Abstract: Europe - Brazil Collaboration of BIG Data Scientific Research through Cloud-Centric Applications (EUBra-BIGSEA) is a medium-scale research project funded by the European Commission under the Cooperation Programme, and the Ministry of Science and Technology (MCT) of Brazil in the frame of the third European-Brazilian coordinated call. The document has been produced with the co-funding of the European Commission and the MCT. The purpose of this report on Use Case Requirements is to provide a comprehensive list of functional and non-functional requirements related to the Traffic Recommendation scenario of the project. This report will serve as a guide to all partners, establishing the guidelines and procedures to be followed in the project for the analysis and management of requirements, and the way to validate and verify them.
**Document identifier: EUBRA BIGSEA -WP7-D7.1**

**Deliverable lead**  
UPV

**Related work package**  
WP7

**Author(s)**  
Ignacio Blanquer (UPV), Nádia Puchalski (UFTPR), Nazareno Andrade (UFCG)

**Contributor(s)**  
Ignacio Blanquer (UPV), Nádia Puchalski (UFTPR), Nazareno Andrade (UFCG), Leandro Balby (UFCG), Wagner Meira (UFMG)

**Due date**  
31/03/2016

**Actual submission date**  
02/05/2016

**Reviewed by**  
Daniele Lezzi (BSC), Carlos Eduardo Pires (UFCG), Sandro Fiore (CMCC)

**Approved by**  
PMB

**Start date of Project**  
01/01/2016

**Duration**  
24 months

**Keywords**  
End-user requirements, traffic recommendation, GES3 data

### Versioning and contribution history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Authors</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>09/03/2016</td>
<td>Ignacio Blanquer (UPV), Nádia Puchalski (UFTPR), Nazareno Andrade (UFCG)</td>
<td>Structure, use case description</td>
</tr>
<tr>
<td>0.2</td>
<td>15/03/2016</td>
<td>Nádia Puchalski (UFTPR), Nazareno Andrade (UFCG)</td>
<td>User stories</td>
</tr>
<tr>
<td>0.3</td>
<td>18/03/2016</td>
<td>Ignacio Blanquer (UPV)</td>
<td>Description on the procedures, software architecture</td>
</tr>
<tr>
<td>0.4</td>
<td>12/04/2016</td>
<td>Ignacio Blanquer (UPV)</td>
<td>Integration of data from the QoS architecture and survey of final users’ applications.</td>
</tr>
<tr>
<td>0.5</td>
<td>18/04/2016</td>
<td>Ignacio Blanquer (UPV), Nazareno Andrade (UFCG)</td>
<td>Homogenisation of the information from User Stories</td>
</tr>
<tr>
<td>0.6</td>
<td>22/04/2016</td>
<td>Ignacio Blanquer</td>
<td>Extension of requirements analysis</td>
</tr>
<tr>
<td>0.7</td>
<td>26/04/2016</td>
<td>Ignacio Blanquer</td>
<td>Final version</td>
</tr>
<tr>
<td>0.8</td>
<td>02/05/2016</td>
<td>Ignacio Blanquer</td>
<td>Comments from reviewers</td>
</tr>
</tbody>
</table>

**Copyright notice:** This work is licensed under the Creative Commons CC-BY 4.0 license. To view a copy of this license, visit [https://creativecommons.org/licenses/by/4.0](https://creativecommons.org/licenses/by/4.0).

**Disclaimer:** The content of the document herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the document is believed to be accurate, the author(s) or any other participant in the EUBra-BIGSEA Consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the EUBra-BIGSEA Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the EUBra-BIGSEA Consortium nor any of its...
# TABLE OF CONTENT

**EXECUTIVE SUMMARY** .................................................................................................................. 5

1. **INTRODUCTION** ......................................................................................................................... 6
   1.1. Scope of the Document ................................................................................................................. 6
   1.2. Target Audience ......................................................................................................................... 6
   1.3. Structure ....................................................................................................................................... 6

2. **PROCEDURE FOR THE ANALYSIS OF REQUIREMENTS** ............................................................. 7

3. **EUBRA-BIGSEA USE CASE OVERVIEW** .................................................................................. 10
   3.1. Target community ....................................................................................................................... 10
   3.2. End-user Perspective .................................................................................................................... 10
   3.3. Data Analytics Perspective ......................................................................................................... 11
   3.3.1. Use Case 1 (UC1) - Data Acquisition ...................................................................................... 12
   3.3.2. Use Case 2 (UC2) - Descriptive Models ............................................................................... 15
   3.3.3. Use Case 3 (UC3) - Predictive Models ................................................................................... 16
   3.4. Analysis of Existing Applications for Traffic Recommendation .................................................. 17

4. **EUBRA-BIGSEA INFRASTRUCTURE OVERVIEW** ................................................................. 19

5. **DESCRIPTION OF USER STORIES** .......................................................................................... 22
   5.1. Use Case 1: Data Acquisition .................................................................................................... 22
   5.2. Use Case 2: Descriptive Models .................................................................................................. 22
   5.3. Use Case 3: Predictive Models .................................................................................................... 23

6. **REQUIREMENT ANALYSIS** ...................................................................................................... 24
   6.1. UC1 requirements ....................................................................................................................... 24
   6.1.1. Functional Requirements ......................................................................................................... 24
   6.1.2. Non-Functional requirements .................................................................................................. 25
   6.2. UC2 requirements ....................................................................................................................... 25
   6.2.1. Functional requirements ......................................................................................................... 25
   6.2.2. Non-Functional requirements .................................................................................................. 26
   6.3. UC3 Requirements ..................................................................................................................... 26
   6.3.1. Functional Requirements ......................................................................................................... 27
   6.3.2. Non-Functional requirements .................................................................................................. 27

7. **CONSOLIDATED ANALYSIS AND ABSTRACTION OF GENERAL REQUIREMENTS** ..................... 28
   7.1. General requirements of data access .......................................................................................... 28
   7.2. General requirements of execution ............................................................................................ 29
   7.3. General requirements for security and logging .......................................................................... 30
   7.4. Analysis and prioritisation ......................................................................................................... 31
   7.5. Final Considerations ................................................................................................................... 32
   7.5.1. Software Licenses ..................................................................................................................... 32
   7.5.2. Version Control ........................................................................................................................ 32

8. **CONCLUSIONS** ............................................................................................................................ 33

9. **ANNEXES** .................................................................................................................................... 34
   9.2. Tables of requirements ............................................................................................................... 36
   9.2.1. Use Case Requirements .......................................................................................................... 36
   9.2.2. Technology Requirements ...................................................................................................... 37
LIST OF TABLES

Table 1 – Phases of the analysis of use case requirements ................................................................. 8
Table 2 – Requirements analysis checklist .......................................................................................... 9
Table 3 – Sources of data ...................................................................................................................... 13
Table 4 – Comparison of Traffic Recommendation Apps ....................................................................... 18
Table 5 – Information from users for Use Case 1 ................................................................................. 24

LIST OF FIGURES

Figure 1 – Projection of the bus stops (blue) and bus stations (red) of Curitiba ...................................... 14
Figure 2 – Sample usage of the transportation network of Curitiba ....................................................... 14
Figure 3 – High-level view of the EUBrA-BIGSEA Architecture ......................................................... 19
Figure 4 – Relations among WP7 and the other technical WPs.............................................................. 20
Figure 5 – Detailed view of the software architecture .......................................................................... 20
Figure 6 – Interactions among requirements .......................................................................................... 31
EXECUTIVE SUMMARY

EUBra-BIGSEA project aims at developing a set of cloud services empowering Big Data analytics to ease the development of massive data processing applications. EUBra-BIGSEA will develop models, predictive and reactive cloud infrastructure QoS techniques, efficient and scalable Big Data operators and a privacy and quality analysis framework, exposed to several programming environments. EUBra-BIGSEA aims at covering general requirements of multiple application areas, although it will showcase in the treatment of massive connected society information, and particularly in traffic recommendation.

The project starts with the analysis of the use case scenario that will be used for demonstration, but considering those requirements in broader way. EUBra-BIGSEA is an API-centric project which main objective is to improve efficiency and throughput of data scientists and data curators.

Therefore, the requirement analysis covers the general problem of traffic data analysis from both the perspective of the data processors and the final end-users, as the requirements of the final end-users also serves to understand the global problem.

The requirements analysis process has gone through four major steps: first, a questionnaire was distributed among data analysis developers and included in this document as an annex to identify the requirements and data sources. Second, we performed an analysis of the general project scenario, which is split into three use cases: Data Acquisition, Descriptive Models and Predictive Models. Third, this analysis has led to the identification of User Stories - individual descriptions of whole interactions of the users with the system per use case. Eleven User stories have been identified covering features such as data integration, data update, data filtering and searching, metadata management, trajectory identification, clustering, model fitting, and model projection. Then, 25 functional and non-functional requirements have been identified from these use stories per Use Case. Finally, from the analysis of those requirements, 18 technical requirements have been identified.

