The original chapter was published in Mukherjee A., De D., Ghosh S.K., Buyya R. (eds) Mobile Edge Computing. Springer, Cham. https://doi.org/10.1007/978-3-030-69893-5_14

Blockchain for Mobile Edge Computing: Consensus Mechanisms and Scalability

Jorge Peña Queralta and Tomi Westerlund

Abstract Mobile edge computing (MEC) and next-generation mobile networks are set to disrupt the way intelligent and autonomous systems are interconnected. This will have an effect on a wide range of domains, from the Internet of Things to autonomous mobile robots. The integration of such a variety of MEC services in a inherently distributed architecture requires a robust system for managing hardware resources, balancing the network load and securing the distributed applications. Blockchain technology has emerged a solution for managing MEC services, with consensus protocols and data integrity checks that enable transparent and efficient distributed decision-making. In addition to transparency, the benefits from a security point of view are evident. Nonetheless, blockchain technology faces significant challenges in terms of scalability. In this chapter, we review existing consensus protocols and scalability techniques in both well-established and next-generation blockchain architectures. From this, we evaluate the most suitable solutions for managing MEC services and discuss the benefits and drawbacks of the available alternatives.

Keywords: Edge Computing; Blockchain; Distributed Ledger Technology; Mobile Edge Computing; Multi-Access Edge Computing; Scalability; Distributed Consensus; Internet of Things (IoT);

Jorge Peña Queralta
Turku Intelligent Embedded and Robotic Systems Lab, University of Turku, Turku, Finland
e-mail: jopequ@utu.fi

Tomi Westerlund
Turku Intelligent Embedded and Robotic Systems Lab, University of Turku, Turku, Finland
e-mail: tovewe@utu.fi

1 Introduction

The scope of the Internet of Things (IoT) has been growing over the past decade, encompassing an ever larger ecosystem that spans multiple domains. Some of the most prominent research directions are smart cities [1, 2], vehicular technology [3, 4], or smart healthcare systems [5, 6, 7]. In all these domains, a common factor is that IoT systems are evolving towards more distributed architectures [8]. This shift from more traditional cloud-centric architectures has crystallized in the edge computing paradigm [9, 10, 11]. At the same time, novel technologies are increasingly designed with decentralization in mind from their inception. Among these, blockchain technology is set to be one of the key drivers behind the disruption of the technological landscape in the near future [12, 13]. Decentralized technologies are also the cornerstone behind the Internet 3.0 and Industry 4.0 revolutions that are undergoing [14].

Blockchain technology is already a driver behind decentralized and distributed IoT systems, providing security [15], trust [16, 17], data management [18], peer-to-peer transactions [19], and fault-tolerand middlewares [20]. Blockchain platforms can be divided in two main types depending on how they manage user credentials, which have a direct impact on their applicability: (i) permissionless, or public, and (ii) permissioned, private, or consortium, blockchains. They differentiate in that public blockchains are based on anonymous nodes with equivalent status, while consortium or private blockchains introduce different types of nodes and permissions, some of which require authentication in order perform certain actions. While trust in permissionless blockchains is shared and distributed, in permissioned blockchains there is a series of validator nodes that represented trusted authorities [21].

One of the main issues stopping a wider adoption of blockchain in IoT systems is scalability, an inherent problem to Bitcoin's architecture that multiple researchers have been addressing [22, 23]. While smart contracts have great potential in the IoT and distributed systems in general, their scalability and performance is closely tied to the overall performance of blockchain systems [24]. Nonetheless, multiple advances in recent years have demonstrated that novel technologies can bring significantly higher degrees of scalability and performance to next-generation blockchain systems. Among these, Elastico provided the first implementation of a sharding protocol in a permissionless blockchain [25]. Sharding is a technique that enables the distribution of nodes in a blockchain into subchains for performing parallel validation, thus increasing throughtput and reducing latency. A more recent scalable blockchain is OmniLedger [26], which reports better scalability than Elastico and promises VISA-level latency and throughout if enough nodes form up the network.

Owing to the distributed nature of blockchain systems, and distributed ledger technology (DLT) in general, IoT systems integrating them must already have a distributed architecture by themselves. Therefore, it is only natural that blockchain is integrated at the edge layer in most occasions, which represents the most distributed and interconnected layer of a typical IoT system. While sensors and actuators could be considered more distributed, they are not necessarily capable of node-to-node communication. Through this chapter, we utilize the terms blockchain and distributed ledger equivalently. However, distributed ledger technology (DLT) is often

utilized to include more general systems that do not implement blockchains *per se*, but instead rely on some other type of network or data management architecture. An example of this is IOTA, which utilizes acyclic directed graphs representing more general data structures. The rest of this introduction delves into more details behind the nature of mobile edge computing and its integration with blockchain/DLT technology.

1.1 MEC and Network Slicing

The European Telecommunications Standards Institute (ETSI) has promoted the standardization of Multi-Access Edge Computing (MEC) [27], which shares the acronym with Mobile Edge Computing (MEC). The "multi-access" term puts an emphasis on the multi-tenant infrastructure and better reflects non-cellular operators [28, 27]. In this chapter, we do not make distinctions between the two terms as our focus lays on the role of blockchain with edge computing. MEC standardization has been led by the MEC Industry Specification Group (ISG) since the end of 2014. One of the main objectives of the ETSI MEC ISG is to define the base technologies for distributed and multi-tenant clouds that are meant to be deployed at the edge of the radio access network (RAN) [9]. By deploying data aggregation and processing tasks directly at the edge of the network, MEC services can provide better reliability, lower latency and higher-throughput [29, 30, 7]. We will specifically discuss throughout this paper how blockchain technology can play a key role in terms of security and robustness for the resource management needed in a multi-tenant edge infrastructure, as well as enhance the services that MEC applications can provide [31, 32, 33, 34].

One of the key architectural cornerstones enabling multi-tenancy and co-existing verticals at the MEC layer is network slicing [35]. Network slicing provides the base for interfacing blockchain with other MEC services for a wide array of application scenarios [36]. Network slicing refers to the co-existence of multiple software defined systems and networks (slices) sharing a common hardware infrastructure. Each of the slices can be thus designed independently and optimized for a particular application or business vertical [37]. In particular, slicing for vehicular communication and offloading, together with 5G-and-beyond connectivity, are set to define the mobility of the future [38].

1.2 Integration of Blockchain and MEC

The integration of blockchain within the MEC layer has been object of extensive research over the past few years. Systems integrating blockchain and edge computing can be roughly divided among those in which edge services are part of a larger blockchain system [39, 40, 41], and those in which blockchain is one of the services

enhancing edge services [31, 42, 34, 43, 44, 33]. In this chapter, we are particularly interested in the latter type, as blockchain can provide a key piece in enabling truly distributed, secure and efficient edge computing. With monetization of MEC being a central topic of discussion since its early proposal [29], multiple works have focused towards either enhancing security or utilizing blockchain as a marketplace framework for users to access different applications at the edge [32, 34, 43]. More recently, other works have also delved into the potential of blockchain as a framework for managing edge resources [45, 46, 47, 44], as well as supporting autonomy in distributed robotic systems [36].

From the security point of view, the integration of blockchain technology brings evident benefits to edge computing. Among the main threats identified in a recent report from the European Union Agency for Cybersecurity (ENISA) on 5G networks and edge infrastructure [48], blockchain and DLT technologies can help address multiple remaining challenges. For instance, permissioned DLTs with built-in identity management naturally provide an extra layer of resilience against malicious diversion of network traffic, manipulation of traffic, or authentication traffic spikes. When blockchain tehcnology is applied to resource management, it can serve as a framework to mitigate risks in terms of abuse of third party hosted network functions, manipulation of the network resources orchestrator, or opportunistic and fraudulent usages of shared resources, among others. Moreover, safety-critical applications can benefit from the enhanced security that blockchains and other DLTs provide. These include the automotive sector with vehicle to everything communication routed at the edge [49, 50], and the healthcare sector [34, 51, 52].

