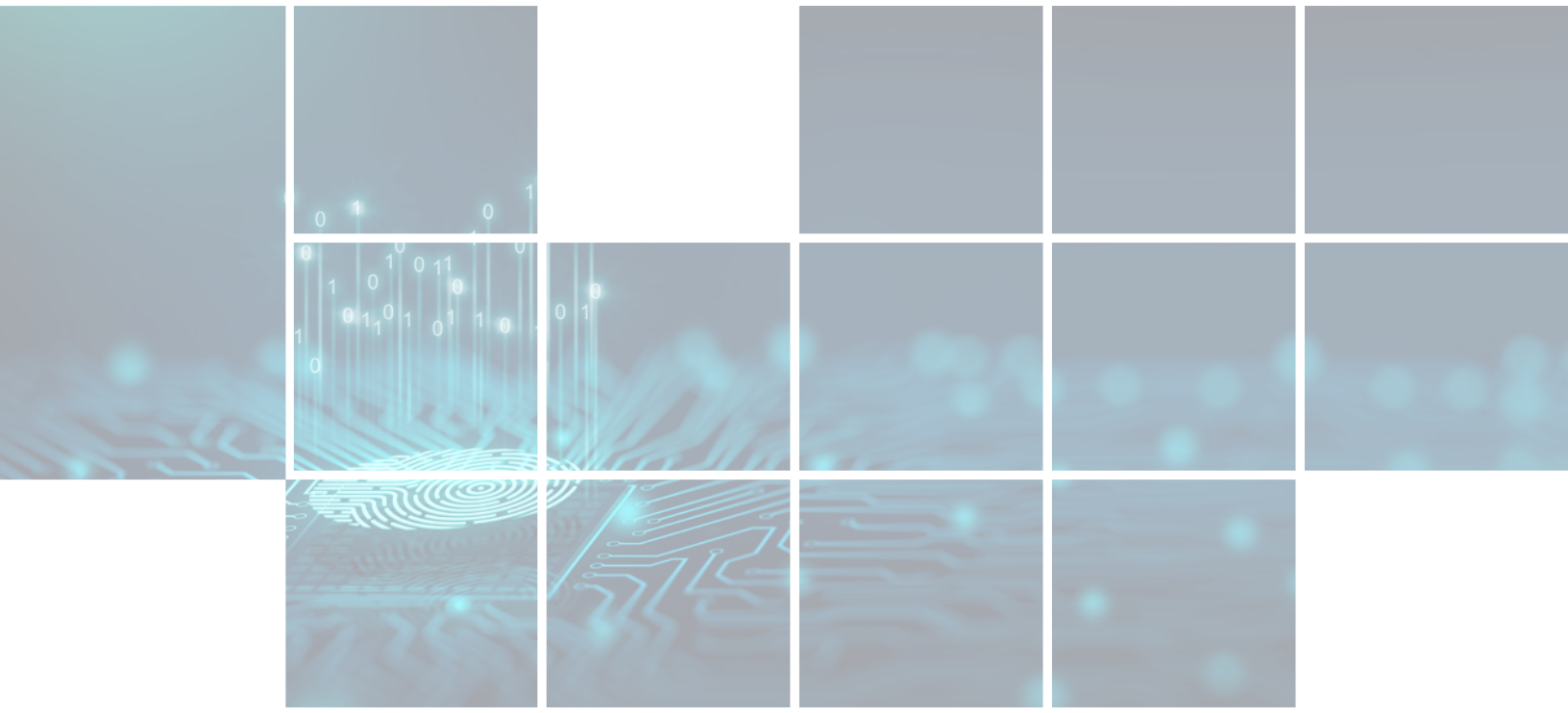
ANONYOME LABS



Innovating

Identity and Access Management with Decentralized Identity

Users will benefit from the transition to decentralized identity.

Identity and access management, or IAM, is the security function that makes it possible for the right users to access the right resources (applications or data).

IAM solutions are widely deployed and very mature, but for users the current approaches fall short. These solutions are difficult to manage, privacy invasive, and often lead to unwanted outcomes such as identity and financial theft.

Decentralized identity (DI) is a new approach to IAM. It overcomes many of the current IAM shortcomings and focuses on the user's needs. DI simplifies how cryptography is applied so that it better protects a user's resources while making it even easier for them to use.

Legacy IAM:
a Centralized Identity Architecture

The first and perhaps still the most common type of IAM is based on a centralized model. In this model, a user signs up directly with a service (e.g., creating an account). During signup (and subsequent logins), the user gives the service personally identifiable information (PII) such as username, password, email address, mobile phone number, credit card information and other types of personal data. Typically these days, the user will also set up two factor authentication (2FA) to strengthen user authentication, which can require even more PII.

Using IAM terminology, the service is both the identity provider (creates the user's account and stores user attributes) and the service provider (provides the application and data). The access control policy is implemented at the service and it's common to find group-based, role-based, and attribute-based models (or combinations) in use.

Often, an enterprise's employees and customers all use this centralized model.

Centralized IAM has some key drawbacks for the user:

Remembering: Over a few years, users could create hundreds of service accounts. For each account, they must remember all their different username (could be email, phone #, or custom) and password combinations. Ideally, each account has a different password (for security) and username (for privacy). If the user has enabled 2FA, they must also give additional information at login (e.g., a one-time password or SMS code). The growth in the password manager products market is testimony to just how difficult users can find managing accounts and authentications.

New PII tracking: Users are providing their PII data to many different services—perhaps 200+. This puts the user's PII data at great risk since any of those services could sell the data they collect or have their user information stolen. Consequently, loss of privacy, identity theft, and financial theft are on the rise.

Figure 1 shows the centralized model. With this model, the service is both an identity provider and a service provider. It also maintains the access control policy. Users create accounts on the service, provide it with PII, and access the service.

