D3.1 Big data and application infrastructure design

DOC. REFERENCE: EDI-WP3-D31-REP-180629v10
RESPONSIBLE: DEUSTO
AUTHOR(S): DEUSTO, ENG
DATE OF ISSUE: 29/06/2018
STATUS: FINAL
DISSEMINATION LEVEL: PUBLIC

<table>
<thead>
<tr>
<th>VERSION</th>
<th>DATE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>v01</td>
<td>15/05/18</td>
<td>First Version of deliverable contents prepared by Unai Aguilera</td>
</tr>
<tr>
<td>v02</td>
<td>05/05/18</td>
<td>Additions to Section 2 by Mikel Emaldi</td>
</tr>
<tr>
<td>v03</td>
<td>06/05/18</td>
<td>Minor changes by Unai Aguilera</td>
</tr>
<tr>
<td>v04</td>
<td>14/05/18</td>
<td>Contributions to Section 3 added by Ciro Formisano</td>
</tr>
<tr>
<td>v05</td>
<td>15/05/18</td>
<td>Executive Summary and Conclusion added by Unai Aguilera</td>
</tr>
<tr>
<td>v06</td>
<td>18/06/18</td>
<td>Revision by Mikel Emaldi and Diego López-de-Ipiña</td>
</tr>
<tr>
<td>v07</td>
<td>19/06/18</td>
<td>Final draft version by Unai Aguilera</td>
</tr>
<tr>
<td>v08</td>
<td>26/06/18</td>
<td>Review by ENG and EHS processed by Unai Aguilera</td>
</tr>
</tbody>
</table>

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 779790
# Table of Contents

Glossary.................................................................................................................................................. 4
Executive summary..................................................................................................................................... 6

1. Introduction ........................................................................................................................................... 7

2. Big Data Infrastructure ........................................................................................................................ 8
   2.1 Usage scenarios ................................................................................................................................. 8
       2.1.1 Data processing ......................................................................................................................... 8
       2.1.2 Data accessibility & multi-user support .................................................................................. 9
       2.1.3 Debugging & performance assessment ................................................................................... 9
       2.1.4 Application infrastructure ..................................................................................................... 10
   2.2 Requirements .................................................................................................................................. 10
   2.3 Data platform selection ................................................................................................................... 12
       2.3.1 Cloudera CDH ......................................................................................................................... 12
       2.3.1.1 Hortonworks HDP ................................................................................................................ 13
       2.3.2 MapR CDP ............................................................................................................................... 14
   2.3.3 Justification of Hortonworks HDP as the selected data platform ............................................... 14
   2.4 Data Platform architecture ............................................................................................................. 15
       2.4.1 Management and monitoring ................................................................................................. 15
       2.4.2 Data processing ....................................................................................................................... 16
       2.4.3 Security .................................................................................................................................... 18
   2.5 Client tool ......................................................................................................................................... 20
   2.6 Hardware requirements .................................................................................................................. 21

3. Application Infrastructure ................................................................................................................... 22
   3.1 Requirements ................................................................................................................................... 22
   3.2 FI-WARE .......................................................................................................................................... 23
       3.2.1 FI-WARE lab ............................................................................................................................. 24

4. Data Catalogue ..................................................................................................................................... 25
   4.1 Requirements ................................................................................................................................... 25
   4.2 Implementation ............................................................................................................................... 26
   4.3 Software contributions .................................................................................................................... 29

5. Conclusion .......................................................................................................................................... 29
Index of Figures

Figure 1: EDI’s incubation process funnel approach (repeated three times during project duration) .... 7
Figure 2: Overview of tools offered by Cloudera CDH ........................................................................... 13
Figure 3: Overview of tools offered by Hortonworks HDP ........................................................................ 13
Figure 4: MapR Converged Data Platform ................................................................................................ 14
Figure 5: Ambari Metrics view .................................................................................................................. 15
Figure 6: Ambari Log Search .................................................................................................................. 16
Figure 7: HDFS architecture. .................................................................................................................... 17
Figure 8: Tools and frameworks offered to sub-grantees by the Big Data Stack ........................................ 17
Figure 9: Apache Ranger auditing dashboard ............................................................................................ 19
Figure 10: WebHDFS from Ambari Views .................................................................................................. 20
Figure 11: EDI’s Big Data Stack hardware requirements. ......................................................................... 21

Figure 12: FI-WARE Generic Enabler’s Catalogue .................................................................................. 24
Figure 13: FI-WARE Lab Interface ........................................................................................................... 25

Figure 14. Customization of the CKAN’s user interface to the style of the EDI project .......................... 27
Figure 15. Data Catalogue top menu customization ............................................................................... 27
Figure 16. Navigable list of data providers .............................................................................................. 27
Figure 17. List of challenge categories in EDI........................................................................................ 28
Figure 18. Account registration form gathering explicit consent for the data processing in EDI......... 28

Index of Tables

Table 1. Data processing requirements ....................................................................................................... 10
Table 2. Security requirements .................................................................................................................. 11
Table 3. Fair sharing requirements ............................................................................................................. 11
Table 4. Accountability requirements ......................................................................................................... 12
Table 5. CKAN modifications ..................................................................................................................... 29
Glossary

