Horizon 2020 Program (2014-2020)

Big data PPP

Research addressing main technology challenges of the data economy

Industrial-Driven Big Data as a Self-Service Solution

D2.3: I-BiDaaS visualization and monitoring framework, and a multi-purpose interface[†]

Abstract: This deliverable describes the design of the interactive visualization tools that will form the front-end of the I-BiDaaS platform. The design methodology is initially presented and then the platform functionalities.

Contractual Date of Delivery	31/12/2018
Actual Date of Delivery	31/12/2018
Deliverable Security Class	Public
Editor	*Ilias Spais (AEGIS)*
Contributors	BSC, IBM, SAG, ATOS, ITML, UNSPMF
Quality Assurance	*Omer Boehm (IBM)* *Dr. Gerald Ristow (SAG)* *Dr. Kostas Lampropoulos (FORTH)*

[†] *The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780787.*

The *I-BiDaaS* Consortium

Foundation for Research and Technology – Hellas (FORTH)	Coordinator	Greece
Barcelona Supercomputing Center (BSC)	Principal Contractor	Spain
IBM Israel – Science and Technology LTD (IBM)	Principal Contractor	Israel
Centro Ricerche FIAT (FCA/CRF)	Principal Contractor	Italy
Software AG (SAG)	Principal Contractor	Germany
Caixabank S.A. (CAIXA)	Principal Contractor	Spain
University of Manchester (UNIMAN)	Principal Contractor	United Kingdom
Ecole Nationale des Ponts et Chaussees (ENPC)	Principal Contractor	France
ATOS Spain S.A. (ATOS)	Principal Contractor	Spain
Aegis IT Research LTD (AEGIS)	Principal Contractor	United Kingdom
Information Technology for Market Leadership (ITML)	Principal Contractor	Greece
University of Novi Sad Faculty of Sciences (UNSPMF)	Principal Contractor	Serbia
Telefonica Investigation y Desarrollo S.A. (TID)	Principal Contractor	Spain

Document Revisions & Quality Assurance

Internal Reviewers

1. *Omer Boehm, IBM*
2. *Dr. Gerald Ristow, SAG*
3. *Dr. Kostas Lampropoulos, FORTH*

Revisions

Version	Date	By	Overview
0.1	20/10/2018	AEGIS	Table of Contents
0.2	28/10/2018	UNSPMF	Comments on ToC
0.3	28/11/2018	AEGIS	First Draft
0.4	06/12/2018	AEGIS	Consolidation of contributions by BSC, UNSPMF
0.5	14/12/2018	AEGIS	Consolidation of contributions by IBM, SAG
0.6	18/12/2018	AEGIS	Draft version for internal review
0.7	19/12/2018	SAG	Added some comments and updated ToC and LoF
0.8	21/12/2018	IBM	Draft review - comments inside
1	22/12/2018	AEGIS	Consolidation of contributions by ATOS and internal review by SAG and IBM. Final version

Table of Contents

List of Tables

List of Figures

Executive Summary

This deliverable presents the User Centric Design methodology to design user interfaces of increased usability and describes the I-BiDaaS approach to it. An overview of the platform functionalities follows and forms the basis for the definition of the basic requirements and workflows of the end users. The document then describes the main characteristics, technologies and integration points of the tools comprising the visualisation and monitoring framework of the I-BiDaaS platform. Moreover, a high-level description of the communication framework between visualisation and the rest of the components of the platform is presented and an initial set of relevant web services per component is outlined.

The multipurpose interface is given in the form of wireframes that depict the information presented in the various screens of the platform and explain how the aforementioned workflows take place using the interface components. Screenshots and a brief description of the deployed first prototype of the interface provide the current implementation status which is closely related with the needs of the MVP as reported in D5.2 'Big-Data-as-a-Self-Service Test and Integration Report'.

1 Introduction

This document is the third deliverable of WP2 and stands upon the user requirements and MVP needs to design the interactions expected between the I-BiDaaS actors and the offered environment. The work, which is presented in this deliverable, strongly relates to the user centric design process that the project follows in order to develop the I-BiDaaS components, in order to ensure that development is aligned to the user needs and the released software receives the acceptance of the stakeholder groups. As such, this deliverable establishes relationships with the work performed in WP6 about the evaluation and validation of the platform through real-life industrial experiments. It is also strongly connected with technical WPs (WP2, WP3, WP4 and WP5) since the described visualisation framework and multipurpose interface receive the outcome of the tools developed in these WPs and expose it to the end users.

The user centric design process described in Section 2 aims to achieve high degree of usability for the I-BiDaaS components. A principal requirement for this is to ensure the engagement of the I-BiDaaS end users in the implementation early in advance in the project time plan and this document presents the outcome of this action. Taking into account the system requirements, the basic workflows that end users will follow when using the platform interface are defined in Section 3. Moreover, the main components used to create the I-BiDaaS interface are presented in detail in Section 4, namely the AEGIS AVT and SAG's MashZone NextGEN. The integration between them but also the communication framework with the rest of the components of the platform are also defined in the document and will serve as a guide for the next steps of development and integration.

Finally, Section 5 describes the multipurpose interface which is the user-facing component of the platform that incorporates visualisations and interactions with all the underlying components. Based on the defined design process, we present the initial wireframes of the platform interface that provides the envisaged functionalities to end users. A subset of these wireframes has been used to produce the first working version of the interface that supports the MVP and will be used for the first round of user evaluation. Conclusion of the performed work and future steps conclude the document.

2 User Centric Design

2.1 Introduction to UCD process

The development of a software system follows the software life cycle processes defined in the ISO/IEC 12207:2017 standard [1]. In this standard, software implementation is evolved into various processes, which refer to the software requirement analysis, the software architecture design, the software construction and integration, the software qualification testing and the software support and reuse processes. The support services include the software quality assurance procedures and validation processes. Software validation is the confirmation that the software specifications conform to user needs and intended uses through examination and provision of objective evidence, and that the particular requirements implemented through software can be consistently fulfilled. Since software is usually part of a larger hardware system, the software validation typically includes evidence that all software requirements have been implemented correctly and completely.

Along the processes of a software implementation project, the level of the stakeholders' involvement in these processes defines whether a user centric design (UCD) approach is being adopted in the development of the software system. A UCD approach, which is, also, called human centric design, targets on the integration of a multidisciplinary group, which focuses on the interactive development of software solutions with emphasis on validating their usability. As mentioned in ISO 9241-210:2010 [2], UCD is the approach to systems design and development that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques. Thus, the scope of a UCD process is to achieve high degree of usability, which is introduced as an inherent measurable property for all interactive digital technologies and is defined in ISO 9241-11:2018 [3] as "the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". In order to meet this objective, the UCD process requires that the software end users and the solution domain stakeholders must be engaged with the implementation activities early in advance, through a structured and systematic approach for gathering all the necessary information, which will guide the software designers in making an effective first approach of the software sketch [4].

The UCD process is strongly dependable on user acceptance criteria, thus, it should approach usability from various perspectives and focus on aspects for the ease of use and the learning curve to adopt the solution in current business practices [5]. This can be measured by the envisaged users through empirical analysis of similar solutions and the actual testing of the software product, as it is progressively made available from a mock-up version, through functional prototyping and up to product testing. In this context, iterative design is a key function in the whole software development process for the continuous evaluation of usability aspects in the software solution.

The usability evaluation aims to assess the extent to which an interactive software solution is easy and pleasant to use. A lot of methods exist to measure the degree that is achieved in the development of a software interactive solution. These methods provide the theoretical background to apply specific robust, objective and reliable metrics that allow a quantifiable approach to usability. The most well-known usability evaluation models are: i) quality in use model, which is used to asses to the use of the software product (effectiveness, efficiency and satisfaction in a particular context of use), and ii) the product quality model, which is used to assess the product user interface and interaction. These quality models are commonly included as part of the software validation process in the ISO/IEC 12207:2017 standard [1]. However, in a UCD process, these models should be employed across all the processes of the software implementation.

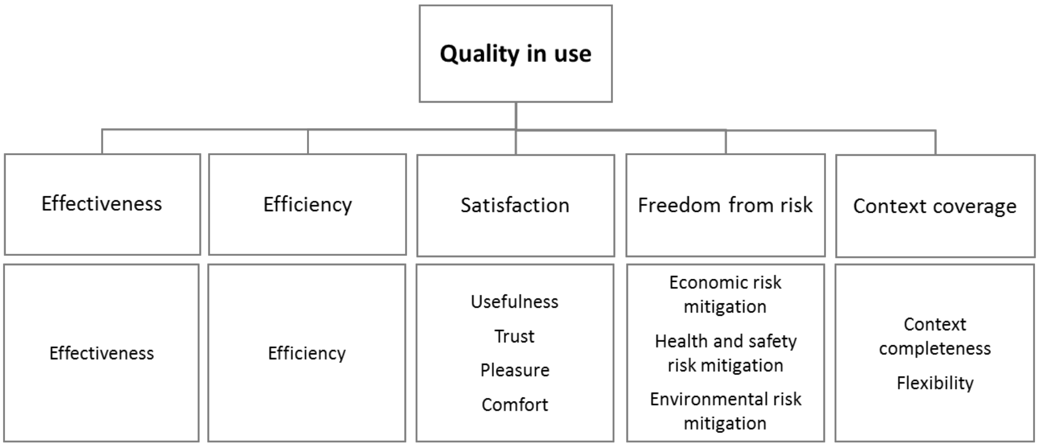


Figure 1. The quality in use model of the ISO/IEC 25010:2011 standard

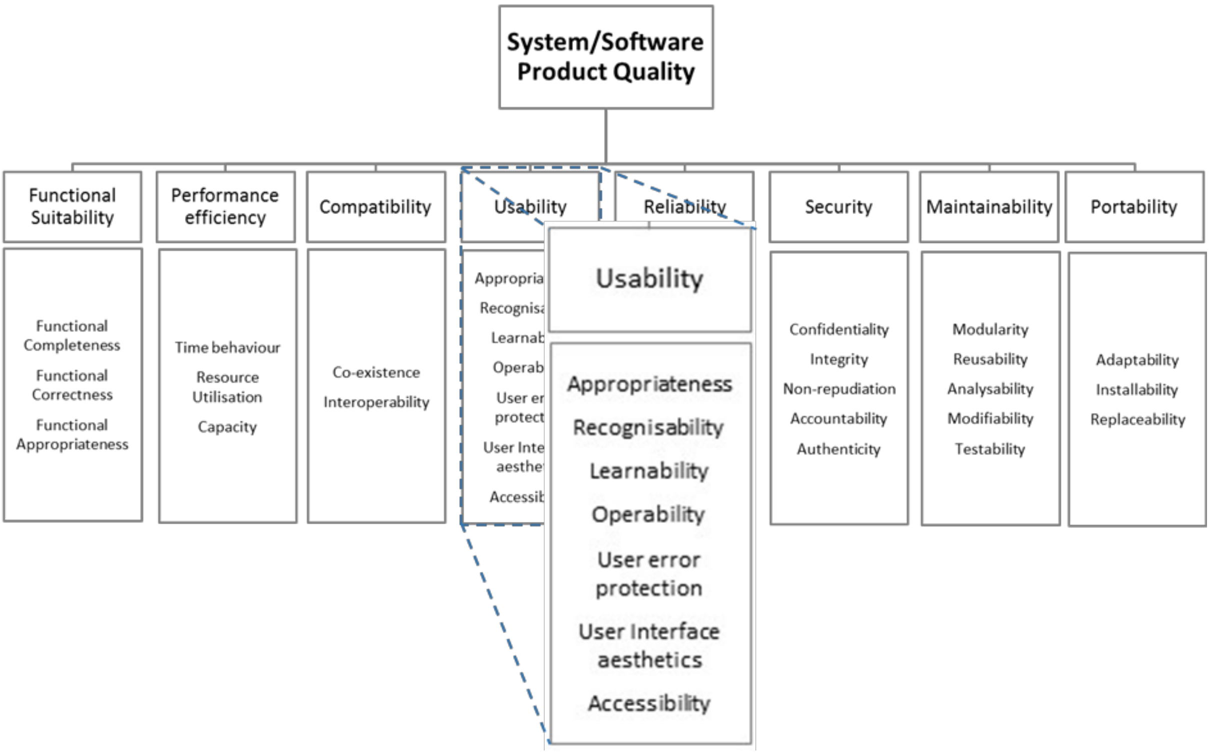


Figure 2. The product quality model of the ISO/IEC 25010:2011 standard

The ISO/IEC 25010:2011 standard [6] is the most widespread reference model for software assurance and it includes eight software quality characteristics, across which the software solution can be assessed. This standard introduces the above-mentioned quality models providing a consistent terminology for specifying, measuring and evaluating system and software product quality. Thus, in ISO/IEC 25010:2011 standard:

- The quality in use model is composed of five characteristics that relate to the outcome of the interaction with the system and characterises the impact that the product can have on the stakeholders. In this model, usability is in the forefront and is realised through the characteristics shown in Figure 1.

