Deliverable
D3.5 Big Data Analytics components (Initial)

Project Acronym:	URBANAGE	
Project title:	Enhanced URBAN planning for AGE-friendly cities through disruptive technologies	
Grant Agreement No.	101004590	
Website:	www.URBANAGE.eu	
Version:	1.0	
Date:	31/01/2022	
Responsible Partner:	ATC	
Contributing Partners:	TEC, IMEC, UH, ENG	
Reviewers:	Jose Luis Izkara (TEC) Ville Santala (FVH)	
Dissemination Level:	Public	x
	Confidential – only consortium members and European Commission	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004590

Revision History

Revision	Date	Author	Organization	Description
0.1	8/10/2021	George Giotis	ATC	ToC
0.2	10/11/2021	George Giotis	ATC	Initial Input in chapters 3,4,5
0.3	10/12/2021	George Giotis	ATC	Input in chapters 3,4,5
0.4	11/01/2022	Manjarres Martinez Diana	TEC	Input in chapters 3 & 4
0.5	14/01/2022	Maritini Kalogerini	ATC	Chapters 1,2,6
0.7	18/01/2022	Ville Santala	FVH	Internal Review & comments
0.6	21/01/2022	Jose Luis Izkara	TEC	Internal Review & comments
0.8	22/01/2022	Maritini Kalogerini & Giorgos Giotis	ATC	Addressing comments
1.0	28/01/2022	Maritini Kalogerini & Giorgos Giotis	ATC	Final version

Table of Contents

Table of Figures

List of abbreviations

Abbreviation	Explanation
AI	Artificial Intelligence
DPPA	Descriptive, Predictive and Prescriptive Analysis
HDFS	Hadoop Distributed File Storage
ID	Identity Provider
ML	Machine Learning
OIDC	OpenID Connect
RDD	Resilient Distributed Datasets
DAG	Directed acyclic graph
LDAP	Lightweight Directory Access Protocol
HDFS	Hadoop Distributed File System

1 Executive Summary

This Deliverable “D3.5 Big Data Analytics Components_Initial” summarizes the work done under Task 3.3 “Big Data analytics” for providing Big Data analytics components to process large amount of data produced in the cities. The main goal is to extract knowledge and present the results through dedicated visual dashboards in URBANAGE platform. More specifically, real-time data coming from IoT devices as well as historical data will be the majority of what is referred as ‘Big Data’ in this Deliverable. These data will be processed and analyzed by the Artificial Intelligence algorithms and by the Big Data Analytics in order to provide valuable aggregations and statistical information. Under that scope, this document firstly summarizes the types of analytics, the proposed architecture of data processing as well as information about the data storage. Finally, an intuitive user interface is being analyzed, in order to visualize the results from the analytics components. This Deliverable is the first version out of two and it aims to set the basis for the upcoming implementation of the big data analytics components.

2 Introduction

Via URBANAGE activities, the consortium plans to implement a framework for decision making in the field of urban planning, with special focus on facilitating the older people aging well in cities. The process of the decision making is data-driven, taking advantage of massive data production and enhanced analytical capacities in the context of the current digital-era. A decision-support Ecosystem will be the basis of the pre-mentioned framework and it will integrate Big Data analysis, modelling and simulation techniques with Artificial Intelligence algorithms and adapted visualization methods through Urban Digital Twins and gamification for enhanced engagement purposes. In order to process large amounts of data produced in the cities and to extract useful knowledge, Big Data analytics components will be made available, through T3.3. Big Data will include real-time data coming from IoT devices as well as historical data collected through the Data Management layer which is defined in T3.1. Moreover, those data will be processed and analyzed by the Artificial Intelligence algorithms in order to provide aggregations and statistical information. An overview of the preliminary Artificial Intelligence algorithms functionalities identified and defined for the three pilot sites of the project (Santander, Flanders and Helsinki) is included in the deliverable "D3.3 AI Algorithms and Simulation Tools". Under this perspective, the deliverables D3.3 and D3.5 can be considered complementary since the deliverable D3.5 offers an overview of the background technologies and techniques for the management, processing and visualisation of big data, whereas, D3.3 provides an overview of the preliminary AI functionalities for the pilot sites, as well as a description of potentially usable data sources and frameworks. The different types of analytics in URBANAGE are a) Descriptive Analytics, for analyzing mainly historical data, b) Predictive analytics, for identifying the likelihood of future outcomes based on previous knowledge, and c) Prescriptive analytics, which involves the use of statistics and modelling to determine future performance, based on current and historical data. The results from the analytics components will be visualized with an intuitive user interface based on the Apache Superset framework, as described below.

