Taxonomy of Edge Blockchain Network Designs

Nguyen Khoi Tran[1][0000−0002−9538−7476] and M. Ali Babar[1][0000−0001−9696−3626]

The University of Adelaide, South Australia, Australia
{nguyen.tran,ali.babar}adelaide.edu.au

Abstract. Blockchains have been increasingly employed in use cases at the network's edge, such as autonomous vehicles and edge computing. These use cases usually establish new blockchain networks due to operation costs, performance constraints, and the lack of reliable connectivity to public blockchains. The design of these edge blockchain networks heavily influences the quality attributes of blockchain-oriented software deployed upon them. This paper presents a taxonomy of edge blockchain network designs successfully utilized by the existing literature and analyzes their availability when facing failures at nodes and networks. This taxonomy benefits practitioners and researchers by offering a design guide for establishing blockchain networks for edge use cases.

Keywords: Blockchain · Distributed Ledger · Design Pattern.

1 Introduction

Blockchain is a type of distributed system that enables tamper-proof transaction recording and facilitates incorruptible judgement without relying on a third party [7]. To establish a new blockchain, practitioners must design and develop an underlying blockchain network – *a distributed software system comprising connected computers that jointly maintain a blockchain according to a blockchain protocol*. Even though the term "blockchain networks" suggests that these systems operate at the computer network level, blockchain networks actually denote the distributed software system running on top of a computer network. Thus, designing blockchain networks is a software architecture topic, similarly to designing and configuring a distributed system.

Designing a blockchain network involves deciding on its protocol and topology. Despite the critical impact of blockchain networks on the characteristics of blockchain-oriented software, the design aspect of blockchain networks has been relatively under-investigated in the software architecture literature comparing to blockchain-oriented software topics such as patterns [9], tactics [6], and decision models [1]. Even less research has focused on edge blockchain networks [2]. While practitioners can selectively apply these existing work to edge blockchain networks, they would benefit further from a comprehensive description, classification, and analysis of concrete designs utilized successfully by the existing edge blockchain network prototypes.

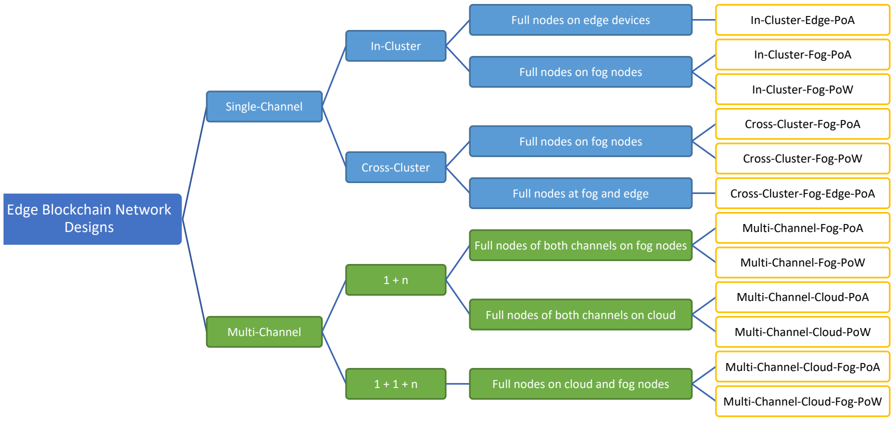


Fig. 1. Taxonomy of edge blockchain network designs

This paper targets the stated gap with a taxonomy of edge blockchain network designs. We constructed this taxonomy by systematically identifying, classifying, and evaluating edge blockchain network designs utilized by prototypes in the literature. First, we identified the relevant papers from a paper dataset of an existing systematic literature review on integrating blockchain and the Internet of Things [5]. Second, we extracted blockchain network design decisions from the prototypes reported by these papers using the blockchain network design space [4]. Third, these decisions were grouped into network designs and organized into a taxonomy (Figure 1). Finally, we evaluated the identified designs regarding their availability for edge devices to read, write, and propagate updates when losing nodes, networks, and cloud backend. We focused on availability because of its criticality to edge use cases and its independence from the implementation technology and hardware infrastructure. The taxonomy of edge blockchain network designs and the availability analysis forms the contributions of this paper.