- The product quality model is composed of eight characteristics that relate to static properties of software and dynamic properties of the computer system. In this model, the eight characteristics can be further divided into sub-characteristics, are shown in Figure 2.

The software quality model of the ISO/IEC 25010:2011 standard is adapted on a case by case basis, in order to define appropriate metrics and to be able to evaluate the software capabilities. These metrics need to reflect the characteristics that they represent. They also need to allow appropriate measurements to be obtained, either through quantitative methods (e.g. by software tests/simulations, usability tests) or qualitative methods (e.g. through user observations).

Three types of classes of metrics are defined in this standard:

- Internal metrics associated with static internal properties of a system such as number of function calls, number of rules.
- External metrics associated with dynamic external properties. These are metrics that are observable when the user interacts with the system (i.e. the user performs a task/function/operation and observes the response in the sense of time required, results obtained etc.).
- Quality-in-use metrics, which refer to metrics that evaluate the extent to which a system meets the needs of the user.

Focusing on the usability requirements, various standardisation bodies have defined common formats to allow usability professionals, end users and software development teams to create usability requirements. Such kind of common formats are published by ISO in the ISO 25062:2006 standard for Common Industry Format for Usability Test Reports [7] and NIST in the NISTIR 7432: Common Industry Specification for Usability – Requirements (CIRU-R) [8].

2.2 The I-BiDaaS UCD approach

This section presents the approach that I-BiDaaS follows to create the interfaces that will be used by the envisaged stakeholders, taking into account the basic steps of the UCS process as previously described.

User involvement in the I-BiDaaS implementation phase includes the analysis of the usability aspects, the design of the interface components with usability requirements being part of the overall I-BiDaaS platform specifications and finally the usability evaluation of the integrated environment. The definition of the groups of end-users, their needs with respect to Big Data capturing, storage, management and analytics and the subsequent functional and non-functional requirements for the definition of the I-BiDaaS functionalities and the context of use of these functions by the target end users has already been presented in Deliverable D1.2 [10].

In this Deliverable, we emphasise on the visual representation of the I-BiDaaS concepts and the expected wireframes that should eventually constitute the user interface components of the I-BiDaaS integrated platform. The wireframes sketch a provisional view on how the defined end users will interact with the I-BiDaaS environment in order to manage their datasets and perform the offered analysis so as to extract useful insights. Therefore, taking into account the groups of end users, the use cases of each group that are to be accomplished via I-BiDaaS, the type of data to be collected, processed and visualised within the I-BiDaaS environment and the expected user requirements to be fulfilled by the I-BiDaaS platform, we evolve the design of the user interface wireframes around the following principles:

- Appropriateness: the wireframes should reflect the suitability of the interfaces to enable the interaction of the specific target end user groups (i.e. non-IT experts, data scientists etc.) with the I-BiDaaS environment.
- Operability: the wireframes should allow the end users to interact in an autonomous way, through self-explained interactive tools that provide easy operation and control.
- Flexibility: the wireframes should enable end users take the control of the workflow execution, through comprehensive navigation.
- Awareness: the wireframes should reflect the capability of the I-BiDaaS platform to allow the end users realise the status of the workflow and the consistency of the information offered through the interface functionalities.
- Efficiency: the wireframes should provide clarity on which is the primary action of each wireframe that should be accomplished by the end user.
- User interface aesthetics: the wireframes should be customised to the behaviour and philosophy of the end users using modern design elements, so that they are visually pleasant and attractive for use.
- Accessibility: UI development will be based on international standards and will target for various access channels (end user devices) as well as for as many users as possible.

It must be noted that relevant work in similar platforms that perform analytics on Big Data or offer machine learning as a service will be taken into consideration while developing the User Interface of I-BiDaaS. The EU-funded project Toreador[1] offers an interface to prepare, store, analyse and display results of analysis on given datasets. As analysed in the following paragraphs, the stepwise process followed in the UI of Toreador is similar to the one that I-BiDaaS will offer. However, I-BiDaaS aims at providing an advanced visualisation framework based on new innovative tools and established commercial frameworks, i.e. AEGIS' Advanced Visualisation Toolkit and SAG's MashZone. Furthermore, the goal to address the needs of both non-IT and IT professionals through the same platform differentiates I-BiDaaS' interface elements with respect to mentioned Toreador project as well as with other mostly IT-expert aiming platforms offering machine learning as a service like Mantra[2] (A Deep Learning Development Kit) or Seldon[3] (Open source platform for deploying machine learning models on Kubernetes). Nevertheless, visualisation options used in the aforementioned or similar tools/platforms will act as useful inspirations during the discussions with users on what is the best fit for their needs.

[1] http://www.toreador-project.eu/

[2] https://github.com/RJT1990/mantra

[3] https://www.seldon.io/

3 The I-BiDaaS platform

3.1 Overview of the platform functionalities

The I-BiDaaS platform aims at helping users working with very high volumes of rapidly increasing, versatile data to address problems that cross sectorial borders (e.g. customer analysis, fraud detection, operational efficiency, etc.), but also improve current operations that assist in decision making. Therefore, by offering efficient Big Data analytics solutions and an easy to use, secure environment, I-BiDaaS will serve as an enabler of advanced Big Data analysis towards improved industrial decision making and problem solving.

I-BiDaaS will develop and evaluate the proposed solution in the three pilot domains of telecommunications, finance and manufacturing. The generic system requirements that have been identified from the respective end-users during the requirements elicitation phase have been analytically described in D1.3 [9] and repeated below so as to complete the context of the current deliverable.

Table 1. System Requirements

No.	System requirement
FR1	The system should enable the generation of anonymized and synthetic data to enable safe experimentation and testing
FR2	The system should enable aggregation of both attribute level and transaction level data coming from a variety of internal data sources and in multiple formats
FR3	The system should be able to accommodate data sets that feature high volume, high velocity, high variety, high variability, high volatility and high data sparsity
NFR1	The system should ensure security of sensitive data
FR4	The system should support diversified, analytic processing, machine learning and decision-support techniques to support multiple stages of analysis
FR5	The system should support interactive data analysis
NFR2	The system should support near real time analytics performance
FR6	The system should support diversified visualization and interaction of results on both desktop and mobile devices
NFR3	The system should support near real-time updating of results
NFR4	The system should support multilevel access control at resource and application level

Based on these requirements, the core workflows that cover the functionalities for the identified user roles are described in the following paragraph.

3.2 End User Roles and Workflows

The main end users of I-BiDaaS that will access the platform interface include non-technical business end-users (strategic or operational managers), who only consume the analytics results, as well as more "technical" roles that configure analytic services and data flows (subject matter experts, data scientists). In this section, we present the basic workflows for the groups that will be using the I-BiDaaS platform interface to perform the desired tasks.

The workflows have been derived from the system requirements described in the previous paragraph and after initial discussions with end users, according to the I-BiDaaS UCD approach which dictates early involvement of the users to the prototype design. The core workflow is presented in Figure 3 and describes the steps that users follow when using the I-BiDaaS interface to load datasets, perform analysis and explore visualised results to discover hidden relationships.

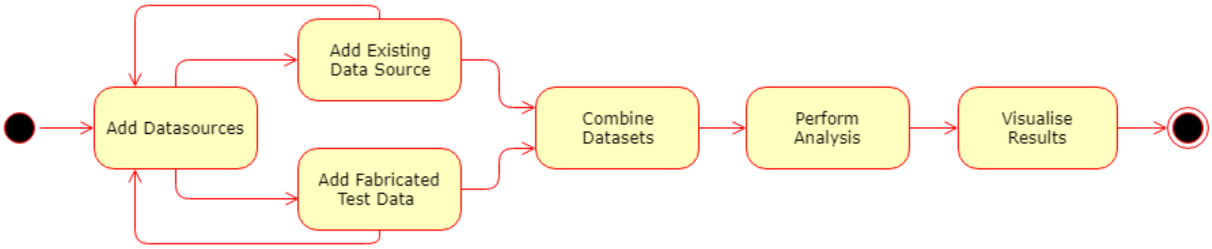


Figure 3. Core Data Analysis Workflow

Users first select the data sources to add. Existing data-sources or fabricated test data can be added. The latter case ensures confidentiality of private data while ensuring realistic and meaningful experimentation with high quality test data. After this, users can combine data and then select the analytics algorithms to perform experiments and extract analytics. At the end, visualisation of results helps users review results and explore extracted information to gain meaningful insights.

The next figure (Figure 4) analyses the activity of analysis which involves a set of sub-activities to cover the needs of expert users that want to further exploit the capabilities of the platform by using their own expertise and former knowledge on Big Data manipulation and analysis.

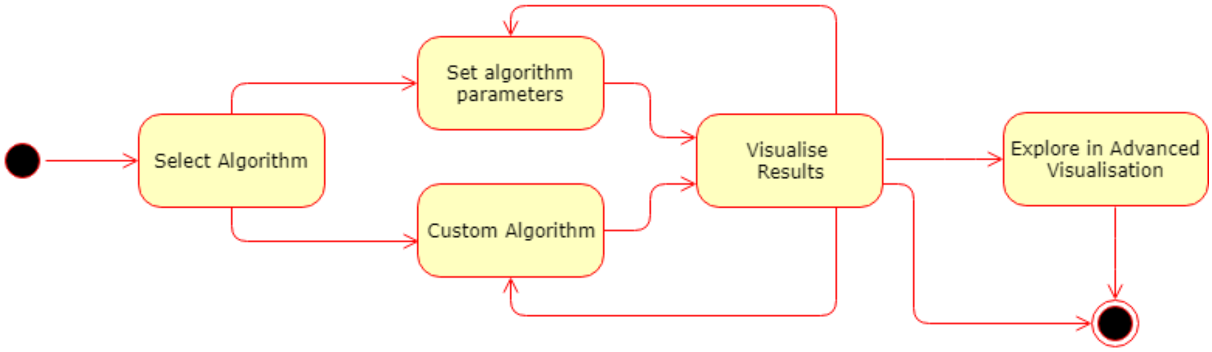


Figure 4. Data Analysis Workflow

The selection of an algorithm to execute on the defined datasets can follow two approaches. The first one, targeting non-experts, lets users select an algorithm from a predefined list offered by I-BiDaaS. These algorithms include state-of-the-art machine learning algorithms as well as traditional analytics procedures that can offer users with a rich set of Big Data analysis functionalities and relevant results. On the other hand, expert users (e.g. data scientists) can take existing algorithms and update them to better match their needs or even load their own implementations of algorithms that can be used to perform advanced analytical tasks. This way extended or specialised versions of current algorithms can derive from such experimentation and enrich the available functions for new datasets of possibly different domains. The possibility of visualising the results and returning to algorithm customisation to change options will foster the latter process and help users identify the most suitable algorithm and its settings for a specific task for given data. Furthermore, visualisation of results initially follows a standard path to represent data results from the analysis. Users with greater expertise and familiarity with advances visualisation can further explore data using more advanced visualisation and try to correlate various dimensions so as to reveal valuable insights.