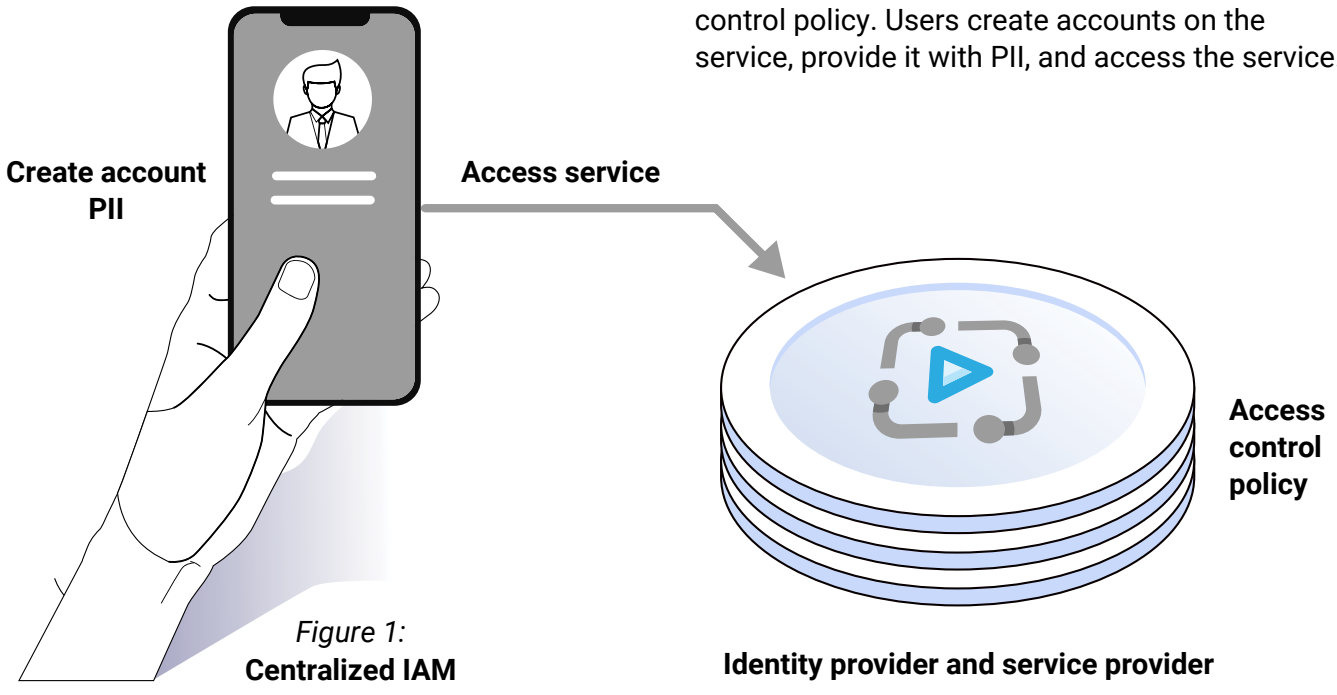


Figure 1:
Centralized IAM

Federated Model for IAM

The second most common type of IAM is where a user signs up with a major identity provider (e.g., Google, Facebook, Twitter, LinkedIn, Apple, etc.). When users want to access a new service, instead of creating a new account, they use federated login (or social login). Usually this appears as a "Sign in with …" login on the service.

Under this model, the identity provider and the service provider are distinctly separate. Using protocols such as SAML, OAuth or OpenID Connect, the user is federated from the identity provider to the service provider. Both providers maintain a trust relationship between them. The user no longer needs to be authenticated at the service provider. At first glance, this improves on the centralized IAM model, because the number of services where a user must create an account is markedly reduced and the user uses fewer username/password/2FA combinations. While this model is much more manageable for the user, it **significantly impacts the user's privacy** since the identity provider knows all the federated services the user accesses, when they connect, and which of the user's PII the service provider is using. [1]

The service provider creates the access control policy and requests the user's PII from the identity provider as required. There can be only one identity provider for the user at this service provider—whereas decentralized identity users can present credentials from multiple identity providers (more later). The downside is that the user's personal data is still being distributed among numerous services, which impacts the user's privacy.

The user doesn't own their identity provider's account. If the identity provider (e.g., Facebook) suspends or deletes the user's account, the user is cut off from the identity provider as well as from any services that were federated through the service provider!

