D4.11 – Big Data Analytics Suite, v2

WP4 – BUILD: Manufacturing Data Analytics for Manufacturing Quality Assurance

i4Q has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958205.

Document Information

GRANT AGREEMENT NUMBER	958205	ACRONYM	i4Q	
FULL TITLE	Industrial Data Services for Quality Control in Smart Manufacturing			
START DATE	01-01-2021	DURATION	36 months	
PROJECT URL	https://www.i4q-project.eu/			
DELIVERABLE	D4.11 – Big Data Analytics Suite v2			
WORK PACKAGE	WP4 – BUILD: Manufacturing Data Analytics for Manufacturing Quality Assurance			
DATE OF DELIVERY	CONTRACTUAL	31-Dec-2022	ACTUAL	30-Dec-2022
NATURE	Other	DISSEMINATION LEVEL	Public	
LEAD BENEFICIARY	UNINOVA			
RESPONSIBLE AUTHOR	Paulo Figueiras - UNINOVA			
CONTRIBUTIONS FROM	1-CERTH, 2-ENG, 5-KBZ, 7-IKER,			
TARGET AUDIENCE	1) i4Q Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community			
DELIVERABLE CONTEXT/ DEPENDENCIES	This document describes the final release of the i4Q Big Data Analytics Suite (i4Q^{BDA}) solution. This is an updated deliverable with respect to its previous version D4.3 Big Data Analytics Suite. - D4.10 Services for Data Analytics: This deliverable describes the base solution (i4Q^{DA}) that will be provided when using the i4Q^{BDA}			
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None			
READING NOTES	None			
ABSTRACT	This deliverable describes the functionalities of the i4Q^{BDA} solution. Emphasis is given on the final release implemented on M24. This is an updated deliverable with respect to its previous version D4.3 Big Data Analytics Suite.			

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	01-Nov-2022	ToC	ToC created for v2	UNINOVA
0.2	15-Nov-2022	1st draft	First draft for v2	UNINOVA, CERTH, ENG, KBZ, IKER
0.3	01-Dec-2022	2nd draft	Second draft for V2	UNINOVA, CERTH, ENG, KBZ, IKER
0.4	09-Dec-2022	Internal Review	Internal Review	IBM, ITI
0.5	12-Dec-2022	3rd Draft	Addressing comments by the reviewers	UNINOVA
1.0	30-Dec-2022	Final Document	Final quality check and issue of final document	CERTH

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
BDA	Big Data Analytics Suite
DA	Data Analytics services
DB	Database
JSON	JavaScript Object Notation
SQL	Structured Query Language
RAM	Random Access Memory
UI	User Interface
YAML	Yet Another Markup Language
WP	Work Package

Executive summary

This document presents the final technical description of the **i4Q Big Data Analytics Suite** (**i4Q**^{BDA}) solution providing the general description, the technical specifications and the implementation status. This document **i4Q** D4.11 v2 is an update of v1 of D4.3, for this reason it contains information of the 1st version together with the updates developed in this 2nd version. The deliverable **D4.11** complements the Source Code of the i4Q^{BDA} solution that is in a private repository of Gitlab: https://gitlab.com/i4q/i4q-bda.

The documentation associated to the i4Q^{BDA} Solution is deployed on the website http://i4q.upv.es. This website contains the information of all the i4Q Solutions developed in the project "Industrial Data Services for Quality Control in Smart Manufacturing" (i4Q). The direct link to the i4Q^{BDA} Solution documentation is http://i4q.upv.es/11_i4Q_BDA/index.html.

Such documentation is structured according to:

- General description
- Features
- Images
- Authors
- Licensing
- Pricing
- Installation requirements
- Installation Instructions
- Technical specifications of the solution
- User manual

The i4Q^{BDA} provides core analytics functions like clustering, regression, classification, anomaly detection on both batch and stream industrial data. It builds on top of the i4Q Services for Data Analytics (**i4Q**^{DA}) solution to provide a complete solution for local or cloud deployments.

Document structure

Section 1: Contains a general description of the **i4Q Big Data Analytics Suite**, providing an overview and the list of features. It is addressed to final users of the i4Q Solution.

Section 2: Contains the technical specifications of the **i4Q Big Data Analytics Suite**, providing an overview and its architecture diagram. It is addressed to software developers.

Section 3: Details the implementation status of the **i4Q Big Data Analytics Suite**, explaining the current status and summarizing the implementation history.

