



## Blockchain orchestration and experimentation framework: A case study of KYC

Wazen M Shbair, Mathis Steichen, Jerome Francois, Radu State

### ► To cite this version:

Wazen M Shbair, Mathis Steichen, Jerome Francois, Radu State. Blockchain orchestration and experimentation framework: A case study of KYC. IEEE/IFIP Man2Block 2018 - IEEE/IFIP Network Operations and Management Symposium, Apr 2018, Taipei, Taiwan. hal-01939865

**HAL Id: hal-01939865**

**<https://inria.hal.science/hal-01939865>**

Submitted on 29 Nov 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Blockchain Orchestration and Experimentation Framework: A Case Study of KYC

Wazen M. Shbair <sup>\*</sup>, Mathis Steichen <sup>\*</sup>, Jérôme François <sup>†</sup>, Radu State <sup>\*</sup>

<sup>\*</sup> University of Luxembourg, SnT, 29, Avenue J.F Kennedy, L-1855 Luxembourg

Email: {wazen.shbair, mathis.steichen, radu.state}@uni.lu

<sup>†</sup> INRIA Nancy Grand Est, 615 rue du Jardin Botanique, 54600 Villers-les-Nancy, France

Email: jerome.francois@inria.fr

**Abstract**—Conducting experiments to evaluate blockchain applications is a challenging task for developers, because there is a range of configuration parameters that control blockchain environments. Many public testnets (e.g. Rinkeby Ethereum) can be used for testing, however, we cannot adjust their parameters (e.g. Gas limit, Mining difficulty) to further the understanding of the application in question and of the employed blockchain. This paper proposes an easy to use orchestration framework over the Grid’5000 platform. Grid’5000 is a highly reconfigurable and controllable large-scale testbed. We developed a tool that facilitates nodes reservation, deployment and blockchain configuration over the Grid’5000 platform. In addition, our tool can fine-tune blockchain and network parameters before and between experiments. The proposed framework offers insights for private and consortium blockchain developers to identify performance bottlenecks and to assess the behavior of their applications in different circumstances.

## I. INTRODUCTION

Since the inception of Bitcoin [1], blockchains are attracting more and more interests. At the heart of any blockchain are cryptographically linked blocks. They contain transactions that modify the state of the blockchain directly or through the execution of smart contracts. Using the public and private keys linked to their respective accounts, users can create transactions and sign them verifiably. This signature is required, as certain actions can only be performed by well specified accounts. This is for example the case when transferring cryptocurrencies from one account to another.

A consensus algorithm defines how blocks can be appended to the blockchain. It dictates which blocks are valid, how they are created and therefore indirectly how quickly they are generated. While Bitcoin relies on *Proof of Work*, other methods such as *Proof of Stake* and *Proof of Authority* exist. The most prominent consensus algorithm, *Proof of Work*, forces nodes to compete in iteratively computing hashes comprising the block data and a nonce that is incremented to find a hash value that satisfies a predetermined condition. If a node believes to be the first to have found a satisfying nonce, it publishes the block. Other nodes can easily verify a received solution by hashing the block and the nonce exactly once [2].

This leads to the following advantages of blockchains. First, because one block is linked to the previous one and because of the consensus algorithm, it is difficult for a single node (or a small group) to arbitrarily modify past transactions. The data stored in a blockchain is thus considered to be immutable. This

is true as long as the network as a whole operates correctly. Second, the blockchain, due to its tolerance of adversarial nodes, allows the collaboration of multiple entities, possibly with differing interests, without compromising the system. A user can therefore rest assured that transactions performed on the blockchain will not be removed or arbitrarily modified and that a small group of participants cannot easily modify the blockchain to gain an advantage over other users.

Since blockchains are still relatively new, especially the second generation blockchains [3], this work proposes an easy to use orchestration framework for private and consortium blockchains. As a result, applications deployed or relying on blockchains can be tested at a scale comparable to the public networks. This allows applications to be verified in conditions that are similar to the production environments, and to further the understanding of the application in question and of the employed blockchain. Also experimental conditions can be actually changed to determine good and bad configurations before a real deployment, for example in terms of the underlying network topology. Moreover, the framework can be used to determine which blockchain to use for a given application, while modeling the scale and the occurrences of public blockchain networks. The framework can further be useful to future research as it can be utilized to verify theories and test proposed improvements or modifications. The framework is distributed and available on Github<sup>1</sup>

This text is presented as follows. Section II describes the mentioned orchestration framework. Section III then introduces KYC (Know Your Customer), a use case that is closely related to the financial industry. The use case is then employed to evaluate and perform experiments on the orchestration framework in Section IV. Section V discusses related work. Finally, the conclusion is given in Section VI.