1.3 Related Works

Multiple surveys and review papers have recently been published on the convergence of blockchain and mobile edge computing [53, 54, 55, 56]. Other surveys in either the blockchain or edge computing domains also mention the potential for integrating one with another [57, 58, 59, 60]. In these and other works, scalability is often identified as one of the key aspects limiting the adoption of blockchain in edge computing. Nonetheless, these works describe the scalability problem either as a systemic blockchain problem [53], or from a system point of view [55]. Most works also focus on a specific blockchain, Ethereum being the most widely researched blockchain for IoT [54]. In a blockchain, consensus algorithms are the main bottleneck in terms of scalability, i.e., the mechanisms enabling all nodes in the blockchain network to validate transaction and stay synced. Depending on the type of consensus algorithm, the scalability of the system might be limited by either the computational complexity of the algorithm, or its communication complexity. We believe there is a gap in the literature describing how the consensus algorithms affect the scalability from these two points of view. Our objective is to bring further insight in this area, providing a literature review and a discussion on the topic.

In this chapter, we introduce the main consensus algorithms that form the backbone of different blockchain solutions, including newer generation distributed ledgers that do not follow many of the paradigms defined within the Bitcoin and successive blockchains. We then describe what can be the role of edge computing when it integrates blockchain/DLT systems. In particular, we discuss the potential for the different solutions in the IoT, from the point of view of scalability but also discussing the different applications that are most suitable for different blockchain/DLT solutions. We do this from the point of view of consensus algorithms and their computational and communication complexity. Compared to previous works surveying the integration of blockchain and edge computing [53], we provide a novel classification of current research directions from an architectural point of view (Section 3), while giving more insight into how the different consensus algorithms affect the integration of blockchain/DLT and edge computing (Section 4).

1.4 Chapter Structure

The rest of this chapter is organized as follows. In Section 2, we introduce the main consensus algorithms in blockchain systems and other DLTs, together with the most prominent results in highly-scalable and low-latency blockchains. Section 3 then reviews specific applications of blockchain at the MEC layer, and discusses how the different consensus protocols integrate at the edge. In Section 4, we discuss on the best blockchain/DLT solutions for different applications in the IoT, and how next-generation systems that are currently under development might change the IoT and MEC landscape. Finally, Section 5 concludes this work.

2 Blockchain Technology: an Evolving Paradigm

In this section we start with the basics of blockchain technology and move into how the field is evolving towards lower-latency, higher-throughput, and new concepts aimed at increasing flexibility and scalability, such as sharding. We provide a historical point of view on the different consensus algorithms that have been proposed for blockchains and other distributed ledgers, and include an overview of the most prominent so-called third-generation blockchains. The main concepts, consensus protocols and applications are summarized in Fig. 1.

Consensus mechanisms are one of the key aspects within the design of decentralized networked systems or distributed computing systems. Consensus mechanisms are those algorithms that enable multiple independent agents to reach an agreement on a certain value, operation, transaction, or other types of data. In a distributed and decentralized system, different agents, or nodes, need to be able to trust each other. Consensus mechanisms are the enablers of trust among agents. The most popular consensus mechanisms to date in blockchain systems, according to a survey from

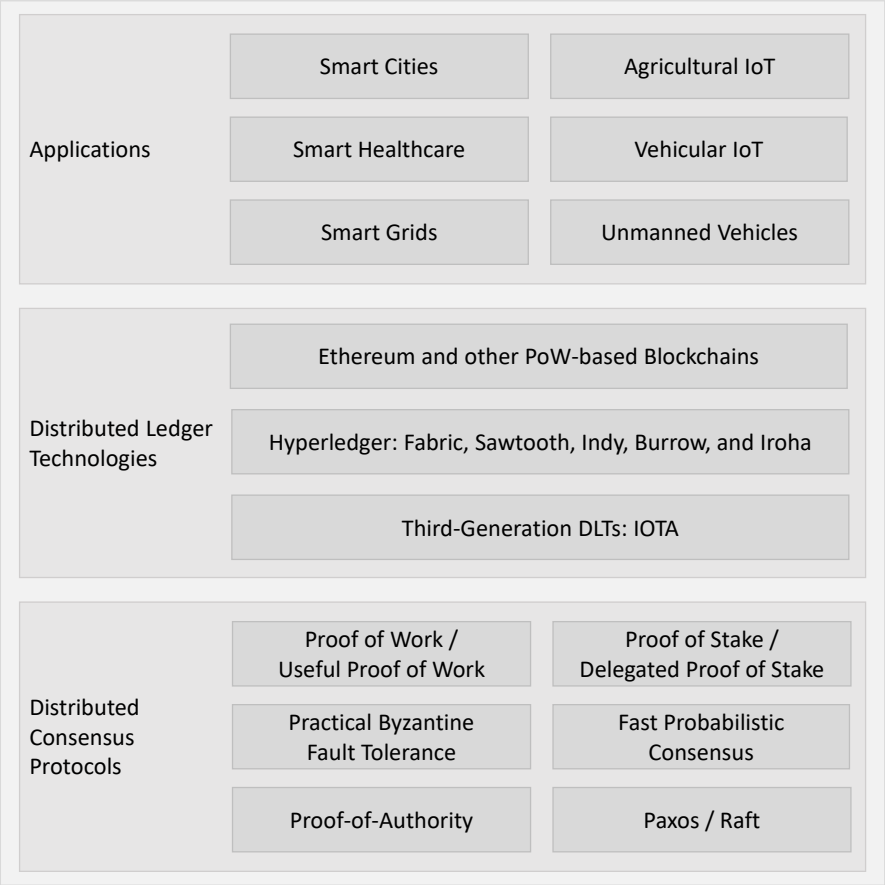

Fig. 1: Blockchain/DLT Consensus protocols, systems, and applications in integration with the Internet of Things.

Li *et al.* [61], are proof of work (PoW), proof of stake (PoS) practical byzantine fault tolerance (PBFT) and delegated proof of stake (DPoS), with other significant approaches including proof of authority (PoA), proof of elapsed time (PoET) or proof of bandwidth (PoB). Apart from some of the more traditional consensus algorithms listed above (e.g. PoW utilized in Bitcoin or Ethereum, and PoS being part of Ethereum 2.0 plans), in this document we also review consensus protocols utilized in third- and fourth-generation distributed ledger systems such as the fast probabilistic consensus (FPC), and the cellular consensus (CC). We also put an emphasis on defining the key technologies behind IOTA, a DLT designed for the IoT and an ideal candidate for integrating DLTs with edge computing.

2.1 Proof of Work

Nakamoto's proof of work designed for Bitcoin [62] has heavily influenced the development of new solutions for newer-generation blockchain systems. The PoW implementation in Bitcoin was a new application for an old algorithm. Originally proposed by [63] as a solution to deter spam activity from email senders, the main idea behind PoW systems has remained unchanged: to request to all networked agents to solve computationally intensive cryptographic problems in order to validate their activity, their identity, or those of another agent. In general terms, a PoW algorithm is, at its most fundamental level, an algorithm that solves a cryptographic problem with a solution that is, in relative terms, hard to find and easy to validate. The computational complexity of the validation of a PoW solution is therefore considerably smaller than the complexity of finding such solution.

Ethereum, the second most popular blockchain system after Bitcoin, also relies on PoW-based consensus to validate new blocks in the blockchain. A block can be roughly defined as each of the entries in the distributed ledger that blockchains implement. A block does not include a single transaction, but often a set of transactions that are near in time. These transactions represent the block's body, where transactions are defined in a generic manner and do not represent only the exchange of cryptocurrencies. Transactions in PoW-based blockchains are not validated individually, but instead all the transactions in a block get validated when the block containing them is validated itself. A block is validated, or mined, by solving a PoW puzzle. The original and most widely used puzzle in blockchains can be summarized as follows: the PoW algorithm must find a block header, which is the result of applying a cryptographic hash function to the content of the block body, satisfying some predefined condition. However, for a fixed hash function and a fixed block body, the resulting hash will always be the same. In order to meet this condition (e.g., finding a hash smaller than a certain value), the algorithm must then find some other value, called a nonce, to be added to the current block body. Finding a nonce is the process often called mining. Once a block is mined, it is added to the blockchain and all other agents in the network can validate the solution. In Bitcoin and other blockchain systems, the miner of a block gets a reward in the form of new cryptocurrency, thus motivating nodes to participate in the transaction validation process.

