

# Orthus: A Blockchain Platform for Smart Cities

Stefano Momo Loss  
momoloss10@gmail.com  
Dept. of Informatics and Applied  
Mathematics  
Federal University of Rio Grande do  
Norte  
Natal, Brazil

Nélio Cacho  
neliocacho@dimap.ufrn.br  
Dept. of Informatics and Applied  
Mathematics  
Federal University of Rio Grande do  
Norte  
Natal, Brazil

Frederico Lopes  
fred@imd.ufrn.br  
Metropole Digital Institute  
Federal University of Rio Grande do  
Norte  
Natal, Brazil

## ABSTRACT

Blockchain has been widely used to store decentralized and secure transactions involving cryptocurrency (for instance, Bitcoin and Ethereum solutions). On the other hand, Smart city applications are concerned about how data and services can be safely stored and shared. In this regard, this research investigates the use of Blockchain in the Smart Cities scenario. Based on that, a platform named Orthus was proposed to support Blockchain in Smart City initiatives focused on scalability. This work presents a case study about how to use the proposed platform in the Natal Smart City Initiative in Brazil to handle land registration. Moreover, it also compares this platform with other implementations that use Blockchain in different domains. Finally, this research confirms that the use of blockchain technology has much to contribute to Smart City solutions once it enables the creation of solutions in distributed networks to meet the demand of the entire population.

## KEYWORDS

Blockchain, Smart Cities, Scalability.

## 1 INTRODUCTION

The world's first utterly decentralized cryptocurrency was the Bitcoin [3]. Bitcoin definition was created in 2008 by an unknown person using the pseudonym Satoshi Nakamoto [9]. One of the main actors responsible for Bitcoin's success is Blockchain technology. The creation of Blockchain is inherent in the creation of Bitcoin. Blockchain can be defined as a distributed database that registers an ordered list of transactions records that are immutably linked together through a chain, on blocks [9].

Blockchain digitally creates a trusted entity that replaces transaction certifying and centralizing entities (banks, governments, notaries). According to [9], some of the potential applications of Blockchain are supply chain, digital identity, voting, healthcare, and government [9, p. 37]. Some of them aim to solve social problems in the most diverse areas, contributing to the construction of Smart Cities. Smart Cities infuse "information into its physical infrastructure to improve conveniences, facilitate mobility, add efficiencies, [...], collect data to make better decisions and deploy resources effectively, and share data to enable collaboration across entities and domains"[7]. This infusion occurs when using the information and management system to solve social problems.

Currently, most municipal public services already use some Management and Information systems developed by different public or private companies and, in most cases, in an isolated way without communication standards. If a set of isolated systems that run independently of each other had a reliable, secure, distributed decentralized database to store and share information between these systems, the integration would decrease the number of frauds since all operations could be verified confirmed. Thus, in this perspective, Blockchain can be used to generate interoperability between systems in addition to sharing data and services in a transparent, secure, and reliable way, without centralizing entities, enabling the integration of city services to improve the quality of life of the population [11].

There are many ways to integrate systems to share data between them in decentralized form, most of which are ad hoc to meet a specific customer need. There are already some ways to integrate system data, such as using middleware or frameworks to compose systems. The problem with sharing data between different systems is the reliability of data exchanged. Blockchain can be used to ensure decentralized and secure data sharing with security through transaction storage. However, Blockchain was initially created to store cash-only transactions [3].

Therefore, Blockchain has a considerable latency in processing to create and verify new transactions and create blocks with them needing a high computational in this process. This consensus algorithm varies by blockchain type and implementation. Blockchain can not handle large amounts of diversified data produced by Smart Cities since a Smart City application has particular requirements. Thus, the Blockchain implementation must use some Big Data technology to store and manipulate a large and varied amount of data at the required velocity.

Thus, based on the context and limitation of current approaches, the objective of this work is to propose a blockchain platform architecture to be used in the context of Smart Cities. Provide an implementation for the proposed architecture. And compare this implementation with other blockchain platforms within that same scope using a case study involving land registry.

The remainder of this paper is organized as follows. Section 2 presents related works presenting and comparing other Blockchain platforms. Section 3 introduces and describes the Orthus architecture, a Blockchain platform for Smart Cities. Section 4 presents a case study about land registry comparing Orthus implementation with another Blockchain approach. Section 5 revisits the achieved contributions and presents perspectives for future work.

