Performance and Cost Evaluation of Smart Contracts in Collaborative Health Care Environments*

Roben Castagna Lunardi*[†], Henry Cabral Nunes*, Vinicius da Silva Branco*, Bruno Hugentobler Lippert*,
Charles Varlei Neu*[‡] Avelino Francisco Zorzo*
*PUCRS, [†]IFRS,[‡]UNISC - Brazil
E-mail: {roben.lunardi, henry.nunes, vinicius.branco, bruno.lippert@acad.pucrs.br, charles.neu}@acad.pucrs.br,
avelino.zorzo@pucrs.br

Abstract—**Blockchain emerged as a solution for data integrity, non-repudiation, and availability in different applications. Data sensitive scenarios, such as Health Care, can also benefit from these blockchain properties. Consequently, different research proposed the adoption of blockchain in Health Care applications. However, few are discussed about incentive methods to attract new users, as well as to motivate the system or application usage by existing end-users. Also, little is discussed about performance during code execution in blockchains. In order to tackle these issues, this work presents the preliminary evaluation of TokenHealth, an application for collaborative health practice monitoring with gamification and token-based incentives. The proposed solution is implemented through smart contracts using Solidity in the Ethereum blockchain. We evaluated the performance of both in Ropsten test network and in a Private instance. The preliminary results show that the execution of smart contracts takes less than a minute for a full cycle of different smart contracts. Also, we present a discussion about costs for using a Private instance and the public Ethereum main network.**

Index Terms—**Blockchain, smart contracts, health care, health activities, performance, Ethereum.**

I. INTRODUCTION

Blockchain emerged as a promising technology after the proposition of the *Bitcoin* [1] cryptocurrency. Additionally, blockchain has also been applied on solutions to solve problems on several other scenarios, such as Domain Name System (DNS) services [2], storing and running programming code parts [3], transaction control [4], electronic voting [5] and copyright control [6]. Many of these different applications have some requirements that are fulfilled by the adoption of blockchain, such as resilience (due to the decentralized characteristic of the network), non-repudiation (by using digital signatures in transactions) and tamper-resistance. Thus, blockchain provides reliability for the data it maintains.

Especially in the context of health data, there still are some problems regarding their handling, such as data that are often not recorded properly, out of date records, and even data may not be accessible by their owners, *i.e.*, the end-users (patients) [7]. Thus, some studies suggest the use of blockchains to provide data integrity and availability [8] [9].

Although blockchain-based systems are used in different research that proposes health care monitoring solutions for digitized data sharing [10], storage [11] and access control [12], there is little discussion about solutions that motivate the adoption and usage by end users. Such solutions could be based on techniques as bonuses [13] or gamification [14], for example, and implemented using blockchains due to their ability on executing code in a distributed manner through smart contracts [14] [15].

This paper aims to present and evaluate the TokenHealth project, a collaborative health practice monitoring system based on blockchain and smart contacts. Thus, the proposed system provides security through data integrity, resilience and availability. It also implements methods that motivate end-user adoption and usage. To evaluate performance in different environments, we used both the test network (Ropsten) and a private instance of Ethereum [3], which is currently one of the most popular blockchains that implement smart contracts. We also evaluate financial costs associated to those environments.

The remaining of this paper is structured as follows. Section II presents some background. Section III discusses some related work. Sections IV and V present a case study, the proposed solution, describing its operation, details about the implementation, technologies and how smart contracts are used. Section VI presents and discusses the preliminary results. Finally, Section VII concludes this paper and indicates some future work.

II. BACKGROUND

With the popularization and success of Bitcoin, other blockchains emerged with different proposals and new technologies. Ethereum, like other blockchains that have a cryptocurrency (such as Bitcoin and Litecoin) associated, uses Proof-of-Work (PoW) as a consensus algorithm [16]. The

consensus algorithm is a mechanism used to ensure that data addition follow a pre-combined business logic. This is necessary because blockchain works in a decentralized peer-to-peer (P2P) network where the nodes are unreliable and may act maliciously. Thus, the consensus algorithm [17] ensures reliability to the data generated by any node in this untrusted environment.

