Horizon 2020 Program (2014-2020)

Big data PPP

Research addressing main technology challenges of the data economy

I-BiDaaS

Industrial-Driven Big Data as a Self-Service Solution

Industrial-Driven Big Data as a Self-Service Solution

Abstract: This report documents different modes and software modules included in the final version of the Industrial-Dirven Big-Data-as-a-Self-Service Solution.

Contributors	*Enric Pages (ATOS)* *Vassilis Chatzigiannakis (ITML)* *Giorgos Vassiliadis, Despina Kopanaki (FORTH)* *Dusan Jakovetic, Nemanja Milošević (UNSPMF)* *Leonidas Kallipolitis, Andreas Alexopoulos (AEGIS)* *Omer-Yehuda Boehm (IBM)* *Raül Sirvent, Cesare Cugnasco (BSC)* *Gerald Ristow (SAG)* *Ramon Martin de Pozuelo (CAIXA)* *Giuseppe Danilo Spennacchio (CRF)* *Ioannis Arapakis (TID)*

Quality Assurance	*Evangelia Kavakli (UNIMAN*

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

Revisions

Version	Date	By	Overview
1.0	20/12/2020	Enric Pages	Final Version
0.8	17/12/2020	Contributors	Reviewed version addressing comments
0.6	16/12/2020	Enric Pages	All contributions
0.3	28/11/2020	All Contributors	New Contributions
0.2	15/11/2020	Enric Pages	Merged Version
0.1	15/09/2020	All Contributors	Initial Inputs
0.0.1	10/03/2020	Enric Pages	TOC

Table of Contents

List of Abbreviations

AVT: Advanced Visualisation Toolkit

BDVA: Big Data Value Association

CI: Continuous Integration

CPU: Central Processing Unit

CVS: Concurrent Versions System

DFP: Data Fabrication Platform

E2E: End-to-End

GPU: Graphics Processing Unit

GUI: Graphical User Interface

ISTQB: International Software Testing Qualifications Board

IOPs: Input/output operations per second

IoT: Internet of Things

JSON: JavaScript Object Notation

MIT: Massachusetts Institute of Technology

MQTT: Message Queuing Telemetry Transport

MVP: Minimum Viable Product

PoCs: Proof-of-concepts

QA: Quality Assurance

RTC: Real-time Computing

SCM: Source Control Management

SQL: Structured Query Language

SVN: Subversion

SW: Software

TDF: Test Data Fabrication

UAT: User Acceptance Testing

UM: Universal Messaging

List of Figures

1 Introduction

1.1 Solution Overview

I-BiDaaS goal is to create new opportunities for self-service analytics aiming to achieve a unified Big Data as-a-service solution that will empower non-expert users to easily take advantages of the Big-Data technologies while at the same time increasing the speed of data analytics. For achieving this vision, one of the key aspects is the capability of the platform to operate handling large data volumes.

1.2 Target Audience

Amid the proliferation of Big Data, not only in terms of datasets but also technologies able to capture, store, manage and analyse large and variable collections of data, it is possible to identify in the data economy vertical submarkets, hereby referred to as target groups, for which the I-BiDaaS solution will carry beneficial value.

The main vertical submarkets that will constitute target groups for the I-BiDaaS solution are:

- Automotive, Aerospace & Transportation
- Banking & Securities
- Insurance
- Defence & Intelligence
- Public Safety & Homeland Security
- Education
- Healthcare & Pharmaceutical
- Smart Cities & Intelligent Buildings

1.2.1 I-BiDaaS End-Users

Within the above target group the I-BiDaaS end-users 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).

Furthermore, I-BiDaaS users include data integrators responsible for interfacing the existing internal enterprise systems with the I-BiDaaS system, administrators of such systems and application developers responsible for the provision or implementation of some analytic application. Such users can be employees of the company, but they could also work for a special service provider.

1.3 Structure of the Document

The outline of this document is as follows: The first chapter introduces the document and its objectives. The second chapter gives an overview of the I-BiDaaS solution and its challenges. A walk through the graphical user interface is described in chapter 3. Chapter 4 provides detailed information of the software modules and its documentation. Finally, the last section chapter provides summarizes lessons learnt from the application of the I-BiDaaS solution in the context of different industrial cases.

2 The I-BiDaaS Solution

2.1 Motivation

2016 was a landmark year for big data with more organizations storing, processing, and extracting value, from data of all forms and sizes. Since then, systems that support large volumes of both structured and unstructured data continued to rise. The market demand platforms that help data custodians govern and secure big data while empowering end users to analyse it. Organizations leverage diverse data pools to drive value, so variety seems to increase significantly in comparison to volume and velocity that drive big-data investments. At the same time, the convergence of Internet of Things (IoT), cloud, and big data create new opportunities for self-service analytics[1] towards a completely new paradigm towards big data analytics. Human and machine created data are being aggregated, transforming our economy and society.

To face these challenges, companies call upon expert analysts and consultants to assist them. The trends above lead us to one of the main challenges of the data economy[2], Big-Data-as-a-Self-Service. A self-service solution will be transformative for organizations; it will empower their employees with the right knowledge and give the true decision-makers the insights they need to make the right decisions. It will shift the power balance within an organisation, increase efficiency, reduce costs, improve employee empowerment, and increase profitability. The domains that can exploit such self-service solutions are numerous; I-BiDaaS explores three critical ones with significant challenges and requirements: banking, manufacturing, and telecommunications.

I-BiDaaS Project statement: *'I-BiDaaS aims to empower users to easily utilize and interact with big data technologies, by designing, building, and demonstrating, a unified framework that: significantly increases the speed of data analysis while coping with the rate of data asset growth, and facilitates cross-domain data-flow towards a thriving data-driven EU economy. I-BiDaaS will be tangibly validated by three real-world, industry-lead experiments.'*

The motivation behind I-BiDaaS led to the definition of five (5) specific objectives that guided the research & innovation efforts of the Consortium:

1. Develop, validate, demonstrate, and support, a complete and solid big data solution that can be easily configured and adopted by practitioners.
2. Break inter- and intra-sectorial data-silos, create a data market and offer new business opportunities, and support data sharing, exchange, and interoperability
3. Construct a safe environment for methodological big data experimentation, for the development of new products, services, and tools
4. Develop data processing tools and techniques applicable in real-world settings and demonstrate significant increase of speed of data throughput and access.
5. Develop technologies that will increase the efficiency and competitiveness of all EU companies and organisations that need to manage vast and complex amounts of data.

[1] Self-Service Analytics. (2017, January 03) http://www.gartner.com/it-glossary/self-service-analytics, (accessed March 30, 2018).

[2] Passlick, J., Lebek, B., & Breitner, M. H. (2017). A self-service supporting business intelligence and big data analytics architecture, 13th International Conference on Wirtschaftsinformatik, St. Gallen, Switzerland.

2.2 I-BiDaaS Platform Modes

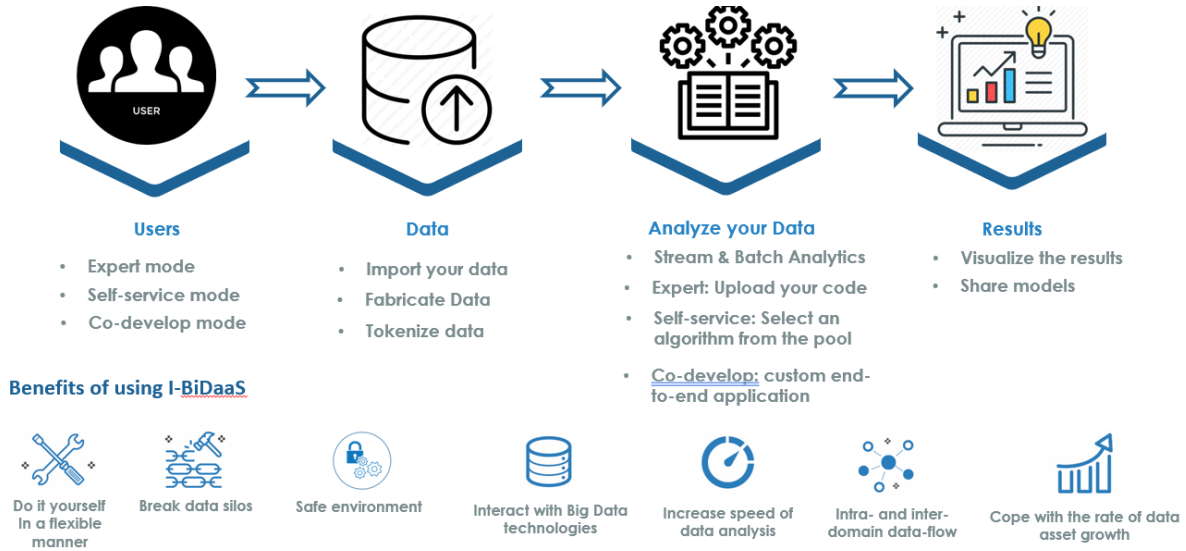

Figure 1. I-BiDaaS Big Data Pipeline

The figure above depicts the Big Data pipeline followed by I-BiDaaS, the platform aims to accommodate the requirements of different end-user according to their data analytics know-how and technical expertise. To this end, it offers three different modes, tailored to different categories of users summarised in the following table.

Expert Mode	Self-Service Mode	Co-develop Mode
Allows user to upload their own code based on pre-defined templates available in I-BiDaaS.	Helps user to construct in an intuitive way a Big Data pipeline & select an algorithm from a pool of available algorithmic implementations	Customized end to end project for specific industrial use case.

In more detail, (i) the Self-service mode, which allows industry in-house personnel that has domain knowledge and some insights about data analysis (non-experts) to easily construct Big Data pipelines in a user- friendly way, selecting a pre-defined data analytics algorithm from an available list; (ii) the Expert mode, allows experts (developers) to upload their own data analytics code based on the available I-BiDaaS highly reusable templates; and (iii) the Co-Develop mode, corresponds to an end-to-end solution for a given industry project developed by the I-BiDaaS team (the I-BiDaaS use cases).

For further information, see section 3 I-BiDaaS Dashboard.

2.3 Overall Architecture

The I-BiDaaS architecture is divided into three principal conceptual layers: the infrastructure layer, the distributed large-scale layer, and the application layer. The infrastructure layer forms the "lower-most vertical" layer which includes the actual underlying storage and processing infrastructure of the I-BiDaaS solution. The distributed large-scale layer is responsible for the orchestration and management of the underlying physical computational and storage infrastructure. It allows the effective and efficient use of the cloud infrastructure and enables the application layer to provide effective big data analytics. The application layer sits on top of the distributed large-scale layer and refers to the architecture aspects and modules that are involved in the actual workflow of extracting actionable knowledge from the big data, starting

from data ingestion, preparation, and fabrication, to batch and streaming analytics, to visualization and delivering analytics results for supporting decision making. Figure 1 depicts the I-BiDaaS platform and its constituent modules, indicating also the three layers.

The I-BiDaaS platform is designed to be able to offer the following functionalities: data ingestion and integration, fabrication of realistic synthetic data for testing, advanced IT services for big data processing tasks, sequential programming (where at the run-time, the platform automatically "parallelizes" the task over the actual distributed infrastructure), open source repository for Big Data processing tasks, data sampling and interactive querying, advanced visualizations and monitoring.

A number of innovative features are implemented. First, the results (e.g., correlations) produced by the batch processing (analytics) module are fed back to Data Fabrication Platform (DFP) via Universal Messaging (UM); these results are then used for training and to help building rules that will be used for future data generation purposes, hence improving the fabrication quality. In addition, parts of the streaming analytics that can be parallelized can be offloaded to the GPU-accelerated streaming analytics and pattern matching sub-module, giving the opportunity to partition the analytics queries, between the high-level stream processing engine (Apama) and the low-level, hardware optimized implementation (GPU-accelerated sub-module). Finally, the system is envisioned to introduce feedback from the batch analytics results to ML problem modelling; in other words, models (e.g., for structured (non)convex optimization) can be upgraded for each application class (e.g., through introducing non-zero weights to additional regularization functions) based on historical analytics performance scores.

The architecture of the 1st integrated version (also called Minimum Viable Product) helps us to illustrate how works the backend of the platform as it includes most of the modules identified in the complete I-BiDaaS architecture (Figure 2); a simplified version of the 1st version architecture is presented in Figure 3. Afterwards, the integration of all modules of I-BiDaaS architecture have been demonstrated in the final integrated version of the platform across several use cases. In addition, in the final solution of the platform the generic I-BiDaaS use case demonstrates the functionalities of the different envisioned modes of the I-BiDaaS platform. In the generic use case, users can either initiate existing projects (with already implemented algorithms, e.g., K-means) or create custom ones (by uploading and running their own code - expert mode). Then they create and execute their experiment(s) based on their data and preferences.

Regarding the 1st integrated version architecture and focusing explicitly on batch processing, the system architecture workflow is shown in Figure 4. The cluster that runs the batch processing jobs is based on docker swarm 1. The swarm consists of a manager node and a set of worker nodes. The orchestrator assigns dynamically a set of workers from the set of available nodes in docker swarm, based on the user's preferences and set-up inserted through the UI. These workers exploit the Cassandra DB and the shared FS in order to complete the requested job. On the other hand, Figure 5 demonstrates the case of stream processing; here, the analysis is carried out through the APAMA analytics module in collaboration with the Universal Messaging (UM) tool, rather than via docker swarm.

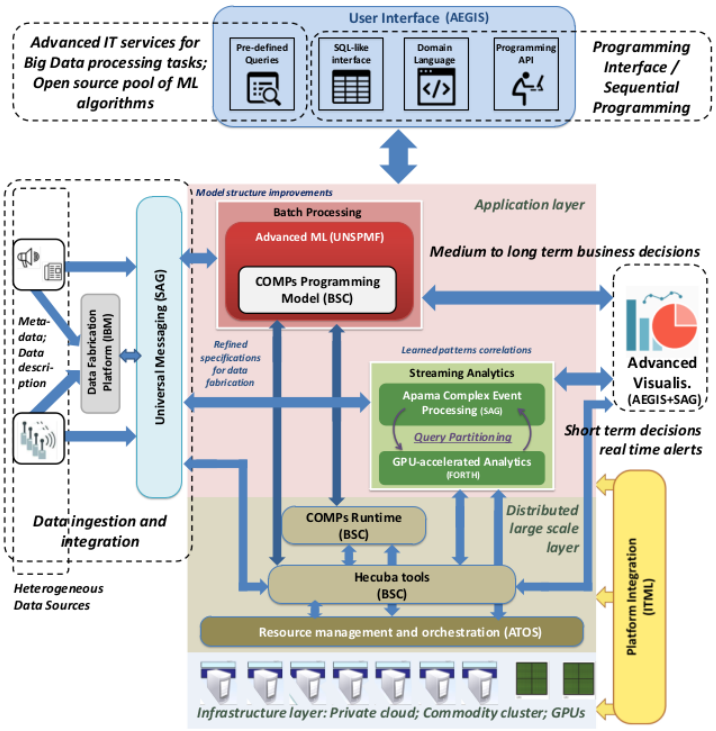

Figure 2. I-BiDaaS architecture

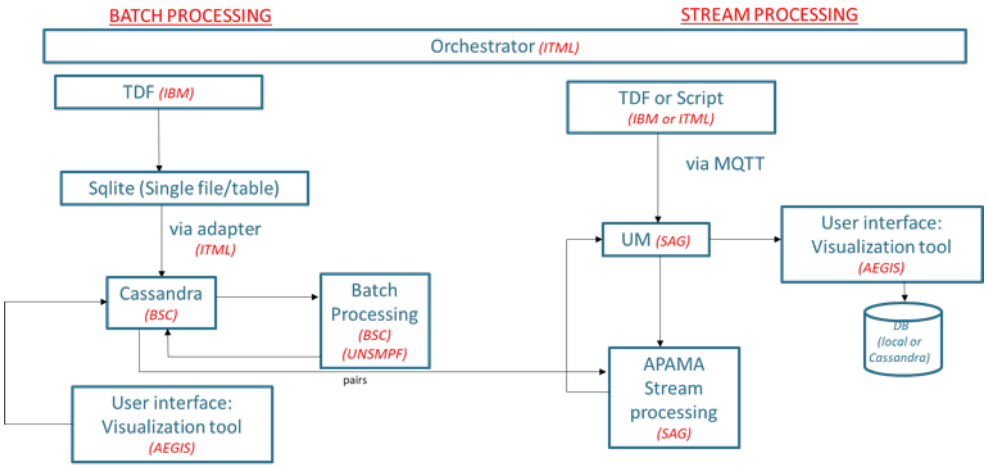

Figure 3. The 1st integrated version architecture (MVP)

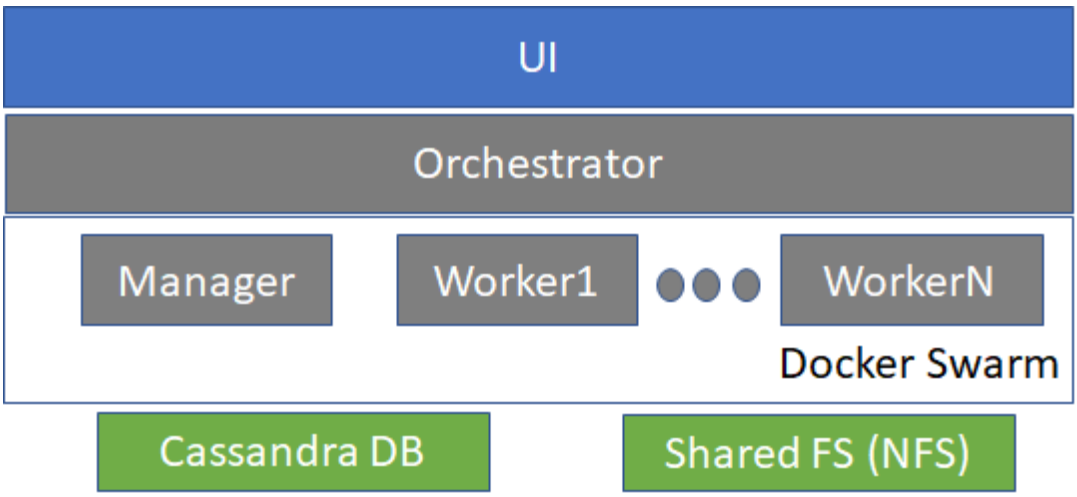

Figure 4. Batch processing architecture: The orchestrator and the docker swarm

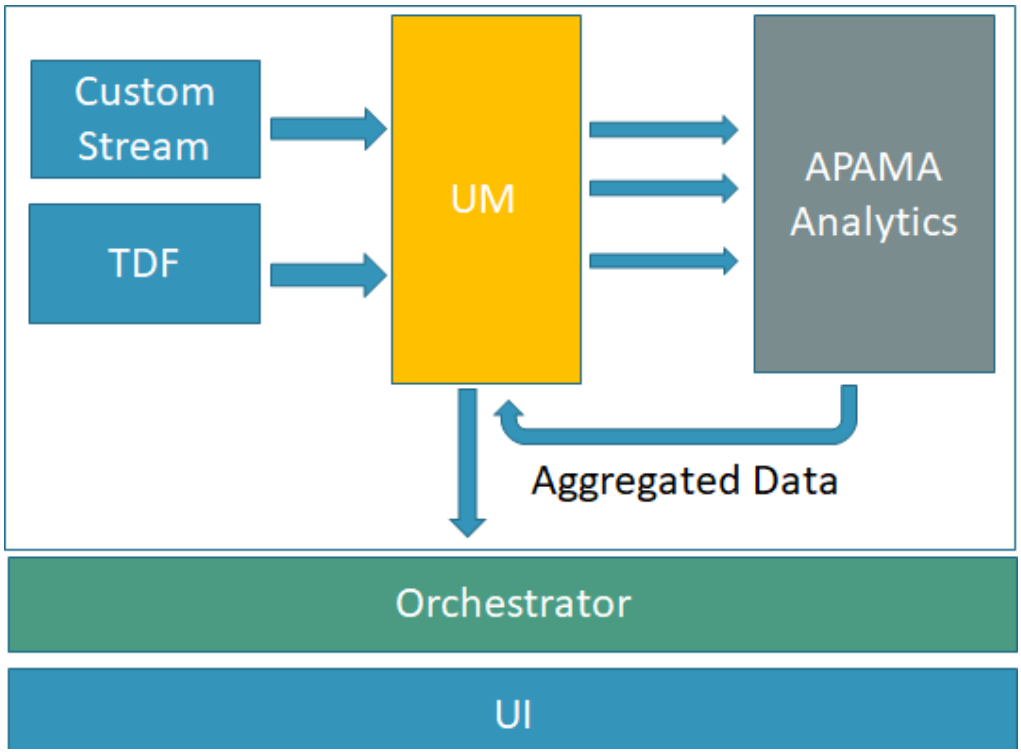

Figure 5. Streaming processing architecture: The orchestrator and APAMA analytics

2.4 Challenges to Overcome

Some of the main challenges that the I-BiDaaS Consortium faced from the beginning of the project are related to breaking data silos and also to data sharing and data availability. I-BiDaaS has carried out several actions towards breaking inter and intra sectoral silos, creating data market, and supporting data sharing, exchange and interoperability.

The I-BiDaaS Data Management Plan (I-BiDaaS DMP) was delivered in M6 (June 2020) having as a main focus the description on how research data will be handled during and after the end of the project, what data will be collected, processed and/or generated, which methodology and standards will be applied, whether data will be shared/made open and how

data will be curated and preserved. I-BiDaaS project participates in the H2020 Open Research Data Pilot (ORDP), a flexible pilot action that aims to improve and maximize open access to, and re-use of, research data generated by projects. The DMP detailed how and which datasets will meet the FAIR Data management principles[3], (i.e., *Findable, Accessible, Interoperable and Re-usable*), and how they will be curated and preserved during and after the lifetime of the project.

It is worth mentioning how I-BiDaaS successfully managed to meet the FAIR Data management principles for all the Open datasets. The actions towards each principle are summarised below.

Making data findable, including provisions for metadata

- Zenodo[4], a well-known data repository, was used to provide open access to the I-BiDaaS datasets.
- A standardized naming convention for all the project datasets was used.
- A DOI was assigned to achieve effective and persistent citation.
- Search keywords describing the dataset or content of the data was provided.
- All the project datasets were documented and uploaded with their related metadata.
- The *minimum DataCite metadata standards* were followed.

Making data openly accessible

- Open access to all peer-reviewed scientific publications relating to its results was ensured.
- The consortium followed 'gold' or 'green' open access in all kinds of publications.
- To raise awareness of the project, increase its impact and ensure its long-term sustainability, the repositories that were used for the project open datasets are Zenodo and the Github platform that the I-BiDaaS knowledge repository[5] is built on.

Making data interoperable

- To increase the interoperability of the provided data, commonly used vocabularies for the metadata within the datasets were used. Interoperability was ensured by using the same standards for data and metadata capture/creation to all datasets.
- I-BiDaaS followed the *minimum DataCite metadata standards* for datasets' description.

Increase data re-use (through clarifying licences)

- Data reusability was achieved by uploading the datasets to the Zenodo repository.
- The datasets are and will be made available to third parties for at least 1 year after the completion of the project.
- The owner is responsible for maintaining the data after the completion of the project. The owner of the data is the beneficiary that generates it.
- To protect the ownership of the publicly provided datasets, *Creative-Commons-Licences (CC)* were used. CC are machine-readable licenses.

Due to the sensitive nature of real-life industrial datasets, as well as the substantial privacy and security risks involved in granting access to third parties, early access was anticipated to be highly challenging. Therefore, to facilitate the early exploration and development phases of the applications in cases where real data is not readily available, data providers shared the data characteristics and format of the respective datasets in order to fabricate data using IBM's Data

[3] FAIR data principles (FORCE11): https://www.force11.org/group/fairgroup/fairprinciples

[4] https://zenodo.org

[5] https://github.com/ibidaas/knowledge_repository

Fabrication Tool to facilitate the development of the I-BiDaaS technologies and the validation of the I-BiDaaS solution against industrial experiments. By realistic synthetic data, we mean the fabricated data that "mimics" a real dataset of a data provider generated by a data fabrication tool (here IBM's data fabrication platform) based on metadata descriptions and other data specification rules provided by the data provider.

The use of synthetic data was a way to by-pass the restrictions without compromising security or privacy. However, they realised that the analysis of rule-based fabricated data did not enable the extraction of new insights from the generated dataset. Thus, they continued the discussions with their legal departments and tried to find other ways to share real data. The result of this process was successful since for all the defined use cases, data providers managed to share real anonymised/encrypted/masked data.

At M6, in the I-BiDaaS DMP, 10 datasets were defined, 9 of which were synthetic and 4 open access. Before the end of the project, the I-BiDaaS data providers successfully shared 12 datasets, 8 of which are real anonymised/encrypted and 9 open access. This change is considered important and drastically affected the overall impact of the project. Table X summarises the final list with the I-BiDaaS datasets along with the owner and the accessibility level.

No	Name	Owner	Data	Accessibility	Link
1	Synthetic Mobility Data	TID	Synthetic	Open Access	https://doi.org/10.5281/zenodo.4274458
2	Real Mobility Data	TID	Real	Confidential (in-house)	-
3	Synthetic Call Center Data	TID	Synthetic	Open Access	https://doi.org/10.5281/zenodo.4274454
4	Real Call Center Data	TID	Real	Confidential (in-house)	-
5	Bank Transfer – Tokenised Dataset	CAIXA	Real Tokenised	Open Access	https://doi.org/10.5281/zenodo.4264086
6	IP Addresses - Synthetic Dataset	CAIXA	Synthetic	Open Access	https://doi.org/10.5281/zenodo.4091026
7	IP Addresses – Tokenised Dataset	CAIXA	Real Tokenised	Open Access	https://doi.org/10.5281/zenodo.4091057
8	Online Banking – Tokenised Dataset	CAIXA	Real Tokenised	Open Access	https://doi.org/10.5281/zenodo.4264239
9	SCADA - Real Dataset	CRF	Real Anonymized	Open Access	https://doi.org/10.5281/zenodo.4265324
10	MES - Real Dataset	CRF	Real Anonymized	Open Access	https://doi.org/10.5281/zenodo.4264473
11	Aluminium die-casting - Synthetic Dataset	CRF	Synthetic	Open Access	https://doi.org/10.5281/zenodo.4274452
12	Aluminum Die-casting - Real Dataset	CRF	Real Anonymized	Confidential	https://doi.org/10.5281/zenodo.4321371

Overall, I-BiDaaS helped data providers to push for internal changes in policies and processes to extract data outside their premises, breaking not only internal but external data silos. 9 datasets are completely open to the research community.

Regarding internal data silos, the I-BiDaaS data providers managed to collect and aggregate data from different sources. For example, CAIXA datapool, the big data infrastructure where most of the collected data is stored, allows all the employees of the company (with all their branches and brands) to access the same data models. The access control management is done in a centralized way from the digital security department. That allows the enterprise to avoid most of the internal silos. However, there are some logs and data that still is collected internally in other systems such as Security Information Event Management (SIEM) tools and not completely integrated in CAIXA datapool. I-BiDaaS allowed to integrate datapool and SIEM. However, the strongest impact to CAIXA comes on breaking external silos and providing tools to export the data of CAIXA with external stakeholders.

TID provided to the consortium in-house access to two different datasets addressing three orthogonal use cases and consist of very diverse signals that span from call logs to streams of mobile phone transactions for 10s of millions of customers. Moreover, the datasets are the product of joins of smaller collections that are curated and maintained by various product teams within TID. To this end, TID met successfully both criteria of data diversity, data complexity and cross-domain interoperability

The Data in the use cases developed in the CRF industrial environment already come from several sources: sensing systems at shopfloor level in the plant, quality control and quality analysis. The sensing data, anonymised by CRF, are provided by the company operating the industrial process, quality control and quality analysis are provided by CRF, demonstrating the integration of data from different sources and layer of the company and breaking of internal and external silos between different companies

Moreover, in the spirit of ensuring that the project's results are shared effectively with the relevant communities and providing maximum visibility and public awareness to companies outside the consortium, CRF and TID successfully organised two Hackathon Days at Campus Melfi in Italy (CRF) and at the premises of Telefonica I+D in Barcelona, respectively. In both cases in-house access of proprietary data to the attendees was given after proper data anonymisation and use of NDAs.

