Journal of Network and Computer Applications 205 (2022) 103440

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

# PriFoB: A Privacy-aware Fog-enhanced Blockchain-based system for Global Accreditation and Credential Verification

Hamza Baniata [*], Attila Kertesz

*Department of Software Engineering, University of Szeged, Szeged, 6720, Hungary*

ARTICLE INFO

*Keywords:*
Blockchain
Fog Computing
Privacy
Global accreditation
Credential Verification
Self Sovereign Identity (SSI)

ABSTRACT

Trusted online credential management solutions are needed for instant and practical verification. Most of the available frameworks targeting this field violate the privacy of end-users or lack sufficient solutions in terms of security and Quality-of-Service (QoS). In this paper, we propose a Privacy-aware Fog-enhanced Blockchain-based online credential management solution, namely PriFoB. Our proposed solution adopts a public permissioned Blockchain model with different reliable encryption schemes, standardized Zero-Knowledge-Proofs (ZKPs) and Digital Signatures (DSs) within a Fog–Blockchain integrated framework, which is also GDPR compliant. We deploy both the Proof-of-Authority (PoA) and the Signatures-of-Work (SoW) consensus algorithms for efficient and secure handling of Verifiable Credentials (VCs) and global accreditation of VC issuers, respectively. Furthermore, we propose a novel three-dimensional DAG-based model of the Distributed Ledger (3DDL), and provide a ready-to-deploy PriFoB implementation. We discuss insights regarding the utilization and the potential of PriFoB, and evaluate it in terms of security, privacy, latency, throughput and power utilization. We analyze its performance in different layers of a Fog-enabled cloud architecture with simulation and emulation, and we show that PriFoB outperforms several Blockchain-based solutions utilizing Ethereum, Hyperledger Fabric, Hyperledger Besu and Hyperledger Indy platforms.

## 1. Introduction

### 1.1. Background

Credential recognition is the process where a (inter)national body, called Verifier, validates the legitimacy of a document that was issued by another body, also called as Issuer. A credential is issued upon an event occurrence to certify that this event has indeed happened, such as educational credentials, vaccination certificates, governmental passports/IDs, etc. Within one country, area, or continent, one can find agreed-on regulations to recognize a named type of credentials for purposes like governmental treatment, hiring, traveling, etc. However, once a person/entity that was issued a legitimate credential needs to approve it abroad, usually a painfully lengthy and costly process needs to be carried out. This is because credential documents generally include different types of stamps, proofs, identification numbers and other data that have to be verified individually and carefully for each credential referring to distinct, centralized, locally maintained databases (DBs). That is, no standard is used by issuers around the world and no constant way to prove different credentials is guaranteed. A credential may even need to be approved by the representing body of the issuer's country at the foreign country, e.g. Embassy, where the representing body needs to communicate with the national recognition, and several governmental, bodies that are related to the management of the credential subject. The higher the sensitivity of a credential, the more complicated and costly it is to validate it abroad.

Fog Computing (FC) is an emerging trend for extending the Cloud Computing (CC) technology to address computing and networking bottlenecks in large scale deployment of CC-assisted systems (Habibi et al., 2020). The basic idea of FC is to create a layer of distributed fog entities between the centralized cloud data centers/processors and end-user devices (or Things in the case of Internet-of-Things (IoT) systems Loffi et al., 2021) at the edge of the network. The FC layer should, conceptually, control the handling of users' data in a private and secure manner, while providing cloud services. Utilizing the fog was proven to be more efficient, than classical, centralized cloud-based services, in terms of overall system throughput (Mutlag et al., 2020), response latency (Khosroabadi et al., 2021), storage efficiency (Wang et al., 2018), and privacy (Arif et al., 2020).

A Blockchain (BC) system is a distributed computer system that is able to store and process a distributed ledger (DL) in a Peer-to-Peer (P2P) network model (Zheng et al., 2018). A BC can be permissioned, where participants added by authorized entities within the system are

[*] Corresponding author.
*E-mail address:* baniatah@inf.u-szeged.hu (H. Baniata).

https://doi.org/10.1016/j.jnca.2022.103440
Received 13 January 2022; Received in revised form 3 May 2022; Accepted 31 May 2022
Available online 14 June 2022

capable of appending (or mining) new blocks onto the DL. Permissionless BCs, on the other hand, allow anyone to participate as a miner in the system without the need for an access grant. Data saved on BC DLs can be either public or private, depending on the domain of users able to request or perform tasks within the system. That is, anyone can access a public BC (e.g. read data on the chain and/or request to be a miner), while only users within a predefined domain within the organization, region, etc. can access data on a private BC. Different BC functionalities appeared in the literature, mostly data management (Vo et al., 2018), payment and trading management (Notheisen et al., 2017), reputation management (Dennis and Owen, 2015), and identity management (Dunphy and Petitcolas, 2018). In the scenarios of Smart Everything (Langley et al., 2021), the required automation criterion needs highly trusted and secure processing of end-users data. The General Data Protection Regulations (GDPR) (European Commission, 2020) enforced in Europe back in 2018, suggests wide variety of practices that should be used in order to protect the privacy of end-users. These regulations are hard to achieve (yet doable) within a BC-based system, due to the immutability of data saved on-chain. Thus, system architects need to take precise and clear measures of what data is saved on/off chain, when BC-based solutions to be utilized.

We have previously surveyed the integration of BC and FC systems in Baniata and Kertesz (2020), and found that the integration of these two technologies revealed good potential lying beyond, if collaboratively interacted. Regarding credential and identity management in such an innovative integration, projects that consider GDPR regulations mostly deployed centralized DBs for handling private data, such as the cloud or a TTP server (Chakroun and Keevy, 2018). Others, that save private data on the immutable DL, proposed some access control mechanisms, so that only authorized entities can read or add private data (which is still a non GDPR-compliant practice).

In summary, state-of-the-art solutions have one, or more, of the following disadvantages:

- The solution is local, resulting in limited utilization and recognition of the solution abroad.
- The solution is limited to a specific application, scope, or type of credential. Adopting the solution for other applications requires extensive modifications.
- The solution is not secure, not privacy-preserving, or not GDPR compliant.
- The solution is not efficient (e.g. high latency, low throughput, high energy consuming, etc.)

### 1.2. Motivations

Several previous studies looked for solutions to overcome the above mentioned drawbacks of centralized solutions. The most recent example of such a solution is the EU Digital COVID Certificate,[1] where authorized governmental bodies within Europe update a central DB that includes personal data on vaccination. Using this service, vaccinated people, or their agents, can prove that they have received a vaccine within a European country, allowing them to travel abroad without, e.g. quarantine, restrictions. However, private data of those agents in such a scheme is not only exposed to national, but also to international governmental personnel/systems. Additionally, locally vaccinated people need to individually register to several different platforms, before they can obtain an EU accredited vaccination certificate. Although such data management schemes are not compliant with the GDPR, it apparently was the only available approach to relax the pandemic's restrictions as soon as possible. In other, more sensitive cases, such as foreign educational diploma recognition or voting systems, privacy awareness is a critical factor that needs to be adopted by design in any proposed solution.

To this end, we were motivated to implement a solution that resolves all of these issues. Specifically, our implemented solution should fulfill the following objectives:

1. The proposed solution must allow multi-party co-operation to ***accredit*** credential issuers.
2. The proposed solution needs to be **generic** to any type of credentials, and scalable to allow ***global credential issuance and verification*** for all of system entities (i.e. accreditation bodies, issuers and end-users).
3. Issued credentials must provably adhere to state-of-the-art ***security and privacy*** measures.
4. The proposed solution must be fully ***GDPR-compliant***.
5. The proposed solution should outperform current solutions, in terms of ***efficiency*** (i.e. Latency, Throughput, Storage and Energy consumption).

### 1.3. Contributions

To achieve the above objectives, we utilize in this work a public-permissioned BC and a Fog layer to propose an efficient system for global institution accreditation and credential verification, namely PriFoB. The BC in PriFoB acts as a Distributed Trusted Third Party (DTTP), in which miners are national accreditation bodies (e.g. national ministry of higher education, ministry of foreign affairs or ministry of health affairs, etc.). On the other hand, fog nodes are realized by credential issuer bodies (e.g. universities, hospitals, vaccination centers, etc.). Our contributions can be concluded as follows:

1. We implement PriFoB to guarantee privacy and security of system entities, by deploying a hybrid, fine-tuned PoA-SoW Consensus Algorithm (CA). PriFoB utilizes the robust PoA (Singh et al., 2019), which provably allows a scalable BC network. The Signatures-of-Work (SoW) (Garay et al., 2020) utilization, along with the public-permissioned BC model, makes practical realization of PriFoB as a global accreditation solution.
2. We use AES and RSA encryption to insure secure data management and sharing among system entities. We use DSs and ZKP mechanisms, while all private data are saved off-chain at the data owners' infrastructure, making PriFoB not only privacy-preserving and GDPR-compliant, but also storage efficient.
3. We implement and test a relaxed and efficient multi-dimensional DL model, where blocks are partially (instead of fully) immutable. Each dimension holds a different type of transactions (TXs), which enhances the overall efficiency of the system. Within each dimension, we designed an improved Directed Acyclic Graph (DAG) based block relations which experimentally outperforms classical linear models in terms of total throughput and response latency (Pervez et al., 2018).

The remaining part of the paper is structured as follows: Section 2 describes our proposed solution and highlights the main methods deployed within. Section 3 presents the evaluation results in terms of security, privacy, latency, throughput and power utilization. Section 4 presents the state-of-the-art regarding BC-based solutions proposed for solving similar problems we target. Finally, Section 5 concludes our work.

## 2. System modeling

The main objective of our proposed system is the simultaneous provision of two major services: (i) Institution Accreditation and (ii) Credential Verification. The proposed system must be privacy-preserving by design, meaning that the deployed communications protocols and interaction/processing methods shall allow no window for private data

[1] https://ec.europa.eu/info/live-work-travel-eu/coronavirus-response/safe-covid-19-vaccines-europeans/eu-digital-covid-certificate.

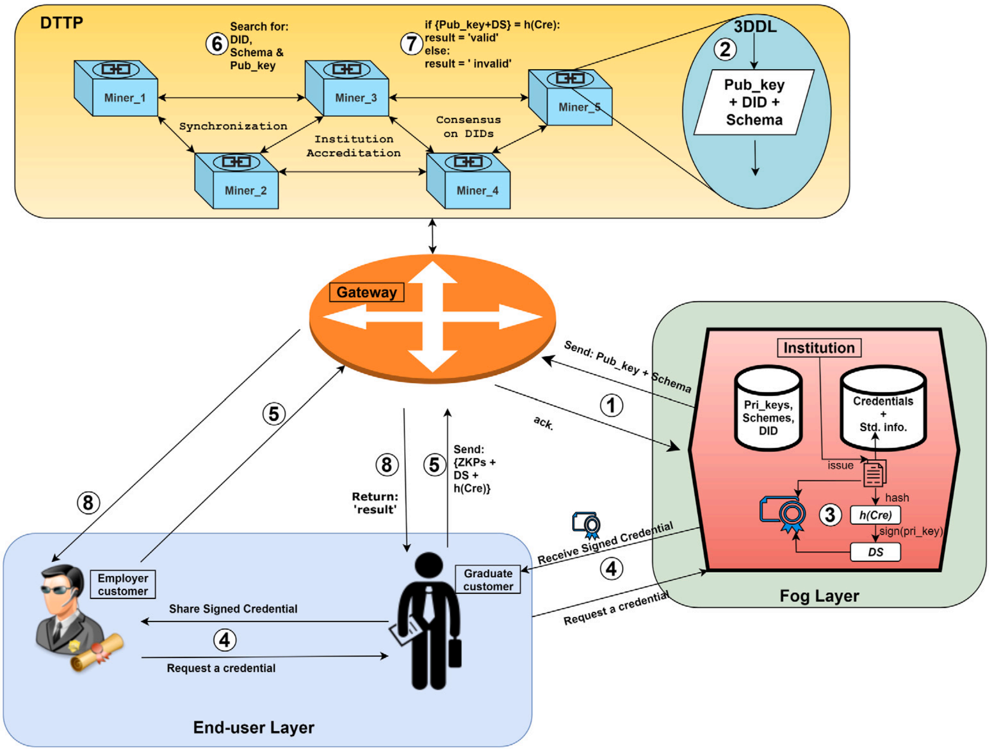


**Fig. 1.** The general architecture and framework of entities in a PriFoB system. The circled numbers indicates the order of steps to remotely accredit an issuer, publish new schemes, issue a new VC by an accredited issuer and verify that VC.

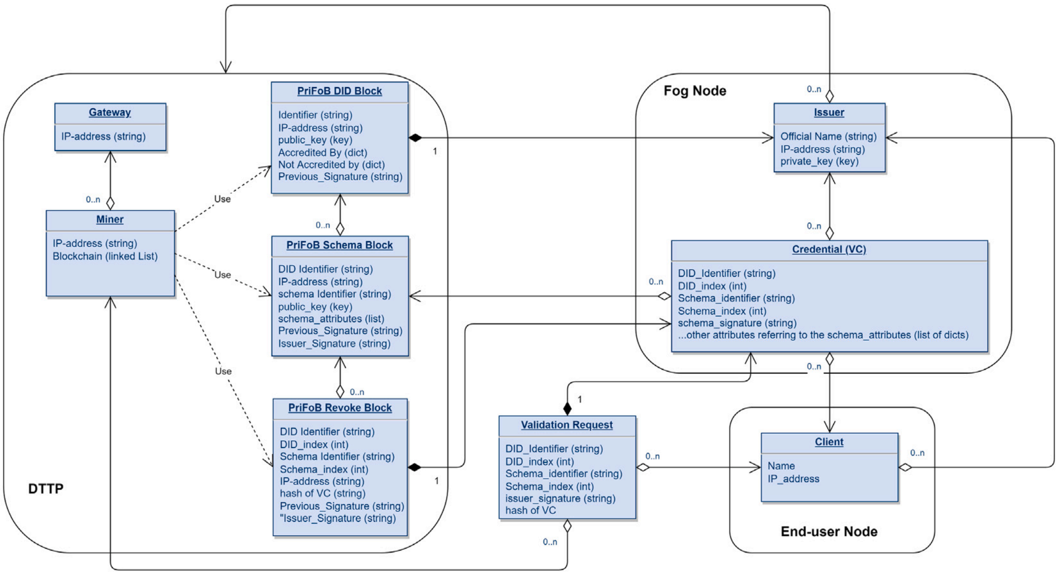


**Fig. 2.** Data model and dependency relations of the PriFoB components.

leakage. The efficiency of the proposed solution can then be evaluated in terms of security, privacy, average throughput, storage cost and energy consumption (or processing cost). To address all of these goals, we propose Privacy-aware Fog-enhanced Blockchain-based Institution Accreditation and Credential Verification (PriFoB), which utilizes different technologies namely FC, BC, DSs and ZKP methods. Specifically, we utilize a public permissioned BC with PoA and a realized SoW CAs. Meanwhile, we use the robust SHA-256, AES and RSA encryption methods. We deploy a BC infrastructure to serve as a DTTP for a decentralized pool of (inter)national accreditation organizations.

The general architecture of the proposed PriFoB, and a simplified control flow scheme within, are depicted in Fig. 1. We made the source-code of the PriFoB solution, along with a tutorial on setup and deployment, publicly available at Github,[2] where the modules interact with each other as demonstrated in Fig. 2. To give clearer description on the roles, groups and components of PriFoB, we further represent

[2] https://github.com/sed-inf-u-szeged/PriFoB.

