Applying Blockchain consensus mechanisms to Network Service Federation: Analysis and performance evaluation

Kiril Antevski[a,*], Carlos J. Bernardos[a]

[a] Universidad Carlos III de Madrid (UC3M), Madrid, Spain

ARTICLE INFO

Keywords:
network service federation
Blockchain
network domains
consensus mechanisms

ABSTRACT

Cutting edge (vertical) applications may swiftly saturate service providers' virtualized infrastructure. In 5G and beyond, service providers seek out for innovative solutions such as Network Service Federation (NSF) which allows orchestration of external domain services/resources to provide a zero-downtime end-to-end vertical service. Distributed ledger technologies, such as Blockchain, are used to enhance the federation process. In this article, we propose and evaluate the application of Blockchain technology for NSF. We evaluate four different consensus mechanisms: Proof-of-Work (PoW), Proof-of-Authority (PoA), Practical Byzantine-Fault Tolerant (PBFT), and Proof-of-Stake (PoS). The experimental evaluation is executed using Ethereum, Tendermint and Cosmos platforms. Results show that the evaluated consensus mechanisms enable the use of NSF for both latency-sensitive and security stringent vertical applications.

1. Introduction

The new generation of networks lay foundations on the benefits of virtualization. Technologies such as Network Function Virtualization (NFV), Software Defined Networks (SDN), and Multi-access Edge Computing (MEC) enable domains to offer new range of network services. They foster better use of the underlying network infrastructure. The use of general purpose hardware boosts the adaptation to unexpected spikes in service usage, however an over-provisioning is needed. The deployment and maintenance of redundant network resources is an expensive solution for service providers.

The latest expansion of Ultra-Reliable Low Latency Communication (URLLC) and Enhanced Mobile Broadband (eMMB) services incline providers to seek for balanced and profitable solutions. Network Service Federation (NSF) is a solution which enables operators to consume network services and resources from other external administrative domains. In other cases, if a telco provider has available and free resources or services, it can lease them as a provider to an external administrative domain, thus lowering the OpEx.

Both the consumer and provider domains have mutual benefit in implementing the federation feature. Telco operators and service providers are aware of the potential gain, however there is a reasonable scepticism in terms of the complexity, security and applicability of the solutions. Namely, federation is usually used for services that require geo-specific service footprint (e.g., sport events such as the Olympics which demand various services on high user density location), sudden increase of users or urgent need of scaling out. The priority for NSF is having a brief execution time and establishment of a new E2E service spanning across multiple administrative domains. The execution itself requires cautious negotiation and interaction among domains. A decentralized domain-to-domain (or peer-to-peer) negotiation and execution is secure, but inefficient and slow approach. A centralized approach is fast and secure even though it has a pricey central entity which is a single-point of failure.

A distributed solution has the potential to have a balanced trade-off between the decentralized and centralized solution. Distributed Ledger Technologies (DLTs) such as Blockchain seem promising in realization of a secure, failure-resistant platform for both negotiation and execution of NSF.

In previous works Antevski, Girletti, Bernardos, de la Oliva, Baranda and Mangues-Bafalluy (2021); J. Baranda et al. (2020a); Antevski, Groshev, Baldoni and Bernardos (2020), we have tackled the federation problem with a goal to explore different approaches of realization of NSF. Different works uri; Boubendir, Guillemin, Le Toquin, Alberi-Morel, Faucheux, Kerboeuf, Lafragette and Orlandi (2018) have tackled a similar problem using distinct mechanisms and blockchain platforms. Various Blockchain platforms use diverse consensus mechanisms to generate and append new blocks. The generation of a new block directly influences the negotiation and execution time of a NSF. Many works Saraf and Sabadra (2018); Gao, Hatcher and Yu (2018); Dabbagh, Choo, Beheshti, Tahir and Safa (2021); Nguyen, Pathirana, Ding and Seneviratne (2020) compare the general characteristics of consensus mechanisms which prove to have a high variance on the block generation times and the performance depends on the application.

*This work has been partially funded by European Union's Horizon 2020 research and innovation programme under grant agreement No 101015956, and the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-NextGenerationEU through the UNICO 5G I+D 6G-EDGEDT and 6G-DATADRIVEN. It has also been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M21), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

*Corresponding author
**Principal corresponding author
kantevsk@pa.uc3m.es (K. Antevski)
ORCID(s): 0000-0001-9057-2691 (K. Antevski); 0000-0003-0708-4983 (C.J. Bernardos)

The goal of this work is to define a common federation scenario and explore how different consensus mechanisms on different Blockchain platforms affect the negotiation and execution of NSF. On top of that, we explore the performance of the Blockchain hosts in terms of CPU, memory, disk and network activity. To the best of our knowledge, it is the first work that executes a performance analysis for network federation scenario using different Blockchain platforms. Additionally, we add remarks based on the experimental realization process.

The rest of the paper is organized as follows. In Section 2 we position our work with the state-of-the-art. In Section 3, we describe the Network Service Federation (NSF) concept, and how Blockchain can be applied. In Section 4, we describe the consensus mechanisms which performance we later evaluate in Section 5 through a defined scenario. The measured results are discussed in Section 6. Finally, we conclude the work in Section 7 along with future works.

2. Related Work

In this section we are positioning this article in the state of the art. The Network Service Federation (NSF) has been explored into different works for NFV Management and Orchestration (MANO) environments Carlos J. Bernardos et al. (2016); Francescon, Baggio, Fedrizzi, Ferrusy, Yahiaz and Riggio (2017) and Edge environments GSM; D32; Antevski et al. (2020). The authors in Baranda, Mangues-Bafalluy, Vettori, Martínez, Antevski, Girletti, Bernardos, Tomakh, Kucherenko, Landi, Brenes, Li, Costa-Pérez, Ubaldi, Imbarlina and Gharbaoui (2020); J. Baranda et al. (2020a,b) have proposed and realized a peer-to-peer network federation between different administrative domains running a NFV MANO orchestration platform over virtualized infrastructure. These works assume that domains have already exchanged information or established relationship through Service-Level Agreements (SLAs). In our case, administrative domains are oblivious of each other existence and the process of discovery and agreement is handled by the Blockchain itself.

The ETSI Permissioned Distributed Ledgers (PDL) group have released several group reports that provide guidelines into applying distributed ledger technology (such as Blockchain) for networking ETSI (2020). Our work extends the guidelines, especially into the example scenario of Section 7.5 in ETSI (2021a) where smart contracts are used for Quality of Service (QoS) monitoring. In our work, we extend the scenario by healing a failed federated service using the same federation procedure. Recently, ETSI PDL released a stable draft ETSI (2021b) explaining the reference architecture for integrating applications and services in a DLT-based platform. Our work contributes into the realization of platform services in every platform category - *alpha* - single underlying DLT technology; *bravo* - multiple underlying DLT technologies using Abstraction layer; *charlie* - Application abstraction layer and DLT abstraction layer to have unified north-bound interface, and *delta* - Application abstraction layer with a single DLT technology. Thus for a DLT-based platform implementation, service providers may decide the platform category based on the supported underlying DLT type.

Other works Bamakan, Motavali and Bondarti (2020); Strobel, Castelló Ferrer and Dorigo (2020); Azbeg, Ouchetto, Andaloussi and Fetjah (2021); De Angelis (2018); Gao et al. (2018); Dabbagh et al. (2021) evaluate the performance of different consensus algorithms. Our article extends these works by evaluating and comparing consensus algorithms in a network service federation experimental scenario.