Each data provider has a different story to share related to the experience gained regarding the challenges, difficulties and obstacles they faced on data availability and data sharing. The lessons learned of the three I-BiDaaS data providers are described in detail in Section 5 of this user guide.

3 I-BiDaaS Dashboard

3.1 Introduction

The I-BiDaaS dashboard is the main entrance point to the platform facilities. It allows the users to experiment with the provided resources and functionalities and to also visit existing applications that were created in close collaboration with the project pilot partners.

The dashboard is mainly split in three main operational modes, introduced in section 2.2. The following sections further describe these modes and provide examples of step by step operation for each one of them.

3.1.1 Home Page

The Homepage offers the links to the 3 available modes:

- **Expert Mode** (Figure 6): In this mode, users are able to run their own code by using platform's resources. Every user-created project is stored under the user's profile and experiments within the project can be executed multiple times, giving users the freedom to experiment on their data with different algorithmic setups and various resource allocations.

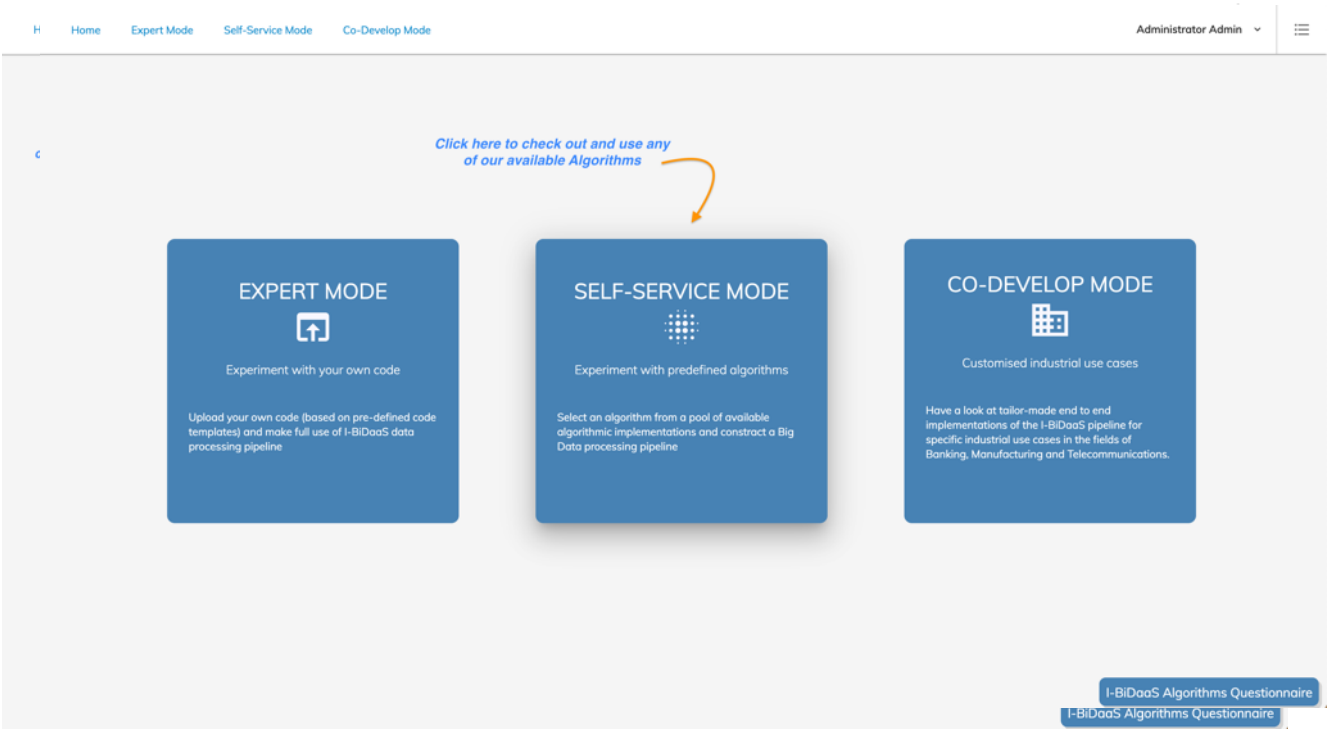

Figure 6. Expert Mode

- **Self-Service Mode** (Figure 7): The self-service mode provides a set of preconfigured algorithms to run on users' data. This mode allows users to perform standard data analytics tasks on their datasets without the need of any coding knowledge.

Figure 7 Self-Service Mode

- **Co-Develop Mode** (Figure 8): This mode presents projects that have been tailored-made to match specific needs in the domains of the project's pilot partners. The I-BiDaaS infrastructure has been configured to address the challenges of these specific use cases and the results showcase the potential of the developed platform in real-world business scenarios.

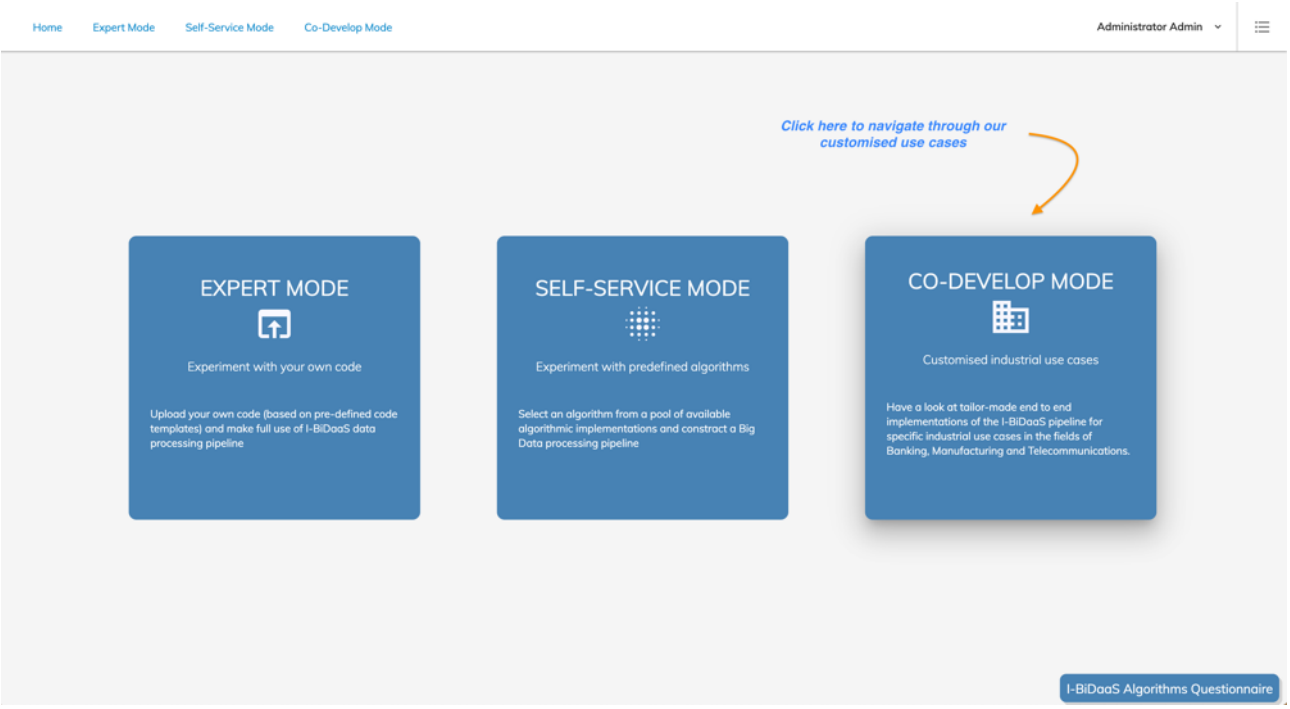

Figure 8. Co-Develop Mode

3.1.2 Algorithms Questionnaire

In order to assist users having no expertise on data analysis algorithms, a self-service questionnaire has been developed exists It can be found at the right bottom corner of the dashboard as seen in Figure 9.

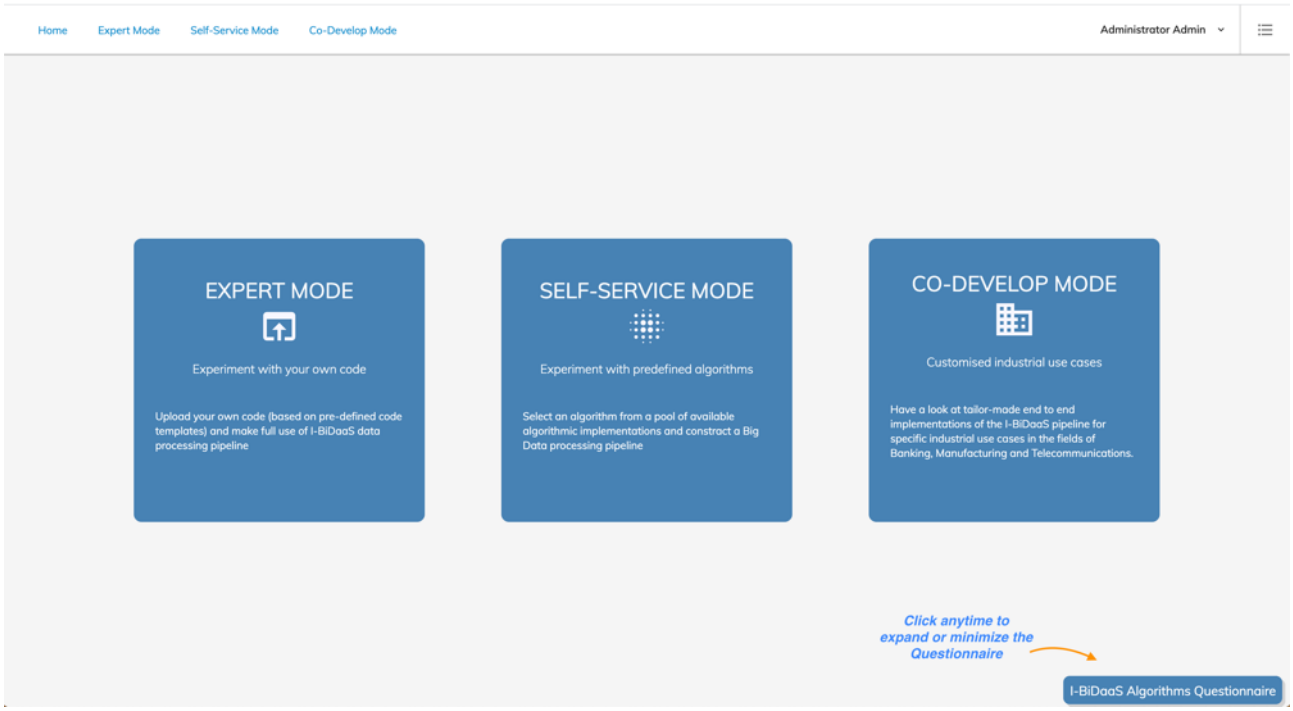

Figure 9. Questionnaire Link

The questionnaire can be answered at any time by users and as many times as they want (Figure 10). Through this process, users can get a recommendation about the more suitable algorithm they may use, depending on the kind of data they wish to analyse.

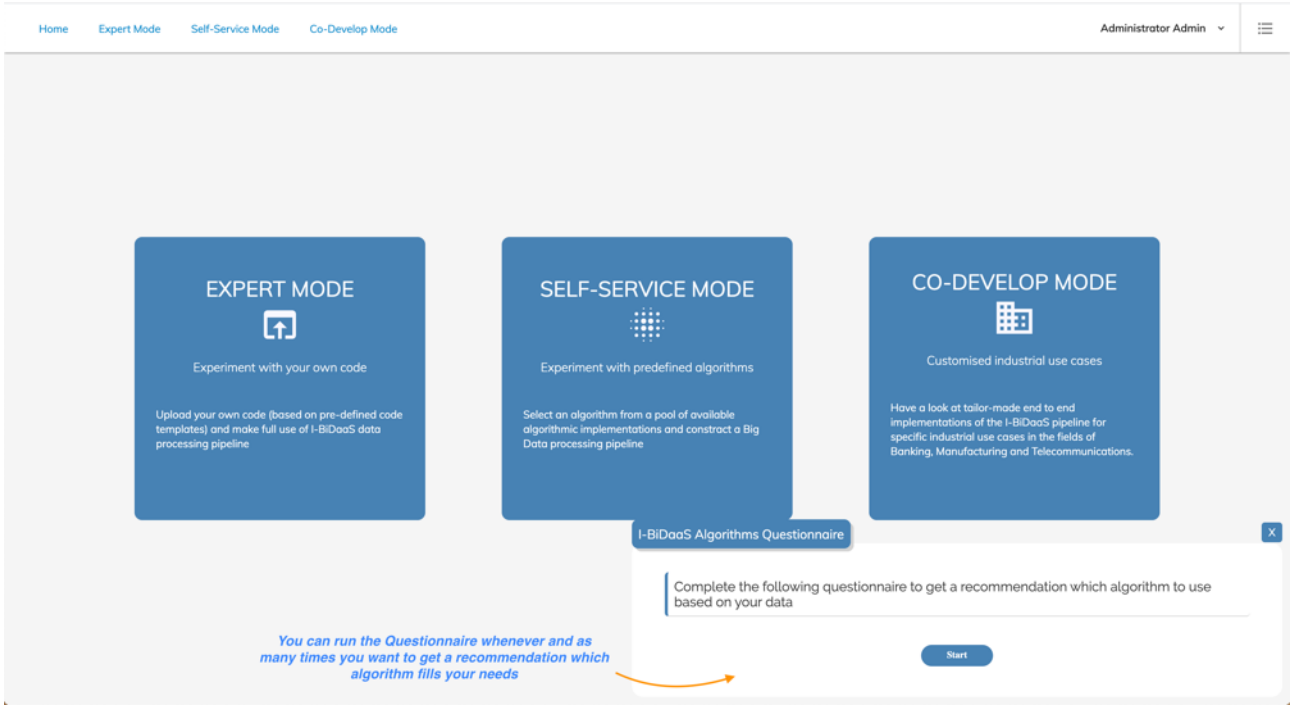

Figure 10. Questionnaire starting screen

When a specific algorithm is suggested, a small description about the algorithm is presented together with a direct link to the algorithm's page (Figure 11).

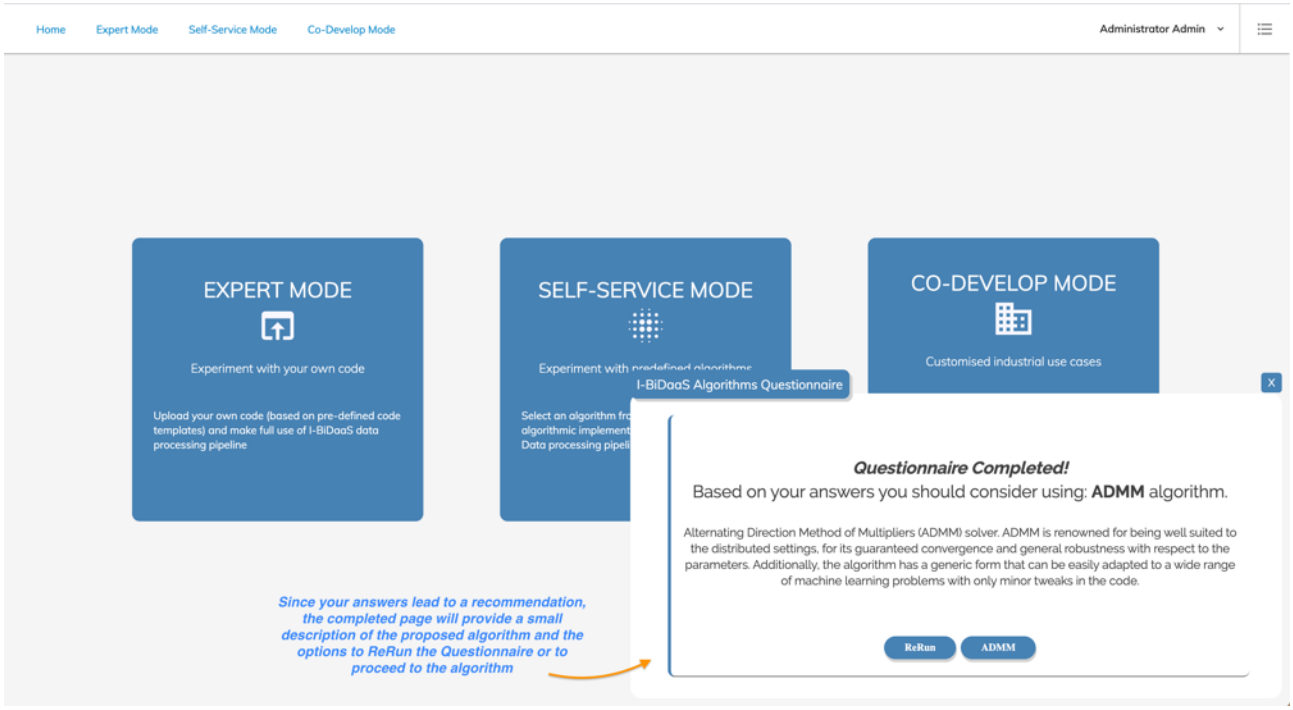

Figure 11. Algorithm Suggestion

In some cases, the questionnaire might not be able to determine which algorithm is the best fit for the user's data, so the user can either re-run the questionnaire (and change the answers) or navigate to the Self-Service Mode page and browse all the available algorithms (Figure 12).

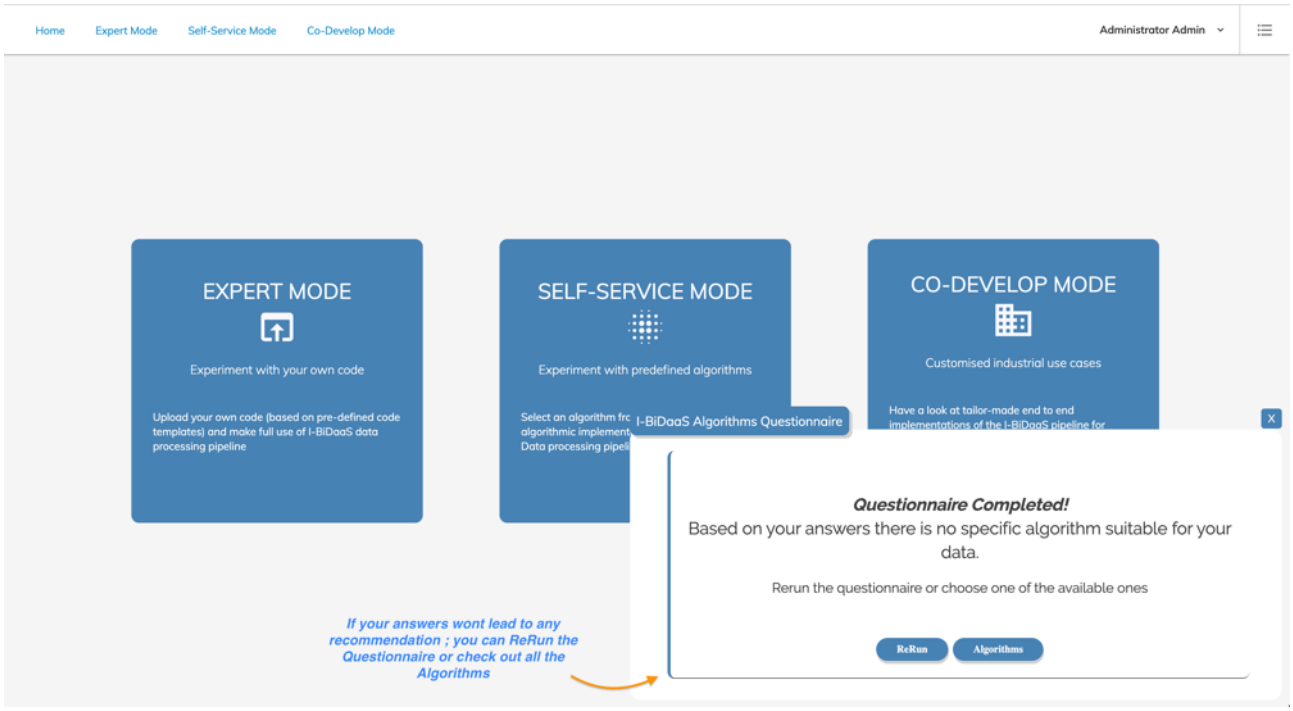

Figure 12. No algorithm proposed

3.2 Walk Through Generic Use Case

This section describes the steps required for setting up a new project and running a new experiment in the Expert Mode of the I-BiDaaS toolbox.

First, navigate to https://toolbox.ibidaas.eu and login with your credentials. After logging in, select EXPERT MODE (Figure 13).

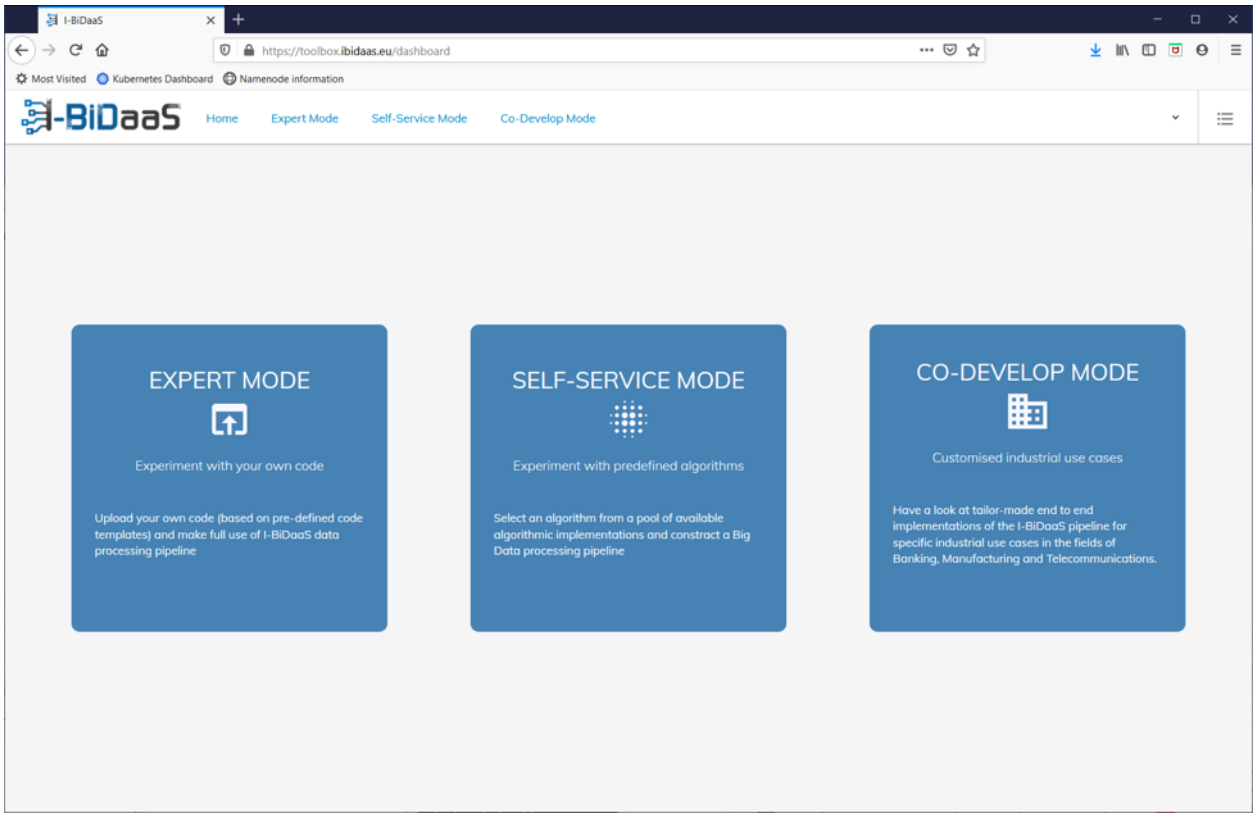

Figure 13. Main page

In the Expert mode screen, select to create a new project (Figure 14)

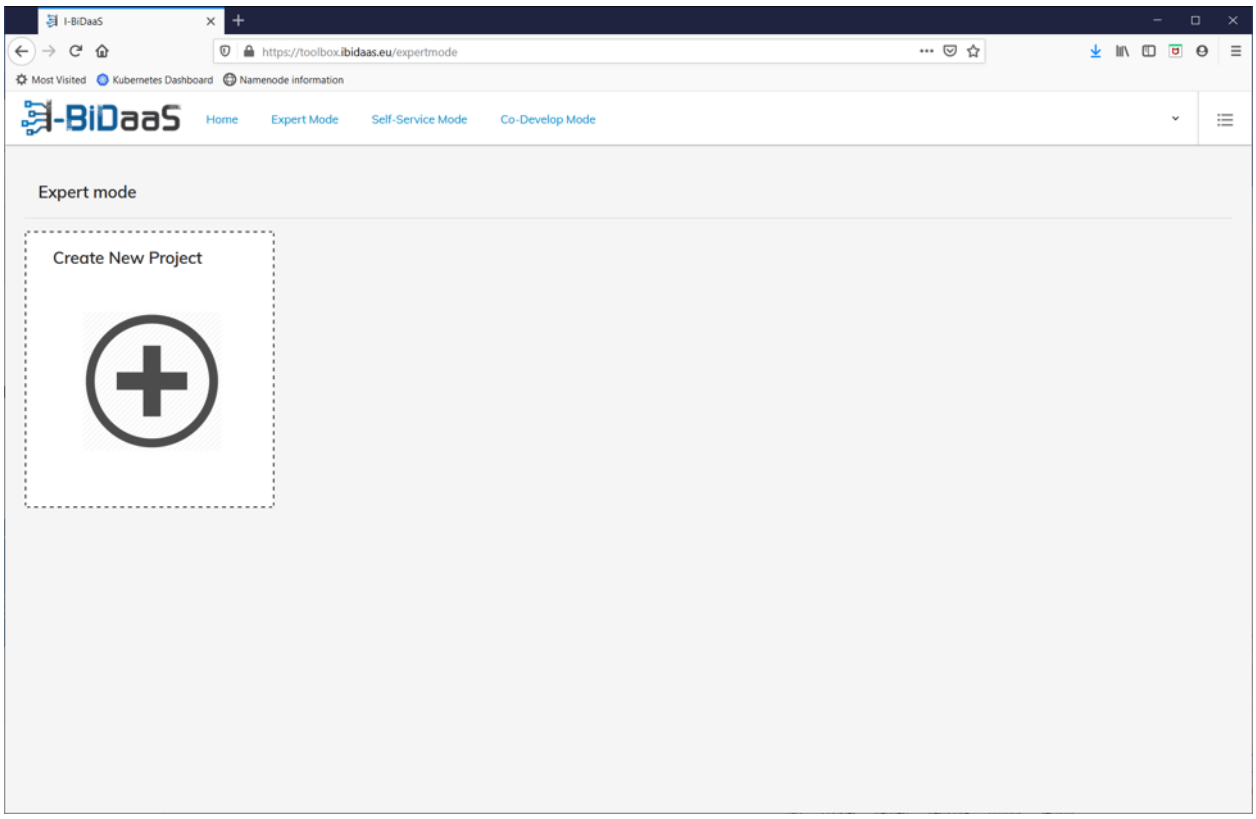

Figure 14. Expert mode

Start adding the new project details. For the new project, enter a name and description, select "None" as input selection and use the default docker image. You can download the random_forest.zip code template (Figure 15) and use it exactly as is, or you can choose to modify its contents (the main.py) file. In any case, the zip must contain the main python script and optionally any dependencies or even a sample input file.

