frontiers in Blockchain

ORIGINAL RESEARCH
published: 28 July 2021
doi: 10.3389/fbloc.2021.626726

Sovrin: An Identity Metasystem for Self-Sovereign Identity

*Phillip J. Windley**

Office of Information Technology, Brigham Young University, Provo, UT, United States

Solving the problems of digital identity in a holistic manner requires that we rethink how we architect identity online. This paper presents the architecture of an identity metasystem called the Sovrin Network that aims to improve the user experience, increase flexibility, and reduce overall costs while supporting better privacy and security. We discuss the problems of online identity on the modern internet, discuss the nature of digital relationships, explore the architectures of identity systems, and detail the combination of these concepts into a comprehensive metasystem for solving the problems of online identity.

Keywords: self-sovereign identity, identity, cryptography, Sovrin, identifiers, verifiable credentials

OPEN ACCESS

Edited by:
Michael Shea,
Independent researcher, Litchfield, Connecticut, United States

Reviewed by:
Prerna Tambay,
Consultant, London, United Kingdom
Maryline Laurent,
Télécom SudParis, France

***Correspondence:**
Phillip J. Windley
pjw@byu.edu

Specialty section:
This article was submitted to Blockchain for Good, a section of the journal Frontiers in Blockchain

Received: *06 November 2020*
Accepted: *14 July 2021*
Published: *28 July 2021*

Citation:
Windley PJ (2021) Sovrin: An Identity Metasystem for Self-Sovereign Identity. Front. Blockchain 4:626726. doi: 10.3389/fbloc.2021.626726

INTRODUCTION

The internet was designed without an identity layer, at least for people (Cameron, 2005). At the time, any network user was identified by proxy through the machine they used to connect and whatever access control system it had. Personal computers and the web led to an internet where many people are online without any sponsoring organization. But the administrative model was so entrenched in the architecture of the internet that we simply perpetuated it with a different administrative identity system, username, and password for every relationship on every site and app.

The internet is a metasystem—a system of systems. The internet is not so much a communications system as it is a system for building communication systems. Metasystems employ protocols, governance, and convention to provide decentralized interoperability between the systems they comprise.

Naturally, an identity system for the internet should be a metasystem as well since no single system can meet the needs of every digital relationship. An identity metasystem is a system for building interoperable identity systems. The concept of an identity metasystem was first introduced by Kim Cameron in 2005 (Cameron, 2005). In describing this system, Cameron said:

> We need a unifying identity metasystem that can protect applications from the internal complexities of specific implementations and allow digital identity to become loosely coupled. This metasystem is in effect a system of systems that exposes a unified interface. . .

An identity metasystem provides the building blocks and protocols necessary for others to build identity systems that meet the needs of any specific context or domain.

This paper explores the architecture of an identity metasystem called the Sovrin Network. An identity metasystem like Sovrin is a prerequisite for an online world where identity is as natural as it is in the physical world. An identity metasystem can remove the friction, decrease cognitive overload, and make online interactions more private and secure.

THE PROBLEMS OF ONLINE IDENTITY

The internet's missing identity layer has resulted in a mishmash of one-off identity systems because every web site, service provider, and application has solved the problem in a unique way (Simmonds, 2015). As a result, people and organizations who use the internet are subject to cognitive overload, friction, increased costs, loss of privacy, and even outright fraud.

Fixing the internet's identity problem is hard. There have been numerous systems, protocols, and standards proposed over the past 20 years (Naik and Jenkins, 2016). While most of them have provided improvements and fixed specific problems, none have offered a holistic solution.

To see why digital identity is so hard, consider the following specific problems that make identity online different from the physical world.

Proximity—Because we are not interacting with people physically, our traditional means of knowing who we are dealing with are useless. None of the familiar signals of the physical world are present. Consequently, it is difficult to reliably recognize and remember people and organizations online (Andrieu 2018). Organizations have built administrative identity systems to serve their own needs in recognizing and remembering their customers, but people do not have the same capabilities. Consequently, we are mired in myriad, incompatible systems built for narrow purposes.

Autonomy—Each of these administrative systems is built for the convenience of the organization who controls it. Design choices for these systems are made to maximize the legibility of people to the organization for its purposes, skewing the balance of power toward the organization (Windley 2020). Consequently, people have very few natural rights and little leverage online. Current online identity systems significantly reduce individual freedom and autonomy.

Flexibility—Closely related to the autonomy problem is one of flexibility. Current online identity systems are built for very narrow purposes. But real life is messy, with billions of use cases (Windley 2018). People are innovative and infinitely diverse. None of us presents the same picture of ourselves to everyone and everything—how we recognize, remember, and respond to others is highly dependent on the context.

Privacy—No one will be surprised to learn that computers are very good at pattern matching. But a consequence of this is that online identity has very different implications for privacy than physical world interactions (Hardman 2019a). When you hand your driver's license to the bartender to establish your legal age, you would be surprised if she could remember all the detailed information it contains, like your address, and do that for every customer she encountered. Computers, on the other hand, retain a perfect memory of all the information they are presented with until they are told to forget.

Anonymity—Anonymity is closely related to privacy. In real life, we do without identity systems for most things. You do not have to identify yourself to the movie theater to watch a movie or log into some system to sit in a restaurant and have a private conversation with friends. Many of our interactions in the physical world are naturally anonymous because they are ephemeral. The ticket taker at a movie theater does "identify" you momentarily for purposes and checking your ticket, but that connection is short-lived and thus anonymous for most purposes. Many online interactions could make use of ephemeral relationships as well to better support privacy.

Interoperability—A consequence of myriad identity silos is that we are unable to carry context from system to system (Simmonds, 2015). Your friend in one system might have a different identifier in another. Consequently, your ability to recognize and remember varies from system to system.

Scale—There are billions of people online. Each of them has dozens, even hundreds of relationships. The internet of things promises to increase that by several orders of magnitude. Consequently, a general-purpose identity system needs to account for trillions of relationships between the many billions of people, organizations, and things that make up the online world. No single, centralized system can do it.

Solving these problems requires building something more abstract and general than the one-off, context-specific identity systems of the past.

RELATIONSHIPS

Identity systems exist to support online relationships. Managing identity information is merely a means to an end. In *Identities Evolve: Why Federated Identity is Easier Said than Done* (Wilson, 2011), Steve Wilson argues that the goal of using federation schemes to create a few identities that serve all purposes is deeply flawed. Wilson's point is that we have hundreds, even thousands, of online identities because we have lots of relationships. The identity data for a given relationship is contextual and highly evolved to fit its specific niche.

Each relationship has a common root, the person being identified, but it is highly contextualized. Some relationships are long-lived, some are ephemeral. Some are personal, some are commercial. Some are important, some are trivial. Still, we have them. The information about ourselves, what many refer to as identity data, that we share with each is adapted to the specific niche that the relationship represents. Once you realize this, the idea of creating a few online identities to serve all needs becomes preposterous.

Because of the proximity problem, we are not interacting with people physically and so our natural means of knowing who we are dealing with are useless. Joe Andrieu defines (Andrieu, 2018) identity as "how we recognize, remember, and respond to" another entity. I add "rely on" to the list.

These activities depend on three properties that any digital relationship must have to overcome the proximity problem:

Integrity—we want to know that, from interaction to interaction, we are dealing with the same entity we were before. In other words, we want to identify them so that we can recognize and remember them.

Lifespan—normally, we want relationships to be long-lived, although we also create ephemeral relationships for short-lived interactions.

Utility—we create online relationships in order to use them within a specific context.

Relationship Integrity

Integrity allows parties to a relationship to recognize each other. Consequently, all identity systems manage relationship integrity as a foundational capability. Federated identity systems improve on one-off, often custom, identity systems by providing integrity in a way that reduces user management overhead for the organization, increases convenience for the user, and increases security by eliminating the need to create one-off, proprietary solutions. An identity metasystem aims to establish relationship integrity with the convenience of the federated model but without relying on an intervening identity provider (IdP) in order to provide autonomy and privacy.