The requirements from the Use Cases serve to guide the implementation of the application and services that will consume the Big Data services of EUBra-BIGSEA platform. These requirements are also of interest for the rest of the platform as high-level needs that require to be addressed through the platform functionality. The 18 technical requirements are directly related to the EUBra-BIGSEA platform developers, and address functionalities such as: integration of external data sources, automatic synchronization, storage of processing products, batch submission, Bag of Tasks, QoS-bounded submission, self-adaptive elasticity, short-jobs support, privacy annotation, authentication, ACLs, group access policies, and data privacy protection.
1. INTRODUCTION

1.1. Scope of the Document

This document describes the user requirements as well as the procedure to identify them, analyse them and prioritise for their implementation. The document also describes the user communities, the data sources and similar applications in the field.

1.2. Target Audience

The document is mainly intended for internal use, although it is publicly released.

At internal level, WP7 members will find in the document the requirements that will be addressed by EUBra-BIGSEA, as well as the expected results at the level of the software components. Technical developers from WP3, WP4, WP5, and WP6 will find the requirements they have to address in their developments. The project success will be also measured in the degree of fulfilment of such requirements.

At external level, Data Scientists (referred here as developers of data-analysis-intensive applications) could evaluate if the requirements addressed are similar to those they have, considering the possibility of adopting the technology. Application developers could also understand the kind of problems that could be addressed using the EUBra-BIGSEA components.

The information of this document will be periodically updated in the internal wiki, and the final outcome of the implementation of such requirements will be analysed in deliverable D7.6 (Validation of requirements), by the end of the project.

1.3. Structure

The document is structured into 8 parts. Section 2 describes the requirement analysis process. Section 3 describes the use cases from the point of view of the target community, the end-user perspective, and the data analytics. Section 4 introduces the software architecture of EUBra-BIGSEA, whereas section 5 describes the user stories for the three use cases. From those user stories, section 6 describes the user requirements. Finally, a consolidated analysis is presented in section 7. Section 8 draws the final conclusions.
2. **PROCEDURE FOR THE ANALYSIS OF REQUIREMENTS**

The analysis of requirements starts understanding the particularities of the user scenario from both the application developers and the end-users. This description is compiled in this section and it has been performed by the Use Case experts. Those experts elaborated a questionnaire that has been circulated among selected representative members, addressing the whole lifecycle of the use case. From the information in the questionnaire, user stories have been identified and requirements from them.

The action plan includes activities that would be completed in 4 phases (elicitation, analysis, specification and validation), shown in Table 1, along the project life cycle.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Requires</th>
<th>Produces</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Elicitation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The expected user classes (actors) and other stakeholders are identified from the scenarios described by users</td>
<td>Report on use case communities (Section 2.1)</td>
<td></td>
</tr>
<tr>
<td>Users and developers reach a common understanding of the tasks and goals of the use case actors</td>
<td>Report on use case communities (Section 2.1), KOM Use case description</td>
<td>Discussion on Research and Technical Board (RTB) telcos</td>
</tr>
<tr>
<td>Users describe the environment in which they usually perform their work</td>
<td>Inventory of applications and data sources (Sections 3.3.1, 3.4)</td>
<td></td>
</tr>
<tr>
<td>User’s quality expectations are discussed</td>
<td>Inventory of applications and data sources (Sections 3.3-3.4), KOM Use case description</td>
<td>Discussion on RTB telcos</td>
</tr>
<tr>
<td><strong>Analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>High-level requirements are described in the form of User Stories</td>
<td>Initial list of Success Stories (section 5)</td>
<td></td>
</tr>
<tr>
<td>New functional requirements are generated from the analysis of the information</td>
<td>Initial list of Success Stories (section 5)</td>
<td>A complete list of requirements is defined (section 6)</td>
</tr>
<tr>
<td>Requirements are allocated to infrastructure functions and components</td>
<td>A complete list of requirements (section 6)</td>
<td>WP3, WP4, WP5, and WP6 advance in the definition of the software architecture</td>
</tr>
<tr>
<td>Implementation priorities are identified</td>
<td>As above</td>
<td>Requirements dependencies (section 7.4)</td>
</tr>
</tbody>
</table>
### Specification

- Requirements are transformed into technical documents and diagrams.
- Final list of use case requirements (Sections 6, 7)

### Validation

<table>
<thead>
<tr>
<th>Action</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documented requirements are reviewed and accepted.</td>
<td>D7.1 is approved</td>
</tr>
<tr>
<td>Early prototypes are developed and validated in the infrastructure.</td>
<td>Integration tests are successfully passed</td>
</tr>
<tr>
<td>Users confirm that the use case implementation meets their needs.</td>
<td>User acceptance tests are successfully passed</td>
</tr>
</tbody>
</table>

Table 1 – Phases of the analysis of use case requirements

Table 2 presents a checklist of 12 tasks to be completed in order to fully address the requirements of the use cases. This methodology was adapted from the approach described in the literature¹. The rest of this report covers the analysis procedure to the phase of specification. Validation will be documented in the subsequent WP7 deliverables D7.2-5 and D7.6. For the elicitation phase, a questionnaire was defined and filled-in by the WP7 partners. The questionnaire is described in the annex of this document.

<table>
<thead>
<tr>
<th>№</th>
<th>Action</th>
<th>Action</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create the questionnaire</td>
<td>PM1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Discuss on a F2F meeting</td>
<td>PM2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Create a collaborative document with the initial list of requirements and user stories</td>
<td>PM4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Verify that all the user classes are represented and requirements are approved</td>
<td>PM4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Verify the level of detail of the requirements</td>
<td>PM6</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Add priorities to requirements</td>
<td>PM6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Allocate requirements to infrastructure</td>
<td>PM6</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Model the requirements</td>
<td>PM6-9</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Develop early prototypes, include APIs and user interfaces</td>
<td>PM9-13</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Requirement</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Evolve the infrastructure to support the use cases</td>
<td>PM9-13</td>
</tr>
<tr>
<td>11</td>
<td>Develop validation tests</td>
<td>PM9-16</td>
</tr>
<tr>
<td>12</td>
<td>Validate use case requirements</td>
<td>PM22-24</td>
</tr>
</tbody>
</table>

Table 2 – Requirements analysis checklist
3. **EUBra-BIGSEA Use Case Overview**

### 3.1. Target community

The potential users of the EUBra-BIGSEA platform include programmers, administrators, and public stakeholders, such as government departments, public authorities, and citizens. A part of this community (such as programmers and administrators) has a vast experience in the use of online databases and research tools, which is an added value for the project, since this experience can be helpful to model, design and validate the project use case. The other part of the community (such as public stakeholders) has experience with part of the use cases explored. As an example, consider that some stakeholders have experience on integrating the online databases, but did not work with prediction and description of data (along with the impacted tools and models).

Finally, the target communities for the final products to be developed in the frame of WP7 include not only data scientists, but also municipality stakeholders and citizens.

### 3.2. End-user Perspective

Our use case application aims to improve urban mobility from a human perspective. For that, the application will bring together navigation algorithms already used in transportation systems, novel data mining models to understand and predict origin-destination characteristics, and visualization, focusing on the need to create cities and transportation systems for people. This focus on people is grounded on a perception that cities should be not only efficient, but also pleasant for its dwellers. It follows that a transportation system should help its users not only to find the fastest routes from A to B, but also the routes that are most interesting, calm, predictable, or lively. For example, a route that leads a traveler to walk through a park and to choose a bus route that will face less traffic, while taking little extra time can be preferred by many people compared to a faster route that is both traffic intense and less pleasant to walk on the way to the bus.

Understanding and predicting such multifaceted attributes for a route is a nontrivial task. Our approach is to leverage Big Data and four data sources to create data mining models that fulfil this goal. First, detailed stationary georeferenced data about city will be used to integrate the other data sources. Next, three Big Data sources are used for the dynamic aspects of the system: the GPS location of all buses and user cards in the city, data output produced by fine-grained weather models, and historical and real-time data posted in social media associated with the city and its locations. This heterogeneous and large volume of data will be integrated and continuously processed to detect patterns, trends and outliers in the behaviour of the transportation system. Given our focus on cities for people, the state of the mobility in the city will be approached not only investigating speed, vehicle flux, traffic disruptions, main origin-destination routes for citizens (according to each day, time, and region), but also the sentiments and topics historically or recently associated with places, stress caused by traffic, landmarks, presence of green, weather conditions and the effects of all of these on the perception one has from a trip in the city.

This use case primarily targets two groups of end-users: citizens and urban planners associated with the municipality. From a citizen perspective, mobile and web interfaces will allow the citizen to query for the state of the route options available for a given trip. Once the citizen manifests a travel need, the system will
provide multiple route options that maximize different criteria in addition to travel time, such as likely stress, pleasantness, interestingness and liveliness of the routes (according to parameters such as day of the week, hour and location). Such route recommendations will be built processing both long-term historical data to identify trends, and short-term data focusing on recent and relevant events. Furthermore, recommendations will be provided through state-of-the-art and novel predictive models from Data Science.

