IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON URBAN COMPUTING & WELL-BEING IN SMART CITIES: SERVICES, APPLICATIONS, POLICYMAKING CONSIDERATIONS

Received July 31, 2019, accepted August 14, 2019, date of publication August 22, 2019, date of current version September 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936941

An Integrated Big and Fast Data Analytics Platform for Smart Urban Transportation Management

SANDRO FIORE[1], DONATELLO ELIA[1,2], CARLOS EDUARDO PIRES[3], DEMETRIO GOMES MESTRE[3], CINZIA CAPPIELLO[4], MONICA VITALI[4], NAZARENO ANDRADE[3], TARCISO BRAZ[3], DANIELE LEZZI[5], REGINA MORAES[6], TANIA BASSO[6], NÁDIA P. KOZIEVITCH[7], KEIKO VERÔNICA ONO FONSECA[8], NUNO ANTUNES[9], MARCO VIEIRA[9], COSIMO PALAZZO[1], IGNACIO BLANQUER[10], WAGNER MEIRA, JR.[11], AND GIOVANNI ALOISIO[1,2]

[1]Euro-Mediterranean Centre on Climate Change (CMCC) Foundation, 73100 Lecce, Italy
[2]Department of Engineering for Innovation, University of Salento, 73100 Lecce, Italy
[3]Departamento de Sistemas e Computação, Universidade Federal de Campina Grande (UFCG), Campina Grande 58429-900, Brazil
[4]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy
[5]Barcelona Supercomputing Center (BSC), 08034 Barcelona, Spain
[6]Department of Software Engineering and Information Systems, University of Campinas (UNICAMP), Campinas 13484-332, Brazil
[7]Department of Informatics, Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba 80230-901, Brazil
[8]Department of Electronics, Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba 80230-901, Brazil
[9]CISUC, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
[10]Institute of Instrumentation for Molecular Imaging (I3M), Universitat Politècnica de València, 46022 Valencia, Spain
[11]Departamento de Ciência da Computação, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte 31270-901, Brazil

Corresponding author: Sandro Fiore (sandro.fiore@cmcc.it)

This work was supported by the European Commission through the Cooperation Programme under EUBra-BIGSEA Horizon 2020 Grant [Este projeto é resultante da 3a Chamada Coordenada BR-UE em Tecnologias da Informação e Comunicação (TIC), anunciada pelo Ministério de Ciência, Tecnologia e Inovação (MCTI)] under Grant 690116.

ABSTRACT Smart urban transportation management can be considered as a multifaceted big data challenge. It strongly relies on the information collected into multiple, widespread, and heterogeneous data sources as well as on the ability to extract actionable insights from them. Besides data, full stack (from platform to services and applications) Information and Communications Technology (ICT) solutions need to be specifically adopted to address smart cities challenges. Smart urban transportation management is one of the key use cases addressed in the context of the EUBra-BIGSEA (*Europe-Brazil Collaboration of Big Data Scientific Research through Cloud-Centric Applications*) project. This paper specifically focuses on the City Administration Dashboard, a public transport analytics application that has been developed on top of the EUBra-BIGSEA platform and used by the Municipality stakeholders of Curitiba, Brazil, to tackle urban traffic data analysis and planning challenges. The solution proposed in this paper joins together a scalable big and fast data analytics platform, a flexible and dynamic cloud infrastructure, data quality and entity matching algorithms as well as security and privacy techniques. By exploiting an interoperable programming framework based on Python Application Programming Interface (API), it allows an easy, rapid and transparent development of smart cities applications.

INDEX TERMS Big data, cloud computing, data analytics, data privacy, data quality, distributed environment, public transport management, smart city.

I. INTRODUCTION

Nowadays, many fields including connected societies, scientific research, Information Technology and business are facing strong big data challenges connected to data heterogeneity, high data production rate as well as the enormous volume of data produced at multiple levels and by different entities. The data deluge is pervasively affecting societal contexts [1]–[4] by representing an invaluable source of information to address the well-being of cities'

The associate editor coordinating the review of this article and approving it for publication was Miltiadis Lytras.

inhabitants [5]. In the smart city [6] context, smart urban transportation management can be considered itself as a very complex and multifaceted big data challenge [7]. It strongly relies on the information collected into multiple, widespread, and various data sources as well as on the ability to extract actionable insights and holistic understanding from them, frequently employing data mining and machine learning techniques [8]. Besides data, to address all the technical challenges and meet users' requirements, a full software stack solution (from platform to services and applications) needs to be specifically adopted [9]. Presently, the growing availability of big data platforms and cloud computing providers, often with their own versions of ready-to-use big data solutions, represents a great opportunity for extracting value from data and gain strategic advancement in many sectors.

In this respect, the *Europe - Brazil Collaboration of Big Data Scientific Research through Cloud-Centric Applications* (EUBra-BIGSEA[1]) [10] project provides a cloud-enabled big data platform to ease the development of highly-scalable, privacy-aware data analytics applications running on top of cloud infrastructures, reducing development cycles and deployment costs. The EUBra-BIGSEA project has been 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. It is aimed at covering general requirements of multiple application areas, although in the context of the project timeline it has focused on the management of massive connected society information. EUBra-BIGSEA has delivered three applications to cope with urban transportation planning aspects. It has not only investigated speed, vehicle flux, traffic disruptions, main origin-destination routes for cities based on the day [11], time and area covered, but also human-side attributes, such as feelings, stress caused by traffic, landmarks, the presence of green areas, weather conditions and their impact on the people moving about in a city.

This work specifically focuses on the City Administration Dashboard, a public transport analytics application that has been used in the Municipality of Curitiba, Brazil, to tackle urban traffic data analysis and planning challenges. In particular, this application provides a wide set of multifaceted, integrated views and aggregate statistics (e.g., by different periods, parts of the city and bus lines) of the public transport system to support reporting and consultation activities, while abstracting from the underlying (big) data management aspects.

This paper provides a clear understanding and an in-depth view of the requirements, goals, workflow, design and implementation of the target application. It comprehensively describes the ICT solution proposed in the EUBra-BIGSEA context, the developed services and algorithms from an application-centric perspective. In this respect, it focuses on data quality and privacy aspects as well as descriptive analytics and graphical user interface (GUI) implementation details, thus providing a novel contribution with respect to previous work ([10], [12]–[15]).

The rest of the paper is organized as follows: Section II provides a high-level view of the City Administration Dashboard whereas the application design specifics are provided in Section III. Section IV presents general aspects of the cloud and security infrastructure, whereas Section V specifically deals with data privacy aspects. Section VI provides a comprehensive, in-depth analysis of the data quality aspects concerning the application pipeline. Section VII introduces the descriptive analytics component whereas Section VIII provides a complete description of the main views developed in the application user interface. Section IX presents a comprehensive state of the art analysis with respect to two different major dimensions of the proposed work: big data infrastructures and end-user applications. Section X is about the societal impact of the application, its exploitation and the users' feedback. Finally, Section XI draws the conclusions and highlights future work.

II. THE CITY ADMINISTRATION DASHBOARD APPLICATION

This section provides a general overview of the City Administration Dashboard application. In particular, the focus is on the public transport system management functionalities for urban planning, reporting and consultation by the relevant stakeholders (e.g., Municipalities). The rest of the section is organized as follows: Section II-A highlights the rationale and main goal of the application, whereas Section II-B presents the high-level application workflow; finally, Section II-C introduces the trans-Atlantic testbed infrastructure hosting the different components of the whole application.

A. MAIN GOAL

The main application target audience is municipality executives who manage a large public transport system. The application is primarily intended to cater for the needs of the municipality of Curitiba, a 1.8 million-inhabitants city in the south of Brazil. Both workers from the municipalities of Curitiba and Campina Grande were consulted during requirement elicitation. In both contexts, it is apparent that a public transport system is a data intensive urban system to operate, but one where municipalities typically cannot perform meaningful data mining.

The main challenges hindering their present analytical capacity are related to the sheer data volume, and to noise and heterogeneity in the data formats and production. There are many sensors/devices installed on buses and terminals, generating vast amounts of data every day, consisting of, but not limited to, bus Global Positioning System (GPS) position and speed records, and passenger boarding information. To give an idea, in the context of Curitiba, the bus system has a fleet of 1,290 vehicles that serve 1.5 million passenger trips on a daily basis. The service performs over 23,000 bus trips a day

[1]https://www.eubra-bigsea.eu/-Last visited on July 2019

and covers the metropolitan area of Curitiba including nearby districts. These entities and transactions generate valuable data which can be mined to extract useful information for administrators, operators and users of the Transportation System. Such data is collected by different systems with differing objectives. For instance: *Automatic Fare Collecting* (AFC) systems collect information about passenger boardings to be used by the transportation consortium as evidence of the share of city passengers, usually recording the passenger card ID, bus route and vehicle (but not the boarding location). *Automatic Vehicle Location* (AVL) systems, in turn, focus on the bus trajectory traces to assess the routes compliance with the predefined trajectory shapes, mainly recording bus route and vehicle, and the time series of their location geographic coordinates, not keeping track of the trip stops timestamps. If one wants to know the boarding location of passengers in a city, it will be necessary to merge the data types from the above systems, which clearly have no direct link to each other. Thereby, given the vast amount of data and the diversity of sources, collection intent, and nature, it becomes hard to integrate and analyze such data.

The chief goal of the City Administration Dashboard is therefore to provide multiple intuitive views of aggregate statistics of the public transport system, completely abstracting data processing and integration. Moreover, municipalities must be able to filter any analysis by different periods, parts of the city and bus lines. These are needed to correlate patterns with other societal and geographical covariates.

B. HIGH-LEVEL APPLICATION WORKFLOW

As mentioned before, the City Administration Dashboard application should produce and make available online to the final stakeholders in the Municipality of Curitiba a set of summary statistics for consultation and public transportation management activities. To this end, the application goes through multiple steps starting from the raw data sources to the final output by exploiting the *EUBra-BIGSEA platform* services. Fig. 1 shows the general application workflow from a high-level perspective. Several and heterogeneous types of data sources (both from the Municipality and the available open data repositories), regarding public transportation means, represent the input of the application. Such data is available as multiple files and stored on a distributed storage system as a setup enabling subsequent processing and analysis steps. Since the input data includes sensitive information, the first step of the workflow is a data anonymization stage. Then, data undergoes Extraction, Transformation & Loading (ETL) and pre-processing steps in order to (i) properly organize data for further processing, (ii) integrate the information from the various data sources and (iii) annotate it with additional descriptive metadata (e.g., data quality information). Once the data has been pre-processed, the descriptive analytics stage computes the actual aggregate statistics over some predefined time ranges (e.g., hourly, daily, weekly, monthly) and multiple dimensions (e.g., bus lines, bus stops, bus users). To address fast data management,

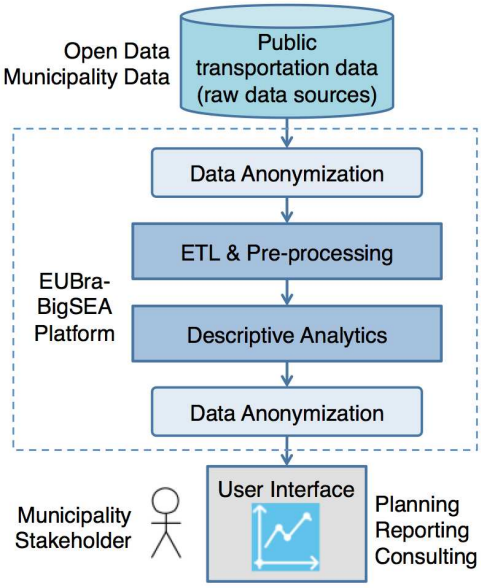


FIGURE 1. High-level workflow of the City Administration dashboard application.

both pre-processing and analytics steps are executed with parallel data analysis solutions joining High Performance Computing (HPC) paradigms and big data approaches. The aggregate data produced undergoes a second anonymization stage to address potential data privacy issues emerging after the integration and aggregation of multiple datasets. Finally, the output data can be accessed via web by the end-users (i.e., Municipality stakeholders) through a graphical user interface, for additional filtering (i.e., subsetting), manipulation (i.e., coarse-grained aggregations) and visualization of the different views.

C. TRANS-ATLANTIC TESTBED INFRASTRUCTURE

The application is a joint trans-Atlantic effort of various institutions from Europe and Brazil involved in the project. From an infrastructural point of view, as it can be seen in Fig. 2, the testbed infrastructure is deployed across multiple sites on both sides of the Atlantic. The pre-processing and the main application are executed through the EUBra-BIGSEA interoperable programming framework in a cloud-based environment deployed at the Polytechnic University of Valencia (UPV) in Spain, while the aggregate statistics (triggered by the application) are computed by a big data analytics framework deployed on a commodity cluster in a private cloud setting at the CMCC SuperComputing Centre (Italy). The final results are then transferred over a secure channel to the Federal University of Campina Grande (UFCG) in Brazil, hosting the web application of the City Administration Dashboard. The web application graphical user interface is consulted by the stakeholders from the Municipality of Curitiba (Brazil) via browser (subject to authentication).

It is worth mentioning that the need for a distributed environment comes from a scenario that combines data providers and computing resource providers, which may require mul-

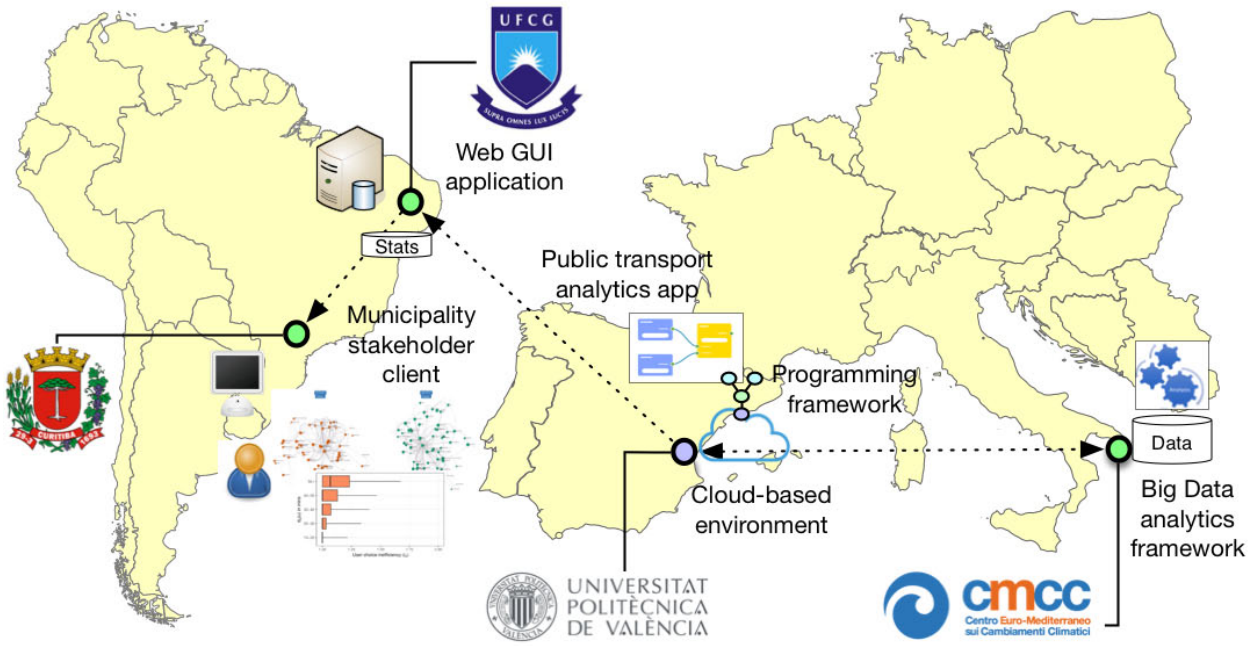


FIGURE 2. City Administration dashboard trans-Atlantic infrastructure.

tiple centres to cooperate. The use of a trans-Atlantic infrastructure was defined to demonstrate that the model is valid even in a geographically-wide scenario that reflects the actual collaboration.

III. APPLICATION DESIGN

This Section deals with the design aspects of the City Administration Dashboard application at the architecture (Section III-A) and data sources level (Section III-B).

A. APPLICATION ARCHITECTURE

The City Administration Dashboard is a descriptive analytics application that provides multiple aggregate statistical trends about bus usage. It handles a full big data pipeline that runs in a cloud infrastructure, starting from the input data sources down to the final visualization output, and it also takes into account data quality and privacy constraints.

