Open Infrastructure for Edge: A Distributed Ledger Outlook

Aleksandr Zavodovski
University of Helsinki

Nitinder Mohan
University of Helsinki

Walter Wong
University of Helsinki

Jussi Kangasharju
University of Helsinki

Abstract

High demand for low latency services and local data processing has given rise for edge computing. As opposed to cloud computing, in this new paradigm computational facilities are located close to the end-users and data producers, on the edge of the network, hence the name. The critical issue for the proliferation of edge computing is the availability of local computational resources. Major cloud providers are already addressing the problem by establishing facilities in the proximity of end-users. However, there is an alternative trend, namely, developing open infrastructure as a set of standards, technologies, and practices to enable any motivated parties to offer their computational capacity for the needs of edge computing. Open infrastructure can give an additional boost to this new promising paradigm and, moreover, help to avoid problems for which cloud computing has been long criticized for, such as vendor lock-in or privacy. In this paper, we discuss the challenges related to creating such an open infrastructure, in particular focusing on the applicability of distributed ledgers for contractual agreement and payment. Solving the challenge of contracting is central to realizing an open infrastructure for edge computing, and in this paper, we highlight the potential and shortcomings of distributed ledger technologies in the context of our use case.

1 Introduction

Originally, edge computing (EC) was introduced to provide a possibility for mobile devices to offload computation to nearby *cloudlets* [59], suggesting that computational capacity will not reside only in clouds but also at the edge of the network near the end-users, hence the name. High demand for low latency services and local data processing needed by Internet-of-Things (IoT) [34,54] has turned EC into a versatile new paradigm with diverse use cases and applications [62]. Naturally, since computational resources are now required in the proximity of end-users, the question arises where those facilities will come from, and who will operate them? So far, we

can identify two major deployment trends. One is an effort by big cloud providers to establish computational facilities near clients/at the "edge", e.g., Amazon CloudFront [31] or Microsoft Azure Stack [53]. The alternative is building an *Open Infrastructure for Edge* (OIE) suggested by [37,42,46,58,70]. Such an infrastructure will comprise of common practices, technologies, and set of open standards that will enable any interested parties to offer their computational capacity for purposes of EC. We will refer to such parties as *independent edge providers* (IEPs). Naturally, a major cloud provider can also be an IEP, as long as their service offering conforms to the OIE standards and practices. Establishing a viable OIE involves addressing a number of challenges, such as the discovery of IEPs by clients and application providers [71], common negotiation protocol for dynamic deployment of services to IEPs [70], migration of services to the IEP and so on. Assuming that most of the aforementioned technical problems can be solved, one final challenge remains, namely coming to an agreement between the IEP and the application provider. This may also include paying compensation to the IEP for the use of its resources. In volatile environments, where an edge-based application must react quickly to changes in users' behavior, conventional electronic payment methods used in e-commerce might not have enough agility. Moreover, both parties will likely require a low-cost escrow service with customizable rules since the application provider cannot be sure whether the IEP will be able to deliver its service as promised. Given the open nature of OIE, we see an open and decentralized mechanism for achieving consensus as a natural choice, and recent progress on distributed ledgers (DLs) makes them an appealing candidate solution.

In this paper, we conduct a case study of using DL for the needs and requirements of OIE. Given the variety of DLs and other means of digital contracting, we specifically identify the strengths and weaknesses of concrete systems for our use case. We also investigate emerging interledger solutions that propose to build an ecosystem utilizing multiple DLs.

The present situation is epitomized in [20]: "Blockchain has not yet bridged the gap from core technology to actual

applications, and at this point, remains theoretical rather than practical." Throughout the paper, the question we investigate is how to bridge this gap in the context of EC and *what would be an "ideal" DL for the EC?* As we will show later in the paper, the answer is not as straight-forward as assumed as none of the existing DL systems have all the desired properties to satisfy requirements of the OIE.

2 The Ricardian Triple

In this section, we describe two major digital contracting techniques, namely, *Smarts Contract* and *Ricardian Contract*. Together with parametrization, they form a *Ricardian triple* which we see as a promising evolution path towards building a DL framework that will satisfy most of the OIE needs.