3. Federation scenario and the use of Blockchain

In this section we present the reader with an in-depth description of the Network Service Federation (NSF) process. In an Network Function Virtualization Management and Orchestration (NFV MANO) environment, the NSF is defined as a feature that allows an administrative domain to orchestrate services or resources across different heterogeneous and external administrative domains. Through the NSF, service providers are able to satisfy the requirements of vertical customers, and additionally may lower the operational cost through offering idle infrastructural resources or services to other external service providers.

The dual actions of consuming and providing NSF indicate that each administrative domain can have a double role: consumer and provider. A consumer domain requests to consume services or additional resources in order to provide an end-to-end (E2E) network service to its vertical customer. A provider domain is an external administrative domain which has the capacity to provide a service or infrastructure resources to a consumer domain.

In the following, we describe the federation process executed and different approaches for various environments. Later, we address how Blockchain can improve the federation process.

3.1. Federation steps

There are several steps that domains follow in order to complete a successful NSF:

1. **Domain registration.**
   Administrative domains may interact among themselves only if they have knowledge of each other point of contact. In a peer-to-peer interaction, it is established through business meetings. In centralized or decentralized platform solution, every domain needs to be registered. Upon registration domains may interact with each other. Security mechanisms and integrity checks are fundamental for both centralized and decentralized solutions.
2. (a) **Advertisement/Discovery.** The second step depends on the nature of the interaction between the administrative domains - the federation environment. When domains establish peer-to-peer connections or register to a centralized platform,

a contract is signed between the parties. The nature of the relationship is already set and it is more static. Hence, when a (consumer) domain desires to federate and consume a service, it issues a federation advertisement. The advertisement is sent to each peer-to-peer domain individually or in a centralized platform it is broadcast to all participants. For a large number of connections, administrative domains may use a polling method or *discovery* instead of advertising. In this way, every domain can create a global view of the available services for federation.

(b) **Announcement/Negotiation.** As opposed to stable environments, in dynamic environments or decentralized platforms, administrative domains are not aware of other participating domains. The relationships are dynamically broken or established. Domains need to re-negotiate federation terms repeatedly. Therefore domains use announcements to reveal the intent of consuming a federated service. Interested provider domains engage into negotiation. Different negotiation techniques can be used, such as: *bilateral*, *match-matching* or *autonomous* V. Scoca et al. (2017). Match-matching and autonomous are more suitable to be used in a centralized environment. Federation begins with a consumer requesting federation of a service with a range of terms. The central entity matches potential provider domains that strictly match terms - *match-matching*, or by close-to-full fulfillment - *autonomous*. Upon a matched domain, both provider and consumer domains receive the connectivity details. The *bilateral* case is more suitable for a decentralized interconnection scheme. The consumer domain broadcasts an *announcement* request for federation. Interested provider domains interplay in a reverse-auction fashion by replying with bidding offers. The consumer domain forms a final decision using internal policy criteria. The selected winning provider domain proceeds into fulfillment of the federation request.

3. **Deployment.** As a next step, the "winning" provider domain obtains the deployment information and proceeds with the deployment of a federated service. The provider domain proceeds notifies the consumer domain of the deployment outcome. The consumer domain and the provider domain establish data plane connectivity and inclusion of the federated service at the end of a successful deployment. In an NFV MANO environment, these steps are repeated for every service extending/scaling/healing operation J. Baranda et al. (2020b). The whole process is transparent for the end-user or vertical, the federated service continues running as an integral part of the consumer domain without any interruption of service.

4. **Life-cycle management & charging.** The consumer domain is the one who manages the federated service life-cycle. For NFV MANO environments the management is executed on a single-layered or hierarchical. In a single layer management communication, the consumer domain orchestrator uses the provider domain orchestrator as a proxy. In a hierarchical management scheme, the control and management plane goes through the north/southbound interfaces (e.g., consumer orchestrator to provider virtualization infrastructure manager) J. Baranda et al. (2020b); Baranda et al. (2020). Both domains monitor the usage and performance of the federated service and calculate the payment fee accordingly. Note that the provider domain has the ability to terminate the federation at any point in time D32. Establishing a monitoring and charging process can be quite challenging in dynamic environments as opposed to static environments. The use of Blockchain is promissing to integrate the charging process seamlessly Refaey, Hammad, Magierowski and Hossain (2019).

3.2. Federation in different environments

Thanks to virtualization (or NFV), external domains can control a network slice or resource in a constituent domain. In NSF, even though the provider domain is the owner of the underlying infrastructure, the orchestration and management of the end-to-end service is under control of the consumer domain. Thus, the NSF is not only a simple lease of services or network resources, it contains a trust dependent element where the consumer domain takes control over the provided services or infrastructure resources in the provider domain. Note that nowadays, a full control is almost impossible since at least the power grid would be always in the hands of the provider domain.

However, the credence element demands consumer and provider domain to establish trust through negotiation process and well-defined Service-Level Agreements (SLAs) for the NSF. In addition, the nature of the services for which a consumer domain requires NSF defines the environment dynamicity. For example, long-term service providing video-streaming services to users in a certain geographical area (e.g., different country, different continent) demands a well-defined static SLA between the consumer domain and the provider domain (with a service footprint in the area of interest). The SLA is usually defined through business meetings between the consumer and provider, on a peer-to-peer basis. The works in Baranda et al. (2020); J. Baranda et al. (2020a,b); Antevski et al. (2021) expand the topic of peer-to-peer NSF.

There are cases when a service or resource needs to be federated on-the-fly with a brief negotiation between the consumer and the provider domains. In such scenarios, a rapid definition of dynamic SLAs for short-term services can be defined through a centralized or distributed-peering approach. The centralized option assumes the presence of a centralized entity (e.g., an exchange platform) responsible

for federating network services or resources. Although it is an efficient way to establish NSF with a brief negotiation, the maintenance of a centralized entity might be an expensive solution. On top of that the centralized entity can be subject to hacker attacks as a single-point of failure, and it can be biased or leaning towards specific providers.

3.3. How Blockchain can be applied for federation

In this work, we focus on the distributed peering approach, in particular the use of Blockchain for NSF. The *Blockchain* technology emerged as a key underlying mechanism for Bitcoin, providing a distributed, secure, and time-stamped ledger that records every transaction between anonymous users. A Blockchain is built of interconnected nodes that share a single ledger. The ledger contains distributed time-stamped blocks filled with transactions that can contain any data. Each block points to the hash of the prior block, generating a chain (or history) of blocks, back until the genesis block (*block 0*). New blocks are generated and validated by the nodes (e.g., any computing device) that are interconnected in a peer-to-peer Blockchain network. There are two types of Blockchain networks: permissionless and permissioned. Each new block needs to be validated by a consensus mechanism which is a key performance component. There are different consensus mechanisms that provide different levels of security, trust, and privacy. Summarizing, the main benefits of applying Blockchain are (more information regarding the use of Blockchain in networking can be found in ETSI (2020, 2021a)):

- **Security.** The transaction data included in each block of the Blockchain is timestamped, tamper-proof and immutable. Data alteration is only feasible if at least 51% of the nodes are malicious/compromised.
- **Verifiability, integrity, and trust.** The state of the Blockchain is easily verifiable by all the members confirming an equivalent observed Blockchain state.
- **Smart Contracts.** Programmable applications that run as independent entities (or members) on top of a Blockchain (e.g., Ethereum). These applications have deterministic and atomic functions that can embed business logic and rules as in regular contract agreements.
- **Balanced privacy and transparency.** All transactions, state transitions, and blocks creations are transparent. Using cryptography enables private data to be encrypted and exclusive while maintaining the defined transition rules.
- **Third party absence.** The consensus mechanism enables collaboration among unknown members in a trusty manner without a third-party authority (e.g., central entity) to guarantee the integrity of the members.