In the next sections we describe the visualisation framework and the multipurpose interface that will be employed to cover the required functionalities for the presented workflows.

4 The visualisation and monitoring framework

4.1 Advanced Visualisation Tookit (AVT)

The constantly increasing amount of data generated via diverse sources, like sensors or data analytics processes, result in a vast amount of information that needs to be analysed in a quick and intuitive way in order to expose the hidden insights. Traditional or innovative ways of presentations must be deployed to generate reports, figures and dashboards that will allow users to correctly interpret data and understand their correlations. Towards this goal, efficient exploration, searching, filtering and aggregation of data in multiple scales (e.g. time, geospatial and other) could be required. This procedure can be quite cumbersome with current visualisation tools that tend to be stove-piped, making it difficult to take information seen in one visualisation tool and obtain a different perspective in another tool. If an interesting relationship is observed, needing to be explored in more depth, the process must be reiterated by manually generating a subset of the data, converting it into the correct format and invoking the new application. The AEGIS Advanced Visualization Toolkit (AVT) consists of a set of interactive visualisation tools developed to allow for a more straightforward exploratory analysis (Figure 5). The selection of the tools and subsequently the definition of the interactions among them depend on the domain and on the end-users' requirements.

AVT can support scalable data visualisation approaches and tools that will enable easy

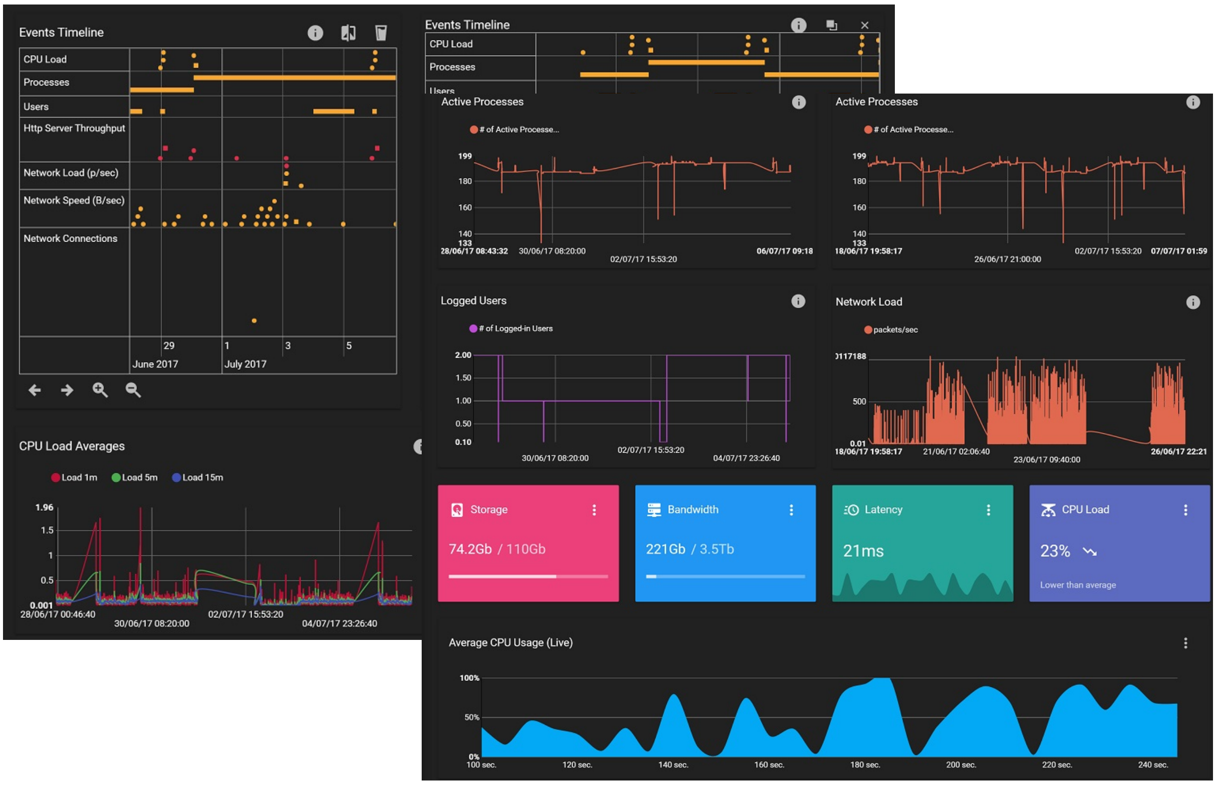


Figure 5. AVT Dashboard

transition from one scale to another or from one form of aggregation to another, e.g. from averages of an indicator per day to total numbers of the same indicator per city or region. Moreover, real-time data analysis, interactive presentations, shareable dashboards and collaborative capabilities are some of the key features that can be utilised. Technologies that support a visualisation framework with the qualities described can vary and they are usually used combined so as to cover as many of the user requirements as possible. Within I-BiDaaS, we will be mainly based on the usage of libraries that produce data visualisations for web browsers, therefore also addressing the need to access data via multiple devices. These are mainly Javascript libraries which include lower-level data manipulation frameworks like D3 (https://d3js.org/) or charting libraries which offer a large number of configurable charts to be used in an interactive and backend-agnostic visualisation environment like e.g. Highcharts (https://www.highcharts.com). Of course, there are also other libraries with special features that can be potentially added to the available tools, e.g. Vis.js (http://visjs.org/) for graph-based visualisations or plotly (https://plot.ly/) for an emphasis to 3D visualisations. Moreover, using the WebGL API (https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API) or a WebGL based library (e.g. https://threejs.org/) can enable AVT to use the GPU of a computer to render 2D/3D graphics on the browser with high performance. AVT is flexible enough to employ one of the above solutions or a combination of them in order to achieve a result that serves the data and user needs in the best possible way.

Except from determining the right visualisation technologies to support the needs of the (IT and non- IT) users of I-BiDaaS, configuration and collaboration features of the tools will also play an important role for data exploration, discovery and querying. These features will be offered via intuitive mechanisms to share visualisations, configurable chart representations of datasets and advanced filtering capabilities offered by the AVT. Such mechanisms together with the capability to use different visualisations in terms of underlying technologies and final output makes AVT a versatile solution that can address different categories of users. Based on the level of expertise of the potential users, the visualisations toolkit can provide different levels of data abstractions and represent them in suitable graphical elements that can be tailored for both non-IT and IT users. Non-IT users will be able to see the data in easily comprehendible ways that will expose key aspects of the visualised information. IT experts will be offered the tools to drill down the data and experiment with advanced visualisations that can help with data exploration and drawing of useful insights such as hidden relationships, patterns or outliers in big volumes of multidimensional data.

Two innovative aspects of the AVT, whose applicability on the I-BiDaaS use-cases will be examined, are the timeline analysis and the preconfigured views:

- Timeline analysis (Figure 6) offers the ability to “travel back in time” and compare the current view of a dataset (which contains data mapped to a time dimension) with similar views of data that were generated in the past. This allows the new data to be compared against patterns encountered before. In this way, new relationships may be identified from re-occurring patterns and end users can discover meaningful correlations that would be otherwise hard to reveal. The example in Figure 6 shows how a dataset containing alerts for off-limit usage of various server metrics (like CPU load, network load, etc.) can be visualised on the timeline graph and immediately reveal that many of the alerts were issued at the same day and time, thus giving the user a hint to further investigate what happened at that time.
- Preconfigured views provide the ability to adapt the display of information based on previously encountered situations. For example, if a data analyst has created a specific “view” consisting of multiple data sources (e.g. using specific relationships

to group and associate datasets) and presentation modes (e.g. specific graphs, charts, etc.) to explore a particular set of data in the past, this view can be saved and reused, either manually or automatically, to present data associated with a new case, or a data stream that is currently being generated.

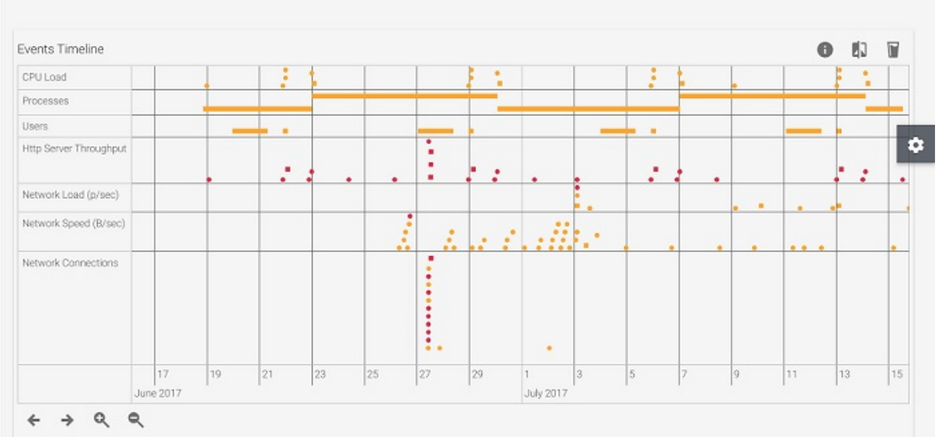


Figure 6. Timeline analysis

Combined, these two aspects of AVT, allow the end user to quickly gain a solid understanding of one dataset (or more) and benefit from existing stored knowledge. Information may be automatically presented in a way that enhances situational awareness and allows the analyst to concentrate on the analysis and exploration rather than the configuration of the visualization system. In addition, the need for technical support at the end-user level is minimized which can be particularly beneficial in forwarding deployments.

AVT follows an internal 3-tier architecture, namely: data, business and presentation as seen in Figure 7. This architecture provides the flexibility to adapt to various domains and data sources.

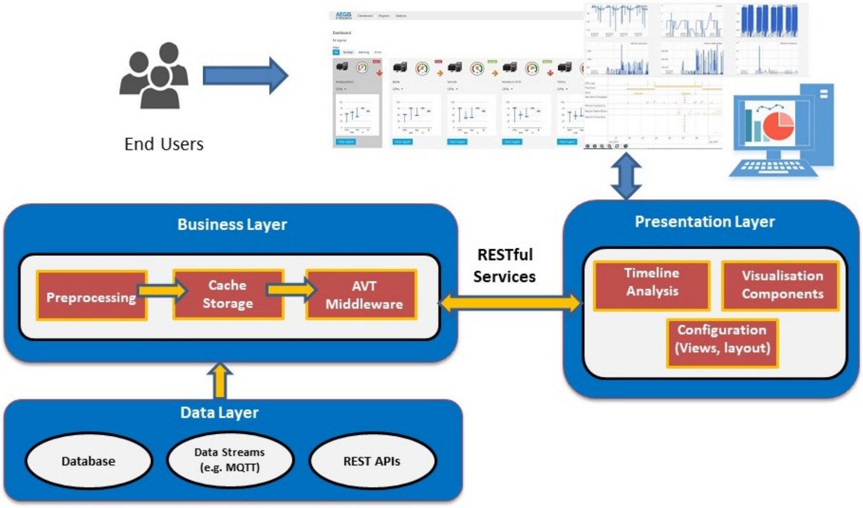


Figure 7. AVT Internal High-Level Architecture

The Data layer includes the original data and the mechanisms that enable their retrieval by the business layer of the AVT. The data include several Indicators that are domain specific and can be calculated from the original data; e.g., for a yearly log of telephone calls an Indicator could be the number of calls per day for the given year. Being domain specific, the Indicators and the means to calculate and retrieve them will be developed within the context of the relevant task of the project (T2.4 – Design and development of interactive visualization tools).

The business layer of AVT handles the processing of the received data in a format suitable for the visualisations. Possibly required filtering, aggregation or any other data transformations take place in this layer. A cache mechanism is also available to enhance performance. Finally, this layer serves the REST services required by the visualisations to offer their functionalities.

