From Self-Sovereign Identity to Fiduciary Identity: A Journey Towards Greater User Privacy and Usability

Frederico Schardong*
Instituto Federal do Rio Grande do Sul
Rolante, Brazil
Universidade Federal de Santa Catarina
Florianópolis, Brazil
frederico.schardong@rolante.ifrs.edu.br

Ricardo Custódio
Universidade Federal de Santa Catarina
Florianópolis, Brazil
ricardo.custodio@ufsc.br

ABSTRACT

In an era defined by the evolving Identity and Access Management (IAM) landscape, this paper shines a spotlight on the critical aspect of usability limitations inherent in the Self-Sovereign Identity (SSI) paradigm. While SSI holds the promise of returning control and privacy to individuals, it presents challenges in terms of user experience for the layman, particularly concerning the management of cryptographic keys, credentials, and the evaluation of service provider requests for private information. To tackle these inherent limitations of SSI, this paper introduces a new digital identity model named *Fiduciary Identity*. It is constructed over the fiduciary relationship, a poorly explored topic in IAM. The fiduciary acts on behalf of the user, simplifying interactions and enhancing usability, bridging the gap between the potential advantages of SSI and the practicality of everyday digital life. The innovative concept of the fiduciary identity acts as a bridge toward achieving a user-centric, secure, and privacy-preserving IAM ecosystem.

CCS CONCEPTS

• **Software and its engineering** → **Software system structures**; • **General and reference** → *Design*; • **Security and privacy** → *Authentication.*

KEYWORDS

Identity and Access Management, Digital Identity, Self-Sovereign Identity, SSI, Fiduciary Identity.

ACM Reference Format:
Frederico Schardong and Ricardo Custódio. 2024. From Self-Sovereign Identity to Fiduciary Identity: A Journey Towards Greater User Privacy and Usability. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3605098.3636061

*This study was supported by the Federal Institute of Education, Science and Technology of Rio Grande do Sul (IFRS) and the Operador Nacional do Registro Civil de Pessoas Naturais (ON-RCPN).

SAC '24, April 8–12, 2024, Avila, Spain

ACM ISBN 979-8-4007-0243-3/24/04…$15.00
https://doi.org/10.1145/3605098.3636061

1 INTRODUCTION

Identity and Access Management (IAM) has been a topic of great interest since the early days of the Internet. With the increasing amount of sensitive data being stored and shared digitally, it has become essential to establish secure ways of authenticating individuals and managing the exchange of digital identity information. IAM encompasses the processes and technologies used to manage users' identities and access privileges within an organization or system. Traditional IAM systems involve centralized databases and complex authorization mechanisms, which can be prone to security breaches. With the increasing digitization of society, the need for a more flexible and secure approach to IAM has emerged.

Self-Sovereign Identity (SSI) represents a shift in how individuals assert and control their digital identities [1]. SSI is built upon the principles of decentralized and user-centric identity management, enabling individuals to manage and share their identity information in a secure, verifiable, and privacy-preserving manner, fostering trust and reducing reliance on third-party intermediaries [9, 12].

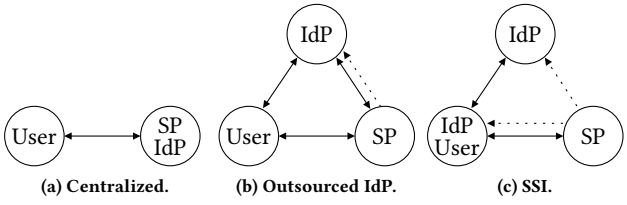
While SSI promises enhanced control and privacy for users, it can also present challenges in terms of user experience [17]. Managing cryptographic keys and digital wallets, which are integral to SSI, can be complex for the average user, potentially leading to issues with key loss, mismanagement, or a lack of understanding. Additionally, the process of sharing and verifying decentralized identities in a user-friendly and standardized fashion across various digital services and platforms remains a challenge. Efforts to simplify the user experience of SSI are ongoing, but these usability limitations must be addressed to ensure its success.

In this paper, we propose the *Fiduciary Identity*, a new digital identity model that emphasizes user freedom through the fiduciary relationship between the beneficiary, *i.e.*, the user, and the fiduciary, *i.e.*, who manages and worries with day-to-day matters related to user identity. This particular kind of relationship implies that the fiduciary puts the beneficiary's interests over its own, not seeking advantages or misusing the data in any shape or form. This relationship is governed by explicit and implicit consent given or revoked by the user to the fiduciary at any time. The ultimate goal is to free the user from cumbersome, repetitive, and often intrusive interactions with Service Providers (SPs) by offloading these interactions to one or more trusted representatives.

Paper Organization. Section 2 provides the preliminaries and related work. We introduce a computational model in Section 3 to describe the Fiduciary Identity model formally. We then thoroughly discuss, formalize, and compare the Fiduciary Identity model in Section 4. Finally, Section 5 concludes the paper.

2 IAM BACKGROUND AND RELATED WORK

This section provides readers with the necessary knowledge about existing IAM models to understand the remainder of the paper. These IAM models are also related works to our proposal. Digital identity models[1] can be generally described according to the roles played by the participants and their interactions. The literature classifies current IAM systems into three digital identity models, namely centralized (Figure 1a), outsourced IdP (Figure 1b), and SSI (Figure 1c) [1, 17].



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

Figure 1: Current digital identity models. Circles represent entities, labels represent roles, solid lines represent interactions, and dashed lines represent trust [17].