One of the problems of PoW-based blockchains is that two agents could solve a PoW puzzle at virtually the same time, for the same or different nonces. This can create two branches, or forks, in the blockchain. Nodes are situated in the branch of the solution that they received first. In Bitcoin, a built-in policy establishes that if one fork is longer than the other (or it accumulates more cryptographic complexity), then all agents in the network judge it as the authentic one. This is a practical solution as it is highly improbable that two consecutive blocks will be solved simultaneously by two pairs of nodes. In any case, even if two or more blocks are solved at the same time, at some point one of the forks will become longer. This defines the so-called 51% or double spending attack, as malicious nodes would need to to control at least 51% of the network's computing power in order to be able to introduce a faulty transaction in a block, validate it, and keep validating consecutive nodes in

the corresponding fork so that it is accepted as the canonical fork by the network. When the size of the network and the number of miners increases, the probability of such attack is reduced, thus giving the blockchain its immutability and data integrity properties.

The benefit of having an expensive PoW solution in terms of hardware, energy consumption and time is that it is equally expensive for malicious nodes to attack the network. Part of the security of PoW thus comes from disincentivizing attackers because of the large a priori investment required in order to be able to attack and gain control of the network, which would not pay off even if the attack is successful [64].

2.2 Proof of Useful Work

Part of the research community has argued that taking into account the humongous amount of computational resources and electric energy put into mining to solve PoW puzzles, at least these could be defined in a way that the solutions found would help research in other fields. As an example, King *et al.* proposed the definition of PoW puzzles that would find long chains of primes [65]. Solving these PoW would be then dedicated to solve a mathematical problem which consists on finding the distribution of the Cunningham prime chain. In this case, the Fermat Primality Test would be used to validate the PoW solutions.

A different research approach is the definition of simpler PoW requiring less computational resources in order to reduce the entry barrier and provide a more uniform distribution of mined currency. Pagh *et al.* introduced the concept of Cuckoo hashing, in which the PoW difficulty would remain constant over time [66].

2.3 Proof of Stake

The basis for security and robustness in a PoW system comes from the amount of computational resources needed in order to gain control over the network. Nonetheless, this computational complexity also brings limitations. First, it limits the probability for news nodes to be able to mine new cryptocurrency by themselves if they join a large network. Second, it also limits the number of transactions that can be validated within a certain time interval. For instance, in Bitcoin, it takes an average time of 10 minutes to validate a block and all the transactions it includes [67]. A different consensus approach that does not rely on computational complexity and that has gained momentum in recent years is Proof of Stake (PoS). One of the main objective of PoS systems, which is being introduced, for instance, as part of Ethereum 2.0, is to reduce transaction validation latency. One of the first implementations of PoS in a blockchain system, which showed clear benefits in this direction, was demonstrated with Nxtcoin [68, 69]. The idea behind PoS is to value the cryptocurrency that validating nodes put *at stake*, instead of their computational power. PoS mechanisms

elect validators with a probability proportional to the size of their stake, which is often closely related to the amount of cryptocurrency that the node, or miner, owns. Nodes can lose the total value of their stake if they incur in fraudulent validations. In [70], a similar PoS system was proposed where the probability of selection of the nodes validating transactions was calculated based on both the pure stake and the state of the block being validated in the blockchain.

The 51% attack discussed in the PoW consensus mechanism is still a potential attack vector in a PoS system. However, while in the PoW case attackers need to obtain control over 51% of the network's computing power, which becomes increasingly easy as larger pools monopolizing the mining process are created, in a PoS system an attacker needs control over 51% of the cryptocurrency's total supply. This is, in theory, a more difficult problem than gathering enough computing power.

Owing to the significant reduction of the computational complexity of the consensus algorithms with PoS when compared to PoW, the energy consumption footprint is also reduced. PoS thus provides a more energy-friendly alternative which in turn enables nodes with lower computational capabilities to participate in the blockchain as equals to all others. Multiple authors, such as [71] or [72], have studied the sustainability of Bitcoin's growth and its energy footprint, which researchers estimate to be the equivalent, on a yearly basis, to non-renewable energy resources consumed by entire nations of the size of Czech Republic or Jordan. Nevertheless, this also means that because miners do not need to dedicate large amounts of computational resources to mining, it is easier to perform Sybil attacks spawning multiple identities within a single malicious node.

In general terms, a PoS system relies on a validator or a set of validators which are eligible after depositing part of their stake. In other words, as described by Buterin *et al.* [73], nodes earn the right to propose a block only after locking part of the coins they own on the blockchain. This is an extended definition over the pure PoS system firstly implemented in [74] as part of PPCoin, in which the total miner's stake is directly considered.

2.4 Practical Byzantine Fault Tolerance

The Practical Byzantine Fault Tolerance (PBFT) consensus algorithm was first proposed by Castro *et al.* in 1999 [75]. PBFT was the first algorithm with the ability to operate in large asynchronous networks such as the Internet, while providing over one order of magnitude in processing power improvement over previous methods, allowing for high-performance Byzantine state machine replication, and demonstrating thousands of requests per second. Byzantine fault tolerance can be described as the capacity of a system to maintain proper operation when multiple errors or unexpected behaviour occur within part of the system, but not its totality [76]. In a distributed network and considering the consensus problem, this is equivalent to the ability of the network to provide a robust consensus even in an scenario where

a subset of nodes act maliciously, failing to forward valid data or sending invalid information.

In a PBFT system, nodes are distinguished between validating and not-validating peers [77]. The validating nodes run the consensus algorithm, in which they replicate a state machine and evaluate its result. A client makes a request that is transmitted over the peer-to-peer network through the non-validating nodes, which act as proxies between clients and validators. Non-validating nodes do not participate in the consensus mechanism, but are able to confirm the results. The PBFT algorithm is able to provide consensus across the network when at most one third of the nodes behave arbitrarily or maliciously. Because the validator nodes need to arrive to the same results regarding the client request, the state machine that is replicated must be deterministic.

In comparison with PoW and PoS systems, in PBFT individual transactions can be confirmed without the need to wait for a block including several transactions to be added to the blockchain. In terms of energy efficiency, PBFT requires less computational resources than a PoW consensus, but increases the probability of a Sybil attack, where a malicious node would create multiple instances pretending to be a large number of parties. In practice, PBFT is often combined with a PoW that must be solved in order to join the network and within certain time intervals to ensure that every node in the network is dedicating some minimum computational resources to the collective validation effort. An important benefit of PBFT over PoW and PoS is the low reward variance, as every node can be incentivized. This lowers the reward variance across miners. Nonetheless, the scalability of PBFT is an issue due to the large number of peer-to-peer communication exchanges required.

2.5 Third-Generation DLTs - Beyond Blockchain

Excluding Bitcoin and Ethereum, which represent the majority of the cryptocurrency market capitalization, one of the most successful blockchains within the IoT and industrial domains has been Hyperledger [78]. Launched in 2016 by the Linux Foundation, the Hyperledger project is divided in five main subprojects where blockchain frameworks for different aims are being developed: Fabric, Sawtooth, Indy, Burrow, and Iroha [79]. Among these, Hyperledger Fabric is the most popular, an enterprise-level and production-ready permissioned distributed ledger framework that has already been applied across various industrial fields [80]. The aims behind the project include open-source and cross-industry development of an scalable framework for smart contracts. Through the rest of this chapter, we utilize Hyperledger to refer to Hyperledger Fabric unless otherwise specified.

The consensus mechanism utilized in Hyperledger vary depending on the subproject. For instance, Hyperledger Fabric relies on RAFT [81], while Hyperledger Indy utilizes Plenum, based on Redundant Byzantine Fault Tolerance (RBFT) [82]. Different blockchains following the hyperledger design ideas rely on PBFT or adapted BFT approaches.

In recent years, blockchain technology has evolved towards a wider range of network definitions that do not keep the original structure of a blockchain in terms of how to store data within a distributed ledger. Among these, one of the most prominent distributed open ledgers under development is IOTA [83]. IOTA's backbone is a directed acyclic graph that defines the *tangle*. The tangle is the underlying network upon which IOTA is built. While Bitcoin was born mainly as a distributed cryptocurrency, Ethereum evolved from it into a platform for smart contracts, and Hyperledger is intended for industrial use, IOTA was specifically designed with the IoT in mind [84]. In IOTA, there are no miner or validator nodes confirming transactions, but instead each user must participate in the validation of two transactions before being able to issue a new one on its own. This approach, together with the tangle's structure, makes IOTA highly scalable and free to use. IOTA's development is open-source and led by the IOTA foundation.

