ABSTRACT

Edge computing has received significant attention from both academic and industrial research circles. The paradigm aims to decentralize the existing cloud infrastructure by incorporating resources co-located alongside its client. Researchers have also proposed solutions for a fully decentralized crowdsourced compute paradigm enabled by Distributed Ledger Technologies (DLTs). This paper investigates the rationale behind DLTs over crowdsourced resource marketplaces to support the requirements of latency-critical applications targeted by edge computing. We develop a fully configurable NEtworked Blockchain emULAtor, or NEBULA, to scrutinize the internal performance bottlenecks of DLTs. We evaluate two blockchain categories – proof-based (popularly used in Bitcoin, Ethereum) and hybrid consensus and find that the enabling factor of DLTs – scale – is also its primary latency contributor. We show that, in reality, the latency overheads due to DLT operation far exceed the operational requirements of edge applications.

CCS CONCEPTS
• Networks → Public Internet; Network measurement; Cloud computing; Computing methodologies → Real-time simulation; Distributed simulation; • Computer systems organization → Peer-to-peer architectures.

KEYWORDS
Distributed Ledger Technology, Blockchain, Edge Computing, Crowdsourcing, Emulation, Network Performance

1 INTRODUCTION

The growth of cloud computing in the early 2000s ushered in a new revolution for Internet-backed application services. Application providers could now utilize seemingly unlimited compute capabilities to match the demand on-the-go – thanks to the highly managed and overly provisioned datacenter silos set up by cloud operators. However, the recent growing interest in next-generation applications such as IoT, AR/VR, etc. [44] has revealed several performance gaps in the current cloud computing setup. The next-gen applications are mission-critical and must operate within strict latency bounds (mostly within 100 ms [52, 58]) – which cannot be satisfied by the cloud due to its remote (and numbered) deployment.

To this end, edge computing has emerged as a compelling solution for such applications [53, 64, 65]. The paradigm aims to decentralize cloud computing by utilizing resources deployed in physical proximity to the users [3]. The computing paradigm is backed by industry and academia alike, both proposing solutions for infrastructure deployments fitting their needs. While existing cloud providers, ISPs, etc. are advocating for deploying edge servers within their managed facilities [33], unmanaged approaches such as fully decentralized crowdsourced marketplaces are simultaneously gaining popularity [40]. While the majority of works in edge computing have focused on how to enable the paradigm through novel networking protocols [47, 54, 55, 66] or designing applications that can best utilize compute capacity near end-users [8, 18, 28], understanding what “edge” is has somehow lost relevance. Recent measurement studies have found that the datacenter networks and deployment has drastically expanded during the last decade, and latencies within the WAN connecting users with cloud is quite low [20, 22, 23]. As a result, there is no clear contender for a “textbook” edge infrastructure as future deployments can (potentially) benefit from unmanaged crowdsourced resources (in terms of latency) and servers in managed environments (in terms of reliability/availability). We believe that edge computing must embrace all possible compute opportunities since its full potential can only be reached if the infrastructure is dense and readily available.

To support such a “collective” compute fabric, researchers have turned to Distributed Ledger Technologies (DLTs) as an enabling technology [60]. Over the past decade, DLTs (and its subclass blockchain) have matured – enabling several commercial-grade systems through cryptocurrencies [29] and smart contracts [72]. However,
the applicability of DLTs for decentralized crowdsourced edge computing from a practical perspective is still a relatively unexplored topic. Conceptually, DLTs can easily enable several sophisticated operations desired by the crowdsourced edge. For example, smart contracts can support task scheduling [42] and resource matching [78], while users can be incentivized to participate through cryptocurrencies. As a result, the majority of research in this space treat DLTs as a “one-size-fits-all” black box that seamlessly supports the operations desired by edge computing. Furthermore, existing work in this space completely ignores (or significantly misrepresents) the additional latencies due to DLT operations within their solutions. This lapse in insight is further fuelled by the significant lack of studies investigating (or measuring) the networking overheads of DLTs at scale. We argue (and later show) that with increasing infrastructure and participant scale, DLTs act more as a bottleneck than enablers for low latency promises of edge computing.

