Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5046

ONSPRIDE: An Ontology-based Framework for Privacy by Design in Distributed Networks Apps

Marco Antonio Colombo da Silva [**Federal Institute of São Paulo** | ***marco.colombo@ifsp.edu.br***]
Luis Hideo Vasconcelos Nakamura [**Federal Institute of São Paulo** | ***nakamura@ifsp.edu.br***]
Geraldo P. Rocha Filho [**State University of Southwest Bahia** | ***geraldo.rocha@uesb.edu.br***]
Luís Veiga [**INESC-ID, Higher Technical Institute, University of Lisbon** | ***luis.veiga@inesc-id.pt***]
Rodolfo Ipolito Meneguette [**University of São Paulo** | ***meneguette@icmc.usp.br***]

Institute of Mathematical and Computer Sciences, University of São Paulo (USP)
Av. Trabalhador São Carlense, 400 - Centro, São Carlos - SP, 13566-590, Brazil.

Received: 05 October 2024 • **Accepted:** 10 March 2025 • **Published:** 15 May 2025

Abstract With the advancement of technologies for data registration in distributed networks, the concern of users and developers of computerized solutions with the privacy of sensitive data has increased. Thus, this work addresses a conceptual solution for an ontology-based framework so that any entity willing to provide a service using Distributed Ledger Technology (DLT) networks can model the set of privacy attributes of its system according to the business rules of its service. The solution proposed in this work encompasses the development of an architecture aimed at providing computational support for the privacy design of the actors involved in the offering and consumption of services implemented in DLTs. The architecture also includes a framework called ONSPRIDE, which uses previously stored domain ontologies to translate business rules into requirements and privacy. We conducted a proof of context by comparing the performance of two Hyperledger Fabric networks. For this purpose, we conducted a controlled experiment in which both networks operate a smart contract that manages attendance records for outdoor events. The main difference between the networks is that one uses a Certificate Authority (CA) to issue access certificates, while the other issues certificates manually. We compared the results obtained through the reports generated by the Hyperledger Caliper tool. In addition, the performance of the initialization and connection of agents in a Self-Sovereign Identity system was measured. The results of this study provide valuable insight that can help developers choose the most suitable ledger type for their Hyperledger projects and support decision-making regarding adopting a Self-Sovereign Identity system.

Keywords: Self-Sovereign Identity, Privacy by Design, Blockchain, Ontology

1 Introduction

Multiple applications belonging to private entities and government agencies require the systematic and secure handling of sensitive information. Regardless of the sector, it is imperative for the corporations involved to establish and rigorously adhere to clear privacy regulations. At the same time, users, whose security depends on strict compliance with these rules, must be aware of such guidelines and assured of their practical implementation in the face of increasing malicious efforts to breach data confidentiality Bu *et al.* [2020].

Currently, we observe that most users' digital identities are centralized and managed by a small number of large corporations, depriving users of control over their personal information Liu *et al.* [2020]. This scenario contributes to data commercialization, marked by a notable need for more transparency. Lux *et al.* [2019]; Maschi *et al.* [2018].

Some technological approaches and tools enable anonymous user authentication in decentralized computer systems securely and reliably, while also allowing for the customization of sensitive data management. These tools use blockchains representing distributed ledgers, are fully accessible to the public, and are recognized for their high security Carlozo [2017].

The privacy characteristics of blockchain are crucial for enhancing the reliability associated with using this technology and for driving innovations in defensive techniques and countermeasures Zhang *et al.* [2019].

This work aims to propose a conceptual solution for an ontology-based framework. It would enable an entity willing to provide a service using DLT networks to model the set of privacy attributes of its system according to the business rules of its service and the concepts of Privacy by Design (PbD).

In this context, the performance of two blockchain networks, one implemented with complex and comprehensive digital certificate management and the other with manual digital certificate management, is evaluated.

Additionally, we conducted experiments to measure the performance of Self-Sovereign Identity Agents, which we can use in conjunction with the ONSPRIDE Framework.

Developers may consider the information gathered based on this work's test results regarding the adoption or non-adoption of a CA and Self-Sovereign Identity (SSI), depending on the volume of certificates and entities their applications expect to manage.

This work contributes a conceptual framework for creating privacy-focused software solutions that utilize DLT networks adaptable to different domains through an ontology that bridges DLT concepts with domains representing the business rules of their creators. It also presents the results of the performance tests that can be a basis for choosing different implementations and approaches.

We organized this work as follows: Section 2 provides information about the essential concepts and technologies needed to understand this work. Section 3 discusses the related literature. Section 4 presents our approach to designing our solution. In Section 5, we present the performance comparison experiments carried out. Section 6 presents the results obtained from the experiments, and Section 7 discusses them. Finally, Section 8 presents our final considerations for this work.

2 Background

The technological context used for creating and applying the ONSPRIDE framework involves a technological framework closely related to Ontologies, Blockchain, Self-Sovereign Identity, Privacy by Design, and related technologies.

In accordance with the concepts observed in PbD, especially those related to end-user access privacy in blockchain systems, we have created a framework that assists developers in designing decentralized applications that use self-sovereign identity for access. Given that the precepts of SSI align with PbD, its adoption should consider the overall system performance, which, in the case of Hyperledger Fabric DLT networks, can be affected by the use of a CA as an access method. The performance of this type of DLT network can be measured using the Hyperledger Caliper tool. For planning software with different business rules, the framework employs domain ontologies that are linked to the characteristics of DLT networks through an ontology responsible for this connection.

2.1 Privacy by *Design*

Privacy, previously understood intuitively as the right to choose which personal data is disclosable and which data should be kept confidential, now takes on more complex nuances. Many researchers find it challenging to reduce the term to a single definition influenced by context, time, or area of application Bawden and Robinson [2020]. In the scope of this work, we will assume a more strictly technical view, avoiding philosophical and ethical perspectives on this concept, given the limitations of the DLT network technology ecosystem explored here.

This work considers the need to protect individuals when forced by collective or institutional pressure to give up their privacy de Souza *et al.* [2020]. In line with this protective movement, a design methodology emerged around 2010 that places privacy as a central element in planning any product, system, process, or service, called PbD Aljeraisy *et al.* [2021].

We embed privacy throughout the product or service life cycle, from design to disposal. The benefits of using PbD include (1) greater awareness of privacy and the handling of personal information in projects, products, services, systems, or processes of an agency, early identification and resolution of possible privacy risks and problems in a more straightforward and less costly manner, (2) greater assurance of compliance with privacy principles provided by law.

We should apply PbD to technology and any activity, such as acquisition, analysis, evaluation, or policy development, where personal information is collected and used. Best practices also recommend considering unidentified or anonymous information derived from personal information.

2.2 Self-Sovereign Identity

SSI is a concept created to describe an identity management system that can work for public, private, and third-sector users. SSI's architecture is based on decentralized technologies and designed to focus on user security, privacy, and individual autonomy Giannopoulou and Wang [2021].

Considering today's digitized world and its needs, SSI management involves storing and maintaining specific attributes data while always controlling access to this data. SSI is rooted in the belief that individuals have the right to an identity independent of reliance on a third-party identity provider, such as the State or any other central authority. Its implementation requires the development of technical standards, as well as sociopolitical adaptations to succeed Giannopoulou and Wang [2021].

Identity management has three main functions: holder, issuer, and verifier. An entity registering an identifier associated with some attribute data in a given system is considered an identity holder. A credential is a verifiable claim of some identity attribute data or facts related to the holder. The claim is certified and digitally signed by a credential issuer. A credential verifier is an entity that requests a specific credential from a trusted issuer and corroborates the authenticity of the credential through the issuer's signature. Most existing identity management solutions require a centralized authority for attribute registration or credential verification Liu *et al.* [2020].

According to Giannopoulou and Wang [2021], SSI is often implemented as blockchain-adjacent but not as blockchain-dependent identity management systems, which are guided by the fundamental principle of user-centred design, using technical standards that allow user-generated and controlled decentralized identifiers (DID), associated credentials, and attestations. This context also counts with legal and policy requirements to ensure that objectives for specific use cases are met, including balancing competing social goals such as user privacy, security, law enforcement, financial inclusion, and risk management.

2.3 Ontologies

Computer Science uses ontologies to represent, name, and define categories, properties, and relationships between concepts, data, and entities in one or more domains. These ontologies are part of the web technologies stack (Figure 1). The semantic web, initially intended for the WWW – World Wide Web, provides a common framework for sharing and reusing data across different applications and enterprises. Most WWW data is human-readable but not machine-

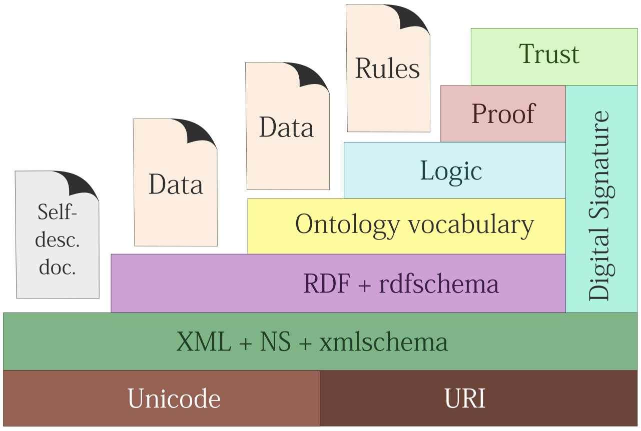

Figure 1. Semantic Web Technology Stack Berners-Lee *et al.* [2001]

readable. The semantic web was created to make this possible. Once data exists in a machine-readable format, it is possible to generate intelligent agents that can relate resources of various natures Hector and Boris [2020].