Section 4: Provides the conclusions.

APPENDIX I: Provides the PDF version of the **i4Q Big Data Analytics Suite** web documentation, which can be accessed online at: **http://i4q.upv.es/11_i4Q_BDA/index.html**

1. General Description

1.1 Overview

The i4Q Big Data Analytics Suite (**i4Q^{BDA}**) enables the encapsulation of the i4Q Services for Data Analytics (**i4Q^{DA}**) solution, which is a data analytics environment that enables the creation of AI workflows in a simple and intuitive, code-free manner, through a visual programming environment For more information about the **i4Q^{DA}** solution, please refer to deliverable **D4.10 – Services for Data Analytics, v2**.

The i4Q Big Data Analytics Suite (**i4Q^{BDA}**) provides this encapsulation in a ready-to-deploy software bundle that can be deployed in any hardware infrastructure, whether it is an on-premises server or a cloud instance. This means that, while the Services for Data Analytics solution is a service that can be accessed through a Web user interface and is housed on a central server, the **i4Q^{BDA}** builds on top of that and provides a way for end users to customize and deploy the **i4Q^{DA}** solution on their own servers or cloud instances, in an optimized way and using already deployed technologies on those servers.

Users will have access to a user interface (UI), where they can select and/or configure the supported technologies from the **i4Q^{DA}**, to be encapsulated and then deployed. This encapsulation will be dependent on the hardware infrastructure where the user wants to execute the **i4Q^{BDA}**, providing the best optimization specifications of the solution for the hardware at hand. However, the current implementation of the **i4Q^{BDA}** requires that users give the required specifications, like the location the service (cloud or local), number of workers to deploy (or that are already deployed) or the amount of RAM memory available as well as free space in disk, for each chosen technology.

The supporting technology responsible for the execution of the encapsulated solutions is the containerization software Docker. This means that the encapsulation that is done from the UI will result in Docker readable images/containers. For more information about the **i4Q^{DA}** solution, please refer to deliverable **D4.10 – Services for Data Analytics, v2**.

1.2 Features

The **i4Q^{BDA}** main focus, as mentioned previously, is the capability of containerization/encapsulation of the functionalities related to the **i4Q^{DA}** solution. As such, the main functionalities for this solution are the following:

- UI for selecting (**Figure 1**) and customizing the resources/parameters (**Figure 2**) for the technologies chosen (e.g., Spark, TensorFlow, MongoDB, etc.), as well as for the services provided by the **i4Q^{DA}** solution (AI workflow creation through drag-and-drop visual programming).
- Encapsulation/containerization of the technologies and services, ready to deploy and run on any hardware infrastructure, in the form of a Docker Compose file.

2. Technical Specifications

2.1 Overview

The i4Q[BDA] solution functionalities can be divided into two, namely:

- Selection and customization of technologies through a UI.
- Backend Logic for the creation of a software bundle, based on Docker Compose containers.

Regarding the selection and customization of technologies a UI provides all the necessary tools for the creation of a bundle of various technologies. Some of these technologies are PostgreSQL, MongoDB for connecting to data, Spark, Pandas for data manipulation or preparation, Airflow to programmatically author, schedule and monitor workflows and TensorFlow or Keras to create train and run AI models.

A user is able to select from the catalogue of technologies already offered through the i4Q[DA] and while selecting the ones that are necessary for a specific use case, users are presented with a configuration window for each of the selected technologies. This provides a personalized customization that will be optimized for the hardware where the containerized solution will be executed. Furthermore, if any of these technologies are already deployed for the use case, then the i4Q[BDA] enables the utilization of these technologies with the services in the bundle as a multi-tenant environment.