IOTA's consensus protocol is defined within the Concordice system [85]. The main differentiating aspect of IOTA's tangle is the fact that multiple disconnected subnetworks can coexist for certain periods of time. This means, for instance, that while a blockchain cannot contain two conflicting transactions in committed blocks, the tangle might temporarily contain two such transactions. IOTA deals with this, however, in a similar manner as Bitcoin does: the fact that a transaction is included in the blockchain does not automatically mean it is valid, as two forks of the chain might exist until one is deemed longer and this valid. Therefore, in both cases there is only information about the *probability* of a transaction being valid, which increases as the blockchain, or the tangle, grow after that given transaction. In order to make a decision on two conflicting transactions in IOTA and reach a consensus across the network, Concordice proposes two consensus protocols: the fast probabilistic consensus (FPC) and the cellular consensus (CC). FPC, introduced in [86], is a leaderless probabilistic binary consensus protocol. FPC has low complexity from the communication point pf view, and is robust in a Byzantine infrastructure. As with PBFT, the basic idea behind FPC is voting. In any case, IOTA is still under development and is not production-ready. More detailed information on IOTA's consensus and CC is available in [87] and [88].

Other DLT solutions claiming to be third-generation blockchain are Nano [89], with its underlying block lattice, and Skycoin [90], aimed at powering the Web 3.0. While Nano and IOTA are recent technologies, Skycoin has been under development for several years and was born out of a series of external audits into Bitcoin, which revealed the different flaws in the PoW consensus protocol.

2.6 Smart Contracts

Second-generation blockchain systems, largely represented by the Ethereum blockchain, were defined as those introducing the ability of executing distributed programs within the blockchain itself, therefore extending their applicability beyond cryptocurrency transactions and into the validation of more general types of transactions.

These programs that can be executed within a blockchain are called *smart contracts*, with one of the most notorious implementations being part of the Ethereum Virtual Machine and its corresponding stack [91], which provides a Turing complete language as part of its framework [92]. Ethereum also introduced a new programming language to be dedicated to the development and implementation of smart contracts: Solidity [93]. Smart contracts as defined with Solidity code can be seen as a set of instructions defining transitions between states of the program, with both the data representing the different states and the code defining the transitions being stored at specific addresses within the Ethereum blockchain.

In Ethereum, smart contracts are part of the Ethereum Virtual Machine (EVM) [94]. The EVM is based on the existence of contract accounts in the blockchain, which extend the functionality of external accounts, those controlled by a human or network node through a public-private key pair. Contract accounts operate in an automated way as a function of the code stored within the account. While external accounts are defined based on their key pair, with an address determined based on the public key being assigned to each node joining the network, contract accounts have addresses that are determined when the contract is created. In Ethereum, the address space is shared among both types of accounts. Contract accounts are created through transactions that have a null or empty recipient. Those transactions must contain code that outputs the smart contract's code, which is then generated when the transaction's code is executed within the EVM. In general terms, transactions including a payload and Ether (Ethereum's cryptocurrency) between external accounts in Ethereum are extended so that when a transaction's target account is a contract account containing a set of code instructions, these are executed given the payload in the transaction. A key concept in Ethereum is gas. Upon creation, transactions are assigned a definite quantity of gas. The gas is a measure of the processing power that will be dedicated to that transaction. In other words, the gas is the transaction fee. The gas is initially charged into the transaction, and its reserve gradually decreases as a function of a set of predefined rules when the EVM executes the different transaction instructions. The gas that is left is refunded to the transaction creator. The gas price, which is paid upfront, is decided by the creator node. Miners, which obtain the gas price as a reward, decide which transactions to mine based on the amount of gas included. Therefore, the gas price is decided based on the market and the desired priority for a specific transaction.

2.7 Sharding and Scalability

While second-generation blockchains introduced new functionality and improvements over Bitcoin-based blockchains at different levels, one of the main challenges in blockchain systems remained: scalability [95]. This is mostly due to the large and increasing amount of computational resources required for mining. From the communication point of view, Bitcoin and other similar blockchains only require one broadcast per block, and therefore the main bottleneck comes from computation

(which cannot be directly decreased while maintaining security). In PBFT-based systems, multicast messages are required for validation, and thus the main scalability problem is the communication cost [22] (which cannot be directly reduced either without compromising security and robustness of the consensus mechanism). Multiple research efforts have been directed towards the realization of more scalable systems, with new blockchains based on PoS and PBFT showing promising results. Elastico, introduced in [25], was one of the first scalable blockchains that introduced the concept of sharding: to divide the network in subnetworks, or *shards*, that would validate transactions in parallel. Elastico was the first blockchain system to provide a full implementation of a sharding scheme for a permisionless blockchain. A different early sharding proposal was presented in [96], where Merklix trees are utilized to merge the state of the different shards into the global blockchain state [97, 98].

Another blockchain system aimed at scalability that has had an important impact on subsequent research is OmniLedger [26]. Omniledger scales linearly with the number of nodes in the blockchain, and reports transaction times able to match credit card standards if the size of the network arrives to a certain threshold. The key difference with Elastico in terms of scalability is that in Elastico the network performance scales with the computational power in a linear fashion, while in Omniledger it does so with the number of validator nodes. In Hyperledger, the scalability of the network has seen significant improvements since the release of Fabric 1.1.0 [99]. Moreover, the number of channels can be scaled with little to no impact on performance according to the same report.

Perhaps the biggest effort that is currently being put into the development of a truly decentralized, permissionless and scalable yet secure blockchain is the design and development of Ethereum 2.0 [100], where huge amounts of computing resources will be no longer required for mining [101]. The Ethereum Foundation and other developers behind Ethereum 2.0 have embraced Proof of Stake as the main consensus mechanism, while still utilizing PoW to secure the network, and the concept of sharding towards scalability. The consensus is based on the Casper protocol [102], which incentives for mining have been described in [73]. The impact that shards have on transaction scalability is relatively clear, with a much larger throughput being possible in terms of transactions validated per second. Nonetheless, it is not straightforward to extend the implementation of smart contracts with sharding. As smart contracts have associated a series of data states corresponding to their code, each state change can be though of as a transaction. Contracts can be executed within a single shard, or a cross-shard synchronization mechanism must exist to allow for data to flow between shards. In [26], the authors introduced introduced Atomix, a client-driven lock/unlock protocol, to ensure that a single transaction can be committed across multiple shards, while enabling the possibility of unlocking rejected transaction proofs in specific shards. The original Atomix state machine can be extended to accommodate the execution of smart contracts across shards.

3 Blockchain Technology for Mobile Edge Computing

This section reviews and classifies the existing research in the integration of blockchain and MEC from an architectural point of view. We classify the different approaches on three main categories, illustrated in Fig. 2. The first category encompasses works providing a system-level integration where a blockchain is one of the key pieces at the heart of the edge infrastructure, managing services and resources. The second category includes approaches that utilize blockchain as a middleware between the edge infrastructure (hardware and software) and the third-party services being provided through MEC. Finally, the last category comprises those works where the blockchain is part of individual applications, for aspects such as security or identity management.

In general terms, Ethereum is the most widely applied blockchain platform in the IoT, owing to the maturity of its smart contracts framework enabling complex interactions between data producers and consumers [103, 104]. In the same area, Hyperledger has potential to disrupt the IoT with more scalable solutions and the ability to run distributed programs as chaincode [105]. In all these cases, nonetheless, the blockchain runs in embedded edge gateways providing stable connectivity, and where enough power and computational resources is available. With the potential to reach embedded devices at the sensor layer, and being developed specifically for the IoT, IOTA is set to play an increasingly important role. Owing to its low inherent computational requirements and being highly scalable, IOTA is the ideal candidate for edge computing systems and hardware.

3.1 MEC Resource and Service Orchestration with Blockchain

One of the most critical points at the edge is resource orchestration [29]. In order to enable a wide variety of use cases, multi-tenant applications, and ad-hoc deployment of different modules, MEC infrastructure needs to be able to manage its resources in real time, while also orchestrating how the network is being utilized. This includes processes from allocating hardware resources for the different virtualized applications to managing the spectrum or the bandwidth that might be in use for computational offloading by different service providers.