A relationship has two parties, let us call them P1 and P2[1]. P1 is connecting with P2 and, as a result, P1 and P2 will have a relationship. P1 and P2 could be people, organizations, or things represented by a web site, app, or service. Recognizing the other party in an online relationship relies on being able to know that you are dealing with the same entity each time you encounter them.

In the identity metasystem represented by the Sovrin Network, a relationship is initiated when P1 and P2 exchange decentralized identifiers (DIDs) (Decentralized Identifiers, 2020). For example, when a person visits a web site or app, they are presented with a connection invitation. When they accept the invitation, they use a software agent to share a DID that they created. In turn, they receive a DID from the web site, app, or service. We call this a "connection" since DIDs are cryptographically based and thus provide a means of both parties mutually authenticating. The user experience does not necessarily surface all this activity to the user[2].

In contrast to the federated model, the participants in the metasystem mutually authenticate and the relationship has integrity without the intervention of a third party because the identifiers are self-certifying (Smith, 2020). By exchanging DIDs, both parties have also exchanged public keys. They can consequently use cryptographic means to ensure they are interacting with the party who controls the DID they received when the relationship was initiated. Mutual authentication based on self-certifying DIDs provides SSI relationships with inherent integrity. P1 and P2 are peers since they both have equal control over the relationship.

In addition to removing the need for intermediaries to vouch for the integrity of the relationship, the peer nature of relationships in the Sovrin Network also means that neither party has access to the authentication credentials of the other. Mutual authentication means that each party manages their own keys and never shares the private key with another party. Consequently, attacks, like the recent attack on Twitter accounts (Conger and Popper, 2020) cannot happen because there is no administrator who has access to the credentials of everyone using the system—there is no trove of high-value data.

Relationship Lifespan

Relationships have lifespans. Some relationships are long-lived, some are short-term, and others are ephemeral, existing only for the duration of a single interaction. We typically do not think of it this way, but every interaction we have in the physical world, no matter for what purpose or how short, sets up a relationship. So too in the digital world, although our tools have been sorely lacking in support for anything by long-lived relationships.

The administrative identity systems we have built to service online relationships usually fail to recognize that some relationships are not permanent. Imagine that if whenever you stopped in the convenience store for a cup of coffee, you had to create a permanent relationship with the coffee machine, the cashier, the point of sale terminal, and the customers in line ahead and behind you? Sounds ridiculous. But that is what most digital interactions require. At every turn, we are asked to establish permanent accounts to transact and interact online.

There are several reasons for this. The biggest one is that every web site, app, or service wants to send you ads, at best, or track you on other sites, at worst. Unneeded, long-lived relationships have come to define the modern online experience and are the foundation of the surveillance economy that Shoshana Zuboff describes (Zuboff, 2020).

Relationship Utility

Relationships are established to provide utility. A university wants you to register for classes. An ecommerce site wants to sell you things. A social media site wants to show you ads. Thus, their identity systems, built around the IAM (identity and access management) system, are designed to do far more than just establish the integrity of the relationship. They want to store data about you and your activities.

Thus, any identity system is much larger and more specialized than the IAM portion. All of the account or profile data these companies use are properly thought of as part of the identity system that they build and run. Returning to Joe Andrieu (Andrieu, 2018):

> Identity systems acquire, correlate, apply, reason over, and govern (the) information assets of subjects, identifiers, attributes, raw data, and context.

Regardless of whether or not they outsource the integrity of their relationships using federation, companies still have to keep track of the relationships they have with customers or users in order to provide the service they promise. They cannot outsource this to a third party because the data in their identity system have evolved to suit the needs of the specific relationship. We will never have a single identity that serves all relationships because their unique contexts demand their own identity data. Change the identity system in a Netflix or Amazon and it will not be the same company anymore.

This leads us to a simple, but important conclusion: You cannot outsource a relationship. Online apps and services

[1]For simplicity, we limit the discussion to two-party relationships, but the model can be generalized to multi-party relationships.

[2]To get a feel for the user experience, see the demo at https://try.connect.me.

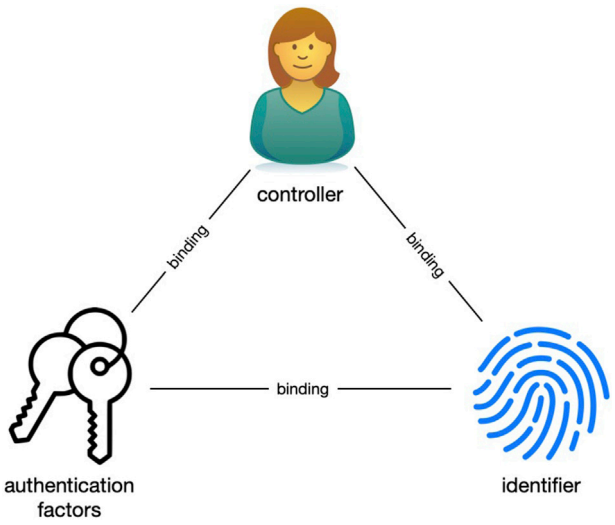

FIGURE 1 | Binding of controller, authentication factors, and identifiers in identity systems.

decorate the relationship with information they observe and use that information to provide utility to the relationships they administer. Doing this and doing it well is the foundation of the modern web.

Consequently, the bad news is that an identity metasystem does not reduce the need for companies to build, manage, and use identity systems. Their identity systems are what make them what they are—there is no "one size fits all" model. But the identity metasystem does make the relationships they form richer, provides a more balanced relationships by providing symmetric value to all parties, and increases flexibility and privacy.

THE ARCHITECTURE OF IDENTITY SYSTEMS

To understand how an identity metasystem like Sovrin Network supports better online relationships, it is useful for clearly understanding the architectures of identity systems.

As we said, identity systems provide the means necessary for remembering, recognizing, and relying on the other parties to the relationship. To do so, they use identifiers, convenient handles that name the thing being remembered. Identifiers are unique within some namespace. The namespace gives context to the identifiers since the same string of characters might be a phone number in one system and a product ID in another.

As shown in **Figure 1**, identifiers are issued to or created by a *controller* who by virtue of knowing the *authentication factors* can make authoritative statements about the identifier (e.g., claiming it by logging in). The controller might be a person, organization, or software system. The controller might be the subject that the identifier refers to, but not necessarily. The authentication factors might be a password, key fob, cryptographic keys, or something else. The strength and nature of the *bindings* between the controller, authentication factors, and identifier determine the strength and nature of the relationships built on top of them.

To understand why that is so, we introduce the concept of a *root of trust*. A root of trust is a foundational component or process in the identity system that is relied on by other components of the system and whose failure would compromise the integrity of the bindings. A primary root of trust cannot be replaced, while a secondary root of trust can be. Together, the roots of trust form the trust basis for the system.

The trust basis enabled by the identity system underlies a particular *trust domain*. The trust domain is the set of digital activities that depend on the binding of the controller to the identifier. For example, binding a customer to an identifier allows Amazon to trust that the actions linked to the identifier are authorized by the controller. Another way to look at this is that the strength of the binding between the identifier and customer (controller) determines the risk that Amazon assumes in honoring those actions.

The strength of the controller–identifier binding depends on the strength of the binding between the controller and the authentication factors and between the authentication factors and the identifier. Attacking either of those bindings reduces the trust we have in the controller–identifier binding and increases the risk that actions taken through a particular identifier are unauthorized.