2 Background and Related Work

2.1 Blockchain Network Design

Blockchains can be considered a singleton, transaction-based state machine replicated across a group of networked computers – *a blockchain network* [7]. These computers operate according to *consensus protocol*, which dictates how a blockchain network agree on the validity and order of blockchain transactions to ensure that a blockchain's state is identical across the entire network. Public blockchains, which are open to anonymous participants on the Internet, generally employ *Proof-of-Work (PoW) consensus protocols*. These protocols are resistant against Sybil attacks, which involve a malicious party creating multiple fake identities to take over the consensus of an open and anonymous system.

PoW protocols make Sybil attack prohibitively expensive by adding a cost to any consensus-related interaction, such as broadcasting a new set of transactions (i.e., a *block*). This cost assumes a cryptography puzzle that is challenging to solve but easy for anyone to validate. All participants that process transactions in a PoW-based blockchain network, also known as *miners*, race to solve puzzles and broadcast new blocks. Due to the independence of miners, PoW-based networks are resilient to network fragmentation because all fragments would continue operating instead of becoming hanged. An alternative to PoW consensus protocol is *Proof-of-Authority (PoA) consensus protocols*, which require participants to reveal their identity and coordinate closely to reach a consensus on the validity and order of transactions. Due to this tight coordination, PoA-based networks are less resource-intensive but more vulnerable to network disruptions.

Designing a blockchain network involves decisions covering both protocol and topology [4]. Protocol decisions include choosing and configuring the consensus protocol for a blockchain network. Topological decisions determine a blockchain network's structure. They cover the distribution of blockchain nodes and mining rights to blockchain participants. They describe the deployment of blockchain nodes on hardware infrastructure. They also cover the segmentation of a blockchain network into *channels*, which maintain separated ledgers and might use different blockchain protocols.

2.2 Related Work

The architecture design of software running on or interacting with blockchains have been studied extensively in the recent software engineering literature. For instance, Xu et al. [8] proposed considering blockchain as a new form of software connector and presented a pattern collection for blockchain-based applications [9]. Wessling et al. [6] proposed architectural blockchain tactics. Comuzzi [1] proposed a design space consisting of eight types of design decisions for consideration when transforming a centralised software system into a blockchain-based one. While these patterns, tactics, and design space are invaluable to blockchain-based software, they do not guide the design of blockchain networks beyond describing the typical characteristics of various blockchain types [10].

A few research have focused on the architectural aspect of the underlying blockchain network. For example, Tran et al. [4] proposed a design space for blockchain networks containing 19 dimensions with 51 choices. This design space applies to blockchain networks in most domains, including edge use cases. It provides a framework for systematically evaluating design decisions and describe blockchain network designs. More closely related to edge blockchain networks are four architectural styles proposed by Liao et al. [2], which specify where blockchain clients should be located in an IoT-based infrastructure. While practitioners can selectively apply these existing work to the design of edge blockchains, none has provided a comprehensive description, classification, and analysis of concrete designs for edge blockchain networks that support the design process of practitioners working on blockchain for edge use cases.

This paper addresses the stated gap by proposing a taxonomy of edge blockchain network designs. We apply the blockchain network design space [4] to identify and describe the common edge blockchain network designs employed by the existing prototypes. Comparing to the architectural styles proposed by Liao et al. [2], the taxonomy has a broader scope, covering both protocol and topology design decisions rather than limiting to blockchain client deployment. Moreover, we also evaluate the availability level of the identified edge blockchain network designs against five failure scenarios.

3 Taxonomy of Edge Blockchain Network Designs

3.1 Methodology

We developed the taxonomy following a three-step process. First, we identified the relevant papers about operating blockchains in an edge computing environment. These papers were selected from a dataset of an existing systematic literature review on Blockchain-integrated Internet of Things systems [5], combined with the top search results from Google Scholar on the keywords "blockchain" and "edge computing".

