Cluster Computing (2023) 26:2115–2128
https://doi.org/10.1007/s10586-022-03732-x

Using FIWARE and blockchain in smart cities solutions

Stefano Loss[1] · Har Preet Singh[2] · Nélio Cacho[1] · Frederico Lopes[3]

Received: 4 April 2022 / Revised: 11 August 2022 / Accepted: 25 August 2022 / Published online: 7 September 2022

Abstract
Nowadays, Blockchain has been widely used to store decentralized and secure transactions involving cryptocurrency, e.g., Bitcoin, Ethereum, etc. However, Blockchain can also store other types of information besides monetary transactions. On the other hand, innovative solutions for smart cities are concerned with how services and information can be safely stored and shared. For this reason, smart city systems can benefit from using Blockchain to integrate their data and services. These smart solutions also demand consistency and standardization across the industry. However, this Blockchain integration varies according to its implementation. FIWARE, a framework of an open-source platform for smart solutions, adopts NGSI Standards (Context Information Management (CIM); NGSI-LD API: Tech. Rep., CIM and ETSI Industry Specification Group (ISG), 2020) to enable the integration of components and provides the basis for interoperability and portability among smart solutions. Unfortunately, FIWARE does not support any integration with Blockchain technology. Hence, this paper proposes a set of new components to allow FIWARE to be integrated with Blockchain technology. With these proposed components, it is possible to support Blockchain technology with smart city applications via the FIWARE platform. For instance, we have designed and implemented a FIWARE Blockchain adapter to submit/listen to transactions from/to FIWARE Context Broker to/from any Blockchain implementation without human intervention. In addition, we present a global post-pandemic vaccination case study to evaluate the proposed approach in the Smart City context.

Keywords Vaccination · Blockchain · Interoperability · FIWARE · NGSI

1 Introduction

Since the Industrial Revolution, the growth of cities in all continents of the world has intensified. According to the report *World Urbanization Prospects 2018*,[1] 54% of the world population in 2018 lived in urban areas. It is estimated that this proportion, in 2050, will increase to 68.4% of the world population living in cities.

Most of the time, this growth occurs disorderly, causing several social problems, such as violence, environmental pollution, insufficient hospital care, floods, traffic jams, and reduced quality in the provision of public services [1]. Public authorities must create policies that stimulate the orderly growth of cities in an innovative way, guaranteeing public services and quality of life for the entire population [2].

However, it is clear that existing public resources are insufficient or not adequately used to meet the growing urban population, making the essential services offered by

✉ Stefano Loss
momoloss10@gmail.com

Har Preet Singh
harpreet.singh@fiware.org

Nélio Cacho
neliocacho@dimap.ufrn.br

Frederico Lopes
fred@imd.ufrn.br

[1] Department of Informatics and Applied Mathematics, Federal University of Rio Grande do Norte, Natal, Rio Grande do Norte, Brazil

[2] FIWARE Foundation, Berlin, Germany

[3] Metropole Digital Institute, Federal University of Rio Grande do Norte, Natal, Rio Grande do Norte, Brazil

[1] Available at: https://esa.un.org/unpd/wup/.

the government increasingly precarious. On the other hand, with the advancement of technology, large cities look for ways to use information and Communication Technologies (ICT) to address those problems. These technologies can be used as computational solutions to make cities more "smart," giving rise to the term *Smart Cities* (SC). A smart city (SC) integrates the city's physical infrastructure with information and communication technology, intending to improve the population's quality of life [3].

Most public services in some metropolis already use some Information and Management System (IMS) to store their data and, in some cases, present them to the population. In most cases, IMS is developed without standardization by different private companies or cities' public bodies. This isolated development causes *siloed* systems, making it challenging to communicate with other systems in an interoperable way.

Interoperability among systems is the ability of a set of systems to communicate, share specific information and perform missions together with a common goal that the systems would not perform individually. Systems of the same city could utilize a fully integrated approach by providing interoperability between them. In that case, they could be better used, enabling the emergence of new functionalities that would benefit even more the entire population [4].

There are many ways to integrate systems, and most of the time, this is done in a *ad hoc* way to meet a specific stakeholder need [5]. In this case, it is necessary to know the candidate systems' implementation details to perform some adjustments. The problem with this approach is that the integration complexity increases with the number of systems to be integrated.

Another way is to use a *middleware* platform to compose Smart Cities systems, for example, Kaa,[2] SOFIA[3] (Smart Objects for Intelligent Applications), CityHub [6], and FIWARE[4]. For instance, FIWARE is a middleware platform widely used by municipalities and businesses. FIWARE is an open-source initiative that defines a universal set of standards and components (named Generic Enablers—GEs) for smart city applications. Orion Context Broker is the FIWARE core component; it stores and exchanges any information (regarding sensors, actuator, state of entities) and enables the systems attached to it to perform many operations regarding any smart city component.

Although the FIWARE can be used to provide some integration between systems through its standardization, it does not guarantee the necessary quality attributes: reliability, integrity, security, and immutability of the data exchanged. For example, If only one integrated system is compromised, this can affect the functioning of the others by reporting false data or requesting wrong functionalities.

In contrast, Blockchain can be defined as a secure distributed database that registers an ordered list of transaction records that are immutably linked together through a chain, on blocks [7]. With the advancement of this technology, other data storage structures were developed without being in a chain, i.e., as a directed acyclic graph (DAG) or a Merkle tree. So, Distributed Ledger Technology (DLT) nomenclature was used to generalize this technology in different structures. Hereafter, DLT and Blockchain will be treated as synonyms in this work.

Blockchain technology can be integrated with FIWARE components, allowing secure and reliable interoperability among systems. Since this system integration with Blockchain can improve its management and security [8]. Moreover, using this integration, it is possible to share data and services transparently, secure, and reliable without centralizing entities. This integration enables the usage of city services to improve the quality of life of the population [9].

Therefore, this work presents an innovative solution to integrating systems securely, interoperable and reliable using middlewares and blockchain in the context of smart cities. This approach can also facilitate the integration of any system to different distributed ledger transactions. Using the NGSI Standards allows the integration of FIWARE components and provides the basis for interoperability and expandability among smart solutions [10].

In this context, the contributions of this paper are threefold. First, we discuss the liabilities of the FIWARE platform to handle the development of reliable and secure applications (Sect. 2). Second, a novel Blockchain adapter for FIWARE, named *Canis Major*, is introduced. *Canis Major* supports transactions in different kinds of DLT and enables the full integration of Blockchain technology with FIWARE components, promoting the basis for smart solutions' interoperability and replication (portability). Third, we described *Taurus*, a new Generic Enabler that aims to listen to a set of Blockchain configured events and store them in FIWARE data store components. Finally, We evaluate our proposed approaches through a quantitative and qualitative assessment using a global vaccination case study against COVID-19.

2 FIWARE plartform

FIWARE is an open middleware developed with support from the European Union (EU), which aims to provide an infrastructure for developing and providing applications for the Future Internet (FI). This platform offers a robust set of

[2] https://www.kaaproject.org/.

[3] https://artemis-ia.eu/project/4-sofia.html.

[4] www.fiware.org.

APIs (Application Programming Interfaces) that facilitate the development of intelligent applications for various industries. In addition, it provides an open-source reference implementation of each of its components, the so-called Generic Enablers (GEs).

These GEs are grouped into technical chapters according to the set of functionalities to which they are related, namely: Interface with the Internet of Things (IoT), which makes it possible to capture updates on context information and translate the necessary actions; Context/API data management and monetization, provides usage control support and the opportunity to publish and monetize part of the managed context data; Processing, analysis, and visualization of context information: provides mechanisms to deal with the intelligent behavior expected from applications and helping end-users in decision making.