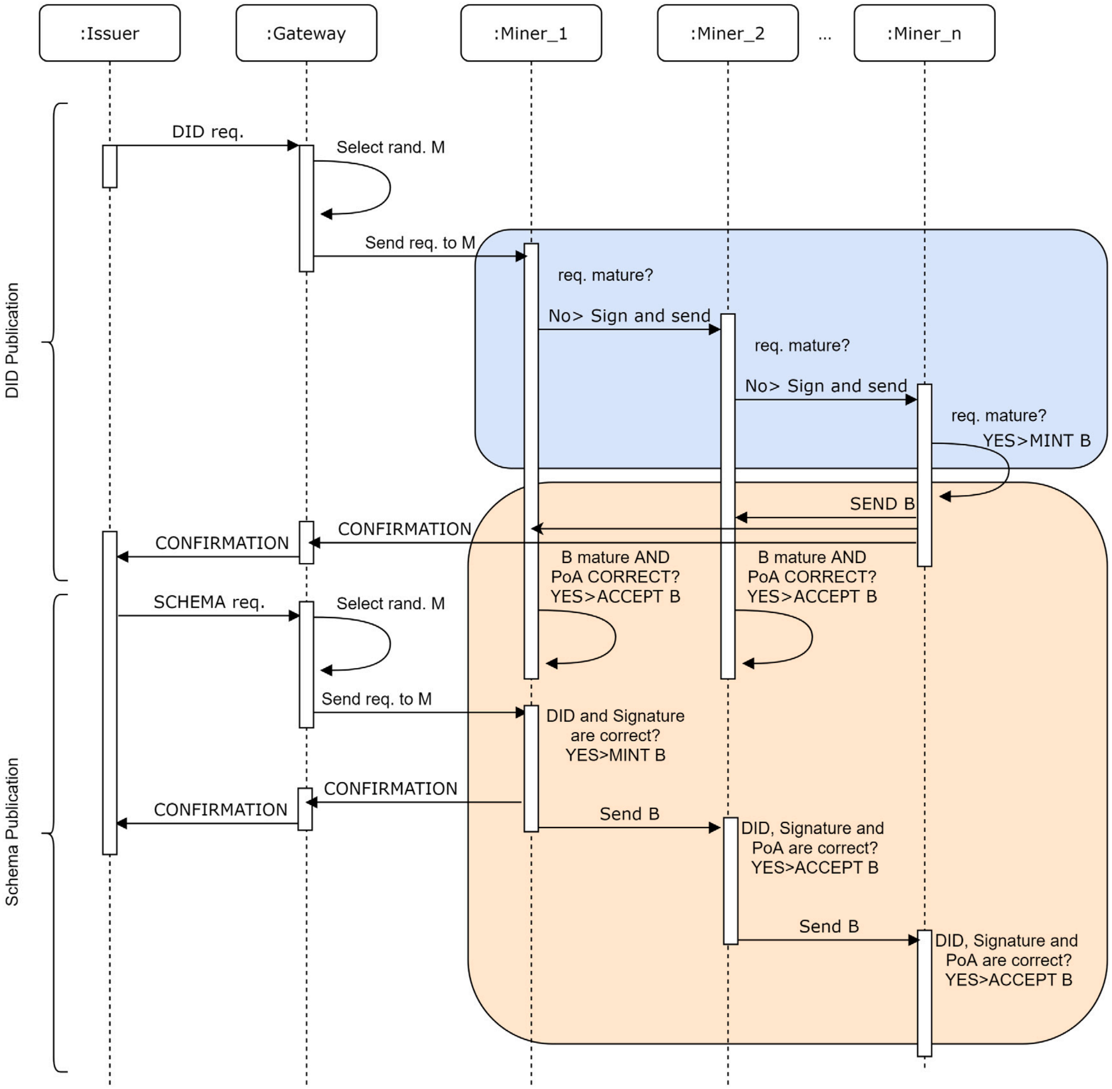


**Fig. 3.** PriFoB Control Flow for publishing DIDs and schemes. Blue and Light Salmon colored areas indicate, where the SoW and the PoA consensus algorithms are utilized, respectively.

PriFoB using the AGR4BS model (Roussille et al., 2021) in Appendix A. As demonstrated in the figures, PriFoB consists of three major layers:

1. The DTTP layer: which consists of the Gateway (GW) and Miners. The GW connects the BC network with the issuers and end-users. Furthermore, it is responsible for bootstrapping new miners with randomly selected peers. Miners, on the other hand are responsible for verifying new blocks and maintaining the consistency of the DL. Furthermore, miners are responsible for validating VCs using DSs and ZKPs.
2. End-user layer: consists of regular end-users requesting to be issued VCs. Those end-users can request, validate, share, or rename their issued VCs. Additionally, those end-users can download the whole or part of the BC. Entities belonging to this layer are not allowed to write on-chain.
3. The Fog layer: consists of issuer(s). An issuer is an extended end-user entity, which is responsible for issuing new VCs to other qualifying end-users. To do so, an issuer is initially required to publish its unique Decentralized Identifier (DID) and a schema(s) (which is a VC template) into the DTTP through the GW. Once both are published, it can issue as many VCs as it needs.

Note that in addition to the functions a non-issuer end-user can perform, an issuer entity can also write on-chain (DID and Schemes), and it can revoke a VC that it has previously issued. Issuers are set in the fog layer because they can be directly connected to regular end-users without a middling element, and they do provide several types of services to them. However, they are still considered end-users from the DTTP point of view, since the DTTP provides services for them. Some issuers can belong to both end-user layer and fog layer at the same time, since they can request VCs from other issuers as well.

Next, we discuss the PriFoB protocol and different types of messages exchanged between PriFoB entities. Accordingly, we arrive at a conclusion about when to use different types of encryption schemes (detailed in following subsections).

### 2.1. PriFoB Protocol

Knowing that using asymmetric encryption is computationally expensive and bounded, we only use it in PriFoB when necessary. In the following subsections, we detail each step of the PriFoB protocol to clarify the simplified framework depicted in Fig. 1.

#### 2.1.1. Accreditation

As depicted in Fig. 1, the first step for an issuer to be able to issue new VCs is to be accredited. To do that, the issuer generates a public key and sends it to the DTTP along with non-private issuer information (e.g. official name, IP-address, etc.). The corresponding private key is then saved and managed locally by the issuer. No encryption is needed in this step, other than ordinary symmetric encryption performed within the frame of the TCP/IP protocol. The combination of data sent by an issuer to request accreditation is called a DID request. Once the request is accepted and the DID is published on-chain (i.e. as a DID block), we say the issuer is accredited and can issue VC schemes. A sample PriFoB DID is provided in Appendix B.

Fig. 3 shows the control flow for publishing DIDs (i.e. accrediting an issuer). Miners perform the SoW CA on DID blocks (detailed later in this manuscript) in order to maintain the DL consistency.

### 2.1.2. Schema publication

A schema is a VC template that is saved on-chain to refer to later when a VC to be issued/verified. In addition to the DID definitions to which a schema relates to, a schema consists of its own public key and the fields that needs to be filled for each VC of this type. That is, each schema corresponds to a unique type of VCs. A sample PriFoB schema is provided in Appendix B.

Publishing a new schema upon issuer accreditation includes sending non-private information (e.g. Schema title, public key, etc.). Fig. 3 shows the control flow for publishing a new schema after accrediting the issuer. Miners perform the PoA CA on schema blocks (detailed later in this manuscript) in order to maintain the DL consistency. Once the issuer is accredited and its schemes have been published, it can generate new VCs to its clients.

### 2.1.3. Issuing a verifiable credential

As depicted in Fig. 1, the third step of the PriFoB protocol after publishing a schema is issuing new VCs with its clients' private data (e.g. name, grade, birth-info, etc.) and perhaps with its own shareable private data as well (e.g. courses, professors' names, admin and registrar's signatures, etc.). Thus, this VC is only saved locally, as we assume that, trivially, customers of an issuer do trust that issuer with their private data. An end-user (e.g. student or hospital patient, we also interchangeably use the terms client, customer, or agent) sends a VC request to the issuer, which also includes the client's public key. The request shall include some private identifying information (e.g. full name, SSN, year of credential issuing, etc.), so that the issuer can share a VC representing the original credential with high confidence that the requester is the client herself. Mandatory identifying data are declared in the schema. Because of that, the client connects with the BC network to ask for the issuer's public key (which was initially saved on-chain in the first step of the protocol) and the mandatory data that needs to be submitted. Using this public key, the client can encrypt her credential request that includes the mandatory information. Consequently, no entity but the issuer can read private data within a VC request, as only its private key can decipher the request. The issuer can then use the client's public key to encrypt its response.

If a new type of VCs to be issued, a new schema needs to be submitted and, only after saved on-chain, the new type of VCs can be issued. Note that old schemes remain saved on-chain as the BC provides immutable storage, thus old fashioned VCs remain verifiable despite a new schema application. This is both beneficial and critical. It is beneficial for old scholars who are guaranteed they will not lose their credibility even if the issuer's system is changed. However, it is critical if, for some reason, the issuer decided that some of its previously issued VCs should be considered invalid. In our proposed PriFoB system, we solve these issues by utilizing our proposed 3DDL as discussed in Section 2.3.4.

Mainly, the issuer response shall include two parts (assuming the requested credential was indeed issued) the digital credential, and $Sig$ (using the schema's private key). The encrypted response, which is in fact the VC, can only be then read by the client, as only her private key can decipher it. This is the fourth step of the protocol in Fig. 1. Once the response is decrypted, the client is free to share and verify the obtained VC (repeat step 4). Fig. 4 shows the control flow for issuing and sharing a VC.

### 2.1.4. Credential verification

The client may send a verification request to the DTTP (step 5 in Fig. 1). The verification request includes only non-private data, including: the DID block identifier and index (e.g. issuer official name and its index on-chain), the schema block identifier and index (which might be similar for different issuers but unique for each issuer), the hash of the credential to be validated, and the signature originally provided by the issuer within the VC.

Note that none of these data can reveal any private information about the client. Accordingly, asymmetric encryption is not required here. Once the BC receives the verification request, a miner, randomly selected by the GW according to the implemented load-balancing criteria, performs a VC verification (steps 6 and 7, technically described in Sections 2.3.2 and 2.3.3). The output of this step defines the response from the DTTP to the client (step 8). That is, the response is either Valid or Invalid, yet the reason for considering a VC invalid shall be also provided. Reasons for considering a VC invalid include: the DID or Schema has not been published on-chain, the hash of the credential is not equivalent to the decrypted $Sig$, or the VC has been revoked by the issuer.

Miners search for a DID with a claimed identifier and index. If found, it searches within the schemes chain within this DID block for the schema identifier and index. Otherwise, a response is sent to the requester that the issuer/schema is not accredited/registered. Once the schema is found, it searches within the revoke chain within this schema block for the hash of the credential provided within the request. If not found, then the signature is verified using the public key of the schema found. If the signature is valid, and the hash is not in the revoke chain, a response is sent to the requester that the credential is valid. if the signature is valid but the hash is in the revoke chain, a response is sent to the requester that the credential is revoked.

Note that no private data are saved on-chain, or provided for validation. This is the ZKPs scheme we use as the VC is validated without any knowledge requirement. The requester need not to expose any private data other than its address to which the response should be send. The VC validation is performed automatically and publicly without any restrictions or conditions. If a client decides to download the publicly available DL, it can verify any VC without referring to the DTTP in PriFoB. Fig. 4 shows the control flow for verifying a VC.

### 2.1.5. Revoking a credential

If an issuer decides to revoke a credential, a revoke request needs to be sent to the DTTP. The revoke request consists of DID and Schema data and the hash of the VC to be revoked. The request should be signed using both the DID private key and the Schema private key and thus the miner handling the request is assured the VC to be revoked is placed correctly and the request is legit. Once the issuer's signatures are verified, a revoke block is added on-chain. Later on, if a client attempts to verify this VC, the DTTP will respond with Revoked instead of Valid.

Fig. 4 shows the control flow for revoking a VC. Miners perform the PoA CA on revoke blocks (detailed later in this manuscript) in order to maintain the DL consistency. A sample revoke block is provided in Appendix B.

## 2.2. Infrastructure modeling

The communication between end-users and issuers can be conceptually viewed as an end-user to fog communication. That is, a client, who is granted a VC, can directly request a VC from a fog server (i.e. issuer server) closer to them, than from a web-based cloud application. Noticing that PriFoB is mainly targeting global VC verification, it is recommended to have a distributed TTP to handle requests from anywhere. The advantage of such DTTP is demonstrated through task distribution leading to higher throughput compared to centralized TTPs. However, the DTTP should consist of trusted bodies representing national accreditation organizations. Examples of such trusted bodies

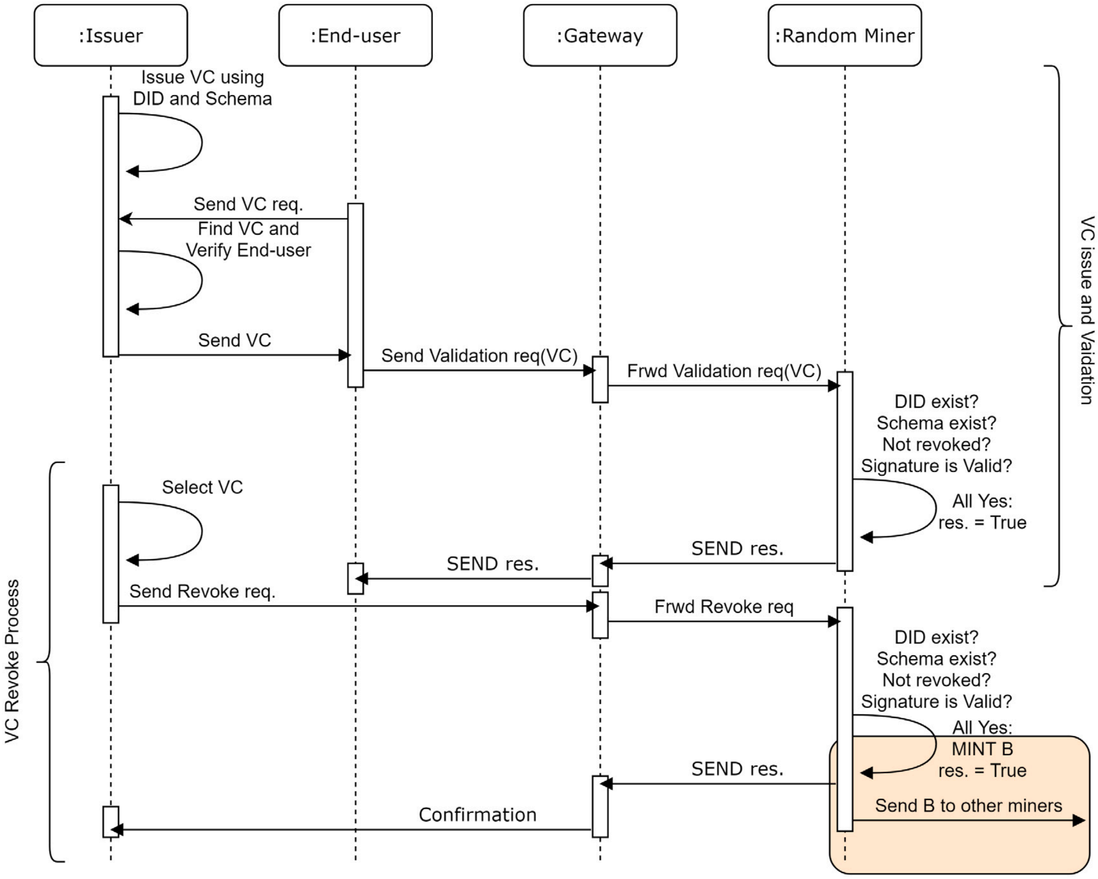


**Fig. 4.** PriFoB control flow for issuing a new verifiable credential and revoke a previously issued one. Light Salmon colored area indicates where the PoA consensus algorithm is utilized.

are: a ministry of higher education, or a governmental regulatory authority. End-users may be PCs, mobile devices or even microcomputers. A GW is then needed to load-balance the tasks received by the DTTP, and to control communications from/to the miners, which are the main components of the DTTP. Connecting miners into a P2P network, and the expected heavy computational and storage load on them, suggest that miner nodes and the GW cannot be handled by resource-limited devices like end-users'. Each miner node, along with the GW, can be administered by a human factor, or by an automation tool according to a regularly updated DB.