According to De Kruijff and Weigand [2017], researchers recognize ontology as a valuable tool for reducing conceptual ambiguities and inconsistencies while identifying value-creation capabilities in a given domain, making ontology an increasingly important tool for reducing complexity through structuring domains of interest.

2.4 Blockchain

According to Androulaki *et al.* [2018]:

> ”A blockchain is an immutable ledger for recording transactions, maintained within a distributed peer-to-peer network of mutually distrusting participants. Each peer maintains a copy of the ledger. The peers execute a consensus protocol to validate transactions, group them into blocks, and build a chain.”

Hyperledger Fabric is an open-source blockchain platform developed under the Linux Foundation umbrella. It allows for the creation of permissioned blockchain networks with pre-authorized participants, and each can have different levels of access and permissions Foundation [2021b].

According to Vukoli [2017], Hyperledger Fabric provides a modular architecture and includes a membership component, allowing developers to create solutions that fit the specific needs of their organizations.

Hyperledger Fabric also supports the execution of chaincodes, known in other blockchain contexts as smart contracts. These are scripts executed on the blockchain that automate, validate, or securely and efficiently execute transactions Foundation [2021b].

2.5 Certificate Authority

In the Hyperledger Fabric architecture, the CA plays a central role in identity management, issuing digital certificates crucial for authentication and authorization within the blockchain network. These certificates validate the participants' identity and enable secure and reliable operations within the networkFoundation [2021b]. A Certificate Authority in Hyperledger Fabric offers greater security, scalability, and flexibility in managing certificates and identities, making it suitable for production environments Gayathri Santhosh and Reshmi [2023].

2.6 Hyperledger Caliper

Hyperledger Caliper is a blockchain benchmarking tool part of the Hyperledger project hosted by the Linux Foundation. Its main goal is to measure the performance of a specific blockchain network across various metrics, such as transactions per second, transaction latency, resource usage (CPU, memory, and others), and throughput under different network conditions and transaction loads Foundation [2022].

Hyperledger Caliper supports multiple blockchain platforms, including Hyperledger Fabric, Hyperledger Sawtooth, and others, allowing users to test and compare the performance of different blockchain technologies with a standard set of benchmarks. Hyperledger Caliper generates several performance reports Foundation [2022].

According to the Hyperledger project website Foundation [2021a], the community includes ”leaders in finance, banking, the Internet of Things, supply chains, manufacturing, and technology.” It is important to note that the solutions produced by the project are open-source and under open technical governance.

3 Related Work

We found articles in the literature that perform tests and performance evaluations on Hyperledger Fabric networks, considering different scenarios and focusing on specific targets in their analyses.

The article Kuzlu *et al.* [2019] evaluates the impact of network workload on the performance of a Hyperledger Fabric blockchain platform. The authors used Hyperledger Caliper to evaluate throughput, latency, and scalability performance. They conducted the experiments on AWS EC2, and the authors concluded that a blockchain network's throughput, latency, and scalability depend on the hardware configuration, blockchain network design, and the complexity of smart contract operations.

The article Wang and Chu [2020] evaluates the performance of the execute-order-validate architecture of Hyperledger Fabric. The execution phase showed good scalability under the OR endorsement policy, when a blockchain transaction is endorsed by at least one organization in the network, with ordering services (Solo, Kafka, and Raft) performing relatively well. In contrast, the validation phase was likely the system bottleneck due to the low validation speed of the chaincode.

The article Melo *et al.* [2022] evaluated the performance of a Hyperledger Fabric platform (v1.4.1) deployed in a private environment. The Caliper benchmarking tool managed and evaluated a single entity (latency and throughput). The authors detected increased resource consumption and identified a software aging issue. The article also includes an interesting evaluation of computing resource consumption, such as CPU, RAM, disk, and cache memory. The authors created a system availability model considering the increase in resource consumption and revealing the impact on the overall system availability.

Finally, the article Mor *et al.* [2024] evaluates and compares the performance of different versions of Hyperledger Fabric (v1.0 to v1.4.4), highlighting that under high workload circumstances, the platform's performance did not match that of contemporary conventional database systems. The authors hope that the results of this study will help the corporate community select the best blockchain platform for their needs.

The mentioned studies primarily focus on analyzing the performance of DLT networks, considering factors such as latency, throughput, and resource consumption. However, these studies do not explore the impact of CA adoption on networks or privacy enhancements. In contrast, this work not only evaluates performance but also introduces the ONSPRIDE framework, adopting an ontology-based approach that enables transaction classification based on business rules, thereby incorporating the principles of Privacy by Design (PbD).

As can be seen in Table 1, although the works above use related tools to perform performance evaluations on Hyperledger networks and fulfill their objectives in various contexts, unlike this work, they were not specifically designed for the performance evaluation and comparison of networks with a CA and networks that do not have this digital certificate manager.

4 The ONSPRIDE Framework

The process for developing the solution proposed in this work includes developing an architecture aimed at providing computational support for the privacy design of the actors involved in offering and consuming services implemented in DLTs. Within this architecture, we propose a framework called ONSPRIDE, which will use previously stored domain ontologies to translate business rules into requirements and privacy. The ONSPRIDE framework includes a new ontology for classifying transactions in DLTs. The following section will provide an overview of the proposed approach and its main components.

Developers should implement a solution with the ONSPRIDE framework for intelligent ontology-based transaction assistance in DLT networks, considering the ecosystem expected in an SSI access situation.

We selected SSI because it is directly related to the requirements of the problem addressed by our solution. These challenges involve privacy, user control over their data, and decentralized identity management. Scenarios that use DLT networks often rely on authentication through centralized entities, which can compromise privacy and introduce risks of excessive control over user credentials. SSI gives individuals full autonomy over their digital identities, allowing them to store their credentials in decentralized repositories and share only the necessary attributes for a given transaction. This approach is based on the principles of PbD, so in our solution, SSI enables the implementation of a secure and verifiable authentication mechanism, ensuring the privacy and data protection required for an attendance system in outdoor events.

The architecture of the solution (Figure 2) includes a layer of actors involved in the process, a layer consisting of the distributed identity repository, a layer composed of the DLT network where transaction records will be stored, and the proposed ONSPRIDE framework.

It is important to note that in the context of this work, actors are any organizations, individuals, devices, or computer systems that may directly interact with the proposed solution. In the layer of actors involved in the process, the following actors exist:

- **Verifier** – The role of the Verifier will be filled by any organization or individual interacting with the solution in two scenarios: (1) When this actor instantiates or creates a type of transaction to be used with DLT networks. (2) When this actor must verify the identity of a consumer of services registered through transactions in a DLT network. Verifiers include cryptocurrency companies, government institutions, and private companies providing services or products.
- **Owner** – This actor's objective in the system is to consume a service involving data recording in DLT networks, securely identifying themselves and with their privacy respected. This is to the point where they are the Owner and masters of their data, choosing what should be omitted and what should be revealed, as deemed necessary for the continuation of the service they wish to obtain from the Verifier. Examples of Owners include students in universities, participants in events with verified attendance, or owners of cryptocurrency wallets.
- **Issuer** – The Issuer actor will be responsible for registering the Owners' identity in the DID, signing, and electronically transferring it to them. They should be entities or individuals widely trusted by Verifiers. In a transaction, an Issuer may be physically the same entity acting as a Verifier. Issuers include government institutions, associations, guilds, or private companies.

The identity repository layer can be implemented using any DLT network capable of providing identities in a distributed manner, following the principles and requirements expected in a standard DLT, with some fundamental attributes for this solution being immutability, availability, reliability, and auditability. This layer receives information from the Issuer to register new identities and can also be queried by the Verifier when they need to verify the authenticity of an identity. Each identity stored in this layer will have an encrypted key to identify its Owner. Although public, without verifiable proof, the data in this repository would be meaningless in the event of improper use. Communication between this layer and the DLT network layer is described below.