Smart Contract. One trend pushing DLs beyond fiscal applications are smart contracts, introduced in [61]. Smart contracts are programs that are published on a peer-to-peer (P2P) blockchain network of miners. Users call methods of smart contracts by posting signed transactions to the network, and execution of such a method may alter the global state of the system, e.g., resulting in cryptocurrency transfer from one account to another. The execution is verified either collectively by miners or by special validator nodes, and only if the consensus is achieved changes to the state are globally propagated.

Ethereum's [65] implementation of smart contracts has enabled their wider adoption, also in the context of EC. Wright et al. [66] propose a system where Ethereum smart contact handles agreement between a group of edge nodes and users running applications at them. A key feature is escrow capability: a smart contract can freeze funds by transferring them to a special account and release only when specific conditions are met. Technically, when a user deploys an application to the edge node, the contract freezes payment of user and transfers it to the owner of the edge node only in the case that application was executed on the edge node appropriately. In [56], the authors present a framework where smart contracts handle tenants of the next generation cloud [48].

The Ricardian Contract. Conceived contemporaneously to smart contracts, the Ricardian Contract [38, 43] enables digitalizing the text of a legal agreement in tamper-proof fashion, wrapping it into machine-readable form, and providing the means to include the reference to a legal agreement in electronic transactions. Whereas smart contracts are responsible for the automated execution of an agreement that participants are already committed to, the Ricardian Contract represents the legal agreement itself put into digitally protected form.

Ricardian triple. The approach in [39, 44] unifies the two concepts above into a Ricardian triple: <*prose, parameters, code*>. The *prose* is the digital formulation of legal agreement, i.e., the Ricardian Contract; *code* is responsible for the enforcement of the *prose*, and corresponds to a smart con-

tract; *parameters* bind formulation and enforcement together, providing flexibility. In a Ricardian triple, *prose*, and *code* form a template-like structure while *parameters* concretize the particular details of the agreement.

The Ricardian triple is a promising candidate for building a decentralized agreement framework for EC since it unites declarative and procedural aspects of contracting. The Ricardian Contract encloses a digital SLA, while the smart contract is responsible for performance and enforcement of the SLA. Technically, the system underneath acts as a ledger, securing transactions and keeping the history of events. Parametrization allows for templating and negotiation: different participants may use similar agreements but negotiate custom price, latency, and other terms of service.

3 The Requirements of OIE

We now identify the key requirements for an open infrastructure for the edge. As edge is a very broad concept, we focus on a general set of requirements to be realized and base them largely on modern cloud services, as those are the yardstick against which edge computing performance is compared.

1) Speed of transaction confirmation. Effective use of EC requires ensuring service providers similar levels of scaling as in cloud datacenters, to meet varying demands. As edge nodes are typically limited in capacity, horizontal scaling is likely more common than vertical scaling. Further, we assume service providers will engage in both active scaling, i.e., adapting their resources on-the-fly, and passive scaling, i.e., pre-allocating resources based on estimated needs.

2) Cost of transaction. Due to relatively low prices for cloud VM deployments[1], the transaction cost of the DL must be no more than a negligible fraction of deployment cost itself.

3) Protection against financial volatility. A DL involves cryptocurrency or some form of digital tokens. The value of such tokens measured in conventional currencies is subject to market speculation. Figure 1a displays the volatility [36] of Ether (Ethereum token), Bitcoin and gold vs. U.S. dollar (USD). Lack of willingness to buy volatile tokens poses a threat to a wide-scale adaption of DL Technologies (DLT).

4) Privacy of transactions. Permissionless, i.e., publicly accessible ledgers, are mostly based on the blockchain, which is visible to all parties. While IEPs are likely willing to let their identities known, e.g., for reputation tracking, they might be hesitant to allow the details of the transactions, e.g., price, be publicly known. Ideally, we would like to be able to publish the existence of transactions, while keeping the details secret and revealed only when explicitly agreed by the IEP.

5) Accountability of participants. As an open environment, OIE does not preclude participation of crowdsourced IEPs. Crowdsourcing improves the availability of resources but

---

[1] AWS VM hourly ERM prices [30] may vary from $0.011 to $0.27.

poses a problem of unfair participants. The dilemma of anonymity vs. accountability [51] that most of the crowd-sourcing systems face must be solved in our case in favor of accountability.