We can place all identity systems into one of three broad architectural categories based on their structure and primary root of trust:

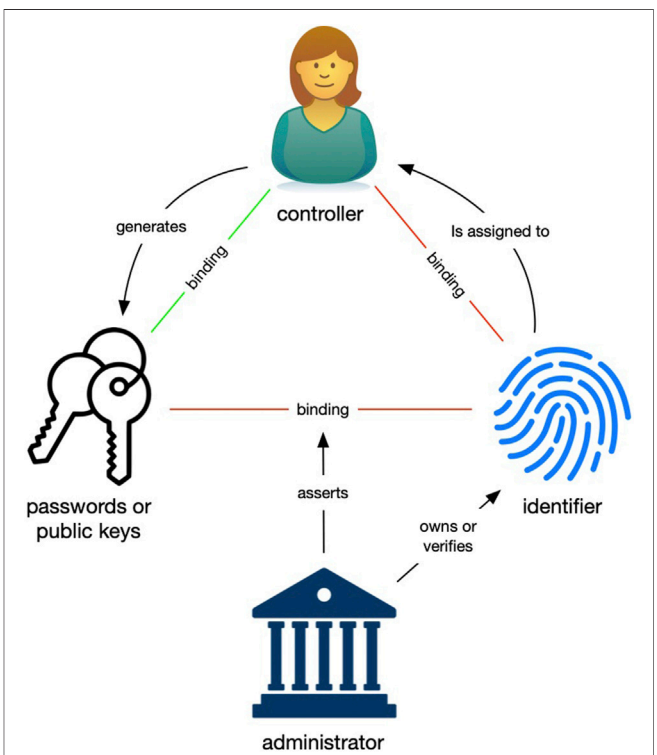

FIGURE 2 | The trust basis in administrative identity systems.

- Administrative
- Algorithmic
- Autonomic

These architectures differ in who controls what. Knowing the locus of control is the primary factor in determining the basis for trust for each. We call this *control authority*. The entity with control authority takes action through operations that affect the creation (inception), updating, rotation, revocation, deletion, and delegation of the authentication factors and their relation to the identifier. How these events are ordered and their dependence on previous operations is important. The record of these operations is the *source of truth* for the identity system.

Administrative Architecture

Identity systems with an administrative architecture rely on an administrator to bind the identifier to the authentication factors. The administrator is the primary root of trust for any domain with an administrative architecture. Almost every identity system in use today has an administrative architecture and their trust basis is founded on the administrator.

Figure 2 shows the interactions between the controller, identifier, and authentication factors in an administrative identity system, the role of the administrator, and the impact these have on the strength of the bindings.

The controller usually generates the authentication factors by choosing a password, linking a two-factor authentication (2FA) mechanism, or generating keys. Even though the identifier might be the controller's email address, phone number, public key, or other ID, the administrator "assigns" the identifier to the controller because it is their policy that determines which identifiers are allowed, whether they can be updated, and their legitimacy within the identity system's domain. The administrator "owns" the identifier within the domain.

The administrator also asserts the binding between the identifier and the authentication factors. An employee's mistake, a policy change, or a hack could affect the binding between the identifier and authentication factors or the identifier and the controller. Consequently, these bindings are relatively weak. Only the binding between the controller and authentication factors is strong because the controller generates them.

The administrator's primary duty is to authoritatively assert the binding between the controller and identifier. Authoritative control statements about the identifier are recorded in the administrator's database, the source of truth in the system, subject to retroactive change by employees and hackers. The administrator might be an ecommerce site that maintains an identity system as the basis for its customer's account. In this case, the binding is private, and its integrity is of interest only to the web site and the customer. Alternatively, the administrator might provide federated login services. In this case, the administrator is asserting the controller–identifier binding in a semi-public manner to anyone who relies on the federated login. A certificate authority is an example of an administrator who publicly asserts the controller–identifier binding, signing a certificate to that effect.

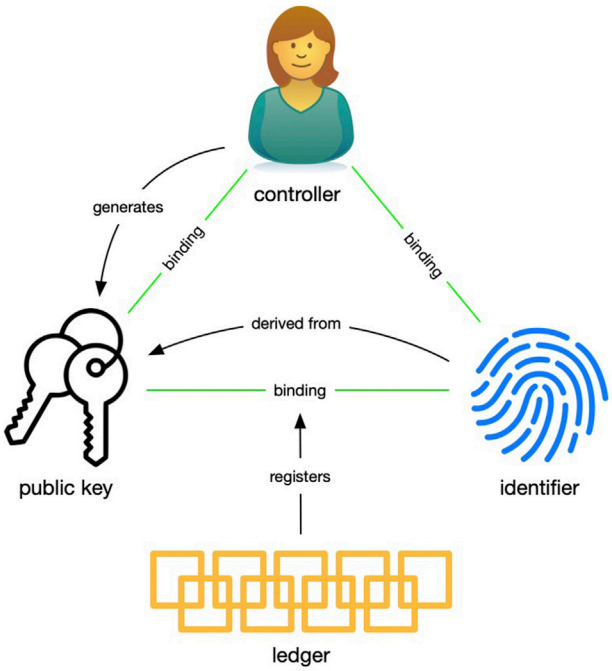

FIGURE 3 | The trust basis in algorithmic identity systems.

Because the administrator is responsible for binding the identifier to both the authentication factors and the controller, the administrator is the primary root of trust and thus the basis for trust in the overall system. Regardless of whether the binding is private, semi-public, or public, the integrity of the binding is entirely dependent on the administrator and the strength of their infrastructure, policies, employees, and continued existence. The failure of any of those can jeopardize the binding, rendering the identity system unusable by those who rely on it.

Algorithmic Architecture

Identity systems that rely on a ledger have an algorithmic architecture. I'm using "ledger" as a generic term for any algorithmically controlled, distributed-consensus-based datastore including public blockchains, private blockchains, distributed file systems, and others. Of course, it is not just algorithms. Algorithms are embodied in code, written by people, running on servers. How the code is written, its availability to scrutiny, and the means by which it is executed all impact the trust basis for the system. "Algorithmic" is just shorthand for all of this.

Figure 3 shows how the controller, authentication factors, identifier, and ledger are bound in an identity system with an algorithmic architecture. As in the administrative identity system, the controller generates the authentication factors, albeit in the form of a public–private key pair. The controller keeps and does not share the private key. The public key, on the other hand, is used to derive an identifier (at least in well-designed SSI systems) and both are registered on the ledger. This registration is the inception of the controller–identifier binding since the controller can use the private key to assert her control over the identifier as

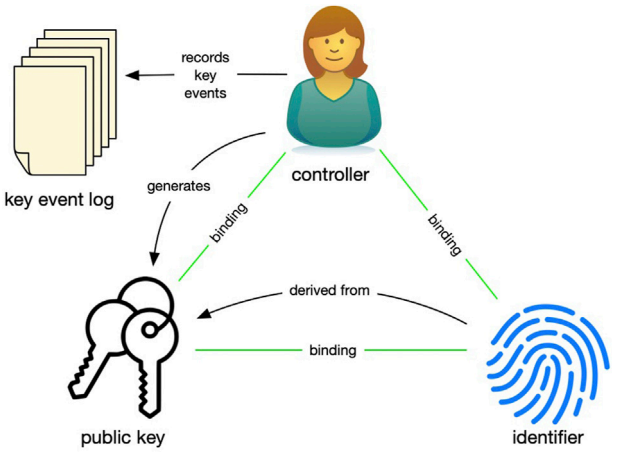

FIGURE 4 | Trust basis in autonomic identity systems.

registered on the ledger. Anyone with access to the ledger can algorithmically validate the controller–identifier binding.

The controller makes authoritative control statements about the identifier. The events marking these operations are recorded on the ledger, which becomes the source of truth for anyone interested in the binding between the identifier and authentication factors.

In an identity system with an algorithmic trust basis, computer algorithms create a ledger that records the key events. The point of the ledger is that no party has the power to unilaterally decide whether these records are made, modified, or deleted and how they are ordered. Instead, the system relies on code executed in a decentralized manner to make these decisions. The nature of the algorithm, the manner in which the code is written, and the methods and rules for its execution all impact the integrity of the algorithmic identity system and consequently any bindings that it records.

