Secure Decentralized IoT Infrastructure

Vasco Santos
Departamento de Eletrónica, Telecomunicações e Informática, Universidade de Aveiro.
3810-193 Aveiro, Portugal
Email: vasco.santos@ua.pt

João Paulo Barraca
Instituto de Telecomnicações,
Campus Universitário de Santiago,
3810-193 Aveiro, Portugal
Email: jpbarraca@av.it.pt

Diogo Gomes
Instituto de Telecomnicações,
Campus Universitário de Santiago,
3810-193 Aveiro, Portugal
Email: dgomes@av.it.pt

***Abstract*—Despite many Internet of Things (IoT) Infrastructures having been implemented in recent years, none of them is truly prepared for a global deployment, where failure tolerance and scalability are an essential requirement. This article presents an alternative concept for IoT Infrastructures, which focuses on enhancing the traditional centralized architecture, usually operated by a single entity, into a decentralized architecture featuring multiple business roles. We propose a dynamic and self-configurable infrastructure on top of a structured Peer-to-Peer network. In addition, a set of communication protocols are provided in order to support heterogeneous devices, as well as data access, streaming and persistence. It is also an important focus of our proposal to have mechanisms that guarantee the privacy and security of the information flow and storage.**

I. Introduction

IoT infrastructures must be flexible and extensible enough to accommodate a high level of diversity, as well as to support billions of devices producing massive amounts of data. Ideally, one could argue that infrastructures should not exist, as they contribute to the existing problem of data and management silos. The Internet grew from an organic attachment of networks and servers, and this proved to be a much successful approach, as a result of the amount of distribution it presents, as well as the adaptation to new use cases and services.

Our work aims at taking a side step from current trends, in order to avoid the many issues created by a platform oriented to the Internet growth. In particular, we consider an Internet where an high number of loosely coupled devices, owned by users or even Telecom operators, provide a decentralized infrastructure, which is responsible for interconnecting all IoT platforms, without the need for any central entity. It focuses on storing and securing data at a global scale, instead of in a set of central entities. Accordingly, in a distributed approach, entities at the edge of the network exchange information and collaborate with each other in a dynamic way, providing a decentralized, self-organized and scalable infrastructure.

We specifically aim at avoiding centralized data storage and processing. Moreover, we consider that data should be strongly encrypted and only accessible to their rightful owners, or someone else with the appropriate authorization. Scalability and organic growth, by the addition of new sensors and systems, are also mandatory. This will have the potential of empowering local businesses and citizens to provide the services required for others to integrate their devices. In the end, we envision that everything will be connected to a global net, acting as a secure common data repository and communication channel, which we see as the real Internet of Things.

II. Related Work

Large players like IBM and Samsung Eletronics developed the Autonomous Decentralized Peer-to-Peer Telemetry (ADEPT) proof-of-concept (PoC)[1]. It aims to demonstrate several capabilities, which are fundamental for building a full decentralized IoT, and clearly demonstrate that these solutions need research, having interest from the market. This PoC selected three open source protocols for its implementation.

Combining the blockchain [1] with the IoT offers a set of capabilities, such as immutable history of transactions, as well as data security and privacy. ADEPT supports different types of devices, according to their performance and storage capabilities. However, it requires considerable resources for taking advantage of the blockchain benefits, which are far away from the common user, as they represent a considerable investment. Moreover, no blockchain solution was tested at a world scale [2]. Thus, this solution, paves the way to decentralized IoT systems, but it is not ready for the current view of the IoT, where the devices are expected to be inexpensive and their real-time communications may be crucial.

III. Proposed Infrastructure

The proposed infrastructure relies on a P2P overlay network, where IoT gateways are the network peers. This network is complemented by several WSN, which will provide data to the infrastructure. In addition, a undefined number of applications may communicate with it. With the proposed topology, it is possible to achieve the Internet level of scalability, once each new node joining the network shares its resources towards the overlay. Therefore, the topology consists of a community driven, decentralized network. Moreover, the system may provide different roles in the overlay, which will result in the share of computational power, storage capacity and network bandwidth, allowing the system to scale easily. In this section, the infrastructure requirements are identified and the design principles are outlined along with a brief motivation for making the associated decisions.

As the IoT continues expanding, researchers and companies search for economical and efficient solutions to secure the infrastructures. Public Key Infrastructure (PKI) has been the backbone of security in the Internet since its inception, relying in the use of digital certificates. Above all, PKI is an economical, reliable and proven technology that can be used in order to set up a secure and high-performance infrastructure [3]. Unlike conventional PKI and connected devices, the IoT will be composed by constrained devices, which can compromise the infrastructure performance.