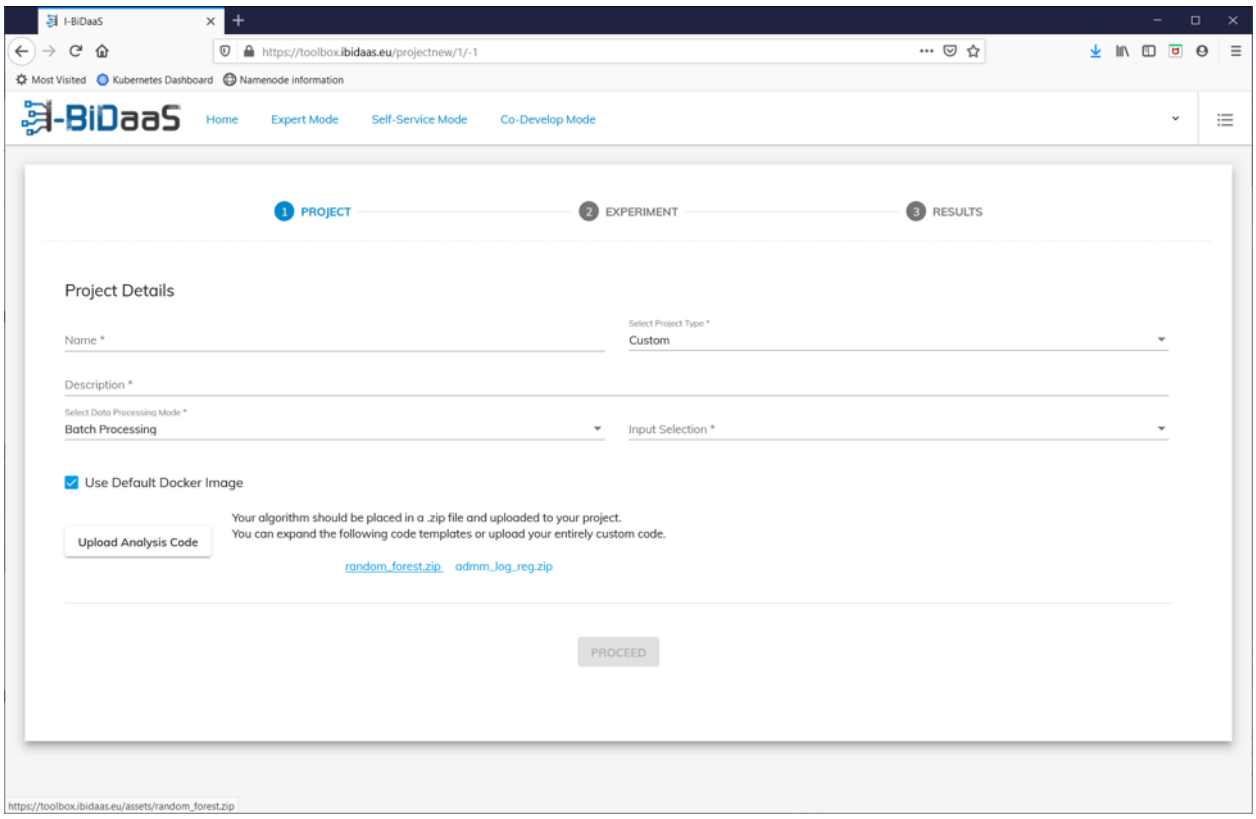

Figure 15. Preparing the project (I)

Upload the code as shown in Figure 16. The zip file must not contain any folders.

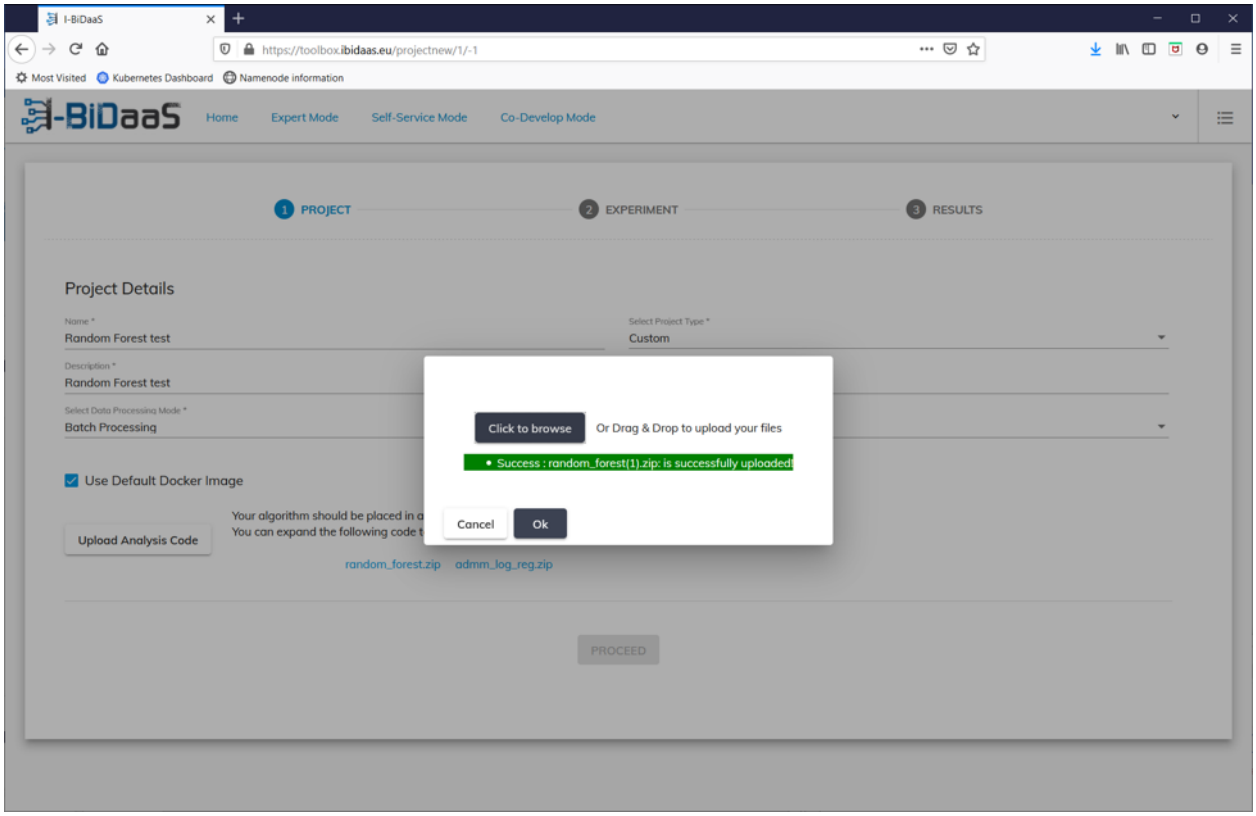

Figure 16. Code uploading

Click “Proceed” and in the next page (Figure 17) set up a new experiment. A project may contain one or more experiments. In each experiment you can modify the command (e.g. provide different parameters) for the same code that was provided during project initialization.

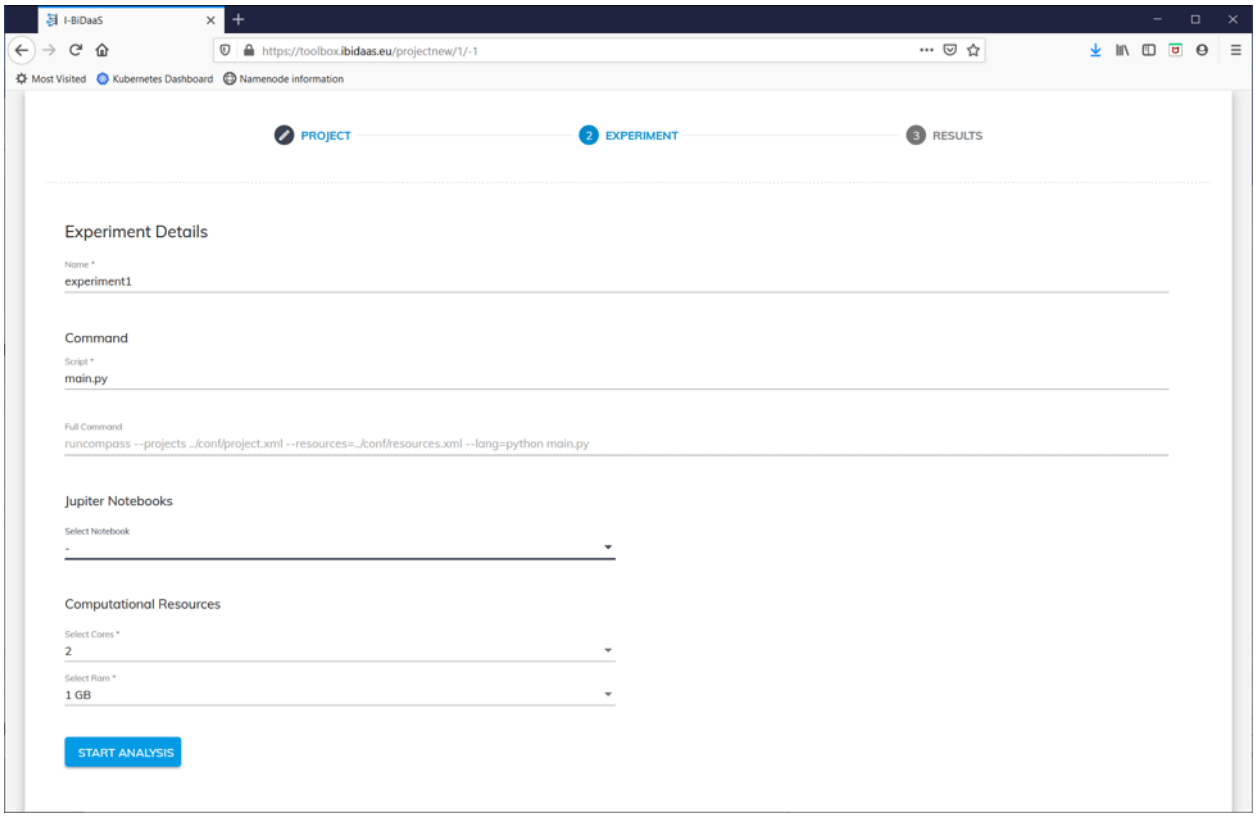

Figure 17. Experiment details

After setting up the required parameters of the new experiment you can click “Start Analysis”. The experiment will run for some time (Figure 18) in case of the random forest code, it should be a couple of minutes , depending on the server load).

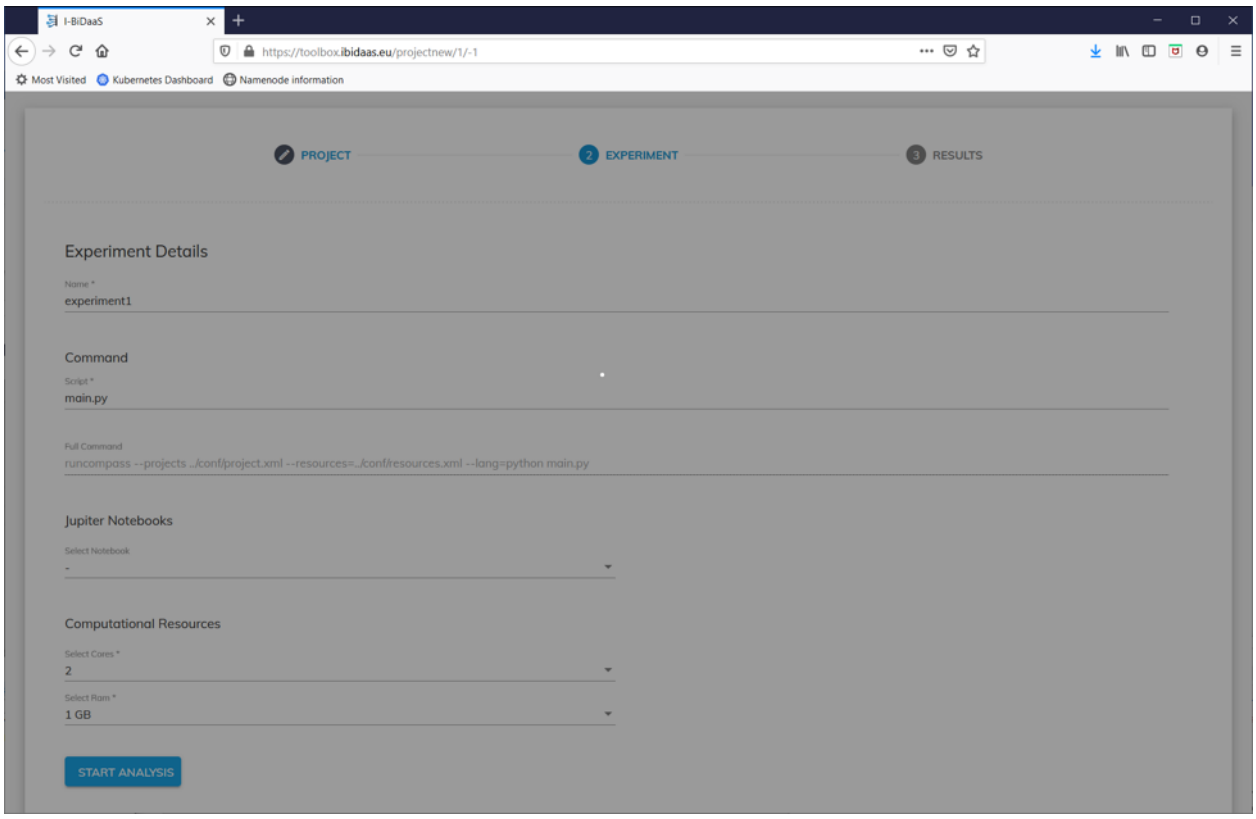

Figure 18. Running experiment

You can leave this page at any time and navigate back the expert mode page and then to the Project details page (Figure 19). There you can see the list of experiments. The new experiment should be in running state for a while.

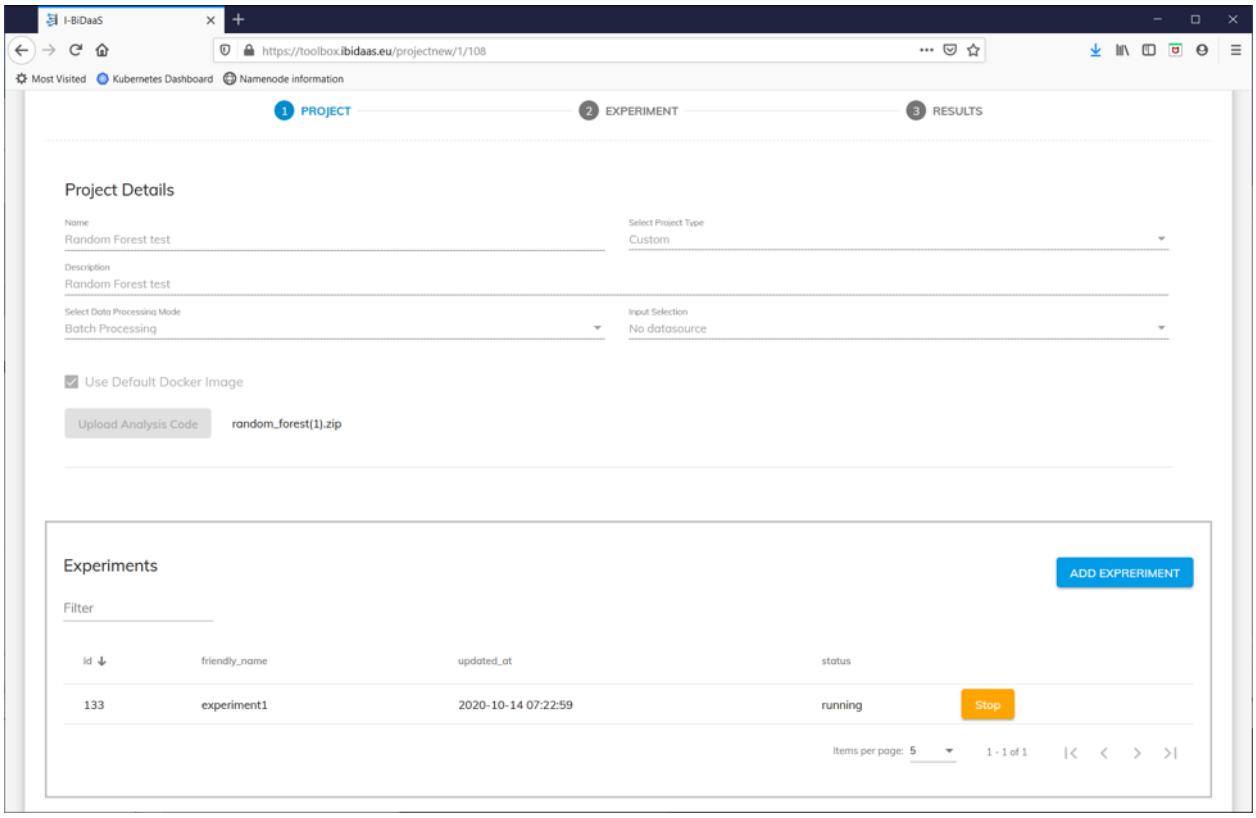

Figure 19. Project details page

When the experiment terminates, the state will be updated to “stopped” and then you can choose to view the results or delete it (Figure 20).

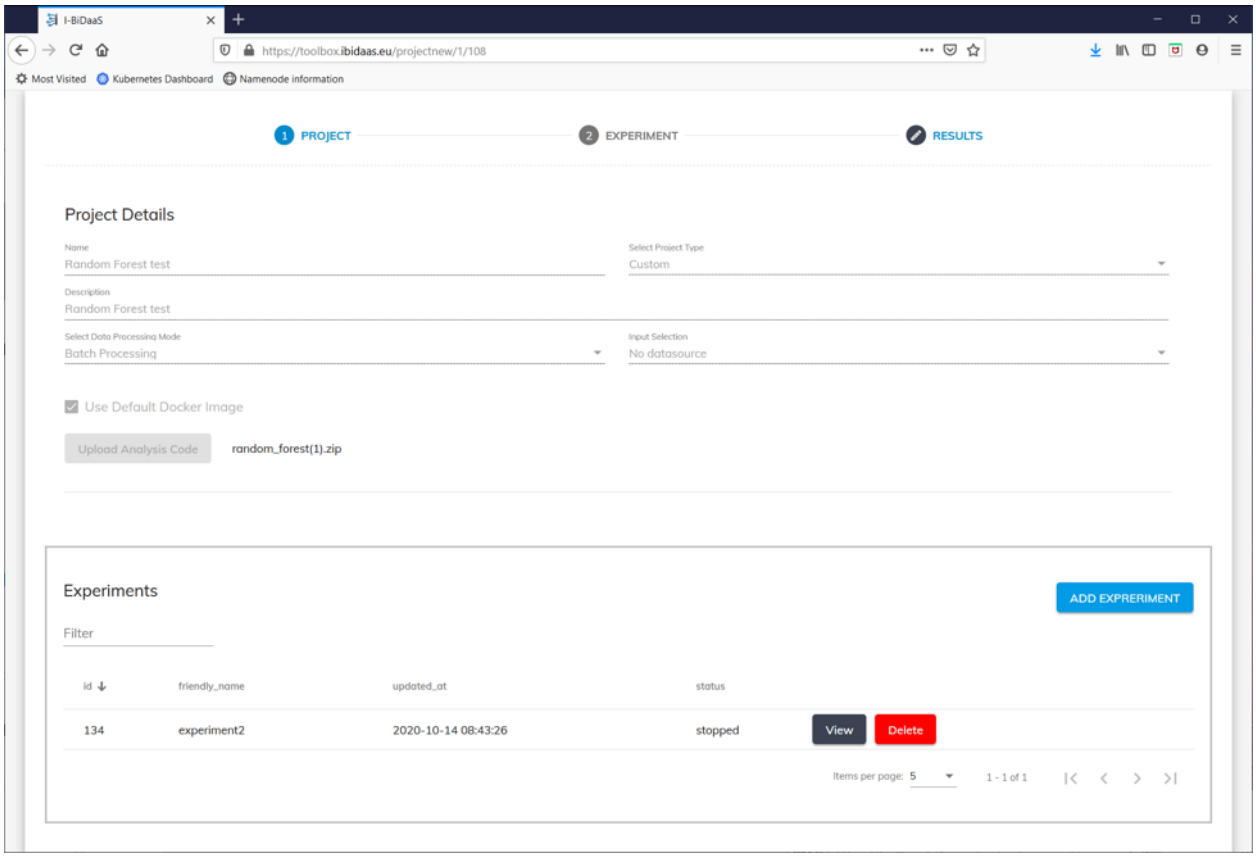

Figure 20. Project details (II)

By selecting “View” you can choose to download a tar ball containing the results (Figure 21).

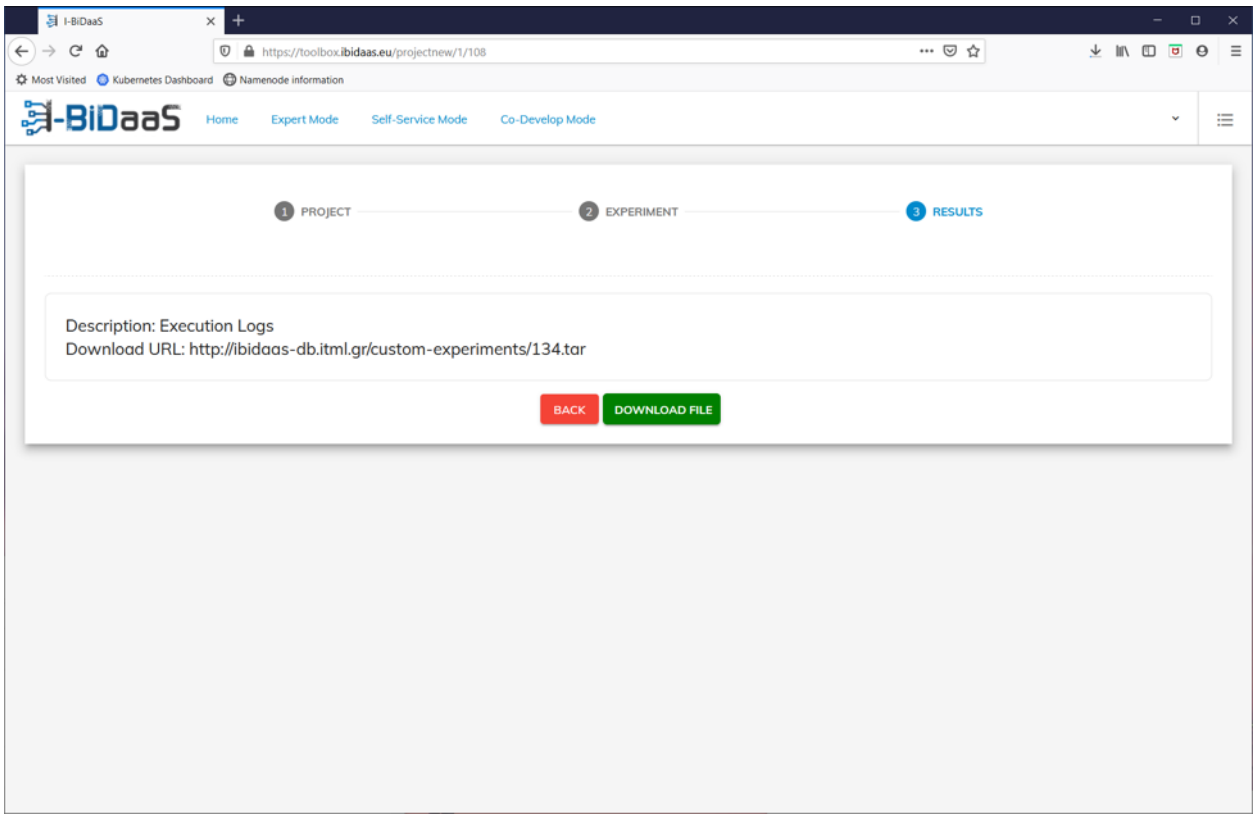

Figure 21. Experiment Results page

The tarball has the following file structure:

- Command.txt: The actual command line used to run the experiment
- Output.txt: Debugging output of the docker swarm. A successful run should look like the screenshot in Figure 22, containing the docker stack creation, execution and removal of the docker stack
- custom-experiment-XXX-results/application_log.out: the output to stdout of the app. Results could be printed there
- custom-experiment-141-results/application_log.err: the output of the stderror of the app

Any other file produced by the script will be also included in the file structure.

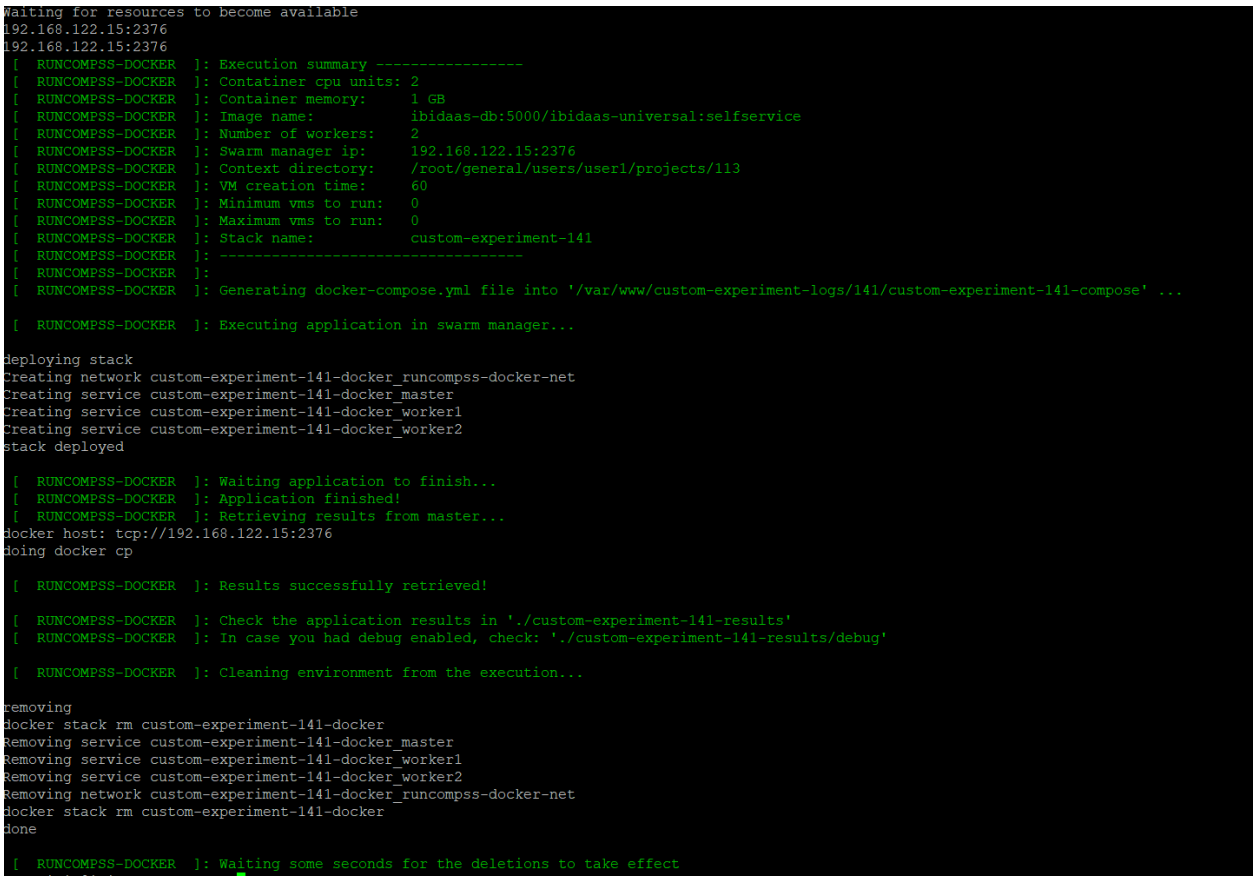

Figure 22. Output.txt contents

In case you chose to run the random_forest template code, which loads a predefined dataset provided with the dislib library, the classification output will be printed (along with a lot of debugging messages in application_log.out (Figure 23).

