Towards a smarter IoT environment with Ethereum Virtual Machine enabling ledgers

Sandi Gec*, Dejan Lavbič*, Vlado Stankovski*, Petar Kochovski*,
*Faculty of Computer and Information Science, University of Ljubljana, Slovenia
Email: sandi.gec@fri.uni-lj.si

***Abstract*—Distributed Ledger Technology (DLT), from the initial goal of moving digital assets, allows more advanced approaches as smart contracts executed on distributed computational enabling nodes such as Ethereum Virtual Machines (EVM) initially available only on the Ethereum ledger. Since the release of different EVM-based ledgers, the use cases to incentive the integration of smart contracts on other domains, such as IoT environments, increased. In this paper, we analyze the most IoT environment expedient quantitative metrics of various popular EVM-enabling ledgers to provide an overview of potential EVM-enabling characteristics.**

***Index Terms*—Ethereum Virtual Machine, Smart Contract, Consensus Mechanism, transactions.**

I. Introduction

Distributed Ledger Technology (DLT) has been in continuous development since the launch of the first public crypto coin, Bitcoin, in the beginning of 2009. The emerging crypto coins initially mainly differentiate from Bitcoin in general properties such as block generation time, transaction cost, cryptographical approach (e.g. anonymity) and others. Vitalik et al. [1] presented a novel Blockchain system that can be described as an update of the system with Touring complete smart contracts that can be compared to ordinary (notary) contracts or, in other words, a smart contract is a self-executing contract with the terms of the agreement between at least two entities being directly written into lines of code. From the infrastructural point of view, Ethereum is a distributed state machine called Ethereum virtual machine (EVM).

The structure of the software accessible from the internet changed tremendously from single server applications to Cloud, Fog and Edge solutions. Ethereum, in this case, played a crucial role since it enabled executing applications in a distributed way through so-called decentralized Applications (dApps) that are executed and run on the EVM as a part of the Ethereum node instance. The cost of the dApps interactions and smart contract operations on the Ethereum ledger is not always sustainable in transaction-heavy environments (e.g. IoT environments) due to and depends on the ongoing network traffic that may be checked on Eth Gas Station Web application[1]. Due to high transaction costs and speed, developers do not deploy their dApps on the Ethereum network. To exploit the potential of EVM technology, many cryptocurrencies were released to primarily reduce the cost of the transactions, increase the speed and integrate other features (e.g. interoperability mechanisms). Choosing the most suitable one from the EVM-enabling ledger pool is not trivial.

Therefore, this paper analyses the quantitative metrics of various popular EVM-enabling ledgers that might be integrated into IoT-based environments. The study consist of the analysis of the mainnet EVM enabling ledgers. We focused on the qualitative metrics (e.g., consensus mechanisms and network topology) and quantitative metrics available during the paper preparation period in the summer of 2022. From the qualitative properties, we provide an overview that may facilitate the ledger selection based on the use case requirements of any dApp.

The rest of the paper is organised as follows. Section 2 puts our work in the context of other related works. Section 3 describes EVM-enabling ledgers in the context of IoT environments. Section 4 presents quantitative and qualitative results, and section 5 concludes.

II. Related works

The idea behind the decentralized ledger technologies emerged approximately twenty years ago [2]. However, in 2015 the technology evolved from an idea to an applicable technology with the integration of business logic inside the ledgers (i.e. the emergence of smart contracts) in networks such as Ethereum [1] and Tezos [3]. This led towards the massive adoption of the newly developed technology in the design and development of secure and trustworthy decentralized applications. As the blockchain implementation grew and applications evolved to become more decentralized, the Scalability Trilemma became unavoidable. It refers to the trade-off between decentralization, security and scalability that decentralized applications must make when deciding how to optimize the underlying architecture of their blockchain. This has motivated the development of different blockchain layers.

Layer-0 is the data-transfer layer, mainly responsible for data interoperability and integration between blockchains and traditional networks. Polkadot [4] is commonly referred to as a Layer-0 blockchain because its mainnet is a relay chain and is only responsible for the security and interoperability between major parachains. Layer-1 represents the layer that covers the data, network, consensus and activation sub-layers in the blockchain logical architecture. Examples of Layer-1 blockchains are Bitcoin, Ethereum, Binance, Cardano, etc. Though Layer-1 blockchains are massively used in various domains, they all have significant scaling limitations due to

[1]https://ethgasstation.info/

979-8-3503-2028-2/22/$31.00 ©2022 IEEE
DOI 10.1109/CSCI58124.2022.00275

the ever-increasing throughput demand and common network congestion. Layer-2 are perceived as secondary networks constructed on top of existing blockchains and work in parallel or independent of the main chain. The Layer-2 solutions are designed to overcome the main blockchains' transaction speed and scalability issues. Common types of Layer-2 solutions are side chain, Plasma, State Channels, and Rollups, whereas the most famous representatives of this layer are: Polygon, Arbitrium and similar [5].

