

## Article

# A Cost Analysis of Implementing a Blockchain Architecture in a Smart Grid Scenario Using Sidechains

Iago Sestrem Ochoa <sup>1,2,\*</sup> , Luis Augusto Silva <sup>1</sup> , Gabriel de Mello <sup>1</sup> ,  
Juan Francisco de Paz <sup>3</sup> , Nuno M. Garcia <sup>4,5</sup>  and Valderi Reis Quietinho Leithardt <sup>1,4,6</sup> 

<sup>1</sup> Laboratory of Embedded and Distributed Systems-LEDS, University of Vale do Itajaí, Itajaí-SC 88302-901, Brazil; luis.silva@edu.univali.br (L.A.S.); gabrieldemello@edu.univali.br (G.d.M.); iago.ochoa@edu.univali.br; valderi.leithardt@ubi.pt (V.R.Q.L.)

<sup>2</sup> Departamento de Informática e Redes de Computadores, Instituto Federal Catarinense (IFC), Brusque 88354-300, Brazil

<sup>3</sup> Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca, Spain; fcofds@usal.es (J.F.d.P.)

<sup>4</sup> Departamento de Informática, Universidade da Beira Interior, 6201-001 Covilhã, Portugal; ngarcia@di.ubi.pt

<sup>5</sup> Instituto de Telecomunicações, Universidade da Beira Interior, 6201-001 Covilhã, Portugal

<sup>6</sup> COPELABS, Universidade Lusófona de Humanidades e Tecnologias, 1749-024 Lisboa, Portugal

\* Correspondence: iago.ochoa@edu.univali.br

**Abstract:** Smart Grid systems have become popular and necessary for the development of a sustainable power grid. These systems use different technologies to provide optimized services to the users of the network. Regarding computing, these systems optimize electrical services by processing a large amount of data generated. However, privacy and security are essential in this kind of system. With a large amount of data generated, it is necessary to protect the privacy of users, because this data may reveal users' personal information. Today, blockchain technology has proven to be an efficient architecture for solving privacy and security problems in different scenarios. Over the years, different blockchain platforms have emerged, attempting to solve specific problems in different areas. However, the use of different platforms fragmented the market, which was no different in the smart grid scenario. This work proposes a blockchain architecture that uses sidechains to make the system scalable and adaptable. We used three blockchains to ensure privacy, security, and trust in the system. To universalize the proposed solution, we used the OSGP protocol and smart contracts. The results show that architecture security and privacy are guaranteed, making it feasible for implementation in real systems. Although scalability issues regarding the storage of data generated still exists.

**Keywords:** blockchain; sidechain; smart grid

## 1. Introduction

Smart Grid (SG) is a large-scale electrical network infrastructure mainly characterized by security, agility, and resilience, capable of handling new threats and unforeseen conditions. In 2005, the authors introduced this concept in [1], known as smart electrical networks. The agents that act on these networks can communicate and cooperate in a self-configuring mode, considering that a new element can join the network, or a random event can cause a requirement for correction. Although SG networks ensure efficiency in electrical systems, problems still exist for its implementation to be efficient in a holistic way.

According to [2], one of the problems to be solved for the implementation of SG networks is the privacy issue. The work developed in [3] states that, in general, data privacy affects the security of who

is connected, due to store information related to the user's life. The use of control access techniques, which guarantee reliable authentication, authorization, and confidentiality of the services, does not ensure a holistic solution to the privacy problem. This difficulty happens due to the data needs to be disseminated in different parts of the network. In addition to privacy issues, another challenge facing the popularization of SG networks is security. Security problems on SG networks can cause disastrous effects on the network. According to [4], an SG network is vulnerable to cyber-attacks such as traffic analysis, social engineering, cracking, spoofing, denial of service, and others. If a security flaw exists in equipment connected to the network that could compromise the system, an update would be necessary to correct all devices, impacting a high monetary cost. To avoid security problems, the use of communication protocols that guarantee security in SG networks is essential.

For the data monitoring and communication in SG networks, different protocols exist for these areas. These protocols aim to ensure efficient solutions to the reliability and security of the network [5]. However, the use of different protocols fragments the development of new applications, generating different network architectures directed to the SG segment. The use of a protocol that satisfies the requirements of an SG network is necessary for the development of new applications.

The Open Smart Grid Protocol (OSGP) is a protocol widely used in SG applications. OSGP Alliance developed the OSGP and published as a standard by the European Telecommunications Standards Institute [6]. The protocol implements all layers of the OSI model and provides security through cryptographic methods for Smart Meters (SM). However, studies expose security flaws on the OSGP encryption method. The work presented by Kursawe and Peters [7] shows a structural weakness in the cryptographic process of OSGP. The main flaw observed was the use of RC4 encryption, with each new key generated for each message transmitted, only the first eight bytes of this new key is different from the others. Another problem observed was that is used only one key for authentication. This same authentication key is used to derive the encryption key, so if the authentication key is exposed, all encryption keys are compromised.

Security flaws are not exclusive of the OSGP protocol. As shown in [5], other SG protocols also have security flaws that can compromise the entire network. Conventional techniques of privacy and security are not sufficient to guarantee these requirements. For this, it is necessary to use an architecture that holistically guarantees security and privacy.

In 2008 Satoshi Nakamoto presented the Bitcoin system to the world. Bitcoin is a virtual currency, also known as cryptocurrency [8]. This technology works based on P2P communication among network users, eliminating the need for a third party to validate transactions between the peers of the network. To ensure integrity, security, privacy, and reliability of data transmitted over the network, bitcoin uses the technology known as the blockchain. Blockchain acts as a distributed reason book. The information is stored on blocks and validated through a consensus algorithm. The process of validating blocks is called mining. To encourage users to participate in the mining process, users that participate in the mining process receive a reward in cryptocurrency. Due to its characteristics, the blockchain got the attention of the applications developers.

Blockchain proved to be an innovative technology due to its characteristics, which can solve security and privacy issues [9]. It is possible to find blockchain usage in medical environments [10], on IoT scenarios [11] and in industrial environments [12]. The trend of the use of blockchain was not different for the SG scenario. Commercial solutions that use blockchain on SG scenarios already exist. Nowadays, the primary use of this technology consists of electrical energy trade between different consumers. However, different works are attempting to develop blockchain architectures for the SG scenario that guarantee users security and privacy.

With the development of blockchain architectures focused on the SG scenario, various authors propose entirely new solutions that lack the use of existing SG protocols. These solutions are difficult to implement because due to the complexity of adapting them to devices that already exist in SG networks. The use of existing protocols on new blockchain architectures can favor the implementation of this technology in SG networks.

To solve the problems previously presented, in this article, we propose a blockchain architecture focused on the SG scenario that uses sidechain. Our architecture uses the OSGP protocol integrated into three different blockchains, proposing to guarantee privacy and security in SG networks holistically. Our architecture allows users to define their privacy preferences in a tamper-proof way, using a privacy blockchain. The access to the information of each user by the electric company is stored in storage blockchain to ensure the reliability of the system. In this way, users and companies benefit from the use of this type of architecture.

The principal contribution of this article is on the definition of a blockchain architecture that uses a protocol widely used in the SG scenario, supporting the implementation of this architecture on existing systems. Regarding the functionalities of our architecture, it provides security, reliability, and privacy for users through the use of different blockchains. Our architecture also provides scalability for SG applications, implementing the architecture in a sidechain concept, specifically designed to enable large-scale application development using the OSGP protocol.

The paper is structured as follows. Section 2 presents the background with the fundamental concepts to understand this work. Section 3 shows the related works. Section 4 exhibits the methodology used in the development of the proposed architecture and the details of our architecture. Section 5 illustrates the results obtained through the tests developed. Finally, section 6 shows the conclusions obtained with the development of this work.

## 2. Background

In this section are presented the fundamental concepts to understand our architecture. In Section 2.1 we present the SG concept focusing on security and privacy issues. We also show the blockchain concept as a solution to security and privacy problems in section 2.2. In Section 2.3 we outline the advantages of using the Ethereum blockchain in application development, the concept of sidechain, and how it can revolutionize blockchain technology.

### 2.1. Smart Grid

According to [13], an SG system is the integration of information technology with the generation, transmission, and distribution systems of electrical energy. It is possible to describe four characteristics of an SG system: (i) Increase efficiency and profitability of the system. (ii) Supply tools for the consumer to manage energy use. (iii) Optimization of the resilience and quality of energy of the system. (iv) Development of new technologies such as renewable energy generation (solar, wind, and others), storage of energy (batteries), and electric vehicles.

One of the critical features in an SG network is that consumers also become producers (or prosumers); this happens because they can produce renewable energy in their houses through alternative sources. Analyzing this from an extended perspective, they acquire the responsibility of generating electricity with the same quality of traditional generation sources. According to the essential characteristics of the SG networks before mentioned, with decentralized prosumers, three of these characteristics are guaranteed [14].