In: III Concurso de Teses e Dissertações (CTD 2021), Minas Gerais, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2021.

© 2021 SBC – Sociedade Brasileira de Computação.  
ISSN 2596-1683

## 2 RELATED WORKS

Some works already use Blockchain in the context of Smart Cities (for example, money, health [8], automotive [10], education [4]). However, none of these applications used a platform that could be generic enough to be used generally for other types of domains. On the other hand, there are other solutions for general purpose that are developed for use by the market, out of the Smart Cities context, which focuses on different approaches, namely:

**EOSIO**<sup>1</sup> is a Blockchain platform for businesses with industry-leading transaction speed and flexible utility. "The smart contract platform claims to eliminate transaction fees and also conduct millions of transactions per second." [5]. **NEO**<sup>2</sup> "is the use of Blockchain technology and digital identity to digitize assets, the use of smart contracts for digital assets to be self-managed, to achieve "smart economy" with a distributed network". [1] **Hyperledger Fabric**<sup>3</sup> is the first ledger developed by The Linux Foundation in partnership with IBM and Digital Assent. It is a framework for application development where the Smart Contracts are named as ChainCode. The communication between nodes in the Hyperledger Fabric occurs through channels, making transactions private to channel members only. The main feature of this solution is that it is modular and configurable according to the application developed. **Corda Platform**<sup>4</sup> Corda is an open-source Blockchain platform that enables businesses to transact directly and in strict privacy using smart contracts, reducing transaction and record-keeping costs and streamlining business operations.

## 3 ORTHUS PLATFORM

Orthus platform was designed to be a permissioned Blockchain platform that supports the requirements to share information between Management Information System (MIS) present in Smart Cities and promote interoperability between them. Blockchain provides system integration in Smart Cities because it involves systems from multiple sources (different private companies and municipal or federal agencies) where there is no reliability between them. With Blockchain, data from these systems would be shared securely, immutable with decentralized management.

To perform this integration, each MIS must implement the Orthus component to be part of the network. Orthus components are instantiated once for each system, and this association (MIS + Orthus components) is called a node. Communication between an MIS and the Orthus Gateway component occurs through REST requests. The Orthus network comprises distributed nodes and a network of brokers that enables data exchange to share transactions and blocks to be validated by all. An example of using Orthus to integrate some systems can be seen in Figure 1. In the diagram, you can see the integration of these systems in two different cities where each city would have its Orthus network giving rise to an Orthus organization. You can also see the integration of the organization of these two or more cities by using a new broker to integrate them.

The platform architecture to achieve integration between the systems must contain the following components, namely: i) an API gateway, to receive the REST requests of creating new transactions

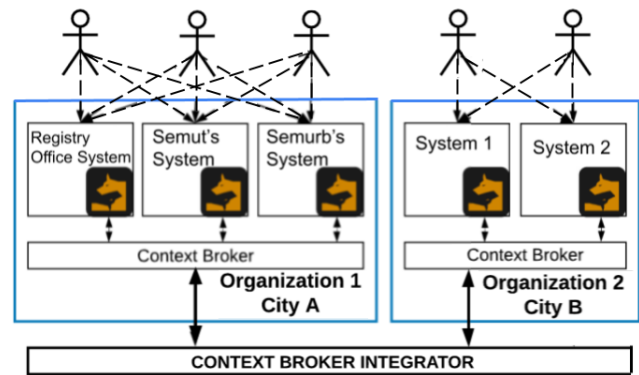


Figure 1: Integration scenario through Orthus

from systems; ii) a Controller Service, to orchestrate the solicitations done in Orthus component; iii) a Transaction Service, this component is responsible for validating and creating new transactions; iv) Block Service, the component responsible for creating a new block with the transactions added before; v) Blockchain Service, this component is responsible close the block with the transactions and inserts the block into the database (adding this block into the Blockchain); vi) a Database, the component responsible for storing the Blockchain; vi) Communication Service, the component responsible for communication with other Orthus peers, it utilizes a context broker bus to facilitate the conversation and viii) Context Broker, this component is responsible for integrating all Orthus peers as a communication bus for them.