Autonomic Architecture

Identity systems with an autonomic architecture function similarly to those with an algorithmic architecture. As shown in **Figure 4**, the controller generates a public–private key pair, derives a globally unique identifier, and shares the identifier and the currently associated public key with the party she wishes to create a relationship with.

The controller uses her private key to authoritatively and non-repudiably sign statements about the operations on the keys and their binding to the identifier, storing those in an ordered key event log[3]. One of the important realizations that make autonomic identity systems possible is that the key event log must only be ordered in the context of a single identifier, not globally. So, a ledger is not needed for recording operations on identifiers that need not be publicly validated. The key event log can be shared with and verified by anyone.

The controller also uses the private key to sign statements that authenticate herself and authorize use of the identifier. A digital signature also provides the means of cryptographically responding to challenges to prove her control of the identifier. These self-authentication and self-authorization capabilities make the identifier self-certifying and self-managing, meaning that there is no external third party, not even a ledger, needed for the controller to manage and use the identifier and prove to others the integrity of the bindings between herself and the identifier. Thus, anyone (any entity) can create and establish control over a personal identifier namespace in a manner that is independent, interoperable, and portable without recourse to any central authority. Autonomic identity systems rely solely on self-sovereign authority.

Autonomic identifiers have a number of advantages:

- Self-Certification—autonomic identifiers have no reliance on a third party.
- Self-Administration—autonomic identifiers can be independently administered by the controller without reliance on a third party.
- Low Cost—autonomic identifiers are virtually free to create and manage.
- Security—because the keys are decentralized, there is no trove of secrets that can be stolen.
- Regulatory—autonomic identifiers need not be publicly shared or stored in an organization's database, and consequently reduce regulatory concern over personal data.
- Scale—autonomic identifiers scale with the combined computing capacity of all participants, not a central system.
- Independent—autonomic identifiers are not dependent on any specific technical system or even being online.

CREDENTIAL EXCHANGE AS THE FOUNDATION FOR ONLINE IDENTITY

In the physical world, people collect and manage credentials from various sources including governments, financial institutions, employers, schools, businesses, family, colleagues, and friends. Individuals also assert information themselves. These various credentials serve different purposes. We have credentials that we use often and carry around with us. We have important credentials we file away and even some we keep in safe deposit boxes. Some, like boarding passes, we use once, then throw away. Others, like birth certificates, we keep for our entire life.

We use credentials, alone or in concert with other credentials, when we need to prove something about ourselves. We present credit cards to prove we are authorized to charge an account. We present a driver's license to prove we are of legal age at a bar. We present letters from our employer to prove our salary when applying for a loan. The credential verifier is free to determine whether to trust the credential or not.

[3]A number of cryptographic systems are trivially self-certifying (e.g., PGP, Ethereum, and Bitcoin). What sets the autonomic identity systems described here apart is the key event log.

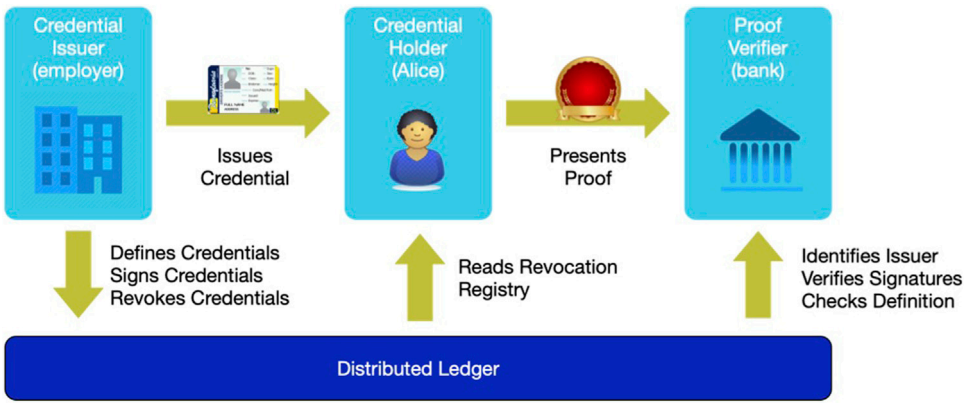

FIGURE 5 | Parties to credential exchange and their interactions with each other and the ledger.

Identity does not work that way online. As we have discussed, online identity has traditionally been administrative, centralized, and built for specific purposes. Various, so-called "identity providers" authenticate people using usernames and passwords and provide a fixed, usually limited, set of attributes about the subject of the identity transaction. The identity information from these systems is usually used within a specific, limited context. For example, federated login (e.g., Log in with Google) allows a login to be used across contexts, but the kind of information shared is limited and its provenance is often difficult to determine. These various administrative identity systems are not interoperable, making it hard to combine attributes from one with those of another. Consequently, online identity is one-dimensional and has limited value.

In credential exchange, there are three parties: the credential issuer, the credential holder (sometimes called the identity owner), and the credential verifier (also known as the relying party).

A credential is a collection of claims (i.e., attributes) that is signed by the issuer and held by the identity owner. Credentials conform to the Verifiable Credential specification (Sporny et al., 2019). While the word "credential" conjures images of formal documents, almost anything representable in JSON that needs to be attested can be a credential. So, while things like passports and driver's licenses fit this bill, so do things like membership cards, boarding passes, school report cards, invoices, purchase orders, and store receipts.

Figure 5 shows how credential exchange works. Suppose Alice (the identity owner) is applying for a loan at her local bank (the credential verifier). The bank requires proof that Alice is employed and makes at least $70,000 per year. Alice's employer (the credential issuer) has issued an employment credential that includes her employment status and her current salary. The credential might also include many other attributes related to Alice's job. Alice holds the employment credential and can present it to prove to the bank that she is employed and makes more than $70,000.

When Alice proves her employment status to the bank online, she does not present the entire credential since doing so would reveal more information than is necessary. Instead, Alice presents just the information the bank needs using a cryptographic technique known as "zero knowledge proof." The ability to limit the information presented from a credential is important to maintain privacy through the principle of minimal disclosure.

The online model for verifiable credentials has five important characteristics that mirror how credentials work in the offline world:

- Credentials are decentralized and contextual. There is no central authority for all credentials. Every party can be an issuer, a holder (identity owner), or a verifier. Verifiable credentials can be adapted to any country, any industry, any community, or any set of trust relationships.
- Credential issuers decide on what data are contained in their credentials. Anyone can write credential schemas to the ledger. Anyone can create a credential definition based on any of these schemas.
- Verifiers make their own decisions about which credentials to accept—there is no central authority who determines what credentials are important or which are used for what purpose.
- Verifiers should not need to contact issuers to perform verification. Credential verifiers do not need to have any specific technical, contractual, or commercial relationship with credential issuers.
- Credential holders are free to choose which credentials to carry and what information to disclose. People and organizations are in control of the credentials they hold (just as they are with physical credentials) and determine what to share with whom.

In addition to these five characteristics, credential presentment *via* zero-knowledge proofs offers important privacy protection to the credential holder (Lodder, 2018). ZKP presentment.

- increases the cost of correlation since the identifier of the holder is blinded and other data can be excluded if the verifier does not need it;

- reduces the parties with whom the data are shared;
- supports incremental disclosure as a relationship becomes more trusted;
- restricts the attributes that are shared to just the needed subset of what is contained in the credential; and
- empowers the holder to restrict resharing.

These powerful protections against correlation increase privacy in a structural way and make possible more effective regulation of verifiers.

RISK AND TRUST: CREDENTIAL FIDELITY AND PROVENANCE

Trust is a popular term in the identity community. Some people rightly ask about risk whenever someone in the identity community talks about trust. Because of the proximity problem, digital relationships are potentially risky. One of the goals of an identity system is to provide evidence that can be used in the risk calculation.