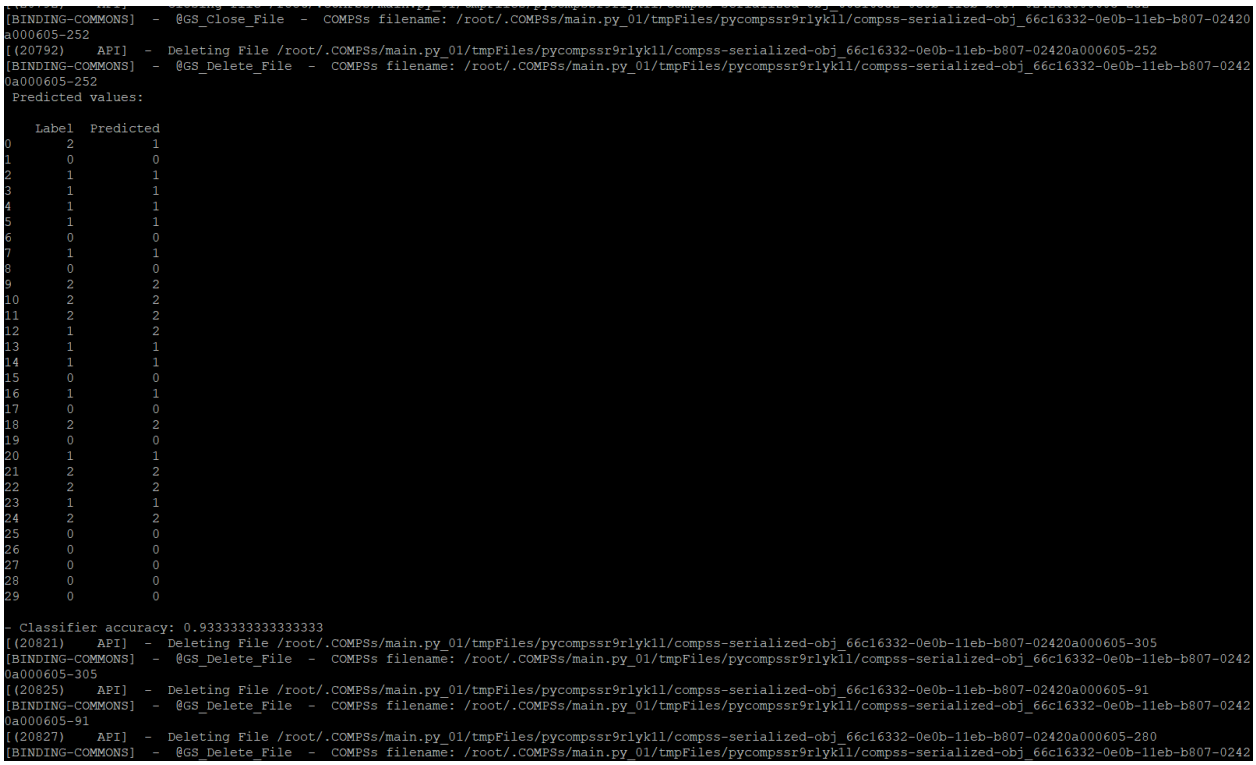

Figure 23. Sample of the successful execution of random forest template

3.3 Walk Through Expert Mode

In this mode, users are able to execute their own experiments by using both their own data and custom code. When entering the expert mode, a list with the previously created projects shows up as seen bellow.

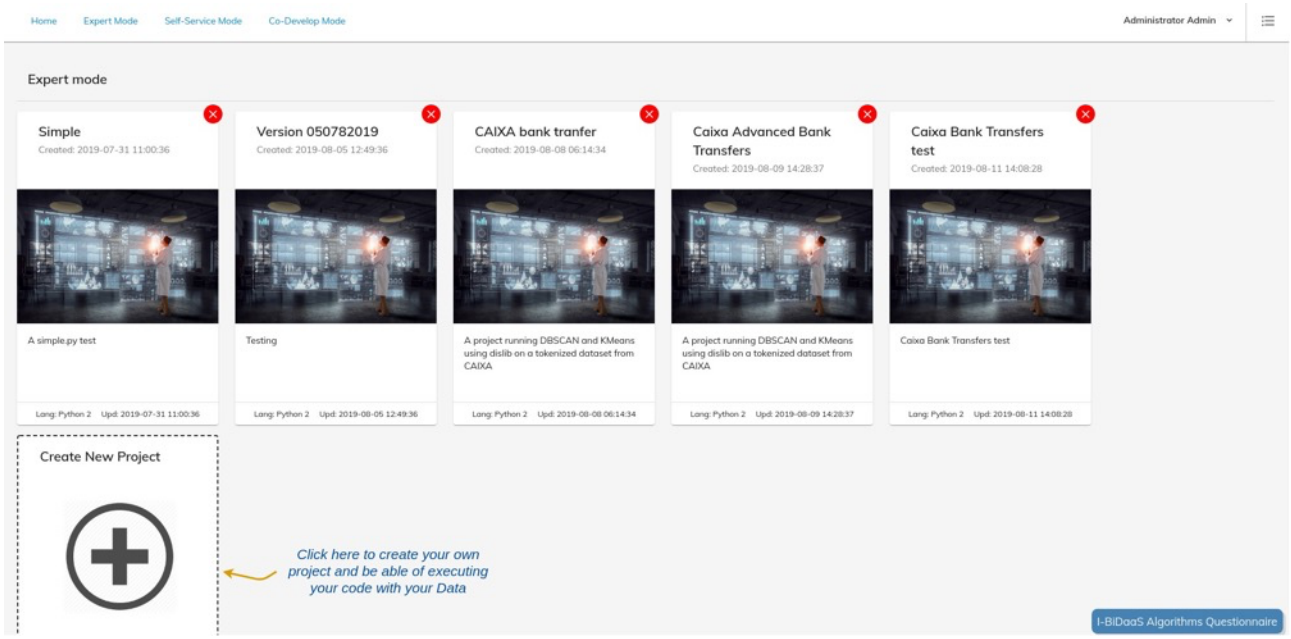

Figure 24. Expert Mode projects

The title, creation and last updated dates and a short description of every project are available in this list. Users have the followings options:

- Add a new project
- Add and run a new experiment in an existing project
- View results of previous experiments in a project
- Delete a project

When creating a new project, the projects details must be filled in. These include the name of the project, a description, the data processing mode and the input type. Moreover, users can choose to use the default docker image for their experiments or point out the name of another docker image from the DockerHub. Finally, the analysis code must be put in a zip file and be uploaded using the given button. Users have the possibility to download available code templates and upload new versions of these ones for their project. These code templates provide sample implementations of various algorithms that exploit the capabilities of the I-BiDaaS platform.

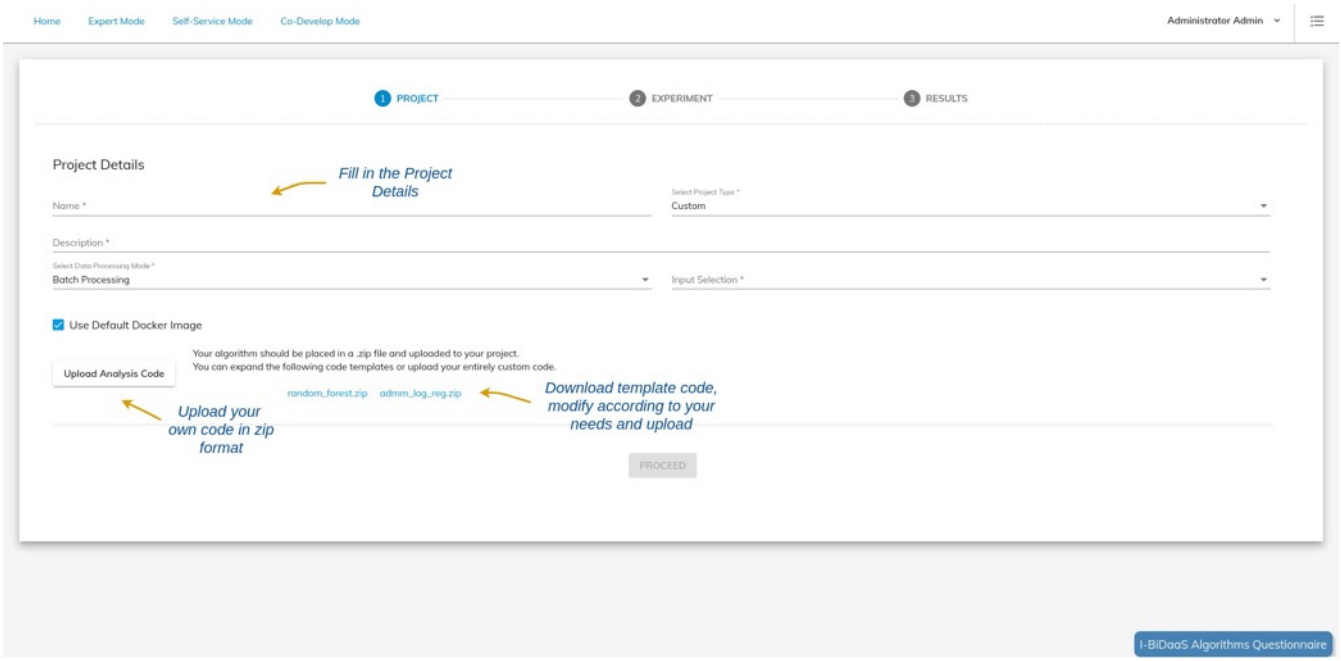

Figure 25. Expert Mode-Project Details

Once project details are set, users can move forward to the next step which is about the experiment details by clicking the proceed button. In this step, users have to specify information such as the name of the experiment, the actual command that has to be executed (including any required arguments) and the required computational resources. Figure 26 show the relevant information filled in using a simple python program whose task is to increase the given numerical input by 1 and return the result.

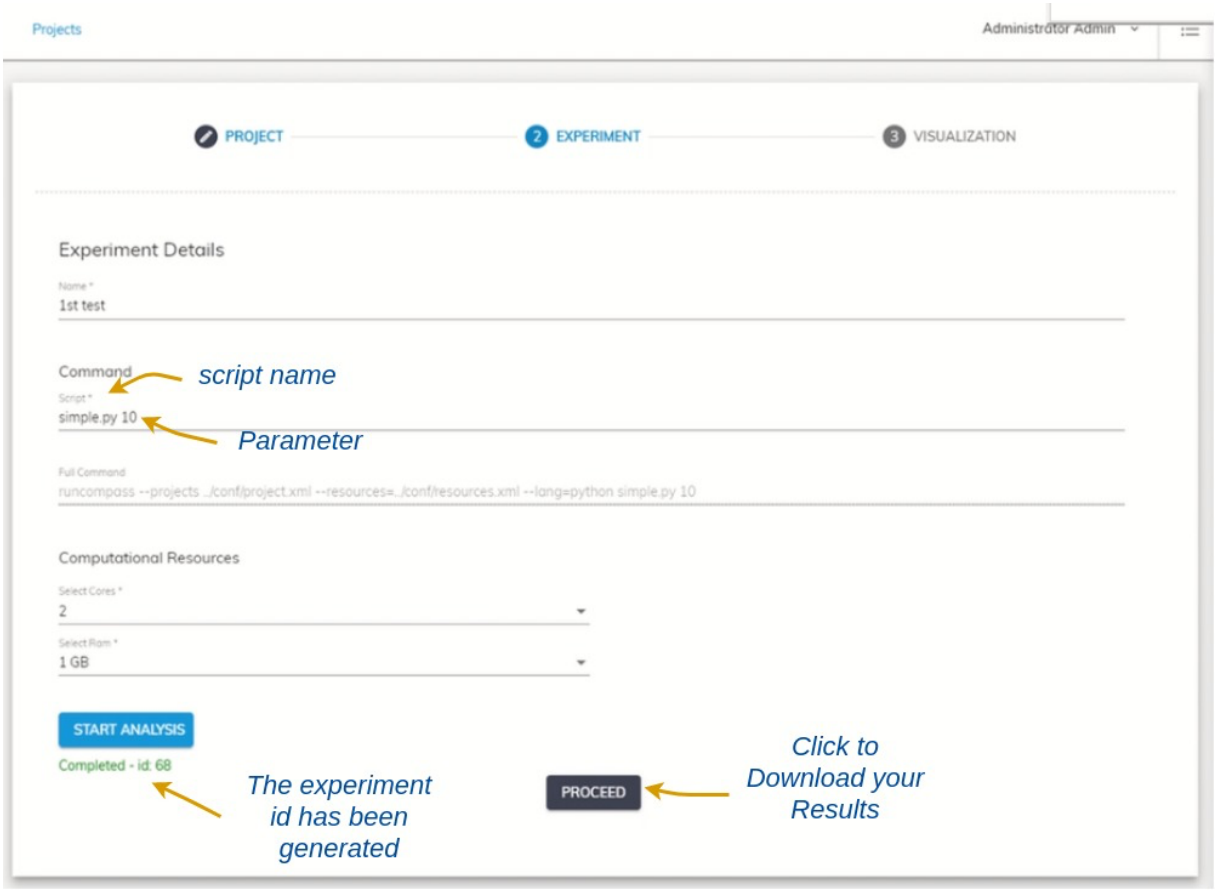

Figure 26. Expert Mode-Experiment Details

Once all the details have been filled in and the start analysis button is clicked, the execution of the experiment starts. Once the analysis ends, the experiment id and a completion message appear. Results (execution logs) are now available in the final step of the experiment (Figure 27).

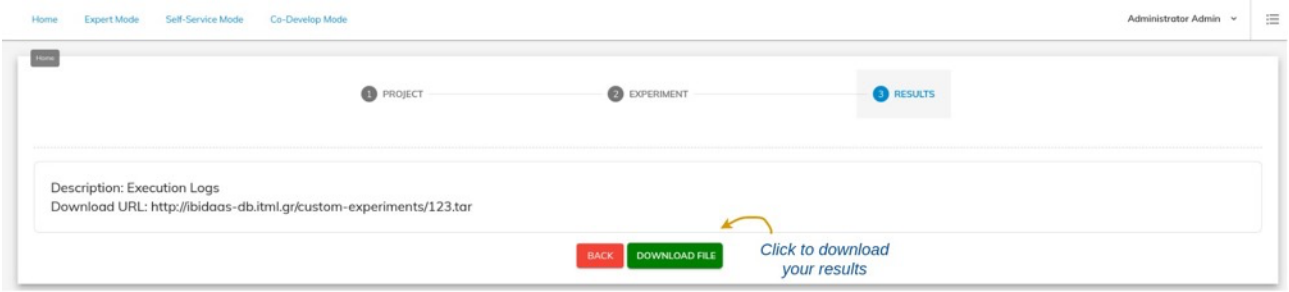

Figure 27. Expert Mode- Download results

The zip file which contains all the logs of the experiment execution has the same name as the experiment id. Inside this zip there is a folder named “custom-experiment-id-results”. In this folder there is a text file named “application_log.out” ().

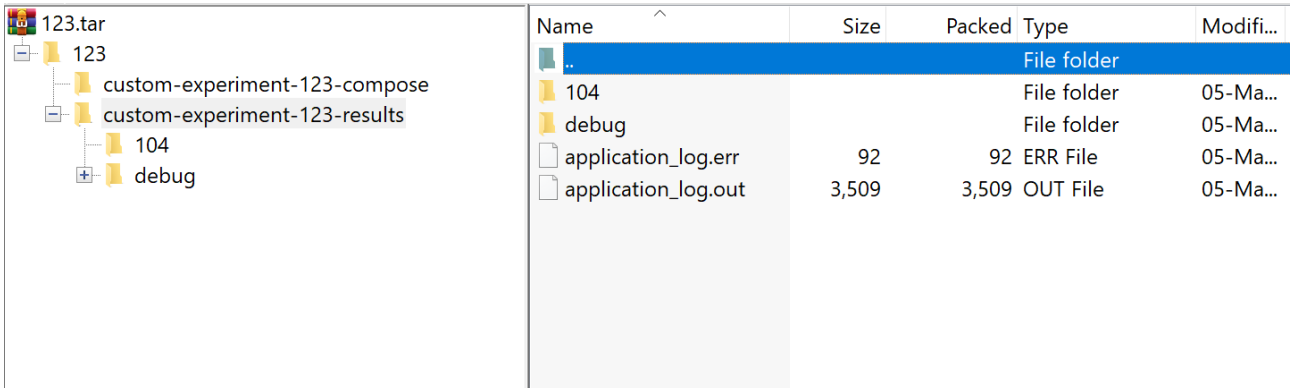

Figure 28. Expert Mode-Results Structure

As mentioned before our simple script prints out the number given as execution argument as well as the numerical result of the number increased by 1 as shown in Figure 29. A list of all the rest of the steps involved in running the experiment are also printed out and inform the user on the execution process.

```
Sourcing generate_project.sh
[  INFO] Using default execution type: compss
[  INFO] Relative Classpath resolved: /root/conf/StorageItf-1.0-jar-with-dependencies.jar:

----------------- Executing simple.py --------------------------

INFO: Storage API successfully imported.
Binding debug is activated
[BINDING-COMMONS]  -  @GS_On  -  Creating the JVM
[BINDING_COMMONS]  -  @create_vm  -  reading file in JVM_OPTIONS_FILE
[BINDING_COMMONS]  -  @create_vm  -  Launching JVM
[BINDING_COMMONS]  -  @create_vm  -  JVM Ready
WARNING: COMPSs Properties file is null. Setting default values
[BINDING-COMMONS]  -  @GS_On  -  Creating runtime object
[(984)    API]  -  Deploying COMPSs Runtime v2.4.rc1905 (build 20190527-1444.rnot-found)
[BINDING-COMMONS]  -  @GS_On  -  Calling runtime start
[(992)    API]  -  Starting COMPSs Runtime v2.4.rc1905 (build 20190527-1444.rnot-found)
[(992)    API]  -  Initializing components
[(1847)    API]  -  Ready to process tasks
[BINDING-COMMONS]  -  @Init JNI Methods
[BINDING_COMMONS]  -  @Init JNI Direction Types
[BINDING_COMMONS]  -  @Init JNI Stream Types
[BINDING-COMMONS]  -  @Init JNI Types
[BINDING_COMMONS]  -  @Init DONE
[BINDING_COMMONS]  -  @Master JNI Init DONE
[BINDING-COMMONS]  -  @GS_Get_AppDir - Getting application directory.
[BINDING-COMMONS]  -  @GS_Get_AppDir  -  directory name: /root/.COMPSs/simple.py_01/
Initial counter value is 10
[BINDING-COMMONS]  -  @GS_RegisterCE - Registering Core element.
[(1919)    API]  -  Registering CoreElement simple.increment
[(1920)    API]  -       - Implementation: simple.increment
[(1920)    API]  -       - Constraints   :
[(1920)    API]  -       - Type          : METHOD
[(1921)    API]  -       - ImplTypeArgs  :
[(1921)    API]  -              Arg: simple
[(1921)    API]  -              Arg: increment
[BINDING-COMMONS]  -  @GS_RegisterCE  -  Task registered: simple.increment
[BINDING-COMMONS]  -  @GS_ExecuteTaskNew - Processing task execution in bindings-common.
[BINDING-COMMONS]  -  @GS_ExecuteTaskNew  -  Processing parameter 0
[BINDING_COMMONS]  -  @process_param
[BINDING-COMMONS]  -  @process_param  -  ENUM DATA_TYPE: 9
[BINDING-COMMONS]  -  @process_param  -  File: /tmp/counter
[BINDING-COMMONS]  -  @process_param  -  ENUM DIRECTION: 2
[BINDING-COMMONS]  -  @process_param  -  ENUM STREAM: 3
[BINDING-COMMONS]  -  @process_param  -  PREFIX: null
[BINDING-COMMONS]  -  @process_param  -  NAME: filePath
[(1938)    API]  -  Creating task from method simple.increment
[(1938)    API]  -  There is 1 parameter
[(1938)    API]  -    Parameter 1 has type FILE_T
[BINDING-COMMONS]  -  @GS_Open_File  -  Calling runtime OpenFile method  for /tmp/counter and mode 2 ...
[(1947)    API]  -  Opening /tmp/counter in mode INOUT
[(5981)    API]  -  File target Location: /root/.COMPSs/simple.py_01/tmpFiles/d1v3_1583336076906.IT
[BINDING-COMMONS]  -  @GS_Open_File  -  COMPSs filename: /root/.COMPSs/simple.py_01/tmpFiles/d1v3_1583336076906.IT
Final counter value is 11
[BINDING-COMMONS]  -  @GS_Off
```

Figure 29. Expert Mode - Results

3.4 Walk Through Self-Service Mode

The Self-service mode allows industry in-house personnel which have domain knowledge and some insights about data analysis (non-experts) to easily construct Big Data pipelines in a user-friendly way, selecting a pre-defined data analytics algorithm from an available list. Figure 30 below presents the current set of available algorithms that user may choose to run an experiment:

- LASSO ADM
- K-Means Prediction
- K-Means Evaluation

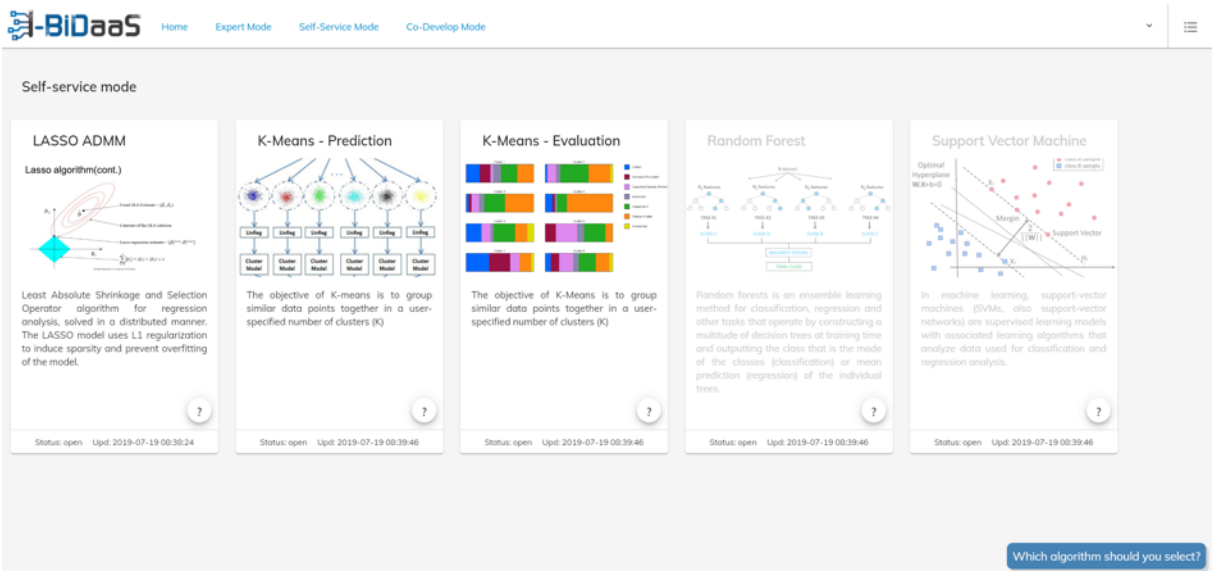

Figure 30. Algorithms in Self-Service Mode

When selecting one of the available algorithms, a new project will be created. The project details form contains predefined values that are specific for the chosen algorithm. Moreover, a table with all past and current experiments of the project is presented below the project details. For each existing experiment users can:

- Stop it if it is still running
- View its details and results
- Delete it

Figure 31 shows the project page that shows up after selecting 'K-means prediction' in the previous screen.

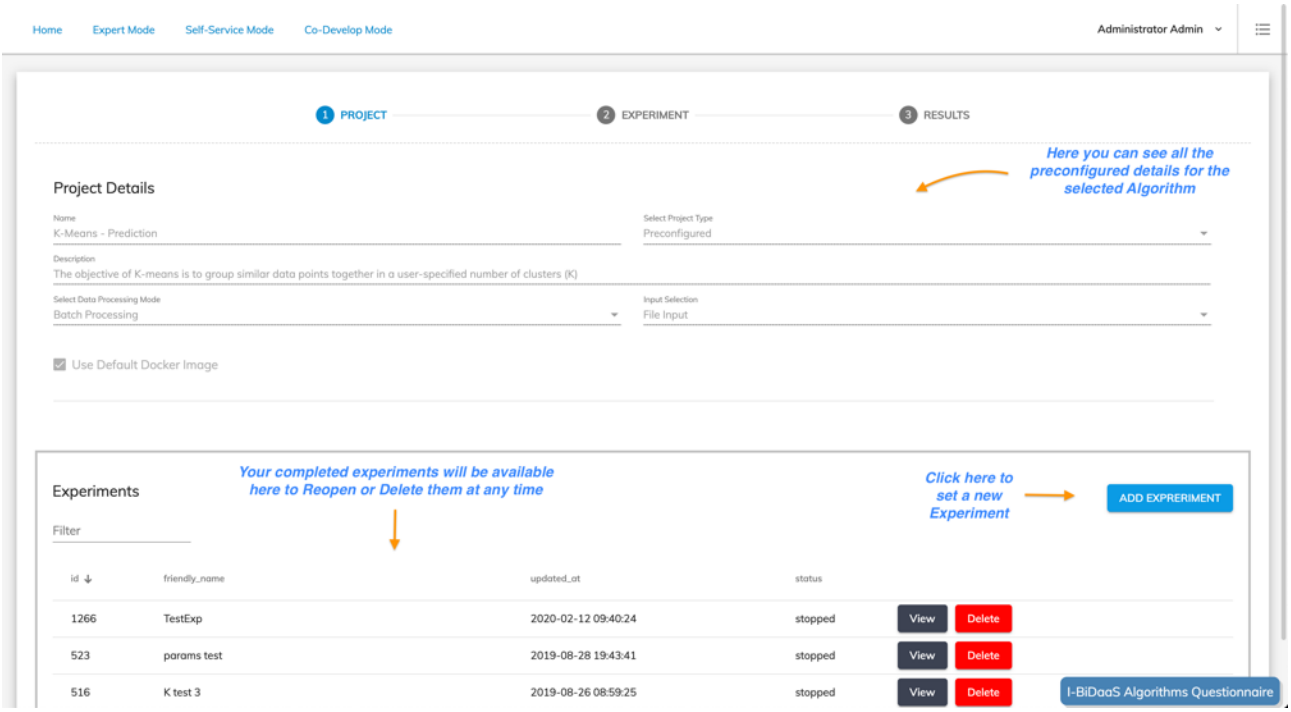

Figure 31. Project page for 'K-means prediction' algorithm

Clicking the "add experiment" button brings up the experiment setup screen (Figure 32). The name, data source, algorithm parameters and computational resources have to be specified at this step. The algorithm parameters come with a predefined set of default values to help users have a typical configuration of the selected algorithm. Once the details are defined and the 'start analysis' button is clicked, the analysis of the given data starts.

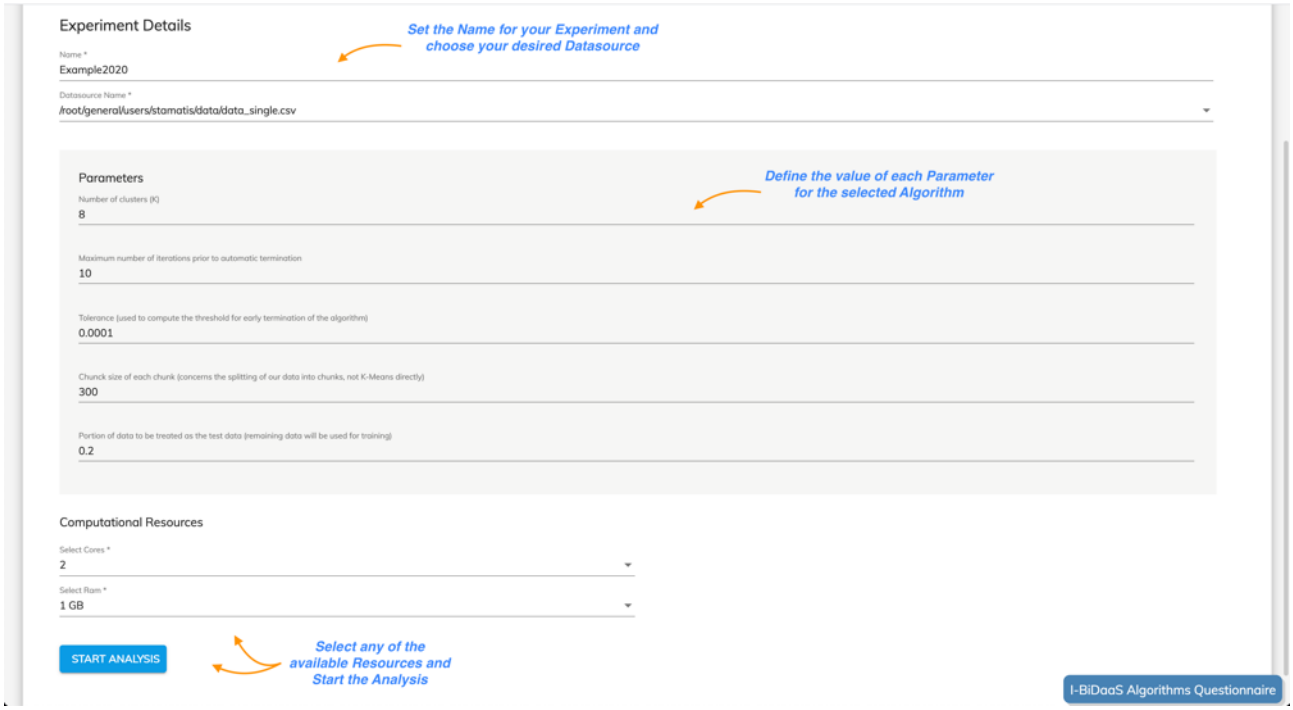

Figure 32. Experiment Details for Selected Algorithm

Upon completion of the experiment, the 'proceed' button is activated and users can navigate to the results page (Figure 33). The results provide visualisations of the analysis results which may vary depending on the selected algorithm. For example, if the selected algorithm is 'K-Means Evaluation' (Figure 34) information about clustering is shown by means of an interactive scatter chart which presents the different identified clusters and their centroids. Users are also given the option to download results which are packaged together with all the execution logs in a zip file as described in the previous paragraph (Figure 35) .

