
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2021 SCPE. Volume 22, Issues 4, pp. 463–468, DOI 10.12694/scpe.v22i4.1947

A REVIEW OF BLOCKCHAIN-ENABLED FOG COMPUTING IN THE CLOUD CONTINUUM CONTEXT

ADRIAN SPĂTARU*

Abstract. This article surveys the literature in search of systems and components that use Blockchain or Smart Contracts to manage computational resources, store data, and execute services using the Cloud paradigm. This paradigm has extended from warehouse-scale data centres to the edge of the network and in between, giving rise to the domains of Edge and Fog Computing. The Cloud Continuum encompasses the three fields and focuses on the management of applications composed of connected services that span from one end to the other of the computational spectrum. Several components that are commanded by Smart Contracts are identified and compared concerning their functionality. Two important research directions are the experimental evaluation of the identified platforms and the identification of standards that can accelerate the adoption of Blockchain-based Fog platforms.

Key words: Blockchain, Decentralized Cloud, Service Orchestration

AMS subject classifications. 68M14,68M15

1. Introduction. New challenges concerning data transfer have been introduced in the scope of the Internet of Things (IoT) advancements. Previous paradigms requiring all data to be moved in the Cloud for processing have been replaced by the Edge and Fog Computing paradigms. Edge Computing tries to push the computation on the end-devices, while Fog Computing tackles the processing of data at an intermediate level, between the data originator and the Cloud. This reduces network congestion, but constructing proprietary Fog networks implies a high cost and both in terms of investment and maintenance.

The emergence of Blockchain technologies (Bitcoin [20], Ethereum [4]) has proved that if economic incentives are high, then the crowd will invest in the hardware required for running the technology. In the case of blockchain mining, the resources are mostly wasted and the probability to mine the next block is decreasing as more nodes join the network. Instead, some of the high-end personal computers and small private Clouds are excellent candidates for exposing their resources to the Fog fabric. In this manner, the nodes will be reimbursed for executing applications, storing data, or monitoring other nodes.

There has been a surge in the number of platforms developed to take advantage of security, transparency, and traceability offered by the Blockchain. Nevertheless, these properties can only be offered for data stored on the Blockchain. Generally, there is a need for further proofs related to objects managed externally, proofs which can be checked using Smart Contracts. FileCoin [2] builds upon the IPFS peer-to-peer file system, using Ethereum Smart Contracts for handling access management, payments, and reward allocation. A Proof of Replication [11] algorithm has been proposed to validate the amount of space a node has dedicated for storing file blocks under different replication standards.

The biggest downside of peer to peer systems constructed using personal computers is the unpredictability of the nodes' availability. The Blockchain use case does not suffer from this because nodes do not participate actively in synchronization procedures. Rather, when a block is mined, the block is sent to other peers which will validate it and append it to the history of blocks. If a peer is not online at the moment, the peer will ask for all new blocks when coming back online. In the case of file systems and cloud services, there is a need to ensure that nodes that hold files or execute applications will be available for a given period, or at least have backup plans in case they become absent. This should also be ensured for the components that manage the platform, as they are also external to the Blockchain and need to maintain a state in a distributed environment.

*Department of Computer Science, West University of Timişoara, Romania. (adrian.spataru@e-uvt.ro).

Table 2.1: Current research directions

Research direction	**Approach**	**Research Items**
File Storage, Content Distribution Networks	Some use Smart Contracts to check an externally generated proof of storage. Others use Smart Contracts to retais a mapping of data blocks to locations and to manage access control. Data is stored on nodes running Ethereum and IPFS.	[2, 11], [7, 18, 26]
Resource Management / Task scheduling	Smart Contract maintains list of resources. Some rely on the Smart Contract to make the matchmaking, but this approach is expensive in terms of gas. Instead, the parties can negotiate off-chain and the Smart Contract can *authorize* their agreement.	[13, 27, 32, 33, 34]
Service Orchestration	Several Smart Contracts are used to manage the resources across multiple clusters, and to instantiate applications composed of one or multiple services which may depend on each other. Several Orchestration components assure the fault tolerance of the deployment and monitoring processes.	[28, 29]
Monitoring and Quality of Service	One approach is to use Oracles (trusted parties) to monitor the services and assess their quality. Instead, a peer to peer network with a robust protocol can ensure the correctness of the monitoring process.	[14, 29, 31]
Result Verification	Credibility-based fault tolerance protocols and zero-knowledge proofs can be used to determine if a node has executed the program as expected. These strategies are encoded in Smart Contracts which verify the proofs. A different approach is to use Trusted Execution Environments which cryptographically ensure that the program is executed as planned.	[8, 15, 23]