Second, we extracted design decisions from the selected papers using the blockchain network design space [4]. The extracted decisions fell into two classes. *Protocol decisions* cover the choice and configuration of blockchain protocols for operating the blockchain network. *Topological decisions* cover the following aspects:

- *Overall network's shape decisions* determine whether the network is segmented into interrelated channels and the coverage of those channels.
- *Full node decisions* dictate the distribution of full nodes to the network's participants and the deployment of full nodes on hardware infrastructure.
- *Mining nodes decisions* control the distribution of the mining rights and the placement of mining nodes on the hardware infrastructure.
- *Lightweight node decisions* determine whether a blockchain network employs lightweight blockchain nodes in its topology.
- *Network interface decisions* govern the type and deployment of blockchain networks' interfaces.
- *Remote node decisions* govern the distribution and deployment of remote nodes, which are programs that users use to query and send transactions to a blockchain.

In this step, we also synthesized a *Reference Edge Infrastructure* from the edge infrastructures utilized by the selected papers (Fig. 2). The reference edge infrastructure is arranged into clusters, representing devices deployed at different locations. Each cluster includes edge devices and a few fog nodes, which act as cluster heads and are responsible for communicating with other clusters and cloud servers. No direct communication is allowed between edge nodes and any device outside its cluster. The reference edge infrastructure normalizes deployment-related design decisions across different papers by defining a

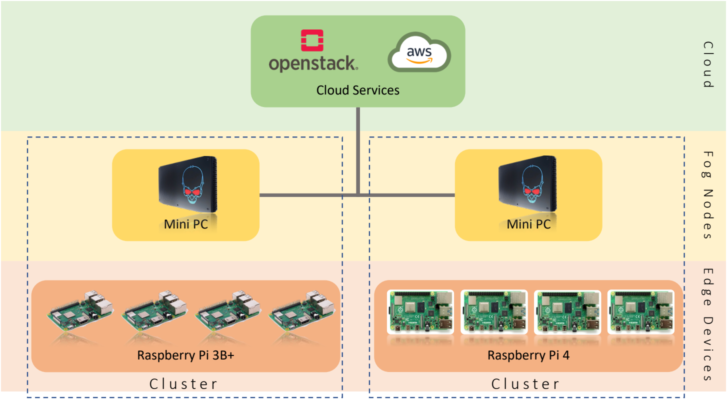


Fig. 2. Reference edge infrastructure for describing deployment decisions

standard vocabulary for describing hardware nodes and networks on which edge blockchain networks are deployed. It also helped minimize the ambiguity of concepts such as edge, fog nodes, cloudlet, and edge computing.

The last step of our process was constructing the taxonomy of edge blockchain network designs. We identified common edge blockchain network designs, which are groups of design decisions that commonly appear together. We clustered these designs into a hierarchical structure, forming the taxonomy (Fig. 1).

3.2 Taxonomy of Edge Blockchain Network Designs

Despite the diversity of options, we found that the existing edge blockchain prototypes converged to a few designs, classified into two groups: single-channel designs and multi-channel designs.

Single-channel designs contain only one blockchain channel in the entire edge infrastructure or many independent channels that have no connection with each other, effectively forming smaller separate blockchain networks. Single-channel designs are divided into in-cluster and cross-cluster designs. *In-cluster designs* limit blockchain channels within the boundary of device clusters. Designers can choose between placing full and mining nodes on edge devices or offloading all blockchain-related responsibilities to fog nodes. Placing miners on edge devices limits the consensus protocol choice to PoA due to resource constraints. *Cross-cluster designs* deploy a blockchain channel across multiple device clusters. A popular design is placing all blockchain-related responsibilities on fog nodes that control clusters, reducing edge devices to simple blockchain clients. Alternatively, designers can place full and mining nodes on all fog nodes and edge devices. As we will see in the availability analysis, this design offers a superior availability level for reading from a blockchain. However, it limits protocol choice to PoA-based consensus protocols.