Blockchain can have different characteristics depending on the purpose for the kind of application it was designed for. Zorzo *et al.* [18] proposed a layer-based model that show different solutions for communication, consensus algorithms, data management, and application layers. For example, hierarchical P2P architecture is better suited for IoT environments. One important feature regarding the application layer is the capability to support smart contracts, *i.e.*, the capability to integrate the business logic of the application into the blockchain.

The concept of smart contracts was introduced in 1994 by Nick Szabo as a *script* representing a contract that can enforce its terms automatically, reducing the need for intermediaries in the event of legal disputes [19]. Smart contracts are processed in a blockchain, being decentralized and allowing different models. It provides flexibility to process any application, providing immutability of the generated data, transparency on the operation and auditability on performed processes and transactions.

In Ethereum, each node has a virtual machine, called Ethereum Virtual Machine (EVM), which can be used to process *bytecodes* representing smart contracts. Users can make special requests to the network by calling these smart contracts, allowing them to change their state or request information about the current state. Nodes process these requests based on the smart contract bytecode in its EVM and store the resulting smart contract state in the blockchain. This whole process is the same as the process of adding standard transactions and needs to be mined and verified by the whole network [3].

III. RELATED WORK

Different research proposed methods for rewarding, solving performance issues and adoption of smart contracts in blockchains. For example, Rouhani *et al.* [20] discuss security issues and performance of smart contract execution based on different talks that intend to measure smart contracts performance. For example, some metrics are cited such as number of transactions per second, contract execution time, and block state update time. For measurement purposes, in this work, the average time to perform each contract will be adopted.

An approach to solve the mining work concentration problem by using a virtual currency service that proposes a new end-user usage incentive based on gamification instead of traditional economic incentives is presented by [21]. Experiments that show the feasibility of adopting the alternative incentive were presented with some discussion, regarding to the positive impact of psychological factors provided by gamification methods.

Parizi *et al.* [14] also discuss the gamification process in blockchain. They argue that gamification has been a trending topic to address human-centric concerns, specially in the online world, both in the industry and business and also in academic works. In this work, the authors also identified and discussed main human-related problems in decentralized blockchain systems and proposed a preliminarily gamified model, which is illustrated in the context of a typical blockchain system.

Another work that discusses bonuses on blockchain-based systems is presented by Chen *et al.* [13]. Their method is built as a new type of decentralized bonus points alliance based on key technologies of blockchain, such as consensus mechanism and smart contract in blockchain and take the "alliance blockchain". This proposal takes advantage of technical features of decentralization, trustconsensus, distributed network, collective maintenance and advanced research on BonusPoints Alliance business model based on blockchain. This model is applied to design a system that could be used as a solution for the shortcomings of traditional alliance, such as a high cost of system development, difficulty of bonus points exchange and difficulty of bonus points circulation.

A next generation repudiation system based on blockchain is proposed by Dennis *et al.* [22]. The authors first discuss current reputation systems, current security vulnerabilities and how new blockchain-based technologies are currently used. Their goal is to propose a new reputation system based on blockchain technologies to solve problems that are, according to the authors, not yet solved on current generation reputation systems. Results are presented and discussed based on simulations. Performance is evaluated and limitations of the proposed solution are indicated and explained. Finally, the authors also present suggestions to overcome current limitations and indicate future directions.

IV. CASE STUDY: TOKENHEALTH

TokenHealth[1] is a system that aims to promote health through a collaborative tool, using methods based on *tokens* and gamification (reputation system) functions. A proof of concept of a vaccination flow is implemented to validate and to evaluate the project. This proof of concept is intended to cover the entire vaccination cycle, from vaccine application, reapplication reminder, gamification and incentives.