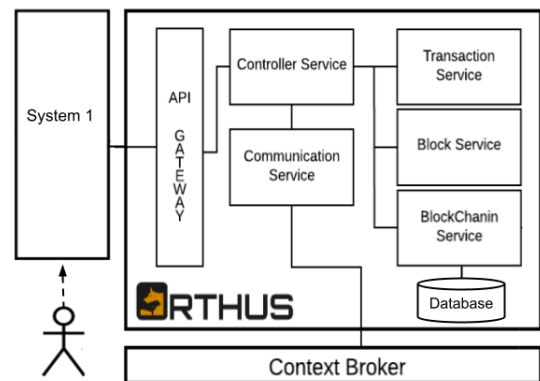


Figure 2: Orthus components integration

Figure 2 shows the integration of the internal components of the Orthus node as previously described. It also shows the integration with a possible system to be integrated into the Orthus platform. In this case, it exemplifies the integration of a System with the Orthus platform. A user requests the System and sends a REST request to Orthus to access the Blockchain data or request a new transaction. Remember that for each system to be integrated into the platform, an instance of the Orthus node is required. In this way, it is possible that a system can share information securely between the systems that are part of the platform. Since, in each of the Orthus nodes, the data is verified and validated.

<sup>1</sup>www.eos.io

<sup>2</sup>www.neo.org

<sup>3</sup>www.hyperledger.org/projects/fabric

<sup>4</sup>www.r3.com/corda-platform/

### 3.1 An Orthus Implementation

A set of technologies was chosen to implement a version of Orthus according to the architecture presented in Section 3. The technologies can be replaced by similar ones that meet the architectural description. So then, open-source technologies powering scalable real-time and data-driven applications were chosen to be used in the implementation. Play Framework<sup>5</sup> for Java is used to create a REST API component to enable communication between the node and the system, but it could also use the Spring Framework.

Already the Akka<sup>6</sup> for Java is used in the implementation of the actors of the Orthus component; these actors communicate by exchanging messages following the Reactive Design Patterns [6]. Messages are exchanged locally, allowing maintain the relative simplicity of the event-driven programming model. As it is one of the leading and most used implementations of the Reactive Design Patterns, it is the reason to use Akka in the Orthus implementation. Cassandra<sup>7</sup> database is used to storing block-generated data (transaction hash, previous block hash, timestamp) by the Blockchain within each node. The data itself will be stored within each of the systems based on the validated blocks. However, other databases could be used to store transactions and block data.

Orion Context Broker<sup>8</sup> is used as a bus to share the transactions and the blocks all over the nodes. Orion will also be used (not yet implemented) to communicate across organizations. For this, one organization's Orion would subscribe to integrating Orion to receive data from the other organization, enabling data exchange between organizations and making their data available in more than one location. The development of the Orthus platform is part of the Smart Metropolis<sup>9</sup>.

## 4 PLATFORM EVALUATION

Currently, Natal has a bureaucratic process to transfer land ownership involving three institutions: Land Registry Office, Municipal Secretary of Environment and Urbanism (Semurb), and Municipal Secretary of Taxation (Semut). The most critical operation involving these different institutions is the land transfer operation between people. It is necessary to ensure that the seller does have the land since there may be disagreement about the landowner among these institutions. Once these institutions are isolated and do not share data information, making fraud possible in this process, in this way, Blockchain can be used to solve this problem as a distributed database across a network to store land information with security, reliability, and privacy and enable interoperability between these institutions. The land transfer transaction was chosen to be implemented by the blockchain platforms to be assessed.

### 4.1 Case Study with Orthus

In implementing the case study using the Orthus platform, each system has an internal Blockchain node that communicates with each other via Orion Context Broker. New transactions can be done by any system of the case study (Registry Office's System, Semut's System, or Semurb's System) where each system is considered an Orthus node, as can be seen in Figure 1. The Land class

was implemented based on the AbstractCapitalGood class with the required terrain information (area, address, status) and the TransferLandTransaction class based on the AbstractTransaction class with the information required (buyer identifier, seller identifier, land identifier, value).

Along with the implementation of the class for this transaction, it is necessary to code in JSON format of the transaction to perform a land transfer, as this data varies with the application and its context. In this case study, a landlord transfer requires the public key of the seller, buyer, and land in addition to the purchase price. Those transactions are validated and shared by the other peers. From time to time, valid transactions are grouped in blocks; the first node that can close the block releases it to the other nodes in the network. When a node receives a block, it validates and inserts it in the Cassandra database.