Our implementation of all PriFoB entities are containerized using Docker (Shah and Dubaria, 2019). This architecture allows for easy enhancement of the system in terms of scalability. That is, a clustering and/or scaling service can be utilized, using e.g. Kubernetes, which allows for container orchestration. For example, multiple GWs can be utilized instead of one, and multiple containers can be utilized per miners. To clarify, a miner is attributed using its IP-address, location, and the administrative organization that owns it. Each miner can then be logically represented to the GW as one machine, yet several machines, with several local containers and addresses, can process the requests received by the cluster. Similar deployment options can be utilized for PriFoB issuers, i.e. the fog layer components, leading to a multi-level fog set-up (Shaik and Baskiyar, 2018). For example, a department within a faculty is allowed to issue a limited number of schemes, or it is allowed to utilize specific schemes' private keys. Thus, no department should issue VCs for degrees that are not obtained within. On a higher level, a VC maybe revoked only by the faculty head office or the university head office, which implies that the private keys of the university should be managed, specifically, by the office allowed to use it.

### 2.3. Data layer modeling

In this subsection, we discuss how and when different types of data can be encrypted, decrypted, signed, and verified. Furthermore, we explain types of messages and ZKPs exchanged between different entities of PriFoB.

#### 2.3.1. One-way, symmetric and asymmetric encryption

In cryptography, there are many types of encryption used to hide shared information. A hashing function $h(.)$, or a one-way encryption function, is a mathematical function that takes a variable-length input string and converts it into a fixed-length binary sequence that is computationally difficult to invert (Thomas Porter et al., 2011). A hashing function enables the determination of a message's integrity: any change to the message will, with a very high probability, result in a different message digest (Johnson, 2019). In PriFoB, we use the SHA-256 (Yoshida and Biryukov, 2005) function for one-way encryption.

Symmetric encryption is the process of turning a readable text (plain text $P$) into a non-understandable text (cipher $C$) using an encryption function $E(.)$ (De Cannière, 2007). The input of a symmetric encryption function is $P$ or $C$, and a key $k$, leading to $E(P,k) = C$. The processes performed by $E(.)$ shall be traversed, using $k$, if $P$ to be derived from $C$. That is, $E^{-1}(C,k) = P$. In PriFoB, we use the AES methods (Akkar and Giraud, 2001) for the symmetric encryption.

Asymmetric encryption is a secure method $S(M,k)$ used to ensure that only the receiver is able to decrypt $D(C,g)$ a cipher and read the original message $M$. We deploy this type of encryption in PriFoB, in addition to one-way and symmetric encryption methods, so that the security of exchanged messages that include private data is guaranteed. This type of encryption implies the generation of two keys, $g$

which decrypts $C$ that was originally encrypted by $k$. A message $M$ that was encrypted using $k$ is computationally hard to be decrypted unless $g$ is known. One of the most secure and famous asymmetric encryption algorithms is the RSA algorithm (Rivest et al., 1978). The RSA key generation, encryption and decryption processes are described in Appendix C.

### 2.3.2. Digital signature and verification

The RSA keys generated above can be swapped without the loss of generality. That is, $g$ may be used to decrypt a cipher $C$ that was encrypted using $k$, or to encrypt a message $M$ to verify the credibility of $M$'s origin. Specifically:

1. The original sender of $M$ computes: $S(h(M), g) \rightarrow Sig$, where $h(.)$ is an agreed on hashing function (e.g. SHA-256),
2. The sender sends the resulting Signature ($Sig$) along with $M$ $[M, Sig]$ to the receiver,
3. The receiver computes $h(M)$,
4. The receiver computes $D(Sig, k) \rightarrow h'(M)$,
5. if $h(M) = h'(M)$, the receiver shall be confident that $M$ was sent by the original sender who is the only one that can read $g$.

Because this method is typically used to prove the sender credibility, while anyone can decrypt $Sig$ using the publicly available $k$, it is called Signing and Verification rather than Encryption and Decryption. Following this remark, both $M$ and $Sig$ shall be encrypted at the sender side using symmetric encryption with a shared key, or asymmetric encryption with the public key $k$ of the *receiver*. As described in the previous subsection, only the receiver then shall be able to $read$ and $verify$ the contents of $M$ and $Sig$, respectively.

### 2.3.3. Zero-Knowledge-Proofs (ZKPs)

A ZKP is a verification technique which, using cryptography, allows one substance to prove to another component that it knows a specific data or fulfills a particular requirement without disclosing any actual data that supports that evidence (Malyan and Madan, 2021). We deploy the ZKP technique in PriFoB with the goal of end-users being able to verify VCs without disclosing any private data within. PriFoB implies that each VC is coupled with a $Sig$ which is the encrypted hash of the issued VC $h(VC)$. Referring to the definitions presented in Bacelar Almeida et al. (2012), BC miners and end-users in PriFoB can be considered verifiers and provers, respectively. Following the notations described in Section 2.3.2, a prover sends the $h(VC)$ and the received $Sig$ accompanied with the VC, which could only be generated by the VC issuer. A verifier then only performs step 5 of Section 2.3.2. The ZKPs in PriFoB fulfill all the properties of a successful ZKP deployment as follows:

1. **Completeness**: an honest prover can always convince an honest verifier. In PriFoB, No system entity is able to generate a *correct* $Sig$ other than the original issuer of the VC because only the issuer knows the private key used for signing VCs.
2. **Proof of knowledge**: a malicious prover not knowing the secret cannot convince the verifier, except with negligible small probability. This is true in PriFoB as a malicious prover needs to know both $h(VC)$ and $g$ in order to generate a correct $Sig$, which is not the case according to the PriFoB protocol.
3. **Zero-knowledge**: an honest verifier that follows the protocol cannot learn additional information about the secret. According to the previously presented definition of hashing functions, any alteration within the input of $h(.)$ results in an unexpected output. Additionally, the input cannot be known from the hash string (hence, the name one-way-encryption). By allowing the verifier to read $h(VC)$, $h'(VC)$, $Sig$ and $k$ the verifier shall not be able to read/deduce any private data that belongs to the prover nor to the issuer.

More in-depth details on how the ZKPs work and their level of security and privacy can be found in Petkus (2019).

### 2.3.4. The distributed ledger

The BC as a PriFoB system element, including all its components, represents a DTTP for different types of agents to hold their public information and/or to securely validate issued VCs. On the other hand, many verifiers are required to perform tasks instead of a single central entity. Thus, it is recommended, in order to fulfill the system globalization feature, to have this DTTP designed as a BC. Additionally, those verifiers need to be granted equal provisioning and maintenance rights of the DL, as their roles in their territories are alike. As general criteria of a system that needs a BC include several equal participants, performing similar tasks, and maintaining a DL using an agreed-on CA, this description perfectly fits the scenario of the DTTP in PriFoB.

The DAG-based DL proposed in PriFoB has three dimensions, each dimension is used for a specific type of blocks. Simplified views of our proposed DL are depicted in Fig. 5. In Fig. 5(a), the order of confirmed blocks, present at a given time slot within the DL, is demonstrated with reference to the depth of each dimension up to the genesis block. It is noticeable here that each child block is pointing to specifically one parent while each parent block is allowed to have several children blocks. Additionally, several blocks at a given dimension can have similar index.

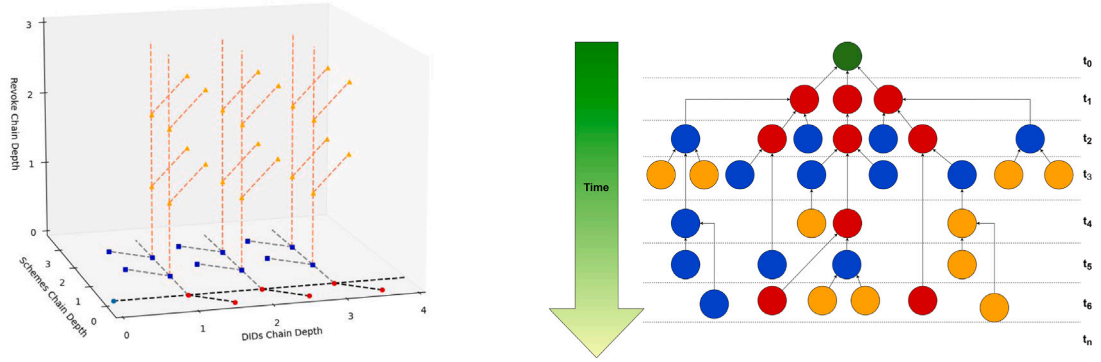
In Fig. 5(b), mature blocks (will be discussed later) appearing in the DTTP are demonstrated with reference to the time they appear. It is noticeable here that DID blocks can either point to the genesis block or to other DID blocks. Schema blocks can either point to the DID block of, specifically, their issuer, or to other schema blocks generated earlier by their issuer. Revoke blocks can either point to the schema block that was used to issue the revoked VC, or to any other revoke block that was generated earlier by the same issuer using the same schema block. It is also noticeable here that the time at which a mature block appears does not necessarily define the index of the block nor the position at which it is placed within the DL.

Although both demonstrations within Fig. 5 are captured from one miner's point of view, other honest miners within the DTTP will have exact similar views. The index and the previous signature of a mature block are decided by the miner who generated that block. In comparison with linear DL models, if a miner receives a valid block with a previous signature that is not of the, specifically, previous block, the new block is rejected. Additionally, the index of a received new valid block is determined by the receiver not the generator of the block.

According to the classification of DAG-based DLs, detailed in Wang et al. (2020), each dimension in our proposed DL is of Type-II DAG, where TXs need to be organized in blocks for packaging and the topology is a natural graph. However, our proposed DL model allows for parent blocks being pointed to by several child blocks, while each child block is allowed to point to only one parent block. This results in a tree-like DL, with several blocks potentially having similar index but unique identifiers (i.e. official name of issuer, type of schema, and hash of revoked VC, respectively). To efficiently allocate a block, an orthogonal parameter (*identifier*, *index*) needs to be provided. The identifier of every newly advertised block is checked and repetitions are not allowed. In a rare temporary case where an orthogonal parameter is similar for two different blocks within the same dimension, the longest-chain extension method (Shi, 2019) is used leaving a DL with no orphaned blocks and no repetitions. This concludes that our proposed DL provides a deterministic finality (Anceaume et al., 2020). The consensus mechanisms utilized to add new blocks will be detailed in later sections.

The DTTP maintains a 3DDL, where each dimension is a DAG structured. The first dimension holds confirmed DID blocks, each of those blocks includes:

1. an Header consisting the block type and miner signature,
2. an IMMUTABLE Body used in miner signature generation, consisting:
   - the DID TX sent by the issuer,



(a)

(b)

**Fig. 5.** A simplified view of our implemented DAG-based 3DDL (dashed links in (a) and arrows in (b) represent the usage of higher depth block of the signature of the linked lower depth block. Green node: Genesis Block, red nodes: DID blocks, blue nodes: schema blocks and orange nodes: revoke blocks).

- signatures of active miners (each indicates whether this issuer is accredited or not by the signer),
- and the signature in the Header of the previous block,

3. a MUTABLE independent chain of schemes (i.e. The second dimension), where future schema blocks issued by this specific issuer shall be saved.

Similar to the first dimension design, the second dimension holds a DAG of confirmed schema blocks each includes:

1. an IMMUTABLE Header,
2. an IMMUTABLE Body (with or without the accreditation signatures of all miners according to application specifications),
3. and a MUTABLE independent chain of revoked credentials (i.e. the third dimension), where future confirmed revoke blocks issued by the DID owner shall be saved.

The third dimension of the proposed 3DDL is dedicated to saving the hashes of revoked credentials. This would allow BC miners to check if the VC was revoked by the issuer, without actually reading any private data within the VC.

In all the three dimensions, the previous signature of a given block is not necessarily the signature in the header of the *last* confirmed block, yet the claimed previous signature must exist in one of the previously confirmed blocks for credibility. Our design allows for flexible data confirmation and non-linear chaining, leading to higher block confirmation rates, higher throughput rates, and lower response latency.

To this end, there is no need to implement a time synchronization method between miners (which typically appears in PoA-based BCs) as there is no restrictions on the order of confirmed blocks. Several blocks can simultaneously appear in the network, instead of a single block per time slot, without a consistency problem as they are all considered valid by all miners who confirmed their claimed previous blocks.

The immutability property of confirmed blocks is still preserved as changing a block, and all its consequent confirmed blocks for a successful attack, requires the attacker to know all private keys of all miners who mined the consequent blocks. The detailed security evaluation will be discussed later.

For obtaining a maximal efficiency reading from the 3DDL, a locally-saved **Sorted_Chain** object is implemented. Once a block is added to the chain, a tuple including its unique identifier and its on-chain index (probably not unique) is added to this object. This way, when a TX is later received, for which a miner needs to search for a block by its unique identifier instead of its index, the miner extracts its index from the **Sorted_Chain** and then reads the block directly on the chain. We implemented a binary search algorithm for this purpose to enhance the search efficiency in PriFoB to reach O(log n). Note that the blocks are ordered on-chain according to their time of confirmation while they are sorted alphabetically in the Sorted_Chain using their unique identifiers. Similar approach was utilized in Bandara et al. (2018) and found most efficient.

We also decided to set the block size to exactly one TX per block, as each TX requires its own group of signatures even if multiple TXs per block are utilized. As increasing block sizes lead to proportional decrease in block generation rate, due to higher verification latency (Baniata and Kertész, 2020), the overall throughput shall not improve with changing the block size as proven in Dinh et al. (2017).

### 2.4. Network layer modeling

In PriFoB, the BC network is a typical interconnected P2P network, with a randomly selected group of neighbors per peer. Each peer represents a national accreditation body, which is usually the focal point for accreditation in a given country. Peers can either be actual computers at the facility or cloud-based instances that can be accessed through the internet. In either cases, each peer shall connect to the PriFoB GW to get updated data regarding other active miners, the current state of the DL, and to receive and respond to the submitted TXs by agents and other miners.

The GW in PriFoB is a load-balancing and a monitoring element for the DTTP network in general. The GW keeps track on active miners, accept new mining requests and broadcasts miners' identification information (including their public keys) into the BC network. Shared data among neighbors are sent directly, while responses from miners to agents are sent through the GW.

To realize the connections between different PriFoB components, we utilize the open Secure Sockets Layer (OpenSSL) framework. Each entity dedicates two threads, namely Server and Client, to asynchronously receive and send messages, respectively. These threads deploy the Transport Layer Security (TLS) Protocol (Dierks and Rescorla, 2008) with a static port number (default is 5050) for sending and receiving.

The communications between system entities are ruled by a strict protocol for the sake of achieving the Privacy-by-Design method. All communications with the GW, and between Miners, are performed in plain text because none of these communications include private data of any system entity. All exchanged messages are JSON strings for facilitating future deployment of heterogeneous system entities.