6) Escrow service and conflict resolution. For the reasons above, we need an escrow service, transferring the payment only in the case of appropriate service delivery by the IEP. However, we need to handle cases when IEP and service owners disagree on the fulfillment of SLA. The problem is not specific for OIE and exists in modern clouds as well, but the possibility of crowdsourcing makes the issue more acute. Since formal verification of computation is too heavy for OIE needs, we see some form of reputation tracking as a solution. Thus, the DL must have the capability to store reputation rankings in a tamper-proof fashion.

7) Legislation for digital agreement. To be effective, Ricardian *prose* needs to have legislative status and variations in legal systems of different countries complicate the problem further. However, for digital contracting to be effective, we need real-world agreement on the legal frameworks (or restrict services and IEPs to operate inside a single jurisdiction). E.g., in Tennessee, there is a strong initiative by lawmakers to accept blockchain technology for legal contracts [35].

8) Truly distributed operation without a central authority. One of the goals of DL is to avoid centralized control and explicit ownership. However, there is evidence that well-established systems having strong community backing began to suffer from centralization [52]. Ideally, the DL should have an implicit mechanism preventing the centralization.

9) Environmentally sustainable operation. Figure 1b displays energy consumption [41] of Bitcoin and Ethereum; these are similar to Portugal and Costa Rica, respectively. Per transaction, Ethereum consumes 46 kWh, while VISA consumes 151 kWh for 100 000 transactions [41]. For sustainability, DLs used in OIE must consensus protocols other than those in proof-of-work (PoW) [55] family.

4 Challenges of the Building Blocks

Next, we examine existing distributed ledgers, their weaknesses and strengths from the perspective of the requirements formulated above. Our analysis of the DL systems is by no means exhaustive since there are over 2000 cryptocurrencies listed [40]; despite some of them being clones of each other (so-called altcoins), hundreds are still unique by the technology and application niche. We include systems that have managed to gain sufficient community support and maturity.

As we envision OIE to be an open infrastructure, we focus purely on permissionless, i.e., open DLs. Closed or permissioned DLs might in some cases offer different benefits and disadvantages, but their further analysis is left for future study.
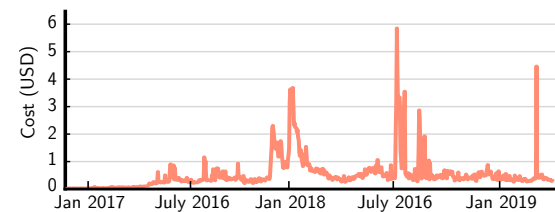
Support for the Ricardian Contract. The first implementation was the system called *Ricardo* [60], which is used by
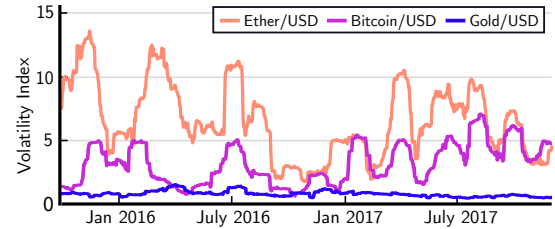
(a) Volatility of Ether, Bitcoin, and gold vs. US dollar [36].
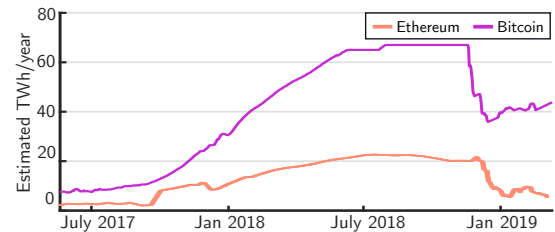
(b) Bitcoin and Ethereum energy consumption [41].

(c) Ethereum transaction fees [10].

Figure 1: Bitcoin and Ethereum infographics.

banks and other financial institutions. However, *Ricardo* is not intended for P2P execution. Among modern P2P ledgers, OpenBazaar [57] is notable for its use of the Ricardian Contract. The application of the contract is extended and broken into four phases: 1) vendor offer 2) buyer order 3) vendor order confirmation 4) buyer receipt. This workflow would also serve well purposes of OIE and it is easy to adopt. There is an initiative [33] to include built-in support of the Ricardian Contract into EOS [7, 45], one of the leading smart contract platforms. Technically, it is feasible to incorporate the Ricardian Contract into most of present DLs, especially those having smart contract support.