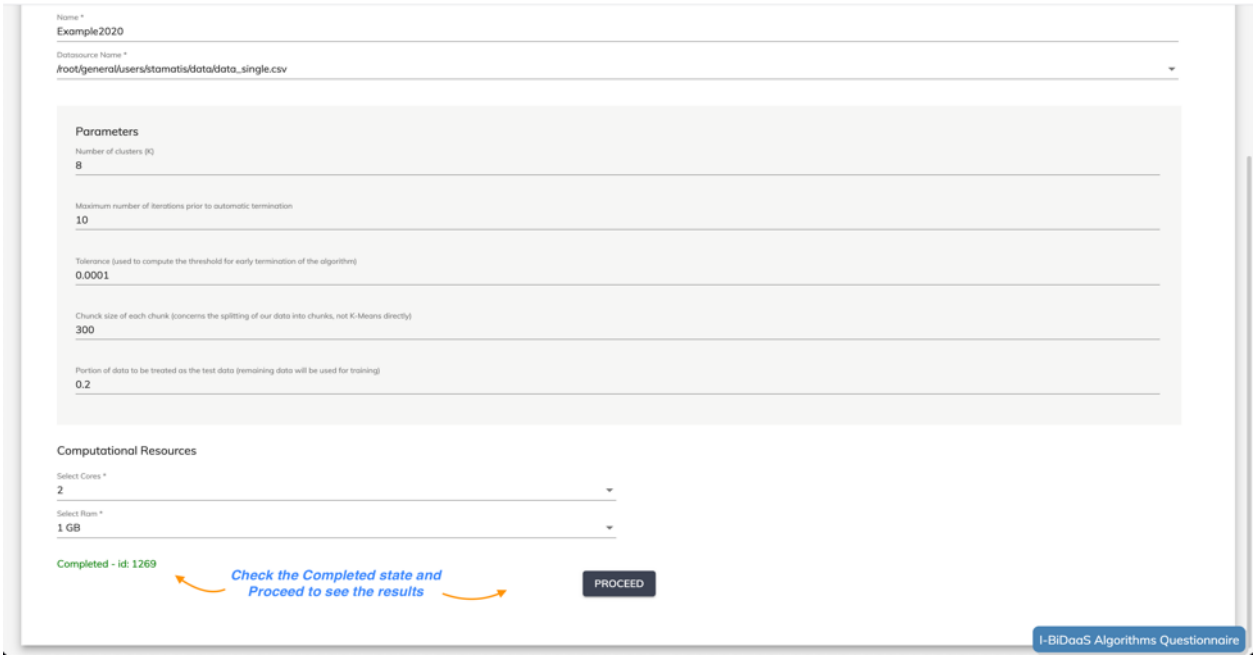

Figure 33. Experiment Completion

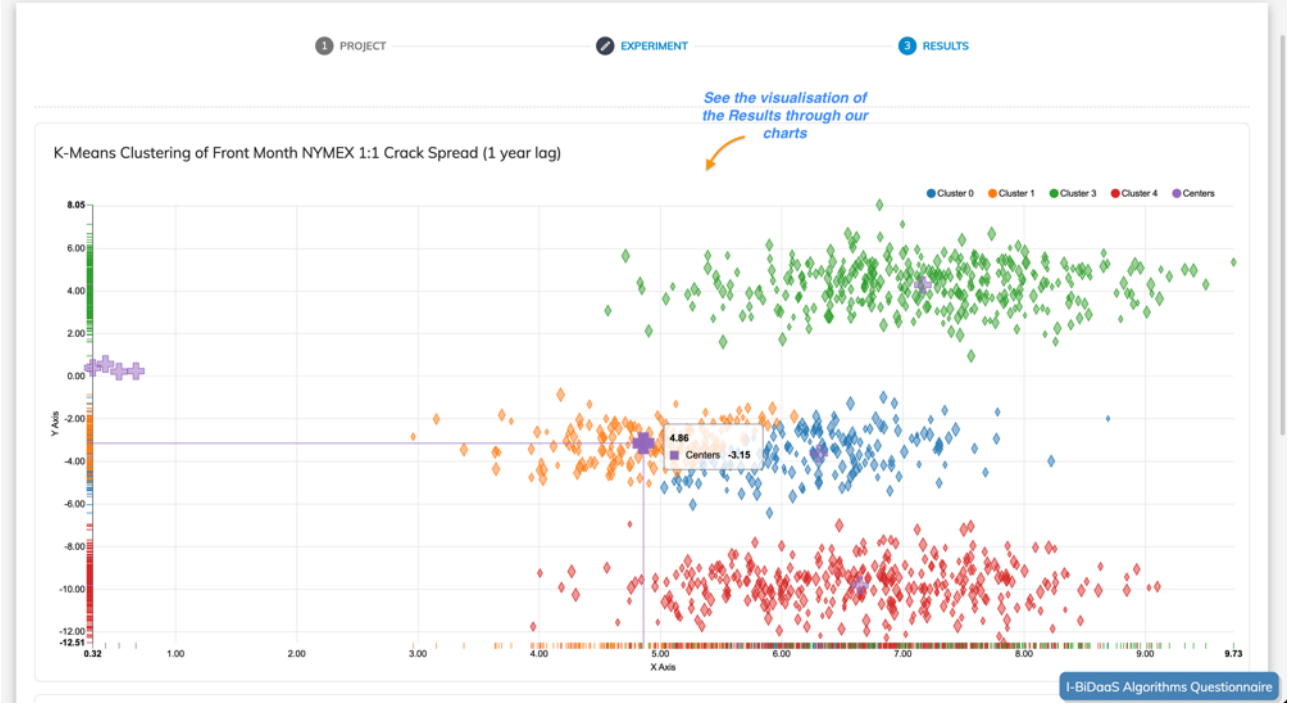

Figure 34 Visualisation of algorithm results

Figure 35. Downloading of experiment results

3.5 Walk Through Co-Developed Mode (Success Stories)

The Co-Develop mode corresponds to an end-to-end solution for a given industry project developed by the I-BiDaaS team (the I-BiDaaS use cases). Projects in this mode involve minimum user interaction since both dataset formats and analysis setup are predefined and custom-made according to the needs of the project's pilot partners. The idea is to present meaningful examples of how the I-BiDaaS platform and tools can be applied to real-world applications based on real business needs.

The following paragraphs present such examples for the three industry pilots participating in I-BiDaaS, namely Caixa Bank (finance), Telefonica ID (telecommunications) and CRF (manufacturing).

3.5.1 Caixa Use Case – IP Relationships

The Caixa use case is about identifying potentially fraudulent or suspicious transactions between users based on relationships of the IPs used to perform these transactions. The use case involves two types of analysis. The first one relies on batch processing of a dataset including bank information about bank transactions, while the second one operates on a data stream and showcases the real-time identification of IP relationships in the given stream of data.

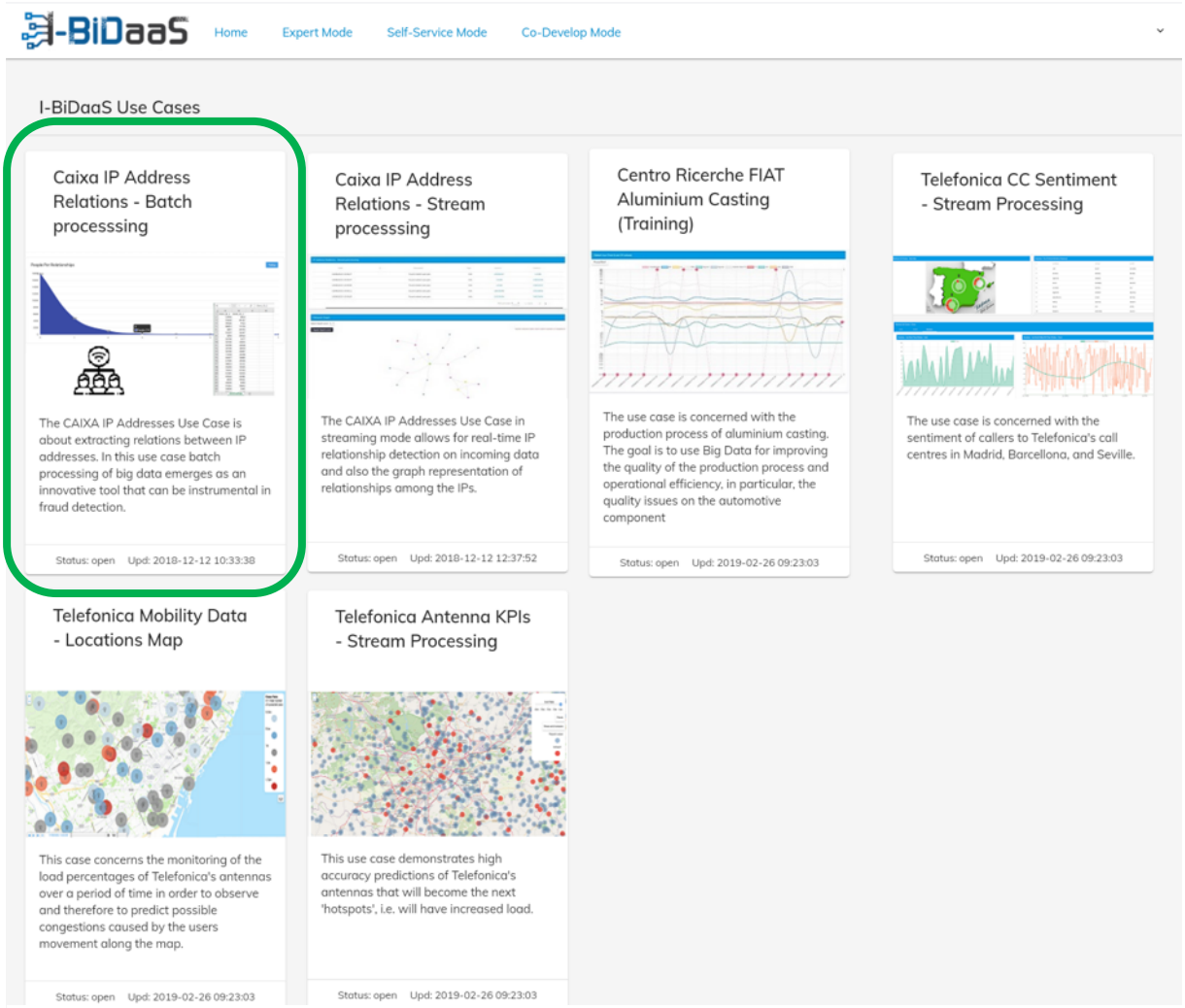

Figure 36. Caixa IP Relationships - Batch mode

After selecting the batch mode option, users are able to choose between fabricated or tokenized data. The rest of the project details as well as the details of a new experiment follow the same logic described for the previous modes (Figure 37). However, in Co-Develop projects users have minimum configuration options since the analysis pipeline is already defined in the backend (Figure 38).

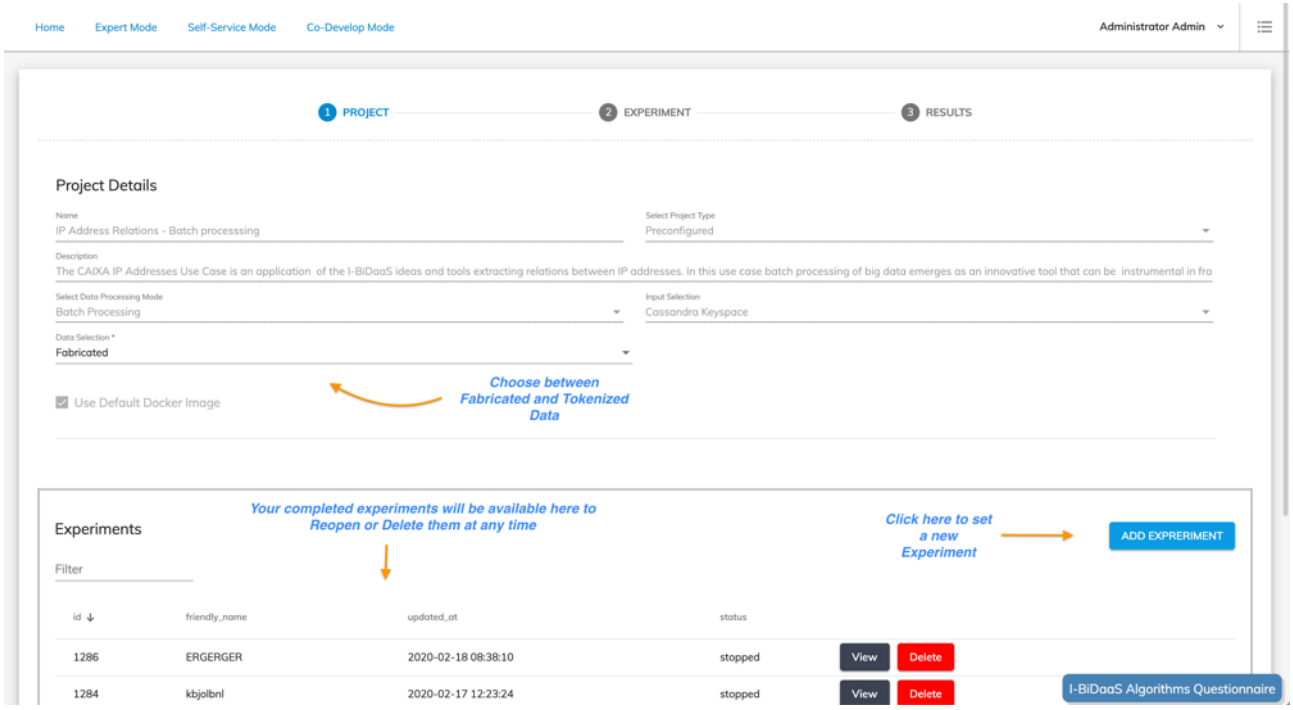

Figure 37 Batch Processing: New Experiment

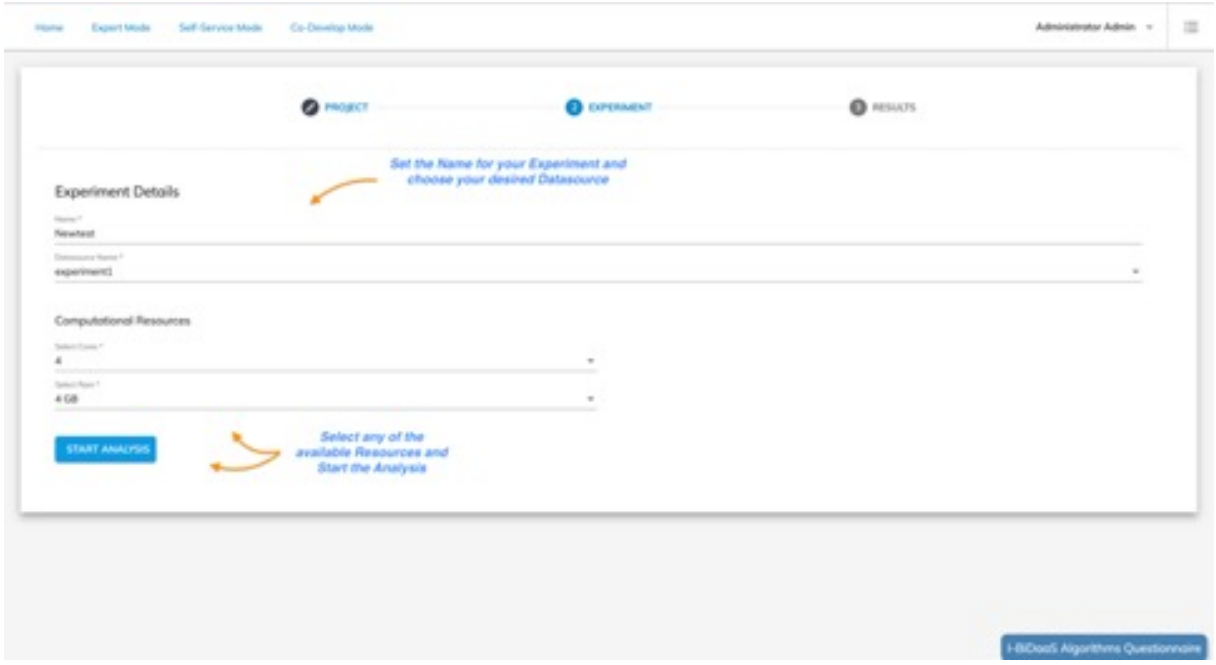

Figure 38. Batch Processing: Experiment Details

Once the experiment is completed, a visualization of the results is available. Figure 39 shows the identified groups of people having 1,2 or more relationships, as discovered by the underlying processing. Once more, results are available for downloading for further analysis if desired.

Figure 39. Batch Processing: Experiment Results

The second operational mode of the Caixa use case includes the data stream processing (Figure 40) which does not resemble the typical project setup.

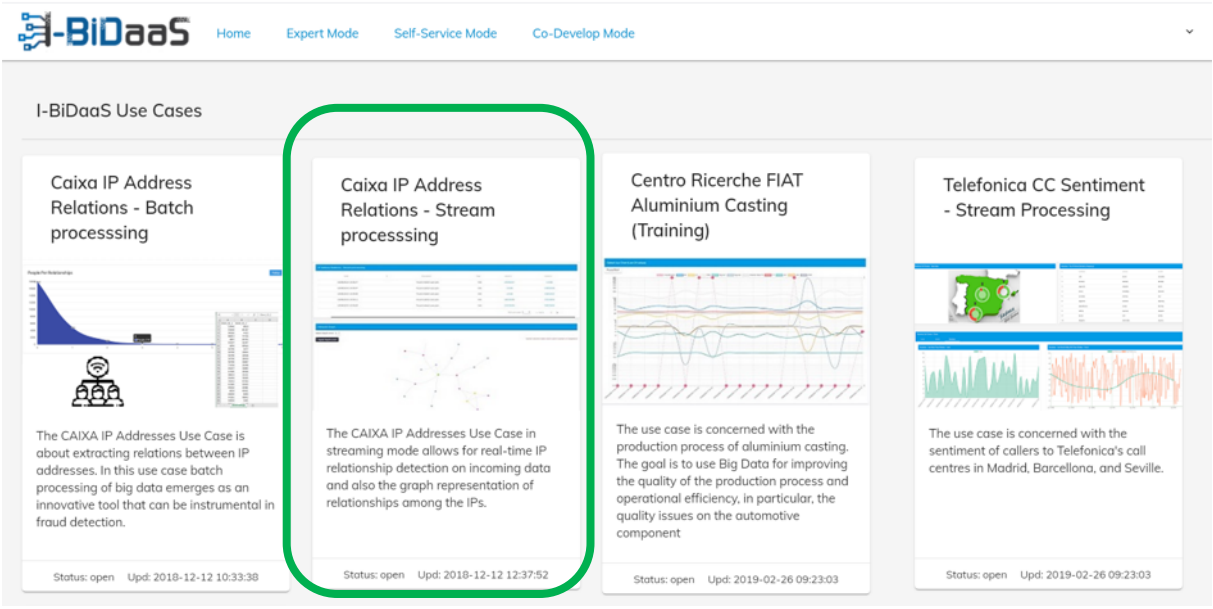

Figure 40. IP Relationships: Streaming

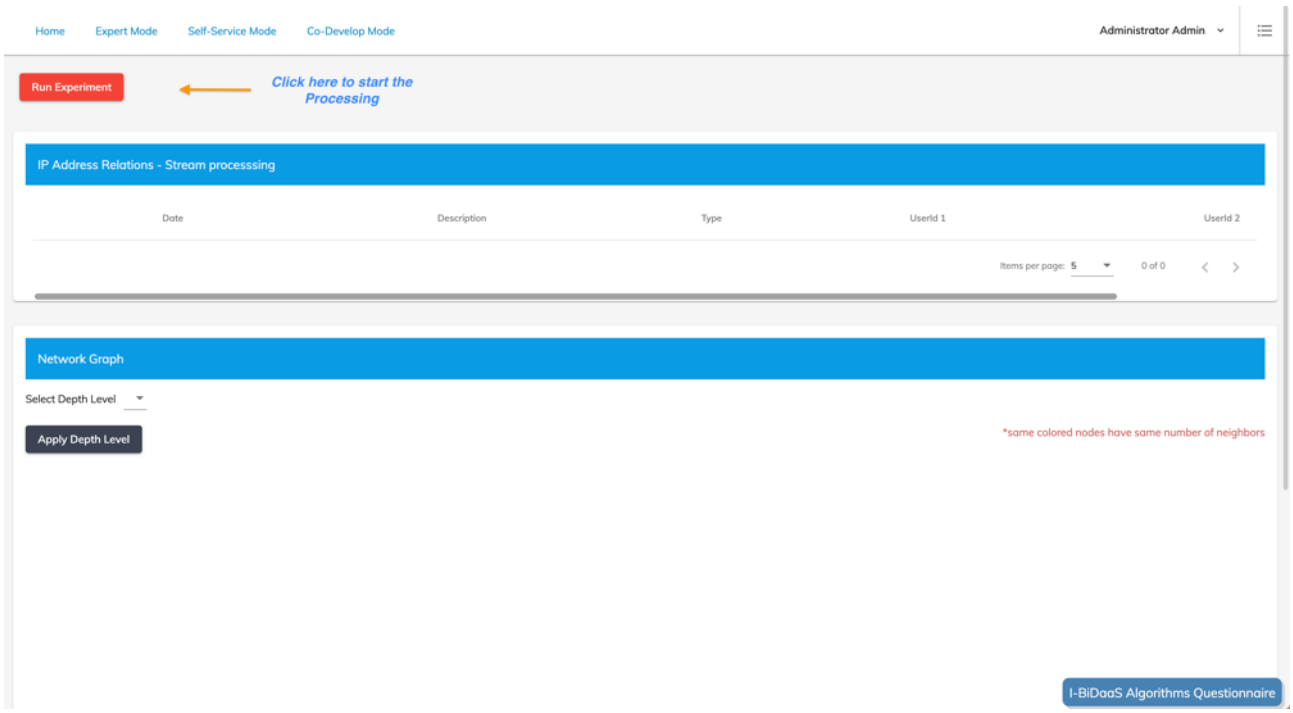

Figure 41. IP Relationships: Start Stream analysis

Figure 41 presents the starting page of the streaming analysis project. The only available option is the 'Run Experiment' button which triggers the data stream processing. A stream of couples of user identification numbers is pushed to the system and the running analysis decides whether these two users are related based on the datasets of relationships extracted in the previous experiment (Figure 42).

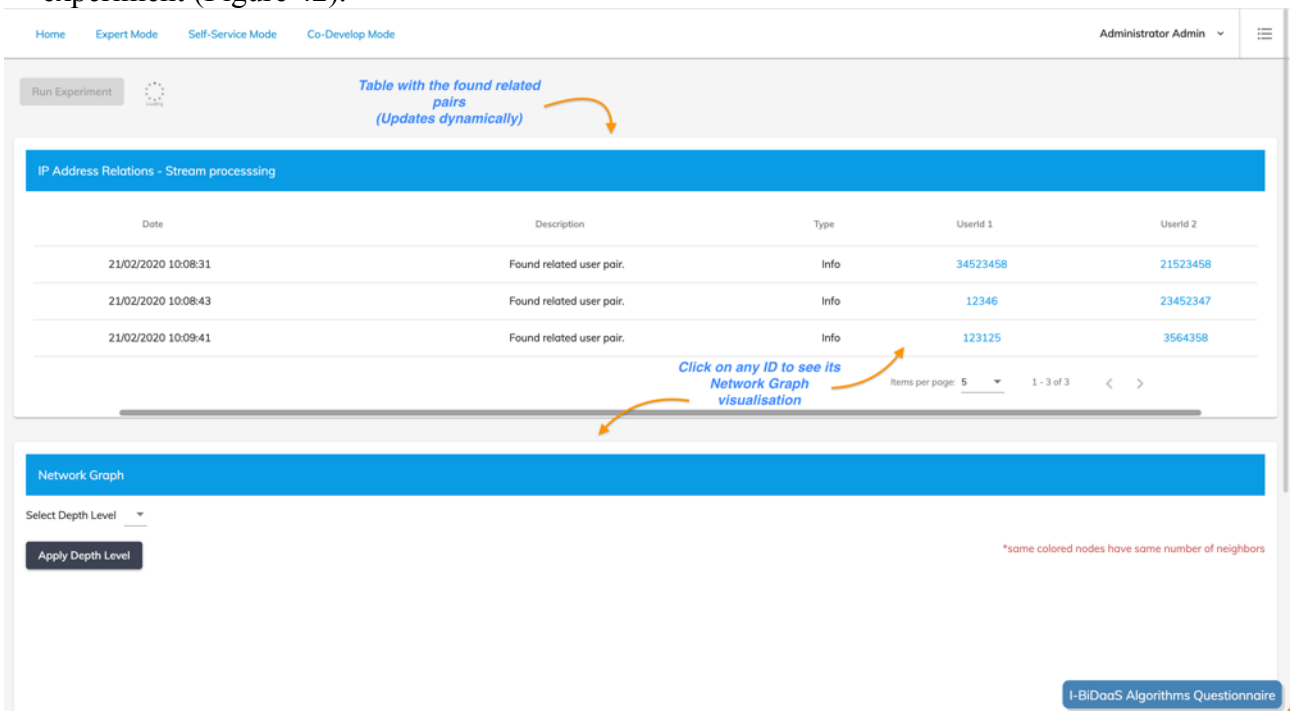

Figure 42. IP Relationships: running stream processing

Furthermore, operators can explore all other relationships of an identified user by exploring a graph representation of the user and their peers (Figure 43). The interface also allows the

selection of the graph's depth (Figure 44), meaning that choosing to visualize the graph of relationships for a given user with depth 2 would present the user with his immediate associated users (depth 1 neighbors) as well as the associated users of the neighbors (depth 2 neighbors). Such a visualization allows operators to quickly explore relationships in an interactive way and can provide a comprehensive overview of the network of any given user.

