Practical Limitations of Ethereum's Layer-2

RAY NEIHEISER[1], GUSTAVO INÁCIO[2], LUCIANA RECH[2], CARLOS MONTEZ[3], MIGUEL MATOS[4] and LUÍS RODRIGUES[4]

[1]Institute of Science and Technology Austria, Klosterneuburg, Austria
[2]Universidade Federal de Santa Catarina, PPGCC, Florianópolis, Brazil
[3]Universidade Federal de Santa Catarina, PPGEAS, Florianópolis, Brazil
[4]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

Corresponding author: Ray Neiheiser (e-mail: ray.neiheiser@protonmail.com).

ABSTRACT Most permissionless blockchains inherently suffer from throughput limitations. Layer-2 systems, such as side-chains or Rollups, have been proposed as a possible strategy to overcome this limitation. Layer-2 systems interact with the main-chain in two ways. First, users can move funds from/to the main-chain to/from the layer-2. Second, layer-2 systems periodically synchronize with the main-chain to keep some form of log of their activity on the main-chain - this log is key for security. Due to this interaction with the main-chain, which is necessary and recurrent, layer-2 systems impose some load on the main-chain. The impact of such load on the main-chain has been, so far, poorly understood. In addition to that, layer-2 approaches typically sacrifice decentralization and security in favor of higher throughput. This paper presents an experimental study that analyzes the current state of Ethereum layer-2 projects. Our goal is to assess the load they impose on Ethereum and to understand their scalability potential in the long-run. Our analysis shows that the impact of any given layer-2 on the main-chain is the result of both technical aspects (how state is logged on the main-chain) and user behavior (how often users decide to transfer funds between the layer-2 and the main-chain). Based on our observations, we infer that without efficient mechanisms that allow users to transfer funds in a secure and fast manner directly from one layer-2 project to another, current layer-2 systems will not be able to scale Ethereum effectively, regardless of their technical solutions. Furthermore, from our results, we conclude that the layer-2 systems that offer similar security guarantees as Ethereum have limited scalability potential, while approaches that offer better performance, sacrifice security and lead to an increase in centralization which runs against the end-goals of permissionless blockchains.

INDEX TERMS Distributed Ledgers, Blockchain, Layer-2, Ethereum

I. INTRODUCTION

Blockchain is a distributed ledger that is maintained by a potentially large set of processes in a fully decentralized manner. Bitcoin [1] is a pioneer cryptocurrency system that uses a blockchain to keep track of financial transactions and prevent double-spending. An important advantage of the blockchain is that participants are not required to trust any centralized authority to maintain the ledger. Instead, all participants engage in a distributed consensus protocol to decide the order by which transactions are recorded on the ledger. As there are many other applications, besides cryptocurrencies, that may benefit from a distributed ledger, proposals to expand the capabilities of the blockchain soon followed. Ethereum [2] is a successor of Bitcoin that popularized the concept of smart contracts [3]. Smart contracts are deterministic computer programs that may be invoked when a transaction is recorded on the blockchain, affecting the outcome of the transaction. This allows to support more complex interactions among users, opening the doors to a wide range of potential applications [4]. For instance, a smart

contract may stipulate that a transfer of funds from one user to another only takes effect if some predicate is true when the contract is executed.

The consensus protocols used in Bitcoin and its variants, such as Ethereum, are *permissionless*, i.e., they do not enact any constraints on the set of users that participate in consensus: any user may join the system at any point, and start participating in the consensus protocol. This is very powerful, as it makes it hard for a coalition of users to control the blockchain (e.g., to decide which transactions are recorded and which are not). Unfortunately, all known protocols that are able to solve fault-tolerant permissionless consensus among a large set of users are either very expensive and very slow [5] or partially sacrifice resilience. As a result, most permissionless blockchains inherently suffer from throughput limitations. For instance, Bitcoin executes roughly 5 transactions per second and Ethereum less than 15 transactions per second [5].

In order to cover the costs of operating the system, users usually have to pay a fee per transaction. Usually, the fee has no fixed value: users may declare how much they are willing to pay for the transaction. Since the throughput of the system is limited, not all pending transactions may be included in a given block. Therefore, the higher the value the user is willing to pay for the transaction, the more likely and quickly it is eventually included in the blockchain. As a result, the throughput limitations paired with the high demand to transact on the blockchain drive the minimum fee a user has to pay per transaction. This led to the emergence of several approaches that often trade security, decentralization, or both in order to improve performance and reduce costs [6].

There are several avenues to circumvent the throughput limitations of existing permissionless blockchain systems. The first is, naturally, to design more efficient permissionless consensus protocols and more efficient mechanisms to maintain the blockchain and execute smart contracts. For instance, Ethereum is expected to deploy, at some point in the future, a number of upgrades that aim to improve the system's scalability (known as "Ethereum 2.0") [7]. However, even if these efforts are successful, it is unlikely that they can boost the performance to a point where permissionless blockchains can compete with logically centralized systems [8]. For instance, the VISA system is able to execute approximately 1700 transactions per second with peaks of up to 24000 [8]. Another avenue consists in offloading transaction processing from the blockchain to an outside system, the so-called layer-2 or off-chain systems [9].

Layer-2 systems usually interact with the main-chain in two ways. The first is when users move funds from the main-chain to the layer-2 and vice-versa. This typically implies locking funds on the main-chain before transactions can be executed on the layer-2, which is required to prevent a user from using the same funds to execute concurrent transactions on different layer-2 systems. Second, layer-2 systems need to keep some form of log of their activity on the main-chain. This log is key to enforcing the security guarantees

of the layer-2 system. Different layer-2 systems use different techniques to log their state on the main-chain, materializing different tradeoffs between the security guarantees offered to the users and the load imposed on the main-chain.

Given the fast emergence of many layer-2 systems, with different security and performance tradeoffs, it becomes difficult for users, researchers, and practitioners to assess the merits of the competing approaches.

In this paper, we report the results of a systematic study on the security and performance properties of existing layer-2 systems. We restrict ourselves to the layer-2 approaches on top of Ethereum, which is one of the most popular and flexible blockchain due to the native support for smart contracts. We selected six popular Ethereum layer-2 projects, based on their transaction volume, namely Polygon [10], Optimism [11], Arbitrum [12], ZKSync [13], Ronin [14], and Gnosis (formerly xDAI) [15], and carefully analyzed their designs. Moreover, we conducted a one year study, encompassing the full year of 2021, where we have collected data about their performance and their impact on the Ethereum main-chain. This allows us to assess the current load that these projects impose on Ethereum and to understand their potential to scale the Ethereum ecosystem in the long-run.

While there are previous studies on Ethereum's layer-2 systems [17], [25], [28], they present several limitations. Several previous works study the different algorithms used by layer-2 systems [25]–[27] but none address their actual performance in practice. Others [19] compare concrete systems from the Ethereum ecosystem but do not offer an experimental assessment of their behavior. The work of Chemaya *et al.* [28] presents an experimental study focused on Polygon, but does not cover the scalability perspective. To the best of our knowledge, this paper is the first systematic study that conducts a year-long experimental study of the most popular, in terms of transaction volume, Ethereum layer-2 systems, and studies the current and long-term impact they have on the main-chain. Table 1 summarizes how our work compares with, and complements, previous studies.