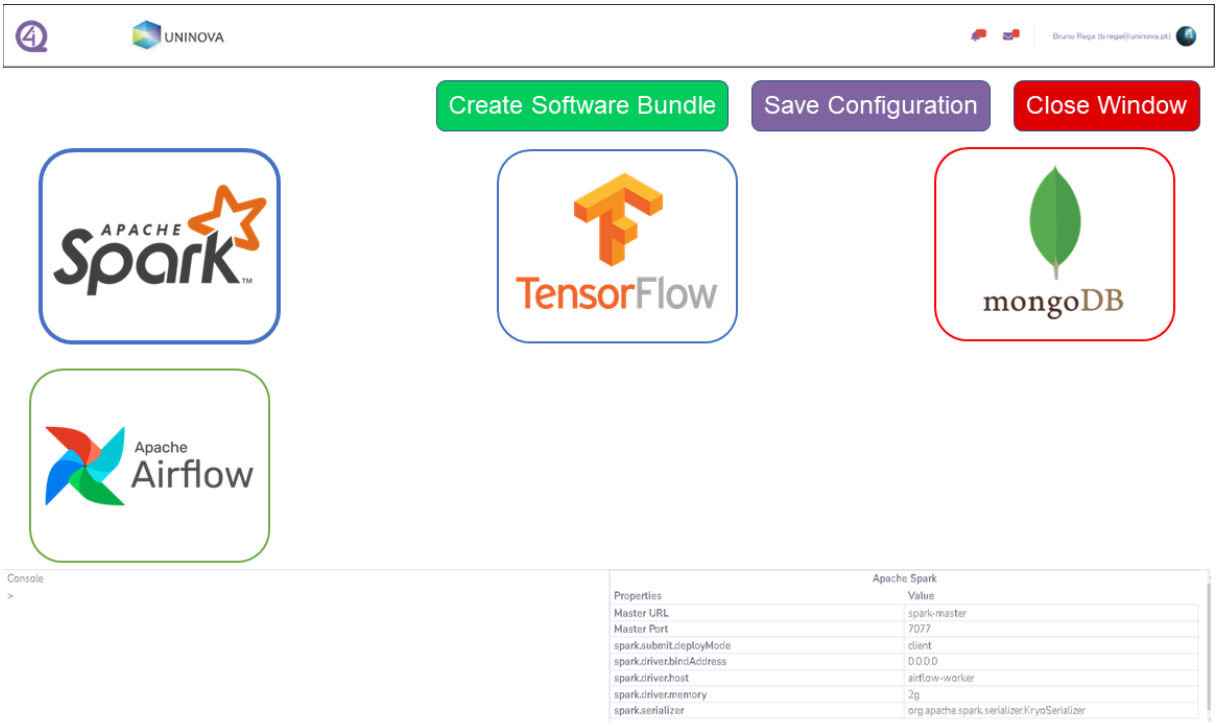

Figure 1.BDA user interface

The backend logic refers to how this solution will build the bundle, in the backend, while the user interacts with the UI. The information about the selected technologies and the respective customization parameters are represented in a JSON format. The JSON format is a very light weight format, has a built-in package in Python to work with JSON data and it is easy to read and

to store. Finally, the JSON-based information is converted to YAML, which is the main format for Docker Compose files. This is the final offering of this solution: A Docker Compose file that can be used to deploy all the selected and customized technologies, as well as the services offered by the i4Q[DA], on the target infrastructure.

