The Role-Artifact-Function Framework for Understanding Digital Identity Models

Frederico Schardong[1,2(✉)] and Ricardo Custódio[1]

[1] Universidade Federal de Santa Catarina, Florianópolis, Brazil
ricardo.custodio@ufsc.br
[2] Instituto Federal do Rio Grande do Sul, Rolante, Brazil
frederico.schardong@rolante.ifrs.edu.br

Abstract. Identity and Access Management (IAM) is a crucial aspect of online interactions, and researchers and practitioners have been exploring this area since the early days of the Internet. Over the years, many digital identity protocols and standards were created. While OAuth 2.0 and OpenID Connect (OIDC) are currently the most widely used and mature protocols, the emerging concept of Self-Sovereign Identity (SSI) and its underlying protocols promise greater user privacy and control. Although much emphasis has been placed on new protocols and implementations, the ontological and formal understanding of digital identity models still needs to be addressed. In this paper, we make three contributions: (i) introduce the Role-Artifact-Function (RAF) framework, which comprises a meta-metamodel and a metamodel; (ii) describe and compare digital identity models through the instantiation of the RAF metamodel; and (iii) present significant insights, including the conceptual equivalence between the established family of x509-based protocols and SSI, as well as the correlation between the chronological evolution of identity models and their increasing linguistic complexity.

Keywords: Identity and Access Management · Digital Identity · Self-Sovereign Identity · Metamodel

1 Introduction

During the early days of the Internet, Service Providers (SPs) established Identity Providers (IdPs) to authenticate users and offer customized services. This resulted in usability challenges, with users frequently resorting to weak password reuse across platforms, thus increasing vulnerabilities [27]. Third-party IdPs emerged, enabling users to register with a select few IdPs for access to various web services. This model concentrates valuable private information within a limited number of large repositories. In contrast, Self-Sovereign Identity (SSI) presents a digital identity paradigm centered on individual autonomy over their data. SSI empowers individuals as sovereign data guardians, granting them authority over its governance, storage, and sharing permissions [2].

W. Maass et al. (Eds.): ER 2024, LNCS 15238, pp. 377–395, 2025.
https://doi.org/10.1007/978-3-031-75872-0_20

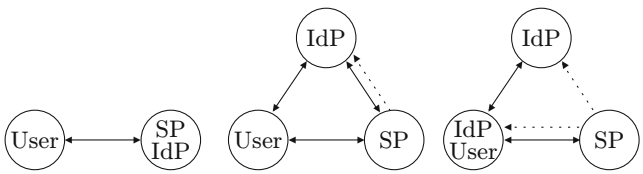
Existing models lack formal descriptions on a shared ground, which will facilitate their comparison and evaluation. The increasing development of SSI has introduced new challenges and complexities to the field of IAM [2,18], making formal matters more urgent [29]. Formal studies can provide a common language and framework for evaluating digital identity models, allowing for more effective communication between researchers, practitioners, and policymakers. Additionally, such studies can help identify the strengths and weaknesses of different models, leading to the development of more robust and effective IAM solutions.

The IAM literature has often conceptualized digital identity models using a bottom-up approach, starting from real-world instances, such as the protocols and concrete implementations [29]. In contrast, this paper adopts a town-down approach, offering a structured methodology to describe and comprehend digital identity models. We introduce the Role-Artifact-Function (RAF) framework, which comprises a meta-metamodel and a metamodel. The latter is instantiated to formally describe three IAM models: (i) offline outsourced IdP; (ii) online outsourced IdP; and (iii) SSI. While this work does not cover the instantiation of these models to concrete identity systems or protocols, we offer high-level protocols that illustrate the foundational concepts of real-world protocols and applications implementing these models.

The remainder of this paper is organized as follows. In Sect. 2, we provide the required background on digital identity models. In Sect. 3, we introduce the RAF framework. In Sect. 4, we describe existing models using RAF, creating a shared ground to compare them in Sect. 5. We explore related work in Sect. 6 and conclude in Sect. 7.

2 IAM Preliminaries: Digital Identity Models

Different IAM models, also called identity models, can be categorized based on the responsibilities and interactions of the roles involved, namely user, IdP, and SP. Based on this classification approach, current identity systems and protocols can be generally categorized into three models, namely centralized (Fig. 1a), outsourced IdP (Fig. 1b), and SSI (Fig. 1c).



(a) Centralized. (b) Outsourced IdP. (c) SSI.

Fig. 1. Current digital identity models. Circles represent entities, labels represent roles, solid lines represent interactions, and dashed lines represent trust [29].

2.1 Centralized

In the early days of the Internet, SPs had no choice but to develop IdPs to authenticate consumers and offer customized products and services. Nevertheless, this resulted in the establishment of *centralized* authorities that served as both SPs and IdPs. Users frequently resorted to reusing weak passwords across multiple systems, resulting in many vulnerabilities and usability issues [27].

In considering adopting the centralized model, it is pertinent to acknowledge the establishment of the server-client architecture as standardized in 1996 through the Hypertext Transfer Protocol (HTTP) version 1.0. This initial version featured a rudimentary method for client identification and authentication, reliant on a straightforward comparison of usernames and passwords lacking cryptographic safeguards. Subsequently, in 1997, Digest Access Authentication was introduced to enhance security. This method introduced a nonce-based challenge, requiring the client to hash its response using MD5 before transmitting [11].

2.2 Outsourced IdP

The next logical evolution was to replace the centralized authorities with third-party IAM solutions, *i.e.*, IdPs. This paradigm shift allowed users to engage with a limited number of IdPs to access a diverse array of web-based services. While this approach streamlines the management of multiple identifiers and passwords for users, it also consolidates significant volumes of sensitive personal data within a few major silos. To standardize the interactions among IdPs, SPs, and end-users, token exchange protocols like SAML [14], OAuth 2.0 [13], and OpenID Connect [24] have been introduced. These protocols are used daily by billions of users through industry-leading IdPs like Google and Meta.

The OAuth 2.0 protocol introduced standardized procedures facilitating SPs to interact with IdPs[1] to obtain access to digital identity attributes or perform actions on behalf of users. However, its primary focus on authorization rather than authentication leads to the exclusion of crucial digital identity elements. OAuth 2.0 does not regulate how SPs should request fundamental digital identity attributes like name and email, leaving this determination to IdPs. Moreover, SPs cannot set a minimum Level of Assurance (LoA) acceptable for authenticating users. In response to these shortcomings, the OpenID Connect (OIDC) protocol was conceived as an extension of OAuth 2.0 [24].

