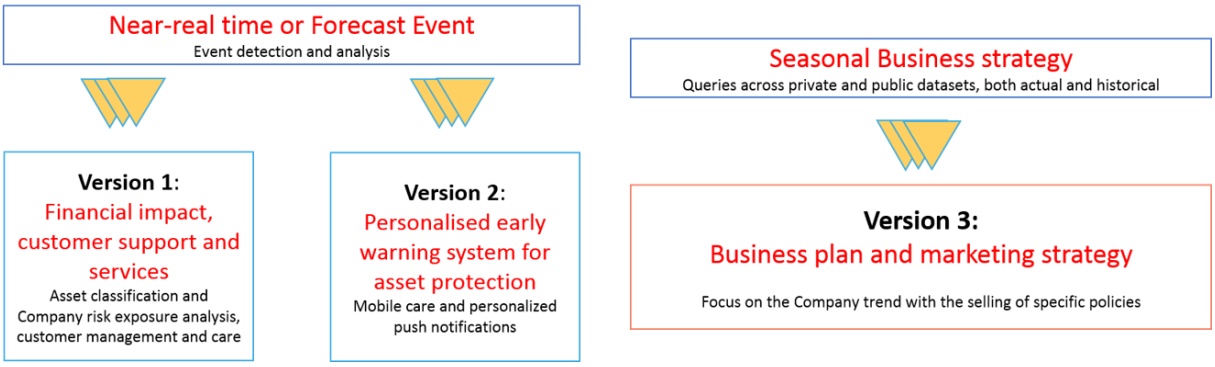


Figure 27: Insurance demonstrator scenarios overview

ID	Scenario	Functionalities	Demonstrator Version
1	Financial impact, customer support and services	Event Detection tool configuration and training[4] Event Detection notification[5] Create Project Uploading in-house dataset Mobile App and geolocation[6]	Early

[4] At the moment the tool is standalone.

[5] At the moment this service has not yet been implemented.

[6] At the moment this service has not yet been implemented.

		Identification of the possibly involved customers Visualisation Priority list (report) sharing Evaluation of the financial impact Customer support and services[7]	
2	Personalised early warning services for asset protection	Additional functionalities required are: Event Detection tool configuration and training (version 2) Identification of the possibly involved customers (version 2) Personalised offer	Medium
3	Marketing strategy and pricing support services	Additional functionalities required are: Business analysis request Business analysis Visualisation (version 2)	Advanced

Table 6: Insurance demonstrator scenarios and required functionalities overview

It is important to note that despite of what has been defined in D5.2 the Early and the Medium Demonstrator have been reversed: considering both the privacy issues (the customers at the moment could not be contacted) and the status of the platform at its first release, the (D5.2) Medium Demonstrator could be adapted better to exploit the AEGIS platform functionalities than the (D5.2) Early Demonstrator.

The actual Early Demonstrator has been changed from what was described as Medium Demonstrator in D5.2 and further test cases are included. Moreover, as remarked with the footnotes of **Fehler! Verweisquelle konnte nicht gefunden werden.**, not all the functionalities have already been implemented within M18 and they will be covered with the next releases of the AEGIS platform, HDI Web App and HDI Mobile App. The next deliverables will also include updates of the first scenario evaluation.

[7] At the moment this service has not yet been implemented.

The first Scenario of the Insurance Demonstrator is mostly related to the near real-time detection of events of interest by the Event Detection Tool as clearly defined by the HDI Data scientists. The HDI Data Scientists through the AEGIS platform can evaluate the risk exposure of the company by identifying the customers that could be affected by the events and the type of the policy/-ies held by them. Through the analysis of some features, for instance the number of accidents and the number of previous injuries, for each customer a priority value is assigned. The list of customers is then sent to the Web App for further processing by the HDI operators that will contact the customers. The main features of the first scenario are shown in Figure 28.

Figure 28: Insurance Demonstrator, Scenario 1, Overview

The involved actors in the first scenario and the actions performed by each of them during the scenario execution are shown in the following schematic (Figure 29), while a detailed description of the workflow executed during the scenario execution is provided in section 6.2.

