computers

MDPI

Article

Combining Blockchains, Smart Contracts, and Complex Sensors Management Platform for Hyper-Connected SmartCities: An IoT Data Marketplace Use Case

Georgios Palaiokrassas [1], Petros Skoufis [1,*], Orfefs Voutyras [1], Takafumi Kawasaki [2], Mathieu Gallissot [3], Radhouene Azzabi [4], Akira Tsuge [2], Antonios Litke [1], Tadashi Okoshi [2], Jin Nakazawa [2] and Theodora Varvarigou [1]

[1] School of Electrical and Computer Engineering, National Technical University of Athens, 15773 Athens, Greece; geopal@mail.ntua.gr (G.P.); o.voutyras@gmail.com (O.V.); litke@mail.ntua.gr (A.L.); dora@telecom.ece.ntua.gr (T.V.)
[2] Faculty of Environment and Information Studies, Keio University, Endo, Fujisawa 252-0882, Japan; drgnaman@sfc.keio.ac.jp (T.K.); tsuge@sfc.keio.ac.jp (A.T.); slash@sfc.keio.ac.jp (T.O.); jin@sfc.keio.ac.jp (J.N.)
[3] Electronics and Information Technology Laboratory, Université Grenoble Alpes, CEA, LETI, DSYS, F-38000 Grenoble, France; mathieu.gallissot@cea.fr
[4] CEA, CEA Tech en Occitanie, 51 Rue de l'Innovation, F-31670 Labège, France; radhouene.azzabi@cea.fr
* Correspondence: skoufis.pet@gmail.com

check for updates

Citation: Palaiokrassas, G.; Skoufis, P.; Voutyras, O.; Kawasaki, T.; Gallissot, M.; Azzabi, R.; Tsuge, A.; Litke, A.; Okoshi, T.; Nakazawa, J.; et al. Combining Blockchains, Smart Contracts, and Complex Sensors Management Platform for Hyper-Connected SmartCities: An IoT Data Marketplace Use Case. *Computers* **2021**, *10*, 133. https://doi.org/10.3390/computers10100133

Academic Editor: Paolo Bellavista

Received: 7 September 2021
Accepted: 12 October 2021
Published: 16 October 2021

Abstract: In this paper, we demonstrate the multiple points of innovation when combining blockchain technology with Internet of Things (IoT) and security frameworks. The deployment and use of IoT device networks in smart city environments has produced an enormous amount of data. The fact that those data are possessed by multiple sources that use independent systems for data collection, storage, and use impedes the exploitation of their value. Blockchains, as distributed ledgers, can be used for addressing the development of a universal system for data collection and distribution. Smart contracts can be used to automate all the processes of such a network, while at the same time, blockchain and the InterPlanetary File System (IPFS) protect sensitive data through anonymity and distributed storage. An innovative and open IoT blockchain market of applications, data, and services is proposed that: (i) provides the framework upon which objects and people can exchange value in form of virtual currencies, for assets (data and services) received; (ii) defines the motivation incentives according to social and business context for humans and smart objects to interact. The specific marketplace is piloted through a cross-border trial between Santander and Fujisawa, in the context of the M-Sec project, validating thus the interoperability, efficiency, and data protection principles.

Keywords: blockchain; Ethereum; smart contracts; internet of things; sensing as a service; system integration

1. Introduction

The term Internet of Things (IoT) refers to wireless sensor networks (WSN) that are deployed seamlessly in several environments and share information to develop a common operating picture (COP) [1]. The impact of the exponential increase in the use of such sensors can be identified in the so-called smart cities. Smart cities are large, complex, distributed and continuous systems that capture and use such mission critical data to improve their processes and their habitats lives [2,3]. However, the security requirements of such systems, and especially the need for lack of a single point of failure, propose the use of a decentralized-distributed schema, for IoT data governance and distribution.

Blockchains are distributed ledgers that store data in a chain architecture, which can store multiple transactions that contain the data in blocks [4]. Certain cryptographic protocols are used for block validation, while the addition of new blocks to the ledger is agreed through a predetermined consensus mechanism of its nodes, providing immutability and consistency.

All the transactions within an IoT network can be integrated on blockchains through the use of smart contracts. Smart contracts utilize protocols and user interfaces to facilitate all steps of the contracting process [5]. The specific nature of the IoT networks' data transmission and action processes can be easily integrated within the business logic structure of smart contracts.

The value of the distributely gathered IoT data can be leveraged through an organized data exchange schema, between data providers and consumers. The Sensing-as-a-Service business model provides a conceptual architecture that can support such a data-exchange schema [6].

Compared to our prior preliminary work [7], this paper presents an extended and more complete version of the system with major enhancements based on the Sensing-as-a-Service (S2aaS) concept that integrates IoT and blockchain to allow data exchanges. By taking into account the latest advancements in the field, as well as the research that has been undertaken to successfully address critical issues and limitations of blockchain and IoT implementations and by integrating state of the art solutions for data gathering and identity management, we propose a complete IoT and blockchain integration that combines most of the advancements in the field to effectively handle security and performance issues and limitations.

The rest of the paper is structured as follows. In Chapter 2, we present the results of the deployed system. In Chapter 3, we present the related work that has been done in the field, which formed the basis of our proposed solution, as well as the M-Sec Project and the smart city context that we chose as a real use case scenario to highlight our solution's functionality and efficacy. Chapter 4 contains the overview of the proposed systems, as well as its requirements and implementation and integration details. Finally, in Chapter 5 we conclude based on our work.

2. Results

Our proposed system was successfully used by an increasing number of users mainly in the context of use cases and field trials of M-Sec Project. The Blockchain Marketplace, integrating very efficient and scientific sound solutions for sensor management, data handling and identity management as described in previous sections, was utilized as a central point for data collection during field trials in use cases. The data and user created content were available for trading in both Japan and Europe while ensuring security on all layers through blockchain and other mechanisms. Different stakeholders participated such as citizens sharing photos, plus using the smart contracts coupons system, and members from universities and research institutions, cities, and companies. During dedicated events, users were able to register and use the functionalities of the platform, exchange information, and provide and purchase sensor data and user-created content as well.

We should note that a lot of effort was given to operate the Blockchain Marketplace in a secure environment while considering the requirements imposed by GDPR (https://gdpr.eu/, accessed on 2 September 2021) and APPI (https://www.ppc.go.jp/en/, accessed on 2 September 2021). Indicatively, we have conducted a lot of research on coupling encrypted databases with blockchain technologies, thus making the synergy of off-chain and on-chain storage and processing of data possible, a characteristic which enhances security considerably, while still ensuring data reliability and users' privacy.

The system was evaluated in five different cross-border pilots in different cities in EU and Japan (https://www.msecproject.eu/use-cases/, accessed on 2 September 2021). Users were able to interact with the proposed distributed system, by registering and accessing the different user interfaces. End-users, having the option of being both buyers and sellers, could upload user-generated content (photos, videos etc.), purchase content, sensor data, datasets and access to past, current and future sensor data as well by using dedicated created blockchain-based Tokens. These Tokens were also used as motivation mechanism and as a means to participate in contests. As a result, more than 10,000 units of virtual Tokens were exchanged for virtual goods trades, regarding thousands of actual resources and data, by verified users of the Blockchain Marketplace.