In terms of NSF, besides the security and trust, domains are interested in the performance of Blockchain in terms of (*i*) the time it takes to federate a service; (*ii*) the required resources to support the NSF using Blockchain (e.g., CPU, RAM, storage, etc.) In our view, different Blockchain platforms can enable different levels of security, performance and execution of NSF.

Fig 1 shows our proposal of applying Blockchain for NSF in NFV MANO domains. An administrative domain is the service provider with the NFV MANO infrastructure. We propose the installation of a Blockchain node within each domain infrastructure with a direct connection on the Eastbound/Westbound interfaces of an NFV MANO platform (such as the ones presented in X. Li et al. (2020); Li, Garcia-Saavedra, Costa-Perez, Bernardos, Guimarães, Antevski, Mangues-Bafalluy, Baranda, Zeydan, Corujo et al. (2021)). By adding and registering a node, the NFV MANO domains gain access to a private/public Blockchain network. Additionally oracles might be part of the same network to realize the vision of QoS monitoring (Sec.7.5 of ETSI (2021a)). The Blockchain network is a straight-forward process of node registration.

A quick execution of NSF demands a Blockchain platform with a fast transactions processing and on-chain appending. The process of including transactions in blocks is part of the consensus mechanism of a Blockchain. Thus the consensus mechanism is directly dictating the transaction processing time, the security of a Blockchain and the resources it utilizes (in terms of CPU, memory and storage) to store transactions on-chain.

Hence, our work focuses in exploring how different consensus mechanisms perform the execution of network service federation. In the following we provide a brief description of the different consensus mechanisms. Later, we provide insight of the performance evaluation we performed on a (simple) experimental scenario.

4. Consensus mechanisms in Blockchain

To achieve our goal of comparing how different consensus mechanisms can influence the performance of NSF, we decided to compare them over a simple scenario. In this section we describe the consensus mechanisms that we are going to compare. Each of these consensus mechanisms are implemented in a platform that can be deployed and used for experimentation. For better description, we are coupling the description of the consensus mechanism with each of the platforms.

4.1. Proof-of-Work (PoW) - Ethereum

Proof of Work is the fist consensus mechanism implemented in the first Blockchain implementation - Bitcoin Nakamoto (2008). The same consensus mechanism is used for Ethereum, the first Blockchain platform supporting smart contracts.

The PoW consists of generation and validation of a new block. The process of generation a new block is when a Blockchain node (*i*) collects a finite number of pending transactions to form a block. The transactions are hashed to form a Merkle tree, the Merkle root is added in the block

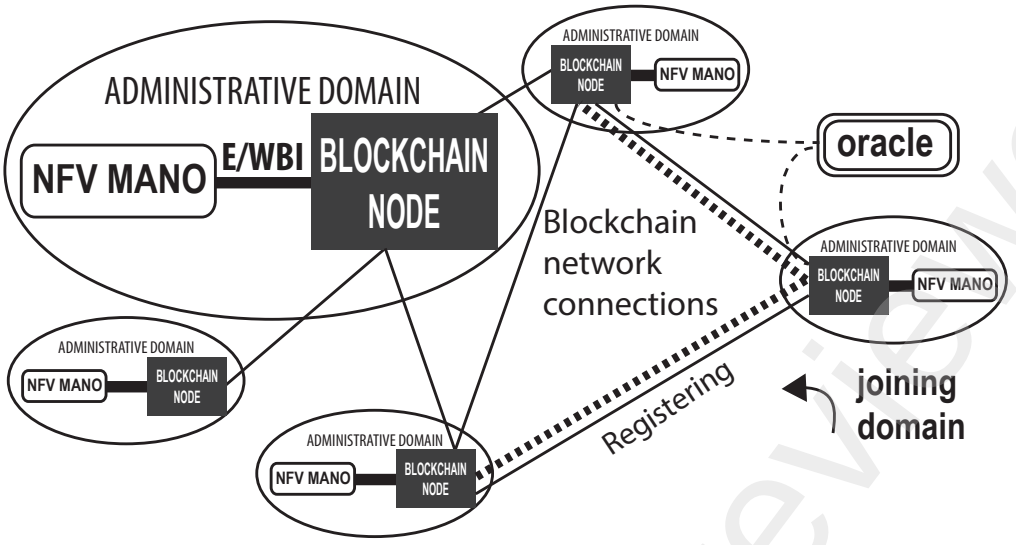

Figure 1: Blockchain application for NSF in NFV MANO domains

header. The Merkle root along with a timestamp, hash of the previously confirmed block, transaction count and nounce are added in the block header. In order the block to be valid, a node needs to compute a hash of a nounce that would produce a SHA-256 number with a defined number of leading zeros. Producing a SHA-256 hash with leading zeros is a computational intensive puzzle-solving work. The number of leading zeros represents the difficulty of the consensus mechanism. This is a fundamental feature that allows the Blockchain to adapt the *mining* difficulty when nodes with extra computational capability join the Blockchain network. Once the mining node successfully solves the puzzle and produces valid nounce, it broadcasts the created block into the network. Rest of the nodes can easily validate the result by simple hash of the nounce and the block header to produce the resulting block hash. The validated block is appended to the Blockchain ledger, and a new round of block creation starts.

The difficulty of the consensus mechanism is also adjusted to maintain the block time - time it takes to append a new block in the ledger. In Bitcoin, the block time is ∼10 minutes while in Ethereum it is ∼14 seconds. On average Ethereum is producing 15 transactions per second eth (a).

Besides the consensus mechanism, it is important to note that Ethereum implements smart contracts (introduced by Nick Szabo Szabo (1997)) on top of Ethereum Virtual Machine (EVM) eth (b). The EVM is a near Turing-complete on top of which the smart contracts are executed. Smart contracts contain set of rules/functions stored at specific account address in a form of a bytecode. Users use accounts to issue transactions to other users, or to smart contracts. When a user makes a message call to a smart contract, the bytecode is executed, and returns a result, changes a state, etc.

4.2. Proof-of-Authority (PoA) - Ethereum

In 2017, as a consequence of a spam attack to the Ethereum test network - Ropsten, a new test network was deployed using Proof of Authority (PoA) consensus mechanism PoA. The PoA consensus was proposed in the EIP-225 and later implemented in the Clique proof of authority protocol poa. The new protocol is maintaining the block structure as in PoW Ethereum, however instead of mining nodes competing to solve a difficult puzzle, there are pre-elected authorized signer nodes that can generate new blocks at any time. Each new block is endorsed by the list of signers and the last signer node is responsible for populating the new block with transactions. The transaction reward for each new block created is shared between all the signers Wang, Hoang, Hu, Xiong, Niyato, Wang, Wen and Kim (2019).

The Ethereum PoA permissionless test network - Kovan, has been released with the initial validators assigned to 12 independent public notaries with active commission license poa. In our experimental scenario, we are using a private instantiation of the Clique Ethereuem network which is explained further in details. The performance of PoA Blockchains depends on the number of signers. In private chains, the performance can reach ∼70 transactions per second Schäffer, Di Angelo and Salzer (2019).