In this paper, we investigate the suitability of DLTs (specifically, blockchains) for addressing the latency requirements of applications over decentralized edge computing. We first identify the key components in crowdsourced marketplaces that necessarily require blockchains for operation. Further, we shed light on the internals of different DLTs, particularly focusing on networking. We develop a NEtworked Blockchain emULAtor – NEBULA – that allows us to scrutinize the realistic operation of different DLTs in a controlled testbed. Through extensive experiments over different parameters (e.g., network size, density, latency, block rates, etc.), we investigate the applicability of two popular blockchains – proof-based (e.g., Bitcoin, Ethereum) and hybrid consensus (e.g., EOS) – for supporting latencies of edge-enabled applications. We realize that the primary success parameter of DLTs – i.e., scale – is also their biggest performance bottleneck. Our aim with this work is not to deter research in this area but to provide a reality-check for future studies. We make NEBULA available for public use at [27].

## 2 CHAINING THE EDGE

With smart contracts, DLTs have become general-purpose replicated state machines [88] and can now enable decentralized applications, such as electronic voting [49], data provenance [61], or item sharing [15]. As a subclass of DLTs, blockchains maintain the distributed ledger by batching transactions into immutable blocks, each block referencing its parent (Figure 1). Peers can participate in the blockchain in a variety of ways. For example, participation can be public or private (based on read access), permissionless, or permissioned (based on consensus participation) [37].

### 2.1 Understanding Blockchains

Table 1 shows the four broad categories of DLTs¹ used today. Proof-based blockchains (PoX) are most popular as they enable cryptocurrencies like Bitcoin [56], Ethereum [81], etc., to operate in public-permissionless mode. New blocks in PoX are proposed via a distributed lottery, e.g., proof-of-work (PoW) [56], proof-of-stake (PoS) [63], proof-of-retrievability (PoR) [51] etc. The lottery winner (or miner) includes a zero-knowledge proof of their win within a new block, which the peers verify. Occasionally, two miners can mine a block referencing the same parent – resulting in a fork (see Figure 1). The participants eventually reach an agreement on transaction history through mechanisms such as longest chain, GHOST [71], etc. (green chain in Figure 1). Transactions in forks outside the main chain, a.k.a. stale blocks (orange chain), are disregarded and implicitly invalidated. To reduce the probability of stale forks and counter potential attacks like selfish-mining, double-spending, etc., [32, 48], participants can wait for additional confirmations before accepting a block. Note that despite these measures, blocks may still become stale if overtaken by a longer fork. Such blocks are referred to as confirmed stale blocks. On the other hand, general stale blocks are blocks outside of the main blockchain with at least one confirmation – the block itself. Such (transient) forks are a common occurrence in PoX blockchains and differ from occasional hard forks caused due to software updates [39]. Hard forks are of interest to this study as they are rare and deliberate events outside of regular blockchain operations.