To address the application purpose, multiple data sources have been exploited and several modules have been developed and consistently integrated with each other. Throughout the whole pipeline, data is processed at various levels, from level-0 (raw data at full resolution) to level-1 (processed data) and level-2 (refined products). Additional details about the data sources levels are provided in Section III-B.

Fig. 3 provides a detailed view of the main application building blocks, their interaction and the data flow within the application. Cyan blocks represent the different data sources (classified by level) exploited or produced by the application. Security and privacy modules/interfaces are reported in light colors.

In terms of input data, the application deals with a consistent set of heterogeneous data sources; some of them have been provided by the Municipality of Curitiba, whereas others are already publicly available with no restrictions, as open data (more details are provided in Section III-B).

The central block of the architecture includes the main processing components and services of the application. The PRIVAaaS (PRIVAcy as a Service) toolkit is used at the beginning of the processing to anonymize the most critical information on the input data before applying the other stages and, at the end, to enforce data privacy constraints on the aggregate data (Section V). The DQaaS (Data Quality as a Service) service performs data quality evaluation on the input data and annotates it with data quality metrics to identify noisy, inaccurate or misleading values, which can negatively affect the final outcome (Section VI-A). On the other hand, the EMaaS (Entity Matching as a Service) module runs entity-matching algorithms integrating and aligning multiple data sources in order to create higher quality integrated geospatial-temporal data (Section VI-B). Both modules are implemented as Spark applications, although EMaaS has also been implemented as a COMPSs application (Section VI-B1). The enriched level-1 data is then used by the descriptive analytics to compute a wide set of aggregated statistics (Section VII). It is a Python application built on top of the COMPSs programming framework to easily enable tasks parallelism and it relies on the Ophidia big data analytics API [16], [17] to address the analysis of parallel, multi-dimensional data, also known as *data cubes*. As a result, it provides level-2 data ready to be consumed by the web application running at UFCG.

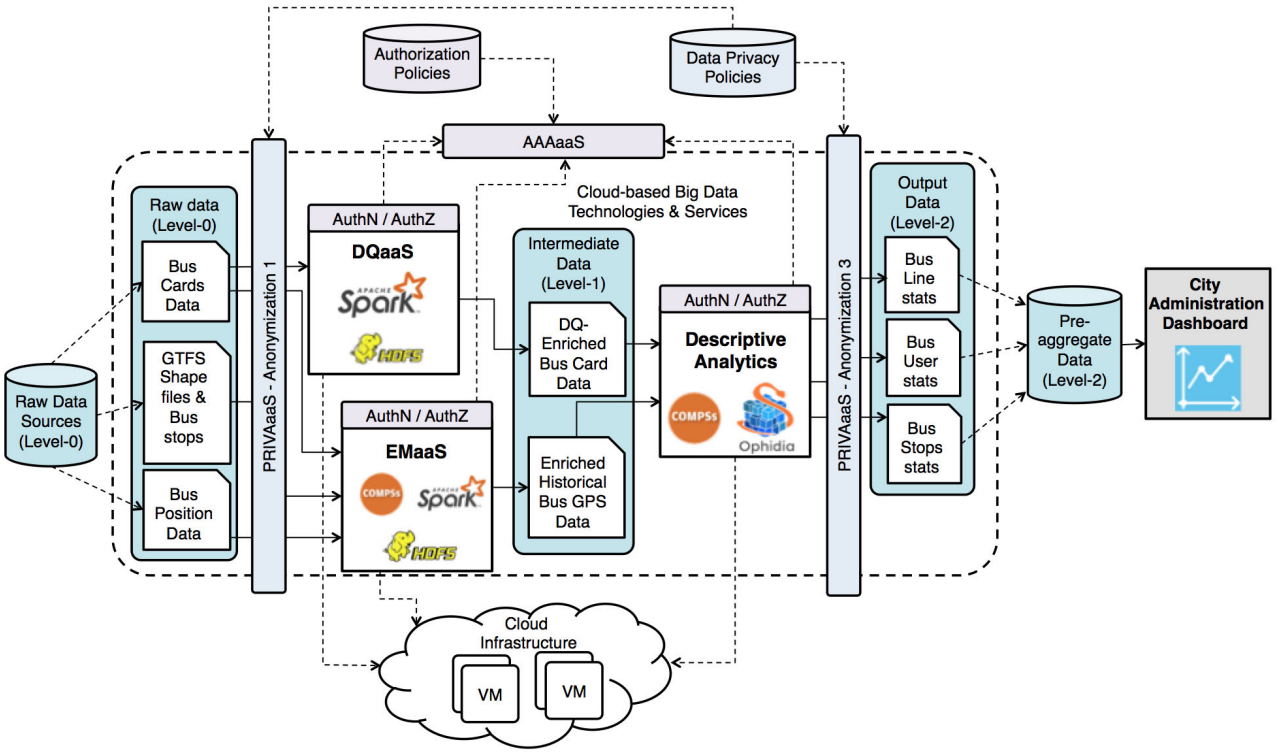


FIGURE 3. Architectural view of the City Administration dashboard application.

In this respect, the web application provides support for real-time exploration, filtering, visualization, and reporting of the statistics through multiple user-friendly and interactive web pages (Section VIII). It also allows stakeholders to perform user-defined/custom aggregations to address specific needs in a very flexible way. It is worth mentioning that, to address responsiveness, the web application caches pre-aggregated level-2 (coarse-grain) data calculated on top of (level-2 fine-grain) data provided by the central block.

The full application pipeline has been secured by design at all levels. To this end, security is provided by the EUBra-BIGSEA Authentication, Authorization and Accounting as a Service (AAAaaS) infrastructure described in Section IV-B.

From a technological standpoint, given the heterogeneous data and requirements involved, a wide set of big data components has been integrated for the development of the application. In particular, the final implementation includes the following technologies for big data management and parallel/distributed processing: *Apache Hadoop Distributed File System (HDFS), PostGIS, Apache Spark Streaming, Apache Spark, Ophidia* and *COMPSs*. Such tools provide the features to (i) handle the whole application pipeline, including, for instance, ingestion, streaming and batch processing, analytics, storage, access, filtering and (ii) support, among others, parallel computation, data partitioning/distribution, caching, and metadata management. All the technological components have been carefully identified, evaluated, partially adapted (when needed) and integrated with those developed by the project.

With respect to the *data storage* and *access*, HDFS, PostGIS and Ophidia have been used. In particular, HDFS has been used as raw storage for several blocks well integrated with the Apache eco-system of services (i.e., EMaaS and DQaaS), thus providing a seamless support to high-level services like Spark, while PostGIS has provided a consolidated technology for storing and querying geo-spatial data (required by EMaaS). Finally, Ophidia has been used as a native component for the On Line Analytical Processing (OLAP)-based analytics, mainly to support the descriptive analytics tasks. With regard to the streaming processing, Spark Streaming has been used in the real-time version of the EMaaS approaches.

As for *batch data analytics and mining*, Apache Spark and Ophidia (in combination with COMPSs) have been used to implement the parallel code of the main application blocks (i.e., DQaaS, EMaaS and the descriptive analytics). The two components provide complementary functionalities in terms of data mining and OLAP-based analytics; they also provide in-memory support, thus representing the fast (besides the big) data management components of the EUBra-BIGSEA platform.

The modules based on Spark and COMPSs run into a cloud environment at UPV exploiting Apache Mesos as cluster manager (Section IV-A), whereas Ophidia runs in a separate

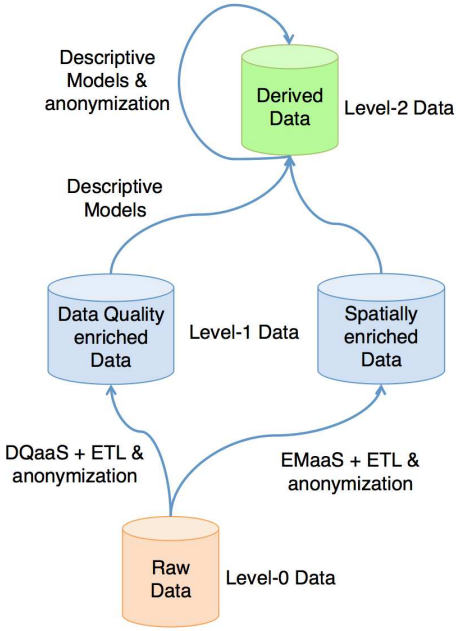


FIGURE 4. Data source levels (three-level hierarchy model).

setup at CMCC and interacts with COMPSs leveraging its server-side approach.

B. DATA SOURCES AND THREE-LEVEL HIERARCHY MODEL

In a real urban context, the analysis of multiple, interrelated, heterogeneous (big) datasets is necessary to distill bus usage and passengers habits knowledge, valuable for decision-making processes related to smart urban transportation. The raw data collected by the Municipality (e.g., electronic bus tickets monitoring) or measured from sensors (e.g., GPS with the bus position) can be too fine grained to be directly explored by the users (level-0, high-resolution); it can require preliminary pre-processing and aggregation stages to make the analysis feasible (level-1, intermediate results). Still, further knowledge can be inferred by interrelating the information from multiple data sources (level-2, refined products).

As a result, a three-level hierarchy model (see Fig. 4) has been defined to represent all the data sources managed/produced by the target application (from raw data to more refined data products).

More specifically:

- *Level-0 data*: it comprises the raw data from the data sources used by the application, which are related to urban traffic as well as to other static and dynamic information. In particular:
  - *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. The datasets used for stationary data came from both official (Instituto de Pesquisa e Planejamento Urbano de Curitiba - IPPUC,[2] the Municipality of Curitiba,[3] along with data from URBS[4]) and non official sources (Open Street Map and Google Maps). Stationary (or static) data might include geo-referenced data from bus lines, bus stops, terminals, and streets that do not change often.
  - *Dynamic spatial data*. This data contains georeferenced information about the vehicles and users, valid for a specific point in time. The datasets used for dynamic spatial data came from both official (URBS) and non official sources (Generic Transit Feed Specification - GTFS).[5] Dynamic data from URBS include data from bus lines, and user cards, transmitted at an average frequency of 5 minutes.
- *Level-1 data*: includes the intermediate, integrated and enriched data used by the application after the execution of the pre-processing steps (e.g., anonymization, ETL, Entity Matching and Data Quality services);
- *Level-2 data*: consists of the data derived from the integrated level-1 data. Such data is produced as a result of the descriptive models applied on level-1 data, and represents the output that is accessed by the application front-end for further aggregation and visualization (i.e., through graphical user interfaces). This level of transformation also includes the additional anonymization step (as shown by the recursive arrow).

It is worth mentioning that the proposed 3-level hierarchy model is not application-specific; it has in fact been designed to meet generality and simplicity principles, in order to work well with other applications too.

The following table (Table 1) provides an overview of the various datasets used or produced by the City Administration Dashboard application, classified according to the aforementioned levels. The datasets are in Comma-Separated Values (CSV) or JavaScript Object Notation (JSON) formats.

IV. CLOUD AND AAA INFRASTRUCTURES

Both the cloud and the Authentication, Authorization and Accounting (AAA) infrastructures represent two core and application-agnostic elements of the EUBra-BIGSEA platform. As it can be easily argued, they play a fundamental role for the proper implementation of the City Administration Dashboard application. The next two sections delve into the details of the cloud and AAA infrastructures.

[2]http://ippuc.org.br/geodownloads/geo.htm-Last visited on July 2019
[3]https://www.curitiba.pr.gov.br/dadosabertos/-Last visited on July 2019
[4]https://www.urbs.curitiba.pr.gov.br-Last visited on July 2019
[5]This specification was developed by Google and defines formats for files to be provided by an operator or authority to describe transit supply at three levels. https://developers.google.com/transit/- Last visited on July 2019

TABLE 1. Data managed/produced by the application. The colors in the tables rows refer to the data sources level as shown in Fig. 4.

Data Source Name	Level	Description	Data Format	Data Availability	Anonymization
Bus card data	0	Provides information about which user and when he boarded a particular bus. This data source comes from the AFC systems. AFC data contains at least a timestamp, card id, and vehicle id for each card tap	JSON	By request (sensitive information[6]). Usage restricted to the project	Not Applicable
Bus GPS position data	0	Provides information about the GPS position of the buses at different time steps	CSV	By request. Usage restricted to the project	Not Applicable
GTFS shape files	0	Provide the variations of a bus service (predefined trajectory) over a given route represented by shape linestrings	CSV	Public	Not Applicable
GTFS bus stop files	0	Provide the list of bus stops geo-positions (ordered by the sequence number of the bus stop) over a given predefined trajectory (shape)	CSV	Public	Not Applicable
DQ-enriched Bus Cards Data	1	Bus card data information enriched with some data quality fields produced by DQaaS algorithms	JSON	Generated by the DQaaS	Anonymization level 1
Enriched Historical Bus GPS Data	1	Provides the integration (association) of the bus GPS data (Bus GPS position data) with the predefined trajectory (GTFS shape files) and the boarding time of passengers based on the ticketing time (Bus card data)	CSV	Generated by the EMaaS	Not required
Bus line aggregate statistics	2	Provide statistical information regarding the number of passengers per bus line over various time ranges	CSV/JSON	Generated by the descriptive analytics module	Not required
Bus users aggregate statistics	2	Provide statistical information regarding the bus usage by each passenger over various time ranges	CSV/JSON	Generated by the descriptive analytics module	Anonymization level 3
Bus stops aggregate statistics	2	Provide statistical information regarding the number of passengers boarding from each bus stop over various time ranges	CSV/JSON	Generated by the descriptive analytics module	Not required

A. CLOUD INFRASTRUCTURE

The proposed application exploits a cloud infrastructure to host the different application components. In this respect, the processing backend includes an elastic Apache Mesos cluster. Mesos [18] is a convenient and large-scale resource management platform for the execution of applications embedded in containers. It provides support for Docker containers as well as generic applications running on Mesos native containers. Mesos abstracts processors, storage and special resources from a distributed infrastructure, offering those resources to executing frameworks, such as Spark, Hadoop, Marathon, Chronos. New frameworks can be developed through plugins.

Mesos is fault-tolerant and dynamically manages resources. Therefore, new resources can be added on the fly, being immediately available for new frameworks. Decommissioned resources are automatically removed, becoming unavailable for the active frameworks. This feature enables the development of elastic management services, which could power on and off resources that will bind to or unbind from the Mesos cluster. For this purpose, Elastic Compute Clusters (EC3) [19] is used, as it provides a service to manage the deployment, configuration and reconfiguration of self-managed elastic clusters. The EC3 plugin for Mesos detects resource starvation and powers on (and configures) enough resources to run the stalled jobs submitted through the registered frameworks. Initially, EC3 powers on only the front-end, deploying new working nodes as new jobs are submitted. The first working node is used to create a reference Virtual Machine Image (VMI) to be reused to deploy and reconfigure further working nodes, speeding up the process. Fig. 5 shows these details. Mesos includes a Container Network Interface overlay network that provides connectivity among the containers running on the distributed system, which are routed through a Domain Name System (DNS) service.

B. AAA INFRASTRUCTURE

Authentication, Authorization and Accounting is a popular and generalized designation to support services that require authentication, authorization and accounting features for accessing resources (e.g., network, processing, storage, data, applications, documents). While there is a wide range of

[6]Municipalities often keep some demographic information associated with a card too

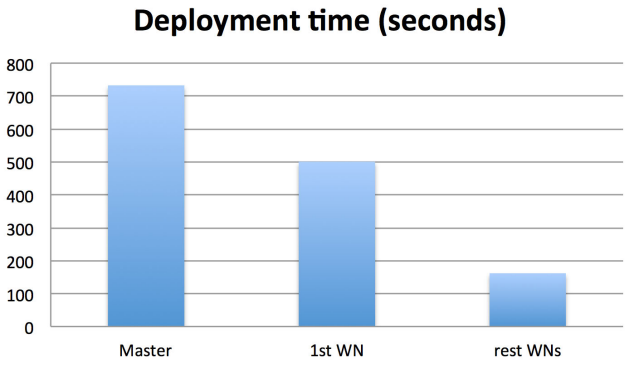


FIGURE 5. Deployment time of the different node types.