2.3 Self-sovereign Identity

Self-Sovereign Identity (SSI) is a new approach to digital identity that emphasizes individual sovereignty over their data. Unlike previous identity models where individuals were viewed as mere users, SSI recognizes individuals as the sovereign owners of their data, granting complete control over it. They can manage and store their data and decide with whom to share it [29].

[1] OAuth 2.0 refers to SP as client and IdP as authorization and resource servers.

Christopher Allen [2] proposed ten principles an SSI system must fulfill. These principles, known as the ten principles of SSI, outline the characteristics of an SSI system and are as follows: (i) users must have an independent *existence*; (ii) users must *control* their identities; (iii) users must have *access* to their own data; (iv) systems and algorithms must be *transparent*; (v) identities must be *persisted*; (vi) information and services must be *portable*; (vii) identities should be *interoperable*; (viii) users must *consent* to identity use; (ix) disclosure of claims must be *minimized*; and (x) the rights of users must be *protected*.

Let us discuss the two most relevant principles for brevity: minimization and access. The *minimalization* principle enhances user privacy about attribute sharing. It allows users to share statements about their attributes, such as their age, rather than sharing their birthdate. This approach is advantageous because less information is shared and potentially retained by the SP. Zero-Knowledge Proof (ZKP) is a popular method employed to share predicates over one or more attributes of a credential without compromising authenticity or integrity [19,20,28]. In brief, ZKP allows users to prove to SPs that some properties about attributes hold without revealing them [29]. The *access* principle means that users have complete control over their data without relying on IdPs to manage them. Users have autonomy over storing, managing, and sharing their data. ZKP also enables the implementation of the access principle by demonstrating the credential's validity without disclosing the credential to the SP [5]. Lastly, it is worth mentioning that trust is another critical aspect of SSI. Although individuals have the autonomy to issue their credentials, others are free to distrust them.

Let us move on from theoretical concepts to more practical topics. In the realm of SSI literature, Verifiable Credentials (VCs) [31] and Decentralized Identifiers (DIDs) [25] are crucial W3C standards [7,21,29]. The VC standard defines the terminology used in SSI, explaining that a *claim* is any statement made about a *subject*, and a *credential* is a collection of one or more assertions made about a subject. A VC is a credential that guarantees the credential's integrity, as well as the issuer's identification and non-repudiation. It contains a revocation list or another method of revocation. A *verifiable claim* or *Verifiable Presentation (VP)* is a tamper-resistant claim derived from a verifiable credential. Additionally, the DID specification outlines how SSI users can issue, update, and revoke identifiers independently without relying on other entities. These identifiers, digital wallets, and protocols, such as DIDComm [12] and DID Exchange [33] for exchanging personal information, are essential for SSI applications.

3 Role-Artifact-Function Framework

In this section's first part, the RAF meta-metamodel establishes a high-level abstract syntax, or a meta-language [17]. In the second part of this section, we introduce the RAF metamodel, an instance of the RAF meta-metamodel.

3.1 RAF Meta-Metamodel

The RAF meta-metamodel is grounded in set theory and is defined as follows.

Definition 1. *The RAF meta-metamodel is a triple* $(\mathbb{R}, \mathbb{A}, \mathbb{F})$*, where:*

- $\mathbb{R}$ is the set of *roles* played in the digital identity model.
- $\mathbb{A}$ is the set of *computational artifacts* employed by the roles.
- $\mathbb{F}$ is the set of *functions* in the metamodel with domain and co-domain in $\mathbb{R} \cup \mathbb{A}$. Subscript symbols represent the roles that can execute said function.

The *roles* are abstract entities with distinctive computational artifacts and functions that the real-world actors in an IAM system can play. For instance, in SSI, the end-user acts as an IdP and user. The *computational artifacts* represent algorithms, data structures, and other IAM-related artifacts. The *functions* are operations executed by roles. To instantiate this meta-metamodel into a metamodel, we need to provide instances for the sets $\mathbb{R}, \mathbb{A}, \mathbb{F}$ as described next.

3.2 RAF Metamodel

Definition 2. *The RAF metamodel is defined as an instance of the RAF meta-metamodel* $(\mathbb{R}, \mathbb{A}, \mathbb{F})$*, where:*

- $\mathbb{R}$=$\{s,\ i,\ v\}$ *represents the roles within the model.*
- $\mathbb{A}$=$\{\mathcal{C},\ \mathcal{R},\ \mathcal{M},\ \mathcal{P},\ \mathcal{L}\}$ *denotes the computational artifacts.*
- $\mathbb{F}$=$\{issue_i,\ revoke_i,\ status_{s,v},\ trust_v,\ authn_s,\ authz_s,\ request_v,\ present_s,\ verify_v\}$ *encompasses the set of functions.*

Next, we will describe each component in detail.
Roles ($\mathbb{R}$):

- s: subject, *i.e.*, the user, with a set of attributes $s_A \subset A$ (*e.g.*, age and name) where A is the set of all possible attributes[2].
- i: issuer, *i.e.*, the IdP.
- v: verifier, *i.e.*, the SP.

Computational Artifacts ($\mathbb{A}$):

- $\mathcal{C}$: set of credentials, and each credential $c \in \mathcal{C}$ asserts a set of attributes $c_A \subset s_A$ about subject s[3]. Each attribute $a \in c_A$ has an associated value, denoted by c_a.
- $\mathcal{R}$: set of revocation registries, where each registry $r_c \in \mathcal{R}$ associated with a credential c is associated with a boolean value indicating whether the credential c is valid.
- $\mathcal{M}$: general-purpose machine, *i.e.*, a Turing machine.
- $\mathcal{P}$: program for a general-purpose machine $\mathcal{M}$, *i.e.*, a tape with instructions for the Turing machine.
- $\mathcal{L}$: language for statements, *e.g.*, predicate or first-order logic, where $\mathcal{L}(c \cdot r_c)$ is a statement about a credential c and a revocation registry r_c in $\mathcal{L}$.

[2] We delegate the specifics, such as the value of attributes or how they change over time, to the instances of the metamodel.
[3] The expiration of credentials based on time is excluded for simplicity of abstraction.