Blockchain technology can provide multiple advantages to orchestration at the edge: enhanced security and identity management, together with distributed consensus algorithms to implement the resource allocation decision processes. In this area, EdgeChain was introduced by Zhu *et al.* as a middleware platform to deploy third-party applications across the MEC layer [31]. In [33], the authors introduce a blockchain framework that relies on smart contracts for managing network bandwidth and resource allocation in a distributed and collaborative computational offloading framework. In [36], a similar idea is extended towards managing network infrastructure and the available computational resources focused at enhancing autonomy of self-driving cars and other autonomous robots forming distributed

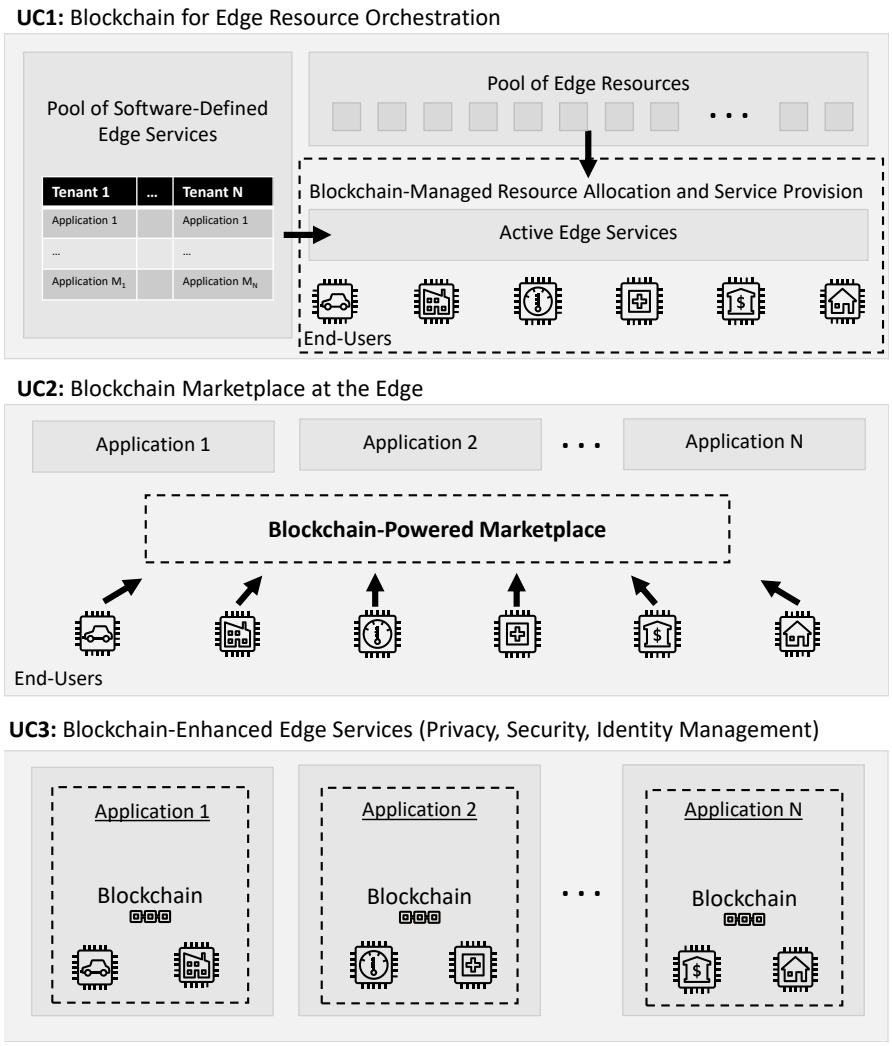

Fig. 2: Main use cases for blockchain within edge computing systems. (UC1) Blockchain-powered resource allocation and service provision; (UC2) Blockchain-powered marketplace for interfacing users and services; (UC3) Blockchain-enhanced individual edge services relying on blockchain technology for security, privacy, data management and audits or identity management, among others.

robotic systems. In this paper, the blockchain MEC slice was the key slice managing the deployment of applications across other MEC slices supporting different verticals within the automotive sector. Further adoption of blockchain for computational offloading will require, however, higher-bandwidth and lower-latency blockchain frameworks enabling real-time sensor data to be streamed for applications such as autonomous mobile robots [30, 106].

Resource orchestration processes have underlying optimization algorithms that can be implemented either in a more traditional deterministic manner, or relying on machine learning models. Several authors have proposed the utilization of deep reinforcement learning for computational offloading in blockchain-powered edge computing. In [107], the authors demonstrate an approach that is able to improve long-term performance in a computational offloading scheme, with an adaptive genetic algorithm to improve the exploration processes while learning. In [43], the authors describe different situations in which blockchain can support resource management at the edge with deep reinforcement learning: spectrum sharing, vehicle-to-vehicle energy trading, computational offloading, or device-to-device content caching.

3.2 Blockchain for a MEC Services Marketplace

DLT can also provide a platform for building a marketplace between end-users and third-party edge application through either a transparent, secure and auditable monetization framework or as a middleware for sharing data securely between producers and consumers. In the former direction, Xiong *et al.* deployed a blockchain at the edge to enable resource-constrained devices producing data to sell it to third-party applications [32]. The pricing scheme introduced in the paper models the interactions within the IoT as market activities and the blockchain represented the framework for regulation of such activities. Distributed marketplaces based on blockchain for MEC services often utilize Ethereum as a base and the InterPlanetary File System (IPFS) for data storage. Examples are available in [108] or [109]. A study describing the different challenges and opportunities is available in [110].

3.3 Blockchain-Powered MEC Services

In [49], the authors describe how blockchain can play a key enabled role in interconnected vehicles from the security point of view. In particular, blockchain is exploited for data management, but also for energy management in electric vehicles, with the authors proposing blockchain inspired data coins and energy coins. An edge computing security scheme is proposed including these two interaction aspects. An approach more related to the nature of blockchain as a cryptocurrency framework was proposed by Liu *et al.* in [41], where the authors present an offloading framework not for data but for the blockchain itself and related mining operations. In

general, blockchain can support edge services by providing enhanced privacy and security [40], decentralized data management [18], or identity management [111].

4 Performance and Scalability of DLTs at the Edge

In this section we describe the benefits and drawbacks of the different consensus protocols and DLT solutions for each of the three main use cases defined in the previous section and illustrated in Fig. 2, as well as for edge computing in the industrial internet of things. A basic classification of some of the protocols introduced in the previous section from the point of view of the capabilities of embedded IoT systems is given in Table 1.

Table 1: Comparison of consensus protocols in terms of their applicability within resource-constrained devices in the IoT.

	PoW	**PoS/DPoS**	**PBFT**	**Concordice**
Computationally-Constrained Devices	✗	✓	✗	✓
Communication-Constrained Devices	-	✗	✗	✓
Intermittent Connectivity	✗	✗	✗	✓
Independent Subnetworks	✗	✗*	✓**	✓***
Production-Ready Platform	✓	✓	✓	✗

*Recent proposals implementing sharding might be considered subnetworks, however here we refer to the ability of specifically creating a subnetwork from a given set of nodes.
*Channels in Hyperledger enable data separation but need to remain connected to the main net.
**The tangle in IOTA enables sets of nodes to be disconnected for certain periods of time and rejoin the network later on.

4.1 Blockchain Technology in Resource-Constrained Devices

Consensus protocols in the different DLTs are the key performance indicators, and they are directly related to the minimum capabilities that nodes in the network must meet. In PoW-based blockchains, including Bitcoin and Ethereum, resource-constrained devices in the IoT that are potentially battery powered do not have the ability to participate as full nodes in the network. In Ethereum, nonetheless, the blockchain has adapted to some extent towards embedded IoT devices. For instance, the Zerynth Ethereum library provides basic capability to embedded microcontrollers running MicroPython [112]. It enables sensor nodes to create signed transactions and execute contract calls.

Hyperledger Fabric and IOTA, designed with scalability in mind, do not have such strong computational requirements. The consensus protocols at the hearth of Hyperledger, however, have high communication complexity and therefore require nodes to be able to communicate frequently and with low-latency. Hyperledger can therefore run in embedded IoT edge gateways with wired internet connection but its extendability to wireless and potentially battery-powered sensor nodes is limited. In this area, IOTA has a comparative advantage. In particular, STMicroelectronics has collaborated with the IOTA foundation in the development of X-CUBE-IOTA [113], a complete middleware that enables IoT sensor nodes based on STM32 microcontrollers to build IOTA applications and access the IOTA distributed ledger directly.

