KNOB: Expanding Blockchain Benchmarking Frameworks: A Web Application

Georgios Michoulis, Marios Touloupou, Evgenia Kapassa
Independent, Greece
gmixoulis@gmail
Institute for the Future, Department of Digital Innovation, University of Nicosia, Cyprus
touloupos.m, kapassa.e. @unic.ac.cy

***Abstract*—Nowadays many Consensus Algorithms and Blockchain networks have been proposed. All the initiatives that appeared, tried to address the blockchain trilemma, by enhancing key features. Corporations and individuals that seek to integrate blockchain networks in their products want to know which Consensus Algorithm will be the best to exploit. Moreover, setting up and configuring a private deployment of blockchain networks is complex, time-consuming, and usually a difficult task for people with a non-technical background. The lack of visualization and complex environments cannot help them to adapt easily to the blockchain that is most fitted to their needs. Thus, an easy-to-interpret information system is required which will be usable by both technical and non-technical teams. This paper introduces the expansion of our previous work, implementing and integrating a web application portal, on top of an already developed blockchain benchmarking framework. Finally, this paper presents the requirements, technical tools, and results of the open-source implementation.**

***Index Terms*—blockchain performance, benchmarking, consensus algorithms, web application, user interface**

I. Introduction

The BitCoin was introduced in 2008 by Satoshi Nakamoto [1], which was also the invention of the Blockchain and the Nakamoto Consensus [2]. To define blockchain one way to do is by quoting Drescher [3], one of the most accurate definitions:

> Blockchain is a purely distributed peer-to-peer system of ledgers that utilizes a software unit consisting of an algorithm, which negotiates the informational content of ordered and connected blocks of data together with cryptographic and security technologies to achieve and maintain its integrity.

Therefore, it is evident that the Consensus Algorithm (CA) plays a crucial role in the core of each blockchain. Specifically, CAs are responsible for the validation of transactions, as well as the synchronization of the ledger throughout the blockchain's life [4], [5]. Currently, there are plenty of known CAs [6] such as Proof-of-Work, Proof-of-Stake etc. Each CA has unique utilities and reaches consensus in the network differently [6].

A number of issues affecting the blockchain industry are known as the "blockchain trilemma". There are three options available, but only two of them can be accomplished at the same time. Security, scalability, and decentralization are the three options in the blockchain trilemma. This challenge makes it hard for developers and organizations to integrate a blockchain with the most suitable CA. In that sense, several research works have been presented trying to address the blockchain trilemma, resulting in several difficulties [7], [8]. Considering the above, when a developer/organization wants to adopt a blockchain, the following questions need to be taken into consideration:

- Does the blockchain perform well based on the requirements?
- Is the selected blockchain secure, scalable or decentralized enough?

The considerations mentioned above introduced the need for an open-source and easy-to-use framework that allows technical and non-technical users to be able to deploy, monitor, analyze, compare, and report various blockchain performance activities, toward the selection of the most appropriate one – based on their needs. In order to address the latter, a Web Application on top of a Blockchain Benchmarking Framework (BBF) is required that will i) automate the deployment processes of various blockchain networks, ii) analyze their performance and iii) expose the user to a User Interface (UI), abstracting the complexities, and time-consuming configuration processes.

The current paper introduces KNOB, which is a user-friendly and easy-to-use web application that interacts with the underlying BBF, enabling seamless interaction with non-expert users. The underlying BBF is outside the scope of this work, while it is already presented and evaluated in authors' previous articles published in [9], [10]. The contributions of this paper are the following:

- **C1:** Develop and present a user-friendly web application on top of the BBF, which will bridge the gap between technical and non-technical teams.
- **C2:** Abstract the technical details from the end-user, while deploying a network and creating benchmark tests.

The rest of the paper is structured as follows. Section 2 discusses BBFs frameworks that have been identified in the literature, while Section 3 provides a conceptual overview of the proposed web application and its system requirements. The detailed architecture of KNOB is presented in Section 4. Finally, Section 5 concludes the paper with some final thoughts and future work.