Our analysis shows that the impact of a given layer-2 system on the main-chain is the result of both technical aspects (how its state is synchronized with the main-chain) and user behavior (how often users decide to transfer funds between the layer-2 and the main-chain). Based on the data we have collected, we hypothesize that, without efficient and secure mechanisms that allow users to transfer funds directly between two layer-2 systems, current layer-2 systems will not be able to scale Ethereum effectively to competitive levels, regardless of their technical merits. While some layer-2 proposals claim to be able to process thousands of transactions per second, this considers the layer-2 system in isolation and without taking into account the synchronization cost with the main-chain. As our results show (§IV), and based on the current workload characteristics, achieving these throughput levels would put a load on the main-chain higher than what it can accommodate. Furthermore, our analysis corroborates previous findings [6] that show a deterioration of security

TABLE 1. Overview of existing Ethereum's Layer-2 studies.

Work	Compares different algorithms	Compares concrete projects	Number of systems evaluated experimentally	Addresses the impact on the Main-chain
Gangwal, et al. (2022) [25]	Yes	No	0	No
Sguanci, et al. (2021) [26]	Yes	No	0	No
Jourenko, et al. (2017) [27]	Yes	No	0	No
Gluchowski (2020) [19]	No	Yes	0	No
Chemaya, et al. (2022) [28]	No	Yes	1	No
This work	Yes	Yes	6	Yes

guarantees of decentralized applications in the Ethereum ecosystem in exchange for increased performance due to the scalability limitations of existing layer-2 systems.

The rest of the paper is organized as follows. §II introduces the background on blockchains and layer-2 systems required to understand our study. §III describes the layer-2 systems covered in this paper and discusses the tradeoffs imposed by their designs. §IV introduces the methodology used to collect the experimental data, and discusses the obtained results. Finally, §V concludes the paper.

II. BACKGROUND

This section starts by introducing the main general concepts underlying a blockchain such as Ethereum, and then details the building blocks used in layer-2 approaches.

A. BLOCKCHAIN

The blockchain is a linked list of blocks maintained by a network of nodes. Each block holds a set of transactions and metadata. The next block to be added to the blockchain is decided through a permissionless consensus protocol, such as Nakamoto consensus and its variants in the case of Bitcoin and Ethereum [1].

Nakamoto consensus works as follows. Any participant can propose the next block to be added to the blockchain. However, in order to do so, it must present a *Proof of Work*, i.e., it must first solve a cryptographic-puzzle, that takes a random period of time to solve – a process known as mining. If a miner solves the crypto-puzzle, it propagates the block in the network. A miner that receives a valid block adopts that block (and abandons its own attempt of producing a competing block). When there are no concurrent proposals, the (single) block proposal is quickly disseminated in the network and adopted by all participants, that subsequently move to propose the next block. If two miners concurrently propose a block, a *fork* in the chain occurs: some participants will adopt one proposal and other participants may adopt the other. However, eventually one of the branches of the fork will grow faster than the other, becoming the longest chain. A miner that realizes it is no longer working on the longest chain abandons the shorter chain and adopts the longest chain. Transactions executed on the shorter chain are invalidated and need to be re-executed on the longer chain. Due to this reason, blockchains based on Nakamoto consensus do not offer deterministic *finality*: a transaction added to

the chain can always be reverted if a longer chain is found. In practice, the probability of a transaction being reverted quickly diminishes as time passes. Thus, transactions are considered definitive after some finite number of blocks have been appended after their own block on a given branch.

The miner of the winning block is rewarded with some cryptocurrency (partially newly created, partially sourced from transaction fees). This reward structure serves a twofold purpose: it incentivizes miners to participate in the system, and it discourages miners from proposing invalid or empty blocks.

Since Proof of Work is very energy intensive, more efficient algorithms have been proposed. As we will further detail in §II-D, some layer-2 systems sidestep the energy cost and throughput limitations of Proof of Work by relying on alternative consensus mechanisms such as Proof of Stake (PoS) and Proof of Authority (PoA). Briefly, in Proof of Stake, a deterministic algorithm chooses the next miner based on the quantity of cryptocurrency this entity is holding. In Proof of Authority, a trusted set of validators is chosen ahead of time (e.g. large companies) and a deterministic algorithm then rotates through this set of validators.

B. SMART CONTRACTS

While Bitcoin focuses mostly on token transfers, i.e., cryptocurrency, Ethereum introduced the notion of smart contracts. As noted before, smart contracts are deterministic computer programs that can be executed when a transaction is included on the blockchain, affecting the outcome of the transaction. These contracts are implemented with the help of a Turing complete programming language and can broaden the applicability of blockchain systems.

Consider for instance that two parties want to exchange digital assets. As such, they may require an entity to hold onto the payment of either party, until both parties submitted the agreed upon quantity, and then release the assets simultaneously. Classically, this was done through a trusted intermediary but, in the context of blockchains, this can be implemented through a smart contract, avoiding the need for a trusted third party.

Smart contracts open the door for a wide range of novel applications but can also make the blockchain more vulnerable to denial-of-service attacks. A malicious user may submit a smart contract with an infinite loop, and hence prevent the entire system from progressing. The ability to run arbitrary

programs in adversarial environments, coupled with the well-known halting problem [18], requires a mechanism to bound the execution time of each transaction.

To address this problem, Ethereum introduced the notion of gas, a processing fee system that the issuer of a transaction must pay. The gas cost is proportional to the computational complexity and storage requirements of the transaction. Hence, smart contract transactions are typically more expensive than cryptocurrency transactions. Interestingly, the gas cost is not fixed. Instead, it depends on the current supply and demand and, as such, users can bid on the price to pay for each gas unit to ensure their transactions are eventually processed.

C. THROUGHPUT LIMITATIONS

As discussed above, blockchains based on proof of work, such as Ethereum, require miners to solve a crypto-puzzle in order to produce a block. This puzzle must be hard enough to ensure that the chances of having two participants concurrently proposing different versions of a block are very small. How hard the crypto-puzzle needs to be depends on several factors, such as: the expected number of miners, the estimated power of each miner, and how fast the network disseminates a new block, among others. In Ethereum, this is configured such that a new block is generated approximately every 13 seconds. Moreover, each block has a maximum size. On Ethereum it is possible to include approximately 200 transactions, on average, in a single block, which yields a throughput of approximately 15 transactions per second. This throughput is very small for most applications.

Furthermore, in Ethereum, the negative effect of low throughput is amplified by the gas mechanism. Given that the throughput is small, users are incentivized to pay higher gas prices. This not only means that the system's throughput is very small, but also that the cost of every transaction is very high (e.g., at the time of this writing, more than $2, for Ethereum transfers, more than $5 for ERC-20 transfers and more than $20 for trading on a decentralized exchange). Again, for many applications, this price is too high to make it economically appealing.

D. LAYER-2

To overcome these limitations, several layer-2 systems have been proposed. The key idea underlying all layer-2 systems is to deploy a third-party service or system that will process the bulk of transactions outside the main-chain. A smart contract deployed on the main-chain mediates the interactions between the main-chain and the layer-2 system, allowing users to deposit funds in that smart contract and receive tokens in the layer-2 system that can be used in the services it provides. To withdraw funds from the layer-2 system, the user issues a transaction (on the layer-2) to a special address. In turn, the layer-2 state is synchronized with the main-chain at regular checkpoint intervals through the smart contract. Thus, after the layer-2 system has synchronized the next checkpoint with

the main-chain smart contract, users may collect their funds on the main-chain, if any [9].

There are several competing approaches to implement these layer-2 systems. One of the most popular and secure approaches are *Rollups*, which can be further classified into *Optimistic Rollups* and *Zero Knowledge Rollups (ZK Rollups)*. In these approaches, the state on the layer-2 system is maintained by one or more nodes, known as Aggregators, which run a system-specific protocol.