### 4.2 Case Study with Hyperledger Fabric

The Hyperledger Fabric platform was used to recreate the same case study. Within the developers' manual<sup>10</sup> there is an example that involves the registration of automobiles, named *Fabcar*, implemented using Java language. This example has been modified to meet the case study requirements of the land record— a Land class where land attributes are defined.

Tests must be carried out under the same conditions on both platforms to maintain fairness between tests. Hyperledger Fabric platform currently does not automatically provide access to new transactions using the REST API. So, a JavaScript REST API web service was developed to communicate with the Hyperledger Fabric following a Medium<sup>11</sup> tutorial.

### 4.3 Performance test definition

The average response time and throughput are used as indicators in the analyses. The Orthus platform was compared with the Hyperledger Fabric platform using the same case study. Hyperledger Fabric platform was chosen for comparison because [2] promises one of the highest transfer rates of the platforms found, more than 3500 transactions per second. Four test cases were defined, changing the number of simultaneous users. The first test case was executed with five users, then 50, 500, 5000, 10000, and ending with 15000 simultaneous users, each requesting ten land transfer transactions between two people in sequence.

The Google Cloud n1-standard-2 (2 vCPUs, 7,5 GB of memory) instance was chosen to run the case study on both platforms. One instance for each platform was used to run the case study test using Docker<sup>12</sup>. Both case studies were implemented with three peers each and made requests via the REST API using Apache JMeter<sup>13</sup> from an external computer to the cloud to capture the information previously described.

### 4.4 Results analysis

After running the case study test for each of the platforms, as described above, shortest and most prolonged time, average request time, standard deviation, error rate, and throughput for each case

<sup>5</sup><https://www.playframework.com/>

<sup>6</sup><https://akka.io/>

<sup>7</sup><https://cassandra.apache.org/>

<sup>8</sup><https://fiware-orion.readthedocs.io/>

<sup>9</sup><http://smartmetropolis.imd.ufrn.br/>

<sup>10</sup><https://hyperledger-fabric.readthedocs.io/en/release-1.4/ledger/ledger.html#example-ledger-fabcar>

<sup>11</sup><https://medium.com/@kctheservant/an-implementation-of-api-server-for-hyperledger-fabric-network-8764c79f1a87>

<sup>12</sup><https://www.docker.com/>

<sup>13</sup><https://jmeter.apache.org/>

test was summarized in Table 1. According to it, the Orthus throughput remains between approximately 380 to 495 transactions per second. The throughput of the five users test case of Orthus is lower than fifty simultaneous users test cases because a characteristic of the Orthus platform can dynamically increase the number of instances of its actors. As for the results of the tests using the Hyperledger Fabric platform, the throughput kept a decrease of approximately 470 in the test case with five simultaneous users to approximately 330 transactions per second in the last test case with 10000 concurrent users.

Users	Samples 10 x #Users	Response time (in ms)			Deviation	Error	Throughput
		Average	Min	Max			
<b>Orthus</b>							
5	50	12	2	36	9.7	0.0%	402.8/s
50	500	45	2	210	36.43	0.0%	494.2/s
500	5000	630	2	1970	303.72	0.0%	486.7/s
5000	50000	1258	3	2586	813.24	0.0%	437.3/s
10000	100000	1680	5	3153	1298.24	0.0%	377.8/s
15000	150000	2974	18	4542	2239.11	2.5%	262.4/s
<b>Hyperledger Fabric</b>							
5	50	15	5	87	23.0	0.0%	473.6/s
50	500	67	7	286	47.32	0.0%	468.8/s
500	5000	834	9	2262	412.17	0.0%	447.1/s
5000	50000	1586	15	3487	1013.39	0.0%	403.4/s
10000	100000	2134	21	4257	1876.71	0.0%	328.3/s
15000	150000	x	x	x	x	x	x

**Table 1: Comparison Orthus vs Hyperledger Fabric**

During the execution of the test cases with 15000 users, the Orthus implementation managed to fulfill all the requests, but with a two and a half percent error in the answers. The implementation using the Hyperledger Fabric did not support the number of requests. This case study implementation crashed the server in the cloud, making it impossible to complete the test. Therefore, the maximum possible number of simultaneous transactions for the case study using Hyperledger Fabric is approximately 13500, since the last JMeter log was for user number 13426. The difference in the average response time for each platform where Orthus has a better average time in all test cases. This result indicates that Orthus performed better in the number of transactions performed per second than the same case study test using Hyperledger Fabric in the same test case. There is a growth in the average response time on both platforms, between 5 and 500 simultaneous users. However, the growth becomes less pronounced after 500 users until the test of 10,000 users in both approaches.