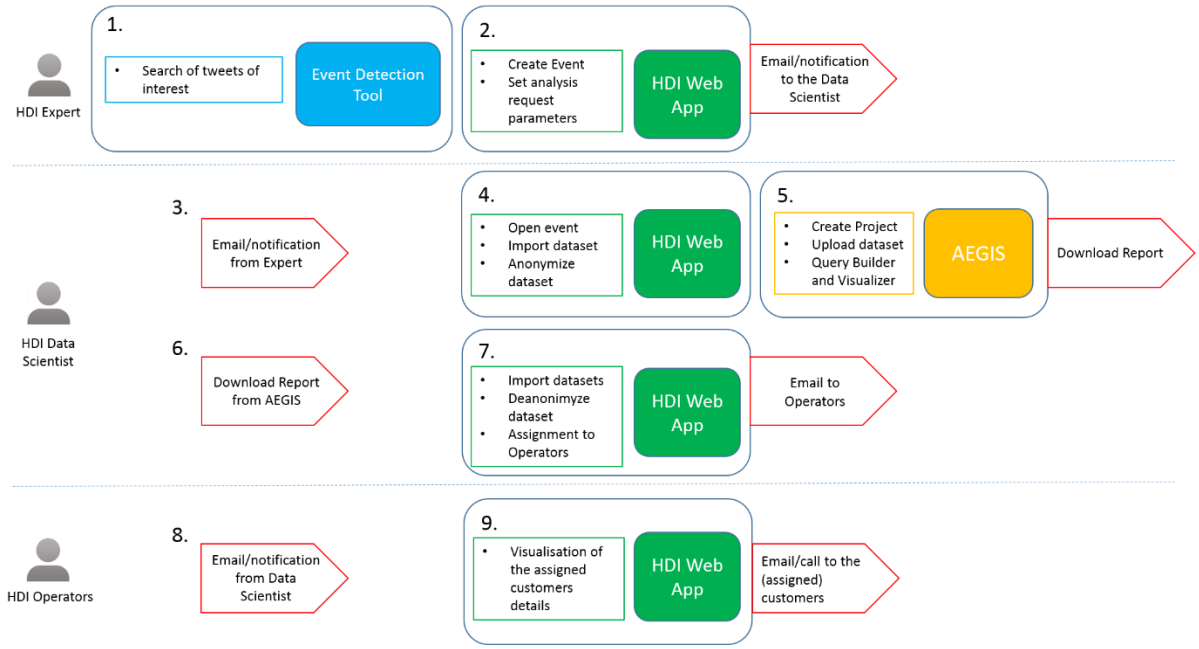


Figure 29: Scenario 1 overview: Actors, environments and main actions

It should be pointed out that for the first evaluation of the Insurance demonstrator the Web App has not been yet integrated with the HDI Systems, although it has been developed taking into account a future integration with the current HDI informative system and databases. The next section will document in detail the scenario execution and its results. Since the Early and the Medium Demonstrators have been reversed, most of the test cases show deviations from what stated in D5.2; for instance, they might not have the same number and name as reported in D5.2, and further test cases have been added in order to describe better each step of the workflow. Additionally, the flows described in the following tables may contain changes from the descriptions in D5.2.

6.2. Scenario execution

6.2.1. Test Case 1 for "Event Detection tool configuration and training"

Actors:	HDI Expert, HDI Data Scientists
Pre-conditions:	1. The AEGIS Consortium provides the Event Detection tool.
Post-conditions:	1. The Event Detection tool detects events of interest for HDI.
Normal Flow	1. The HDI Expert selects a set of interesting data sources and keywords to be taken into account by the Event Detection tool considering the Italian language. 2. A Python script streams the tweets that contain the keywords of interest and creates a csv file to collect them. 3. The Data Scientist manually assigns to each tweet in the csv file a label from -1 to 1, where: • -1 means that the tweet is not of interest, • 0 means that it is relevant to an event but it is not something that happened recently or it is an opinion about an event, • 1 means that it is exactly the kind of tweet needed. 4. After reaching a sufficient amount of labelled records the text of each tweet has been cleaned through a Python script. The result is a csv file with 3 columns: tweet_id, clean-text and label, and another csv with just the clean text (1 tweet per line). 5. These files have been then used as a corpus for the TF-IDF algorithm that transforms the dataset to numeric values. The file is ready for the training.
Pass Metrics	1. The Event Detection tool is trained in detecting events of interest for HDI.
Fail Metrics	1. The Event Detection tool is not well trained, does not detect the right events due to wrong keywords setting or misunderstanding on the keywords meaning in the tweet context.
Notes and Issues:	The time necessary for the collection of a significant number of tweets of interest is unpredictable. The Python code could be also used almost as-is for the streaming of tweets related to other keywords, languages and further Scenarios.

Execution Results	The Event Detection tool has been trained for the early demonstrator with the Italian language and the keywords 'alluvione' and 'allagamento' (flood and flooding) both in the singular and in the plural forms. The number of tweets collected and labelled is around 1000 and neither retweets nor answers have been considered. The machine-learning algorithm of the Event Detection Tool has been improved.