To ensure efficient management of energy usage, SG networks need to allow the prosumers to perform real-time monitoring of electricity consumption and generation. In this way, they can choose to store or sell the energy excess produced for other SG network users. SG networks require the development of communication infrastructures that support the growth and density of the system, guaranteeing the quality of service necessary for the operation for large scale applications.

The Smart Meter (SM) is the critical part of an SG network. An SM is responsible for collect, process, and manage the information obtained about the electrical usage on a residence. They are also responsible for collecting data from the electricity grid. The functionalities of an SM are various, these functionalities are intended to provide the consumer a wide range of information such as, the amount of energy consumed in real-time, amount of energy used in the last hour, week and month (and how

much it cost), classify the use of electricity at high, medium or low levels, and others. With the amount of information obtained from the user through an SM, privacy issues appear [15] [16].

According to [17], even transporting the data obtained by the SM in a secure communication channel to the electrical company, the electrical company still get a large amount of data. With this data, it is possible to know when the consumer is at home, as shown in [18]. Some devices have specific consumption patterns, so it is possible to determine, for example, when television or a washing machine is turned on.

Regarding security, failures can seriously compromise network performance. An example of a situation that can happen with a security breach in SG systems is a fraud in SMs. If a hacker has access and controls the information contained in SMs, it is possible to manipulate metering data and send forged data to the electrical company. Considering a catastrophic scenario, a hacker could have full access and control over someone’s SM [19].

Based on the information presented, security and privacy are two critical points in SG systems. Thus, it is necessary to ensure these requirements holistically for SG systems to operate efficiently. For this, the use of different technologies can help in the development of architectures that guarantee security and privacy in SG environments.

2.2. Blockchain

Proposed by Nakamoto in 2008 [8], Bitcoin proposes to promote the exchange of its cryptocurrency in a decentralized way using the blockchain technology to store the data and guarantee the authenticity of the transactions made. Over the years, blockchain technology has caught the attention of researchers because it can enable the development of various applications besides the exchange of cryptocurrencies between users.

The blockchain is a technology that its primary function is to guarantee information security by storing transactions in a decentralized way between the nodes of the network. The blockchain can be compared to a public ledger, creating consensus and direct communication between two parties, without the intermediary of third parties.

Transactions consist of information stored inside blocks that are later validated through a consensus algorithm and added to the blockchain. Inside the blocks is stored the transactions performed, the hash of the current block, the hash of the previous block, the number of the last block, and a nonce value. Figure 1 illustrates the blockchain structure described.

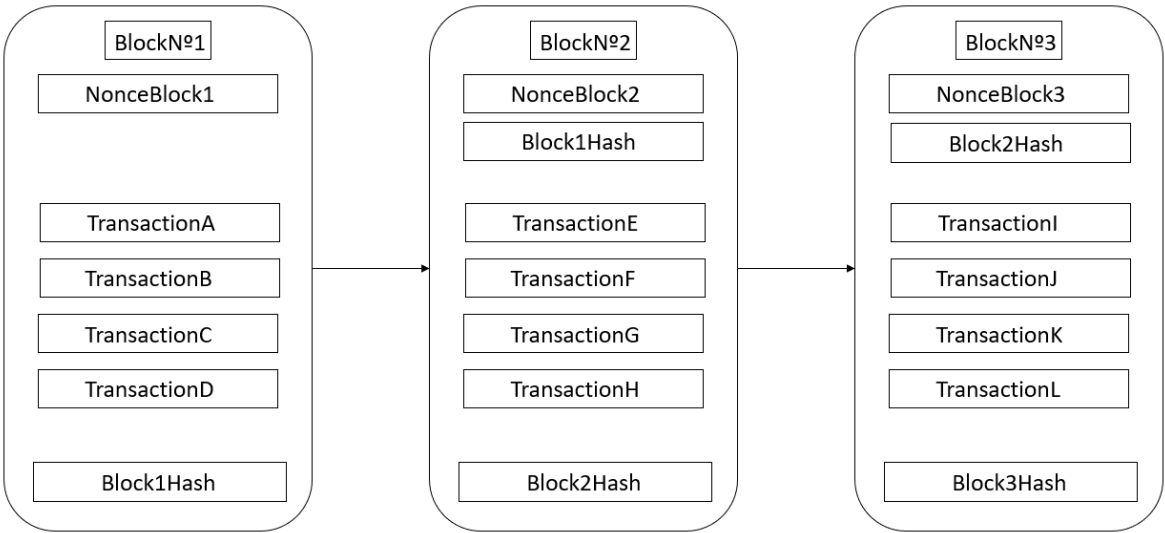


Figure 1. Blockchain structure.

The information stored in the blockchain refers to transactions done by network users. The hash consists of encrypting all the information stored on the blocks. The hash of the previous block links one block to another, thus ensuring the integrity of the entire chain. The block number corresponds to the identification of that block in the blockchain. Finally, the nonce has as objective to be a unique number for the hash function, that is, to be able to generate the hash of a block, the miners, responsible for the creation of the blocks, must hit the nonce value to obtain the correct hash value.

With the popularization of blockchain technology, developers began to create their blockchains, each with its purpose and particular characteristics. One of the most innovative blockchains developed is the Ethereum platform, also known as the Ethereum network.

Ethereum is a platform able to execute smart contracts (SC) and store them in a blockchain. Contracts executed on the Ethereum platform are immutable and work precisely as programmed, without any possibility of changing the code after it is created and stored in the blockchain. Ethereum was established in 2014 by Vitalik Buterin through a crowdfunding project [20].

Figure 2 illustrates the execution of smart contracts in the Ethereum platform. Considering the scenario of a person that intends to finance a project, the rule of the contract is that the payment is released to the project developer only when the project is done and working. If this condition is not satisfied, the contractor receives the money back. As seen in Figure 2, SC can ensure that both parties are satisfied in each case of contract operation.

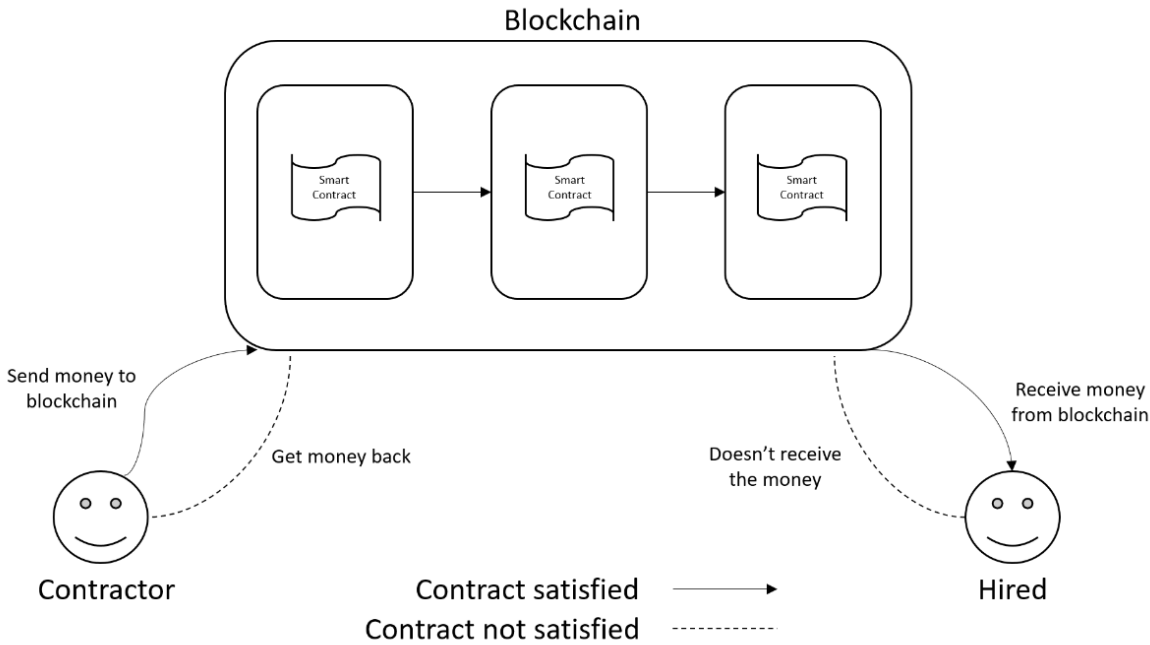


Figure 2. Smart contract operation.

The consensus algorithm used to mine new blocks in the Ethereum blockchain is known as EtHash Proof-of-Work (PoW). This algorithm was proposed to solve the ASIC mining problem of the bitcoin consensus algorithm [21,22]. Although, it is intended to migrate the consensus algorithm of the Ethereum network to the Proof-of-Stake (PoS). In the PoS algorithm, a random number generator guided by the amount of cryptocurrency that users own determines the miner of the next block [23].