Also, this proof of concept implements security, by providing integrity, availability and transparency through the use of a blockchain. Additionally, *Solidity*, a smart contract programming language, was used to implement the business rules in the blockchain. Thus, a generic model of operation is proposed, where users can keep their vaccines records updated and receive bonus for taking care of their health. Fig. 1 presents an overview of the operation, flow of the main system components and the interaction with the actors that are involved.
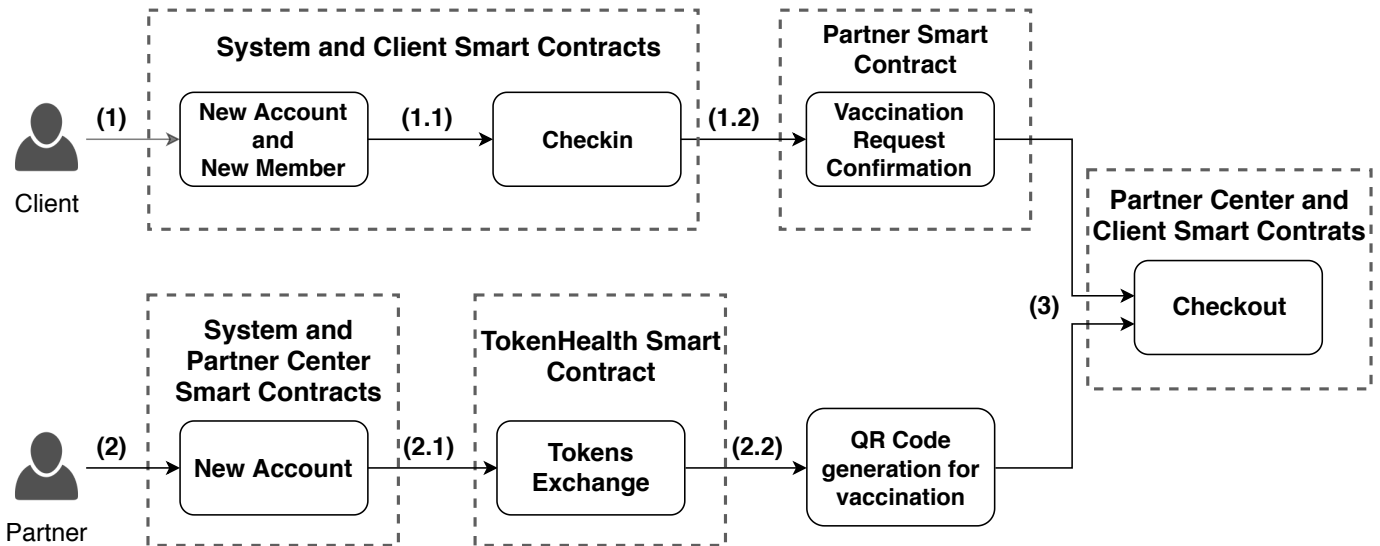
Fig. 1: Main flows for the vaccination system

The TokenHealth flow depends on two main actors: (*(i)* the client, or their dependent, wants to be vaccinated and *(ii)* a vaccination place (for example, a pharmacy, as illustrated in Fig. 1). The purpose of this flow is to connect clients and vaccination companies, encouraging the client to keep their vaccination grid up to date, and also to be rewarded with *tokens*, which can be used to receive discounts on other purchases, for example. Thus, TokenHealth can also link customers to companies, improving fidelity. To do so, the blockchain technology plays a key role, as it allows the creation of transactions that persist data according to the business rules, as well as sending *tokens*, a form of "cryptocurrency" that allows the exchange of values.

The flow starts with the customer and company registration (Flow 1 and 2, respectively, in Fig. 1). Through smart contracts, the customer then informs their personal data, registers their dependents and the vaccines that have already been received by each one. In another stream, the pharmacy registers its business data in "System and Partner" smart contract, and also informs its available vaccines that can be purchased. After that, the pharmacy can perform the purchase of *tokens* that will be transferred to customers as bonus, as illustrated on Flow 2.1 in Fig. 1. TokenHealth provides many functionalities to the client, such as to list what vaccines should be taken, to select a partner (*e.g.*, a pharmacy) that has the vaccines available, and to perform *checkin* at the partner's place to inform that a vaccine has been taken. Also, the customer can choose to pay the full amount for the vaccine and receive *tokens* or use their *tokens* to get a discount, as shown in Flow 1.1 in Fig. 1

After the *checkin* process is completed, the customer can go to the pharmacy, where a store attendant can see the customers *checkin* on the "TokenHealth Partner" system, the partner side of TokenHealth. Then, an unique QR Code is generated for the current flow, so that the customer can read and confirm the release of the vaccine in TokenHealth. Flows 2.2 and 1.2,

in Fig. 1, show this process. Once the vaccine is applied, the pharmacy can also confirm this process in "TokenHealth Partner", by using the *checkout* process. Flow 3 illustrates this step. The whole *checkout* process is completed when both customer and pharmacy inform that the vaccine process is finished. If the customer has chosen to pay the full amount of the vaccine (without spending their *tokens*), the partner must confirm it, so that the system then automatically sends rewarding *tokens* to the customer, performing the bonus process. On the other hand, if the customer has chosen to spend available *tokens* as a payment method (and/or receive a discount, for example), after the partner confirmation, the customer must read a (new) QR Code to perform the *checkout*. Thus, TokenHealth sends the *tokens* to the partner automatically, in order to complete the payment.