6.2.2. Test Case 2 for "Event Detection Tool Streaming"

Actors:	HDI Expert, HDI Data Scientist
Pre-conditions:	1. The Event Detection tool has been trained with the keywords of interest for HDI.
Post-conditions:	1. The Data Scientist receives an email and a notification on the Web App from the HDI Expert and is enabled to start his/her analysis on AEGIS.
Normal Flow	1. The HDI Expert through the Event Detection Tool API is informed that an event that has been detected. 2. The HDI Expert using his/her knowledge (eventually making some research on the internet) evaluates if the event could be of interest for the company. 3. If the event is of interest, the HDI Expert sends an analysis request to the Data Scientist (by filling a predefined form within the Web App) (Figure 30).
Pass Metrics	1. The Event Detection Tool API detects events of interest for HDI. 2. The HDI Expert fills the form and assigns the event to a Data Scientist, who receives an email and a notification with the details of the analysis to be performed.
Fail Metrics	1. The HDI Expert cannot fill the form in an exhaustive way. 2. The Data Scientist does not receive an email and a notification with the details of the analysis to be performed.
Notes and Issues:	During the evaluation time, no significant events occurred in the area of Rome. The steps hereinafter have been simulated.
Execution Results	The interaction between the HDI Expert with the Event Detection tool and the Web App has been tested many times by different users. When the Expert assigns a new event to a Data Scientist, he/she receives both an email and a notification on the Web App. The need of sending an email is

	in order to access the Web App, hence to see the notification, the Data Scientist has to be logged in the Web App. The form that the HDI Expert fills to request a new analysis from the Data Scientist has been jointly defined by them with the aim to drive the analysis that is going to be performed. The fields that should be filled include general information about the event (for instance type and date) and instructions for the analysis and its sharing. When the analysis request is sent, a new event is created and an ID is associated to the event: both the Data Scientist and the Expert from the Web App can display the event details and status.

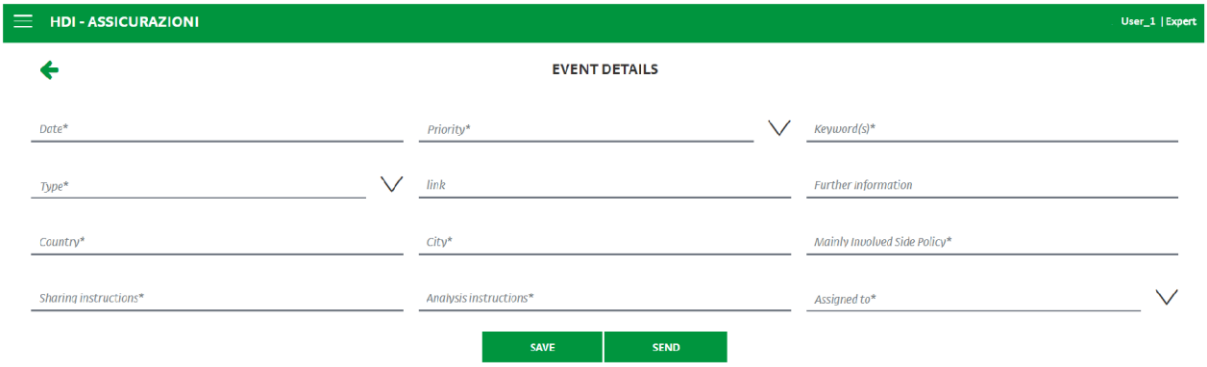
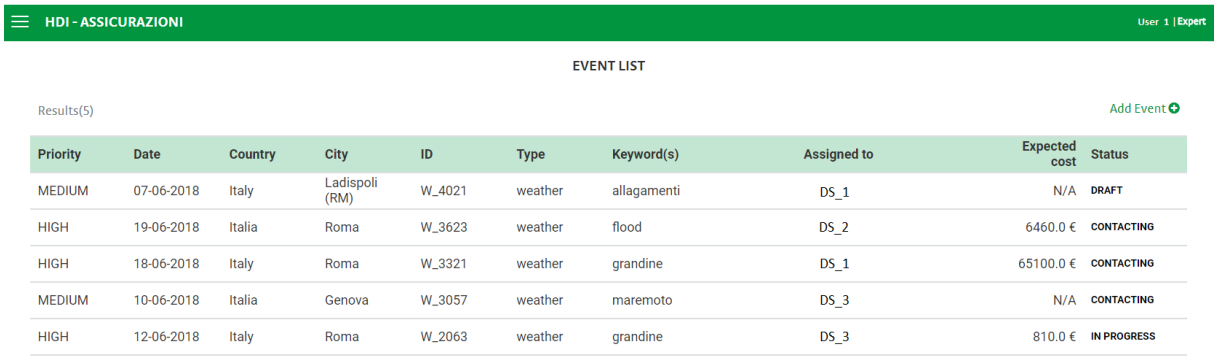


Figure 30: HDI Web App - HDI Expert, Event Details (Empty Analysis Request Form)



Priority	Date	Country	City	ID	Type	Keyword(s)	Assigned to	Expected cost	Status
MEDIUM	07-06-2018	Italy	Ladispoli (RM)	W_4021	weather	allagamenti	DS_1	N/A	DRAFT
HIGH	19-06-2018	Italia	Roma	W_3623	weather	flood	DS_2	6460.0 €	CONTACTING
HIGH	18-06-2018	Italy	Roma	W_3321	weather	grandine	DS_1	65100.0 €	CONTACTING
MEDIUM	10-06-2018	Italia	Genova	W_3057	weather	maremoto	DS_3	N/A	CONTACTING
HIGH	12-06-2018	Italy	Roma	W_2063	weather	grandine	DS_3	810.0 €	IN PROGRESS

Figure 31: HDI Web App - HDI Expert, Event List

6.2.3. Test Case 3 for "Create Account and Project"

Actors:	HDI Data Scientists
Pre-conditions:	1. The Data Scientist receives an email and a notification on the Web App from the HDI Expert and starts his/her analysis on AEGIS.