2.1 Structure of the report

This document is divided in 6 main sections and it is structured as follows:

Section 1, includes the executive summary of this report.
Section 2, includes a short introduction and the structure of the report.
Section 3, the Types of Big Data analytics, including Descriptive and Predictive analytics.
Section 4, summarizes the main tools that are going to be used in the context of Big Data analytics.
Section 5, refers to the visualization of the results from the analytics components with a user interface based on the Apache Superset framework.
Finally, Section 6 highlights the main outcomes and concludes this report.

3 Types of Big Data analytics

Big Data analytics is the process of investigating a large amount of data, trying to uncover useful information like hidden patterns and relations between the various entities of interest, trends etc, aiming this way to aid in the business decisions of an organisation. In this section, we are presenting the main types of Big Data analytics processes and briefly discuss how these can be used in the context of the project based on the user stories and functional requirements described in D6.1 [1] and D5.1 [2] respectively. The Big Data analytics components are highlighted in the overall architecture that is shown in Figure 1.

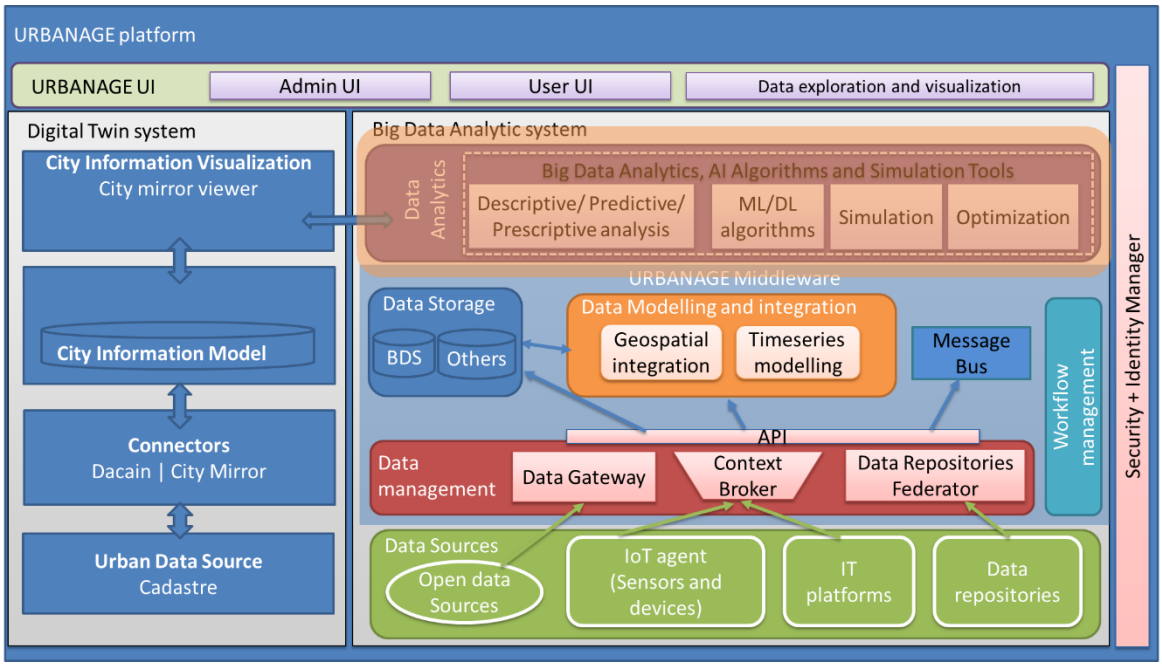

Figure 1: URBANAGE overall architecture

The functionalities and tools presented in this section and the next ones will be part of the Descriptive, Predictive and Prescriptive Analysis component (DPPA) that will be implemented in the context of WP3, however the Optimisation and Simulation components are also mentioned and briefly summarized in this section, since they are part of the Big Data analytics system that we envision for the project.