Optimistic Rollups require the layer-2 service to leave a security deposit on the main-chain smart contract. For each transaction in the layer-2 system, the resulting state and raw transaction data are published on the main-chain. Thus, Optimistic Rollups move the cost of computation (i.e., transaction processing) and user interactions to the layer-2 but keep storage on the main-chain. Therefore, there is still a linear relationship between transactions on layer-2 and storage usage on the main-chain. The main advantage of this approach is that, because the raw transaction data is available on the main-chain, any third party can verify the correctness of the resulting state, report any conflict and slash/claim the security deposit if the state proves to be invalid. This is done by storing the security deposit in a smart-contract on the main-chain alongside the raw transaction data and layer-2 state. If a user or any third party detects misconduct, they can invoke the smart-contract which executes the raw transaction data and compares it to the published layer-2 state. If the computation reveals that the raw transaction data does not lead to the published layer-2 state, the security deposit is slashed and a part of it is rewarded to the party that uncovered the misconduct. The downside is that users have to wait for a long period (7 days on average) to allow anyone to verify the correctness of the published state before they can move their funds back to their main-chain account. In summary, Optimistic Rollups impose three main costs on the main-chain: a transaction depositing funds in the main-chain smart contract, the publication of the state and raw transaction data on the main-chain by the layer-2 system, and a transaction to withdraw the user's funds.

Zero Knowledge Rollups do not require the publication of the raw transaction data on the main-chain. Instead, the Aggregator computes a zero knowledge proof of the layer-2 state, and submits it together with the resulting state to the main-chain where it is verified by the main-chain smart-contract. As a result, the proof size is constant and not linear to the number of transactions as in Optimistic Rollups. To withdraw funds, users compute a zero knowledge proof of their layer-2 state which they submit in a transaction to the main-chain smart contract. Similar to Optimistic Rollups, ZK Rollups also have three main cost components in the main-chain. While their storage cost is much smaller than of Optimistic Rollups, the computation cost on the main-chain is higher because the verification of Zero Knowledge proofs is computationally expensive. As the computation of this proof is significantly more expensive than for Optimistic Rollups, fund withdrawals have a much higher cost for the user. Due to

this, withdrawals are usually executed in batches by the layer-2 operator, which significantly reduces the cost but results in additional client-side latency (up to 1h). Nonetheless, the client can always explicitly request the proof computation, which results in lower latency but higher costs.

By publishing the raw transaction data, or the corresponding zero knowledge proof, Rollups offer good security guarantees as misbehavior of the layer-2 system can either be detected and punished by slashing the initial security deposit (Optimistic Rollups) or are directly detected and prevented by the main-chain smart contract (ZK Rollups).

One important aspect that is often overlooked is that the security of these approaches relies on the correctness of the smart contract implementation. Therefore it is fundamental that the implementation is publicly available as open-source, such that it can be verified by third parties, and also to confirm that the compiled smart contract that runs on the main-chain matches the open-source implementation.

Side-chains are an alternative approach to implement layer-2 systems. Side-chains are loosely coupled to the main-chain [9]. The key idea is to have a parallel blockchain that tracks the main-chain but runs completely independently and periodically checkpoints its state on the main-chain — this mechanism is known as a two-way peg. These checkpoints consist of a digest of the side-chain state (a Merkle Tree Root) and contain enough information to allow the main-chain smart contract to verify if a user has funds to withdraw. The state of the side chain is maintained with the help of a consensus algorithm usually based on Proof of Stake or Proof of Authority. Regardless of the consensus algorithm, the number of participants is much smaller than the number of miners on the main-chain, and, therefore it is substantially easier to attack the side-chain as the number of resources necessary to overtake the side-chain is significantly smaller compared to the main-chain. Proof of Authority side-chains, in particular, require trust in a small and limited group of validators compared to the trustless approach of Ethereum. In summary, side-chain based approaches trade security guarantees for better performance when compared to the Proof of Work algorithm used in Ethereum.

An approach known as Plasma [9] improves on the security guarantees of side-chains by requiring the side-chain to leave a security deposit on the main-chain, and publishing the raw transaction data on the main-chain, similarly to Optimistic Rollups. In case of misbehavior, the procedure to recompute the state and slash the security deposit is the same as with Optimistic Rollups. Therefore, Plasma withdrawals also take 7 days to process but are backed by main-chain security guarantees. However, Plasma is incompatible with most smart-contract operations on the side-chain, and, as such, only supports a limited set of applications [19].

Notably, one of the main reasons these solutions have an impact on the main-chain performance comes from the necessity to go through the main-chain to deposit and withdraw funds to/from the layer-2. While it is possible to use a centralized service to perform layer-2 deposits and with-drawals at a reduced cost, as a result, this further degrades the security guarantees. The risks of these approaches become obvious when considering the number of reported exchange hacks [20], that have resulted in hundreds of millions of dollars of lost user funds.

Table 2 summarizes the characteristics of the different layer-2 approaches in regards to security, smart contract support, withdrawal time and maintenance costs on the main-chain. There is no approach that excels in all metrics. Rollups offer good security guarantees and smart contract support but have a high main-chain maintenance cost. Optimistic Rollups have a significantly longer withdrawal time and require more storage space on the main-chain, but, in turn, consume significantly less computational resources of the main-chain compared to Zk Rollups. Plasma also offers good security guarantees, but its very limited smart contract support makes it only viable for a small subset of applications. Side-chains are fully independent from the main-chain and therefore have a low main-chain maintenance cost and offer a good scalability potential. However, the provided security of this layer-2 approach is significantly weaker than the other alternatives. Finally, and regardless of the security guarantees offered by each approach, an application that runs on layer-2 has to trust the continuous availability of the layer-2 provider. This is especially noteworthy for the layer-2 systems that are run by a single entity.

III. SELECTED APPROACHES

Due to the very large number of layer-2 systems, it is infeasible to assess them all in detail in the scope of this work. We selected six of these systems that cover the different approaches discussed in §II (and, for each approach, we selected those with a large volume of main-chain transactions). The selected systems are: Polygon [10], Optimism [11], Arbitrum [12], ZKSync [13], Ronin [14], and Gnosis [15].

Table 3 summarizes the main characteristics of the selected approaches, which we discuss next. The latency, throughput, and finality values presented in the table have been taken from the values reported by each system (extracted from their respective white papers). The "TX lat." column captures the time it takes until a given transaction is accepted by the validators. As Rollups are mostly centralized services, the time it takes for a client to receive a positive or negative response is dominated by the roundtrip time (RTT). Meanwhile, side-chains like Gnosis and Polygon require their side-chain consensus to terminate until a client is able to verify the success of their operation. The "Max tput." column captures the number of transactions per second each of the approaches claims to be able to process, solely based on the side-chain capabilities and disregarding their impact on the main-chain. The "L2-Approach" column describes the layer-2 approach taken by each system, as per the discussion on §II-D. Next, the "Consensus" column captures the consensus algorithm used by each layer-2 system. While, in the case of Side-Chains this is usually Proof of Authority (PoA) or Proof of Stake (PoS), all current services that leverage Rollups are

TABLE 2. Layer-2 design space and comparative characteristics.