To encourage network participants to mine new blocks, the Ethereum network has a cryptocurrency called Ether, denoted by the ETH pseudonym. The Ethereum white paper documents the denominations of each part of Ethereum cryptocurrency [24].

Different references show the use of the Ethereum platform for the development of applications in different areas. In [25,26], the authors developed implementations to ensure privacy in IoT

environments. In [27], the authors developed a blockchain architecture to prevent fake news in social media. In [28], the authors used the Ethereum platform to optimize agricultural services. In [29], the authors described a blockchain architecture for charging electric vehicles. In [30] shows the use of the Ethereum network for Healthcare systems.

With the development of new applications due to platform popularity, the Ethereum network registered in January 2018 a peak of approximately 1.25 million transactions made in its blockchain [31]. It is necessary to mention that a value called gas limit determines the limit of transactions per block in the Ethereum platform. The gas limit value is defined by the users who mine new blocks in the Ethereum network. Considering that the Ethereum network has been overgrowing due to its popularity, scalability issues also appear with the development of new applications. The processing of transactions that operate on the Ethereum platform can be affected due to a large number of transactions processed at the same time. To solve the scalability problem, it is necessary to use blockchains that work in parallel with each other. This concept is known as a sidechain.

2.3. Sidechain

Sidechain is a type of blockchain that validates data from other blockchains. This technology has been developed to avoid fragmentation of existing markets. Since the creation of Bitcoin, different blockchains have been created fragmenting the market. Sidechains allow integration between blockchains, without modifying basic scripts of existing blockchains [32].

According to [33], sidechains must satisfy the following requirements: cryptocurrency moved between sidechains must be able to be retrieved by whomever the owner is. Transfers must be binary (e.g., happen or not happen; there should not be failures that create cryptocurrency fragmentation). If there is a bug in a sidechain, this bug cannot interfere in another sidechain. The sidechains must be independent. Finally, users should not need to find sidechains that they are not actively using. Figure 3 illustrates the situation described in this paragraph.

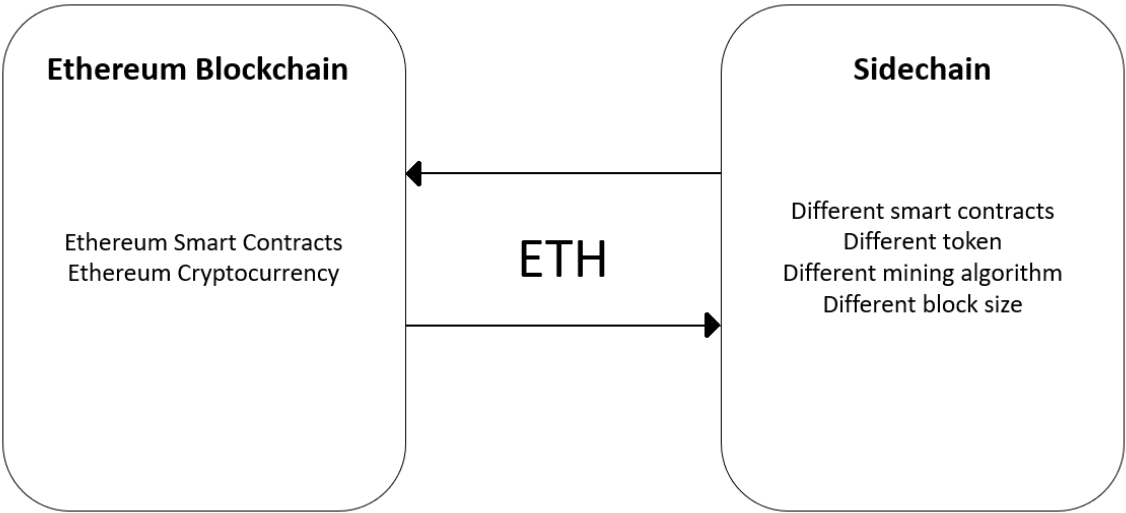


Figure 3. Sidechain concept.

The decentralized validation process, known as pegging, allows cryptocurrencies to be imported from a blockchain and returned to other blockchains. Pegging is a symmetric validation process, to transfer cryptocurrency from a blockchain to a sidechain, it must be sent to an output address in the main blockchain and can be unlocked through a sidechain work test. The pegging can also be asymmetrical. In this process, the sidechain users are miners of the main blockchain, the transfers between the main blockchain and the sidechain do not require a PoW method [32].



Another type of validation for transactions between blockchains and sidechains is using a method called federation. The federation is an intermediate layer between blockchains and sidechains whose function is to manage the users' cryptocurrency transactions. The owner of the sidechain chooses the members of the federation.

Regarding security, each sidechain is responsible for securing its network. In case a failure happens that compromises the security of the sidechain, this failure must not affect the main blockchain. However, if a security flaw compromises the main blockchain, sidechain still works, but the pegging method loses its value.

Sidechain is a new technology, but it offers numerous advantages in the development of decentralized applications. The interaction between different cryptocurrencies happens through the sidechain architecture. Also, when a sidechain exists for a determined purpose of operation, it is not necessary to create another sidechain with the same functionality. Finally, sidechains help to improve the scalability of the system since transactions made in sidechains can be processed independently of the main blockchain.

### 3. Related Work

This section presents the related works listed. State-of-the-art solutions are presented to support the architecture proposed in this article.

Guan et al. [34] propose an architecture that divides users into groups, and each group has its private blockchain, and each user is associated with a pseudonym to disguise their identity. The authors use the bloom filter to validate aliases and check for fake users. This way, an attacker would only know the sum of the group's electricity consumption, without knowing each user's data. Using aliases and encryption ensures the privacy of network users. However, the authors do not specify the platform and communication protocol used.

Gür et al. [35] describe a blockchain-based system for metering and billing with privacy protection. According to the authors, the decentralized blockchain architecture and cryptographic algorithms ensure data privacy and security of the network. The authors used the Hyperledger platform because it has an architecture that allows blockchain construction in a modular way. To simulate the smart meters, they used Raspberry Pi 3, generating random measurements. In the proposed architecture, the authors manage privacy by keeping data on devices, being shared only when necessary, and using encryption in communication, although the authors do not address the communication protocol or the cryptographic algorithm used.

Gai et al. [36] proposed an architecture for solar panels. The authors used blockchain to ensure reliability in energy trade among users. To avoid storage vulnerabilities, the authors implemented a distributed ledger. To ensure privacy, they used a method of account creation based on the user's energy use. The proposed implementation uses the Hyperledger platform. A private blockchain was developed, and comparisons were made with a public blockchain. The authors not specified the use of communication protocols.

Li et al. [37] proposed a blockchain architecture for managing transactions in microgrids. The authors no addressed User-to-user transactions, so it is not possible to analyze each user's data separately. The blockchain is a distribution operator between microgrids. The authors only specified that a private Ethereum blockchain developed in Go language was used, not addressing data privacy concepts or communication protocols involved.

Niu and Zhang [38] describe a blockchain system for power distribution networks. The authors applied data compression with the blockchain DPOS algorithm for storage on resource-limited nodes, and these nodes can verify transactions by themselves. This proposal improves the efficiency of block generation and validation in the network. The work does not address the privacy or communication protocol used. The authors used a private blockchain but not specified which. However, data compression has proven to be an effective method for storing the blockchain on devices with limited storage capacities.

Vashista and Barbhuiya [39] show a blockchain architecture where each machine and user has its address for direct communication with services. The system was implemented on the Ethereum platform using the Solidity language. The authors used the IPFS distributed data storage protocol. The IPFS stores the billing document, and the blockchain stores the IPFS hash. The authors did not show why using the Ethereum platform, but report that the tests were developed on a private network. The work superficially addresses privacy, only storing bill documents through IPFS. The authors do not describe which communication protocol is used.

Li et al. [40] proposed a layered architecture of grid, edge, and cloud. Each layer performs data analysis for quick response to users. The blockchain is above these layers using dynamic pricing and smart contracts to maintain Smart Grid stability and allowing users to monitor details of their production and power consumption. The authors chose the Ethereum platform because it has more freedom to work with smart contracts and has stability. They used a private network for prototyping, but they do not address data privacy, and communication protocols used in the blockchain.

Table 1 shows a comparison synthesis of related works. The table is divided into eight columns, as described below:

1. Reference;
2. Year;
3. Address privacy;
4. Blockchain type;
5. Blockchain used in implementation;
6. Communication protocol.