Cloud-to-Edge environments such as IoT systems are in continuous growth nowadays due to the support of various communication protocols (e.g. IPv6, ZigBee, LoRaWAN and others), heterogeneous services and IoT devices able to communicate through a gateway. Even though such systems are widely production ready in multiple domains, there are many security concerns and research challenges on components such as (micro)services in charge of data management [6]. Our work aims to overcome some of the security concerns through implicit blockchain properties and dedicated distributed dApps empowered by smart contracts. B. Glendenning et al. [7] proposed a hierarchical blockchain framework for providing scalability and security in IoT environments to minimize single points of failure. A broader analysis of different Layer-1 blockchain platforms packed in an evaluation framework was proposed by G. A. Di Lucca et al. [8], which also covers public non-EVM-based ledgers unlike our approach focusing only on public EVM-based ledgers. The practical usability of smart contracts in the health care domain was proposed by A. Maghraby et al. [9], where the authors focused on privacy and controlled exchange of Electronic Health Records.

Given the importance of the smart contract, the different layers and the significant performance difference between them, this paper delivers quantitative metrics and a comparison of various popular EVM-enabling ledgers suitable for various IoT domains. This will significantly simplify selecting the underlying blockchain technology for a decentralized application. We believe this is the first of its kind at the moment of writing.

III. Enabling smart contracts in IoT environments

This section describes the most common development tools and approaches to designing smart contracts. From the perspective of the IoT domain, we present a guideline on where to integrate the EVM-based ledgers to allow interaction of on-chain and off-chain data.

A. Solidity Smart Contracts development tools

Since the execution and interaction of smart contracts on a public Ethereum network, the development does not fully follow standard design patterns. Thus, many smart contract attacks were identified, providing a taxonomy of common programming pitfalls which may lead to vulnerabilities as analysed by N. Atzei et al. Ethereum attacks [10]. Many tools and frameworks were proposed to facilitate the development of smart contracts, minimise vulnerabilities, and even optimise the transaction costs (e.g. function triggering and deployment). *Truffle Suite*[2], which is an excellent development environment, testing framework and asset pipeline for Blockchains using the Ethereum Virtual Machine (EVM) to facilitate the development of SCs. Luu et al. [11] presented the tool *OYENTE*, a novel, symbolic, Web tool which discovers security bugs in Ethereum SCs. *OpenZeppelin*[3] is a framework of reusable SCs for Ethereum and other (EVM and eWASM) Blockchains, which has been widely tested by using many existing Ethereum tokens. Development can be performed on local Integration Development Environments (IDEs) or Web-based EVM enabling Ethereum IDE Remix[4].

B. Integration of smart contracts in IoT environments

Distributed ledgers such as Bitcoin provide primary blockchain monetization use cases. With the introduction of smart contracts as a new concept to offer more complex features, the ledger architecture upgraded from distributed ledger to distributed state machine. These state machines can hold data structures and machine states, which can change from block to block according to a pre-defined set of rules written as a (Solidity) programming script and execute arbitrary machine code.

Moreover, due to the presented characteristics, smart contracts allow improved use cases and trustless interaction among involved human or machine entities. For example, IoT environments ordinarily consist of monitoring components that react to exceeded thresholds based on the monitor data provided by one or multiple sensors. In such cases, the problematic provision of transparent auditing during such events may be improved with an adequate definition of smart contracts. Additionally, smart contracts in IoT environments may provide a facilitated advanced monetization policy between the IoT environment and available stakeholders (e.g. customer, distributor, manufacturer and others). The most relevant interactions among stakeholders and other components in the IoT environment, presented in a smart contract-based IoT architecture, are depicted in Figure 1. An example of a smart contract-enabled IoT environment is intelligent home surveillance. Instead of using the conventional billing process in cases of alarms triggering on-premises human security checks, the entire process can be fully autonomous and transparent. The level of a single point of failure decreased due to the increased distribution of the components. The billing policy is entirely defined in the smart contracts that require payment from the stakeholders (e.g. customer) in cases of emergencies or monitoring actions to provide the payment to other stakeholders (e.g. support, distributor). The interaction with the smart contract in such a system can be performed by: (i) the stakeholder wallet within a Web browser by using the bridge Metamask[5], (ii) in system components with Web3 libraries and (iii) through dedicated Web pages which enable Web3 dedicated smart contract interactions. The example use case data management could

[2]https://truffleframework.com/
[3]https://github.com/OpenZeppelin/openzeppelin-solidity
[4]https://remix-project.org
[5]https://metamask.io/

be further elaborated on on-chain and off-chain interaction, which is not in the scope of the paper.