From the urban planning perspective, our goal is to provide a descriptive view on the state of the mobility in the city as a whole, identifying its status, trends and the impacts of relevant events. Contrasting with the citizen view, the goal of the application in this perspective is to use data mining descriptive models and to allow urban planners to interactively explore it, as well as to provide notifications and alarms for these stakeholders. This view will inform urban planners in both operating and planning transportation systems in a more effective way.

The project will ensure dealing with the mentioned characteristics, while respecting privacy, quality concerns, raises ethical, political, social, scientific, technological and engineering challenges. From the computational perspective, the implementation of the two described views of the use case will require non-trivial resources and services from a cloud platform. Large volumes of data will be continuously acquired, processed and stored. At the same time, this data needs to be pre-processed in different manners to extract features relevant for the higher-level functionalities and mining. In addition to this continuous processing, models must be created periodically so as to provide description and predictions about the transportation system that meet recency requirements. The creation of such models is compute-intensive, and providing such creation with time and budget constraints will only be possible through the creation of novel mechanisms in the EUBra-BIGSEA project. This is particularly challenging given the likelihood that the execution time of data mining processing will be affected by events external to the computing system, such as traffic incidents or football matches, for example.

Finally, we aim to investigate concepts such as cloud, analytics, mobile, social, and security from what is recently called Smart Cities. The urgency around related challenges (traffic congestion, waste management, etc.) along with sustainable and liveable city, triggers the society to find smarter ways to manage resources. Initiatives toward this direction include open data and civic hackathons. Mobility, from the city perspective of open data, data integration, cloud, and data mining, besides being signed as challenges in computer science, can change the outcomes of economic investments and employment opportunities.

### 3.3. Data Analytics Perspective

The data analytics requires three different actions or Use Cases. First, data must be retrieved from the appropriate sources. This means identifying the proper sources, the access protocols, the data formats and the data features. Second, descriptive models are implemented to infer higher-level information, such as trajectories, transportation patterns, outliers, and hotspots. Descriptive models will also be used for the Data Quality as a Service activity. Finally, predictive models project new conditions on the models of the data to identify the outcome of future or hypothetical situations. In this sense, we define three use cases:

- Use Case 1 (UC1), devoted to the acquisition of data in the system using different data sources.
- Use Case 2 (UC2), related to the processing of the data to extract the historical knowledge from the data.

---

Use Case 3 (UC3), aims at creating new knowledge by projecting existing models in the future or under different conditions.

These three use cases are briefly described below. Details on the user stories and the requirements are provided in the next sections.

3.3.1. Use Case 1 (UC1) - Data Acquisition

EUBra-BIGSEA must deal with multiple sources and types of data. These data sources are related to urban traffic and cover static and dynamic information. Despite that they are related to the city of Curitiba, where the pilot case is being constructed, the EUBra-BIGSEA framework should be extensible to other scenarios.

Four main data types are identified:

- Stationary data. This data is related to long-living data that describes the topology of the traffic network of the city, the street map, relevant city spots (such as bus stops and bus terminals) and other geographic information that is relevant to understand the location of the components present within the urban mobility scenario (illustrated in Figure 1).
- Dynamic spatial data. This data contains georeferenced information of the vehicles and users, valid for a specific point in time.
- Environmental data. This data provides information about the environmental conditions and the weather forecasts that are also relevant for understanding citizens’ mobility.
- Social network data. Refers to immediately consumed data that can provide information about sentiments and unpredictable events.

In summary, these data is related to three entities:

- Place. It consists on a georeferenced point or region, including the weather, stationary data, people and vehicles related to it (who have been there and when), as well as crowdsourced info.
- Vehicle. It is characterized by its type (e.g. bus, tram), trajectories, and people who could use it.
- Person. Basically a user of the transportation or social network data source, with the associated information.

Regarding the data types, we can mention: (i) Non-structured (user-generated content), such as tweets, text from news, etc.; (ii) Tabular; (iii) Spatial, according to different geographic formats and coordinates; (iv) Temporal, such as different environmental data for the week, month, etc.; (v) Relational, within a spatial database; and (vi) Hybrid data that may embed any combination of the above.

Example of Stationary GIS Data (Shapefiles) can be found at http://ippuc.org.br/geodownloads/geo.htm. A sample entry of Streaming Data (Web-based service, with data at every 5 minutes) is:

```
{"CODLINHA":"515","NOMELINHA":"IGUAPEII","CODVEICULO":"EA195","NUMEROCARTAO":"0002480682","DATAUTILIZACAO":"07/10/1518:05:50,000000"
}
```

```
{"CODLINHA":"515","NOMELINHA":"IGUAPEII","CODVEICULO":"EA195","NUMEROCARTAO":"0003415843","DATAUTILIZACAO":"07/10/1517:36:44,000000"
}
```

According to the information retrieved through the questionnaire, the characteristics of the information relevant to this use case in the project are summarized in Table 3.
The data created from the four different sources might present discrepancies in terms of resolution, orientation and displacements. In particular, spatial data quality might be explored through:

- **Data Precision**: Precision can be termed as the degree of details that are displayed by the different sources.
- **Data Accuracy**: This can be termed as the discrepancy between the actual attributes value and coded attribute value.
- **Data Consistency**: Data consistency can be termed as the absence of conflicts in a particular database.

Several preprocessing steps will be applied within this phase. From the spatial data perspective, we can mention a standard coordinate system and projection, possible format changes, location filtering, (vectorial or raster) and any other process that our data requires in order to be effectively used for the type of analysis that we want to perform. Other preprocessing steps (e.g. deduplication, anonymisation, filtering, etc.).
In order to understand the volume of the information from the network of Curitiba, main target of this use case, Figure 1 shows the projection of the bus stops and bus stations of the transportation network, and Figure 2 shows the transportation dynamic data obtained from a Sunday by hours. This data corresponds to 339 lines and 1762 vehicles.
3.3.2. Use Case 2 (UC2) - Descriptive Models

A fundamental abstraction of the descriptive models are trajectories, that is, the path traversed by each end user while using public transportation. Trajectories comprise not only dynamic spatial data, but also the other types of data that enrich the trajectory information. Notice that building the trajectories is a challenge by itself, since matching the various types of data to a specific end user trajectory may be very tricky and demand advanced and complex techniques. Building trajectories may be seen as a clustering task, although it fits better as a record linkage or entity resolution task.

Once we have trajectories, we may perform three descriptive tasks. The first task is characterization, which analyses the trajectories in detail, deriving aggregated and non-aggregated statistics, where the latter consider the characteristics of several entities. The characterization also aims to identify characteristics probability distributions for the various statistics, as a strategy to extrapolate the findings and the underlying phenomena.

The second task is to determine correlations among trajectories, looking for features that characterize them. There are at least two types of correlations: sets and sequences. In both cases, the elements may be any of the data discussed in the previous section. Usually, we look for sets and frequencies that occur above a predefined threshold, which corresponds to the frequent pattern mining problem. The main difference between sets and sequences is that the latter also express precedence relations among elements. Frequent patterns are important to determine the common behaviours, which may be often surprising.

The third task is to cluster similar trajectories, which demands the answer of an intriguing question: what makes two trajectories similar? In our use case, we may exploit several dimensions: location, time, weather, sentiment/opinion, among others. Determining the similarity criterion is a challenge by itself, and may be based on the other two descriptive tasks.

All tasks must be implemented in conjunction with visualizations that allow a user to interactively explore its results, zooming and filtering data on demand.

The Descriptive Models (DM) service is accessed by the Routes for People application through a RESTful API. In turn, it needs to access services from the EUBra-BIGSEA platform that provides: (i) access to the GS3 data, (ii) the execution of data mining algorithms, and (iii) the storage, update and retrieval of data mining models. All of these services will be accessed through HTTP calls to RESTful APIs.

The data used by the Predictive Model (PM) service to create predictive models will come primarily from the GS3 integrated data service. Raw data from stationary elements of the city, traffic status, environmental predictions and social media posts will be acquired by the data acquisition service. This data will be continuously processed by a module common to both the DM and the PM services to create a second storage consisting the higher-level features of the raw data used to mine the patterns relevant to the Routes for People application. Such features will be accessed to create the descriptive models that are used to answer queries from end-users from the municipality. The descriptive models themselves are also saved and accessed.

The continuous processing of raw data to extract higher-level features is expected to process all data coming in the GS3 service. It is expected that the features extracted from the raw data are on the same order of magnitude as the raw data itself.
The maximum amount of data required by a single execution of the descriptive models creation is in the order of dozens of hundreds of GBs. The average case is likely a few hundred GBs. Model storage and answering end-user queries require access to MBs of data.