978-1-5090-5856-3/17/$31.00 ©2017 IEEE

For a secure infrastructure, it is required to guarantee the peers identity and authenticity. Currently, the two most frequently adopted approaches are considerable different. One approach consists of the existence of a Certification Authority (CA), which acts as a trusted third party, in order to issue digital certificates, for certifying an entity public key. The other one resides in the use of a blockchain, which provides a distributed entity certification. After analyzing the ADEPT approach before, it is possible to understand that the current blockchain cannot be used to achieve the requirements of the proposed infrastructure. Above all, the real-time data and scalability required will not be guaranteed using it. For instance, a bitcoin transaction takes about 10 minutes to be validated [2]. For this reason, this infrastructure makes use of a centralized CA, instead of a blockchain.

A Distributed Hash Table (DHT) is an excellent choice for an IoT infrastructure, since it provides a guarantee (precise or probabilistic) on query cost, which makes possible that a request can be routed to a peer, who maintains the desired data quickly and accurately. Kademlia was the chosen DHT implementation, since it guarantees good performance, while also providing caching, resource replication and asynchronous range queries [4]. Taking into consideration that the overlay peers are required to communicate in a secure way, it is necessary to expand this DHT implementation to a trusted and secure overlay network. Considering the PKI, each peer of the overlay will keep an asymmetric key pair.

While the number of networked devices is becoming larger, their capabilities will diversify. Consequently, this infrastructure must be flexible enough to allow peers with different purposes in the overlay, according to their hardware constraints. Therefore, the infrastructure should accept peers with a set of different services enabled, namely Data Access, Data Collector, Persistence and Stream.

The infrastructure is required to guarantee data anonymity, in order to ensure that an attacker cannot link data to its owner. Thus, it must manage its users, as well as their information, in a secure way, without compromising the users privacy. Consequently, each entity should have a pseudonym, which is an anonym identifier of an entity, since the generated data should not be publicly associated with a certain entity [5]. The entity's pseudonym must be generated on the client side, using for example a specific hardware token, in order to assure that only the entity can produce its own identity.

Therefore, when an application intends to sign up a new user in the infrastructure, it must create two RSA key pairs on the client side, one for digital signatures and another one for encryption and decryption. Moreover, the user's password must be derived using a key derivation algorithm. Afterwards, the resulting derivation is used to cipher the private keys previously generated. In addition, the pseudonym is created using one of the generated private keys. Finally, the application provides the pseudonym, public keys and ciphered private keys to the infrastructure. As a result, the infrastructure propagates the users data trough the overlay. From that moment on, when a user intends to log in the application, it provides its pseudonym to the infrastructure and receives its data in response. Afterwards, it generates the password derivation to decrypt the private data (on client side).When a new sensor is connected to the infrastructure, the user must bind it to its account. Therefore, the infrastructure keeps a list of sensors each user may access, as well as an access list of users who may access each sensor.

The tremendous number and diverse nature of IoT deployments brings new considerations to the table, concerning how to actually implement interoperable infrastructures. The collector interface consists of a public interface, through which the sensors of a WSN will send their gathered data. Consequently, the store interface must accept different protocol communications, in order to be prepared to the heterogeneous environment of sensors. Therefore, the proposed infrastructure must accept data from protocols such as HTTP and MQTT.

Data gathered by sensors may be an attack target, which may compromise the privacy of business processes. As a result, the infrastructure must ensure the data storage security. When an IoT gateway receives data from a sensor (message contains its sensor identifier), it encrypts it through a randomly generated symmetric key. Afterwards, the generated key is encrypted using the public keys of the entities who have access to the received data, in order to generate an access code for each entity decrypt its data. Finally, having the encrypted data, as well as the sensor assertion, it is necessary to spread this data through the network.

The IoT global scale deploy promises to change the way we live. To achieve this, the data generated by sensors has to be retrieved by its owners. Consequently, the infrastructure must assure access control, through the analysis of the sensor's access list. The data access may be divided into two different interfaces, one for retrieving the last data of a sensor (stored in the DHT) and one for retrieving the data history of a sensor (Persistence). Moreover, it is crucial to guarantee request authenticity and data integrity through digital signatures.

Considering the IoT premise where millions of devices will be connected and producing massive volumes of data, the capacity and efficiency of data storage are imperative for an IoT infrastructure. Therefore, this data should be stored in a distributed database, which provides data consistency, as well as efficient data access and replication. In this context, time series databases are optimized for sequential writes, which will be indexed by timestamps [6]. Moreover, they also provide efficient retrieval for data in a time interval, as well as a fast removal of data that is no longer relevant. Consequently, a distributed time series database is an excellent choice for persisting the infrastructure data.

Some critical IoT scenarios demand real-time data. Consequently, a global IoT infrastructure is required to provide a stream of data, from sensors to applications, using protocols such as MQTT. This decentralized infrastructure behaves as a decentralized broker, where a peer whose data stream service is enabled may receive subscribe messages from applications, publish messages from sensors and must distribute published data to its subscribers. Thus, all stream peers must be subscribed among them, in order to allow sensors and applications to communicate with a single peer. For security purposes, subscribe messages must proceed to an access control mechanism. Moreover, when a publish message is received, its data must be encrypted before being distributed through its subscribers.