The presentation layer includes the timeline analysis component and a set of multiple visualisations to support the various kinds of representations. The timeline analysis component offers the functionalities for temporal analysis as mentioned before. Through the timeline control, the end user may move back in time, narrow the viewed time window, and compare two different time points, in order to get insights of the data. If present, the timeline control drives the displayed data in all the other visualisations of the currently opened view inside the AVT by propagating the selected timeframes to the rest of the visualisations. These visualisations include a set of interactive graphs and charts that form the heart of the AVT. Bar charts, line charts and pie charts are some of the standard forms of data representations, whereas geographical maps, tree maps, chord diagrams or linked graphs are some of the most advanced types of visualisations that can be put in place. Different visualizations are used to display different types of data, in order to make the understanding of them easier and help users uncover hidden relationships.

4.2 MashZone NextGen

MashZone NextGen[4] (MashZone) visual analytics software provides a fast way to explore and analyze both real-time (e.g. streaming data) and data at-rest. Data from multiple sources like e.g. Apama, spreadsheets, databases, flat files, etc. can be connected to MashZone for analysis at any time. Accessing the original data directly from the source lets business users respond to changing conditions as they happen. The connection of multiple data sources is afterwards combined in dashboards comprising analytics widgets that visualise data coming from the linked data sources in real-time or at configurable time intervals. This way, meaningful visualisations of raw data can be created by end users with no need for IT knowledge and help them get meaningful insights using any kind of device to access the information.

The key features of MashZone can be summarised in the following list:

- "Point and click" to build dashboards
- Intuitive user interface to identify visual patterns quickly
- Live, multi-source connectivity
- Real-time interactive dashboards
- Extensible, embeddable and secure

Figure 8 below depicts an overview of MashZone architecture having the data sources on the left providing data feeds to the MashZone interface, which in turn facilitates dashboard creation to reveal insights via web interfaces running in web or mobile browsers that are shown in the right side of the schema.

[4] https://www.softwareag.com/corporate/products/apama_webmethods/mashzone_nextgen/default

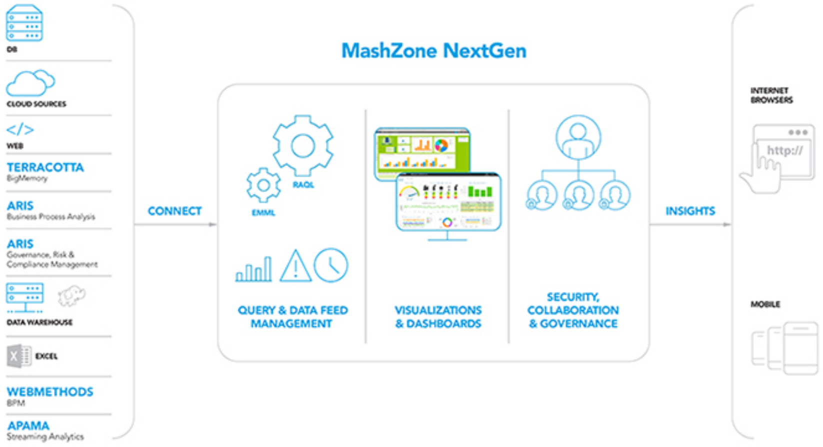
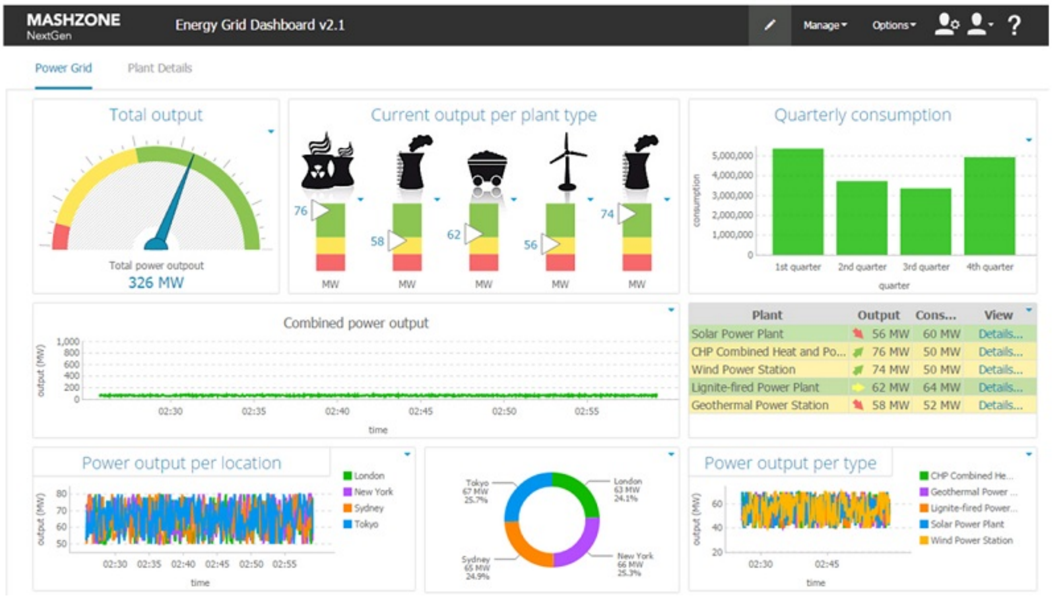


Figure 8. MashZone Overview

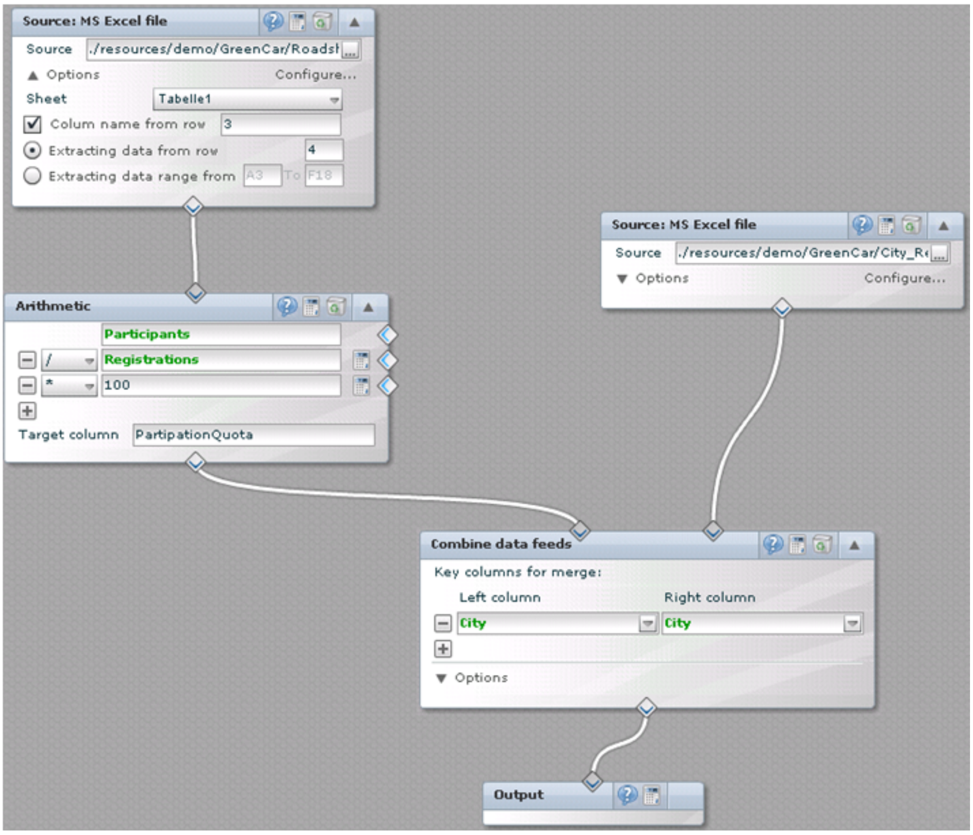
The main concepts to achieve the described functionality are dashboards and data feeds. A dashboard is an interactive application that collects data from different data sources, combines it, and visualizes it. The data can come from different sources. Possible data sources include Excel or CSV files, reports from ERP or CRM systems, queries from data warehouses, or freely available, machine-readable data from the Internet. Dashboards are composed of individual graphical components (e.g., business graphics, tables, maps, etc.), which obtain their data from data feeds and display it. Users can combine the individual display components to filter the displayed results interactively and thus analyse them intuitively.



Plant	Output	Cons...	View
Solar Power Plant	56 MW	60 MW	Details...
CHP Combined Heat and Po...	76 MW	50 MW	Details...
Wind Power Station	74 MW	50 MW	Details...
Lignite-fired Power Plant	62 MW	64 MW	Details...
Geothermal Power Station	58 MW	52 MW	Details...

Figure 9. Mashzone Dashboard Example

A data feed is a table containing prepared data, which is accessed by the individual display components of a dashboard. A data feed consists of several columns that contain numerical values (e.g., figures), text, or date values. Each row in the calculated result of a data feed corresponds to one data record. The data in a data feed is calculated based on various data sources (e.g., data from MS Excel, CSV, or XML files) using feed definitions. Feed definitions aggregate, extend, transform, or calculate data from one or more data sources. A feed definition can consist of any number of operators and data sources, which are linked together using connections. Data is calculated for each data source and each operator and then passed on to the operators linked to them for further processing. A feed definition delivers a data structure in the form of a list table as its result. All individual processing steps in the feed definition are based on this data structure. The source data is not held redundantly in the data feed, but remains in its original sources, ensuring that it is constantly up to date. In addition to the external data sources, direct user entries in the data feeds can also be processed. Only one data feed can be assigned to each display component, with the same data feed being able to supply the data for several display components. An online drag and drop editor is available to users to create their data feeds as depicted in Figure 10 below:



Source: MS Excel file
Source ./resources/demo/GreenCar/Roadst
▲ Options Configure...
Sheet Tabelle1
☑ Colum name from row 3
● Extracting data from row 4
○ Extracting data range from A3 To F18

Arithmetic
Participants
/ Registrations
* 100
Target column PartipationQuota

Source: MS Excel file
Source ./resources/demo/GreenCar/City_R
▼ Options Configure...

Combine data feeds
Key columns for merge:

Left column	Right column
City	City

▼ Options

Output

Figure 10. MashZone Data Feed Editor

The visual components that are connected to a data feed and included in a dashboard are referred to as widgets. MashZone ships with a variety of widgets out of the box including different chart types, speedometers, image and text widgets. Additionally, it provides an API to creation custom visualizations. These "custom widgets" are integrated in MashZone and behave like native widgets. They consume the same data sources and support all the main functionality of native widgets, for example, selection handling, emitting and receiving filter events or the configuration of actions. The base technologies used to create custom widgets for MashZone are HTML, CSS (LESS) and Javascript (AngularJS).

The development of new custom widgets in the context of I-BidaaS will offer a great opportunity to offer innovative visualisations and rich data representations to users while keeping the easy process of “point and click” creation of dashboards that MashZone can offer.

4.3 Integration of AVT and MashZone

As mentioned in the previous paragraph, MashZone allows the creation of custom widgets based on a specified development framework[5] which relies on HTML, CSS and AngularJS. This fits nicely with AEGIS' AVT which is also developed using AngularJS as the basic frontend technology. Therefore, as briefly mentioned in D1.3 [9], AVT's visualisation components can be ported to MashZone as custom widgets that will allow the creation of complex dashboards satisfying user needs. This way, data feeds that will be created using the MashZone facilities will have the option to be connected to the custom widgets offering the advanced visualisations of AVT, such as the timeline component.

The conceptual representation of the integration method described above is presented in the following figure, showing how could the timeline component of AVT be integrated and offered as a custom widget within MashZone.