Agents can communicate directly with each other or with the GW. Only communications between agents, described as step no. 4 in Fig. 1, must be encrypted. Thus, only the requested agent can read the VC request. Public keys of issuers are requested by end-users from the DTTP in plain text. An issuer behaves as a fog node because fog nodes are, fundamentally, entities that communicate directly with end-users/edge devices, and perform computations instead of a classical centralized server (i.e. mostly a cloud server).

**Table 1**
Computational complexities required to generate new blocks referring to different types of requests, and the expected appearance rates of different types of TXs throughout the life-cycle of an accredited institution.

| Request type | Appearance | Computational complexity (wrt. no. Miners) |
|---|---|---|
| DID TX | Few | (n+1) signing + (2n−1) verification |
| Schema TX | Moderate | 1 signing + n verification |
| VC validation | Most | 1 verification |
| Revoke TX | Rare | 1 signing + (2n − 1) verification |

### 2.5. Consensus modeling

In PriFoB, we utilize two types of CAs for two different layers of consensus regarding new blocks, namely PoA and SoW. By default, we assume that a successful verification of a new schema TX or a new revoke TX means a successful verification of the TX issuer (among other things to validate). Thus, those two types of TXs are processed by only one miner leading to mining a new block using the signature of this miner (i.e. PoA). For DID TXs, we decided to propagate the TX throughout the BC network asking each miner to declare whether the DID requester is accredited or not in its country. The decision can be performed both manually by the miner admin, or automatically referring to a local/remote DB (both approaches are implemented).

Once a miner's declaration is ready, the miner's signature using its private key is added to the declaration and the signed declaration is added to the TX. The TX is propagated throughout the network, in the state of unready/immature TX, until the following strict conditions are met. When a miner receives a DID TX that meets all these conditions, the TX is considered ready or mature to be mined:

1. All active miners (periodically pinged by the GW, e.g. every 5 s) have signed the TX
2. Recipient miner has already signed the TX
3. All signatures are correct
4. TX is not found in any previously confirmed DID block (i.e. the claimed DID is unique)

Unlike classical PoA CA, no restrictions are enforced in PriFoB regarding the number of blocks that can be generated within a single time slot. The miner who finds a mature DID TX mines and broadcasts it immediately.

All miners are authorized to mine new blocks, according to the data layer definitions, in any time slot by signing the block body, and adding the signature to the Header of this new block. Note that the accumulated and complete group of correct miners' signatures (i.e. the SoW) represents a trigger for the last miner to start the mining process. In other words, the trigger to mine a new DID block in PriFoB is pulled by the network (instead of a single leader node in typical PoA-based BCs). Consequently, the miner adds a PoA to the new block and propagates it throughout the network. The PoA is then verified, as well as the block body, by all recipient miners, and is added upon successful validation and verification.

Table 1 presents the computational complexities expected, for each type of TXs, from the time it is delivered to the GW, until it is confirmed and responded to. Note that the expected relative appearance of different TX types can be deduced logically. That is, each issuer is allowed only one unique DID, while it is expected to issue unlimited number of VCs referring to a limited number of schemes. Additionally, it is relatively rather rare that an institution would revoke a VC it issued. This being said, an expiration data of a VC can be injected within by the issuer, and consequently the accompanying signature, so that it can be checked once the VC is found valid. A similar approach can be used for DIDs and schemes as well.

Compared to different consensus complexities presented in Eichhorn et al. (2021), PriFoB performs optimally for schema and revoke TXs as both require O(1) computational complexity by the system (i.e. signing). For DID blocks, PriFoB requires O(n) computational complexity by the system, which is better than the PBFT and RBFT CAs with complexities O($n^2$) and O($n^3$), respectively.

It can be observed from the values presented in the table that the verification is used much more for all types of TXs than signing. It can also be observed that the most appearing type of TXs is the VC validation request which requires no signing by any BC entity as it is not saved on-chain. This, in fact, was our motivation to adopt the RSA encryption methods. For more details, we present our thorough comparison between the mostly used ECC and our adopted RSA encryption methods in Section 4.

### 2.6. PriFoB applications

In an abstract description of the ZKP-based verification protocol of PriFoB, a prover can be either a client of an issuer or the issuer entity itself. An issuer entity can then issue a valid VC for itself. However, this should not be a trust problem since the application of PriFoB does not give a meaning of such behavior. For example, consider a hospital that issues vaccination certificates to patients, or an educational institution that issues degree certificates to graduate students. Both of these entities do not need to be certified as vaccinated or graduate, respectively. On the other hand, an issuer shall be able to validate a VC before/after it is issued, in order to deliver guaranteed services to clients. Thus, issuers are able to perform ZKP-based verification just as their clients.

Note as well that issuers can be clients to each other, which is indeed a realistic scenario. That is, organizations (e.g. hospitals, education institutions, etc.) shall be able to be granted certificates from other, higher-level organizations (e.g. international committee of specialized hospitals, international corporation of education quality, etc.).

PriFoB schema blocks are much flexible for any type of VCs. Most previous works have targeted a specific type of credentials and only allowed specific, poorly reconfigurable attributes per credential (if any). In our work, however, we provide three levels of credential definition, namely DIDs, Schemas, and VCs. With issuers being able to define any type of VC by simply publishing a new schema, VCs can cover a wide variety of credential applications.

Issuer accreditation is also optional at the time of deployment, and if it does not apply for the business model, then it can be simply deactivated by the miners. Accordingly, any received DID TX will be verified only depending on the correctness of the signatures within. In addition to the educational diploma and the global vaccination certificates examples, PriFoB can be used for realizing other applications such as a BC-based smart parking solution (Al Amiri et al., 2020). Depending on the business model, we can envision a scenario where several international companies need to co-operate their parking lots that are distributed in different countries. Each parking lot is maintained by a local management team and needs to digitally verify cars entering their areas. The international companies then can be connected to perform a DTTP, which accepts DID TXs from the parking lots. Each miner in the DTTP can individually accredit (or not) a parking lot according to the agreement between the companies.

Once a parking lot has a confirmed DID, it can issue its own different schemes relating to different sections/floors or different types of vehicles (e.g. small cars, trucks, etc.). Once a schema is confirmed, each vehicle is granted a VC certifying that it is allowed to enter and park at the lot which, specifically, owns this DID, in which a specific section is allowed to be utilized. Such a VC can be validated by security checkers at the gates of the parking lots and/or the sections to be entered. The security checker might be a human factor with a simple end-user account or an automated verification machine using, e.g. a QR-code scanner.

The mentioned scenario is one of many several applications where PriFoB could be used for global co-operation. PriFoB, in short, provides the highly required flexibility for distributing a global server, while maintaining a unified interface for end-users at different locations, regions and countries. Other examples, including (inter)national e-Voting, e-Health, IoV applications, can be easily realized using our open-source PriFoB solution.

It is also beneficial to highlight the award policy in PriFoB, as all system entities do perform computational and storage tasks throughout the life cycle of a VC. As a start, miners are not awarded for newly mined blocks, but rather granted a fare share of revenues according to their provided computational and storage capacities. Every TX submitted to the BC can be accompanied with a proof of pre-payment, either using cryptocurrency or bank-based accounts. The amount of valid pre-payment can be adjusted during system setup according to the application business model (e.g. type of TX, expected cost of a block confirmation, etc.). A verified TX can then be checked by the recipient miner for valid pre-payment TX on another platform. End-users can use the services of PriFoB (i.e. the PriFoB wallet interface) freely as revenue can be obtained by ads on the application UI. However, issuers can individually enforce a prepayment scheme for issuing a VC depending on their own business model as well.

For example, a valid DID TX can be defined to cost \$10, a valid schema TX can be defined to cost \$5 and a valid revoke TX can be defined to cost \$50. Accordingly, if 10k issuers are expected to register to the PriFoB-based system, with an average of 20 schemes per issuer, then an initial revenue of \$100k is expected for DID blocks plus \$1 m is expected for schema blocks. If each schema is expected to be used for issuing 100 VCs per year, then the number of yearly active end-user wallets can be estimated by 100 users × 20 schemes × 10k issuers = 20 million active wallets per year. Assuming that only 1% of those users open their wallets per day (i.e. 200k users), and an average revenue rate of \$11.5 per 1000 views (Dogtiev, 2021) as of 2021, the expected daily revenue can be estimated then by \$2300 (i.e. \$839,500 per year). Assuming the DTTP is composed of 300 miners, each is providing equal computational and storage capacities, each of those miners can be granted ≈ \$2798 per year. As a basic 24/7 virtual machine at a public cloud provider platform (e.g. E-2 instance at the Google Cloud Platform) may cost ≈ \$350 per year (Google corp., 2021), then each miner is left with a yearly revenue of more than \$2440. Note that in the case where the DTTP miners have agreements/collaboration with issuers, such as described in parking lots case above, the revenue for issuing the 20 million VCs can also be divided amongst the collaboration entities. That is, assuming that a VC is issued for \$10 with an annual expiration date, then a total yearly revenue of \$200 million is expected. Similar revenue calculations, can be conducted with different TX predefined costs for different applications.

## 3. Evaluation

In this section, we analyze the security in the Data Layer and the Consensus Layer of PriFoB. The security discussion of the remaining layers (i.e. infrastructure, network, and application layers) are beyond the scope of this work. We also discuss the privacy in PriFoB in its generality referring to the GDPR rules and the detailed description of requests and responses among system entities. After that, we compare real measures of PriFoB deployed in the cloud against well documented implementations of Ethereum and different Hyperledger platforms, including Indy, Besu and Fabric. Finally, we parameterize a Discrete Event Simulation environment, using real data we obtained, to predict the throughput and power utilization of PriFoB in different settings.

### *3.1. Security*

Distributed systems are usually evaluated in terms of security referring to several concepts. The **Strong Validity** requirement implies that if all honest nodes propose the same value $v$, then no honest node decides a value different from $v$. The value $v$ here corresponds to the state of the DL. In PriFoB, this property is guaranteed as all honest nodes that are connected to the network will eventually receive and accept a new block, leading to its addition to their local ledgers. Thus, all honest nodes will eventually agree on the contents of the DL.

The **Agreement** requirement implies that no two honest nodes decide differently. This is also referred to as the consistency of the DL (Kertesz and Baniata, 2021). This property is usually the hardest to achieve in BC-based systems due to the need of agreeing not only on valid blocks, but also on the order of those blocks. PoW-based systems solve this by hardening the puzzle difficulty so that only one block is generated every predefined time (e.g. 10 min in Bitcoin), giving sufficient time to new blocks to propagate throughout the network. However, PriFoB does not suffer in this regard as the order maintenance requirement is relaxed because we utilize our previously described DAG-based 3DDL. This being said, PriFoB is not suitable for applications that require strict chronological order of blocks, such as digital currency applications, but is suitable for credential verification frameworks as described earlier.

The **Termination** and the **Integrity** requirements imply that every honest node eventually decides some value and that no honest node decides twice, respectively. Both of these properties are true for PriFoB as we fine-tuned all system elements to either accept or reject new TXs/blocks, according to strictly defined conditions. Furthermore, if an immature TX is already signed by a miner then this miner does not sign it again.

Originally, a generic SoW scheme for consensus in distributed systems was described in Garay et al. (2020). The generic scheme was proven correct, secure against Tampering and Chosen-Message Attacks, and t-Verifiable (the verifier is able to verify a SoW in t steps, where t is a lot smaller than the time needed to produce a signature). The generic SoW scheme has been shown feasible for deployment in even less trusted environment, namely permissionless BCs, where miners are allowed to join and leave the network without restrictions.

Specifically, for $n$ miners collaborating to produce a group of signatures that represents a proof of correctness for a given block, $h$ is the hardness level of signature generation, $H$ is a collision resistant hash function, $\lambda$ is a security parameter and $T$ is the number of failed/adversary nodes ($T < \frac{n}{2}$), the lower bound on the rate of generating uniquely successful blocks by an adversary, denoted $\gamma$, for such system can be calculated using Eq. (1).

$$\gamma_{DID} = (n - T).(1 - (\frac{h}{2^{\lambda}})^{T'_H}).(\frac{h}{2^{\lambda}})^{(n-1)T_H} \tag{1}$$

For example, a SoW-based system, such as ours, that consists of $n = 6$ nodes, with $h = 50$ milliseconds, $H$ is SHA-256, $\lambda = 1024$ bits and $T = 2$, generates each DID block with its unique SoW with $\gamma = 6.66 \times 10^{-1533}$. Note how small this value is, indicating nearly Zero probability of an adversary miner to be able to change any DID block saved on the DL. To clarify, differential cryptanalysis cannot be used by a dishonest miner to deduce other miners' private keys and generate forged blocks. Since $\gamma_{DID}$ is directly proportional to $n$, adding more miners to the DTTP in PriFoB shall enhance the security of the first dimension in our proposed 3DDL.

For other types of TXs, i.e. schema and revoke TXs, that consist of one signature proving its correctness, the probability of forging is determined by $\lambda$, $h$ and $T$. We can deduce Eq. (2) from Eq. (1) to assess the probability of a successful schema or revoke block forging. Note that, in this case, adding more miners does not affect the security level of the second and third dimensions of our proposed 3DDL. The simplest way to describe it is to assume that the longer the key (in bits) the stronger it is. More details can be found in several references such as Mahto and Yadav (2017) and Garay et al. (2020).

$$\gamma_{scheme,revoke} = (1 - (\frac{h}{2^{\lambda}})^{T'_H}).(\frac{h}{2^{\lambda}})^{T_H} \tag{2}$$

We further assess PriFoB using the Common Vulnerability Scoring System (CVSS v3.1) (Mell et al., 2006). The CVSS v1.0 was originally implemented and proposed in 2006 and had been continuously developed to v2.0 in 2007, v3.0 in 2015 and finally v3.1 in 2019 (Nowak et al., 2021). CVSS is a tool used to quantify the fuzzy security of a given system into a numeric score. The CVSS scores given for a system range between 0.0 to 10.0. The lower the score of a given system, the more secure it is against security attacks. To obtain an accurate security score of a system, the CVSS tool needs to be parameterized according to the system design and the probable attacks characteristics.

To obtain a security score for our proposed PriFoB, we parameterized the CVSS tool as follows:

- Attack Vector: Local (attacker must be connected to the DTTP network)
- Attack Complexity: High (attacker needs high computational power to forge a block)
- Privileges Required: High (attacker must be accepted as a miner by the GW)
- User Interaction: Required (other miners must accept the attacker as a legitimate miner and start using its public key distributed by the GW)
- Scope: Changed (forged blocks with correct signatures leads to changing the state of the DL)
- Confidentiality: None (a forged block does not imply a confidentiality vulnerability)
- Integrity: Low (a forged block does imply an integrity vulnerability but only for one customer not for the whole system)
- Availability: None (a forged block does not imply an availability vulnerability)

The above parameterization resulted in a rather low CVSS **Base Score** of 2.3/10.0. Hardening the assessment, PriFoB received another low **Environmental Score** of 3.9/10.0 as follows:

- Modified Attack Vector: Network (attacker can join the DTTP network from anywhere)
- Modified Privileges Required: low

Regarding the security of our proposed 3DDL, such model implies higher efficiency and throughput compared to linear models as proven in several previous works. However, the drawback of utilizing it is demonstrated by higher probability of launching a selfish mining breach leading to double-spend attacks (Begum et al., 2020) (e.g. payment and smart contract solutions such as Bitcoin and Ethereum). Such type of attacks indeed makes sense for applications that require a chronological order of confirmed TXs/blocks. However, those attacks do not make sense within the framework of PriFoB (De Souza et al., 2019). That is, the simultaneous confirmation of multiple blocks/TXs, leading to different order of blocks at different physical locations of the DL (termed forks), does not imply that one of the blocks in the tie/fork is incorrect or misleading. Each of the blocks/TXs in a tie of a fork shall represent a different issuer with a different unique identifier although they may have similar block indices. To solve a probable mistaken response for a given block index that is common for multiple blocks, a requested miner searches for all the blocks with the given index and requires an equal value of the unique identifier in the request to the value of the unique identifier in the block.