<p>| AD   | Active Directory                  |
| AMQP | Advanced Message Queuing Protocol |
| API  | Application Programming Interface |
| CDH  | Cloudera Distribution including Apache Hadoop |
| CDP  | Converged Data Platform           |
| CLI  | Command Line Interface            |
| CPU  | Central Processing Unit           |
| GE   | Generic Enabler                   |
| HDF  | Hortonworks Data Flow             |
| HDFS | Hadoop Distributed File System    |
| HDP  | Hortonworks Data Platform         |
| IaaS | Infrastructure as a Service       |
| IP   | Internet Protocol                 |
| JMS  | Java Message Service              |
| LDAP | Lightweight Directory Access Protocol |
| RAM  | Random Access Memory              |
| SSH  | Secure Shell                      |
| SQL  | Structured Query Language         |</p>
<table>
<thead>
<tr>
<th>UI</th>
<th>User Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>YARN</td>
<td>Yet Another Resource Negotiator</td>
</tr>
</tbody>
</table>
Executive summary

This deliverable, named “D3.1 Big Data and Application Infrastructure Design”, reports the main results of the selection, adaptation and deployment of the different tools that will be available to participants and sub-grantees during the different phases of the three incubation rounds planned in the EDI project. In short, the document contains the description of three main elements. First, the Big Data infrastructure, a computation platform freely provided to sub-grantees to perform the analysis of the data during the Experiment and Evolve stages of the incubation process. Second, the Application infrastructure, which provides an application deployment environment during the Explore, Experiment and Evolve stages of each of the three incubation rounds. Finally, the Data Catalogue, which contains the data samples available during the Open Calls, which precede the execution of the incubation process, when the applicants are writing their proposals.
1. Introduction

“D3.1 Big Data and application infrastructure design” reports the main results of the selection, adaptation and deployment of the different tools that will be available to participants and sub-grantees during the three different rounds of incubation comprised by the Open Call << Explore << Experiment << Evolve stages (see Figure 1) planned by the EDI project.

In particular, this document focuses on describing the following components of the infrastructure:

- **Big data tools.** A toolset containing different complementary big data tools which allows developers the processing of massive datasets in a distributed highly scalable manner. This computation infrastructure is available during the Experiment and Evolve phases. Section 2 describes the envisioned usage scenarios for the Big Data Platform, the requirements extracted from these scenarios, the selected set of tools and its justification that fulfil the identified requirements and a description of the minimum hardware and computation infrastructure required to deploy the selected tools.

- **Application infrastructure.** This infrastructure allows developers to deploy their own application containers to execute software not provided by the Big Data Stack and their own final applications. Section 3 discusses the characteristics, requirements and tools of this infrastructure.

- **Data Catalogue.** A CKAN-based extension to publish metadata and excerpts of datasets from providers, including details about the authentication and authorization and exploitation mechanisms opened for third party usage. In order to allow all participants to evaluate the challenges and their associated data samples, the Data Catalogue is available as soon as the Open Call stage, part of the EDI incubation process, starts. Section 4 provides detailed information about the aim of this component, its requirements and its architecture.

![Figure 1: EDI’s incubation process funnel approach (repeated three times during project duration)](image-url)
2. Big Data Infrastructure

During the Experiment and Evolve stages of each of the three rounds of incubation of the EDI project, the selected sub-grantees have the possibility to use the computation infrastructure that the EDI project offers freely, as part of the incubation process.

The Big Data community is continuously producing new tools and architectures, resulting in a heterogeneous ecosystem of solutions and techniques that are currently available to solve a wide range of computation problems. In addition, the data providers of the EDI project can propose a great variety of challenges and expect very different type of results. Furthermore, there are multiple different ways that the sub-grantees could tackle the same challenge, requiring the integration of many tools to process the data and obtain valuable results.

Therefore, the aim of the proposed Big Data infrastructure is to provide a generic set of tools and computational resources that are ready to use and directly applicable to the majority of the possible solutions that sub-grantees could propose during the Explore phase. Due the required generality of the Big Data infrastructure, the provided infrastructure offers a common set of general tools and computation resources. However, if the freely available set of tools and computational resources are not appropriate for their requirements, sub-grantees are free to deploy and use their own resources from their particular infrastructures.

With the aim of selecting a set of tools that can be used during the Experiment and Evolve stages by the greatest number of sub-grantees, the Big Data infrastructure design process has started with the definition of the minimum supported usage scenarios, which have been later used to extract the minimum requirements of the platform. Part of these requirements concerns data processing and security: in particular, the latter is essential due to the shared nature of the computation environment. Finally, the best set of tools that fulfil the identified functional and non-functional requirements have been selected.

The last part of this section provides useful information about the minimum hardware requirements for the deployment of the Big Data tools. This information will serve as input for the selection and configuration of the computation infrastructure that will be developed as part of T3.3 – Big Data and Application Infrastructure operation.

2.1 Usage scenarios

This section describes the expected usage, which the sub-grantees will made, of the Big Data infrastructure during the Experiment and Evolve stages of the EDI incubation process. The current state of the art of the available tools for Big Data together with the proposed challenges and sample data offered by the data providers for the 1st Open Call of the project provide the basis for these descriptions. In addition, these descriptions aim to achieve a general infrastructure that sub-grantees can use to solve the widest range of problems.

2.1.1 Data processing

There are the main approaches in the current trends for processing in Big Data: batch and stream processing. Therefore, the Big Data infrastructure should provide sub-grantees with the basic set of tools...
allowing them to implement their algorithms using the approach that better adapts to their problem, data and computation needs.