The subject of peer to peer networks has been investigated thoroughly with respect to distributed consensus [21, 16, 25, 5, 6], file sharing protocols like Kademlia [17], BitTorrent [22], IPFS [3], and protocols for volunteer computing (BOINC [1], XtremWeb[10]) and achieving correct results in the presence of saboteurs [23].

This report surveys the literature in search of platforms or components that use Smart Contracts for the management of computational resources and deployment of applications. Two surveys complement our work. The survey presented in [30] inspects three Blockchain-based Cloud platforms. It provides an abstraction based on the three architectures and presents standards that can be used to ease the integration between Cloud Components. An in-depth investigation is pursued in [9], which focuses on Blockchain protocols with built-in logic for specific delivery paradigms (IaaS, PaaS, SaaS, and more).

The contributions of this paper are the identification of Fog platforms and components that can be used in the context of Cloud Continuum. Several components are identified and categorized based on their functionality, and components within the same category are compared. Finally, future research directions required for the integration of the Cloud, Fog, and Edge devices are presented.

2. Algorithms and Components. Several research papers tackle the decentralization of the Cloud by proposing solutions for specific Cloud processes such as resource management, data storage, or service orchestration, quality of service and result verification. The directions are summarized in Table 2.1.

Resource management and task scheduling have been investigated in [27]. The authors investigate the operational constraints and costs for managing resources and applications through a Smart Contract and analyze the impact in terms of latency and gas usage of three scheduling methods. The authors conclude that the best approach is to negotiate the resource selection off-chain and use the Smart Contract to seal the agreement. The other option is for the Smart Contract to handle the matchmaking algorithm, which proves to be expensive both in terms of gas and in terms of time until the transaction is mined. A different paper, [13] approached the problem via the Serverless computing paradigm, but by using Hyperledger Fabric the team did not focus on the economic cost of deploying the solution to a public blockchain. have been proposed in [34, 33, 32], but do not provide the same level of fine-grained evaluation analysis.

Several platforms making use of the Blockchain to verify data storage and transfers have been proposed,

making use of known protocols like BitTorrent or IPFS. FileCoin [2] and others [18, 7, 26], use IPFS as the backbone for file storage and transfer and apply diverse logic applicable for different purposes. FileCoin makes use of a Proof-of-Replication mechanism [11] to decide that a node has spent some space for a given amount of time. The solutions presented in [18] and [26] propose a secure document communication protocols based on IPFS that uses the blockchain for establishing communication channel properties. In [7], the authors distinguish between hot and cold data and apply different replication mechanisms depending on their type. The blockchain is used for data access management and to keep track of the nodes holding the data. We further refer the reader to [12], for a comparison between centralized and blockchain-based decentralized storage solutions.

Service Orchestration intermediated by the blockchain has been investigated in a few papers. In [29], a Component Administration Network stores Orchestrator checkpoints which ensure the continuity of Application Deployment in the presence of Orchestrator failures. A decentralized, modular platform has been presented in [28] and investigated in detail. The aforementioned work has focused on the management and fault tolerance of the Fog platform, which can later ensure the fault tolerance of the running applications. A series of Smart Contracts are used to manage resource selection and service deployment. A Registry Contract maintains a catalogue of resource managers (local Clouds, ad-hoc clusters) that a customer can query for specific resources. The resource manager can create an Application Contract which is used by an Orchestration component to deploy and monitor the status of the services composing the application. Orchestration components periodically update the Application Contract regarding the status of the services, thus ensuring a fair payment. Some nodes part of the Blockchain peer-to-peer network will form the administration network instead of mining. These nodes check on each other's availability and on any components that are running (e.g. Orchestrator). Components are run by a subset of the administration nodes and store checkpoints on this network using a distributed file system. All nodes that contribute to the running and monitoring of an application are reimbursed for their contribution using the checkpoints (checkpoint metadata is stored in the application Smart Contract).