4.3. Practical Byzantine-Fault Tolerant (PBFT) - Tendermint

The Byzantine-Fault Tolerant consensus mechanism is based on a property of a system that can resist the failures derived from the *Byzantine Generals' Problem* Lamport, Shostak and Pease (2019). The main characteristic of a BFT system is the ability of continuous nominal operation even if some of the participating nodes fail or act maliciously. When applied to a Blockchain realization, it has the ability to rule out validations from malicious nodes Salah, Rehman, Nizamuddin and Al-Fuqaha (2019).

Practical BFT aims for Blockchain with high performance (e.g., high transactional throughput, low latency, etc.), and high execution time. PBFT nodes of a permissioned Blockchain are sequentially ordered and all permitted nodes assist in attaining a consensus. The PBFT Blockchain is able to maintain the consensus if the maximum number of malicious nodes is not more than a third of all the participating Blockchain nodes. The Blockchain security increases with the increase of participating nodes.

Tendermint is an application-based Blockchain with a default Byzantine Fault-tolerant (BFT) consensus Kwon (2014); Amoordon and Rocha (2019). Tendermint enables users to turn any deterministic application into a Blockchain application through the use of the Tendermint BFT state-machine replication. Simplified, an application (as a state-machine) needs to be adapted to use an Application BlockChain Interface (ABCI) in order to communicate any state-transitions in form of transactions to the Tendermint Blockchain. On run-time, the Tendermint BFT consensus handles the state transitions by recording them into blocks of transactions. The state transistions are then replicated in each of the Blockchain nodes that run the same application. Hence, each application would run its own Blockchain (network) making the Tendermint an application-based Blockchain.

Unlike Bitcoin, blocks in Tendermint are added through voting by validators or validator nodes. The validators depend on how they are set. This can define if the set Tendermint network would be public or private. On top of that, a Proof-of-Stake (PoS) consensus can be employed. In that case, validators are user accounts/nodes that lock coins in a bond deposit transaction. In return, the validators gain voting power equal to the amount of bonded coins. In all cases, a block is validated and added to the Tendermint Blockchain when 2/3 of the voting power has signed and committed the block. Thus even if 1/3 of the validators fail, the Tendermint is still generating new blocks. Additionally users can run full-nodes or light nodes (suitable for IoT applications). A block is added in three rounds: (*i*) Proposal, (*ii*) Pevote and (*iii*) Precommit.

Tenderimint is a high performance Blockchain which can handle a maximum of 10^4 transactions per second Buchman (2016) with an average block latency of one second.

4.4. Proof-of-Stake (PoS) - Cosmos

The Proof of Stake consensus Blockchain is based on a Blockchain network of nodes that generate and validate new blocks differently than solving a complex puzzle as a proof of work. A PoS validator can generate (mint) or validate a new block with a probability equal to the Blockchain tokens/coins it holds. In PoS Blockchains the competition to generate a block is minimized compared to the PoW Blockchains. The node that generates the subsequent block is randomly chosen in a pseudo-random-selection process based on a combination of various Blockchain specific variables or processes (e.g., token staking) Saleh (2021).

Blockchain nodes that compete in the block generation process need to secure and lock a certain number of coins into the network as their stake. The size of the stake provides a linear probability for a node to be elected as the next-block validator - the bigger the stake, the higher the chances Bach, Mihaljevic and Zagar (2018). PoS is characterized Saleh (2021); Javed, Antevski, Mangues-Bafalluy, Giupponi and Bernardos (2022) as not fully decentralized Blockchain mechanism with high scalability, 50% fault tolerance and relatively high transaction throughput.

Cosmos is a network of many Tendermint Blockchains that are joined in a single Blockchain with a global transaction ordering cos. Considered as an upgrade of the Tendermint with a goal of enabling inter-operability between different applications realized as Tenderemint Blockchains. The mechanism for enabling the inter-communication is referred as Inter-Blockchain Communication (IBC). A first public Cosmos Blockchain is the Cosmos Hub which serves as a central ledger for multiple Zones or Tendermint Blockchains. The Cosmos Hub is PoS based and it has its own cryptocurrency - Atom. Users can stake Atoms to become validators or delegate their Atoms to trusted validator in order to earn portion from transaction fees. To maintain performance, there are limited amount of validators (e.g., up to 100 in the first year). Cosmos inherits the Tendermint performance and it is useful for connecting different Blockchains Schulte, Sigwart, Frauenthaler and Borkowski (2019) or realization of specific use-cases such as Decentralized Exchange (DEX) Lin (2019).

5. Performance of different consensus mechanisms in a federation scenario

In the previous sections we have described the federation steps to realize a NSF (Section 3.1), and described the different Blockchain platforms running over different consensus mechanisms (Section 4). In the following section we describe the experimental scenario and setup used to evaluate the performance of different consensus mechanisms when used for NSF. The obtained results are elaborated for each Blockchain platform in terms of execution time and utilization of resources.

5.1. Experimental scenario

The experimental setup contains three independent administrative domains. Each administrative domain contains an orchestrator, underlying infrastructure and a Blockchain node. The characteristics of each component are described in the following Section 5.2. The experimental scenario is divided into two parts: federation and healing.

The federation part simulates an extension of a network service through federation. In our experiment, we are not using an NFV MANO platform to deploy a network service. We use mininet to emulate the underlying infrastructure, and an orchestrator script that deploys network services as a predefined number of hosts organized in a network topology on a constituent mininet. Every domain runs an independent mininet instance. The federation is realized when a consumer domain deploys part of the service (two hosts in our

evaluation) in its constituent mininet and a number of hosts in a provider domain's mininet instance. The federation is considered successful when the hosts from the consumer domain are able to maintain a continuous communication (with no packet loss for finite amount of time) with the hosts from the provider domain. The healing procedure is similar, the consumer domain maintains the hosts active on its constituent mininet and issues establishes new federation with another provider domain.

At the start, the consumer domain sends a transaction to announce the desired extension of the service or additional hosts to be deployed. As both provider domains receive the announcement transaction and generate a bid-offer as a transaction, containing all the service details. The consumer domain receives the offers (transactions) and elects a winning provider domain which selection may be based on various things. In our case, we elect the domain using first-come-first-serve (FCFS) strategy. Both domains receive the consumer selection outcome, and the winning provider (#1) starts the deployment of the federated service (the requested number of hosts). The provider #2 returns to idle state. While the deployment is running, the consumer domain sends the connection details through a transaction. Upon deployment of the federated service, the provider #1 and the consumer domain establish the interconnection between both domains using VxLAN The deployed hosts are up and running promptly after the federated service deployment finished and inter-connection is established. Beside utilizing the service by having access to the deployed federated hosts, the consumer domain continuously monitors the connection if it satisfies a zero packet loss requirement. In order to emulate a healing procedure for the experimental duration, the federated service is kept up and running only for 10 seconds.

While in the second phase - healing, the federated service fails, and it is healed by performing a new federation procedure with another provider domain (provider #2). The healing is successful when the two hosts establish again an uninterrupted communication with hosts from the provider #2, similarly as in the federation procedure. In our experiments, the federated service is set to fail after 10 seconds which marks the start of the healing part. The consumer domain upon detection of two consecutive packet losses, issues a new federation announcement. For the new federation procedure, the provider #1 is blacklisted as an unreliable domain which leaves provider #2 as only winner. The provider #2 deploys the newly healed federated service using the same deployment steps.