Post-conditions:	1. The Data Scientist has an AEGIS account. 2. The project is created.
Normal Flow	1. The Data Scientist opens the AEGIS platform in the Web browser and creates a user account. 2. He/she then logs into the platform and creates a new project named "HDI_EARLY_D".
Pass Metrics	1. The user account and the new project are created.
Fail Metrics	1. No user or no project is/are created.
Notes and Issues:	The AEGIS Administrator has to set manually the number of projects allowed for each user. In the next platform releases a functionality to contact directly the Administrator to request an increase of the projects number allowed should be added.
Execution Results	While creating the accounts and the projects no issues have been encountered by the Data Scientists.

6.2.4. Test Case 4 for "Anonymised in-house dataset upload"

Actors:	HDI Data Scientists
Pre-conditions:	1. The Data Scientist has anonymised the in-house datasets of interest for the analysis he/she is going to perform (following the instructions given by the HDI Expert). 2. The Data Scientist has an AEGIS account. 3. The Data Scientist is logged in. 4. The Data Scientist has previously created a project.
Post-conditions:	1. The dataset of interest is uploaded and associated with the "HDI_EARLY_D" project.
Normal Flow	1. The Data Scientist logs into the AEGIS platform, opens the "HDI_EARLY_D" projects and within that project creates a new dataset named "HDI_Customers_inHouse". 2. He/she selects the "upload data" functionality and uploads customers data (csv-file) from his/her local computer to the platform.
Pass Metrics	1. The dataset is uploaded. 2. The dataset is associated with the "HDI_EARLY_D" project.
Fail Metrics	1. The dataset is not uploaded and/or not associated with the right project.

Notes and Issues:	-
Execution Results	The activities described were performed easily by the Data Scientists, even if the upload of heavier files is highly time consuming as well as the preview of these files.

6.2.5. Test Case 5 for "Identification and visualisation of the possibly involved customers"

Actors:	HDI Data Scientists
Pre-conditions:	1. The HDI Expert has filled the form and the Data Scientists has received an email with the details of the analysis to be performed. 2. The dataset of interest is uploaded in AEGIS and associated with the "HDI_EARLY_D" project.
Post-conditions:	1. The Data Scientist obtains a dataset about the possibly involved customers; the list of customers depends on the impact area of event.
Normal Flow	1. The Data Scientist on the AEGIS platform through the Query Builder correlates the features of the event with the in-house dataset regarding the customers' policies and location (Figure 32). 2. The Data Scientist obtains a csv-file with the list of the customers residents in the area, that have an asset or a real estate with a valid policy coverage in the area. The file contains the ID of these customers, the type of the valid policy held and the location of the insured asset(s). 3. To have a fast overview of the customer interested he/she can visualise them on a map (each customer is represented with a pin).
Pass Metrics	1. The query filters the rows of the file related to the customers that could be involved in the event. The content of the rows is not changed. 2. The Data Scientist can visualise the customers pointed out from the Query Builder analysis on a map (Figure 33). The customers' location is identified with a marker.
Fail Metrics	1. The analysis is not well performed, the dataset was not appropriate or the query not properly defined.
Notes and Issues:	-
Execution Results	The Data Scientists, initially supported by the AEGIS partners that have built the Query Builder and the Visualiser have executed this step by using

	many (anonymised) customer's csv-files from the HDI databases. No issues have been encountered.