**Table 1.** Related Work

Work	Year	Privacy	Blockchain Type	Platform Used	Communication Protocol
Guan et al. [34]	2018	Yes	Private	NS	NS
Gür et al. [35]	2019	Yes	Private	Hyperledger	NS
Gai et al. [36]	2019	Yes	Private	Hyperledger	NS
Li et al. [37]	2019	No	Private	Ethereum	NS
Niu and Zhang et al. [38]	2019	No	Private	NS	NS
Vashista and Barbhuiya [39]	2019	Yes	Private	Ethereum	NS
Li et al. [40]	2019	No	Private	Ethereum	NS
Our work	2020	Yes	Sidechain	Ethereum	OSGP

NS: Not Specified

The works of Li [37], Vasistha [39] and Li [40] used the Ethereum platform.. Considering this, the use of Ethereum blockchain is most appropriate due to the integration of smart contracts and network robustness. It is not necessary to develop a specific mainchain for the SG scenario, considering that sidechains can solve the problem presented. Considering the use of private blockchains, all related works used private blockchains. The problem regarding the use of private blockchains is that only users who have access authorization can join in the blockchain. In this way, scalable solutions become infeasible for SG scenarios. Four related works address the privacy issue. In [34], different blockchains and pseudonymization are used to guarantee privacy through a group of users. In [? ], the authors stored the data on the device itself, and for sharing, they used encrypted communication. None of the related work presented shows in detail the use of communication protocols and how this can change the operation of blockchain architecture for SG scenarios. Our work uses the OSGP protocol to guarantee the generalization of the developed application. A sidechain was also used to enable the scalability of the system. We also consider that privacy is an essential point in SG applications, so our solution guarantees different privacy policies according to user preferences.



#### 4. Methodology and Architecture

This research has as objective to build a possible response or solution to a problem. In this case, the problem addressed is to make an efficient SG system using blockchains. To define the ideal solution to the problem, we compared diverse state-of-the-art solutions.

As observed in the state-of-the-art, we noticed that there is a research gap regarding the use of sidechains in SG systems. There is no definitive answer to this problem concerning privacy, scalability, and universality of existing solutions.

Our solution is a sidechain architecture that is built up of three different blockchains named BlockPRI, BlockSEC, and BlockTST. BlockPRI consists of storing each user's privacy preferences. BlockSEC stores users' data. Finally, BlockTST manages and validates information regarding the energy trade between consumers/prosumers and consumers/companies.

We developed an architecture using the three blockchains to validate the proposed idea. The Loom Network, a sidechain testing platform, was used to develop each blockchain. Three layers identify the proposed architecture to abstract each part of it. Table 2 shows the acronyms used to identify each device shown in the architecture.

**Table 2.** Acronym Definition

Acronym	Definition
PPC	Privacy Preference Contract
BL	Blockchain Layer
PL	Protocol Layer
UL	User Layer
EC	Electric Company
CP	Consumer/Prosumer
CC	Consumer/Company
UI	User Interface
ET	Energy Trade
EC	Electrical Company

Our architecture ensures system privacy, security, and reliability through the use of three distinct blockchains, one for each feature mentioned. This choice was made considering that it is easier to handle each of the requirements using individual blockchains. However, to perform the communication between the blockchains, a federation is used to establish the connection.

Three layers divide our architecture, and each layer has a specific function on each system requirement. The first layer is called the user layer. In this layer, the information is obtained and presented to users. The second layer is the protocol layer, whose function is to define the data according to the OSGP protocol. The third layer is called the blockchain layer. This layer manages the data obtained at the protocol layer. Figure 4 illustrates the described model.

The UL registers users in the blockchain through a UI. When a user wants to join the system, the user registers his SM through the UI (if the SM is not compatible, it is not possible to register). The UI is also used to make interactions with the blockchain, such as changing privacy preferences, requesting ET, verifying SM information in real-time, among others.

The PL uses the OSGP protocol to obtain and model the data package. According to [41], three standards defines the OSGP protocol. ETSI TS 103 98 rules the physical layer, ISO/IEC 14908.1 defines the network layer, and finally, ETSI GS OSG 001 standard corresponds to the application layer. In our architecture, we used only the ETSI GS OSG 001 standard. This standard was used to define the data packet stored in the security blockchain. Considering this, applications running the ISO/IEC 14908.1 standard have compatibility to use our architecture. It is necessary to mention that the OSGP protocol is compatible with other SG protocols (i.e., G3-PLC and DLMS/COSEM), thus enabling the use of this architecture in other protocols as well.

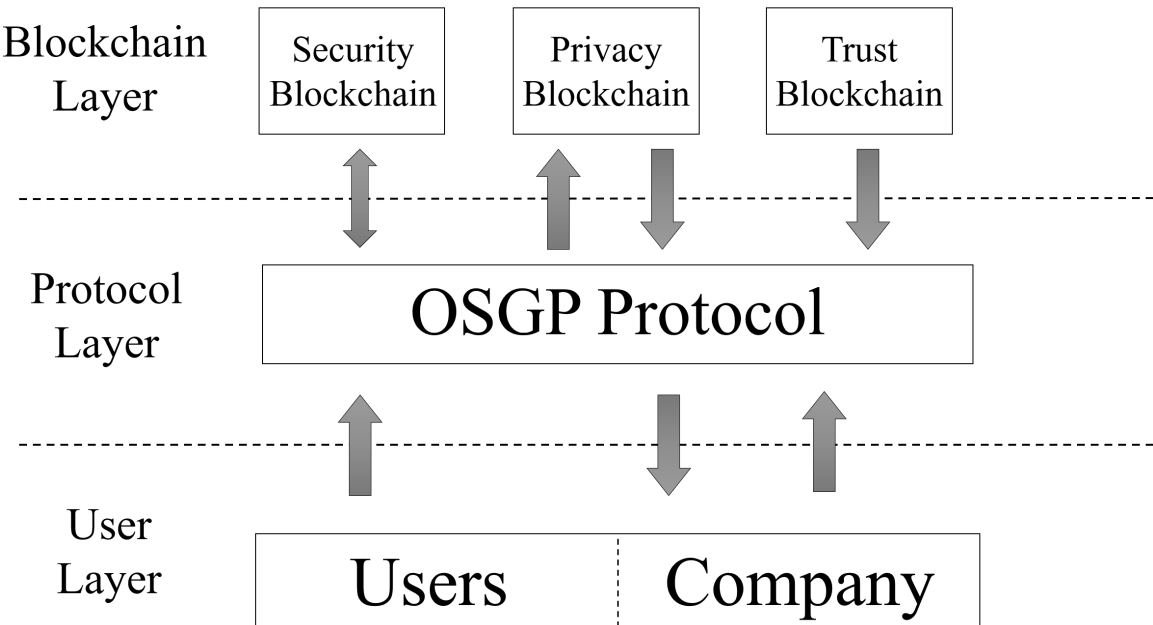


Figure 4. Proposed Arhchitecture.

The BL is responsible for ensuring privacy, security, and trust in the system. BlockPRI stores smart contracts with the privacy preferences of each user; these preferences are defined by the user when registering on the network. Considering that SC is immutable when a user registers a contract in the network with his privacy preferences, this same contract can be changed only by the user who created it. The privacy SC it is called PPC. Figure A1 illustrates the code used for the situation described.

As can be seen in Figure A1, the setMonitorTransfPref() and setMonitorConsPref() functions have a bool value to define the privacy preference for monitoring ET and consumption data. The getConsumer() function returns the consumer’s address, and this should be done to verify that the person who is trying to change the SC is the owner of the same. The getConsumerPrefs() function returns the user or concessionaire the privacy preferences of a user. Finally, the consumerRegister() function registers the user in the blockchain on the first access.

BlockSEC is responsible for storing user information. The blockchain stores data about energy usage, transactions, and other information. Whenever the EC needs to store a user’s information, it must connect to BlockPRI and verify if the monitored user allows that the EC can store such data (i.e., the user can define that the EC stores data of energy usage but not of transactions made with other users). However, BlockSEC only deals with the storage issue, so the EC can monitor the data in real-time to have control over the network even if it does not store it. Figure A2 illustrates the SC used in BlockSEC.

As described in Figure A2, the setAddress() function checks if the preferences stored in the PPC allow monitoring the user. If the user has enabled the option to be monitored, the setTransLog(), setEnergyUsage() and setCEnergyUsage() functions store the monitored data in BlockSEC. The getTransLog(), getEnergyUsage(), and getCEnergyUsage() functions retrieve the data stored in the blockchain.

BlockTST validates transactions between users. This blockchain uses smart contracts to ensure payments through the token created. The SC confirms that the token is paid only at the time the power trade is confirmed. Energy commerce can be made between CP and CC, thus allowing a dynamic and efficient system of energy trade. Figure A3 illustrates the operation of BlockTST SC.

Figure A3 illustrates the contract of ET. The functions buyEnergy() and sellEnergy() are responsible for the process of buying and selling electricity, linking buyers and sellers through the parameters of

each function. The setStorageAddress() function sends the data to the storage contract to store the transaction information on BlockSEC.

Figure 5 illustrates the operation of the proposed architecture in different situations. The illustrated scenario consists of the integration of different SG zones and the connection of different environments through the proposed architecture. To exemplify the use of the architecture, different application scenarios are used, such as privacy preference register (i), energy trade (ii), and monitoring (iii).