Several works have tackled the Quality of Service (QoS) aspect of the deployed applications. Some of the solutions (e.g. [14]) depend on external oracles which are trusted parties for monitoring the QoS. Alternately, the monitoring job can be delivered by the crowd. The concept of "Crowd-based Oracle-as-a-service For Consensus On Qos Monitoring" presented in [31] is similar to the Component Administration Networks presented in [29], making use of a Smart Contract to manage a network of trusted peers. The advantages of the platform presented in [31] and [14] are represented by the usage Service Level Agreements (SLA) and Service Level Objectives (SLO), while the solution presented in [29] only accounts for the service being responsive for a given amount of time.

In the case of batch tasks, several papers investigate the correctness of the result. Sarmenta's approach to deal with saboteur nodes using credibility-based fault tolerance [23] is currently the base of the Proof of Computation protocol (PoCo) at the heart of the iExec platform (described further below). Another approach is to use zero-knowledge proofs to verify the results of computation by providing a small proof to a Smart Contract [15]. The introduction of the Intel SGX technology [8] paved the way for Trusted Execution Environments to become popular. A hardware key is used for ensuring no corruptions have been made to the code, while memory is encrypted to prevent a root user of the machine to access the computation data.

3. Platforms. Several platforms that offer service deployment through the means of Blockchain exist. Ethereum itself is taught as the *world computer*, though the capabilities of storing data and execution are drastically limited by the price of smart contract operations. Thus, platforms rely on using the Ethereum Blockchain in order to create tokens for their platform, and raise investment funds through *Initial Coin Offerings* (ICOs).

Golem[1] uses IPFS [3] as the means to distribute file blocks in a network of worker nodes which process data at the block level and later collect and merge the results computed in parallel. The platform intended to offer Software as a Service, but starting in 2020 the focus was shifted to Platform as a Service, providing an SDK for creating Golem applications.

SONM[2] uses Docker for executing Container Images and achieves a higher level of abstraction, getting close to a generic Cloud platform. An Ethereum side chain is used to manage the orders. *Suppliers* must

[1]https://golem.network/
[2]https://docs.sonm.com/concepts/main-entities

Table 3.1: Blockchain based Cloud Platforms

Name	Blockchain	Market	Storage	Services
Golem	Ethereum	No	No	PaaS
SONM	Ethereum	Yes	Yes	IaaS
iExec	Ethereum	Yes	Yes	PaaS
Decenter	Ethereum	Yes	External	IaaS

Table 3.2: iExec entities

iExec Hub and Marketplace	an auditable smart contract used to manage the stakes and keep track of the history of the actors.
Dataset providers	individuals which will sell access to their data on the platform.
Application providers	individuals which will deploy applications on the platform; applications can be free or ask for a price.
Workers	individuals or companies which expose their resources on the marketplace.
Worker pools	smart contracts to which workers can subscribe; the smart contract will take care of workers contributions and will receive fees for managing the underlying infrastructure. This contract stores scheduler settings, required in order to handle the stake and payments.

interact with this chain to instantiate worker nodes that will act on their behalf. Resources (CPU, RAM, storage, bandwidth) are exposed using benchmark identifiers such as GFLOPS, IOPS, etc. The platform considers two delivery models: fixed-time or pay-as-you-go. There are no peer-reviewed experiments presenting the performance of the platform, and the limited amount of documentation does not present implementation details.

iExec makes use of Xtremweb middleware [10, 19] to handle task placement and the validity of results. The platform logic is implemented using a side chain and the public Ethereum Blockchain is used to create and handle the platform tokens. The platform defines several entities that are shown in Table 3.2.

A *Proof of Contribution* (*PoCo*) protocol is used for acknowledging the correct result of an Application, using the sabotage tolerance introduced in [23]. The *PoCo* links two entities: the iExec marketplace (where deals are made) and the computing infrastructure (based on XtremWeb-HEP middleware [10]).

DECENTER is a Horizon 2020 financed project aiming at providing a federated brokering platform for fog resources [24]. Their proposed architecture is centred around the Resource Exchange Broker (REB) Smart Contract. Resource providers deploy an REB Contract which manages the selection of resources and signals Orchestration Components to deploy applications. A resource provider is required to have installed a full Cloud management software stack: infrastructure management and provisioning, together with service orchestration components.