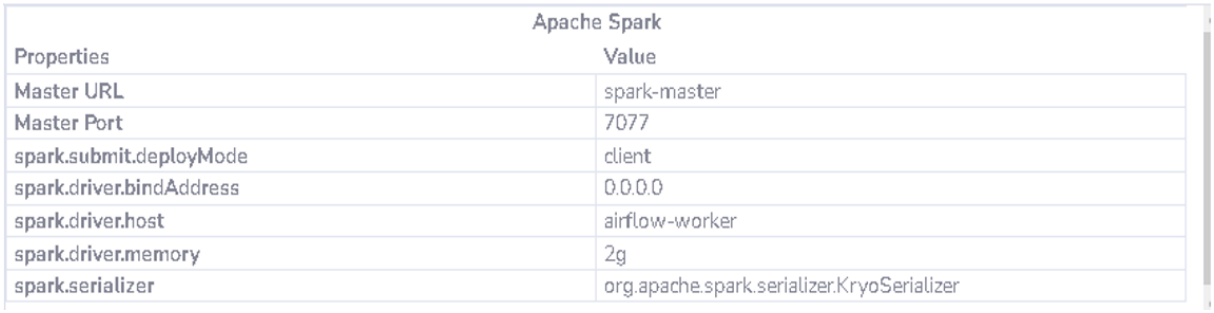

Figure 2. Apache Spark configuration

In this case, the user selected TensorFlow and Apache Spark to be present in the software bundle and is customizing the Apache Spark instance that will be contained in the bundle. For Apache Spark, for instance, users can select the number of workers that they wish to deploy, the memory and number of cores for each worker, as well as several other Apache Spark configuration parameters. Technologies highlighted in blue (Apache Spark and TensorFlow) are the ones selected by users to be contained in the bundle, the thicker blue selection indicates that this is the technology that the configuration parameters are displayed, technologies highlighted in red are excluded from the software bundle (MongoDB) and technologies highlighted in green are mandatory to be contained in the bundle. In the example, the user already has a MongoDB instance running on premises, and so MongoDB is not selected to be bundled (highlighted in red). When the user deploys the bundle, the Configure Technologies interface of the **i4Q**[DA] can be used to configure the deployed instance to use the already existing MongoDB instance, as explained in deliverable ***D4.10 – Services for Data Analytics, v2***.

The final release of the **i4Q**[BDA] solution comprises a proof-of-concept feature for the creation of the software bundle from hardware specifications. When a user clicks outside any of the technology catalogue boxes, the configuration window shown in **Figure 3** is presented. Users are prompted to input some hardware specifications, such as number of cores and available RAM memory in the single node in which the software bundle will be deployed. This proof-of-concept only concerns single node instances for now.

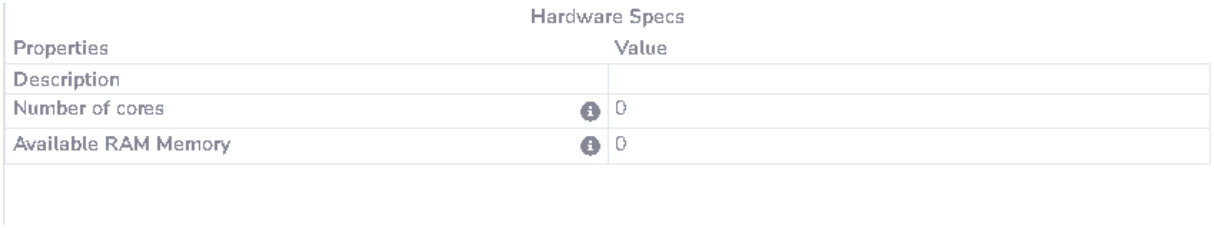

Figure 3. Hardware specification configuration

From the information about the single node instance's specifications, the **i4Q**[BDA] solution will calculate the best distribution of worker instances for each distributed technology that was

selected to be part of the bundle. For now, the proof-of-concept only calculates the number of Apache Spark workers, depending on the available cores and RAM memory.

To calculate the right distribution of cores and RAM memory for each worker, the following assumptions were used:

- Let M_A be the available RAM memory and C_A the available number of cores on the single node instance;
- First, a portion of memory has to be left for other processes than the workers' processes. This portion is calculated as one tenth of the available memory, ceiled to the next integer value above:

$$M = M_A - ceil\left({}^{M_A}/_{10}\right)$$

- The number of Spark processing nodes (driver + workers) is calculated as the integer division of the memory M by 4 (minimum RAM amount, in GB, for each instance):

$$Nodes = M\%4$$

- The memory for the Spark driver is calculated as the minimum memory for one of the nodes, plus the remainder of the above integer division:

$$M_D = 4 + r(M\%4)$$

- The memory for each Spark worker is calculated as:

$$M_W = \frac{M - M_D}{Nodes - 1}$$

- Regarding the cores, a similar calculation is performed. First, a tenth of the available cores are left for other processes outside the Spark processing processes:

$$C = C_A - ceil\left({}^{C_A}/_{10}\right)$$

- The same number of cores is saved for the Spark driver:

$$C_D = C - ceil\left({}^{C_A}/_{10}\right)$$

- Finally, the number of cores per worker is calculated as follows:

$$C_W = \frac{C - C_D}{Nodes - 1}$$

These figures will be used in the creation of the final software bundle, in which the number of Spark workers is given by $Nodes - 1$.