Functions ($\mathbb{F}$):

- $issue_i : \{s\} \times \wp(s_A) \nrightarrow \mathcal{C} \times \mathcal{R}$—issuing a credential by issuer i takes the subject s, a subset of their attributes $s_{A'} \subset s_A$, and generates a credential $c \in \mathcal{C}$ with attributes $c_A = s_{A'}$ and a revocation record $r \in \mathcal{R}$.
- $revoke_i : \mathcal{C} \times \mathcal{R} \nrightarrow \mathcal{R}$—given a credential $c \in \mathcal{C}$ and its associated revocation record $r \in \mathcal{R}$, this function updates the record to indicate that c is revoked, producing a new revocation record $r' \in \mathcal{R}$.
- $status_{s,v} : \mathcal{R} \rightarrow \{\texttt{true}, \texttt{false}\}$—determines the revocation status of a credential c. If the revocation record r_c in the set $\mathcal{R}$ indicates that c is revoked, it returns `true`; otherwise, it returns `false`.
- $trust_v : \{i\} \rightarrow \{\texttt{true}, \texttt{false}\}$—the trust function of verifier v returns a boolean indicating whether it trusts issuer i.
- $request_v : \wp(\mathcal{L}(\mathcal{C} \cup \mathcal{R})) \nrightarrow \{s, i\}$—verifier v can inquire subject s or issuer i to provide a program $\mathcal{P}$ that meets the statements in the language $\mathcal{L}(\mathcal{C} \cup \mathcal{R})$.
- $present_{s,i} : \wp(\mathcal{C}) \times \wp(\mathcal{R}) \times \wp(\mathcal{L}(\mathcal{C} \cup \mathcal{R})) \nrightarrow \{\mathcal{P}\}$—presenting function takes sets of credentials, revocation registries, and statements as input and produces a program $\mathcal{P}$. The execution of $\mathcal{P}$ in $\mathcal{M}$ is designed to confirm the veracity of the provided statements without compromising the privacy of the associated credentials. Specifically, it ensures that no additional information about the credentials is disclosed.
- $authn_s : \{\mathcal{P}\} \rightarrow \{\texttt{true}, \texttt{false}\}$—the authentication function of the subject s takes a program $\mathcal{P}$ as input and returns a boolean value indicating whether the subject is the owner of the credentials expressed in $\mathcal{P}$.
- $authz_s : \{\mathcal{P}\} \times \{v\} \rightarrow \{\texttt{true}, \texttt{false}\}$—subject s authorization decision function receives a program $\mathcal{P}$ and a verifier v and returns `true` if program $\mathcal{P}$ can be shared with verifier v or `false` otherwise.
- $verify_v : \wp(\mathcal{L}(\mathcal{C} \cup \mathcal{R})) \times \{\mathcal{P}\} \rightarrow \{\texttt{true}, \texttt{false}\}$—verification function returns `true` if the execution of program $\mathcal{P}$ in $\mathcal{M}$ satisfies the set of statements in language $\mathcal{L}$ about credentials and revocation records and `false` otherwise.

As we conclude our exploration of the RAF metamodel, we focus on its practical applications in the next section. We apply the RAF metamodel to two pivotal identity models: outsourced IdP and SSI. This shift is a move from understanding the foundational elements of the RAF framework to seeing it in action, highlighting how it can be used to break down and analyze specific identity management approaches. By examining different digital identity models through the RAF lens, we aim to showcase the framework's utility in describing the nuances that transverse identity systems.

4 Digital Identity Models Through RAF

In this section, we instantiate the RAF metamodel into the outsourced IdP and SSI models. This process involves defining concrete instances of the abstract components of the RAF metamodel. For the sake of simplicity, we instantiate the most abstract and general elements of the metamodel, *i.e.*, we provide an instance of the language $\mathcal{L}$ and the function $present_s$ for each model. We demonstrate that these instances meet the functional requirements defined for each

digital identity model. This study does not instantiate these models to actual digital identity protocols, but we provide high-level protocols that demonstrate the fundamental notions of real-world protocols implementing these models.

Firstly, we decompose the outsourced IdP model into two: offline and online. The former pertains to outsourced IdP interactions in which the IdP plays no active role in transmitting personal information when the end-user utilizes their credentials, which improves privacy. In contrast, the latter involves the IdP actively engaging in this process, reducing the end-user's privacy but easing the burden of storing and managing their credentials.

4.1 Offline Outsourced IdP Model

This section delineates the Offline Outsourced Identity Provider Model (OffOIdPM), a paradigm that allows the credential presentation to occur asynchronously. The model revolves around issuing a credential and a corresponding revocation record to the subject by the IdP, facilitating a verification process that does not necessitate continuous online access to the IdP. It has been a cornerstone of the Internet through the x509 digital certificate [15] and its family of protocols [1,3,4,16,22,26]. Definitions 3 and 4 lay the foundation for this model, which is further explored through Lemma 1 and Theorem 1.

Definition 3. *In the OffOIdPM, the issuer i issues a credential c and a revocation record r_c to the subject s through executing the $issue_i$ function.*

Definition 4. *In the OffOIdPM, subject s shares statements about themselves in language $\mathcal{L}(\mathcal{C} \cup \mathcal{R})$ with verifier v without the involvement of issuer i.*

Let us instantiate the RAF metamodel to describe the OffOIdPM.

Lemma 1 ($\mathcal{L}^{\alpha}$ as an instance of $\mathcal{L}$). *In the OffOIdPM, the set of statements regarding credentials and their revocation status can be represented by the language $\mathcal{L}^{\alpha}$, which is an instance of the RAF metamodel's abstract language $\mathcal{L}$.*

Proof of Lemma 1. Proof by construction. The language $\mathcal{L}^{\alpha}$ is a regular language over the alphabet $\Sigma = \{c, r_c\}$, and can be defined by the regular expression $\mathcal{L}^{\alpha} = (cr_c)^*$. This formulation establishes $\mathcal{L}^{\alpha}$ as an instance of the abstract language $\mathcal{L}$, applicable to modeling offline credential verification processes. □

Corollary 1. *The regular language $\mathcal{L}^{\alpha}$ encompasses the empty string $\{\varepsilon\}$ and any sequence of credential c and its revocation record r_c. This regularity confirms that $\mathcal{L}^{\alpha}$ aligns with the formal requirements of $\mathcal{L}$, demonstrating its suitability for modeling credential and revocation record handling in the specified model.*

Lemma 2 ($present_s^{\alpha}$ as an instance of $present_s$). *In the OffOIdPM, the function that produces programs $\mathcal{P}^{\alpha}$ to prove statements in $\mathcal{L}^{\alpha}$ is $present_s^{\alpha}$, which is a particular instantiation of RAF metamodel's abstract function $present_s$.*