AAA protocols and implementations that correspond to a variety of usage scenarios and protected assets ([20]–[24]), they all share the same base concepts: AAAaaS features the support for traditional AAA and for Identity and Access Management (IAM), also allowing the integration with external identity providers. AAAaaS has been developed following three fundamental cloud principles, namely *scalability*, *elasticity* and *resilience*, and it is based on a RESTful service developed in Python and a MongoDB database for storing authorization policies and other required information. The CloudFlare Secure Socket Layer[7] (CFSSL) tool supports the generation and management of the certificates used for the communication between the RESTful service and the database, making all internal communication encrypted.

Through an API or via web pages, the solution provides: i) authentication (sign in, token verification, read user information, sign up, sign out, update user information, delete user accounts, change password, reset password, and resend account confirmation email), ii) authorization (create rules, update rule, show rule, delete rule, use resource), and iii) accounting and other features (traditional accounting, i.e., read accounting of a user, and also other actions such as creating email associations, reading email associations, and deleting email associations).

The AAA architecture has been designed in such a way that it can be easily maintained or further developed [10]. AAAaaS is provided as a containerized solution with three main elements: web server, web application container and database containers. The architecture represents the interaction among the containers. The front-end block is a Docker container with Nginx acting as a reverse proxy and redirecting all the traffic to the web application container back-end (Nginx provides load balancing and resilience capabilities). The web application container queries the database container represented by Data Storage. The use of containers allows several instances of the different components, thus providing a scalable solution. Also, it is possible to load the SSL certificate to ensure secure communications with all clients through HyperText Transfer Protocol Secure (HTTPS). By providing the location of the web application instances, the requests can be redirected according to the introduced settings (e.g., instance weights, least-connected or other settings).

The web application handles all the HTTP(S) requests made to the service. Every request is then validated using secure methodologies. For instance, passwords are encrypted with SALT functions. Passwords must fulfill three out of four conditions (e.g., minimum length, letters, numbers, capital), they cannot be the same as the user name, as well as other criteria. As mentioned before, queries to the database are secured with SSL certificates. The service is based on tokens, randomly issued at each sign-in session and with an expiry date that can be up to seven days (when the ''stay signed in'' option is checked). After the expiry date, tokens are no longer valid and a new sign in is required.

AAAaaS also provides an iAA (infrastructure Authentication and Authorization). The iAA deals with the authentication and authorization of infrastructure accesses instead of applications, services or end-users. It provides a graphical user interface and a RESTful API. The iAA module provides an end-point for Mesos agents or frameworks to authenticate themselves and gain clearance to access certain resources. The module acts as a middleware between Mesos agents or frameworks and the Mesos Master. It can be easily adapted to support different frameworks executed from the Command Line Interface (CLI) and allows changes (e.g., updates) to be made to the Mesos system without having a disruptive effect on the iAA process. The iAA control is carried out in accordance with the authorization mechanisms (i.e., credentials and Access Control Lists - ACLs) available on the Mesos Master. To achieve this, a mapping between the credentials created by the user and a set of credentials previously loaded on the Mesos Master is provided. This means a pair of Mesos credentials (principal and secret in the Mesos terminology) is assigned to each registered user in a completely transparent manner.

In the context of the City Administration Dashboard application, the AAAaaS has been exploited to support the Authentication and Authorization on the various services and models running with Spark (e.g., DQaaS and EMaaS) and Ophidia (descriptive models) on top of the EUBra-BIGSEA platform. In particular, each big data component/service involved in this application (i.e., Ophidia, EMaaS and DQaaS) has been developed or extended to check the validity of the authentication tokens and the authorization rules, exploiting the AAAaaS, before granting permission for the actual requested processing.

V. DATA PRIVACY

To guarantee data privacy protection, *PRIVAaaS* (*PRIVAcy as a Service*) has been integrated into the City Administration Dashboard. PRIVAaaS [13] is a software toolkit developed in the context of the EUBra-BIGSEA project. It provides a set of libraries and tools that allow controlling and reducing data leakage in the context of big data processing and, consequently, help protect sensitive information (e.g., names,

[7]https://www.cloudflare.com/ssl/-Last visited on July 2019

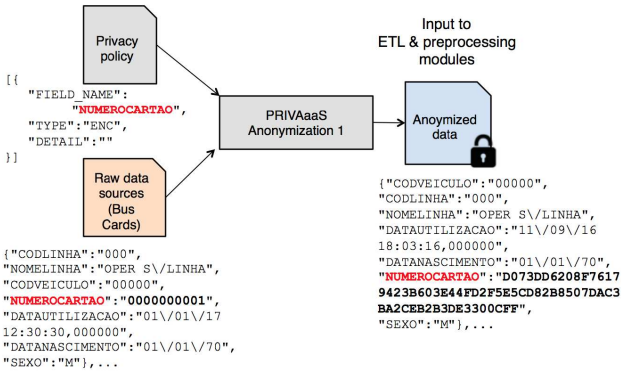


FIGURE 6. Detail of *PRIVAaaS Anonymization 1* process on bus card data.

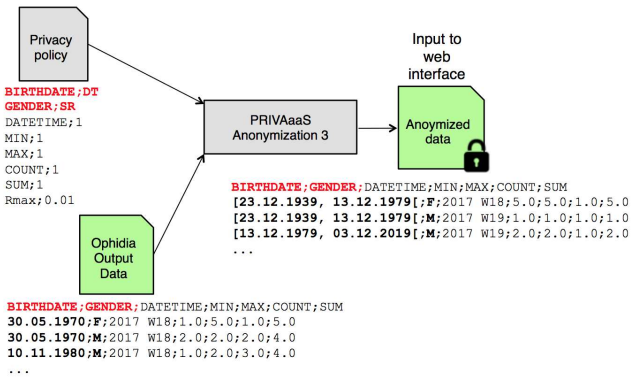


FIGURE 7. Detail of *PRIVAaaS Anonymization 3* process on descriptive model output data.

addresses, social security IDs, credit card numbers, etc.) that is processed by data analytics algorithms. PRIVAaaS is based on anonymization techniques (e.g., generalization, suppression, encryption, masking [25]) and anonymization policies [26]. It is a free and open source tool, developed in Java language and suitable for big data and cloud computing contexts.

As shown in Fig. 3, the PRIVAaaS component is used both on input (*Anonymization 1*) and output data containing statistical values (through risk re-identification process - *Anonymization 3* [27]).

The *Anonymization 1* phase is applied, in this case, on bus cards data. In this phase, the raw input data is updated with PRIVAaaS by applying a specific privacy policy and producing an intermediate anonymized version of the input data. In particular, the bus card identifier field - the unique ID number identifying the bus card user (referred to as *"NUMEROCARTAO"* in Fig. 6) - represents a sensitive attribute that must be updated by PRIVAaaS through the application of an anonymization technique (i.e., encryption). The other fields in the bus card data represent non-sensitive (or less sensitive) information, i.e., the bus line code (*"CODLINHA"*) and name (*"NOMELINHA"*), the vehicle identifier (*"CODVEICULO"*), the date and time when the bus card was used (*"DATAUTILIZACAO"*), the bus user birth date (*"DATANASCIMENTO"*) and gender (*"SEXO"*). In this case, the anonymization policy requires the encryption of this field in order to remove the direct reference to a specific user, while retaining the usage correspondence of the same bus card. Fig. 6 shows an example of how this kind of anonymization is performed in the application.

The output produced by the descriptive analytics block includes some aggregate data that could potentially be used to infer sensitive information. Hence, the PRIVAaaS (*Anonymization 3* phase) is applied to the output data of the descriptive analytics block to evaluate the re-identification risk of the dataset and, as necessary, to increase its level of anonymity, based on a risk threshold defined in the policy file [27]. In particular, some statistics contain information regarding the bus user's birth date and gender (represented as *"BIRTHDATE"* and *"GENDER"* in Fig. 7), which are considered sensitive fields (more specifically, quasi-identifier fields) and require proper anonymization techniques to reduce the risk of bus user's re-identification from the output. Under these circumstances, the *k*-anonymity algorithm [28] is applied over the quasi-identifiers fields, using the generalization technique for the user's birthdate and the suppression technique for the gender respectively (these techniques are defined in the policy). For this type of output, the suggested risk threshold lies within the range of 0.01 to 0.05. These values were obtained from the literature ([29], [30]). Fig. 7 shows an example of this anonymization process.

As it can be seen, the *"BIRTHDATE"* field values are replaced with a generalized version within a range, while the *"GENDER"* field has not been suppressed, since the risk threshold has already been reached by the *k*-anonymity application with the generalization of the birthdate field (which is considered a higher priority by the algorithm due to the higher variability of the field's values).

To sum up, in the target application, *Anonymization 1* is performed on input raw data during the ETL or pre-processing steps to anonymize sensitive attributes (e.g., the bus card identifier field), whereas *Anonymization 3* is executed on the output data produced by the applications to prevent values re-identification.

VI. DATA QUALITY-AWARE SERVICES: DQaaS AND EMaaS

In the City Administration Dashboard application, the first stage of the pipeline focuses on data quality-aware preparation and pre-processing. In particular, at this stage the analysis and annotation of the data sources is performed. The analysis aims to understand if the quality of the input data is suitable for obtaining high quality results. Data Quality (DQ) is, in fact, a fundamental ingredient for the effective exploitation of big data [31]. Data quantity can create a real value only if combined with data quality: *good decisions and actions are the results of correct, reliable and complete data* [32], [33]. In such a scenario, methods and techniques able to evaluate the quality of the available data are needed. Most of the literature contributions in this field are about structured data [34], so new algorithms have to be designed in order

to deal with novel requirements concerning variety as well as volume and velocity issues. In the proposed applications, such methods are provided by the following two *data quality-aware* modules:

- *Data Quality as a Service* (DQaaS, see Section VI-A): it is in charge of providing a descriptive view of data sources quality, with the aim of supporting the analytics applications in understanding which data is relevant and useful to be considered in more advanced analyses;
- *Entity Matching as a Service* (EMaaS, see Section VI-B): it supports data integration by providing approaches for entity matching management.

A. DATA QUALITY AS A SERVICE

Data Quality is often defined as *fitness for use*, i.e., the ability of a data collection to meet users' requirements [35]. It can be considered as a good starting point for filtering out non-significant information and improving the effectiveness of the results of processes and applications. In fact, especially in a big data scenario, not all data is relevant: *one of the fundamental difficulties is that extracted information can be biased, noisy, outdated, incorrect, misleading and thus unreliable* [36]. Poor data quality can prevent applications from exploiting the potential value of big data. Therefore, it is important to analyze and pre-process the provided data before using big data sources.

Data Quality is evaluated by means of different dimensions, whose definition mainly depends on the context of use [34]. Anyway, most studies focus on a small set of DQ dimensions that are considered relevant; such set includes: accuracy, completeness, timeliness, and consistency [34]. For structured data, Data Quality literature offers several contributions that propose assessment algorithms for these consolidated dimensions but big data pose new challenges [37], [38]: (i) in order to manage the increasing data volume and minimize the execution time of the assessment algorithms, new methods designed to exploit sampling and parallel computing are needed; (ii) the presence of heterogeneous sources (i.e., variety) requires the design of an adaptive system able to trigger the proper assessment algorithms on the basis of the data types and sources. In fact, not all the data quality dimensions are always computable: the set of DQ dimensions to consider depends on the type of source and the type of data.

Based on these requirements, the Data Quality service (DQaaS) designed within the scope of the EUBra-BIGSEA project is able to provide information about the quality of the analyzed big data sources. Quality metadata is provided to make users, analytics or data mining applications aware of the quality of input data and, in particular, to support the selection of relevant data. In fact, DQ is able to identify and eliminate *noises* that can affect data interpretation and the effectiveness of decision support systems.

1) DQaaS ARCHITECTURE

This section describes the components of the Data Quality service and how the assessment module will work also considering big data constraints. The proposed Data Quality service architecture is depicted in Fig. 8 [39].

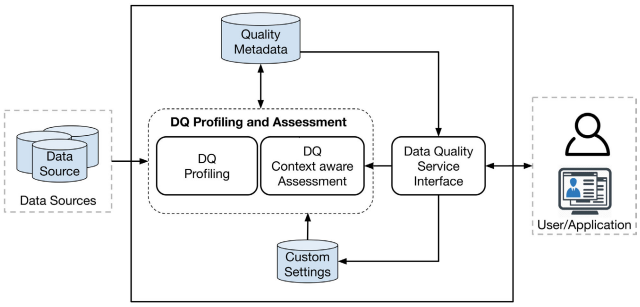


FIGURE 8. Data quality service architecture.

The core of the architecture is the DQ profiling and assessment module that consists of two main components: the *Profiling module* and the *Assessment module*.

The Profiling module is in charge of providing some metrics useful to *measure and monitor the general quality of a dataset* [40]. It contains a Source Analyzer service that takes a data source as input and, if feasible, defines its principal characteristics: list (i.e., source schema) and type of attributes. On the basis of this information, the source analyzer provides the list of data quality dimensions that can be evaluated. The DQ profiling module then collects statistics and information about data. In particular, on the basis of the input data source, it provides general quality metadata such as the volume (i.e., number of registrations or size of the source) and metadata associated with each attribute, such as number of null values, number of distinct values, number of duplicates, maximum, minimum, mean and standard deviation (only for numerical values). This profiling metadata is important in order to understand the content of the sources and it often reveals the presence of errors. For example, if an attribute age is associated with a maximum value equal to 150, it would be evident that the source contains an error. Again, the number of distinct values is a useful indicator for revealing errors especially if the respective attribute can only have values of a finite domain.

The Assessment module is in charge of computing Data Quality dimensions. It takes the data to analyze and the metadata provided by the Profiling module as input and then triggers the suitable assessment algorithms for the dimensions that can be evaluated by considering the type of input source and data. For this reason, it is called a context-aware Data Quality assessment.

The Assessment module periodically analyzes the entire source, calculating all the possible quality dimensions and storing the results in the quality metadata repository. In order to enrich the context for the assessment, it is possible to add the requirements of the users/applications that need to access the analyzed data sources. A Data Quality Service Interface allows users to submit a configuration file through which they can set the portion of data to analyze, the quality dimensions to consider and the granularity selected to perform the assessment. There are four types of granularity that can be

selected. *Data object granularity* level provides an array of aggregated quality values, where each value is associated with one dimension and expresses the quality level of the specific aspect (e.g., completeness, accuracy) for the entire dataset. The dataset can be both the entire dataset or a portion of data of interest for an application. *Attribute granularity* level, for which the assessment provides a matrix of aggregated quality values, where each cell reveals the aggregated quality value of one dimension for a specific attribute. *Value granularity* level, where the values of a selected attribute are used as grouping keys. The assessment module provides a set of aggregated quality values for each group. *Tuple granularity* level provides data quality values for each attribute.

As regards data quality dimensions, the data quality assessment module is able to evaluate the accuracy, completeness, consistency, distinctness, precision and timeliness, as also described in [39]. *Accuracy* is a measure of correctness: a value is considered correct if it belongs to the domain of values accepted for a specific attribute [41]. If a value is not correct, accuracy can be measured as its minimum closeness to a value of the considered domain. *Completeness* measures the degree with which a dataset is complete [35], by considering the amount of values currently available in the dataset and the expected amount of values. Note that the expected amount of values considers both null values in available registrations and missing registrations. *Consistency* refers to the violation of semantic rules defined over a set of data items [34]. In our approach, semantic rules are automatically derived from significant functional dependencies gathered by the Profiling module. *Distinctness* considers duplicates and measures the percentage of unique registrations or distinct attribute values in a dataset. *Precision* is calculated only for numerical attributes and can be defined as the degree of closeness among the subsequent values of an attribute. *Timeliness* is the degree with which values are temporally valid [35] and is calculated by considering the age and the frequency of changes for a specific value.

As better described in [42], the DQaaS is also enriched with a *DQ Adapter* that tunes the precision of the results according to the specification of the user. Note that Data Quality computation can be time expensive, especially if a high volume of data has to be analyzed. If the user needs fast responses, sampling techniques can be applied to address the velocity issues by selecting (through the adapter) a subset of the available data to provide a faster evaluation. Such evaluation will be clearly affected by a lower precision that has been defined as an index called *confidence*. In this way, the output of the DQ profiling and assessment modules is a set of metadata expressing a Data Quality evaluation of sources, sometimes coupled with a confidence value. This information is written in the Quality Metadata repository.

From an implementation standpoint, the DQaaS module has been built on top of Apache Spark running over a distributed Hadoop File System - HDFS. In the City Dashboard Administration application, the execution of Spark is handled by the underlying elastic Mesos infrastructure, described in IV-A.