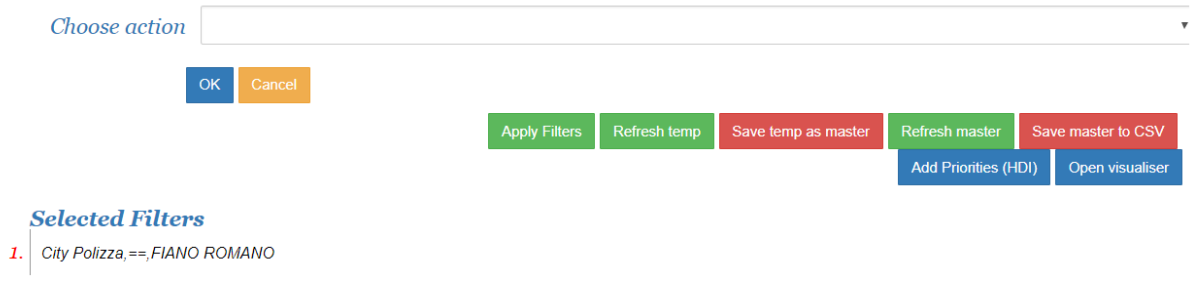


Figure 32: Example of usage of the Query Builder

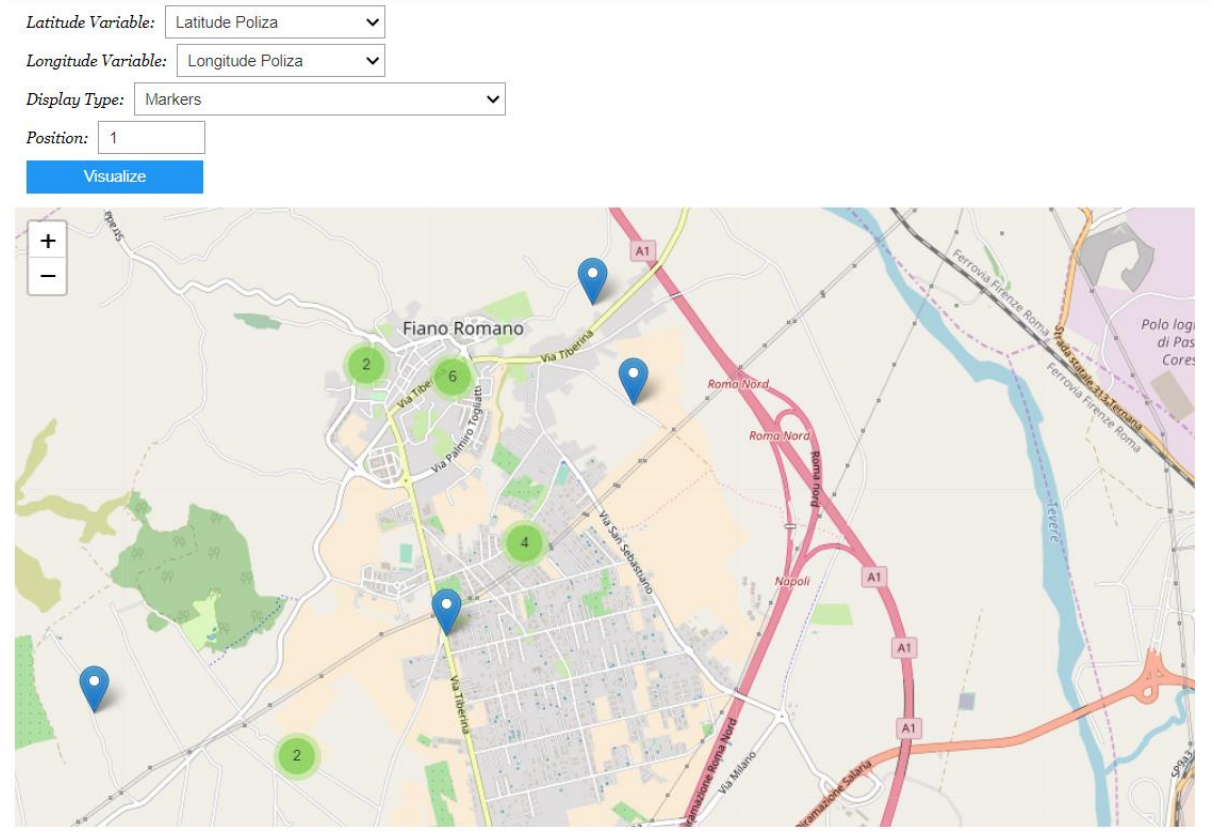


Figure 33: Query Builder output is visualised: the markers indicate the customers' asset location

6.2.6. Test Case 6 for "Priority list creation"

Actors:	HDI Data Scientists

Pre-conditions:	1. The Data Scientist has identified the customers possibly involved in the event through the Query Builder.
Post-conditions:	1. A column with a priority value is added to the dataset, a new csv-file "HDI_priorityList_customers" is created and downloaded by the Data Scientist.
Normal Flow	1. The Data Scientist in the same Jupyter notebook containing the Query Builder and the Visualiser, implements a prioritization algorithm. The priority rules have been defined by the HDI Expert as specified in the form with the analysis instructions (Figure 30). 2. Running the paragraph, a column is added to the dataset. The column contains the priority value associated to each customer. 3. The Data Scientist saves the new dataset, "HDI_priorityList_customers", and downloads it in his/her own device.
Pass Metrics	1. The priority rules are well defined. 2. The new csv-file includes a column with the priority value associated to each customer. 3. The "HDI_priorityList_customers" file is saved as new dataset in the AEGIS platform and then downloaded by the Data Scientist.
Fail Metrics	1. The Data Scientist has no knowledge about the priority rules. 2. The paragraph is not well implemented, the values are not the right or the column is not added. 3. The Data Scientist is not able to download the file from AEGIS.
Notes and Issues:	1. The HDI Expert has to define precise rules for the prioritization.
Execution Results	The Data Scientists performed this test case with different datasets and rules. The execution of this test case has been made easier by the 'Add priorities (HDI)' button (Figure 32) although the priority rules should be defined/changed each time following the analysis needs. By pushing the 'Save as Master' button (Figure 32), the Data Scientists automatically uploaded the resultant file in a specific dataset folder related to the project. From here, the file is downloaded locally. While executing this test case no issues were encountered.