3. Discussion

3.1. IoT Platforms and Blockchain Solutions

The model of Sensing-as-a-Service (S2aaS) consists of four conceptual architectural layers [6]. Sensor and sensor-owners layer is the deepest layer, which consists of sensor devices that capture data and sensor-owners, who have ownership of the data captured and decide whether to publish the data publicly or not. On top of that lies the sensor publishers (SPs) layer, with main responsibility to detect available sensors, communicate with the sensor-owners, and obtain permission to publish the sensors in the cloud. The next layer is the extended service providers (ESPs) layer, which is an intelligent layer that communicates with multiple SPs and provides added value services, by serving high-level user requests with data provided by multiple SPs. The top layer is the sensor data consumers layer, consisting of all registered sensor data consumers with a valid certificate, who acquire data in return of a proposed fee or offer. Such a model can have a significant impact on computational distribution, increased data collection, and data democratization. However, certain challenges underlie.

The S2aaS concept can be integrated with blockchain by extending the above-mentioned architecture with an extra layer of blockchain on top [8]. Blockchain uses smart contracts to handle user registration and guarantee anonymity through multiple pseudonyms for each of the users. At the same time, the extra layer handles transactions, managing the task of data purchase between data providers and consumers.

Apart from the S2aaS model, the general case of blockchain and IoT system integration shares a common architecture with that of the S2aaS. IoT is mostly modeled through a three-layer architecture, with a physical layer, a network layer, and an application layer. Each of these layers is vulnerable to certain types of attacks that challenge the security of IoT systems [9,10]. Unauthorized access to the network resources, as well as trust management and authentication issues are regarded as some of the most important threats for the network and application layer.

At the same time, there are certain limitations. As is described in [11], the fact that several nodes of a blockchain network store a full copy of the chain data leads to scalability and storage issues, since these nodes need to fulfill significant storage requirements. An oversized chain can cause negative effects on performance of the whole network processes. The latest work on the field takes into account these threats and limitations and attempts to produce solutions that tackle them.

A solution for blockchain and IoT systems that attempts to solve some common limitations of this combination and achieve scalability, high throughput, transparency and lightweight communication between the blockchain and the IoT devices is proposed in [12]. A permissioned blockchain is deployed, where nodes in the network are fully mutually trusted, and apply lightweight and fast consensus algorithms like byzantine fault tolerance (BFT) to achieve high throughput. Additionally, they use RESTful interfaces for the communication between IoT and blockchain, offloading the blockchain network from the IoT devices. The transparency and security of transactions is handled through a strict authorization schema. Additionally, the proposed architecture is modular, allowing for changes to a certain layer without affecting the rest of the system.

In another work, the blockchain layer bridges the physical layer of sensor data capturing devices, with the top application layer that provides users with interfaces to access the system's services [13]. The blockchain layer combines multiple lightweight nodes that perform cryptographic functions to encrypt the data and provide access to the second part of the layer, which is the private blockchain. This allows authorized-only users to access the blockchain and perform transactions through certain smart contracts. Despite the robustness of this architecture, the fact that it uses an on-chain data storage schema may lead to scalability issues.

Besides scalability, Yu et al. stress the issues of data supervision and management, trust among participating entities, and device lifecycle management in IoT systems [14]. In the area of smartwatch IoT devices, they propose a solution based on a permissioned Hyperledger Fabric blockchain, which uses smart-contracts to handle data ownership and trading between all stakeholders of this IoT ecosystem, smartwatch owners, manufacturers, and data analytics companies. Thus, the proposed solution addresses the issues of trust and data management and they agree that future research should explore issues of data-privacy and the tradeoff between public verifiability and privacy, as well as effective model delegation.

A great amount of research has focused on the privacy issues that arise, regarding the storage of sensitive personal user data on the blockchain, due to their irreversibility and transparency [15]. They propose the use of the decentralized peer-to-peer repository, InterPlanary File System (IPFS), as the off-chain storage of the personal data, and they handle the access to this data through a private smart contract, only for administrator users.

A similar use of the IPFS has been proposed as an off-chain storage for sensitive medical data [16]. IPFS is considered a great fit for this purpose due to the fact that it features high throughput, security with hash mapping of transactions, and concurrent access of transactions by peers in the network. In both [15,16], only the content-addressed hash of the data needs to be stored on the blockchain, off-loading it from a great amount of information that can decrease its performance, since blockchains are inefficient for saving large-size data [17].

3.2. IoT Sensor Data Handling and Blockchain

Many IoT systems have already been introduced with various sensing data in different use cases all over the world, but there are still many vertically integrated systems which cannot share sensing data horizontally. We have not reached the point yet, where people who need to utilize data from sensors all over the world can efficiently retrieve and use them. The Publish/Subscribe messaging model (Pub/Sub model) is effective as an IoT system that handles various kinds of sensing data. It is a model that matches supply and demand, which is common in such Internet businesses.

This messaging model provides the functionality of sending data to many clients at the same time. In addition, data senders (publishers) and receivers (subscribers) do not depend on each other in the context of this model. This is an important advantage, when many clients connect, aiming to exchange data, because they do not affect each other status. For instance, even if a publisher application is down, a subscriber is able to continue performing his regular operations. The reason behind this is that clients do not connect directly with each other, but via a broker server.

Currently, several Pub/Sub protocols are developed, such as XMPP [18], MQTT [19] and AMQP [20], and more. These protocols are implemented based on the topic-based Pub/Sub model. The notion of a Topic is used and in the context of this paper, we use the Topic's name to specify the sensor's name. In our topic-based Pub/Sub model, clients specify the topic name and by the defined convention within a topic's name, a "/" is used in the expression. For instance, for an environment sensor in a house, the topic name is expressed as "house/environment/temperature". This way, the topic name is quite descriptive and provides sufficient information for the supported data.

However, if a subscriber is not aware of the different publishers and the sensor/topic names, he cannot find the topic name in which he wishes to subscribe to. To this direction, Yonezawa et al. developed SOXFire as a Pub/Sub model platform for sensors effectively handling the previously mentioned limitation [21,22]. SOXFire is implemented based on Sensor-Over-XMPP [23], and extends Openfire [24]. This platform can support and manage complex sensors by extracting many different topics from a single compound topic.

To accomplish this, a topic on the platform is configured to have a "meta node" and "data node". The most important features lie within the meta node, which has the information and metadata related to the topic. It additionally includes the sensor category and the unique name of the topic.

A complex sensor includes various values such as geolocation and information about some other sensors. At the platform, a sensor is managed as an independent transducer, so the complex sensor has many transducers. Subscribers can get the meta-information from SOXFire and easily confirm the information about the topic.

In addition, SOXFire uses XMPP [21]. The overhead of this protocol is larger generally than AMAP and MQTT, but XMPP can more effectively handle various data types. The reason XMPP effectively treats textual data having high reliability is related to the implemented protocol for instant messages, while also being efficient for handling bigger objects such as pictures. In the context of this research work, we treat several information and topics as well, so using SOXFire has many advantages and allows us to manage those data effectively.

However, it is difficult to implement an advanced data commerce mechanism based only on SOXFire. The main reason lies in the fact that while Pub/Sub messaging model is good at handling data for many clients at the same time, this model does not guarantee enough security regarding the integrity of the data. In addition, in a Pub/Sub model-based platform, a sender cannot directly connect to a receiver, so while the receiver purchased the data of the sender, the sender cannot receive notifications about it. To this direction, in this research work, we are implementing a data-secure marketplace by collaborating SOXFire and another information technology like blockchain, as will be described in detail in the sections that follow.

3.3. Identity Management and Blockchains

One of the key requirements for the development of a successful platform that uses blockchain is the secure identity management. The three aspects of identity management are authentication, privacy, and trust [25].

Ren et al. [26] propose blockchain-based identity management with the addition of an access control mechanism as a solution to the authentication problem in edge computing IoT environments. The registration and authentication processes are secured through the use of self-certified cryptography, where certificates and network entities are implicitly connected to create an identity and certificate management mechanism based on blockchain. The identity management system is also enhanced with the use of a Bloom filter.