2) DATA QUALITY ASSESSMENT - EXPERIMENTS

As described above, the DQaaS module is used to evaluate the quality of a data source with respect to data usage. This section illustrates the results obtained by analyzing the streaming data source considered in the municipality scenario, including both Bus Position Data and Bus User Card Data. The results for the Bus User Card Data are reported in the experiments. The fields describing the Birth Date and Sex of the user have not been considered for privacy issues. The goal of the evaluation is twofold:

- Providing information about the Data Quality of the streaming data sources, considering different levels of granularity. In particular, Data Quality details are provided both for the overall data source, computing a set of DQ metrics that can provide an overview of the data source, and at the tuple level. At the tuple level, a subset of the metrics is evaluated for each entry of the dataset, to provide information about the quality of the specific tuple. In particular, as shown in the following, the output is an extended version of the input source. The original schema is extended with the data quality attributes that can be calculated at the tuple level. In this way, each tuple is composed of its values and the respective quality attributes.
- Providing an evaluation of the performance of the Data Quality service in terms of the five DQ dimensions.

The experiments presented in this section concern the quality results of the analysis of all the data contained in both the available data sources. The fact that the entire dataset has been considered, guarantees the highest accuracy of the results. The next two subsections show the results of the experiments conducted on the biggest datasets.

3) DATA QUALITY PROFILING

As described in Section VI-A1, Data Quality Profiling consists of a set of activities that have to be executed the first time a data source is processed. It is used to extract some basic information from the data source that enables the following assessment step. Analyses have been executed on this source of data collected in May 2017, with a volume of approximately 1 GigaByte (GB). The first time the data source is uploaded, it is initially analysed to automatically detect the type of collected attributes. The results of this task are: type string for attributes *CODLINHA*, *CODVEICULO*, *NOMELINHA* and *NUMEROCARTAO*, and type datetime for *DATAUTILIZACAO*.

Since the Data Quality assessment depends on the type of attributes, the use of experts knowledge coded in a specific repository enables the association of the attributes with the quality dimensions that can be evaluated and the related granularity level (see Table 2).

TABLE 2. Data Quality dimensions that can be assessed on the considered source.

DQ dimensions	Data source/ Data object	Attribute	Value	Tuple
Accuracy	-	Yes, NUMEROCARTAO	-	-
Completeness	Yes	Yes, all attributes	Yes	Yes
Consistency	Yes	-	Yes	Yes
Distinctness	Yes	Yes, all attributes	Yes	-
Precision	-	Yes, NUMEROCARTAO	-	-
Timeliness	Yes	Yes, all attributes	Yes	Yes
Volume	Yes	Yes, all attributes	Yes	-

Note that the NUMEROCARTAO attribute was initially considered as a numerical type, thus leading the system to wrongly associate accuracy with precision. Here, a domain expert has been needed to check and correct the results. In this particular case, accuracy and precision have not been evaluated.

Finally, during the profiling phase, the automatic extraction of consistency rules between the attributes in the data source is also performed. In this phase, the values of the attributes are analysed to detect causal dependencies among them. In the considered data source, three consistency rules were detected: (i) CODLINHA $\rightarrow$ CODVEICULO, (ii) CODLINHA $\rightarrow$ NOMELINHA and (iii) CODLINHA, CODVEICULO $\rightarrow$ DATAUTILIZACAO. Rules state that a bus line is associated with the same vehicle and the same name, and that the combination of bus line and vehicle is associated with the same date (suggesting that for different dates this combination might change). Note that our evaluations have been performed by using a blocking approach that analyzes registrations day by day. This is the reason why our tool considered the first and the third rules as significant. Each day a specific vehicle is assigned to a specific bus line. As a final note, it is important to remark that data source profiling is performed only once.

4) DATA QUALITY ASSESSMENT

Data quality assessment has been performed by considering a specific configuration setup for the target application. In particular, such settings define: (i) which attributes of the data source is worth of consideration (CODLINHA; CODVEICULO; NUMEROCARTAO; DATAUTILIZACAO; NOMELINHA); (ii) the grouping attributes, that are the attributes that can be used for the ''value level'' granularity (CODLINHA, CODVEICULO; CODLINHA); (iii) a set of filters, by using the variables ''value intervals'' and ''selected values'' (for a comprehensive assessment no filters were defined); (iv) the list of dimensions to assess (completeness, consistency, distinctness, timeliness, volume); (v) the volatility value (i.e., the days a value is temporally valid for) to use for the assessment of the timeliness dimensions (in our example, a long period has been chosen since data is also used for historical analysis); (vi) the desired level of granularity for each selected attribute (global; attribute; value; tuple).

TABLE 3. Data source quality.

DQ dimension	Global Value
Completeness	0.99
Distinctness	1
Timeliness	0.57
Volume	7781637
Consistency	0.99


```
{
"CODLINHA":"811",
"CODVEICULO":"BA022",
"DATAUTILIZACAO":"30/04/17 14:04:55,000000",
"NOMELINHA":"SATURNO",
"NUMEROCARTAO":"D073DD6208F76179423B603E44FD2F5E5CD82B8507DAC
3BA2CEB2B3DE3300CFF",
"COMPLETENESS_MISSING":"1.0",
"TIMELINESS_DATAUTILIZACAO","0.55664500698",
"ASSOCIATION_CONSISTENCY":"1.0"
}
```


FIGURE 9. Enriched data source with data quality evaluation (in bold) per tuple.

Considering the source level granularity, Table 3 describes the results obtained from the Data Quality assessment of the Bus Users Card dataset at the source level.

As regards *completeness*, there were no missing values in the dataset but there were some missing registrations. Such missing registrations have been detected by considering historical values and estimating the frequency with which users were expected to enter the buses. The *distinctness* value reveals that no duplicates were included in the dataset. *Timeliness* values is the mean value of the timeliness associated with the considered registrations, whose amount is shown by the volume dimension. The evaluation of *consistency* has been executed for the rules extracted in the profiling phase. However, only the second rule is relevant for the whole dataset and it has been evaluated to assess the consistency value shown in Table 3. As discussed before, the other rules are only valid if a single block (one day of data) is considered, whereas they are not valid when the whole dataset is considered.

The assessment also generates an enriched dataset by adding information about quality to each tuple in the dataset. Not all metrics can be computed at the tuple level, and here the focus is on three different metrics: completeness, consistency, and timeliness. Completeness expresses if there are any missing values in the tuple. Consistency evaluates the validity of the consistency rules extracted in the profiling phase for the considered tuple. Finally, timeliness evaluates the temporal validity of the timestamp of the considered tuple. An extract of the resulting enriched dataset is shown in Fig. 9.

B. ENTITY MATCHING AS A SERVICE

In the EUBra-BIGSEA project, EMaaS has been developed to address important problems of data acquisition and descriptive models development. The data acquisition problem EMaaS deals with is the lack of accuracy and precision of official and non-official sources, which ends up producing incoherent information and unalignment of buildings, streets and bus stops. EMaaS can support the detection and measurement of matching problems when linking these data sources. Regarding the descriptive models, a fundamental abstraction of these models is a trajectory, i.e., 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. Building such 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. Thus, a couple of EMaaS approaches have also been developed to deal with trajectories matching and provide high-quality integrated geospatial-temporal training data, in order to support the predictive machine learning algorithms (predictive models) developed during the project. With respect to Table 1, the EMaaS considers the following data sources as input: (i) shapes and schedules for bus operation; (ii) automatic vehicle location, either instantaneous or historical; and (iii) automatic fare collection data informing when transit users boarded and sometimes left the vehicles.

1) THE EMaaS ARCHITECTURE

EMaaS is capable of performing efficient (data-intensive) matching tasks by means of the programming models developed in the context of EUBra-BIGSEA project. It includes the implementation of *BULMA* (BUs Line MAtching), *BULMA-RT* (Real-Time BUs Line MAtching) and *BUSTE* (BUs Stop Ticketing Estimation).

Briefly, BULMA has been developed to address the task of identifying bus trajectories from the sequences of noisy geospatial-temporal data sources. It performs the linkage between the bus GPS trajectories and their corresponding road segments on a digital map (i.e., predefined trajectories or shapes) running a batch processing. In this sense, BULMA is a novel unsupervised technique capable of matching a bus trajectory with the *correct* shape, considering the cases in which there are multiple shapes for the same route (usual cases in many Brazilian cities, e.g., Curitiba and São Paulo). Furthermore, BULMA is able to detect bus trajectory deviations and mark them in its output.

The Real-Time BULMA (BULMA-RT) challenge has been raised to support the predictive machine learning algorithms that must often be trained in short periods of time (e.g., every two hours). The objective is to answer the following question: *can we train and update predictive models in real-time mode using updated bus trajectory information?* Additionally, the objective can be extended to address other types of questions, such as: *can we use the BULMA output to identify bus trajectories and thus monitor bus trajectories in real-time?* Such questions have motivated the investigation of a new technique able to produce integrated geo-spatio-temporal data of bus trajectories by processing real-time streaming GPS data.

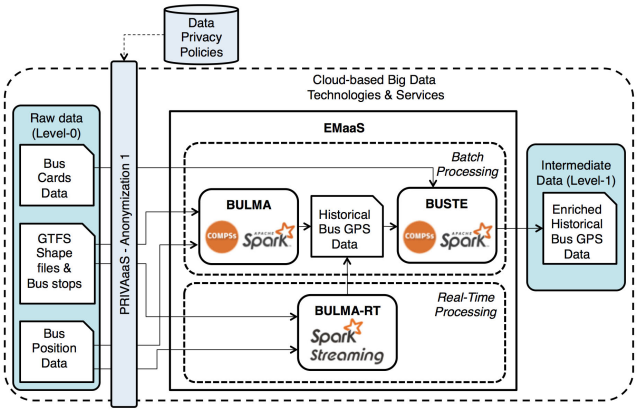


FIGURE 10. The proposed EMaaS Architecture to address the problem of bus trajectory matching and ticketing data integration.

The main difference between the BULMA and BULMA-RT is the Map-matching context. The input of BULMA contains all GPS information regarding the bus trajectories performed during the entire day. This historical context of bus trips enables a broader analysis of the trajectories performed by buses, such as the definition of complementary or circular trajectories. Unlike this scenario, the input of BULMA-RT only contains a few GPS geo-spatial points and the algorithm must identify the correct shape considering these small pieces of trajectory information. This means that, this time, the usage of the ending point of the trip is no longer possible. Consequently, the context information about the bus trajectories is no longer available.

In order to enrich the BULMA and BULMA-RT output, BUSTE (BUs Stop Ticketing Estimation) is used to perform a time interpolation over the shapes (based on BULMA output). Furthermore, BUSTE positions the bus stops over the interpolated shape and groups the boarding passengers according to each bus stop. In other words, the idea of BUSTE is to provide an estimate of the number of passengers boarding at each bus stop. BUSTE also provides rich and high-quality integrated geospatial-temporal data to support the City Municipality Dashboard application, see Fig. 3. Note that BUSTE receives the anonymized ticketing data as input and produces the enriched Historical Bus GPS data. This means that the BUSTE computation is not influenced by the presence of anonymized values in ticketing data. The BULMA and BUSTE architecture is depicted in Fig. 10. See [43], for more details about BULMA and BUSTE implementation.

As we can see (according to EMaaS architecture), BULMA and BUSTE approaches can be executed through a Spark or COMPSs job. The execution of BULMA-RT can be performed through a Spark Streaming job. BULMA output files are partitioned into n files assigned to BUSTE COMPSs

workers to be processed in parallel. As for the other components, the EMaaS is executed over the underlying elastic Mesos environment (see IV-A). The first step of BUSTE enriches the historical bus trips (generated by BULMA) by positioning the bus stops over the interpolated shape selected by BULMA. Afterwards, BUSTE groups the passengers boarding at each bus stop. Regarding the crowdedness prediction, i.e., a feature of the Bus Trip Recommender application, it is also generated based on historical bus GPS data (generated by the EMaaS approaches BULMA and BUSTE). The prediction model is trained using a state-of-the-art machine learning technique based on Spark over the BUSTE output. Thus, the trained predictive model is used to predict future trip duration and crowdedness.

2) EMaaS EVALUATION

Since the output of BULMA is the keystone of bus trajectories matching, we evaluate the *BULMA* technique against the *BoR-tech*, a strategy utilized in [44] regarding a critical factor in performance: the trade-off (relation) between the map-matching effectiveness (quality) and the efficiency (execution time) of each technique. The *BoR-tech* work presented in [44] looks similar to ours. It shows a method for inferring transit network topology from commonly available data feeds. It makes use of a Bag-of-Roads strategy, which is a sparse vector containing the number of road segments traversed by a bus b, where its i-th element denotes the frequency of bus b traversing the road segments. Then, it selects the top-k nearest predefined routes according to the Euclidean distance and Cosine similarity. Their results show good performance using the routing between the bus trajectory stops as a form of map-matching. In this respect, two real-world data sources have been utilized: the first dataset (DS-GPS) contains five days of GPS information (from 2016-10-30 to 2016-11-04) collected from the public transportation agency of Curitiba (Brazil). The second one (DS-shapes) contains all the shapes (i.e., the trajectories that a bus should follow) extracted from the GTFS, also provided by the public transportation agency of Curitiba. The DS-GPS and DS-shapes datasets have been utilized to study the performance of both techniques in terms of execution time.

For the evaluation of the map-matching effectiveness, we utilized a gold-standard data source. This ground-truth data source was manually (visually) labeled by a human data specialist and contains all the trips of nineteen routes performed by the buses on 2016-10-30. To measure the effectiveness, we applied three quality metrics: i) recall, which estimates the portion of correspondences that were correctly identified, denoted by

$$recall = \frac{TP(l_b)}{TP(l_b) + FN(l_b)}$$

where $TP(l_b)$ is the amount of correctly labeled GPS points (true positives) and $FN(l_b)$ is the amount of incorrectly labeled GPS points (false negatives). The recall value is included in the interval [0, 1], with higher values indicating better results.

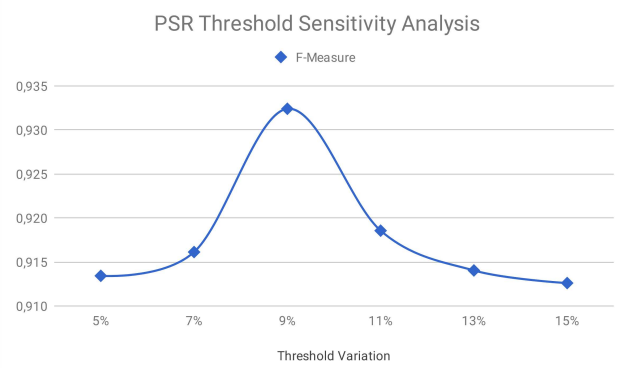


FIGURE 11. Sensitivity analysis of φ_{PSR}.

Before performing the map-matching effectiveness evaluation, we first needed to define the φ_{PSR} value, i.e., a threshold used to decide whether a GPS point is a start or finish point, to be utilized as a parameter of BULMA. To define that, we performed a sensitivity analysis using six different thresholds (from 0.05 to 0.15) over the gold standard data source, as depicted in Fig. 11. We only show this threshold range because a threshold value under 0.05 or above 0.15 would result in worse values of F-measure. As we can see, the φ_{PSR} value chosen was 0.09 due to the higher Map-matching effectiveness achieved (in terms of F-measure). Thus, all the experiments in this work utilized the φ_{PSR} value of 0.09.

Tables 4 and 5 show the map-matching effectiveness values of the two techniques. The evaluation is organized in two scenarios. The first one is related to the classification performed by the techniques regarding the two types of shapes (described in the VI-B1): complementary shapes (i.e., shapes that must join to other shapes to form a complete route) and circular shapes (i.e., shapes that describe a complete route by themselves), while the second one compares the execution of the two techniques over problematic routes existing in the DS-GPS dataset.

Table 4 shows the observed execution times along with the number of buses and F-measure collected values for the execution of the two techniques over common routes existing in the DS-GPS. As we can see, in all cases, none of the approaches achieved the (highest) F-measure of 1.0. The main reason for this result is related to the significant amount of sparse and missing GPS data within a few bus trips. Such situation leads the techniques to make erroneous decisions during the detection of the start and finish points. Also, note that, in Table 4, *BoR-tech* outperforms *BULMA* in terms of execution time due to the lack of more complex mechanisms to deal with the problem of multiple shapes describing the same route. However, in all cases, *BULMA* outperforms *BoR-tech* in terms of map-matching effectiveness. The main reason for that, regarding the F-measure of complementary and circular shapes, is its capability to analyze all the trips performed by a bus during a day. Since *BULMA* selects the best sequence

TABLE 4. Comparative table of common cases.