- **Batch processing.** In this case, the data is usually available in the form of “complete” datasets that have been stored and are ready for an algorithm to process. The transformations and algorithms in this approach usually require minutes or hours to finish and produce the results. Batch processing can be viewed as a general case of the stream processing, allowing the application of algorithms from the latter approach by considering that the data is composed of small chunks or data (e.g. processing the data using a row at a time), which is a paradigm known as micro-batching [2].

- **Stream processing.** The processing of stream data requires the infrastructure to be able to compute a function, or a sequence of functions, to a single data element (or some small chunk of data) to produce some output and, in some cases, to generate a real-time response to some external event that has produced the data. Computations for each data element are generally independent from each other, however, in some problems, there could be the need to apply some analysis to the data received inside a temporal window.

### 2.1.2 Data accessibility & multi-user support

During the Experiment and Evolve stages, multiple sub-grantees will use the infrastructure to read the data provider’s data, mix it with their own source and apply their computation algorithms, while, at the same time, they will generate their own output data, writing it again to the internal storage of the infrastructure.

During the Experiment and Evolve phase, each sub-grantee will work on different challenges, requiring them to sign an agreement with the data provider in order to guarantee an adequate usage and exploitation of the supplied data. On the other hand, multiple sub-grantees could be working on the same challenge, competing to produce the best solution to the data provider’s problem. This means that their algorithms and their data should be isolated from other users, avoiding an unfair access to other sub-grantees’ solutions and complying with the data provider’s agreements and data regulations.

On the other hand, the data providers are interested in the usage that the users will make of their data: access, copying, etc. Therefore, the infrastructure will provide mechanisms to allow administrators and data providers to control and check this usage, in order to guarantee that all the processes perform in accordance of the agreements and the required data privacy.

### 2.1.3 Debugging & performance assessment

As the computational resources provided by the Big Data infrastructure are limited and all the sub-grantees will share them during the Experiment and Evolve stages, we expect that developers of the Big Data infrastructure will debug and test that their algorithms execute correctly using some sample data before submitting the works to the infrastructure.

However, the development of computation algorithms requires measuring the performance during the real deployment of the execution. In addition, the administrators of the platform will also monitor the usage of the platform by the different users in order to detect an incorrect usage or, if required, to change the configuration of the different parameters to adapt them to the current workload of the platform.
Therefore, the Big Data infrastructure will include a set of tools allowing users and administrators to assess the behaviour of the particular executed algorithms and of the different components, respectively. This information will be useful to improve the Big Data infrastructure during the Experiment and Evolve phases and for the next Open Calls.

2.1.4 Application infrastructure

Although the Big Data infrastructure focuses on providing the computational resources and tools for the data processing, the sub-grantees are free to create their own applications on top of the Application infrastructure. For example, specific databases, indexes, APIs or web interfaces. Therefore, the project provides the Application infrastructure, detailed in Section 3, as a resource freely available to participants during all stages of the incubation process (even in Explore stage whilst they are developing the first mock-up), allowing them to host these specific application services.

The Application Infrastructure and the Big Data infrastructure will require to communicate with each other, for example, to propagate the latest results from the data processing, or to include new data pushed from the application level into the analysis. Therefore, the Big Data infrastructure will provide mechanisms to perform this communication between the different layers of the proposed solutions.

As the deployment of the Big Data infrastructure and the Application infrastructure during the Experiment and Evolve stages will take place in different subnetworks, there would be some impact in the performance of the developed applications. However, the combination of the two infrastructures and the range of available tools should be enough for the sub-grantees to demonstrate the feasibility of their proposals and prototypes during the length of their incubation process.

2.2 Requirements

The following tables contain the requirements extracted from the scenarios previously described that the Big Data infrastructure must satisfy to provide a general and usable computation platform for the sub-grantees.

<table>
<thead>
<tr>
<th>Data Processing</th>
<th>Batch Processing</th>
<th>Stream Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Batch Processing</strong></td>
<td>The Big Data platform must provide users with a set of tools to perform batch computation on datasets. Although the same computation could be done applying a stream-processing paradigm, these tools will enable to use the wide range of existing computation for batch processing and to provide a more flexible set of computation tools to the sub-grantees.</td>
<td>The platform must include tools for stream processing, allowing users to perform computation on single (or small sets) of data elements and, therefore, enabling the possibility to process data as it is produced.</td>
</tr>
</tbody>
</table>
### Table 2. Security requirements

<table>
<thead>
<tr>
<th>Security</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User authentication</strong></td>
<td>As the Big Data platform is a shared computation environment, the platform must contain mechanisms to identify and authenticate users (i.e. user/password). The platform must allow to create different users assigning them to groups, and to perform the related management processes (deletion, blocking, etc.).</td>
</tr>
<tr>
<td><strong>User authorization</strong></td>
<td>The platform must control and restrict the access of the users to the different resources and data hosted at the platform. Each sub-grantee will be only allowed to read the data that corresponds to its assigned challenge. Furthermore, due the competition and privacy regulations that will exist for each challenge, each sub-grantee team must only access to the data that they have produced, in addition to the data provided for their assigned challenge, creating isolated work spaces from other sub-grantees. The access restriction must not only include the data but also other resources as job and message queues, databases, source code, and so on, guaranteeing that only the authors can access to their own resources.</td>
</tr>
</tbody>
</table>