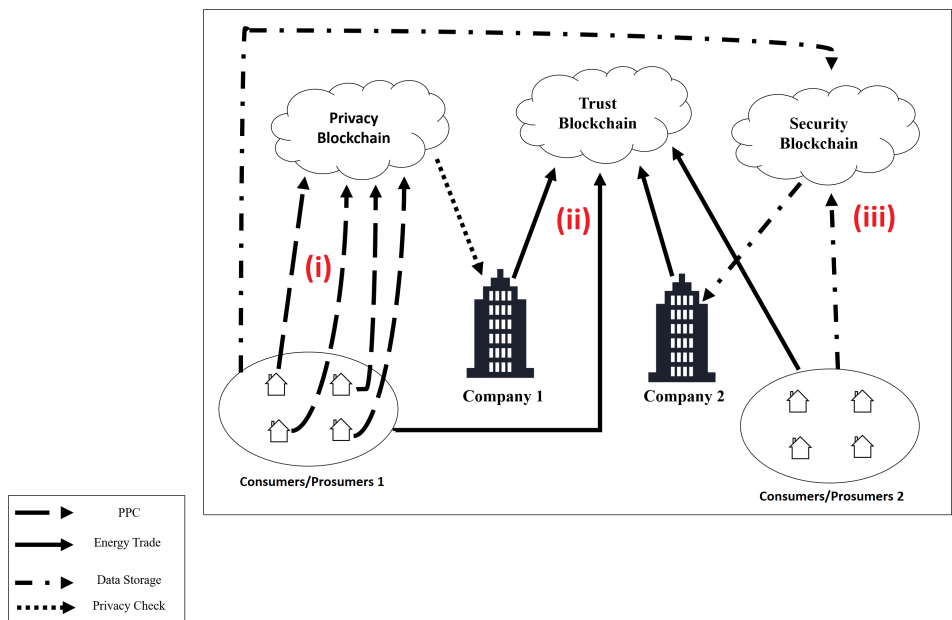


Figure 5. Application Scenario.

In situation (i), a user registers his SM in the system through an UI. The systems registers a wallet address for each SM. At this moment, the user sets his privacy preferences. The privacy preference information is registered on the PPC and stored in BlockPRI. Only the contract owner can change privacy preferences defined. Figure 6 illustrates this process.

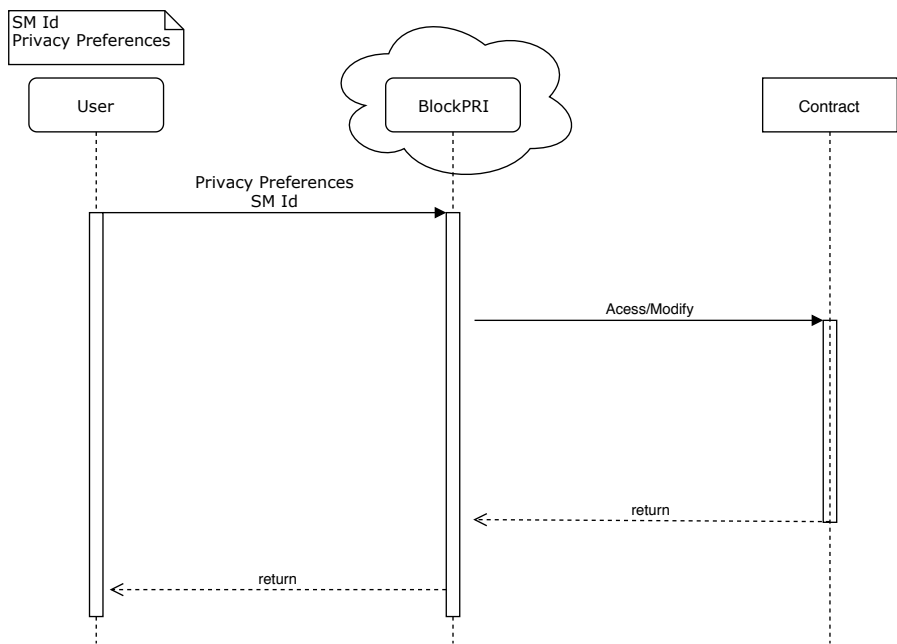


Figure 6. Privacy preferences registering.

In situation (ii) is illustrated the ET between CP and CC. The user who purchases the electric power sets in the UI the amount of energy to be purchased and the price to be paid. When the ET happens, the SM sends a confirmation signal to BlockTST through the UI, so the SC can validate the trade process and give the tokens to the power seller. The same process occurs for ET between the electrical company and the user. Figure 7 illustrates the described situation.

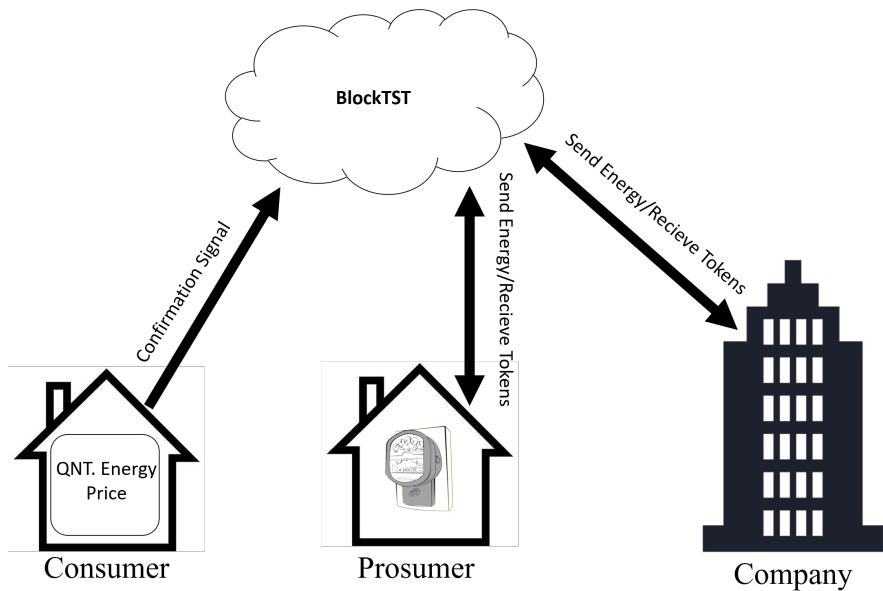


Figure 7. Energy trade situation.

In the situation (iii), the EC monitors a user. When this happens, the company connects to BlockPRI and informs the address of the user who wants to monitor. If the user’s PPC allows monitoring, BlockSEC stores the monitored data in private blockchain that only EC has access. If a user suspects that it has been monitored in an unauthorized way, that user may request a court order to verify BlockSEC, considering that information cannot be deleted from the blockchain, the system is entirely auditable. Figure 8 shows in detail the described situation.

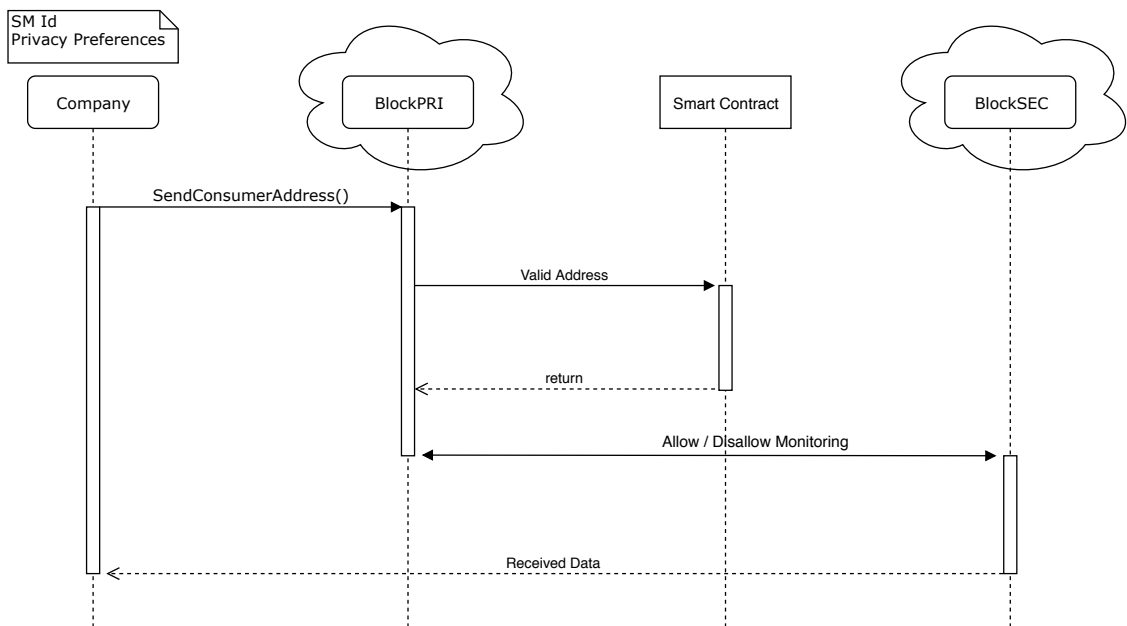


Figure 8. Energy trade situation.