SHAPE TYPES/METRICS	COMPLEMENTARY		CIRCULAR		OVERALL	
TECHNIQUE	BULMA	BoR-tech	BULMA	BoR-tech	BULMA	BoR-tech
NUMBER OF BUSES	9	9	6	6	15	15
NUMBER OF TRIPS	147	147	68	68	215	215
EXECUTION TIME (s)	21	**13**	19	**13**	25	**17**
F-MEASURE	**0.98**	0.94	**0.87**	0.26	**0.94**	0.70

TABLE 5. Comparative table of noisy, missing and sparse GPS data.

SHAPE TYPES/METRICS	PROBLEMATIC		OVERALL	
TECHNIQUE	BULMA	BoR-tech	BULMA	BoR-tech
NUMBER OF BUSES	4	4	19	19
NUMBER OF TRIPS	69	69	284	284
EXECUTION TIME (s)	17	**11**	28	**19**
F-MEASURE	0.89	**0.91**	**0.93**	0.74

of shapes associated with the entire trajectory performed by a bus during a day, it is able to optimize the "best fit" sequence of shapes according to the trajectory performed by the bus. For this reason, in the case of typical complementary shapes, *BULMA* (0.98) slightly outperforms *BoR-tech* (0.94) in terms of F-measure.

However, the superior performance of *BULMA* is highlighted by the map-matching of circular shapes in Table 4. Note that *BULMA* achieved a F-measure of 0.87 against 0.26 of *BoR-tech* regarding typical circular routes. This result shows the lack of robustness of *BoRtech* in detecting the correct shape among multiple shapes that refer to the same route. The overall results (shown in the third column of Table 4) present the performance values for the execution of the techniques over all the common complementary and circular shape types. Table 5 shows the observed execution times along with the number of buses and F-measure collected values for the execution of the two techniques over problematic routes existing in the DS-GPS dataset. The status of problematic is defined by *BULMA* and *BoR-tech* when it is impossible to properly detect the start point (start or finish points in the case of *BULMA*). This issue can occur due to the presence of noisy, missing or sparse GPS data or the usage of an insufficient threshold (to determine the start and finish points). Thus, all trajectories with such detection restriction for both approaches are marked as problematic. Table 5 shows the results regarding the execution of both techniques over four problematic routes (with noisy, missing or sparse GPS data).

VII. DESCRIPTIVE ANALYTICS

The descriptive analytics module represents a core component of the application; it is in charge of computing the aggregated statistics (level-2 data) delivered to the web GUI for final end-user access and visualization. The application runs through COMPSs on top of a cloud infrastructure hosting an Apache Mesos cluster (see Section IV-A) and exploits Ophidia to enable in-memory, parallel data analytics tasks. The next two subsections delve into the details of the descriptive analytics application (Section VII-A) and cloud-enabled runtime execution (Section VII-B).

A. SOFTWARE IMPLEMENTATION

The descriptive analytics module starts from the set of level-1 data produced by the data quality-aware services and computes a set of statistics regarding:

- *bus users*, e.g., the number of buses each user (anonymized) took in a given time range;
- *bus lines*, e.g., the number of passengers on a given bus line in a given time range;
- *bus stops*, e.g., the number of passengers boarding at each bus stop in a given time range.

From an implementation standpoint, the descriptive analytics module has been developed as a Python application using COMPSs and Ophidia at two different levels: COMPSs as the programming framework and runtime execution engine and Ophidia for the parallel I/O and data analytics tasks. The Python language has been chosen as it is largely used within the Data Science community and there is a huge Python eco-system of libraries for data analysis already available. With respect to the target application, the Python bindings provided by the two frameworks through PyCOMPSs[8] and PyOphidia[9] have been exploited.

As shown in Fig. 12, the module implements three steps: (i) an additional preprocessing stage (ETL) performed on the level-1 data, (ii) the descriptive model to compute the aggregate stats (eventually pre-filtering data based on DQ constraints) and (iii) a final anonymization stage. More in detail:

- Step 1 performs the ETL to: (i) extract the data from the CSV or JSON files produced by the DQaaS and

[8]https://pypi.org/project/pycompss/ - Last visited on July 2019
[9]https://anaconda.org/conda-forge/pyophidia - Last visited on July 2019

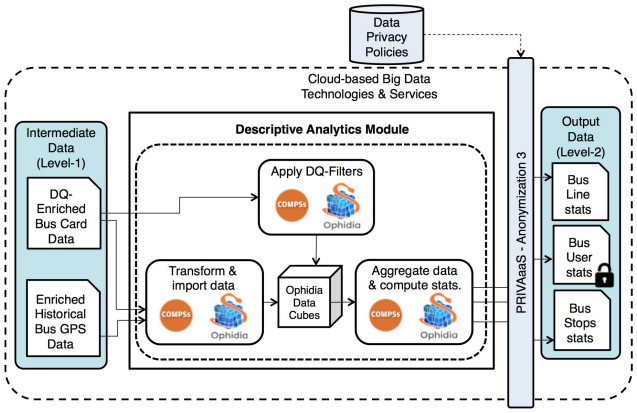


FIGURE 12. Descriptive analytics module internals.

the EMaaS, (ii) transform the data format and structure to speed up the ingestion in Ophidia and (iii) import the data (and metadata) into the Ophidia in-memory data store [45]. To speed up the process, these steps are performed on multiple input files in parallel. Data ingestion relies on the Ophidia parallel I/O support; data partitions make use of the Ophidia distributed storage model;

- Step 2 performs the actual computation of the aggregate statistics by running a set of parallel operators at the level of the Ophidia framework, to properly manipulate (e.g., slide/dice, reduce, transform) the data (cubes) ingested during Step 1. Additionally, before computing the statistics, some data can be filtered out by taking into account data quality constraints related to *consistency*, *completeness* and *timeliness* (as defined in previous section VI-A1). These constraints are dynamically applied to the imported data, thus making possible the statistics computation under different data quality constraints. At the end of the in-memory computation, the final results are stored in the output file;
- Step 3 applies the Anonymization (level 3) to the output file produced by the descriptive model. This stage produces a new file, by applying the *k*-anonymity algorithm described in Section V to some sensitive information.

Concerning the computation of the aggregate statistics, several algorithms have been implemented according to the type of metrics to be computed; in particular, the following Ophidia operators have mainly been used: data subsetting, time aggregation (to statistical values), dimension reduction, data masking and data import/export. As stated at the beginning of the section, multiple types of aggregation on both the *time* dimension (e.g., hourly, daily, weekly or monthly, but also on specific weekdays or groups of related weekdays - like weekends) and the *bus*-related dimensions (e.g., on the whole set of bus lines/stops/users to show an aggregate view of the entire bus transportation system) are implemented. For each combination of aggregations, various statistics are computed on the defined temporal aggregation (e.g., minimum,

```
1  #Ophidia metrics implementation
2  def totalAggregation(startCube, metric, nCores,
       user, pwd, host, port):
3    cube.Cube.setclient(username=user,
       password=pwd, server=host, port=port)
4    reducedCube =
       startCube.reduce(group_size='all',
       operation=metric, ncores=nCores)
5    data = reducedCube.export_array(show_time='yes')
6    return data
7
8  #COMPSs Task implementation
9  @task(startCube=IN, metric=IN, nCores=IN,
       user=IN, pwd=IN, host=IN, port=IN,
       returns=dict)
10 def compssTotAggregation(startCube, metric,
       nCores, user, pwd, host, port):
11   return internal.totalAggregation(startCube,
       metric, nCores, user, pwd, host, port)
12
13 ...
14 #Main code of the application
15 subsettedCube =
       startCube.subset2(subset_dims='time',
       subset_filter=filter_list, time_filter='no',
       ncores=nCores)
16 for i, m in enumerate(METRICS_BUS):
17   cubeList[idx][i] =
       compssTotAggregation(subsettedCube, m,
       nCores, user, pwd, host, port, mode)
18
19 cubeList = compss_wait_on(cubeList)
20 ...
```

Listing 1. Python code snippet about one type of aggregate metrics computation.

maximum, total or average number of passengers boarding at a bus stop/using the bus line).

These steps are computed concurrently on various blocks of data, e.g., for different time periods or metrics, through COMPSs, which transparently makes different tasks of the application run in parallel through its runtime engine, by taking advantage of the inherent code parallelism. As it can be seen in lines 17 and 9-11 of the code in Listing 1, this is achieved very easily using the COMPSs `@task` decorator on the `totalAggregation` function. At run time, COMPSs detects the multiple calls to the function (based on the number of cycles in the loop in line 16) and schedules concurrent instances of the task, also taking care of data dependencies among tasks.

Each task executed by COMPSs, in turn, triggers the remote execution of parallel data analytics operators in the Ophidia server, through PyOphidia, as shown in lines 4-5 of Listing 1, where a data reduction task with `nCores` number of cores is performed. As it can be clearly argued from Listing 1, the adopted approach has two major advantages: (i) an easier development of big data applications, enabling fast prototyping of applications thanks to the available high-level libraries and (ii) the ability to transparently handle inter-task and intra-task parallelism, as well as managing tasks dependencies over distributed computational/storage resources. In our tests, the COMPSs computational tasks have been run in the Mesos cluster at UPV while the data analytics operators have been executed in the Ophidia server at CMCC.

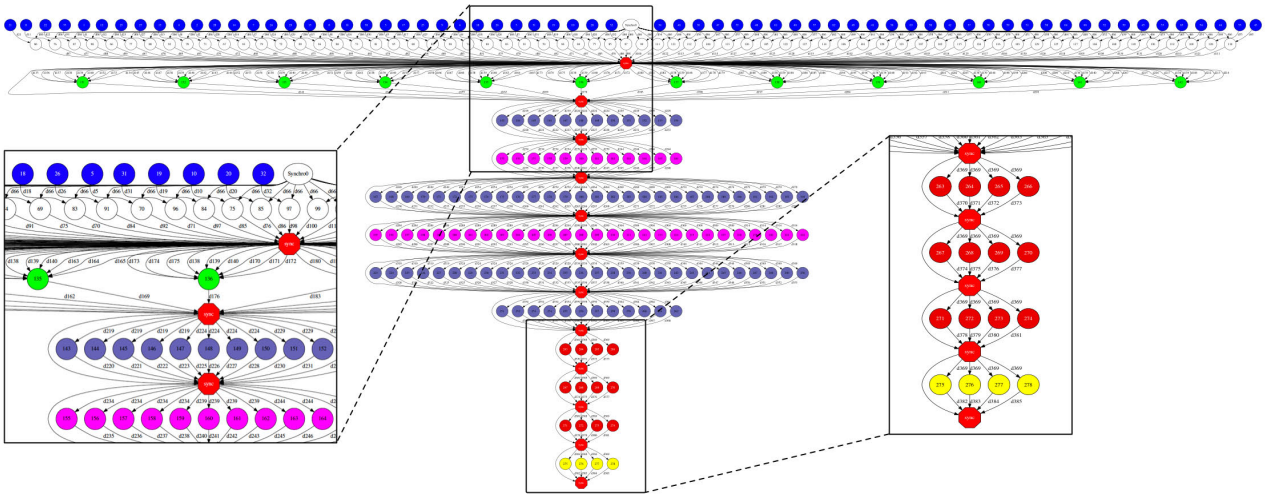


FIGURE 13. Execution graph of the descriptive analytics block of the City Administration Dashboard from the COMPSs perspective.

B. CLOUD-ENABLED RUNTIME EXECUTION

Fig. 13 shows the complete execution graph of the descriptive analytics block from the COMPSs runtime perspective. Each circle represents a COMPSs task executing a block of code with one or several instructions. For the sake of understanding, the graph is related to the application execution on a subset of input files related to 3 months, i.e., 65 files. In a real setting, several days could be missing for different reasons, i.e., AFC network downtime, AFC on-board unit malfunctioning. In fact, each blue circle at the top of the graph is associated with one of the input files; in particular, it refers to the pre-processing stages. Then, the extraction (white circles) and transformation (green circles) phases of the ETL procedure are executed. The following 10 stages (purple, magenta, red and yellow circles) refer to the computation of the different aggregate statistics, where each circle represents the sequence of Ophidia operators required to compute a specific metrics. The number of tasks per level defines the maximum degree of parallelism that can be achieved with respect to the specific type of computed statistics. For example, in the last levels, only four tasks are required since the four metrics (i.e., max, min, avg, count) have to be computed on the full time range, whereas in the middle layer 28 tasks can be executed in parallel by running the computation of the four metrics on each weekday (e.g., Monday, Tuesday, etc.). It is important to mention that the circles at the same level can be executed concurrently, based on the cloud resources availability, whereas the different levels are executed sequentially due to the inherent input/output dependencies in the algorithm (the red hexagons in the graph). Moreover, most of the tasks associated with Ophidia are executed in parallel. As a result, such two-level parallelism strategy allows running multiple parallel jobs concurrently, while also exploiting (at a finer level) data-level parallelism to speed up I/O and analysis.

From an infrastructural point of view, it is worth noting that the COMPSs runtime is executed as a Mesos Framework [14] at UPV. COMPSs processes are embedded as Docker containers, which run on the Mesos resources allocated to the COMPSs framework. COMPSs requests a different amount of resources depending on the concurrency degree of the executing graph of the application. In turn, EC3 manages the Mesos underlying infrastructure (i.e., VMI) transparently and dynamically (see Section IV-A).

Fig. 14 depicts the deployment of COMPSs workers on a set of Mesos agents, each one running a Docker container with two cores (in the picture, using the Mesos terminology, *tasks* represent resources assigned to a framework). The lower part of the picture shows the release of resources by COMPSs related to completed tasks.

VIII. USER INTERFACE

The user interface for the City Administration Dashboard is composed of a set of interactive dashboards and reports. The dashboard allows decision makers to monitor Public Transit with continuously updated information about deviations from schedule, extra and missing bus trips, boarding per time of day, ticketing share by company, total number of passengers per bus line or month, day or hour of the day, etc. The visualizations that compose the dashboard leverage visual analytics techniques to encode information so that it is easy to detect meaningful patterns, such as delays in the start or end time of a trip and trips that take longer than expected, as depicted in Fig. 15a. The data aggregation level shown in the interface can be interactively defined by the user, to drill down or summarize statistics over bus lines, days and periods of a day. It starts from level-2 data provided by the descriptive analytics module, although some level-1 data may also be directly exploited for the report production. There is also another visualization that aggregates the boarding events per company, characterizing the global ticketing share in the system, as shown in Fig. 15b.

Additionally, as depicted in Fig. 16, other views provide (a) hourly, daily, weekly and monthly aggregations for each bus line (to compare the overall boarding across the

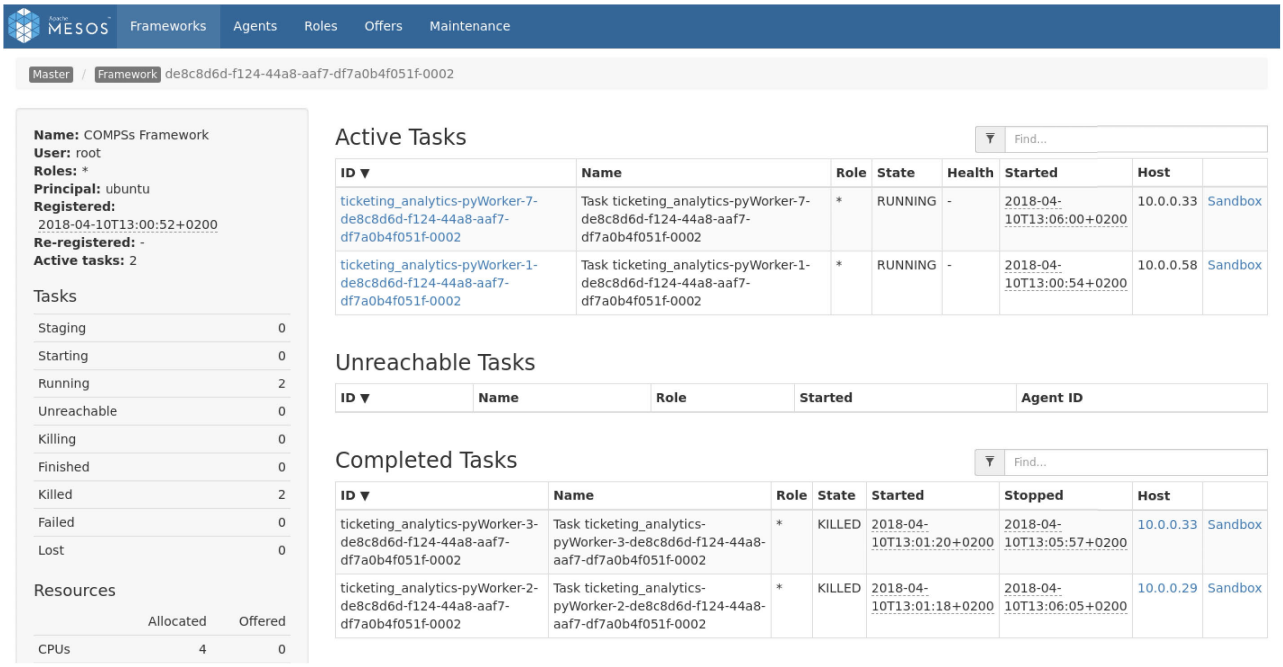


FIGURE 14. Deployment of COMPSs workers in Mesos and dynamic release of resources (completed tasks).