II. Related Work

The current literature introduces some BBFs that are going to be presented within this section. Although, it was noticed that they are diverse from real-world scenarios since they are based on several assumptions and hypotheses. To begin with, one of them was presented by Carlsten et al [11] who based on a theoretical analysis and a Bitcoin simulator, worked on the so-called "selfish mining". As a result, the authors have demonstrated that a miner with an arbitrarily low hash power share and who is arbitrarily dispersed over the network can be made profitable. Additionally, Gervais et al [12], introduced a quantitative framework to examine how different Proof of Work (PoW) based blockchain networks affect the security and performance of the network. Additionally, the proposed architecture allowed them to record both PoW-based deployments and PoW blockchain variations that are constructed with various settings. Moreover, BlockSim [13], is a framework that allows the user to build event-based and dynamic system models for blockchain systems. It is also worth mentioning that the framework is mostly focused on the modeling and simulation of block creation, based on PoW.

On the other hand, as it is discussed by Touloupou et al, [10], the identified blockchain simulators make use of several assumptions and hypotheses. Thus the simulated results are often far from the real-world scenarios. For that reason, different benchmarking frameworks, such as [14]–[16] have been introduced trying to address that challenge.

However, to the best of our knowledge, none of the current works provide a user-friendly application to interact with underlying benchmarking tools. During the research and development of the authors was realized that the developed BBF has some limitations [17]. Specifically, (i) it is complex and time-consuming, (ii) it has many dependencies, and (iii) it needs high technical expertise in order to be used by non-technical experts. Therefore, the need for a friendly user interface that minimizes these implications is highlighted. The current paper expands the already proposed benchmarking framework, towards the development of a user-friendly, web-based application that is responsible for the communication between the user and the core components of the BBF.

III. Concept Overview

Through the analysis presented in the previous sections, it became evident that current benchmarking frameworks lack the appropriate level of abstraction. Therefore, this section will present the conceptual overview of the proposed web application named KNOB. The main aim of KNOB is to expand the usability of the implemented BBF abstracting the technical details from the end user and reducing the time needed.

The proposed application is based on the established blockchain layer stack as presented in [18], [19], consisting of the infrastructure layer, the network layer, the protocol layer, the service layer, and the (optional) application layer. The proposed web application is part of the application layer, and its conceptual overview is depicted in Figure 1. KNOB consists of three parts: (a) the front-end, which enables the interaction with the end-user, (b) the back-end, which is the underlying BBF and (c) the middle layer, which provides the necessary communication functionalities.