The data manipulated by the Descriptive Models Service is a combination of relational data, NoSQL records in JSON format and georeferenced data. A considerable amount of this data models are time series.

3.3.3. Use Case 3 (UC3) - Predictive Models

Predictive models will be based on one or several data mining paradigms. The approach selected will be key for the QoS metrics, which will have to determine the expected resources required by the algorithm to run. The project will explore the following four paradigms:

- Combinatorial models, where the models are selected from a larger search space that must be built, stored and traversed, increasing the storage and processing resources. Output models may be large.
- Probabilistic models, where the aim is to determine and calibrate the parameters of the probabilistic models through an iterative process.
- Algebraic models, which involve solving a Linear Algebra System by means of direct or iterative methods, which are computationally intensive.
- Graph-based models, where isomorphisms and canonicals play a major role, which increases complexity and cost.

In terms of predictive models, we envision at least two tasks to be designed, implemented and assessed: classification and regression. The first task aims to predict which is the route that best matches the user profile and preferences, considering all data we discussed in Section 2.3.1. This recommendation task is generally framed as classification, but with very specific requirements. There are two challenges here: (i) selecting the features to be used; (ii) choosing the technique to be employed, which should be capable to recommend routes that match the user profile and expectations as much as possible.

The second task is regression, which will be used for estimating (i) the duration of a trajectory considering current conditions and historical data and (ii) the level of stress of the users while using a given public transportation line (e.g. by means of sentiment analysis). These tasks are complementary to the classification one, since the estimates may become a selection criterion for the trajectories to be recommended. Again, it is necessary to select the features for the model and choose the model function and its characteristics so that we are able to maximize its accuracy for a wide variety of the scenarios.

Besides the requirements identified for the descriptive models, we identify two extra requirements for the predictive models: first, such models are expected to be periodically created, and although involving non-trivial computation and Big Data processing, must happen timely. Second, after creation, these models must be able to provide answers for end-user queries in near real time.

From a platform perspective, three main types of processing are required: continuous pre-processing of raw data, batch processing for creating predictive models, and online processing for answering prediction queries.

The continuous pre-processing of raw data processes a stream of events resulting from the acquisition of GS3 data. The amount of processing over time is likely variable, given its dependency on the volume of data.
acquired in a given period. This volume, in turn, is a function of the number of events exogenous to the system that are captured by the environmental, traffic and specially in the social media data sources.

For the batch processing necessary to create predictive models, it is expected that the system is able to create a new predictive model two times a day. This creation happens in a batch process and it is important that such models can be created with recent information and are available soon before critical periods, typically the morning, midday and evening rush hours. It follows that the PM service requires from the platform the possibility of scheduling the batch processing of model creation to finish by a certain deadline with some confidence. Naturally, given the possibility of elasticity, a budget must be also specified by the PM service to be used by the platform.

The online processing required to answer a prediction query by an end-user requires a trivial amount of computation from a single VM per user. Similarly to a web information service, this processing has a predictable amount of computation. It is required that the PM service can rely on the platform to answer up to 100 queries per second with a processing time of at most 2s per query.

### 3.4. Analysis of Existing Applications for Traffic Recommendation

Widely used applications for traffic recommendation target mobile applications as the most suitable platform to provide real-time information for the user’s location. Several features are analysed:

- **Data sources.** Applications typically use static information about the transportation routes, stops and schedules and when available, real-time information of the position of the vehicles and expected arrival time based on the route.
- **Real-time reaction.** Applications may get information from traffic news, but only to notify the user. Routes are not recalculated.
- **Coverage.** The cities and countries supported by the application.
- **Push / Pull approach.** All the applications enable defining routine routers for specific time, enabling the application to warn in advance. Some applications enable notifying the user as an emerging message (e.g. notification) the arrival of the vehicle or the stop where the user should leave it.
- **License mode.** Normally the applications follow a free “ad-ware” license mode, with the possibility of removing commercial announcements through a fee.
- **Recommendation.** The application recommends routes from two destinations (in several cases only the current location and a target destination), combining several transportation means. They do not include recommendation based on social networks trends, traffic conditions (although they may show traffic congestion) or weather.

Table 4 shows a comparative of three of the most popular traffic recommendation apps.

<table>
<thead>
<tr>
<th></th>
<th>Moovit</th>
<th>Transit</th>
<th>CityMapper</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notification</strong></td>
<td>Based on predefined destinations and timeslots</td>
<td>Based on predefined destinations and timeslots</td>
<td>Based on predefined destinations and timeslots</td>
</tr>
<tr>
<td><strong>Real time</strong></td>
<td>When available from the</td>
<td>When available from the</td>
<td>When available from the</td>
</tr>
</tbody>
</table>
The services and applications to be developed in the frame of WP7 aim at projecting predictive models on the real-time conditions to provide more accurate route plans and automatic recommendations based on population information. This approach strongly differs from the static routes of current applications. In this case, we expect to have recommendations of routes such as the most pleasant one, given the weather conditions and the path, the safest one, uncrowded lines, etc. In order to reach these recommendations, more complex algorithms are needed, thus reaching higher computational demand and near real-time response. EUBra-BIGSEA aims at providing such QoS-based cloud framework.
4. **EUBra-BIGSEA Infrastructure Overview**

The EUBra-BIGSEA general Infrastructure comprises 4 main blocks:

- **QoS Cloud Infrastructure services**, which integrates the modelling of the workload, the monitoring of the resources, the implementation of vertical and horizontal elasticity and the contextualization.

- **Big Data Analytics services**, which provide operators to process huge datasets and which can be integrated in the programming models. Analytics services are characterized in the QoS cloud infrastructure models of the underlying layer, which automatically (or explicitly driven by the analytics services) will adjust resources to the expected workload and considering its specificities.

- **Programming Models**, which provide a higher-level programmatic framework (Python, Java, Spark) and are also characterized by the models of the infrastructure. The programming models will ease the parallelisation of the applications developed on top of them.

- **Security framework**, which provides the means to annotate data and processing and ensures the proper protection of privacy and security.

On top of those four blocks, applications are developed using the programming models and the data analytics extensions. Application developers are expected to use the programming models and may use other features of underlying layers, such as the user-level QoS metrics.

![Figure 3 - High-level view of the EUBra-BIGSEA Architecture](image)

Figure 3 shows the high-level view of the EUBra-BIGSEA architecture depicting the interactions among the main blocks.

**WP7** is an application-centric work-package that will rely on the APIs and consume the components developed in the frame of the other WPs, as depicted in Figure 4.
Figure 4 – Relations among WP7 and the other technical WPs

Figure 5 shows a more detailed schema of the architecture. More details will be provided in the deliverables D3.1 QoS Monitoring Architecture and D5.1. BIGSEA Software Architecture.

In order to implement the execution life cycle from each one of the user scenario levels, it is necessary to define:

- Application binary and associated dependencies, embedded on a container. The application dependencies can be coded as a dependency file ("a la dockerfile") or directly registering the container on a repository. The basic execution unit is therefore the container.

- QoS Policies. The different applications will be analysed in terms of dependencies, graph execution path, resource demand and performance. Static QoS policies will be defined for each application type, which will define the estimated best resource allocation for each executed case. Real-time QoS policies will also be defined for each algorithm to drive elasticity.
- A specific execution description, programmatically coded or as a job description, to be run on the scheduler of the platform.

Most of this information is inherent to the programming model and part to the specific algorithm to be executed on it. Therefore, the user will not need to provide additional information, but the analysis of the requirements should derive this information or the methods to obtain it.

Additionally, the requirement analysis will provide input to the following components:
- Authentication. The use cases should define whether user identification is needed or not, at which levels will be needed and if there is already an existing mechanism.
- Authorisation. The authorisation on the access to the data could be done at the level of individuals or groups, and the actions that the authorisation could grant can be different. User scenarios have to identify such needs.
- Privacy management. The project does not involve the management of protected personal data, but it has an activity to develop components for this. Therefore, the requirement analysis should identify the information needed to code and protect the level of privacy of the data, either from its acquisition or after its processing. It may happen that raw data will not require data protection, but post-processed data could discover re-identifiable data, so it could not be stored in the same way.
- Data acquisition. Data sources, data formats, data volumes, data acquisition rate, expected storage needs and data validity.
- Programming models. The project will start from the existing expertise of the data analytics groups, who already attain experience on popular execution environments. Application developers and data scientists will express their requirements in the form of programming languages and frameworks.
- Execution patterns. Different use cases identified in this document will have different execution patterns (event-based, bag of task batch, interactive).
- Logging and monitoring. The use cases will also define metrics that will be logged to feedback and adjust the static and real-time policies. Use cases should define which are the metrics they can expose (additionally to the basic resource monitoring, such as CPU, memory and disk usage, or the ones that could be obtained from the scheduler, as the job waiting time) and which are relevant to the use case.