FIGURE 15. City Administration dashboard visualizations: (a) route trip analysis (deviation from schedule) and boarding stats; (b) ticketing share by company.

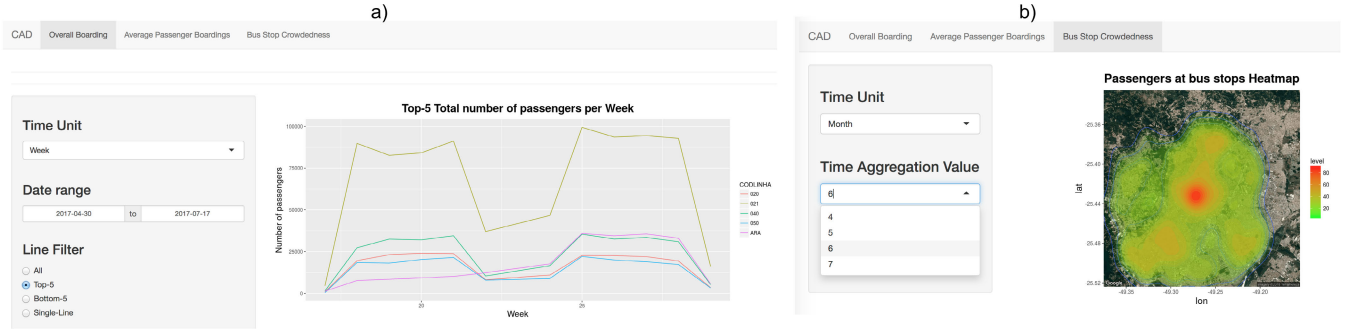


FIGURE 16. (a) Statistics on total number of passengers per week over the Top-5 bus lines (b) and heatmap regarding the distribution of boardings at bus stops.

different bus lines) or hourly and daily aggregations for each weekday on all lines (useful to identify the bus peak hours) and (b) heat maps regarding the distribution of boardings at bus stops. Still, filters over specific time frames or bus lines can be dynamically applied jointly with further aggregation levels to get additional insights into particular sets of data.

Besides these live interactive views of the data, the City Administration Dashboard also provides analytics reports (with aggregate metrics over longer periods) that are periodically produced and made available to the stakeholders. Examples of such reports are shown in Fig. 17. The People paths report (a) examines the city neighborhoods as nodes in a network, where there is an edge between two nodes if there was a minimum number of trips between these neighborhoods. This network allows the municipality to detect neighborhoods that act as hubs in a period of time, and relationships between groups of neighborhoods as communities in the graph. The second report (Fig. 17b) examines the degree with which passengers take trips that are suboptimal with respect to time in the Public Transit system. A trip is suboptimal if there was another trip with the same start and end points that was shorter and started around the same time of the one taken. Such trips indicate that passengers are not able to navigate and plan their daily travels efficiently. Finally, the third report (Fig. 17c) provides an *Origin-Destination matrix* (OD-Matrix) [46] for Public Transport, a common tool for city and transport planners, which aggregates the estimates of trips duration between each pair of sectors in the city. The creation of the OD-Matrix available through the City Administration Dashboard is fully automated.

IX. STATE OF ART

This section presents a comprehensive state of the art analysis with respect to two different major dimensions of the proposed work: *big data infrastructures* (Section IX-A) and *public transport applications issues* (Section IX-B).

A. SOLUTIONS FOR BIG AND FAST DATA MANAGEMENT

As mentioned in Section III, given the heterogeneity of data sources and the types of computation and functionalities required by the City Administration Dashboard application, various data management technologies have been taken into consideration during the implementation. Since the big data landscape provides a wide set of Open Source technologies, a sound evaluation has been performed on three different classes of systems (data storage and management Section IX-A1, data analytics and mining Section IX-A2, data streaming Section IX-A3) to identity those capable of better addressing the application requirements.

1) DATA STORAGE AND MANAGEMENT

As regards data storage and management, some well-known solutions, both non-relational (NoSQL - Non-Structured Query Language) and relational, include the Apache Hadoop Distributed File System (HDFS), MongoDB, PostGIS or Apache HBase. HDFS [47], [48] is a high-throughput, fault tolerant, distributed file system based on the Google File System (GFS) [49]. HDFS, together with YARN and MapReduce, is one of the core modules of the Apache Hadoop project.[10] MongoDB[11] is a multi-platform, document-oriented database providing high performance, high availability and scalability. It works on the concept of document collections and provides a flexible storage architecture. PostGIS[12] is an open source spatial database extension for PostgreSQL Object-Relational Database Management System (ORDBMS) that adds support for geographic objects and follows the Open Geospatial Consortium's "Simple Features for SQL Specification" [50]. It provides several features such as processing of vector and raster data, spatial reprojection, import/export of Environmental Systems Research Institute (ESRI) shapefiles, 3D object support. Finally, Apache HBase[13] is an open-source, distributed, versioned, non-relational database modeled after Google's "Bigtable: A Distributed Storage System for Structured Data" [51]. Similarly to Bigtable, which leverages the distributed data storage provided by GFS, Apache HBase provides Bigtable-like capabilities on top of Hadoop and

[10] http://hadoop.apache.org - Last visited on July 2019
[11] https://www.mongodb.com/ - Last visited on July 2019
[12] https://postgis.net - Last visited on July 2019
[13] https://hbase.apache.org - Last visited on July 2019

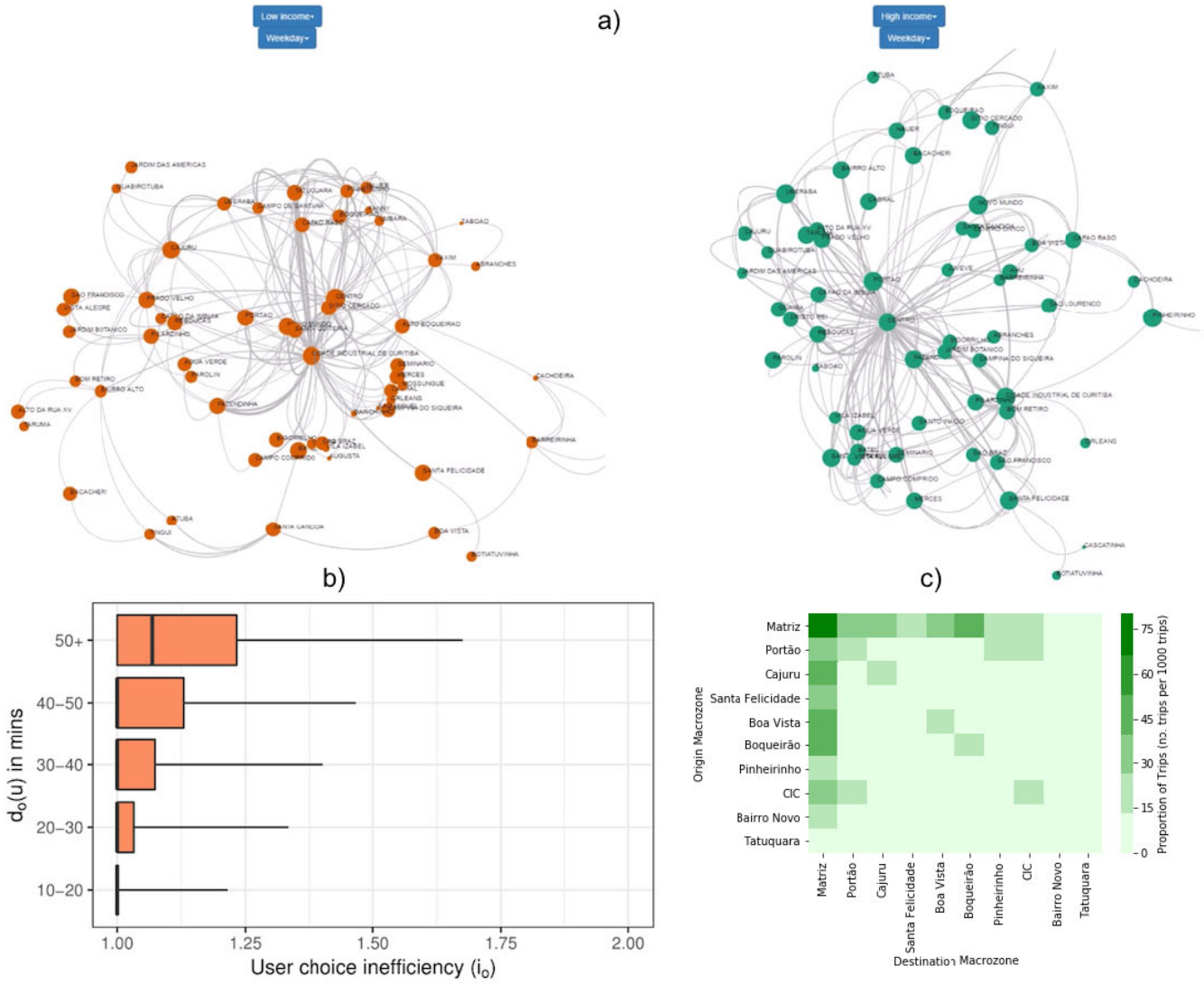


FIGURE 17. Periodically-constructed analytics reports: (a) People paths; (b) User-choice inefficiency analysis; (c) Origin-matrix estimation.

HDFS. As it can be easily argued, there are no one-size-fits-all solutions, which motivates the need for a technology ecosystem rather than a single service or component. In fact, while PostGIS provides strong support for Geographic Information System (GIS)-based data, HDFS is thoroughly integrated with several other technologies in the Apache eco-system. The other technologies have not been adopted since document-oriented or columnar-store solutions were not required.

2) DATA ANALYTICS AND MINING

Concerning data analytics and mining, a variety of big data-enabled solutions have been evaluated. Among these, Apache Spark [52] is a well-known, fast and general-purpose cluster computing framework, based on the abstraction of Resilient Distributed Dataset (RDD), a distributed, fault-tolerant, read-only collections of (in-memory) objects [53], for large-scale data processing.[14] Apache Hadoop MapReduce is an implementation of the MapReduce programming model [54] that allows the processing of massive data in a parallel and distributed fashion on computer clusters. It is one of the main components of the Hadoop framework. Another such solution, Apache Hive[15] [55] is a data warehouse software that facilitates reading, writing and managing large datasets that reside in distributed storage through SQL. It is built on top of Hadoop and provides tools to enable easy access to data via SQL, access to files stored directly in HDFS or in other storage systems like Apache HBase, and query execution via various frameworks. Instead, Apache Kylin[16] is an open source distributed analytics engine designed to provide SQL interface and multidimensional

[14]http://spark.apache.org - Last visited on July 2019
[15]https://hive.apache.org - Last visited on July 2019
[16]http://kylin.apache.org - Last visited on July 2019

analysis (OLAP) on top of Hadoop supporting large datasets. Similarly, Druid[17] [56] is another open source data store designed for OLAP queries. It provides a custom language with operations to load, index, query and group (roll-up data) and a fault-tolerant architecture with data partitioning and replication. Finally, Ophidia [17], [57] is a data analytics framework for scientific data.[18] It provides a complete environment for the execution of data-intensive analysis through parallel computing techniques, smart data distribution and in-memory I/O. It exploits an array-based storage model and a hierarchical storage organization to partition and distribute multidimensional scientific datasets over multiple nodes. Among all these technologies, Ophidia and Spark have been considered for the implementation of the application due to their complementarity in dealing with data mining and OLAP-based analysis. Apache Hive was not selected since its native support would have required to run on top of Hadoop which, as reported in literature, does not address performance in the proper way; the support to run on top of Spark has been added after the initial application design. In this regard, it has been demonstrated in various papers that Spark outperforms Hadoop in a very large set of use cases [52], [58], [59]. Furthermore, even though Hive represents the SQL interface on top of the Hadoop stack, Spark also provides an integrated support for relational queries through the Spark SQL module [60], which can also exploit HiveQL. From the OLAP perspective, Druid, Apache Kylin and Ophidia are all able to run OLAP analytics. Similarly to Hive though, Apache Kylin originally used Hadoop MapReduce to create cubes from data, while the support for fast cubing with Spark has only been added after the design of the application (during 2017 with Kylin v2.0 [61]). On the other hand, Druid shares some similarities with Ophidia since it is an OLAP-based system with a main memory database, capable of handling time series data and providing aggregation queries [56]. However, Druid is a column-oriented data store primarily used for applications on event data, whereas Ophidia implements a multi-dimensional, array-based data store mainly addressing scientific data needs [16]. Hence, Ophidia has been selected mainly due to (i) the native support for both scientific and array-based data jointly with parallel Message Passing Interface (MPI)-based operators, which provides an optimized analytics engine for multidimensional datasets, (ii) the native support for fast in-memory data analysis and (iii) the wide range of data analytics operators (about 50) and primitives (about 100) well suited for the statistical computation required by the application.

3) DATA STREAMING

In terms of streaming data solutions, Apache Kafka[19] [62] is a distributed, partitioned and fault-tolerant publish-subscribe messaging system that can handle hundreds of megabytes of reads/writes per second with low latency. Its scalable design allows streams to be partitioned over a cluster of multiple nodes. Another solution in this field, Apache Storm[20] [63], provides a distributed real time computing system which allows the reliable processing of unbounded streams of data and provides inherent parallelism to process high throughputs of messages with very low latency. It can be very easily coupled with Apache Kafka to ingest and process streams of data. Apache Flink[21] [64] is an open source platform for distributed stream and batch data processing and ''it has native support for iterations, incremental iterations and programs consisting of large Directed Acyclic Graphs (DAGs) of operations'. Finally, Apache Spark Streaming,[22] part of Apache Spark, is an execution framework that also allows processing streams of data with high scalability, throughput, and fault tolerance. Data streams can be absorbed from many sources and then processed by the Spark engine. In particular, Spark streaming has been selected due to a stronger and seamless integration with Apache Spark and its data analytics functionalities. In fact, as described in [65], which provides a comparison of technologies for machine learning from the Hadoop ecosystem, Storm does not come with a native machine learning library and must rely on external libraries, such as Apache SAMOA,[23] whereas Spark Streaming can rely on Spark native MLlib. Apache Flink can also exploit Apache SAMOA for machine learning and, additionally, it provides its own machine learning library, although it is a more recent effort and does not support the same wide set of functions as Spark MLlib.

B. PUBLIC TRANSIT ANALYTICS

Transportation stands out as a key area to be cared for and closely monitored, as it directly impacts on citizens' daily lives. In recent years, the European Union has established guideline documents and policies stimulating the adoption of data-driven decisions, specifically related to the field of transportation, ranging from mobility data collection to the deployment and optimization of smart mobility systems [66], [67].

There are sources of big data that can be used for a variety of purposes in this context: vehicle GPS streams [68], mobile phone data [69], smart card data [70], Points of Interest (POI) [71], among others. Out of these, the most common available and used data sources for public transport operation analyses are GPS and smart card data [8].