Figure 2:
Federated model for IAM

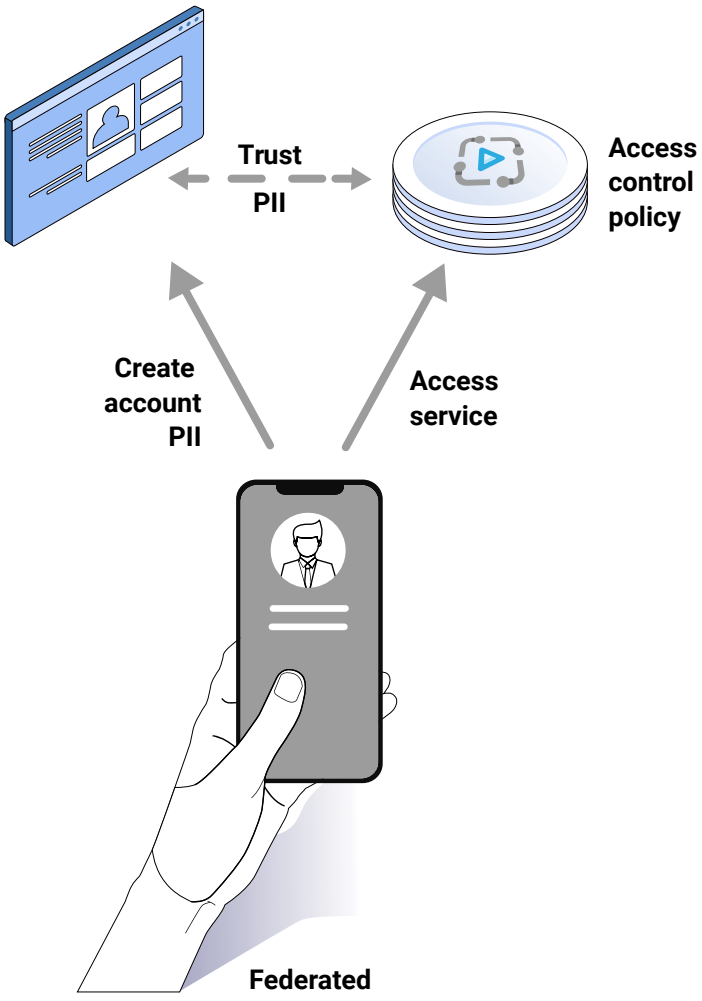


Figure 2 gives an overview of the federated model. The identity provider and the service provider are distinctly separate, but there is still a trust relationship between them. The service provider also creates the access control policy, which determine how a user's PII is used and disseminated. The service provider requests PII from the identity provider.

[1] **It is possible to implement federated IAM in a way that the identity provider doesn't track the user's access to services, but historically federations haven't been created in this way.**

Decentralized Identity Model for IAM

DI is specifically designed to improve security and the user experience. DI is described as:

… an approach to identity and access management (IAM) that seeks ways to allow individuals to manage their own personally identifiable information (PII) instead of using a central authority. An important goal of decentralized identity is to create standards that will allow internet users to control which applications and services can have access to specific types of PII.

The DI approach is quite different from the centralized and federated identity models. While identity providers still exist in this model, they are providers of cryptographic verifiable credentials (VCs). VCs are signed electronic data structures that carry claims about the user, which the identity provider asserts. VCs have many security benefits for the receiver of the credential (service provider). Specifically, the receiver of the credential can:

- Be assured of the identity of the identity provider (creator) of the credential
- Be assured of the integrity of the credential (it can't be modified without detection)
- Be assured the identity provider hasn't revoked the credential
- Accept credentials from different identity providers (credential issuers), which can contribute to the access control policy. This creates resilience when some credentials may have expired (e.g., use a passport's claims when a driver license may have expired).

Accept credentials from different identity providers (credential issuers), which can contribute to the access control policy. This creates resilience when some credentials may have expired (e.g., use a passport's claims when a driver license may have expired).

An identity provider issues a VC to the user for storing in the user's identity wallet. The identity wallet is an element that DI systems introduce.

A decentralized ledger (e.g., Sovrin, Indicio, cheqd, etc.) provides the foundation of trust in the DI environment. This new component replaces the functionality that centralized identity systems (e.g., centralized PKI) currently provide.[2]

When the user wants to access a service (the service provider), they assert claims from the VCs (received from one or more identity providers). This assertion/presentation of claims process is done via a proof of possession and control of the credential. The user uses their identity wallet to present the requested claims to the service provider. The access control policy at the service provider is based on attribute-based access control (ABAC) since the decisions are made on the attributes/claims present in the credentials.

[2] Work is underway on the option of reducing the need for a decentralized ledger. The did:web method is anchored in the X509 SSL certificate as the security around a DIDDoc.

Two important aspects of DI relate to the user presenting VC proofs to the service provider:

1. The identity provider doesn't need to be aware of the legal identity of the user accessing the service provider and there is no interaction between service provider and identity provider.[3]

2. The user is in full control of approving or denying what PII (from their VCs) is presented to the service provider. Both aspects improve privacy and security over the centralized and federated models by asserting authentication information rather than providing it.