### 3.2. Privacy

As PriFoB complies with the W3C standard and further does not allow for any private date to be saved on the DL, PriFoB can be trivially considered a GDPR-compliant system. We have described earlier how VCs are only saved locally at the end-user layer and are never sent to the DTTP. Additionally, the deployed encryption schemes, the ZKPs with collision resistant hash functions, along with the utilization of DSs, all lead to a Privacy-by-Design implementation.

Nevertheless, we referred to the GDPR checklist for data controllers available at Proton Technologies AG (2021). In our paper, we have conducted an information audit to determine what information PriFoB processes and who has access to it. Thus, we provided clear information about PriFoB data processing and legal justification in its privacy policy (PriFoB-based solutions will have a legal justification for its data processing activities as end-users voluntarily sign up into the system). Encryption, pseudonymization, and anonymization of personal data wherever possible is performed in PriFoB. Additionally, it is easy for PriFoB customers to request and receive all the information it has about them. That is, the whole BC is directly downloadable via the application interface. Finally, different types of TXs can be revoked and regenerated, while private data is only saved locally, making the private data controllable only by its owners (there is no need to request private data deletion).

### 3.3. Latency

Searching recently proposed solutions, we found the following works that are similar to PriFoB in terms of objectives and BC-deployment. Baniata et al. (2022b) analyzed the latency of BC-based solutions utilizing Hyperledger Indy. Urbančok (2019) described and compared four open-source BC platforms, namely Ethereum, Hyperledger's Fabric, Besu, and Iroha, in different terms including latency. Xu et al. (2021) analyzed the latency of BC-based solutions utilizing Hyperledger Fabric. Zhang et al. (2019) experimentally evaluated the performance of Ethereum testnets in terms of Account balance query latency, Block generation time and End-to-end TX acceptance latency. Härer and Fill (2019) utilized the main net of Ethereum platform and proposed a credential verification solution, on which they performed latency assessment. Bampatsikos et al. (2019) proposed a solution for mitigating probable Computational Denial of Service (CDoS) attacks when utilizing Ethereum for credential verification purposes, and provided latency assessment for their solution. We clarify the differences and similarities, along with brief comparison of the results, for those works and PriFoB in Table 2.

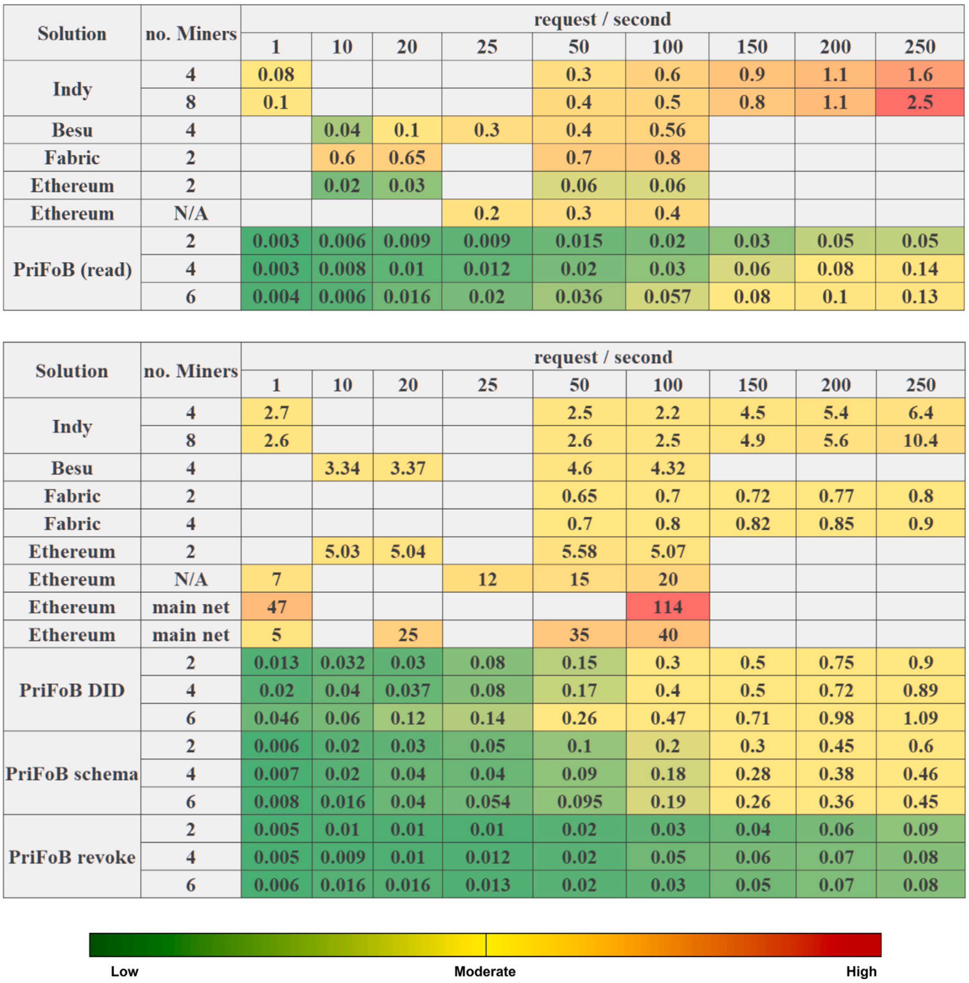
Some of these works evaluated their solutions using simulation (all miner nodes run on a single machine), while others evaluated their solutions using real test-beds (each miner is allocated a different machine). We compare the latency of PriFoB, however, with all of them, to prove PriFoB outperformance and provide insights regarding expected latency with different scalability measures. Additionally, we run several test scenarios in order to comprehensively compare PriFoB with all of them. The results of the scenarios we tested, along with each scenario parameters, are detailed in Fig. 6. To facilitate reading and comparing the performance results, we color-coded the tables' cells according to the scale provided within the figure.

We evaluated the performance of PriFoB using a Proof of Concept (PoC) system composed of a single GW, a network of miners with different sizes (2, 4 and 6 miners), and a script we implemented, that emulates several issuers and agents simultaneously communicating with the DTTP. We deploy each of those entities on a separate VM at the Google Cloud Platform, each is of type E2-medium (2 Intel(R) Xeon(R) vCPUs clocked at 2.30 GHz with 10 GB of RAM), running Linux 20.10 OS. Alternatively, issuer and agent implementations are available and tested within the project repository but they cannot be used to stress-test the DTTP in PriFoB.

According to the aforementioned description of PriFoB, the GW component is the bottleneck of a PriFoB-based system. Thus, we tested PriFoB using one GW to obtain the worst results possible. The bottleneck effect can trivially justify the observable proportional relation, in Fig. 6, between the number of miners and the average latency. However, it will be shown in the following subsection that the Read latency in PriFoB should not be affected by the number of miners when deploying a computationally powerful GW. As a result, our containerized implementation of PriFoB elements, including the GW, shall show better measurements than those presented here, if more powerful machines and/or more GWs were deployed using e.g. Kubernetes orchestrator. It is worth noting here that despite the apparent bottleneck effect in our experiments, PriFoB still outperforms all the compared related systems.

**Table 2**
PriFoB comparison with previous related works, that proposed solutions for distributed credential verification systems, in terms of utilized Blockchain platform, granting institution accreditation services, number of Miners (M), assisted request type (T), lower and upper bounds of request per second rates (req/s) and the lower and upper bounds of response Latency measured as second per request (s/req).

| Solution | CA | w/Accreditation? | M | T | req/s | Latency (s/req) |
|---|---|---|---|---|---|---|
| Hyperledger Indy (Baniata et al., 2022b) | Plenum (PBFT) | NO | 4 | DID/Schema (write) | 1–250 | 2–6 |
| | | | | Any (read) | | 0.08–1.6 |
| | | | 8 | DID/Schema (write) | | 2–10 |
| | | | | Any (read) | | 0.1–2.5 |
| Hyperledger Besu (Urbančok, 2019) | PoA | NO | 4 | Any (write) | 10–100 | 3.34–4.60 |
| | | | | Any (read) | | 0.04–0,56 |
| Hyperledger Fabric (Xu et al., 2021) | RAFT | NO | 2 | Any (wirte) | 50–250 | 0.6–0.8 |
| | | | 4 | Any (write) | | 0.7–0.95 |
| Hyperledger Fabric (Urbančok, 2019) | | NO | 2 | Any (read) | 10–100 | 0.6–0.8 |
| Ethereum (Urbančok, 2019) | PoW | NO | 2 | Any (write) | 10–100 | 5.03–5.58 |
| | | | 2 | Any (read) | | 0.02–0.06 |
| Ethereum (Zhang et al., 2019) | PoA | NO | N/A | Any (write) | 25–100 | 5–34 |
| | | | N/A | Any (read) | | 0.2–0.4 |
| Ethereum (Härer and Fill, 2019) | PoW | YES | Main Net | DID (write) | 1–100 | 47–114 |
| Ethereum (Bampatsikos et al., 2019) | PoW | YES | Main Net | DID (write) | N/A | 5–40 |
| **PriFoB** | SoW+PoA | YES | **2–6** | **DID (write)** | **1–250** | **0.013–1.09** |
| | | | | **Schema (write)** | | **0.006–0.6** |
| | | | | **Revoke (write)** | | **0.005–0.09** |
| | | | | **Any (read)** | | **0.003–0.14** |



| Solution | no. Miners | request / second | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 10 | 20 | 25 | 50 | 100 | 150 | 200 | 250 |
| Indy | 4 | 0.08 | | | | 0.3 | 0.6 | 0.9 | 1.1 | 1.6 |
| | 8 | 0.1 | | | | 0.4 | 0.5 | 0.8 | 1.1 | 2.5 |
| Besu | 4 | | 0.04 | 0.1 | 0.3 | 0.4 | 0.56 | | | |
| Fabric | 2 | | 0.6 | 0.65 | | 0.7 | 0.8 | | | |
| Ethereum | 2 | | 0.02 | 0.03 | | 0.06 | 0.06 | | | |
| Ethereum | N/A | | | | 0.2 | 0.3 | 0.4 | | | |
| PriFoB (read) | 2 | 0.003 | 0.006 | 0.009 | 0.009 | 0.015 | 0.02 | 0.03 | 0.05 | 0.05 |
| | 4 | 0.003 | 0.008 | 0.01 | 0.012 | 0.02 | 0.03 | 0.06 | 0.08 | 0.14 |
| | 6 | 0.004 | 0.006 | 0.016 | 0.02 | 0.036 | 0.057 | 0.08 | 0.1 | 0.13 |

| Solution | no. Miners | request / second | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 10 | 20 | 25 | 50 | 100 | 150 | 200 | 250 |
| Indy | 4 | 2.7 | | | | 2.5 | 2.2 | 4.5 | 5.4 | 6.4 |
| | 8 | 2.6 | | | | 2.6 | 2.5 | 4.9 | 5.6 | 10.4 |
| Besu | 4 | | 3.34 | 3.37 | | 4.6 | 4.32 | | | |
| Fabric | 2 | | | | | 0.65 | 0.7 | 0.72 | 0.77 | 0.8 |
| Fabric | 4 | | | | | 0.7 | 0.8 | 0.82 | 0.85 | 0.9 |
| Ethereum | 2 | | 5.03 | 5.04 | | 5.58 | 5.07 | | | |
| Ethereum | N/A | 7 | | | 12 | 15 | 20 | | | |
| Ethereum | main net | 47 | | | | | 114 | | | |
| Ethereum | main net | 5 | | 25 | | 35 | 40 | | | |
| PriFoB DID | 2 | 0.013 | 0.032 | 0.03 | 0.08 | 0.15 | 0.3 | 0.5 | 0.75 | 0.9 |
| | 4 | 0.02 | 0.04 | 0.037 | 0.08 | 0.17 | 0.4 | 0.5 | 0.72 | 0.89 |
| | 6 | 0.046 | 0.06 | 0.12 | 0.14 | 0.26 | 0.47 | 0.71 | 0.98 | 1.09 |
| PriFoB schema | 2 | 0.006 | 0.02 | 0.03 | 0.05 | 0.1 | 0.2 | 0.3 | 0.45 | 0.6 |
| | 4 | 0.007 | 0.02 | 0.04 | 0.04 | 0.09 | 0.18 | 0.28 | 0.38 | 0.46 |
| | 6 | 0.008 | 0.016 | 0.04 | 0.054 | 0.095 | 0.19 | 0.26 | 0.36 | 0.45 |
| PriFoB revoke | 2 | 0.005 | 0.01 | 0.01 | 0.01 | 0.02 | 0.03 | 0.04 | 0.06 | 0.09 |
| | 4 | 0.005 | 0.009 | 0.01 | 0.012 | 0.02 | 0.05 | 0.06 | 0.07 | 0.08 |
| | 6 | 0.006 | 0.016 | 0.016 | 0.013 | 0.02 | 0.03 | 0.05 | 0.07 | 0.08 |

**Fig. 6.** Color-coded average response latency for PriFoB verification requests per second (upper table), and average response latency for PriFoB DID, Schema and Revoke write requests per second (lower table) against average read and write latency measurements, reported in the literature, for major Blockchain solutions utilized for objectives similar to PriFoB.

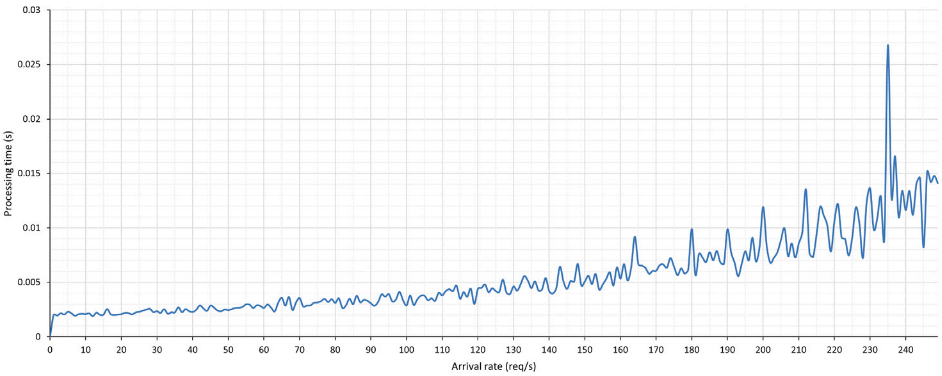


**Fig. 7.** Processing time per READ request in PriFoB for $\lambda \in [1, 250]$, $n = 10$ and $\mu = 335$.

### 3.4. Throughput

Next, we mathematically model our system to predict its general behavior while the rate of requests per second ($\lambda$) and the number of miners ($n$) increase. To do so, we refer to the Queuing Theory (Li et al., 2018) and characterize PriFoB using the Kendall's notation as ($\lambda$ = M/D = M/$n$):(FCFS/∞/∞), where M, D and FCFS stands for Poisson distribution, output distribution and the First-Come-First-Served discipline, respectively. Utilizing such approach to model BC-based systems has been predominantly used in the literature (Smetanin et al., 2020). To compute the expected average processing time ($W_s$) in PriFoB, we use Eq. (3):

$$W_s = \frac{\mu(\lambda/\mu)^n}{(n-1)!(n\mu-\lambda)^2} P_0 + \frac{1}{\mu} \tag{3}$$

where $\mu$ and $P_0$ are the mean service rate per busy server (request per second i.e. $\mu$ = 1/Latency) and the probability that there are Zero requests in the system, respectively. For all $n \times \mu > \lambda$, we compute $P_0$ using Eq. (4):

$$P_0 = \frac{1}{[\sum_{i=0}^{n-1} \frac{1}{i!}(\frac{\lambda}{\mu})^i] + \frac{1}{n!}(\frac{\lambda}{\mu})^n \frac{n\mu}{n\mu-\lambda}} \tag{4}$$

All the following experiments were run using the Discrete Event Simulation tool provided at Gonzalez (2020), with suitable modifications and tuning referring to our model and real parameterization. The tool provides analysis for a single queuing case per run so we injected several FOR loops as needed to go through all the cases in each scenario we attempted to test. Using our mathematical model of PriFoB, we measured $W_s$ for $\lambda \in [1, 250]$ with a static $n = 10$ and using $\mu = 335$ referring to the real READ results we detailed previously. Accordingly, we could simulate the general effect of increasing $\lambda$ and compare it with our real data to validate the simulation tool. The results we obtained for the first scenario are depicted in Fig. 7. As can be noticed, the simulation results comply with the real measures we obtained as increasing $\lambda$ results in linear increment of $W_s$. However, average $W_s$ is less than the real data presented earlier, as expected, due to the deployment of more mining nodes. This indeed shall increase the overall throughput of the system as more servers are able to process an equivalent $\lambda$.