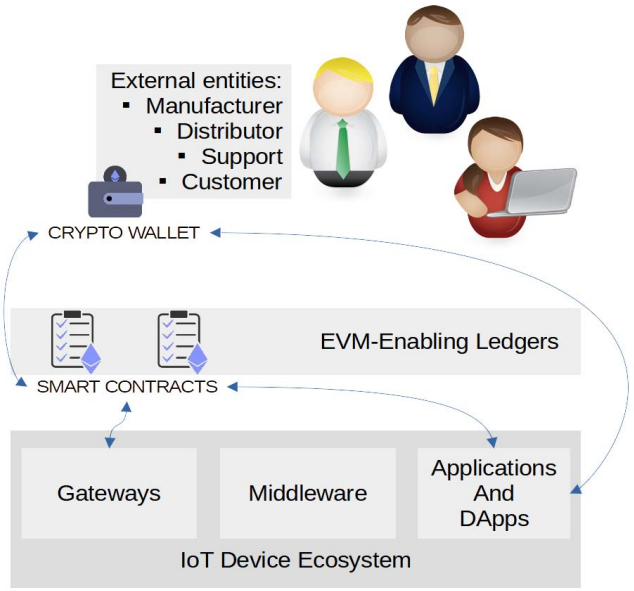

Fig. 1. Proposed IoT architecture depicting key components and smart contract interaction among them

IV. Experimental Analysis

This section focuses on the fundamental properties that may lead developers to deploy smart contracts enabling dApps to certain chains. We focus on critical qualitative and quantitative metrics.

A. Comparison of the top EVM-compatible layers

EVM-enabling ledgers do not differ among them just from the quantitative properties, such as transaction cost and average block time, but also from the qualitative attributes that may be important to consider when deploying dApps on a specific blockchain. It is essential to emphasise the ongoing progress and future directions, which may also need to be more accurate since it is difficult to deterministically plan development in contrast to research work. In practice, it happened to many blockchain solutions that some updates contained vulnerabilities that were not identified due to the lack of testing phase or other development issues (e.g. scalability limitations due to the architectural design of a DLT). Table I summarises the key comparison metrics among selected popular EVM-enabling ledgers.

IoT environments differ by sensor topology and other vital requirements such as power consumption, performance, latency and security. Therefore, the selection and integration of EVM-compatible ledgers should be made based on these requirements. In our work, we compared the most known quantitative metrics as follows:

- **transaction speed** presented as average block time and

TABLE I
Base comparison of popular EVM-based ledgers

Ledger Name	Gas Coin	Consensus Mechanism(s)	Network Topology (Origin)	Mainnet
Ethereum	ETH	PoS	Layer-1	Yes
BNB Chain	BNB	PoS & PoA	Layer-1	Yes
Project Aurora	AURORA	PoS	Layer-2 (NEAR)	Yes
Fantom	FTM	PoS	Layer-1	Yes
Polygon	MATIC	PoS	Layer-2 (Ether)	Yes
Arbitrum	ETH	AnyTrust Guarantee	Layer-2 (Ether)	Yes (beta)
Heco	HT	Hybrid PoS	Layer-1	Yes
Harmony	ONE	PoS	Layer-1	Yes
Gnosis Chain	GNO	PoS	Layer-1	Yes
OKex	OKT	PoS	Layer-1	Yes
KCC chain	KCS	PoS & PoA	Layer-1	Yes
Crypto.com Cronos	CRO	PoS	Layer-2 (Cronos)	Yes
TomoChain	TOMO	PoS Voting	Layer-1	Yes
Emerald ParaTime	ROSE	PoS	Layer-2 (Oasis)	Yes
Avalanche	AVAX	PoS	Layer-1	Yes
Cardano Milkomeda C1	milkADA	PoS	Layer-2 (Cardano)	No
Tron	TRON	PoS	Layer-1	Yes

- **transaction cost** presented as average transaction cost.

The quantitative metrics are measured based on the actual performances analyzed through native blockchain explorer Web services (e.g., Etherscan[6] blockchain explorer for Ethereum). The average block times vary among ledgers from less than 1 second to 15 seconds (see Fig. 2), thus allowing to cover the vast part of IoT performance requirements. Each potential IoT domain should also consider block size and typology of a block (fixed or flexible size) complementary to block generation time to avoid bottlenecks like block overload and, thus, transaction delays.