Recent publications present an architecture that allows for the definition of Service Level Agreements (SLAs) using Smart Contracts [14]. Quality of Service (QoS) parameters such as network throughput, or the latency between different tiers of the same Application are then used by a decision-making layer, which is composed of the monitoring and orchestration components.

4. Conclusion. This report has investigated the current efforts in the direction of Blockchain-based platforms offering SaaS, PaaS, IaaS. The most difficult problem is the verification of work, generally implying one of two options: work replication, or workload monitoring. The first option is employed by iExec and uses the results computed by multiple replicas to decide on the validity of a result. This, however, implies a batch model for the Application in order to decide and this is a requirement that cannot be satisfied by today's Cloud Services, which are generally interactive.

Golem is using monitoring, but the monitoring metrics are self-reported by the node participating in the network. This raises security concerns as the metrics can be fabricated by the node. SONM and DECENTER choose to dedicate some nodes for running the monitoring agents that inspect the worker nodes and applications

running on top of them, yet they consider these to be trusted by default. Instead, research efforts are pushing the boundaries by focusing on fault-tolerant mechanisms for ensuring the quality of work execution and ensuring the fair payment of all entities taking part in managing the platform.

There is still a limited amount of experimental research that compares the performance and scalability of Blockchain-based systems and components that expose computational resources. This is mostly due to the infancy of the domain, researchers focusing on publishing architectures and protocols as the first materials. New simulation platforms need to be implemented to tackle the vast distribution of future services and their interconnectivity.

An important future direction for Blockchain-mediated Fog service delivery should focus on new standards for the deployment and monitoring of applications spanning the Cloud Continuum. Future applications will be represented by workflows that start at the Edge of the network (processing raw data obtained from sensors), flow through the Fog (aggregating data from multiple Edge devices) and reach the Cloud (for archival, analytics, visualisation). Adaptation of existing standards can increase the chances for a blockchain-based Fog network to be integrated with the Cloud Continuum.

Acknowledgment. This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project number PN-III-P4-ID-PCE-2020-0407, within PNCDI III.

REFERENCES

[1] David P Anderson. Boinc: A system for public-resource computing and storage. In *proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10. IEEE Computer Society, 2004.
[2] J Benet and N Greco. Filecoin: A decentralized storage network. *Protoc. Labs*, pages 1–36, 2018.
[3] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.
[4] Vitalik Buterin et al. Ethereum white paper, 2014. *URL https://github. com/ ethereum/wiki/wiki/White-Paper*, 2013.
[5] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.
[6] Bernadette Charron-Bost, Fernando Pedone, and André Schiper. Replication. *LNCS*, 5959:19–40, 2010.
[7] Yongle Chen, Hui Li, Kejiao Li, and Jiyang Zhang. An improved p2p file system scheme based on ipfs and blockchain. In *Big Data (Big Data), 2017 IEEE International Conference on*, pages 2652–2657. IEEE, 2017.
[8] Victor Costan and Srinivas Devadas. Intel sgx explained. *IACR Cryptol. ePrint Arch.*, 2016(86):1–118, 2016.
[9] MR Dorsala, VN Sastry, and S Chapram. Blockchain-based solutions for cloud computing: A survey. *Journal of Network and Computer*, 2021. Query date: 2021-12-03 19:28:36.
[10] Gilles Fedak, Cecile Germain, Vincent Neri, and Franck Cappello. Xtremweb: A generic global computing system. In *Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on*, pages 582–587. IEEE, 2001.
[11] Ben Fisch, Joseph Bonneau, Nicola Greco, and Juan Benet. Scaling proof-of-replication for filecoin mining. *Benet//Technical report, Stanford University*, 2018.
[12] T. Gabriel, A. Cornel-Cristian, M. Arhip-Calin, and A. Zamfirescu. Cloud storage. a comparison between centralized solutions versus decentralized cloud storage solutions using blockchain technology. In *2019 54th International Universities Power Engineering Conference (UPEC)*, pages 1–5, 2019.
[13] Sara Ghaemi, Hamzeh Khazaei, and Petr Musilek. Chainfaas: An open blockchain-based serverless platform. *IEEE Access*, 8:131760–131778, 2020.
[14] Petar Kochovski, Vlado Stankovski, Sandi Gec, Francescomaria Faticanti, Marco Savi, Domenico Siracusa, and Seungwoo Kum. Smart contracts for service-level agreements in edge-to-cloud computing. *Journal of Grid Computing*, pages 1–18, 2020.
[15] Benjamin Körbel, Marten Sigwart, Philip Frauenthaler, Michael Sober, and Stefan Schulte. Blockchain-based result verification for computation offloading. In Hakim Hacid, Odej Kao, Massimo Mecella, Naouel Moha, and Hye-young Paik, editors, *Service-Oriented Computing*, pages 99–115, Cham, 2021. Springer International Publishing.
[16] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.
[17] Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.
[18] Mruthyunjaya Mendu, B Krishna, Sallauddin Mohmmad, Y Sharvani, and Ch Vinay Kumar Reddy. Secure deployment of decentralized cloud in blockchain environment using inter-planetary file system. In *IOP Conference Series: Materials Science and Engineering*, volume 981, page 022037. IOP Publishing, 2020.
[19] Mircea Moca, Cristian Litan, Gheorghe Cosmin Silaghi, and Gilles Fedak. Multi-criteria and satisfaction oriented scheduling for hybrid distributed computing infrastructures. *Future Generation Computer Systems*, 55:428–443, 2016.
[20] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *URL https:// bitcoin.com/bitcoin.pdf*, 2008.