Therefore, the next sections address the previous items to drive the design and development of the architecture to fulfil the use case requirements.
5. **DESCRIPTION OF USER STORIES**

The user stories describe a complete interaction and can be seen as macro-requirements that outline interactions and non-functional requirements. The user stories are defined at the level of the three use cases.

5.1. **Use Case 1: Data Acquisition**

The type of user stories in the data acquisition and integration focus on retrieving the data, the periodicity and mechanism for retrieving the data, the basic filtering of the data, raw and filtered data data visualization. This use case is mainly intended for data curators and data scientists that have to prepare and understand the main features of the data sources. The user stories for data integration include:

US1.1.- A process, that can be invoked manually by a data curator user, retrieves the stationary data from the original sources - including the metadata, transforms it to a common format (geographic reference system, data and time, etc.) and stores it on a proper database system.

US1.2.- A process can be executed to filter the dynamic spatial data according to transportation lines, geographic zones, specific users or data periods.

US1.3.- It should be possible to visualize stationary data on a map view along with the results of a processing.

US1.4.- A user can have customised views, filters and searching macros as well as filter results on a persistent storage.

US1.5.- A user can visualize both stationary data (such as bus stops) with dynamic data, in order to understand if a bus route is regular or not.

5.2. **Use Case 2: Descriptive Models**

The type of user stories in the descriptive models focus on the analysis of trajectories and the associated variables that could affect their distribution (weather, date and time, social networks information, etc.). The Descriptive Models are built as a service (DM service), targeting data scientists on traffic management to discover correlations and build up higher-level services.

US2.1 - As the DM Service, a user can query a storage for trajectory data and its aggregated and disaggregated statistics, which include estimates for probability distributions of variables related to these trajectories, so the data can feed statistical analyses that aim to characterize trajectories. Queries may specify a period of time or geographical region to filter the trajectories.

US2.2 - As the DM Service, a user can query the platform for summaries of frequent patterns of sets and sequences in trajectories, so that the data can feed statistical analyses that aim to characterize correlations in trajectories.

US 2.3 - As the DM Service, the user can request the platform a clustering of the trajectories in the system, specifying the clustering algorithm, its parameters and a similarity function, so that the data can feed statistical analyses that aim to characterize correlations in trajectories.
5.3. **Use Case 3: Predictive Models**

The user stories in the Predictive Models deal with the training of the predictive models on the descriptive data obtained in the previous Use Case. These user stories are apparently more computing-intensive than data-intensive bounded. Prediction will also include projection of models, which is not computing intensive but should work in interactive time.

As final product, we envisage the development of an application of routes for people, whose main goal will be to help citizens to check how mobility conditions are and give hints on how to better reach destinations with respect to multiple criteria, such as time, predicted traffic (and hence stress), pleasantness to walk, sights, and interestingness (taken from social media).

The user stories for the Predictive Models (PM) Service are:

US3.1 - As the PM Service, the user can request the immediate training and storage of a predictive model with the current most recent data in the system and a time window, so that he/she can later access this model to make predictions. The request specifies model type (eg. random forests, recurrent ANN), training procedure (eg. 10-fold cross validation) and data (eg. last month) to be considered.

US3.2 - As the PM Service, the user can request a prediction to be made with a previously trained model, so that the data can serve user-facing applications with accurate and timely predictions. Predictions should run in near real time.

US3.3 - As the Predictive Models Service (PM Service), the user can schedule a periodic training of a predictive model that updates the previously stored model.
6. **Requirement Analysis**

As described in section 2, the procedure will go through the analysis of the User Stories to derive use case requirements that could drive the implementation of the EUBra-BIGSEA services. By the analysis of the User Stories and the data sources, several technical requirements are identified. The requirements have been coded as issues in the redmine project site to follow-on their discussion and progress.

Requirement analysis is first presented by Use Cases. This analysis implies the development of components and APIs on top of the services of EUBra-BIGSEA. Specific requirements for WP3-6 are described in Section 7.

6.1. **UC1 requirements**

The data acquisition will provide the integration of four different data sources, with customised views, filters and searching tools. From the point of view of an end-user, there must be only one entry point to the integrated environment, and inputs and outputs should be consistent across all operations. This imposes several challenges that will be addressed in the following sections. Table 3 shows the information gathered from UC1 users in the elicitation phase.

<table>
<thead>
<tr>
<th>Scope</th>
<th>User-Provided Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Interface</strong></td>
<td>Web-based access.</td>
</tr>
<tr>
<td><strong>Data Access</strong></td>
<td>The environment will use public data sources, and will produce data that must be protected from unauthorized access. It is expected that the data access volume required by a single execution does not exceed few MB, although this value may be exceeded in specific uncommon cases. Standard data formats and access protocols are widely used and there exist open-source software libraries that can be used to access and manipulate the data.</td>
</tr>
<tr>
<td><strong>Execution Model</strong></td>
<td>Experiments are modelled and executed as repeatable workflows. Users select a workflow template and customize input and output parameters.</td>
</tr>
<tr>
<td><strong>Network Traffic</strong></td>
<td>Moderated traffic is expected.</td>
</tr>
<tr>
<td><strong>Security Constraints</strong></td>
<td>User data must be protected from unauthorized access. A user can grant access to other user or group of users.</td>
</tr>
</tbody>
</table>

![Table 5 – Information from users for Use Case 1](image)

### 6.1.1. Functional Requirements

R1.1. The application must integrate the GIS data sources included in the Table 1. The integration of data should be done in a standardized way to facilitate the future integration of any other data source. The information should be updated accordingly. The data integration procedure should be clearly described, as well as the storage architecture required.
R1.2. The application must integrate meteorological/climate data sources as identified in the table 1. This information should be attached to historic records and new (forecast) information should be accessible.

R1.3. Metadata must be included into the application to describe the area covered, the year of acquisition of the data, the type/format of the data and any other technical specification that is necessary.

R1.4. Besides user classes, the service should distinguish the following groups of users with different access and their roles:

a. Administrator or system manager: full control access over all users, user groups and data.

b. Data administrator: managing database.

c. Scientific administrator: managing geo-processing tools, ensuring data accuracy.

d. End-users and possible municipality stakeholders (research/education/individual).

6.1.2. Non-Functional requirements

The user interface should display many different data using maps, tables, and parameters. This should be approached by prototyping and iterating over the user interface several times before the final version of the application is released to the community. This requirement targets the usability of the application.

An application interface should enable the access and filtering of the data sources integrated. We expect two mains of access: through an API capable of providing the functionalities that fulfil the above requirements, and exposed to the programming models, and through a simple graphic application build up on top of those APIs.

R1.5. An API must be exposed to deal with the storage resources to authenticate, populate data, retrieve and filter data, update data. Same operations for metadata. Data access should have a short latency (near real-time access).

R1.6. A basic user interface should be developed to browse, filter and update data and metadata. User interface should present the data in the form of a map.

6.2. UC2 requirements

The main aim of UC2 - Descriptive Models is to extract, characterize trajectories and to obtain correlations among them and other associated metadata. In order to do so, several functional requirements are identified that depend on WP7 (for the implementation of the DM Service) and other WPs to provide convenient APIs and components.

6.2.1. Functional requirements

The functional requirements of Use Case 2 involve both WP7 and the rest of technical WPs.

R2.1. The service must extract trajectories from the data in UC1, including not only dynamic spatial data, but also the other types of data that enrich the trajectory information.

R2.2. The service must be able to characterize trajectories, deriving aggregated and non-aggregated statistics, where the latter consider the characteristics of several entities. The characterization also
aims to identify characteristics probability distributions for the various statistics, as a strategy to extrapolate the findings and the underlying phenomena.

R2.3. The service **must** be able to determine correlations (sets and sequences) among trajectories, looking for features that characterize them. Besides, the system should provide outlier detection, which could identify abnormal situations in traffic or weather.

R2.4. The service **must** cluster similar trajectories, exploiting several dimensions: location, time, weather, sentiment/opinion, among others. This analysis could derive typical travel patterns, passenger spatio-temporal hotspots, desired lines and places, landmarks or trajectories with similar sentiment or topics associated to a given trajectory. The analysis will be noise-tolerant.

R2.5. The service **must** facilitate the end-user to select the data sources, temporal and spatial scales and output format for historical time-series analysis.

R2.6. The service **must** facilitate the end-user to select an area of interest (e.g. the Batel District in Curitiba, Brazil), for that area the available data (e.g. GIS stationary data) **must** be retrieved from the application and the end-user **must** have the possibility through the service to select derived information for that area. Trajectory analysis algorithms may be implemented in an incremental way, therefore processing just recent data available as the whole data will not fit in memory.

R2.7. The service **must** present and visualise the key output of the use case in different ways: maps, graphs, tables, and comparative charts. Export of results should be also supported (CSV, JPEG formats).

R2.8. It **should** be possible to download aggregated results and products to be used in subsequent analysis.