## II. EXPERIMENTATION FRAMEWORK

The experimentation of distributed applications like blockchains needs a highly reconfigurable and controllable environment for fine-tuning blockchain and network parameters in different scenarios. Therefore, there might be significant manual operations which lead to human errors and make it hard to reproduce experiments. Experiment management

<sup>1</sup><https://github.com/wshbair/Blockchain-Orchtration-Framework>

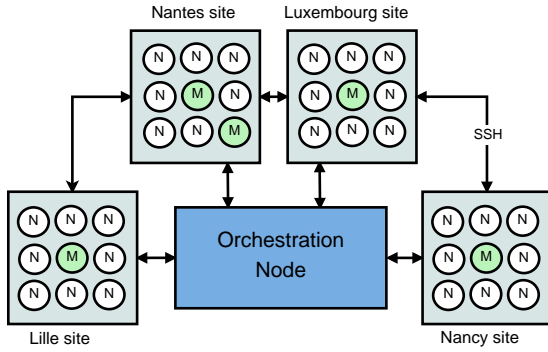


Fig. 1: Grid'5000 as blockchain experimentation testbed (N: transaction node, M: mining node).

tools are valuable means for researchers to automate low-level tasks like operating system installation and networking configuration. Also, testbeds offer controlled environments that are normally shielded from the randomness of production environments. The researchers can focus on the scientific questions to answer and forget about time-consuming, low-level details of experiments. In [4] the authors provide an extensive list of features offered by general-purpose experiment management tools and common testbeds.

In the context of blockchain technologies, there are still many open questions related to scalability, security, privacy and service level management. These features need to be ensured if this technology is deployed in operational and sensitive environments, such as the financial sector. Most existing approaches for assessing the limits of different blockchain technologies rely either on simulators or virtual environments. However, these cannot accurately predict the performance in the real world, where heterogeneous blockchain nodes are geographically distributed.

Hence, in the next sections we present a large scale experimentation testbed and a novel orchestration tool for dynamically deploying blockchain platforms and smart contracts over a testbed.

#### A. Experimentation platform

Building a representative evaluation testbed needs a flexible environment for fine-tuning blockchain and network parameters in different scenarios. Grid'5000<sup>2</sup> is a large-scale testbed environment for experiment-driven research. A large amount of resources (1000 nodes, 8000 cores, 10G Ethernet links), scattered over many geographic sites (in France and Luxembourg), makes it an environment that provides conditions that are very close to real blockchain networks. Indeed, blockchain applications operate over a network, like the Internet, shared by different applications making congestion and message delays unpredictable. Therefore, the blockchain applications should be assessed under real conditions as accurately as possible.

The Grid'5000 platform provides a real physical hardware and networking infrastructure that is equipped with advanced measurement features to evaluate any application, such as blockchains, from different aspects with extensive tests. This can be thus leveraged to gain new and valuable knowledge related to the monitoring and orchestration of blockchain infrastructures. This knowledge is vital for enterprises that use private/consortium blockchain platforms to run their blockchain network within predefined requirements (e.g., transaction throughput and latency). So we can critically study, if a given configuration provides the optimal or an acceptable performance for a particular application scenario.

Figure 1 shows how a possible deployment of blockchain nodes on Grid'5000 can build a blockchain network over different geographic sites (i.e., Lille, Nancy, Luxembourg, Nantes), where each site can host transactions nodes (i.e., that send transactions to the network) and/or mining nodes. This ability to adapt the available Grid'5000 resources makes it possible to test various blockchain technologies and applications with heterogeneous configurations. For instance, for a given blockchain platform (e.g., Ethereum [5], [6]) we can assess the performance of the network while varying the number of mining nodes.