Multi-channel designs contain multiple blockchain channels covering different sets of hardware nodes in an edge computing infrastructure. We further classify these designs into $1 + n$ and $1 + 1 + n$ designs. The $1 + n$ designs combine n in-cluster channels with a cross-cluster channel. Both types of channels can be designed using the corresponding single-channel designs described above. The cross-cluster channel is generally hosted by cloud servers or fog nodes.

The $1 + 1 + n$ designs contain a cloud-level channel, a cross-cluster fog-level channel, and n in-cluster channels. For instance, Nyamtiga et al. [3] employed a three-level edge blockchain network to securely store and manage the data collected by edge devices. The bottom-level channels covered edge devices within clusters. The middle-level channel spanned fog nodes, connecting clusters. The top-level channel was deployed within cloud infrastructures to control and coordinate cloud resources. Even though $1 + 1 + n$ designs offer a high degree of flexibility, they were among the least popular edge blockchain network designs.

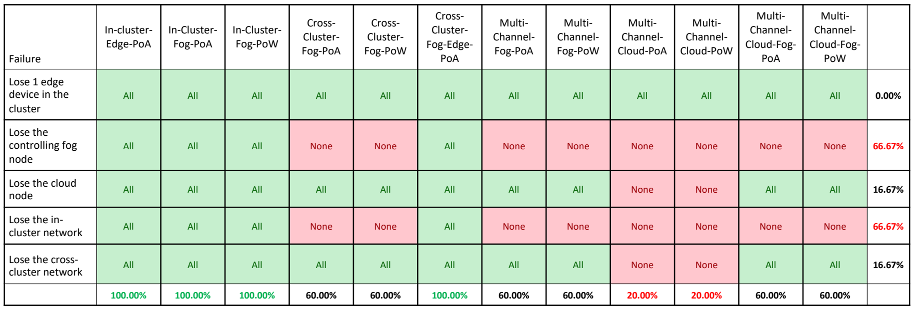
4 Availability Evaluation

We evaluated the availability level of edge blockchain network designs from the perspective of edge devices such as robots and autonomous vehicles because they are primary end-users of edge blockchain networks. We distinguished three types of availability as follows. *Read-availability* of a blockchain network design assesses whether edge devices can read the content stored on a blockchain. *Write-availability* assesses whether updates sent by edge devices can be committed to a blockchain, meaning that a device can reach a blockchain node and that node can process and append new transactions to a blockchain. *Update-availability* assesses whether the updates sent by an edge device can reach every device, only local devices, or no devices in an edge infrastructure. It should be noted that being write-available does not imply being update-available. For instance, a disconnected group of PoW-based blockchain clients is available for writing, but these updates would not reach all edge devices.

We analyzed the availability of edge blockchain network designs listed in the taxonomy in the context of the following failure scenarios: (1) losing (n/3) of edge devices in a local cluster, (2) losing a local fog node, (3) losing a cloud node, (4) losing a local edge network, and (5) losing a cross-cluster network.

As an example, let us consider the availability of the Cross-Cluster-Fog-Edge-PoA design when losing the cross-cluster network. All edge devices can still read from a blockchain when a failure happens because they host full blockchain nodes and thus have access to a complete ledger replica. However, the PoA consensus protocol used by a Cross-Cluster-For-Edge-PoA blockchain network would hang if its miners are disconnected. Therefore, none of the edge devices can write to a blockchain, which means none of the edge devices can receive any update from others. Thus, the Cross-Cluster-Fog-Edge-PoA design achieves the ”all” level for read availability and ”none” for both write availability and update availability. Following the a similar analysis, we evaluated all blockchain network designs identified in the taxonomy. Figure 3 and 4 presents the results.