Approach	Main-Chain security	Smart Contract support	Withdrawal time	Maintenance cost
Optimistic Rollups	Yes	Yes	7 days	High
Zk Rollups	Yes	Yes	10min - 1h	Medium
Plasma	Yes	Limited	7 days	Low
Side-Chain	No	Yes	10min	Low

centralized, and as such do not have a consensus mechanism. The "Finality" column describes the maximum delay until either a fund withdrawal transaction is processed and/or the side-chain state is irrevocably persisted on the main-chain. Finally, and for completeness, the "API" column indicates the API URL of each system.

Polygon is a Proof of Stake (PoS) Plasma side-chain and is the best performer in terms of throughput. It produces, on average, a block every 2 seconds and claims to be able to reach a maximum throughput of 65,000 tps. Depending on the chosen layer-2 approach, withdrawing tokens from Polygon takes between 20 minutes and 3 hours for Proof of Stake backed withdrawals, and 7 days for Plasma based withdrawals. Thus, Polygon actually offers users two modes of interaction: one through Plasma and another through Proof of Stake.

Optimism relies on Optimistic Rollups with a centralized Aggregator, and as such, is able to approve transactions instantly (bounded by the RTT). However, at the current state, it supports at most 200 tps, and, similarly to Polygons Plasma approach, withdrawals take 7 days to process.

Arbitrum, similarly to Optimism, also relies on Optimistic Rollups with instant transaction approval. However, the authors claim to support up to 4,500 tps, a much higher theoretical throughput than Optimism.

ZKSync is the only system relying on Zero Knowledge Rollups and a centralized Aggregator and hence it is able to approve transactions instantly. ZKSync claims to be able to scale up to 3,000 tps. In terms of finality, it takes approximately 10 minutes to compute the fraud proof on top of awaiting the finality of the Ethereum block it is included in. Nonetheless, to reduce the withdrawal costs, withdrawals are usually automatically computed by Zksync in batches, taking on average 1 hour.

Gnosis is a Proof of Authority (PoA) side-chain that produces a block every 5 seconds. It claims to offer a max throughput of 90 tps and, as it fully relies on the side-chain approach, offers instant finality through Casper [7].

Finally, *Ronin* is a Proof of Authority side-chain with a small pre-selected set of validators. Ronin, is a more recent system at the time of this writing, and as such, the authors have not yet officially disclosed the maximum theoretical throughput, nor the expected finality.

In summary, each system has significantly different characteristics in terms of security, latency, throughput, and finality. The main tradeoffs offered by each approach are summarized in Table 2, following the discussion of §II-D, while Table 3 presents the characteristics of each concrete layer-2 implementation. One can observe a large gap between the throughput potential of side-chain approaches like Polygon, which claims to be able to scale up to 65,000 tps, and approaches using Rollups, that estimate an upper limit of 4,500 tps. This is in line with previous reports [19] and, as discussed in § II, comes with an inherent performance versus security tradeoff. Despite relying on the main-chain for security, and hence offering better guarantees than side-chains, all current systems that use Rollups are centralized providers and hence can be subject to downtimes and reduced availability. Furthermore, there are also large differences between systems using the same approaches. For instance, despite both being side-chains, Gnosis offers a fraction of the potential throughput of Polygon. This is intentional, according to Gnosis developers, who state that they intentionally offer a lower max throughput in order to avoid growing the blockchain state too quickly [16]. In the following section, we study the impact these tradeoffs have on the main-chain.

IV. MEASUREMENTS

Our main goal is to assess the load that layer-2 systems impose on the main-chain and understand their scalability potential in the long-run. To assess this, we conducted a 12-month study, from January 1, 2021 to December 31, 2021, and collected, for each selected system, the following performance indicators:

- *Throughput*: the daily average the number of transactions per second;
- *Main-chain load*: the daily average load imposed by the layer-2 system on the main-chain, measured as the fraction of gas consumption by layer-2 transactions that appear on the main-chain over the maximum gas limit per block;
- *Maintenance cost*: fraction of the total layer-2 gas consumption on the main-chain excluding withdrawal and deposit requests.

A. METHODOLOGY

To collect data, we started by obtaining the layer-2 smart contract addresses for each of the selected systems. Next, we traversed the full Ethereum blockchain for the given period to determine, from the set of all main-chain transactions, which ones correspond to layer-2 deposits, withdrawals, checkpoints, and other maintenance operations. This allows us to measure the main-chain load and the maintenance cost. To obtain the throughput, which is inherent to each layer-2 system, we consulted the respective APIs and downloaded their individual states (each block and its respective meta-

TABLE 3. Selected layer-2 systems. RTT means that the transaction latency is given by the network round trip time.

System	TX lat.	Max tput.	L2-Approach	Consensus Alg.	Finality	API
Polygon [10]	2s	65, 000 tps	Side-Chain + Plasma	PoS	3h (PoS), 7 days (plasma)	https://polygonscan.com/chart/tx
Optimism [11]	RTT	200 tps	Optimistic Rollups	None	7 days	https://optimistic.etherscan.io/chart/tx
Arbitrum [12]	RTT	4, 500 tps	Optimistic Rollups	None	7 days	https://arbiscan.io/chart/tx
ZKSync [13]	RTT	3, 000 tps	ZK Rollups	None	1h	https://zksync.io/api/v0.1.html
Ronin [14]	RTT	-	Side-Chain	PoA	-	https://proxy.roninchain.com/free-gas-rpc
Gnosis [15]	5s	90 tps	Side-Chain	PoA	5s	https://xdai-archive.blockscout.com

data) of the selected time period. The only layer-2 system that, at this time, does not offer a public API node to collect the data is Ronin. In this case, to collect the data, we did setup a read-only node, let it synchronize with the Ronin network, and then processed the collected data similarly to what we did for the other systems.

Next, we detail how each of the metrics of interest has been extracted from the logs.

- *Throughput:* To obtain the throughput of each layer-2 system, in transactions per second (tps), we relied on the API provided by each layer-2 system. For each system, we obtained the blocks produced during our period of interest, and then counted the number of transactions present in each block. Following that, we ordered the entries by timestamp to calculate the number of transactions on each given day.
- *Main-chain Load:* As in Ethereum there is no fixed maximum block size, but instead a maximum gas limit that can be adjusted by miners, we use the latter as the maximum potential load per block. With this in mind, we have computed the layer-2 load imposed on the main-chain by considering the total gas spent by layer-2 transactions over the maximum possible gas limit per block. During our one year measurement period, there were two major adjustments to the gas limit: one in April (Berlin hardfork [21]) that increased the miner defined gas limit from 12.5M units to 15M units and another one in August (London hardfork [22]) that set a soft cap of 15M units and a hard cap at 30M units. If the soft cap is reached, the gas price is automatically increased for the following blocks. Even though the soft cap was regularly surpassed in our observations, to simplify the model, we consider the maximum gas limit to be 12.5M units in the beginning of our observations (from January 1, 2021 until April 2021), and then 15M units for the rest of the observation period (from April 2021 until December 31, 2021).
- *Maintenance Cost:* The maintenance cost provides insight on the overhead of operating a given layer-2 system. The cost is given, for each layer-2 system, by the ratio between deposit and withdrawal transactions on the main-chain and the total transactions for that system on the main-chain. Note that, as different transactions can have different storage and processing costs, we calculate the cost in terms of gas rather than number of transactions as this is a more realistic approximation to the real financial cost of running layer-2 systems.

B. ANALYSIS

We now present and discuss the obtained results considering the metrics defined above.