<table>
<thead>
<tr>
<th>Category</th>
<th>Participation Mode</th>
<th>Consensus Algorithms</th>
<th>Consensus Complexity</th>
<th>Consensus Finality</th>
<th>Fault Tolerance</th>
<th>Throughput (TPS)</th>
<th>Latency</th>
<th>Example Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proof-based (PoX)</td>
<td>Public-Permissionless</td>
<td>PoW, PoS, PoR</td>
<td>$O(1)$</td>
<td>Probabilistic: LCR/GHOST, confirms.</td>
<td>50% compute, stake value, storage space</td>
<td>Hundreds</td>
<td>Minutes</td>
<td>Bitcoin [56], Ethereum [81], Permacoin [51]</td>
</tr>
<tr>
<td>BFT</td>
<td>Public/Private-Permissioned</td>
<td>PBFT, RBFT</td>
<td>$O(n^3) - O(n^2)$</td>
<td>Total</td>
<td>33% servers</td>
<td>Thousands</td>
<td>Seconds</td>
<td>Hyperledger Indy [77]</td>
</tr>
<tr>
<td>Hybrid Permissioned</td>
<td>DPOS-BFT, Tendermint</td>
<td>$O(n) - O(n^3)$ in delegates</td>
<td>Probabilistic, Total</td>
<td>33% delegates, stake value</td>
<td>Thousands</td>
<td>Seconds</td>
<td>EOS [14], Cosmos Hub [43]</td>
<td></td>
</tr>
<tr>
<td>DAG Permissioned</td>
<td>PoW, parent approval</td>
<td>$O(1)$</td>
<td>Probabilistic</td>
<td>50% participants / comp. power</td>
<td>Thousands</td>
<td>Seconds, Minutes</td>
<td>IOTA [76], Spectre [70]</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Categorizing Distributed Ledger Technology (DLT) platforms

![Figure 1: Stale blocks (SB) and confirmed SB in DLT.](image)

¹We use "DLTs" and "blockchains" interchangeably for the sake of simplicity.
In contrast to PoX, blockchains utilizing traditional byzantine fault tolerant (BFT) protocols (e.g., Hyperledger Indy [77]) need identifiable peers and permissioned access. BFT blockchains reach consensus after every block – sacrificing fault tolerance for scalability. Hybrid or committee-based blockchains improve upon these shortcomings by allowing peers to vote for delegates responsible for proposing blocks in fixed rounds. Delegates reach consensus using protocols like BFT (Cosmos [43]), pipelined BFT (EOS [14]), etc., with slightly relaxed finality. DAG-based DLTs depart from regular chain architectures, allowing blocks (SPECTRE [70]) or transactions (IOTA [76]) to reference multiple parents. To restore a consistent transaction history, the resulting partial ordering is then transformed into a total order algorithmically. However, such approaches lack maturity and cannot achieve similar performance as chain-based DLTs.

2.2 Blockchains for Resource Marketplaces

While researchers have employed blockchains for several dimensions of edge computing, its most popular use is to enable decentralized crowdsourced marketplaces [1, 42, 57, 78, 82, 86, 87]. Such marketplaces aim to facilitate a pervasive platform that integrates both managed and unmanaged resources at the edge. Figure 2 shows the internals of such a marketplace. Resource providers, including large organizations (ISPs, city admin) and individual operators, offer their hardware for an asking price. These resources are sought after by clients/developers to deploy their applications. The glue between the two parties is the blockchain that acts both as a decentralized auctioneer and as a payment portal. The blockchain can be supported by entities with dedicated resources (governments, etc.) or the community at large.

Several components are required for enabling such a marketplace to operate over blockchains, highlighted in blue. The auction protocol allows providers to advertise their hardware (as offers) and clients to “bid” for resources (as requests). The content of bids varies for different approaches [13, 86] but primarily includes resource valuation, costs, SLAs, etc. The matching algorithm is then responsible for finding optimal pairings between offers and requests, e.g., achieving maximum economic, QoS or runtime performance [73, 86]. Each match is treated as a new transaction, which is bundled with other matches and published as a block. Once bidders come to a conclusion on the best match, the resource provisioning allocates resources to the winning client for the agreed duration. To minimize overheads, provisioning can take place off-chain [46].

Since such a marketplace can include selfish actors, verification of final results to SLA in the original bid becomes necessary [78]. Operations like log verification [9], correctness checking [11], etc., can be carried out by incentivized external entities off-chain [75]. In case the verification indicates a dispute, the aggrieved party can initiate the resolution process over smart contracts, which may result in monetary or reputational punishments. While all components in Figure 2 are integral for enabling crowdsourced edge marketplaces over blockchains, only auctioning and matching operations are critically dependent on blockchains. Therefore, the wait time for gathering bids and matching them will strongly depend on underlying blockchain performance.