3.1 Descriptive

Descriptive analytics refer to a data analysis performed mainly on historical data in order to get insights with regard to what has already happened in the context of public domain with a focus on older peoples' needs. It's a rather simplistic type of analytics that uses simple mathematical and statistical methods such as average and percentage changes, and the results can be visualised using for example line graphs, pie charts and bar

charts. A key method in this type of analytics is the data aggregation. The historical data can be grouped in different ways so that it can address the need for representing the data at different spatial scale, whether it is on a neighbourhood, a masterplan or a city level. Another example where this type of analytics is applicable is about the maintenance interventions of temporary pedestrian routes so that the user knows whether they are clearly marked or whether they are maintained properly at all times of year. The same applies for the maintenance interventions of pavements so that the user can avoid slippery conditions and snow piles near points of interest (i.e., bus and metro stations, roads, sidewalks etc.). Moreover, data about health care services (i.e., doctors, hospitals, pharmacies, physiotherapists, care centres etc.), their availability and accessibility will be analysed to allow a user to know whether there are facilities close by.

3.2 Predictive

Predictive analytics refer to a type of data analysis that allows to identify the likelihood of future outcomes based on previous knowledge. In the context of URBANAGE, machine learning (ML) algorithms will be applied to predict future potential needs for health care services accessibility for older people as well as for people with disabilities. In addition, critical situations (e.g., slippery conditions and snow piles) about pavements and pedestrian routes will be modelled using ML techniques so that it can be predicted the location of infrastructure that older people should avoid.

The predictive analytics will be implemented as Machine Learning (ML) pipelines using the Spark Machine Learning library (MLlib) [3]. MLlib offers toolsets for common ML algorithms such as classification, regression, and clustering. Moreover, it supports a wide range of preprocessing utilities on the raw datasets such as feature extraction, transformation, dimensionality reduction, and selection. In addition, it offers tools for persisting and loading the trained models and pipelines.

Spark ML pipelines provide a set of high-level APIs (built on top of DataFrames API) that allow users to create and tune Machine Learning pipelines. A pipeline can be composed by multiple subcomponents, each one responsible to perform a specific action on the data such us tokenization, hashing, regression etc. The key concepts in the Spark Pipelines API are dataframes, transformers and estimators.

Part of the predictive analytics process can be considered the simulation component, especially in cases when historical data are not adequate to train or test the machine learning algorithms. For example, in cases like these, simulation could potentially provide synthetic training data.

3.2.1 Simulation component

The principal objective of the simulation component is to offer a strong simulation tool which copes with the limitations or complexity of movement of older people. In the context of this project, this simulation module will congregate the mechanisms needed for building the simulation engine for the optimal deployment of urban accessibility. Among other functionalities, implemented tools will simulate a remarkable number of scenarios aiming to find the best possible locations for different urban furniture. It will also consider the deployment of vertical transport and automatic ramps with the specific extract of older people in the context of global multimodal urban mobility and accessibility. Moreover, this simulation component will also include

functionalities regarding the long-term use aiming at improving the age friendliness of a specific neighbourhood (urban accessibility, access to public services etc).

3.3 Prescriptive

Prescriptive analytics in the context of URBANAGE can help users to answer questions about what should be done to ensure that older people or people with disabilities have a safe environment to move. For example, the lighting capacity of city areas could be analysed to learn what the optimal conditions are for providing a safe environment. Another key component in this type of analytics is the optimization component that is described in the section that follows. It should be noted that the Simulation component (described in high level in section 3.2.1 and in detail in D3.3) can potentially fit into the prescriptive analytics too, in the sense that it can be used to validate the results of the Optimization component (described in high level in section 3.3.1 and in detail in D3.3). The Optimization component will be mainly developed in the context of T3.2 however it can be considered that it can possibly aid the Big Data analytics procedures.

3.3.1 Optimization component

The optimization component is comprised of the algorithms, heuristics and metaheuristics which will solve all optimization problems defined on URBANAGE. As an example, techniques and functionalities regarding the age-friendly route planning system provide comfortable routes for older people considering several variables such as: the state of the street, obstacles, noise, temperature, pollution, benches, public toilets, urban accessibility infrastructures, shadowed places among others, will be placed within this component. Moreover, this component will include Machine Learning and deep learning methods able to obtain data and provide valuable predictive insights. It will try to give detailed information about the reason for the choice of the route.

This component will feed both from Big Data Storage and from the updates found in the City Information Model (CIM) [4] to achieve better precision.