To validate the proposed architecture, we performed performance, safety, and efficiency tests of the system in the described scenarios. The tests aim to prove the feasibility of the implementation of our architecture.

5. Tests and Results

We used the Loom platform to perform the tests and obtain the results. Loom is a platform focused on the development of sidechain applications. The results obtained regards to the number of transactions processed by the network, the contracts cost, the comparison of the energy price purchased by blockchain about the purchase of conventional energy, the adjust of the energy price according to the PPC, the system response to unauthorized monitoring attempts, and BlockSEC’s data structure view when monitoring a user. All of the tests were performed on a Kubuntu OS 18.04 notebook with an Intel Core i5-8625U @ 1.60 GHz processor and 8 GB of RAM.

5.1. Transaction processing time

Figure 9 shows the processing time spent on each set of transactions performed. The samples chosen for the test were 10, 50, 100, 250, 500, and 1000 transactions. The linear growth corresponds to the behavior expected within a stress test of the system, ranging from 0.8 to 94 seconds of processing time. The environment used for testing relied on immediate mining, and we discarded any additional waiting time that would interfere with the results. The data size used in this test was 144 bytes; this corresponds to the OSGP protocol data packet size.

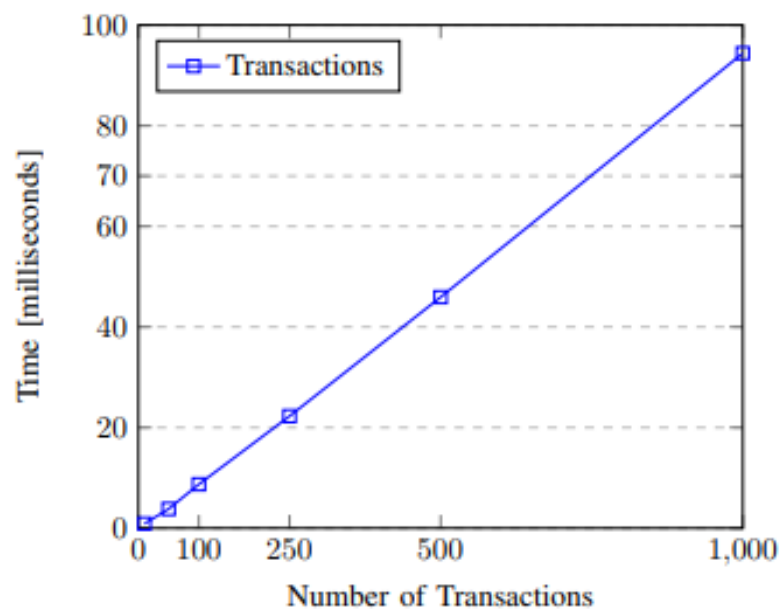


Figure 9. Transactions per seconds.

5.2. Token cost with different privacy settings

Table 3 shows four situations based on different levels of demand, generation, privacy preferences, and type of trade. The types of possible trades are between CP or CC. The amount of energy used is the same as all for ease of comparison. In Situations 1 and 3, where consumption (or demand) is high, the SGT (Smart Grid Token) values involved in the exchange tended to be higher due to the shortage of the product. Considering that the consumer allowed their monitoring was indifferent because it is a relationship of trade between consumers, neither is allowed to monitor the other, with no benefits to any party. In situations 2 and 4, however, we can observe two aspects that make the price involved

in the transaction cheaper. The first is high supply and low demand, denoting excess production. The second is the importance of monitoring when in an exchange with a dealership. In Situation 2, when dealing specifically with this aspect, there is no difference in price, since the privacy options are disabled, but in Situation 4, having the options enabled made it even possible to cheap the transaction.

**Table 3.** Token Cost in Different Situations

	Demand	Generation	Privacy Preferences	Trade Type	Energy Ammount	Price (SGT)
Situation 1	75	25	Disabled	CP	50	30
Situation 2	25	75	Disabled	CC	50	20
Situation 3	75	25	Enabled	CP	50	30
Situation 4	25	75	Enabled	CC	50	10

### 5.3. Smart Contract Cost

Table 4 shows the deploy cost of each contract developed. We used the Ropsten TestNet to evaluate all contracts. The Storage contract, which is responsible for storing information, which is the most expensive functionality in a blockchain, had the highest cost in ETH. The other contracts have a lower cost because the functions used in each of them do not have the main purpose of storing data, but access control.

**Table 4.** Deploy Cost by Contract

Contract	Cost (ETH)
Access	0.001413
Storage	0.003389
Transfer	0.001417
Token	0.001862

In Table 5, we show the relationship of a function to its contract and the cost of operation. As can be seen, the predominance of the most expensive ones is almost entirely from Storage class functions, since they are blockchain write operations (even getters, since the use of functions results in saving the corresponding get-log operation to the address). Transfer functions also had some cost, even if less excessive.

**Table 5.** Function Cost

Function	Contract	Cost (ETH)
setTranslog	Storage	0.000821
buyEnergy	Transfer	0.000536
getEnergyUsage	Storage	0.000519
getTranslog	Storage	0.000447
sellEnergy	Transfer	0.000452

### 5.4. Privacy violation test

In Figure 10 we demonstrate, by the console, the result of an attempt of unauthorized access to the data stored in the storage. First, we define an address as being an electrical company to try to access. Then, another address is registered and stored on BlockPRI, with the distributor mode disabled, as well as the settings that allow access to data stored in the blockchain. Finally, we had the distributor address try to access the historical data of the consumer who chose to protect himself. The result of the



procedure was an error coming directly from the execution of the operation, whose implementation takes care of interrupting it if the request made is not by the client's privacy options.

addr = 0x7567a0ccdc8fc4475734bb69ef051725c2640f39	<a href="#">index.html:621</a>
isDistributor = true	<a href="#">index.html:622</a>
	<a href="#">index.html:623</a>
addr = 0x432642464a7d8a848de201225b4dc5f447dbc8fa	<a href="#">index.html:624</a>
isDistributor = false	<a href="#">index.html:625</a>
canMonitorTransactions = false	<a href="#">index.html:626</a>
canMonitorEnergyConsume = false	<a href="#">index.html:627</a>
	<a href="#">index.html:628</a>
Address 0x7567a0ccdc8fc4475734bb69ef051725c2640f39 trying to access transaction historic from 0x432642464a7d8a848de201225b4dc5f447dbc8fa	<a href="#">index.html:629</a>
<div> <span>✖</span> <div> <p>► Uncaught Error: VM Exception while processing transaction: revert</p> <p>at Object.InvalidResponse (errors.js:35)</p> <p>at RequestManager.send (requestmanager.js:61)</p> <p>at Eth.send [as sendTransaction] (method.js:145)</p> <p>at SolidityFunction.sendTransaction (function.js:148)</p> <p>at SolidityFunction.execute (function.js:234)</p> <p>at index.html:632</p> </div> <div>errors.js:35</div> </div>	

**Figure 10.** Unauthorized access to storage.

### 5.5. Data structure stored in BlockSEC

Figure 11 shows one of the data structures inside the data storage blockchain. This structure is a shared vision for the concessionaires whose consumers have allowed the monitoring of their data. All transactions performed by an address are stored in the same structure, with the destination address, transaction value (in SGT), and the table encrypted and formatted within the specifications of the OSGP model. We can view the transaction history of a particular consumer, with all the security and standardization found in the current state-of-the-art.

address = 0x7567a0ccdc8fc4475734bb69ef051725c2640f39	<a href="#">index.html:623</a>
	<a href="#">index.html:624</a>
transaction[0]	<a href="#">index.html:625</a>
transDest = 0x432642464a7d8a848de201225b4dc5f447dbc8fa	<a href="#">index.html:626</a>
transValue = 3 SGT	<a href="#">index.html:627</a>
transCode = E29023AD25EA834DDB73FDE546AAB0DCE7BF	<a href="#">index.html:628</a>
	<a href="#">index.html:629</a>
transaction[1]	<a href="#">index.html:630</a>
transDest = 0x432642464a7d8a848de201225b4dc5f447dbc8fa	<a href="#">index.html:631</a>
transValue = 8 SGT	<a href="#">index.html:632</a>
transCode = CC4EB482FF6C4880465128F2A29656C95B1D	<a href="#">index.html:633</a>
>	

**Figure 11.** Stored Contents.

## 6. Conclusion

Our work presented a blockchain architecture for SG systems using sidechains. Our architecture ensures privacy, security, and trust in the system through the use of three distinct blockchains. Our architecture also guarantees universality through the use of the OSGP protocol, since it ensures a universal architecture for SG environments.