2.3 Blockchains as Plug-and-Play?

The majority of research encouraging an alliance between edge and blockchains has focused on “edge” as it provides more room for novel research solutions [38, 73]. Most of these approaches consider existing blockchain technologies as plug-and-play “blackboxes” [42, 86, 87]. The impact of potential overheads within blockchains on overall performance is either not considered or usually dismissed as out-of-scope.

We argue that using DLTs to support edge computing is analogous to finding the right tool to hammer a nail. The utilities and trade-offs of each DLT variant differ significantly and must be taken into consideration when proposing solutions reliant on the technology [37]. Considering a DLT is a distributed networked system, its notable performance metrics are end-to-end latency and throughput, in other words, “time taken to mine and confirm a new block to the chain”. Parameters, such as consensus mechanism and block sizes, can significantly affect these metrics at scale. However, similar to traditional networks, optimizing for both high throughput and low latency is difficult (and almost contradictory) in DLTs. For example, Rainblock [59] plugs a performance bottleneck in Ethereum, allowing the technology to achieve ≈20K transactions per second. However, the solution packs 480x transactions in each
block which increases the block size from 40-60 KB to 24 MB – resulting in a significant jump in propagation latency. On the other hand, next-generation applications care for both throughput and latency, with the latter being stricter of the two (most apps must operate under 100 ms) [52]. While many simulators and models evaluating different protocols for blockchains have been proposed in the past [5, 32, 68], the potential bottlenecks due to networking have remained largely unexplored in the literature. We attempt to bridge this gap by developing a configurable blockchain emulator called NEBULA, which allows us to investigate different blockchain technologies at scale.

3 BLOCKCHAINS UNDER MICROSCOPE

Figure 3 shows the internals of our Java-based NEtworked Blockchain Emulator – NEBULA. We model blockchains as a network of virtualized peers that communicate via simple message-passing of Protobuf [2] over TCP. Each peer runs on a separate thread, allowing us to emulate n nodes on m coordinator machines (n >> m). One coordinator generates a network topology based on parameters such as the number of nodes, bandwidth, density and latency of the links, etc. Additionally, we also allow replicating networks from synthetic (e.g., random, scale-free [4]) and realistic (e.g., CAIDA AS [36], user-cloud connections [26]) datasets. Nodes in the graph are split between coordinator machines depending on their computing capabilities. The underlying network is modeled as TCP connections with application-level delays. The nodes are then instructed to establish interconnections to mimic the specified topology and collaborate as a blockchain. NEBULA is available publicly at [27].

In this work, we evaluate proof-based (PoX) and hybrid blockchain approaches (DPoS-BFT). While PoX is most popular, DPoS-BFT is the most promising performance-wise [7, 12]. For PoX, emulated peers create blocks randomly and concurrently at a configurable rate. For DPoS-BFT, NEBULA selects a subset of BPs that generate blocks at a constant rate in a round-robin fashion. The consensus protocol is pipelined BFT, where a block is published only if 2/3 BPs verify the block. Consensus protocols in NEBULA are implemented as Protobuf message definitions allowing for easy extensions, e.g., Tendermint [17], with minimal overhead. The emulator also allows fine-tuning several other performance-affecting parameters, e.g., transaction rate, size, and fee, block verification time, number of confirmations, etc., to name a few.