4 Data processing architecture

In this section we present the main tools that are going to be used in the context of Big Data analytics. This list is not limited to the tools presented but will be updated if needed as the project progresses, in order to tackle better the user needs and the technical challenges that may arise. For the same reasons, these tools will be accompanied / complimented by custom implementations when needed in the context of the DPPA component.

For development purposes of the URBANAGE Platform, the project established a CI/CD (Continuous Integration and Deployment) process that includes among the tools to be used, a code repository (i.e. GitLab). The code repository collects the prototypes of the main components (e.g. baseline tools, libraries, datasets, etc.) constituting the DPPA and their future developments. The deliverable "D5.2 Initial Platform Prototype" provides details about the CI/CD process and the code repository.

Figure 2 depicts the data processing workflow for the various Big Data analytics to be implemented in the URBANAGE context. The data processing cluster is shown in Figure 2 and it refers to a cluster of Spark workers. The Spark workers consume the previously ingested data objects from the data lake which is a data storage backed by the MinIO server [5]. In between the Spark-Select component allows to offload SQL queries and return the relevant data object subsets that needs to be processed.

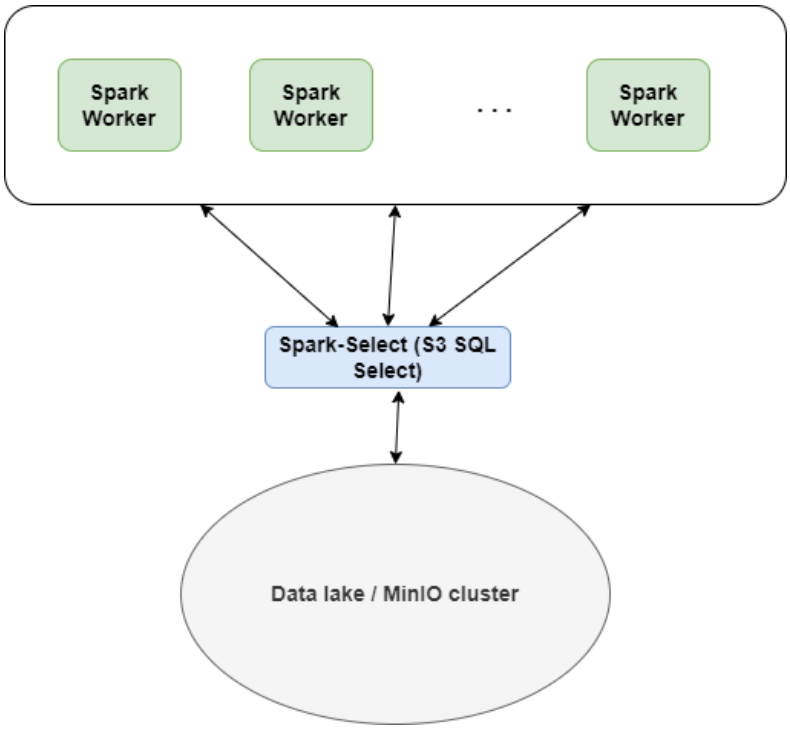

Figure 2: Data processing workflow for the URBANAGE Big Data analytics

A detailed description of the processing and storage components follows.

4.1 Technologies

An important requirement for choosing a framework suitable for the needs of Big Data analytics in the context of URBANAGE is the ability to support both batch and streaming processing. An initial evaluation of the available and most popular Big Data analytics frameworks has been performed focusing on the open source ecosystem. Apache Hadoop [6] is a framework for scalable and distributed processing that has been extensively used for Big Data analytics but it exclusively supports batch processing. Apache Storm [7] is another candidate but Storm mostly focuses on complex event processing by implementing a fault tolerant method to pipeline different computations on an event as they flow into the system. There are separate components in the URBANAGE architecture that already address the need for event processing (i.e. Kafka broker and context broker). Apache Flink [8] provides stateful computations over data streams. It lacks the maturity with regard to the SQL support that Spark provides through the SparkSQL functionality.

The Big Data analytics layer is based on the Apache Spark [9] framework. Spark is a cluster computing framework that fits well with large scale data processing. It consists of a set of distributed worker processes called executors. An executor is a distributed process responsible for the execution of tasks, as shown in Figure 3**Error! Reference source not found.**.