Proof of Lemma 2. Proof by construction. The function $present_s^\alpha$ first ensures by a matching operation that the inputted credentials and revocation records are those referenced by the inputted statement in language $\mathcal{L}^\alpha$. Since the statements in language $\mathcal{L}^\alpha$ are plain sequences of credentials and revocation records, the program $\mathcal{P}^\alpha$ is a sequence of instructions to output the inputted statements. □

Theorem 1. *The instantiation of the RAF metamodel's $issue_i$ function following Definition 3 is trivial and is thus omitted. Language $\mathcal{L}^\alpha$ and function $present_s^\alpha$ are explicitly defined to align with the requirements specified in Definition 4. Together, these elements provide a complete description of the OffOIdPM.*

Proof. We prove Theorem 1 by construction using language $\mathcal{L}^\alpha$ and function $present_s^\alpha$. We present a high-level protocol that instantiates the OffOIdPM and describes the flow of information respecting Definitions 3 and 4. We refrain from instantiating other functions of the RAF metamodel as their implementation is straightforward. The protocol is depicted in Fig. 2 and proceeds as follows:

1. The issuer i generates a credential c and its revocation record r_c through the $issue_i$ function, dispatching both to the subject s.
2. The verifier v requests the credential c and its revocation record r_c from the subject s utilizing the $request_v$ function, specifying the desired language representation $\mathcal{L}^\alpha(c \cdot r_c)$.
3. Upon request, the subject s constructs program $\mathcal{P}^\alpha$ through the $present_s^\alpha$ function, embedding c and r_c within $\mathcal{L}^\alpha$ to fulfill the verifier's request, and transmits $\mathcal{P}^\alpha$ after evaluating $authz_s(P^\alpha, v)$ as `true`.
4. Authentication of ownership over program $\mathcal{P}^\alpha$ is achieved by s through the $authn_s$ function, establishing credibility in the presented credentials.

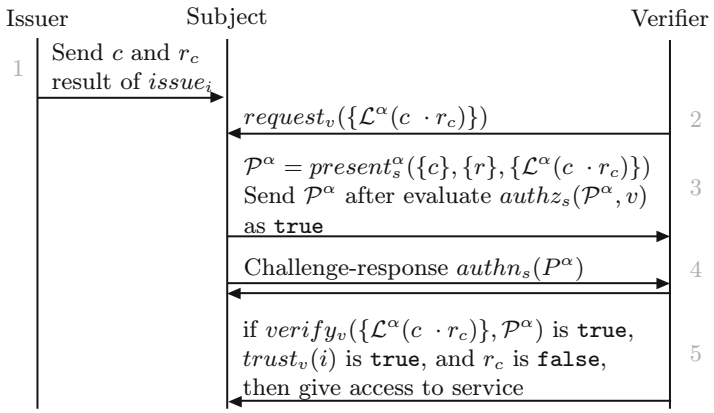


Fig. 2. The RAF-described OffOIdPM through a high-level protocol: the flow of information between issuer i, subject s, and verifier v. Numbers denote protocol steps, and arrows represent messages.

5. The verifier v grants access if it validates the received program $\mathcal{P}^{\alpha}$ against the requested information, the issuer's trustworthiness, and the credential's validity based on the revocation record, completing the verification process.

This high-level protocol exemplifies how the RAF metamodel's roles, computational artifacts, and functions can be instantiated to implement the OffOIdPM under Definitions 3 and 4. □

Our OffOIdPM model's description and the high-level protocol closely resemble the x509 certificate system and their related protocols [15]. For example, issuing x509 certificates, especially within national Public-Key Infrastructure (PKI) systems [1], involves rigorous verification steps to confirm the accuracy of subject attributes, akin to our $issue_i$ function. Similarly, the Automatic Certificate Management Environment (ACME) protocol allows internet servers to secure x509 certificates for domain names [3], another practical application of the $issue_i$ function. For revocation, the Certificate Revocation List (CRL) [4] serves as an example of revocation records R, with the Online Certificate Status Protocol (OCSP) [26] acting as an instance of the $status_v$ function. In our description, we emphasize offline verification, similar to how CRL might accompany a certificate without needing issuer contact due to both being digitally signed. Authentication is facilitated through the Certificate Management Protocol (CMP), a challenge-response system verifying private key ownership [16]. Finally, the Certificate Management over CMS protocol illustrates the $revoke_i$ function, showing how certificates are efficiently revoked [22].

The proposed high-level protocol and the digital certificate examples encounter two main problems: privacy and usability. Privacy is compromised because subjects cannot choose which credential attributes to share. Usability must improve since individuals must always have their credentials for verification. The privacy concern can be addressed with cryptographic schemes that enable selective attribute disclosure [23], which we present in Sect. 4.2. However, the usability issue is fundamental to the model and necessitates changing how information flows, which we explore in Sect. 4.3.

4.2 Self-sovereign Identity

The Self-Sovereign Identity (SSI) model is portrayed in the literature as a revolutionary identity model that gives individuals control over their credentials, including who can access them and how they are used [2,29,31]. However, it is simply a specialization of the OffOIdPM model, as seen below.

Definition 5. *In the SSI model, issuer i issues a credential c and a revocation record r_c to subject s by executing the $issue_i$ function.*

Definition 6. *In the SSI model, subject s shares statements about themselves in language $\mathcal{L}(\mathcal{C} \cup \mathcal{R})$ with verifier v without the involvement of issuer i.*

Definition 7. *In the SSI model, the language $\mathcal{L}(\mathcal{C} \cup \mathcal{R})$ must allow to: (i) express the disclosure of any subset of attributes of a credential c; and (ii) express properties and relations of any subset of attributes of a credential c.*

Note that Definitions 5 and 6 are copies of Definitions 3 and 4 from OffOIdPM framed for SSI. Thus, SSI is an extension of OffOIdPM that includes an extra bound. We formally describe SSI instantiating the RAF metamodel as follows.