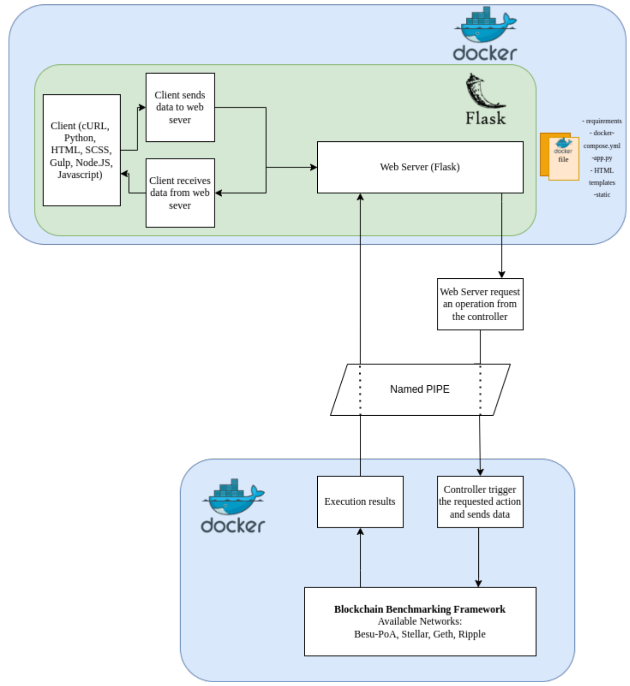

Fig. 1. KNOB concept overview

KNOB, (i.e., part of the front-end) consists of a web server and the client interface. The BBF (i.e. part of the back-end) is using a Docker container [20]. The necessary communication takes place with standardized API calls. The developed APIs work locally and execute shell scripts. To be able to use them from the Docker containers, we incorporated the "pipe method", enabling a more radical and user-friendly way of executing shell scripts at the host machine. The "pipe" is an expansion to the standard pipe notion in Unix and Unix-like systems [21]. Thus, the middle layer of the KNOB consists of the pipe and the communication processes. The final steps of the typical process for using the application are as follows:

- The client (i.e., end-user) sends a request to the web server through the KNOB application.
- The web server receives the inquiry from the client and pushes it to the named pipe.
- The named pipe calls the proper method from the exposed APIs of the BBF.
- The BBF controller executes the inquiry and returns the results to the named pipe.
- The Named pipe stores the logs and/or the results of the BBF in a local file.
- The front-end application receives/reads the execution and returns it to the user/client.

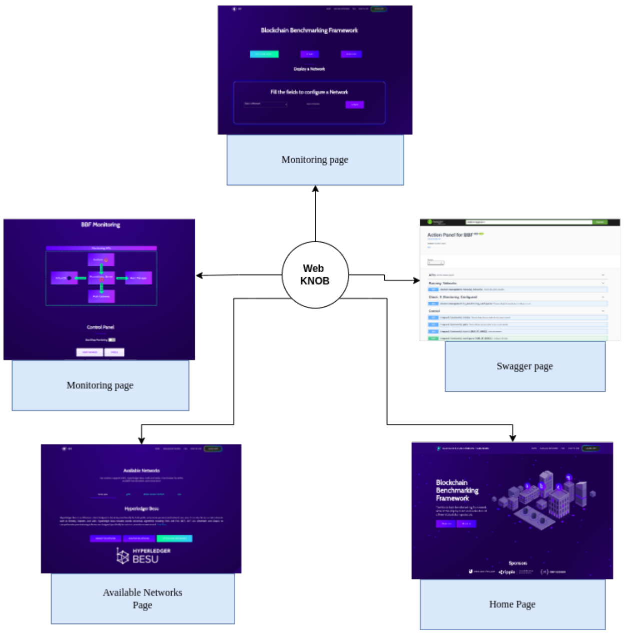

Fig. 2. The KNOB's available Components

IV. Architecture Description

Following the concept overview, within this section, the architecture of the proposed web application will be presented. This section will first describe the working environment and the present KNOB's components. The architecture consists of four main components named: (a) Swagger UI, (b) Available Networks, (c) Monitoring, and (d) Benchmarking Engine as depicted in Figure 2. Each component corresponds to a dedicated service (i.e., web page) in the implemented web application, described in the following sub-sections. The core functionalities of the proposed application take place within the Benchmarking Engine service.

A. Working environment

To begin with, KNOB was also implemented using JavaScript (JS) [22] and Python [23],], allowing fast development, compatibility, and easy integration with the underlying BBF. Specifically, we used Node.js, Gulp.js, and jQuery. Concerning the UI/UX design, KNOB utilized Bulma [24] and Bootstrap [25] open-source frameworks. The communication from the web application to the named pipe is trigged by a Flask API call [26]. The front end fetches the Flask API using JavaScript functions, and the results are returned to the user.

Additionally, KNOB requires a set of minimal resources to be developed locally (e.g., memory usage, CPU usage, Disk space, NET I/O, and Block I/O), as those summarized in Table I. It is pointed out that the development of the blockchain networks from the BBF (i.e. out of this paper's scope since it is described in our previous article in [10]) requires more resources. Currently, the KNOB is publicly available in our GitHub repository [27] and hosted on an Amazon web server[1].