However, evaluating the performance of a blockchain application assuming a single parameter is too narrow because there is a multitude of parameters such as mining difficulty and Gas limit that potentially affect the general performance of the blockchain network. Moreover, some other parameters are inherited from the fact that we are dealing with distributed and decentralized systems, for example the number of participants, network topology, hardware configuration (e.g., CPU frequency, network bandwidth). Thus, an experiment management tool is necessary to control the various experimental parameters. The next section details our orchestration tool for conducting blockchain experiments on the Grid'5000 platform.

#### B. Blockchain orchestration tool

In the area of cloud computing, orchestration refers to the management of resources including placement and configuration of virtual machines, for example [7]. In our context, the blockchain orchestration tool automates the deployment of blockchain infrastructure (nodes, network) and smart contracts in a given environment. Here, we focus on two types of blockchain networks; the private and consortium ones. Because installing and managing a private blockchain environment includes significant manual work due to the many components that need to be installed, configured and monitored properly, we developed an orchestration tool to automate the blockchain infrastructure installation and deployment, and to provide a mean to control the whole life cycle of experiments.

The Grid'5000 platform offers different experiment management tools, such as experiment scheduling, resource allocation and node reconfiguration [8]. However, still there is no dedicated tool that can manage blockchain-based experiments. It was an important design goal to not reinvent the wheel by developing a completely new tool, but to extend the

<sup>2</sup><https://www.grid5000.fr>

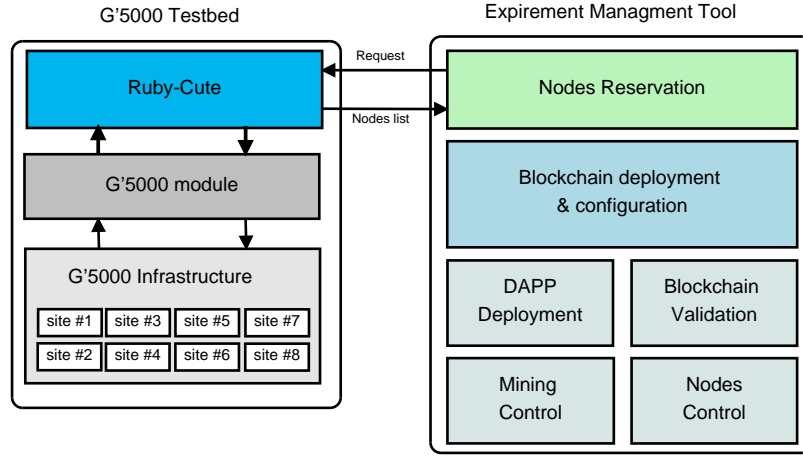


Fig. 2: Blockchain orchestration framework components.

functionality of a currently existing tool that works with the Grid'5000 platform. Thus, we developed our orchestration tool on top of Ruby-Cute<sup>3</sup> to handle blockchain deployment and configuration. Ruby-Cute is a set of commonly used tools for experiments. It is a Ruby library (gem) aggregating various snippets useful in the context of development of experiment control scripts on Grid'5000. It interacts with the Grid'5000 infrastructure through an API<sup>4</sup> to manage the most common activities like resource reservation and deployment of environments. The current version of our tool is built to orchestrate private/consortium Ethereum blockchain platforms. In the future, we plan to cover more blockchain platforms.

Figure 2 depicts the architecture of the proposed framework. The Grid'5000 testbed (on the left) is interfaced with the Ruby-Cute module. The experiment management tool (on the right hand side) comprises the node reservation module that sends the reservation request to Ruby-Cute, which handles the resource reservations and operating system deployments. Once the resource are ready, a list of nodes (i.e., machine names) is forwarded to the blockchain deployment and configuration module, which has four main tasks to prepare the blockchain environment; Genesis file configuration setting, network initialization, miners startup, and finally configuration validation. **Genesis file configuration:** the genesis block is the starting block of the blockchain (block 0) and "genesis.json" is the file defining it. It looks like the "configuration" of the blockchain, as example we can set the level of difficulty to mine blocks or the Gas limit. Indeed, changing these parameters in the genesis file facilitates the creation of customized private blockchains. For instance, we can tune the "difficulty" parameter to control the speed of the block generation.