Consensus protocols and scalability. Bitcoin and Ethereum are not sustainable because of their energy-wasteful PoW [55] consensus protocol, where miners compete with each other. In Ethereum, the situation is further exacerbated by the need to validate execution of smart contracts, i.e., every full node executes all the calls to the smart contracts contained in the block before accepting a new block. Thus, a program is executed as many times as there are full nodes in the Ethereum network with a sole purpose of achieving the consensus.

Recent progress has produced entire families of new consensus protocols, of which proof-of-stake (PoS) family is of particular interest for us since many DL systems already use variants of PoS. Notably, Ethereum is planning the transition to Casper [69], which also belongs to the PoS family. The first

	Transactions per second (TPS)	Transaction confirmation time	Transaction fee	Number of nodes	Consensus algorithm
Ethereum	10-30	6 min	$0.35	8717	PoW, moving to PoS
EOS	3000	1.5 sec	$0	21 (528)	DPoS
Stellar	1000	4-5 sec	0.00001 XLM[†]	434	SCP
TRON	748	5 min	$0	1058	DPoS
Cardano	50-250	5 min	0.155381 ADA[‡]	NA	PoS (Ouroboros)

[†]XLM is Stellar Lumen token, [‡]ADA is Cardano token.

Table 1: Performance of platforms supporting smart contracts.

system implementing PoS was Peercoin [18], and basically, the idea behind the protocol is to entrust block generation to nodes having the most assets at stake, i.e., the more tokens node has the more trusted it is. Naturally, to prevent the richest node from usurping control over the system sophisticated randomization techniques are also applied. PoS is eco-friendly and improves the performance of a system, e.g., transactions per second (TPS) by order of magnitude. Delegated PoS (DPoS) [50] is used often in recent systems (see Table 1). In this protocol, a quorum of *block producers* is elected by PoS in rounds.

TrueBit [63] brings a solution for the scalability of smart contracts by introducing the *challenger* role. Expressed simply, challengers will selectively examine the execution of smart contracts and upon exposing incorrect result will receive a substantial reward. Sharding [68] is another approach to increase throughput: a system is split in a number of sidechains or shards, enabling parallel processing. Ethereum's Plasma [47] project is expected to combine sharding with techniques of TrueBit.

Given the state of the art in consensus protocols, we conclude that sustainability, scalability and performance issues are no insurmountable obstacles in broader DL adaption.

Smart contract systems. The leading platform in smart contracts has been Ethereum. Despite high reliability and security due to the large community backing, Ethereum has been criticized for: i) low transaction throughput ii) long confirmation time iii) high costs. Indeed, given metrics in Table 1, for the time being, Ethereum is not an appropriate platform for OIE, unless transition to Casper protocol changes the outlook.

Fortunately, the popularity of smart contracts has propelled a rapid development of the technology, spawning new systems that surpass successors by performance, usability, and other characteristics. Table 1 summarizes the most prominent of them in order of their capitalization [40].

Ethereum [65]. In practice, TPS stays around 15 transactions per second. Transaction fee varies significantly depending on the congestion of the network (see Figure 1c), the value in the table is average for January 2019.

EOS [7]. Use of DPoS protocol improves performance metrics compared to Ethereum. There are constantly 21 block producers in the system. The number of candidates for producer role may vary, and at the moment of measurement, there were 528 candidates [8]. There is only a block reward but no transaction fee, which is very attractive for end-users. There

was a remarkable criticism blaming the platform for security flaws, lack of accountability, and centralization [67].

TRON [25]. DPoS platform targeted for entertainment and content services. The platform promises zero transaction fees and theoretically maximum TPS of 748 which in practice stays near 20 [26].

Stellar [21]. Originally, the system shared many similarities with Ripple, but now it has evolved relatively far from it. The network is a set of servers (Stellar Cores) running Stellar Consensus Protocol (SCP), which is based on Federated Byzantine Agreement (FBA) instead of PoW or PoS. One might consider Stellar to be somewhere on the boundary of permissionless and permissioned DLs since anyone can set up a Stellar Core node but if no other node in the network is configured to trust this new node its validation opinions will have no weight. This can be seen as a downside; otherwise, the system's performance is high, costs low and operation sustainable.