Lemma 3 ($\mathcal{L}^\beta$ as an instance of $\mathcal{L}$). *In the SSI model, the set of statements regarding credentials and their revocation status can be represented by the language $\mathcal{L}^\beta$, which is an instance of the RAF metamodel's abstract language $\mathcal{L}$.*

Proof of Lemma 3. Proof by construction. The language $\mathcal{L}^\beta$ can be created using the First-Order Logic (FOL) constructs to encapsulate SSI's entities, relationships, and processes. The syntax and semantic integration of RAF with FOL forms language $\mathcal{L}^\beta$ as follows:

- Variables: Let RAF roles and computational artifacts denote the variables of $\mathcal{L}^\beta$.
- Predicates: Let RAF functions be incorporated as FOL predicates by aggregating the inputs and outputs of RAF functions as parameters of the FOL predicates. Non exhaustive list of examples:
  - $issue_i(s, s_A, c, r_c)$: reflects the RAF function $issue_i$, where issuer i issues credential c with attributes s_A and revocation record r_c to subject s. This predicate represents the action of credential issuance.
  - $trust_v(i, \texttt{true})$: reflects the RAF function $trust_v$ indicating that verifier v trusts issuer i.

  Moreover, auxiliary predicates to access and manipulate properties of RAF roles and computation artifacts are defined as follows:
  - $has(a, b)$: indicates if entity a posses, or contains entity b.
  - $comp(op, a, b)$: indicates if the boolean operator op (*e.g.*, $<$, $\geq$, $=$, $\neq$, $\Leftrightarrow$, etc.) between the operands a and b is `true`.
- Functions: No function is required besides standard FOL logical connectives and quantifiers. □

Corollary 2 (Direct mapping of $\mathcal{L}^\beta$ to RAF.). *By directly mapping RAF functions into FOL predicates, this construction ensures that the language $\mathcal{L}^\beta$ accurately reflects the operational semantics of the RAF framework. This direct mapping affirms the language's completeness in representing SSI dynamics.*

Language $\mathcal{L}^\beta$ enables the creation of comprehensive statements tailored for the SSI model, in alignment with Definition 7. To illustrate the capability of $\mathcal{L}^\beta$ in formulating such statements, we present two examples:

- A valid credential c which includes an *age* attribute and ensures that the value of this attribute exceeds 18.

$$\mathcal{L}^\beta(\exists c\, \exists r_c\, ((status_s(r_c) \Leftrightarrow \texttt{false}) \wedge has(c, age) \wedge comp(>, c_{age}, 18)) \tag{1}$$

– A valid credential c from issuer i (*e.g.*, a government body) with a *name* attribute whose value matches that of another credential c''s *name* attribute. Additionally, this second credential c' must be valid and possess a *category* attribute valued at *premium* and should be issued by issuer i' (*e.g.*, a ticket seller for a football game):

$$\begin{aligned}
&\mathcal{L}^{\beta}(\exists c\ \exists s_A\ \exists c'\ \exists s'_A\ (issue_i(s, c, s_A, r_c)\\
&\wedge issue_{i'}(s, c', s'_A, r_{c'}) \wedge (status_s(r_c) \Leftrightarrow status_s(r_{c'}) \Leftrightarrow \texttt{false})\\
&\wedge has(c, name) \wedge has(c', name) \wedge has(c', category)\\
&\wedge comp(\Leftrightarrow, c_{name}, c'_{name})\\
&\wedge comp(\Leftrightarrow, c'_{category}, premium)))
\end{aligned} \tag{2}$$

Lemma 4 ($present^{\beta}_s$ as an instance of $present_s$). *In the SSI model, the function that produces programs $\mathcal{P}^{\beta}$ that proves statements in $\mathcal{L}^{\beta}$ without revealing additional information is $present^{\beta}_s$, which is a particular instantiation of the RAF metamodel's abstract function $present_s$.*

Proof of Lemma 4. Proof by construction. The construction of $present^{\beta}_s$ is demonstrated through the pseudo-algorithm presented in Algorithm 1. To fulfill Definition 7, we adopt ZKP and use a widely recognized notation [19,20,28], symbolized by $\Sigma = (Gen, Prove, Verify)$. Within this scheme, Gen generates common parameters, $Prove$ for creating a cryptographic proof, and $Verify$ for validating said proof.

Theorem 2. *The instantiation of the RAF metamodel's $issue_i$ function following Definition 5 is trivial and is thus omitted for brevity. However, language $\mathcal{L}^{\beta}$ and function $present^{\beta}_s$ are explicitly defined to align with the requirements specified in Definitions 6 and 7. Together, these elements provide a complete description of the SSI model.*

Proof. Proof by construction. Let $\mathcal{L}^{\beta}$, $present^{\beta}_s$ and $\mathcal{P}^{\beta}$ be employed in a protocol that describes the flow of information respecting Definitions 5, 6 and 7. We borrow the protocol from the Proof of Theorem 1, replacing $\mathcal{L}^{\alpha}$ by $\mathcal{L}^{\beta}$, function $present^{\alpha}_s$ for $present^{\beta}_s$, and program $\mathcal{P}^{\alpha}$ by $\mathcal{P}^{\beta}$. Moreover, the statement $\mathcal{L}^{\alpha}(c \cdot r_c)$ of steps two, three, and five are replaced by the FOL statement presented in Eq. 1. □

The SSI model preserves the user's privacy by allowing them to disclose only the necessary information. In this high-level protocol, the subject selects an attribute of one of their credentials to present to the verifier. The verifier does not receive the credentials or revocation registries directly. Instead, the subject constructs a program $\mathcal{P}^{\beta}$ using the selected credential and revocation registry and sends it to the verifier. The verifier trusts the program $\mathcal{P}^{\beta}$ because it is convinced of its truthfulness after executing it to verify the presented statements. The program $\mathcal{P}^{\beta}$ employs ZKP, a technique heavily used in the SSI literature [29].