In their excellent paper, Risk and Trust (Nickel and Vaesen, 2012), Philip Nickel and Krist Vaesen define trust as the "disposition to willingly rely on another person or entity to perform actions that benefit or protect oneself or one's interests in a given domain." From this definition, we see why crypto proponents often say, "To trust is good, but to not trust is better." The point being that not having to rely on some other human, or human-mediated process is more likely to result in a beneficial outcome because it reduces the risk of non-performance.

Relationships imply a shared domain, context, and set of activities (Wilson, 2011). We often rely on third parties to tell us things relevant to the relationship. Our vulnerability, and therefore our risk, depends on the degree of reliance we have on another party's performance. Relationships can never be "no trust" because of the very reasons we create relationships. Bitcoin, and similar systems, can be low or no trust precisely because the point of the system is to reduce the reliance on any relationship at all.

The architecture of the identity metasystem significantly limits the ways we must rely on external parties for the exchange of information *via* verifiable credentials and thus reduces the vulnerability of parties inside and outside of the relationship. The design of the identity metasystem clearly delineates the parts of the system that are low trust and those where human processes are still necessary.

Our basis of trust in the physical world is other humans We interact with people directly, recognizing, remembering, responding to, and relying on them (Andrieu 2018). As we pointed out in *The Problems of Online Identity*, in the digital realm, we are not proximate to the parties we have a relationship with. As a result, credential exchange replaces the human basis of trust in the physical world with algorithmic and autonomic bases of trust.

Returning to the example in the last section, Alice's bank needs two levels of trust: first it needs to know the credential is authentic. Second, the bank wants to verify the veracity of the contents of the credential.

With respect to credential authenticity, the bank wants to know:

1. Who issued the credential,
2. That the credential was issued to Alice,
3. That the credential has not been tampered with, and
4. That the credential has not been revoked.

The metasystem provides these properties cryptographically (Hardman 2018). We call the properties that the metasystem provides *credential fidelity*. Fidelity is cryptographic. The bank can verify these four properties by looking at the credential definition on the ledger, retrieving the issuer's public DID from the definition, resolving the DID to get the public key of the issuer, and using the public key to check the signature of the credential to ensure it has not been tampered with. The bank can also cryptographically verify that the credential was issued to Alice. As part of making her proof from the credential, Alice also proves that it has not been revoked by referencing a revocation registry on the ledger. The ledger ensures that the bank can do all of this without contacting the employer, helping preserve Alice's privacy.

Fidelity allows the bank to verify the credential as a container, but fidelity does not prove the veracity of the statements within the credential. Generally, credential veracity depends on the reputation of the issuer. More specifically, we establish it through *credential provenance*.

In this example. the bank wants to know that the issuer identifier in the credential is associated with a legitimate business[4], the details of that business, and what others have said about that business so they can judge the veracity of the statements made in the credential. The bank has several options depending on their internal policies.

- They could use an out-of-band method to validate the identifier of the issuer by, say, looking up the public DID of the issuer on the issuer's web site.
- They could ask that the bank prove things to them by establishing a direct DID-based relationship with the bank and requesting data from the credentials the bank holds (e.g., their FIDC membership).
- The banking industry could create an industry-specific governance framework and list the public DIDs of its members in a public registry that anyone could access.

Determining the provenance of the credential's content cannot be done through purely technical means. Clearly, technology can help, but unlike credential fidelity where cryptography alone can prove credential authenticity, provenance is a matter of human process, policy, regulation, and law.

[4]I am saying business, but in fact this could apply to any entity that can issue credentials including people and things.

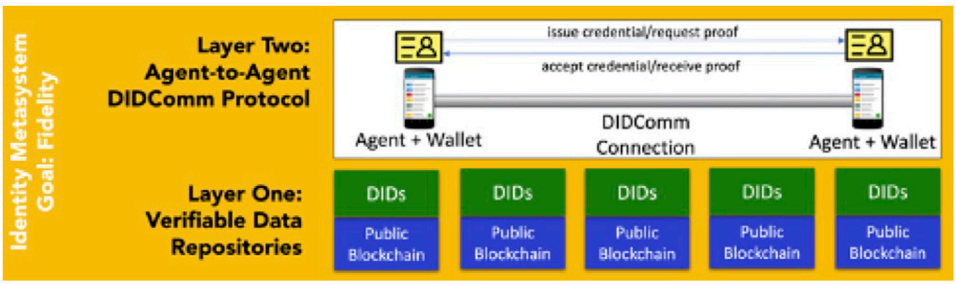

FIGURE 6 | The identity metasystem.

THE IDENTITY METASYSTEM

The identity metasystem embodied in the Sovrin Network provides three primary capabilities that allow it to be used as the basis for any context-specific identity system that is needed:

- Relationships—the architecture must allow people, organizations, and things to have relationships with each other.
- Messaging—the architecture must support messaging between the parties to those relationships.
- Trustworthy Attribute Exchange via Verified Credentials—parties to relationships must be able to reliably exchange information about attributes (often called claims by identity professionals).

The architecture for the identity metasystem supplies these features using layers that build on each other as depicted in **Figure 6**.

The metasystem is a hybrid architecture, using both algorithmic and autonomic identifiers to provide these capabilities.

Autonomic Identifiers Support Relationships and Messaging at Layer 2

When Alice forms a relationship with her friend, her colleague, her doctor, her employer, an ecommerce web site, or even her thermostat, she uses an autonomic identifier based on the Peer DID specification (Peer DID, 2020). Alice and other parties use agents based on the Hyperledger Aries open-source code (Hyperledger Aries, 2020). The user interface to these agents is called a wallet.

To form a relationship, Alice and the other party each generate a new peer DID and send it to the other. Peer DIDs need not be publicly resolved since both parties know about the other. The result is a network of peer-to-peer relationships between agents under the control of the people and organizations forming relationships. This forms Layer 2 of the identity metasystem in **Figure 6**.

Because the parties have exchanged DIDs, each party can authenticate the other. Mutual authentication allows the relationship to have integrity without an intervening third part (Young, 2020).

The relationship created by exchanging Peer DIDs is useful for more than mutual authentication. The mutually authenticated channel supports a uniform and democratic protocol for secure interaction called DIDComm (DID Communications, 2020). The DIDComm protocol allows parties to a relationship to securely and privately share authenticated messages. The security and authority of a DIDComm channel are rooted in DIDs and their associated authentication factors. DIDComm can be used over a wide variety of transports.

One of the primary uses of the DIDComm channel is to support the verification of key events following a key rotation. Whenever one of the parties needs to rotate their keys, they make an entry in their key event log (called "deltas" in the Peer DID specification) that records the relevant operations on the keys in a cryptographic manner. The key event log is a chain of signed change records that can be cryptographically verified. The parties in a DID-based relationship share (using a CRDT) key event logs for each identifier. If either party updates the keys associated with the DID, the other is informed of change.

But beyond that core functionality, DIDComm support message exchanges for many purposes. Aries RFCs (Hyperledger Aries RFCs, 2021) describe protocols for several core use cases for DIDComm including:

- Establishing peer DID Connections
- Requesting and issuing credentials
- Presenting a credential proof

Future use cases could include protocols for the following:

- Payments
- Interactions with IoT devices
- Buying and selling

Vic Cooper likened DID-based P2P messaging to the Batphone (Windley, 2020b). When Batman picks up the Batphone to talk with Commissioner Gordon, Commissioner Gordon does not start off the conversation with "Who am I speaking to?," "Can you give me your account number?," "What's your date of birth?," or "What street did you live on in Junior High?" When Commissioner Gordon picks up the Batphone, he knows it is Batman on the other end. Only Batman can call on the Batphone.

DIDComm-based messaging is like having a Batphone for every digital relationship you have. You and they know they are communicating with the right party. All the messages are authenticated and protected from eavesdroppers.

DID Messaging could revolutionize how we talk to each other and how we communicate with businesses.

- We no longer have to rely on a correlatable identifier like an email or phone number, to identify, discover, or connect to the other party.
- We no longer have to use centralized systems to talk to other parties with the attendant risk of the system being down or the conversation not being private.
- We save time and money using frictionless, direct communications with companies we need to work with.
- We can verify who is at the other end by asking them to prove things to us.
- We can sever one relationship without affecting others since everyone has a different identifier for us.