[1] https://bbf-gui.ddns.net/home

	Values
Memory	166 MiB
CPU %	1.48% of an Intel i7-10750H CPU @ 2.60GHz × 6
Disk space	953,1 MB (after the development of the container, including the BBF)
Net I/O	66.8MB / 1.47MB
Block I/O	51.8MB / 274kB

TABLE I
KNOB's resource requirements

B. Swagger UI

The first component is linked with the Swagger UI. Swagger allows machines to read API structures. It can produce API client libraries in various languages and automate testing. Some of the currently available APIs are the following:

- Configure a network: It deploys a network and builds the required dependencies.
- Start a network: It starts the Docker container and the communication between the blockchain nodes
- Stop a network: It stops the communication between the blockchain nodes and the Docker containers that represents the nodes.
- Delete a network: It cleans and deletes permanently the deployed dependencies for a network.
- Traffic Generation: Generate traffic to examine it at the Monitoring Engine.
- APIs that returns different kind of information, such as Docker statistics, node logs, server logs, etc.

C. Available Networks

The Available Networks component provides all the necessary information related to the supported network within the web application. Currently, KNOB supports the following networks, although the application can be parameterized in order to support additional ones:

- *XRPL*. It employs a CA in which certain servers (i.e. known as validators) agree on the sequence and result of the XRP transactions, every 3-5 seconds. All network servers execute transactions using the same rules, and those that follow the protocol are promptly confirmed. All transactions are accessible to anyone, and the integrity of the network is ensured through cryptographic techniques. Furthermore, as the validator pool increases, the CA assures that the blockchain becomes more decentralized over time [28].
- *HyperLedger Besu*. Besu is an Ethereum client developed for corporate use cases in both public and private permissioned networks. It may also be tested on Rinkeby, Ropsten, and Goerli test networks. It supports CAs such as PoW and PoA [29].
- *Ethereum*. Ethereum allows end-users and developers to construct applications keep assets, trade, and communicate without relying on a central authority. Within

Ethereum, there is no need for sharing data with third-parties since the end user retains control of their personal information [30].
- *Stellar*. The main aim of Stellar is to track ownership. Specifically, Stellar employs a ledger, similar to the one accountants use. Stellar has no central authority, so no one can halt the network or covertly change the numbers, yet the ledgers are validated and updated every five seconds [31].

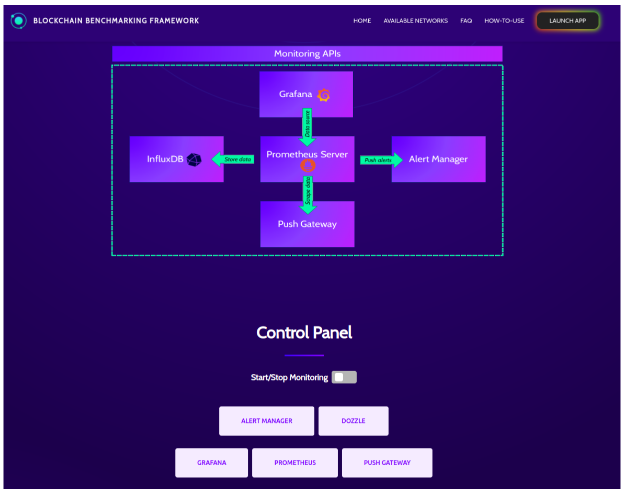

Fig. 3. Monitoring Engine

D. Monitoring

One of the most important components of KNOB is the Monitoring Engine. The UI of this component is depicted in Figure 3. One of the functions of this component is the start/stop the Monitoring Engine. Moreover, it presents a control panel where the user can leverage all the functionalities of the underlying BBF. KNOB in association with the BBF takes advantage of cutting-edge technologies in order to constantly monitor and manage the deployed networks. These are the following:

- *Prometheus*: Prometheus saves data locally on the disk for quick data storage and querying (with the ability to store them remotely as well). Under the scope of this work, the Prometheus monitoring server was used to store the data generated from the executed benchmarking tests [32].
- *Grafana*: Grafana is used to visualize the end-user's metrics, as well as query, explore and generate alerts. Grafana provides the necessary visualization of the benchmarking metrics and the monitoring results [33].
- *InfluxDB*: IKNOB takes advantage of InfluxDB in order to store and analyze time series data produced by the benchmarking tests, in real time. Additionally, InfluxDB can be configured as a data source in the Grafana service [34].

E. Benchmarking Engine component

The Benchmarking Engine provides all the core functionalities of KNOB. Specifically, this component allows the end-user to configure the networks, control the configured networks (start/stop), generate traffic in the form of payment transactions, and analyze advanced details of each network. Figure 4 depicts the workflow of the Benchmarking Engine, where a step-by-step workflow is presented. The latter are also described below:

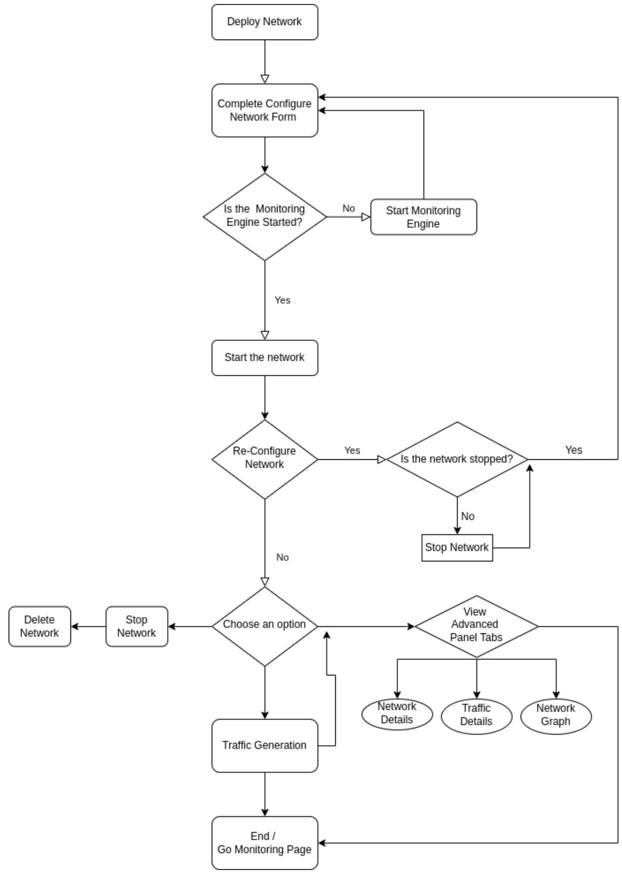

Fig. 4. WorkFlow of the Benchmarking Engine

1) The user initializes the deployment of a network, through a user-friendly web form, providing all the necessary configurations needed. The configuration starts as soon as the user has selected the preferred network and the desired number of nodes.
   a) In case the Monitoring Engine is not started, a message will prompt the user to start it and try again, as shown in Figure 5. It must be highlighted here that the workflow cannot be completed if the monitoring engine is down.
   b) User retries to configure the network.
2) In case the end-user wants to re-configure the network (e.g. wrong parameters were inserted), he/she should first stop the network, and then try again.
3) Once the network is started, the Traffic Generation can be configured. While configuring it, a prompt will ask the end-user to complete once again a user-friendly web form with the number of wallets and the number of tokens to be exchanged.