2.1 Centralized

In the early days of the Internet, SPs were compelled to establish their own IdPs for client authentication and personalized service delivery due to a lack of alternatives. However, this practice gave rise to *centralized* authorities assuming dual roles as both SPs and IdPs. This architectural approach resulted in usability problems for users, as they frequently resorted to employing weak passwords across various systems, ultimately leading to numerous security vulnerabilities. Moreover, the lack of portability for user identities further complicated the landscape. Consequently, initiatives were launched to raise awareness among users about the dangers associated with using simple passwords and recycling them [15, 21].

2.2 Outsourced IdP

In pursuit of more inclusive IAM solutions, centralized authorities were substituted with third-party providers known as IdPs. This transformation allows users to register with a limited number of IdPs to gain access to a broad spectrum of web-based services. At the same time, SPs are required to register with IdPs to operate in tandem with them. This approach simplifies the management of user identifiers and passwords but does give rise to a few significant silos containing valuable private information [1]. To standardize the interactions among IdP, SP, and end-users, token exchange protocols such as SAML [7], OAuth 2.0 [6], and OpenID Connect [13] were introduced. These protocols are used daily by billions of users through leading industry IdPs like Google and Meta.

2.3 Self-Sovereign Identity

Self-Sovereign Identity (SSI) represents a shift in digital identity, prioritizing individual autonomy and control over personal data. Unlike earlier identity models that regarded individuals as mere users, SSI empowers individuals as the owners of their data, affording them complete authority. Under SSI, individuals can exercise full control over the management and storage of their data, as well as the discretion to determine with whom they share it [17]. Christopher Allen [1] introduced a set of ten principles that any SSI system must adhere to, collectively known as the ten principles of SSI. These principles provide a comprehensive framework that outlines the fundamental characteristics of an SSI system [1].

(1) Users must have an independent *existence*;
(2) Users must *control* their identities;
(3) Users must have *access* to their own data;
(4) Systems and algorithms must be *transparent*;
(5) Identities must be *persisted*;
(6) Information and services must be *portable*;
(7) Identities should be *interoperable*;
(8) Users must *consent* to identity use;
(9) Disclosure of claims must be *minimized*;
(10) The rights of users must be *protected*.

SSI handles attribute sharing in a manner reminiscent of traditional outsourced IdPs, granting users the discretion to select which attributes they wish to disclose to SPs. However, SSI employs the *minimalization* principle to bolster user privacy. This principle enables users to share statements about their attributes, such as their age, rather than revealing the raw attribute itself, for instance, their birthdate. This approach offers the advantage of sharing and potentially retaining less information on the part of the SP, thereby enhancing user privacy. Zero-Knowledge Proof (ZKP) is a widely adopted technique in SSI to share components of a credential and predicates related to one or more attributes without compromising the authenticity or integrity of the data [10, 11, 16]. In a nutshell, ZKP allows a user to demonstrate to an SP that they possess knowledge of a specific attribute without exposing the value itself [18].

Within the SSI model, the *access* principle is pivotal, granting users control over their data without reliance on IdPs. Users enjoy complete autonomy in storing, managing, and sharing their data. The implementation of the access principle is further facilitated by ZKP, allowing the verification of credential validity without revealing the actual credential to the SP [4]. Trust is another fundamental pillar of SSI. While individuals possess the autonomy to issue their credentials, others retain the right to exercise skepticism. Users can engage with SPs using credentials, either self-issued or third-party-issued, but it is the prerogative of the SP to determine the level of trust in the issuer.

Figure 2 shows a typical interaction between the subject and the SP[2]. This diagram shows a subject requesting access to a service. It triggers the SP to reply with the required statements, *i.e.*, statements about the attributes in credentials, such as being over a certain age. Here lies a cumbersome and error-prone usability matter of SSI: the user is presented with said required statements by their digital wallet, marked in red in the diagram. The user may lack the mental capacity to comprehend the request, be fatigued or stressed, or have encountered numerous permission prompts, leading to a tendency to accept any request. Drawing a parallel, consider browsing the modern web, where most websites seek user permission to store cookies and approve privacy policies.

[1]We use the terms digital identity models and IAM models interchangeably.

[2]We use the terms subject and user, SP and verifier, and IdP and issuer interchangeably.

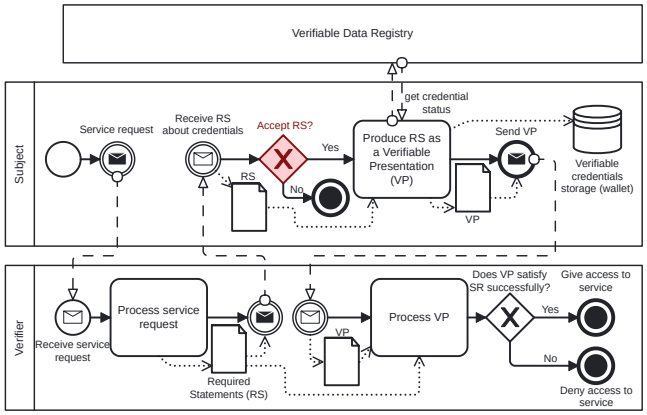


Figure 2: A typical interaction between subject and SP in SSI using the Business Process Model and Notation (BPMN).

3 COMPUTATIONAL MODEL

In the preceding section, we have laid the groundwork for various digital identity models. In this section, we delve into a general-purpose computational model, which will serve as the foundation for formally describing the Fiduciary Identity.

The Norma Machine (NM) [19] is similar to the Von Neumann architecture employed in contemporary computers and is equivalent to a Turing Machine [22]. A distinguishing feature of the NM is its unlimited number of registers. Registers are memory locations that can store natural numbers of arbitrary sizes and can be accessed and modified at any point. The concept of a program is rigorously defined within the NM framework; it comprises a set of instructions that carry out operations on registers[3]. Programs consist of four fundamental instructions: increment, decrement, conditional jump, and stop.