Zhao and Liu [27] suggest the development of a decentralized identity management system based on blockchain and smart contracts, where users have full control over their personal data access rights. The anonymity of the users is preserved through the use of an attribute-based authentication schema, where each user's identity is decomposed into a set of attributes, which can be validated independently. The credibility of each user's identity is being assessed by an attribute reputation model, which performs adequately even in decentralized environments.

There are also privacy compliant authentication protocols that have been proposed, using in most cases Zero-Knowledge proof attesting the identity of one without revealing it. Such examples include for example ZCash, Hawk, and SNARK [28–30].

Despite the current scientific achievements in the field of identity management, the issues of "identity wallet leakage" and identity changes constitute serious risks for the success of blockchain solutions that handle identity management [25].

3.4. Hyper-Connected Smart Cities: The Example of M-Sec

A great deal of state-of-the-art IoT systems in smart cities tends to be built around the concept of the integration of heterogeneous data streams within one or more infrastructures. This wave of IoT systems enables IoT systems to benefit from the scalability, performance, and capacity of the cloud, which has been proven very efficient for certain classes of IoT applications such as large-scale data processing problems. Nevertheless, these architectures promote a centralized data collection and processing approach, which introduces several limitations both in terms of the supported applications and in terms of the business models that they enable. In particular, smart city platforms are mainly centralized IoT/Cloud infrastructures

Today, we also have new approaches in P2P systems, cryptocurrencies (like Bitcoin, Ether and many others), blockchain ledgers that can provide a common reference for distributed, and decentralized systems for collaboration increasing the levels of trust in trustless environments. Blockchains at the same time can provide a tamper-proof framework for data to be exchanged between smart city platforms, while forming the underlying technology for building the internet of value that can make a marketplace of sensors in smart city context a reality.

An example of such an approach for smart cities is the M-Sec (https://www.msecproject.eu/, accessed on 2 September 2021) project. M-Sec in particular makes use of today's key technologic enablers by applying the blockchain concept to a highly decentralized IoT. A smart object, part of smart city IoT infrastructure, can be registered to a blockchain and the object remains a unique entity within the blockchain throughout its life. This gets even more apparent with plans for creating a disruptive way IoT will work in the next years, proposing new the business models and usage patterns of IoT infrastructures, providing the way where economic value can be generated from the devices for both the devices and humans.

By architecting approaches to develop different delivery and communication patterns such as P2P, publish/subscribe, message queuing for heterogeneous participants, etc. different objectives are in the focus such as:

(i) design the future decentralized architecture of IoT that will unlock the capacity of smart objects, by allowing to instantly search, use, interact, and pay for available assets and services in the IoT infrastructures.

(ii) enable seamless, highly autonomous, and secure interaction between humans and devices in the context of a smart city, through the use of blockchains and for business contexts relevant to specific smart city use cases enabling innovative machine–human and machine–machine interactions.

(iii) engineer new levels of security and trust in large scale autonomous and trustless multipurpose smart city platforms, define and implement trust-ensuring mechanisms that will enable virtual currency transactions with transparency and all necessary security aspects (authentication, authorization, accounting, etc.).

(iv) define, design, and implement a novel marketplace where smart objects can exchange, information, energy, and services through the use of virtual currencies, allowing real-time matching of supply and demand, enabling the creation of liquid markets with profitable business models of the IoT stakeholders.

In order to enable the M-Sec paradigm, the project defines and implements a platform (comprising middleware, protocols, and tools) for the implementation of applications involving peer-to-peer decentralized interactions between objects and people in a hyper-connected smart city context. The M-Sec project brings together two smart cities (Santander, Fujisawa) and five use cases, such as securing IoT devices to enrich strolls across smart city parks in Santander, cross-border handling of heterogeneous user data (e.g., photos), preventing attacks, and allowing trading in a secure environment during festivals and cultural events with the participation of local stakeholders, taking advantage of the M-Sec Token as a reward mechanism, as will be described in the following sections.

4. Materials and Methods

4.1. System Overview and Requirements

This paper introduces a distributed application that blends several technologies, including smart contracts deployed on the blockchain, web services, and databases to produce an Ethereum blockchain Dapp, enabling users to easily buy and sell IoT sensor data, by using a custom token as the payment currency. The proposed system acts as a widely used decentralized repository where sensor owners can gain value from their data. An indicative use case includes a potential sensor data buyer who compensates the data provided by paying with cryptocurrency. To accomplish this, our deployed contracts communicate with each other, verify that the buyer has sufficient funds, and perform the payment following the billing schema. M-Sec Tokens (our specifically created cryptocurrency) are transferred from the buyer's account to the owner's account and they both receive the transaction hash, as a proof of this deal. An overview of our proposed system is shown in the following Figure 1 and consists of four main components: (i) Blockchain Marketplace, (ii) SOXFire, (iii) Security Manager, (iv) Off-chain Data Handler, which are presented in the following subsections.

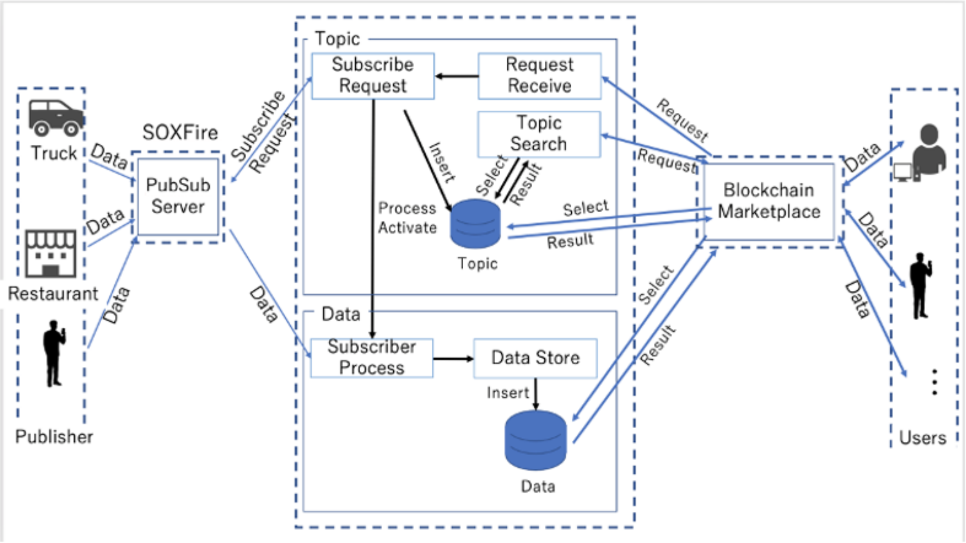


Figure 1. System Architecture of a Bridge System.

4.1.1. Blockchain Marketplace

The idea of data marketplace is to construct a marketplace where data integrity is present and tamperproof data can be securely distributed with secure multi-layer technologies. Marketplace's users can exchange IoT sensor data and media easily, securely and with complete anonymity. It allows the real-time matching of supply and demand, enabling the creation of liquid markets with profitable business models of the IoT stakeholders. IoT devices and humans using mobile applications and APIs are able to exchange data and value through the blockchain-based implementations, as shown in Figure 2.