**Network initialization:** it disseminates the genesis-file and the experiment scripts to the participating nodes. Then it initializes the blockchain client in each node. Once all nodes are running, it grabs the addresses of the nodes (e.g, enode://address:@IP:port) to construct the Static-nodes file which in-

cludes the peers that all nodes should connect to in order to join the blockchain network.

**Miners startup:** it runs all the nodes to start mining. So each node will gain some Ethers (in case of Ethereum blockchain), consumed later to deploy smart contracts and to send transactions to the network.

**Configuration validation:** it deploys a smart contract and runs workloads. This task is essential to test the configuration of the blockchain and to make it ready and steady.

To summarize this section, we presented the functionality of the blockchain orchestration tool and how it can be used with Grid'5000 to provide a missing tool to control blockchain experiments. The next section elaborates a case study of a FinTech application. Our motivation is to show how the proposed orchestration tool and the Grid'5000 testbed can be used to evaluate the performance of the application over a large-scale testbed.

### III. USE CASE: KYC SMART CONTRACT

Nowadays there is a huge number of banking transactions around the world. For instance, the Society for Worldwide Interbank Financial Telecommunication (SWIFT) has recorded 28.4<sup>5</sup> million financial transactions per day. Hence, to avoid transaction fraud it is vital to have proper techniques for verifying the identity of the parties involved in a payment transaction and also their rights to perform and/or receive a payment transaction.

The term 'Know Your Customer' (KYC) is widely used in the financial world by financial institutions and other regulated companies to identify their clients before doing financial business with them. The process of KYC is as follows: if a client needs to make a payment transaction from his/her bank account to another bank account through a payment provider, the payment provider does the KYC to validate the client's identity for instance by the client's name, from the bank where the money was transferred from. The client's name is then crosschecked with the name stated by the client. during the

<sup>3</sup><https://github.com/ruby-cute/ruby-cute>

<sup>4</sup><http://www.rubydoc.info/github/ruby-cute/ruby-cute/Cute/G5K/API>

<sup>5</sup><https://www.swift.com/about-us/swift-fin-traffic-figures>

sign-up of the payment provider's service [9]. Also the client's name may also be crosschecked against third-party databases like World-Check<sup>6</sup>, which helps organizations fight financial crime.

#### A. Traditional KYC problems

Performing KYC is an easy task in countries that provide electronic services to verify a persons identity. However, if such services are not available to financial institutions it is risky to accept individuals as clients [10]. Therefore, the KYC process performed when on-boarding a new client is time-consuming and each financial institution must do its own KYC. For instance, when a client needs to open a bank account, the bank sends the client's information to the registries, which store the client's information in their databases and the customer becomes "KYC Compliant". This process is repeated every time the client needs to open a new bank account.

#### B. Blockchains to the rescue

Blockchain technology can solve many of the problems related to KYC like the on-boarding issue. The blockchain of the public distributed ledger can disseminate the client's information across many banks, once verified. Thus, a KYC once performed can be accessed by other financial institutions with unique authorization from the client. This will make the KYC process much easier, simpler, less time consuming and cost-effective. Second, the centralized databases-based KYC solutions also expose a vulnerability. While in blockchain, the KYC data is replicated across many various nodes, which makes it immutable and traceable, because of blockchain's append-only data structure [9].

#### C. Blockchain-based KYC POC

This use case demonstrates how the blockchain can be used to serve the KYC process. Figure3 illustrates the use case of the KYC. The steps are explained as follows:

- 1) A client allows a bank to perform KYC by providing documents like identity card, financial situation, etc.
- 2) The bank will review and validate identity and financial information about the client and approve him/her as 'KYC Compliant'.
- 3) The bank adds KYC information and status to a blockchain platform to confirm it has been verified.
- 4) The bank give the client a token that can be used as a proof of his/her KYC status.
- 5) A third party can be given the authorization by the client to check his KYC status.
- 6) The intended bank can check the KYC information.