6.2.7. Test Case 7 for "Priority list (report) de-anonymization and assignment, risk exposure evaluation"

Actors:	HDI Data Scientists
Pre-conditions:	1. The Data Scientist obtains a prioritized list with the possibly involved customers from the AEGIS platform.
Post-conditions:	1. The Data Scientist has assigned all the customers of the "HDI_priorityList_customers" file to the designed operators. 2. The HDI Expert can visualise on his/her Web App the estimated cost of the event. 3. The operators receive a notification on the Web App and an email informing them that they have to contact some HDI customers that could be involved in an event.
Normal Flow	1. The downloaded file from AEGIS is uploaded in the Web App and de-anonymised (Figure 34). 2. The Data Scientist sets on the Event Details page of the Web App the mean cost of the event for each policy type (Figure 34). 3. The risk exposure of the company is automatically calculated when the file is uploaded considering the number of customers in the list and their type of policy(-ies). 4. The Data Scientist assigns the customers on the list to the proper Operator(s), and the Web App automatically sends them an email and a notification on the Web App (Figure 35).
Pass Metrics	1. The file is correctly de-anonymized. 2. The risk exposure of the company is evaluated and the HDI Expert can display its value. 3. All the customers in the list have been assigned to the proper Operator.
Fail Metrics	1. The file is not de-anonymized or the 'anonymised' customers are not correctly related to their original data. 2. The risk exposure of the company is not evaluated and/or the HDI Expert can't display its value. 3. The Web App does not work correctly and the Data Scientist is unable to assign the customer to the Operator(s) and/or the email/notification are not sent to the proper Operator.
Notes and Issues:	While the Data Scientist assigns the customers to the Operator(s), the list is split accordingly, in order to provide them only the data of the customers they have to contact.

	Before assigning the customers, the Data Scientist has to provide sharing instructions to the Operator, following the rules defined by the HDI Expert (Figure 30).
Execution Results	No issues have been found by none of the Data Scientists since these steps are totally automated. The most important things are the knowledge of the Data Scientist for the risk exposure estimation and the accuracy of the rules defined by the Expert (for both of the assignments and the sharing instructions). The filter on the top of the customer assignment page (Figure 35) has been inserted to made easier the assignment to Operators.