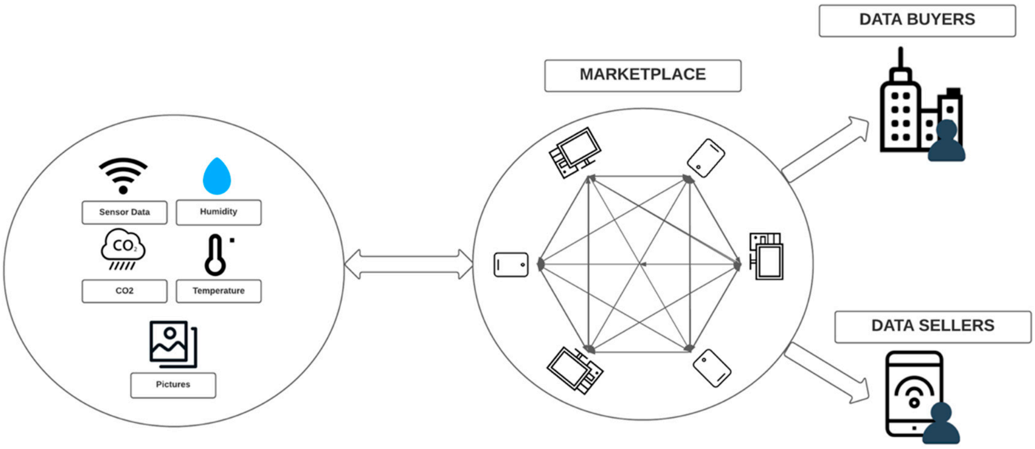


Figure 2. Overview of the Blockchain Marketplace allowing the monetization of data and the exchange of information and services.

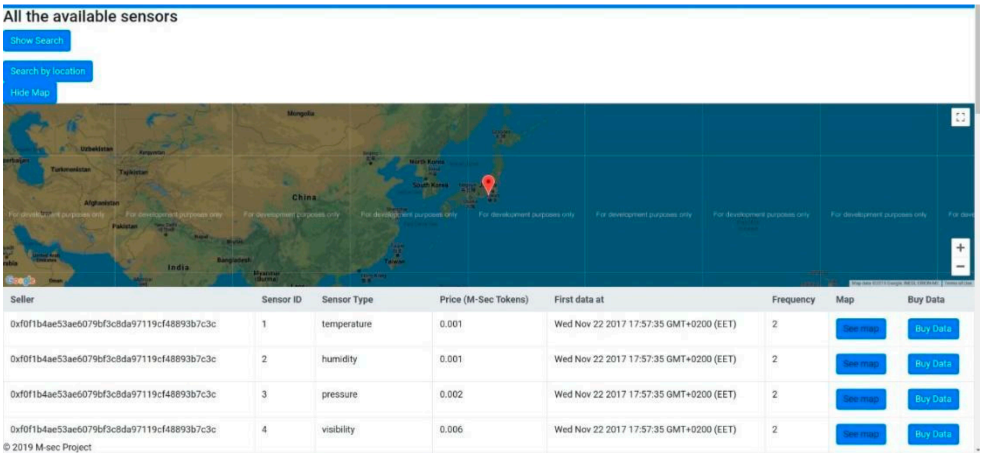
The implementation details of this novel marketplace, where smart objects can be exchanged through the use of virtual currencies is publicly available (https://github.com/MSec-H2020/IoT_Marketplace, accessed on 2 September 2021) and is based on three main parts:

1. Smart Contacts: implemented in Solidity (https://docs.soliditylang.org, accessed on 2 September 2021) programming language, a contract-oriented, high-level language whose syntax is similar to Javascript and supports inheritance, libraries, complex-defined types, and is designed to target the Ethereum Virtual Machine.
2. Web Application: The interactions among end-users, the Blockchain Marketplace and deployed smart contracts are handled by the developed web service. It is accessible through its web browser and API endpoints and the data exchanged can be distributed on off-chain and on-chain databases. We used Nodejs and Node-Red (https://nodered.org/, accessed on 2 September 2021) (a flow-based development tool) for the Back End. A relational database management system PostgreSQL (https://www.postgresql.org/, accessed on 2 September 2021) was used for optimizing system's performance and caching frequent requests.
3. User Interfaces: Implemented using HTML, Javascript, jQuery, Bootstrap to develop the different Front-End interfaces, Web3 Javascript API to handle the interaction with smart contracts. The Interfaces between the users and the blockchain provide functionalities helping users interact with the smart contracts deployed on blockchain, browse available media items and datasets from smart cities, sensors and other users, and access data they have bought. Owners are able to register their sensors to the marketplace, while a buyer browses sensors, specifies desired time period, and buys data using our dedicated created M-Sec cryptocurrency. Through the Marketplace sending transactions and reading data of transactions and smart contracts is also allowed, while users are "protected" from misreading or mistyping info when sending a transaction. The user searches in all the available sensors registered in the Smart Contracts the sensors of interest specifying details in the corresponding fields such as the location, the type the data (temperature, starting date and time, frequency etc.). Some example GUIs are shown in the following Figures 3 and 4.

4.1.2. SOXFire and Data Collection

In this research, SOXFire collects some information. We collect various data from Fujisawa city in Japan. First, we collect environmental data in the city. At this fieldwork, we focus on garbage trucks because they drive around the whole the city almost every day [31]. We attached a sensor box to the garbage truck, which has some sensors measuring temperature, PM2, illuminance, and more. In addition, we installed other sensors in some restaurants in the city. Currently, checking CO_2 concentration in the restaurant is important because there is a concern by the public about this because of the increasing population density. Additionally, because of measures related to the virus, COVID-19, people keep distance among each other to prevent transmission and infection. Thus, the data from sensor boxes are sent to SOXFire and many clients are able to have access to these data by using SOXFire.

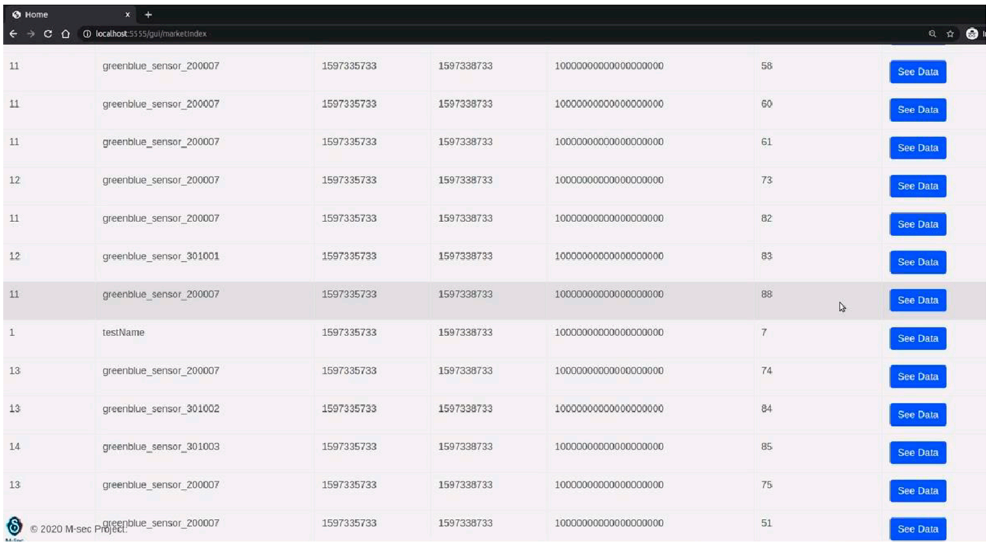
Except from sensor data, pictures are also captured by citizens in the city. We implemented a client application to collect information related to the city [32]. This application can take a photo captured by smartphone and post it to Social Networks. Since, many times personal information is also included in the data sent to SOXFire, other clients cannot have access to part of these data. A collaboration among SOXFire and the blockchain marketplace is presented in the following sections.