For transaction processing time tests, the results illustrate a linear growth. The processing time was adequate for the current scenario, considering that 1000 transactions were processed in approximately 94 milliseconds. For the SGT Token created, we realized that the smart contract architecture provided a price decrease in consumer-company transactions for users who chose to share their information. Regarding the cost of deploying the contract, the most expensive contract was the storage contract, this result was expected as storage is the most costly operation on a blockchain. The privacy test developed showed that the privacy preferences stored in smart contracts are met; this has been proven through an interaction trying to access data not allowed by the contract preferences. Finally, concerning the data structure stored at BlockSEC, it can be observed that the stored data illustrates the information required for the utility when a user allows its monitoring.

With the tests developed, we realized that the proposed architecture is feasible to be used in real scenarios. Using different blockchains for each system requirement allowed ensuring privacy, security, and trust holistically. To stimulate users to share their data with the electric company, a system of price discounts on the purchase of electricity is given to users who share their data for electrical companies. However, as the system grows, the processing time of transactions may increase and generate performance problems that must be treated in the future. We also mention that BlockSEC, which stores user data, may need large storage capacity to be able to hold all data.

The developed architecture meets the requirements presented in the introduction of this work. The developed solution presented has its differential based on state of the art using sidechains in smart grid systems. Thus, we conclude that the proposed work presents a scientific and technical contribution proposing a different approach in the use of blockchains in smart grid systems.

For future work, we suggest the application of ISO/IEC 14908.1 normative in the proposed architecture. We also suggest applied studies on the issue of transaction processing time and data storage in BlockSEC, seeking to optimize these functionalities to ensure a completely efficient system.

**Author Contributions:** Conceptualization, I.S.O. and V.R.Q.L.; Investigation, I.S.O., L.A.S and G.d.M ; Methodology, I.S.O.; Project Administration, I.S.O. and V.R.Q.L.; Resources, V.R.Q.L., J.F.d.P. and N.M.G.; Supervision, L.A.S., V.R.Q.L.; Validation, L.A.S., V.R.Q.L.; Writing – original draft, I.S.O.; Writing – review and editing, L.A.S, V.R.Q.L., N.M.G. and J.F.d.P.

**Funding:** This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001 and by Fundação de Amparo à Pesquisa de Santa Catarina—Brasil (FAPESC)—Grant 2019TR169. Supported by project PLATAFORMA DE VEÍCULOS DE TRANSPORTE DE MATERIALES Y SEGUIMIENTO AUTÓNOMO—TARGET. 463AC03, SA063G19. Project co-financed with Junta Castilla y León, Consejería de Educación and FEDER funds, including a cooperation with the project international cooperation project Control and History Management Based on the Privacy of Ubiquitous Environments—Brazil/Portugal. This work was supported by FCT project UID/EEA/50008/2013 (Este trabalho foi suportado pelo projecto FCT UID/EEA/50008/2013). The authors would also like to acknowledge the contribution of the COST Action IC1303—AAPELE—Architectures, Algorithms and Protocols for Enhanced Living Environments.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

Figure A1 illustrates the PPC. This contract is responsible for storing the privacy preferences of each user. The algorithm is described detailed in Section 4.

```
pragma solidity ^0.5.1;
pragma experimental ABIEncoderV2;
contract Access {
    struct ConsumerPref {
        bool isDistributor;
        bool canMonitorTransf;
        bool canMonitorCons;
    }
    address[] public consumerDB;
    mapping (address => ConsumerPref) public ownerToPrefs;
    function setMonitorTransfPref(bool _pref) public {
        ownerToPrefs[msg.sender].canMonitorTransf = _pref;
    }
    function setMonitorConsPref(bool _pref) public {
        ownerToPrefs[msg.sender].canMonitorCons = _pref;
    }
    function getConsumer(uint _id) public view returns(address){
        return consumerDB[_id];
    }
    function getConsumerPrefs(address _addr) public view returns(ConsumerPref memory){
        return ownerToPrefs[_addr];
    }
    function consumerRegister() public{
        consumerDB.push(msg.sender);
        ownerToPrefs[msg.sender] = ConsumerPref(false, false, false);
    }
}
```

**Figure A1.** Privacy Preference Contract.

## Appendix B

Figure A2 illustrates the Storage SC. This contract is responsible for check the preferences stored in PPC and perform an action based on these preferences. The algorithm is described detailed in Section 4.

```
pragma solidity ^0.5.1;
pragma experimental ABIEncoderV2;
import "./Access.sol";
contract Storage {
    Access access;
    struct AccessLog {
        address accessOrigin;
        uint accessTime;
        uint accessDay;
    }
    struct TransactionLog {
        address transDest;
        uint transValue;
        string transCode;
    }
    struct EnergyUsage {
        uint usageDay;
        uint usageQuant;
    }
    struct ConsumerInfo {
        AccessLog[] accessLog;
        TransactionLog[] transLog;
        EnergyUsage[] energyUsage;
        uint currentEnergyUsage;
    }
    mapping (address => ConsumerInfo) public consumerInfos;
    function setAddress(address _addr) public{
        access = Access(_addr);
    }
    function setTransLog(address _transDest, uint _transValue, string memory _transCode) public{
        consumerInfos[msg.sender].transLog.push(TransactionLog(_transDest, _transValue, _transCode));
    }
    function getTransLog(address _consumerAddr) public returns (TransactionLog[] memory){
        require(access.getConsumerPrefs(msg.sender).isDistributor == true);
        require(access.getConsumerPrefs(_consumerAddr).canMonitorTransf == true);
        consumerInfos[_consumerAddr].accessLog.push(AccessLog(msg.sender, 0, 0));
        return consumerInfos[_consumerAddr].transLog;
    }
    function getEnergyUsage(address _consumerAddr) public returns (EnergyUsage[] memory){
        require(access.getConsumerPrefs(msg.sender).isDistributor == true);
        require(access.getConsumerPrefs(_consumerAddr).canMonitorCons == true);
        consumerInfos[_consumerAddr].accessLog.push(AccessLog(msg.sender, 0, 0));
        return consumerInfos[_consumerAddr].energyUsage;
    }
    function getCurrentEnergyUsage(address _consumerAddr) public returns (uint){
        require(access.getConsumerPrefs(msg.sender).isDistributor == true);
        consumerInfos[_consumerAddr].accessLog.push(AccessLog(msg.sender, 0, 0));
        return consumerInfos[_consumerAddr].currentEnergyUsage;
    }
}
```

Figure A2. Storage Contract.

## Appendix C

Figure A3 illustrates the Energy Transfer SC. This SC is responsible for validating the energy trade between users. The algorithm is described detailed in Section 4.

```
pragma solidity ^0.5.1;
pragma experimental ABIEncoderV2;
import './Token.sol';
import './Storage.sol';
contract Transfer{
    Token token;
    Storage stor;
    struct EnergyOffer{
        uint energyQuant;
        uint energyValue;
    }
    mapping (address => EnergyOffer) public energyOffers;
    function buyEnergy(address _to, uint _quant) public{
        uint totalCost = energyOffers[_to].energyValue * _quant;
        require(energyOffers[_to].energyQuant >= _quant);
        energyOffers[_to].energyQuant -= _quant;
        token.transfer(_to, totalCost);
        stor.setTransLog(_to, totalCost, "003F00340000000006F52F5481599DF7BCF192C236");
    }
    function sellEnergy(uint _energyQuant, uint _energyValue) public{
        energyOffers[msg.sender].energyQuant = _energyQuant;
        energyOffers[msg.sender].energyValue = _energyValue;
    }
    function setTokenAddress(address _addr) public {
        token = Token(_addr);
    }
    function setStorageAddress(address _addr) public {
        stor = Storage(_addr);
    }
}
```