These GEs can be assembled together with other components from third-party platforms to accelerate the development of intelligent applications. FIWARE's context information manager, Context Broker, is the central component of the platform since from it, all other components can produce or consume information and perform their functions.

Figure 1 shows the FIWARE overall architecture. The first top layer represents the portals, such as dashboards and open data portals, in a top-down explanation. The second layer comprises GEs responsible for processing the data, i.e., Real-Time and big data processing and Business intelligence (BI). The core of the FIWARE architecture is the Context Broker (named Orion), which receives data from the bottom layer and sends it as NGSI context data for all other GEs. The penultimate layer comprises the GEs interacting with sensors and actuators (last layer). In addition, to control the actuators and send this sensor data to Orion, the IoT Broker and IoT Management transform the sensor data into the standard platform format—NGSI.

The generic enablers on the right of Fig. 1: Account & Payment are responsible for FIWARE accounts and ways to pay. The IDM & Auth is already responsible for Identifier Management and authorizations with data and services access rules based on its ID. On the left, some GEs responsible for interacting with Smart Cities Systems powered by FIWARE, whether to provide some information or orchestrate some service provided by the platform.

Orion Context Broker can provide interoperability among systems. However, Orion does not guarantee the reliability of the data exchanged. When Orion receives a Context (i.e., sensor or actuator data) update, it forwards that change to the stakeholders who subscribe to that context. However, it does not store metadata (when and by who) of this context change. Nevertheless, it does not store metadata (when and by whom) of that context update where more than one system can do these changes. In this way, it is impossible to verify these operations by other systems. Moreover, a malicious system can insert false data into the FIWARE platform, go unnoticed, and affect other systems. Reliability between exchanged data can be one of the main requirements when sharing critical data.

3 Proposed FIWARE blockchain integrators

As previously mentioned, FIWARE can provide interoperability among systems, although no reliability is provided for data exchange. In contrast, Blockchain technology could fulfill such a gap by providing secure and reliable data exchange. Unfortunately, FIWARE does not support integration with any Blockchain technology.

Therefore, we propose two new Generic Enablers (GE) named *Canis Major* and *Taurus* to fulfill these integration issues. These new GEs allow FIWARE-based applications to store data safely and reliably using Blockchain technology using the NGSI data model. These two brand new Generic Enablers are described in the following subsections.

3.1 Canis major GE

Canis Major is the Blockchain adapter that supports transactions in different kinds of DLT. Into the current version 1.1, it supports Ethereum and IOTA[5] as DLTs. It works with two different NGSI versions: NGSI-V2[6] and NGSI-LD[7] (Linked Data) as the communication model. This communication standard enables the integration of components, promoting the basis for smart solutions' interoperability and replication (portability).

Canis Major is implemented in JavaScript, offering a Rest API responsible for configuring a personalized new contract or querying entities already inserted into a Blockchain. Canis Major source code is available under the Apache License at https://github.com/FIWARE-Blockchain/CanisMajor.

The Canis Major workflow in FIWARE platform architecture, as shown in Fig. 2, is as follows:

1. Request from the user consists of the Payload, Header with the token, and DLT_ID (base64 of public key and private key of the Blockchain).
2. Wilma PEP Proxy[8] validate the token and check with the KeyRock[9] Identity Management (IDM) and

[5] https://www.iota.org/.
[6] https://fiware.github.io/specifications/ngsiv2/stable/.
[7] https://ngsi-ld-tutorials.readthedocs.io/en/latest/.
[8] https://fiware-pep-proxy.readthedocs.io/.
[9] https://fiware-idm.readthedocs.io/.

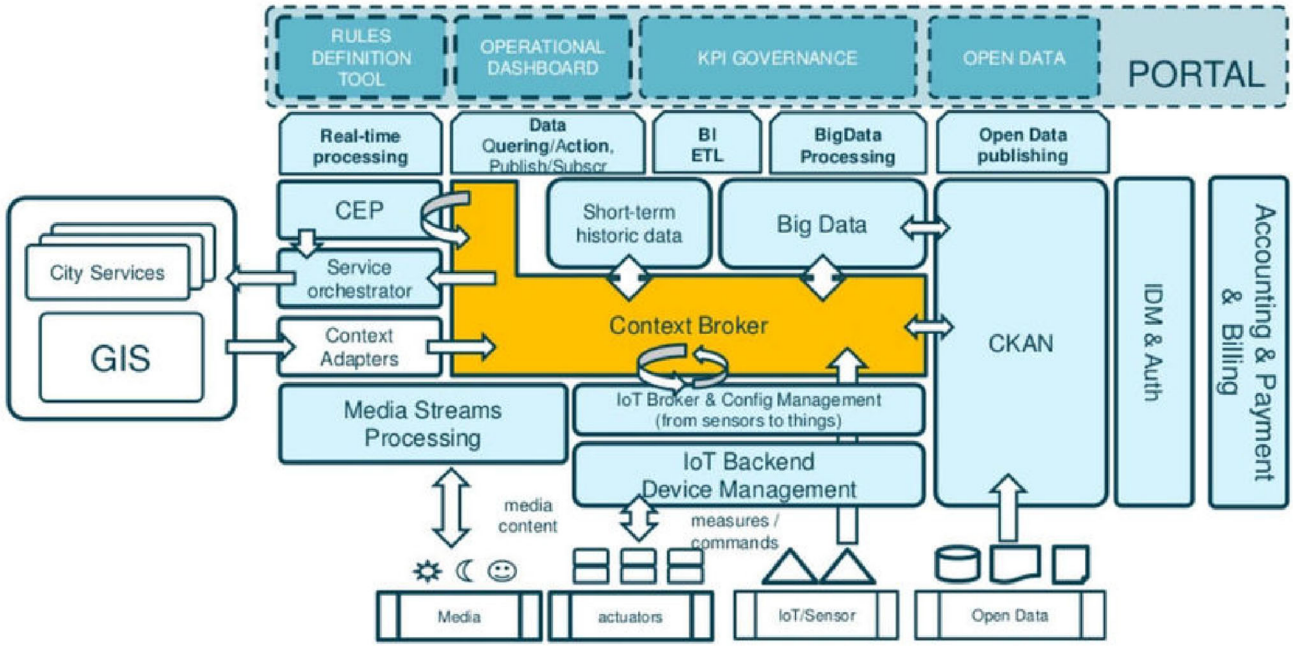

Fig. 1 FIWARE platform architecture. *Source* [11]

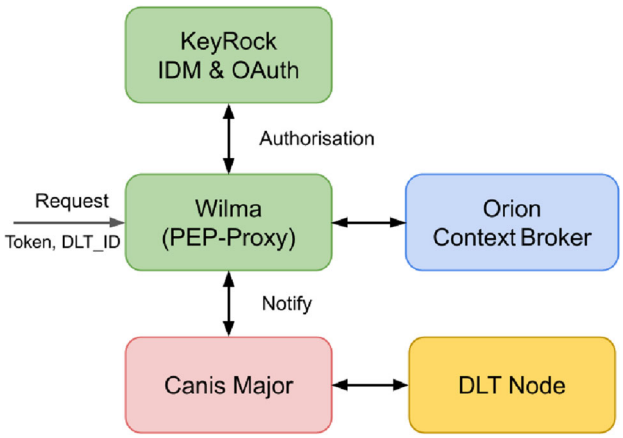

Fig. 2 Canis major integration architecture with FIWARE

validate the user, permission (Authentication and Authorization).