Seller	Sensor ID	Sensor Type	Price (M-Sec Tokens)	First data at	Frequency	Map	Buy Data
0xf0f1b4ae53ae6079bf3c8da97119cf48893b7c3c	1	temperature	0.001	Wed Nov 22 2017 17:57:35 GMT+0200 (EET)	2	See map	Buy Data
0xf0f1b4ae53ae6079bf3c8da97119cf48893b7c3c	2	humidity	0.001	Wed Nov 22 2017 17:57:35 GMT+0200 (EET)	2	See map	Buy Data
0xf0f1b4ae53ae6079bf3c8da97119cf48893b7c3c	3	pressure	0.002	Wed Nov 22 2017 17:57:35 GMT+0200 (EET)	2	See map	Buy Data
0xf0f1b4ae53ae6079bf3c8da97119cf48893b7c3c	4	visibility	0.006	Wed Nov 22 2017 17:57:35 GMT+0200 (EET)	2	See map	Buy Data

© 2019 M-sec Project

Figure 3. Graphical User Interface enabling searching of sensors in the smart contracts running on blockchain.



11	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	58	See Data
11	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	60	See Data
11	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	61	See Data
12	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	73	See Data
11	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	82	See Data
12	greenblue_sensor_301001	1597335733	1597338733	10000000000000000000	83	See Data
11	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	88	See Data
1	testName	1597335733	1597338733	10000000000000000000	7	See Data
13	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	74	See Data
13	greenblue_sensor_301002	1597335733	1597338733	10000000000000000000	84	See Data
14	greenblue_sensor_301003	1597335733	1597338733	10000000000000000000	85	See Data
13	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	75	See Data
	greenblue_sensor_200007	1597335733	1597338733	10000000000000000000	51	See Data

© 2020 M-sec Project.

Figure 4. User browsing all purchased data from sensors.

4.1.3. Security Manager

We have specified and implemented a tool to provide interoperable security functions such as a Public Key Infrastructure (PKI), a User Federation system, and some administration component proposed by the requirements for a Trusted Computing Group related to trusted platform modules and trusted networking. We called this tool the "Security Manager" and its overview is shown in Figure 5. Its goal is to enable mutual trust between all components in an IoT infrastructure.

At a device level, devices are enrolled using their TPM for identification. Once enrolled, device can be provisioned with the infrastructure keys and certificates. A "Remote Attestation" procedure enables manufacturer and services to verify the integrity of the device in a secure manner. Additionally, the device is provisioned with ECDAA keys and certificate in order to enable anonymous attestation that may be required depending of use cases and their privacy requirements.

At the infrastructure level, the PKI leverages trust by providing multiple interfaces and code snippets to use the same security backend for multiple components thanks to LDAP, SASL, and Kerneros. Finally, at the End-User level, user federation brings authentication and accounting facilities using OpenID and Oauth2.

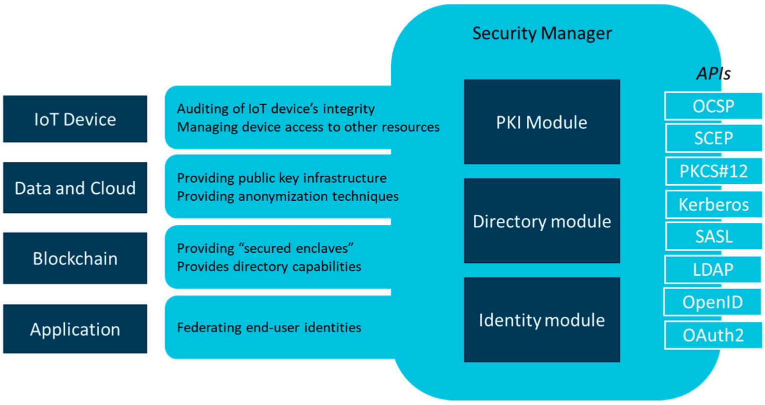

Security Manager
APIs
IoT Device
Data and Cloud
Blockchain
Application
Auditing of IoT device's integrity
Managing device access to other resources
Providing public key infrastructure
Providing anonymization techniques
Providing "secured enclaves"
Provides directory capabilities
Federating end-user identities
PKI Module
Directory module
Identity module
OCSP
SCEP
PKCS#12
Kerberos
SASL
LDAP
OpenID
OAuth2

Figure 5. Overview of Security Manager.

4.1.4. Offchain Data Storage Interacting with IPFS

The integration of IPFS as a distributed off-chain data storage for blockchain solutions has already been applied in several works. In the field of healthcare, Kumar et al. [16] propose the use of the IPFS as a means to preserve the security of sensitive data, by avoiding the use of a single centralized database that stores the entire information. Ali et al. [33] combined blockchain with the IPFS to develop a decentralized effortlessly privacy-preserving IoT-data storage solution. Additionally, Zou et al. [34] propose a blockchain-based incentive anonymous reporting system that uses IPFS as an off-chain storage solution to achieve security, high-throughput, high-capacity storage, and high concurrent access, while at the same time preserving user anonymity.

Aiming to a more decentralized design, we integrated blockchain with IPFS (Inter-Planetary File System), a peer-to-peer version-controlled protocol and filesystem, run by multiple nodes, storing files submitted to it [35]. It combines distributed Hash Tables, Block Exchanges, and Merkle Trees. Using middleware, users are able to upload content to IPFS and place its unique hash code (address of the file) to the smart contracts running on Quorum blockchain. If we use a central database for storage, we benefit from the high throughput but this centralization does not coincide with the decentralized nature which blockchain advocates can lead to a Single Point of Failure (SPOF) of the whole application. Facing the aforementioned drawback, IPFS being a peer-to-peer (p2p) file sharing system and blockchain's complementary component, settled exceptionally the SPOF problem, furnishing low latency and data distribution.

4.2. Implementation and Integration Details

4.2.1. Blockchain Framework

After deployment to the blockchain, smart contracts' code is final and cannot be modified. Thus, it is of vital important to trace any bugs during the design and implementation process and through many iterations and testing end up to final smart contracts which will be deployed to the blockchain and cannot be altered. The different smart contracts are written in the programming language Solidity (http://solidity.readthedocs.io, accessed on 2 September 2021), so it is feasible to migrate from Quorum permissioned blockchain framework to public blockchain frameworks (e.g., Public Ethereum Network), since Solidity is the common programming language to Ethereum-based blockchain frameworks. We employed and experimented with different blockchain technologies and implementations featuring main differences, for example in access rights and permissions, before concluding to the use of the Alastria blockchain platform.

1. Private Ethereum blockchain: During the development process, we used a local private blockchain named Ganache (https://www.trufflesuite.com/ganache, accessed on 2 September 2021), which allowed us extensive testing of the developed smart contracts. It provides a personal Ethereum blockchain which we can use to run tests,

execute commands, and inspect the state while controlling how the chain operates. It provides a built-in explorer and allows us to quickly see the current status of all accounts, including their addresses, private keys, transactions, and balances.