In terms of communication-constrained devices, low-power wide area networks (LPWANs) have emerged in recent years as a solution for extending the range of applications, with LoRa and LoRaWAN being the most prominent radio and network technologies [7, 114]. Edge computing is a natural paradigm to be integrated with LPWAN networks owing to the low-bandwidth available and thus the need to preprocess large amounts of raw data [115, 116, 117]. However, the integration of blockchain into LPWAN networks is not direct [118]. Current efforts deploy the blockchain either at the LPWAN gateways, which often have wired internet connection, or at the back-end servers [119, 120]. More interesting use cases will be possible when the blockchain nodes can be interconnected via low-bandwidth and high-latency LPWAN links, which might be soon possible with IOTA and STM.

4.2 Application Scenarios

From the point of view of edge computing as a system encompassing multiple independent applications, the simplest use case is such in which blockchains are managed by each application independently. This allows for the same orchestration algorithms to remain in place, as well as co-existence of blockchain-based and other applications running at the edge. Depending on the nature of each of the applications, all of the DLT solutions presented in this chapter might be applied. For general IoT systems where data is gathered from sensor nodes and transactions between either the user or the sensors and the application back-end (which may or may not be deployed entirely at the edge) are relatively simple, then IOTA stands out by providing free transactions. This can be a key differentiating point in applications where data is routinely gathered and does not have specific value. Because IOTA's consensus is built in a way that all nodes need to take the validator role before being able to commit transactions, nodes do no need an additional incentive to validate and therefore there is no need for a transaction fee as with other blockchain platforms. If more complex transactions are required, with either real-time interaction between users or a user and sensor data being processed, then smart contracts might be needed. Ethereum is by far the most extended and used blockchain platform for smart contracts, and therefore it would be natural to rely on it. This will be an even better solution when Ethereum 2.0 is available. Nonetheless, relying on Ethereum or

similar solutions involves an extra transaction cost, due to the need for mining new cryptocurrency to compensate nodes participating in the validation process. Alternatively, private Ethereum networks can be deployed and infrastructure managed by the application developers. This is specially important in PoW-based systems, but also in PoS systems as otherwise nodes would have no incentives on putting their stakes at risk.

When blockchain is utilized to power a marketplace of services at the edge, the cryptocurrency that blockchains build upon might play a more important role with the introduction of monetization. In this sense, monetization does not necessarily refer only to paying for services, but can also encompass the edge resources that services rely on [29]. Similar to the previous case, the choice of DLT framework has a significant dependence on the type of data management and processing that needs to be done. For simple applications in which services and end-users are predefined and communicate independently, then IOTA can provide a fast and scalable framework, while Hyperledger could be an alternative if there is enough infrastructure set to sustain the blockchain and validate transactions. These applications can cover a wide variety of scenarios: paying a highway toll, exchange of information for coordination between autonomous cars, track-and-trace in the logistics sector, or providing digital identity to citizens in a smart city. In all these cases, a common denominator is that the transfers of value, or information, are small and frequent in time, and therefore there is not enough incentive to utilize other blockchain platforms such as Ethereum where transactions involve a fee. Hyperledger, nonetheless, is only a viable option if either public or private infrastructure supports its use without an impact on the end-user. For more complex applications, both Hyperledger and Ethereum provide extensive support for smart contracts and execution of distributed applications.

The last of the use cases presented in the previous section, and involving the most complex system-level integration of DLT technology at the edge layer is resource allocation and service provision. In this case, different optimization algorithms in which the resource orchestrator relies need to be implemented on top of the blockchain for transparent management of resources. The processes involved in dynamic resource allocation and service provision are complex and therefore require blockchains able of running smart contracts. Ethereum provides a suitable platform from the functionality point of view, but lacks the ability to scale and the low control over latency would significantly affect the real-time allocation of resources. Moreover, the computational power needed to validate transactions would reduce the availability of edge resources. Until Ethereum 2.0 or a more scalable solution is available, Hyperledger has multiple competitive advantages in this area.

A different application scenario that has not been directly covered in the previous section is the industrial IoT. Industrial scenarios often differentiate in that they operate on private networks. Moreover, safety-critical applications require more control over the network parameters as well as over the data management itself. In these directions, Hyperledger Fabric stands out, with design decisions targeting industrial use cases since its inception. Not only does a permissioned Hyperledger blockchain provide a secure framework for management of identities and network control, but

it is the ability to separate data across channels that can provide wider adoption in privacy-critical and safety-critical use cases.

5 Conclusion and Future Work

We have reviewed the most important consensus protocols in traditional blockchains and novel distributed ledger technologies, together with the different applications and use cases resulting of the integration of blockchain and edge computing. In particular, we have described how the underlying consensus protocols affect the applicability of the different DLT systems for edge computing, with an emphasis on the current research trends in terms of scalability and performance. We have outlined the main benefits and drawbacks of Ethereum, Hyperledger and IOTA in four main use cases: (i) orchestration of edge resources and services, (ii) implementation of a marketplace of edge services, (iii) enhancing security, privacy or identity management of individual edge services, and (iv) providing a framework for data management in the industrial Internet of Things.

Acknowledgements This work was supported by the Academy of Finland's AutoSOS project with grant number 328755.

References

1. Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.
2. Ola Salman, Imad Elhajj, Ali Chehab, and Ayman Kayssi. Iot survey: An sdn and fog computing perspective. *Computer Networks*, 143:221–246, 2018.
3. Celimuge Wu, Zhi Liu, Di Zhang, Tsutomu Yoshinaga, and Yusheng Ji. Spatial intelligence toward trustworthy vehicular iot. *IEEE Communications Magazine*, 56(10):22–27, 2018.
4. Jorge Peña Queralta, Tuan Nguyen Gia, Hannu Tenhunen, and Tomi Westerlund. Collaborative mapping with ioe-based heterogeneous vehicles for enhanced situational awareness. In *2019 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. IEEE, 2019.
5. Charalampos Doukas and Ilias Maglogiannis. Bringing iot and cloud computing towards pervasive healthcare. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 922–926. IEEE, 2012.
6. Ammar Awad Mutlag, Mohd Khanapi Abd Ghani, Net al Arunkumar, Mazin Abed Mohammed, and Othman Mohd. Enabling technologies for fog computing in healthcare iot systems. *Future Generation Computer Systems*, 90:62–78, 2019.
7. Jorge Peña Queralta, Tuan Nguyen Gia, Hannu Tenhunen, and Tomi Westerlund. Edge-AI in LoRa based healthcare monitoring: A case study on fall detection system with LSTM Recurrent Neural Networks. In *2019 42nd International Conference on Telecommunications, Signal Processing (TSP)*, 2019.
8. Sam Edwards and Ioannis Profetis. Hajime: Analysis of a decentralized internet worm for iot devices. *Rapidity Networks*, 16, 2016.
9. Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computing—a key technology towards 5g. *ETSI white paper*, 11(11):1–16, 2015.

10. Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.
11. L. Qingqing, F. Yuhong, J. Peña Queralta, T. N. Gia, Z. Zou, H. Tenhunen, T. Westerlund. Edge Computing for Mobile Robots: Multi-Robot Feature-Based Lidar Odometry with FPGAs. In *12th ICMU*. IEEE, 2019.
12. Melanie Swan. *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.
13. Sarah Underwood. Blockchain beyond bitcoin, 2016.
14. Yang Lu. Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6:1–10, 2017.
15. Yongfeng Qian, Yingying Jiang, Jing Chen, Yu Zhang, Jeungeun Song, Ming Zhou, and Matevž Pustišek. Towards decentralized iot security enhancement: A blockchain approach. *Computers & Electrical Engineering*, 72:266–273, 2018.
16. Juah C Song, Mevlut A Demir, John J Prevost, and Paul Rad. Blockchain design for trusted decentralized iot networks. In *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, pages 169–174. IEEE, 2018.
17. Mohamed Tahar Hammi, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni. Bubbles of trust: A decentralized blockchain-based authentication system for iot. *Computers & Security*, 78:126–142, 2018.
18. Gbadebo Ayoade, Vishal Karande, Latifur Khan, and Kevin Hamlen. Decentralized iot data management using blockchain and trusted execution environment. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 15–22. IEEE, 2018.
19. Jollen Chen. Devify: Decentralized internet of things software framework for a peer-to-peer and interoperable iot device. *ACM SIGBED Review*, 15(2):31–36, 2018.
20. Penn H Su, Chi-Sheng Shih, Jane Yung-Jen Hsu, Kwei-Jay Lin, and Yu-Chung Wang. Decentralized fault tolerance mechanism for intelligent iot/m2m middleware. In *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pages 45–50. IEEE, 2014.
21. Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*, pages 557–564. IEEE, 2017.
22. Marko Vukolić. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International workshop on open problems in network security*. Springer, 2015.
23. Ghassan Karame. On the security and scalability of bitcoin's blockchain. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1861–1862, 2016.
24. Mattias Scherer. Performance and scalability of blockchain networks and smart contracts, 2017.
25. Loi Luu, Viswesh Narayanan, Chaodong Zheng, Kunal Baweja, Seth Gilbert, and Prateek Saxena. A secure sharding protocol for open blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 17–30, 2016.
26. Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 583–598. IEEE, 2018.
27. Sami Kekki, Walter Featherstone, Yonggang Fang, Pekka Kuure, Alice Li, Anurag Ranjan, Debashish Purkayastha, Feng Jiangping, Danny Frydman, Gianluca Verin, et al. Mec in 5g networks. *ETSI white paper*, 28:1–28, 2018.
28. Sonia Shahzadi, Muddesar Iqbal, Tasos Dagiuklas, and Zia Ul Qayyum. Multi-access edge computing: open issues, challenges and future perspectives. *Journal of Cloud Computing*, 6(1):30, 2017.
29. Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3), 2017.
30. L. Qingqing, J. Peña Queralta, T. N. Gia, Z. Zou, H. Tenhunen, T. Westerlund. Visual Odometry Offloading in Internet of Vehicles with Compression at the Edge of the Network. In *12th International Conference on Mobile Computing and Ubiquitous Networking*, 2019.