First, we analyze the throughput of layer-2 systems, as depicted in Figure 1, where each datapoint represents the average throughput on a given day for each system. During our observation period, the sum of all layer-2 systems reached peaks of over 100 tps, stabilizing at around 90 tps close to the end of the observation period. The overall throughput is mostly dominated by Polygon, which contributes with over 80% of all layer-2 throughput, followed by Ronin averaging at around 10 tps and Gnosis at up to 6 tps. Thus, side-chains contribute the vast majority of the overall layer-2 throughput while Rollups, outside of rare peaks, contribute less than 1 tps each. We can observe that the type of rollup that is used has insignificant influence on the observed throughput.

The results in terms of main-chain load, depicted in Figure 2, offer some interesting insights on the relative cost of each approach. The results show how much each of the layer-2 systems consumed of the daily available resources on Ethereum. As one can observe, layer-2 systems impose a peak in load of over 10% of the main-chain capacity. However, in the last months of the observation period, this dropped to a daily average of around 2%. We also observe that, over the majority of the time span, the actual main-chain load is also dominated by the layer-2 systems with the highest layer-2 throughput (namely Polygon and Ronin). This is interesting as both projects are side-chains with relatively little coupling to the main-chain when compared to Rollups.

Nonetheless, over the last months of the analysis period, the main-chain load of both solutions reduced significantly even though the overall throughput remained mostly constant. We conjecture that this cost decrease is the result of the support Ronin and Polygon received from centralized exchanges which started to allow deposit and withdraw funds directly to those systems without having to go through the main-chain: Polygon support was announced in July and Ronin support was announced in early September, aligning with the decrease in the main-chain load [23], [24].

We can also observe a strong increase over time in the load Rollups imposed on the main-chain, namely Optimism, Arbitrum and ZkSync. Finally, the load imposed by Gnosis is negligible. This is the case as Gnosis has much lower coupling than Polygon (which offers Plasma support) and significantly lower coupling than Rollups. As such, it is much more comparable to Ronin in terms of coupling to the main-chain. We analyze this behavior in detail in the next section
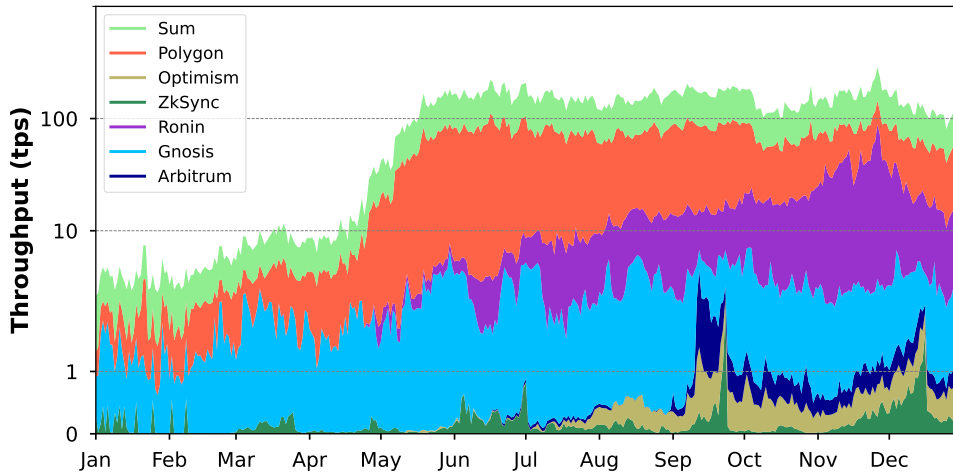
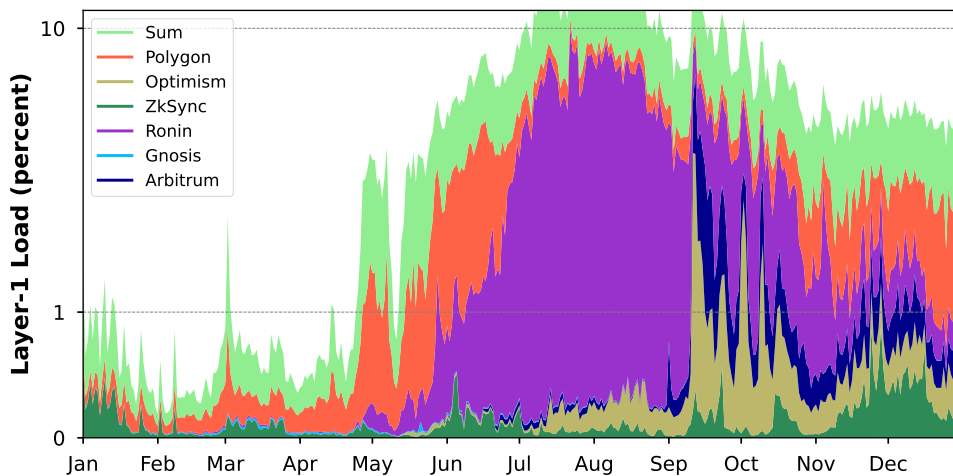**FIGURE 1.** Average layer-2 daily throughput throughout 2021.

FIGURE 2. Average layer-2 daily main-chain load throughout 2021.

by studying where the differences in terms of cost stem from.

The results for the maintenance costs are depicted in Figure 3. As expected, most projects exhibit a very high maintenance overhead in the early phases since, while the adoption by the users is still low, they have to distribute fixed maintenance costs over a small set of transactions. In the case of Polygon, in the early months of the observation period, the maintenance overhead was over 50% with peaks of up to 75%, but during most of the year 2021 it stabilized at around 10% with little variance. Rollups like Optimism, Arbitrum and ZKSync, on the other hand, are different as, due to the stronger coupling to the main-chain, the maintenance overhead is consistently high. Each of these approaches displays a maintenance overhead of above or around 50% (i.e. 50% of the cost the layer-2 system exhibits on the main-chain is related to maintenance and not deposit/withdrawals). While Polygon is also a side-chain, when compared to Gnosis and Ronin, it offers Plasma functionality, which results in

stronger coupling and, as such, also results in a higher maintenance overhead. Gnosis' maintenance load is consistently low as it is one of the oldest side-chains and displays a very consistent throughput level over the recorded period.

Ronin, however, only shows a high maintenance cost in the early phase and then, eventually, reaches a level similar to Gnosis due to the very low coupling required. As such, the decrease of load Ronin imposes on the main-chain must stem from a decrease of deposit/withdrawal operations. We discuss the potential reasons for this in §V.