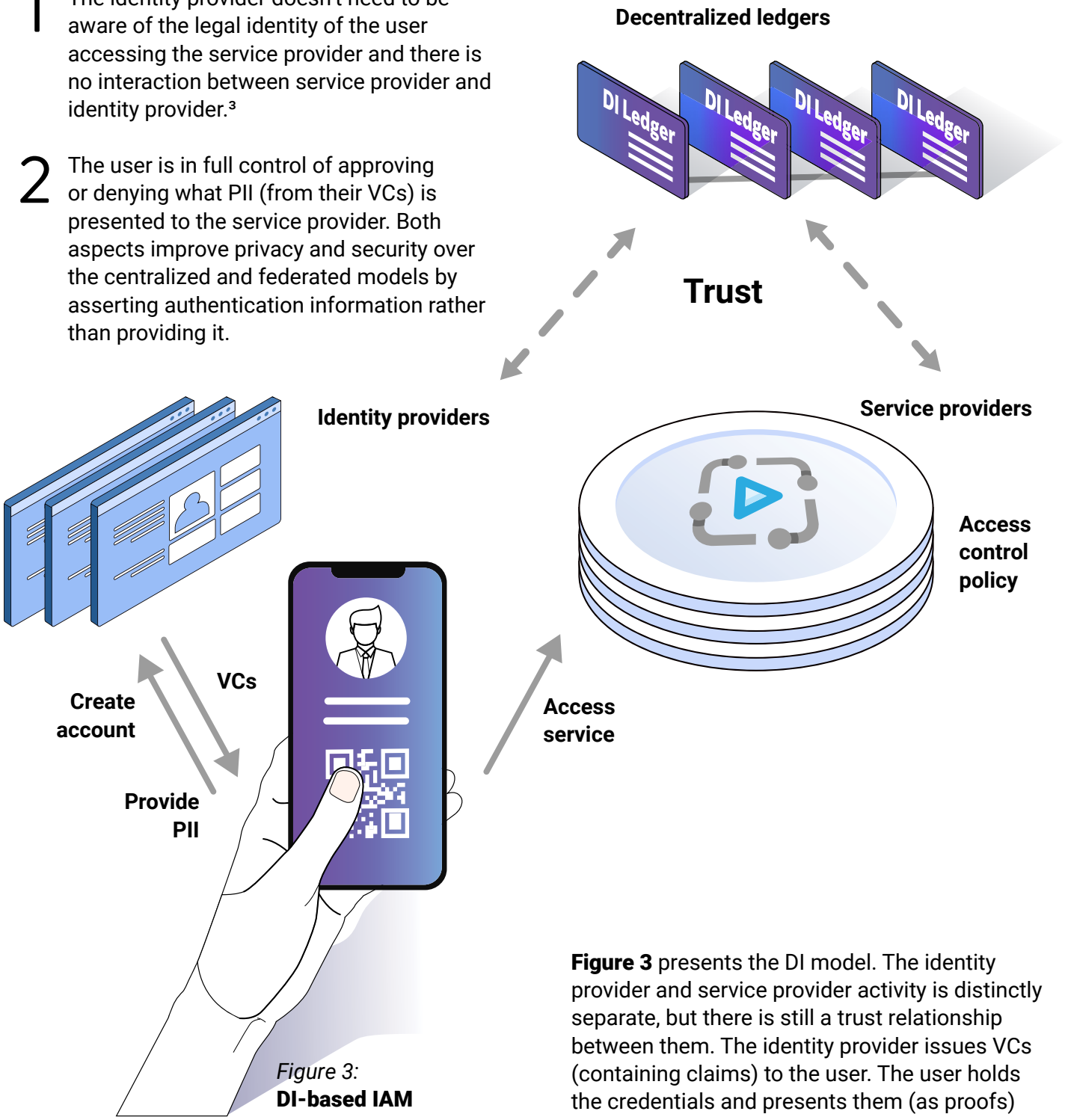


Figure 3: **DI-based IAM**

Figure 3 presents the DI model. The identity provider and service provider activity is distinctly separate, but there is still a trust relationship between them. The identity provider issues VCs (containing claims) to the user. The user holds the credentials and presents them (as proofs) when required at the service provider. The service provider creates the access control policy, which is attribute-based.

Decentralized

[3]In some cases, the identity provider and service provider may be the same enterprise, such as when a single company issues credentials for their own employees' access.

The Role of the Identity Wallet

The identity wallet is at the heart of the DI user experience. In the wallet, the user creates and manages their decentralized identities and any VCs they receive. For each new decentralized identity, the wallet creates a:

- Private/public key pair (usually based on elliptic curve cryptography such as ed25519)
- Decentralized identifier (DID).

Once the wallet has created a key pair/DID, the user can interact with a DI ecosystem. To communicate and receive a VC from an identity provider, the wallet establishes a DIDComm connection (a durable, secure connection to allow two parties to communicate) between the identity wallet and the identity provider. In the same way, the identity wallet establishes a DIDComm connection with the service provider to present (as proof) claims from the VCs. This process introduces strong cryptography, but the user is no longer required to remember and present a username/password to either the identity provider or the service provider. This simplification comes from allowing the user to be authenticated after demonstrating control of the decentralized identity's private key, rather than by presenting a secret credential (e.g., password).