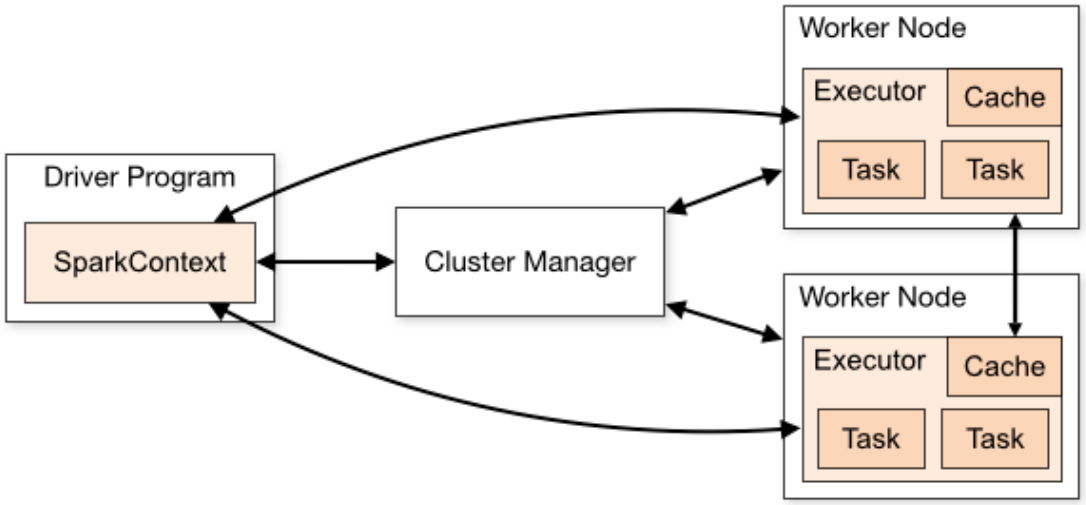

Figure 3: Spark internal architecture

The building blocks of a Spark application are the Resilient Distributed Datasets (RDDs) and the Directed Acyclic Graph (DAG). Eventually all data to be processed have to be stored inside an RDD which is actually a distributed data structure. Spark avoids data loss from hardware failures by providing fault tolerance on a RDD level where an RDD can rebuild itself in the event of failure. Typically, a single RDD is stored on a series of different nodes in the cluster to address the need for no single point of failure. This way the cluster can operate on the RDDs in parallel. A series of transformations are applied to the RDDs and finally a “Reduce” action that aggregates all the elements of an RDD using some function is applied. Whenever an action is performed on an RDD, Spark creates a DAG which is a finite direct graph of functions to be applied on the dataset.

4.1.1 MinIO Spark Select

The integration between the processing framework (Spark) and the URBANAGE data lake (MinIO based) is achieved using the Spark-Select [10] functionality. The MinIO S3 Select API (that actually adds SQL query capabilities to the native MinIO API) allows to send query jobs to the MinIO server make it possible to speed up the analytics workflow. The workflow is the following: when the application sends an SQL query, the MinIO server loads only the relevant subset of the data objects into memory. This way the Spark jobs are executed faster and less network/compute/memory resources are utilized. Moreover, it allows the Spark jobs to run concurrently and faster. The Spark-Select API acts as a Spark data source implemented as a DataFrame interface. It actually converts any filters into SQL Select queries and then returns the data subset results as DataFrames to be directly consumed and processed by the Spark jobs. The supported file formats are JSON, CSV and Parquet. Spark-Select can be integrated with Spark via spark-shell, pyspark, spark-submit etc. or can be added as Maven dependency or simply as a jar import.

4.2 Data storage

To address the need for processing large scale data a data lake solution is implemented. It serves as a centralized repository that holds any structured or unstructured data that will be generated from the various URBANAGE data sources at any scale, as shown in Figure 4. Different types of analytics can run on top of the data lake like visualizations, real time analytics, Big Data processing and machine learning.