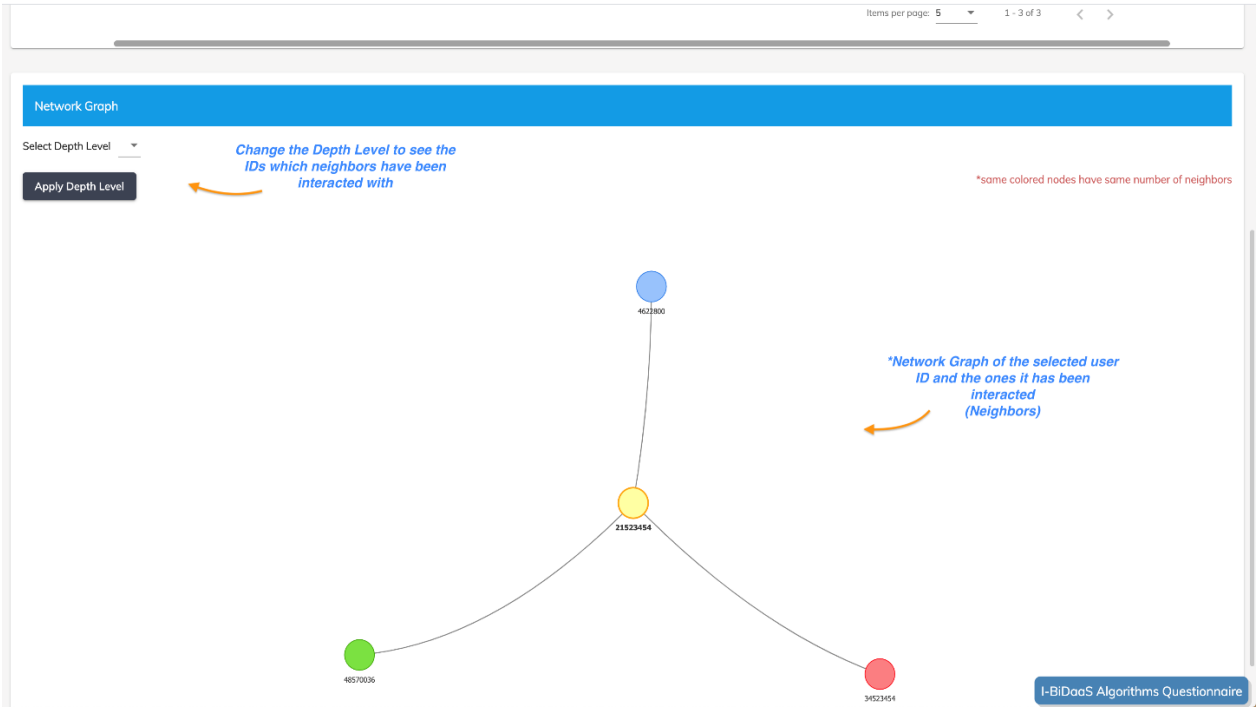

Figure 43. IP Relationships: Network Graph

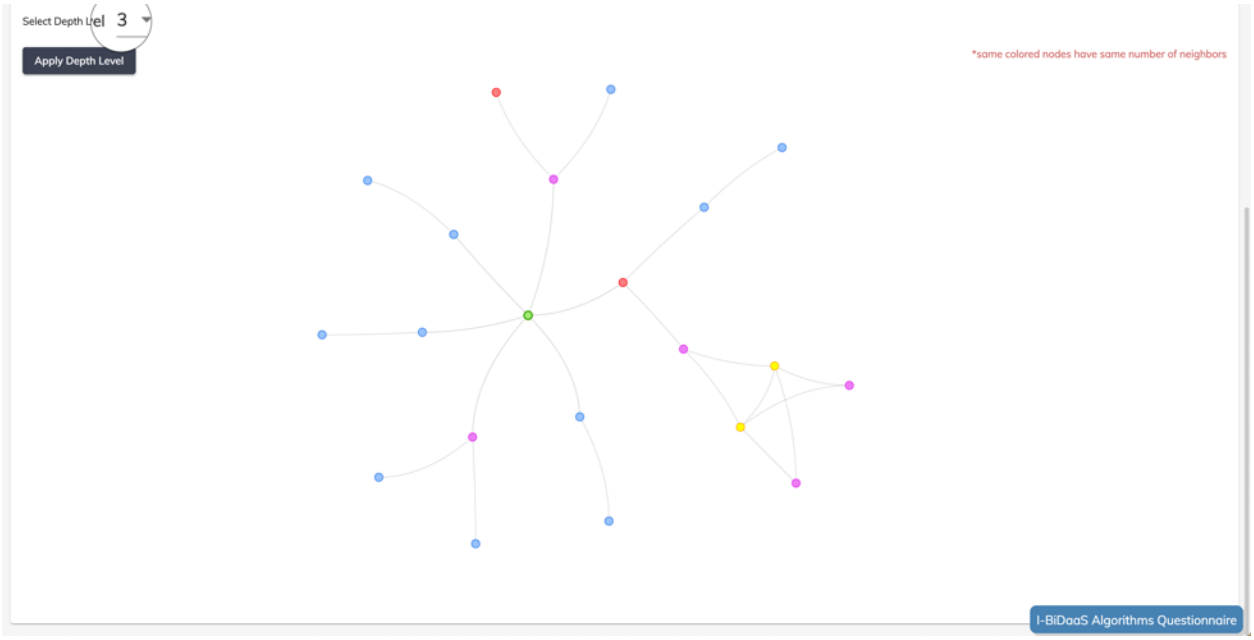

Figure 44. IP Relationships: Network graph of depth 3

3.5.2 CRF Use Case: Aluminium Casting

The CRF use case is about the production process of aluminium casting. The goal of this use case is to improve the quality of the processing by analyzing streaming data coming from sensors mounted on the machinery that execute the process.

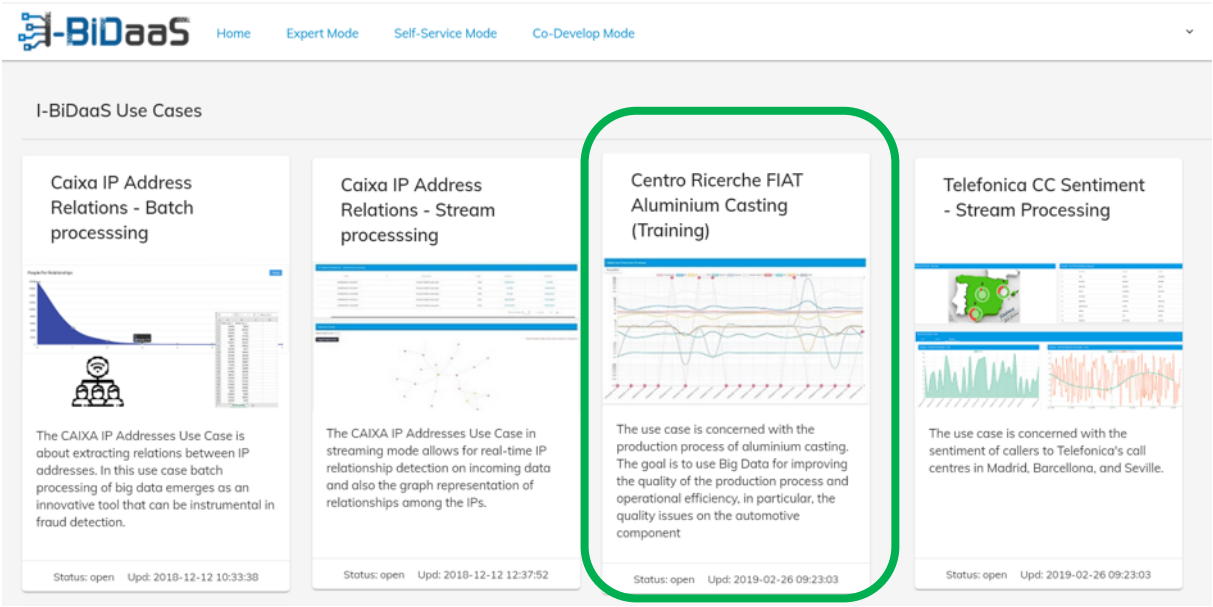

Figure 45. CRF Use Case: Aluminium Casting

The project starting page provides the option to 'Run the Experiment', similarly to the previous use case of Caixa (Figure 46). Once started, the experiment receives a stream of data records containing the values of a set of monitored metrics. These are actually the sensor measurements that monitor various production machine features, presented in an anonymized way. The stream also includes a special value for each of these records which represents the final quality classification result for the record. This means that depending on the rest of the values, the final product of this process is classified in one of the three defined levels of quality, namely 0,1 or 2 (given in an anonymized manner). The overall aim of the experiment is to monitor the received values and watch which of the configuration produce the best classification of the end product.

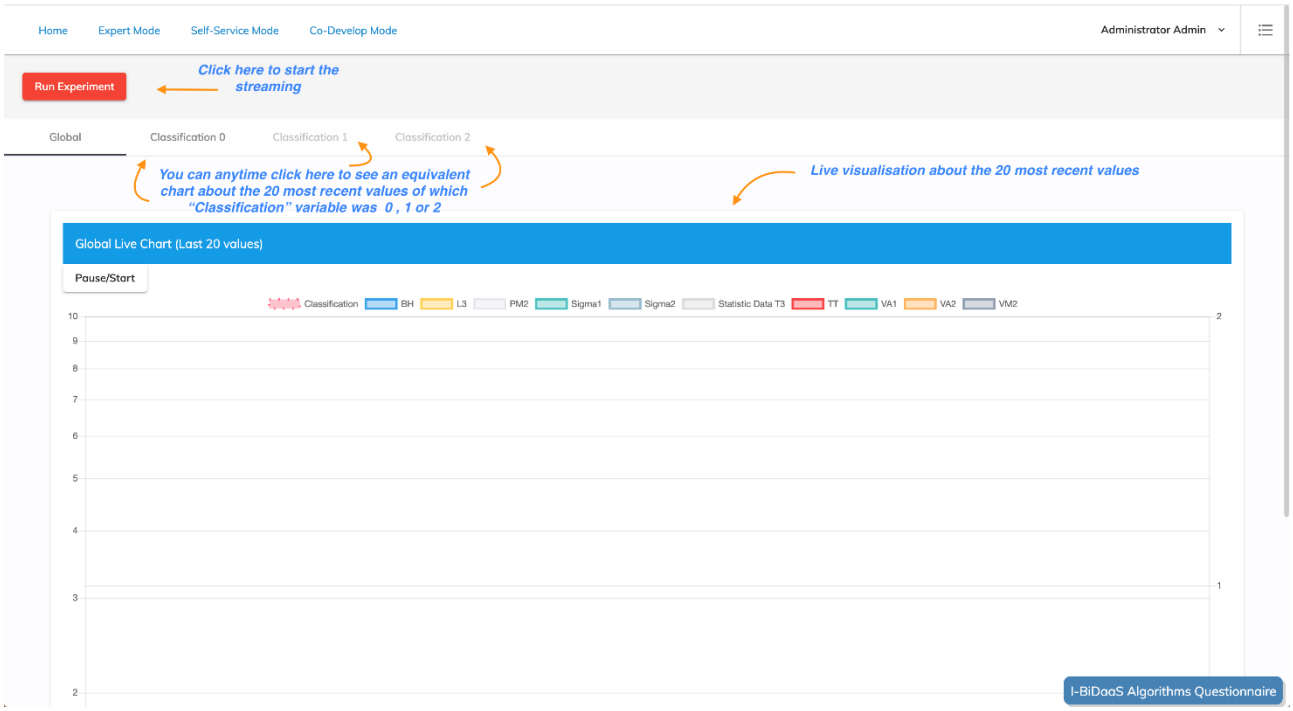

Figure 46. CRF: Starting the experiment

The 10 coloured rectangular boxes are associated to the aforementioned metrics. The first coloured box represents the classification result for every record. After clicking the 'Run experiment' button, the data streaming starts, and the chart displays the received values in real time. The 20 most recent values of each variable are displayed while the user can pause/restart data reception at any time (Figure 47).

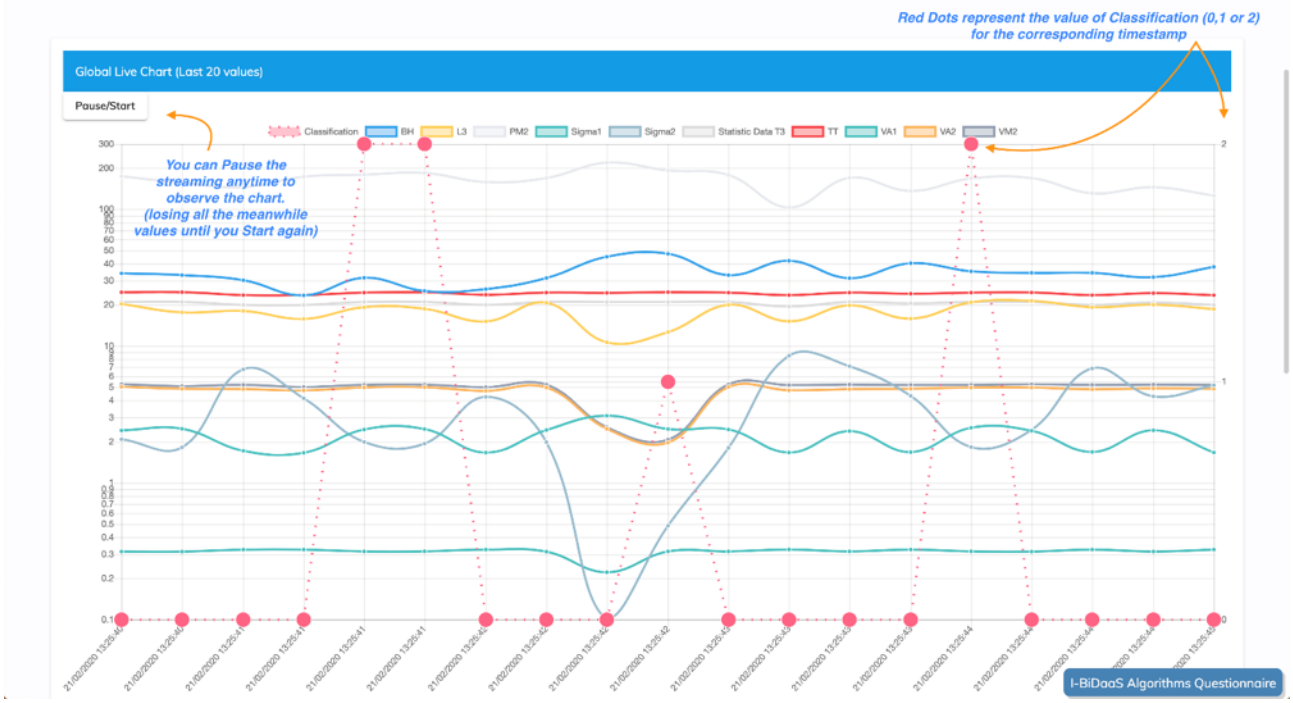

Figure 47. CRF: Real-Time data stream

The user is also able to watch the streaming values of variables for a specific classification. If a certain value of classification is selected (for example class 0, Figure 48) only the records resulting in this classification are displayed.

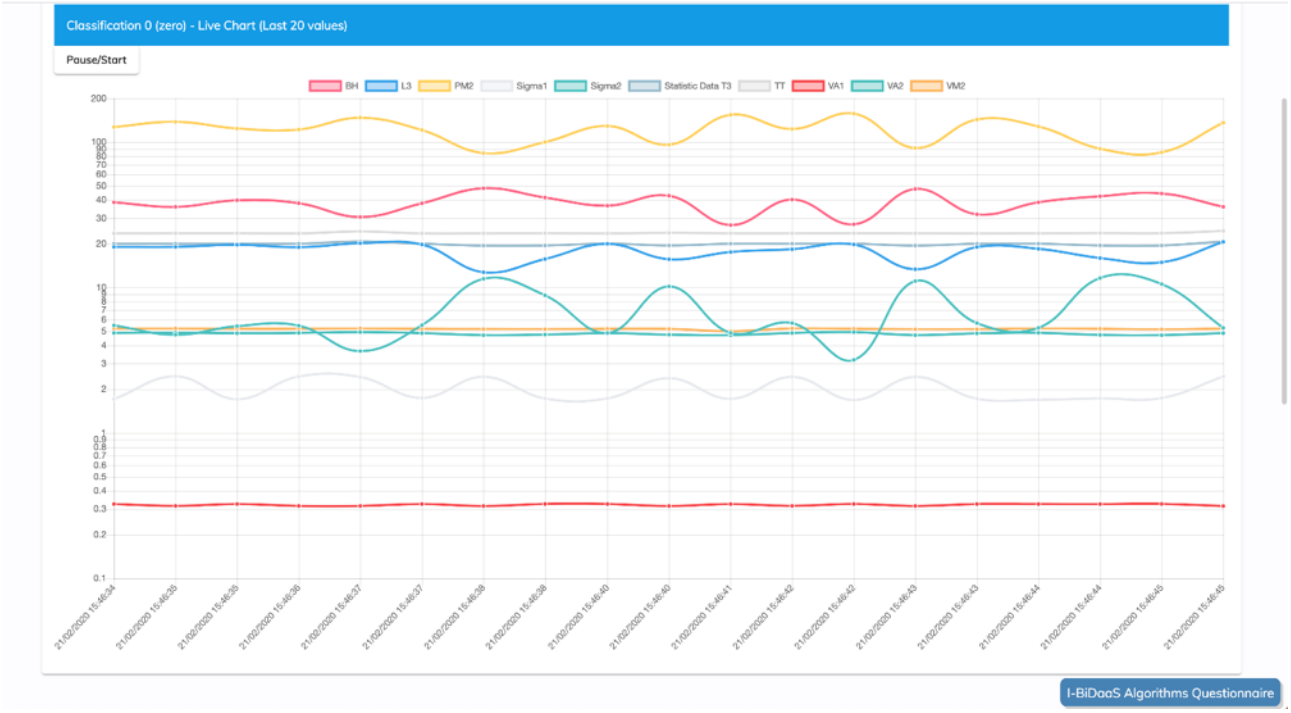

Figure 48. CRF: Classification Selection

In parallel, the dashboard of this use case includes a set of widget that get dynamically updated as the data stream is received. Figure 49 presents these widgets, namely a pie chart which displays the current percentages of every classification value, a metric table (on the top right) that displays the min, max and average value of a selected metric and finally a dynamic table which includes all the records of the data stream using a color code to denote the classification of each set of values.

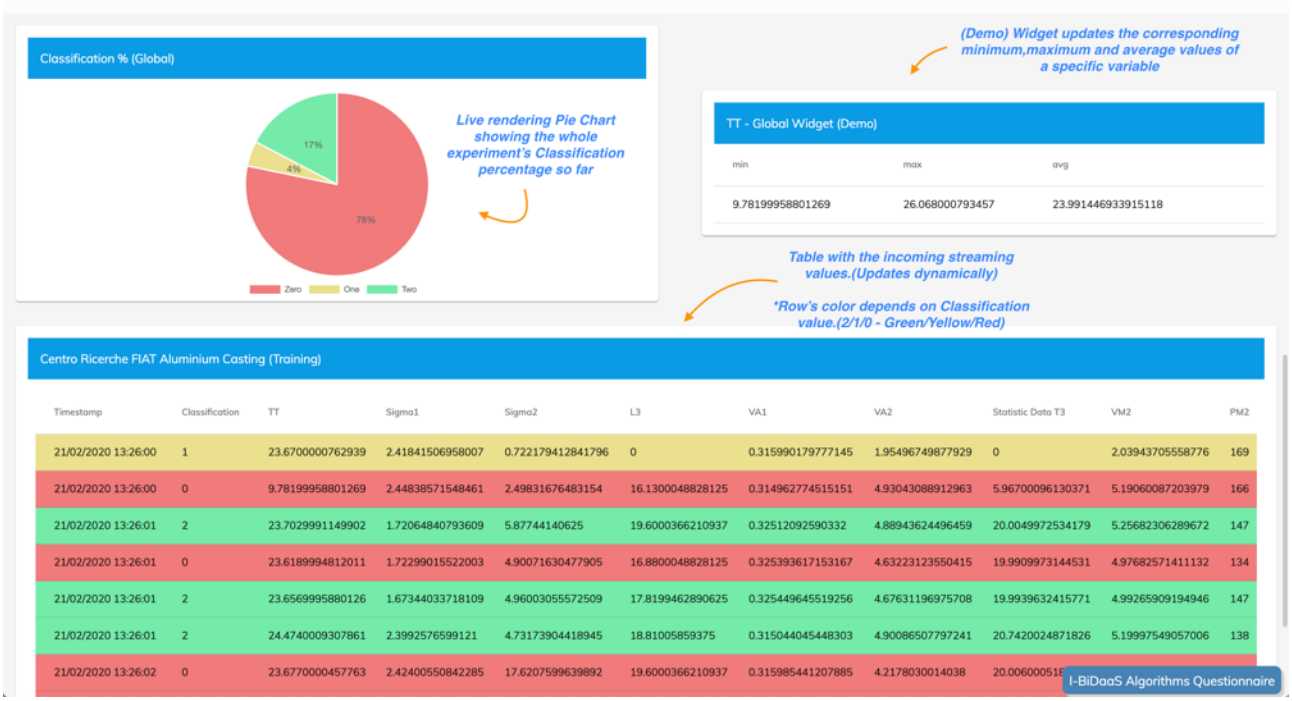

Figure 49. CRF: Visualisation widgets

3.5.3 Telefonica Use Case

The Telefonica (TID) use case (Figure 50) is about visualizing the sentiment analysis performed in TID's call centers . The values are coming as a data stream of positive, negative and dropped calls per call center. Data is anonymized and the displayed call centers names (cities) do not correspond to reality but are rather used to ease readability of the visualisations.

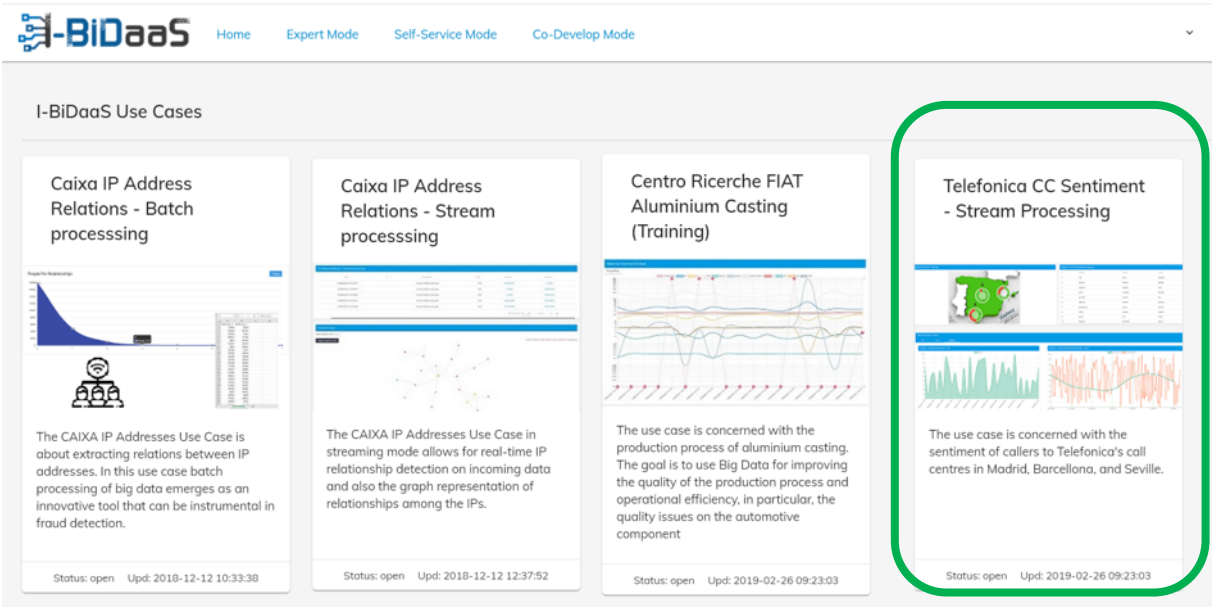

Figure 50. Telefonica I+D use case: Call Centers

The dashboard consists of 3 main visualisations as seen in Figure 51. The map on the left displays the 3 call centers in the form of a pie chart which shows the current state of the sentiment for all the previous calls (since the beginning of the experiment). The overall country color is based on the average of all 3 call centers. On the right side, a table presents the actual

numbers of calls belonging to each call center and having been classified as positive or not. Average waiting time and resolution have been simulated so as to showcase information that could be potentially added in this table. The last chart depicts the number of calls per sentiment per call center as time progresses, thus allowing users to see the historical evolution of the analysis for every call center.

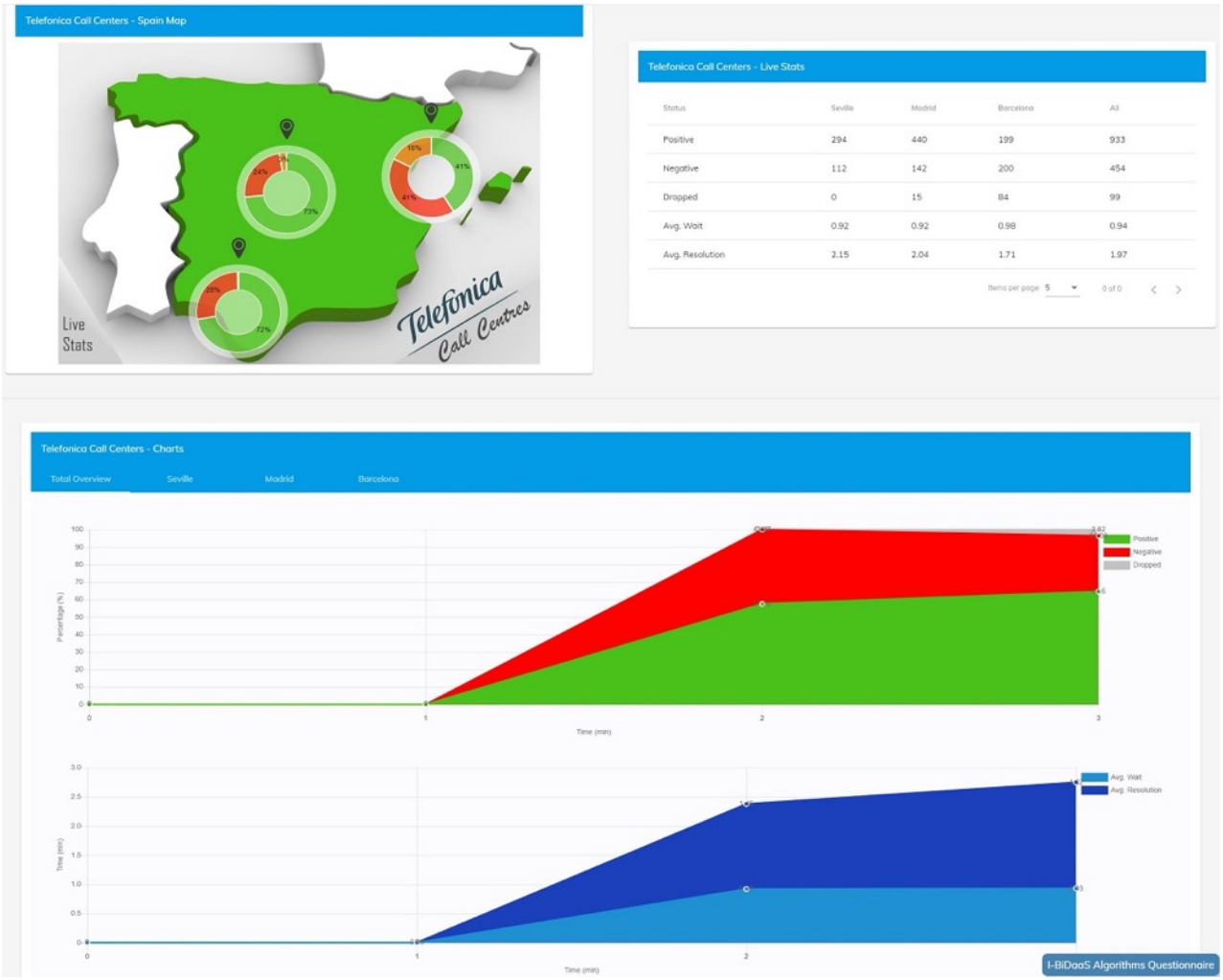

Figure 51. TID: Call center sentiment analysis

4 Detailed Software Modules

This section includes the detailed guidelines for each of the software modules within the final I-BiDaaS solution.

The content is structured by layers:

- Application Layer
  - Data Ingestion and Integration
  - Batch Processing
  - Streaming Analytics
- Distributed Large Scale Layer
  - Hecuba
  - I-BiDaaS Orchestrator
  - Resource Management and Orchestration
- Infrastructure Layer
  - Cloud Infrastructures
  - GPU Cluster

4.1 Application Layer

4.1.1 Data Ingestion and Integration

4.1.1.1 Universal Messaging module – UM (SAG)

Installation Guide

Pre-Installation Requirements

See http://techcommunity.softwareag.com/ecosystem/documentation/universal_messaging/num10-5/index.htm

Pre-Installation Checklist

See http://techcommunity.softwareag.com/ecosystem/documentation/universal_messaging/num10-5/index.htm

Installation and Configuration Procedure

See http://techcommunity.softwareag.com/ecosystem/documentation/universal_messaging/num10-5/index.htm

User/Admin Guides

See http://techcommunity.softwareag.com/ecosystem/documentation/universal_messaging/num10-5/index.htm

Development Guide

See http://techcommunity.softwareag.com/ecosystem/documentation/universal_messaging/num10-5/index.htm

API

See http://techcommunity.softwareag.com/ecosystem/documentation/universal_messaging/num10-5/index.htm

4.1.1.2 Data Fabrication Platform – TDF (IBM)

Installation Guide

Pre-Installation Requirements

Hardware requirements

The following hardware is required: 64 Bit Linux/x86 Server (Red Hat 6+, Ubuntu 14+), 8GB RAM, 4 CPUs, 300MB DASD for Core Component Software and 100GB DASD.