The DLT network layer is responsible for storing the transaction record related to the service offered by the Verifier and consumed by the Owner. Similarly to the previously mentioned layer, this one also requires the basic attributes of a DLT network. For the transaction to occur, this layer must first receive access permission from the Identity layer and,

Paper	Compares networks	Uses Caliper	Focus on the CA	Privacy Issues
Kuzlu *et al.* [2019]	No	Yes	No	No
Wang and Chu [2020]	Yes	No	No	No
Melo *et al.* [2022]	No	Yes	No	No
Mor *et al.* [2024]	Yes	No	No	No
Present Paper (ONSPRIDE)	Yes	Yes	Yes	Yes

Table 1. Characteristics of Interest in Related Work

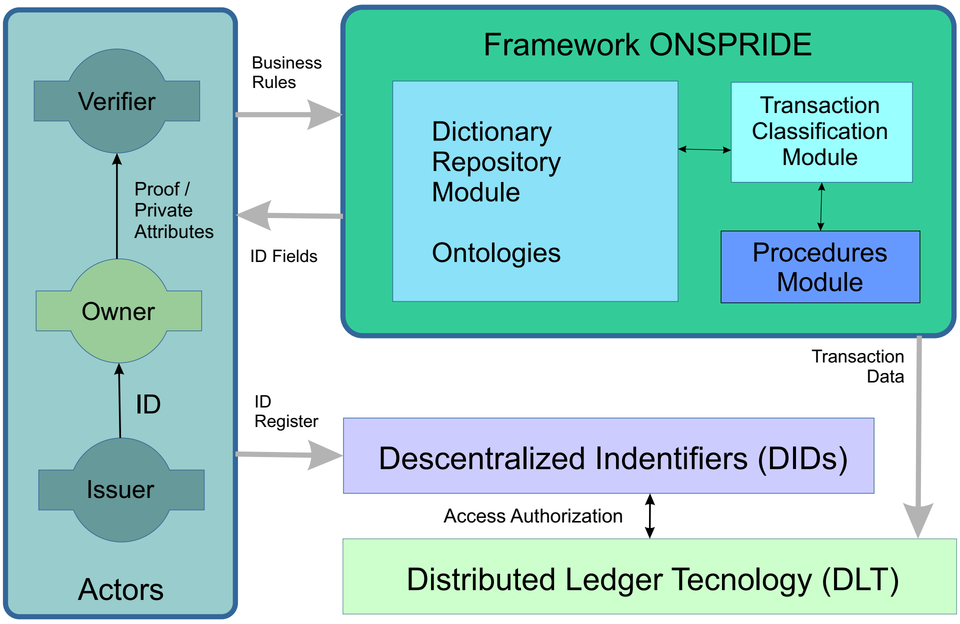
Verifier
Proof / Private Attributes
Owner
ID
Issuer
Actors
Business Rules
ID Fields
ID Register
Framework ONSPRIDE
Dictionary Repository Module
Ontologies
Transaction Classification Module
Procedures Module
Transaction Data
Descentralized Indentifiers (DIDs)
Access Authorization
Distributed Ledger Tecnology (DLT)

Figure 2. Solution Architecture with the ONSPRIDE Framework

subsequently, the transaction data sent by the ONSPRIDE framework, described below.

4.1 ONSPRIDE Modules

The ONSPRIDE framework proposed in this work aims to receive the business rules defined by the nature of the DLT transaction provided by the Verifier, classifying it according to a query in the domain ontologies defined at the beginning of the process. It is important to note that users of services in DLT networks can only use the ONSPRIDE framework in contexts where the ontologies have been pre-loaded in the solution's domain ontology component. ONSPRIDE includes an ontology module, a transaction classification module, and a procedures module. The following subsections will discuss each of these modules.

4.1.1 Transaction Classification Module

The Transaction Classification Module (Figure 3) receives the Verifier actor's business type and performs a query in the transaction repository through a procedure call from the procedures module.

If a corresponding transaction is found, the Verifier actor can analyze the transaction type's characteristics and decide whether it aligns with their business. If affirmative, the Verifier will receive fields to fill in according to the transaction found; these fields correspond to the desired privacy modeling. If negative, the Verifier will provide new characteristics of their business rules, which will be used as input for a query in the system's domain ontologies, starting with the Transaction Ontology proposed in this work.

The result of the previous query will be privacy attributes that, after processing through the Verifier's interface, will allow a new transaction to be inserted into the system via a specific procedure call in the Procedures Module.

4.1.2 Procedures Module

The Procedures Module (Figure 4) provides the routines to be used in queries with the ontologies, procedures for inserting new transaction types, and storage of inferences in domains available within the solution.

Firstly, the Pre-existing Transaction Query procedures will be executed at the beginning of each new solution. Its objective is to check if similar business rules have been entered into the system in any previous use. If so, this transaction will be returned and presented to the Verifier, who must confirm or reject its reuse.

Next, the Domain Ontology Query procedures are required to search through the system's preexisting ontologies, returning information related to the business rules provided by the Verifier. This return will enable the development of privacy attributes.

The set of Pre-existing Inference Query procedures will return inferences obtained in previous executions that have already produced reusable privacy attributes.

The Inference Insertion procedure aims to store the infer-

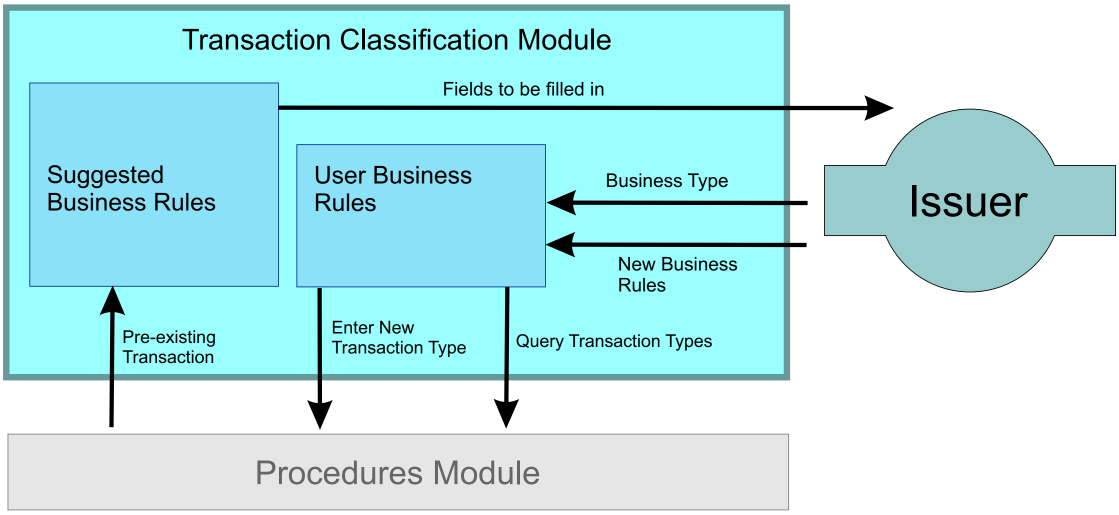

Figure 3. Transaction Classification Module of the ONSPRIDE Framework

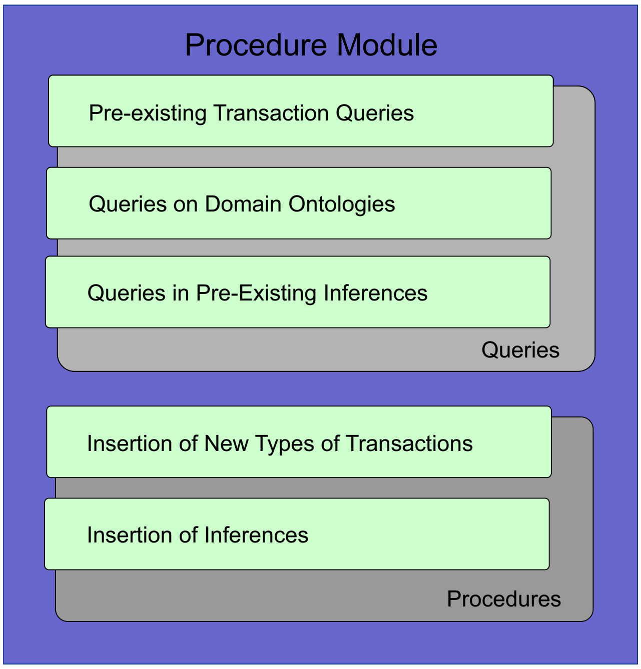
Procedure Module
Pre-existing Transaction Queries
Queries on Domain Ontologies
Queries in Pre-Existing Inferences
Queries
Insertion of New Types of Transactions
Insertion of Inferences
Procedures

Figure 4. Procedures Module of the ONSPRIDE Framework

ences obtained through the methods above, including their privacy attributes and all the data necessary for their traceability in the ontologies that generated them.

Finally, the New Transaction Type Insertion procedure aims to create a set of privacy attributes related to the business rules of a specific DLT service in the system.

Respecting the architecture's structural integrity, these procedure calls originate exclusively from the Transaction Classification Module.

4.1.3 Ontology Module

The Ontology Module (Figure 5) contains the domain ontologies necessary for transaction classification through queries from the Procedures Module.

The Domain Ontologies are pre-stored and are crucial for the solution to function. In other words, if the DLT service to be offered does not have its domain ontology already stored

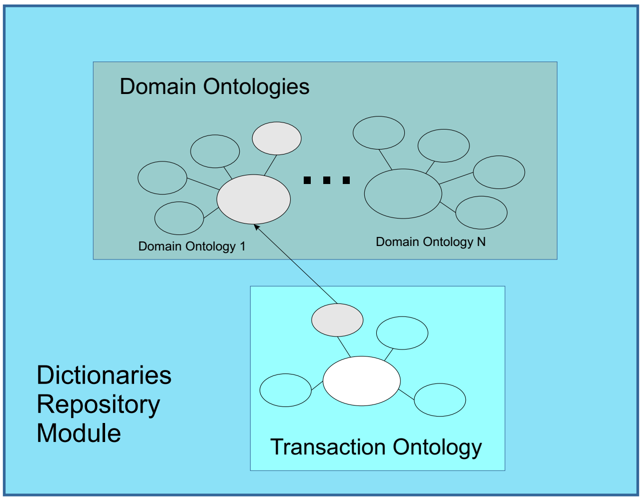

Figure 5. Ontology Module of the ONSPRIDE Framework

in the system, the modeling of privacy rules will only be possible once the system administrators implement this solution and update it with the desired domain ontology. As a proof of concept, the use case scenario in this work utilizes the TransTypO ontology Silva *et al*. [2024] for transaction types.