C. LAYER-2 COST PER TRANSACTION

To better understand the tradeoffs between throughput and cost, we now study how much resources per transaction each of the systems consumes compared to an average Ethereum transaction. As such, we first divide the average number of transactions (per day) by the total gas consumption (per day) to estimate the average cost of an Ethereum transaction. Next,
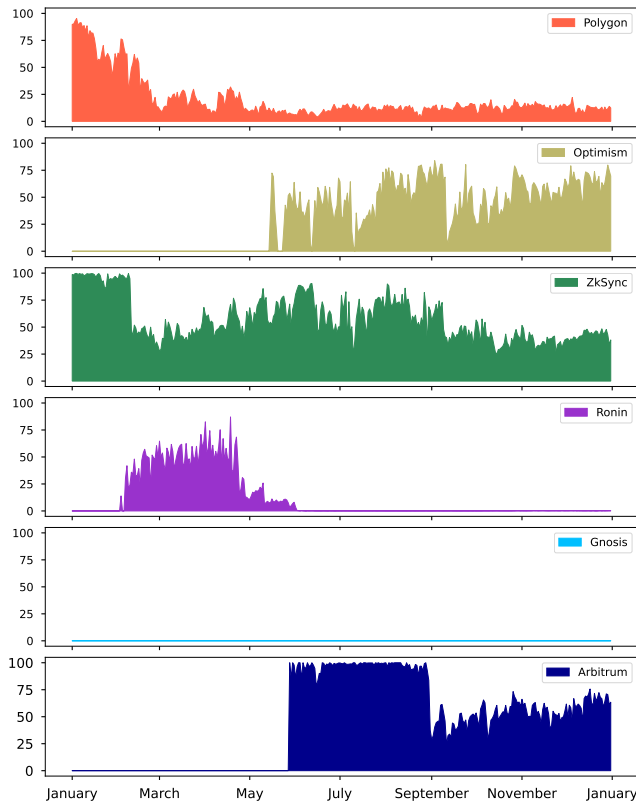
FIGURE 3. Layer-2 daily maintenance costs during 2021.

we divide the average load each layer-2 solution imposed on the main-chain (see §IV-A) at a given throughput level by the number of layer-2 transactions that were processed at that throughput level. With this, we can then compare the average main-chain cost of a layer-2 transaction with the average Ethereum transaction. The result are shown in Figure 4 which shows the relationship of the throughput of each layer-2 system (x-axis) with the relative transaction cost (y-axis) at the given throughput. As explained above, the relative transaction cost is given by the cost difference to an average Ethereum transaction, and aims to capture how much it would cost a user to submit the transaction to the layer-2 system when compared to submitting the same transactions directly to the main-chain. As an example, a relative cost of 50% indicates that a transaction on the layer-2 system is half the cost of a transaction that is submitted directly to the main-chain. This analysis helps us understand the transaction cost as the throughput of the system evolves. Given the economies of scale, we expect the transaction cost to drop as the throughput increases - as for a given fund withdrawal periodicity, say once every week, the number of processed layer-2 transactions will grow.

As the first observation, while side-chains, in some cases, also reach high cost factors at low throughput, in all cases, at the higher throughput levels, the cost difference between side-chain (Polygon, Ronin and Gnosis) and Rollups (Op-
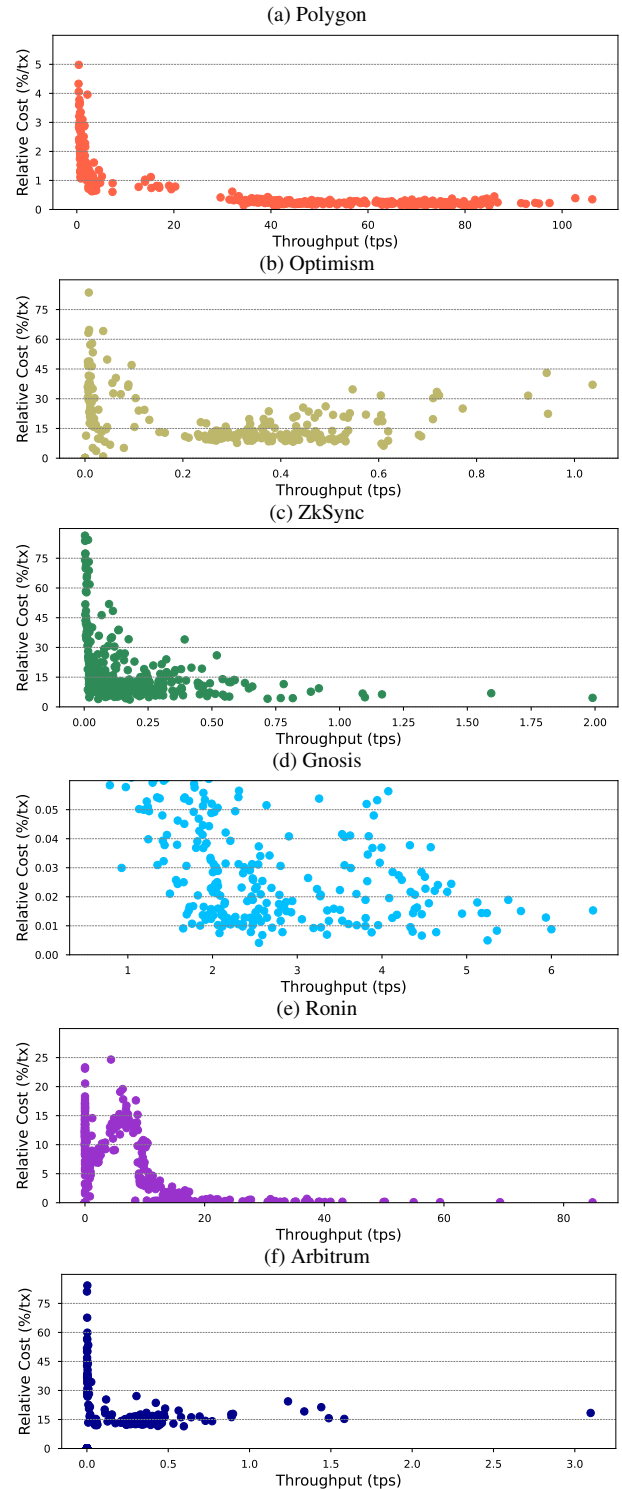

FIGURE 4. Relative transaction cost as the throughput evolves.

System	Estimated Throughput Potential		
	Observed	No Dep./With.	Theoretical
Polygon	6,900	63,000	65,000
Optimism	110	194	200
Arbitrum	102	191	4,500
Zksync	165	293	3,000
Ronin	5,700	2,500,000	-
Gnosis	56,000	Unlimited	90

timism, Arbitrum, ZKSync) approaches several orders of magnitude. This is explained by the fundamental differences between each approach, as discussed in §II, and it presents a clear tradeoff between the security and costs of each approach. Aside of that, the notable difference between Optimistic and Zero Knowledge Rollups is also worthy of note. Layer-2 systems using Optimistic Rollups (Arbitrum and Optimism) have a relative cost above 15%, while ZKSync only has a cost of around 7-8%. This comes from the inherent disadvantage of Optimistic Rollups compared to Zero Knowledge Rollups which, while not requiring computation on the main-chain, have a significant storage overhead as all layer-2 transaction data must be published on the main-chain. Note that, at low throughput levels, the fixed maintenance cost makes up a larger part of the overall cost. As such, with increasing throughput, the cost per transaction decreases continuously until the fixed cost makes up only a considerably small percentage of the overall cost. In addition to that, depending on the relative number of withdrawals and deposits, the overall cost per transaction may fluctuate.

Next, and as expected, very low throughput levels result not only in a generally much higher cost in all cases, but also, quite often, in a large variance, as the general maintenance overhead makes up a more significant percentage and deposits and withdrawals, or lack thereof, might skew the results significantly in either direction.

However, with increasing throughput, there are visible plateaus where the cost remains stable with low variance. We can observe these plateaus very visibly at Polygon, Ronin, ZkSync, and Arbitrum. Ronin is especially interesting in this regard as we can observe two plateaus: one higher plateau between 2.5 and 10 tps and one lower plateau above 12 tps. Optimism has a very high variance which is due to it not reaching throughput level comparable to approaches like Arbitrum and ZkSync which stabilized only at higher throughput levels. The case of Gnosis is similar, while it displays much higher throughput than the Rollups, in comparison to the other side-chains its throughput is still lower than their respective stabilization levels.