Correctness. We emulate four popular PoX blockchains, specifically Bitcoin [24], Ethereum [5], Dogecoin, and Litecoin [32], with similar block rates and propagation delays as their real-world counterparts. We generate up to 10K blocks over multiple iterations and compare the total stale blocks to previous real-world measurements [5, 24, 32]. As noted in §2.1, generated stale blocks is a factor of consensus protocol and network latencies, and thus is an accurate representation of blockchain behavior. Figure 4a showcases that it is possible to emulate real-world blockchains in controlled settings through NEBULA. The slight deviation in accuracy is likely due to our lack of knowledge about other internal parameters of these blockchains. To address this, we compare NEBULA to a blockchain simulator [32]. Figure 4b shows that in similar networks with equal parameter settings, our emulations closely follow the simulations with a minimal error of 8.7%. Compared to the simulator, NEBULA

![Figure 4: NEBULA correctness to (a) Ethereum, Bitcoin, Dogecoin & Litecoin, and (b) blockchain simulator.](image)

is more fine-grained as it exposes additional configuration parameters and opens up the future potential for hardware-in-the-loop experiments.

3.1 Dissecting Blockchain Performance

We now attempt to empirically understand the possibility of employing blockchains for edge computing, especially for the model use-case of decentralized resource marketplace (§2.2). Among all components, potential bottlenecks in blockchain’s performance directly impact auction and matching stages as new auctions can only be entertained once existing ones are dealt with. In such an architecture, resource bids are sent to the auction protocol in the blockchain as traditional transactions that are added to a pool of unmatched bids. The matching is completed during block creation, and the block is finalized in accordance with the underlying consensus protocol (e.g., PoW, DPoS, etc.). The final block includes a pairwise match between all included transactions. The block – thus, the auction – is accepted by the clients once additional blocks extend it and confirm it as per the protocol. We use confirmation latency as our primary performance metric, i.e., the time elapsed between receiving resource bids and confirming the block that contains the match of those bids. Additionally, we measure the percentage of stale blocks and investigate their impact on consistency, efficiency, and fault-tolerance of auctions executed on the chain.

Setup Configuration. We conducted our experiments on a compute cluster of 15 Linux-based VMs, totaling ≈60 CPU cores. All VMs are interconnected by 1 Gbps Ethernet. We emulate three distinct blockchain network sizes for our experiments: small, medium, and large. Small and medium are scale-free networks of 100 and 500 nodes, respectively, with topology similar to Ethereum [80]. The small network follows Gaussian latency distribution with a mean latency of 10 ms (representative of a smart city [69]). Configurations of the medium and large network are inspired from user-to-cloud latencies and traversal measurements in [21]. The average end-to-end propagation delay in the medium network is set to 63 ms, similar to user to data center latencies in the study. Our large network is a topology of 2700 ASes residing between end-users and the cloud. Considering these ASes are the best contenders for potential edge server deployments, the network topology reflects an edge infrastructure spanning multiple organizations. Due to
the comparatively high block rates of our experiments, individual blocks are smaller and contain only a few transactions – allowing us to focus on the accuracy of our latency measurements without exhausting bandwidth limits. We conduct multiple iterations for each experiment, generating up to 30K blocks per iteration.

Proof-based Blockchains (PoX). We first evaluate the suitability of PoX blockchains for supporting edge marketplace tasks. Figure 5a shows how an increase in the frequency of blockchain-backed auctions (translating to increasing block rate) affects overall transaction latency. The dashed horizontal lines are the limits of the fitted Weibull functions. At first glance, it appears that PoX blockchains can effectively support high-frequency auctions at the edge, as the latency decreases to an asymptotic limit with increasing block rate. Naturally, as the block generation rate rises, the waiting time for transactions to be included in the next block approaches zero, and the remaining latency is primarily due to network propagation. However, higher rates lead to higher concurrency in block creations and, thus, increased stale block percentages (see Figure 5c). More stale blocks in the chain not only imply wastage of computational resources but also a possibility of invalid auctions – as the same bid can be included in multiple concurrent blocks.