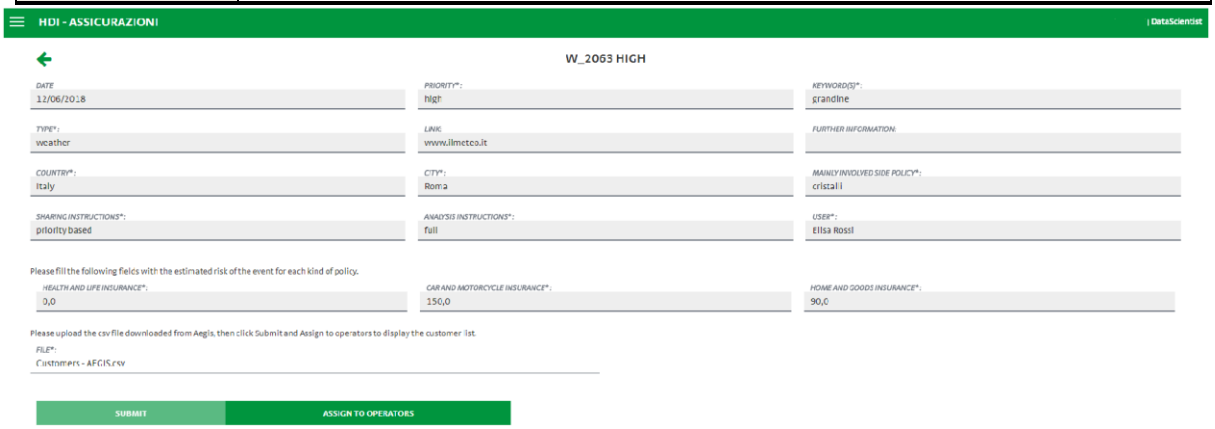


Figure 34: HDI Web App - Data scientist, Event Details page

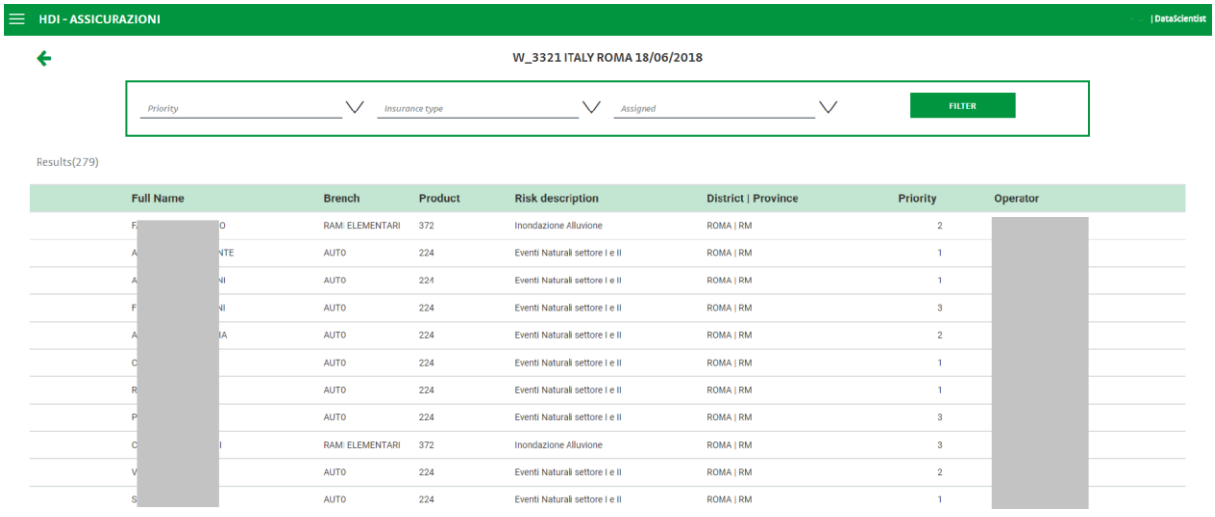


Figure 35: HDI Web App - Data scientist, Customers assignment

6.2.8. Test Case 8 for "Priority list (report) de-anonymization and assignment"

Actors:	HDI Operators

Pre-conditions:	1. The Data Scientist has assigned all the customers of the "HDI_priorityList_customers" file to the designed operators. 2. The operators receive a notification on the Web App and an email to informing them that they have to contact some HDI customers that could be involved in an event.
Post-conditions:	1. The HDI operator has contacted the customers.
Normal Flow	1. The HDI Operator displays the list of the customers assigned to him and starts contacting them with an email or a call depending on the instructions received. 2. The HDI Operator contacts the customers and if the customer has been hit by the event, provides support (for instance instructions and documentation).
Pass Metrics	1. The proper HDI Operator receives a list of the customers that has to contact. 2. The customer is contacted by the HDI Operator and receives support.
Fail Metrics	1. The HDI Operator does not receive the list of the customers. 2. The Web App does not work correctly and he/she cannot modify the status of the contacting of each customer.
Notes and Issues:	-
Execution Results	The steps have been performed by various operators until the last one: within the Early Demonstrator the customers have not been alerted (the events were simulated). Each operator in the Web App homepage displays the list of events, and for each event the list of customers he/she has to contact (Figure 36). The Operator can filter the list (priority, customer surname, contacted) and add a note for each customer.

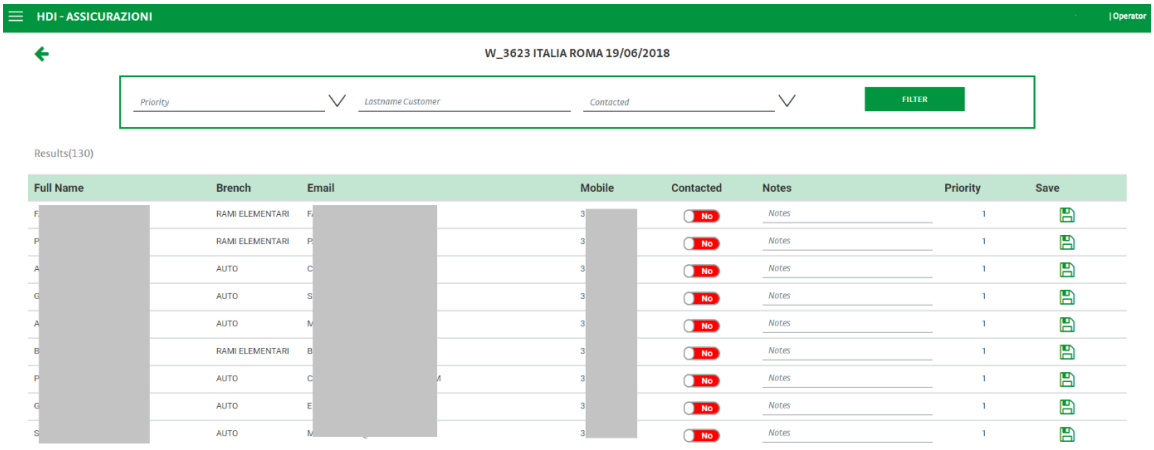


Figure 36: HDI Web App - Operator, Contacts

6.3. Demonstrator Evaluation

6.3.1. Quantitative Evaluation

Most of the KPIs defined in D5.2 are not all applicable to the Early Demonstrator since the developed Web App has not been yet integrated with the HDI Systems and the customers have not been actually contacted. Further KPIs were added to the previous list in order to provide a quantitative evaluation of the first scenario.