Here we summarize the exact events measured during the experiments:

1. Service announced - *consumer*
2. Announce received - *providers*
3. A bid offer sent to consumer - *providers*
4. Winner choosen and broadcasted - *consumer*
5. Winner announcement received - *providers*
6. Deployed federated service - *winning provider*
7. Connection details sent to winning provider - *consumer*
8. E2E Service running - *consumer and winning provider*
9. Service stopped - *winning provider*

Note that the events of the healing service are measured in the same order. Additionally the 10 seconds countdown of the federated service starts at event (6), and it is stopped in the last step (9).

5.2. Experimental setup

The experimental setup is shown on Fig. 2. Each administrative domain consists of two host machines. The orchestrator and the underlying infrastructure are coupled in a mininet VM, an Ubuntu 14.04 virtual machine with 2 CPU cores, 2 GB of RAM, and 5 GB of disk memory. Each of the blockchain nodes are in different machines, an Ubuntu 18.04 virtual machine with 2 CPU cores, 6 GB of RAM, and 25 GB of disk. The experimental environment is isolated from the public network. Each machine has minimum features running to minimize the activity of background processes. Besides the different dependencies both on the mininet and Blockchain platforms, the decoupling would represent a real integration of Blockchain nodes into an existing infrastructure of a service provider or a mobile operator. A different hardware setup should reflect similar results. In other words, regardless of the hardware characteristics, each of the consensus mechanism experiments should show similar performances.

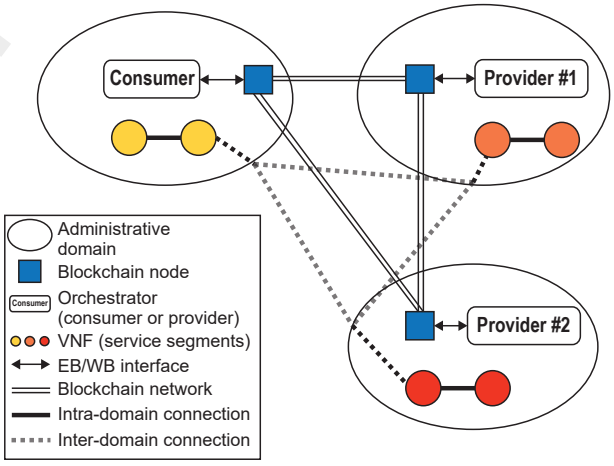

Figure 2: Experimental setup

5.3. Proof-of-work consensus profiling

First, we executed the experimental scenario using the Ethereum platform with Proof-of-work consensus mechanism. Note that the three PoW Ethereum nodes were mining simultaneously, competing each of them for the block reward. Each orchestrator (consumer or provider) is posting the transactions directly to the local Blockchain node. Fig. 3 presents the occurrence of all events listed in the previous section (Sec. 5.1). The narrow ticks represent the average

time of each event. Each event is numbered and labeled in the legend. The colored area show the variance of each event. Note that the color opacity increases where the subsequent events overlap with the variance ranges.

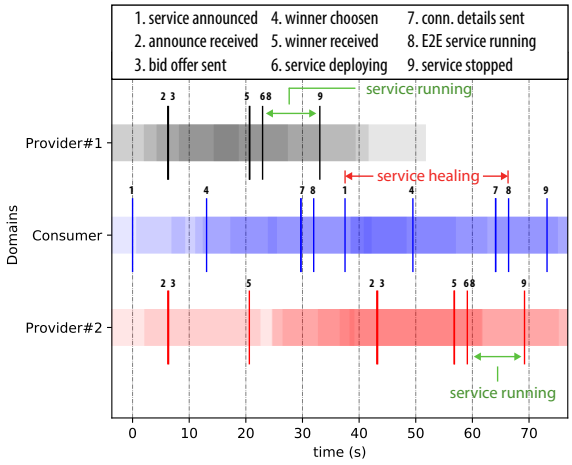

Figure 3: PoW event variance

As mentioned before, the provider domain is deploying and keeping the service up and running for a limited time of 10 seconds. From Fig. 3 is visible that in the case of PoW, the service is barely consumed by the consumer domain, due to high variance in transaction propagation.

For better view of what is happening in each Blockchain node, we monitored the CPU usage, the memory usage, the storage and network receiving for the duration of the experiments. Fig. 4 presents the profiling obtained for the duration of 20 consecutive experiments. From the obtained results, it is clear that the PoW consensus mechanism is saturating the CPU in each node up to 100%. The memory usage is constant while runing the experiments. Due to exchange of pending transactions and mined blocks, both the disk and the network activities are at moderate level. Note that an experimental execution on a more powerful hardware platform should provide similar hardware utilization (e.g., CPU, memory, disk, network). Additionally, we have isolated the experimental setup from any external network. Each VM contains minimal installed features to measure only the impact of the experimental process.

5.4. Proof-of-authority consensus profiling

We repeated the experimental scenario for Ethereum platform using PoA (Clique) consensus mechanism. The experiments consist 20 consecutive repetitions. On Fig. 5 are shown the event occurrences. In the PoA case, compared to PoW, the submitted transactions are mined more regularly, which reflects in lower federation time as well as lower variance ranges per events. The service federation is established within 15 seconds with no overlapping average times of events occurrences. The completed time of the PoA experiment is less than 50 seconds while in the PoW case is over 70 seconds.

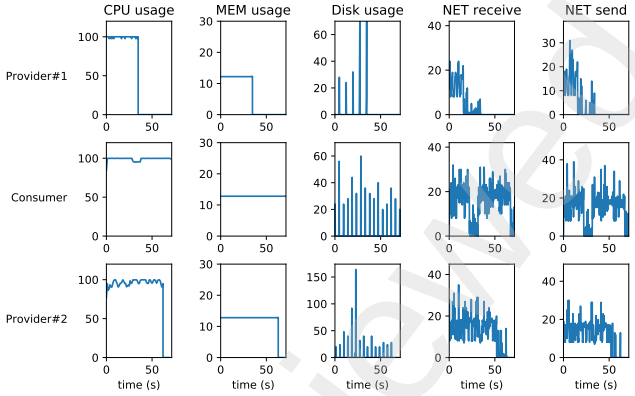

Figure 4: PoW profiling

Similar as in the PoW case, events overlap and are triggered in the same mainly due to inter-block times.

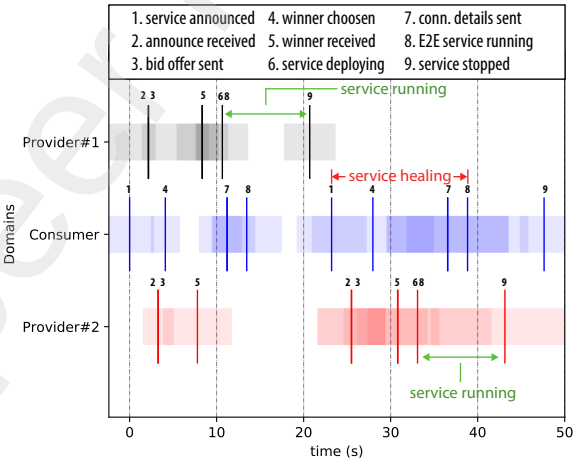

Figure 5: PoA event variance

The profiling of the PoA Blockchain nodes is presented in Fig. 6. The most evident is the low CPU usage. In contrast to PoW, the CPU load is less than 10% with small peaks in the Provider #1 domain. These peaks are due to the Provider #1 domain being the last validator and sealer of each newly created block. As mentioned in Section 4.4, the last validator is in charge of running the smart contract bytecode and sealing the block. The memory consumption is similar to the memory consumption of the PoW Ethereuem. There is a significant increase of disk and network activity. Even though the disk activity peaks are not significantly higher, the network activity of PoA is around 100% higher than the PoW driven Ethereum.