Agents exchange attributes over the channel created in Layer 2 using a flexible, decentralized system of credential exchange as discussed in *Credential Exchange as the Foundation for Online Identity*.

Algorithmic Identifiers Support Credential Exchange

The metasystem's algorithmic identifiers also take the form of DIDs. But rather than the peer DIDs used at Layer 2, DIDs at Layer 1 are public DIDs. DIDs have a number of important properties that make them ideal as identifiers in an algorithmic system. Specifically, they are non-reassignable, resolvable, cryptographically verifiable, and decentralized.

As algorithmic identifiers, DIDs allow the controller to make cryptographically authoritative statements about the identifier and the keys it is bound to. Those statements are recorded on a ledger to provide a record of the key events that anyone with access to the ledger can evaluate.

The DID specification provides for many DID methods such that DIDs may be recoded on a variety of data stores. There is nothing in the DID specification itself that requires that the data store be a blockchain or ledger, but that is the primary use case. The collection of ledgers supporting the binding of public DIDs to their authentication factors forms Layer 1 of the metasystem shown in **Figure 6**.

The record on the ledger is public since the purpose of putting DIDs on a ledger is to allow parties who do not have an existing relationship to evaluate the identifier and its linkage to the controller and public keys. The ledger provides several important features:

- The ledger creates a circuit breaker so that issuers do not know when and where credentials are being used, increasing the privacy of the transaction. Consequently, the metasystem structurally supports the privacy of participants.
- The ledger enables offline exchange of credentials. This not only supports verification of a credential when the issuer is offline, but support for state proofs in the ledger allows exchange to occur when all the parties are offline but the holder and verifier can connect over some local network (e.g., Bluetooth).

The metasystem makes use of the resolvability of DIDs to support credential exchange. Issuers (in the role of controller) register DIDs on a public ledger and issue credentials using that identifier. When the credential holder proves attributes to a verifier, she also proves the identifier of the issuer. The verifier can resolve the DID for the issuer from the ledger as part of ensuring the fidelity of the credential exchange.

BUILDING IDENTITY SYSTEMS ON THE METASYSTEM

The capabilities of the identity metasystem provide a sure foundation for creating identity systems that are secure and support the autonomy and privacy of people and organizations. The goal of an identity metasystem, like Sovrin Network, is to connect individual identity systems and allow them to interoperate since no single system meets the needs of every digital identity scenario.

As we discussed in *Relationships*, the goal of the metasystem is to support relationships between parties online and provide a secure, private means of exchanging verified credentials. The metasystem uses credential exchange on top of DIDComm messaging at Layer 2 as the unifying protocol for exchanging identity information. In credential exchange, an issuer issues a credential to a person or organization called the holder. The holder holds one or more credentials and uses the protocols provided by the metasystem to prove things about themself to a verifier who needs trustworthy attributes. **Figure 7** shows the layers of this system.

The blue box on the top of **Figure 7** represents an identity system built on top of the metasystem. There is more than one identity system. In fact, there are tens of millions, maybe more. Every credential definition represents a new identity system created for a specific context. Anyone can define a credential for any purpose. And even though each identity system stands alone for its own purpose, they are interoperable because they are built on top of the metasystem and employ common protocols.

For example, Alice may have a credential representing her driver's license and one representing her employee ID. These are designed for a specific purpose by the DMV and the employer. Yet, because they are based on a metasystem and use a common protocol, she could go to the bank and use those in concert to prove that she is employed (employee ID) and her date of birth (driver's license) in one operation.

The two systems shown in **Figure 7** have different properties. The identity metasystem (orange box) provides important assurances about the fidelity of the credential. A credential verifier who receives a proof is concerned about credential fidelity, but they are also concerned with the credential's provenance. The fidelity provided by the identity metasystem, combined with the credential provenance provided by the

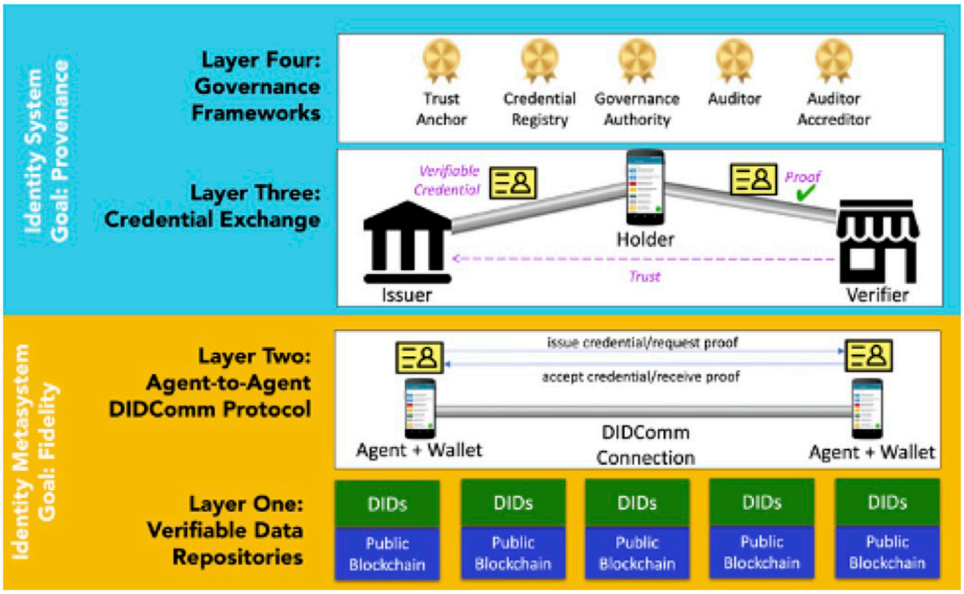

FIGURE 7 | Identity on the metasystem.

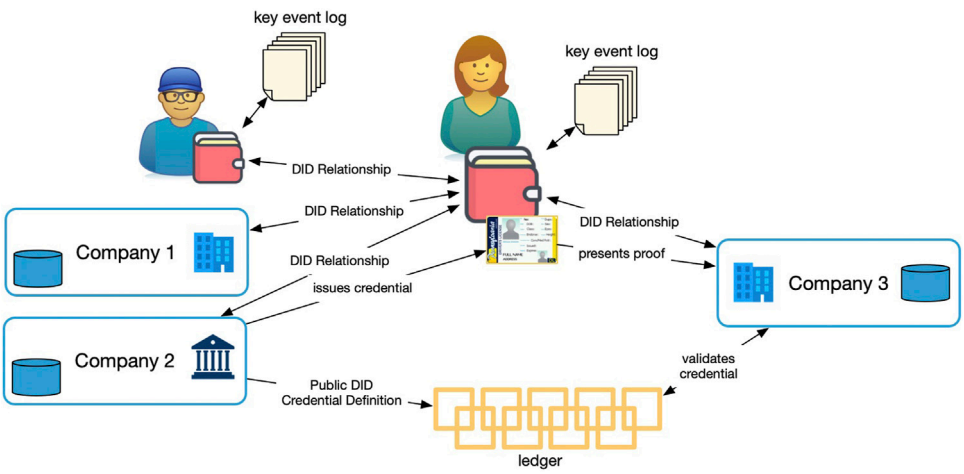

FIGURE 8 | Relationships and interactions in the Sovrin Network.

context-specific identity system operating on top of it, provides the basis for trusting the information that the holder has conveyed through credential exchange.

Operationalizing Digital Relationships

Figure 8 shows the relationships and interactions in the Sovrin Network. In the figure, Alice has an SSI wallet[5]. Alice's SSI wallet is like other wallets she has on her phone with several important differences. First, it is enabled by open protocols, and second, it is entirely under her control. She uses the wallet to manage her relationship with Bob as well as a host of organizations.