3. Once the user is validated, Wilma forwards the request to the Orion Context Broker[10] and persists it.
4. Once the Payload is stored in Context Broker, Wilma notifies Canis Major of the configuration, such as what attribute of the payload should be stored Blockchain Identity of the user.
5. Further, Canis Major persists the data in a supported Blockchain using an AEI contract.

For the Ethereum clients (such as Geth,[11] Quorum,[12] Besu[13]), it is recommended to use the AEI contract model to describe the Smart Contracts. AEI (Asset, Event, and Identity) Contract Model is written in Solidity[14] language using ERC721[15] standard and can be used to describe your contract. An ERC 721 Contract follows OpenZepplin standards; security audits are trusted by leading organizations building decentralized systems.

The AEI Contract Design, as shown in Fig. 3, where an Entity or Asset has a unique identity with a 1:1 mapping (one asset to one identity). At the same time, one asset/entity has a 1:n mapping with Events or Metadata. An Asset has a 1:n relationship with any other assets. It is compatible with FIWARE once Canis Major Adaptor configures it using Canis Major rest API to store any data in this model type in Blockchain. It is a standard interface for non-fungible tokens (NFTs), also known as deeds. The following standard allows for implementing a new standard API for NFTs within smart contracts, making it possible to track and transfer NFTs more easily.

[10] https://github.com/FIWARE/context.Orion-LD.

[11] https://geth.ethereum.org/.

[12] https://consensys.net/quorum/.

[13] https://www.hyperledger.org/use/besu.

[14] https://docs.soliditylang.org/.

[15] https://eips.ethereum.org/EIPS/eip-721.

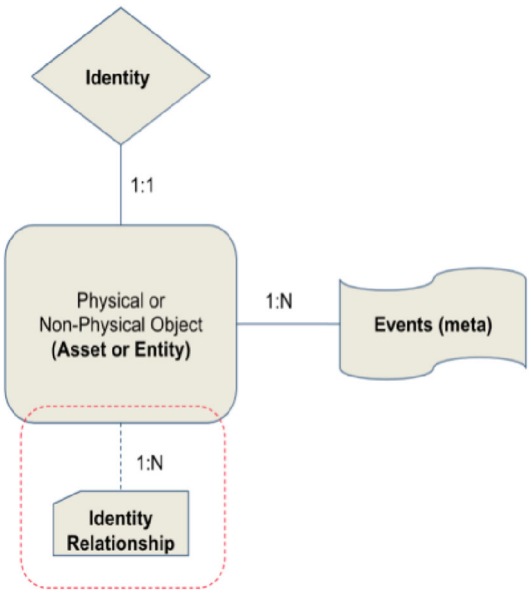

Fig. 3 AEI contract design

Currently, version 0.2 of Canis Major supports different storage types (IOTAMaM,[16] IPFS[17], Merkle Tree[18] and a personalized Contract[19]) according to the need of the developed application.

3.2 Taurus GE

Taurus is another Generic Enabler under development in the FIWARE platform. It is a Distributed Ledger Technology listener that supports various implementations of DLT. This listener aims to listen to configured events and store this event's data in FIWARE. This component complements FIWARE as an off-chain database once it stores the information from DLT as context data through Orion to be delivered to the services that have subscribed to it.

Taurus is implemented in Python3 and composed of a Rest API responsible for creating a personalized listener to request specific data updates in a Blockchain at intervals of a specific time and send it to Context Broker, see Fig. 4. It can be used by solutions that only have to consult data from DLT, being Canis Major's opposite that adds data in DLT. Taurus source code is available under the Apache License at https://fiware-Blockchain.github.io/Taurus/.

In its current version, Taurus is in version 0.0.1, where for now, it only has integration with Ethereum, where it uses the

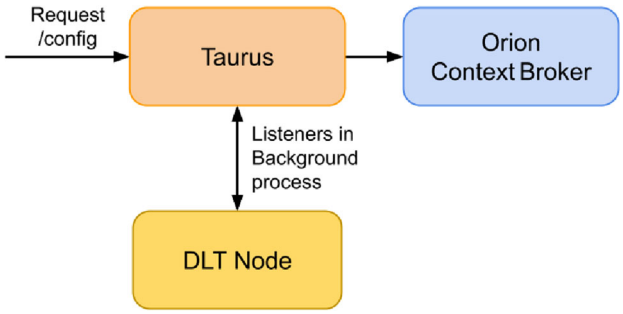

Fig. 4 Taurus integration architecture with FIWARE

Web3[20] library to make requests. Future versions are expected to integrate with IOTA and Hyperleadger Fabric.

The Listing 1 exemplifies, in the NGSI standard, the needed information to create a new Taurus listener. Lines 2, 3, and 4 contain the identifier name of the event to be monitored, the time interval in seconds between requests, and the *contractAddress* of the operation to be stored in Ethereum. Lines 5 to 30 contain the Application Binary Interface (ABI)

```
1 {
2     "id":"event-identifier",
3     "interval":10,
4     "contractAddress":"0x1349f3e1b8d71
          effb47b840594ff27da7e603d17",
5     "abi":[
6         {
7             "inputs":[
8                 {
9                     "name":"_x",
10                    "type":"uint256"
11                }Brazil and Australia
12            ],
13            "anonymous":false,
14            "name":"LogEvent",
15            "type":"event"
16        },
17        {
18            "inputs":[
19                {
20                    "indexed":true,
21                    "name":"_sender",
22                    "type":"address"
23                }
24            ],
25            "anonymous":false,
26            "name":"LogOtherEvent",
27            "type":"event"
28        }
29    ]
30 }
```

Listing 1 Taurus configuration in NGSI format

[16] https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e/.

[17] https://ipfs.io/.

[18] https://www.javatpoint.com/blockchain-merkle-tree.

[19] Using an eth_sendRawTransaction to create your own Smart Contract.

[20] https://web3js.readthedocs.io/.

that describes the event as a function with each input parameter (name and type) according to Ethereum's Smart Contracts written in Solidity language.

The Fig. 5 shows a sequence diagram with Taurus running details. The creation process for each new Taurus listener starts with a POST "/config" request passing the contract Address and attributes to listen to when a new transaction was added in a DLT, as previously detailed. With this information, Taurus creates the specific listener that, from time to time, verifies if a new transaction is added with the same ContractAddress previously registered. A listener captures this information and sends it to FIWARE context Broker.

4 Evaluation

The evaluation of the two proposed FIWARE GEs were divided into three major phases: (1) the design and implementation of the target case study, (2) the setup of software and hardware for the testing environment, and (3) the assessment of target case study (developed in phase 1). The following subsection will present in detail each of these phases.

4.1 Target case study

COVID-19, caused by the SARS-Cov-2 virus, spread worldwide, becoming a pandemic and affecting people's travel worldwide. Some restrictions were adopted to control the pandemic propagation, such as restricting travel from uncontrolled pandemic countries. With the emergence of different vaccines to reduce the death tax caused by SARS-Cov-2, international travel will be back for business, tourism, and vacation. However, each country already has its vaccination control system. Integrating them in a decentralized and reliable approach is essential, providing interoperability between them. For instance, suppose a person wants to travel from Brazil to Germany. The European Union has defined some criteria for vaccination migration rules to approve immigration ranging from certain mandatory types of vaccination and a specific time after each dose of vaccine.