6.2.2. Non-Functional requirements

The service will expose a user interface and execute processing actions on the filtered data for the analysis of trajectories, as stated above. For this purpose, the project will develop an API and a user interface.

R2.9. An API **must** be developed to expose the functionality of the above requirements from the Use Case 2 perspective.

R2.10. A user interface **must** display the aggregated and non-aggregated statistics, and parameters. This should be approached by prototyping and iterating over the user interface several times before the final version of the application is released to the community. This requirement targets the usability of the application.

6.3. UC3 Requirements

The main goal of UC3 is to build an application for the recommendation of routes to people that will help citizens to check how mobility conditions are and give hints on how to better reach destinations with regards to multiple criteria, such as time, predicted traffic (and hence stress), pleasantness to walk, sights, and interestingness (taken from social media).

In terms of Use Case 3 (Predictive Models), the project will implement two tasks: classification and regression. The first task aims to predict which is the route that best matches the user profile and preferences. There are two challenges here: (i) selecting the features to be used; (ii) choosing the technique to be employed, which should be capable to recommend routes that match the user profile and
expectations as much as possible. The second task is regression, which will be used for estimating (i) the duration of a trajectory considering current conditions and historical data and (ii) the level of stress of the users while using a given public transportation line (e.g. by means of sentiment analysis).

6.3.1. Functional Requirements

R3.1. The Predictive Models (PM) must construct models using the information from UC1 and UC2 using a batch-model execution service. The models will be noise-tolerant.

R3.1. The PM construction must recompute the models periodically (at least twice per day), with updated information. The system must guarantee that the results are likely to be obtained in a given time frame.

R3.3. The PM must be projected in interactive time to fulfil the traffic recommendation requests from end-users. It is required that the PM service can rely on the platform to answer up to 100 queries per second with a processing time of at most 2s per query. The recommendation will include both paths and destinations.

R3.4. The application must present and visualise the key output of the use case in different ways: maps, graphs, tables, and comparative charts. Export of results should be also supported (CSV, JPEG formats).

R3.5. Download of aggregated results and products must also be supported.

R3.6. The user interface must facilitate the end-user to select the data sources, temporal and spatial scales and output format for historical time-series analysis.

R3.7. The user interface must facilitate the end-user to select an area of interest (e.g. the Batel District in Curitiba, Brazil), for that area the available data (e.g. GIS stationary data) must be retrieved from the application and the end-user must have the possibility through the web-interface to select derived information for that area.

6.3.2. Non-Functional requirements

The user interface needs displaying many different data using maps, model types (eg. random forests, recurrent ANN), training procedures (eg. 10-fold cross validation) and data (eg. last month). This should be approached by prototyping and iterating over the user interface several times before the final version of the application is released to the community. This requirement targets the usability of the application.

A distinctive feature of the UC3 is that end-users are mostly interested in the final products rather than on raw data. Therefore, the access to/visualization of products from the web-interface must be fast. This requirement targets the performance of the gateway to post-process and display the products.

Finally, the Predictive Models Service will expose the functionality through to means:

R3.8. An API must be developed to expose the functionality of the above requirements from the Use Case 3 perspective.

R3.9. A user interface must display the results of the recommended routes by means of the projected models. This requirement targets the usability of the application.
7. CONSOLIDATED ANALYSIS AND ABSTRACTION OF GENERAL REQUIREMENTS

This section presents a consolidated analysis of the use case requirements. From this analysis, a set of general functional requirements have been identified to guide the design of the EUBra-BIGSEA infrastructure. These requirements will be prioritised and all of them are expected to be enforced in the final infrastructure. This document sets the starting point for the discussion and common understanding of the infrastructure needs and capabilities.

The requirements have been classified into three types: requirements for data access, requirements for execution, and requirements for security and logging.

All the use cases will contribute to the objective of creating a federation of heterogeneous infrastructures for computing and data resources. UC1, UC2 and UC3, will contribute to the objective of providing intensive computing using federated computing resources. UC1 and UC3 will execute complex workflows on a wide geographic area using the EUBra-BIGSEA infrastructure. Finally, UC1, UC2 and UC3 will process massive amounts of data.

7.1. General requirements of data access

Access to large-scale data storage and data processing facilities is the most important requirement of the project. As open-access databases continue to grow in scale and more information becomes available, stakeholders are finding more ways in which this information can be exploited to develop the new generation of scientific tools and applications. For example, UC1 uses moderated traffic network in order to process the data in integration of four different public data sources, containing georeferenced information.

The comprehensive analysis of the use case requirements yielded the following global requirements for data access:

RD.1. The infrastructure must support the integration of external data from existing data sources. This integration must be complemented with methods for referencing the data in their original locations, and to pre-process and annotate the data with additional information. Metadata standards must be used when available to annotate the data.

RD.2.Automatic synchronization with original data sources must be addressed (updating the infrastructure with the latest releases of the data), considering the individual needs of each case, which range from simply discovering and downloading new data when it becomes available, to running complex data pre-processing before storing the data in the infrastructure.

RD.3. The infrastructure must store the data processing products, taking the necessary steps to ensure data persistence and data protection, when necessary.

RD.4. The infrastructure must provide access to authorized applications to access and process the data, supporting the application data processing model.

RD.5. The infrastructure must facilitate the end-user to access the data, providing the most appropriate protocols and data formats to enable developers with the necessary means to build usable user interfaces. Data must be queried in a variable granularity.

RD.6. User Internet connection is a potential bottleneck for performance, especially low bandwidths as expected in field conditions. Therefore, the infrastructure should facilitate the access to the data even in poor Internet connections.
7.2. General requirements of execution

From the data mining point of view, four types of algorithms are envisaged:

- Approximative algorithms, providing lower-quality but faster/cheaper models.
- Privacy-preserving algorithms, which could find a trade-off among filtering vs. anonymization vs. obfuscation vs. privacy guarantees (based on metadata).
- Data-Quality-aware algorithms, handling uncertainty in data may require significant changes, also finding the balance between filtering and tolerance to uncertainty.
- Adaptive algorithms, whose demands vary as a function of instantaneous availability.

As described in section 3.3.3, there are four types of paradigms, which could have different demands in terms of computing:

- Combinatorial paradigm involves irregular, input- and parameter-dependent computation. Demand may vary significantly (including storage) during one execution and it will require to change the infrastructure resources allocated to maintain application state across the execution.
- Probabilistic paradigms are usually regular, but involve iterative computation. The demand does not vary significantly, unless it is necessary to record the execution evolution. Storage has fixed and predefined size and the response projection time is much shorter that the building of the models.
- Algebraic paradigms are usually regular, even when using iterative algorithms, but computationally intensive, as they involve significant numerical computation. The elasticity is not feasible in some cases, as they rely on existing libraries that typically cannot be updated dynamically (e.g. when using MPI-based libraries) - although it could possible for multi-core software libraries (e.g. when using OpenMP). Fixed output storage and same conditions as above for the response time.
- Graph-based paradigms are usually irregular in terms of computing, and computing isomorphisms is an expensive task. Computing demand may vary significantly and managing canonicals may require elasticity, also in terms of storage. The same conditions apply with respect to the response time.

The data analytics services to be developed in EUBra-BIGSEA must take into account the different execution models required by the three use cases. Data Acquisition (UC1) will require running filtering and data quality analysis continuously as data is retrieved by the platform. Descriptive Methods (UC2) will require running batch jobs to extract, characterize and cluster trajectories. Predictive Models (UC3) will require executing both batch jobs for the training and adjustment of parameters (even periodically or when changes in the environment are expected or detected) and interactive jobs for the projection of the models. Batch executions may require their completion at a given time, so QoS models should be defined to schedule the execution on the right time.

Therefore, from a platform perspective, three main types of processing are required: continuous preprocessing of raw data, batch processing for creating predictive models, and online processing for answering prediction queries. This analysis has reached several requirements on the need of computing resources:

RE.1. The infrastructure must support unrestricted batch execution of data analytic jobs. Unrestricted in the sense of no QoS-bounded batch jobs, where latency is not a key issue. Single jobs will be those that normally could fit in memory.

RE.2. The infrastructure must support unrestricted batch execution of a bag of data analytics jobs. A Bag of jobs is a model that fits the High-Throughput Computing (HTC) paradigm.

RE.3. The infrastructure must execute batch jobs with associated QoS. Executions should be characterized on time and could also require a bounded budget expressed in the maximum number
of resources’ time to be spent. The scheduler should adjust the resources to meet the expected QoS.

RE.4. The execution service of the infrastructure should provide deadline-based execution requests, which will have to finish at a given time, and are characterized in terms of resources and expected execution time. If the deadline is not feasible at submission time, it will notify the user and run immediately as resources are available. If not, it will schedule the execution for the future. If the execution takes place closer to the deadline, the data will be more up-to-date.