The KPIs reported in D5.2 will be evaluated during the next period when the two Apps will be integrated and the customers could be contacted.

Sub-characteristics	KPIs	Calculation Type	Mandatory / Optional	Value
Early demonstrator functional completeness	Portion of successfully completed early stage Test Cases	[Successfully Completed test cases] / [No of tested cases] * 100%	M	100%
Early and Medium Demonstrator functional completeness	Event Detection Tool reliability	[No of events of interest] / [No of events detected] * 100%	M	70%
Early and Medium Demonstrator functional completeness	Event Detection Tool trained events	Sum	M	1

All Demonstrators	Number of datasets uploaded	Sum	M	52

The Web App that has been developed to allow the information exchange between the three actors involved in the process is working as expected, and the execution flow was tested by different users for each role and no issues were observed.

The Data Scientists' analyses within the AEGIS platform have also succeeded. The problems encountered in the first tests have been fixed in cooperation with the other partners of the consortium.

6.4. Challenges and recommendations

The main issue related to the Early Demonstrator and, in general, to the Insurance Demonstrator is the privacy and security regulations. Since the topic is crucial and it needs a deep knowledge and accurate handling, one of the Ethical Advisory Boards has supported HDI for the scenarios' definition. In order to respect the Italian and European Legislation about data treatment, and the Insurance specific policies, the in-house datasets stored in HDI databases are uploaded on the platform after their anonymization. The data are managed and handled only by the HDI employees that are working on the AEGIS project and only the columns of interest for the project evaluation purposes are kept. Additionally, at the end of each analysis the in-house datasets are deleted from the AEGIS platform. More details about security and privacy within the Insurance Demonstrator are available in D1.3, D9.1 and D9.3.
The security and privacy issues are reflected in the workflow described in the previous paragraph where there is the need to upload and download data from the HDI databases to the AEGIS platform. The challenge for the Medium and the Advanced Demonstrator is to avoid these multiple steps while guaranteeing data protection.

In general, the Early Demonstrator has satisfied all the HDI actors, in terms of usability of the tools and accuracy of the analysis. Toward this end, the cooperation between the HDI employees and the technical team of the project was fundamental, from the definition of the user stories (D3.1) to the test cases execution.

7. CONCLUSION

The objective of this deliverable was to document the efforts undertaken within the context of Tasks 5.2, 5.3, 5.4, 5.5 and 5.6 of WP5. This deliverable builds on top of the work and outcomes of deliverables D5.1 and D5.2 in order to report the progress of the AEGIS demonstrators and perform the evaluation of the AEGIS platform and the AEGIS demonstrators during the first (early) phase of demonstrators' implementation.

At first, the evaluators involved in the AEGIS evaluation framework were presented, describing their knowledge and technical expertise, their role in the project and their involvement in the evaluation process.

Following the presentation of the evaluators, the AEGIS platform v1.00 evaluation was performed. In accordance to the AEGIS evaluation framework, the AEGIS platform evaluation followed the twofold approach incorporating both quantitative evaluation based on the list of KPIs, as defined in D5.2, and the qualitative evaluation based on guided interviews with key stakeholders. The results of both methods were documented followed by a description of the key challenges faced in regards to the AEGIS platform v1.00 in the course of the first (early) version of the demonstrators and a list of recommendation for the enhancement of the platform.

Following the AEGIS platform evaluation, a comprehensive description of the current status and work performed during the implementation of the first (early) version of the demonstrators is documented. For each of the scenario(s) executed during this phase the results of the corresponding steps are provided along with the related implementation details. The evaluation of each demonstrator is performed and the challenges faced during the implementation are discussed. Finally, a list of recommendations is presented for the upcoming versions of the demonstrators.

In the next steps the knowledge extracted from this deliverable will be further analysed by the AEGIS platform developers so as to identify the updates and enhancements required in the upcoming version AEGIS platform in order to address the demonstrators' needs. It should be noted at this point that the demonstrator evaluation and feedback is a living process that will last till M30, when the final demonstrator evaluation and feedback (corresponding to D5.5) and the final evaluation, impact assessment and adoption guidelines (corresponding to D5.6) will be delivered. The evaluation of the upcoming version of the platform, as well as the demonstrators' second (medium) phase will be documented in the upcoming deliverable D5.4 entitled "Demonstrators Evaluation and Feedback – v2" which will be delivered in M24.