Cardano [2]. As stated by its creators, the distinction of Cardano from other DPoS platforms is that the Ouroboros protocol is the first one of DPoS family with mathematically proven properties. The TPS of the next version (Hydra) is expected to be around 5000 transactions per second, due to the usage of sharding. The transaction costs are dependent on the size of transactions and a couple of parameters; the value given in Table 1 is the minimal possible value [24].

Privacy. Cryptocurrencies such as ZCash [29] or Monero [15] use publicly available encrypted ledger. Encrypted transactions remain validatable due to the usage of zero-knowledge proofs and other advanced techniques. Such a feature is appealing for OIE since many IEPs would prefer to hide their bookkeeping from competitors while preserving accountability. Unfortunately, at the moment neither ZCash nor Monero support smart contracts. Dero project [5] aims to bring smart contracts to CryptoNote [4] protocol (used by Monero). The problem is the utilization of PoW and lack of acceptance from the community (at present, Dero's rank is 374 [40]). Technically, the privacy issue is solvable, e.g., AZTEC protocol [1] is designed for the execution inside smart contracts and provides functionality similar to CryptoNote. However, present privacy solutions are clearly lacking convenience.

Protection against volatility. Exceptionally high volatility, i.e., unpredictable value fluctuations of cryptoassets has been one of the major disincentives for broader take up of DLs. The problem has opened a niche for so-called *stablecoins* [32].

Stablecoin is cryptocurrency which has a fixed exchange rate with some fiat currency or another asset. The most prominent stablecoins by their capitalization are Tether [23], USD Coin [28], and TrueUSD [27]. Technically, these cryptocurrencies run either as ERC20 [9] tokens on Ethereum or inside Omni Layer [17] of Bitcoin (or its altcoins), thus sharing all the shortcomings of the underlying network. Stable exchange rate is guaranteed by the acquisition of real assets, so stablecoins require some formal institutionalized entity or private company to actually obtain and hold those assets. In this sense, stablecoins contradict decentralization philosophy implicit for Bitcoin and cryptotokens in general. Essentially, stablecoins replace the risk of volatility with the risks of trusting a foundation, company, or other non-governmental financial institution. There were allegations that there is no hard evidence that dollars backing Tether actually exist [64]. To summarize, at present there is no mechanism to protect cryptocurrencies against volatility that would avoid centralization and risks related to it.

Interledgers. The concept behind interledger technology is to enable seamless interaction between different DLs via a commonly defined interface or intermediary in such way, that the end-user can utilize DL as a service, without any specific platform lock-in [49]. Interledgers also partially address the problem of volatility and trust, since in the case client and an IEP prefer different DL platforms, the contracting is still possible via an interledger. In the scope of this article, we look at Interledger Protocol (ILP) [14], Cosmos [3], and Polkadot [19]. ILP developers emphasize using a methodology similar to networking protocols, resulting in a layered outcome with clearly defined interfaces and responsibilities. From our point of view, the downside of this solution is the relatively tight binding of its implementations to Ripple (e.g., Moneyd [16]), thus orienting more towards interbank transactions than community-based blockchain systems. Cosmos and Polkadot on the other hand, target blockchain developers. They are both taking advantage of Tendermint [22] consensus protocol and share many other similarities on both conceptual and technical sides. Cosmos offers Tendermint coupled with a gossip networking engine as a building block for the creation of new custom blockchain platforms that will communicate with each other easily through the Cosmos Hub. For those systems that do not use Tendermint as a consensus protocol (e.g., Bitcoin or Ethereum), interaction mechanism is more complicated and involves the creation of so-called *Peg Zones* (*Bridges* in Polkadot) with special validator nodes. The arrangement is complex and participants have to trust the operator of the Peg Zone. The need to trust the operator violates the principle of decentralization. On a positive note, according to Cosmos developers [3], Ethereum's transition to Casper protocol will obsolete the need for Peg Zone since Casper belongs to the same PoS family as Tendermint. So far, the vision of putting major DLs behind a single interface is still a distant mirage. Nevertheless, in the case systems like Cosmos or Polkadot will become prevalent, a Tendermint DL will be an attractive option for the OIE.