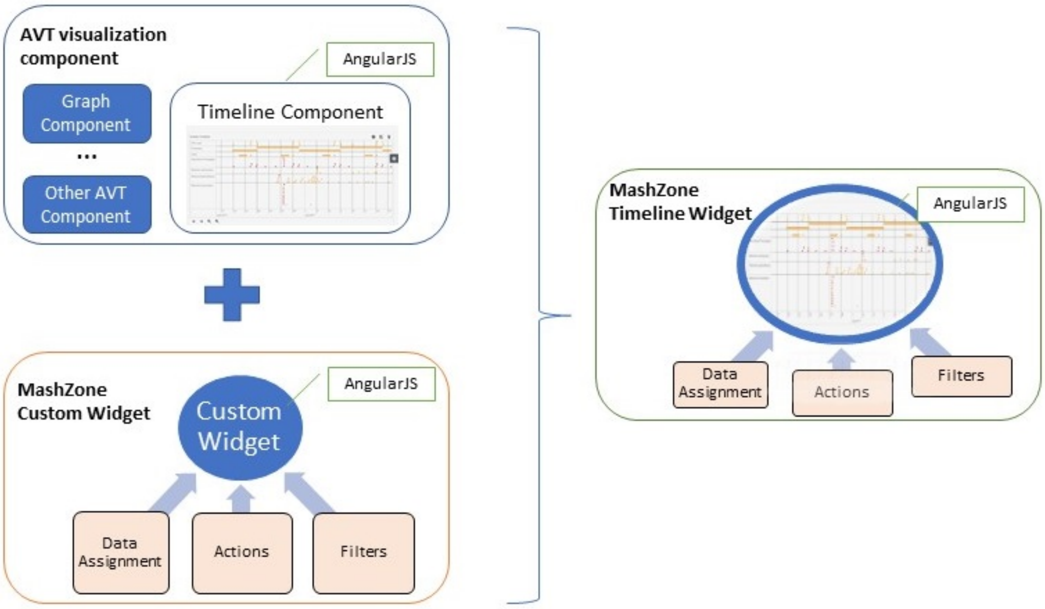


Figure 11. AVT-MashZone integration

4.4 Communication with other components

The visualization framework will be built based on the input received by the rest of the components of the I-BidDaaS platform. This communication is foreseen to take place via a service-oriented approach where each component exposes its outputs via web services that can be afterwards consumed by the visualisation framework. The selected architectural style is REST and the next paragraphs present a high-level description of the service signatures that are going to be available. It must be noted that since the project is still at an early stage, additions of new services and updates to the existing ones are expected as development advances and user needs evolve.

[5] http://techcommunity.softwareag.com/web/guest/pwiki/-/wiki/Main/Custom+Widgets+Introduction+and+Prerequisites

4.4.1 Resource management and orchestration module

Function Name	Description	Input	Output
Create deployment	Interpret the blueprint manifest file to provide the best placement	Blueprint manifest file describing the resource requirements and their relationships	Success or Failure
Destroy deployment	Destroys all entities created for the deployment and frees any resources that are tied to this deployment.	Blueprint manifest file	Success or Failure
Scale down	Scales down the resources used by the deployment	N/A	Success or Failure
Scale up	Scales up the resources used by the deployment	N/A	Success or Failure

4.4.2 Test Data Fabrication

Function Name	Description	Input	Output
Start Data Fabrication	Fabricates the data based on the given model	Data Model, Fabrication parameters	Fabricated Data
Create Data Model	Generates a data model from the given database	Structure, rules, fabrication configuration	Data Model
Load Data Model	Loads the given data model	Data Model	Success or Failure

4.4.3 Batch Processing - Advanced ML module

Function Name	Description	Input	Output
List Available Algorithms	KNN classification, K Means clustering, Decision Tree classification, Random Forest classification and ADMM LASSO for sparse regression	N/A	A list of algorithms
Get Algorithm Parameters	Each algorithm has its own set of parameters (for KNN: *K* – the number of nearest neighbours to consider, *num* – number of chunks the data is being splitted for processing. K Means: *K* – number of clusters to create, *max_iter* – maximum number of iterations before terminating the training, *tol* – allowed distance tolerance for terminating the training and *num* – number of chunks the data is being splitted for processing. LASSO: *lmbd* – lambda parameter for the optimization problem, *rho* – rho parameter for the optimization problem, *itter* - maximum number of iterations before terminating the algorithm, *reltol* – relative tolerance for terminating the algorithm, *abstol* – absolute tolerance for terminating the algorithm, *num* – number of chunks the data is being splitted for processing. Decision Tree: *max_depth* – maximum allowed depth of the tree, *min_sample* – minimum allowed number of samples in a leaf, *impurity_tol* – allowed entropy in a leaf node. Random Forest: *n_estimators* – number of trees to create for a model, *n_features* – number of column to use for each tree, *max_depth* – maximum allowed depth of the tree,	Algorithm ID or name	A list of parameters (data type, range, etc)

	min_sample – minimum allowed number of samples in a leaf, *impurity_tol* – allowed entropy in a leaf node.)		
fit[6]	Trains (fits) the specified algorithm with a given parameters and input data	Data, Algorithm parameters	Data results
predict	Predicts (transforms) an input data with respect to the train phase of the specified algorithm	Data	Data results
fit_predict	Simultaneously performs fit and predict methods (equivalent to invoking .fit( ).predict( ) on the same data set)	Data, Algorithm parameters	Data results

4.4.4 COMPSs

Function Name	Description	Input	Output
Get job info	Returns information and statistics about the distributed execution of the job	Job ID	Dependency graph, statistics

4.4.5 Hecuba module

Function Name	Description	Input	Output
Store stream	Tells Hecuba to listen for a topic and to store events into Cassandra	Channel ID, Event data format and table destination	
Managed stream persistence	Manages the event persistence		Stream status/ insertion statistics

4.4.6 Apama module, communication via Universal Messaging [7]

Function Name	Description	Input	Output
Subscribe to channel	Allows a listener process to subscribe to a channel and receive all its messages	Channel ID	Data Stream (messages), normally in JSON format
Publish to channel	Allows a writer process to send messages to a channel	Channel ID	Data Stream (messages), normally in JSON format

4.4.7 Integrated platform

Function Name	Description	Input	Output
Create project	Creates a new project for the current user	Project Title	Project ID

[6] All the algorithms in the algorithm pool have the same functions (methods) – fit, predict and fit_predict.

[7] Apama can easily connect to SAG's Universal Messaging component either with the MQTT or the UM connectivity adapter, this should be the easiest and preferred way to communicate with Apama.

List Projects	Returns the list of projects for the given user	User ID	List of projects
Add User Dataset	Adds given dataset to the list of the user's datasets	Dataset location, User ID	Dataset ID
Get User Datasets	Returns the list of datasets for the given user	User ID	List of datasets
Save Project Configuration	Saves all selected options of the user for the given project	User ID, Project ID, Datasets List, Algorithm ID, Algorithm Parameters	Project Configuration ID
Run experiment	Deploys and runs a project configuration	User ID, Project Configuration ID	Experiment ID
Get project experiments	Returns the list of experiments for the given project	User ID, Project ID	List of experiments
Download Results	Exports the results of an executed experiment in the given format	Experiment ID, format	Data Results in requested format
Delete project	Deletes the given project	Project ID	Deletion status
Delete dataset	Deletes the given dataset	Dataset ID	Deletion status
Delete project configuration	Deletes the given project configuration	Project Configuration ID	Deletion status
Delete experiment	Deletes the given experiment	Experiment ID	Deletion status

5 The multipurpose interface

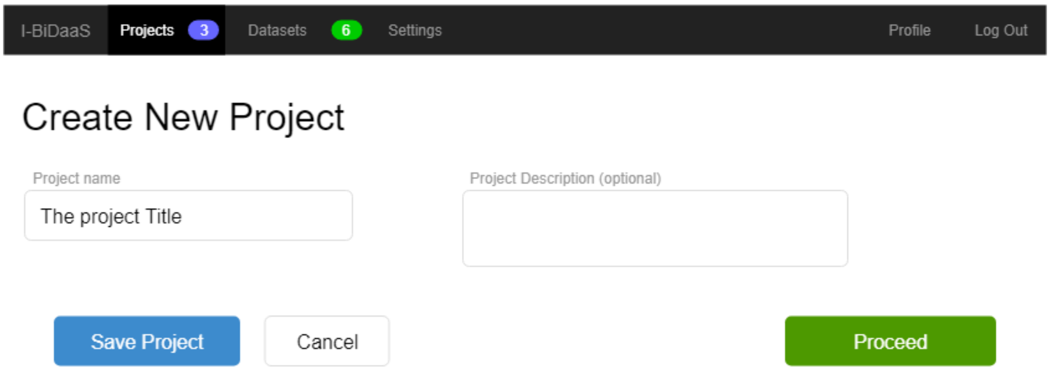
The goal of the multipurpose interface of the I-BiDaaS platform is to provide functionalities and data analytics in different levels of abstractions, tailored for the different categories of the target user groups. By using a business-style dashboard, the functionalities of the platform that are required to cover the needs of the various user roles will be organised and presented in an easy to use and highly operational manner.

Following the workflows described in paragraph 3.2, a set of wireframes have been created to depict the way that the various functional elements will be presented to the user. The wireframes present the core elements required to fulfil the specified functionalities and will be, of course, enhanced with additional elements and auxiliary functions as the work progresses towards the first integrated prototype.

Screenshots of the first working prototype follow the wireframes section and present the current status of both the interface and visualisation framework prototype. The implementation has mainly focused on providing the functionalities needed for the MVP as they have been described in D5.2 [11].

5.1 Wireframes

Figure 12 presents the project setup page. The user provides the project title and optionally a description for the project. The next step is to define the datasets.



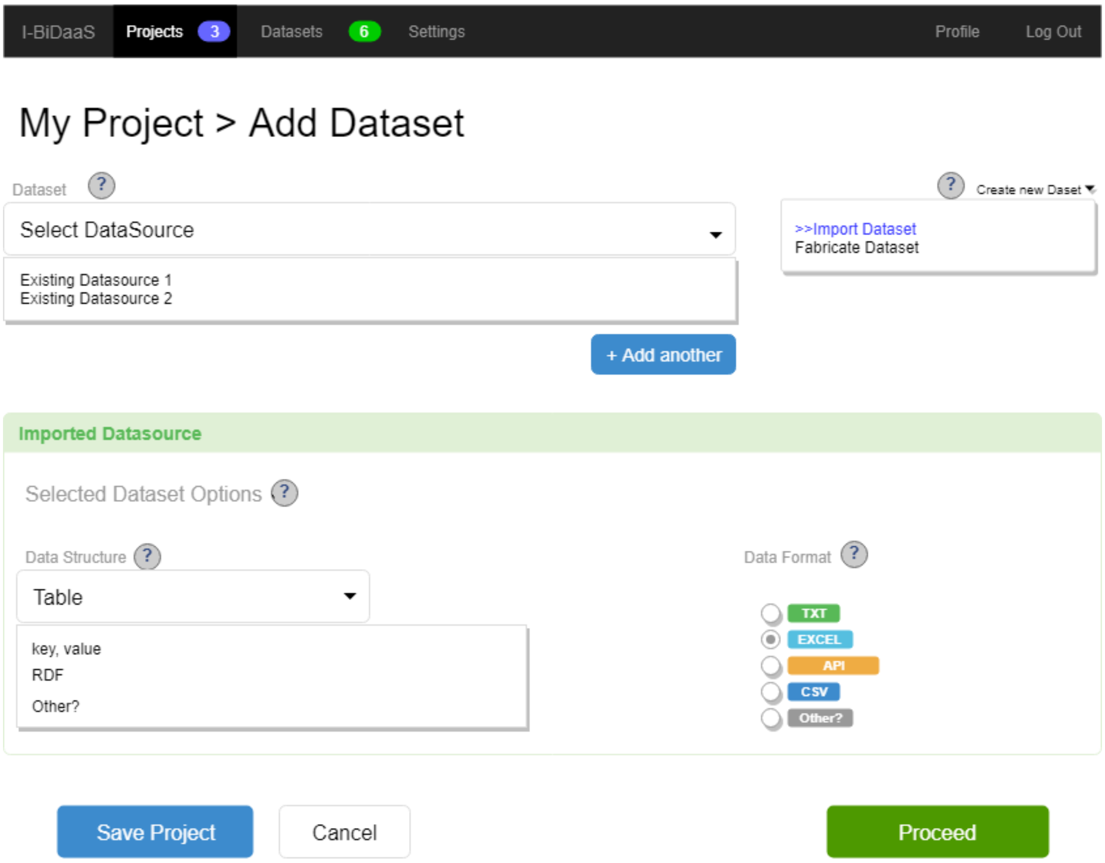
Create New Project

Project name: The project Title

Project Description (optional):

Figure 12. Create Project

The next screen allows users to add one or more datasets to their project. Existing data sources will potentially include a number of supported formats that will be derived from the realisation of the pilot use cases and the user needs.