V. IMPLEMENTATION AND TECHNOLOGIES

To develop the proposed solution, a set of technologies was chosen. The development can be divided into four layers, as presented in Fig. 2: (*i*) developed applications, such as Client App and TH Web application (gray in the Fig. 2); (*ii*) technologies used for the application implementation, such as programming languages, libraries and Application Programming Interface (API) (in blue); (*iii*) technologies used to communicate different technologies (in red); and (*iv*) the smart contracts that implement the back end solution.

The *web* interface was implemented using JavaScript as a programming language. This language was chosen because it is one of most used to connect through WEB3 (Ethereum interface) and has integration with the chosen libraries. For the client application, we used the *Flutter* framework, as it generates applications for both *Android* and *iOS* platforms based on the same source code. Additionally, the Solidity language was chosen for the development of smart contracts, which is the main language on Ethereum. Also, the *React.js*
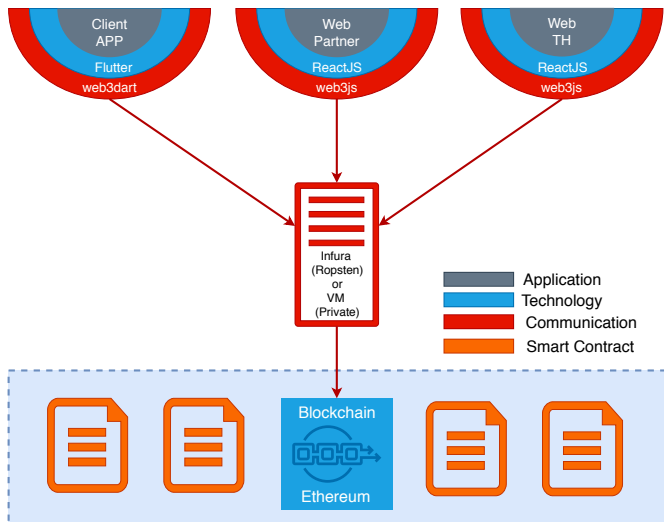
Fig. 2: Implementation in layers

libraries were used to create the *web* interface and *Web3.js* to establish connection to the blockchain.

VI. EVALUATION

In order to evaluate the performance of the proposed solution, we used the *Ropsten Test Net*, a testing blockchain environment maintained by Ethereum. This environment is available to perform tests in an emulated environment, which contains similar characteristics of the main Public Ethereum network. An important advantage on evaluating the solution in Ropsten is that no financial investments are required, as Ropsten provides a faucet to request Ethers to this testing network. Additionally, we used a private instance of Ethereum in a cloud-based environment (using Google Cloud Platform [23] virtual machines). It is important to notice that we did not performed evaluation through the main Public Ethereum network due to financial costs associated with it.

One important advantage in private instances is that the mining difficulty can be set in the genesis block. This capability can help to start the blockchain with a difficulty that have a Proof-of-Work (PoW) adjusted to the infrastructure that will be used to maintain the blockchain. It is important because it allows to set a difficulty that enables time evaluation to produce new blocks in a higher throughput than in the main Public Ethereum network.

We present a qualitative discussion in Table I about different Ethereum options that can be used by the proposed solution. In our evaluation, both Ropsten and Private instances had a mining time lower than 1 minute. Consequently, the behaviour in both was similar. However, in the main Public Ethereum network, mining time is higher than 1 minute. It happens because of the high difficulty present in main Public Ethereum due to the dynamic difficulty increase over time, especially due to high computing power of the miners. However, when a distributed Application (dApp) uses a private instance of Ethereum, there is an associated infrastructure cost that should

be considered. For example, for a small application, 5 nodes can be used to validate produced blocks. However, for larger applications, more nodes should be used to guarantee resilience and performance in the smart contracts execution.