D. SCALABILITY ESTIMATES

Based on the data we have collected, we now estimate the scalability potential of each system taking into account the average cost per transaction of each layer-2 system and the main-chain capacity. Given the observed average load and cost of each evaluated layer-2 approach, and the Ethereum transaction throughput and maximum gas capacity, we can now reason on the practical throughput and cost levels achievable by each approach, and compare it with the theoretical throughput predicted in their respective white papers. This analysis provides us with some critical insight on whether these approaches are sufficient in the long term or whether new designs are needed. Note that, while some of these solutions could theoretically process a large number of transactions, this does not take the main-chain capacity into account and, depending on how tightly coupled to the main-chain each solution is, the maximum achievable throughput in practice can be much lower (i.e. up to the point where the load imposed in the main-chain surpasses its capacity).

We simplify the calculation of the estimated scalability potential of each of the approaches in two ways. First, we do not consider Cross-Chain-Rollups as these are not yet available and we have no data to predict their impact in the future. Second, to simplify the analysis, we assume that 100% of the main-chain load can be moved to layer-2, even though, in practice, there are certain operations that may not feasibly be moved to a layer-2 solution (for example applications that rely exclusively on main-chain storage). As such, the presented throughput potential is higher than realistically possible.

We compare the theoretical limit of each system as introduced in their respective white papers, with the estimated practical limit derived from our experimental observations. The practical limit considers the resource consumption each system imposes on Ethereum and Ethereum's maximum capacity imposed by the gas limit per block. The practical limit is calculated assuming an idealized scenario where each layer-2 system would be the sole user of Ethereum and therefore could consume 100% of the main-chain resources. This scenario, therefore, represents the maximum each system would be separately able to offer under such idealized conditions.

Recall that in Figure 4 we depict the average cost a layer-2 transaction imposes on the main-chain compared to an average Ethereum transaction at different throughput levels. In order to calculate the maximum potential throughput a given layer-2 system can process before exceeding the main-chain capacity, we want to use the lowest cost plateaus (at the highest throughput level) for this estimate that we identified by analyzing the respective graphs under different zoom levels. As Gnosis does not show any visual plateaus we've used a value after which there is a visible pattern change. As a result, we set the respective thresholds for this to 60 tps for Polygon, 0.2 tps for Optimism, 0.5 tps for ZKSync, 3 tps for Gnosis, 20 tps for Ronin and 0.25 tps for Arbitrum.

In addition to that, we also want to evaluate the scalability potential without considering withdrawals and deposits which correspond to the most favorable conditions. In order to calculate this, we have to obtain the maintenance rates at the throughput thresholds (described earlier), resulting in

10.93% for Polygon, 57.07% for Optimism, 56.5% for ZkSync, 53.47% for Arbitrum, and 0.23% for Ronin. As Gnosis solely interacts with the main-chain for deposits and withdrawals, the potential scalability is entirely independent of the main-chain.

The results of this analysis are presented in Table 4, where we compare the estimated throughput potential of the different systems considering the current load (Observed column) and without deposits and withdrawals (No dep./with. column) with the theoretical maximum throughput (Theoretical column) which is based on what each of the solutions claims to be able to achieve. We obtained these results by dividing the current maximum Ethereum capacity (in gas) by the respective per transaction costs of the layer-2 solutions at the identified plateaus.

Without considering deposits and withdrawals, Polygon reaches over 90% of their theoretical limit. However, if we consider the practical workload, it is closer to 10% as withdrawals and deposits occur very regularly and make up a large percentage of the total load. In particular, approaches based on Rollups impose a very high maintenance load (due to the strong main-chain coupling) and thus, even without considering deposits and withdrawals, they are unable to offer more than 300 tps. Ronin, in theory, due to the loose coupling, could scale to very high throughput levels. However, in practice, due to the large quantity of deposits and withdrawals, Ronin may only contribute up to around $5,700$ transactions to the Ethereum ecosystem. This way, Ronin stays even behind Polygon which offers Plasma based transaction and, as such, significantly higher security guarantees. The only approach that, in practice, in terms of cost, could offer very large throughput is Gnosis due to the very limited coupling to the main-chain and very rare deposits and withdrawals. However, Gnosis only offers a comparably low theoretical max throughput (90 tps).

V. CONCLUSIONS

Layer-2 systems emerged as a promising approach to circumvent the throughput limitations of permissionless blockchain systems. Despite previous studies on different aspects of layer-2 systems, a systematic experimental study that assesses the current and long-term impact these systems have on the main-chain, and a side-by-side theoretical and practical comparison of these systems were missing. In this paper, we conducted this study over a year-long period, and conclude that despite the diversity of existing layer-2 systems, all solutions fall short of fulfilling their promises given that the load they impose on the main-chain constitutes a severe bottleneck, preventing these systems from reaching their alleged maximum throughput levels. Moreover, we observe that regardless of the underlying technology, the performance of layer-2 systems will always be limited by the user behavior. In particular, the performance is heavily dependent on the frequency of deposits and withdrawals, which require regular main-chain interactions. As far as we could observe, at the current state of affairs, there is no application and/or

subsystem that encourages users to keep their funds in single layer-2 system and to avoid transfers via the main-chain. Given this scenario, it is questionable if the savings justify the use of most layer-2 systems given that, in the end, many offer significantly weaker security guarantees.

Furthermore, as one could probably expect, the approaches that do a better job in avoiding the costs of interacting with the main-chain are the ones that put trust on centralized operators, such as Ronin or Gnosis, driving a trend that undermines the decentralization of the Ethereum ecosystem (as also observed by other works [6]). This is unfortunate, as it comes at odds with the original motivation for the use of blockchain systems, i.e., to avoid trust on centralized components.

As future work, we plan to extend our study to cover the layer-2 systems of other blockchain ecosystems, such as the lighting network of Bitcoin [29], and study whether the same trends we observe here are present in those ecosystems. We also plan to use the logs collected for our experimental study to create a synthetic workload that captures the features observed on the layer-2 systems and that can be used to benchmark different approaches in a reproducible manner.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf. Accessed on: Apr. 18, 2022

[2] V. Buterin, "Ethereum whitepaper", 2013. [Online]. Available: https://ethereum.org/en/whitepaper. Accessed on: Apr. 18, 2022

[3] N. Szabo, "Formalizing and securing relationships on public networks," *FM*, vol. 2, no. 9, Sep. 1997, DOI: 10.5210/fm.v2i9.548.

[4] R. Neiheiser, G. Inácio, L. Rech, and J. Fraga, "HRM smart contracts on the blockchain: emulated vs native," *Cluster Comput.*, vol. 23, no. 3, pp. 2105–2122, Feb. 2020, DOI: 10.1007/s10586-020-03063-9.

[5] M. Vukolić, "The quest for scalable blockchain fabric: proof-of-work vs. BFT replication," in *Open Problems in Network Security*, Cham, Switzerland: Springer, 2016, pp. 112–125.

[6] A. Sai, J. Buckley, B. Fitzgerald, and A. Le Gear, "Taxonomy of centralization in public blockchain systems: A systematic literature review," *Inf. Process. Manag.*, vol. 58, no. 4, Jul. 2021, DOI: 10.1016/j.ipm.2021.102584.

[7] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv:1710.09437*, Oct. 2017. [Online]. Available: https://arxiv.org/abs/1710.09437. Accessed on: Apr. 18, 2022

[8] K. Wüst and A. Gervais, "Do you need a blockchain?," in *CVCBT*, Zug, Switzerland, pp. 45-–54, 2018.