My Project > Add Dataset

Dataset ?

Select DataSource

Existing Datasource 1
Existing Datasource 2

? Create new Daset

>>Import Dataset
Fabricate Dataset

Imported Datasource

Selected Dataset Options ?

Data Structure ?

Table

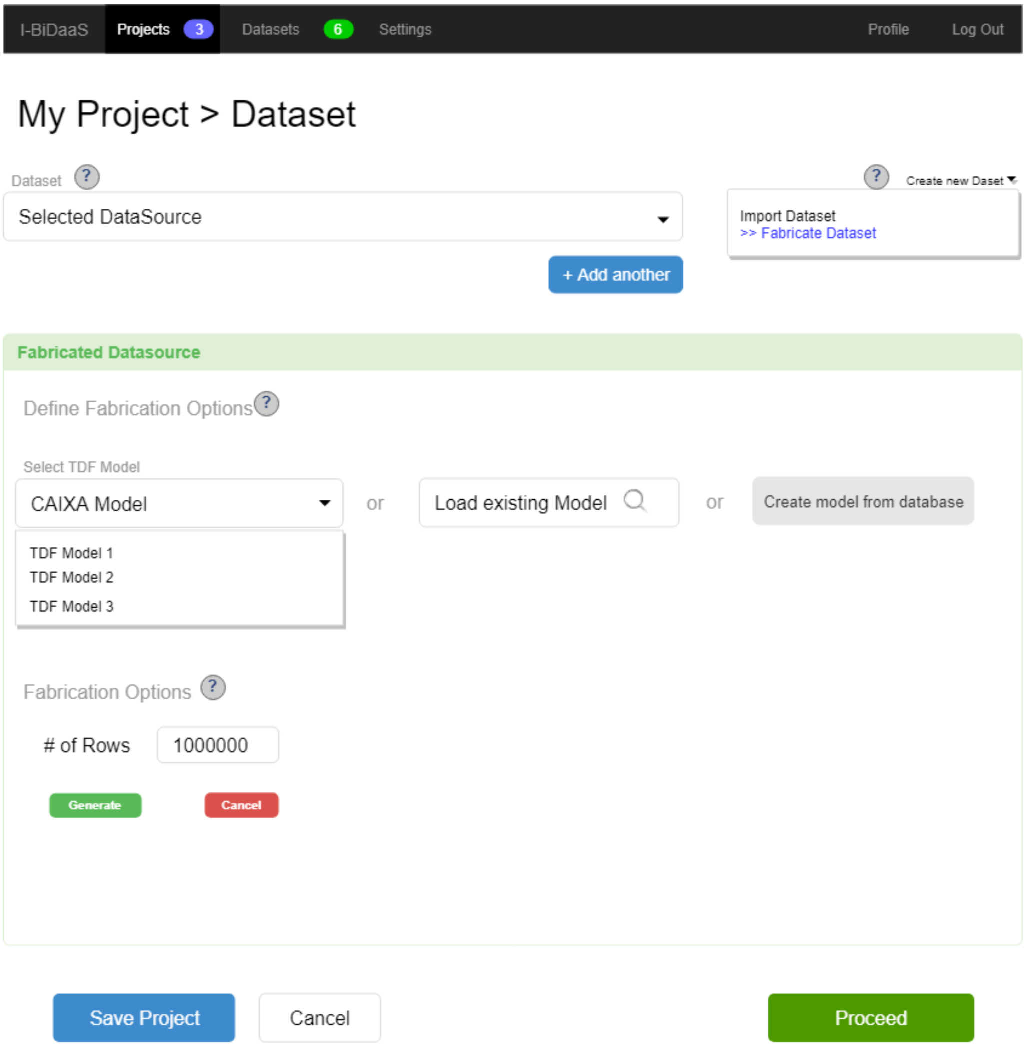
key, value
RDF
Other?

Data Format ?

○ TXT
● EXCEL
○ API
○ CSV
○ Other?

Figure 13. Add Dataset(s)

When the ‘Fabricate Dataset’ option is selected, users are presented with a set of options regarding the test data fabrication. These options include the selection of the data model that will be used to fabricate data and then the size of data that should be fabricated. The model can either be one of the preloaded ones or users can load their own or create one on the fly. These options are depicted in Figure 14 below:



My Project > Dataset

Dataset: Selected DataSource

Fabricated Datasource

Define Fabrication Options

Select TDF Model: CAIXA Model

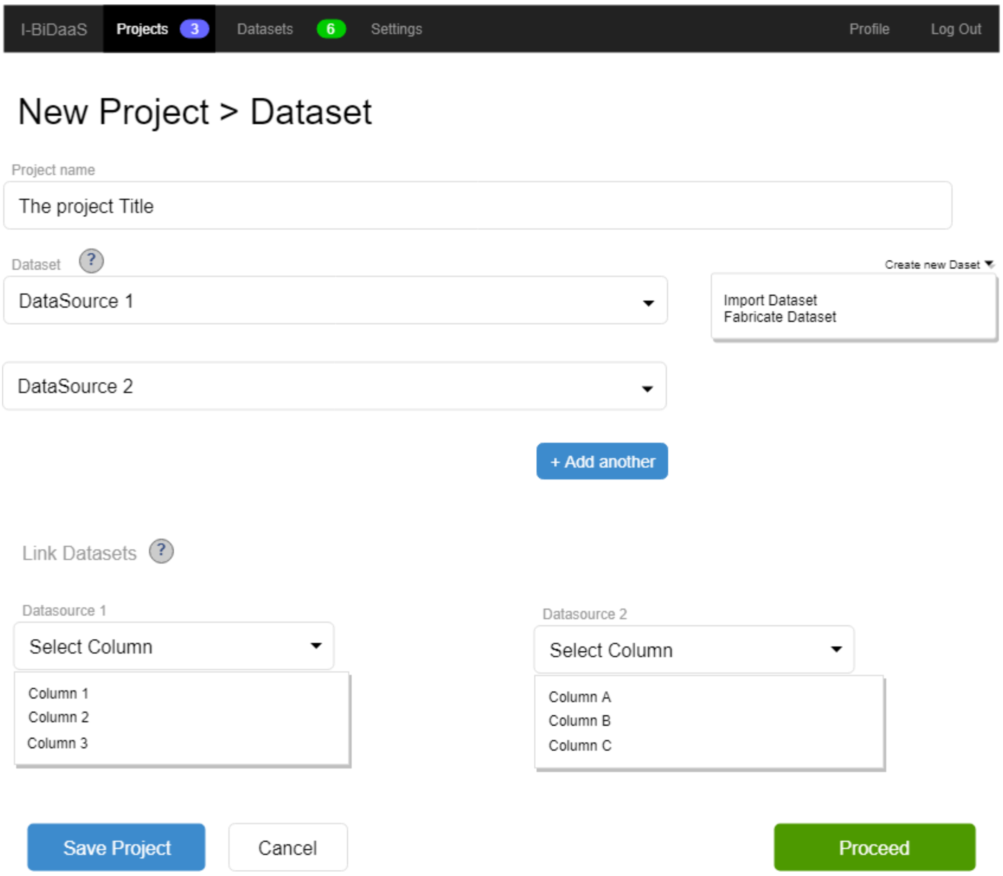
Load existing Model

Fabrication Options

# of Rows: 1000000

Figure 14. Fabricate Dataset

If the users select more than one datasets, they have to provide the connection among them. This can be done by e.g. selecting the data elements that represent the same entity. For example, the columns were the joining between two tables should happen. This functionality implies a very good knowledge of the dataset structures and can also expose high complexity if dataset formats are significantly different. Nevertheless, the following wireframe presents one of the cases and a more detailed approach to the matter will be analysed in future deliverables where the implementation of the use cases will be of higher maturity.



New Project > Dataset

Project name: The project Title

Dataset: DataSource 1

DataSource 2

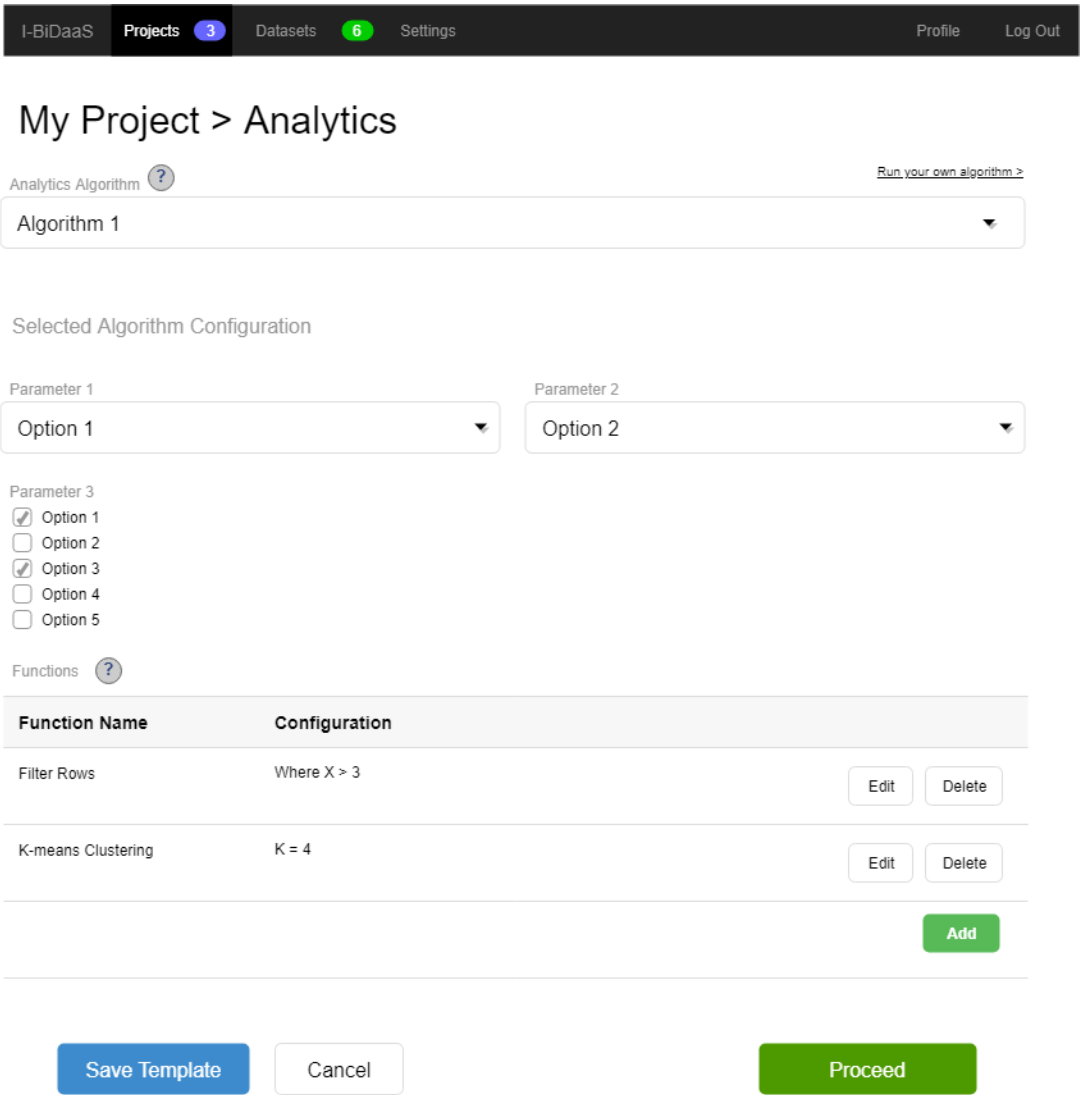
Link Datasets

Datasource 1: Select Column

Datasource 2: Select Column

Figure 15. Dataset linking

When the selection of datasets is finished, users have to select the algorithm they want to use to run their experiment. Every algorithm includes a set of configurable parameters which defines its execution and affects the generated results. Users have to define these parameters (or leave the defaults) and then proceed to executing the experiment.