TABLE I: Ethereum networks comparison

	Ethereum (Main)	Ropsten (testnet)	Private Instance
Mining Time	>5 minutes	<1 minute	<1 minute
Mining Difficulty	High	Medium	Settable
Financial Cost	Yes (Ethers)	No	Yes (infrastructure)

Table II shows execution costs for each smart contract to present an overview of maintenance costs of the proposed solution. First, by analyzing the main Public Ethereum network, we estimated the cost in *gas* (smart contract execution fee). The highest cost for an individual smart contract (as shown in Table II) is the cost to create a new member, corresponding to a total of 0.002718 Ethers (or US$0.453906, using the average quotation of $167 dollars per Ether on September 26, 2019 [24]). A full user costs at least 0.003136 Ethers (sum of new account and one new member costs). Although this function has the highest cost, it should only occur once per user.

The total cost for complete execution of the full vaccination cycle (checkin, confirmation and checkout) requires 0.000776 Ethers (or approximately $0.129592). It is justified by the size and few processing required by the smart contracts used in the vaccination cycle. Those values are explained in Table II.

TABLE II: Smart Contracts execution costs

	Ethereum (Main Network)	Private Instance
Create new account	0.000418 Ethers (∼US$0.069806)	-
Add a member	0.002718 Ethers (∼US$0.453906)	-
Checkin	0.000739 Ethers (∼US$0.123413)	-
Vaccination confirmation	0.000003 Ethers (∼US$0.000501)	-
Checkout	0.000034 Ethers (∼US$0.005678)	-
Full vaccination cycle	0.000776 Ethers (∼US$0.129592)	-
Infrastructure (Monthly)	-	US$123,75

Another strategy to deploy a dApp is to use a private instance of Ethereum to maintain and execute smart contracts. For example, one can instantiate using cloud services with predefined infrastructure costs. For example, when allocating 5 specific machines to run Ethereum nodes in Google Cloud Platform [23], the monthly fixed cost would be around $123.75 dollars (using 5 instances of $24.75). It is worth noting that this cost represents only the basic configuration, *i.e.*, it was not considered any elastic service or any kind of costs with maintenance and system configuration.

Some observations can be made when comparing the costs from both the main Public Ethereum network and a Private Instance. For example, Public Ethereum does not require maintenance cost and the cost for each execution is

based on the number of cycles. However, using a private instance can allow a higher number of transactions with a fixed cost. As a comparison, the main Public Ethereum network can perform almost 1,000 full vaccination cycles for the same $123.75 dollars (considering 26 September 2019 Ether exchange value [24]). Considering the purpose of the developed application, one thousand cycles are not enough. Consequently, the main Public Ethereum network has a higher cost to perform the smart contracts considering a system only for vaccination. This discussion should be exploited in a future evaluation, considering other entities of health care, such as hospitals, health insurance, gym and others.

For the preliminary performance evaluation of the developed smart contracts, we used the Ropsten testing network and a Private Instance using Google Cloud [23] with 2 processing cores, 8GB of memory and 80GB of storage. The experiments were repeated 10 times and the results of the median executions are presented. Both in the private instance and in the Ropsten test network, good response time results related to the performance were obtained. An overview of executions can be observed in Fig. 3.

We can observe, in Fig. 3, that the execution of some smart contracts have a similar performance. For example, the smart contract to create a New Account executed in 27,744.5 milliseconds (median execution time) in a private instance and in 31,560.5 milliseconds in Ropsten, *i.e.*, a difference of around 13%. However, if we consider the execution of a smart contract with few processing requirements (shorter bytecode), the difference is higher. For example, the smart contract to add a New Member was executed in 8,906 milliseconds in the private instance and in 69,533.5 milliseconds in Ropsten. Also, it is important to note that Ropsten has a similar behavior present in the main Public Ethereum network, *i.e.*, in some moments the throughput can be affected by problems such as fork resolutions or other issues. For the Full Vaccination Cycle, *i.e.*, the sum of time spent in execution of Checkin, Vaccination Confirmation and Checkout smart contracts, it took 44,273.5ms in the private instance and 69,899.5ms in the Ropsten test network. The results demonstrates the viability, considering performance, to execute the main smart contracts for a dApp for vaccination. However, it was not possible to compare with the main Public Ethereum network due to the financial costs for acquiring Ethers.