4.2 Use Case

This section describes a use case scenario with practical examples of its utilization and software architecture as an instance of the concept to facilitate the understanding and evaluation of the solution. In the chosen use case, organizers of outdoor events, such as public demonstrations, artistic performances, or parades, wish to calculate the number of participants in a reliable and auditable manner. However, the participants want to avoid being identified but want their presence to be counted as support for the event. Using decentralized applications and DLT network servers with the ONSPRIDE solution already implemented, the organizers will take on the roles of Verifiers and Issuers. At the same time, the participants will act as Owners.

The sequence of events (Figure 6) for registering a check-

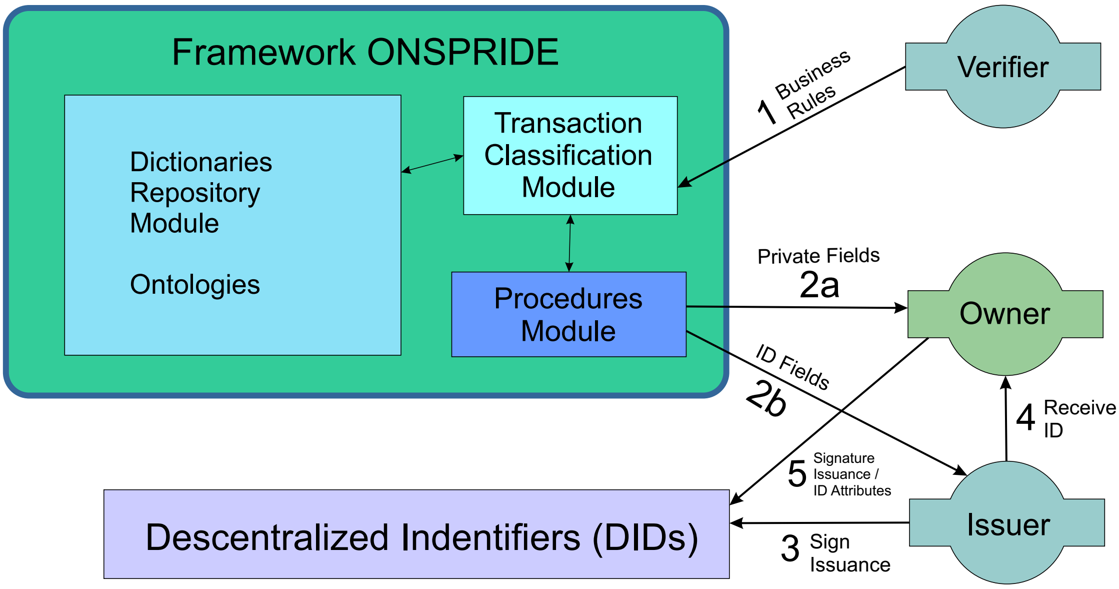

Figure 6. Use Case of the Solution with the ONSPRIDE Framework - ID Generation

in type transaction begins when the event organizers access the web solution to register their event. To complete this step, they must provide the system with the business rules related to outdoor event organization, according to its specificities. Still, in this step, they are guided by the application, which offers options based on results obtained from the ontologies of the chosen domain.

As a result of the previous step, the ONSPRIDE framework provides the sensitive fields that must be filled in by the event participant, who, as the Owner, must install a DApp (Decentralized Application). At this point, another output of the framework is the identification fields, which the Issuer must know, a role also performed by the event organizers, for the next step. The event organizers register and sign the participant's digital identification. The participant receives their electronic identification document from the organizer, with countersignature and identification fields to fill in.

The participant records their countersignature and provides encrypted identification data via the DApp (Figure 7). If the event participant agrees to proceed with the transaction, they can present proof of identity and private attributes. Armed with the digital file identifying the participant, the event organizers confirm it with the identity repository through the web application. Transaction data of the check-in type, such as timestamp, latitude, and longitude, are sent to the ONSPRIDE framework via the Web. Procedures for accessing the final ledger are executed to release the registration of the check-in type transaction. Finally, once access is granted, the framework records the check-in data in the event ledger.

Among the preexisting ontologies for DLT networks, we highlight one we can use for blockchain-type DLT networks because it adheres to the instance chosen to prove the concept proposed in this work. For this reason, we chose the BLONDiE ontology (Hector and Boris [2020]).

The BLONDiE ontology (Figure 8) covers the blockchain domain across three blockchain technologies: Bitcoin, Ethereum, and Hyperledger Fabric. It is a domain-specific ontology whose scope describes the native structure of these data structures and related information. Its creators used a leading ontology development tool called Protegé for its development, and the implementation is available on the GitHub repository. The fact that the BLONDiE ontology is extendable to other projects and can be enhanced to cover other ledgers led to its consideration for inclusion in the present proposal. At this point, it is essential to emphasize that the TransTypO ontology can couple with other ontologies created for different DLT networks or even with the BLONDiE ontology extended to other DLT networks.

For the ontology usage scenario, we chose the Simple Event Model (SEM) ontology (van Hage *et al.* [2011]) because it is capable of supporting the business rules of an application designed to control the attendance of people at outdoor events, such as conventions, parades, processions, artistic performances, and other gatherings of people. The central layer of entities in this ontology includes the representation of the event itself (Figure 9), the actors involved in the process, and the time and place where the event is to take place. It is worth noting that the proposed ontology can be used with other business models and, therefore, integrated with other ontologies as needed.

4.3 TransTypO Ontology

In Figure 10, the representation of the TransTypO ontology (Silva *et al.* [2024]) includes fragments of the BLONDiE ontology and the SEM ontology to provide a better understanding. To clarify the distinction, TransTypO ontology classes are prefixed with TyO, BLONDiE classes with BLND, and SEM classes with Sem.

TransTypO Entities The main class of the TransTypO Ontology is Transaction Type, and its purpose is to represent the type of transaction to be recorded in the blockchain ledger

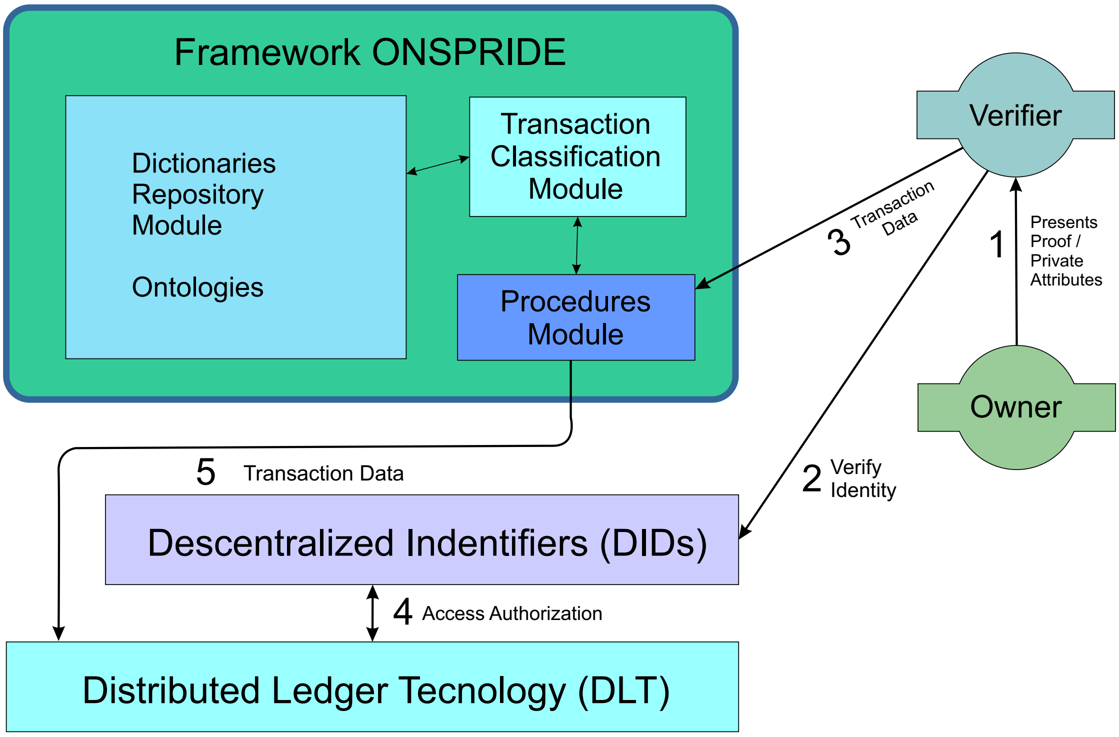

Figure 7. Use Case of the Solution with the ONSPRIDE Framework - ID Presentation and Access

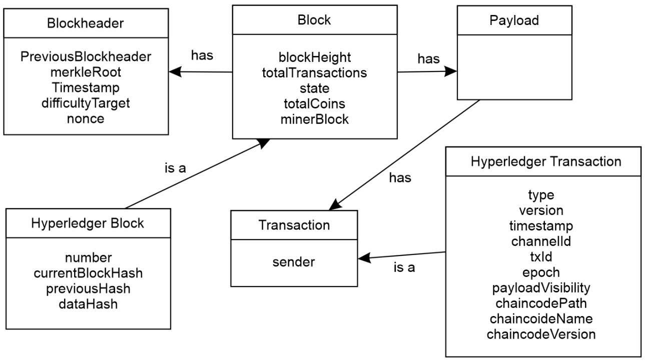

Figure 8. Excerpt from the BLONDiE Ontology. Adapted from Hector and Boris [2020]

according to the business rules corresponding to the application to be developed. Its properties include (1) domain, which stores the general area of application or target interest. When the ontology is used in an ontology-based system equipped with an ontology repository, its value will reference an instance of that repository. Otherwise, it will be a string with a textual description; (2) Purpose, stores a string with the textual description of a record in the domain specifically related to the transaction itself; (3) Application, stores a string with the textual description of the nature and objective of the application to be developed. This entity is connected to the Transaction entity from the BLONDiE ontology. In any other ontology for different DLT networks, the connection would be made with the entity equivalent to the transaction.