5 Summary

In this paper, we examined the potential for edge computing to move towards OIE and formulated the requirements that such an extensive infrastructure would demand from the point of the contractual agreement and transaction handling. While there are other challenges in realizing an open edge infrastructure, such as discovery of resources and their management, those are more of a technical nature and we expect most of them to be efficiently solvable. Contractual issues in a large-scale, open infrastructure, on the other hand, require a decentralized solution for managing the agreements and consensus. Distributed ledgers seems like an attractive choice and in this paper we focused on evaluating the suitability of existing DLs for OIE. We ended up with a slightly negative result since, at the time of writing, no existing system fully satisfies the requirements that OIE would impose. This is mainly due to the operation of present highly distributed DL systems supporting smart contracts, e.g., Ethereum, not being environmentally sustainable, while TPS is too low and costs are inappropriately high.

Fortunately, the outlook for the future is promising due to the rapid technological development in the field of DLT. As of now, there are fully operational systems[2] using smart contracts or alike technology to trade computational resources. On a broader scale, there is a trend toward the transition from energy wasteful PoW family of protocols towards, e.g., ecological PoS and other solutions alike. However, having an operational, robust and reliable public DL is not only a question of the technical solution and its implementation. Such DLs are operated by a community, comprising of miners either pursuing an economic profit or volunteering for a non-monetary cause. Fortunately, such a community already exists: as of the moment, there are ≈ 8717 Ethereum nodes operated by miners [11]. Moreover, the emergence of the new communities backing novel systems is also highly probable. Upcoming interledgers are obsoleting the need for a dominant DL solution and, hopefully, will enable a seamless interaction between various ledgers, diversifying the ecosystem. Thus, despite present hindrances we believe that in the near future there will be community-backed open systems capable of undertaking the challenges of OIE, necessitating further research into the field.

Acknowledgments

---

[2]Golem [12], iExec [13], and DFINITY [6], just to mention a few.

6 Workshop Discussion Topics

a) What kind of feedback are we looking to receive?

We are most interested in getting feedback on the general stance of the paper, i.e., our starting point that an open edge infrastructure is a viable and likely deployment direction. Much of the future of edge computing will also depend on actual applications and any kind of feedback from people actually running edge applications would be appreciated. Our paper is based on the assumption that open infrastructures, like the Internet, will eventually triumph over closed, proprietary systems.

b) The controversial points of the paper

Distributed ledgers are very much a controversial topic and their suitability for OIE is very much open to debate. The performance of DLs is a moving target and making predictions about their future suitability for edge computing is hard.

c) The type of discussion this paper is likely to generate in a workshop format

We see the discussion revolving around the direction of development for edge, i.e., will it be the monopoly of a small number of providers or will the open edge infrastructure we envision in this paper take over. The technical issues that need to be solved in these two cases are very different and for the community it would be extremely valuable to have discussions on the tradeoffs inherent in either of these directions, as well as their advantages and disadvantages.

d) The open issues the paper does not address

We do not address the use of permissioned, i.e., closed, DLs. These could provide an interesting alternative to open, permissionless DLs, but would bring their own tradeoffs. Also, because of space limitations, we had to leave out several existing DLs.

e) Under what circumstances the whole idea might fall apart

We can see a couple of possibilities where the need for a distributed, open edge infrastructure will not emerge, or where there would be no need to use distributed ledgers. One, edge computing could go the way of cloud computing or content distribution, which are monopolized by a few large players, obviating the need for flexible contracting. Two, distributed ledgers might turn out to be a dead-end from a business point of view and turn out to be unworkable. Three, commercial entities, like VISA, could take over smart contracting via their current infrastructures.

References

[1] AZTEC protocol, 2019. https://www.aztecprotocol.com/.

[2] Cardano, 2019. https://www.cardano.org.

[3] Cosmos Network, 2019. https://cosmos.network/.

[4] CryptoNote - the next generation cryptocurrency, 2019. https://cryptonote.org/.

[5] DERO project, 2019. https://dero.io/.

[6] DFINITY, 2019. https://dfinity.org/.

[7] EOS, 2019. https://eos.io/.

[8] EOS Tracker, 2019. https://eostracker.io/producers.