VII. Final Considerations

Health care is a recurring and important topic to the society, as several advances, new techniques, activities and medicines are constantly emerging. A plethora of systems and applications for health care and health activities monitoring are also currently available. However, it is important to create methods to promote end-users adoption and usage, in special for a collaborative approach that can help preventing diseases and health problems. Thus, this paper presents a solution for collaborative health economics systems using blockchain, exemplifying the usability of this technology in order to improve health and disease prevention through gamification and loyalty.

In addition, we presented some benefits of using blockchain in private instances or in blockchain public networks. As shown, the financial costs on the main Ethereum Public network are higher when the number of transactions is also high. However, when choosing to use private instance, the cost of infrastructure and personnel must be considered. Also, it was observed that the performance in the testing network were very similar to the values on the private instance, but no results were obtained with the main Ethereum Public network. Finally, we can concluded that blockchain can be used as an alternative to a collaborative health monitoring system, as it makes the system safe by providing data immutability, ensuring that a business logic is preserved and the possibility of gamification by completing preventive health activities.

As a next step, we intend to expand the system to include medicines, medical consultations data and other activities regarding to preventive health. Also, we intend to expand the tests and to evaluate our solution on the main Ethereum Public network. Additionally, we intend to evaluate the Smart Contracts in different blockchains, especially in blockchains with different consensus algorithms, such as Hyperledger Fabric [25] and SpeedyChain [26], [27].

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf[AccessDate:12November,2019].

[2] T. H. Chang and D. Svetinovic, "Data analysis of digital currency networks: Namecoin case study," in *21st International Conference on Engineering of Complex Computer Systems (ICECCS 2016)*, 2016, pp. 122–125.

[3] Ethereum, "A next generation smart contract and decentralized application platform," 2017. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper[AccessDate:12November,2019].

[4] X. Min, Q. Li, L. Liu, and L. Cui, "A permissioned blockchain framework for supporting instant transaction and dynamic block size," in *2016 IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 90–96.

[5] T. Moura and A. Gomes, "Blockchain voting and its effects on election transparency and voter confidence," in *18th Annual International Conference on Digital Government Research*. ACM, 2017, pp. 574–575.

[6] J. Kishigami, S. Fujimura, H. Watanabe, A. Nakadaira, and A. Akutsu, "The blockchain-based digital content distribution system," in *2015 IEEE Fifth International Conference on Big Data and Cloud Computing*, 2015, pp. 187–190.

[7] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2nd International Conference on Open and Big Data (OBD 2016)*, 2016, pp. 25–30.

[8] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom 2016)*, 2016, pp. 1–3.

[9] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11 676–11 686, 2018.

[10] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[11] L. Mertz, "(block) chain reaction: A blockchain revolution sweeps into health care, offering the possibility for a much-needed data solution," *IEEE Pulse*, vol. 9, no. 3, pp. 4–7, May 2018.