Definition 3.1. Norma Machine is as tuple $NM = (R, Q)$.

Where R denotes the infinite set of indexed registers r_k, where $k \in [0, \infty]$, with each register capable of storing natural numbers ($\forall k \in [0, \infty], r_k \in \mathbb{N}$). Q comprises the set of operations performed on the registers. There are five operations: (i) e represents the initialization operation, where $e_k(w)$ stores the value w in register r_k, with all other registers set to zero; (ii) s is the retrieval operation, where s_k retrieves the value stored in register r_k; (iii) z denotes the test operation, with z_k returning `true` if $r_k = 0$ and `false` otherwise; (iv) ad signifies the increment operation, as ad_k adds one to register r_k and updates it with the result; and (v) sub is the decrement operation, reducing the value in r_k by one and storing the result in the same register. Importantly, if r_k is already at zero, it remains at zero.

With these operations, the NM can execute both arithmetic and logical operations on the registers, facilitating the implementation of algorithms and the resolution of computational problems. Operations on registers enable arithmetic and logical operations, thus enabling algorithm implementation. The notation e_k, s_k, z_k, ad_k, and sub_k indicate that the respective operation is applied to register r_k.

Definition 3.2. A program for the NM is represented as a tuple $P = (x, y, I, C, i_0)$.

Here, r_x denotes the register where the input is stored, and r_y is the register designated for the output. I is the set of indexes for the program's instructions, while C comprises the instruction set of the program P. Finally, i_0 designates the program's initial instruction index, with $i_0 \in I$. It is worth noting that the instruction pointed to by the address of the initial instruction i_0 must contain the operation e.

The instruction set C comprises tuples with three elements: $c = (i_\triangledown, o, i_\triangleright) \in C$. The first element of the tuple, $i_\triangledown \in I$, represents the index of the current statement c. The second element, $o \in Q$, specifies the operation to be executed by the instruction c. The third element, $i_\triangleright \in I$, designates the index of the statement immediately following c. If $i_\triangleright = \varepsilon$, it indicates a stop instruction.

Let mem be defined as the set of pairs that represent all indexed registers r_k and their respective values, denoted by v_k, where k is the register index:

$$mem = \{(r_0, v_0), (r_1, v_1), ...\} \tag{1}$$

Definition 3.3. The NM execution history is a set of pairs $H = \{(c_0, mem_0), (c_1, mem_1), ..., (c_{n-1}, mem_{n-1})\}$

Each pair (c_j, mem_j) consists of the next instruction to be executed, c_j, and the state of the registers at that moment, denoted as mem_j. The index 0 to $n-1$ indicates the states of the program from its inception to its conclusion.

Each valid instruction c_j of the program P in the execution history H evolves the state of the machine from (c_j, mem_j) to (c_{j+1}, mem_{j+1}):

$$(c_0, mem_0) \vdash_{NM} (c_1, mem_1) ... \vdash_{NM} (c_{n-1}, mem_{n-1}) \tag{2}$$

The program accepts input w only if there exists a computation of the form:

$$((i_0, e_x(w), i_\triangleright), mem_0) \vdash^n_{NM} ((i_\triangledown, s_y, \varepsilon), mem_{n-1}) \tag{3}$$

To enhance convenience, we have compiled a table (Table 1) that includes mathematical symbols associated with the NM computational model, as referenced in the subsequent sections.

Table 1: Notation table for the computational model.

Symbol	Definition
$NM = (R, Q)$	Norma Machine
$P = (x, y, I, C, i_0)$	Program executable in a NM
H	Execution history of a program

[3]A clear differentiation exists between an operation and an instruction. Specifically, every non-empty instruction encompasses an operation, and this operation, in turn, represents the procedure carried out on one or more registers.

4 FIDUCIARY IDENTITY

Both the centralized, outsourced IdP and SSI models have shortcomings concerning user autonomy in managing their data, user-friendliness, or both. The *Fiduciary Identity* model is a novel digital identity approach that seeks to tackle the limitations mentioned above by relieving users from the burdensome and repetitive interactions with SPs. It is based on the fiduciary relationship, where the subject establishes consent rules that the fiduciary will use to reason and act on behalf of the subject.

This section is structured as follows. We initiate by introducing the concept of the fiduciary relationship to the reader. Next, we introduce and motivate the Fiduciary Identity model. Subsequently, we delve into the core concepts of the Fiduciary Identity model through a formal description aided by the computational model discussed earlier. Following this, we present a high-level protocol outlining the information flow within this model. An introductory discussion of security aspects is given, and finally, we engage in a comprehensive discussion that includes comparisons with existing models.

4.1 Foreword about the Fiduciary Relationship

Fiduciary relationships, deeply rooted in trust and responsibility, play a crucial role in our lives. Consider, for example, the doctor-patient relationship or the attorney-client dynamic, in which professionals are ethically and legally obligated to act in the best interests of their clients. These are illustrative instances of fiduciary relationships commonly delineated by legal principles.

While a universally accepted definition of the fiduciary relationship remains unresolved [2], Rotman [14] contends that its inherent flexibility contributes to this lack of consensus. Nonetheless, there are identifiable characteristics of fiduciary relationships as described in common law [3].

In the fiduciary relationship, one party assumes a position of authority and responsibility over another, necessitating utmost care, loyalty, and transparency. In broad terms, a fiduciary is an individual who bears distinct responsibilities of loyalty and reliability towards another individual. The fiduciary must exercise caution and prioritize the other party's best interests, which may be referred to as the principal, beneficiary, or client. The beneficiary places their trust or confidence in the fiduciary, and it is the fiduciary's responsibility to refrain from breaching that trust or confidence [3].