The Privacy class represents the privacy-related characteristics that make up a specific type of transaction. A transaction type can be linked to multiple instances of privacy, and privacy attributes may vary depending on the user's preferences and the type of application domain. The Privacy entity's properties are as follows: (1) Public Data, a data set that must be provided as necessary for identifying the user and correctly recording the transaction, i.e. is essential for the business rules. (2) Sensitive Data, a set of data that, although adding value to the information, can be omitted according to the user's preference. (3) Period, composed of its start and end; this attribute refers to the period during which the user wishes their sensitive data to be exposed.

The Actors class has its counterparts in the SEM ontology, but as TransTypO may be connected to other ontologies in different situations, its presence is necessary here. It supports the individuals involved in the process, which will be stored in the blockchain. The actors can be real people, software subroutines, or even devices.

The Self-Sovereign Identity class represents an instance of a user identity present in a decentralized repository. Its attributes are (1) ID, the data needed to identify the credential in the repository; (2) Service, the type of service to be used with the repository; (3) Authentication, the proof generated by the repository to be used for access registration in the application; (4) Public Key, an identification key for validating the authenticity of the user's identity; (5) Context, a general description of the environment in which the user's identity will be used.

The Thing class is the root of the ontology, and connected to it is the event, which, in this use case, is the object of interest of the business rules. The connection would be made with its main entity in any other business ontology.

The SEM ontology's Event class is an example of a connection point between the TransTypO ontology and the ontology representing the transaction's business rules. In this

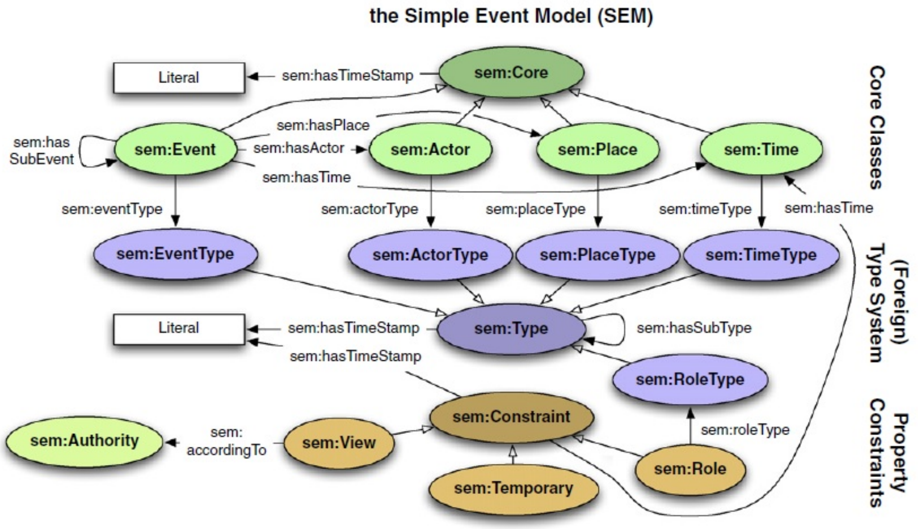

Figure 9. Simple Event Model (SEM) ontology. Adapted from van Hage *et al.* [2011]

example, the basic properties of an event are covered.

The Transaction class from the BLONDiE ontology is an example of a connection point between the TransTypO ontology and the ontology representing the DLT network used. This example covers the basic properties of a transaction in the Hyperledger Fabric ledger.

5 Experiments

Two simulations were conducted: the first was a simulation of a blockchain network for recording events in open spaces, and the second was a simulation of the initialization and connection of self-sovereign identity agents within the same scenario. Both simulations were executed in the same computational environment: a virtual environment running Ubuntu 20.04 on a Dell G7 computer with 16GB of RAM and an Intel Core i7 eighth-generation processor.

5.1 Simulation I

The chosen scenario for implementing ledger instances using Write and Read methods was to record attendance at urban outdoor events, such as parades, demonstrations, and rallies, in a controlled manner while preserving participants' identities with reliable registration. The smart contract, benchmarks, workloads, and tables with the results of this work are available at https://github.com/macs20/JISA2024. The outdoor event registration included the following attributes:

- **User ID** - Information representing the user's identification in a system for recording attendance at outdoor events. A pseudorandom number between 1 and 5000 was generated to simulate the choice of an ID.
- **Event Type** - The type of outdoor event in which attendance is recorded on the blockchain. A type of event was randomly selected from a predetermined list to simulate the choice of the event the user would attend.
- **Latitude** - Represents the first element to form the coordinate needed to establish the user's location during attendance registration. A valid latitude value was generated pseudo-randomly.
- **Longitude** - Represents the second element to form the coordinate. A valid longitude value was generated pseudo-randomly.
- **Date** - The complete date of attendance registration, including the timestamp. A valid, complete date value was pseudorandomly generated.

5.1.1 Metrics

Hyperledger Caliper evaluates the performance of blockchain systems using several key metrics, which provide a comprehensive understanding of the network's efficiency, scalability, and robustness. The metrics used in this work are:

- **Transactions Per Second (TPS)**: Measures the number of transactions the network can process per second. It is an important metric for understanding the network's throughput capacity.
- **Transaction Latency**: Refers to the time required for a transaction to be confirmed by the blockchain network. This includes the time from the transaction submission to its inclusion in a block and the network's eventual confirmation of that block.
- **Resource Usage**: Involves monitoring the consumption of computing resources by the blockchain network, such as CPU and memory usage. This metric is vital for assessing system efficiency and identifying potential performance bottlenecks.
- **Throughput**: The total capacity of the network to process transactions over a specific period. It differs from transactions per second, as it can also consider the network's ability to handle varying workloads over time.

These metrics aim to provide network developers and administrators with insight into the operational performance and efficiency of a blockchain deployment, allowing them to make optimized adjustments to improve the network's capacity and reliability.

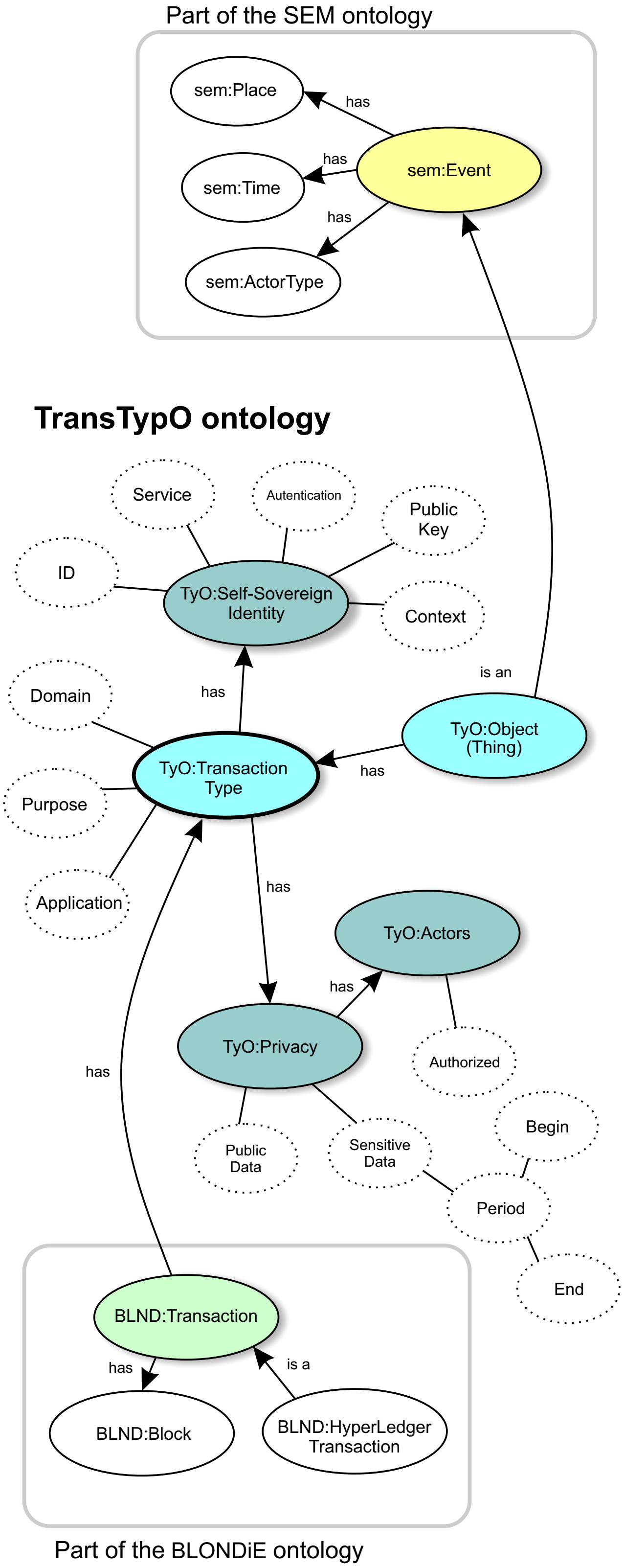

Figure 10. TransTypO Ontology Associated with Domain Ontologies of the Use Case - Silva *et al.* [2024]

5.1.2 Specifications

The selected version of Hyperledger Fabric was 2.5.6, and the Hyperledger Caliper version was 0.4.1. One network was implemented with a CA, and the other with manual certificate creation and management. The chaincode was written in JavaScript in this experiment, and the read and write operations were executed separately on each network. Hyperledger Caliper generated the workload and performance metrics collection, and the configured range for the request submission rate was between 50 and 200 transactions per second. Three test executions were performed: the first with 1,000 transactions, the second with 5,000 transactions, and the third with 10,000 transactions.