#### D. Blockchain KYC POC Implementation

To implement the blockchain KYC we used the Ethereum blockchain platform and the Solidity<sup>7</sup> programming language to write the smart contract. The source code of the smart

<sup>6</sup><https://risk.thomsonreuters.com>

<sup>7</sup><https://solidity.readthedocs.io/en/develop/>

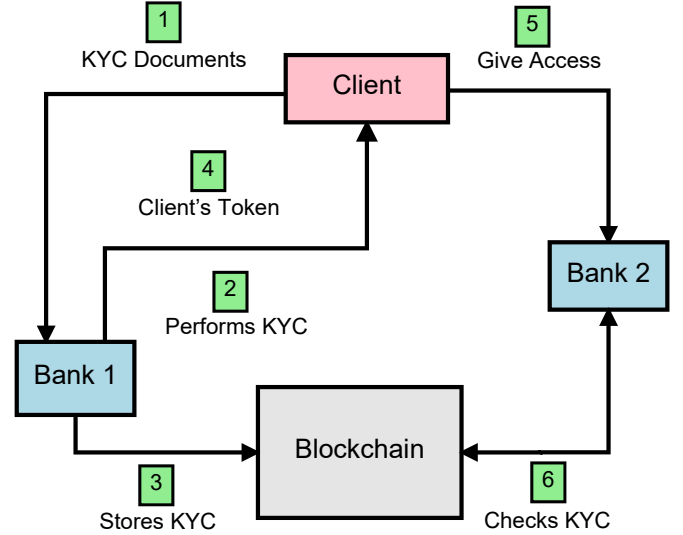


Fig. 3: The workflow of the KYC POC

contract is available on Github<sup>8</sup> Due to the storage limitation of blockchains, we use a Peer-to-Peer file system, the Inter-Planetary File System (IPFS), to store the KYC documents like identity card and passport [11]. The main advantages of IPFS are: it provides a permanent and decentralized method of storing and sharing files and it has no single point-of-failure compared to centralized data storage systems. Therefore, IPFS is in use at: banks, legal archives, blockchain companies and smart contract applications [12]. In the KYC case, each file stored in IPFS is given a unique fingerprint (cryptographic hash), which will be stored in the KYC smart contract.

#### IV. EVALUATION AND EXPERIMENTAL RESULTS

As explained in section II the performance of a blockchain-based application is difficult to evaluate with respect to one single parameter, since there are many parameters that control the performance of the blockchain environment. In this section, we elaborate how the orchestration tool presented in SectionII-B facilitates the evaluations using the Grid'5000 testbed. The evaluation starts with reserving nodes, so as input to our orchestration tool we specify the number of nodes to be reserved per site and the duration of the experiments. Once the nodes are ready, our tool goes through the steps explained in II-B to deploy and to configure a private blockchain and setup the KYC application.

A scripted workload deploys the KYC smart contract and starts the workload generator. The workload simulates KYC client registration events as a Poisson distribution, as its widely used to simulate queues like the arriving of emails or calls [13]. Our aim is to show how the proposed framework can be used to study the behavior of blockchain, with specific configurations, under massive KYC client registration requests.

For this experiment, we reserved 25 nodes scattered over five different geographic sites. The Gas-limit

<sup>8</sup><https://github.com/wshbair/Blockchain-Orchtration-Framework/KYC>



TABLE I: KYC POC evaluation results (Tuning: number of mining nodes)

# Nodes	# Mining nodes	Gas Limit	Mining difficulty	# Client	Mean Poisson (ms)	Latency (sec)
25	25	0x8000000000000	0x5ffffff	100	10	383.48
25	20	0x8000000000000	0x5ffffff	100	10	419
25	18	0x8000000000000	0x5ffffff	100	10	413
25	15	0x8000000000000	0x5ffffff	100	10	381.8
25	13	0x8000000000000	0x5ffffff	100	10	418.72

TABLE II: KYC POC evaluation results (Tuning: Mean Poisson distribution)

# Nodes	# Mining nodes	Gas Limit	Mining difficulty	# Client	Mean Poisson (ms)	Latency (sec)
25	25	0x8000000000000	0x5ffffff	100	500	376.1
25	25	0x8000000000000	0x5ffffff	100	50	386.34
25	25	0x8000000000000	0x5ffffff	100	10	383.48

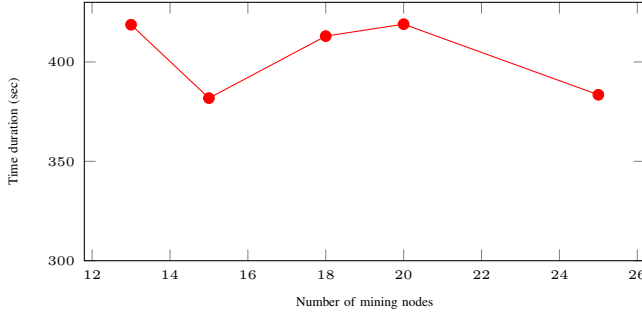


Fig. 4: Tuning the number of mining nodes vs. the latency time to handle 100 clients registration