Another important quantitative metric to be considered is the transaction cost. In the integration of EVM-compatible ledgers, the IoT architecture should be studied to identify all potential interactions (e.g., among stakeholders, monitoring

[6]https://etherscan.io/

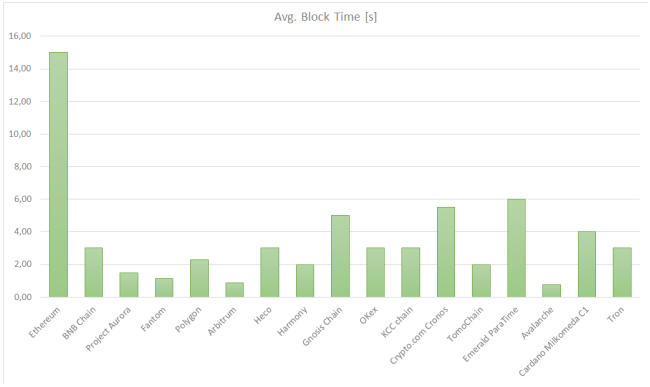

Fig. 2. Average block times presented in seconds of popular EVM-compatible ledgers.

triggers, etc.). Production-ready IoT environments should be sustainable in the long term.

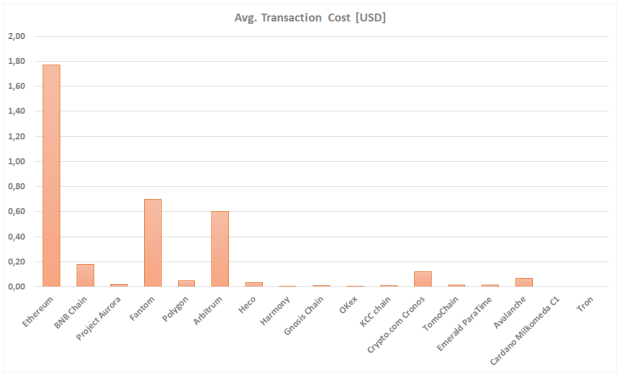

Fig. 3. Average transaction cost presented in USD currency of popular EVM-compatible ledgers.

V. Conclusion

Blockchain has become a leading technology that plays an essential role in the global economy and digitization process. Given that the technology is still in its early phase, there are many challenges to be addressed. Balancing between decentralization, security, and scalability is an important decision when designing decentralized applications requiring high-quality service and experience. Therefore, from the emerging variations of EVM-enabling ledgers, it is possible to select an underlying blockchain technology among the wide variety of possible choices to enable sustainable smart contracts in IoT environments.

In this paper, we presented the popularity of EVM-enabling technology and the possibility of integrating EVM-based smart contracts into most IoT environments. By analyzing the ledgers' quantitative and qualitative metrics, we provide guidelines to facilitate selecting developers' specific use case solutions packed into dApps. In our future work, we plan to expand the analysis to an empirical study by deploying and triggering use-case-specific smart contracts on testnet environments that most likely match the mainnet ledgers. Moreover, we will analyze the potential of integrating smart oracles approaches to increase interoperability in IoT environments.

Acknowledgment

The research and development reported in this paper have received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 957338 (ONTOCHAIN: Trusted, traceable and transparent ontological knowledge on blockchain) and from the Research Agency of the Republic of Slovenia under the research programme P2-0426 Digital Transformation for Smart Public Governance 1/1/22 - 12/31/27.

References

[1] "Buterin, vitalik, et al. "ethereum white paper", updated september 30, 2015," https://github.com/ethereum/wiki/wiki/White-Paper, accessed: 2022-05-30.

[2] S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, vol. 4, p. 2, 2008.

[3] L. Goodman, "Tezos: A self-amending crypto-ledger position paper," *Aug*, vol. 3, p. 2014, 2014.

[4] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White Paper*, vol. 21, pp. 2327–4662, 2016.

[5] ""ethereum development documentation", updated june 17, 2022," https://ethereum.org/en/developers/docs/scaling/, accessed: 2022-05-30.

[6] M. Driss, D. Hasan, W. Boulila, and J. Ahmad, "Microservices in iot security: Current solutions, research challenges, and future directions," *Procedia Computer Science*, vol. 192, pp. 2385–2395, 2021, knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921017440

[7] B. Glendenning, R. Kiefer, and A. Patel, "Ziggurat: A framework for providing scalability and security in iot blockchains," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021, pp. 1548–1550.

[8] G. A. Di Lucca and M. Tortorella, "Towards a framework to compare blockchain platforms for smart contracts," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021, pp. 1931–1937.

[9] A. Maghraby, A. Numan, A. A. Mashi, A. Aljuhani, R. Almehdar, and N. Abdu, "Applied blockchain technology in saudi arabia electronic health records," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021, pp. 1250–1254.

[10] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts sok," in *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*. New York, NY, USA: Springer-Verlag New York, Inc., 2017, pp. 164–186.

[11] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 254–269. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978309