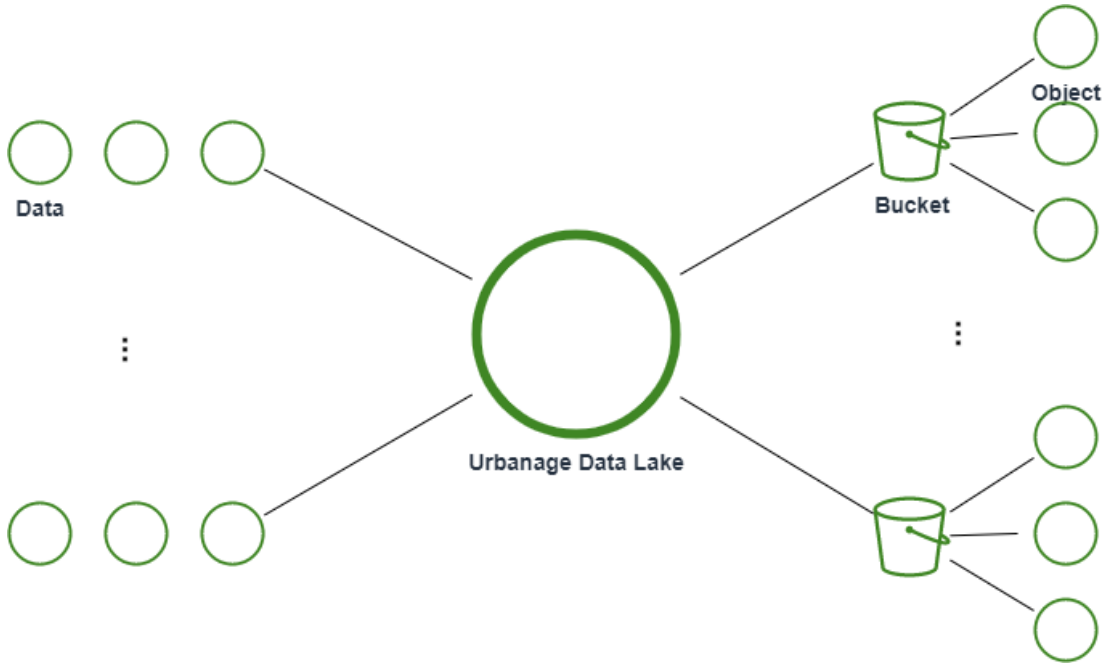

Figure 4: URBANAGE data lake storage

The URBANAGE data architecture ensures compliance by following a unified way to secure, monitor and manage access to the data. Moreover, it allows to scale without compromising the performance of the analytics services.

4.2.1 Technologies

The Hadoop distributed file storage is the most common solution for Big Data storage to consider. Though it imposes some limitations. For example, it provides support for batch processing only and lacks support for real time processing. In Hadoop, MapReduce framework is comparatively slower, since it is designed to support different data formats and structures. In MapReduce, Map takes a set of data and converts it into another set of data, where individual elements are broken down into key-value pairs and Reduce takes the output from the map as input and process it further. MapReduce requires a lot of time to perform these tasks thereby increasing latency. Moreover, HDFS is not efficient for caching. In Hadoop, MapReduce cannot cache the intermediate data in memory for further processing which eliminates the performance of Hadoop/HDFS.
The reference implementation of the URBANAGE data lake storage is based on the MinIO [11] object storage. MinIO is a distributed and high-performance object storage system that is open source under the GNU AGPL v3 license [12]. It provides enhanced performance compared to a typical storage solution for Big Data analytics which is the Hadoop Distributed File Storage (HDFS). The MinIO object storage organises the data objects into buckets, as shown in Figure 5. A bucket is similar to a folder or directory in a file system, where each bucket can hold an arbitrary number of objects.