Another aspect of the fiduciary relationship is the consequence that may arise when one entity places its trust and confidence in another and, as a result, gains power and superiority over the other [5]. The beneficiary's susceptibility to the fiduciary's abuse of power is a worrisome trait that legal foundations can mitigate.

Our exploration of the fiduciary relationship began in this subsection by examining its fundamental characteristics. By understanding the principles of this relationship, we can analyze its advantages over conventional trust relationships. In the subsequent subsection, we will delve into the motivations driving the shift towards the fiduciary relationship, highlighting its indispensable role in addressing the limitations of existing trust-based paradigms. Through this progression, we aim to show the compelling reasons behind adopting the fiduciary relationship and its implications in a digital identity model.

4.2 The Fiduciary Identity Model

In [3], the notion of considering SPs as information fiduciaries was introduced. Building on this concept, [2] re-framed it specifically for the realm of health data SPs. We propose a different perspective. Instead of establishing a fiduciary relationship directly between end-users and SPs, our thesis advocates for the creation of an independent fiduciary entity, distinct from IdPs or SPs. This entity is designed to represent users and prioritize their interests in the digital realm. To be more pragmatic, this fiduciary must actively manage, make decisions, and take actions concerning the digital selves of those it represents.

It is essential to highlight that relying solely on trust, transparency, and consent is insufficient for protecting against privacy harms, as emphasized in [2]. The fiduciary relationship, however, establishes stronger bonds and emerges as the ideal choice for the connection between end-users and the fiduciary representing them.

We introduce the *Fiduciary Identity*, a new digital identity model. It builds upon established concepts from existing identity models [12, 17, 20] to create a more comprehensive, inclusive, and robust digital identity. An architectural overview of the Fiduciary Identity model is depicted in Figure 3. In this model, the fiduciary is commissioned to represent the user, and thus interacts with the IdP and SP on their behalf. These interactions are guided by implicit and explicit consent expressed as rules, policies, or guidelines for the fiduciary to consider when representing the user.

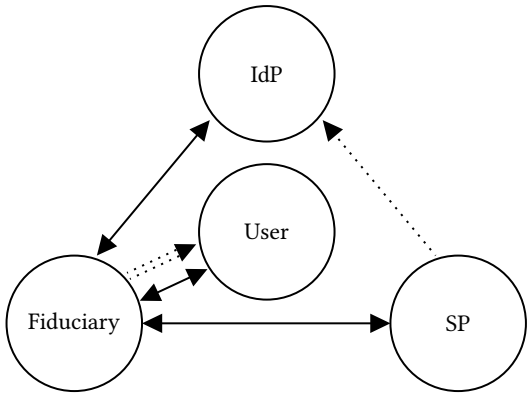


Figure 3: Fiduciary Identity: solid lines for interactions, dashed lines for trust, and double dashed lines for fiduciary.

In the outsourced IdP model, user consent is often implicit or assumed, such as when a user provides personal information to a company in exchange for access to a service. While the SSI model offers several advantages over earlier identity models, it still faces challenges and limitations. Users must actively consent to share their information with verifiers. This presents a challenge because users may need help understanding the full implications of their consent and are prone to abusive verifiers that might require excessive information. In the Fiduciary Identity model the subject must express their consent through rules, policies, or guidelines for the fiduciary to consider when representing the subject. In this model, the fiduciary and the verifier have their own computing

devices, which can be used independently or in conjunction to execute algorithms that utilize the subject's data. This model allows the fiduciary to negotiate with verifiers where programs that operate over subject private data are executed. For example, a fiduciary may not trust a verifier to receive the subject's birthdate but may agree to run the verifier's algorithm on the fiduciary's computing device to ensure the verifier cannot access their birthdate.

Let us show how a vanilla service request interaction, *i.e.* the SP requires private information from the subject to allow access to the requested service, requires less effort from the user in the Fiduciary Identity model than previous models. To do that, we step up the example given in Figure 2 to the Fiduciary Identity model, shown in Figure 4. Note that all the subject has to do is make the service request. The previously cumbersome and tiring experience of interacting with the verifier is now performed by the user's fiduciary, which reasons upon previously expressed consent rules, marked in red. This example provides an overview of the possibilities. Naturally, the fiduciary might bother the user to request permission if the previously given consent rules ask for such behavior.

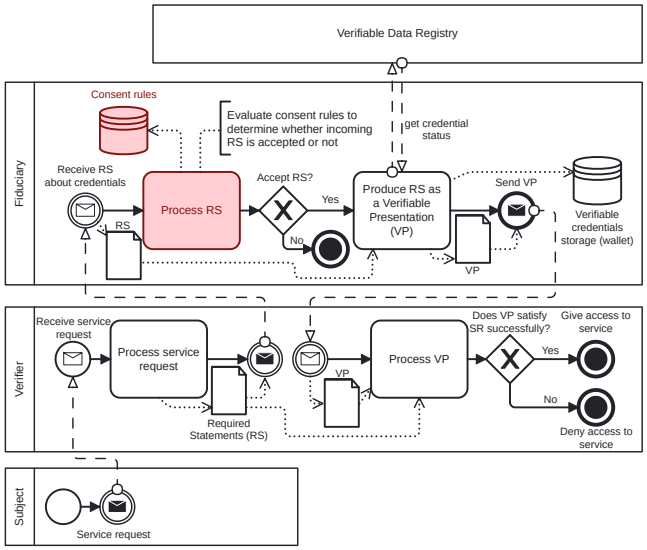


Figure 4: An example of interaction between subject, fiduciary, and SP (verifier) in the Fiduciary Identity model.