Software requirements

The following software must be installed on your workstation: Apache Tomcat Application Server version 7.0.59 (included in product download) and Oracle Java 7+ or IBM Java 7+. One of the following operating systems is also required: Red Hat Enterprise Linux Server 6 x86, Red Hat Enterprise Linux Server 7 x86, Ubuntu 14 x86 or Windows Enterprise Edition 7.

Pre-Installation Checklist

Performed by the provided installer

Installation and Configuration Procedure

Performed by the provided installer

User/Admin Guides

IBM InfoSphere Optim Test Data Fabrication User Manual is provided with the tool.

For more information see:

https://www.ibm.com/cloud/garage/dte/tutorial/ibm-infosphere-optim-test-data-fabrication

https://www.ibm.com/il-en/marketplace/infosphere-optim-test-data-fabrication

https://www.ibm.com/downloads/cas/A0NE7EGV

4.1.2 Batch Processing

The Batch Processing Module has been extensively described in Deliverables D3.1, D3.2 and D3.3. Its goal is to provide batch data analytics and machine learning capacities to the architecture. It is essentially composed by two components:

- The Advanced Machine Learning algorithms: a set of innovative Machine Learning and data analytics algorithms, that use both Python and COMPSs.
- The COMPSs programming model: able to parallelise and distribute a sequentially programmed algorithm.

This module also uses two extra tools in some use cases: the Qbeast indexing system (facilitating data exploration and experimentation) and the dislib library (providing big data analytics algorithms).

4.1.2.1 Advanced ML Module (UNSPMF)

Installation Guide

Pre-Installation Requirements

I-BiDaaS platform is available online for use on the following link: https://github.com/ibidaas

If, however, it is needed to be ran at the premises for local data analysis it can be used as a Docker container.

HW requirements:

1. Recent Multi-Core CPU (at least 4 cores)
2. 8GB RAM
3. 50GB storage (preferred SSD storage)
4. Fast network (100Mbps at least)

SW requirements:

1. Recent version of Windows, macOS, or Linux (CentOS, Debian, Ubuntu) – Linux is preferred – compatible with Docker Engine
2. Docker Engine

Network requirements:

1. Static IP address

Pre-Installation Checklist

1. Your server has configured OS and Docker Engine is running (Linux: systemctl show --property ActiveState docker)
2. Your server has a static IP address
3. Your server has sufficient storage

Installation and Configuration Procedure

I-BiDaaS platform can be installed and ran with the following command:

```
docker run -d -p 8080:8080 --name ibidaas -v /tmp/data:/data ibidaas:latest
```

The application is then available at the server port 8080.

User/Admin Guides

Currently, the Batch processing module offers the following algorithms:

- Random Forrest (RF),
- Kmeans (KM),
- Decision tree (DT),
- Support vector machines (SVM),
- Alternating direction method of multipliers (ADMM) based algorithms.

The algorithms can be roughly divided into two groups: supervised learning algorithms (RF, DT, SVM, ADMM) and unsupervised learning algorithms (KM). The unspervised learning algorithms are used for clusterization tasks, i.e., the tasks where we do not have predefined classes or knowledge of the structure of the data, but are instead aiming to extract this knowledge from the data itself. For the supervised learning tasks, we can once more roughly split the algorithms in two groups: classification algorithms (RF, DT, SVM) and regression algorithms (SVM). The classification algorithms are used in problems where the data corresponds to a finite set of values and we want to associate each data point with a single value. The regression algorithms are used in problems where the set of values to which the data corresponds to is infinite (e.g., real valued).

Concept and Features

- RF - A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset. After each decision tree gives its prediction, majority voting is performed to improve the predictive accuracy and control over-fitting. RF's offer an internal feature importance ranking, that can point to the most important features in the dataset used for learning. Also, one of the main advantages of this model is its ability to work with either numerical or categorical data.
- KM - The Kmeans algorithm clusters data by trying to separate samples in K groups of equal variance, minimizing the within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields.
- DT - Decision Trees are a non-parametric supervised learning method used for classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Advantages include being able to handle both numerical and categorical data, being simple to understand and interpret, and requiring little data preparation, to name a few.
- SVM - Cascade support vector machines have been introduced as extension of classic support vector machines that allow a fast training on large data sets. The basic support vector machines are a set of supervised learning methods used for classification, among other things. Advantages of the model include being effective in high dimensional spaces and in situations where the dimensionality is greater than the number of samples, as well as using a subset of training points in decision function (called support vectors), which makes them memory efficient.
- ADMM - Alternating Direction Method of Multipliers (ADMM) solver. ADMM is renowned for being well suited to the distributed settings, for its guaranteed convergence and general robustness with respect to the parameters. Additionally, the algorithm has a generic form that can be easily adapted to a wide range of machine learning problems with only minor tweaks in the code.

Inputs

- RF – The user needs to input a data matrix, containing samples as rows and features as columns, as well as the target vector. The algorithm offers a few hyperparameters that the user can set manually, such as:
  1. **n_estimators** - number of trees to fit.
  2. **try_features** - the number of features to consider when looking for the best split.
  3. **max_depth** - the maximum depth of the tree.
  4. **random_state** - the seed used by the random number generator.

- KM – The user needs to input a data matrix, containing samples as rows and features as columns. Since this is an unsupervised learning algorithm, no target values are needed. The algorithm offers a few hyperparameters that the user can set manually, such as:
  1. **n_clusters** - the number of clusters to form.
  2. **max_iter - maximum number of iterations of the k-means algorithm for a single run.**
  3. **n_init - number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.**
  4. **tol – relative tolerance for the early stoppage of the algorithm.**
- DT - The user needs to input a data matrix, containing samples as rows and features as columns, as well as the target vector. The algorithm offers a few hyperparameters that the user can set manually, such as:
  1. **try_features** - the number of features to consider when looking for the best split.
  2. **max_depth** - the maximum depth of the tree.
  3. **random_state** - the seed used by the random number generator.
- SVM - The user needs to input a data matrix, containing samples as rows and features as columns, as well as the target vector. The algorithm offers a few hyperparameters that the user can set manually, such as:
  1. **max_iter** - maximum number of iterations to perform.
  2. **tol – relative tolerance for the early stoppage of the algorithm.**
  3. **kernel - the kernel type to be used in the algorithm. Supported kernels are 'linear' and 'rbf'.**
  4. **C - penalty parameter C of the error term.**
- ADMM - The user needs to input a data matrix, containing samples as rows and features as columns, as well as the target vector. The algorithm offers a few hyperparameters that the user can set manually, such as:
  1. **rho** - the penalty parameter for constraint violation in ADMM.
  2. **abstol** - The absolute tolerance used to calculate the early stoppage criterion for ADMM.
  3. **reltol** - The relative tolerance used to calculate the early stoppage criterion for ADMM.
  4. **warm_start** - cvxpy warm start option.
  5. **objective_fn** - objective function to be solved by the ADMM framework.

Outputs

- RF – The model outputs the predicted class label for each data sample.
- KM - The model outputs the predicted cluster label with which the data sample is associated with.
- DT - The model outputs the predicted class label for each data sample.
- SVM - The model outputs the predicted class label for each data sample.
- ADMM - The model outputs the predicted regression value for each data sample.

Development Guidelines

Architecture

The Batch processing module relies on the COMPSs framework (with Python) when developing the algorithms. This means that the parallelization of the algorithms can be easily achieved by adding some decorators i.e. annotations to the code. As the work is being distributed to a set of workers, some parts of the algorithm should be defined as tasks (using the @task decorator). When a method or function is defined as a task, it will be executed on each worker separately. This demands that the data are split among the workers. The most important concern for the developer is to identify parts of the code, that could be executed on the distributed data, in parallel by the workers. After defining the tasks, it is important to utilize the compss_wait_on function, in order to synchronize the results gained from the workers. The input data should be provided either already separately for each worker, or as one whole that can be divided to the workers programmatically, before executing the tasks.

Prepare Development Environment

In order to develop algorithms for the Batch processing module, it is necessary to have installed Python (as well as the needed Python libraries) and the COMPSs framework.

4.1.2.2 COMPSs Programming Model (BSC)

Documentation Guidelines

For the COMPSs programming model, extensive documentation exists, which covers prerequisites and requirements, installation, user and admin guides. The link is:

https://www.bsc.es/research-and-development/software-and-apps/software-list/comp-superscalar/documentation

The main resource is the COMPSs Documentation (https://compss-doc.readthedocs.io/en/stable/), which includes these detailed sections:

- What is COMPSs?
- Quickstart
- Installation and Administration
- Application development
- Execution Environments
- Tracing
- Persistent Storage
- Sample Applications
- PyCOMPSs Player
- PyCOMPSs Notebooks
- Troubleshooting

The rest of documents are very specific, and detail how to work with Distributed Data Sets (DDS) and to exploit Redis as persistent storage. Older versions of the documentation are also kept as reference.

4.1.3 Streaming analytics

4.1.3.1 Apama Streaming Analytics Sub-module (SAG)

Installation Guide

Pre-Installation Requirements

See http://www.apamacommunity.com/docs/

Pre-Installation Checklist

See http://www.apamacommunity.com/docs/

Installation and Configuration Procedure

See http://www.apamacommunity.com/docs/, it differs if you install it via the SAG installer or from a Docker image, both is described in detail.

User/Admin Guides

See http://www.apamacommunity.com/docs/

Development Guide

See http://www.apamacommunity.com/docs/

API

See http://www.apamacommunity.com/docs/

Comparison of Community and Commercial Edition

The main differences are that: the Community Edition is supported by the community, rather than Software AG; The number of threads that the correlator runs on is reduced to four; the number of query definitions is limited to five; and that the Correlator is restricted to using 1GB of memory.

The table below shows the complete list of differences between the two products:

Restrictions	Community Edition	Full
Support	Community	Software AG
Correlator process memory usage	1024MB	Unlimited
Machine CPU threads	4	Unlimited
Log contains "Community Edition" entries	Notes in log ×	Not applicable ✓
Number of correlators used in project	10	Unlimited
Commercial OEM/indirect use cases permitted	No ×	Yes ✓
Attribution in application Help/About	Attribution required	Not applicable
Number of Apama queries definitions	5	Unlimited
JMS messaging	Best effort	Best effort, reliable
Apama queries uses distributed stores for HA	No ×	Terracotta ✓
Number of EPL contexts	20	Unlimited
Number of instances per Apama query	5	Unlimited
Number of unique partitions per Apama query	50	Unlimited
Number of EPL persistent monitor types	5	Unlimited
Correlator Deployment Packages (CDPs)	No ×	Yes ✓

4.1.3.2 Streaming Analytics and Pattern Matching Module (FORTH)

Installation Guide

Pre-Installation Requirements

Software requirements: Java version 7 or later.

Pre-Installation Checklist

Download and install Terraccotta-db (http://www.terracotta.org/downloads/)

Installation and Configuration Procedure

1) Run TerracottaDB by running ./start-server.(sh|bat)

2) Run the Producer by running ./start-producer.(sh|bat)

3) Run the Consumer by running ./start-consumer.(sh|bat)

4) (Optional) Tun ./start-reader.(sh|bat) to see the filtered entries in the new database.

4.1.4 Advanced Visualization

4.1.4.1 Advanced Visualisation Toolkit sub-module – AVT (AEGIS)

Installation Guide

Pre-Installation Requirements

- **HW Requirements**
  >4GB RAM, >2-Cores, >30GB storage
- **SW Requirements**
  Apache 2 Web server, Node.js
- **Network Requirements**

Open ports to components serving streaming data (e.g. SAG's Universal Messaging, etc.)

Pre-Installation Checklist

See 2 options in SW Requirements

Installation and Configuration Procedure

AVT is developed using the Angular web framework. Therefore, installation only involves the deployment of the application on a web server (Apache2 preferably – see SW requirements).

Before running the application, a set of endpoint urls must be provided so that the tool can retrieve data. This takes place in a configuration file that can be found under the path "/src/assets/config.json". The format of the file follows a simple key:value approach where the base urls of the relevant endpoints are given. The following figure shows an example of the urls used in the context of I-BiDaaS project.

```
{
    "apiBaseUrl": "http://ibidaasurl1.eu",
    "messagingUrl": http://ibidaasurl2.eu,
    "usecaseUrl": http://ibidaasurl3.eu,
    "histogramUrl": http://ibidaasurl4.eu,
    "endpointUrl": http://ibidaasurl5.eu,
    "kpisUrl": http://ibidaasurl6.eu

}
```

Figure 52. AVT configuration file

User/Admin Guides

Concepts/Features

AVT supports a set of visualization widgets that can be customized to suit the needs of the data to be visualized. Data can be available in batch or streaming modes.

The customization of AVT to cover the needs of I-BiDaaS introduced the concepts of:

- Project, that contains the type of data to be processed, the data processing mode and the algorithm to execute
- Experiment, that serves as a specific instantiation of the project with given resources and algorithm parameters.

Building on these basic concepts, the AVT empowers the I-BiDaaS Dashboard with three modes of operation which give users a variable flexibility of configuring the elements of their projects and experiments:

1. Expert mode, where all project and experiment details are configurable by users.
2. Self-service mode, where users only provide data and parameter values to given algorithms.
3. Co-Developed mode, where users have limited or no configuration options at all. Experiments in this mode use tailor-made visualisations for specific use cases and therefore follow a specific pipeline and configuration.

Inputs

- Data in streaming mode via Universal Messaging and in batch mode via I-BiDaaS' storage.
- Project and Experiment details
- User actions on running visualisations that allow interaction with the visualized results of the underlying analysis.

Outputs

- Data in streaming mode via Universal Messaging and in batch mode via I-BiDaaS' storage.
- Project and Experiment details
- User actions on running visualisations that allow interaction with the visualized results of the underlying analysis.

Development Guides

Architecture

The application uses angular modules to implement the various visualization elements required by the various use cases. Moreover, a number of Javascript Libraries were integrated to cover aspects like e.g. the communication with SAG's Universal Messaging via its Universal Javascript API (nirvana.js library).

Addressing the required management process of the 'project' and 'experiment' concepts as described in the previous paragraphs, the AVT is based in the context of I-BiDaaS on the orchestration API that provides all the necessary endpoints for CRUD operations on projects and experiment.

Finally, adaptors that allow data retrieval directly from repositories used in I-BiDaaS like e.g. the Cassandra DB, have been developed and successfully integrated in the AVT's architecture. Such adapters serve as a middleware between data sources and the web application and in AVT's architecture they reside outside the Angular app as complementary components which offer internal endpoints that handle the relevant communications.

The following figure depicts the high-level architecture of AVT context of I-BiDaaS.

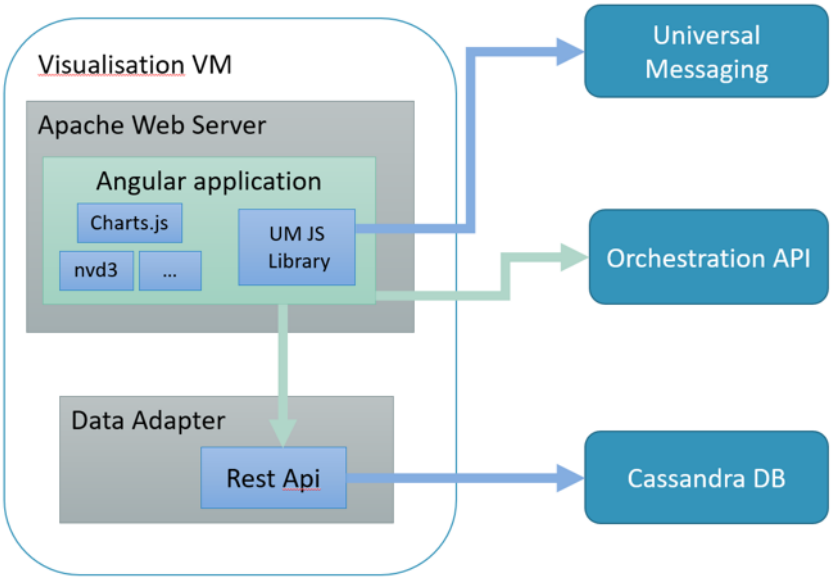

Figure 53. AVT Architecture

4.2 Distributed Large Scale Layer

4.2.1 Hecuba (BSC)

Hecuba is in charge of the data storage, acting as an easy and non-changing interface to several database systems (e.g. Cassandra). In addition, Hecuba is able to interact with the COMPSs programming model runtime to achieve advanced features such as exploiting data locality.

Installation Guide

Hecuba's installation procedures (both automatic and manual) are described at the readme file of GitHub (https://github.com/bsc-dd/hecuba)

User/Admin Guides

The User and Admin guides can be located at the GitHub Wiki (https://github.com/bsc-dd/hecuba/wiki), Sections 1 and 3.

- https://github.com/bsc-dd/hecuba/wiki/1:-User-Manual
- https://github.com/bsc-dd/hecuba/wiki/Cache,-prefetch-and-distributed-arrays

Development Guide

A Developer Guide in the form of a Hands-on tutorial can be found at the GitHub Wiki, Section 2 (https://github.com/bsc-dd/hecuba/wiki/2:-Hecuba-Hands-On).

4.2.2 I-BiDaaS Orchestrator (ITML)

Installation Guide

Installing the software is part of the I-BiDaaS installer, which contains the opensource components of the I-BiDaaS solution. The installer creates a docker swarm with one node acting as manager, also running the orchestrator and the I-BiDaaS AVT, and one or more workers used for running the PyCompss jobs.

To run the installer, there must be at least two (three recommended) available centos hosts (or virtual machines), with SSH with public key authentication enabled.

Pre-Installation Requirements

One host running ansible V2.7 or higher, access (via SSH with public key authentication) to two or more centos nodes. A two-node installation will result in installing the software shown in *Figure 54*.

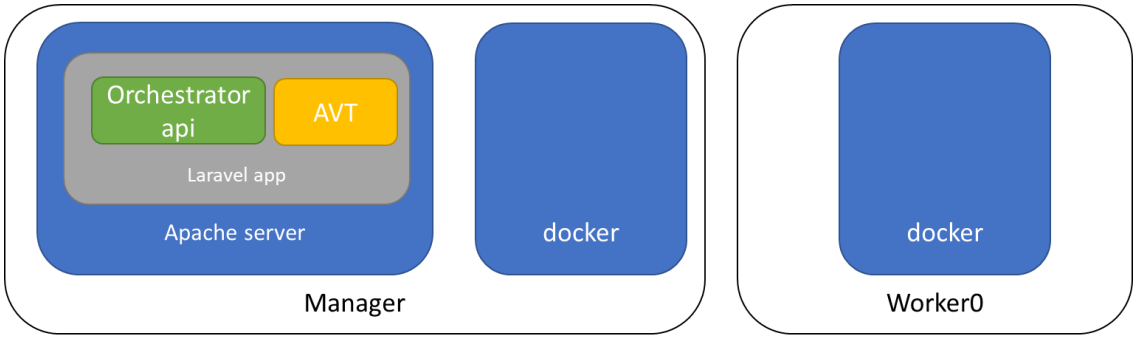

Figure 54. Installed software schema

Installation and Configuration Procedure

Auto deployment of Docker Swarm (multimanager) cluster (including docker installation)

- Edge version of docker will be used for cluster deployment
- Docker cluster will be deployed with N-managers, N-workers
- Hosts are used in swarm-cluster already will be skipped

Swarm hosts requirements:

- CentOS \ RedHat
- Internet connection (to get packages from repos)

Ansible host requirements:

- Ansible v2.4 (or newer)
- Jinja 2.9 (or newer)
- Ssh-keys should be copied to all hosts of your inventory (ssh-copy-id can be used)

Cloning repo from gitlab

git clone https://github.com/ibidaas/installer

Edit hosts file

In this file, we define the hosts to be used. An example configuration with one manager and two workers is the following:

```
manager ansible_ssh_host=192.168.15.100
worker1 ansible_ssh_host=192.168.15.101
worker2 ansible_ssh_host=192.168.15.102


[docker-swarm-manager]
manager

[docker-swarm-node]
worker1
worker2

[docker-swarm:children]
docker-swarm-manager
docker-swarm-node
```

Running ANSIBLE-PLAYBOOK

```
ansible-playbook -h hosts playbook swarm-cluster.yml
ansible-playbook -h hosts playbook site.yml
ansible-playbook -h hosts playbook manager.yml
```

Development Guide

The Orchestrator in the I-BiDaaS platform was created to fill the gap in the original architecture for orchestrating the I-BiDaaS elements; it operates as a middleware between the UI and the rest of the infrastructure.

The orchestrator is a REST API with DB support that provides:

- **Scalability**: nodes/core number, RAM selection per experiment
- **Security/isolation**: Every experiment runs in docker containers that are disposed after experiment completion
- **Self-service-ness**:
  - Users do not need to maintain a COMPSs environment in the machines
  - Users can experiment on ready-to-use docker images with templates/examples
- **CRM Features**
  - User provisioning
  - Authentication, Authorization, Accounting
  - History (Custom project management, Experiment history)

The REST API is implemented in PHP Laravel[6], supported by Mysql RDBMS[7]. Its functionality in terms of the available endpoints is provided in Appendix 1.

The Orchestrator provides integration between the UI and the docker swarm executing the requested experiments. The sequence diagram for running a new Experiment is depicted in Figure 55.

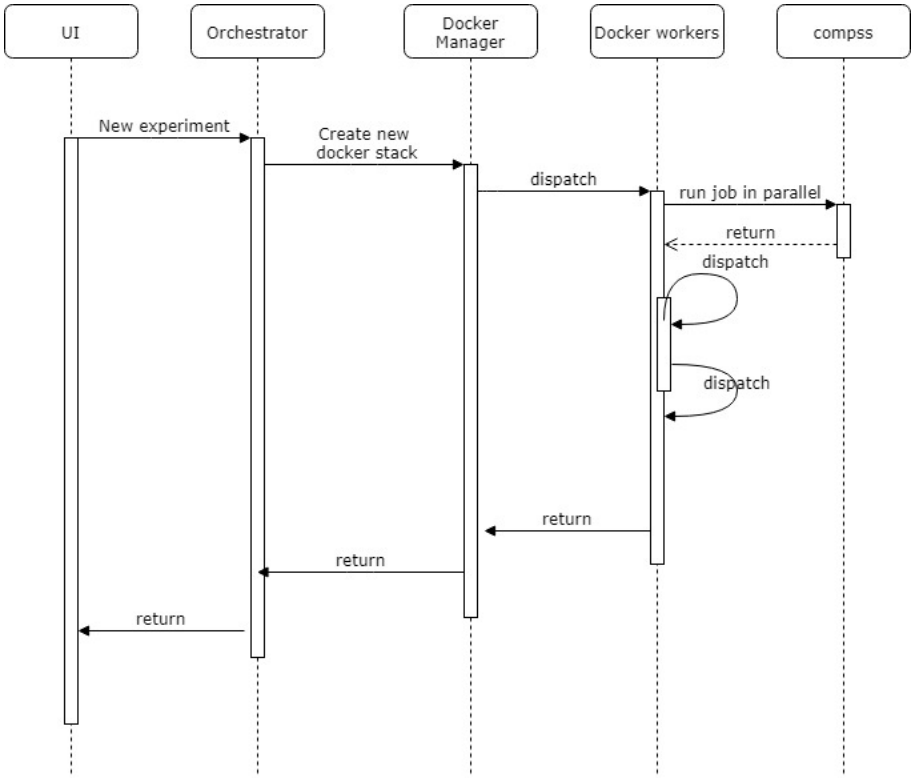

Figure 55. The Generic use case process sequence diagram

The relational model of the Mysql database supporting the REST API is depicted in Figure 56. The projects that each user can initiate have a specific code (an implementation of an I-BiDaaS use case) or a generic algorithm implementation, *e.g.*, K-means. In Custom projects, users can upload and run their own code (expert mode).

[6] https://laravel.com/

[7] https://www.mysql.com/

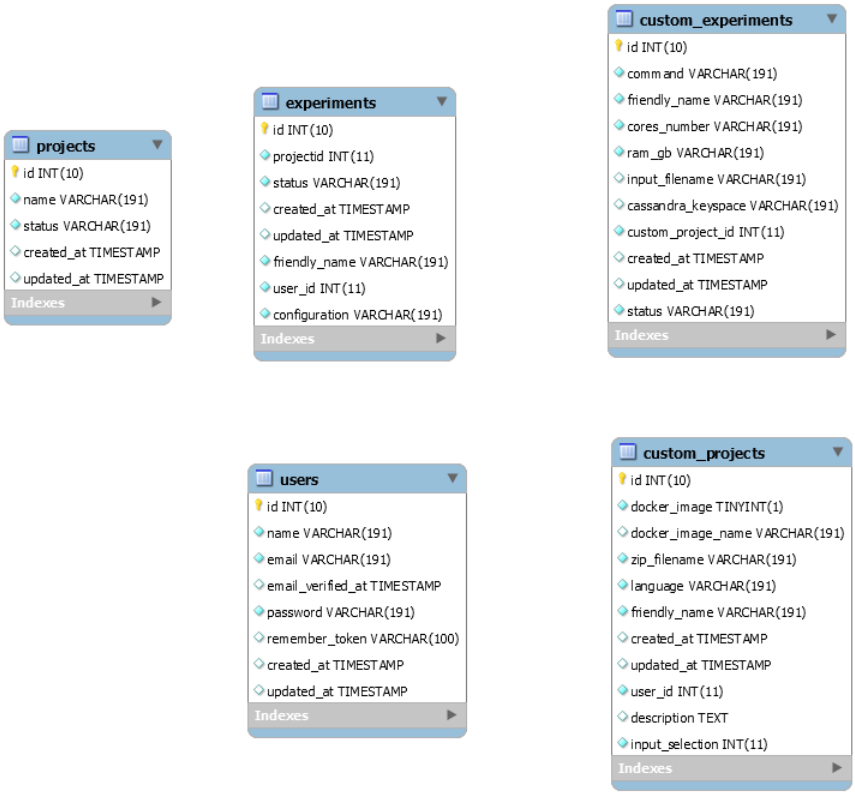

Figure 56. The I-BiDaaS generic use case database structure

The cluster that runs the batch processing jobs is based on docker swarm[8]. The swarm consists of a manager node and a set of worker nodes. The orchestrator assigns dynamically a set of workers from the set of available nodes in docker swarm, based on the user's preferences and set-up inserted through the UI. These workers exploit the Cassandra DB and the shared FS in order to complete the requested job.

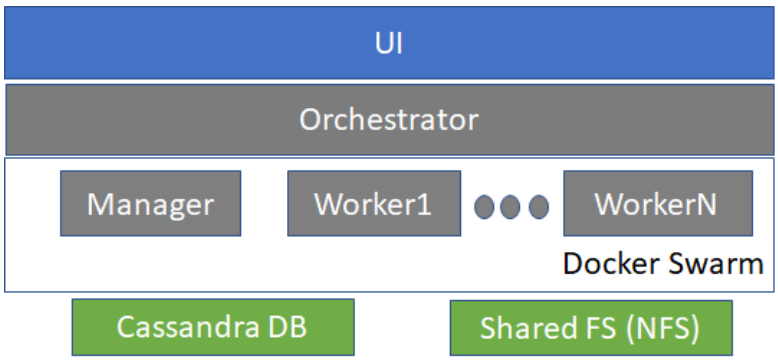

Figure 57. Batch processing architecture: The orchestrator and the docker swarm

For further information see Appendix 2 – Orchestrator REST API.

[8] https://docs.docker.com/engine/swarm/

4.2.3 Resource Management and Orchestration – RMO (ATOS)

The brain of the RMO modules is the resource orchestrator, which is based on Cloudify. This technology allows us to operate at large-scale while describe our deployments using a common specification across different types of providers. In this way, the solution aims to avoid vendor lock-in situation through the abstraction of different providers that could be integrated via plugins to our environment.