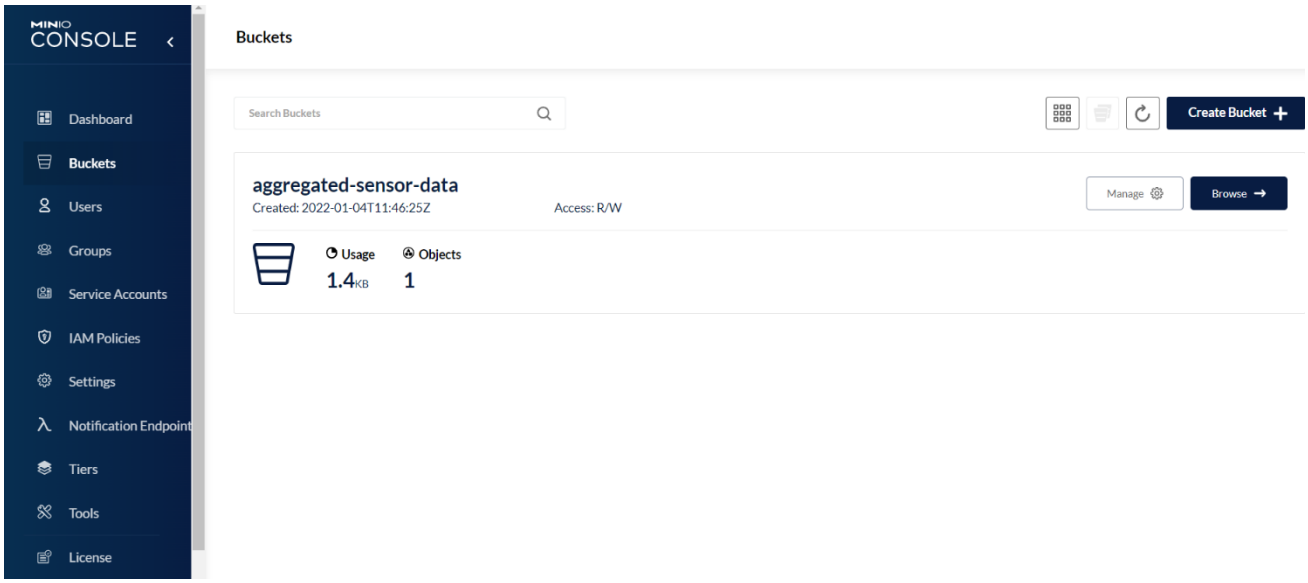

Figure 5: MinIO server setup on URBANAGE staging environment

A key feature that MinIO supports is the bucket notifications that allows to automatically publish notifications to one or more configured notification target endpoints when specific events occur in a bucket. Moreover, MinIO supports end-to-end encryption of objects not only while data objects are transmitted over the network (network encryption in transit) but also while objects reside in the storage buckets (at-rest).
In terms of security, MinIO provides built-in identity management functionality through an internal Identity Provider [13] (IDP) mechanism. Specifically, MinIO requires clients to authenticate using the AWS Signature version 4 protocol [14]. In order to connect, a client is required to provide a valid access key as well as a valid secret key on every API call. In addition, it supports external identity management either through an OpenID Connect (OIDC) compatible service or through an Active Directory or LDAP service. Once authenticated, the

client request is either allowed or rejected depending on whether the authenticated identity is authorized to perform the operation on the specified bucket/object. The authorized actions and resources that a user has access to are defined through a policy.

5 Data exploration and visualizations

The results from the analytics components will be visualized with an intuitive user interface based on the Apache Superset [15] framework. Superset is considered highly available, and it is designed to scale out to large, distributed environments. Many types of graphs are supported, as shown in Figure 6, for example:

- Funnel chart
- Force directed graph chart
- Circular graph chart
- Mixed timeseries chart
- Gauge chart
- Radar chart
- Tree chart
- Pie chart
- Heatmap

Superset provides a SQL editor for preparing data for visualization, including a rich metadata browser. Moreover, it allows to build custom visualization plugins. The option to include a Superset dashboard in a user interface build with the React library will be investigated.
Superset provides out-of-the-box support for most SQL-speaking databases so the integration with Spark SQL fits well. Superset is agnostic with regard to the connectivity to databases, except for SQLite, which is part of the Python standard library. One needs to install the required packages for the database of choice as the metadata database as well as the packages needed to connect to the backed database.