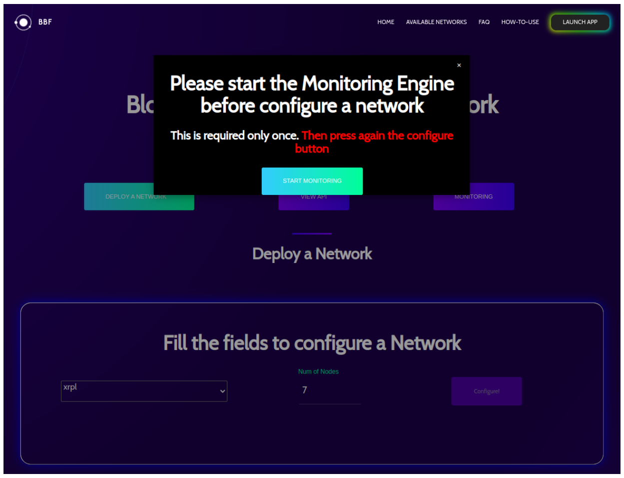

Fig. 5. Start Monitoring Engine prompt and panel of configuring a network

4) Then, the end-user can view the advanced options of the applications. KNOB web application allows the user to access the following information related to the deployed network:
   - *Network Details*: Information about the server and Docker's container data for the running containers is provided.
   - *Traffic Details*: Generated wallets, transaction details, and logs are presented.
   - *Network Graph*: The network's validated nodes are visually presented through an interactive graph
5) Finally, the end-user can stop the network, and then delete it or start it again.

V. Conclusions & Future Work

The main contribution of this work was to propose and develop a user-friendly environment on top of a BBF. In this paper, we proposed an open-source web application, the so-called, "KNOB", which aims to abstract the technical details from the end-user, while deploying a blockchain network and executing different benchmarking tests.

During the implementation of our proposition, different limitations have been identified. The named "pipe" used for the communication between the docker containers and the host machine was evaluated as an unstable solution because it prevents easy troubleshooting and debugging when the application was failing. Therefore, in our future plans, we aim to replace the latter with other solutions identified from the literature. An example would be the usage of an SSH connection where you would be able not only to communicate with the local machine but also with other hosts. Moreover, the UI is planned to be enhanced, taking into consideration the user experience (UX) and focusing also on mobile devices.

Acknowledgment

This work has been supported by the University Blockchain Research Initiative (UBRI) project, funded by the Ripple's Impact Fund, advised funds of the Silicon Valley Community Foundation (Grant ID: 2018-188546 and Grant ID: 2021-244121). Also, this work was supported by the XRP Ledger Developer Program, and Wave 2 grants offered by Ripple Labs Inc.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Bitcoin.org*, May 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[2] L. Ren, "Analysis of nakamoto consensus," *Cryptology ePrint Archive*, 2019.

[3] D. Drescher, *Blockchain Basics : A Non-Technical Introduction in 25 Steps*. New York: APRESS, 2017.

[4] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[5] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[6] G.-T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *Journal of Information processing systems*, vol. 14, no. 1, pp. 101–128, 2018.

[7] S. Reno and M. M. Haque, "Utilizing off-chain storage protocol for solving the trilemma issue of blockchain," in *Emerging Technologies in Data Mining and Information Security*, P. Dutta, A. Bhattacharya, S. Dutta, and W.-C. Lai, Eds. Singapore: Springer Nature Singapore, 2023, pp. 169–179.

[8] G. D. Monte, D. Pennino, and M. Pizzonia, "Scaling blockchains without giving up decentralization and security: A solution to the blockchain scalability trilemma," in *Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, ser. CryBlock '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 71–76. [Online]. Available: https://doi.org/10.1145/3410699.3413800

[9] M. Touloupou, K. Christodoulou, A. Inglezakis, E. Iosif, and M. Themistocleous, "Towards a framework for understanding the performance of blockchains," in *2021 3rd Conference on Blockchain Research and Applications for Innovative Networks and Services (BRAINS)*, 2021, pp. 47–48.