To capture the general effect on $W_s$ when increasing $n$, we tested $\lambda = 10$, $n \in$ [1–100] and using $\mu = 335$. The results we obtained are depicted in Fig. 8. Here, the simulation assumes that all incoming requests are immediately distributed among available miners (i.e. no bottleneck effect). Accordingly, the latency is nearly constant between 3–5 millisecond per request. This proves that the proportional relation between $n$ and the Read latency is mainly attributed to the limited computational power of the GW. Uncontrollable communication delays and/or unpredicted hit ratios when searching the DL can also cause such effect. For all of these reasons, we used a static $\mu$ in both scenarios we simulated so far, as $\mu$ is not related to $n$ for all types of TXs, except for DID TXs.

For DID TXs, however, $\mu$ is affected by changing $n$ as more miners in the system implies more signature generations. Consequently, increasing $n$ should, theoretically, increase $W_s$ for DID TXs. For this reason, we could state that the results depicted in Fig. 8 do not represent the expected behavior of PriFoB for DID TXs.

To capture the general effect of increasing $n$ on $W_s$ for DID TXs, we estimate the values of $\mu$ for increased $n$ using the latency observations we obtained in the previous subsection. Let $X_{i,l}$ be the real DID TX latency for $i \in Y$ = [2, 4, 6], $l \in Z$ = [1–250]. We can then calculate the increase in latency per each added miner, denoted by $\gamma_{i,j,l}$, using Eq. (5):

$$\gamma_{i,j,l} = \frac{X_{i,l} - X_{j,l}}{j-i} \ \forall\, i, j \in Y,\ l \in Z \ and\ i < j \tag{5}$$

Using data in the set $\Gamma$, which consists of all $\gamma$ values we obtained, we calculated $\gamma_{avg} = 0.03$ second per request per added miner and we calculated the Maximum Likelihood Estimation (MLE) $\gamma_{MLE} = 0.01$ second per request per added miner. The MLE was obtained from a Poisson Distribution obtained from the set $\Gamma$. Note that MLE is generally more accurate than the average (Severini, 2000). We injected both $\gamma_{avg}$ and $\gamma_{MLE}$ into Eq. (3) resulting in Eq. (6):

$$W_{s_{DID}} = \frac{\mu(\lambda/\mu)^n}{(n-1)!(n\mu-\lambda)^2} P_0 + \frac{1}{\mu} + (n \times \gamma) \tag{6}$$

Using Eq. (6) to predict $W_{s_{DID}}$ values for different $n$ values is now realistic using both of $\gamma$ values. We tested $\lambda = 10$ DID TXs, $n \in [1, 100]$. The results we obtained are depicted in Fig. 9.

### 3.5. Power utilization

Finally, deploying more miners may indeed increase the latency for DIDs, a type of TXs that is used less frequently than Schema TXs, and further appears even more rarely compared to VC validation requests where PriFoB system flourish. However, we need to further evaluate PriFoB in terms of power utilization. To achieve this, we injected several variables into our mathematical model implementation. We could then assess the percentage of system power utilization for known $\lambda$ and $n$. We tested for $\lambda = 250$, $n \in [1, 100]$, a dynamic $\mu$ for DID TXs with $\gamma_{MLE} = 0.01$, and a static $\mu = 335$ for all other types of TXs. Fig. 10 describes the percentage of computational power consumption out of the total available computational power, to process all received requests. For all types of TXs other than DID TXs, it is clear from the results we obtained that adding more miners to the system shall enhance the overall efficiency in terms of computational power consumption per request. However, increasing $n$ implies higher power consumption rates due to the required SoW computational complexity. Note that if the optional accreditation service in PriFoB is deactivated, DID TXs shall consume as much energy as any other type of TXs. Also note that the

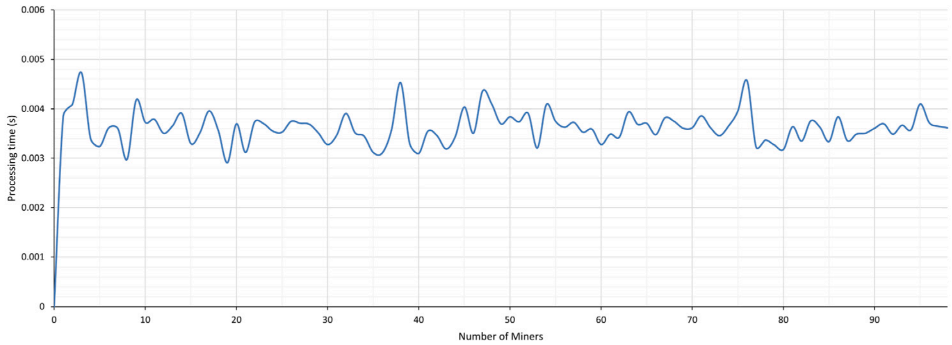


**Fig. 8.** Processing time per READ request in PriFoB for $\lambda = 10$, $n \in [1, 100]$ and $\mu = 335$.

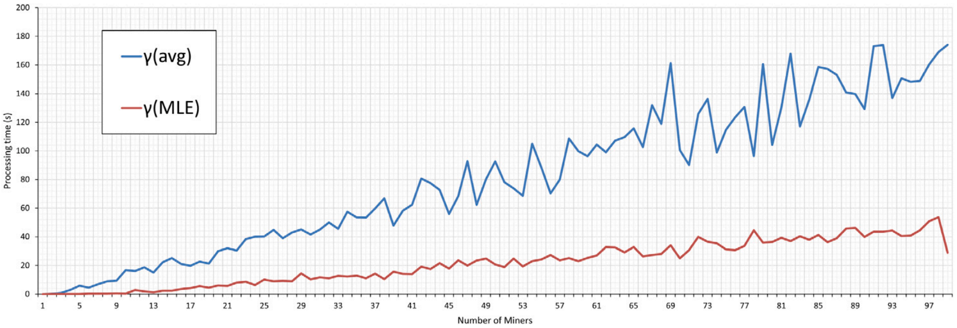


**Fig. 9.** Processing time per DID request in PriFoB for $\lambda = 10$ DID TXs, $n \in [1, 100]$, $\gamma_{avg} = 0.03$ and $\gamma_{MLE} = 0.01$.

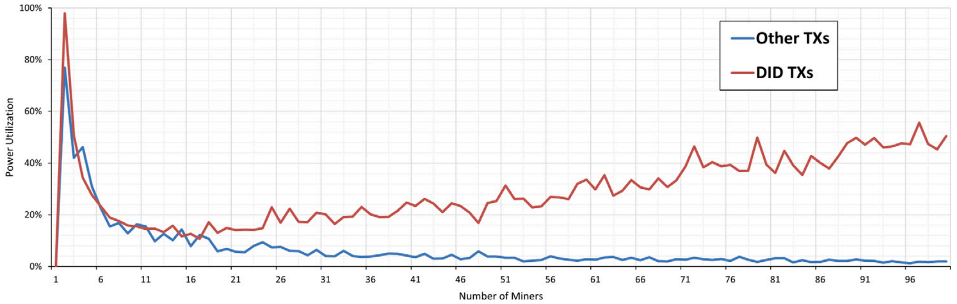


**Fig. 10.** Power utilization with $\lambda = 250$, $n \in [1, 100]$ for DID TXs ($\gamma_{MLE} = 0.01$) and all other types of TXs ($\mu = 335$).

power utilization here is inversely proportional with $\mu$ which is rather low in our experiments, due to the low computational capacity of our test-bed infrastructure.

## 4. Related work

Several previous works approached BC-based solutions for realizing digital credential verification. Table 3 summarizes most relevant solutions we found in the literature, where we highlight main properties of PriFoB against those solutions. As provided in the table, the only previous work that addressed both the globalization and online issuer accreditation properties was only discussed theoretically. Furthermore, most works saved VCs on-chain which resulted in non-GDPR compliant solutions. Other technical details, such as storage optimization and DL modeling were not considered thoroughly as well. All of these works adopted linear DL models (instead of DAG-based model in PriFoB). Works that used the PoW and the PoA CAs were realized using an Ethereum-based BC network, while those who used the Raft CA were realized using a Hyperledger Fabric BC network.

Fauteux et al. (2020) stated how BC deployment can solve the long list of challenges when an academic credential needs to be validated. The BC solution was discussed along its pros and challenges, and some technical details were further presented. Only in the years of 2019 and 2020, some few testable implementations were proposed to address credential verification problem using BC technology. Recently launched BC-based credential verification projects, namely BlockCerts (BlockCerts, 2021), OpenCerts (OpenCerts, 2021), and trustED (Batzavalis et al., 2021) were surveyed in Bhumichitr and Channarukul (2020), where the most mature and suitable consensus methods, BC architectures, and BC platforms were presented and discussed. Consequently, the authors proposed AcaChain, which is a private, permissioned BC system that allows issuers to track the achievements of their agents, and then issue the graduation proof once all conditions of the credential are fulfilled. As the three BC-based credential systems, namely AcaChain, BlockCerts and OpenCerts, save all credentials information, along with relevant personal identifiers of students on the immutable chain, they are non GDPR-compliant. Furthermore, saving all data on the chain is considered an inefficient approach of using a BC system, as this can

**Table 3**
Summary comparison of various BC-based solutions similar to PriFoB.

| Solution | Consensus algorithm | Implemented? | GDPR compliant? | Storage optimization? | Globalization | Issuer accreditation? |
|---|---|---|---|---|---|---|
| BlockCerts (BlockCerts, 2021) | PoW | ✔ | ✘ | ✘ | ✘ | ✘ |
| OpenCerts (OpenCerts, 2021) | PoW | ✔ | ✘ | ✘ | ✔ | ✘ |
| TrustED (Batzavalis et al., 2021) | PoA + PoS | ✘ | ✘ | ✔ | ✔ | ✘ |
| AcaChain (Bhumichitr and Channarukul, 2020) | Raft | ✔ | ✘ | ✘ | ✔ | ✘ |
| Anant et al. (2020) | PoW | ✔ | ✘ | ✘ | ✘ | ✘ |
| MS & MH (Ahammad M.S. Tomal MH, 2020) | PoW | ✘ | ✘ | ✘ | ✔ | ✘ |
| Singh et al. (2020) | Raft + DSs | ✔ | ✔ | ✔ | ✘ | ✘ |
| Hyperledger Indy (Linux Foundation, 2020) | Plenum | ✔ | ✔ | ✔ | ✔ | ✘ |
| Cerberus (Tariq et al., 2019) | PoA | ✘ | ✘ | ✘ | ✔ | ✔ |
| EBSI and BCDiploma | Raft | ✔ | ✔ | ✔ | ✔ | ✘ |
| **PriFoB** | PoA + SoW | ✔ | ✔ | ✔ | ✔ | ✔ |

rapidly drain storage and computation resources. Thus, these projects may not be considered practical in industrial deployment. Similar approach was utilized by other solutions as well, e.g. Li et al. (2021).

Anant et al. (2020) proposed deploying BC for SRM Institute of Science and Technology digital credential validation, using Ethereum smart contracts. This approach also saves all students credentials and data on the public chain. Similarly, Ahammad M.S. Tomal MH (2020) proposed a BC-based system that saves only the signed hash of each certificate instead of the certificate itself, while deployed the inefficient Proof-of-Work (PoW) CA to maintain the consistency of the DL. In Singh et al. (2020), a BC-based privacy preserving protocol is proposed so that users can access on-chain services. The proposed protocol suggests that a verifier is needed to personally verify the correctness of a claimed credential. Furthermore, in the generality of this protocol, the assumption of the user having to be a member of the BC raises some questions regarding deployment feasibility.

Hyperledger Indy (Linux Foundation, 2020) is an open source project, administered by the Linux foundation, which aims at providing a BC-based Verifiable Credential (VC) system. The project implementation provides a platform for DIDs rooted on BCs or other DLs so that they are inter-operable across administrative domains, applications, etc. The project deploys privacy preserving mechanisms such as ZKPs, and takes good care on what data is saved on-chain. Indy uses the Plenum CA which is a special purpose RBFT CA (Aublin et al., 2013). The Sovrin BC (Windley, 2016) is built on top of Indy project for providing a general purpose, global DIDs. After years of development, this BC was officially launched in September 2019. Similarly, The GraphChain (Sopek et al., 2018) was proposed for exploiting the advantages of Indy and facilitate the issuance of Legal Entity Identifiers.

Tariq et al. (2019) proposed Cerberus, where Ethereum-based, private-permissioned BC architecture was utilized. Here, VCs and (if applicable) revoke TXs are saved on-chain. A QR code-scanning approach is deployed then to verify a VC is indeed on-chain and that it is not revoked. The network in Cerberus is monitored and maintained by a single accreditation body that refers to its local DB for a full list of issuers. Thus, an issuer is automatically accredited by the system only if it was accredited through a legacy accreditation channel. Although the architecture seemed very promising, the authors have only discussed their proposal theoretically while no implementation or experimental comparisons were provided.

In De Souza et al. (2019), a strategy for modeling, designing and developing BC-based healthcare accreditation and verification solutions was discussed. Although the paper aimed at e-Health applications, it provided rich insights on design principles, trade-offs, and critical terminology definitions. Similarly to most proposals for BC-based accreditation and verification solutions, this proposal was discussed only in theory.

In November 2019, the World Wide Web Consortium[3] (W3C) published a standard recommendation for solutions targeting VC solutions, including data models, system concepts, approved methods, privacy and security challenges with proposed solutions, and validation. EBSI[4] and BCDiploma[5] are examples of services built according to this standard with the BC as the TTP where system entities save their data. However, the standard recommendation of W3C[6] and the solutions following it did not consider the issuer accreditation challenge assuming any entity should be able to own a DID and issue any type of VCs.

Comparing EBSI with our proposed solution, EBSI utilizes the general purpose Hyperledger Fabric[7] which uses the Raft CA. VC issuers in EBSI apply to an accreditation body for accreditation outside EBSI scope. Once an issuer is accredited in a legacy approach, the accreditation body issues a VC on the EBSI network certifying that this issuer is eligible to issue credentials. Miners in EBSI refer to bodies who attain accreditation confirmation. The BCDiploma platform directly saves all issued VC hashes on the BC which means that they can never be deleted, and it uses a PoW-based BC with linear DL model. Additionally, issuer accreditation service is not provided.