### Table 3. Fair sharing requirements

<table>
<thead>
<tr>
<th>Fair sharing of resources</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage</strong></td>
<td>The Big Data Platform provides a shared storage to all the users of the platform. In order to guarantee a fair usage of the available resources, the platform will initially assign an equal amount of storage to all the sub-grantees during the Experiment and Evolve phases. In addition, the platform will provide the functionality to manually increase or reduce these initial assigned quotas to accommodate to the specific needs of the sub-grantees.</td>
</tr>
<tr>
<td><strong>Processing</strong></td>
<td>The platform must guarantee a fair usage of the computation resources by all the sub-grantees. Therefore, the platform must assign an equal amount of computational power to all the sub-grantees accessing at the same time to the platform, while maximizing, at the same time, the usage of the available resources. However, some of the total computational power will be reserved for administration purposes.</td>
</tr>
</tbody>
</table>
### Accountability

<table>
<thead>
<tr>
<th><strong>Logging</strong></th>
<th>The platform must register all the actions performed by users, enabling to apply controls and audits to the platform’s usage. In particular, the platform must log the following data: users’ login, file reads, writes and code executions and guarantee that this information must be only accessible by the platform administrators during the Experiment and Evolve stages, in order to guarantee a fair participation of all the sub-grantees. Regarding to data providers, the platform will allow them to monitor that the allowed sub-grantees are the only ones accessing to their data.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report assessment</strong></td>
<td>The platform must provide a set of tools to inspect and control the logged data in an efficient way. The collected data will be used to perform compliance checks with the GDPR and privacy requirements that could arise from the usage of sensitive data. In addition, these tools must allow the generation of reports that will be very valuable for the different members of the EDI consortium (e.g. data providers) to improve for the following incubation rounds to be organized by EDI (up to 3 during the project duration). Furthermore, a subset of this information must be accessible to the users in order to allow them to improve/tune their proposed algorithms and solutions. Their solutions should not only be feasible technically but also economically in terms of their associated computation and storage costs.</td>
</tr>
</tbody>
</table>

#### 2.3 Data platform selection

As introduced in Section 2, during the past years a heterogeneous ecosystem of tools has been created around Big Data. This ecosystem is continuously evolving and producing new and innovative tools. With the aim of selecting, integrating and managing a set of tools for covering different needs at the processing and analysis of Big Data, different vendors provide their own “Apache Hadoop distribution”. Those distributions, although they offer a set of common tools, have their own particular characteristics. At this section a set of popular and well-known solutions, i.e. Cloudera CDH, Hortonworks HDP and MapR CDP, are analysed.

#### 2.3.1 Cloudera CDH

Cloudera Distribution Including Apache Hadoop (Cloudera CDH) [3] delivers the core elements of Hadoop along with a Web-based user interface for managing CDH based clusters. Cloudera CDH offers the most popular tools for different purposes, like HDFS as a file system; Hive, Pig and MapReduce for batch processing; Spark for both batch and stream processing; SQL and NoSQL tools like Impala or HBase; and tools for accessing security like Sentry and RecordService. While those tools offered by CDH are open-source, it is not the case of the cluster management web-interface, named Cloudera Manager. Its commercial approach consists on selling commercial software on top of its open source platform. In addition to its open source version, Cloudera provides different paid services. Figure 2 depicts the basic architecture of Cloudera CDH.
2.3.1.1 Hortonworks HDP

Hortonworks Data Platform (HDP) [4] is a fully open source Apache Hadoop distribution. HDP bases on a centralized architecture, which leverages on YARN the allocation of resources need, by different tools. Among those tools, HDP offers Apache Ambari for managing HDP based clusters and different tools for batch and stream processing, SQL and NoSQL databases, data workflow management, data governance, authentication and authorization, and other administrative and data processing tools, as Figure 3 shows.

Opposite to Cloudera, the commercial approach of Hortonworks advantages on offering paid support over their fully open source distribution. In addition, Hortonworks is backed by its own and the Apache community and has a complete support documentation. Its wide community and the open source nature
of all its components have been determinant aspects for selecting Hortonworks HDP for deploying the Big Data Stack.

2.3.2 MapR CDP
MapR Converged Data Platform [5] is a distribution combining open-source engines and tools with commercial engines and applications. Opposite than other distributions, MapR CDP offers its own file system named MapR-FS, providing APIs for connecting them with applications written for interacting with HDFS. Figure 4 shows the main components of the architecture of the MapR distribution. Although it offers a wide variety of open source tools, this distribution was finally discarded due the big number of proprietary tools and systems offered.

![Figure 4: MapR Converged Data Platform](image)

2.3.3 Justification of Hortonworks HDP as the selected data platform
The following features have been key for selecting Hortonworks as the data platform for implementing the Big Data Stack:

- **Fully open source platform** that allows sub-grantees applying their algorithms and their knowledge in another Hortonworks HDP instance after the ending of the incubation process, without tying them to any proprietary solution.
- **Efforts put on security** by Hortonworks HDP by distributing tools like Apache Ranger, and the integration of the whole system with Kerberos and LDAP/AD, make this distribution appropriate for implementing a multi-tenant Big Data Stack.
- **Well documented and community based platform**, being fostered by its open source nature, Hortonworks HDP has a wide community formed by support forums, blogs, manuals and a complete official documentation provided by Hortonworks.
- **Integration with other products provided by Hortonworks and third party software** thanks to Ambari Management Packs, which allow integrating software not included by default in HDP, i.e. Hortonworks HDF (Hortonworks Data Flow) [17].
The usage of Hortonworks HDP allows deploying a secure and multi-tenant Big Data Stack, allowing the fair sharing of different resources provided by the stack, the definition of quotas, access rules and the monitoring of almost all the activity of users inside the system.