The number of confirmed stale blocks in PoX can be reduced by waiting for additional confirmations (see §2.1). As expected, Figure 5b shows that stale block percentages decrease exponentially if more confirmations are required before accepting the block. However, the additional wait time for confirmations also results in an increase in overall latency. Note that the confirmations do not affect general stale blocks in PoX (denoted by stale blocks with one confirmation in Figures 5b and 5c). Moreover, the trade-off between latency and stale blocks increases with network size (see Figure 5c). We find that for larger networks (>2K nodes), even with the most lenient settings, PoX blockchains add ≈ 500 ms overhead to edge marketplace operation (see Figure 5a) – far exceeding the 60-100 ms desired threshold [52].

Hybrid Blockchains (DPoS-BFT). By design, delegated proof-of-stake with BFT (DPoS-BFT) avoids waiting for additional confirmations by centralizing consensus to a subset of block producers (BP). BP’s are elected by the network through stake-based voting [79]. Figure 6 shows the transaction latency of DPoS-BFT at different block rates and network sizes. Here, we restrict the consensus to 21 BPs (default in EOS [14]) and show the results when BPs are the best or the worst connected nodes in the network.

Similar to PoX, transaction latency in DPoS-BFT is primarily influenced by network sizes (see Figure 6). While optimal BP selection can reduce latencies in large networks, it has a negligible effect in smaller networks. Our result, however, indicates that in small networks, DPoS-BFTs can support <100 ms latencies at high block rates – showing some promise for edge-based applications. However, while DPoS-BFT ensures that only one BP proposes blocks at any given time, stale blocks can still occur in the chain. The primary contributor is network propagation delay, as BP responsibility changes between nodes at the end of each round. Therefore, preventive mechanisms, like increasing the time allocated for the last block per BP, are still needed in DPoS-BFT – which will result in additional latencies. Furthermore, as noted in Table 1, the fault-tolerance of DPoS-BFT is lower than PoX, and the blockchain security can be attacked with 33% BP control (which is quite possible in relatively small networks). Nevertheless, in terms of performance, a local DPoS-BFT blockchain seems to be most promising for our use-case.

4 DISCUSSION

Our results show that it is largely unclear if blockchains can be proponents for crowdsourced edge computing. While the performance differs for different consensus technologies, achieving the coveted 60-100 ms end-to-end latency [52] is usually not possible for blockchains spanning large networks. Additionally, one must sacrifice between latency (confirmations) or processing cycles when dealing with stale blocks. Due to stale forks, expensive roll-back mechanisms [16] would still be required in edge marketplaces to avoid monetary losses. Here, two things must be noted to appreciate our results. First, while we discuss our results in context to edge computing requirements, our experiment design is fairly generic, and our results are also relevant for understanding blockchain networking bottlenecks. Second, the transaction latency in §3.1 does not include the processing overhead of the auction/matching protocol itself, which is also known to be significant [30]. We do not empirically explore alternate blockchains like DAGs in this work since recent measurements report their confirmation latencies to last several minutes [35] – making them unsuitable for edge.
Despite its popularity in literature [42, 86], propagating and confirming local edge transactions through a global blockchain network within stringent latency bounds is an uphill battle. While recent developments such as ZK- or Optimistic-rollups [34] attempt to improve scalability, adding consensus from a global network to the critical path of an edge transaction is a massive operational overhead. In small networks, DLTs show some promise for decentralized computing (Figures 5a and 6). However, restricting the size of the network is tough in reality. One possible way is through fully-permissioned DLTs, e.g., HotStuff [85], which restricts participation to only a few pre-approved nodes (likely to be under the same management – city administration, industry, etc.). While several factors favor such a deployment in the future, the primary benefits of DLTs (trust, privacy, openness) are no longer present in such gated participation policies.

Another solution would be to collocate DLTs alongside edge networks. Figure 7 shows a fragmented (a.k.a. sharded) blockchain which localizes data and task operation to different edge markets distributed across geography. Still, each sharded blockchain cannot be permissionless in nature, as access needs to be restricted within geography. Localized blockchain markets are lazily synchronized by a global interledger [31, 67]. Applications can operate in local markets and are administered globally via the interledger. Such a blockchain can concurrently support highly frequent localized tasks and occasional global operations with minimal latency overhead. Interestingly, the approach of blockchain sharding describes a hybrid centralization solution. While general membership is retained to smaller regions using a “centralized” third party (i.e., city admin, interledger), some decentralization in control/accountability is retained by individual members who need not necessarily trust each other. Still, significant research efforts are required to understand the trade-offs in such a blockchain architecture, especially for tasks over multiple shards and potential double-spends [67].