When the selection and customization process is finalized, users click on the "Create Software Bundle" button (**Figure 1**) and the backend logic for the creation of the software bundle starts. The bundle is delivered as a Docker Compose file that is provided to the user, through a direct download dialog. **Figure 4** depicts an excerpt of a Docker Compose file created through the **i4Q**^{BDA} solution, in which Apache Airflow (mandatory) and Apache Spark (selected by the user) are defined and customized. This file can then be executed to deploy the respective Airflow and Spark containers on the user's server.

```yaml
airflow-cli:
  <<: *airflow-common
  profiles:
    - debug
  environment:
    <<: *airflow-common-env
    CONNECTION_CHECK_MAX_COUNT: "0"
  # Workaround for entrypoint issue. See: https://github.com/
  command:
    - bash
    - -c
    - airflow

flower:
  <<: *airflow-common
  command: celery flower
  ports:
    - 5555:5555
  healthcheck:
    test: ["CMD", "curl", "--fail", "http://localhost:5555/"]
    interval: 10s
    timeout: 10s
    retries: 5
  restart: always
  depends_on:
    <<: *airflow-common-depends-on
    airflow-init:
      condition: service_completed_successfully

spark:
  #image: docker.io/bitnami/spark:3
  build: ./spark
  environment:
    - SPARK_MODE=master
    - SPARK_RPC_AUTHENTICATION_ENABLED=no
    - SPARK_RPC_ENCRYPTION_ENABLED=no
    - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
    - SPARK_SSL_ENABLED=no
  ports:
    - '8081:8080'
  volumes:
    - ./_plugins:/opt/airflow/plugins
spark-worker:
  #image: docker.io/bitnami/spark:3
  build: ./spark
  environment:
    - SPARK_MODE=worker
    - SPARK_MASTER_URL=spark://spark:7077
    - SPARK_WORKER_MEMORY=1G
    - SPARK_WORKER_CORES=1
    - SPARK_RPC_AUTHENTICATION_ENABLED=no
    - SPARK_RPC_ENCRYPTION_ENABLED=no
    - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
    - SPARK_SSL_ENABLED=no
  ports:
    - '8082:8081'
  volumes:
    - ./_plugins:/opt/airflow/plugins
  depends_on:
    spark:
      condition: service_started
```

Figure 4. Excerpt of the resulting Docker Compose file that comprises the software bundle

2.2 Architecture Diagram

The **i4Q**[BDA] solution, as in the case of **i4Q**[DA], includes processes and services to create AI pipelines which are mapped to the Platform and Edge Tiers of the i4Q Reference Architecture as depicted in **Figure 5**.

- **Platform Tier**: The services that are being used in this tier are the "Distribution Services", the "Data Brokering and Storage", "Data Transformation", "Orchestration management" and the "Data Analytics". Amongst these services, the most important for this solution is the "Data Analytics and services", which is in charge of analysing big amounts of data that can come in streams or in a batch while providing many types of data analytic techniques aided by machine learning, that can be adapted based on the data type and the type of analysis. This customization, which can be done by the end-user, makes this a versatile analytical tool.