[10] M. Touloupou, M. Themistocleous, E. Iosif, and K. Christodoulou, "A systematic literature review toward a blockchain benchmarking framework," *IEEE Access*, vol. 10, pp. 70 630–70 644, 2022.

[11] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 154–167. [Online]. Available: https://doi.org/10.1145/2976749.2978408

[12] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 3–16. [Online]. Available: https://doi.org/10.1145/2976749.2978341

[13] M. Alharby and A. van Moorsel, "Blocksim: A simulation framework for blockchain systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, p. 135–138, jan 2019. [Online]. Available: https://doi.org/10.1145/3308897.3308956

[14] 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*, ser. SIGMOD '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1085–1100. [Online]. Available: https://doi.org/10.1145/3035918.3064033

[15] D. Saingre, T. Ledoux, and J.-M. Menaud, "Bctmark: a framework for benchmarking blockchain technologies," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, 2020, pp. 1–8.

[16] C. Natoli, H. Benoit, V. Gramoli, and R. Guerraoui, "Diablo: A distributed analytical blockchain benchmark framework focusing on real-world workloads," 05 2021.

[17] "GitHub - UNIC-IFF/blockchain-benchmarking-framework: Backbone scripts and configuration for blockchain benchmarking framework." [Online]. Available: https://github.com/UNIC-IFF/blockchain-benchmarking-framework

[18] P. Dhulavvagol, V. Bhajantri, and S. Totad, "Blockchain ethereum clients performance analysis considering e-voting application," *Procedia Computer Science*, vol. 167, pp. 2506–2515, 01 2020.

[19] U. Roth and T. N. Djoua, "Message exchange on base of a blockchain-based layered architecture," *it - Information Technology*, vol. 60, no. 5-6, pp. 253–261, 2018. [Online]. Available: https://doi.org/10.1515/itit-2017-0037

[20] "Home - Docker," May 2022. [Online]. Available: https://www.docker.com/

[21] "Named pipe," apr 2022, page Version ID: 1082902383.

[22] A. Wirfs-Brock and B. Eich, "Javascript: The first 20 years," *Proc. ACM Program. Lang.*, vol. 4, no. HOPL, jun 2020. [Online]. Available: https://doi.org/10.1145/3386327

[23] G. van Rossum, *Python Tutorial*, ser. CS-R. Centrum voor Wiskunde en Informatica, 1995. [Online]. Available: https://books.google.gr/books?id=hHMVPwAACAAJ

[24] "Bulma: Free, open source, and modern CSS framework based on Flexbox." [Online]. Available: https://bulma.io

[25] J. Thornton and B. M. Otto, "Bootstrap." [Online]. Available: https://getbootstrap.com/

[26] M. Grinberg, *Flask web development: developing web applications with python*. O'Reilly Media, Inc., 2018.

[27] G. Michoulis, M. Touloupos, E. Kapassa, and K. Christodoulou, "BBF-FLASK-API," Jun. 2022, original-date: 2022-06-07T12:08:24Z. [Online]. Available: https://github.com/UNIC-IFF/BBF-FLASK-API

[28] "XRPL Overview | XRPL.org." [Online]. Available: https://xrpl.org/xrp-ledger-overview.html

[29] R. Dawson, PegaSys, M. Baxter, and PegaSys, "Announcing Hyperledger Besu – Hyperledger Foundation." [Online]. Available: https://www.hyperledger.org/blog/2019/08/29/announcing-hyperledger-besu

[30] "What is Ethereum?" [Online]. Available: https://ethereum.org

[31] "Open Source Blockchain for Currencies & Payments - Stellar." [Online]. Available: https://www.stellar.org/learn/intro-to-stellar

[32] Prometheus, "Prometheus - Monitoring system & time series database." [Online]. Available: https://prometheus.io/

[33] "Grafana: The open observability platform." [Online]. Available: https://grafana.com/

[34] "InfluxDB: Open Source Time Series Database." [Online]. Available: https://www.influxdata.com/