5.5. Practical Byzantine tolerance consensus profiling

Fig. 7 shows the experimental results obtained from the Tendermint platform using the PBFT consensus mechanism. From the obtained results, the average occurrence times of the events have lower variance compared to the the PoW

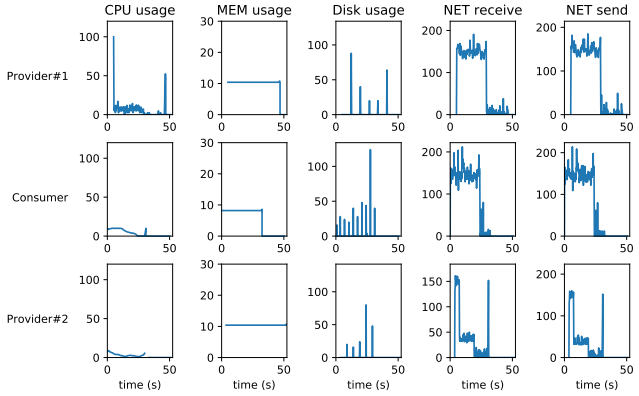

Figure 6: PoA profiling

or the PoA results. The system and execution stability is generally preserved for the all repetitive trials. The average completion time is lower than both the Ethereum PoA and the Ethereum PoW. The transaction propagation is almost instantaneous.

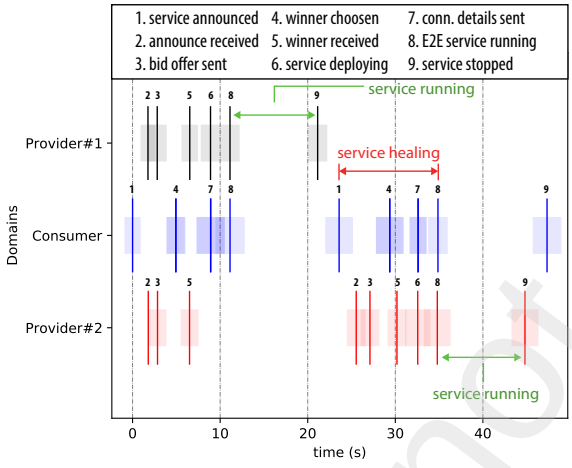

Figure 7: PBFT event variance

The performance of the Tendermint platform is displayed on Fig. 8. The CPU load shows that the Tendermint platform is very efficient is appending and exchanging transaction. The validation is not computationally demanding. Since the Tendermint is application-based Blockchain, only a single application can run on top of the Blockchain. In this case it is the federation application. Thus the CPU is significantly lower in contrast to the Ethereum PoW platform, where as a general purpose Blockchain many smart contracts can run on top. However there are not many differences compared to the Ethereum PoA. Memory-wise, the Tendermint platform takes over around 10% of the available memory, similar to the Ethereum platform. On the other hand, the disk activity is significantly increased compared to the Ethereum platform. This can potentially be problematic on the long run, mainly depending on the storage hardware used for the Tendermint nodes. The network activity is in the range of the Ethereum PoW platform, with increased picks when new federation announcements are submitted, mainly due to increased data exchange.

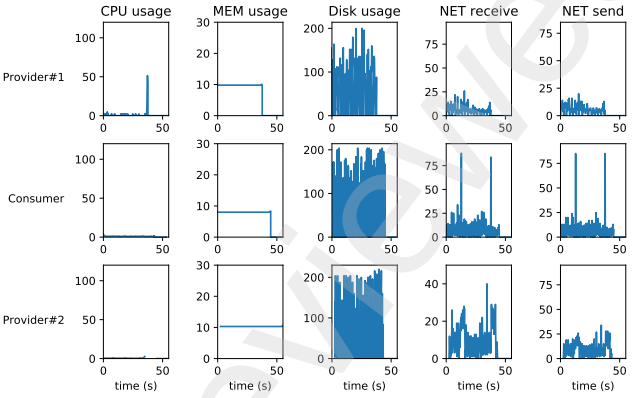

Figure 8: PBFT profiling

5.6. Proof-of-stake consensus profiling

The last evaluated consensus mechanism, the Cosmos PoS platform the averaged event occurrences are shown on Fig. 9. The average time to complete a federation is ∼ 26 seconds which is shorter in duration than Ethereum PoW, but longer than Ethereum PoA and PBFT Tendermint. The variance is significantly lower than Ethereum PoW.

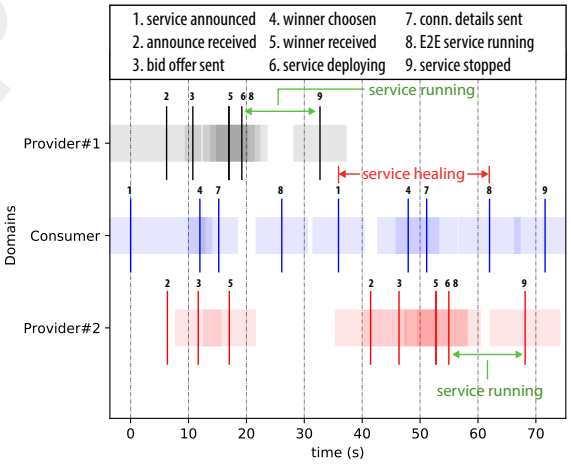

Figure 9: PoS event variance

The performance analysis of the Cosmos PoS is shown on Fig. 10. Even though the Cosmos platform is built on top of Tendermint, the computational overhead in verifying all the blocks is evident in the CPU load. The CPU increase of up to 50% is related to the generation of transactions from the given nodes. When a node is only validating and relaying blocks, the CPU load drops significantly, as in the case with Provider #1. The memory usage is standard up to 10% for all Blockchain platforms.

The storage activity is relatively high as in the Tendermint case. However the network activity is significantly

higher than rest of the Blockchain platforms. In our view, this is due to the increased size of data exchanged. The Cosmos PoS is an application of Tendermint itself, which by default adds an data overhead.

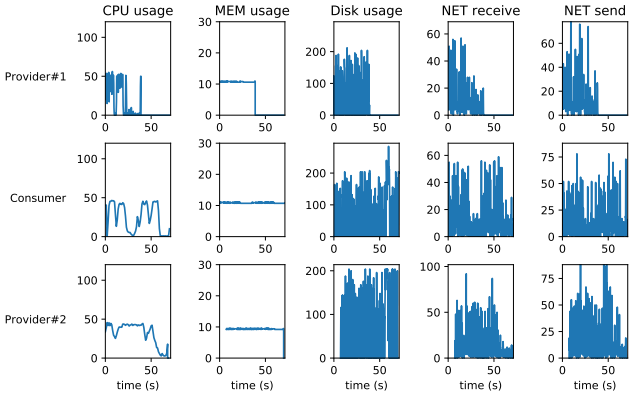

Figure 10: PoS profiling

6. Discussion

In this section we elaborate over the results presented in Section 5 and summarized in Table 1. Besides the measured results, we generated additional empirical metrics through the setup and running of the experiments which we find useful for evaluation of the platforms. These empirical metrics are: setup complexity, application development complexity, public chain portability, support for multiple applications and community support through documentation. Note that these are more subjective metrics given we have 5 developing experience on Ethereum platform. However, we argue that these metrics provide significant insight into the transition and adoption of Blockchain as a technology for NSF or other network applications, given the maturity level of the Blockchain technology (at the time of writing).