In this context, we select a real post-pandemic vaccination scenario as a target case study to assess our proposed approach's feasibility and effectiveness. This target case study should be able to certify with a high degree of certainty that someone was vaccinated. Moreover, this implementation should fulfill specific criteria defined by a destination country more securely than a paper certificate that can be easily defrauded or lost. Overall, this case study should be able to integrate most of the vaccination systems worldwide.

One option to implement this case study is by using FIWARE, which utilizes NGSI Standards, allowing the integration of components and providing the basis for the interoperability and portability of smart solutions. Unfortunately, FIWARE alone does not guarantee that data exchanged between vaccination systems are reliable and secure. Therefore, Blockchain can be used together with NGSI as an immutable decentralized transaction ledger to integrate systems to overcome these challenges.

This case study was selected because it met several relevant criteria for our intended evaluation. First, it is a fundamental and non-trivial system. This case study is particularly rich in several recurring concerns and technologies common in day-to-day software development for smart city solutions. Second, our case study's design and implementation choice has been extensively discussed in Sect. 4.1.1, allowing other researchers to correlate our results with future studies.

The following sub-section presents how the proposed Fiware GEs (Canis Major and Taurus) can be used to provide secure interoperability between systems to integrate vaccination systems worldwide in a post-pandemic vaccination scenario (see Fig. 6).

4.1.1 Implemented architecture

Figure 7 depicts the case study implemented architecture. It allows any vaccination control system to submit vaccination certificates as transactions in NGSI format from the FIWARE middleware components to any DLT implementation supported by the proposed Canis Major and Taurus GEs without any intervention of humans.

The proposed architecture comprises many components, which are described as follows:

Draco The Draco Generic Enabler[21] is an alternative data persistence mechanism for managing context history. It is based on Apache NiFi and is a data flow system based on the concepts of flow-based programming. It supports robust and scalable directed graphs of data routing, transformation, and system mediation logic and offers an intuitive graphical interface. Draco was used to capture data from databases of vaccination systems that have no connection to FIWARE. In addition to capturing the data, it is also responsible for transforming it into the NGSI format (following the model in Listing 2) and sending it to Canis Major via Pep Proxy Wilma. In this manner, it becomes possible to integrate the data of vaccination systems in an automated way worldwide.

Canis major The Canis Major Generic Enabler is under development to be a FIWARE Generic Enabler DLT Adapter. It is responsible for connecting data (in NGSI

[21] https://fiware-draco.readthedocs.io/.

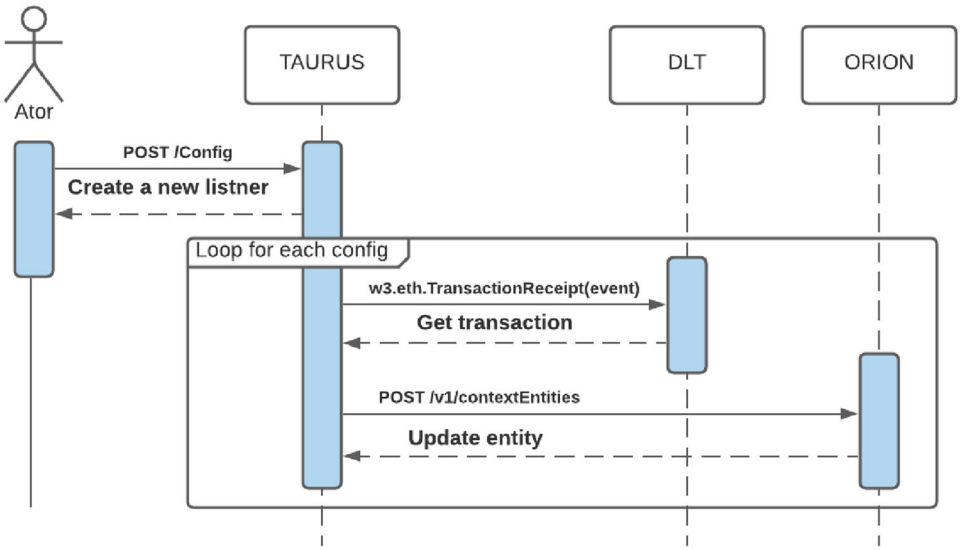

Fig. 5 Sequence diagram of Taurus integration with DLT

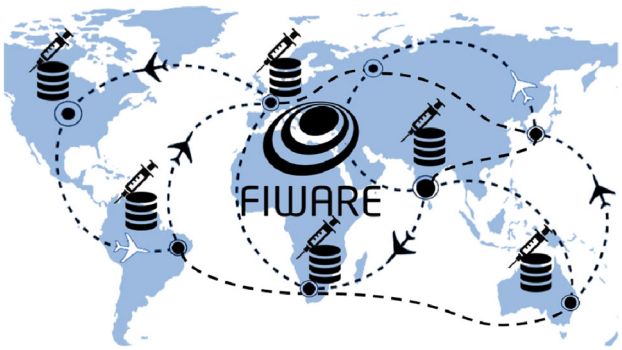

Fig. 6 Global integration vaccination scenario using FIWARE

format) from any FIWARE Generic Enabler stored in a DLT supported by Canis Major. When data in an NGSI format arrives on Canis Major, it creates a new transaction. The DLT network peers validate this transaction according to the Blockchain consensus mechanism. After transaction creation, a confirmation is sent back to Canis Major to notify the requesting system through the Orion Context Broker.

Canis Major allows a user/system to simultaneously store any data in one or more than one Blockchain. For this, Canis Major's API provides three main methods: (i) *store*, sending the data and the determined Blockchain as a parameter, and it will return the transaction hash; (ii) *fetch*, sending a transaction hash as a parameter, it will return the transaction data; (iii) *migrate*, sending a transaction and two Blockchain ids as a parameter. It will fetch the

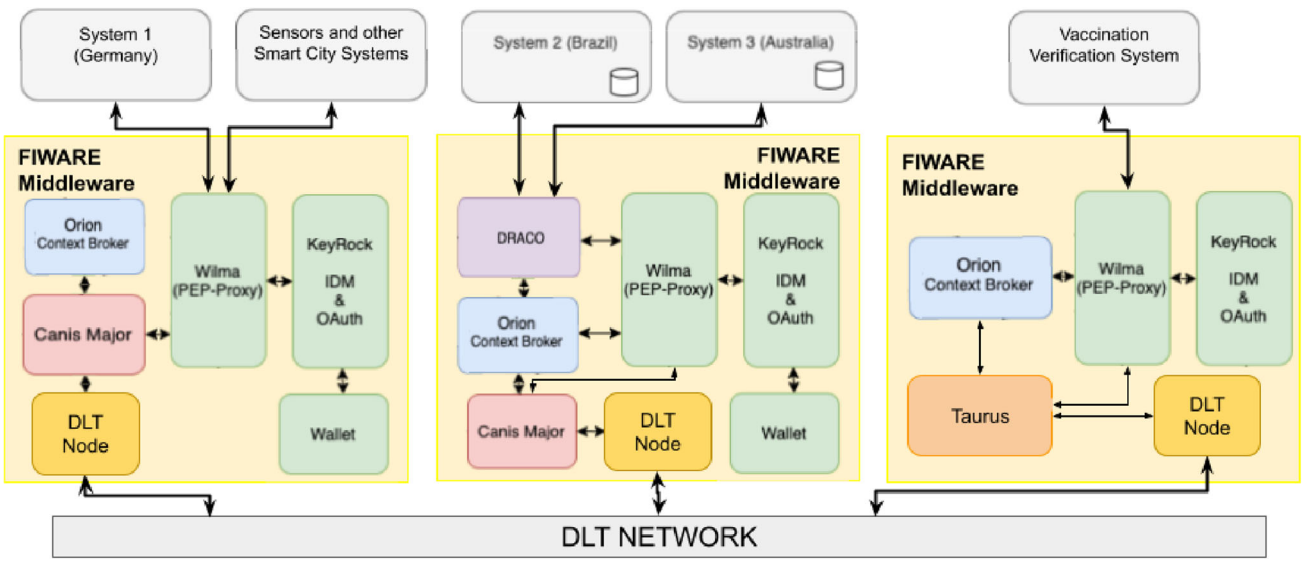

Fig. 7 Proposed architecture—FIWARE components integration

transaction from one Blockchain and store it in the other Blockchain.