RE.5. The algorithms will be described in a way that the infrastructure can dedicate more resources to fulfill the QoS. The infrastructure must be reactive in both allocated computing resources and allocated memory. The infrastructure must be self-adapting in order to accommodate the workload peaks that can appear in HTC applications.

RE.6. The infrastructure must support the execution of short-jobs, finishing in interactive time, which could arrive massively (hundreds per minute).

RE.7. The infrastructure must support the execution of big data workflows, where the input data and the products can be large (in the order of tens of GBs) and they can be stored on servers geographically distant from each other.

7.3. General requirements for security and logging

In general, there are no strong requirements for security, since the data will be made available to the community openly. UC1, UC2 and UC3 will require user registration for accounting purposes and to support group-level policies to access the data.

On the other hand, UC2 will require privacy protection when mobility data is used to compute the models. However, models will be computed using HPC resources, so this requirement has to be met by the HPC provider.

Major security requirements are expected from other project activities that deal with interoperability. This fact might cause the use cases to adopt a specific security mechanism in order to successfully interoperate with other major cloud computing provider, such as the EGI federated cloud.

RA.1. The infrastructure must support end-user authentication for access control and accounting purposes.

RA.2. The infrastructure must support end-user authorization for accessing the data and the applications deployed with the infrastructure.

RA.3. The infrastructure must support group-level policies, defined by the different roles of the application, to access the data.

RA.4. Data privacy protection must be ensured in the resources where the UC2 will run.

RA.5. Proper annotation privacy issues. The execution of a given algorithm on a given data could produce sensible results, which must be properly reflected on the metadata of the produced results.
7.4. Analysis and prioritisation

Use Case and technical requirements have a deep relation. The carefully planning of the implementation of the requirements should take into account the expectancies of the use cases, not only in terms of functionality and performance, but also with respect to the timeline.

This document sets up the first step for this process, by means of the identification of the requirements and the potential relations among them. This plan is dynamic and will be updated continuously during the project lifetime.

Figure 6 shows a diagram of identified interactions among the requirements. All the requirements have been coded as issues that will be followed through the issue tracker system. Blue arrows denote dependencies among the same category of requirements (either at Use Case level or at Execution, Data or Security levels). Green arrows denote interactions among Use Cases and technology requirements. Orange arrows denote relations among technology requirements. Finally, purple arrows denote interaction among Use Cases.

These dependencies are used by the different workpackages leaders to define the workplan of their activities.

This analysis clearly depicts the following conclusions:

- The Use Case of Data Acquisition is critical for the development of the rest of the Use Cases, which will consume the data. The workplan is already defined in a way that this Use Case will produce a first prototype by PM12.
- The Use Case of Predictive Models requires information from the trajectories obtained from the Use Case of Descriptive Models, although the work to integrate the job submission and the algorithm characterization can be performed in parallel.
- Authentication and authorization policies should be defined as soon as possible to guide the implementation of both data privacy and privacy metadata. Job submission can be implemented without a strong dependency on the AA policies.
- Batch job submission, QoS execution and Bag of Tasks are key requirements that have a strong dependency on the rest.
- Storage of processing products is a key requirement since the way to consume the results of the different Use Cases is through the storage.
7.5. Final Considerations

This section presents additional information that arises from the analysis of the use cases that are not strictly speaking part of the requirements, but are important for the implementation of the use cases.

7.5.1. Software Licenses

Software developed in the project that implements the use cases will be released under open source licenses. The project will rely on existing developments under Apache License 2.0, GPL V3. The developments of the use cases will follow a European Union Public Licence (EUPL)\(^3\) in the case of the European Partners and a similar compatible license in other cases. The software will be released under a multiple license consisting of the EUPL, the Apache License 2.0 and other licenses that could arise.

7.5.2. Version Control

The EUBra-Brazil organization was registered in Redmine facilitate community involvement in the development and validation of the use cases: [http://eubrazilcc-rm.i3m.upv.es/projects/eubrazilcropdata/](http://eubrazilcc-rm.i3m.upv.es/projects/eubrazilcropdata/). Each use case will create a repository within this organization for managing and distributing its own source code.

---

\(^3\) [https://joinup.ec.europa.eu/software/page/eupl](https://joinup.ec.europa.eu/software/page/eupl)
8. CONCLUSIONS

This deliverable provides the whole project consortium the expectancies by the user scenario developers with respect to the functionality they require to the platform. The document clearly identifies three different Use Cases that have clear interactions but some degree of concurrency at the same time.

Use Case 1 poses several requirements on the data integration, data synchronization and data access general requirements, especially in terms of the type of data, formats, heterogeneity and volume of the sources, retrieval policies and indexing. Use Case 2 will consume Use Case 1 data to identify and process (characterize, correlate and cluster) trajectories which will be stored back and consumed by the predictive models. Therefore, although trajectory data is necessary to construct the models, interactions will be done through the data storage.

Use Cases 2 and 3 have an incremental development cycle, starting from the construction of trajectories identification and basic models, which are used by subsequent stages.

Technology requirements must address the key requirements in terms of functionality of the Use Cases: data integration, data update, data filtering and retrieval, data products storage, batch execution, workflows and interactive execution. Then, additional features will be progressively added: privacy management, self-adaptive elasticity, Quality of Service execution, and Group policies.

The project uses a ticketing system to follow-on and track the development of the requirements. Each requirement is coded as an issue and inserted into the system with the dependencies from other requirements. This facilitates to follow-on the degree of development and to identify bottlenecks. Tickets are assigned to the WP leader for each activity who re-assign them to the specific technical people.

The evolution of the requirements according to the plan will be described in the periodic progress reports of the project.
9. **ANNEXES**

9.1. **Questionnaire for the identification of the use case requirements**

1. **Describe your type of application.**

   This refers basically the type of interaction that you expect from your users. For example, your application is executed in a command line, it is used through a web interface, or it is an executable program with its own graphical interface.

2. **Describe your data and your data sources.**

   First, you should list the data sources that your application will use. Then, you should complete this information by describing: 1) the nature of the data (is the data publicly available on the Web or, on the contrary, is there a license required to access the data?); 2) what is the expected data access volume required by a single execution (in your maximum and average cases); 3) what protocols are available to access the data (e.g. ftp, http)?; and 4) in which format is the data stored? In case that you use any domain-specific data, please, provide as many details as possible about the format, including links to existing standards or definitions and software libraries that can be used to manipulate your data.

3. **Describe your execution model.**

   Define the CPU time and memory required by a single execution in your maximum and average cases. Does your application involve any parallelization strategy (high-performance or high-throughput computing, multiple cores, multiple processing workflows)? How many concurrent executions do you expect to have? How many concurrent users?

4. **Describe your network traffic.**

   Define the amount of data that will be transferred over the network (bandwidth consumed) in your maximum and average cases. Note that you don’t need to measure the amount of data transmitted and received by your application. Instead, you can report here the size of your data and the expected frequency of download and upload operations.

5. **Describe your security constraints on accessing data and computing.**

   Do your data include sensitive, personal information that require any special security measures? For example, working with patients’ data requires privacy protection measures. Similar protection against data breaches could be applied also to computing resources, for example, by enforcing strong isolation between virtual machines, re-deploying applications instead of reusing the previously allocated ones, etc.

6. **Describe your constraints and preferences regarding programing languages, applications and platforms.**
Provide references to the tools your application uses. Are they freely available for download? Include information about licenses. Give some examples of how you normally use the tools. Explain your reasons for using or not using a particular framework or programming language to implement your application.

7. Provide examples of existing applications or projects that illustrate your needs.

Include links to the documentation of the applications and to the website of the projects. Describe the positive and negative features of the example applications that you provide.

3.2 Questionnaire for the identification of user communities

1. Define your user classes and representative users.

From your application’s point of view, what are the different user classes that you will support? Describe the roles played by each group that you define. An example of such a role is: a person who runs data processing, or a person who validates the data entered to the system, finds errors and repairs them. Note that a single user may play multiple roles.

2. Describe your users from a quantitative point of view.

How many users are in your community? How many of them are expected to use your application?

3. Which are the most representative IT tools of your domain?

Mention any Information Technology (IT) tool that is extensively used in your community (Web portals, online databases). For example, PubMed and GenBank are widely used in Bioinformatics.

4. What is the relation of your community with the new generation of Internet technologies and applications?

Is your community actively participating in the development of Internet-based applications? If so, list the most representative projects, the projects that are more close to you and to your application. Describe the experience of your community in the use of Cloud computing and other distributed computing infrastructures, such as Grid computing infrastructures.

5. Define your requirements decision-makers.

In your opinion, who can mediate over disagreements which may arise with regard to the requirements and ultimately decide upon using one approach over the other?
9.2. Tables of requirements
This annex includes a summary information of all the requirements. Descriptions are available in the proper section of this document.