5.2 Simulation II

Simulation II simulated the initialization and connection between two self-sovereign identity agents. To measure the performance of these agents, one was the student and the other the event-organizing institution. Agents were created using the Credo framework, which is specific for self-sovereign identity, in TypeScript. The steps of the simulated process were:

- Initialization of the Student agent, the individual attending the event.
- Initialization of the USP agent, the University of São Paulo as the institution organizing the event.
- Creation of a connection invitation by the institution.
- Acceptance of the invitation by the Student.
- Establishment of a connection between the agents. We measured the time at two points: the time elapsed from the call to the Student's initialization and the time elapsed from the initial call to the establishment of the connection.

The above steps were executed 33 times in the same computational environment as Simulation I.

6 Results

The results obtained from the two simulations are presented in this section.

6.1 Results of Simulation I

To differentiate between the two networks, we will refer to the network implementing the Certificate Authority as the CA Network and the MSP Network as the one that does not (and that entails manual certificate creation and management).

Regarding the latency test for the function of recording attendance, as shown in the graph in Figure 11, for the execution with 1,000 transactions, the MSP Network had a latency of 1.03 seconds, and the CA Network had a latency of 0.88 seconds. For the execution with 5,000 transactions, the recorded latencies were 4.34 seconds for the MSP Network and 4.2 seconds for the CA Network. For the execution with 10,000 transactions, the MSP Network had a latency of 8.48 seconds, and the CA Network had a latency of 7.43 seconds.

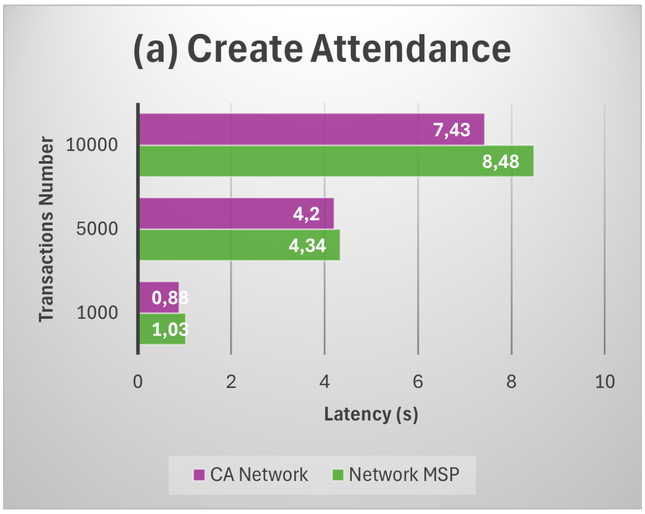

Figure 11. Latency Results for the Record Attendance Method.

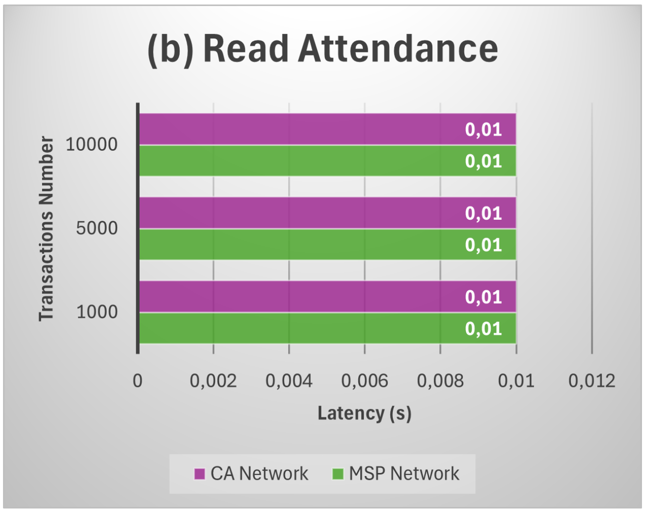

Figure 12. Latency Results for the Read Attendance Method.

When the function of reading an attendance record was executed, as shown in the graph of Figure 12, as a result of all executions, the result was 0.01 seconds for both networks.

Regarding throughput, for the function of recording attendance, as shown in the graph in Figure 13, for the execution with 1,000 transactions, the MSP Network achieved a throughput of 46.1 transactions per second. The CA Network achieved a throughput of 45.3 transactions per second. For the execution with 5,000 transactions, the throughputs recorded were 49 transactions per second for the MSP Network and 48.6 transactions per second for the CA Network. For the execution with 10,000 transactions, the MSP Network achieved a throughput of 49.3 transactions per second, and the CA Network achieved a throughput of 48 transactions per second.

When the function of reading an attendance record was executed, as shown in the graph in Figure 14, for the execution with 1,000 transactions, the MSP Network achieved a throughput of 61.2 transactions per second. The CA Network achieved a throughput of 64.1 transactions per second. For the execution with 5,000 transactions, the throughputs recorded were 64.9 transactions per second for the MSP Network and 64.8 transactions per second for the CA Network.

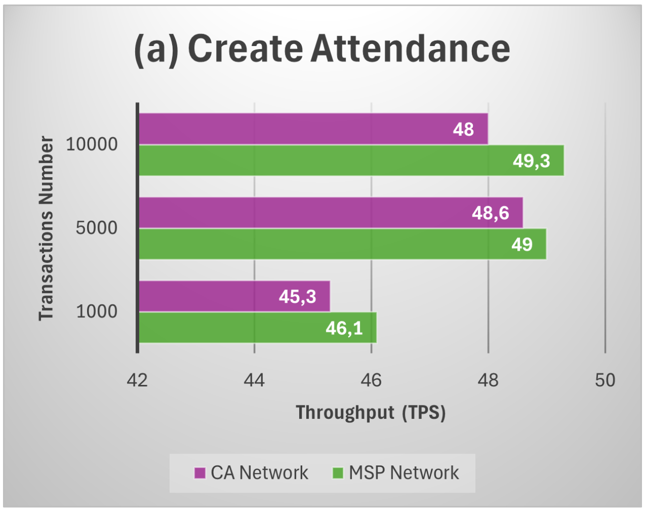

Figure 13. Throughput Metric Results for the Record Attendance Method.

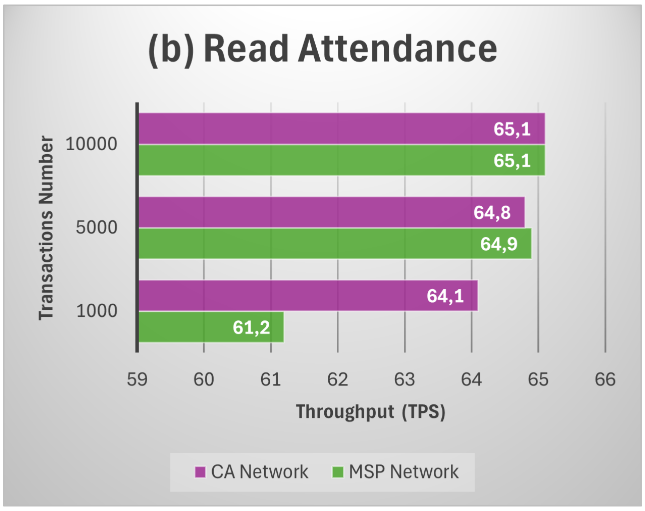

Figure 14. Throughput Results for the Read Attendance Method

For the execution with 10,000 transactions, the MSP Network achieved a throughput of 65.1 transactions per second, and the CA Network also achieved a throughput of 65.1 transactions per second.

In the measurement of CPU usage for the function of recording attendance, as shown in the graph in Figure 15, for the execution with 1,000 transactions, the MSP Network used 14.79% of the CPU capacity. In contrast, the CA Network used 15.32% of the CPU capacity. For the execution of 5,000 transactions, the MSP Network used 17.1% of the CPU capacity, while the CA Network used 18.12%. For the execution of 10,000 transactions, the MSP Network used 17.33% of the CPU capacity, while the CA Network used 17.39%.

When the function of reading an attendance record was executed, as shown in the graph in Figure 16, for the execution with 1,000 transactions, the MSP Network used 8.38% of the CPU capacity. In contrast, the CA Network used 8.43% of the CPU capacity. For the execution of 5,000 transactions, the MSP Network used 15.87% of the CPU capacity, while the CA Network used 16.04%. For the execution of 10,000 transactions, the MSP Network used 18.55% of the CPU capacity, while the CA Network used 18.76% of the CPU capacity.