a. Read Availability



Failure	In-cluster-Edge-PoA	In-Cluster-Fog-PoA	In-Cluster-Fog-PoW	Cross-Cluster-Fog-PoA	Cross-Cluster-Fog-PoW	Cross-Cluster-Fog-Edge-PoA	Multi-Channel-Fog-PoA	Multi-Channel-Fog-PoW	Multi-Channel-Cloud-PoA	Multi-Channel-Cloud-PoW	Multi-Channel-Cloud-Fog-PoA	Multi-Channel-Cloud-Fog-PoW	
Lose 1 edge device in the cluster	All	All	All	All	All	All	All	All	All	All	All	All	0.00%
Lose the controlling fog node	All	All	All	None	None	All	None	None	None	None	None	None	66.67%
Lose the cloud node	All	All	All	All	All	All	All	All	None	None	All	All	16.67%
Lose the in-cluster network	All	All	All	None	None	All	None	None	None	None	None	None	66.67%
Lose the cross-cluster network	All	All	All	All	All	All	All	All	None	None	All	All	16.67%
	100.00%	100.00%	100.00%	60.00%	60.00%	100.00%	60.00%	60.00%	20.00%	20.00%	60.00%	60.00%	

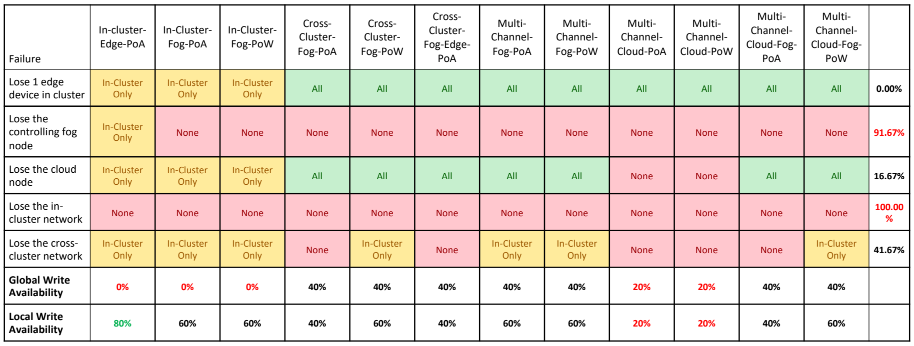
b. Write Availability

Failure	In-cluster-Edge-PoA	In-Cluster-Fog-PoA	In-Cluster-Fog-PoW	Cross-Cluster-Fog-PoA	Cross-Cluster-Fog-PoW	Cross-Cluster-Fog-Edge-PoA	Multi-Channel-Fog-PoA	Multi-Channel-Fog-PoW	Multi-Channel-Cloud-PoA	Multi-Channel-Cloud-PoW	Multi-Channel-Cloud-Fog-PoA	Multi-Channel-Cloud-Fog-PoW	
Lose 1 edge device in cluster	All	All	All	All	All	All	All	All	All	All	All	All	0.00%
Lose the controlling fog node	All	None	None	None	None	None	None	None	None	None	None	None	91.67%
Lose the cloud node	All	All	All	All	All	All	All	All	None	None	All	All	16.67%
Lose the in-cluster network	None	None	None	None	None	None	None	None	None	None	None	None	100.00%
Lose the cross-cluster network	All	All	All	None	All	None	All	All	None	None	None	All	41.67%
	80.00%	60.00%	60.00%	40.00%	60.00%	40.00%	60.00%	60.00%	20.00%	20.00%	40.00%	60.00%	

Fig. 3. Read and write availability levels of the evaluated blockchain network designs

5 Concluding Remarks

This paper presents and evaluates the availability levels of twelve edge blockchain network designs commonly used by the existing edge blockchain prototypes. We found that the further we push blockchain clients to the edge, the more available an edge blockchain network becomes, particularly in read availability. Designs that place blockchain clients directly on edge computing devices enjoy the best read availability and in-cluster update availability. Cloud-only designs consistently offer the worst availability levels. Protocol-wise, we find that PoW-based designs generally provide better levels of write availability and update availability. While designs that use PoW consensus protocol and push blockchain clients to the edge offer higher availability levels, they are also significantly more resource-intensive and, as a result, potentially slower. These drawbacks make these designs infeasible for some edge use cases. This observation shows the trade-off between resource consumption, performance, and the availability of blockchain network designs. Thus, future research should empirically analyze the performance and resource consumption offered by edge blockchain network designs to establish a multi-dimensional decision support system for designing blockchain networks that consider trade-offs on all three aspects.