9.2.1. Use Case Requirements
Requirements per Use Case

<table>
<thead>
<tr>
<th>Req</th>
<th>UC</th>
<th>Description</th>
<th>Priority</th>
<th>WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1.1</td>
<td>UC1</td>
<td>To integrate the GIS data sources included in the table 1</td>
<td>MUST</td>
<td>WP4</td>
</tr>
<tr>
<td>R1.2</td>
<td>UC1</td>
<td>To integrate meteorological/climate data sources</td>
<td>MUST</td>
<td>WP4</td>
</tr>
<tr>
<td>R1.3</td>
<td>UC1</td>
<td>Metadata must be included into the application</td>
<td>MUST</td>
<td>WP4</td>
</tr>
<tr>
<td>R1.4</td>
<td>UC1</td>
<td>To distinguish from different groups of users with different roles</td>
<td>MUST</td>
<td>WP6</td>
</tr>
<tr>
<td>R1.5</td>
<td>UC1</td>
<td>Availability of an API</td>
<td>MUST</td>
<td>WP4/5</td>
</tr>
<tr>
<td>R1.6</td>
<td>UC1</td>
<td>Development of a basic user interface</td>
<td>SHOULD</td>
<td>WP7/5</td>
</tr>
<tr>
<td>R2.1</td>
<td>UC2</td>
<td>Extract trajectories</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R2.2</td>
<td>UC2</td>
<td>Characterize trajectories</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R2.3</td>
<td>UC2</td>
<td>Determine correlations</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R2.4</td>
<td>UC2</td>
<td>Cluster trajectories</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R2.5</td>
<td>UC2</td>
<td>Selection of data sources for time-series analysis.</td>
<td>MUST</td>
<td>WP7/4</td>
</tr>
<tr>
<td>R2.6</td>
<td>UC2</td>
<td>Selection of the area of interest</td>
<td>MUST</td>
<td>WP7/4</td>
</tr>
<tr>
<td>R2.7</td>
<td>UC2</td>
<td>Export key ouput data</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R2.8</td>
<td>UC2</td>
<td>Reuse aggregated results</td>
<td>SHOULD</td>
<td>WP7/4</td>
</tr>
<tr>
<td>R2.9</td>
<td>UC2</td>
<td>API interface to DM service</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R3.1</td>
<td>UC3</td>
<td>To construct Predictive Models</td>
<td>MUST</td>
<td>WP7/5</td>
</tr>
<tr>
<td>R3.1</td>
<td>UC3</td>
<td>To recompute the PMs</td>
<td>MUST</td>
<td>WP7/5</td>
</tr>
<tr>
<td>R3.3</td>
<td>UC3</td>
<td>To project the PMs</td>
<td>MUST</td>
<td>WP7/5</td>
</tr>
<tr>
<td>R3.4</td>
<td>UC3</td>
<td>Visualize output data</td>
<td>MUST</td>
<td>WP7</td>
</tr>
<tr>
<td>R3.5</td>
<td>UC3</td>
<td>Download of aggregated results</td>
<td>MUST</td>
<td>WP7/4</td>
</tr>
<tr>
<td>R3.6</td>
<td>UC3</td>
<td>Selection of data sources</td>
<td>MUST</td>
<td>WP7/4</td>
</tr>
</tbody>
</table>
R3.7. UC3 Selection of area of interest
R3.8. UC3 API
R3.9. UC3 Web application

9.2.2. Technology Requirements
Requirements for execution, data access and security.

<table>
<thead>
<tr>
<th>Req #</th>
<th>Description</th>
<th>Priority</th>
<th>WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD.1</td>
<td>Integrate external existing data sources</td>
<td>MUST</td>
<td>WP4</td>
</tr>
<tr>
<td>RD.2</td>
<td>Automatic synchronization with original data sources</td>
<td>MUST</td>
<td>WP4</td>
</tr>
<tr>
<td>RD.3</td>
<td>Storage of processing products</td>
<td>MUST</td>
<td>WP4/6</td>
</tr>
<tr>
<td>RD.4</td>
<td>Authentication and Authorization</td>
<td>MUST</td>
<td>WP4/6</td>
</tr>
<tr>
<td>RD.5</td>
<td>Data Access</td>
<td>MUST</td>
<td>WP4</td>
</tr>
<tr>
<td>RD.6</td>
<td>Deal with poor-Internet connection limitations</td>
<td>SHOULD</td>
<td>WP4</td>
</tr>
<tr>
<td>RE.1</td>
<td>Unrestricted Batch jobs</td>
<td>MUST</td>
<td>WP3</td>
</tr>
<tr>
<td>RE.2</td>
<td>Unrestricted Bag of Tasks</td>
<td>MUST</td>
<td>WP3</td>
</tr>
<tr>
<td>RE.3</td>
<td>QoS Batch jobs</td>
<td>MUST</td>
<td>WP3</td>
</tr>
<tr>
<td>RE.4</td>
<td>Deadline-based scheduling</td>
<td>SHOULD</td>
<td>WP3</td>
</tr>
<tr>
<td>RE.5</td>
<td>Self-adapting elasticity</td>
<td>MUST</td>
<td>WP3</td>
</tr>
<tr>
<td>RE.6</td>
<td>Short-jobs</td>
<td>MUST</td>
<td>WP3</td>
</tr>
<tr>
<td>RE.7</td>
<td>Big Data Workflows</td>
<td>MUST</td>
<td>WP5</td>
</tr>
<tr>
<td>RA.1</td>
<td>End-user authentication</td>
<td>MUST</td>
<td>WP6</td>
</tr>
<tr>
<td>RA.2</td>
<td>Data and applications ACL</td>
<td>MUST</td>
<td>WP6/4/5</td>
</tr>
<tr>
<td>RA.3</td>
<td>Group-level policies</td>
<td>MUST</td>
<td>WP6</td>
</tr>
<tr>
<td>RA.4</td>
<td>Data privacy protection</td>
<td>MUST</td>
<td>WP6/4</td>
</tr>
<tr>
<td>RA.5</td>
<td>Privacy annotation</td>
<td>MUST</td>
<td>WP6/5</td>
</tr>
</tbody>
</table>
## GLOSSARY

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Explanation</th>
<th>Usage Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Authentication and Authorization</td>
<td>Security</td>
</tr>
<tr>
<td>ACL</td>
<td>Access Control List</td>
<td>Security</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
<td>Requirement Analysis</td>
</tr>
<tr>
<td>BoT</td>
<td>Bag of Tasks</td>
<td>Programming Model</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Value</td>
<td>Data Type</td>
</tr>
<tr>
<td>DM</td>
<td>Descriptive Models</td>
<td>Use Case 2</td>
</tr>
<tr>
<td>ESRI</td>
<td>Environmental Systems Research Institute</td>
<td>Data type</td>
</tr>
<tr>
<td>EUPL</td>
<td>European Union Public License</td>
<td>License Mode</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
<td>Data type</td>
</tr>
<tr>
<td>HPC</td>
<td>High-Performance Computing</td>
<td>Computing Architecture</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
<td>Data type</td>
</tr>
<tr>
<td>KOM</td>
<td>Kick-off Meeting</td>
<td>Project meeting</td>
</tr>
<tr>
<td>NETCDF</td>
<td>Network Common Data Form</td>
<td>Data type</td>
</tr>
<tr>
<td>NoSQL</td>
<td>Not Only SQL</td>
<td>Database paradigm</td>
</tr>
<tr>
<td>OSN</td>
<td>Online Social Network</td>
<td>Data type</td>
</tr>
<tr>
<td>PM</td>
<td>Predictive Models</td>
<td>Use Case 3</td>
</tr>
<tr>
<td>POSTGIS</td>
<td>Spatial and Geographic Objects for PostgreSQL</td>
<td>Database engine</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
<td>WP3</td>
</tr>
<tr>
<td>RTB</td>
<td>Research and Technical Board</td>
<td>Project Board</td>
</tr>
<tr>
<td>SAD69</td>
<td>Geodesic Reference system in Brazil</td>
<td>Reference system</td>
</tr>
<tr>
<td>UC1</td>
<td>Use Case 1 - Data Acquisition</td>
<td>Use Scenario part</td>
</tr>
<tr>
<td>UC2</td>
<td>Use Case 2 - Descriptive Models</td>
<td>Use Scenario part</td>
</tr>
<tr>
<td>UC3</td>
<td>Use Case 3 - Predictive Models</td>
<td>Use Scenario part</td>
</tr>
<tr>
<td>US</td>
<td>User Story</td>
<td>Requirement Analysis</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td>Category</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>UTF8</td>
<td>8-bit Unicode Transformation Format</td>
<td>Data Type</td>
</tr>
<tr>
<td>UTM</td>
<td>Universal Time Zone</td>
<td>Reference system</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
<td>WP3</td>
</tr>
<tr>
<td>WRF</td>
<td>Weather Research and Forecasting</td>
<td>Data Type</td>
</tr>
</tbody>
</table>