The average federation time shows the overhead of the NSF application if it is realized through the application of Blockchain technology Baranda et al. (2020); Antevski et al. (2021); J. Baranda et al. (2020b). The different consensus mechanisms, as we previously evaluated, have different security characteristics. To that end, the choice of more time-efficient consensus is a trade-off for choosing less secure and more centralized or permissioned systems. For example, the Ethereum PoW would be more suitable for open federation, where the participants does not demand stringent authentication procedures and anonymity is allowed. On the other hand, PBFT Tendermint or Ethereum PoA would be more suitable for rapidly changing dynamic environment, where a service demands a volatile edge infrastructure Antevski et al. (2020).

There is a big distinction in the CPU utilization for each of the consensus mechanisms. In the case of Tendermint and Ethreuem PoA, the Blockchain process activity has low effect on the CPU usage. The saturated CPU utilization in

Table 1
Consensus mechanisms and platforms comparison

		PoW - Ethereum	PoA - Ethereum	PBFT - Tendermint	PoS - Cosmos
Measured	Avg. federation time	32 seconds	14 seconds	11 seconds	26 seconds
	CPU utilization	High	Moderate Low	Low	Moderate
	Memory utilization	Low	Low	Low	Low
	Disk utilization	Moderate	Moderate-Low	High	High
	Network activity	Low	Moderate-High	Low	Low
Empirical	Setup complexity	Low	Medium	High	High
	App development complexity	Solidity (medium)	Solidity (medium)	Golang (medium-high)	Golang (medium-high)
	Support for multiple applications	Yes	Yes	No	Partially
	Portability to public Blockchain	Simple	Simple	Complex	Complex
	Community support	High	Medium-high	Low	Medium-low

Ethereum PoW demands higher performance computational infrastructure.

In terms of memory usage, every platform use a low memory usage (around 10%). The disk activity might be severe for the long-run, especially in the case of Tendermint and Cosmos platforms. Except in the Ethereum PoA case, there is a low network overhead which is suitable for federation of network services as well as other applications.

As mentioned before, the observed metrics are useful for future application of any of the evaluated Blockchain platforms. The setup complexity is straight-forward and well documented for both Ethereum PoW and PoA. The access points are well-defined, with various tools for deployment of smart contracts (e.g., Truffle, Hardhat). The block creation process is familiar to the original Bitcoin block creation process. The complexity of setting up Tendermint and Cosmos private Blockchain instances is significantly higher. Although running a single node environment is straightforward, the setup of multiple networks is not well documented and not very intuitive. We also want to note that this was the case at the time of setting up the experimental environment which might be improved afterwards.

The application development is not significantly different in terms of application logic. The Ethereum Virtual Machine (EVM) provide universal functions, definitions (e.g., addressing, balances) and variables (block numbers, states) that are not differ significantly between different EVM compilers. Smart contracts can be interconnected with other smart contracts and interact. An application might be distributed over several smart contracts. In Tendermint, it is up to the service providers to develop all the utility libraries on top (addressing, balances, etc.). Cosmos contains some of the default utilities, however it still demands very detailed application code which defines behaviors at each stage of the block creation.

The support for multiple applications or smart contracts might be crucial for future implementations. Both Ethereum platforms support running multiple smart contracts over the same Blockchain (EVM) instance. In this case the computational utilization is not (significantly) increasing for every new Blockchain smart contract in Ethereum PoW. This is not the case the PBFT Tendermint and PoS Cosmos platforms

which demand newly deployed Blockchain instance for each new application. Although there are Cosmos extensions that allow for enabling an Ethereum Virtual Machine (EVM) to run over Cosmos eth (c), the Cosmos performance might be degraded due to high overhead. Hence, in our view Cosmos has partial support for running multiple applications at the same Blockchain.

The portability to a public Blockchain (main network) is tightly related with the support for multiple applications. In the case with Ethereum, it is a straight-forward process that requires use of the provided tools (e.g., Truffle, Hardhat), or well defined APIs. There is no public Tendermint network, and in case of Cosmos, the portability is not straight-forward. Although the Inter-communication Blockchain Protocol (IBC) is designed to allow different Blockchain application instances to be able to communicate, the process is not simple.

Finally, the development communities of Ethereum, Tendermint and Cosmos is significantly different. Ethereum has already established and very active community. The Tendermint and Cosmos community is tightly working together and although they are quite centralized, the development is very active and constantly improving with the goal to catch-up the Ethereum.

7. Conclusion and future work

In this article, we have analyzed how different consensus mechanisms can be applied to the process of Network Service Federation. We are confident that any of the Blockchain applications can successfully improve the NSF process, especially in dynamic scenarios with unknown users.

We managed to set up an experimental scenario where we tested four different consensus mechanisms. Through the measurements and the process of adapting the scenario for each of the platforms, we managed to provide additional empirical observation. From our experience, every platform and consensus mechanism has its own benefits and drawbacks. In our view, the choice mainly depends of the application nature and the longevity of the solution.

As a future work, we plan to add other consensus mechanisms and execute the scenario using real NFVO MANO infrastructure while federating a real end-to-end NFV network service.

References

, . Blockchain-based decentralized cloud/fog solutions: Challenges, opportunities, and standards .

, . Clique PoA protocol. https://github.com/ethereum/EIPs/issues/225. (Accessed on 11/25/2021).

, . cosmos/whitepaper.md at master · cosmos/cosmos. https://github.com/cosmos/cosmos/blob/master/WHITEPAPER.md. (Accessed on 04/20/2021).

, . Deliverable 3.2 - Refined Design of 5G-CORAL. http://5g-coral.eu/wp-content/uploads/2019/06/D3.2.pdf. (Accessed on 02/05/2021).

, . Eip-225: Clique proof-of-authority consensus protocol. https://eips.ethereum.org/EIPS/eip-225. (Accessed on 11/25/2021).

, a. Ethereum (eth) blockchain explorer - etherchain.org - 2021. https://etherchain.org/. (Accessed on 11/25/2021).

, b. Ethereum Virtual Machine (EVM). https://ethereum.org/en/developers/docs/evm/. (Accessed on 11/25/2021).

, c. Ethermint - Ethereum Virtual Machine (EVM) as a Cosmos SDK module. https://docs.ethermint.zone/modules/evm/. (Accessed on 12/10/2021).

, . GSMA | Operator Platform Concept Whitepaper - Future Networks. https://www.gsma.com/futurenetworks/resources/operator-platform-concept-whitepaper/. (Accessed on 12/14/2021).

, . PoA Network Whitepaper. https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper. (Accessed on 11/25/2021).

V. Scoca et al., 2017. Smart contract negotiation in cloud computing, in: 2017 IEEE 10Th international conference on cloud computing (CLOUD), IEEE. pp. 592–599.

Amoordon, A., Rocha, H., 2019. Presenting tendermint: Idiosyncrasies, weaknesses, and good practices, in: 2019 IEEE International Workshop on Blockchain Oriented Software Engineering (IWBOSE), IEEE. pp. 44–49.

Antevski, K., Girletti, L., Bernardos, C.J., de la Oliva, A., Baranda, J., Mangues-Bafalluy, J., 2021. A 5g-based ehealth monitoring and emergency response system: Experience and lessons learned. IEEE Access 9, 131420–131429.