Figure 4: **Sudo Platform identity wallet**

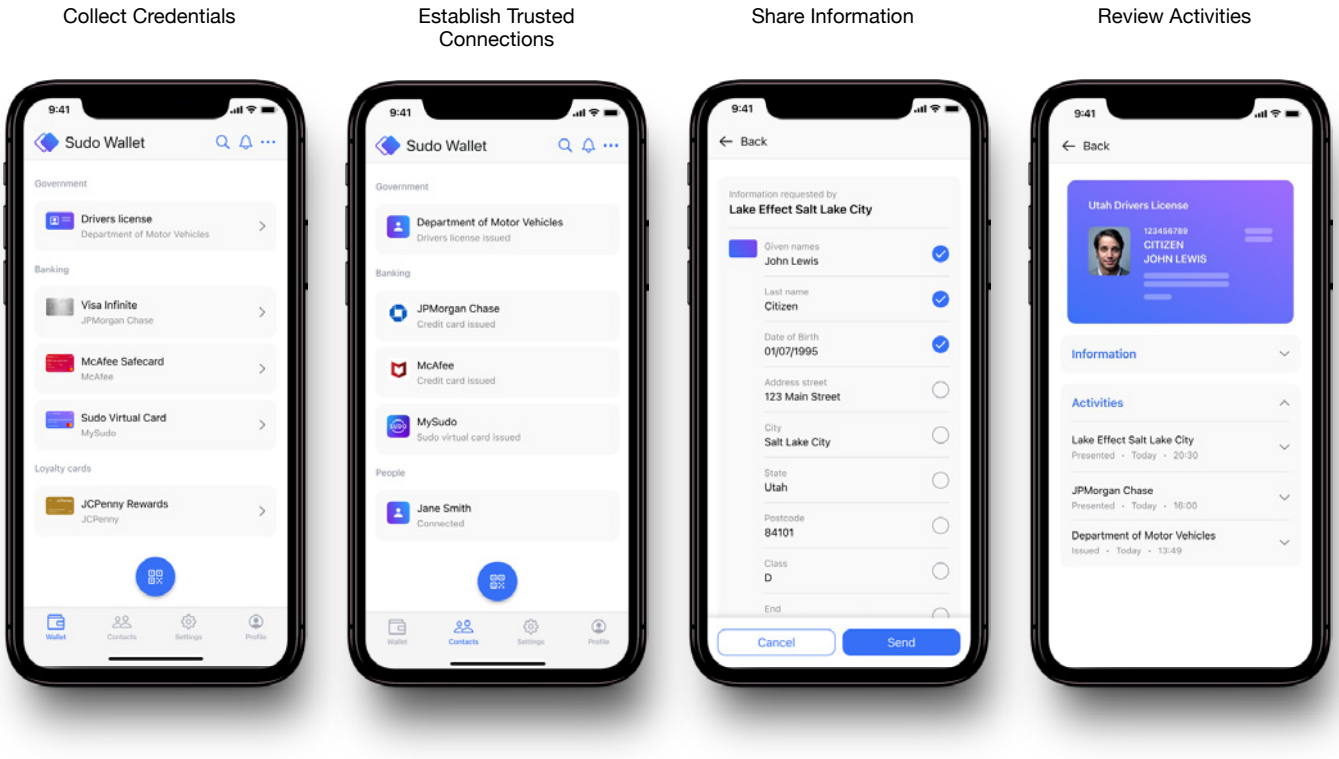
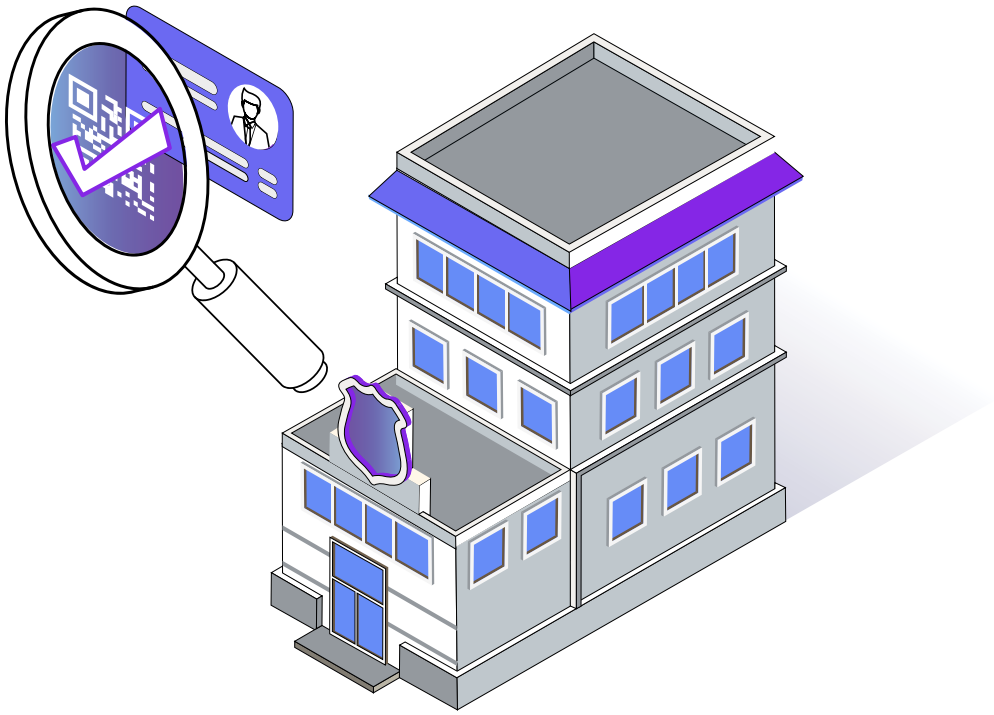


Figure 4 displays a few screens from the Sudo Platform's mobile identity wallet. On the left, the user can view any VCs currently stored in the identity wallet. The second screen is a list of (DIDComm) connections the wallet has established. The third screen shows the user selecting specific PII elements to share from a credential. The fourth screen shows activities related to that credential.

The Role of VCs

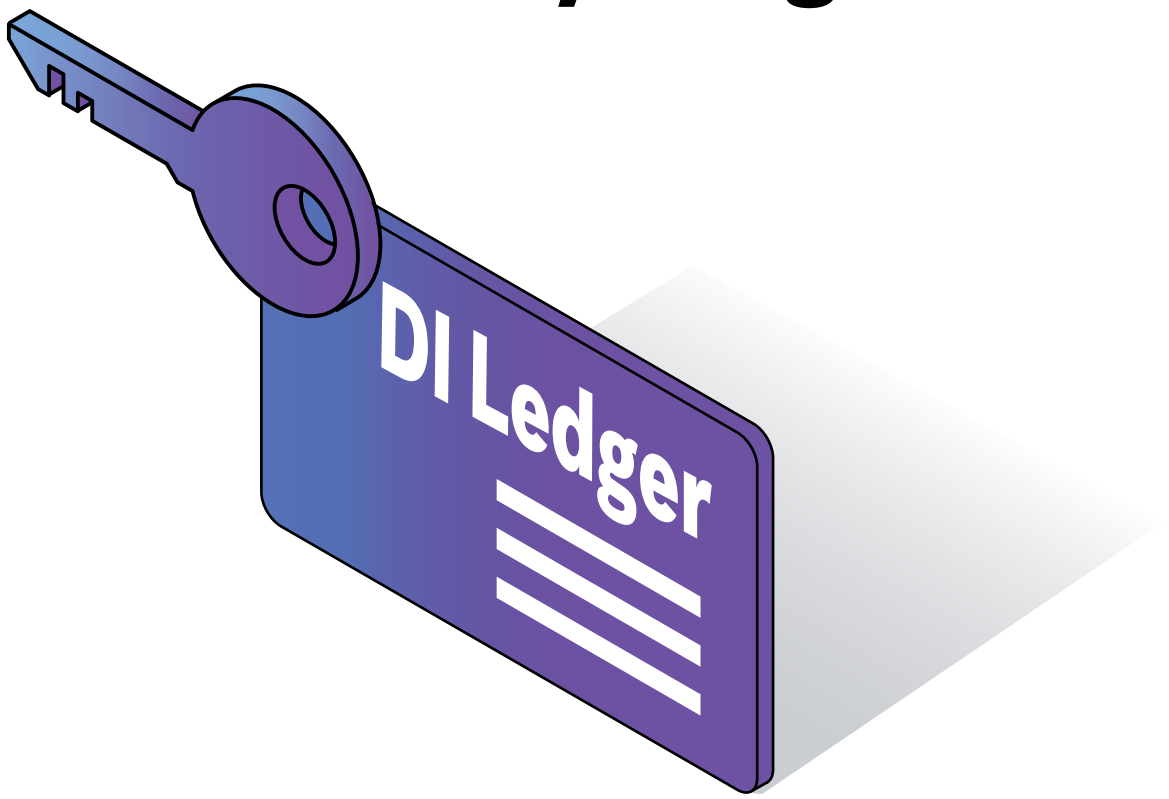


VCs are the core technology enabling DI-based IAM. An identity provider creates VCs and issues them to a user who can later assert them (or a subset of their claims) to a service provider. VCs are designed for protecting the user's privacy, and have privacy/security preserving innovations:

- Service providers never need to communicate the user's actions to the identity provider.
- Users can control what PII from a VC is available to the service provider. They might disclose all of a credential's claims, a subset of the claims, or even privacy-preserving zero knowledge proofs or ZKPs (more later).
- Service providers can be sure of the identity of the VC creator, that the credentials haven't been modified, and that the identity provider hasn't revoked the credentials.
- Users no longer need username and password combinations to connect with the identity provider and the service provider.

There are two VC standards: W3C Credentials and Anonymous Credentials (AnonCreds).

The Role of the Decentralized Identity Ledger



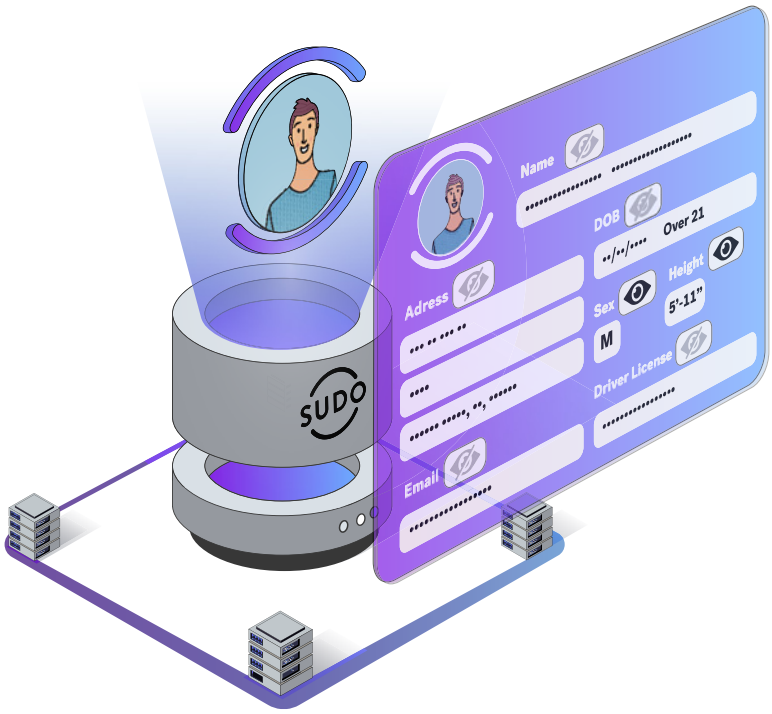
The DI ledger is another new component of the DI-based IAM model. It provides the foundation of trust to the environment and functions loosely like a decentralized PKI. Many different DI ledgers exist; the most common are Hyperledger Indy-based ledgers.

Before an identity provider can issue credentials to a user, they must first establish their own identity foundation on a DI ledger. This process (using Hyperledger Indy as an example) includes writing to the ledger the:

- Issuer's DID
- Schema of the credential
- Credential definition (relates the Issuer DID and credential schema)
- Revocation definition (in the case where the service provider needs to check on whether a credential has been revoked).

Since the DI ledger stores this information, the service provider can accept the security of a VC without having to check with the identity provider directly.

The Role of Attribute-Based Access Control (ABAC)



ABAC is the natural embodiment of the DI-based IAM system. This process enables the user to present claims (based on elements contained within VCs) to the service provider, and the service provider should be able to authenticate a user's identity based on those claims. For example:

If a user presents a claim from a valid US driver license stating they are over age 21, they can avoid presenting their actual birth date. This allows a receiving merchant/verifier to verify whether a customer is over the minimum age required for certain activities, such as purchasing alcohol.

If a user presents proofs from their US passport, COVID-19 vaccine certificate, and a country visa, a receiving airline can verify whether a traveller meets current travel requirements and thereby is allowed to purchase plane tickets to that country.