2.4 Data Platform architecture

This section describes the different components deployed within Hortonworks HDP for completing the Big Data Stack. In addition to the components and tools offered to the sub-grantees, this section also describes the tools used for the internal management and monitoring of the cluster. Furthermore, the Big Data stack could offer more tools and services in later iterations of the incubation process, responding to sub-grantees’ demands.

2.4.1 Management and monitoring

For the management and monitoring of the cluster, three tools will be used: Apache Ambari, Apache Ambari Metrics and Apache Ambari Log Search.

Apache Ambari [6] is a management dashboard that eases the provisioning, managing and monitoring of a Hadoop cluster. Ambari allows adding or decommissioning nodes in the cluster and deploying different services over the cluster through an intuitive step-by-step wizard, handling the management and versioning of the configuration. It provides a dashboard for starting and stopping installed services, and for monitoring the health of those services, allowing defining different alerts for noticing failing services.
Further, Ambari eases the management of YARN computation quotas for each user or service through its **Ambari Views** component.

In addition to the core Ambari system, **Ambari Metrics** provides a built-in metrics collection for Ambari. It collects system-level and Hadoop-level metrics for each host on the cluster and visualizes them through the Ambari dashboard. Figure 6 shows a screenshot of the view showed by this management tool.

![Figure 6: Ambari Log Search](image)

On the other hand, **Ambari Log Search**, whose interface is shown in Figure 6, aggregates logs from different components deployed within Ambari, enabling to compare logs from different components at a concrete time window for analysing possible malfunctions through its web interface.

### 2.4.2 Data processing

This section describes the different tools offered to the sub-grantees for data storing and processing. The set of tools included in EDI’s Big Data Stack has been selected in order to cover most of the needs that sub-grantees could have during the Experiment and Evolve phases. This initial setup could change during the incubation process if the sub-grantees express different needs, using the experience gained during the first incubation process for the setup of the Big Data Stack at later iterations of the incubation process.

As Figure 8 shows, data storage leverages on **Hadoop Distributed File System (HDFS)** [7]. HDFS is a distributed file system designed to be used in conjunction with MapReduce jobs. HDFS is highly fault-tolerant and it is designed to be deployed on low-cost hardware. A HDFS deployment is formed by a **Namenode**, which maintains the file-tree and establishes in which **Datanode** the data blocks are stored. Each Datanode is in charge of storing the data, which is usually replicated by a factor of three.
YARN (Yet Another Resource Negotiator) [8] is deployed on top of the distributed file system delivered by HDFS. YARN provides resource management and a central platform in which other services can execute their operations. A YARN deployment is composed by a ResourceManager, which allocates resources in

---

different *nodemanagers*. Usually, a worker node on a Hadoop deployment has the *HDFS Datanode* and YARN *Nodemanager* roles assigned.

On top of YARN, different services are offered to the sub-grantees. *MapReduce* [9] is the original framework for processing large amount of data stored at HDFS. MapReduce splits data into key-value pairs that are processed in parallel, taking advantage of the aggregation of the processing capacity of all worker nodes.

YARN opened Hadoop to other data processing engines, being *Spark* [10] one of those. Spark is an in-memory data processing engine for executing streaming, machine learning or SQL workloads that require fast iterative access to datasets. Spark consists of Spark Core, a set of libraries (MLlib for machine learning, Spark SQL, Spark Streaming and GraphX for graph processing) and a set of APIs for developing applications in different languages like Scala, Java, Python and R.

Another processing engine running over YARN is *Hive* [11]. Hive allows executing SQL queries over petabytes of data in Hadoop. In addition, Hive allows creating databases and tables over different types of data stored in HDFS like CSV, Sequence or JSON files. On the other hand, *Pig* [12] is a scripting platform that eases the processing of large datasets, implementing multiple typical MapReduce operations, avoiding user coding them.

*HBase* [13] is a NoSQL database engine running on top of YARN and HDFS. HBase is designed for handling billions of rows and millions of columns. An HBase deployment consists on a single HBase Master and a set of *RegionServers* managing data regions stored on HDFS.

*Kafka* [14] is a scalable, fault-tolerant messaging system used to replace traditional message brokers like JMS and AMQP because of higher throughput, reliability and replication. In the same way of those traditional systems, Kafka relies on message producers, message consumers and topics.

For orchestrating those services, two frameworks are offered by EDI’s Big Data Stack. The first of them, *Oozie* [15], provides job scheduling over the cluster, allowing launching jobs from different nature thanks to a previously designed workflow with the shape of a direct acyclic graph. On the other hand, *NiFi* [16] automates the movement of data between different systems of the cluster, and even outside the cluster. In a similar way to Oozie, NiFi allows creating a conditional workflow for processing data through different systems. Although NiFi is not provided by default by Hortonworks HDP, it is accessible thanks to the Hortonworks HDF and HDP integration.

### 2.4.3 Security

Due to the shared nature of the computation environment, in which different sub-grantees will share storage and computation resources and, the requirements that arise from the management, storage and processing of some sensitive data, security is a key asset for the Big Data Stack. The secure access to every resource in the cluster must be guaranteed, including files stored in HDFS, Hive and HBase databases, Kafka topics, and NiFi workflows. For achieving those security challenges, a combination of Kerberos, OpenLDAP and Apache Ranger is used in the Big Data Stack.