It is worth noting that different asymmetric encryption schemes are available in the literature. Many previous works deployed the most famous alternative scheme of the RSA namely Elliptic Curve Cryptography (ECC) (Hankerson et al., 2006). For this we performed a brief comparison between RSA and ECC to decide on which is more suitable to deploy in PriFoB. Table 4 presents a brief comparison between the two schemes referring to different benchmarks. To clarify, ECC keys are shorter than RSA keys for an equivalent level of security. Thus, ECC scheme is considered more secure than RSA comparing their keys of the same length. ECC is relatively fast in key generation and cipher decryption, while RSA is relatively fast in plain-text encryption and signature verification. As will be discussed later, we found that PriFoB entities mostly utilize the asymmetric encryption framework for signature verification, while signing is occasionally utilized. We also found that an RSA key with length of 1024 bits is currently considered post-quantum and secure, while such length of a key performs fast enough to outperform state-of-the-art solutions. For these reasons, we decided to use the RSA encryption framework with modifiable 1024-bit keys.

In the late 2020, Wang et al. (2020) have comprehensively investigated the state-of-the-art regarding DAG-based DL systems. They provided a general mathematical model of such systems, and categorized existing structures into six types. They could then systematically define potential applications and drawbacks of those structures as they were found either rough in summaries, superficial in analysis, or incomplete in evaluations. The main performance bottleneck of linear DL structure, which is not the case in DAG-based ones, was identified to be the utilized consensus mechanisms. Specifically, the competition

[3] https://www.w3.org/.

[4] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/EBSI.

[5] https://www.bcdiploma.com/en-GB.

[6] https://www.w3.org/TR/vc-data-model/#introduction.

[7] https://www.hyperledger.org/use/fabric.

**Table 4**
Comparison between RSA and ECC encryption schemes.

| Benchmark | ECC | RSA | Ref. |
|---|---|---|---|
| Post-Quantum? | Yes | Temporarily yes | Bafandehkar et al. (2013) |
| Security level | Better | Worse | Bafandehkar et al. (2013) |
| Key generation time | Better | Worse | Maletsky (2015) |
| Encryption time | Worse | Better | Mahto and Yadav (2017) |
| Decryption time | Better | Worse | Mahto and Yadav (2017) |
| Signing time | Better | Worse | Maletsky (2015) |
| Verification time | Worse | Better | Maletsky (2015) |

among a group of miners for the right of block packaging does not appear in DAG-based DLs as each newly added block is allowed to refer to more than one parent (many-to-many cardinality model of the BC). To solve this issue in linear DLs, block confirmation must be artificially suppressed (e.g. adjust the puzzle difficulty in the consensus method) so that each block is fully attached before the next one's arrival (resulting in one-to-one cardinality model of the BC). DAG-based DLs, on the other hand, support concurrent operations as multiple nodes can simultaneously add TXs/blocks to the DL, thereby significantly improving the throughput.

Type-II DAGs, specifically, have been utilized in several previous works including Spectre (Sompolinsky et al., 2016), Phantom (Sompolinsky and Zohar, 2018) and Meshcash (Bentov et al., 2017). However, Spectre uses it temporarily for swift processing but the final DL is linearized by a majority voting. Phantom utilizes a PoW consensus and enforces a strict linear ordering over blocks and transactions in the network, resulting in a DL with probabilistic finality. Meshcash utilizes a PoW consensus as well, with blocks pointing for necessity to *every* block confirmed in the previous round. None of those solutions offer a Tree-like multi-dimensional DL as PriFoB does. Specifically, all previous works adopt a many-to-many cardinality model; each newly added block may refer to more than one parent, and each parent may have several children. Our 3DDL, on the other hand, adopts a one-to-many cardinality model; each newly added block refers to specifically one parent, while each parent may have several children. To the best of our knowledge, our 3DDL is the first to propose a permanent, multi-dimensional and PoA- and SoW-based DL with deterministic finality.

Although our implementation is similar to few several recently proposed credential verification solutions (e.g. the Indy and Sovrin BCs) in terms of purpose and the BC access control modeling, we argue that our implementation is more appropriate. In PriFoB, accreditation is performed remotely and directly operated by the PriFoB network/platform, which reduces the whole process complexity. This is because miners in PriFoB are administered by the accreditation bodies themselves. Additionally, PriFoB has the following properties:

1. The DL model in PriFoB is Multi-Dimensional, implying enhanced and more efficient structure compared to the classical linear BC model.
2. PriFoB provides both services, namely global accreditation and credential validation, with GDPR- and W3C-compliance.
3. PriFoB utilizes two consensus layers, namely SoW and PoA. Accordingly, all nodes, rather than a majority of nodes in Plenum, should validate and sign, and later confirm a DID block.
4. The maximum number of miner nodes in systems deploying versions of the RAFT CA is recommended to be 25 nodes at maximum. This observation is inline with the findings presented in Dinh et al. (2017), where Hyperledger projects were reported to be limited by a maximum of 16 nodes. On the other hand, PriFoB can dynamically scale up and down according to the necessary number of miner nodes, with more efficient latency and throughput compared to PoW as well.

## 5. Conclusion

In this paper, we have proposed PriFoB: a Privacy-aware Fog-enhanced Blockchain-based solution for global accreditation and digital credential verification. We used Zero-Knowledge-Proofs, Digital Signatures, SHA-256, AES and RSA encryption schemes, and two different consensus algorithms, namely PoA and SoW. Furthermore, we have proposed a novel Three-Dimensional DAG-based Distributed Ledger (3DDL) with efficient deterministic finality. We evaluated PriFoB in terms of security, privacy, latency, throughput and power utilization. Additionally, we compared our realized cloud-based deployment of PriFoB with similar blockchain-based solutions, for various system parameterization. PriFoB outperformed all of the recently proposed solutions utilizing Ethereum and Hyperledger projects (Besu, Fabric and Indy). We provided a ready-to-deploy implementation of PriFoB, and we made it available at a public, open-source repository.

Our future directions include the investigation of Sharding and deploying Merkle Trees with light nodes. We will research the effects of increasing the block size in PriFoB, as it is currently set to 1 TX/B. We also plan to provide a web-based implementation of PriFoB entities and a wallet application for mobile devices. We further plan to enhance future versions of PriFoB by deploying the DONS protocol (Baniata et al., 2022a) to optimize the block finality and message propagation.

## CRediT authorship contribution statement

**Hamza Baniata:** Conceptualization, Methodology, Software, Writing – original draft, Validation, Investigation. **Attila Kertesz:** Supervision, Promotion, Writing – review and editing, Methodology, Validation, Investigation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The research leading to these results has received funding from the TruBlo project of the European Union's Horizon 2020 research and innovation program under grant agreement No. 957228, and from the national project TKP2021-NVA-09 implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, and from the University of Szeged Open Access Fund under the grant number 5750. All authors have read and agreed to the published version of the manuscript.

## Appendix A. PriFoB modeling using AGR4BS

AGR4BS (Roussille et al., 2021) is a generic organization-centric multi-agent model for BC systems relying on high-level abstractions (i.e., agents, groups, roles, and interaction types). This allows for a clear division of the different building blocks of BC systems, while leaving the possibility to explore behavioral divergence in a well defined framework.

Following the abstractions of the AGR4BS model, PriFoB consists of three structural groups, namely the DTTP layer, the Fog layer, and the end-user layer. Alternatively, we could have used the recommended grouping by the AGR4BS model authors (i.e. TX management group, Block management group, etc.), but we prefer to use our own representative abstractions.

There are also four types of elements in PriFoB, namely Gateways, Miners, Issuers, and End-users. Each element can perform one or more of the following eight roles:

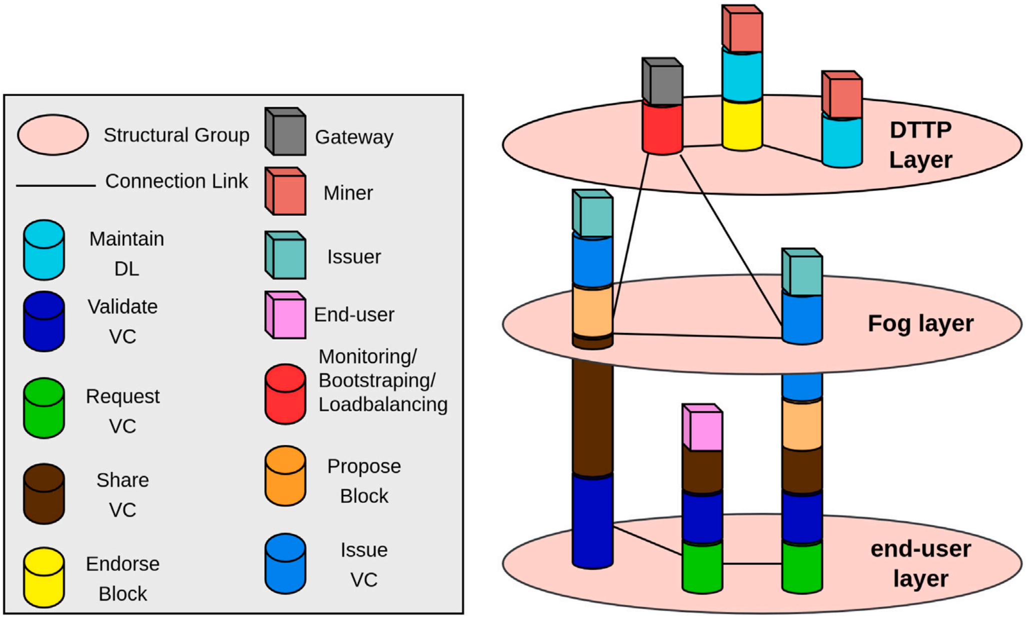


**Fig. 11.** Cheeseboard diagram for an organizational view of the PriFoB solution.

```
{"Header": {"Type": "DID_block",
           "miner_id": "10.128.0.32",
           "Signature": "\u000b... ("Body" signed by authorized Miner with its private key)",
           "Index": 10},
"Body": {"Transaction": {"Identifier": University of Szeged,
                      "institution_address": "10.128.0.29",
                         "institution_public_key": "MEgC...",
                         "Accredited By": {"USA": {"Address": "10.128.0.28",
                                               "Person Who Signed This": "General Registrar",
                                               "Signature": "0\u009f... (of Accreditation Body
                                                   ↪ using its private key)"},
                                           "UAE": {"Address": "10.128.0.27",
                                                "Person Who Signed This": "General Registrar",
                                                "Signature": "UPXd\u..."},
                                           "Jordan": {"Address": "10.128.0.33",
                                                 "Person Who Signed This": "General Registrar",
                                                 "Signature": "d<\u00..."}},
                      "Not Accredited by": {"Hungary": {"Address": "10.128.0.32",
                                                 "Person Who Signed This": "General Registrar",
                                                 "Signature": "D^\u00f..."}}},
       "Previous_Signature": "\u000f...(in the "Header" of the previous DID block )"},
"schemes_chain": [... new Schema blocks issued by this issuer are appended here]}
```

**Fig. 12.** A sample DID block in PriFoB, newly confirmed into the Blockchain.

1. Maintain the DL.
2. Validate a VC.
3. Request a VC.
4. Share a VC.
5. Endorse/Sign a Block.
6. Monitor/Bootstrap/load-balance the BC network.
7. Propose a Block.
8. Issue a VC.

Accordingly, PriFoB can be represented using the Cheeseboard diagram depicted in Fig. 11, where differently-colored boxes and cylinders represent different system elements, and the Roles they may perform, respectively. Note that a regular end-user is unable to propose a block or issue a VC. However, an issuer may perform some/all of regular end-user roles in addition to other functions. Specifically, an issuer belongs to both end-user and Fog layer because it can act differently in different situations. An issuer can be a customer for another higher-level issuer and request a VC.

## Appendix B. Sample PriFoB blocks

Fig. 12 presents a sample DID block in PriFoB, newly confirmed into the DTTP. Fig. 13 presents a sample Schema block in PriFoB, newly confirmed into the DTTP. Fig. 14 presents a sample Revoke block in PriFoB, newly confirmed into the DTTP.

## Appendix C. Encryption modeling

### C.1. RSA keys generation

To generate RSA Public and Private keys ($[n, e]$ and $d$):

```
{"Header": {"Type": "Schema_block",
            "miner_id": "10.128.0.28",
            "Signature": "\u000...("Body" signed by authorized Miner using its private key)",
            "Index": 1},
"Body": {"Transaction": {"institution_name": University of Szeged,
                        "institution_address": "10.128.0.29",
                        "Identifier": "PhD_schema",
                        "schema_public_key": "mDc...",
                        "schema_attributes": [["name","Mandatory"], ["Grade","Not Mandatory"],...(
                            ↪ more can be added here by issuer)],
                        "DID_index": 10},
      "Previous_Signature": "\u0006j...(in the "Header" of the previous Schema block)",
      "Issuer_Signature": "-jjK?\u...("Transaction" signed by issuer using DID private key)"}
"Hashes_of_revoked_credentials": [... new Revoke blocks, which uses this schema of this
    ↪ specific issuer are appended here]}
```

**Fig. 13.** A sample Schema block in PriFoB, newly confirmed into the Blockchain.

```
{"Header": {"Type": "Revoke_block",
         "miner_id": "10.128.0.27",
         "Signature": "\u00b6\u...("Body" signed by authorized Miner using its private key)",
         "Index": 8},
"Body": {"Transaction": {"DID_Identifier": University of Szeged,
                        "DID_index": 10,
                        "Schema_index": 1,
                        "institution_address": "10.128.0.29",
                        "Schema_Identifier": PhD_schema,
                        "Identifier": "943fc4a7...(hash of VC to be revoked)"},
    "Previous_Signature": 00e4\u...(in the "Header" of the previous Revoke block),
    "Issuer_Signature": "3\u00e4\u...(signed by the issuer using Schema private key)"}}
```

**Fig. 14.** A sample Revoke block in PriFoB, newly confirmed into the Blockchain.

1. Select two primes $P, Q \in \mathbb{Z}^+$, and compute $\langle n \leftarrow P \times Q \rangle$
2. Find $\langle \lambda(n) \leftarrow lcm$[8]$(P-1, Q-1)\rangle$
3. Select a third prime number $e \in \mathbb{Z}^+$, which is not a factor of $n$ nor $\lambda(n)$, such that $1 < e < \lambda(n)$.
4. compute the modular multiplicative inverse of $e \mod \lambda(n)$. That is $\langle e \times d( \mod \lambda(n)) \rightarrow 1\rangle$). Note that $d$ shall then satisfy: gcd[9]$(d, \lambda(n)) \rightarrow 1$.

### C.2. RSA encryption and decryption

To utilize the RSA algorithm for encrypting a plain text $M$ into a cipher $C$ and vice versa:

1. Represent $M$ as an integer using one of the standard representation methods (say $F(.)$).
2. The cipher $C$ can then be computed using the function $C \leftarrow F(M)^e \mod n$.
3. To decipher $C$, one can compute $C^d \mod n \rightarrow F'(M)$, which must equal $F(M)$.
4. Consequently, $M$ can be deduced from $F'(M)$ by using the same representation method in step 1.

## References

Ahammad M.S. Tomal MH, I.M., 2020. DistB-CVS: A distributed secure blockchain based online certificate verification system from Bangladesh perspective. In: 2nd International Conference on Advanced Information and Communication Technology 2020 (ICAICT 2020). IEEE, pp. 1–6.

Akkar, M.-L., Giraud, C., 2001. An implementation of DES and AES, secure against some attacks. In: International Workshop on Cryptographic Hardware and Embedded Systems. Springer, pp. 309–318.