4.3 Formal Description

To articulate the *Fiduciary Identity model*, we employ set theory to describe three key sets: roles, computational artifacts, and functions. The set of roles, denoted as $\mathbb{R}$, represents the various actors within the IAM system. The set of computational artifacts, denoted as $\mathbb{A}$, encompasses algorithms, data structures, and other artifacts. Lastly, the set of functions, denoted as $\mathbb{F}$, comprises operations applied to both roles and artifacts.

Definition 4.1. The *Fiduciary Identity model* is a triple ($\mathbb{R} = \{s, i, v, f\}$, $\mathbb{A} = \{C, \mathcal{R}, NM_f, NM_v, P, \mathcal{L}, \mathcal{P}, \mathcal{E}, E\}$, $\mathbb{F} = \{issue_i, revoke_i, status_i, trust_v, request_v, present_f, authz_f, exec_{v,f}, verify_v, fiduciary_f, announce_s, express_s, audit_s\}$), where:

Roles ($\mathbb{R}$):

- s: subject, *i.e.*, the user, with a set of attributes s_A (*e.g.*, age and name).
- i: issuer, *i.e.*, the IdP.
- v: verifier, *i.e.*, the SP.
- f: fiduciary of subject s.

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

- C: set of credentials, and each credential $c \in C$ asserts a set of attributes $c_A \subseteq s_A \mid c_A \subset s_A$ about subject s and is controlled by fiduciary f.
- $\mathcal{R}$: set of revocation registries, where each registry $r_c \in \mathcal{R}$ is associated with a credential c.
- NM_f: fiduciary's Norma Machine, following Definition 3.1.
- NM_v: verifier's Norma machine, following Definition 3.1.
- P: program following Definition 3.2.
- $\mathcal{L}$: language for statements, *e.g.*, predicate or first-order logic, where $\mathcal{L}(C \cup \mathcal{R})$ is the set of statements about credentials and revocation registries, and $\mathcal{L}(\mathcal{E})$ represents the set of statements about *executions*, refer to $\mathcal{E}$'s definition.
- $\mathcal{P}$: set of plans, each *plan* is a tuple $(\mathcal{L}(C \cup \mathcal{R}), P)$, where $\mathcal{L}(C \cup \mathcal{R})$ represents statements about credentials attributes and their revocation status, and $P = (x, y, I, C, i_0)$ is a program created by $present_f$ over $\mathcal{L}(C \cup \mathcal{R})$. A *plan* ensures that the register r_y with the output of the execution is always shared with verifier v, regardless of which set of NM ran it.
- $\mathcal{E}$: set of *executions*, where each *execution* represent running a *plan* in a set of NM, *i.e.*, *execution* is a tuple $(plan, NM')$ where $plan \in \mathcal{P}$ and $NM' \in \wp(NM_f, NM_v)$.
- E: set of evidence, where the evidence of an *execution* is $e^{execution} = (H', M')$ consisting of an execution history H' and metadata M' such.

Functions ($\mathbb{F}$):

- $issue_i : \{s\} \times \wp(s_A) \nrightarrow C \times \mathcal{R}$ — issuing a credential by issuer i takes the subject s into consideration, a subset of their attributes $A' \subseteq s_A$, and generates a credential $c \in C$ with attributes $c_A = A'$ and a revocation record $r \in \mathcal{R}$.
- $revoke_i : C \times \mathcal{R} \rightarrow \mathcal{R}$ — given a credential $c \in C$ 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 $r' \in \mathcal{R}$.
- $status_i : C \rightarrow \{\texttt{true}, \texttt{false}\}$ — checks if credential c is revoked and returns `true` if not and `false` otherwise.
- $trust_v : \{i\} \rightarrow \{\texttt{true}, \texttt{false}\}$ — the trust function of verifier v returns a boolean indicating whether it trusts issuer i.
- $request_v : \mathcal{E} \nrightarrow \{f\}$ — verifier v can request to fiduciary f an *execution*, which has: (i) a *plan* with statements in the language $\mathcal{L}(C \cup R)$ and no program P, as P will be created by f; and (ii) a set of NMs where program P will run.
- $present_f : \wp(C) \times \wp(\mathcal{R}) \times \wp(\mathcal{L}(C \cup \mathcal{R})) \nrightarrow \{P\}$ — presenting function takes sets of credentials, revocation registries, and statements in $\mathcal{L}(C \cup R)$ as input and produces a program P, ensuring that the execution of P eliminates any uncertainty regarding the truthfulness of the inputted statements, while simultaneously safeguarding the confidentiality of the associated credentials, *i.e.*, without revealing additional information about the credentials.

- $authz_f : \mathcal{E} \times \wp(\mathcal{L}(\mathcal{E})) \rightarrow \{\texttt{true}, \texttt{false}\}$ — fiduciary f's authorization decision function to authorize or reject the *execution*, *i.e.,* running a *plan* in a set of NM, also receives a set of consent statements $\mathcal{L}(\mathcal{E})$, and returns `true` if *execution* may be carried out or `false` otherwise.
- $exec_{v,f} : \mathcal{E} \rightarrow \{(H, M)\}$ — running a *execution*, *i.e.,* running a *plan* in a set of NM, results in producing a pair containing the execution history H and a set of metadata M. Additionally, *plan*'s result register r_y, and an evidence $e = (H' \subseteq H, M' \subset M)$ are shared with v and f.
- $verify_v : E \times R \rightarrow \{\texttt{true}, \texttt{false}\}$ — verification function takes as input the evidence $e \in E$, and the output register $r_y \in R$ produced by running the *execution*. It returns `true` if the *execution* holds and `false` otherwise.
- $fiduciary_f : s \rightarrow \{\texttt{true}, \texttt{false}\}$ — fiduciary function takes as input a subject s and returns whether fiduciary f has a fiduciary relationship with subject s or not.
- $announce_s : \{f\} \rightarrow \{i, v\}$ — subject s announces their fiduciary f to issuer i or verifier v.
- $express_s : \wp(\mathcal{L}(\mathcal{E})) \nrightarrow \{f\}$ — subject s express function receives sets of statements about the execution of plans in language $\wp(\mathcal{L}(\mathcal{E}))$ describing the subject's consent about the usage of their data in different contexts to fiduciary f.
- $audit_s : \wp(\mathcal{L}) \times \{f\} \nrightarrow \mathbb{R} \cup \mathbb{A} \cup \mathbb{F}$ — audit function of subject s receives statements $\wp(\mathcal{L})$ and a fiduciary f, returning any information in $\mathbb{R} \cup \mathbb{A} \cup \mathbb{F}$ that the fiduciary has.