In order to perform analyses such as those previously described in the context of public transportation, a number of challenges must be overcome, more specifically related to data integration [66]. First of all, the different data sources to combine in order to perform the analysis were designed and currently operate with diverse purposes, and they

[17] http://druid.io/ - Last visited on July 2019
[18] http://ophidia.cmcc.it - Last visited on July 2019
[19] https://kafka.apache.org - Last visited on July 2019
[20] http://storm.apache.org - Last visited on July 2019
[21] https://flink.apache.org/ - Last visited on July 2019
[22] https://spark.apache.org/streaming/ - Last visited on July 2019
[23] https://samoa.incubator.apache.org - Last visited on July 2019

usually have no explicit reference to each other [11]. In addition, there are numerous devices asynchronously generating data for the system, which increases the likelihood of measurement errors [43]. Such challenges are tougher when a city-wide analysis is performed, as opposed to a limited (single/few-route) analysis, as in some cases [46].

Particularly in the case of smart card data, it is important to state that most transportation systems are entry-only, which means the user only needs to tap in the card at the beginning of the trip, and not again at the end. Therefore, an important information to be inferred in order to have a complete dataset is where the user exited the bus [72]. The studies found in literature use different models to perform the destination estimation task, ranging from the trip chaining model [73], passing through probability models [74], up to deep learning models [75].

Another challenge faced in some cities is the special boarding stations, where the passenger can get in from the outside, by tapping the smart card on the station reader; or from a bus by alighting at the station, and then being able to board another bus without having to use the smart card again. This creates a challenge as the second boarding is not recorded in the data and it gets harder to track users' movements around the city. Curitiba, our case study city, has many such stations distributed throughout the city. Such problems have been treated in literature by applying heuristics to decide which bus the user might have taken based on the availability of routes and their next boarding record [76]. We used a similar approach in our experiments.

The use of a combination of GTFS, GPS and smart card data allows a number of analyses and various applications in the public transportation field, such as: passenger travel pattern (Origin-Destination Matrices estimation) [73], [76], performance simulation/evaluation of transport changes (e.g., adding a new bus route or removing a bus stop) [77], and crowdedness estimation (either focusing on the user or on the transit operator point of view) [78].

X. SOCIETAL IMPACT AND EXPLOITATION

The City Municipality Dashboard has been made available to the Municipality of Curitiba, and a reduced version has also been provided to the Municipality of Campina Grande, a 400-thousand inhabitants city in the Northeast of Brazil. Furthermore, since the application creation, continuous support for data availability, quality, security and formats has been given to the Curitiba municipality. The proposed solution has fostered the adoption of standards such as ISO NP/37166 (Specification of multi-source urban data integration for smart city planning) [79] and raised awareness of data quality issues in the Curitiba Municipality, to gain better insight into the challenges related to real urban transit scenarios. In particular, the knowledge acquired from the data exploited in the proposed application, promoted the participation of the team members in the ISO Brazilian team. Different scenarios (noise, security, accessibility, gender) have been considered and integrated with mobility data to understand the impact of the whole project along the city. The integration of other data complements (such as traffic and Brazilian accident data) through masters and doctoral theses has been carried out by leveraging the results presented in this work. In terms of data openness, EUBra-BIGSEA has been instrumental to facilitate the discussion with the Municipality on making historical data for the city of Curitiba publicly available,[24] along with the integrated dataset.

XI. CONCLUSION

The dynamicity and diversity of the urban computing applications as well as the large spectrum of users' expertise in terms of big data require a platform that supports easy, scalable and manageable development. The software stack of the EUBra-BIGSEA fulfills such requirements and has been proven to work effectively in the application scenario of Urban Computing. This work presents a public transport analytics application named City Administration Dashboard, designed to fulfill the requirements of the Curitiba municipality in Brazil and built on top of the EUBra-BIGSEA platform. The paper describes the proposed solution in terms of big and fast data analytics platform, flexible and dynamic cloud infrastructure, data quality-aware components, security and privacy techniques, rich programmable framework layer and web GUI with several views. It is important to note that the application components (e.g., PRIVAaaS, DQaaS, EMaaS, Descriptive Analytics) have been developed as independent services to be used either as stand-alone components or easily assembled at the developer's convenience for other applications, not only limited to the public transportation domain. Moreover, the integration of PyOphidia and PyCOMPSs allows an easy development of parallel applications, hiding at the same time the underlying infrastructure complexity, parallelization and resource management aspects.

Furthermore, the entire application has been designed with *flexibility* and *generality* in mind, to address *replicability* with respect to other cities. It has been deployed and demonstrated in a real trans-Atlantic testbed across Europe and Brazil, and represents a major output of the EUBra-BIGSEA project.

The future work concerns the extension of the set of statistics (and the web interface views accordingly) to meet new requests from the stakeholders and the development of real-time features and predictive analytics algorithms to address new and dynamic urban transport management scenarios.

ACKNOWLEDGMENT

The authors would like to thank Antonio Aloisio for his editing and proofreading work on this paper.

REFERENCES

[1] C. Lim, K.-J. Kim, and P. P. Maglio, "Smart cities with big data: Reference models, challenges, and considerations," *Cities*, vol. 82, pp. 86–99, Dec. 2018. doi: 10.1016/j.cities.2018.04.011.

[24] http://dadosabertos.c3sl.ufpr.br/curitiba/ - Last visited on July 2019

[2] I. A. T. Hashem, V. Chang, N. B. Anuar, K. Adewole, I. Yaqoob, A. Gani, E. Ahmed, and H. Chiroma, "The role of big data in smart city," *Int. J. Inf. Manage.*, vol. 36, no. 5, pp. 748–758, 2016.

[3] A. Souza, M. Figueredo, N. Cacho, D. Araújo, and C. A. Prolo, "Using big data and real-time analytics to support smart city initiatives," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 257–262, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896316325599

[4] P. Thakuriah, L. Dirks, and Y. Keita, "Digital Infomediaries and Civic Hacking in Emerging Urban Data Initiatives," in *Seeing Cities Through Big Data: Research Methods and Applications in Urban Informatics*, P. Thakuriah, N. Tilahun, and M. Zellner, Eds. New York, NY, USA: Springer, Oct. 2016, pp. 189–207. [Online]. Available: http://eprints.gla.ac.uk/118320/

[5] A. J. Jara, D. Genoud, and Y. Bocchi, "Big data in smart cities: From poisson to human dynamics," in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 785–790.

[6] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities: Definitions, dimensions, performance, and initiatives," *J. Urban Technol.*, vol. 22, no. 1, pp. 3–21, 2015.

[7] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities," *J. Internet Services Appl.*, vol. 6, p. 25, Dec. 2015.

[8] M. Karatsoli and E. Nathanail, "Big data and understanding change in the context of planning transport systems," *J. Transp. Geography*, vol. 76, pp. 235–244, Apr. 2018.

[9] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[10] A. S. Alic *et al.*, "BIGSEA: A Big Data analytics platform for public transportation information," *Future Gener. Comput. Syst.*, vol. 96, pp. 243–269, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X18304448

[11] T. Braz, "Inferring passenger-level bus trip traces from schedule, positioning and ticketing data: Methods and applications," M.S. thesis, Dept. Syst. Comput., Universidade Federal de Campina Grande, Grande, Brazil, 2019.

[12] D. G. Mestre, C. E. S. Pires, D. C. Nascimento, A. R. M. de Queiroz, V. B. Santos, and T. B. Araujo, "An efficient spark-based adaptive windowing for entity matching," *J. Syst. Softw.*, vol. 128, pp. 1–10, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121217300559

[13] T. Basso, R. Moraes, N. Antunes, M. Vieira, W. Santos, and W. Meira, "PRIVAaaS: Privacy approach for a distributed cloud-based data analytics platforms," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2017, pp. 1108–1116.

[14] C. Ramon-Cortes, A. Serven, J. Ejarque, D. Lezzi, and R. M. Badia, "Transparent orchestration of task-based parallel applications in containers platforms," *J. Grid Comput.*, vol. 16, no. 1, pp. 137–160, Mar. 2018. doi: 10.1007/s10723-017-9425-z.

[15] R. M. Badia, J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent, "COMP superscalar, an interoperable programming framework," *SoftwareX*, vol. 3, pp. 32–36, Dec. 2015. doi;[p 10.1016/j.softx.2015.10.004.

[16] S. Fiore, A. D'Anca, C. Palazzo, I. Foster, D. Williams, and G. Aloisio, "Ophidia: Toward big data analytics for escience," *Procedia Comput. Sci.*, vol. 18, pp. 2376–2385, Aug. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913005528

[17] S. Fiore and A. D'Anca, D. Elia, C. Palazzo, D. Williams, I. Foster, and G. Aloisio, "Ophidia: A full software stack for scientific data analytics," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, Jul. 2014, pp. 343–350.

[18] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. J. Joseph, R. H. Katz, S. Shenker, and I. Stoica, "Mesos: A platform for fine-grained resource sharing in the data center," in *Proc. 8th USENIX Conf. Networked Syst. Design Implement.*, Apr. 2011, pp. 295–308. [Online]. Available: http://dl.acm.org/citation.cfm?id=1972457.1972488

[19] A. Calatrava, E. Romero, G. Moltó, M. Caballer, and J. M. Alonso, "Self-managed cost-efficient virtual elastic clusters on hybrid cloud infrastructures," *Future Gener. Comput. Syst.*, vol. 61, pp. 13–25, Aug. 2016. doi: 10.1016/j.future.2016.01.018.

[20] J. Arkko, G. Zorn, V. Fajardo, and J. Loughney, *Diameter Base Protocol,' Internet Requests for Comments*, document RFC 6733, Oct. 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6733

[21] A. C. Rubens, S. Willens, W. A. Simpson, and C. Rigney, *Remote Authentication Dial in User Service (Radius)*, document RFC 2865, Jun. 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2865

[22] L. G. D. Carrel, *The Tacacs+ Protocol*. Internet Engineering Task Force, Jan. 1997. [Online]. Available: https://tools.ietf.org/html/draft-grant-tacacs-02.txt

[23] OASIS Committee Note 01. *Identity in the Cloud Use Cases Version 1.0*. Accessed: May 8, 2012. [Online]. Available: http://docs.oasis-open.org/id-cloud/IDCloud-usecases/v1.0/cn01/IDCloud-usecases-v1.0-cn01.html

[24] A. Saldhana, R. Marian, F. G. Marmol, and C. Kappler, Eds. Cloud Authorization Use Cases Version 1.0. OASIS Committee Note Draft 01/Public Review Draft 01. Accessed: Mar. 17, 2014. [Online]. Available: http://docs.oasis-open.org/cloudauthz/CloudAuthZ-usecases/v1.0/CloudAuthZ-usecases-v1.0.html

[25] T. Basso, R. Matsunaga, R. Moraes, and N. Antunes, "Challenges on Anonymity, Privacy, and Big Data," in *Proc. 7th Latin-Amer. Symp. Dependable Comput. (LADC)*, Oct. 2016, pp. 164–171.

[26] R. Matsunaga, I. Ricarte, T. Basso, and R. Moraes, "Towards an ontology-based definition of data anonymization policy for cloud computing and big data," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2017, pp. 75–82.

[27] H. Silva, T. Basso, R. Moraes, D. Elia, and S. Fiore, "A re-identification risk-based anonymization framework for data analytics platforms," in *Proc. 14th Eur. Dependable Comput. Conf. (EDCC)*, Sep. 2018, pp. 101–106.

[28] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[29] K. El Emam, F. K. Dankar, R. Vaillancourt, T. Roffey, and M. Lysyk, "Evaluating the risk of re-identification of patients from hospital prescription records," *Can. J. Hospital Pharmacy*, vol. 62, no. 4, p. 307, Jul. 2009.

[30] K. El Emam, D. Paton, F. Dankar, and G. Koru, "De-identifying a public use microdata file from the canadian national discharge abstract database," *BMC Med. Inform. Decis. Making*, vol. 11, no. 1, p. 53, Dec. 2011.

[31] H. B. Sta, "Quality and the efficiency of data in 'smart-cities'," *Future Gener. Comput. Syst.*, vol. 74, pp. 409–416, Aug. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16308081

[32] M. Mirzaie, B. Behkamal, and S. Paydar, "Big data quality: A systematic literature review and future research directions," 2019, *arXiv:1904.05353*. [Online]. Available: https://arxiv.org/abs/1904.05353

[33] J. M. Tien, "Big data: Unleashing information," *J. Syst. Sci. Syst. Eng.*, vol. 22, no. 2, pp. 127–151, 2013.

[34] C. Batini and M. Scannapieco, *Data and Information Quality—Dimensions, Principles and Techniques* (Data-Centric Systems and Applications). New York, NY, USA: Springer, 2016. doi: 10.1007/978-3-319-24106-7.

[35] R. Y. Wang and D. M. Strong, "Beyond accuracy: What data quality means to data consumers," *J. Manage. Inf. Syst.*, vol. 12, no. 4, pp. 5–33, 1996.

[36] L. Berti-Equille and J. Borge-Holthoefer, *Veracity of Data: From Truth Discovery Computation Algorithms to Models of Misinformation Dynamics* (Synthesis Lectures on Data Management). New York, NY, USA: Morgan Claypool, 2015. doi: 10.2200/S00676ED1V01Y201509DTM042.

[37] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data Sci. J.*, vol. 14, p. 22, May 2015.

[38] I. Caballero, M. Serrano, and M. Piattini, "A data quality in use model for big data," *Future Gener. Comput. Syst.*, vol. 63, pp. 65–74, Oct. 2014.

[39] C. Cappiello and W. Samá, and M. Vitali, "Quality awareness for a successful big data exploitation," in *Proc. 22nd Int. Database Eng. Appl. Symp.*, Villa San Giovanni, Italy, Jun. 2018, pp. 37–44. doi: 10.1145/3216122.3216124.

[40] F. Naumann, "Data profiling revisited," *SIGMOD Rec.*, vol. 42, no. 4, pp. 40–49, Dec. 2013. doi: 10.1145/2590989.2590995.

[41] T. C. Redman, *Data Quality for the Information Age*, 1st ed. Norwood, MA, USA: Artech House, 1997.

[42] D. Ardagna, C. Cappiello, and W. Samá, and M. Vitali, "Context-aware data quality assessment for big data," *Future Gener. Comput. Syst.*, vol. 89, pp. 548–562, Apr. 2018. doi: 10.1016/j.future.2018.07.014.

[43] T. Braz, M. Maciel, D. G. Mestre, N. Andrade, C. E. Pires, A. R. Queiroz, and V. B. Santos, "Estimating inefficiency in bus trip choices from a user perspective with schedule, positioning, and ticketing data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3630–3641, Nov. 2018.

[44] R. Raymond and T. Imamichi, "Bus trajectory identification by map-matching," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 1618–1623.

[45] D. Elia, S. Fiore, A. D'Anca, C. Palazzo, I. Foster, and D. N. Williams, "An in-memory based framework for scientific data analytics," in *Proc. ACM Int. Conf. Comput. Frontiers*. New York, NY, USA: ACM, May 2016, pp. 424–429. doi: 10.1145/2903150.2911719.

[46] M. Tanaka, T. Kimata, and T. Arai, "Estimation of passenger origin-destination matrices and efficiency evaluation of public transportation," in *Proc. 5th IIAI Int. Congr. Adv. Appl. Inform. (IIAI-AAI)*, Jul. 2016, pp. 1146–1150.

[47] D. Borthakur, "HDFS architecture guide," *Hadoop Apache Project*, vol. 53, nos. 1–13, p. 2, 2008.

[48] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Proc. IEEE 26th Symp. Mass Storage Syst. Technol. (MSST)*, Washington, DC, USA, May 2010, pp. 1–10. doi: 10.1109/MSST.2010.5496972.

[49] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The google file system," in *Proc. 19th ACM Symp. Operating Syst. Princ.* New York, NY, USA: ACM, 2003, pp. 29–43. doi: 10.1145/945445.945450.

[50] *OGC - Simple Feature Access*. Accessed: Dec. 16, 2017. [Online]. Available: http://www.opengeospatial.org/standards/sfs

[51] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 4:1–4:26, Jun. 2008.

[52] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. 2Nd USENIX Conf. Hot Topics Cloud Comput.* Berkeley, CA, USA: USENIX Association, 2010, p. 10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863113

[53] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proc. 9th USENIX Conf. Networked Syst. Design Implement.* Berkeley, CA, USA: USENIX Association, Apr. 2012, p. 2. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228301

[54] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. doi: 10.1145/1327452.1327492.

[55] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: A warehousing solution over a map-reduce framework," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1626–1629, Aug. 2009. doi: 10.14778/1687553.1687609.