This diagram has elements of each architectural style described in *The Architecture of Identity Systems*. Alice has relationships with four different entities: her friend Bob and three different companies. These relationships are based on autonomic identifiers in the form of peer DIDs.

Company 2 has an algorithmic identifier in the form of a public DID that has been recorded on the ledger along with a credential definition. Company 2 has, based on that credential definition and its associated public DID, issued a credential to

[5]I am using the term "wallet" fairly loosely here to denote not only the wallet but also the agent necessary for the interactions in an SSI ecosystem. For purposes of this article, delineating them is not important. In particular, Alice may not be aware of the agent, but she will know about her wallet and see it as the tool she uses.

Alice. The contents of that credential are based on the information Company 2 knows from its relationship with Alice, stored in its *internal* administrative identity system.

Alice has presented a proof based on the credential to Company 3 who can validate its fidelity using the credential definition on the ledger. Company 3 likely has its own internal administrative identity system where it stores information about its relationship with Alice.

The peer DIDs that Alice presented to Company 2 and Company 3 are different. Nevertheless, the cryptographic procedures of the zero-knowledge proof (ZKP) that Alice presents to Company 3 ensure that Company 3 can know that the credential used as the basis of the proof was issued to the same person who they have a relationship with. More generally, Company 3 knows that the same entity controls the keys for the DID Alice shared with Company 2 and the DID she shared with them.

Company 2 does not issue the verifiable credential to the peer DID Alice gave them. In Hyperleger Aries credential proof, Alice creates a blinded link secret and sends it to Company 2 in response to a credential offer. The verifiable credential contains the blinded link secret. When Alice uses ZKP to prove attributes from her credentials, the blinded secret is what proves Alice is the same Alice to whom all the credentials she used were issued. The proof contains a special predicate showing that the link secret in the credential, if unblinded, would be the same as the link secret Alice shared with Company 3, if unblinded. No unblinding actually happens. Since the credential is not linked directly to the peer DID, but indirectly through the blinded link secret, Alice is free to rotate the DID-associated keys underneath the credential without invalidating it. And the DID continues to serve its purpose of identifying Alice to Company 2 (Hardman 2018).

Because Alice uses different peer DIDs for Company 2 and Company 3, they cannot correlate data they have about her through the identifier independently. They need Alice, who controls the link secret, to correlate the information for them. That ensures Alice is in control of what information is shared and correlated based on the peer DID relationships.

Identity Systems

When we say "digital identity system", most people probably think of just one thing: authentication. The digital identity systems we have built over the last 30 years are so anemic that it is difficult for us to imagine the kind of rich identity systems that exist in the physical world being available online.

In the offline world, we use credentials to prove things about ourselves to others. Each of these credentials constitutes an identity system, designed and built for a specific purpose in a given context. For example, businesses frequently give employees ID cards. I have one for Brigham Young University (BYU), my employer. I can use it to open doors, get a discount at the bookstore, get a car from the motor pool, and even ride a local bus or train. This flexible identity system allows the university to add new functionality over time as needs change. The university sets the rules about who gets an ID card and what it means. Of course, it also has use outside the context of the university, say, for example, at a store that gives discounts to university employees and is willing to accept the ID card as proof of employment.

Businesses are full of credentials. Each one represents an identity system designed and built for a specific context. Every form or official piece of paper is a potential credential. Every bundle of data transmitted in a workflow is a potential credential. Here are a few examples of common credentials:

- Employee badges
- Driver's license
- Passport
- Wire authorizations
- Credit cards
- Business registration
- Business licenses
- College transcripts
- Professional licensing (government and private)

Here are some others that may not be typically thought of as credentials, but fit the definition:

- Invoices and receipts
- purchase orders
- Airline or train ticket
- Boarding pass
- Certificate of authenticity (e.g., for art, other valuables)
- Gym (or any) membership card
- Movie (or any) tickets
- Insurance cards
- Insurance claims
- Titles (e.g., property, vehicle, etc.)
- Certificate of provenance (e.g., non-GMO, ethically sourced, etc.)
- Prescriptions
- Fractional ownership certificates for high value assets
- CO2 rights and carbon credit transfers
- Contracts

Since even a small business might issue receipts or invoices, have customers who use the company web site, or use employee credentials, most businesses will define at least one credential, and many will need many more. There are potentially tens of millions of different credential types. Many will use common schemas but each credential from a different issuer constitutes a different identity credential for a different context.

With the ongoing credential work in Hyperledger Aries (Hyperledger Aries 2020), these use cases expand even further. With upcoming "redeemable credentials" feature, issuers can double-spend-proof proving credential possession without a ledger. This works for all kinds of redemption use cases like clocking back in at the end of a shift, voting in an election, posting an online review, or redeeming a coupon.

You might notice that many of the things listed above are solutions some people advocate building entire blockchains for. That is overkill when you can use a credential to get the job done. Especially when that credential is interoperable with others in a ubiquitous identity metasystem. By double-spend-proofing

credentials, you create a system capable of representing value of all sorts. An identity metasystem for trustworthy credential exchange has uses far beyond what we might typically think of as an "identity system."

A Marketplace for Credentials

Many credentials will be created for internal or non-commercial purposes (like the employee credential). But some will have a supporting business model. This is exactly what happens offline where many credentials are exchanged for money. The metasystem should support credential business models to achieve ubiquity. Daniel Hardman discusses this in his excellent blog post about Categorizing Verifiable Credentials (Hardman, 2019b).

Credentials may intersect with payment in different ways. Some may be issued and used for free; others may be purchased; still others may incur a fee with every use. And while payment could be viewed as entirely independent from credentials, the binding is actually more interesting. This is because economics and levels of assurance are intertwined. For example, a top-secret security clearance may require thousands of dollars of field work and investigation and bump its holder's salary by even more. Thus, business models that allow economic value to be harvested in credential interactions are important.

With non-free credentials, who pays whom is interesting. The most straightforward model is holder-pays-issuer; we already expect to pay a fee when we apply for a passport. But other variations are equally possible, and they represent potential innovation that is impractical with physical credentials. For example, a holder who is applying to a university might pay the university a fee to verify their academic credentials. A potential employer with stringent security requirements might pay an issuer to achieve assurance that an applicant has a government security clearance. A medical researcher might pay a holder for the privilege of verifying genetic information from credentials, as part of a study they are conducting.

While it is impossible to anticipate every possible credential use case that includes a reciprocal exchange of value, looking at a few use cases is instructive. The following use cases are just for the Holder-Pays-Issuer pattern, but other patterns, like Verifier-Pays-Issuer, are possible.

Driver's License—Driver's licenses are an excellent example of a credential people pay for. There are 112 million licensed drivers just in the US. If we assume each license costs $30 and is renewed every 5 years, almost $700 million is paid per annum for driver's licenses.

Memberships—Memberships in gyms are just one example of a membership credential where the credential holder pays the issuer. Gym membership revenues in the US in 2018 was $32 billion according to Wellness Creatives[6]. There are many more membership types that could be built on top of Sovrin Network.

Movie Tickets—Movie tickets are another credential that is bought. In 2018, 1.3 billion movie tickets were sold in the US[7]. At $10 per ticket, that is $13 billion.

Airline Tickets—Airline tickets are a special kind of credential that is purchased. According to IATA, there were 4.1 billion airline passengers in2017[8]. The US Department of Transportation Bureau of Transportation Statistics reports that average airfare was $347 that same year[9]. We can estimate that worldwide airfare was about $1.4 trillion in 2017.

Online Sales—Online sales could be accomplished using Holder-Pays-Issuer credential exchange. By paying for the receipt (a credential) equal to the amount of the order, we can view all of ecommerce as a form of paid credential issuance. Linking payment to a credential and placing it inside a wallet that emphasizes relationships and credential management may make credential-related payments an important component of online retail. US online retail sales were $519 billion in 2018[10].