*Kerberos* [18] is an authentication protocol for client/server applications by using secret-key cryptography. Ambari uses Kerberos to avoid identity impersonation inside de cluster, avoiding non-
authenticted users accessing to its resources. As later explained in section 2.5, a client must authenticate itself using the *kinit* command and insert her Kerberos principal (*username@realm*) and password. As Kerberos is limited to the command line interface (CLI), an **OpenLDAP** [19] service is used as Kerberos backend for storing user credentials. As Ambari allows synchronizing its users with the ones at an LDAP/AD directory, a user can access to both CLI tools and web-based tools with the same username and password.

For managing authorization, the platform provides **Apache Ranger** [20]. Apache Ranger is centralized security framework that allows establishing fine-grained authorization policies for multiple components from Hadoop ecosystem. Ranger allows granting access to different user or groups of users, at different permission levels like read, write, update, create, drop, alter, index, and other operations, depending on the component. For some components, like Hive, Apache Ranger allows defining policies even at column level. Figure 9 shows a screenshot of the auditing dashboard of Apache Ranger. In addition, Ranger provides a dashboard in which all accesses to different resources in the cluster can be monitoring, ensuring that only allowed sub-grantees are accessing to their corresponding challenge’s dataset.

For the Big Data Stack, the following components will be secured using the mechanisms provided by Apache Ranger:

- Access to HDFS files and directories.
- Access to YARN queues, Hive and HBase databases.
- Kafka topics
- NiFi workflows.
2.5 Client tool

Because of the multi-tenant and shared nature of the Big Data Stack, the usage of an edge node has been discarded. An edge node is a node in the cluster to which users connect through SSH for launching their MapReduce jobs, upload files to HDFS, and so on. Enabling an edge node requires large administration efforts for monitoring how users are using this node, as they could use this node for launching other apps and processes not related with services offered by the Big Data Stack, consuming too much memory or computation resources.

As sub-grantees can launch their own applications at the Application Container Infrastructure described in section 3, they will connect remotely to services offered to the Big Data Stack from their own machines or personal computers. In order to allow this direct connection a Docker [21] image has been prepared, in which all the clients for connecting to services offered are installed, in addition to their required configurations. Sub-grantees can execute this image in their own computers and start working with the Big Data Stack, without wasting any time installing or configuring clients for connecting to different services offered by the Big Data Stack.

The provided Docker image allows authenticating users using Kerberos, operating files and directories at HDFS, submitting MapReduce, Spark, and Pig jobs, querying Hive and HBase databases and managing Kafka topics. This image will be fully documented including examples on how to interact with different services.

In addition to this Docker image, some Ambari Views are enabled for allowing users interacting with the cluster: WebHDFS, a web based HDFS navigator, shown in Figure 10; Hive2 view, a web based dashboard for launching and saving Hive queries; Pig view, a view for executing Pig scripts; and Oozie workflow manager, a dashboard for creating and executing Oozie workflows.

![Figure 10: WebHDFS from Ambari Views](image-url)
2.6 Hardware requirements

This section exposes the minimum hardware requirements for running the Big Data Stack. As Hadoop is designed for running in commodity hardware, no especial hardware needs are required. Furthermore, thanks to Ambari, a new worker node can be added into the cluster easily, allowing adapting the capabilities of the cluster on demand. Figure 11 shows the minimum hardware architecture needed for deploying the Big Data Stack.

One management server, one master server and at least three worker servers conform this basic architecture:

- The **management server** hosts all tools and services needed for managing and operating the cluster. Among those tools, we can find Apache Ambari and its related components, authentication and authorization functionalities like Kerberos, OpenLDAP and Apache Ranger, and other tools for supporting some deployed services like Ambari Infra or NiFi Certificate Authority. This server must have at least 16GB of memory.

- The **master server** hosts all the master processes of deployed applications and services. Those services are HDFS Namenode and Secondary Namenode, YARN Resourcemanager and App Timeline Server, HBase master, Hive Server2, Hive WebHCat Server, Oozie Server, Spark History Server and NiFi. In addition, it hosts Tez client (required by Hive and Pig), a Kafka broker (although multiple Kafka brokers could be deployed in a cluster) and Zookeeper (which will be replicated at different worker nodes). In order to perform all these tasks, the master server must have at least 64 GB of memory.

- **Worker nodes** will be the ones storing the data and executing the massive processing of data. For doing that, they will host the HDFS Datanode, YARN Nodemanager and HBase RegionServer. In addition, they will host different clients for executing different applications over YARN, like MapReduce, Pig, Hive, Oozie and Spark. They will host an instance of Zookeeper, enabling the replication of this service. At least three worker nodes must be deployed, as this is the recommended minimum replication factor required by HDFS and Zookeeper. As Hadoop scales horizontally, more worker nodes will be added depending on the final needs of the Big Data Stack.
The specific requirements will be finally known at the Explore phase, when sub-grantees will start to use the Big Data Stack and the volume of the datasets associated to their challenges will be measured. However, it is recommended that those worker nodes have at least 64 GB of EEC memory.

As Figure 11 shows, tools provided by Hortonworks HDP and those ones installed specifically for EDI’s Big Data Stack (Kerberos and OpenLDAP) will share the computation environment.