Failure	In-cluster-Edge-PoA	In-Cluster-Fog-PoA	In-Cluster-Fog-PoW	Cross-Cluster-Fog-PoA	Cross-Cluster-Fog-PoW	Cross-Cluster-Fog-Edge-PoA	Multi-Channel-Fog-PoA	Multi-Channel-Fog-PoW	Multi-Channel-Cloud-PoA	Multi-Channel-Cloud-PoW	Multi-Channel-Cloud-Fog-PoA	Multi-Channel-Cloud-Fog-PoW	
Lose 1 edge device in cluster	In-Cluster Only	In-Cluster Only	In-Cluster Only	All	All	All	All	All	All	All	All	All	**0.00%**
Lose the controlling fog node	In-Cluster Only	None	None	None	None	None	None	None	None	None	None	None	**91.67%**
Lose the cloud node	In-Cluster Only	In-Cluster Only	In-Cluster Only	All	All	All	All	All	None	None	All	All	**16.67%**
Lose the in-cluster network	None	None	None	None	None	None	None	None	None	None	None	None	**100.00 %**
Lose the cross-cluster network	In-Cluster Only	In-Cluster Only	In-Cluster Only	None	In-Cluster Only	None	In-Cluster Only	In-Cluster Only	None	None	None	In-Cluster Only	**41.67%**
Global Write Availability	**0%**	**0%**	**0%**	**40%**	**40%**	**40%**	**40%**	**40%**	**20%**	**20%**	**40%**	**40%**	
Local Write Availability	**80%**	**60%**	**60%**	**40%**	**60%**	**40%**	**60%**	**60%**	**20%**	**20%**	**40%**	**60%**	

Fig. 4. Update availability levels of the evaluated blockchain network designs

References

1. Comuzzi, M., Unurjargal, E., Lim, C., Workshops, A.I.S.E.: Towards a Design Space for Blockchain-Based System Reengineering. pp. 138–143. Springer International Publishing, Cham (2018)
2. Liao, C.F., Hung, C.C., Chen, K.: Blockchain and the Internet of Things: A Software Architecture Perspective. In: Business Transformation through Blockchain, pp. 53–75. Springer (2019)
3. Nyamtiga, B.W., Sicato, J.C.S., Rathore, S., Sung, Y., Park, J.H.: Blockchain-based secure storage management with edge computing for IoT. Electronics (Switzerland) **8**(8), 1–22 (2019). https://doi.org/10.3390/electronics8080828
4. Tran, N.K., Ali Babar, M.: Anatomy, concept, and design space of blockchain networks. In: Proceedings - IEEE 17th International Conference on Software Architecture, ICSA 2020 (2020). https://doi.org/10.1109/ICSA47634.2020.00020
5. Tran, N.K., Ali Babar, M., Boan, J.: Integrating blockchain and Internet of Things systems: A systematic review on objectives and designs. Journal of Network and Computer Applications **173**(September 2020), 102844 (2020). https://doi.org/10.1016/j.jnca.2020.102844
6. Wessling, F., Ehmke, C., Meyer, O., Gruhn, V.: Towards blockchain tactics: Building hybrid decentralized software architectures. In: 2019 IEEE International Conference on Software Architecture Companion (ICSA-C). pp. 234–237. IEEE (2019)
7. Wood, G.: Ethereum: A Secure Decentralized Generalized Transaction Ledger. Ethereum Yellow Paper (2014)
8. Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A.B., Chen, S.: The Blockchain as a Software Connector. In: 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA). pp. 182–191 (2016). https://doi.org/10.1109/WICSA.2016.21
9. Xu, X., Pautasso, C., Zhu, L., Lu, Q., Weber, I.: A Pattern Collection for Blockchain-based Applications (2018). https://doi.org/10.1145/3282308.3282312
10. Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., Rimba, P.: A taxonomy of blockchain-based systems for architecture design. In: 2017 IEEE International Conference on Software Architecture (ICSA). pp. 243–252. IEEE (2017)