2. Public Ethereum blockchain: We deployed smart contracts on Ethereum-based blockchains using browser IDE "Remix" (https://remix.ethereum.org/, accessed on 2 September 2021). Remix is an open-source tool that supports smart contracts development on the browser and facilitates the deployment on local or public Ethereum-based blockchain platforms. We explored the use of public test Ethereum blockchain.
3. Quorum blockchain framework: The different smart contracts were deployed on Quorum blockchain framework (https://docs.goquorum.com/en/latest/, accessed on 2 September 2021). Quorum is a permissioned implementation of Ethereum, which allows certified members to build and run decentralized applications that run on blockchain technology. It is an open-source platform and supports smart contract privacy. Both private and public smart contracts are validated by every node within the blockchain network. Additionally, Quorum provides privacy and transparency, both at the transaction-level and network wide. In each Quorum node, consensus is achieved with the Raft or Istanbul BFT consensus algorithms instead of using Proof-of-Work. The P2P layer has been modified to only allow connections to/from permissioned nodes. In Ethereum, the notion of Gas was introduced (the fee or pricing value required to successfully conduct a transaction or execute a contract on Ethereum blockchain platform), while in Quorum the pricing of Gas has been removed, although Gas itself remains. One of the features of Quorum that are of great value for the component is the network and peer-to-peer permission management. This feature enables only the validated and authorized users to have access and be a part of the network. Also, Quorum provides enhanced transaction and smart contract privacy features. The permission-based nature of Quorum enables the constitution of private and public transaction getting the best of both worlds; open transactions are analogous to Ethereum but when it comes to the private transaction then it is confidential, and the data is not exposed to the public. Quorum adds privacy functions that allow for private transactions that are only visible to the transacting parties, while the other parties in the network would only see a hash. Finally, Quorum is considered to be very fast due to its efficient consensus mechanism, which belongs to the family of Byzantine Fault Tolerance (BFT) mechanisms [36]. In order to develop and deploy the smart contracts to Quorum blockchain, we used Truffle suite (https://www.trufflesuite.com/, accessed on 2 September 2021). It is a development environment and testing framework using the Ethereum Virtual Machine (EVM). Before we deploy the smart contracts to the blockchain network, we extensively tested them on the private blockchain previously presented. Additionally, before being deployed on a larger Quorum Network, we used a Quorum test network consisting of seven nodes (https://github.com/jpmorganchase/quorum-examples/tree/master/examples/7nodes, accessed on 2 September 2021).
4. ALASTRIA: It is a public-permissioned network, sharing some of the properties of both types of networks public-permissionless and a private consortium, while trying to overcome scalability problems of traditional blockchain platforms related to their consensus mechanisms. There are a lot of efforts being made to solve or alleviate blockchain's scalability problem, but as of today, the problem still exists and permissioned networks demonstrate significant better performance than public-permissionless. On the other hand, the high transaction cost in public blockchains such as Bitcoin and blockchain and the fact that all nodes store every transaction and smart contract constitute a restriction for certain types of applications. Alastria (https://github.com/alastria/alastria-platform/blob/master/en/Alastria-Core-Technical-Platform.md#alastria-core-technical-platform, accessed on 2 September 2021) is a Public-Permissioned network compatible with regulation, while implementations are based on Quorum. Each participant node has to be identified before it can

participate in the network, while there is no cryptocurrency embedded as in the case of Bitcoin or Ethereum.

4.2.2. Smarts Contracts Details

Ethereum (https://www.ethereum.org, accessed on 2 September 2021) provides a Turing-complete scripting language onto an authenticated data structured inspired by the Bitcoin blockchain. This language can be used to specify contracts to be enforced in the distributed network without a central arbiter (so-called "smart contracts"). Smart Contracts are instances of a computer program that runs on blockchain. In the case of permissioned blockchain such as Quorum, where only authorized users are able to interact with the ledger, an authorized user can create a contract by posting a transaction to the blockchain. The code's execution is provoked by a received message either from a user of another contract and could provide utility to other contracts or require assistance from other Smart Contracts.

In our proposed system the power of smart contracts is combined with the sensor capabilities of the IoT, aiming to leverage the flexibility and trust of self-enforcing, distributed property transfers from purely virtual goods to all IoT-enabled objects. This securitization of IoT devices will enable these devices to act as oracles: agents, which can gather and measure real-world data and trusted enough to submit this information to a blockchain for being used by smart contracts. In this section, we describe the different smart contracts developed to support the proposed system and related-use cases as well as some of the functionalities they provide.

M-Sec Token

A custom token was created specifically for research purposes. It is actually a cryptocurrency in the form of a smart contract running on Quorum blockchain. It follows the ERC223 (https://github.com/ethereum/EIPs/issues/223, accessed on 2 September 2021) token standard. Preliminary implementations followed the ERC20 token standard, but ERC223 is a superset of the previous standard offering security improvements and more usability and backwards compatibility with any services and functionalities designed and developed for ERC20. As fully compliant with ERC223, it implements a set of functions and events, such as name(), transfer(), totalSupply() and Transfer event, which is emitted to the blockchain when an amount of Tokens is transferred from a user to another.

This Token has different applications for the end-users of the Blockchain Marketplace. It is firstly used as a payment currency to exchange value among the users of the Marketplace. Another implementation and configuration of the M-Sec Token allows us to use it as a "Social Token". This Token acts as a mean to tokenize a loyalty points program with rewards. Users of the platform have an initial balance and particular users are rewarded with more tokens based on specific criteria such as for example:

(i) the most active user,
(ii) the most social user,
(iii) the user who uploaded the most popular content.

Item Manager Smart Contract

The Item Manager Smart Contract allows the interaction of item/content creators (e.g., photos, multimedia items etc.) with the platform and the blockchain. A user is able to upload all the information and metadata related to an item. To this direction, we created dedicated "structs", as shown in the following code snippet, which are a special feature of solidity contract-oriented programming language, in order to store for each item, the details (e.g., tags, information, metadata) and the unique address of its owner. It is important to note that the actual data are stored in an off-chain mode, exploiting the integration with IPFS.

```
struct item {
   address owner;
```

```
    string URI;
    uint256 price;
    string tag;
    string info;
}
```

Sensors Smart Contract

This smart contract records all the registered IoT sensors. It gives the possibility to register a sensor and to change its information afterwards as well. Dedicated Solidity structures were created to store this information and functions to allow its retrieval. A structure that allows the storing of the information is the following:

```
struct sensor {
    address sensor-Owner ;
    uint8 type-of-Sensor ;
    uint MSec-Token-Price ;
    uint32 timestamp-of-start ;
    uint16 frequency ;
    int32 latitude ;
    int32 longitude ;
    string url ;
    string name
}
```

It is important to note that functions altering sensors information such as the seller, the price, the url succeed only when the owner of the sensor (specific address) attempts to change the fields; otherwise, the access is denied.

The function for purchasing sensor data directly communicates with M-Sec Token smart contract, when a user wishes to buy data for a specific period. If the user has sufficient funds and the information is correct, then the transaction will be successfully completed. Upon success, the event Transfer is emitted to the network informing the users who watch the smart contracts that this transaction took place. Regarding the sensor type, the possible values are described in Table 1 that follows.

Table 1. Different types of sensors and units of measurement supported by Sensors Smart Contract.

Serial Number	**Type of Sensor**	**Proposed Unit of Measurement**
1	Temperature	°C
2	Relative humidity	%
3	Pressure Hectopascal	hPa
4	Visibility	Km
5	Wind speed and direction	m/s
6	Sky cloud coverage	%
7	Dew point	°C
8	Solar Radiation	watt/m^2
9	UV index	0 to 11
10	Columnar density of total atmospheric ozone layer Dobson	DU
11	Smart Plug	Volts
12	Smoke Bool	on/off
13	Mattress Bool	on/off

Smart City Data Smart Contract

This smart contract acts as a template for managing data from smart cities. It was successfully used in the context of M-Sec Project for managing data from the smart cities of Santander and Fujisawa. It directly communicates with other smart contracts and parts of the Blockchain Marketplace, which allow encrypted data storage and off chain storage.