Figure 5 encapsulates the proposed architecture, featuring a visually intuitive representation that includes all previously established components and the addition of dashed directional arrows to convey nuanced associations embedded into the model's mathematical definitions. The rounded rectangles symbolize roles, the square-cornered rectangles represent computational artifacts, and the continuous directional arrows depict established functions. Notably, the dashed directional arrows introduce a layer of semantic richness by conveying supporting associations embedded into the model's definitions. Each dashed arrow serves as a visual cue, providing insights into relationships such as ownership ("owns"), relevance ("about"), composition ("of"), connection ("to" and "outputs to"), storage ("stores"), and functionality ("uses").

4.4 High-Level Protocols

We introduce two key high-level protocols to demonstrate typical interactions within the Fiduciary Identity model, emphasizing secure and private identity information handling. These protocols illustrate how the model manages credential requests and verification processes, ensuring adherence to user-defined consent policies.

4.4.1 Protocol 1: Credential Request and Management.

(1) *Credential Request Initiation*:
Subject s initiates a credential request to issuer i, signaling their fiduciary f using $announce_s$ to manage subsequent operations.
(2) *Credential Verification*:
Issuer i confirms the identity of subject s and verifies if they fulfill the criteria for the requested credential c.
(3) *Credential Issuance*:
Upon meeting the requirements, issuer i generates the credential c and a matching revocation record r using $issue_i$.
(4) *Credential Transfer*:
Issuer i transmits the credential c and revocation record r to the subject's fiduciary f, who securely stores them.
(5) *Consent Policy Granting*:
Subject s assigns, via $express_s$, a set of consent policies in language $\mathcal{L}(\mathcal{E})$. These policies dictate the conditions under which fiduciary f can use $present_f$ to create programs for plans in executions.

4.4.2 Protocol 2: Identity Verification for Service Access.

(1) *Service Access Request*:
Subject s seeks a service from verifier v, informing via $announce_s$ that fiduciary f will handle further interactions.
(2) *Verifier's Execution Proposal*:
Verifier v proposes an $execution$ to fiduciary f using $request_v$, detailing credential requirements in a *plan* and actions to be executed on NM_v.
(3) *Fiduciary's Authorization Check*:
Fiduciary f evaluates the proposed $execution$ against consent policies in $\mathcal{L}(\mathcal{E})$ using $authz_f$ and denies it as unauthorized, notifying verifier v.
(4) *Revised Plan Submission*:
Verifier v submits a revised $execution'$, specifying the same plan to be executed on fiduciary's device NM_f.
(5) *Fiduciary's Plan Approval*:
Fiduciary f reassesses $execution'$ and, finding it compliant with $\mathcal{L}(\mathcal{E})$, grants authorization.
(6) *Credential Preparation*:
Fiduciary f selects relevant credentials C_s and their revocation registries $\mathcal{R}_s$, and checks their status using $status_i$.
(7) *Statement Generation*:
Fiduciary f generates a set of statements $\mathcal{L}_s$ in language $\mathcal{L}(C \cup \mathcal{R})$ about credentials C_s and their status R_s.
(8) *Program Creation*:
By using $present_f(C_s, R_s, \mathcal{L}_s)$, fiduciary f builds program P with necessary credential assertions and revocation status
(9) *Execution and Evidence Generation*:
Fiduciary f updates $execution'$ with P, executing it on NM_f to produce output r_y and evidence $e^{execution'}$. The latter is sanitized producing evidence e''. Outputs and sanitized evidence are sent to verifier v.
(10) *Verifier's Final Verification*:
Verifier v receives r_y and e'', conducting a verification with $verify_v(r_y, e'')$.
(11) *Service or Data Access Decision*:
If $verify_v(r_y, e'')$ returns `true`, then verifier v accepts the $execution'$ and grants access to the requested service or data. If it fails, verifier v denies access.

These protocols showcase the Fiduciary Identity model's use of consent policies $\mathcal{L}(\mathcal{E})$ in decision-making, ensuring that fiduciary actions align with user authorization. Executing plans within the fiduciary's trusted environment NM_f heightens the security and privacy of the process.