(0x8000000000000), mining difficulty (0x5ffffff) and number of requests are fixed, while we tune the number of mining nodes. As consensus algorithm, we use the PoW one to keep the experiments simple. Table I shows the effect of changing the number of mining nodes (shaded column) while all other parameters are fixed. The duration column shows the latency time taken by the blockchain to handle these requests.

As shown in Figure 4, minimizing the number of mining nodes does not have a significant effect on the latency, because the workload is small. However, there is a trade-off between lowering the number of mining nodes and the security of blockchain network, because more mining nodes means more nodes validate the transactions.

Table II illustrates the results for tuning the Mean of the Poisson distribution (the shaded column) while other parameters are fixed. We also noticed that the latency does not change significantly if the delay between transactions varied from 500 ms to 10 ms. Thus, we plan to study the location of the nodes sending transactions to the network. For example, if the transactions are generated from Lille (north of France) and the mining nodes are located in Sophia (South of France). As the Grid'5000 infrastructure is well connected there is no issue to have it distributed, so we expect to get results that reflect the pure behavior of blockchain network protocol. We believe it will enrich our understanding of the application in question and of the employed blockchain.

## V. RELATED WORK

Since Bitcoin [1], blockchains have been gaining in popularity. They are used in vastly different areas, for example to perform payments as in Bitcoin, together with IoT devices as in [14], or to allow users to implement smart contracts that are executed on the blockchain [5].

Although the public blockchain networks comprise several thousand nodes (Bitcoin and Ethereum), there is also a need for the private execution of blockchain transactions as mentioned in [15]. For example, Multichain [16] is a permissioned blockchain, wherefore only participants who have been given the corresponding permissions can view or interact with the blockchain. Furthermore, [15] proposes an approach with a system of interacting satellite chains with which it is possible to share transactions only with selected entities.

As discussed in [2], different blockchain variants share a common basis, the chain of blocks, but may differ greatly in other aspects, such as the consensus algorithm. The latter also impacts the performance of blockchains, for example the scalability [17]. Consequently, *Proof-of-Work* blockchains in particular have been given consideration, especially with respect to improving the scalability. [18] proposes a new blockchain protocol based on epochs and leaders that propose new blocks at a fast rate, while [19] and [20] use off-blockchain payment channels.

In addition, research has been conducted on analyzing blockchains and providing tools to stress different blockchain components. On one hand, [21] analyzes the performance of private blockchains. To do this, they utilize a single machine located in a cloud and deploy either Hyperledger [22] or Ethereum blockchains. Due to the single machine setup, consensus mechanisms are not considered in that work. On the other hand, [23] considers multiple layers within a blockchain system and proposes workloads that stress a target layer or the blockchain system as a whole. Multiple machines located in a cluster are used simultaneously, and the consensus algorithm is considered.

As opposed to these works, this paper provides an easy to use framework that distributes nodes to distant geographical sites, thus providing conditions that are similar to real blockchain networks that can be scattered either over a country, a continent or the world. Furthermore, the tool allows

the configuration of the blockchain nodes. The proposed tool can be used for the following purposes. First, custom blockchain deployments can be tested, verified and adapted before being used in a production environment. Second, new implementations dealing with the performance of blockchains can be analyzed at a large scale. Third, measurements can be performed under varying conditions, such as the location of the nodes or the blockchain configuration, again, at large scale.

To achieve this, the tool proposed herein uses the Grid'5000 platform. Grid'5000 has been presented in [8] and is supported by multiple sites in France and one in Luxembourg.

## VI. CONCLUSION

A blockchain experimentation framework is a valuable means for researchers and developers to further their understanding of the blockchain technology. Therefore, there is a pressing need for a highly reconfigurable and controllable environment to be able to experiment with blockchains extensively. In this paper, we proposed using the Grid'5000 a large scale distributed platform that can easily be controlled and reconfigured. In parallel, we present our orchestration tool that provides a means to control and manage private blockchain environments over Grid'5000. As a direct application of using this framework, the blockchain KYC application has been discussed. A POC has been implemented. We provided preliminary and basic results of how the framework (the testbed and the orchestration tool) can be used to conduct blockchain evaluation experiments on a large scale real environment.