This smart contract constitutes an extension of the Sensor Smart Contract oriented to better handle datasets provided by smart cities. Among others, this smart contract was used to integrate the data and datasets from Santander Open Data Platform (www.datos.santander.es, accessed on 2 September 2021). Different datasets are provided about transport, urban planning and infrastructure, culture and leisure, environment, science and technology, society, well-being, and more. As part of the integration, where the open data API is also utilized (http://datos.santander.es/documentacion-api/, accessed on 2 September 2021), this smart contract stores all the available metadata, making available the datasets to the Blockchain Marketplace.

Value Handler Smart Contract

This smart contract is responsible for handling values in different formats such as hashed values and is used to enhance the security aspects among the different components and APIs' communication with Blockchain Marketplace and facilitate convergence of IoT security with blockchains to support an innovative smart city platform.

4.2.3. Integrating Pub/Sub Model with Blockchain Marketplace

A point of integration between Blockchain Marketplace and SOXFire was implemented, facilitating different end-users of the system. To this direction, a middleware component was developed, namely "SOXFire–Blockchain Marketplace Bridge" allowing registration of sensors, purchase/exchange of data, and their visualization. An overview of this component is shown in Figure 1.

This integration among Blockchain Marketplace and SOXFire, realized by using the bridge system (Figure 1), provides some important functionalities. It allows subscription to different topics available in the Blockchain Marketplace, since the bridge system can store not only the values but also various information regarding each topic. Pub/Sub model platform generally does not provide storing functionality, so SOXFire does not support this as well. The bridge system implements a flow, transmitting data from SOXFire to the Blockchain Marketplace. The data format used is shown in the following Table 2.

Table 2. The data format and fields arriving in the bridge.

Name	Proposed Unit of Measurement
Topic_ID	Unique topic name
Transducer_ID	Each transducer's name
Value	Value
Pub_Timestamp	Timestamp when published from publisher
Timestamp	Timestamp of arrival in the Bridge

In addition, having SOXFire managing and handling all topics could require big computation cost as well as fast and reliable connection. In addition, there is a possibility that the server becomes at some random point temporarily unavailable for technical reasons. In this case, the Blockchain Marketplace would attempt to communicate with SOXFire servers, but would have experienced unexpected latency or in the worst case would be temporarily impossible. To this direction, the bridge system is able to act as a Proxy and manage some of the connections with SOXFire servers. And, all of the data can be fetched to the local storage, using a combination of traditional databases and IPFS. So, in the end, the Blockchain Marketplace will be able to retrieve all the requested data and serve them to the end users.

Additionally, the bridge system has a searching function optimized for SOXFire. If a client has the meta-information of a topic, he is able to subscribe to the topic once. This means the clients would have to execute programming commands to retrieve the desired meta-information from SOXFire. Instead, the bridge system executes these commands, by using dedicated HTTP requests, and returns the parsed result to the client program. Moreover, the bridge system can search and subscribe by prefix matching. When clients want to receive data from the SOXFire server, they subscribe by providing the topic name. This command is the same as in the case of the bridge system. However, if a client wants to subscribe to some topics, the client has to repeat this operation. In this work, to facilitate the subscription, we ended up to a convention. We defined the regulation of registered topic names. The head name of similar sensors includes a common keyword. For example, sensors of restaurants are named "greenblue_XXX" ("greenblue" is a registered company name). By this, topic names have a regularity. A bridge system can subscribe to all topics that have the same head name by using prefix matching. Therefore, clients are able to subscribe to different topics in a more easy and automated way.

4.2.4. Integrating Identity Management with Blockchain Marketplace

The Security Manager tool provides an extra layer of safety in the authentication process of the application, while at the same time ensuring controllable user anonymity within the app. It operates as a standalone service that retains all the sensitive data that each user provides upon registration and connects them with the anonymous identity keys that define each user within the Marketplace. This tool is responsible for the processes of user registration, authentication, and management. The Security Manager can therefore be presented as a lightweight, yet secure, alternative to a Know Your Customer (KYC) authentication service [15], since it acts as a validator of a user's identity and as a bridge between the user's real identity that contains sensitive-data and the public and private keys constitute a user's identity within the Marketplace. Upon authentication, the Security Manager service exposes only the public and the private key of the authenticated user to the Marketplace application, therefore preserving the sensitive-user data and allowing for anonymous, yet validated, browsing of the users within the app and transactions within the blockchain network. At the same time, it is important to stress that the existence of two distinct environments, the one of the Security Manager and the other of the Marketplace, with two distinct types of data storage, traditional databases for the Security Manager and immutable distributed storage solutions for the Marketplace, protect the rights of each user to their personal data, including this of the erasure of data that contradicts with the nature of blockchain.

The Security Manager tool is integrated with the Marketplace by using OAuth 2.0, OpenID and Keycloak. OAuth 2.0 is an authorization framework that lets an authenticated user grant access to third parties via tokens. A token is usually limited to some scopes with a limited lifetime. Therefore, it is a safe alternative to the user's credentials. OpenID Connect (OIDC) is built on top of OAuth 2.0 to add an identity management layer to the protocol. Hence, it allows clients to verify the end user's identity and access basic profile information via a standard OAuth 2.0 flow. Finally, Keycloak is used within the Marketplace app to bridge it with the Security Manager tool since it can federate existing external user databases. Keycloak supports LDAP and Active Directory, as user storage providers. When a user logs in, Keycloak will look into its internal user store to find the user. If it cannot find it there, it will iterate over every user storage provider configured until it finds a match.

5. Conclusions

The proposed system manages to administrate the whole process of IoT data gathering and exchange successfully by leveraging distributed storage technologies like blockchain and the IPFS, as well as automated business-logic scripts called smart-contracts. The modular nature of its architecture allows for excessive security between the several layers of the platform, while at the same time providing freedom and ease in terms of changes in a specific layer. Additionally, all the upper layers of the platform are developed in such a way that guarantees both validity and anonymity for all the participants in the data exchange. In that way, users are allowed to operate freely within the platform and benefit without compromises from the data-value exchange that takes place.

The present work creates a path for future developments in several parts of the proposed platform. In terms of the data collection layer, research could be made to simplify the data submission process to support a larger gamut of sensor types. Additionally, the token exchange functionality of the platform could be enhanced to support a wider range of crypto tokens, as an exchange fee for the purchased data, and advancements could be made in the integration of many crypto-wallets' functionalities within the platform.