### 3. Application Infrastructure

The computation services provided by the Big Data Infrastructure are sometimes not enough to enable the sub-grantees to provide a complete solution. In many cases, specific applications to process the data in a customized way should be added, for example, to expose interfaces, temporarily store data or include specific APIs. The Application Infrastructure is freely available for these purposes starting from the Explore phase of the three planned incubation interactions. In particular, starting from that phase, the selected sub-grantees can use a set of assigned virtual machines with pre-determined features in term of processing power and memory. These features could increase in the Experiment phase, since it is supposed that the Explore phase could include some basic tests, while the Experiment phase should be the period in which the applications will be developed and deployed.

The use of the Application Infrastructure is not mandatory and, if all the legal constraints related to data security and confidentiality are fulfilled, each participant can use his/her own infrastructure.

This section will describe the requirements that the infrastructure meets and the motivations that led the consortium to select FIWARE as the platform on which the infrastructure has been built.

#### 3.1 Requirements

The Application Infrastructure tries to meet the following requirements:

- **Virtual Machines provisioning.** In order to host the developed applications and, if required, the application containers used to deploy them: a private network must be available to connect the internal services that interoperate to provide the final application.

- **Public IP addresses.** At least a public IP address for each sub-grantees must be provided in order to expose the deployed application on the internet: the possibility to have more than a single public address will depend on the availabilities of the infrastructure provider.

- **Possibility to deploy different services supporting the final application.** In particular, it could be necessary to deploy local databases, brokers and other services interoperating with the application container.

- **Data and application security.** Possibility and easiness to apply the usage terms established by the data providers and the legislations about data and application security.

The identified requirements represent the minimum set that a candidate Application Infrastructure must fulfill in order to be used for supporting the incubation interactions. At time of writing this document, no final decision have been taken about the dimension of the infrastructure. In particular, any limitations on
the maximum number of virtual machines instances or about the resources assigned to each participant will be decided according with the requirements of the published challenges and the budget of the project and could vary among the different interactions.

Another important point, strictly connected with the last requirement of the presented list, concerns the secure connection with the Big Data Infrastructure. Since data security and the respect of all the legal terms must be guarantee in all the processes concerning data, also the communication between the two infrastructures, that will heavily interact to get data and present the results, must be secure.

3.2 FI-WARE

FI-WARE Platform [22] is an *innovation ecosystem* that provides cloud services, including virtual machines and storage, along with a set of functionalities based on public specifications, called *Generic Enablers* (GEs). FI-WARE meets all the requirements presented in the previous section and, furthermore, its Generic Enablers could be useful to support the applications developed by the sub-grantees.

In more detail, FI-WARE cloud infrastructure is based on *Open Stack* and includes the following services:

- **IaaS Generic Enabler**, providing the cloud infrastructure, aimed at provisioning and management of compute, storage and network resources
- **Cloud Object Storage Service**, based on *OpenStack Swift*, providing a scalable, resilient and efficient storage facility
- **Cloud Application Management Service**, based on *OpenStack Murano*, providing the ability to manage provisioning and configuration of complex applications, including basic resources as well as software configuration within the VMs/containers
- **Cloud Policy Service**, providing the ability to define rules and apply actions in response to certain events associated with cloud resources and their state
- **Cloud Monitoring Service**, providing the ability to collect and distribute resource metrics associated with VMs or hosts.

Generally speaking, each well-defined functionality (also, for instance, the IaaS) is defined as an *Enabler*. If the specifications of an Enabler are *generic* enough to be used in different contexts, it is defined *Generic Enabler*. In addition, it is possible to define specifications intended only for certain contexts, the known as *Domain Specific Enablers*.

For each Enabler, at least one reference implementation is proposed: the implementation of an Enabler must expose **at least** all the functionalities described in the specification. This means that a Generic Enabler implementation may also expose **more** functionalities than the specified ones. For this reason, some GEs have more than one implementation and it is always possible to use the public specification to propose new implementations of any Enabler.

The Generic Enablers are listed in a catalogue, shown in Figure 12, and are categorized on different technological fields, such as *Authentication and Authorization, Message Delivery* and *Internet of Things*. Examples of Generic Enablers are *KeyRock* (Identity and Access Management), *Wilma* (Authorization), *Orion Context Broker* (Publish-Subscribe).
The Generic Enablers can be used by the sub-grantees to integrate pre-implemented functionalities (e.g. Authentication) to their customized software.

### 3.2.1 FIWARE lab

**FIWARE Lab** ([https://cloud.lab.fiware.org](https://cloud.lab.fiware.org)) is a working instance of FIWARE Platform available for experimentations: it enables to build virtual infrastructures, deploy applications and use the Generic Enablers. FIWARE lab Accounts can be categorized into two types: **Trial** and **Community**. Trial accounts provide limited resources for a reduced period. A trial account provides the following default quotas:

- **14 days** of duration
- **one** public IP address
- **two** virtual machines instances, each of which has
  - **two** virtual CPUs
  - **40 GB** of hard disk space
  - **4096 Mb** of RAM.

All the participants selected for the **Explore** phase will be provided with a Trial account. EDI team will discuss with FIWARE community to extend the 14 days limit to at least 9 **weeks** (one week more than the duration of the Explore phase). These accounts will be useful to start the first tests and to learn how to use the services provided by the infrastructure that will be extensively adopted in the **Experiment** phase.
The accounts of the participants selected for the Experiment phase will be upgraded to Community accounts. Community accounts do not have any explicit limitation in term of duration and quotas; according to the potential requirements of the challenges and the capabilities, which FI-WARE community will be able to provide for them, some limitations may be introduced and communicated to the participants before the Experiment phase. Figure 13 shows a capture of the interface provided by FI-WARE Lab.