These are just a few potential use cases where credentials and value are exchanged. While not all of these will necessarily come to pass, it is easy to conclude that the potential marketplace for credentials is in the trillions of dollars. The identity metasystem, with its mutually authenticated messaging protocol, is an excellent platform for supporting commercial credential exchange. These workflows, with built-in value exchange, can be developed on the identity metasystem.

An identity metasystem like the Sovrin Network provides the foundation for creating tens of millions of interoperable identity systems for every conceivable context and use. By virtue of being built on the metasystem, these identity systems share a common protocol and similar user experience. The metasystem is available to all and is decentralized, allowing each participant to make their own decisions about what identity systems they will build and participate in to support their goals and ambitions.

CONCLUSION: LIFE-LIKE DIGITAL IDENTITY

We use identity in the physical world without thinking about it. And when we do, there are patterns that are so ingrained in our ways of interacting that we do not give them a second thought. If we are to move more and more of our lives to the digital realm while also preserving agency and autonomy, we must create a digital world that allows us to jump the trust gap we inevitably have with people, organizations, and things when our interaction is digital.

An identity metasystem provides the long-missing identity layer for the Internet that will allow this to happen. The metasystem can be incorporated into every digital tool and system providing a consistent, trustworthy experience that feels as frictionless and natural as identity in the physical world.

The identity metasystem overcomes the problems of digital identity described in *The Problems of Online Identity*. We have described a system that carefully uses cryptography to overcome the problems introduced by distance while providing autonomy

[6]https://www.wellnesscreatives.com/gym-market-statistics/.

[7]https://www.statista.com/statistics/187073/tickets-sold-at-the-north-american-box-office-since-1980/.

[8]https://www.iata.org/en/pressroom/pr/2018-09-06-01/.

[9]https://www.bts.gov/content/annual-us-domestic-average-itinerary-fare-current-and-constant-dollars.

[10]https://www.digitalcommerce360.com/article/us-ecommerce-sales/.

and flexibility for people and organizations without compromising strong privacy and workable anonymity. The nature of credential exchange based on an interoperable protocol specification introduces a system for building myriad identity systems that provide a more life-like experience than current, disconnected administrative identity systems.

Decentralized, self-sovereign identity depends on an identity metasystem and is the foundation for a decentralized web—a web that flexibly supports the kind of ad hoc interactions people have with each other all the time in real life. We will never get an online world that mirrors real life and feels frictionless and life-like until we do.

Consequently, the arguments for creating the identity metasystem provided by Sovrin Network are not narrow or technical issues. Sovrin Network does not merely provide narrow technical benefits. Rather, the identity metasystem is vital for personal autonomy and ultimately human rights. Computers are coming to intermediate every aspect of our lives. Our autonomy and freedom as humans depend on how we architect this digital world. Unless we put digital systems under the control of the individuals they serve without intervening administrative authorities, the internet will undermine the quality of life it is meant to bolster. The identity metasystem is the foundation for doing that.

AUTHOR'S NOTE

Parts of this article have appeared previously on the author's blog at https://www.windley.com.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

PW is solely responsible for the content of this publication.

REFERENCES

Andrieu, Joe. (2018). Five Mental Models of Identity. Available at: https://github.com/WebOfTrustInfo/rwot7-toronto/blob/master/topics-and-advance-readings/five-mental-models-of-identity.md (Accessed Nov 5, 2020).

Cameron, Kim. (2005). The Laws of Identity. Available at: https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf (Accessed Nov 5, 2020).

Conger, Kate., and Popper, Nathaniel. (2020). *Florida Teenager Is Charged as 'Mastermind' of Twitter Hack*. New York Times. Available at: https://www.nytimes.com/2020/07/31/technology/twitter-hack-arrest.html (Accessed Nov 5, 2020).

Decentralized Identifiers (DIDs) specification v1.0 (2020). Decentralized Identifiers (DIDs) Specification v1.0. W3C Working Draft 08 October 2020, Available at: https://www.w3.org/TR/did-core/ (Accessed Nov 5, 2020).

DID Communication Working Group (2020). Available at: https://identity.foundation/working-groups/did-comm.html (Accessed Nov 5, 2020).

Hardman, Daniel. (2019b). Categorizing Verifiable Credentials. Available at: https://www.evernym.com/blog/categorizing-verifiable-credentials/ (Accessed Nov 5, 2020).

Hardman, Daniel. (2018). How DIDs, Keys, Credentials, and Agents Work in Sovrin, Sovrin Technical Report. Available at: https://sovrin.org/library/how-dids-keys-credentials-and-agents-work-in-sovrin/ (Accessed Nov 5, 2020).

Hardman, Daniel. (2019a). The Dangerous Half-Truth of "We'll Be Correlated Anyway". Available at: https://www.evernym.com/blog/well-be-correlated-anyway/ (Accessed Nov 5, 2020).

Hyperledger Aries Project (2020). Available at: https://www.hyperledger.org/use/aries (Accessed Nov 5, 2020).

Hyperledger Aries RFCs (2021). Available at: https://github.com/hyperledger/aries-rfcs (Accessed June, 2021).

Lodder, Mike. (2018). The Sovrin Network and Zero Knowledge Proofs. Available at: https://sovrin.org/the-sovrin-network-and-zero-knowledge-proofs/(Accessed Nov 5, 2020).

Naik, N., and Jenkins, P. (2016). "An Analysis of Open Standard Identity Protocols in Cloud Computing Security Paradigm," in 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, Auckland, New Zealand, 8-12 Aug. 2016 (IEEE), 428–431. doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2016.85

Nickel, Philip. J., and Vaesen, K. (2012). "Risk and Trust," in *Handbook of Risk Theory*. Editors Sabine Roeser, Rafaela Hillerbrand, Martin Peterson, and Per Sandin (Springer).

Peer DID Method Specification (2020). Peer DID Method Specification. W3C Document 25 August 2020, Available at: https://identity.foundation/peer-did-method-spec/index.html (Accessed Nov 5, 2020).

Simmonds, P. (2015). The Digital Identity Issue. *Netw. Security* 2015, 8–13. doi:10.1016/s1353-4858(15)30069-6

Smith, Samuel. (2020). Key Event Receipt Infrastructure (KERI). Available at: https://arxiv.org/abs/1907.02143 (Accessed Nov 5, 2020).

Sporny, M., Noble, G., Longley, D., Burnett, D., and Zundel, B. (2019). Verifiable Credentials Data Model 1.0. Nov 19 2019. Available at: https://www.w3.org/TR/2019/REC-vc-data-model-20191119/.

Wilson, Stephen. (2011). "Identities Evolve: Why Federated Identity Is Easier Said Than Done," in AusCERT 2011 Conference: "Overexposed" Gold Coast, Australia. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2163241 (Accessed Nov 5, 2020).

Windley, Phillip. J. (2020). *Authentic Digital Relationships*. Lindon, UT, United States: Technometria. Available at: https://www.windley.com/archives/2020/08/authentic_digital_relationships.shtml (Accessed Nov 20, 2020).

Windley, Phillip. J. (2020b). *DIDComm And the Self-Sovereign Internet*. Lindon, UT, United States: Technometria. Available at: https://www.windley.com/archives/2020/11/didcomm_and_the_self-sovereign_internet.shtml (Accessed Nov 20, 2020).

Windley, Phillip. J. (2018). *You've Had An Automobile Accident: Multi-Source Identity To the Rescue*. Lindon, UT, United States: Technometria. Available at: https://www.windley.com/archives/2018/08/youve_had_an_automobile_accident_multi-source_identity_to_the_rescue.shtml (Accessed Nov 5, 2020).

Young, Kaliya. (2020). Understanding DIDComm. Available at: https://medium.com/decentralized-identity/understanding-didcomm-14da547ca36b (Accessed Nov 5, 2020).

Zuboff, Shoshana. (2020), The Age Of Surveillance Capitalism: The Fight For a Human Future At the New Frontier Of Power, (PublicAffairs) January 2019.

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.