Author Contributions: Conceptualization, G.P., O.V., M.G., A.T., A.L., J.N. and T.V.; methodology, G.P., O.V., M.G., A.T., A.L., T.O., J.N.; software, G.P., P.S., T.K., M.G., R.A., T.O.; validation, G.P., O.V., M.G., A.T., A.L., T.O. and J.N.; formal analysis, G.P., O.V., M.G., A.T., A.L. and T.O.; investigation, O.V., M.G., R.A., A.T., A.L., T.O., J.N.; writing—original draft preparation, G.P., P.S., T.K., M.G., A.T., A.L.; writing—review and editing, G.P., P.S., T.K., A.L.; project administration, G.P., O.V., M.G., A.T., A.L., T.O., J.N. and T.V. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results received funding from the European Commission under the H2020 Programme's project M-Sec (grant agreement nr. 814917). Additionally, this work received support from the French State under the Investments for the Future program, IRT Nanoelec, with the reference ANR-10-AIRT-05.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
2. Fernández-Caramés, T.M.; Fraga-Lamas, P. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* **2018**, *6*, 32979–33001. [CrossRef]
3. Chakrabarty, S.; Engels, D.W. A secure IoT architecture for Smart Cities. In Proceedings of the 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2016; pp. 812–813.
4. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
5. Szabo, N. Formalizing and Securing Relationships on Public Networks. *First Monday* **1997**, *2*. [CrossRef]
6. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Sensing as a service model for smart cities supported by Internet of Things. *Trans. Emerg. Telecommun. Technol.* **2014**, *25*, 81–93. [CrossRef]
7. Papadodimas, G.; Palaiokrasas, G.; Litke, A.; Varvarigou, T. Implementation of smart contracts for blockchain based IoT applications. In Proceedings of the 2018 9th International Conference on the Network of the Future (NOF), Poznan, Poland, 19–21 November 2018; pp. 60–67.
8. Lin, C.; He, D.; Zeadally, S.; Huang, X.; Liu, Z. Blockchain-based Data Sharing System for Sensing-as-a-Service in Smart Cities. *ACM Trans. Internet Technol.* **2021**, *21*, 40:1–40:21. [CrossRef]
9. Mohanta, B.K.; Jena, D.; Ramasubbareddy, S.; Daneshmand, M.; Gandomi, A.H. Addressing Security and Privacy Issues of IoT Using Blockchain Technology. *IEEE Internet Things J.* **2021**, *8*, 881–888. [CrossRef]
10. Malik, A.; Gautam, S.; Abidin, S.; Bhushan, B. Blockchain Technology-Future of IoT: Including Structure, Limitations and Various Possible Attacks. In Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 5–6 July 2019; Volume 1, pp. 1100–1104.
11. Reyna, A.; Martin, C.; Chen, J.; Soler, E.; Díaz, M. On blockchain and its integration with IoT. Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *88*, 173–190. [CrossRef]
12. Hang, L.; Kim, D.-H. Design and Implementation of an Integrated IoT Blockchain Platform for Sensing Data Integrity. *Sensors* **2019**, *19*, 2228. [CrossRef] [PubMed]

13. Latif, S.; Idrees, Z.; Ahmad, J.; Zheng, L.; Zou, Z. A blockchain-based architecture for secure and trustworthy operations in the industrial Internet of Things. *J. Ind. Inf. Integr.* **2021**, *21*, 100190. [CrossRef]
14. Yu, B.; Wright, J.; Nepal, S.; Zhu, L.; Liu, J.; Ranjan, R. IoTChain: Establishing Trust in the Internet of Things Ecosystem Using Blockchain. *IEEE Cloud Comput.* **2018**, *5*, 12–23. [CrossRef]
15. Kapsoulis, N.; Psychas, A.; Palaiokrassas, G.; Marinakis, A.; Litke, A.; Varvarigou, T. Know Your Customer (KYC) Implementation with Smart Contracts on a Privacy-Oriented Decentralized Architecture. *Future Internet* **2020**, *12*, 41. [CrossRef]
16. Kumar, R.; Marchang, N.; Tripathi, R. Distributed Off-Chain Storage of Patient Diagnostic Reports in Healthcare System Using IPFS and Blockchain. In Proceedings of the 2020 International Conference on COMmunication Systems NETworkS (COMSNETS), Bengaluru, India, 7–11 January 2020; pp. 1–5.
17. Cong, R.; Liu, Y.; Tago, K.; Li, R.; Asaeda, H.; Jin, Q. Individual-Initiated Auditable Access Control for Privacy-Preserved IoT Data Sharing with Blockchain. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
18. Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Core. Available online: https://xmpp.org/rfcs/rfc6120.html (accessed on 2 September 2021).
19. MQ Telemetry Transport. Available online: https://mqtt.org/ (accessed on 2 September 2021).
20. Caiza, G.; Llamuca, E.S.; Garcia, C.A.; Gallardo-Cardenas, F.; Lanas, D.; Garcia, M.V. Industrial shop-floor integration based on AMQP protocol in an IoT environment. In Proceedings of the 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM), Guayaquil, Ecuador, 11–15 November 2019; pp. 1–6.
21. Yonezawa, T.; Ito, T.; Nakazawa, J.; Tokuda, H. SOXFire: A Universal Sensor Network System for Sharing Social Big Sensor Data in Smart Cities. In Proceedings of the 2nd International Workshop on Smart, Trento, Italy, 12–16 December 2016; pp. 1–6.
22. Rowe, A.; Berges, M.E.; Bhatia, G.; Goldman, E.; Rajkumar, R.; Garrett, J.H.; Moura, J.M.F.; Soibelman, L. Sensor Andrew: Large-scale campus-wide sensing and actuation. *IBM J. Res. Dev.* **2011**, *55*, 6:1–6:14. [CrossRef]
23. Bhatia, G.; Rowe, A.; Berges, M.; Spirakis, C. Sensor-Over-XMPP. Available online: https://xmpp.org/extensions/inbox/sensors.html (accessed on 2 September 2021).
24. Openfire. Available online: https://www.igniterealtime.org/projects/openfire/ (accessed on 2 September 2021).
25. Liu, Y.; He, D.; Obaidat, M.S.; Kumar, N.; Khan, M.K.; Raymond Choo, K.-K. Blockchain-based identity management systems: A review. *J. Netw. Comput. Appl.* **2020**, *166*, 102731. [CrossRef]
26. Ren, Y.; Zhu, F.; Qi, J.; Wang, J.; Sangaiah, A.K. Identity Management and Access Control Based on Blockchain under Edge Computing for the Industrial Internet of Things. *Appl. Sci.* **2019**, *9*, 2058. [CrossRef]
27. Zhao, Z.; Liu, Y. A Blockchain based Identity Management System Considering Reputation. In Proceedings of the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 28–30 September 2019; pp. 32–36.
28. Hopwood, D.; Bowe, S.; Hornby, T.; Wilcox, N. *Zcash Protocol Specification*; GitHub: San Francisco, CA, USA, 2016.
29. Kosba, A.; Miller, A.; Shi, E.; Wen, Z.; Papamanthou, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 839–858.
30. Garoffolo, A.; Kaidalov, D.; Oliynykov, R. Zendoo: A zk-SNARK verifiable cross-chain transfer protocol enabling decoupled and decentralized sidechains. *arXiv* **2020**, arXiv:2002.01847.
31. Chen, Y.; Nakazawa, J.; Yonezawa, T.; Tokuda, H. Cruisers: An automotive sensing platform for smart cities using door-to-door garbage collecting trucks. *Ad. Hoc. Netw.* **2019**, *85*, 32–45. [CrossRef]
32. Sasaki, W.; Eigen, Y.; Medela, A.; Litke, A.; Nunez, V.C.; Okoshi, T.; Nakazawa, J. SmileCityReport: Emotion-aware Participatory Sensing for Smart Cities with Double-sided Photo Shooting. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Austin, TX, USA, 23–27 March 2020; pp. 1–6.
33. Ali, M.S.; Dolui, K.; Antonelli, F. IoT data privacy via blockchains and IPFS. In Proceedings of the Seventh International Conference on the Internet of Things, Linz, Austria, 22–25 October 2017; pp. 1–7.
34. Zou, S.; Xi, J.; Wang, S.; Lu, Y.; Xu, G. Reportcoin: A Novel Blockchain-Based Incentive Anonymous Reporting System. *IEEE Access* **2019**, *7*, 65544–65559. [CrossRef]
35. Chen, Y.; Li, H.; Li, K.; Zhang, J. An improved P2P file system scheme based on IPFS and Blockchain. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 2652–2657.
36. Vukolić, M. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In *Open Problems in Network Security*; Camenisch, J., Kesdoğan, D., Eds.; Springer International Publishing: Cham, Switzerland, 2016; Volume 9591, pp. 112–125, ISBN1 9783319390277, ISBN2 9783319390284.