Antevski, K., Groshev, M., Baldoni, G., Bernardos, C.J., 2020. Dlt federation for edge robotics, in: 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE. pp. 71–76.

Azbeg, K., Ouchetto, O., Andaloussi, S.J., Fetjah, L., 2021. An overview of blockchain consensus algorithms: Comparison, challenges and future directions. Advances on Smart and Soft Computing , 357–369.

Bach, L., Mihaljevic, B., Zagar, M., 2018. Comparative analysis of blockchain consensus algorithms, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE. pp. 1545–1550.

Bamakan, S.M.H., Motavali, A., Bondarti, A.B., 2020. A survey of blockchain consensus algorithms performance evaluation criteria. Expert Systems with Applications 154, 113385.

Baranda, J., Mangues-Bafalluy, J., Vettori, L., Martínez, R., Antevski, K., Girletti, L., Bernardos, C.J., Tomakh, K., Kucherenko, D., Landi, G., Brenes, J., Li, X., Costa-Pérez, X., Ubaldi, F., Imbarlina, G., Gharbaoui, M., 2020. Nfv service federation: enabling multi-provider ehealth emergency services, in: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1322–1323. doi:10.1109/INFOCOMWKSHPS50562.2020.9162873.

Boubendir, A., Guillemin, F., Le Toquin, C., Alberi-Morel, M.L., Faucheux, F., Kerboeuf, S., Lafragette, J.L., Orlandi, B., 2018. Federation of cross-domain edge resources: a brokering architecture for network slicing, in: 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), IEEE. pp. 415–423.

Buchman, E., 2016. Tendermint: Byzantine fault tolerance in the age of blockchains. Ph.D. thesis.

Carlos J. Bernardos et al., 2016. 5GEx: realising a Europe-wide multi-domain framework for software-defined infrastructures. Transactions on Emerging Telecommunications Technologies 27.

Dabbagh, M., Choo, K.K.R., Beheshti, A., Tahir, M., Safa, N.S., 2021. A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities. computers & security 100, 102078.

De Angelis, S., 2018. Assessing security and performances of consensus algorithms for permissioned blockchains. arXiv preprint arXiv:1805.03490 .

ETSI, 2020. ETSI ISG PDL 003 V1.1.1, Permissioned Distributed Ledger (PDL); Application Scenarios .

ETSI, 2021a. ETSI ISG PDL 004 V1.1.1, Permissioned Distributed Ledgers (PDL) Smart Contracts System Architecture and Functional Specification .

ETSI, 2021b. ETSI ISG PDL 012 V0.0.3, Permissioned Distributed Ledgers (PDL); Reference Architecture; GS based on PDL-003 "Application Scenarios" .

Francescon, A., Baggio, G., Fedrizzi, R., Ferrusy, R., Yahiaz, I.G.B., Riggio, R., 2017. X–mano: Cross–domain management and orchestration of network services, in: 2017 IEEE Conference on Network Softwarization (NetSoft), IEEE. pp. 1–5.

Gao, W., Hatcher, W.G., Yu, W., 2018. A survey of blockchain: Techniques, applications, and challenges, in: 2018 27th international conference on computer communication and networks (ICCCN), IEEE. pp. 1–11.

J. Baranda et al., 2020a. 5G-TRANSFORMER meets Network Service Federation: design, implementation and evaluation. IEEE International Conference on Network Softwarization (NetSoft'20) .

J. Baranda et al., 2020b. Realizing the Network Service Federation Vision: Enabling Automated Multidomain Orchestration of Network Services. IEEE Vehicular Technology Magazine 15, 48–57.

Javed, F., Antevski, K., Mangues-Bafalluy, J., Giupponi, L., Bernardos, C.J., 2022. Distributed ledger technologies for network slicing: A survey. IEEE Access 10, 19412–19442.

Kwon, J., 2014. Tendermint: Consensus without mining. Draft v. 0.6, fall 1.

Lamport, L., Shostak, R., Pease, M., 2019. The byzantine generals problem, in: Concurrency: the Works of Leslie Lamport, pp. 203–226.

Li, X., Garcia-Saavedra, A., Costa-Perez, X., Bernardos, C.J., Guimarães, C., Antevski, K., Mangues-Bafalluy, J., Baranda, J., Zeydan, E., Corujo, D., et al., 2021. 5growth: An end-to-end service platform for automated deployment and management of vertical services over 5g networks. IEEE Communications Magazine 59, 84–90.

Lin, L.X., 2019. Deconstructing decentralized exchanges. Stanford Journal of Blockchain Law & Policy 2.

Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review , 21260.

Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A., 2020. Blockchain for 5g and beyond networks: A state of the art survey. Journal of Network and Computer Applications 166, 102693.

Refaey, A., Hammad, K., Magierowski, S., Hossain, E., 2019. A blockchain policy and charging control framework for roaming in cellular networks. IEEE Network 34, 170–177.

Salah, K., Rehman, M.H.U., Nizamuddin, N., Al-Fuqaha, A., 2019. Blockchain for AI: Review and open research challenges. IEEE Access 7, 10127–10149.

Saleh, F., 2021. Blockchain without waste: Proof-of-stake. The Review of financial studies 34, 1156–1190.

Saraf, C., Sabadra, S., 2018. Blockchain platforms: A compendium, in: 2018 IEEE International Conference on Innovative Research and Development (ICIRD), IEEE. pp. 1–6.

Schäffer, M., Di Angelo, M., Salzer, G., 2019. Performance and scalability of private ethereum blockchains, in: International Conference on Business Process Management, Springer. pp. 103–118.

Schulte, S., Sigwart, M., Frauenthaler, P., Borkowski, M., 2019. Towards blockchain interoperability, in: International Conference on Business Process Management, Springer. pp. 3–10.

Strobel, V., Castelló Ferrer, E., Dorigo, M., 2020. Blockchain technology secures robot swarms: a comparison of consensus protocols and their resilience to byzantine robots. Frontiers in Robotics and AI 7, 54.

Szabo, N., 1997. The idea of smart contracts. Nick Szabo's Papers and Concise Tutorials 6.

Wang, W., Hoang, D.T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., Kim, D.I., 2019. A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access 7, 22328–22370.

X. Li et al., 2020. Automating Vertical Services Deployments over the 5GT Platform. IEEE Communications Magazine 58, 44 – 50. doi:10.1109/MCOM.001.1900582.

Kiril Antevski received the B.S. degree in telecommunication engineering from the The Saints Cyril and Methodius University of Skopje, Macedonia in 2012 and the M.S. degree in telecommunication engineering from the the Politecnico di Milano, Milan, Italy in 2016. He is currently pursuing the Ph.D. degree in telematics engineering at University Carlos III Madrid (UC3M), Spain. His research interest includes the development of mechanisms to integrate and enhance NFV and MEC for 5G Networks in dynamic and heterogeneous environments.

Carlos J. Bernardos received a Telecommunication Engineering degree in 2003, and a PhD in Telematics in 2006, both from the University Carlos III of Madrid, where he worked as a research and teaching assistant from 2003 to 2008 and, since then, has worked as an Associate Professor. His research interests include IP mobility management, network virtualization, cloud computing, vehicular communications and experimental evaluation of mobile wireless networks. He has published over 70 scientific papers in international journals and conferences. He has participated in several EU funded projects, being the project coordinator of 5G-TRANSFORMER and 5Growth.