**4. Data Catalogue**

The aim of the Data Catalogue is to allow potential participants to download, when the Open Call is publicly opened, the data samples associated with the challenges registered in the project main page ([https://edincubator.eu](https://edincubator.eu)). Using the Data Catalogue, participants can browse and download the available datasets and view information about their structure. The Data Catalogue only contains the sample data associated with each challenge, and its related information, while the full description of the challenge is contained in the project’s main page. The EDI project’s main page links to the data catalogue from the description of each challenge, being directly accessible at [https://data.edicunbator.eu](https://data.edicunbator.eu).

**4.1 Requirements**

During the analysis of the intended usage of the Data Catalogue, we have identified the following requirements:
- **Registry of data providers.** The Data Catalogue must allow the registration of an entry per data provider. Each entry must contain the name, logo and short description of the data provider; allowing to easily identify them when participants navigate through the data catalogue.

- **Dataset registration.** Each of the data samples must contain a meaningful title, a description that includes its internal structure, associated tags, license, data format (CSV, JSON, etc.) and its classification according to the internal EDI taxonomy for the challenges. In addition, each dataset must be associated with the provider offering the data using a unique ID in the form of `EDI-YEAR-N-XXX`. Tasks developed within WP2 manually generate and assign these IDs to each challenge during the preparation of each Open Call.

- **Browse by data provider, dataset and category.** The Data Catalogue must allow participants to browse by the available data providers, the registered datasets, and the categories identified in the EDI project. Currently, the main categories for the challenges are the following: Energy & Environment, Industry 4.0, Internet & Media, Retail and Smart Cities.

- **Unique URL per dataset.** As the datasets must be linkable from the challenges published in the main page of the project, each of them must have a unique URL.

- **User registration.** Although the navigation of the datasets is public, user registration into the Data Catalogue is mandatory prior downloading any of the available data samples. This registration requirement allows a finer control and further analysis of the downloaded data.

- **GDPR compliance.** As the Data Catalogue stores some personal data during the user registration, it must comply with the GDPR, allowing users to explicitly consent the collection, storage and processing of the data obtained during the registration and the its usage for statistical reporting.

- **Usage analytics.** The Data Catalogue must collect some statistical data that will enable to analyse the usage of the different datasets published in the Data Catalogue.

- **Unified UI:** The design of the user interface must be adapted to follow a design compatible with the one used for main page of the project and the dissemination materials.

### 4.2 Implementation

The Data Catalogue is an adapted CKAN, which is an open source management system for the storage of open data. CKAN already provides most of the required functionality identified for the Data Catalogue; however, we have adapted and configured its basic functionality to fulfil the specific requirements for the project.

In order to use CKAN as the Data Catalogue for the project, we have applied the following adaptations and guidelines:

- The default user interface distributed with CKAN has been adapted to follow the style of the project website, as shown in Figure 14.
We have modified the CKAN default menu, shown in Figure 15, to adapt it to the nomenclature used in the project: datasets, data providers and categories, allowing users to navigate easily through the information and datasets stored in the data catalogue and including the login and registration options.
We have renamed the default “Organizations” category provided by CKAN to “Data Providers” to adapt it to its particular usage in the EDI project. This modification enables to reuse the default functionality of CKAN to navigate through the list of registered data providers, as shown in Figure 16.

The default functionality for “Groups” in CKAN has been adapted to “Categories” and used to classify each dataset according to the type of its associated challenge (see Figure 17).

User registration is required to download the sample datasets from the data catalogue. We have extended the basic registration mechanism provided by CKAN to gather the users’ explicit consent about the processing of their personal data, in accordance with the GDPR, as shown by Figure 18.
4.3 Software contributions

The modifications made to CKAN’s base distribution are available as open source code contributions at https://github.com. Table 5 summarizes the contributions and modifications made to the CKAN default distribution.

<table>
<thead>
<tr>
<th>Software contributions</th>
<th>Description</th>
<th>Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>ckanext-edi_theme</td>
<td>Extension for CKAN providing the EDI’s theme.</td>
<td><a href="https://github.com/edincubator/ckanext-edi_theme">https://github.com/edincubator/ckanext-edi_theme</a></td>
</tr>
<tr>
<td>ckanext-gdpr</td>
<td>A CKAN extension for adapting the registering/login mechanism to the GDPR requirements.</td>
<td><a href="https://github.com/edincubator/ckanext-gdpr">https://github.com/edincubator/ckanext-gdpr</a></td>
</tr>
<tr>
<td>ckanext-protect_resources</td>
<td>Extension for CKAN that enables user registration and login for downloading the data samples.</td>
<td><a href="https://github.com/edincubator/ckanext-protect_resources">https://github.com/edincubator/ckanext-protect_resources</a></td>
</tr>
<tr>
<td>ckanext-ga</td>
<td>An extension for including Google Analytics support into CKAN.</td>
<td><a href="https://github.com/edincubator/ckanext-ga">https://github.com/edincubator/ckanext-ga</a></td>
</tr>
</tbody>
</table>

5. Conclusion

This deliverable has reported about the three main components that are available for the use of applicants and sub-grantees during the different stages of the EDI incubation process: the Big Data Infrastructure, the Application Infrastructure and the Data Catalogue. This document contains the justification for the selection of the chosen tools and the main characteristics of the provided solutions.
References

[22] FI-WARE https://www.fiware.org/, 2018