My Project > Analytics

Analytics Algorithm ?

Algorithm 1

Selected Algorithm Configuration

Parameter 1: Option 1

Parameter 2: Option 2

Parameter 3
- ☑ Option 1
- ☐ Option 2
- ☑ Option 3
- ☐ Option 4
- ☐ Option 5

Functions ?

Function Name	Configuration
Filter Rows	Where X > 3
K-means Clustering	K = 4

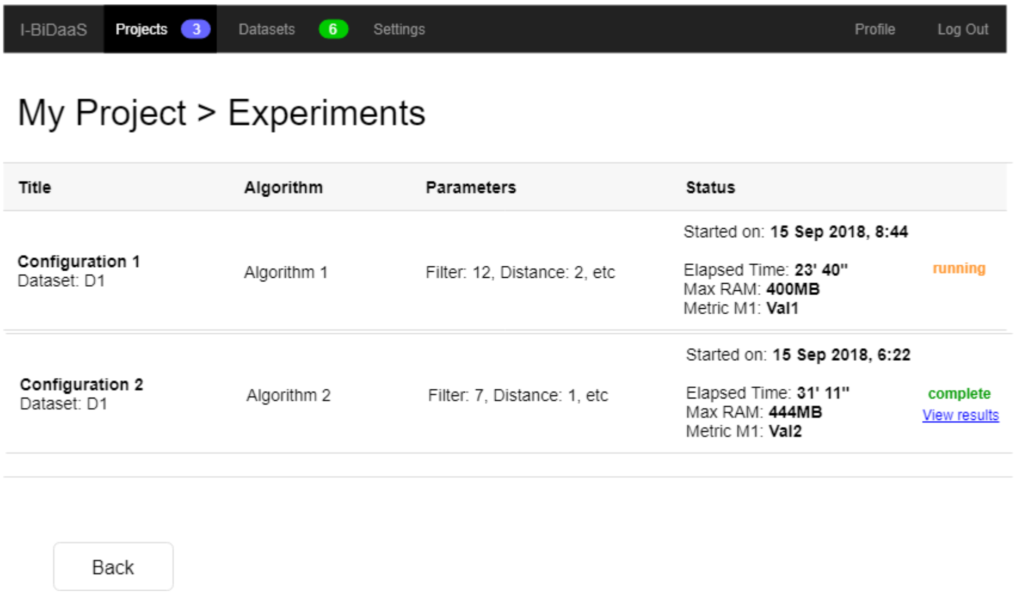
Figure 16. Algorithm Selection

It must be noted that expert users can follow a different option and setup their own algorithm on the given data as presented in the relevant workflow. Solutions like the cloud-based Collaboratory[8] or JupyterLab[9] will be examined so as to provide a solution that lets experts write and run the entire script inside I-BiDaaS platform, thus utilising the offered resources and visualisation capabilities.

Running an experiment on big volumes of data may require significant amount of time to get completed. Therefore, when users setup their experiments the interface provides an overview screen where they can monitor the execution status and get information retrieved by the underlying components that are responsible to run the experiment. This information may include the elapsed execution time, the maximum amount of RAM consumed so far and other metrics that will be made available by the components. This way users get a quick update on their experiments and can select to see the final results for the ones that have finished.

[8] https://colab.research.google.com

[9] https://jupyterlab.readthedocs.io/en/latest/

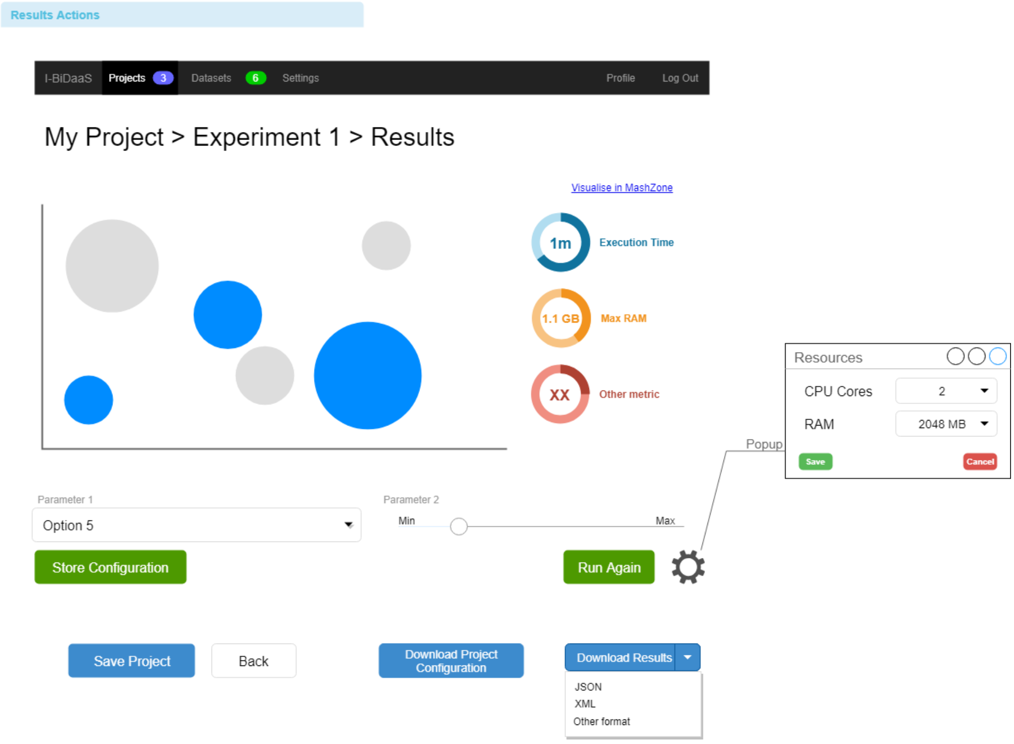


My Project > Experiments

Title	Algorithm	Parameters	Status	
Configuration 1 Dataset: D1	Algorithm 1	Filter: 12, Distance: 2, etc	Started on: **15 Sep 2018, 8:44** Elapsed Time: **23' 40"** Max RAM: **400MB** Metric M1: **Val1**	running
Configuration 2 Dataset: D1	Algorithm 2	Filter: 7, Distance: 1, etc	Started on: **15 Sep 2018, 6:22** Elapsed Time: **31' 11"** Max RAM: **444MB** Metric M1: **Val2**	complete View results

Figure 17. Project Experiments

When an experiment is over, users can go to the results page and start exploring the outcome. This page includes visualisations of the results and also the possibility to open up the advanced visualisation framework to explore more advanced visualisation inside the environment of MashZone. Furthermore, in order to foster productivity and effectiveness of the visualisation tool, parameters and even algorithms can be updated in this page and saved as different configurations. This way, users can quickly see the results in the case of fast executing experiments or easily change parameter values without leaving the page for time consuming experiments. Figure 18 shows these options:



My Project > Experiment 1 > Results

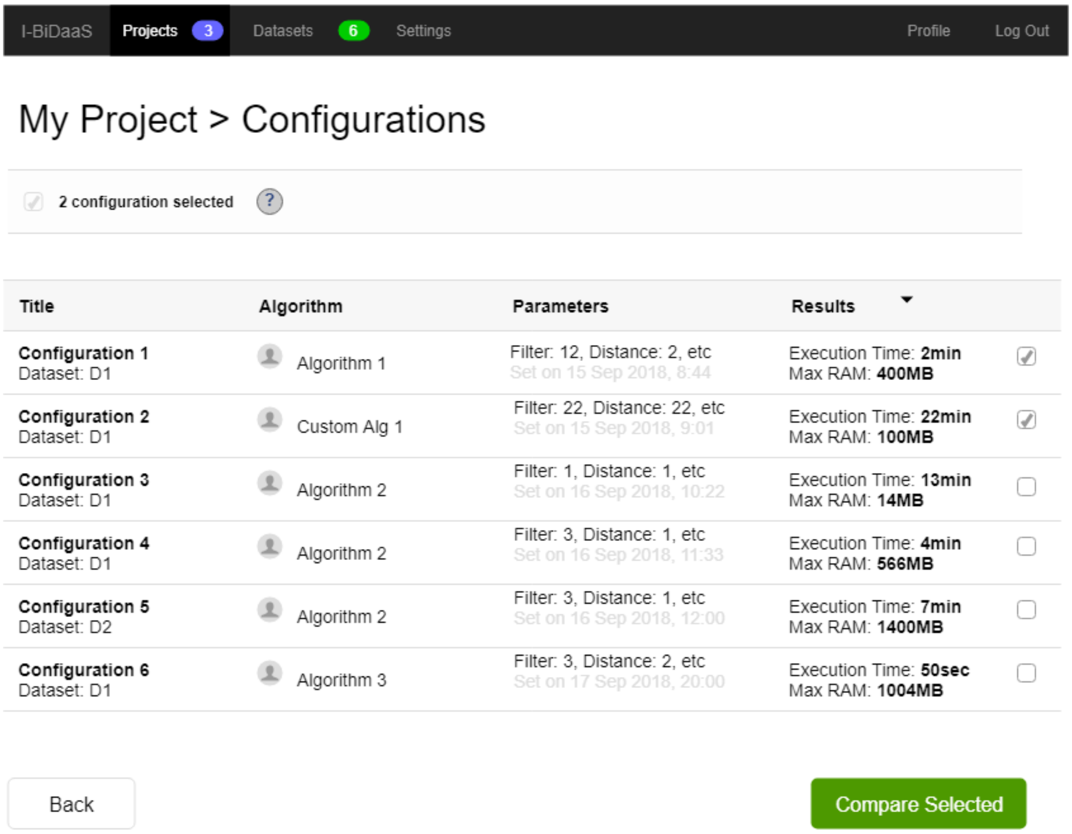
CPU Cores: 2
RAM: 2048 MB

Parameter 1: Option 5
Parameter 2: Min — Max

Figure 18. Experiment Results

Once the user is taken to this page, two download options are available. The first one is Project Configuration file which includes all the options of the current experiment, namely datasets, algorithms and defined parameter values. This functionality aims at increasing interoperability and reusability of the work performed using the I-BiDaaS interface. The configuration file can be potentially used to run the same experiment in another infrastructure and overcome difficulties such as privacy of the used data or resource limitations. The second option is to download the actual results in a format suitable for the user's needs. This option is also towards the same direction as the first one; users can feed the results into their own proprietary visualisation systems and explore them using familiar tools that are tailor-made for very specific datasets.

The last wireframe presents an overview screen where users can check configurations and compare results. Upon selecting 2 or more of them, they can see results side by side and extract useful conclusions that can foster further analysis or lead them to hidden insights.