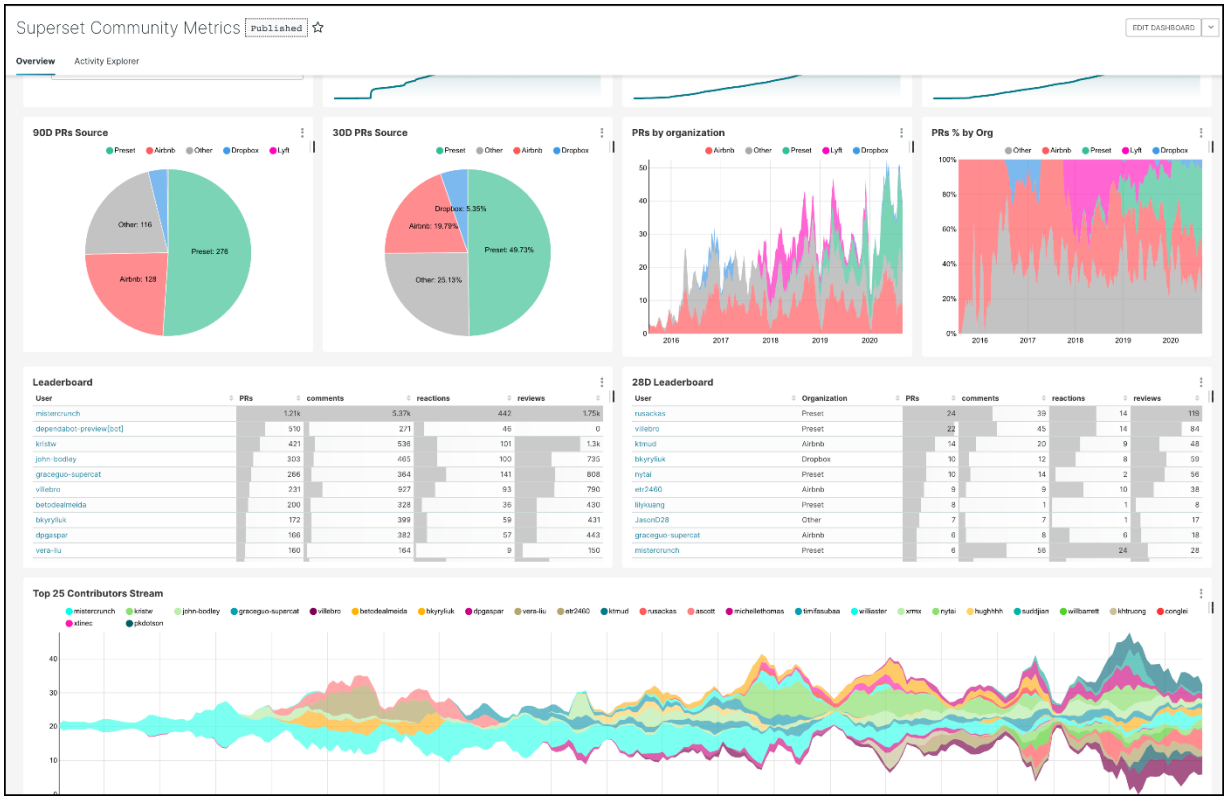

Figure 6: An example of a User Interface composed of Superset charts

In terms of security, Superset provides an extensible security model that allows configuration of rules regarding the access to specific datasets. The tool can be integrated with the Identity Manager component described in D5.1 in order to enable access control on user data.

6 Conclusion

This document summarizes the work done for providing Big Data analytics components to process large amount of data produced in the cities, in order to extract knowledge and present the results through dedicated visual dashboards in URBANAGE platform. Real-time data coming from IoT devices and historical data will be processed and analyzed by the Artificial Intelligence algorithms and by the Big Data Analytics in order to provide valuable aggregations and statistical information. For this reason, this document firstly clarifies the types of analytics in URBANAGE, as well as the relevant components such as the 'Optimization component' and the 'Simulation component'. Moreover, the data processing workflow is analyzed, with special focus in a detailed description of the processing and storage components. Finally, with regards to the data exploration and visualization activities, the results from the analytics components will be visualized with an intuitive user interface based on the Apache Superset framework.

It should be taken into account that this Deliverable is the initial version that aims to set a clear basis for the upcoming implementation of the Big Data analytics components, during the URBANAGE's lifetime.

References

[1] D6.1 Use case implementation and validation Plan
[2] D5.1 System Architecture & Implementation Plan
[3] https://spark.apache.org/docs/latest/ml-guide.html
[4] D4.1 CIM Structure definition and Existing components
[5] https://min.io/
[6] https://hadoop.apache.org/
[7] https://storm.apache.org/
[8] https://flink.apache.org/
[9] https://spark.apache.org/
[10] https://github.com/minio/spark-select
[11] https://min.io/product/overview#
[12] https://github.com/minio/minio/blob/master/LICENSE
[13] https://docs.min.io/minio/baremetal/security/minio-identity-management/basic-authentication-with-minio-identity-provider.html#minio-internal-idp
[14] https://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-authenticating-requests.html
[15] https://superset.apache.org/