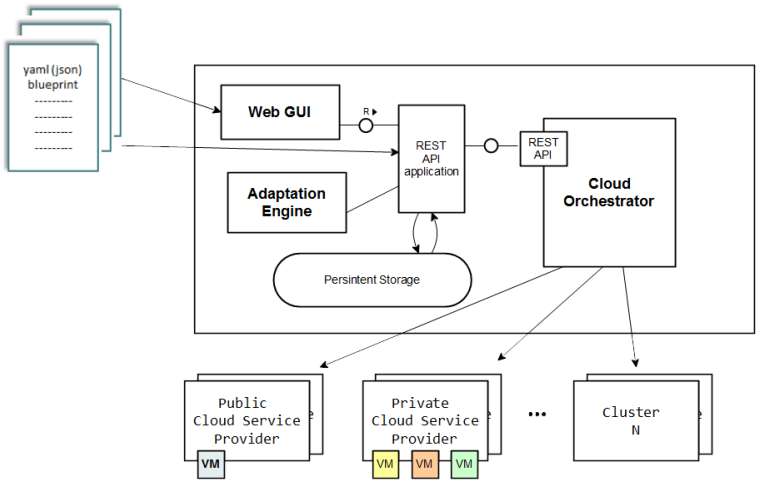

Figure 58. RMO module architecture

Installation Guide

Pre-Installation Requirements

The RMO module is composed by several submodules each of them with their own requirements.

- For the Cloud Orchestrator is recommended:
  - CO is supported for installation on a 64-bit host with RHEL/CentOS 7.4.
  - 8 VCPUs, 16GB RAM, 64GB Storage
  - 2 network interfaces
- The Adaptation engine is composed by a configuration management system which includes configuration recipes, template images and deployment descriptor templates.
- Any Private or Public Cloud Service Provider account:
  - For Private Cloud we used during the project lifetime:
    - DevStack in single-node - https://docs.openstack.org/devstack/latest/
    - Openstack Kolla in multi-node - https://docs.openstack.org/kolla/latest/
  - For Plublic Cloud: AWS, Azure, Google Cloud, … several providers are supported via plugin.
    - AWS has been used for testing purposes.

Installation and Configuration Procedure

Cloudify Manager images are present on the Cloudify website. It provides images for AWS, OpenStack and Docker.

wget http://repository.cloudifysource.org/cloudify/19.01.24/community-release/cloudify-docker-manager-community-19.01.24.tar

sudo docker run --name cfy_manager -d --restart unless-stopped -v /sys/fs/cgroup:/sys/fs/cgroup:ro --tmpfs /run --tmpfs /run/lock --security-opt seccomp:unconfined --cap-add SYS_ADMIN --network host docker-cfy-manager:latest

cfy profiles use <manager-ip> -u admin -p admin -t default_tenant

Start Docker Container

Docker container name is cfy_manager.

sudo docker container start cfy_manager

Stop Docker Container

sudo docker container stop cfy_manager

Navigate to the docker container CLI

sudo docker exec -it cfy_manager bash

User/Admin Guides

- Adding new Cloud Service Provider

This action can only be performed by an I-BiDaaS platform operator. When a new Cloud Provider type needs to be on-boarded, a new plugin needs to be configured in the Resource Orchestrator module. The system includes multiple compiled modules that provide an abstraction for using 3rd party APIs by providing TOSCA types and matching implementation code that can be used via blueprints.

#OpenStack v3.1.0

cfy plugins upload -y openstack_plugin/plugin.yaml http://repository.cloudifysource.org/cloudify/wagons/cloudify-openstack-plugin/3.1.0/cloudify_openstack_plugin-3.1.0-py27-none-linux_x86_64centos-Core.wgn

As soon as a new provider type has been uploaded (see sample command above), the concrete stack to be used can be configured. The parameters needed to on-board a new Cloud Service Provider are:

• The endpoint where the CSP is accessible.

• The credentials needed to access the provider

Development Guide

In order to define the logical representation of a cloud-based application, it is needed to specify various elements like the computing nodes to be used, how they relate to one another, or how they need to be deployed and maintained across the lifecycle of the application. The OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) specification offers a well-known standard approach for enhancing interoperability across cloud providers. The deployment descriptors or blueprints are written in YAML and are packaged together with the configuration files and the installation scripts required for running the application. Once the blueprints have been defined, the Cloudify Dashboard allows us to visualize their topology in a graphical way.

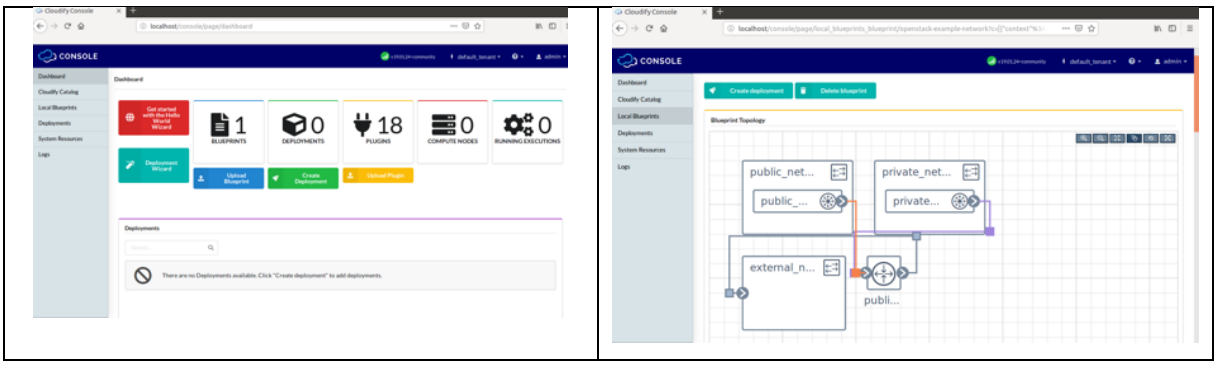

Figure 59. Cloud Orchestrator dashboard.

```
cfy blueprints upload <blueprintFileName.yaml> -b <blueprint-id>
```

For further information see https://docs.cloudify.co/4.2.0/blueprints/overview/

4.3 Infrastructure Layer

4.3.1 Cloud Infrastructure (ATOS)

ATOS' Infrastructure Services offered in I-BiDaaS are delivered on top of the hardware resources from Instituto Tecnológico de Energías Renovables (ITER) Data Centre located in the Canary Islands. Among the different high-performance computing facilities and data infrastructures providers considered during the elicitation of requirements phase, the final candidate selected for I-BiDaaS is the "Teide Supercomputer", which is a general-purpose High-Performance Computing infrastructure, housed in a D-Alix datacentre which provide high available electrical and cooling infrastructure, and high-speed internet connectivity.

The Teide-HPC supercomputer is composed of 1100 Fujitsu computer servers, the core of the compute nodes is Fujitsu PRIMERGY CX250 S1 servers housed in a PRIMERGY CX400 chassis grouped in 4 nodes per chassis and featured with the latest Intel Sandy Bridge processors.

For further details, check "D5.1 /D5.3 /D5.5 Federated Resource Management for Data Analytics".

4.3.2 GPGPUs Commodity Cluster (FORTH)

FORTH's commodity cluster contains several modern off-the-shelf commodity GPGPUs, such as NVIDIA GeForce GTX 1080 Ti and NVIDIA TITAN Xp. Such GPGPUs offer extremely high processing throughput for parallelizable workloads with a low power cost.

In addition to this kind of GPUs, also known as discrete or dedicated, there is another type of such processing unit, called integrated GPU. An integrated GPU is packed inside the CPU's die, such as in the case of the Intel HD Graphics. The main characteristic that differentiates these two kinds of GPUs is the memory space, where discrete GPUs have their own dedicated memory space, while integrated GPUs share the same memory space with the CPU. Apparently, integrated GPUs are more power-efficient than discrete GPUs. FORTH's commodity cluster also contains another type of hardware accelerator, which is the Intel Phi coprocessor, as well

as powerful Intel processors (based on the Skylake micro-architecture or newer) enabled with the Intel Software Guard Extensions (Intel SGX), which is a set of CPU instruction codes that allows user-level code to allocate private regions of memory, called enclaves, that are protected from processes running at higher privilege levels.

The GPU-Accelerated Pattern Matching engine has been described in Deliverables D4.1, D4.2 and D4.3. Its goal is to provide GPU-accelerated string searching capabilities, tailored for filtering or counting streaming data. It has been integrated with the TerraccottaDB, hence it can be used transparently seamlessly through it.

The module uses as input a file with the patterns that need to be searched, as well as the input database that will read data from, and the output database that will write data to.

For further details, check “D5.1 /D5.3 /D5.5 Federated Resource Management for Data Analytics”.

5 Lessons Learnt

After a significant period of research, design and experimentation towards the development of the I-BiDaaS solution, the project's data providers, CAIXA, TID and CRF, shared their experience regarding the challenges, difficulties and obstacles they faced on data availability and data sharing.

Their stories have been transformed into lessons that provide valuable knowledge to our stakeholders and also necessary insight on how to further improve the I-BiDaaS solution. A dedicated blog article[9] has been created having as a main goal to spread their experience to the community.

5.1 CaixaBank: I-BiDaaS Application to the Financial Sector

In the case of CaixaBank, as with many entities in critical sectors, there was an initial reluctance to use any big data storage or tool outside its premises. Therefore, **the primary goal of CaixaBank when starting its involvement in I-BiDaaS was to find an efficient way to perform big data analytics outside its premises**. Achieving this would speed up the process of granting new external providers access to CaixaBank data (typically a bureaucratic process that takes weeks). Additionally, CaixaBank wanted to become **much more flexible in adopting proof-of-concept (PoC) technological solutions (i.e., to test the performance of new data analytics technologies to be integrated into CaixaBank infrastructure).** Usually, for any new technology testing, even simple ones, if hardware is needed, then it should be done through the infrastructure management subsidiary who will be in charge of deploying it. Due to the level of complexity, the size of CaixaBank's infrastructure, and the processes rigidity, deployment can also take months.

CaixaBank needed to find ways to by-pass these processes without compromising security or privacy. GDPR really limits the usage of customer data, even if used for fraud detection and prevention, or for enhancing the security of customer accounts. It can be used internally to apply certain security policies but sharing this data with other stakeholders remains an issue. Furthermore, the banking sector is strictly regulated, and National and European regulators are supervising all security measures taken by banks to provide a good level of security while maintaining the privacy of customers. The current trend of externalizing many services to the cloud also implies the establishment of strict control of the location of data, as well as who has access to it.

To be more efficient in terms of amount of money and time spent, CaixaBank proposed three different use cases to be tested within I-BiDaaS, exploring ways to conduct data analytics outside of its premises, assuring the maximum level of security and privacy possible and evaluating I-BiDaaS overall solution and specific tools, such as IBM's TDF (Test Data Fabrication)[10] for providing synthetic, non-sensitive data.

Results obtained from the first use case validated the usage of rule-based synthetically generated data and indicated that it could be very useful in accelerating the onboarding process of new data analytics providers (consultancy companies and tools). **CaixaBank validated that it could be used as high-quality testing data outside CaixaBank premises for testing new technologies and PoC developments, streamlining the grant accesses of new external providers to these developments, and thus reducing the time of accessing data from an**

[9] http://www.ibidaas.eu/blog/Data-sharing-availability/

[10] https://www.ibm.com/cloud/garage/dte/tutorial/ibm-infosphere-optim-test-data-fabrication

average of 6 days to 1.5 days. This analysis was beneficial for CaixaBank purposes but **was also concluded that the analysis of rule-based fabricated data did not enable the extraction of new insights from the generated dataset**, simply the models and rules used to generate the data.

The other two use cases focused on how extremely sensitive data can be tokenized to extract real data for use outside CaixaBank premises. By tokenizing, we mean encrypting the data and keeping the encryption keys in a secure data store that will always reside in CaixaBank facilities. This approach implies that the data analysis will always be done with the encrypted data, and it can still limit the results of the analysis. One of the challenges of this approach is to **find ways to encrypt the data in a way that it loses as little relevant information as possible**. Use case 2 and use case 3 experimentation was performed with tokenized datasets built by means of three different data encryption algorithms: (1) Format preserving encryption for categorical fields; (2) Order preserving encryption for numerical fields; (3) A Bloom-filtering encryption process for free text fields. This enabled CaixaBank to extract the dataset, **upload it to I-BiDaaS self-service big data analytics platform and analyse it with the help of external entities without being limited to the corporate tools available inside CaixaBank facilities**. We proceeded with unsupervised anomaly detection in those use cases, identifying a set of pattern anomalies that were further checked by CaixaBank's Security Operation Center (SOC). This helped increase the level of financial security of CaixaBank. However, beyond that, we consider this experimentation very beneficial, and provide us the guidelines to be replicated when analysing other commercial big data analytics tools, previously to their acquisition. In summary, the next table highlights some of the benefits of CaixaBank due to its participation in I-BiDaaS:

Benefits	KPIs
To increase the efficiency and competitiveness in the management of its vast and complex amounts of data.	75%-time reduction data access from external stakeholders using synthetic data (From 6 to 1.5 days).
To break data silos not only internally, but also fostering and triggering internal procedures to open data to external stakeholders.	Real data accessed by at least 6 different external entities skipping long-time data access procedures.
To evaluate Big Data analytics tools with real-life use cases of CaixaBank in a much more agile way.	I-BiDaaS overall solution and tools experimentation with 3 different industrial use cases with real data.

On the one hand, usage of rule-based synthetically generated data and indicated that it can be very useful in accelerating the onboarding process of new data analytics providers (consultancy companies and tools). CaixaBank validated that it could be used as high-quality testing data outside CaixaBank premises for testing new technologies and PoC developments, streamlining the grant accesses of new external providers to these developments, and thus reducing the time of accessing data from an average of 6 days to 1.5 days. This analysis was beneficial for CaixaBank purposes but was also concluded that the analysis of rule-based fabricated data did not enable the extraction of new insights from the generated dataset, simply the models and rules used to generate the data.

On the other hand, CaixaBank explored how extremely sensitive data can be tokenised to extract real data for its usage outside its premises. By tokenising, we mean encrypting the data and keeping the encryption keys in a secure data store that will always reside in CaixaBank facilities. This approach implied that the data analysis will always be done with the encrypted

data, and it can still limit the results of the analysis. One of the challenges of this approach is to find ways to encrypt the data in a way that it loses as little relevant information as possible. Experimentation was performed with tokenised datasets built by means of three different data encryption algorithms: (1) Format preserving encryption for categorical fields; (2) Order preserving encryption for numerical fields; (3) A Bloom-filtering encryption process for free text fields. This enabled CaixaBank to extract the dataset, upload it to I-BiDaaS self-service Big Data analytics platform and analyse it with the help of external entities without being limited to the corporate tools available inside CaixaBank facilities. I-BiDaaS beneficiaries proceeded with an unsupervised anomaly detection in those use cases, identifying a set of pattern anomalies that were further checked by CaixaBank's Security Operation Center (SOC), helping to increase the level of financial security of CaixaBank.

In summary, we were able to speed up the implementation of Big Data analytics applications, test algorithms outside CaixaBank premises and test new tools and algorithms without data privacy concerns by exploring and validating the usage of synthetic data and tokenised data in three different use cases, improving the efficiency in time and cost by means of skipping some data access procedures and being able to use new tools and algorithms in a much more agile way. User requirements regarding the availability of 'Intermediate and Non-IT users' to analyse and process the data of the use cases were also validated through several internal and external workshops[1] in which the attendants from several departments of CaixaBank and other external entities (data scientists, business consultants, IT and Big Data managers) provided very positive feedback about the platform usability. It is important to highlight that those results should be applicable to any other financial entity that faces the same challenges and tries to overcome the limitations of data privacy regulation, the common lack of agility of large-scale on-premise Big Data infrastructures and very rigid but necessary security assessment procedures

5.2 TELEFONICA I+D: I-BiDaaS Application to the Telecommunication Sector

There are **several challenges** related to Telefonica's setup. One of the most critical ones is the **use of big data outside the company's premises**. The I-BiDaaS initiative served as a **pivoting point** for re-visiting existing practices of performing **big data analytics within a wider ecosystem**.

More specifically, when considering the telecommunications sector, the utility of data generated across the network depends on how the data are aggregated across time, users, locations involved, etc. Depending on where and how the aggregation and modeling of such data happens, it can be extremely powerful for marketing and other purposes (e.g., infrastructure planning) but also raise great privacy concerns over the anonymity and privacy of the users producing them. For example, on the one hand, browsing data of particular users can be very valuable for targeted advertisements but also be considered as very invasive and creepy by some users. On the other hand, aggregated mobility data of users in a city extracted from antenna logs can be useful in deploying new infrastructure, services, bootstrapping other telco companies, or even planning physical retail stores without violating users' privacy due to aggregation.

Second, leakage of private data happening over the network can impact or diminish the utility of such data, and threaten the users involved, as well as their trust in the ISP and its services. In addition, sharing across companies of insights or data based on user information and activity is

[1] https://www.ibidaas.eu/blog/%e2%80%9cI-BiDaaS-Application-to-the-Financial-Sector%e2%80%9d-Workshop/

regulated and can be difficult to achieve, again diminishing the utility of such data and insights. Also, such sharing is not easy to audit and monitor, especially when this needs to be done at the user-level, or at least with the user's consent. This is especially true if these data are user-generated (e.g., web browsing logs), and must be stored, processed and shared in a GDPR-compliant manner.

In fact, insights based on machine learning models are typically very useful for telcos and other organizations (for the reasons mentioned above), but again **very difficult to audit and even engage the users to participate** and control the model creation. In particular, it is difficult to build a machine learning model across many users' data while respecting their privacy, possible opt-ins in particular analysis to be performed on their data, etc.

These challenges become even more difficult to address when data are combined from different sources within a company, or taken to the extreme case, combined across companies. Therefore, in order for such diverse valued data to be utilized in a **secure, privacy-preserving, industry-supported, future EU data market, safeguarding policies and procedures are put in place**, and innovative technologies are needed in the near future. In more detail, privacy-preserving machine learning methods applied to the data beforehand could enable higher or even exact compliance with GDPR-type of regulations, and provide better privacy guarantees to the end-user. Also, user models built with a privacy-preserving fashion could be traded or shared with 3rd party companies (e.g., other telcos, advertisers, etc.) **via the 4th platform strategic effort of TID**.

In many cases, an anonymised dataset can still present residual risk to data subjects. Indeed, even when it is no longer possible to precisely retrieve the record of an individual, it may remain possible to glean information about that individual with the help of other sources of information that are available (publicly or not).

It has to be highlighted that beyond the direct impact on data subjects produced by the consequences of a poor anonymisation process (annoyance, time consumption and feeling of lost control by being included in a cluster without awareness or prior consent), other indirect consequences of poor anonymisation may occur whenever a data subject is included in a target erroneously by some attacker, as a consequence of processing anonymised data - especially if the attacker’s intents are malicious. Therefore, anonymisation techniques can provide privacy guarantees, but only if their application is engineered appropriately - which means that the prerequisites (context) and the objective(s) of the anonymisation process must be clearly set out in order to achieve the targeted anonymisation level.

Users interact with various telco services and trigger the production of large and diverse types of data at different levels in the network infrastructure of the ISP. Some data are produced at the user level (i.e., end-user devices such as smartphones, laptops, etc.), and include data regarding activities of the user such as web browsing, TV viewing, radio streaming, video calling, etc. Other data are produced at the networking infrastructure level (e.g., routers, antennas, etc.), as a reaction to the user activity in the network such as phone calls, activity with SMS and other message services, roaming, traveling in the city (i.e., changing antennas), web browsing, and other user-level activities.

Both types of data are useful for modeling user behavior, and can be utilized by the telco owning them, or by other telcos and 3rd party companies. However, with **strict EU regulations on user personal data such as the GDPR and ePrivacy**, safeguarding methods were needed to put in place in order to prevent privacy-breaching activities. To this end, TID has made **concrete steps to utilize its data by providing new services on top of them, or supporting other 3rd party businesses in making efficient and effective, data-driven decisions**.

Towards this goal, **TID created a novel strategy called the 4th platform**, and a first of its kind vertical on top of it, called Aura, which aims to revolutionize the way telcos and other big data players can participate in the data markets in a user privacy-respecting fashion. Moreover, in the context of I-BiDaaS, TID has provided the necessary support so that IBM can deliver synthesized data **to support the relevant use cases at early-stage**. In addition, TID has **achieved the data silo breach and has successfully shared resources with authorized third parties**.

5.3 CRF: I-BiDaaS Application to the Manufacturing Sector

Automotive production processes are complex in that production lines have several robots and digital tools. At the shop floor level, massive amounts of raw data are gathered; data that not only help to monitor processes but can also improve process robustness and efficiency.

Within the **I-BiDaaS project**, the data provider CRF identified two scenarios in which complex and initial structured/unstructured data sets are retrieved from real processes, and defined two use cases:

1. Production process of Aluminium die-Casting
2. Maintenance and Monitoring of Production assets

The project focuses on providing **a self-service solution** that will give CRF employees the insights and tools they need **to develop a methodology to be implemented in the production sites for improving the quality of the processes and products in a much more agile way** through the collaborative effort of self-organizing and cross-functional teams.

1. Production process of Aluminium die Casting

The Aluminium die-casting process is complex, and it is important to not only carefully design parameters but also to control them because they have a direct impact on the quality of the casting. The goal of the use case is **to predict whether an engine block will be manufactured correctly**, i.e., without anomalies. The correct prediction during the die-casting process would **avoid subsequent further processing and scraps**, **which would lead to financial savings for the manufacturer.**

Initially, we received large unstructured volumes of heterogeneous data from different sources and at different levels. In the preliminary stage, we analysed all the information and interacted with the plant in order to understand the obscure parts. Furthermore, needing to guarantee the anonymization of the data, a process that took more time than expected, synthetic data were fabricated in order not to delay the technical development of the individual components and the I-BiDaaS solution. Later, when real anonymised data became available, we shared them to test the complexity of the process with the analytics developed within the project.

Unstructured, noisy and incomplete data has been collected, aggregated, pre-processed and converted into structured messages of a common, unified format for the analysis. In summary, the first lesson we learned is that, in parallel with the process of data anonymization, creating structured data, etc., it is useful to generate synthetic data in parallel. This is useful for the early stages of development but requires caution when extracting insights from synthetic data. **The high-level algorithms**, developed during the first part of the project, **identified the critical values from the dataset and determined the parameters that affect the quality of the process**. According to the first results, we proposed to the plant some strategies for better control of data integrity (e.g., a second level of control for changes in the parameters of the operators) and two thermal imaging cameras have been installed in all die-casting machines.

Therefore, a large dataset of thermal images has been provided in addition to the process data, assuming that there is a correlation between the sensor data, the thermal data and the process result. Several models have been developed to utilize both sensor and thermal image data, as reported in D3.3[11], available on the I-BiDaaS' website. The second lesson we learned is that the initial data analysis, jointly with the CRF hackathon results (see the related blog article 'I-BiDaaS – CRF Hackathon'[12]), revealed that an additional dataset of different nature is to be included due to inherent performance limitations. This was not possible to assess beforehand, and the benchmarking and analysis through the hackathon helped to improve the process. In summary, it may be worthwhile for the data collection process-experiment design process and the data analysis process be iterative rather than sequential "in one shot".

The results can be made available to the end-user via the AEGIS's Advanced Visualization[13] module on which the I-BiDaaS User Interface is built upon and allows us to timely check the status of the process and to classify the quality levels that we are using as KPIs. The third lesson we learned is that Advanced visualizations are extremely useful for developing high value Big Data analytics solutions for domain experts and operators.

2. Maintenance and Monitoring of Production assets

The Maintenance and monitoring of production assets use case gathers data from the production process to define timely "predictive" maintenance measures **to identify and prevent failures before they occur. Different sensors are installed on the production line**, acquiring different types of data (e.g., sensors mounted on different machines, accelerometers mounted in linear stages). The sensors are connected to the control units on the line, while the control units are connected to a data server. The data consists of two different datasets, the SCADA and the MES. The SCADA dataset contains production, process and control parameters of the daily vehicle production. The MES data contains specific data associated with the type of vehicle being produced.

Initially, our intention was to use both types of data, but over time, we faced **problems retrieving MES data because of the scheduled activities and changes in the production lines, partially due to the Covid-19 Pandemic**. Therefore, we decided to utilize only SCADA data to obtain thresholds for abnormal measurements for all sensors. The fourth lesson we learned is that when defining an industrial use case, it turns out that it is very important to allow flexibility and redundancy in the use case definition so that the work can proceed under unexpected or severe conditions, including subsystem failure, unavailability of data, etc.

Due to our internal constraints, the lack of the necessary infrastructure and the high volume of available data, the **I-BiDaaS technical partners decided to bridge the I-BiDaaS infrastructure with the CRF internal server for real-time data transfer and near to real-time data analysis.**

[11] http://www.ibidaas.eu/sites/default/files/docs/ibidaas-d3.3.pdf

[12] http://www.ibidaas.eu/blog/I-BiDaaS–CRF-Hackathon/

[13] https://aegisresearch.eu/solutions/advanced-visualization-toolkit/

6 Appendix 1 – Software Checklist

Here is an example of the Software Readiness list collected for every component of the I-BiDaaS platform:

Software Readiness Checklist	YES	NO	N/A	Comments
DOCUMENTATION				
1. Readme file using markdown format (.md)	X			
2. Installation/configuration guides	X			
3. User/admin guides			X	
3. License file and license headers			X	
4. Documentation of interfaces and usage	X			
5. Usage of tools to auto-generate documentation from source code (JavaDoc, Sphinx, Swagger, AsciiDoc, …)	X			**Swagger**
SOURCE CODE LIFECYCLE MANAGEMENT				
1. Use of source code management system	X			**git**
2. Use of build management system			X	
3. SW artefacts uploaded to a repository (Nexus)			x	
4. Are the required support systems containerized? (Docker)	x			
FULL SOFTWARE RELEASE AUTOMOTION				
1. Use of continuous integration system			x	
2. Use of configuration management system			x	
2. Provision scripts	x			**Ansible installation script**
DEMO & SHOWCASE				
1. Dashboard			x	
2. Demo video (installation and configuration)		x		
3. Demo video (usage)	x			
QUALITY ASSURANCE				
1. Unit and End-to-End testing		x		
2. Code coverage up to 70%		x		
3. 'A' and technical debt bellow 5 days (Sonar)	x			**sonarcube**

7 Appendix 2 – Orchestrator REST API

Route	Type	Description
experiment/update-status/{id}	GET	Sets status of experiment to terminated. Used internally when compss terminates
custom-experiment/update-status/{id}	GET	Sets status of experiment to terminated. Used internally when compss terminates
project	GET	Gets the list of projects (predefined projects)
project/experiments/{id}	GET	Gets the list of project experiments owned by the current user, project is specified by project_id
project/{id}	GET	Gets the details of a project, specified by project_id
experiment/run/{project_id}	POST	Starts a new experiment for project, specified by project_id
experiment/{id}	GET	Gets the details of experiment, specified by id
experiment/stop/{id}	GET	Requests the termination of an experiment, specified by id
experiment/download/{id}	GET	Returns a package (tarball) of the results of an experiment, specified by id
experiment/{id}	DELETE	Deletes the specified experiment
experiment/output/{id}	GET	Returns the results of an experiment, in JSON format
input-dirs	GET	Returns a JSON representation of the available directories to be used as input, for the current user
input-files	GET	Returns a JSON representation of the available files to be used as input, for the current user
keyspaces	GET	Returns a JSON representation of the available Cassandra keyspaces to be used as input, for the current user
custom-project	POST	Creates new project, owned by the current user
custom-project	GET	Gets the list of custom projects of the current user
custom-project/{custom_project_id}	GET	Gets the details of a project, specified by project_id
custom-project-update	POST	Updates the parameters of a project, specified by project_id
custom-project/{custom_project_id}	DELETE	Deletes the specified project
custom-experiment/run/{custom_project_id}	POST	Starts a new experiment for project, specified by project_id

custom-experiment/{custom_experiment_id}	DELETE	Deletes the specified experiment
custom-project/experiments/{custom_project_id}	GET	Returns a list of experiments, belonging to a custom project, specified by custom_project_id
custom-experiment/stop/{custom_experiment_id}	GET	Requests the termination of an experiment, specified by custom_project_id
custom-experiment/download/{id}	GET	Returns a package (tarball) of the results of the specified experiment