Wilma The Wilma proxy Generic Enabler brings support to proxy functions within OAuth2-based authentication schemas. It also implements PEP functions within an XACML-based access control schema that has been part of security FIWARE GEs. Wilma was used as a proxy to provide security through authentication schemas.

KeyRock (IDM & Auth) The Keylock Identity Management Generic Enabler provides OAuth2-based authentication of users and devices. In addition, it also provides Identity Management (IDM). KeyRock works together with Wilma to redirect only authorized users or systems to the internal components of the proposed solution. KeyRock was used in addition to the authentication and authorization security service as a DLT wallet controller.

Wallet (Accounting & Payment) The Wallet component is responsible for managing all the keys used within the FIWARE platform and the keys used by DLTs.

Orion (Context Broker) The Orion Context Broker Generic Enabler is the core and mandatory component of any "Powered by FIWARE" platform or solution. The context broker usage enables the management of context information in a highly decentralized and large-scale manner. Orion is a publish-subscribe context broker that notifies the consumers, sending a message in NGSI format when a new context was updated or created.

In the proposed architecture, Orion integrated vaccination systems powered by FIWARE to a DLT implementation through Canis Major GE. When a new person is vaccinated, this context is inserted on Orion that notifies Canis Major to create a transaction in chosen DLT.

DLT This component represents a DLT implementation supported by the proposed Canis Major Adaptor. Nowadays, the current Canis Major version 1.1 supports only IOTA and Ethereum using the AEI Contract. In the subsequent updates, we will integrate it into other DLT implementations such as Hyperledger Fabric[22] among others.

Taurus Taurus can be used by the Vaccination Verification system that only needs to consult specific vaccination information without the need to search for historical information within the DLT. Therefore, Taurus will be an off-chain database that will communicate through the Orion Context Broker to notify systems that need this information.

The components in the topmost part of Fig. 7 represent the many vaccination systems that may be integrated. There are two types of systems to be integrated into this context. The first type is developed systems using the FIWARE middleware platform that uses the NGSI format. At the same time, the second type comprises systems that were not developed using FIWARE middleware, which need to be integrated into the platform first to be later integrated into DLT. The way to integrate each system changes according to the type, as shown in Fig. 7.

System 1 of Fig. 7, which represents Germany's vaccination control system, was developed using FIWARE middleware composed of different GEs and already uses NGSI to communicate its internal components. According to the defined data model, it needs to integrate the Canis Major GE to its middleware and send it to the configured DLT. While System 2 and System 3 of Fig. 7 represent the vaccination control system from Brazil and Australia, respectively. The fact that these countries use the same FIWARE infrastructure is for illustrative purposes only. In real life, ideally, each of them has its own separate.

Brazil and Australia systems do not integrate with the FIWARE middleware platform. Then, DRACO must provide the system integration with a DLT. Each FIWARE Integrator Node can be used and managed by one or more countries according to the scalability needed by the controlling vaccination system. This solution can be used to integrate vaccination data from different systems of the same country.

The FIWARE Integrator Node integrates the system database to Canis Major (a DLT Adapter) and Orion Context Broker GE using the data format presented at Listing 2. After integrating all systems on the FIWARE platform, an initial setup to migrate the vaccination data to a DLT (if the system already has data stored before integration), the proposed solution node will be able to receive new requests.

The new request pass through the Wilma Pep Proxy that uses KeyRock to authenticate this request and authorize the access, as shown in Fig. 7. Once the data arrives at Canis Major, it is automatically sent as transactions to the previously configured DLT. A new transaction is created in the DLT automatically when the vaccination system inserts new vaccination data, as shown in the Sequence Diagram of Fig. 8. This process starts when a new request arrives to be authenticated by the security GEs (Wilma and KeyRock). This request may come from the system developed using FIWARE or when Draco captures the event of a new vaccine registration in the system database without integration with FIWARE.

After authentication, Wilma sends the request to Orion to be stored, which returns the storage status. Simultaneously, Wilma also notifies Canis Major to validate the context mapping using the AEI contracts. Canis Major would send the transaction to the DLT approval and store it if it were correct.

Sequence Diagram of Fig. 8 also shows the possibility of any system in the network verifying the vaccine

[22] https://www.hyperledger.org/use/fabric.

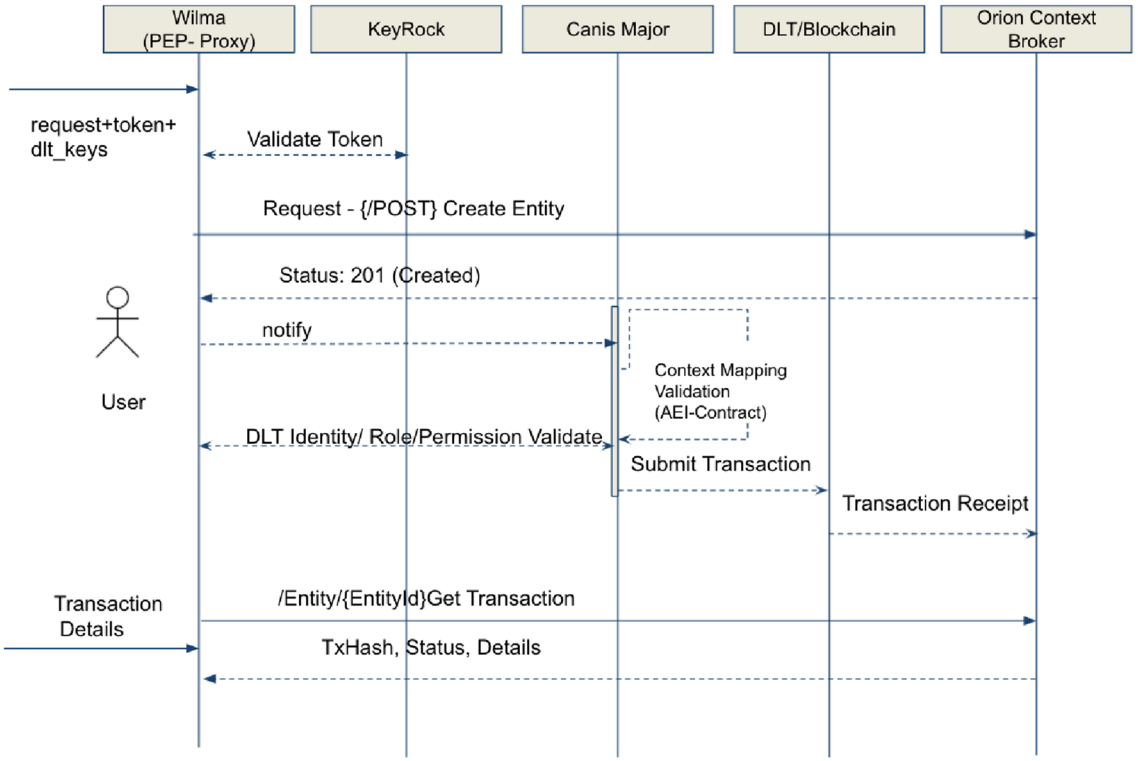

Fig. 8 DLT transaction sequence diagram

certificated using the Orion Context Broker GE. To do that, is necessary make a *GetTransaction* request to Orion passing the entity ID, in this case the *VaccinationCertificate* ID. In this manner, verifying the integrity and veracity of a person's vaccination data during their immigration will be possible.