[9] ERC20 Token Standard, 2019. https://theethereum.wiki/w/index.php/ERC20_Token_Standard/.

[10] Ethereum Charts and Statistics, 2019. https://etherscan.io/charts.

[11] Ethereum Nodes Explorer, 2019. https://www.ethernodes.org.

[12] Golem, 2019. https://golem.network/.

[13] iExec, 2019. https://iex.ec/.

[14] Interledger Protocol, 2019. https://interledger.org/.

[15] Monero, 2019. https://www.getmonero.org/.

[16] Moneyd, 2019. https://github.com/interledgerjs/moneyd.

[17] Omni Layer, 2019. https://www.omnilayer.org/.

[18] Peercoin, 2019. https://peercoin.net/.

[19] Polkadot, 2019. https://polkadot.network/.

[20] Polkadot Lightpaper, 2019. https://polkadot.network/Polkadot-lightpaper.pdf.

[21] Stellar, 2019. https://www.stellar.org/.

[22] Tendermint Core, 2019. https://tendermint.com/.

[23] Tether, 2019. https://tether.to.

[24] Transaction Fees in Cardano Settlement Layer, 2019. https://cardanodocs.com/cardano/transaction-fees/.

[25] TRON, 2019. https://tron.network/.

[26] TRON BlockChain Explorer, 2019. https://tronscan.org.

[27] True USD, 2019. https://www.trusttoken.com/trueusd/.

[28] USD Coin, 2019. https://www.centre.io/usdc.

[29] Zcash: Privacy-protecting digital currency, 2019. https://z.cash/.

[30] Amazon. Amazon EMR Pricing, 2018. https://aws.amazon.com/emr/pricing/.

[31] Amazon. CloudFront, 2019. https://aws.amazon.com/cloudfront/.

[32] Bilal Memon. Guide to Stablecoins, 2019. https://masterthecrypto.com/guide-to-stablecoin-types-of-stablecoins/.

[33] Block.one. Block.one's Proposal for EOS Constitution v2.0, 2018. https://block.one/news/block-ones-proposal-for-eos-constitution-v2-0/.

[34] Flavio Bonomi et al. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[35] Jordan Buie. Lawmaker's bill would accept technology behind Bitcoin for contracts in Tennessee, 2018. Tennessean, https://eu.tennessean.com/story/news/2018/02/06/bitcoin-law-tennessee-cryptocurrency-bill/311641002/.

[36] Buy Bitcoin Worldwide. The Bitcoin Volatility Index, 2019. https://www.buybitcoinworldwide.com/volatility-index/.

[37] Ronan-Alexandre Cherrueau, Adrien Lèbre, Dimitri Pertin, Fetahi Wuhib, and João Soares. Edge computing resource management system: a critical building block! initiating the debate via openstack. In *The USENIX Workshop on Hot Topics in Edge Computing (HotEdge'18)*, 2018.

[38] Usman W. Chohan. What Is a Ricardian Contract? *SSRN*, 2017.

[39] Christopher D Clack, Vikram A Bakshi, and Lee Braine. Smart contract templates: foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*, 2016.

[40] CoinMarketCap, 2019. https://coinmarketcap.com/.

[41] Digiconomist. Ethereum Energy Consumption Index, 2019. https://digiconomist.net/ethereum-energy-consumption.

[42] I. Farris, T. Taleb, H. Flinck, and A. Iera. Providing ultra-short latency to user-centric 5g applications at the mobile network edge. *Transactions on Emerging Telecommunications Technologies*, 29(4):e3169, 2018.

[43] I. Grigg. The ricardian contract. In *Proceedings. First IEEE International Workshop on Electronic Contracting, 2004.*, pages 25–31, July 2004.

[44] I. Grigg. The sum of all chains — let's converge!, 2015. http://financialcryptography.com/mt/archives/001556.html.

[45] Ian Grigg. EOS - An Introduction, 2017. http://www.iang.org/papers/EOS_An_Introduction.pdf.

[46] David Haja, Marton Szabo, Mark Szalay, Adam Nagy, Andras Kern, Laszlo Toka, and Balazs Sonkoly. How to Orchestrate a Distributed OpenStack. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops*, pages 293–298. IEEE, 2018.

[47] J. Poon, V. Buterin. Plasma, 2019. https://plasma.io/.