My Project > Configurations

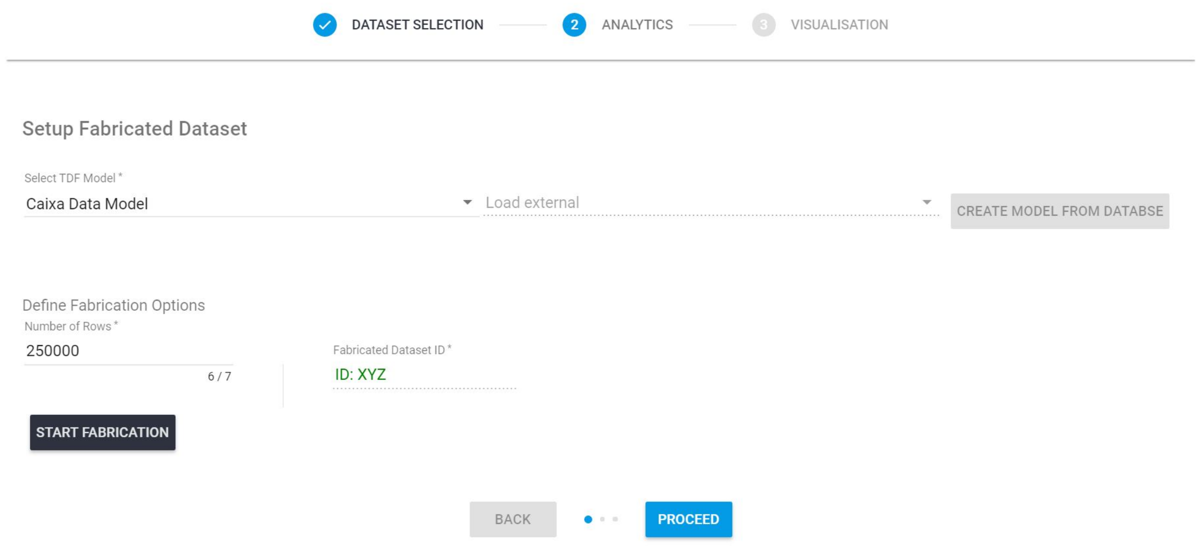
☑ 2 configuration selected ?

Title	Algorithm	Parameters	Results ▾	
Configuration 1 Dataset: D1	Algorithm 1	Filter: 12, Distance: 2, etc Set on 15 Sep 2018, 8:44	Execution Time: **2min** Max RAM: **400MB**	☑
Configuration 2 Dataset: D1	Custom Alg 1	Filter: 22, Distance: 22, etc Set on 15 Sep 2018, 9:01	Execution Time: **22min** Max RAM: **100MB**	☑
Configuration 3 Dataset: D1	Algorithm 2	Filter: 1, Distance: 1, etc Set on 16 Sep 2018, 10:22	Execution Time: **13min** Max RAM: **14MB**	☐
Configuration 4 Dataset: D1	Algorithm 2	Filter: 3, Distance: 1, etc Set on 16 Sep 2018, 11:33	Execution Time: **4min** Max RAM: **566MB**	☐
Configuration 5 Dataset: D2	Algorithm 2	Filter: 3, Distance: 1, etc Set on 16 Sep 2018, 12:00	Execution Time: **7min** Max RAM: **1400MB**	☐
Configuration 6 Dataset: D1	Algorithm 3	Filter: 3, Distance: 2, etc Set on 17 Sep 2018, 20:00	Execution Time: **50sec** Max RAM: **1004MB**	☐

Figure 19. Configuration Comparison

5.2 Screenshots of the prototype

The following screenshots depict the screens of the prototype that handle the functionality of the MVP. Figure 20 depicts the data fabrication options:



Setup Fabricated Dataset

Select TDF Model *
Caixa Data Model

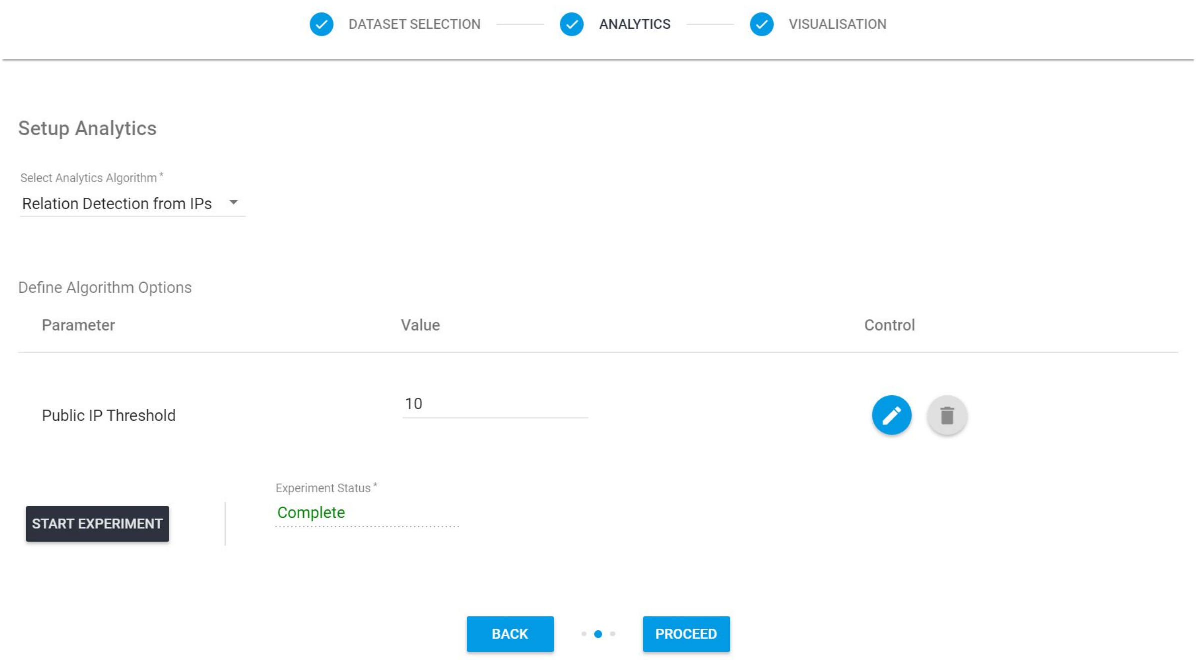
Define Fabrication Options

Number of Rows *
250000

Fabricated Dataset ID *
ID: XYZ

Figure 20. MVP Prototype: Data Fabrication

In the following figure, users select the analytics algorithm to run the experiment. Options of the selected algorithm are also set in the same page.



Setup Analytics

Select Analytics Algorithm *
Relation Detection from IPs

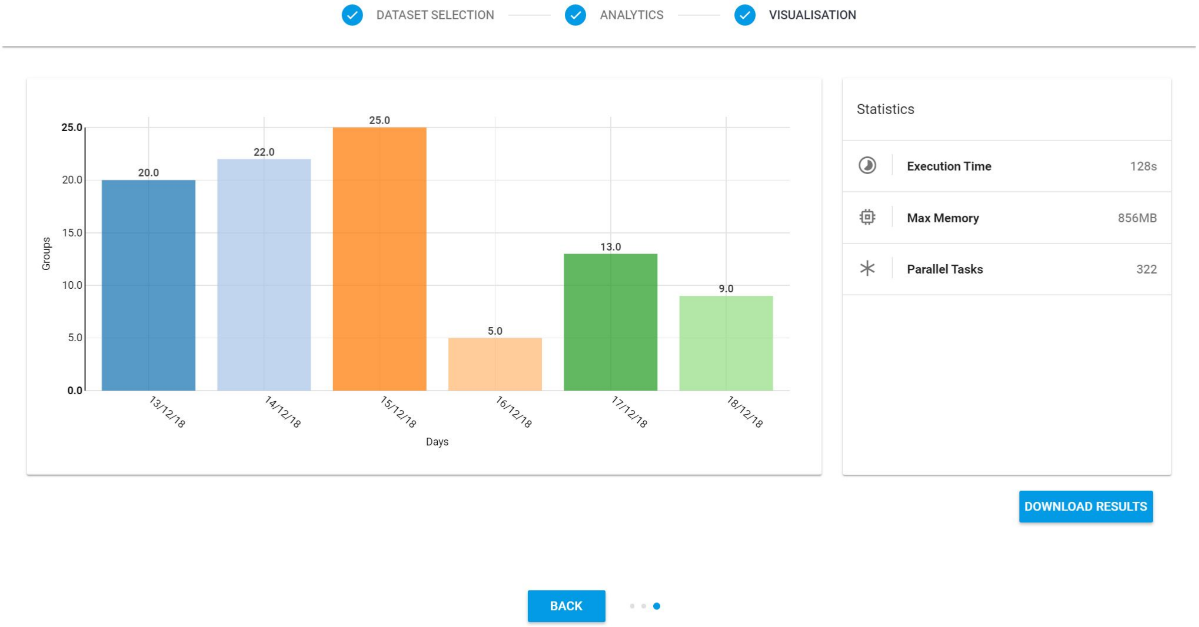
Define Algorithm Options

Parameter	Value	Control
Public IP Threshold	10	

Experiment Status *
Complete

Figure 21. MVP Prototype: Analytics Selection

The visualisation of the results is presented next. For the MVP purposes, the visualisation contains a bar chart showing the number of identified groups of users having a relationship according to the algorithm analysing the IPs of their access points. Various statistics are available on the right-hand side of the screen as well as the option to download the results of the experiment. Figure 22 displays the layout of the results visualisation page:



Statistics

Execution Time	128s
Max Memory	856MB
Parallel Tasks	322

Figure 22. MVP Prototype: Results Visualisation

6 Conclusion

This deliverable presents the methodology for the design of the user interface and the tools that consist its building elements. More specifically, the document presented the outcome from the involvement of end users in the early stages of the I-BiDaaS development activities. Having described the User Centric Design process, we proceeded with an overview of the platform functionalities and the definition of the main workflows that end users will follow when using the platform. The tools comprising the visualisation and monitoring framework have been presented in details and their use inside the platform has been set.

Based on a well-defined user centric design process that aims to achieve high degree of usability for the I-BiDaaS components, we described the initial wireframes of the multipurpose interface. These wireframes provide the initial conceptualisation of the functionalities expected from the I-BiDaaS software for IT and non-IT experts. Moreover, the first version of the interface has been released and includes the functionalities required to support the MVP and its use cases.

The next steps of this work include the evaluation of the interfaces and the validation of the prototype in real-world scenarios. The evaluation process will start right after the delivery of the MVP and the first report will be delivered in M14. This validation will allow us to better position the user expectations in the context of individual functionalities provided by the I-BiDaaS tools and components and adjust the future development activities. Furthermore, usability evaluation will reveal any possible early deviations from what the users expect from the system, thus allowing us to perform corrective actions and update the way that platform capabilities are presented to the end users in view of the first prototype delivery in M18.

7 References

[1] International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 12207:2017, "Systems and software engineering - Software life cycle processes", November 2017.

[2] International Organization for Standardization (ISO) 9241-210:2010, "Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems", March 2010.

[3] International Organization for Standardization (ISO) 9241-11:2018, "Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts", March 2018.

[4] Jeffrey Rubin, Dana Chisnell, Jared Spool, "Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, 2nd Edition", Wiley ed., ISBN: 978-0-470-18548-3, May 2008.

[5] Notes on User Centered Design Process (UCD), https://www.w3.org/WAI/redesign/ucd

[6] International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 25010:2011, "Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models", March 2011.

[7] International Organization for Standardization (ISO) 25062:2006: "Software engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Common Industry Format (CIF) for usability test reports", April 2006.

[8] National Institute of Standards and Technology (NIST), NISTIR 7432: "Common Industry Specification for Usability – Requirements (CIRU-R)", June 2007

[9] I-BiDaaS Consortium, Deliverable "D1.3: Positioning of I-BiDaas", September 2018.

[10] I-BiDaaS Consortium, Deliverable "D1.2: Architecture definition", August 2018.

[11] I-BiDaaS Consortium, Deliverable "D5.2: Big-Data-as-a-Self-Service Test and Integration Report", December 2018.