This architecture also allows a person to be vaccinated in a first dose against Covid-19 in one country and his second dose in another since the systems can exchange information with each other and understand it securely and trust way. Thus, the proposed architecture promotes interoperability among System 1, System 2, and System 3.

As additional functionality, after the integration, any system can be easily added to a Smart Solution by merely using additional FIWARE GEs[23] or third-party components to integrate with the Orion component. This integration is simplified by FIWARE modularity since all components comply with the FIWARE NGSI standard interface, eliminating vendor lock-in.

Another alternative is to use GE Taurus to access only the information registered in the DLT. In Fig. 7, a vaccination verification system uses Taurus as an off-chain database. After the system identification and authorization process, this verification request checks in the Orion Context Broker if the vaccination certificate presented by the user is authentic without looking for it in the DLT.

4.1.2 Data format

In addition to the implemented architecture, it is also necessary to define a data format so that it is possible for all systems to understand the data exchanged, providing interoperability in this way. Thus, as the FIWARE platform already uses the NGSI data model format to exchange information between their Generic Enablers (GEs), this same format was used in the solution.

Then, we define a *VaccinationCertificate*[24] using the NGSI data format to store vaccination information as a transaction, based on paper [12]. Listing 2 shows a example of a vaccination certificate. To the model proposed by [12], was added patient information (see lines 14 to 18) and information about the vaccination local (see lines 10 to 13, where respectively presents a batch number, local of administering, country of vaccination, health professional). Also, vaccine information was added (see lines 19 to 22, presenting vaccine name, act code, medicinal product name, and marketing authorization holder) and proof with verification method (see lines 25 to 28).

[23] https://www.fiware.org/developers/catalogue/.

[24] https://github.com/smart-data-models/dataModel.COVID19/.

```
{
  "id": "urn:ngsi-ld:COVID19:ca3f1295-500c-4aa3-b745-d143097d5c01",
  "type": "VaccinationCertificate",
  "description": "COVID-19 Vaccination Certificate",
  "issuanceDate": "2021-01-01T01:20:00Z",
  "expirationDate": "2021-01-01T01:20:00Z",
  "dateCreated": "2021-01-01T01:20:00Z",
  "issuer": "urn:ngsi-ld:did:f85bb01d-28ed-4847-aa07-57bb53bc3d94",
  "credentialSubject": {
    "batchNumber": "1183738569",
    "administeringCentre": "MoH",
    "healthProfessional": "MoH",
    "countryOfVaccination": "BR",
    "recipient": {
      "givenName": "Stefano",
      "familyName": "Loss",
      "gender": "male",
      "birthDate": "1995-01-01",
      "vaccine": "COVID-19",
      "atcCode": "J07BX03",
      "medicinalProductName": "COVID-19 Vaccine Moderna",
      "marketingAuthorizationHolder": "Moderna Biotech"
    }
  },
  "proof": {
    "created": "2021-01-01T01:20:00Z",
    "proofValue": "eyJhbGciOiJFZERTQSIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..vD_vXJCWdeGpN-qKHDIlzgGC0auRPcwp3O1sOI-gN8z3UD4pI0HO_77ob5KHhhU1ugLrrwrMsKv71mqHBn-dBg",
    "verificationMethod": "urn:ngsi-ld:did:7bdc0b3e-12cc-4164-ac21-f026ec29ba49"
  }
}
```

Listing 2 NGSI format for store vaccination information

Besides *VaccinationCertificate* NGSI data format has defined an architecture using FIWARE Generic Enablers aims to create a network to share vaccination information in a decentralized and verifiable way using DLT technology, as shown in Fig. 6. Therefore, each country will integrate its vaccination control system using this proposed architecture that facilitates integration with a DLT implementation supported by the FIWARE platform.

4.2 Experimental validation

In order to assess the proposed architecture, we have performed a stress test against the implemented case study. Considering that the proposed solution aims to integrate any system of a chosen DLT implementation easily (being an adapter), the time until the creation of a new transaction, in addition to the middleware, will depend on the consensus mechanism of the DLT used. For this reason, only the time related to the receipt of the request, credential validation, and context mapping validation (that use the AEI-Contract) were taken into account.

The stress test aims to verify the average response time in milliseconds (ms) for many simultaneous requests to register a new vaccinated person record, which is quite common in a vaccination management system in a mass vaccination scenario.

To facilitate the test process, we choose to make a POST *Create Entity()* request of the type *VaccinationCertificate* with a JSON body similar of Listing 2. However, this request sends an empty *"proof"* object, this field will be filled after DLT validation, and the next time this entity is consulted, *"proof"* will return its information if the transaction was approved.

The number of simultaneous users making requests increases in each test batch for these tests. And, each concurrent user requests 30 the POST *Create Entity()*. The test batch is made with 10, 50, 100, and 500 simultaneous users, where each user sends 30 requests. Therefore, the total of requests equals the number of users times 30.

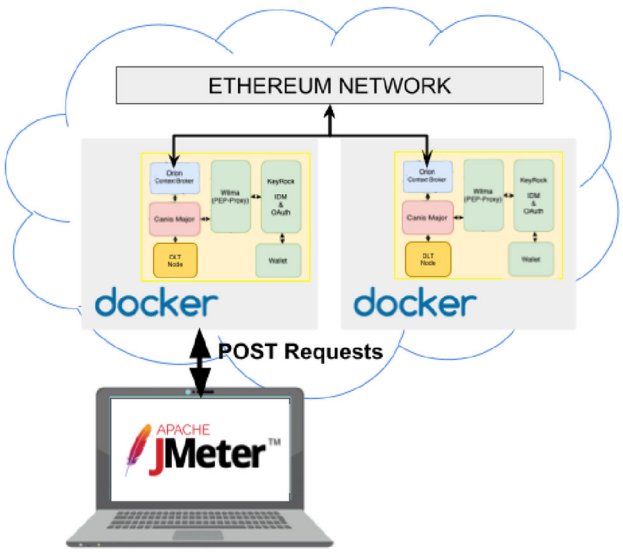

Fig. 9 Test environment configuration

4.2.1 Test environment

A cloud was used to deploy and simulate the proposed solution in a distributed way, see Fig. 9, where Docker Compose[25] was used to configure and deploy containers of all components at the FIWARE Integrator Node. The cloud used was 2vCPU, 4GB RAM, and Ubuntu 18.04.2 LTS.

After that, Apache JMeter,[26] running outside the cloud, was used as a load testing tool for measuring and analyzing the solution's performance for this test. Some requisitions were made to simulate a high number of simultaneous requests.

4.2.2 Test results

For these tests, it has been configured to reach up to 500 instances. After this test, the average time of each requisition for the different scenarios (10, 50, 100, and 500 users) was analyzed.