Finally, an example of a data flow, from a sensor to an application is illustrated in Figure 1. Periodically, each sensor

of a WSN sends its data to a Gateway (message 1). When the gateway receives data, it encrypts the data, as specified previously and propagates it through the overlay (message 2). A user (properly authenticated) asks for the sensor's data, through an application (message 3). The application decrypts the received data and displays it. Moreover, the application may subscribe a set of sensors, and receive a data stream of their periodic data.

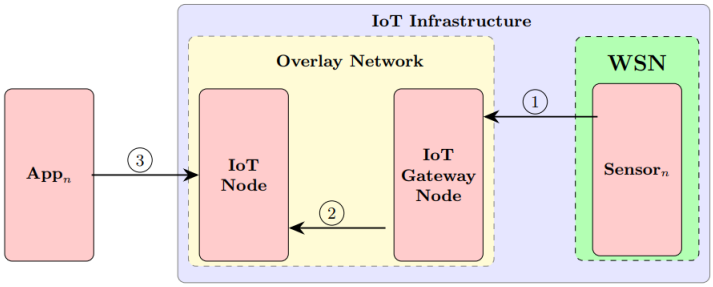

Fig. 1: Infrastructure Data Flow.

IV. Prototype and Evaluation

The proposed infrastructure was implemented to verify its behavior according to the defined requirements. Accordingly, gateways were developed using Python Twisted, an event-driven networking engine. As a result of its asynchronous nature, it can handle thousands of connections in a single thread, providing great performance and scalability. In addition, the gateways, whose Persistence Service is enabled, have InfluxDB enabled, a distributed time-series database.

Complementing the overlay peers, a CA was developed using Python's AsyncIO, as well as a web application composed by a Node.js server and an AngularJS application. Finally, two simulators were developed for sending periodic data to the infrastructure simulating a set of sensors. Both were developed using Python Twisted, but one communicates with the infrastructure using MQTT protocol, while the other one uses HTTP.

The implemented prototype is composed by seven peers. Four of them are RaspberryPi's, which have hardware limitations and consequently, do not have persistence and stream services enabled. The remaining ones are virtual machines with capabilities to have all the services enabled.

Considering a 7 days test scenario in a local network, the infrastructure received messages from the simulators, where each sensor had its own periodicity varying between 20 and 150 seconds. The MQTT simulator contains 66 sensors, while the HTTP has 77 sensors. On the one hand, the HTTP simulator sent a total of 998889 messages, with an average response time of 116.2ms. On the other side, the MQTT simulator provided 908445 data messages, with an average response time of 0.58ms. Thanks to its connection-oriented nature, MQTT provided excellent and uniform results, with a standard deviation of about 0.7. In contrast, HTTP presented more disperse response times, with a standard deviation of approximately 101.1. It is expected that in the Internet scale, the response times will increase. However, the response times will be uniform over time, as a result of the use of a time-series database.

During the test, peers' CPU usage was collected each 60 seconds. Taking into account the obtained results, the Stream Service, as well as the Persistence Service are considerable heavy, as result of the need to have a local message broker and a local database running. It is important to notice that InfluxDB recommends a minimum of 2-4 CPUs and it was used only one.

In the context of the security and privacy requirements, the data security during its flow and storage has to be validated. Therefore, it is possible to verify the database content using the InfluxDB shell. Moreover, as the decryption is processed on the client side, the content of the packets that flow through the network is also encrypted. Each chunk of data contains a `code`, which consists of the symmetric key necessary to decrypt the `data`. This symmetric key has to be decrypted using the user's private key before being used.

V. Conclusions

A global decentralized infrastructure has potential to reduce the platform and maintenance costs. Moreover, decentralization provides fault tolerance by removing single points of failure, as well as scalability, thanks to the community driven nature of P2P. In addition, the proposed infrastructure may provide privacy to the users, as a result of their data being encrypted and dispersed over the network. Finally, it provides all the necessary resources to interconnect multiple business processes and use cases, each one with its own requirements.

The proposed infrastructure matches the stated requirements and is a good solution for a global IoT infrastructure. This results from the current state of the IoT, where devices are intended to be inexpensive and lightweight, as well as the current state of blockchain, which does not scale to the Internet level.

References

[1] IBM, "Empowering the edge - practical insights on a decentralized internet of things," 2015.

[2] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, and E. Gün, "On scaling decentralized blockchains," *Proc. 3rd Workshop on Bitcoin and Blockchain Research*, 2016.

[3] Digicert, *White paper: pki - the security solution for the internet of things*.

[4] P. Maymounkov and D. Mazières, "Kademlia: a peer-to-peer information system based on the xor metric," *First International Workshop on Peer-to-Peer Systems*, pp. 891–921, 2002.

[5] R. Lu, X. Lin, T. H. Luan, X. Liang, S. Member, and X. S. Shen, "Pseudonym changing at social spots(an effective strategy for location privacy in vanet)," vol. 61, no. 1, pp. 86–96, 2012.

[6] D. Namiot, "Time series databases," 2015.