The OffOIdPM requires users to disclose their credentials to the verifier, who may disseminate this information to other entities. On the other hand, the SSI

Algorithm 1: $present_s^\beta$

Input: Set of credentials $C \subset \mathcal{C}$, set of revocation records $R \subset \mathcal{R}$, query Q in language $\mathcal{L}^\beta$
Output: Program $\mathcal{P}^\beta$

Initialize $\mathcal{P}^\beta$ as an empty list.
Initialize CR'_s as an empty set.
Derive a set of conditions D from Q.
Create tuples linking each credential to its revocation record $CR_s = \{(c_1, r_{c1}), (c_2, r_{c2}), ..., (c_n, r_{cn})\}$ from C and R.
foreach $d \in D$ *related to attribute* a*, its value* c_v *or* r_c **do**
    Add all tuples (c, r_c) to CR'_s in which any attribute $a \in c_A$ its value c_v or revocation record r_c is relevant to d.
end
foreach $(c', r_{c'}) \in CR'_s$ **do**
    Establish common parameters $p \leftarrow Gen()$.
    foreach d *requiring verification of* a, c'_v *or* $r_{c'}$ **do**
        Extract witness w showing a, c'_v or $r_{c'}$ satisfies d.
        Create ZKP: $\pi \leftarrow Prove(p, \{a, c'_v, r_{c'}\}, w)$ without revealing w.
        Append $Verify(p, \pi)$ to $\mathcal{P}^\beta$.
    end
end
Combine all verification elements in $\mathcal{P}^\beta$ ensuring $\mathcal{P}^\beta$ ends with an evaluation that returns `true` if and only if all $Verify$ operations are successful, otherwise `false`.
return $\mathcal{P}^\beta$

model exhibits greater flexibility as it empowers users to generate statements regarding their credentials, enabling them to exert more precise control over disseminating their attributes on the Internet.

4.3 Online Outsourced IdP Model

Lastly, we diverge from the asynchronicity of OffOIdPM and SSI. This section delineates the Online Outsourced Identity Provider Model (OnOIdPM), which brings usability gains to the detriment of privacy towards the IdP. This model is today's *de facto* standard way of using digital identity on the web through the OAuth 2.0 and OIDC protocols. Let us formally define the OnOIdPM as follows:

Definition 8. *In the OnOIdPM, the issuer i issues a credential c and a revocation record r_c and either stores them or sends them to the verifier v.*

Definition 9. *In the OnOIdPM, the verifier v inquires the issuer i with statements about a credential c and its revocation record r_c.*

The Definitions 8 and 9 describe the OnOIdPM, which behaves very distinctively from OffOIdPM and SSI. From the beginning, it is evident that the user is not bothered by the need to store and manage credentials. Let us formally describe it, instantiating the RAF metamodel as follows.

Lemma 5 ($\mathcal{L}^{\gamma}$ as an Instance of $\mathcal{L}$). *In the OnOIdPM, the set of statements regarding credentials and their revocation status can be represented by the language $\mathcal{L}^{\gamma}$, which is a particular instantiation of the RAF metamodel's abstract language $\mathcal{L}$.*

Proof of Lemma 5. Proof by construction. The language $\mathcal{L}^{\gamma}$ can be created using a Context-Free Grammar (CFG), which allows for queries regarding attributes in a credential. Let us define the CFG as $\mathcal{L}^{\gamma} = (V, \Sigma, O, S)$, where V is a set of variables (non-terminal symbols), Σ is a set of terminal symbols, O is a set of production rules, S is the start symbol.

Variables (V):

- Q - A query.
- L - A list of attributes to return the value.
- B - An attribute.

Terminal Symbols (Σ):

- $age, name, ...$ - Attributes in A.
- s - The subject s.
- of - The subject from which to obtain the credential with the attributes.

Start Symbol:

- $S = Q$ - The start symbol is the query.

Production Rules (O):

1. $Q \rightarrow L\ of\ s$
2. $L \rightarrow B \mid B\ L$
3. $B \rightarrow age \mid name \mid ...$

Corollary 3. *The language $\mathcal{L}^{\gamma}$ facilitates querying one or more attributes from a credential. For example, the query "age name of s" is accepted by $\mathcal{L}^{\gamma}$, indicating it conforms to the grammar's rules for valid queries.*

Lemma 6 ($present_i^{\gamma}$ as an instance of $present_i$). *In the OnOIdPM, the function that produces programs $\mathcal{P}^{\gamma}$ to prove statements in $\mathcal{L}^{\gamma}$ without revealing additional information is $present_i^{\gamma}$, an instance of RAF metamodel's $present_i$.*

Proof of Lemma 6. The concrete function $present_i^{\gamma}$ has three steps. First, the query, *i.e.*, the statements in language $\mathcal{L}^{\gamma}$, are decomposed into a list of attributes and their credential using the CFG described in Proof 4.3. Second, for each existing attribute a in a queried credential c, an instruction is added to program $\mathcal{P}^{\gamma}$. This instruction outputs a and its value, namely c_a. Third, an instruction to output the revocation status of c, namely r_c, is added.

Theorem 3. *The instantiation of the RAF metamodel's $issue_i$ function following Definition 8 is trivial and thus omitted. Language $\mathcal{L}^{\gamma}$ and function $present_i^{\gamma}$ are explicitly defined to align with the requirements specified in Definition 9. Together, these elements provide a complete description of the OnOIdPM.*

Proof. We prove Theorem 3 by construction using language $\mathcal{L}^{\gamma}$ and function $present_i^{\gamma}$. We present a protocol that describes the flow of information respecting Definitions 8 and 9. The protocol is depicted in Fig. 3 and operates as follows:

1. Subject s requests a credential from issuer i, which creates a credential c with attributes *age* and *name* and the revocation record r_c, using function $issue_i$ and stores them.
2. Verifier v requests to issuer i for the attribute *name* of subject s using $request_v(\mathcal{L}^{\gamma}(\texttt{name of s}))$.
3. Issuer i uses $present_i^{\gamma}(\{c\}, \{r_c\}, \{\mathcal{L}^{\gamma}(\texttt{name of s})\})$ to obtain program $\mathcal{P}^{\gamma}$, then asks subject s to prove they are s referenced in $\mathcal{P}^{\gamma}$ through $authn_s(P^{\gamma})$ followed by obtaining authorization to share program $\mathcal{P}^{\gamma}$ with v using the function $authz_s(P^{\gamma}, v)$.
4. Issuer i shares the program $\mathcal{P}^{\gamma}$ with verifier v, which grants access to subject s if $\mathcal{P}^{\gamma}$ satisfies the requested statements, and if they trust the issuer, and if the credential is not revoked.

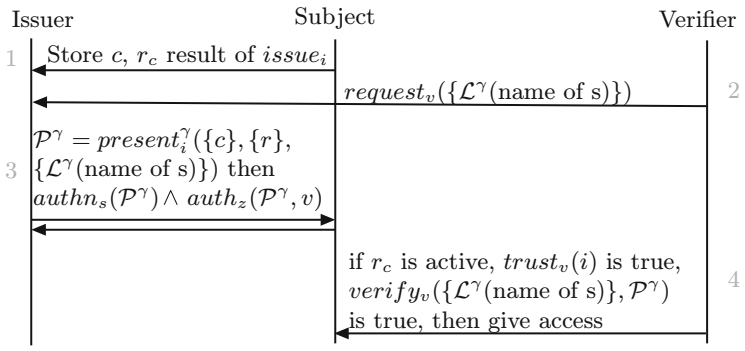


Fig. 3. Online outsourced IdP protocol: the flow of data between issuer i, subject s, and verifier v. Numbers denote protocol steps, and arrows represent messages.