[56] F. Yang, E. Tschetter, X. Léauté, N. Ray, G. Merlino, and D. Ganguli, "Druid: A real-time analytical data store," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. New York, NY, USA: ACM, 2014, pp. 157–168. doi: 10.1145/2588555.2595631.

[57] S. Fiore, C. Palazzo, A. D'Anca, I. Foster, D. N. Williams, and G. Aloisio, "A big data analytics framework for scientific data management," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2013, pp. 1–8.

[58] J. Ekanayake, S. Pallickara, and G. Fox, "MapReduce for data intensive scientific analyses," in *Proc. 4th IEEE Int. Conf. Sci.*, Washington, DC, USA, Dec. 2008, pp. 277–284. doi: 10.1109/eScience.2008.59.

[59] J. Shi, Y. Qiu, U. F. Minhas, L. Jiao, C. Wang, B. Reinwald, and F. Özcan, "Clash of the titans: MapReduce vs. Spark for large scale data analytics," *Proc. VLDB Endow.*, vol. 8, no. 13, pp. 2110–2121, Sep. 2015. doi: 10.14778/2831360.2831365.

[60] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, and M. Zaharia, "Spark SQL: Relational data processing in spark," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2015, pp. 1383–1394. doi: 10.1145/2723372.2742797.

[61] S. Shi. (2017). *By-layer Spark Cubing*. Accessed: Jun. 29, 2019. [Online]. Available: http://kylin.apache.org/blog/2017/02/23/by-layer-spark-cubing/

[62] N. Garg, *Learning Apache Kafka—Second Edition*, 2nd ed. Birmingham, U.K.: Packt Publishing, 2015.

[63] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, and N. Bhagat, "Stormtwitter," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2014, pp. 147–156. doi: 10.1145/2588555.2595641.

[64] A. Alexandrov, R. Bergmann, S. Ewen, J. C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, and F. Naumann, "The stratosphere platform for big data analytics," *VLDB J.*, vol. 23, no. 6, pp. 939–964, Dec. 2014. doi: 10.1007/s00778-014-0357-y.

[65] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," *J. Big Data*, vol. 2, no. 1, p. 24, 2015.

[66] A. Urbanek, "Data-driven transport policy in cities: A literature review and implications for future developments," *Proc. Sci. Tech. Conf. Transp. Syst. Theory Pract.*, 2019, pp. 61–74.

[67] Té Commissions. (2018). *Report for Europe Parliament: Mapping Smart Cities EU. The European Commission*. [Online]. Available: http://www.europarl.europa.eu/RegData/etudes/etudes/join/2014/507480/IPOL-ITRE_ET(2014)507480_EN.pdf

[68] Y. Zheng, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proc. Int. Conf. World Wide Web*, May 2009, pp. 1–9. [Online]. Available: https://www.microsoft.com/en-us/research/publication/mining-interesting-locations-and-travel-sequences-from-gps-trajectories/

[69] F. Calabrese, M. Diao, G. Di Lorenzo, J. Ferreira, and C. Ratti, "Understanding individual mobility patterns from urban sensing data: A mobile phone trace example," *Transp. Res. C, Emerg. Technol.*, vol. 26, pp. 301–313, Jan. 2013.

[70] M. Bagchi and P. R. White, "The potential of public transport smart card data," *Transp. Policy*, vol. 12, no. 5, pp. 464–474, 2005.

[71] C. Anda, A. Erath, and P. J. Fourie, "Transport modelling in the age of big data," *Int. J. Urban Sci.*, vol. 21, no. 1, pp. 19–42, Jun. 2017.

[72] T. Li, D. Sun, J. Peng, and K. Yang, "Smart card data mining of public transport destination: A literature review," *Information*, vol. 9, no. 1, p. 18, Jan. 2018.

[73] A. A. Nunes, T. G. Dias, and J. E. F. Cunha, "Passenger journey destination estimation from automated fare collection system data using spatial validation," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 133–142, Jan. 2016.

[74] X. Zhou, X. Yang, and X. Wu, "Origin-destination matrix estimation method of public transportaion flow based on data from bus integrated-circuit cards," *J. Tongji Univ. Natural Sci.*, vol. 40, pp. 1027–1030, Jul. 2012.

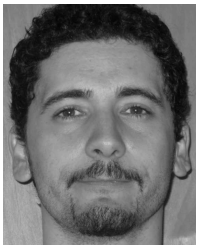
[75] J. Jung and K. Sohn, "Deep learning architecture to forecast destinations of bus passengers from entry-only smart-card data," *IET Intell. Transp. Syst.*, vol. 11, no. 1, pp. 334–339, 2017.

[76] M. A. Munizaga and C. Palma, "Estimation of a disaggregate multimodal public transport Origin–Destination matrix from passive smartcard data from Santiago, Chile," *Transp. Res. C, Emerg. Technol.*, vol. 24, pp. 9–18, Oct. 2012. doi: 10.1016/j.trc.2012.01.007.

[77] K. S. Kim, S.-H. Cheon, and S.-J. Lim, "Performance assessment of bus transport reform in Seoul," *Transportation*, vol. 38, pp. 719–735, Sep. 2011.

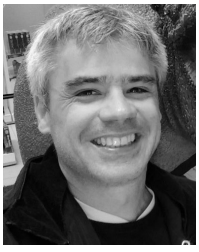
[78] F. Toqué, M. Khouadjia, E. Come, M. Trepanier, and L. Oukhellou, "Short long term forecasting of multimodal transport passenger flows with machine learning methods," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 560–566.

[79] *Smart Community Infrastructures—Specification of Multi-Source Urban Data Integration for Smart City Planning(SCP)*, Int. Org. Standardization, Geneva, Switzerland, 2018. [Online]. Available: https://www.iso.org/standard/69252.html



SANDRO FIORE received the Ph.D. degree in innovative materials and technologies from the University of Lecce, Italy. Since 2010, he has been the Principal Investigator of the Ophidia Project, a research effort on high-performance data analytics for eScience. He has been a Visiting Scientist with the Lawrence Livermore National Laboratory, from 2011 to 2013, working in the context of the Earth System Grid Federation/CMIP, and the University of Chicago, in 2019, working on HPDA for eScience, provenance, and reproducibility. He is currently a Senior Scientist and the Head of the Data Science and Learning Research Team of the Advanced Scientific Computing Division at the Euro-Mediterranean Center on Climate Change (CMCC) Foundation. He has been involved in several national and EU (FP6, FP7, and H2020) projects, working on scientific data management topics. As a Scientific Data Management Expert, he has been involved in the IESP, EESI, EXDCI, and BDEC projects on exascale challenges. He is the author of more than 80 scientific papers, the Editor of the book *Grid and Cloud Database Management*, and the coauthor of *The International Exascale Software Project Roadmap*. His research interests include high-performance database management, big data analytics and mining, large-scale data warehouses, array databases, and in-memory analytics. He is also a member of ACM.

DONATELLO ELIA received the M.Sc. degree in computer engineering from the University of Salento, Italy, in 2013, where he is currently pursuing the Ph.D. degree in engineering of complex systems. In 2013, he joined the Advanced Scientific Computing (ASC) Division, Euro-Mediterranean Center on Climate Change (CMCC) Foundation. His main research interests include high-performance and distributed computing, data-intensive analytics, big data management, and data mining. He has been involved in various European projects, such as the FP7 and Horizon 2020 programs. He has authored or coauthored various papers in refereed journals and conference proceedings. He is currently a member of the IEEE Computer Society.



CARLOS EDUARDO PIRES received the Ph.D. degree in computer science from the Universidade Federal de Pernambuco, Brazil. Since November 2009, he has been a Professor of computer science with the Computing and Systems Department, Universidade Federal de Campina Grande, Brazil, where he currently collaborates with research in the area of information systems and databases at the Data Quality Laboratory, Universidade Federal de Campina Grande. He has experience in computer science, with an emphasis on databases, acting on the following topics: decision support systems, knowledge discovery, data quality, and information integration.

DEMETRIO GOMES MESTRE received the Ph.D. degree in computer science from the Universidade Federal de Campina Grande, Brazil. He has experience in computer science, focusing on information systems and databases. His research interests include data quality, big data, and cloud computing.

CINZIA CAPPIELLO received the Ph.D. degree in information technology from the Politecnico di Milano, in 2005. She is currently an Associate Professor of computer engineering with the Politecnico di Milano, Italy. Her research interests include data and information quality aspects in big data, service-based and Web applications, and sensor data management. On such topics, she has published papers in international journals and conferences.

MONICA VITALI received the Ph.D. degree in information technology from the Politecnico di Milano, Milan, Italy, in 2014. She is currently a Research Assistant with the Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, and a Senior Lecturer with the Computing Science Department, Umeå University, Sweden. She is interested in adaptation and monitorability in cloud computing, adaptive and self-adaptive systems and services, and machine learning techniques for adaptation. She participated in several EC funded projects, such as HUMANOBS, GAMES, ECO2CLOUDS, and DITAS.

NAZARENO ANDRADE received the Ph.D. degree in electrical engineering from the Federal University of Campina Grande, where he is currently a Professor. His research interests include data analytics, collaborative systems, and social computing.

TARCISO BRAZ received the M.Sc. degree in computer science from the Universidade Federal de Campina Grande, Brazil. He has experience with the application of data analytics to various contexts, including public transportation, politics, and public policy. His research interests include data science, artificial intelligence, big data, smart cities, and public policy.

DANIELE LEZZI received the Ph.D. degree in information technology engineering from the University of Salento, Italy, in 2007. He is currently a Senior Researcher with the Computer Science Department, Barcelona Supercomputing Center. In particular, his research addresses the design of programming frameworks for the porting and execution of scientific applications on distributed computing infrastructures, such as grid and clouds with a special emphasis on interoperability. He has participated in several EC funded projects, such as GridLab, CoreGRID, BEinGRID, OGF-Europe, SIENA, VENUS-C, IS-ENES, EGI Federated Cloud, EU-Brazil OpenBio, EU-Brazil Cloud Connect, and EUBra-BIGSEA. His research interests include high performance, distributed, grid and cloud computing, and programming models. He is currently involved in the mobile For to Cloud (mF2C) Project, the BioExcel Center of Excellence for Computational Biomolecular Research, and the Landsupport initiative.

REGINA MORAES received the Ph.D. degree in software engineering from the University of Campinas. She joined the University of Coimbra, Portugal, and LAAS/CNRS, France, for research internship. She is currently a Full Professor with the Department of Informatics, School of Technology, University of Campinas, Brazil. Her research interests include data privacy, security, and dependability assessment of software systems. She is also a member of the IEEE Computer Society.

TANIA BASSO received the Ph.D. degree in computer engineering from the University of Campinas, in 2015. She is currently a Postdoctoral Researcher with the University of Campinas. She has been working with privacy concerns and data anonymization in big data and cloud computing context. She has been involved as a Researcher in many research projects, both at the national and European levels. Her research interests include privacy, security, testing techniques, and dependability assessment.

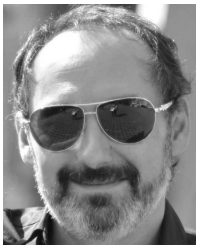
NADIA P. KOZIEVITCH received the Ph.D. degree in computer science from the Universidade Estadual de Campinas, Brazil. Since 2012, she has been a Professor with the Federal University of Technology, Brazil. Her interests include GIS, databases, and smart cities.

KEIKO VERÔNICA ONO FONSECA graduated in electrical engineering from the Federal University of Paraná, in 1985. She received the M.Sc. degree from the State University of Campinas, in 1988, and the Ph.D. degree from the Federal University of Santa Catarina, in 1997, both in electrical engineering. She carried her postdoctoral studies at the Faculty of Informatics, TU Dresden, in 2013. She is currently a Full Professor with the Federal University of Technology—Paraná (UTFPR). She is also the EU-BR H2020 SecureCloud Project Leader at UTFPR. She is also the Smart Cities Project Leader at UTFPR (Sweden-Curitiba project funded by Vinnova). Her research interests include real-time communication systems, data security and privacy, and image processing. She is a member of the IEEE ComSoc, the Brazilian Computer Society (SBC), and IEICE, Japan. She acts as a volunteer for the development of the Badminton sport in Brazil.

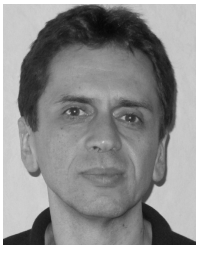
NUNO ANTUNES received the Ph.D. degree in information science and technology from the University of Coimbra (CISUC), in 2014, where he is currently an Assistant Professor. He has been with the Centre for Informatics and Systems, CISUC, since 2008, working on security and dependability topics. His expertise includes testing techniques, fault injection, vulnerability injection, and benchmarking that are applied to the assessment of cloud systems, Web services and applications, virtualized environments, and data management systems.

MARCO VIEIRA received the Ph.D. degree from UC, Portugal, in 2005. He is currently a Full Professor with the University of Coimbra, Coimbra, Portugal. His research interests include dependability and security assessment and benchmarking, fault injection, software processes, and software quality assurance, subjects in which he has authored or coauthored more than 200 papers in refereed conferences and journals. He has participated in and coordinated several research projects, both at the national and European levels. He has served on the program committees of major conferences of the dependability area and acted as a Referee for many international conferences and journals in the dependability and security areas.

COSIMO PALAZZO received the degree *(cum laude)* in computer engineering from the University of Lecce, Italy, in 2003, and the Ph.D. degree in information engineering from the University of Salento, in 2007. He worked at the NATO Consultation, Command and Control Agency (NC3A), The Hague, The Netherlands, on wireless networking in the military. He was a Research Assistant with the Department of Innovation Engineering, University of Salento. He joined the Euro-Mediterranean Centre on Climate Change (CMCC) Foundation, in 2012. His activities concern the design, development, and performance analysis of software solutions for scientific data analysis in the context of climate change. He is the author of several papers published in international journals and proceedings of international conferences. His research interests include the design, modeling, and performance evaluation of communication protocols for wireless networks.



IGNACIO BLANQUER has been a member of the Grid and High Performance Computing Research Group (GRyCAP), since 1993, being the Leader of the group, since 2016. He is currently a Full Professor with the Computer System Department, Universitat Politècnica de València (UPV). He has been involved in parallel computation and medical image processing, participating in 57 national and European research projects. He has authored or coauthored 42 articles in indexed journals and book chapters and more than 100 papers in national and international journals and conference proceedings. He has served as a Coordinator of the application area in the Spanish Network for eScience. He has been the Project Coordinator of EUBrazilCloudConnect (FP7), EUBra-BIGSEA (H2020), CLUVIEM (National Research Project). He is also the Project Coordinator of and a Co-Principal Investigator in ATMOSPHERE (H2020) and the BigCLOE National Research Project. He is also the Spanish Delegate for the e-IRG.



WAGNER MEIRA, JR. is currently a Full Professor with the Computer Science Department, Universidade Federal de Minas Gerais, Brazil. He has been the Project Coordinator of EUBra-BIGSEA (H2020). He is also the Vice-Coordinator of INCT-Cyber, the Brazilian Institute of Science and Technology for a Massively Connected Society. He has published more than 300 papers in top venues and is a coauthor of the book *Data Mining and Analysis: Fundamental Concepts and Algorithms* (Cambridge University Press, 2014). His research interests include scalability and efficiency of large-scale parallel and distributed systems, from massively parallel to the Internet-based platforms, and on data mining algorithms, their parallelization, and application to areas, such as information retrieval, bioinformatics, and e-governance.

GIOVANNI ALOISIO is currently a Full Professor of information processing systems with the Department of Innovation Engineering, University of Salento, Lecce, Italy, where he leads the HPC Laboratory. He is also the Former Director of the Advanced Scientific Computing (ASC) Division at the Euro-Mediterranean Center on Climate Change (CMCC). He is a member of the CMCC Strategic Council and the Director of the CMCC Supercomputing Center. He is the author of more than 250 papers in refereed journals on high-performance computing, grid and cloud computing, and distributed data management. His expertise concerns high-performance computing, grid and cloud computing, and distributed data management. He has been involved in several EU projects, such as GridLab, EGEE, IS-ENES, EESI, EXDCI, EUBrazilCC, INDIGO-DataCloud, OFIDIA, and EUBra-BIGSEA. He has also contributed to the International Exascale Software Project (IESP) exascale roadmap. He is also a member of the ENES HPC Task Force. He has been the Chair of the European panel of experts on WCES that has contributed to the PRACE strategic document "The Scientific Case for HPC in Europe 2015–2020."