Table 1 shows the number of simultaneous users, number of samples, average time (in milliseconds), minimum time, maximum time, standard deviation, error percentage, and throughput of each test. It is essential to highlight the average response time for each request to create a transaction; for 10, 50, 100, and 500 concurrent users are 3 ms, 35 ms, 99 ms, and 245 ms, respectively, without any error.

5 Related work

This paper is a revised and expanded version of the paper [13] including a description of the FIWARE ecosystem; Added more information about FIWARE Generic Enabler (GE) for blockchain Canis Major; a new Related Work presenting some solutions that already use Blockchain in the context of Smart Cities in a generic way in diverse areas; a new Taurus GE responsible for listening to DLTs events and notifying other GEs or services throw Orion Context Broker; Added Taurus on the architecture of the Global Vaccination Scenario.

Many other solutions have been using Blockchain in the context of Smart Cities. Some of these solutions use Blockchain to meet the requirements of smart cities for general purposes, while others are concerned with creating new Smart cities applications in the specifics domain. Other proposals present the use of APIs to facilitate Blockchain usage. The following subsections will be presented some related works divided into two categories: Blockchain for Smart Cities and Vaccination case study.

5.1 Blockchain for smart cities

The [14] presents how the emerging blockchain technology can be implemented into smart city services. This survey also discusses some areas where Blockchain can be implemented to enhance the growth of smart cities like smart government, smart transportation, smart management, trade & finance, smart grid, smart healthcare, smart home, and others.

Already [15] proposes an efficient, secure, and scalable distributed architecture for the continued growth of data amount and number of IoT devices producing real-time information in the Smart Cities context. Therefore, the architecture is divided into two parts: core network and edge network. The core network comprises miner nodes with high computation and storage resources, while the edge node has limited storage and computation power. The edge node receives a new transaction request for the services required by the IoT device/user, and it sends a transaction request to each miner in the core network.

The challenges of integrate Blockchain and IoT environments is presented in [16] and [17]. While [16] is concerned about the computation capacity and scalability needed for the IoT environments present in smart cities. Whereas [17] proposes a solution for moving IoT devices using Blockchain. It presents a blockchain-enabled system that relies on edge device service capabilities to compose services and provide communication connectivity for IoT and robotic devices.

CitySense is proposed in [18] that uses the Ethereum platform to record measurements arriving from the IoT network of sensors. CitySense is structured in four layers: (i) interface, the application layer; (ii) database and control, Blockchain layer; (iii) communication, network, and protocols layer; and (iv) physical, sensor, and Smart Objects. This structure aims to make sensor and object data available to facilitate the development of applications in the context of smart cities.

In [19] presents the Orthus platform that uses Blockchain to provide interoperability between systems in the context of Smart Cities. It uses the FIWARE Orion context broker to communicate with peers in the NGSI standard.

Another Blockchain framework for Smart Cities is SpeedyChain, proposed by [20] to provide a scalable and private architecture for trusted data sharing. It has allowed smart vehicles to share data while maintaining privacy, integrity, resilience, and non-repudiation decentralized and tamper-resistant.

Although the solutions presented above are concerned with Blockchain usage, none proposes integrating smart cities' middleware with existing Blockchain to facilitate

[25] https://docs.docker.com/compose/.

[26] https://jmeter.apache.org/.

Table 1 JMeter summary report table (time in millisecond)

Users	Samples	Average	Min	Max	Deviation	Error (%)	Throughput transactions per second (s)
10	300	3	2	13	2.4	0.0	520.7
50	1500	35	2	107	31.74	0.0	483.1
100	3000	99	2	352	94.27	0.0	462.7
500	15,000	245	3	1634	401.71	0.0	449.6

the development of new solutions or integrate with existing solutions using NGSI communication standards.

5.2 Vaccination case study

Blockchain is already used in some applications of vaccination scenarios. Some are related to vaccination data storage and verification in this context, and others are worried about vaccine production in a supply-chain context. In contrast, other solutions present Blockchain usage in a vaccine passport used in a travel scenario. Some of these solutions will be presented below.

One solution is in the early stages in New York, an app developed by IBM named "Excelsior Pass." According to [21] it can "confirm vaccination or negative SARS-CoV-2 test status through confidential data transfers to fast-track business re-openings". The authors claim that digital health passes can be used to return to commerce, recreation, and travel. Similarly, WHO started to use a belated solution in August 2021 called Digital Documentation of COVID-19 Certificates: Vaccination Status (DDCC:VS)[27] to provide some systems integration. This solution is standardization to share vaccination information between countries, where it is necessary to change the systems to provide data in the specified format by DDCC:VS.

Another app, proposed by [22] aims to provide proof of record owned, allowing the end-user to reveal test results and Vaccination Certification against COVID-19. It uses a consortium Ethereum-based Blockchain to store this data and support a decentralized verification of data confidentiality.

In [23] a Blockchain-based system is proposed for a vaccine supply chain that stores information from production to inoculation. Its objective is to supervise the total vaccine production and distribution process, supporting vaccine traceability. It also detects near expired vaccines, sending reminders automatically to regulators and institutions in the vaccine supply chain. However, it is not concerned with the interoperability between the systems involved.

The use of Blockchain to store and verify the vaccination process is also proposed by [24, 25] and [26]. They show use cases that use Blockchain, with acceptable performance, to store and validate vaccination certificates, but they are not worried about interoperability between existing vaccination control systems.

In a preliminary stage, a vaccine passport using Blockchain with personal privacy is proposed by [27]. The most similar article states, "Skepticism in global data sharing and communication becomes a bottleneck for health technology development and an obstacle for global management of infectious diseases." A digital vaccine passport can improve data provenance, security, and integrity in a migration scenario.

While the solutions presented above are concerned with creating new systems to meet local demands, it is necessary to consider integrating most of the systems used by different countries. Therefore, it is necessary to provide interoperability between these systems, which are not satisfied by these solutions. Therefore, sharing health information through a safe and reliable middleware solution providing interoperability between vaccination systems already used by governments worldwide can be beneficial in containing pandemics.

6 Conclusions and future work

Blockchain technology has a lot to contribute to Smart City solutions. One of the applications that Blockchain can be helpful is in the vaccination scenario. Mass vaccination is used to control Covid-19. Besides, it is also necessary to guarantee that someone was vaccinated when they migrate to another country.

Thus, this paper presented a Generic Enabler named Canis Major to integrate the system using Blockchain and the middleware FIWARE securely and verifiable. Moreover, we proposed an architecture using the Canis Major DLT Adaptor to assess the proposed solution to integrate vaccination systems using Blockchain in a global scenario. We also presented the Taurus, a DLT listener FIWARE GE responsible for catching new data stored in DLT from a

[27] http://apps.who.int/iris/handle/10665/344456.

specific event, making it secure and reliable data consumption from DLT.

One of the gaps that need to be investigated is the possibility of testing this architecture proposed by considering and comparing different DLTs implementations. Since Canis Major and Taurus are still in development, as soon as it supports new DLT implementations, it is necessary to perform performance tests to compare the time needed to create new transactions in each DLT implementation.

Future work must integrate a generalization involving a decentralized registry for cryptographic keys, allowing every public key to have its unique address, known as a Decentralised Identifier (DID). DID should be Integrated using the solution as KeyRock GE and Hyperledger Indy to generate and maintain the keys as a Self-sovereign identity (SSI) that gives individuals control of their digital identities.