The protocol describes a different information flow than the SSI model. The subject s does not have to store its credentials as the issuer i does that and shares programs with verifier v. This protocol is similar to OAuth 2.0 [13] in three ways. Firstly, the responsibility of implementing an authentication mechanism is delegated to the implementer. Secondly, it requires an expression of intention from the user for the issuer to share their data with the verifier, *i.e.*, the $authz_s$ function. Lastly, the program $\mathcal{P}^{\gamma}$ represents an `access token` containing the authorized user attributes intended for sharing with the verifier.

5 Discussion

The literature abstracts real-world applications that use identity in digital identity models. However, a thorough analysis demands more than abstracting roles and information flow. The RAF framework facilitates a deeper understanding, more effectively distinguishing between identity models. This section explores insights derived from the instantiation of models using the RAF metamodel.

The x509 Family and SSI have a Lot in Common. The literature points to SSI as a strong driver of innovation in the IAM field [21]. Indeed, significant innovation is happening in ZKP and privacy-preserving protocols [29]. Nevertheless, the Proof of Theorem 2 shows that SSI is, from a conceptual point of view, the result of the inclusion of privacy-preserving techniques to express statements about attributes, credentials, and revocation records to the OffOIdPM. This questions the need to abandon three decades of technological advances made in the x509 family of protocols and the myriad of applications that support them in favor of a new stack of technologies. Instead, integrating ZKP and similar modern technologies into existing x509 frameworks might offer users privacy benefits without discarding the existing infrastructure.

Chronological Development and Linguistic Complexity of e-ID Models. It is worth noting that the chronology of the creation and popularization of digital identity models aligns with Chomsky hierarchy [30], depicted in Fig. 4. Although we presented the models in the order of OffOIdPM, SSI, and OnOIdPM, their chronological development occurred as follows: first OffOIdPM, then OnOIdPM, and finally SSI. By analyzing the languages associated with these models in their chronological order and classifying them according to Chomsky hierarchy, we find that the languages $\mathcal{L}^{\alpha}$, $\mathcal{L}^{\gamma}$, $\mathcal{L}^{\beta}$ correspond to regular, context-free, and context-sensitive grammars, respectively. The adoption and widespread use of one model over another may be influenced by the increased expressiveness of the languages associated with each model, in addition to the improvements in privacy and usability.

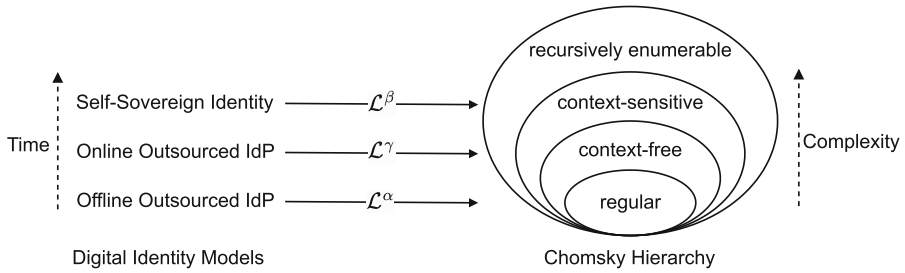


Fig. 4. Digital identity models and Chomsky hierarchy.

A Look Into the Future. The chronological development of digital identity models highlights the early establishment of OnOIdPM, which was distinguished by its usability improvement for the end-user. Despite SSI's later introduction, which conceptually echoed OffOIdPM's approach, there are research and market opportunities to leverage OnOIdPM's inherent usability advantages and tackle its privacy disadvantages. To this end, privacy-preserving techniques are being integrated into OnOIdPM's protocols, exemplified by OpenID for Verifiable Presentations, aiming to meld OAuth 2.0 and OIDC with ZKP to reconcile the privacy divide between SSI and OnOIdPM [32].

6 Related Work

There have been previous attempts to provide a meta-discussion on digital identity. The "The Laws of Identity" by Kim Cameron [6], for instance, attempts to develop an understanding of identity systems for the Internet. The author draws seven statements, named laws of identity, that they argue should guide the construction of digital identity systems. They range from user consent and minimal disclosure to the pluralism of technologies and consistent user experience. This work offers general guidelines instead of mathematically describing meta-identity with entities and their relationships as this paper does. This is also the case for the seminal work of SSI, *i.e.* the "The Path to Self-Sovereign Identity" by Christopher Allen [2]. Instead of detailing rules of a metamodel for identity or the SSI model itself, ten general directives for SSI are given. We have presented them in Sect. 2.

On the other hand, some related works formally describe identity models. In [9], an abstract identity model named the Digital Identity Model (DIM) is introduced. It defines domains of organizations, users, and their attributes, and a mapping between users, attributes, domains, and their values. In other words, users' attribute value depends on the domain. Other functions are also defined: *e.g. ident*, which returns the identifier for a given domain; *checkCredC*, which checks if an identifier and a credential with values match in a context. These are combined to describe existing identity models [8,9] and protocols [10]. In contrast to utilizing general functions as foundational elements that define identity models, *i.e.*, a bottom-up approach, our research presents a metamodel that is instantiated into a model, *i.e.*, a top-down approach. Through this process, entities, computation artifacts, and relationships are established, providing greater versatility and enabling the description of new models with novel relationships, a capability that DIM lacks.

7 Final Remarks

Engaging in meta-discussions about scientific subjects is an essential endeavor that facilitates a deeper understanding of the objectives and methods employed within a particular field of study. These attempts facilitate additional innovation

by utilizing shared notation and establishing mutual knowledge about the various components and their interconnections.

This research paper introduced the RAF meta-metamodel and metamodel as a framework for examining identity and access management dynamics. It was used to formally describe three digital identity models, specifically the offline outsourced IdP, the online outsourced IdP, and SSI. The utilization of the RAF metamodel facilitated the comparison between the models, effectively highlighting their shared and distinctive characteristics. We hope future IAM-related works employ our suggested notation to articulate their novel contributions.