[9] C. Sguanci, R. Spatafora, and A. Vergani, "Layer 2 blockchain scaling: a survey," *arXiv:2107.10881*, Jul. 2021. [Online]. Available: https://arxiv.org/abs/2107.10881. Accessed on: Apr. 18, 2022

[10] J. Kanani, S. Nailwal, and A. Arjun, "Polygon whitepaper," 2021. [Online]. Available: https://whitepaper.io/document/646/polygon-whitepaper. Accessed on: Apr. 18, 2022

[11] Optimism, "Optimism," 2020. [Online]. Available: https://optimism.io. Accessed on: Apr. 18, 2022

[12] Offchain Labs, "Arbitrum", 2022. [Online]. Available: https://offchainlabs.com/. Accessed on: Apr. 18, 2022

[13] Matter Labs, "Introduction to zkSync for developers," 2022. [Online]. Available: https://docs.zksync.io/dev. Accessed on: Apr. 18, 2022

[14] S. Mavis, "Ronin Ethereum sidechain," 2021. [Online]. Available: https://whitepaper.axieinfinity.com/technology/ronin-ethereum-sidechain. Accessed on: Apr. 18, 2022

[15] I. Barinov, V. Arasev, A. Fackler, V. Komendantskiy, A. Gross, A. Kolotov, and D. Isakova, "POSDAO: Proof of stake decentralized autonomous organization," Apr. 2019. [Online]. Available: https://ssrn.com/abstract=3368483. Accessed on Apr. 18, 2022.

[16] Gnosis Chain, "xDai & Polygon differences and similarities," 2022. [Online]. Available: https://developers.gnosischain.com/about-gc/news-and-information/comparisons/matic#similarities. Accessed on: Jul. 18, 2022

[17] P. Silva, D. Vavricka, J. Barreto, and M. Matos, "Impact of geo-distribution and mining pools on blockchains: A study of Ethereum," in Proc. *DSN*, Valência, Spain, pp. 245—252, 2020.

[18] L. Burkholder, "The halting problem," *ACM SIGACT News*, vol. 18, no. 3, pp. 48—60, Apr. 1987

[19] A. Gluchowski, "Evaluating Ethereum l2 scaling solutions: A comparison framework," 2020. [Online]. Available: https://blog.matter-labs.io/evaluating-ethereum-l2-scaling-solutions-a-comparison-framework-b6b2f410f955. Accessed on: Apr. 18, 2022

[20] K. Groves, "Cryptocurrency exchange hacks,", 2022. [Online]. Available: https://www.hedgewithcrypto.com/cryptocurrency-exchange-hacks/. Accessed on: Apr. 18, 2022

[21] A. Beregszaszi, "EIP-2070: Hardfork Meta: Berlin [DRAFT], "Ethereum improvement proposals, no. 2070,", 2019. [Online]. Available: https://eips.ethereum.org/EIPS/eip-2070. Accessed on: Apr. 18, 2022

[22] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta, "EIP-1559: Fee market change for ETH 1.0 chain,", 2019. [Online]. Available: https://eips.ethereum. org/EIPS/eip-1559. Accessed on: Apr. 18, 2022

[23] Binance Team, "Binance completes Axie Infinity (AXS) & smooth love potion (SLP) Ronin network integration,", 2021. [Online]. Available: https://www.binance.com/en/support/announcement/9f2e474636a348d19e28bbd4c32f8384. Accessed on: Apr. 18, 2022

[24] Binance Team, "Binance completes Polygon (MATIC) mainnet token wallet integration,", 2021. [Online]. Available: https://www.binance.com/en/support/announcement/4515e97c82d64189885da7cd7d606b86. Accessed on: Apr. 18, 2022

[25] A. Gangwal, H. Gangavalli, and A. Thirupathi, "A Survey of layer-two blockchain protocols," *arXiv:2204.08032*, Apr. 2022. [Online]. Available: https://arxiv.org/abs/2204.08032. Accessed on: Jul. 18, 2022

[26] C. Sguanci, R. Spatafora and A. Vergani, "Layer 2 blockchain scaling: a survey," *arXiv:2107.10881*, Jul. 2021. [Online]. Available: https://arxiv.org/abs/2107.10881. Accessed on: Jul. 18, 2022

[27] M. Jourenko, K. Kurazumi, M. Larangeira and K. Tanaka, "SoK: A taxonomy for Layer-2 scalability related protocols for cryptocurrencies," *Cryptology ePrint Archive, Paper 2019/352*, Oct. 2017. [Online]. Available: https://eprint.iacr.org/2019/352. Accessed on: Jul. 18, 2022

[28] N. Chemaya, and D. Liu, "Cost of security of layer 2 network — evidence from Polygon network," May. 2022. [Online]. Available: https://ssrn.com/abstract=4119827. Accessed on: Jul. 18, 2022

[29] J. Poon, and D. Thaddeus, "The Bitcoin Lightning network: scalable off-chain instant payments," Jan. 2016. [Online]. Available: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf. Accessed on: Jul. 29, 2022

[30] Ethereum, "The Merge,", 2022. [Online]. Available: https://ethereum.org/en/upgrades/merge/. Accessed on: Sep. 20, 2022

GUSTAVO INÁCIO received his Bachelor degree in computer science from the Federal University of Santa Catarina in May 2022. He participated in this work in the context of a scientific internship in the distributed systems laboratory. His main interests include Cloud Computing, Security, Distributed Systems, Blockchain and Games.

LUCIANA RECH is an Associate Professor at the Informatics and Statistics Department (INE) of the Federal University of Santa Catarina(UFSC) and a member of the Distributed Systems Research Laboratory (LAPESD). Graduated from University of Cruz Alta in Computer Science, with Master's degree in Computer Science (Field: Parallel and Distributed Computing) from Federal University of Santa Catarina and Ph.D. in Electrical Engineering (Field: DAS/Information System). She has experience in the field of computer science with a focus on computational systems working more closely with: Distributed Systems, Intelligent Systems, Real Time Systems and Applied Informatics.

CARLOS MONTEZ (Member, IEEE) is a full professor and researcher at Automation and System Department, UFSC. Since 2005, he has advised and coadvised 23 master's and 12 Ph.D. students, and authored or coauthored more than 140 publications. His research interests include wireless sensor networks, industrial communication protocols, big data sensing, and real-time systems.

MIGUEL MATOS (Member, IEEE) is an assistant professor at the Engineering School of the University of Lisbon (Instituto Superior Técnico) and a Senior Researcher at INESC-ID. His research interests lie in broad the area of systems. His work has been published in SOSP, Eurosys, TPDS, JPDC, ICDCS, DSN, IPDPS and Middleware among others.

RAY NEIHEISER obtained his Bachelor's degree in computer science in 2015 from the University of Media in Stuttgart and subsequently obtained his Master's degree in 2017 from the Federal University of Santa Catarina in Brazil. He defended his PhD Thesis in March 2022 in the context of a Cotutelle Agreement between the Federal University of Santa Catarina in Brazil and the University of Lisbon in Portugal. His primary areas of interest are distributed systems, byzantine fault tolerant consensus and decentralization. Currently he works at IST Austria as a Post-Doc with Parallel Execution Engines for Blockchains.

LUÍS RODRIGUES (Senior Member, IEEE) is a professor at Instituto Superior Técnico, Universidade de Lisboa and a researcher at INESC-ID. His research interests lie in the area of reliable distributed systems. He is co-author of more than 200 papers and 3 textbooks on these topics. He is also a senior member of the ACM.