With the proposed architecture, it is possible to store vaccination information by submitting it as a transaction in NGSI format from FIWARE Context Broker to any Blockchain implementation supported by Canis Major, without any human intervention in a secure and verifiable way. It also provides interoperability, sharing vaccination data and new functionalities between vaccination systems worldwide that potentially use this solution. As a result, it can improve data provenance, security, and integrity in a migration scenario and help us back to everyday life as before the pandemic.

Author contributions All authors contributed to the study conception and design of Canis Major and Taurus. HPS and SL performed the GE implementation and contributed to writing the manuscript. SL wrote the first draft of the manuscript and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by the Smart Metropolis Project http://smartmetropolis.imd.ufrn.br.

Data availability The code and datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing interest The authors declare that they have no competing interests.

References

1. Wenge, R., Zhang, X., Dave, C., Chao, L., Hao, S.: Smart city architecture: a technology guide for implementation and design challenges. China Commun. **11**(3), 56–69 (2014). https://doi.org/10.1109/CC.2014.6825259
2. Boyko, C., Cooper, R., Davey, C., Wootton, A.: Addressing sustainability early in the urban design process. Manag. Environ. Qual. **17**(6), 689–706 (2006). https://doi.org/10.1108/14777830610702520
3. Moss Kanter, R., Litow, S.S.: Informed and interconnected: a manifesto for smarter cities. Harvard Business School General Management Unit Working Paper (09-141) (2009)
4. Miller, P.: Interoperability: what is it and why should i want it? Ariadne 24 (2000)
5. Guédria, W., Naudet, Y., Chen, D.: Interoperability Maturity Models—Survey and Comparison, pp. 273–282. Springer, New York (2008)
6. Lea, R., Blackstock, M.: City hub: a cloud-based IoT platform for smart cities. In: IEEE 6th International Conference on Cloud Computing Technology and Science, pp. 799–804. IEEE (2014)
7. Rull Aixa, D.: Analysis and study of data security in the internet of things paradigm from a blockchain technology approach (2018)
8. Berdik, D., Otoum, S., Schmidt, N., Porter, D., Jararweh, Y.: A survey on blockchain for information systems management and security. Inf. Process. Manag. **58**(1), 102397 (2021)
9. Sun, J., Yan, J., Zhang, K.Z.: Blockchain-based sharing services: what blockchain technology can contribute to smart cities. Financ. Innov. **2**(1), 26 (2016)
10. Context Information Management (CIM); NGSI-LD API: Tech. Rep., Context Information Management (CIM) and ETSI Industry Specification Group (ISG) (2020). https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.02.02_60/gs_CIM009v010202p.pdf
11. Saracco, R.: Smart cities and tech evolution—xxvi service infrastructure—fiware. https://www.eitdigital.eu/newsroom/blog/article/smart-cities-and-tech-evolution-xxvi-service-infrastructure-fiware/
12. Alamo, T., Reina, D.G., Mammarella, M., Abella, A.: Covid-19: open-data resources for monitoring, modeling, and forecasting the epidemic. Electronics **9**(5), 827 (2020). https://doi.org/10.3390/electronics9050827
13. Loss, S., Singh, H.P., Cacho, N., Lopes, F.: Using fiware and blockchain in post pandemic vaccination scenario, pp. 143–150. IEEE (2021)
14. Makani, S., Pittala, R., Alsayed, E., Aloqaily, M., Jararweh, Y.: A survey of blockchain applications in sustainable and smart cities. Clust. Comput. 1–22 (2022)
15. Sharma, P.K., Park, J.H.: Blockchain based hybrid network architecture for the smart city. Futur. Gener. Comput. Syst. **86**, 650–655 (2018)
16. Tseng, L., Yao, X., Otoum, S., Aloqaily, M., Jararweh, Y.: Blockchain-based database in an IoT environment: challenges, opportunities, and analysis. Clust. Comput. **23**(3), 2151–2165 (2020)
17. Al Ridhawi, I., Aloqaily, M., Karray, F.: Intelligent blockchain-enabled communication and services: solutions for moving internet of things devices. IEEE Robot. Autom. Mag. (2022)
18. Ibba, S., Pinna, A., Seu, M., Pani, F.E.: Citysense: blockchain-oriented smart cities, pp. 1–5 (2017)
19. Loss, S., Cacho, N., Lopes, F., Valle, J.M.: Orthus: a blockchain platform for smart cities. IEEE (2019)
20. Michelin, R.A., et al.: Speedychain: a framework for decoupling data from blockchain for smart cities, pp. 145–154 (2018)
21. Gostin, L.O., Cohen, I.G., Shaw, J.: Digital health passes in the age of covid-19: are "vaccine passports" lawful and ethical? JAMA (2021)
22. Eisenstadt, M., Ramachandran, M., Chowdhury, N., Third, A., Domingue, J.: Covid-19 antibody test/vaccination certification:

there's an app for that. IEEE Open J. Eng. Med. Biol. **1**, 148–155 (2020)
23. Yong, B., et al.: An intelligent blockchain-based system for safe vaccine supply and supervision. Int. J. Inf. Manag. **52**, 102024 (2020)
24. Hernández-Ramos, J.L., et al.: Sharing pandemic vaccination certificates through blockchain: case study and performance evaluation. arXiv preprint arXiv:2101.04575 (2021)
25. Rotbi, M.F., Motahhir, S., Ghzizal, A.E.: Blockchain technology for a safe and transparent Covid-19 vaccination. arXiv preprint arXiv:2104.05428 (2021)
26. Deka, S.K., Goswami, S., Anand, A.: A blockchain based technique for storing vaccination records, pp. 135–139. IEEE (2020)
27. Tsoi, K.K., et al.: The way forward after Covid-19 vaccination: vaccine passports with blockchain to protect personal privacy. BMJ Innov. **7**(2), 337–341 (2021)

Stefano Loss is a Ph.D. student at the Dept. of Informatics and Applied Mathematics, Federal University of Rio Grande do Norte, Natal, Brazil. His main research interests are Blockchain, Distributed Systems, Middleware, Fiware, Smart Cities, Interoperability, Microservices, and System-of-system. He has been the recipient of some prestigious awards including, the 2019 IEEE International Smart Cities Conference (ISC2) Best Paper Award and The Third IEEE International Conference on Blockchain Computing and Applications (BCCA) Best Paper Award.

Har Preet Singh received a Master's degree in Computer Science specializing in Distributed Networks from TU-Berlin, Researcher at Fraunhofer FOKUS, Technical Expert and Evangelist at FIWARE, and Blockchain Researcher at Ignite (formally Terndermint), Cosmos Network.

Nélio Cacho is an associate professor at the Federal University of Rio Grande do Norte (UFRN) in Brazil. He holds a Ph.D. in Computer Science from the University of Lancaster (UK). He is a researcher at DIMAp since 2010, as a member of the Distributed Systems LAB, with research interests in Smart Cities solutions, mobile and ubiquitous environments. Cacho has worked in software engineering and distributed systems areas for the last 20 years. His contributions emerged from industrial and academic collaborations in distinct projects with different funding scales. He currently coordinates the Smart Metropolis project which has already deployed many smart solutions to improve the quality of life of Natal and many other cities in Brazil. He is the author or co-author of over 100 peer-reviewed scientific papers, including many best paper awards at international conferences.

Frederico Lopes is Associate Professor at the Federal University of Rio Grande do Norte (UFRN) since 2012. He was a post-doctoral researcher at the University of British Columbia (UBC), Canada. He has experience in Computer Science, acting on the following subjects: ubiquitous applications, context-based middleware, pervasive computing, smart cities, and cloud computing. He is co-author of dozen smart city systems deployed in Rio Grande do Norte State.