References

1. Adams, C., Lloyd, S.: Understanding public-key infrastructure: concepts, standards, and deployment considerations. Sams Publishing (1999)
2. Allen, C.: The path to self-sovereign identity. Available online: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html (accessed on 04 June 2023) (2016)
3. Barnes, R., Hoffman-Andrews, J., McCarney, D., Kasten, J.: Automatic Certificate Management Environment (ACME). RFC 8555 (Mar 2019). https://doi.org/10.17487/RFC8555
4. Boeyen, S., Santesson, S., Polk, T., Housley, R., Farrell, S., Cooper, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280 (May 2008). https://doi.org/10.17487/RFC5280
5. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_5
6. Cameron, K.: The laws of identity. Available online: https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf. Accessed 04 June 2023 (2005)
7. Čučko, Š., Bećirović, Š., Kamišalić, A., Mrdović, S., Turkanović, M.: Towards the classification of self-sovereign identity properties. IEEE Access **10**, 88306–88329 (2022). https://doi.org/10.1109/ACCESS.2022.3199414
8. Ferdous, M.S., Chowdhury, F., Alassafi, M.O.: In search of self-sovereign identity leveraging blockchain technology. IEEE Access **7**, 103059–103079 (2019). https://doi.org/10.1109/ACCESS.2019.2931173
9. Ferdous, M.S., Norman, G., Poet, R.: Mathematical modelling of identity, identity management and other related topics. In: Proceedings of the 7th International Conference on Security of Information and Networks, pp. 9–16. SIN '14, Association for Computing Machinery (2014). https://doi.org/10.1145/2659651.2659729
10. Ferdous, M.S., Poet, R.: Formalising identity management protocols. In: 2016 14th Annual Conference on Privacy, Security and Trust (PST), pp. 137–146. IEEE (2016). https://doi.org/10.1109/PST.2016.7906948
11. Hallam-Baker, P., et al.: An Extension to HTTP : Digest Access Authentication. RFC 2069 (Jan 1997). https://doi.org/10.17487/RFC2069
12. Hardman, D.: DIDComm Messaging. Available online: https://identity.foundation/didcomm-messaging/spec/. Accessed 04 June 2023 (2020)
13. Hardt, D.: The OAuth 2.0 Authorization Framework (Nov 2012). https://doi.org/10.17487/RFC6749

14. Hughes, J., Maler, E.: Security assertion markup language (saml) v2.0 technical overview. Available online: https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html. Accessed 30 Aug 2024 (2005)
15. ITU-T: Recommendation x.509. Available online: https://www.itu.int/rec/T-REC-X.509-201910-I/en. Accessed 04 June 2023 (2000)
16. Kause, T., Peylo, M.: Internet X.509 Public Key Infrastructure – HTTP Transfer for the Certificate Management Protocol (CMP). RFC 6712 (Sep 2012). https://doi.org/10.17487/RFC6712
17. Kern, H.: Model interoperability between meta-modeling environments by using M3-level-based bridges. Ph.D. thesis, Universität Leipzig (2016)
18. Kuperberg, M.: Blockchain-based identity management: a survey from the enterprise and ecosystem perspective. IEEE Trans. Eng. Manage. **67**(4), 1–20 (2019). https://doi.org/10.1109/TEM.2019.2926471
19. Lee, J., Choi, J., Oh, H., Kim, J.: Privacy-preserving identity management system. Cryptology ePrint Archive 2021/1459 (2021)
20. Lee, J., Hwang, J., Choi, J., Oh, H., Kim, J.: SIMS: self-sovereign identity management system with preserving privacy in blockchain. IACR Cryptol. ePrint Archive **1**(2019/1241), 1–13 (2019)
21. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. Comput. Sci. Rev. **30**, 80–86 (11 2018). https://doi.org/10.1016/j.cosrev.2018.10.002
22. Myers, M., Schaad, J.: Certificate Management over CMS (CMC). RFC 5272 (Jun 2008). https://doi.org/10.17487/RFC5272
23. Polk, T., Lee, J., Park, S., Park, J., Lee, H., Lee, H.: Internet X.509 Public Key Infrastructure Subject Identification Method (SIM). RFC 4683 (Oct 2006). https://doi.org/10.17487/RFC4683
24. Recordon, D., Reed, D.: Openid 2.0: a platform for user-centric identity management. In: Proceedings of the Second ACM Workshop on Digital Identity Management., pp. 11–16. ACM, Alexandria, USA (2006). https://doi.org/10.1145/1179529.1179532
25. Reed, D., Sporny, M., Longley, D., Christopher, A., Grant, R., Sabadello, M.: Decentralized Identifiers (DIDs). Available online: https://www.w3.org/TR/did-core/. Accessed 04 June 2023 (2019)
26. Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, D.C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 6960 (Jun 2013). https://doi.org/10.17487/RFC6960
27. Scarfone, K., Souppaya, M.: Guide to enterprise password management. Available online: https://csrc.nist.gov/publications/detail/sp/800-118/archive/2009-04-21. Accessed on 04 June 2023 (2009)
28. Schanzenbach., M., Kilian., T., Schütte., J., Banse., C.: Zklaims: Privacy-preserving attribute-based credentials using non-interactive zero-knowledge techniques. In: Proceedings of the 16th International Joint Conference on e-Business and Telecommunications - SECRYPT,. pp. 325–332. INSTICC, SciTePress, Prague, Czech Republic (2019). https://doi.org/10.5220/0007772903250332
29. Schardong, F., Custódio, R.: Self-sovereign identity: a systematic review, mapping and taxonomy. Sensors **22**(15), 5641 (2022). https://doi.org/10.3390/s22155641
30. Sipser, M.: Introduction to the Theory of Computation. Cengage Learning (2013)
31. Sporny, M., Longley, D., Chadwick, D.: Verifiable Credentials Data Model 1.0. Available online: https://www.w3.org/TR/vc-data-model/. Accessed 04 June 2023 (2017)

32. Terbu, O., Lodderstedt, T., Yasuda, K., Looker, T.: Openid for verifiable presentations. Available online: https://openid.net/specs/openid-4-verifiable-presentations-1_0.html Accessed 30 August 2024 (2024)
33. West, R., Bluhm, D., Hailstone, M., Curran, S., Curren, S., Aristy, G.: Aries rfc 0023: Did exchange protocol 1.0. Available online: https://github.com/hyperledger/aries-rfcs/blob/master/features/0023-did-exchange. Accessed on 30 August 2024 (2019)