The Full DI-Based IAM Model—Putting It All Together

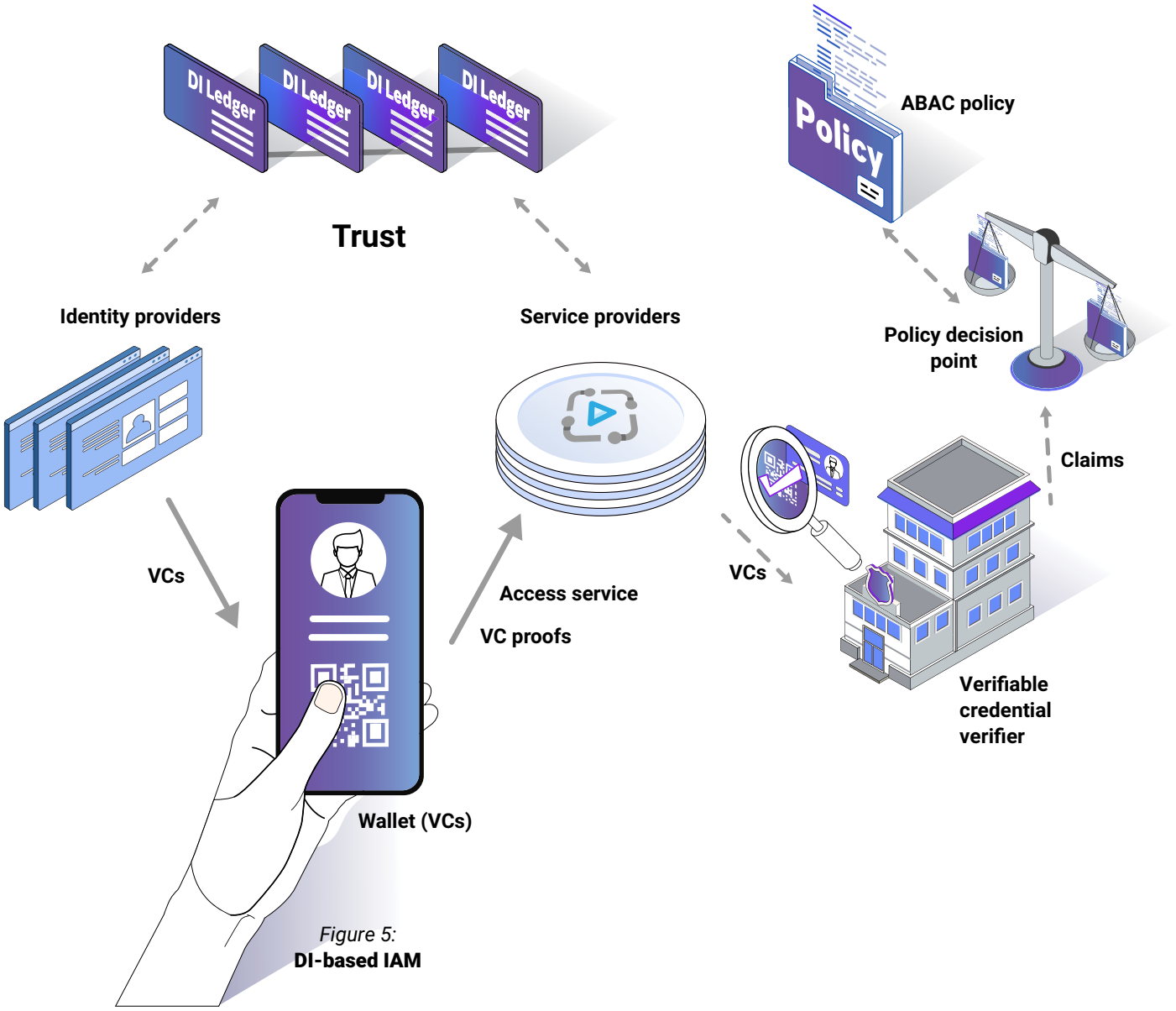
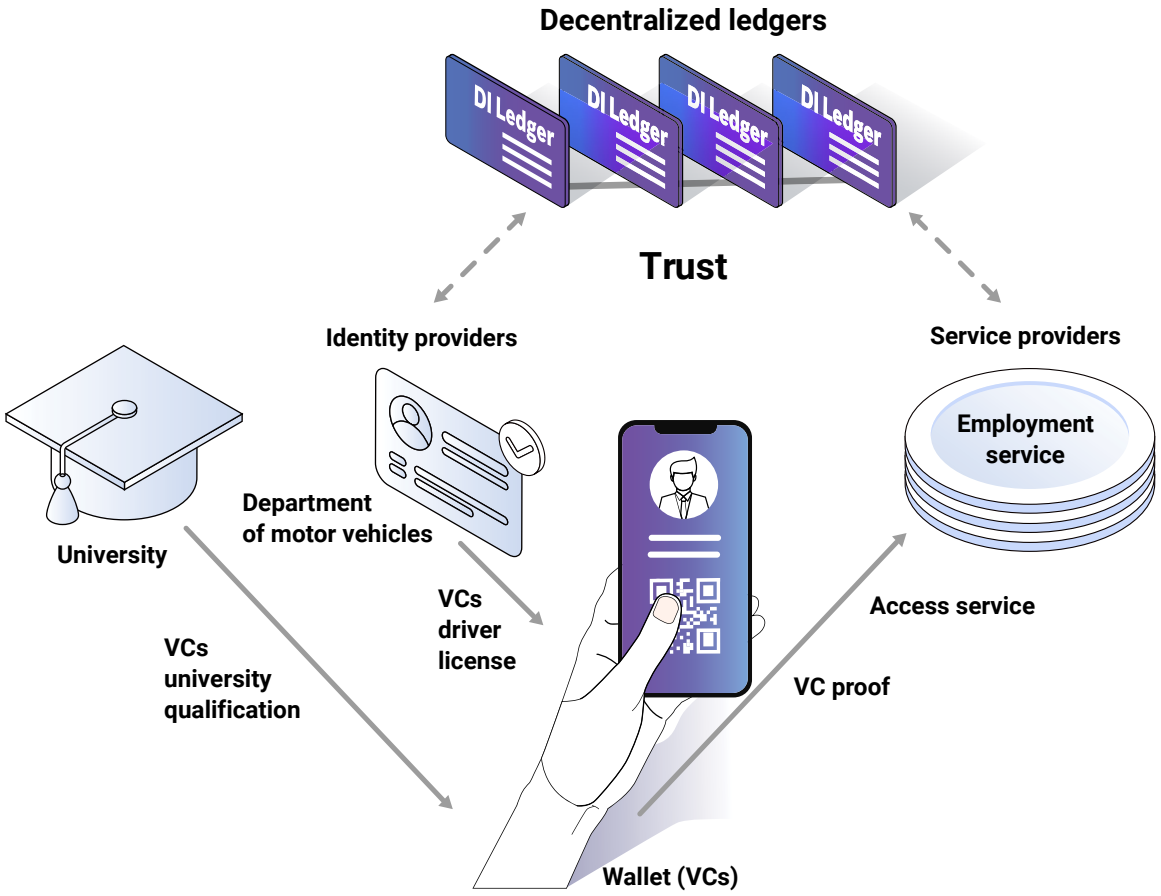


Figure 5:
DI-based IAM

Figure 5 gives an overview of the full DI-based IAM model. The identity providers issue VCs to users. Users store those VCs in their identity wallets. To access a resource from a desired service, the user presents verifiable claims (i.e., elements of the VCs) to the service provider. The service provider's verifiable credential verifier checks the claims have not been altered, are issued by a trusted identity provider, and the containing credentials have not been revoked. If the verifier is satisfied in all those conditions, it can present the claims to the policy decision point so they can make a decision according to the ABAC policy.