Al Amiri, W., Baza, M., Banawan, K., Mahmoud, M., Alasmary, W., Akkaya, K., 2020. Towards secure smart parking system using blockchain technology. In: 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC). IEEE, pp. 1–2.

Anant, S., Sneha, I., Tanisha J., T., Ujjwala, S., 2020. Academic credential verification technique using blockchain. Int. J. Adv. Sci. Technol. 29, 4244–4254.

Anceaume, E., Pozzo, A., Rieutord, T., Tucci-Piergiovanni, S., 2020. On finality in blockchains. arXiv preprint arXiv:2012.10172.

Arif, M., Wang, G., Balas, V.E., Geman, O., Castiglione, A., Chen, J., 2020. Sdn based communications privacy-preserving architecture for vanets using fog computing. Veh. Commun. 26, 100265.

Aublin, P.-L., Mokhtar, S.B., Quéma, V., 2013. Rbft: Redundant byzantine fault tolerance. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems. IEEE, pp. 297–306.

Bacelar Almeida, J., Barbosa, M., Bangerter, E., Barthe, G., Krenn, S., Zanella Béguelin, S., 2012. Full proof cryptography: verifiable compilation of efficient zero-knowledge protocols. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. pp. 488–500.

Bafandehkar, M., Yasin, S.M., Mahmod, R., Hanapi, Z.M., 2013. Comparison of ECC and RSA algorithm in resource constrained devices. In: 2013 International Conference on IT Convergence and Security (ICITCS). IEEE, pp. 1–3.

Bampatsikos, M., Ntantogian, C., Xenakis, C., Thomopoulos, S.C., 2019. BARRETT BlockchAin regulated remote attestation. In: IEEE/WIC/ACM International Conference on Web Intelligence-Companion Volume. pp. 256–262.

Bandara, E., Ng, W.K., De Zoysa, K., Fernando, N., Tharaka, S., Maurakirinathan, P., Jayasuriya, N., 2018. Mystiko—blockchain meets big data. In: 2018 IEEE International Conference on Big Data (Big Data). IEEE, pp. 3024–3032.

Baniata, H., Anaqreh, A., Kertesz, A., 2022a. DONS: Dynamic optimized neighbor selection for smart blockchain networks. Future Gener. Comput. Syst. 130, 75–90.

Baniata, H., Kertész, A., 2020. PF-BVM: A privacy-aware fog-enhanced blockchain validation mechanism. In: CLOSER. pp. 430–439.

Baniata, H., Kertesz, A., 2020. A survey on blockchain-fog integration approaches. IEEE Access 8, 102657–102668.

Baniata, H., Pflanzner, T., Feher, Z., Kertesz, A., 2022b. Latency assessment of Blockchain-based SSI applications utilizing Hyperledger Indy. In: Proceedings of the 12th International Conference on Cloud Computing and Services Science - CLOSER. SciTePress.

[8] Least Common Multiple.

[9] Greatest Common Divisor.

Batzavalis, K., Bala, R., Norta, A., Norta Partners, O., 2021. A platform for leveraging blockchain technology for the storage, issuance and authentication of academic credentials. https://www.trusteducation.io/. Accessed: 2021-12-16.

Begum, A., Tareq, A., Sultana, M., Sohel, M., Rahman, T., Sarwar, A., 2020. Blockchain attacks, analysis and a model to solve double spending attack. Int. J. Mach. Learn. Comput. 10 (2), 352–357.

Bentov, I., Hubácek, P., Moran, T., Nadler, A., 2017. Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies. IACR Cryptol. ePrint Arch. 2017, 300.

Bhumichitr, K., Channarukul, S., 2020. AcaChain: Academic credential attestation system using blockchain. In: Proceedings of the 11th International Conference on Advances in Information Technology. pp. 1–8.

2021. BlockCerts home page. https://www.blockcerts.org/. Accessed: 2022-04-26.

Chakroun, B., Keevy, J., 2018. Digital Credentialing: Implications for the Recognition of Learning across Borders, Vol. 7. United Nations Educational, Scientific and Cultural Organization.

De Cannière, C., 2007. Analysis and Design of Symmetric Encryption Algorithms (Doctoral Dissertaion). KULeuven.

De Souza, Jr., J., De Araújo, D., Barbosa, G., Letouze, P., 2019. An international accreditation system for healthcare professionals based on blockchain. Int. J. Inf. Educ. Technol. 9 (7), 462–469.

Dennis, R., Owen, G., 2015. Rep on the block: A next generation reputation system based on the blockchain. In: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE, pp. 131–138.

Dierks, T., Rescorla, E., 2008. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, pp. 1–104.

Dinh, T.T.A., Wang, J., Chen, G., Liu, R., Ooi, B.C., Tan, K.-L., 2017. Blockbench: A framework for analyzing private blockchains. In: Proceedings of the 2017 ACM International Conference on Management of Data. pp. 1085–1100.

Dogtiev, A., 2021. Mobile advertising rates (2021). businessofapps.com/ads/research/mobile-app-advertising-cpm-rates/. Accessed: 2021-12-01.

Dunphy, P., Petitcolas, F.A., 2018. A first look at identity management schemes on the blockchain. IEEE Secur. Priv. 16 (4), 20–29.

Eichhorn, L., Shreedhar, T., Zavodovski, A., Mohan, N., 2021. Distributed ledgers for distributed edge: Are we there yet? In: Proceedings of the Interdisciplinary Workshop on (de) Centralization in the Internet. pp. 26–33.

European Commission, 2020. 2018 Reform of EU data protection rules. ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf.

Fauteux, L., Alexandre, P., Maxime, L., Guillaume, M., 2020. Blockchain–Tackling the Academic Credential Verification Problem. Technical Report, Tampere University, Faculty of Management and Business, p. 26.

Garay, J.A., Kiayias, A., Panagiotakos, G., 2020. Consensus from signatures of work. In: Cryptographers' Track at the RSA Conference. Springer, pp. 319–344.

Gonzalez, M.A.R., 2020. Discrete event simulation of a queue using python. github.com/miguelrizzog96/Queue_Simulation_Python.

Google corp., 2021. VM instances pricing. cloud.google.com/compute/vm-instance-pricing. Accessed: 2021-12-01.

Habibi, P., Farhoudi, M., Kazemian, S., Khorsandi, S., Leon-Garcia, A., 2020. Fog computing: a comprehensive architectural survey. IEEE Access 8, 69105–69133.

Hankerson, D., Menezes, A.J., Vanstone, S., 2006. Guide to Elliptic Curve Cryptography. Springer Science & Business Media.

Härer, F., Fill, H.-G., 2019. Decentralized attestation of conceptual models using the ethereum blockchain. In: 2019 IEEE 21st Conference on Business Informatics (CBI), Vol. 1. IEEE, pp. 104–113.

Johnson, L., 2019. Security Controls Evaluation, Testing, and Assessment Handbook. Academic Press.

Kertesz, A., Baniata, H., 2021. Consistency analysis of distributed ledgersin fog-enhanced blockchains. In: European Conference on Parallel Processing. Springer.

Khosroabadi, F., Fotouhi-Ghazvini, F., Fotouhi, H., 2021. SCATTER: Service placement in real-time fog-assisted IoT networks. J. Sensor Actuator Netw. 10 (2), 26.

Langley, D.J., van Doorn, J., Ng, I.C., Stieglitz, S., Lazovik, A., Boonstra, A., 2021. The internet of everything: Smart things and their impact on business models. J. Bus. Res. 122, 853–863.

Li, Q.-L., Ma, J.-Y., Chang, Y.-X., 2018. Blockchain queue theory. In: International Conference on Computational Social Networks. Springer, pp. 25–40.

Li, Z., Wu, H., Lao, L.H., Guo, S., Yang, Y., Xiao, B., 2021. Pistis: Issuing trusted and authorized certificates with distributed ledger and TEE. IEEE Trans. Parallel Distrib. Syst. 33 (7), 1636–1649.

Linux Foundation, 2020. HyperLedger indy. hyperledger.org/use/hyperledger-indy.

Loffi, L., Westphall, C.M., Grüdtner, L.D., Westphall, C.B., 2021. Mutual authentication with multi-factor in IoT-Fog-Cloud environment. J. Netw. Comput. Appl. 176, 102932.

Mahto, D., Yadav, D.K., 2017. RSA and ECC: a comparative analysis. Int. J. Appl. Eng. Res. 12 (19), 9053–9061.

Maletsky, K., 2015. RSA vs ECC Comparison for Embedded Systems, Vol. 5. White Paper, Atmel.

Malyan, R.S., Madan, A.K., 2021. Blockchain technology as a tool to manage digital identity: A conceptual study. In: Advances in Manufacturing and Industrial Engineering. Springer, pp. 635–647.

Mell, P., Scarfone, K., Romanosky, S., 2006. Common vulnerability scoring system. IEEE Secur. Priv. 4 (6), 85–89.

Mutlag, A.A., Khanapi Abd Ghani, M., Mohammed, M.A., Maashi, M.S., Mohd, O., Mostafa, S.A., Abdulkareem, K.H., Marques, G., de la Torre Díez, I., 2020. MAFC: Multi-agent fog computing model for healthcare critical tasks management. Sensors 20 (7), 1853.

Notheisen, B., Cholewa, J.B., Shanmugam, A.P., 2017. Trading real-world assets on blockchain. Bus. Inf. Syst. Eng. 59 (6), 425–440.

Nowak, M., Walkowski, M., Sujecki, S., 2021. Conversion of CVSS base score from 2.0 to 3.1. In: 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, pp. 1–3.

2021. OpenCerts home page. https://opencerts.io/. Accessed: 2022-04-26.

Pervez, H., Muneeb, M., Irfan, M.U., Haq, I.U., 2018. A comparative analysis of DAG-based blockchain architectures. In: 2018 12th International Conference on Open Source Systems and Technologies (ICOSST). IEEE, pp. 27–34.

Petkus, M., 2019. Why and how zk-snark works. arXiv preprint arXiv:1906.07221.

Proton Technologies AG, 2021. GDPR checklist for data controllers. https://gdpr.eu/checklist/, Accessed: 2021-12-12.

Rivest, R.L., Shamir, A., Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21 (2), 120–126.

Roussille, H., Gürcan, O., Michel, F., 2021. AGR4BS: A generic multi-agent organizational model for blockchain systems. Big Data Cogn. Comput. 6 (1), 1.

Severini, T.A., 2000. Likelihood Methods in Statistics. Oxford University Press.

Shah, J., Dubaria, D., 2019. Building modern clouds: using docker, kubernetes & Google cloud platform. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, pp. 0184–0189.

Shaik, S., Baskiyar, S., 2018. Hierarchical and autonomous fog architecture. In: Proceedings of the 47th International Conference on Parallel Processing Companion. pp. 1–8.

Shi, E., 2019. Analysis of deterministic longest-chain protocols. In: 2019 IEEE 32nd Computer Security Foundations Symposium (CSF). IEEE, pp. 122–12213.

Singh, K., Dib, O., Huyart, C., Toumi, K., 2020. A novel credential protocol for protecting personal attributes in blockchain. Comput. Electr. Eng. 83, 106586.

Singh, P.K., Singh, R., Nandi, S.K., Nandi, S., 2019. Managing smart home appliances with proof of authority and blockchain. In: International Conference on Innovations for Community Services. Springer, pp. 221–232.

Smetanin, S., Ometov, A., Kannengießer, N., Sturm, B., Komarov, M., Sunyaev, A., 2020. Modeling of distributed ledgers: Challenges and future perspectives. In: 2020 IEEE 22nd Conference on Business Informatics (CBI), Vol. 1. IEEE, pp. 162–171.

Sompolinsky, Y., Lewenberg, Y., Zohar, A., 2016. SPECTRE: a fast and scalable cryptocurrency protocol. IACR Cryptol. ePrint Arch. 2016 (1159).

Sompolinsky, Y., Zohar, A., 2018. Phantom. IACR Cryptol. ePrint Arch. Report 2018/104.

Sopek, M., Gradzki, P., Kuzinski, D., Trojczak, R., Trypuz, R., 2018. Legal entity identifier blockchained by a hyperledger indy implementation of GraphChain. In: Research Conference on Metadata and Semantics Research. Springer, pp. 26–36.

Tariq, A., Haq, H.B., Ali, S.T., 2019. Cerberus: A blockchain-based accreditation and degree verification system. arXiv preprint arXiv:1912.06812.

Thomas Porter, C., CCNP, C., Gough, M., et al., 2011. How to Cheat at VoIP Security. Syngress.

Urbančok, D., 2019. Blockchain Open-Source Software Comparison (Master's thesis). Masaryk University, Faculty of Informatics, Brno, The Czech Republic.

Vo, H.T., Kundu, A., Mohania, M.K., 2018. Research directions in blockchain data management and analytics. In: EDBT. pp. 445–448.

Wang, Q., Yu, J., Chen, S., Xiang, Y., 2020. SoK: Diving into DAG-based blockchain systems. arXiv preprint arXiv:2012.06128.

Wang, T., Zhou, J., Liu, A., Bhuiyan, M.Z.A., Wang, G., Jia, W., 2018. Fog-based computing and storage offloading for data synchronization in IoT. IEEE Internet Things J. 6 (3), 4272–4282.

Windley, P.J., 2016. How sovrin works. windely.com.

Xu, X., Sun, G., Luo, L., Cao, H., Yu, H., Vasilakos, A.V., 2021. Latency performance modeling and analysis for hyperledger fabric blockchain network. Inf. Process. Manage. 58 (1), 102436.

Yoshida, H., Biryukov, A., 2005. Analysis of a SHA-256 variant. In: International Workshop on Selected Areas in Cryptography. Springer, pp. 245–260.

Zhang, L., Lee, B., Ye, Y., Qiao, Y., 2019. Ethereum transaction performance evaluation using test-nets. In: European Conference on Parallel Processing. Springer, pp. 179–190.

Zheng, Z., Xie, S., Dai, H.-N., Chen, X., Wang, H., 2018. Blockchain challenges and opportunities: A survey. Int. J. Web Grid Serv. 14 (4), 352–375.

**Hamza Baniata** is a Ph.D. candidate at the Doctoral School of Computer Science at University of Szeged, Hungary. He is a member of the IoT-Cloud research group at the Department of Software Engineering, actively contributing to the Fog-Block4Trust sub-grant project of the TruBlo EU H2020 project, the CERCIRAS EU Cost Action, and the OTKA FK 131793 project. He received his B.Sc. degree in Computer and Military Sciences from Mutah University (Jordan, 2010), And his M.Sc. degree with excellence in Computer Science from the University of Jordan (Jordan, 2018). His work experience includes different roles in the domains of ICT and Security, while his current research interests fall in the domains of Security, Privacy and Trust of Blockchain, Cloud/Fog Computing, and Internet of Things systems.

**Attila Kertesz** is currently with the University of Szeged, Szeged, Hungary. He is an associate professor at the Department of Software Engineering, leading the IoT-Cloud research group. He graduated as a program-designer mathematician in 2005, received his Ph.D. degree at the SZTE Doctoral School of Computer Science in 2011, and habilitated at the University of Szeged in 2017. His research interests include the federative management of Blockchain, IoT, Fog and Cloud systems, and interoperability issues of distributed systems in general. He is the leader of the FogBlock4Trust sub-grant project of the TruBlo EU H2020 project, and the OTKA FK 131793 national project financed by the Hungarian Scientific Research Fund. He is also a Management Committee member of the CERCIRAS and INDAIRPOLLNET EU Cost Actions. He has more than 130 publications with more than 1100 citations.