**Figure A3.** Energy transfer Contract.

## References

1. Amin, S.M.; Wollenberg, B.F. Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine* **2005**, *3*, 34–41. doi:10.1109/MPAE.2005.1507024.
2. Simmhan, Y.; Kumbhare, A.G.; Cao, B.; Prasanna, V. An Analysis of Security and Privacy Issues in Smart Grid Software Architectures on Clouds. 2011 IEEE 4th International Conference on Cloud Computing, 2011, pp. 582–589. doi:10.1109/CLOUD.2011.107.
3. Fan, Z.; Kulkarni, P.; Gormus, S.; Efthymiou, C.; Kalogridis, G.; Sooriyabandara, M.; Zhu, Z.; Lambbotharan, S.; Chin, W.H. Smart Grid Communications: Overview of Research Challenges, Solutions, and Standardization Activities. *IEEE Communications Surveys Tutorials* **2013**, *15*, 21–38. doi:10.1109/SURV.2011.122211.00021.
4. Wadhawan, Y.; AlMajali, A.; Neuman, C. A Comprehensive Analysis of Smart Grid Systems against Cyber-Physical Attacks. *Electronics* **2018**, *7*. doi:10.3390/electronics7100249.
5. Certsi. Security guide for Industrial Protocols Smart Grid.
6. Alliance, O. OSGP Protocol.
7. Kursawe, K.; Peters, C. Structural Weaknesses in the Open Smart Grid Protocol. 2015 10th International Conference on Availability, Reliability and Security, 2015, pp. 1–10. doi:10.1109/ARES.2015.67.
8. Nakamoto, S. Bitcoin. <https://bitcoin.org/bitcoin.pdf>, 2008. Accessed: 2018-10-18.
9. Song, D. Privacy-first Cloud Computing on Blockchain; USENIX Association: Burlingame, CA, 2019.
10. Shae, Z.; Tsai, J.J.P. On the Design of a Blockchain Platform for Clinical Trial and Precision Medicine. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 1972–1980. doi:10.1109/ICDCS.2017.61.
11. Samaniego, M.; Deters, R. Blockchain as a Service for IoT. 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016, pp. 433–436. doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2016.102.
12. Teslya, N.; Ryabchikov, I. Blockchain-based platform architecture for industrial IoT. 2017 21st Conference of Open Innovations Association (FRUCT), 2017, pp. 321–329. doi:10.23919/FRUCT.2017.8250199.
13. Heirman, D. What makes Smart Grid — Smart — And who is in the “game”? *IEEE Electromagnetic Compatibility Magazine* **2012**, *1*, 95–99. doi:10.1109/MEMC.2012.6244981.
14. Hashmi, M.; Hänninen, S.; Mäki, K. Survey of smart grid concepts, architectures, and technological demonstrations worldwide. 2011 IEEE PES CONFERENCE ON INNOVATIVE SMART GRID TECHNOLOGIES LATIN AMERICA (ISGT LA), 2011, pp. 1–7. doi:10.1109/ISGT-LA.2011.6083192.
15. Rubio, J.E.; Alcaraz, C.; Lopez, J. Recommender system for privacy-preserving solutions in smart metering. *Pervasive and Mobile Computing* **2017**, *41*, 205 – 218. doi:https://doi.org/10.1016/j.pmcj.2017.03.008.
16. Reinhardt, A.; Englert, F.; Christin, D. Averting the privacy risks of smart metering by local data preprocessing. *Pervasive and Mobile Computing* **2015**, *16*, 171 – 183. doi:https://doi.org/10.1016/j.pmcj.2014.10.002.
17. Mármol, F.G.; Sorge, C.; Ugus, O.; Pérez, G.M. Do not snoop my habits: preserving privacy in the smart grid. *IEEE Communications Magazine* **2012**, *50*, 166–172. doi:10.1109/MCOM.2012.6194398.
18. Efthymiou, C.; Kalogridis, G. Smart Grid Privacy via Anonymization of Smart Metering Data. 2010 First IEEE International Conference on Smart Grid Communications, 2010, pp. 238–243. doi:10.1109/SMARTGRID.2010.5622050.
19. McDaniel, P.; McLaughlin, S. Security and Privacy Challenges in the Smart Grid. *IEEE Security Privacy* **2009**, *7*, 75–77. doi:10.1109/MSP.2009.76.
20. Fundation, E. Ethereum Whitepaper. <https://github.com/ethereum/wiki/wiki/White-Paper>, 2018. Accessed: 2018-10-25.
21. Dwork, C.; Naor, M. Pricing via Processing or Combatting Junk Mail. Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology; Springer-Verlag: London, UK, UK, 1993; CRYPTO '92, pp. 139–147.
22. Dinh, T.T.A.; Liu, R.; Zhang, M.; Chen, G.; Ooi, B.C.; Wang, J. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering* **2018**, *30*, 1366–1385. doi:10.1109/TKDE.2017.2781227.



23. Lisk. Proof of Stake (PoS). <https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/proof-of-stake>, 2018. Accessed: 2018-10-25.
24. Wood, D.G. Ethereum: a Secure Decentralised Generalised Transaction Ledger. 2014.
25. Sestrem Ochôa, I.; Silva, L.A.; de Mello, G.; Alves da Silva, B.; de Paz, J.F.; Villarrubia González, G.; Garcia, N.M.; Reis Quietinho Leithardt, V. PRICHAIN: A Partially Decentralized Implementation of UbiPri Middleware Using Blockchain. *Sensors* **2019**, *19*. doi:10.3390/s19204483.
26. Ochôa, I.; Calbusch, L.; K.Viecelli.; De Paz, J.; Leithardt, V.; Zeferino, C. Privacy in the Internet of Things: A Study to Protect User's Data in LPR Systems Using Blockchain. 2019 17th Annual Conference on Privacy, Security and Trust (PST), 2019.
27. Ochôa, I.S.; de Mello, G.; Silva, L.A.; Gomes, A.J.P.; Fernandes, A.M.R.; Leithardt, V.R.Q. FakeChain: A Blockchain Architecture to Ensure Trust in Social Media Networks. *Quality of Information and Communications Technology*; Piattini, M.; Rupino da Cunha, P.; García Rodríguez de Guzmán, I.; Pérez-Castillo, R., Eds.; Springer International Publishing: Cham, 2019; pp. 105–118. doi:10.1007/978-3-030-29238-6\_8.
28. Shih, D.H.; Lu, K.C.; Shih, Y.T.; Shih, P.Y. A Simulated Organic Vegetable Production and Marketing Environment by Using Ethereum. *Electronics* **2019**, *8*. doi:10.3390/electronics8111341.
29. Jin, R.; Zhang, X.; Wang, Z.; Sun, W.; Yang, X.; Shi, Z. Blockchain-Enabled Charging Right Trading Among EV Charging Stations. *Energies* **2019**, *12*. doi:10.3390/en12203922.
30. Dwivedi, A.D.; Srivastava, G.; Dhar, S.; Singh, R. A Decentralized Privacy-Preserving Healthcare Blockchain for IoT. *Sensors* **2019**, *19*. doi:10.3390/s19020326.
31. Etherscan. Ethereum Transaction Chart. <https://etherscan.io/chart/tx>, 2018. Accessed: 2018-10-28.
32. Ray, S. What are Sidechains? <https://hackernoon.com/what-are-sidechains-1c45ea2daf3>, 2018. Accessed: 2018-10-30.
33. Back, S.A.; Corallo, M.; Dashjr, L.; Friedenbach, M.; Maxwell, G.; Miller, A.; Poelstra, A.; Timón, J. Enabling Blockchain Innovations with Pegged. 2014.
34. Guan, Z.; Si, G.; Zhang, X.; Wu, L.; Guizani, N.; Du, X.; Ma, Y. Privacy-Preserving and Efficient Aggregation Based on Blockchain for Power Grid Communications in Smart Communities. *IEEE Communications Magazine* **2018**, *56*, 82–88. doi:10.1109/MCOM.2018.1700401.
35. Gür, A.; Öksüzler, ; Karaarslan, E. Blockchain Based Metering and Billing System Proposal with Privacy Protection for the Electric Network. 2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG), 2019, pp. 204–208. doi:10.1109/SGCF.2019.8782375.
36. Gai, K.; Wu, Y.; Zhu, L.; Qiu, M.; Shen, M. Privacy-Preserving Energy Trading Using Consortium Blockchain in Smart Grid. *IEEE Transactions on Industrial Informatics* **2019**, *15*, 3548–3558. doi:10.1109/TII.2019.2893433.
37. Li, Z.; Bahramirad, S.; Paaso, A.; Yan, M.; Shahidehpour, M. Blockchain for decentralized transactive energy management system in networked microgrids. *The Electricity Journal* **2019**, *32*, 58 – 72. Special Issue on Strategies for a sustainable, reliable and resilient grid, doi:https://doi.org/10.1016/j.tej.2019.03.008.
38. Niu, D.; Zhang, D. Research and Implementation of Power Distribution Systyem Security Protection Technology Based on Blockchain. Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications; ACM: New York, NY, USA, 2019; HP3C '19, pp. 113–117. doi:10.1145/3318265.3318279.
39. Vashistha, M.; Barbhuiya, F.A. Blockchain in Smart Power Grid Infrastructure. Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure; ACM: New York, NY, USA, 2019; BSCI '19, pp. 89–96. doi:10.1145/3327960.3332388.
40. Li, Y.; Rahmani, R.; Fouassier, N.; Stenlund, P.; Ouyang, K. A Blockchain-based Architecture for Stable and Trustworthy Smart Grid. *Procedia Computer Science* **2019**, *155*, 410 – 416. The 16th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2019), The 14th International Conference on Future Networks and Communications (FNC-2019), The 9th International Conference on Sustainable Energy Information Technology, doi:https://doi.org/10.1016/j.procs.2019.08.057.
41. Ochôa, I.S.; Leithardt, V.R.Q.; Zeferino, C.A.; Silva, J.S. Data Transmission Performance Analysis with Smart Grid Protocol and Cryptography Algorithms. 2018 13th IEEE International Conference on Industry Applications (INDUSCON), 2018, pp. 482–486. doi:10.1109/INDUSCON.2018.8627195.