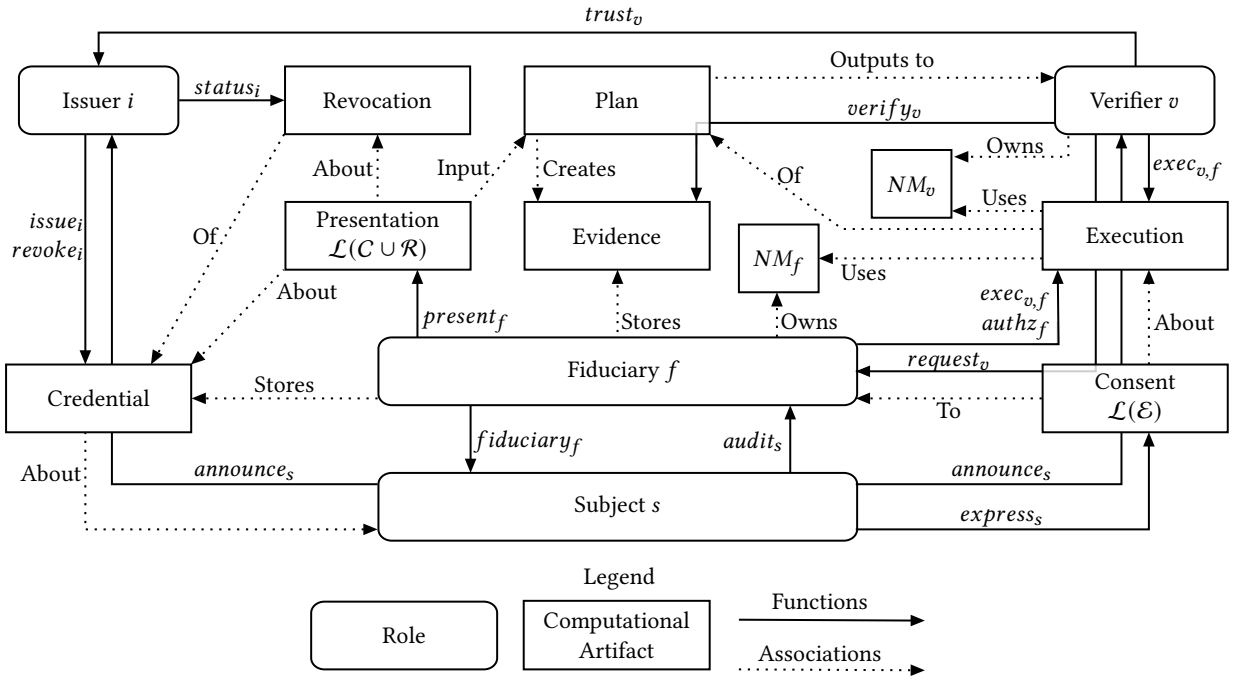


Figure 5: The Fiduciary Identity model: rounded rectangles for roles, square-cornered rectangles for computational artifacts, continuous arrows for functions, and dashed arrows for supporting associations.

4.5 Security Considerations

Like any other digital identity framework, the Fiduciary Identity model warrants careful consideration of security aspects. One of the central concerns in this model is the safeguarding of personal data. This model relies on fiduciary entities to vouch for individuals. This places significant responsibility on fiduciaries to protect user information from data breaches or unauthorized access.

The architecture's reliance on fiduciaries introduces potential vulnerabilities, making it crucial to establish robust identity verification and data protection mechanisms. Additionally, secure key management and encryption practices are essential to prevent identity theft and fraud, as malicious actors may attempt to compromise the fiduciary relationship between the subject and the fiduciary.

Specific security measures, including the definition of attack surfaces and threat models, can vary depending on how the Fiduciary Identity model is instantiated. Different applications and implementations may face unique security challenges and adversaries. Therefore, any protocols or systems implementing this model must conduct a rigorous and context-specific analysis of potential security threats and vulnerabilities. This process should include a detailed attack surface assessment, identifying potential weak points where malicious actors might exploit the system. By tailoring security considerations to the particular instantiation of the Fiduciary Identity model, developers can design more resilient and adaptive security measures to protect digital identities effectively.

4.6 Discussion

Although the proposed mathematical formalization provides a fundamental building block for describing interactions in the Fiduciary Identity model, more is needed to express concrete interactions among the subject, fiduciary, issuer, and verifier. Because we are proposing a model and not a specific instance of this model, many implementation elements need to be filled in by practitioners. For instance, the model does not provide details about how entities transfer information, *e.g.*, how subjects communicate consent or perform audits on fiduciaries. Moreover, expressing consent policies to a fiduciary is not simple, as it requires defining expressive semantics that enable the fiduciary to reason about unforeseen requests from verifiers. Finally, developing two-party computation is difficult, which this model supports (see the definition of $exec_{v,f}$).

Table 2 presents a comprehensive comparison of digital identity models, each with its unique set of features. Notably, one of the most striking features of this table is the decentralization level, which highlights how different models vary in terms of centralization. While centralized and outsourced IdP models have relatively low decentralization levels, relying heavily on a single entity (IdP), the SSI and Fiduciary Identity models achieve a high level of decentralization, placing greater control in the hands of the individual (subject) or fiduciary.

Another essential feature to analyze is the usability cognitive load, which we classified according to the three-level scale (low, medium, and high) used in [8]. Only the fiduciary and the outsourced IdP models have a low cognitive load, as the management of user information is made by someone else. Nonetheless, in the outsourced IdP model, the user has to accept or deny sharing their data with SPs. In the fiduciary model, authorization decisions are made by the fiduciary following the subject's consent rules.

Table 2: Comparison between the main characteristics of the digital identity models.

IAM Model	Decentralization Level	Driving Relationship	Usability Cognitive Load	Private Data Manager	Consent Management	Compute Site
Centralized	low	dependence	medium	SP	none	SP
Outsourced IdP	medium	trust	low	IdP	on demand	SP
SSI	high	trust	high	subject	on demand	SP
Fiduciary Identity	high	fiduciary	low	fiduciary	a priori; on demand; or a posterior	SP; fiduciary; or both

Another noteworthy feature is the private data manager category. In the Fiduciary Identity model, the fiduciary manages private data, emphasizing the role of trusted intermediaries. This distinct characteristic sets it apart from other models, where management may fall on a service provider or the individual themselves. The variety in these approaches to data management showcases how different models handle user privacy and data protection, aligning with their overarching principles and trust relationships.