[21] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.

[22] Johan Pouwelse, Paweł Garbacki, Dick Epema, and Henk Sips. The bittorrent p2p file-sharing system: Measurements and analysis. In *International Workshop on Peer-to-Peer Systems*, pages 205–216. Springer, 2005.

[23] Luis FG Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems*, 18(4):561–572, 2002.

[24] Marco Savi, Daniele Santoro, Katarzyna Di Meo, Daniele Pizzolli, Miguel Pincheira, Raffaele Giaffreda, Silvio Cretti, Seung-woo Kum, and Domenico Siracusa. A blockchain-based brokerage platform for fog computing resource federation. In *2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 147–149. IEEE, 2020.

[25] Fred B Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys (CSUR)*, 22(4):299–319, 1990.

[26] Meet Shah, Mohammedhasan Shaikh, Vishwajeet Mishra, and Grinal Tuscano. Decentralized cloud storage using blockchain. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 384–389. IEEE, 2020.

[27] Adrian Spataru, Laura Ricci, Dana Petcu, and Barbara Guidi. Decentralized cloud scheduling via smart contracts. operational constraints and costs. In *The International Symposium on Blockchain Computing and Applications (BCCA2019)*, 2019.

[28] Adrian Spătaru. *Decentralized Cloud Computing*. PhD thesis, 2021.

[29] Adrian Spătaru. Decentralized and fault tolerant cloud service orchestration. *Scalable Computing: Practice and Experience*, 21(4):709–725, 2020.

[30] Rafael Brundo Uriarte and Rocco De Nicola. Blockchain-based decentralized cloud/fog solutions: Challenges, opportunities, and standards. *IEEE Communications Standards Magazine*, 2(3):22–28, 2018.

[31] Rafael Brundo Uriarte, Huan Zhou, Kyriakos Kritikos, Zeshun Shi, Zhiming Zhao, and Rocco De Nicola. Distributed service-level agreement management with smart contracts and blockchain. *Concurrency and Computation: Practice and Experience*, 33(14):e5800, 2021.

[32] Zehui Xiong, Shaohan Feng, Wenbo Wang, Dusit Niyato, Ping Wang, and Zhu Han. Cloud/fog computing resource management and pricing for blockchain networks. *IEEE Internet of Things Journal*, 6(3):4585–4600, 2018.

[33] Chenhan Xu, Kun Wang, and Mingyi Guo. Intelligent resource management in blockchain-based cloud datacenters. *IEEE Cloud Computing*, 4(6):50–59, 2017.

[34] H. Zhu, Y. Wang, X. Hei, W. Ji, and L. Zhang. A blockchain-based decentralized cloud resource scheduling architecture. In *2018 International Conference on Networking and Network Applications (NaNA)*, pages 324–329, 2018.