31. He Zhu, Changcheng Huang, and Jiayu Zhou. Edgechain: Blockchain-based multi-vendor mobile edge application placement. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pages 222–226. IEEE, 2018.
32. Zehui Xiong, Yang Zhang, Dusit Niyato, Ping Wang, and Zhu Han. When mobile blockchain meets edge computing. *IEEE Communications Magazine*, 56(8):33–39, 2018.
33. Jorge Peña Queralta and Tomi Westerlund. Blockchain-powered collaboration in heterogeneous swarms of robots. *Frontiers in Robotics and AI (to appear)*, 2020. Presented at the Symposium on Blockchain for Robotic and AI Systems, MIT Media Lab.
34. MD Abdur Rahman, M Shamim Hossain, George Loukas, Elham Hassanain, Syed Sadiqur Rahman, Mohammed F Alhamid, and Mohsen Guizani. Blockchain-based mobile edge computing framework for secure therapy applications. *IEEE Access*, 6:72469–72478, 2018.
35. 3GPP. Study on architecture for next-generation system rel. 14. *Techical Report*, 2016.
36. Jorge Peña Queralta, Li Qingqing, Zhuo Zou, and Tomi Westerlund. Enhancing autonomy with blockchain and multi-acess edge computing in distributed robotic systems. In *The Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2020.
37. N. Alliance. Description of network slicing concept. *NGMN 5G P*, 1:1, 2016.
38. Fabio Giust, Vincenzo Sciancalepore, Dario Sabella, Miltiades C Filippou, Simone Mangiante, Walter Featherstone, and Daniele Munaretto. Multi-access edge computing: The driver behind the wheel of 5g-connected cars. *IEEE Communications Standards Magazine*, 2(3):66–73, 2018.
39. Roberto Casado-Vara, Fernando de la Prieta, Javier Prieto, and Juan M Corchado. Blockchain framework for iot data quality via edge computing. In *Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems*, pages 19–24, 2018.
40. A. Nawaz, J. Peña Queralta, T. N. Gia, H. Kan, T. Westerlund. Edge AI and Blockchain for Privacy-Critical and Data-Sensitive Applications. In *The 12th International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*, 2019.
41. Mengting Liu, F Richard Yu, Yinglei Teng, Victor CM Leung, and Mei Song. Joint computation offloading and content caching for wireless blockchain networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 517–522. IEEE, 2018.
42. Jorge Peña Queralta, Li Qingqing, Tuan Nguyen Gia, Hong-Linh Truong, and Tomi Westerlund. End-to-end design for self-reconfigurable heterogeneous robotic swarms. In *The 16th International Conference on Distributed Computing in Sensor Systems*. IEEE, 2020.
43. Yueyue Dai, Du Xu, Sabita Maharjan, Zhuang Chen, Qian He, and Yan Zhang. Blockchain and deep reinforcement learning empowered intelligent 5g beyond. *IEEE Network*, 33, 2019.
44. Nguyen Cong Luong, Zehui Xiong, Ping Wang, and Dusit Niyato. Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.
45. Mayra Samaniego and Ralph Deters. Hosting virtual iot resources on edge-hosts with blockchain. In *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pages 116–119. IEEE, 2016.
46. Mayra Samaniego and Ralph Deters. Using blockchain to push software-defined iot components onto edge hosts. In *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, pages 1–9, 2016.
47. Mayra Samaniego and Ralph Deters. Virtual resources & blockchain for configuration management in iot. *Journal of Ubiquitous Systems & Pervasive Networks*, 9(2):1–13, 2017.
48. The European Union Agency for Cybersecurity. Threat assessment for the fifth generation of mobile telecommunications networks (5g). *ENISA*, 2019.
49. Hong Liu, Yan Zhang, and Tao Yang. Blockchain-enabled security in electric vehicles cloud and edge computing. *IEEE Network*, 32(3):78–83, 2018.
50. Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 6(3):4660–4670, 2018.

51. T. N. Gia, A. Nawaz, J. Peña Queralta, T. Westerlund. Artificial Intelligence at the Edge in the Blockchain of Things. In *8th EAI International Conference on Wireless Mobile Communication and Healthcare*, 2019.
52. Eduardo Castelló Ferrer, Ognjen Rudovic, Thomas Hardjono, and Alex Pentland. Robochain: A secure data-sharing framework for human-robot interaction. *arXiv preprint arXiv:1802.04480*, 2018.
53. Ruizhe Yang, F Richard Yu, Pengbo Si, Zhaoxin Yang, and Yanhua Zhang. Integrated blockchain and edge computing systems: A survey, some research issues and challenges. *IEEE Communications Surveys & Tutorials*, 21(2):1508–1532, 2019.
54. Pietro Danzi, Anders E Kalør, Čedomir Stefanović, and Petar Popovski. Delay and communication tradeoffs for blockchain systems with lightweight iot clients. *IEEE Internet of Things Journal*, 6(2):2354–2365, 2019.
55. Seyednima Khezr, Md Moniruzzaman, Abdulsalam Yassine, and Rachid Benlamri. Blockchain technology in healthcare: A comprehensive review and directions for future research. *Applied Sciences*, 9(9):1736, 2019.
56. Dinh C Nguyen, Pubudu N Pathirana, Ming Ding, and Aruna Seneviratne. Blockchain for 5g and beyond networks: A state of the art survey. *Journal of Network and Computer Applications*, page 102693, 2020.
57. Weichao Gao, William G Hatcher, and Wei Yu. A survey of blockchain: techniques, applications, and challenges. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–11. IEEE, 2018.
58. Archana Prashanth Joshi, Meng Han, and Yan Wang. A survey on security and privacy issues of blockchain technology. *Mathematical Foundations of Computing*, 1(2):121–147, 2018.
59. Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. Edge computing: A survey. *Future Generation Computer Systems*, 97:219–235, 2019.
60. Jose Moura and David Hutchison. Fog computing systems: State of the art, research issues and future trends. *arXiv preprint arXiv:1908.05077 [v2]*, pages 1–32, 2020.
61. Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 2017.
62. Satoshi Nakamoto et al. *Bitcoin: A peer-to-peer electronic cash system*. 2008.
63. Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Annual International Cryptology Conference*, pages 139–147. Springer, 1992.
64. Giang-Truong Nguyen and Kyungbaek Kim. A survey about consensus algorithms used in blockchain. *Journal of Information processing systems*, 14(1), 2018.
65. Sunny King. Primecoin: Cryptocurrency with prime number proof-of-work. 1:6, 2013.
66. Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. *Journal of Algorithms*, 51(2):122–144, 2004.
67. Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better—how to make bitcoin a better currency. In *Financial Cryptography and Data Security*. Springer, 2012.
68. Serguei Popov. A probabilistic analysis of the nxt forging algorithm. *Ledger*, 1:69–83, 2016.
69. Nxt Wiki. *Whitepaper: Nxt*. Nxtwiki. org [online] https://nxtwiki. org, 2018.
70. Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. In *International Conference on Financial Cryptography and Data Security*. Springer, 2016.
71. Karl J O'Dwyer and David Malone. *Bitcoin mining and its energy footprint*. IET, 2014.
72. Alex De Vries. Bitcoin's growing energy problem. *Joule*, 2(5):801–805, 2018.
73. Vitalik Buterin, Daniel Reijsbergen, Stefanos Leonardos, and Georgios Piliouras. Incentives in ethereum's hybrid casper protocol. *arXiv preprint arXiv:1903.04205*, 2019.
74. Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19, 2012.
75. Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186, 1999.
76. RM Keichafer, Chris J. Walter, Alan M. Finn, and Philip M. Thambidurai. The maft architecture for distributed fault tolerance. *IEEE Transactions on Computers*, 37(4), 1988.
77. Joao Sousa, Alysson Bessani, and Marko Vukolic. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In *48th DSN*, pages 51–58. IEEE, 2018.

