BUILD-IT2023 Workshop
19-20 October, Rome, Italy

AN IT PLATFORM FOR MANAGING AND SHARING DATA AMONG DIGITAL TWINS OF AN URBAN INTELLIGENCE SYSTEM

Stefano Silvestri *and* Mario Ciampi
National Research Council of Italy, Institute for High Performance Computing and Networking, Via Pietro Castellino, 111 - 80131 Naples, Italy
e-mail: {stefano.silvestri, mario.ciampi}@icar.cnr.it

This paper describes a novel architecture devoted to the implementation of an IT platform for the deployment and integration of the different Digital Twin subsystems that compose a complex Urban Intelligence system. The proposed IT platform has the following main purposes: i) facilitating the deployment of the subsystems in a cloud environment; ii) effectively storing, integrating, managing, and sharing the huge amount of heterogeneous data acquired and produced by each subsystem using a data lake; iii) supporting data exchange and sharing; iv) managing and executing workflows, to automatically coordinate and run processes; and v) provide and visualize the required information. A prototype of the IT platform has been implemented leveraging open source frameworks, allowing to test its functionalities and performance. The results of the tests confirmed that the proposed architecture can efficiently and easily support the deployment and integration of heterogeneous subsystems, making them able to share and integrate their data and to select, extract, and visualize the information required by a user, also promoting the integration with other external systems.

Keywords: *urban intelligence, IT platform, data lake, data management*

1. Introduction

A comprehensive Urban Intelligence (UI) system for a smart city includes multiple virtual representations of heterogeneous subsystems (Digital Twins, DTs) that model complex interacting assets, infrastructures, and social aspects of the city, such as mobility, social and computer networks, weather, pollution, tourist flow, and so on [1, 2]. To allow a correct functioning of the DTs according to an integrated approach, several IT paradigms and technologies should be integrated in an UI system, such as dedicated IoT sensor networks, network infrastructures, information systems, high-level web services, cloud or edge-computing environments [3], able to collect and manage the heterogeneous data of the city and provide the hardware and software infrastructures to implement and run the required models and analytics, visualization, and user interfaces functionalities [4, 5].

In light of such considerations, the complexity and heterogeneity of the DTs require an advanced enabling IT platform that facilitates their deployment, enables their intercommunication and integration, provides data management and sharing capabilities, supports information searching and retrieving, and offers visualization and user interface capabilities, allowing to fully integrate the DTs.

In this contribution, the architecture of an IT platform able to support the management and integration of DTs for UI systems is presented. The proposed IT platform has the following main purposes: i) facilitating the deployment of the DTs in a cloud environment; ii) effectively storing, integrating, managing, and sharing the huge amount of heterogeneous data acquired and produced by each DT; iii) integrating and supporting data exchange and sharing; and iv) managing workflows, to automatically execute the requested processes and provide and visualize the required information. A prototype of the proposed architecture has been implemented, relying on a cloud infrastructure, a data lake which acts as both a data collector and repository, a set of specifically designed APIs, a workflow engine, and a user interface. The preliminary experiments showed that the proposed IT platform can be effectively adopted for the deployment in real UI projects.

2. IT Platform Architecture

The layers of the proposed IT platform architecture are depicted in Figure 1.

In the lower layer, the *Cloud computing* layer provides and dynamically manages and coordinates the hardware and software resources in the cloud environment. This layer is also devoted to container deployment and orchestration, facilitating the installation and execution of containerized software modules that implement each user DT subsystem in any cloud-based environment, as well as facilitating their integration with other systems.

The next layer is the *Data lake* layer, whose purpose is to acquire, store, integrate, query, and retrieve data collected or produced by any UI subsystem, as well as obtained from external sources. The data lake is based on a Distributed File System and a NoSQL database, able to manage and process structured, semi-structured, and unstructured raw data. To the end of preventing the data swamp degeneration, a relational database technology is integrated into the data lake, with the purpose of enriching the heterogeneous data with specific metadata and a catalog. This way, it supports their retrieval and integration, enriches the structured and unstructured data with additional information, and offers basic Extract, Transform, and Load (ETL) functionalities for data aggregation. Moreover, the data lake layer also acts as a common data collector for the DTs, allowing them to easily share their own data and communicate with the other subsystems. Asynchronous notification services are included in this layer, to the end of notifying specific modules of possible data updates in the data lake according to the publish/subscribe pattern, thus optimizing bandwidth and performances.

The *Workflow engine* layer includes a workflow management and execution component aimed to provide the UI system with an automatic business process engine, whose purpose is to easily implement and execute the workflows that are used to model each analysis requested, where the interaction of more subsystems needs. The workflow engine coordinates the process and data flows through the various subsystems, in order to obtain the required information or perform the requested operation.

The upper *Data visualization and user interface* layer provides the visualization and user interface features (including dashboards), adopting a web-based approach.

Finally, the communication with the data lake and the other UI subsystems exploits the transversal *Communication interface* layer, where a set of dedicated APIs are available to provide common and standard way to get or send data and/or access the exposed functionalities. The Communication interface layer also offers the possibility to connect the IT platform with external systems of the city (like local databases or IT services) and the authentication and authorization functionalities, able to implement single-sign on through standardized claims.