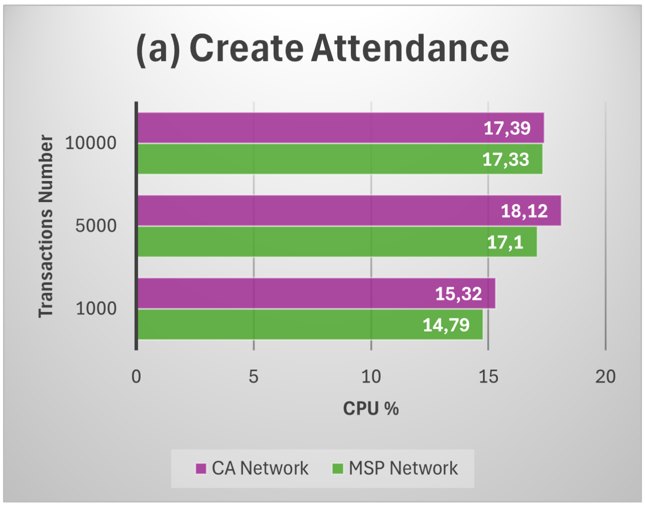

Figure 15. CPU Usage Results for the Record Attendance Method.

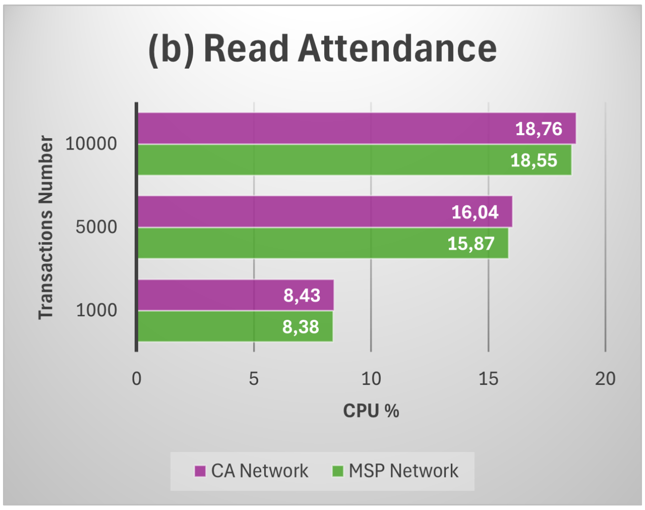

Figure 16. CPU Usage Results for the Read Attendance Method.

When measuring memory usage for the function of recording attendance, as shown in the graph in Figure 17, for the execution with 1,000 transactions, the MSP Network used 259 megabytes of memory. In contrast, the CA Network used 172 megabytes of memory. The MSP Network used 236 megabytes of memory to execute 5,000 transactions, while the CA Network used 155 megabytes of memory. The MSP Network used 197 megabytes of memory to execute 10,000 transactions, while the CA Network used 109 megabytes of memory.

When the function of reading an attendance record was executed, as shown in the graph in Figure 18, for the execution with 1,000 transactions, the MSP Network used 246 megabytes of memory. In contrast, the CA Network used 72.9 megabytes of memory. For the execution with 5,000 transactions, the MSP Network used 251 megabytes of memory, while the CA Network used 71.5 megabytes. For the execution with 10,000 transactions, the MSP Network used 249 megabytes of memory, while the CA Network used 68.1 megabytes.

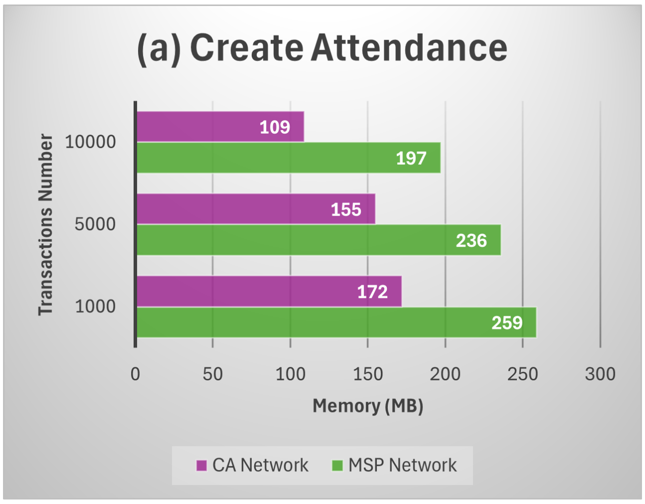

Figure 17. Memory Usage Results for the Record Attendance Method

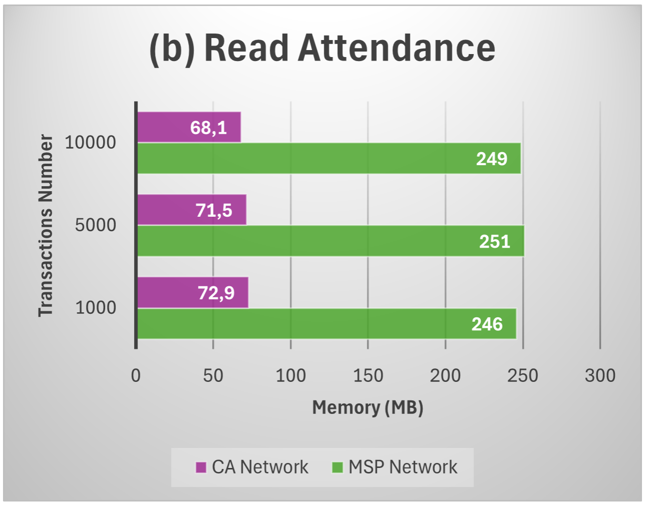

Figure 18. Memory Usage Results for the Read Attendance Method.

6.2 Results of Simulation II

The results of the simulation are listed in Table 2. Analyzing these results, it was possible to observe that the average time to initialize the Student's scheduling was 749.78 milliseconds, with a maximum time of 809.78 milliseconds and a minimum of 726.20 milliseconds. The variation between the maximum and minimum elapsed time is 12%.

The average total execution time, until the establishment of the connection between the agents, was 2007.13 milliseconds, with a maximum time of 2442.52 milliseconds and a minimum time of 1843.74 milliseconds. The variation between the maximum and minimum elapsed time is 32%. Figure 19 shows the elapsed time during the simulation executions.

7 Discussion

Based on the results obtained from the tests, it can be observed that during recording attendance, the latency values in seconds obtained by the CA Network were slightly lower in all executions. Latency gradually increased as the number of transactions sent in the test increased. The same did

Time for SSI Agents Initialization and Connection Established.		
	Initialize Student (milliseconds)	Total elapsed time (milliseconds)
Average Time	749,78043	2007,13389
Maximum Time	809,78004	2442,52127
Minimum Time	726,20503	1843,74656
Variation	12%	32%

Table 2. Results of Simulation II

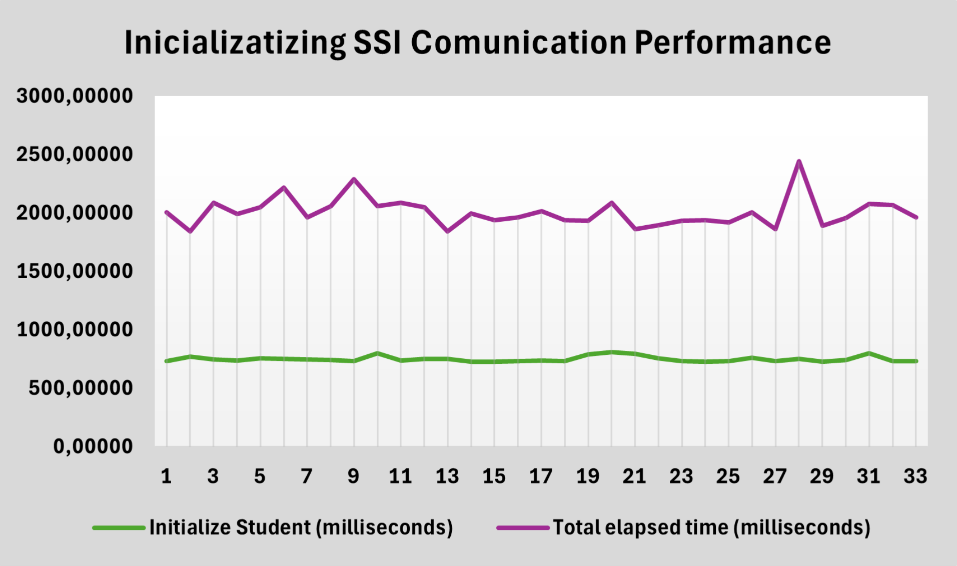

Figure 19. Performance Results of SSI Agents Initialization and Connection.

not occur during the reading of attendance records, where all latency values remained similar for any number of transactions.

When analyzing the data obtained from the throughput test, it can be observed that the MSP network achieved higher values in all executions to record attendance. Regarding this function, the number of transactions per second increased in the MSP Network as the number of transactions increased, with the increase from 5,000 to 10,000 transactions being more discrete than from 1,000 to 5,000 transactions. The same did not happen with the CA network, which experienced a slight reduction in throughput when the number of transactions increased from 5,000 to 10,000. Observing the data obtained from the reading attendance throughput tests, the most significant difference in values occurred when reading attendance for both networks with only 1,000 transactions, as the tests with 5,000 and 10,000 transactions showed similar values for both networks.

Regarding CPU usage data, it can be noted that, when performing the attendance record function, the values were slightly higher for the CA Network than for the MSP Network, except for the execution with 10,000 transactions, which recorded similar values. It can also be noted that with an increase in the number of transactions, this function did not show a significant gap between the two networks. This differs from the CPU usage values for the reading function, where a gradual increase in CPU usage can be observed as the number of transactions increases, and the values obtained between the two networks are close.