As future work, we still have many open questions to answer; regrading the evaluation of blockchain environments from different aspects and more focus will be given to security and privacy issues of private blockchain applications. Also we need to take our orchestration tool a step further to support other common-known blockchain platforms like IBM Hyperledger. More advanced technology will be integrated in our orchestration framework, mainly the Ansible<sup>9</sup> automation technology.

## VII. ACKNOWLEDGEMENTS

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations. The present project is supported by the National Research Fund, Luxembourg under grant agreement number 11785982.

## REFERENCES

- [1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [2] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Big Data (BigData Congress), 2017 IEEE International Congress on*. IEEE, 2017, pp. 557–564.
- [3] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money," in *Banking Beyond Banks and Money*. Springer, 2016, pp. 239–278.
- [4] T. Buchert, C. Ruiz, L. Nussbaum, and O. Richard, "A survey of general-purpose experiment management tools for distributed systems," *Future Generation Computer Systems*, vol. 45, pp. 1 – 12, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167739X14002003>
- [5] G. Wood. (2018) Ethereum: A secure, decentralised generalised transaction ledger. [Online]. Available: <http://yellowpaper.io/>
- [6] (2017) Ethereum web site. Ethereum Foundation. [Online]. Available: <https://www.ethereum.org/>
- [7] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, "Scientific cloud computing: Early definition and experience," in *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*. Ieee, 2008, pp. 825–830.
- [8] F. Cappello, E. Caron, M. Dayde, F. Desprez, Y. Jégou, P. Primet, E. Jeannot, S. Lanteri, J. Leduc, N. Melab *et al.*, "Grid'5000: A large scale and highly reconfigurable grid experimental testbed," in *Grid Computing, 2005. The 6th IEEE/ACM International Workshop on*. IEEE, 2005, pp. 8–pp.
- [9] M. Mainelli, M. Smith *et al.*, "Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers (aka blockchain technology)," *The Journal of Financial Perspectives*, vol. 3, no. 3, pp. 38–69, 2015.
- [10] J. De Geer, "Method and system for identity and know your customer verification through credit card transactions in combination with internet based social data," May 14 2012, uS Patent App. 13/470,737.
- [11] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.
- [12] "Blockchains and the web: IPFS," accessed 7, December 2017. [Online]. Available: <https://www.w3.org/2016/04/blockchain-workshop/slides/benet-ipfs.pdf>
- [13] "Statistics: The poisson distribution," 2007, accessed 7, December 2017. [Online]. Available: <http://www.umass.edu/wsp/resources/poisson/index.html>
- [14] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
- [15] W. Li, A. Sforzin, S. Fedorov, and G. O. Karame, "Towards scalable and private industrial blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, ser. BCC '17. New York, NY, USA: ACM, 2017, pp. 9–14. [Online]. Available: <http://doi.acm.org.proxy.bnl.lu/10.1145/3055518.3055531>
- [16] Multichain. Accessed 26 January 2018. [Online]. Available: <https://www.multichain.com/>
- [17] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, J. Clark, S. Meiklejohn, P. Y. Ryan, D. Wallach, M. Brenner, and K. Rohloff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–125.
- [18] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, ser. NSDI'16. USENIX Association, 2016, pp. 45–59. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2930611.2930615>
- [19] J. Poon and T. Dryja. (2016) The bitcoin lightning network: Scalable off-chain instant payments. Accessed 26, January 2018. [Online]. Available: <https://lightning.network/lightning-network-paper.pdf>
- [20] Raiden network. Accessed 26, January 2018. [Online]. Available: <https://raiden.network/>
- [21] S. Pongnumkul, C. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," in *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, July 2017, pp. 1–6.
- [22] (2017) Hyperledger fabric web site. The Linux Foundation. [Online]. Available: <https://www.hyperledger.org/projects/fabric>
- [23] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 2017, pp. 1085–1100.

<sup>9</sup><https://www.ansible.com/>