5 FINAL REMARKS

In conclusion, this paper has navigated the intricate realm of digital identity models, revealing the diverse spectrum of approaches in the ever-evolving digital landscape. From centralized and outsourced IdP models to the decentralized realm of SSI, each model offers unique trade-offs, implications, and considerations.

Notably, the Fiduciary Identity model emerges as a promising innovation, offering distinct advantages in terms of usability. Through the fiduciary relationship, this model significantly reduces the cognitive load on users, simplifying the digital identity experience. Unlike other models where users must grapple with complex identity management tasks, the Fiduciary Identity model entrusts these responsibilities to a fiduciary, streamlining user interactions. This reduction in cognitive load enhances user convenience. By relieving individuals from the intricacies of identity management, the Fiduciary Identity model paves the way for a more user-friendly and accessible digital identity landscape.

REFERENCES

[1] Christopher Allen. 2016. The path to self-sovereign identity. Available online: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html (accessed on 04 June 2023).

[2] Chirag Arora. 2019. Digital health fiduciaries: protecting user privacy when sharing health data. *Ethics and Information Technology* 21, 3 (2019), 181–196. https://doi.org/10.1007/s10676-019-09499-x

[3] Jack M Balkin. 2015. Information fiduciaries and the first amendment. *U.C. Davis Law Review* 49 (2015), 1183.

[4] Jan Camenisch and Anna Lysyanskaya. 2002. Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. In *Advances in Cryptology — CRYPTO 2002*, Moti Yung (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 61–76. https://doi.org/10.1007/3-540-45708-9_5

[5] Deborah A DeMott. 1988. Beyond metaphor: An analysis of fiduciary obligation. *Duke Law Journal* (1988), 879.

[6] Dick Hardt. 2012. The OAuth 2.0 Authorization Framework. https://doi.org/10.17487/RFC6749

[7] John Hughes and Eve Maler. 2005. Security Assertion Markup Language (SAML) V2.0 Technical Overview. Available online: https://www.oasis-open.org/committees/download.php/14361/sstc-saml-tech-overview-2.0-draft-08.pdf (accessed on 04 June 2023).

[8] M Asif Khawaja, Fang Chen, and Nadine Marcus. 2014. Measuring cognitive load using linguistic features: implications for usability evaluation and adaptive interaction design. *International Journal of Human-Computer Interaction* 30, 5 (2014), 343–368. https://doi.org/10.1080/10447318.2013.860579

[9] Michael Kuperberg. 2019. Blockchain-Based Identity Management: A Survey From the Enterprise and Ecosystem Perspective. *IEEE Transactions on Engineering Management* 67, 4 (2019), 1–20. https://doi.org/10.1109/TEM.2019.2926471

[10] Jeonghyuk Lee, Jaekyung Choi, Hyunok Oh, and Jihye Kim. 2021. Privacy-preserving Identity Management System. *Cryptology ePrint Archive, Report 2021/1459* (2021).

[11] Jeonghyuk Lee, Jungyeon Hwang, Jaekyung Choi, Hyunok Oh, and Jihye Kim. 2019. SIMS: Self-Sovereign Identity Management System with Preserving Privacy in Blockchain. *IACR Cryptology ePrint Archive 2019/1241* (2019).

[12] Alexander Mühle, Andreas Grüner, Tatiana Gayvoronskaya, and Christoph Meinel. 2018. A survey on essential components of a self-sovereign identity. *Computer Science Review* 30 (11 2018), 80–86. https://doi.org/10.1016/j.cosrev.2018.10.002

[13] David Recordon and Drummond Reed. 2006. OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*. ACM, Alexandria, USA, 11–16. https://doi.org/10.1145/1179529.1179532

[14] Leonard I Rotman. 2011. Fiduciary Law's Holy Grail: Reconciling Theory and Practice in Fiduciary Jurisprudence. *Boston University Law Review* 91 (2011), 921.

[15] Karen Scarfone and Murugiah Souppaya. 2009. 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).

[16] Martin Schanzenbach., Thomas Kilian., Julian Schütte., and Christian Banse. 2019. 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,*. INSTICC, SciTePress, Prague, Czech Republic, 325–332. https://doi.org/10.5220/0007772903250332

[17] Frederico Schardong and Ricardo Custódio. 2022. Self-Sovereign Identity: A Systematic Review, Mapping and Taxonomy. *Sensors* 22, 15 (2022), 5641. https://doi.org/10.3390/s22155641

[18] C. P. Schnorr. 1990. Efficient Identification and Signatures for Smart Cards. In *Advances in Cryptology — CRYPTO' 89 Proceedings*, Gilles Brassard (Ed.). Springer New York, New York, NY, 239–252. https://doi.org/10.1007/0-387-34805-0_22

[19] J. C. Shepherdson and H. E. Sturgis. 1963. Computability of Recursive Functions. *J. ACM* 10, 2 (April 1963), 217–255. https://doi.org/10.1145/321160.321170

[20] Manu Sporny, Dave Longley, and David Chadwick. 2017. Verifiable Credentials Data Model 1.0. Available online: https://www.w3.org/TR/vc-data-model/ (accessed on 04 June 2023).

[21] Shuhaili Talib, Nathan L Clarke, and Steven M Furnell. 2010. An Analysis of Information Security Awareness within Home and Work Environments. In *2010 International Conference on Availability, Reliability and Security*. IEEE, IEEE, Krakow, Poland, 196–203. https://doi.org/10.1109/ARES.2010.27

[22] Débora Pandolfi Álves. 2007. Equivalência de Máquinas Universais: Demonstração, Análise e Simulação.