The memory usage data generated by the tests shows that when the recording attendance function was used, the CA network used less memory in all executions. Memory usage also decreased as the number of transactions increased.

Therefore, when choosing whether or not to adopt the CA in a project, the developer must consider two factors: (a) the volume of identities and certificates to be managed by the application and (b) the frequency with which the read and write methods will be executed.

Identity verification through a structured and automated process, such as CA, enhances security by reducing the risk of unauthorized access and identity fraud. However, this increased security requires a higher computational load. However, the network without CA offers a lighter and more straightforward approach, which can be more advantageous in scenarios where a high transaction rate and lower latency are priorities. From a scalability perspective, the CA-based system facilitates long-term digital identity management in large-scale applications. In contrast, the manual certificate issuance process may become impractical as the number of users and transactions grows.

When adopting SSI, the developer must consider the additional time it adds to the process and weigh the benefits this technology offers. It is important to note that the time attributed to the initialization of the Student Agent will occur on the end user's device. Regarding this value, the results showed that the time variation was smaller than the total time since the processing of the Institution Agent will take place in an environment with greater computational power in an actual situation.

8 Conclusion

This work proposes the ONSPRIDE framework, which can guide users from the planning phase of the blockchain service. It allows entities to efficiently incorporate their own business needs and rules into decentralized apps, allowing for the creation of privacy attributes applicable to blockchain networks that are aligned with the principles of Privacy by Design.

ONSPRIDE can be used in any application area with its domain ontologies as long as they cover the business rules or are extended for this purpose. In this context, developers should consider developing networks using a CA, a crucial structure for certificate management in the network, when the application requires complex certificate management. When this is not the case and the nature of the application allows for simplified certificate management, this work provides results that can support the decision on whether to adopt the CA.

To this end, the performance of a network created with a CA was compared to that of a network without this implementation, where certificates were issued and distributed manually. The tests showed similar results for latency, throughput, and CPU usage metrics, but for the Memory Usage metric, the network implemented with CA performed better.

In addition, simulations of the connection between the two SSI agents in the use case were conducted. The results suggest that developers considering the adoption of SSI should take into account the additional time required for this service.

In future work, we will conduct tests using hardware with greater computational capacity in scenarios with more complex structures, a higher number of transactions, and repetitions to establish a confidence interval. We will also consider other certificate issuers, hybrid approaches such as homomorphic encryption and zero-knowledge proofs (ZKP), including Hyperledger Indy, which implements ZKP for SSI, and the use of other types of databases for transaction creation. Further developments in this line of investigation will include a quantitative comparison with other blockchain platforms and enhancements in the SSI simulation for identity verification and issuance. Further developments in this line of investigation will include a quantitative comparison with other blockchain platforms, such as Ethereum and Stellar, and enhancements in the SSI simulation for identity verification and issuance. Finally, we intend to conduct tests in an actual deployment environment, with feedback from users and administrators, as well as a simulation evaluating the impact of SSI on security and network load.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Rodolfo Meneguette gostaria de agradecer a Fundação de Amparo à Pesquisa do Estado de São Paulo (número 2022/00660-0 e 2020/07162-0) pelo suporte financeiro.

Funding

This research was funded by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) e Fundação de Amparo à Pesquisa do Estado de São Paulo.

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 (DOI:10.54499/UIDB/50021/2020).

This work is supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 952226, project BIG (Enhancing the research and innovation potential of Técnico through Blockchain technologies and design Innovation for Social Good).

This work was supported by the CloudStars project, funded by the European Union's Horizon research and innovation program under grant agreement number 101086248.

Authors' Contributions

The authors equally contributed to the elaboration of this paper. All the authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The programming code, datasets, and reports generated and analyzed during the current study are available at https://github.com/macs20/JISA2024.

References

Aljeraisy, A., Barati, M., Rana, O., and Perera, C. (2021). Privacy laws and privacy by design schemes for the internet of things: A developer's perspective. *ACM computing surveys*, 54(5):1–38. DOI: 10.1145/3450965.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., *et al*. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15. DOI: 10.1145/3190508.3190538.

Bawden, D. and Robinson, L. (2020). "the dearest of our possessions": Applying floridi's information privacy concept in models of information behavior and information literacy. *Journal of the Association for Information Science and Technology*, 71(9):1030–1043. DOI: 10.1002/asi.24367.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, 284(5):34–43. DOI: 10.1038/scientificamerican0501-34.

Bu, F., Wang, N., Jiang, B., and Liang, H. (2020). "privacy by design" implementation: Information system engineers' perspective. *International journal of information management*, 53:102124. DOI: 10.1016/j.ijinfomgt.2020.102124.

Carlozo, L. (2017). What is blockchain? *Journal of Accountancy*, 224(1):29. DOI: 10.61557/zcow6716.

De Kruijff, J. and Weigand, H. (2017). Towards a blockchain ontology. *The Netherlands, pdfs. semanticscholar*. Available at: `https://www.list.lu/fileadmin/files/Event/sites/tudor/files/Training_Center/OTHERS/VMBO2017_paper_5.pdf`.

de Souza, E. A., Villa, R. M., and Gonzalez, E. T. Q. (2020). Privacy and autonomy in the big data era/privacidade e autonomia na era de big data. *Acta scientiarum. Human and social sciences*, 42(3).

Foundation, L. (2021a). Hyperledger - open soucer blockchain technologies. Available at: `https://www.hyperledger.org/`. Accessed in 20/10/2021.

Foundation, L. (2021b). Hyperledger fabric- hyperledger foundation. Available at: `https://www.hyperledger.org/use/fabric`. Accessed in 20/10/2021.

Foundation, L. (2022). Hyperledger caliper- hyperledger foundation. Available at: `https://www.hyperledger.org/use/caliper`. Accessed in 05/03/2022.

Gayathri Santhosh, M. and Reshmi, T. (2023). Enhancing pki security in hyperledger fabric with an indigenous certificate authority. In *2023 IEEE International Conference on Public Key Infrastructure and its Applications (PKIA)*, pages 1–5. DOI: 10.1109/PKIA58446.2023.10262412.

Giannopoulou, A. and Wang, F. (2021). Self-sovereign identity. *Internet policy review*, 10(Issue 2). DOI: 10.14763/2021.2.1550.

Hector, U.-R. and Boris, C.-L. (2020). Blondie: Blockchain ontology with dynamic extensibility. DOI: 10.48550/arXiv.2008.09518.

Kuzlu, M., Pipattanasomporn, M., Gurses, L., and Rahman, S. (2019). Performance analysis of a hyperledger fabric blockchain framework: Throughput, latency and scalability. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 536–540. DOI: 10.1109/Blockchain.2019.00003.

Liu, Y., Lu, Q., Paik, H.-Y., Xu, X., Chen, S., and Zhu, L. (2020). Design pattern as a service for blockchain-based self-sovereign identity. *IEEE software*, 37(5):30–36. DOI: 10.1109/ms.2020.2992783.

Lux, Z. A., Beierle, F., Zickau, S., and Göndör, S. (2019). Full-text search for verifiable credential metadata on distributed ledgers. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 519–528. IEEE. DOI: 10.1109/iotsms48152.2019.8939249.

Maschi, L. F. C., Pinto, A. S. R., Meneguette, R. I., and Baldassin, A. (2018). Data summarization in the node by parameters (dsnp): Local data fusion in an iot environment. *Sensors*, 18(3). DOI: 10.3390/s18030799.

Melo, C., Oliveira, F., Dantas, J., Araujo, J., Pereira, P., Maciel, R., and Maciel, P. (2022). Performance and availability evaluation of the blockchain platform hyperledger fabric. *The Journal of Supercomputing*, 78(1):12505–12527. DOI: 10.1007/s11227-022-04361-2.

Mor, P., Tyagi, R. K., Jain, C., and Verma, D. K. (2024). Enhanced hyperledger fabric network set-up for remittance and settlement process. In Goyal, S. K., Palwalia, D. K., Tiwari, R., and Gupta, Y., editors, *Flexible Electronics for Electric Vehicles*, pages 157–167, Singapore. Springer Nature Singapore. DOI: 10.1007/978-981-99-4795-$9_1$5.

Silva, M. A. C., Nakamura, L. H. V., Veiga, L., and Meneguette, R. (2024). Transtypo - transaction type ontology for distributed ledger technology networks with privacy rules observance. In *2024 19th Iberian Conference on Information Systems and Technologies (CISTI)*. Available at: `http://www.transtypo.org/TransTypO/CISTI_2024_Silva_et_al.pdf`.

van Hage, W. R., Malaisé, V., Segers, R., Hollink, L., and Schreiber, G. (2011). Design and use of the simple event model (sem). *Journal of Web Semantics*, 9(2):128–136. Provenance in the Semantic Web. DOI: 10.1016/j.websem.2011.03.003.

Vukoli, M. (2017). Rethinking permissioned blockchains [c]. In *ACM Workshop. ACM*. DOI: 10.1145/3055518.3055526.

Wang, C. and Chu, X. (2020). Performance characterization and bottleneck analysis of hyperledger fabric. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1281–1286. DOI: 10.1109/ICDCS47774.2020.00165.

Zhang, R., Xue, R., and Liu, L. (2019). Security and privacy on blockchain. *ACM computing surveys*, 52(3):1–34. DOI: 10.1145/3316481.