**Practical relevance of blockchains & crowdsourcing.** Irrespective of the technical challenges in supporting edge marketplaces over DLTs, the relevance of both these domains, in reality, remains to be seen. In the past 5–6 years, the interest and knowledge about blockchains in the general public have increased significantly as cryptocurrencies like Bitcoin and Ethereum, are now accepted as blockchains in the general public have increased significantly as evident from the $1200% surge in Folding@Home contributors in 2020 [45]. Crowds are more open to participating as miners of new DLTs to invest early as coin holders. Even without reward mechanisms, crowdsourcing as a concept is now more accepted, as evident from the $1200% surge in Folding@Home contributors in 2020 [19]. On the other hand, cloud providers, such as Amazon and Google, are dominating the computing market and are investing significantly to expand their network reach globally [21]. Many have also established specialized servers within ISP facilities – allowing them to reduce latencies while retaining network control [50]. Interestingly, many cloud players also have a significant presence in the edge as manufacturers and operators of smartphones, smart devices, etc. By incorporating “user-owned-but-manufacturer-operated” hardware with existing cloud network, these organizations have the unique opportunity to build a hybrid crowdsourced infrastructure without involving blockchains. On the other hand, efforts to dethrone the stronghold of cloud providers and usher in the era of decentralized computing are strongly backed by independent investors. For example, “Internet computer”, a blockchain-based non-proprietary compute fabric from Dfinity raised $195M [74] and was launched earlier this year [25].

**Security.** Our results show that blockchains can only support latencies for edge tasks in smaller networks (typically implying lower node counts), which directly inhibits its own paradigm of success – strength-in-numbers. By restricting the network size, inherent benefits of DLTs, e.g., security, privacy and trust, start to crumble and avenues for novel attack vectors open up while existing security loopholes are further magnified [6]. For example, 51% attacks over PoX are now realistically achievable [32], and DPoS blockchains, which are already vulnerable to malicious cartels influencing its voting procedure [83], are prone to complete takeovers. Permissioned blockchains under a central authority might overcome several such security challenges, but they come at the cost of trust and privacy.

**Other Applications.** Despite potential drawbacks in latency, other edge computing use cases not reliant on network delay should be mentioned [84]. Here, decentralized storage systems [62], weather sensors [41], or certain smart home/city applications [52] (i.e., bus timetables, smart parking meters) have been proposed in the past. While these applications are located outside of edge latency feasibility zones [52], they could still benefit from higher bandwidth and localized traffic as promised by edge computing. Due to the decentralized and distributed nature of edge in general, building a self-sufficient ecosystem on top of a secure and trustless DLT platform is undoubtedly enticing.

**5 CONCLUSION**

In this paper, we investigated the rationale behind crowdsourced edge computing backed by distributed ledgers. We identified popular approaches in crowdsourced marketplace research that rely on blockchain for operation. We designed the generic, scalable and configurable blockchain emulator NEBUA to shed light on the internal overheads of proof-based and hybrid blockchain technologies. We highlight several research directions which might make DLTs fit for the edge, albeit with potential limitations, such as sharded interledgers. In conclusion, we believe that there are several challenges to be tackled for enabling distributed edge over blockchains. Future research in this direction must not blindly offload performance-critical functionality to be handled by DLTs.

**ACKNOWLEDGMENTS**

This work was supported by the Swedish Foundation for Strategic Research with grant number GMT-14-0032 (Future Factories in the Cloud) and EU Celtic project Piccolo (C2019/2-2).