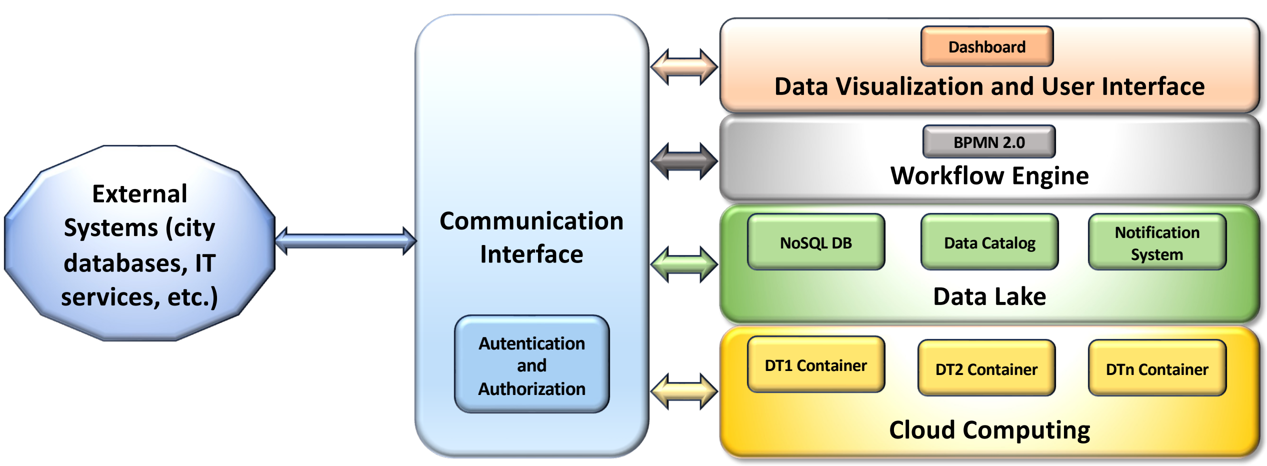

Figure 1: Layers of the proposed IT platform architecture.

3. Implementation and Testing Details

A prototype of the proposed architecture has been implemented exploiting different open source frameworks.

The cloud environment for DTs deployment adopts Docker for module containerization and Kubernetes for their orchestration. This approach makes the IT platform easily deployable, customizable, and embedded within different environments and other external systems.

The data lake and its data catalog are implemented using a MongoDB NoSQL database and a GridFS distributed file system (the latter is devoted to the storage of large raw data). These frameworks ensure high-performance levels, scalability, and the capability of managing very large and heterogeneous data collections. Moreover, the NoSQL database does not require a strict data schema, allowing a simple extension of data models and the catalog. The asynchronous notification of data updates has been implemented adopting a publish/subscribe approach based on Message Queue Telemetry Transport (MQTT) standard protocol, allowing the selection of data that requires notification and the corresponding subscriber modules that receive such notifications. The interfaces to the data lake layer have been developed as RESTful APIs on the HTTPS application protocol, leveraging Python Flask framework and PyMongo library. The authentication and authorization mechanisms for the data lake and other modules use claims based on Java Web Token, leveraging the OAuth 2.0 and OpenID Connect protocols.

The workflow engine layer leverages a Business Process Management engine, which allows the workflows to be easily designed according to the BPMN 2.0 standard. Finally, the visualization and user interface layer is obtained by using a web-based data visualization software developed in JavaScript.

The implemented prototype made it possible to carry out some preliminary tests, exploiting the data and subsystems of two Italian UI projects in the cities of Matera and Catania. The functional and performance tests carried out demonstrated that the proposed IT platform can acquire from the sensor network of a city high rates of data, with a rate of 5 minutes, sharing them at the same time among the UI DTs. It is also able to store and integrate the huge volume of data from both internal and external sources. RESTful interfaces effectively facilitate intercommunication and data sharing between UI subsystems, while workflows can be easily configured and executed.

4. Conclusions

This paper presented a general architecture of a novel IT platform that allows for the deployment and integration of DT subsystems of an UI system. The IT platform leverages the cloud environment and container technology for DTs deployment, adopts a data lake approach and dedicated APIs to collect, store, retrieve, integrate, and share the data produced inside and outside the UI system. It also includes a workflow engine to design and execute the requested processes, a visualization tool, and a user interface. A prototype of the architecture has been implemented and tested in real UI projects, demonstrating that it effectively supports the deployment and the integration of DT subsystems for UI.

REFERENCES

1. Deng, T., Zhang, K. and Shen, Z.-J. M. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities, *Journal of Management Science and Engineering*, **6** (2), 125–134, (2021).

2. White, G., Zink, A., Codecá, L. and Clarke, S. A digital twin smart city for citizen feedback, *Cities*, **110**, 103064, (2021).

3. Habibzadeh, H., Kaptan, C., Soyata, T., Kantarci, B. and Boukerche, A. Smart city system design: A comprehensive study of the application and data planes, *ACM Comput. Surv.*, **52** (2), (2019).

4. Allam, Z. and Dhunny, Z. A. On big data, artificial intelligence and smart cities, *Cities*, **89**, 80–91, (2019).

5. Han, Q., Nesi, P., Pantaleo, G. and Paoli, I. Smart city dashboards: Design, development, and evaluation, *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, pp. 1–4, (2020).