[12] N. Rifi, E. Rachkidi, N. Agoulmine, and N. C. Taher, "Towards using blockchain technology for ehealth data access management," in *4th International Conference on Advances in Biomedical Engineering (ICABME 2017)*, 2017, pp. 1–4.
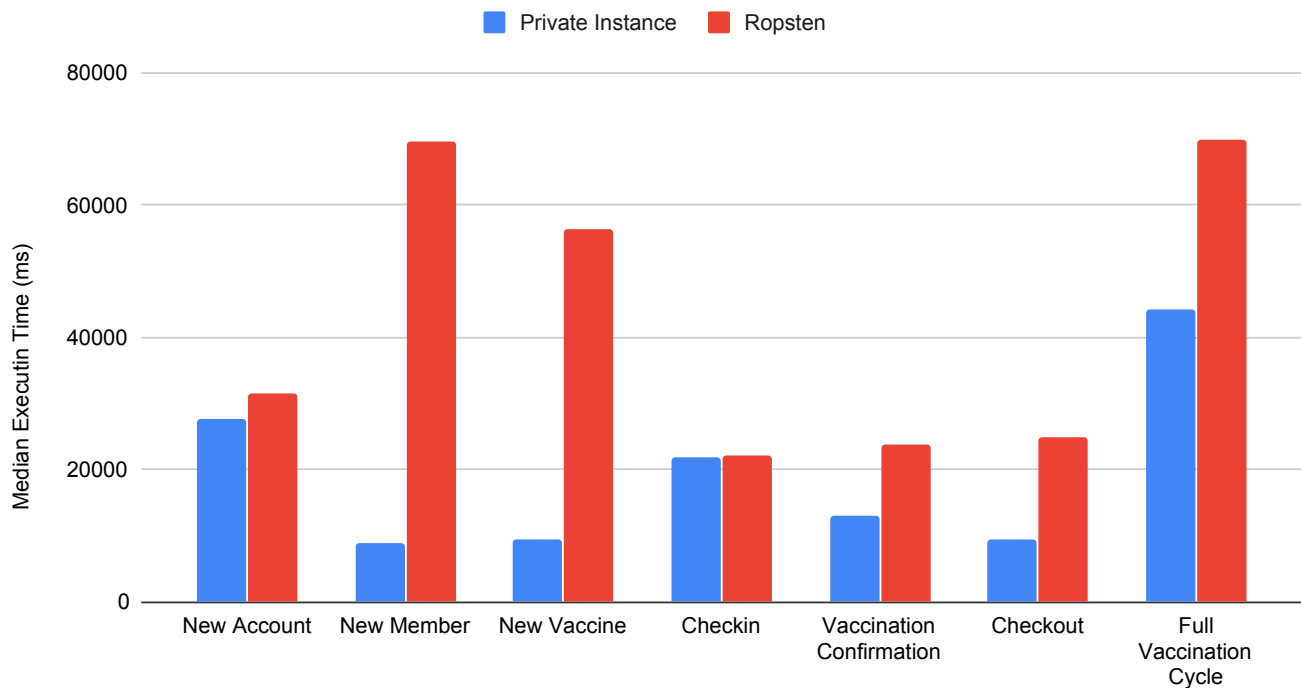
Fig. 3: Smart Contracts performance in Ropsten and Private Instance

[13] C. Chen, X. Sun, G. Lu, H. Kang, and Y. Shen, "Bonus points alliance based on the blockchain," in *14th International Conference on Semantics, Knowledge and Grids (SKG 2018)*, 2018, pp. 229–234.

[14] R. M. Parizi and A. Dehghantanha, "On the understanding of gamification in blockchain systems," in *6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW 2018)*, 2018, pp. 214–219.

[15] G. Zyskind, O. Nathan, and A. . Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, 2015, pp. 180–184.

[16] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[17] R. C. Lunardi, R. A. Michelin, C. V. Neu, A. F. Zorzo, and S. S. Kanhere, "Impact of consensus on appendable-block blockchain for iot," in *16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2019)*, 2019, pp. 1–10.

[18] A. F. Zorzo, H. C. Nunes, R. C. Lunardi, R. A. Michelin, and S. S. Kanhere, "Dependable iot using blockchain-based technology," in *8th Latin-American Symposium on Dependable Computing (LADC 2018)*, 2018, pp. 1–9.

[19] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[20] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50 759–50 779, 2019.

[21] Y. Kano and T. Nakajima, "An alternative approach to blockchain mining work for making blockchain technologies fit to ubiquitous and mobile computing environments," in *10th International Conference on Mobile Computing and Ubiquitous Network (ICMU 2017)*, 2017, pp. 1–4.

[22] R. Dennis and G. Owen, "Rep on the block: A next generation reputation system based on the blockchain," in *10th International Conference for Internet Technology and Secured Transactions (ICITST 2015)*, 2015, pp. 131–138.

[23] Google Inc., "Google cloud platform," 2019. [Online]. Available: https://cloud.google.com/[AccessDate:12November,2019].

[24] Coin Market Cap, "Cryptocurrencies by market capitalization," 2019. [Online]. Available: https://coinmarketcap.com/

[25] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[26] R. C. Lunardi, R. A. Michelin, C. V. Neu, and A. F. Zorzo, "Distributed access control on iot ledger-based architecture," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2018, pp. 1–7.

[27] R. A. Michelin, A. Dorri, M. Steger, R. C. Lunardi, S. S. Kanhere, R. Jurdak, and A. F. Zorzo, "Speedychain: A framework for decoupling data from blockchain for smart cities," in *15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2018)*, 2018, pp. 145–154.