[48] Wolfgang John, Joacim Halén, Xuejun Cai, Chunyan Fu, Torgny Holmberg, Vladimir Katardjiev, Tomas Mecklin, Mina Sedaghat, Pontus Sköldström, Daniel Turull, et al. Making cloud easy: design considerations and first components of a distributed operating system for cloud. In *Proceedings of the 10th USENIX Conference on Hot Topics in Cloud Computing*, pages 12–12. USENIX Association, 2018.

[49] Arto Karila, Yki Kortesniemi, Dmitrij Lagutin, Pekka Nikander, Santeri Paavolainen, Nikos Fotiou, George C. Polyzos, Vasilios A. Siris, and Theodore Zahariadis. SOFIE. Network and Distributed System Security Symposium 2018, page 6, 2018.

[50] Daniel Larimer. Delegated Proof-of-Stake (DPoS). 2014.

[51] Yuan Lu, Qiang Tang, and Guiling Wang. ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 853–865. IEEE, 2018.

[52] Dahlia Malkhi. Blockchain in the Lens of BFT, 2018. USENIX ATC'18 Keynote, https://www.usenix.org/conference/atc18/presentation/malkhi.

[53] Microsoft. Azure Stack, 2019. https://azure.microsoft.com/en-us/overview/azure-stack/.

[54] N. Mohan and J. Kangasharju. Edge-Fog cloud: A distributed cloud for Internet of Things computations. In *2016 Cloudification of the Internet of Things (CIoT)*, pages 1–6, Nov 2016.

[55] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[56] Sambit Nayak, Nanjangud C Narendra, Anshu Shukla, and James Kempf. Saranyu: Using smart contracts and blockchain for cloud tenant management. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 857–861. IEEE, 2018.

[57] OpenBazaar. Decentralized Reputation in OpenBazaar, 2019. https://openbazaar.org/blog/decentralized-reputation-in-openbazaar/.

[58] M. Satyanarayanan, R. Schuster, M. Ebling, G. Fettweis, H. Flinck, K. Joshi, and K. Sabnani. An open ecosystem for mobile-cloud convergence. *IEEE Communications Magazine*, 53(3):63–70, March 2015.

[59] Mahadev Satyanarayanan et al. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 2009.

[60] Systemics. Ricardo by Systemics, 2019. http://www.systemics.com/docs/ricardo/.

[61] Nick Szabo. Smart Contracts: Building Blocks for Digital Markets, 1996.

[62] 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):1657–1681, 2017.

[63] Jason Teutsch and Christian Reitwießner. A scalable verification solution for blockchains, 2019. https://truebit.io/.

[64] Paul Vigna and Steven Russolillo. The Mystery Behind Tether, the Crypto World's Digital Dollar. *The Wall Street Journal*, 2018. https://www.wsj.com/articles/the-mystery-behind-tether-the-crypto-worlds-digital-dollar-1534089601.

[65] Gavin Wood. Ethereum: A Secure Decentralised Generalised Transaction Ledger. *Ethereum Project Yellow Paper*, 2014.

[66] Kwame-Lante Wright, Martin Martinez, Uday Chadha, and Bhaskar Krishnamachari. SmartEdge: A Smart Contract for Edge Computing, 2018.

[67] Brent Xu, Dhruv Luthra, Zak Cole, and Nate Blakely. EOS: An Architectural, Performance, and Economic Analysis, 2018. Whiteblock, https://www.whiteblock.io/library/eos-test-report.pdf.

[68] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 931–948. ACM, 2018.

[69] Vlad Zamfir, Nate Rush, Aditya Asgaonkar, and Georgios Piliouras. Introducing the 'Minimal CBC Casper' Family of Consensus Protocols. *Ethereum Research*, 2018. https://github.com/cbc-casper/cbc-casper-paper/blob/master/cbc-casper-paper-draft.pdf.

[70] Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. ICON: Intelligent Container Overlays. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, HotNets '18, pages 15–21, New York, NY, USA, 2018. ACM.

[71] Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. ExEC: Elastic Extensible Edge Cloud. In *Proceedings of the 2Nd International Workshop on Edge Systems, Analytics and Networking*, EdgeSys '19, pages 24–29, New York, NY, USA, 2019. ACM.