Orthus' throughput (transaction per second) for the test case with five users is lower than the test case with fifty users because the number of Akka actors is dynamically controlled according to the need. The general throughput on the Orthus platform shows a small decrease while on the Fabric platform, this decrease is much higher compared to Orthus. Moreover, after 500 users, the decline is practically the same on both platforms, but Orthus' throughput remains higher than Hyperledger Fabric.

## 5 CONCLUSION

The use of Blockchain technology has much to contribute to integrating systems in the Smart City context. Once it enables the

creation of solutions in distributed networks to share data and services in a transparent, secure, and reliable way without centralizing entities, enabling the integration of city services to improve the population's quality of life. Therefore, this work presented how Blockchain can be used in Smart Cities environments through the definition and implementation of the Orthus platform. Orthus enables interoperability by securely exchanging data between many Information Systems.

The current limitations on Orthus implementation can be resolved with the following future work. In the current implementation of Orthus, there is no use of Smart Contracts. Smart Contracts execution must be performed by the systems that make up Orthus nodes, thus enabling interoperability between systems. Another gap is that Orthus only uses one instance of Orion Context Broker. However, it is necessary to use an Orions network to ensure communication between organizations and avoid having only one way of communication between nodes. Orthus was designed to be a private Blockchain platform, and a management mechanism is required to control the addition of new nodes to the platform. Thus, developing a Node Management Engine and a voting mechanism is still necessary to use Byzantine Fault Tolerance as a consensus mechanism and obtain a better transaction rate per second.

## 5.1 Resulting Publications

Stefano Loss, Nélio Cacho, Frederico Lopes, João Marcos do Valle. Orthus: A Blockchain Platform for Smart Cities. **Best Paper Award** in: 5th IEEE International Smart Cities Conference (IEEE ISC2 2019) 2019, Casablanca/Morocco. Proceedings of IEEE International Smart Cities Conference ISC2 2019, 2019.

## ACKNOWLEDGMENT

This work is supported by the Brazilian funding agency CNPq and SmartMetropolis Project<sup>14</sup>.

## REFERENCES

- [1] 2018. NEO White Paper. Available at <https://docs.neo.org/en-us/whitepaper.html> (2018).
- [2] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*. 1–15.
- [3] Jerry Brito and Andrea Castillo. 2013. *Bitcoin: A primer for policymakers*. Mercatus Center at George Mason University.
- [4] Guang Chen, Bing Xu, Manli Lu, and Nian-Shing Chen. 2018. Exploring blockchain technology and its potential applications for education. *Smart Learning Environments* 5, 1 (2018), 1.
- [5] Ian Grigg. 2017. EOS-An introduction. (Whitepaper) [iang.org/papers/EOS\\_An\\_Introduction.pdf](http://iang.org/papers/EOS_An_Introduction.pdf) (2017).
- [6] Roland Kuhn, Brian Hanafee, and Jamie Allen. 2017. *Reactive design patterns*. Manning Publications Co.
- [7] Rosabeth Moss Kanter and Stanley S Litow. 2009. *Informed and interconnected: A manifesto for smarter cities*. (2009).
- [8] Peter B Nichol, CSSMBB PMP, and CQM CMPL. 2016. *Co-Creation of Trust for Healthcare: The Cryptocitizen Framework for Interoperability with Blockchain*. (2016).
- [9] David Rull Aixa. 2018. Analysis and study of data security in the Internet of Things paradigm from a Blockchain technology approach. (2018).
- [10] Pradip Kumar Sharma, Seo Yeon Moon, and Jong Hyuk Park. 2017. Block-VN: A distributed blockchain based vehicular network architecture in smart City. *Journal of Information Processing Systems* 13, 1 (2017), 84.
- [11] Jianjun Sun, Jiaqi Yan, and Kem ZK Zhang. 2016. Blockchain-based sharing services: What blockchain technology can contribute to smart cities. *Financial Innovation* 2, 1 (2016), 26.

<sup>14</sup><http://smartmetropolis.imd.ufrn.br>