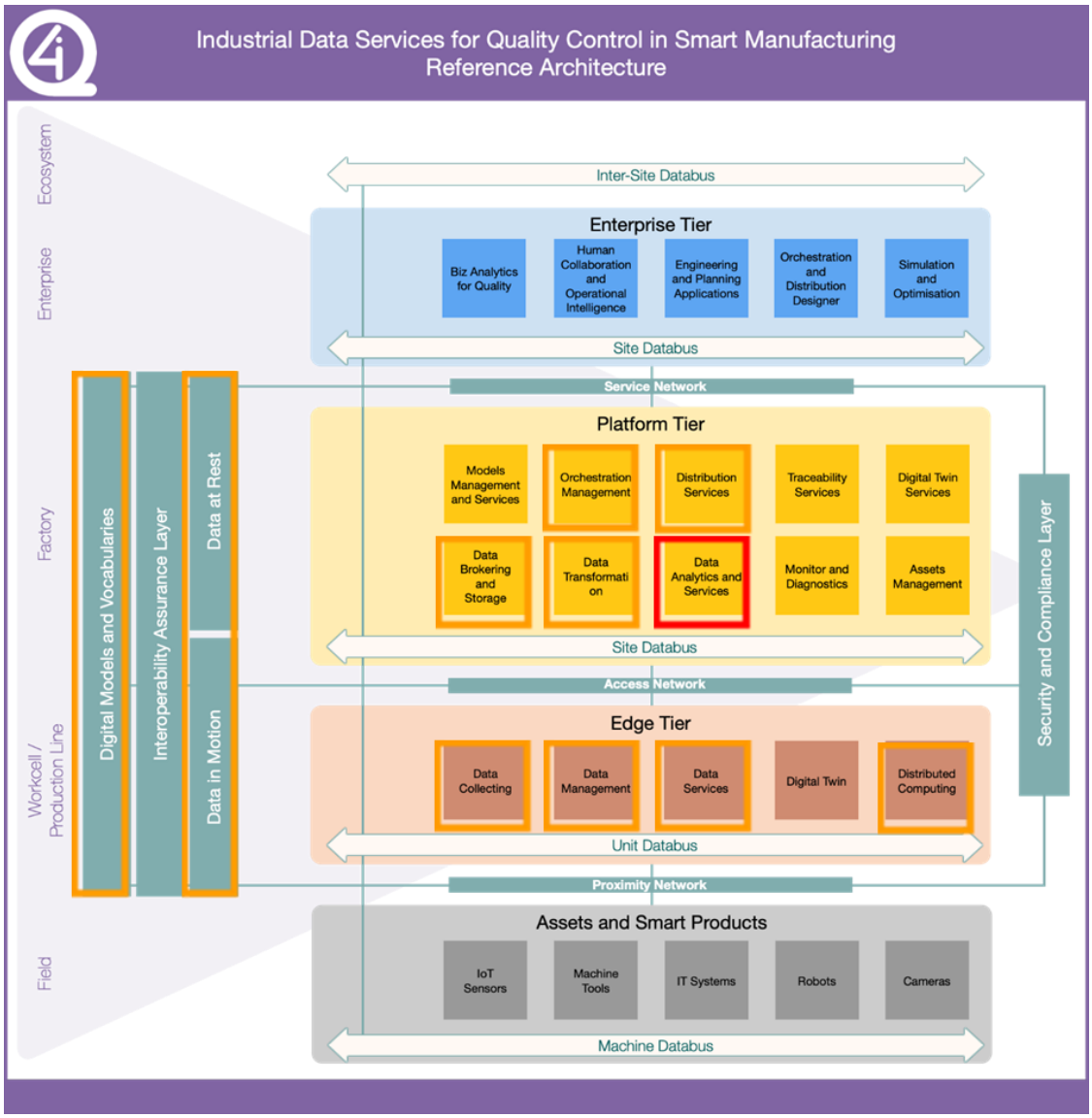

Figure 5. Reference Architecture

- **Edge Tier**: The **i4Q**[BDA] solution is comprised of the "Data Collecting" and "Data Management" services, which will be present in the tool itself and are fundamental for the data analytics, and the "Distributed Computing" services, which will allow this tool to be executed in any type of computer/device.

3. Implementation Status

3.1 Current implementation

A summarized list of the features of the software bundle creation are presented below:

- User interface for the configuration of the bundle to be deployed.
- Library of technologies to be deployed.
- Customizable technology configuration.
- On-Demand deployment files.
- Proof-of-concept solution for the distribution of worker instances for the available distributed technologies (only Apache Spark is supported at this time).

3.1.1 Solution features analysed and mapping with user requirements

Since the **i4Q**[BDA] solution will be based on the encapsulation of the **i4Q**[DA] solution as a software bundle, along with all the necessary technologies, the mapping of the user requirements and analyzed features is similar to the one presented in ***D4.10 – Services for Data Analytics, v2***. Please refer to the aforementioned deliverable for the mapping.

3.2 History

Version	Release date	New features
0.1	30-03-2022	Convert JSON to YAML
0.2	30-04-2022	Backend logic for the creation of Docker Compose files
0.3	30-05-2022	i4Q Big Data Analytics Suite User Interface (version 1)
1.0	30-06-2022	Release 1.0
1.1	30-11-2022	Proof-of-concept for the distribution of worker node instances (Apache Spark)
2.0	20-12-2022	Release 2.0

Table 1. History

4. Conclusions

Deliverable D4.11 *Big Data Analytics Suite, v2* is a technical specification document, providing an in-depth technical overview of the i4Q^{BDA} solution. It describes in detail the role, the functionalities, and the conceptual architecture of i4Q^{BDA}. The major features of the solution are detailed in order to clarify the fundamental functionalities and objectives of i4Q^{BDA}. A technical overview is presented, describing its architecture diagram with respect to the i4Q Reference Architecture.

The final implementation status of i4Q^{BDA} is detailed, presenting the significant progress of its overall development as well as the final release of the solution.

Appendix I

The PDF version of the **i4Q Big Data Analytics Suite** web documentation can be accessed online at: **http://i4q.upv.es/11_i4Q_BDA/index.html**