78. C. Cachin. Architecture of the hyperledger blockchain fabric. In *Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, page 4, 2016.
79. Chinmay Saraf and Siddharth Sabadra. Blockchain platforms: A compendium. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6. IEEE, 2018.
80. Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.
81. Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, pages 305–319, 2014.
82. Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. Rbft: Redundant byzantine fault tolerance. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 297–306. IEEE, 2013.
83. S Popov. The tangle, iota whitepaper. Technical report, IOTA, Tech. Rep.[Online]. Available: https://iota. org/IOTA_Whitepaper. pdf, 2018.
84. M Divya and Nagaveni B Biradar. Iota-next generation block chain. *International Journal Of Engineering And Computer Science*, 7(04):23823–23826, 2018.
85. Serguei Popov, Hans Moog, Darcy Camargo, Angelo Capossele, Vassil Dimitrov, Alon Gal, Andrew Greve, Bartosz Kusmierz, Sebastian Mueller, Andreas Penzkofer, et al. The coordicide, 2020.
86. Serguei Popov and William J Buchanan. Fpc-bi: Fast probabilistic consensus within byzantine infrastructures. *arXiv preprint arXiv:1905.10895*, 2019.
87. Daniel Ramos and Gabriel Zanko. Review of iota foundation as a moving force for massive blockchain adoption in different industry sectors.
88. KENRIC NELSON and ANDRÉ VILELA. Majority vote dynamics for iota transaction consensus. 2020.
89. Colin LeMahieu. Nano: A feeless distributed cryptocurrency network. *Nano [Online resource]. URL: https://nano. org/en/whitepaper (date of access: 24.03. 2018)*, 2018.
90. Skycoin.com. Skycoin whitepaper v1.2. Technical report, [Online]. Available: https://downloads.skycoin.com/whitepapers/Skycoin-Whitepaper-v1.2.pdf, 2020.
91. Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.
92. Everett Hildenbrandt, Manasvi Saxena, Nishant Rodrigues, Xiaoran Zhu, Philip Daian, Dwight Guth, Brandon Moore, Daejun Park, Yi Zhang, Andrei Stefanescu, et al. Kevm: A complete formal semantics of the ethereum virtual machine. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 204–217. IEEE, 2018.
93. Ethereum Revision 7709ece9. *Solidity Documentation*. Solidity Read The Docs [online] https://solidity.readthedocs.io/en/v0.5.12/., 2016-2019.
94. Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.
95. Dejan Vujičić, Dijana Jagodić, and Siniša Randić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *17th INFOTEH-JAHORINA*, pages 1–6. IEEE, 2018.
96. Deadalnix's den. *Using Merklix tree to shard block validation*. [online] https://deadalnix.me/2016/11/06/, 2016.
97. Deadalnix's den. *Introducing Merklix tree as an unordered Merkle tree on steroid*. Accessed October 2019 [online] https://www.deadalnix.me/2016/09/24/introducing-merklix-tree-as-an-unordered-merkle-tree-on-steroid/, 2016.
98. Bo Qin, Jikun Huang, Qin Wang, Xizhao Luo, Bin Liang, and Wenchang Shi. Cecoin: A decentralized pki mitigating mitm attacks. *Future Generation Computer Systems*, 2017.
99. C. Ferris. "does hyperledger fabric perform at scale? *Blockchain Pulse: IBM Blockchain Blog*, 2, 2019.
100. Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3:37, 2014.

101. Serenity Ethereum Foundation et al. *Ethereum 2.0 Specifications*. [online] https://github.com/ethereum/eth2.0-specs, 2018.
102. Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437*, 2017.
103. Seyoung Huh, Sangrae Cho, and Soohyung Kim. Managing iot devices using blockchain platform. In *19th ICACT*, pages 464–467. IEEE, 2017.
104. Matevž Pustišek and Andrej Kos. Approaches to front-end iot application development for the ethereum blockchain. *Procedia Computer Science*, 129:410–419, 2018.
105. Martin Valenta and Philipp Sandner. Comparison of ethereum, hyperledger fabric and corda. *no. June*, pages 1–8, 2017.
106. Li Qingqing, Jorge Peña Queralta, Tuan Nguyen Gia, and Tomi Westerlund. Offloading Monocular Visual Odometry with Edge Computing: Optimizing Image Compression Ratios in Multi-Robot Systems. In *The 5th International Conference on Systems, Control and Communications (ICSCC)*, 2019.
107. Xiaoyu Qiu, Luobin Liu, Wuhui Chen, Zicong Hong, and Zibin Zheng. Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. *IEEE Transactions on Vehicular Technology*, 68(8):8050–8062, 2019.
108. Kazim Rifat Özyilmaz, Mehmet Doğan, and Arda Yurdakul. Idmob: Iot data marketplace on blockchain. In *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018.
109. Vishnu Prasad Ranganthan, Ram Dantu, Aditya Paul, Paula Mears, and Kirill Morozov. A decentralized marketplace application on the ethereum blockchain. In *4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2018.
110. Blesson Varghese, Massimo Villari, Omer Rana, Philip James, Tejal Shah, Maria Fazio, and Rajiv Ranjan. Realizing edge marketplaces: challenges and opportunities. *IEEE Cloud Computing*, 5(6):9–20, 2018.
111. Yongjun Ren, Fujian Zhu, Jian Qi, Jin Wang, and Arun Kumar Sangaiah. Identity management and access control based on blockchain under edge computing for the industrial internet of things. *Applied Sciences*, 9(10):2058, 2019.
112. Zerynth Docs r2.5.2. Ethereum modules. Technical report, [Online]. Available: https://docs.zerynth.com/latest/official/lib.blockchain.ethereum/docs/index.html, 2020.
113. IOTA Distributed Ledger Technology software expansion for STM32Cube. X-cube-iota1. Technical report, [Online]. Available: https://www.st.com/en/embedded-software/x-cube-iota1.html, 2020.
114. Jorge Peña Queralta, Tuan Nguyen Gia, Hannu Tenhunen, and Tomi Westerlund. Comparative study of LPWAN technologies on unlicensed bands for M2M communication in the IoT: beyond LoRa and LoRaWAN. *Procedia Computer Science*, 2019.
115. V. K. Sarker, J. Peña Queralta, T. N. Gia, H. Tenhunen, T. Westerlund. A survey on lora for iot: Integrating edge computing. In *SLICE- FMEC*, 2019.
116. T. N. Gia, L. Qingqing, J. Peña Queralta, H. Tenhunen, T. Westerlund. Edge AI in Smart Farming IoT: CNNs at the Edge and Fog Computing with LoRa. In *IEEE AFRICON*, 2019.
117. T. N. Gia, J. Peña Queralta, T. Westerlund. Exploiting LoRa, Edge and Fog Computing for Traffic Monitoring in Smart Cities. In *Book Chapter: LPWAN Technologies for IoT and M2M Applications*. Elsevier, 2020.
118. Kazım Rıfat Özyılmaz and Arda Yurdakul. Work-in-progress: integrating low-power iot devices to a blockchain-based infrastructure. In *2017 International Conference on Embedded Software (EMSOFT)*, pages 1–2. IEEE, 2017.
119. Jun Lin, Zhiqi Shen, Chunyan Miao, and Siyuan Liu. Using blockchain to build trusted lorawan sharing server. *International Journal of Crowd Science*, 2017.
120. Arnaud Durand, Pascal Gremaud, and Jacques Pasquier. Resilient, crowd-sourced lpwan infrastructure using blockchain. In *CryBlock*, pages 25–29, 2018.