A Practical Security Use Case



One of the major advantages of DI is better security but simpler interactions. A user can be issued VCs into their wallet and subsequently present them on request to a service provider.

Figure 6 gives an example of a user who is trying to find a new job. They apply for positions listed at an employment service. Before the employment service will put forward the user's application to prospective employers, the service wants to verify two things:

- User identity (KYC): The employment service asks the user to present their driver license (via a VC proof).
- University qualifications (as listed on their resume): The employment service asks the user to present their university qualification (again, via a VC proof).

Since the user has previously received a VC-based driver license from their state department of motor vehicles (DMV) and a verifiable credential-based university qualification from their university, this process is convenient and easy for both the user and the employment service. Neither of them has to deal with paper copies or credentials (copying them and having them certified) or force in-person meetings. Everything can be done immediately and easily via electronic transfers.

The other advantage for the user is privacy. The employment service does not have to communicate with either the DMV or the university to confirm the VC proofs. Just by accessing the ledger, the employment agency can be sure the credentials are authentic and have not been revoked.

Privacy-Preserving Use Cases

The employment service use case we just discussed shows how DI increases identity security while keeping interactions simpler and more private for the user. In that example, a user conveyed their full identity and their university qualifications to an employment service. This required the secure transfer of specific information elements from the VCs. Many use cases follow this paradigm.

Another type of use case focuses on preserving the user's privacy. These use cases do not require the user to disclose actual data elements from VCs. Instead, they require only a 'true' or 'false' assertion based on those values. The proof method is cryptographically generated using a function known as a zero knowledge proof (ZKP). ZKPs allow a prover to prove a data item to a verifier without disclosing the data item itself.

These next uses cases illustrate when a ZKP-based verification is better than disclosing specific data elements:

- Proof of age: In some countries, individuals must attain a set age before they can perform certain actions, such as purchasing alcohol. To prove their age, the person could use a driver license or passport, because these documents show their actual birth date. But disclosing their birth date would be providing too much PII since they are only really required to prove they are older or younger than the specified age. Instead, the person could use a ZKP to assert they are over the minimum age and safeguard their actual birth date.

- Proof of qualification: Some jobs require a college degree as a qualification of employment. When applying for these jobs through an employment service, a person might only need to use a ZKP to assert they hold a college degree rather than disclose PII such as their alma mater, dates of enrolment, GPA, transcript information, etc. Giving only the required information (possession of a degree) means the user keeps their more unique information hidden and therefore more secure.

- Proof of residency: In some locales, certain rights are conveyed only to residents since they pay taxes and visitors do not. A resident could use their driver license or passport to assert eligibility, but these documents would reveal more PII, such as street address and date of birth. If the resident used a ZKP, they could assert their residency without disclosing any unnecessary personal information that could be misused.

In each use case, the user could prove they meet the specified criteria (e.g., age, college degree, residency) without disclosing more specific information than necessary. This type of credential proof allows people to stay in control of their private information and grant access to it only as needed.

Appendix A: Comparison of IAM Models

	Centralized	Federated	Decentralized
Identity provider	Service provider is the identity provider. User creates an account at the service provider.	One of the major providers (e.g., Google, Apple, Meta, LinkedIn, Twitter …).	Identity providers are verifiable credential issuers (e.g., federal government, state government, health care provider, etc.).
Service provider	Provides the application. Also the identity provider in this case.	Provides the application. Relies on the external identity provider to authenticate user and provide claims.	Provides the application. Relies on the user (through their wallet) to authenticate and provide claims (from VCs issued by the identity provider).
User authentication	User authenticates to service provider with username/password with 2FA.	User authenticates to identity provider with username/password with 2FA. Service provider receives an authentication token from identity provider (e.g., SAML, OpenID Connect).	User approves release of PII to service provider during VCs proof presentation protocol.
User PII	User provides their PII to service provider – typically, email, mobile phone number, credit card, name, birthdate …	User provides their PII to identity provider – typically, email, mobile phone number, credit card, name, birthdate … Service provider will request PII data from identity provider.	User approves release of PII to service provider during VCs proof presentation protocol.

Claims (information about the user)	Service requests user supply PII directly.	Service provider requests PII data from identity provider. Service provider receives token with claims from identity provider (e.g., SAML, OpenID Connect).	Claims about the user are held in VCs in their wallet. User is in control of the release of the PII to service provider during presentation protocol.
Policy	Set by service provider. Can be based on role, group, or user attributes. Used for employees and customers.	Set by service provider. Can be based on role, group, or user attributes. Used mostly for customers.	Set by service provider. Attribute-based access control is the dominant access control model. Could be used for employees and customers.
Trust	Service provider only. No external trust required.	Service provider trusts the identity provider. Identity provider provides signed tokens to the service provider. In some cases, the identity provider issues tokens directly to the user who then can decide whether to present them to the service provider.	Service provider trusts the identity providers. Identity providers create VCs for the user. Service provider can verify the VC without needing to contact the identity provider (by checking Issuer DID, schema and credential definition stored on ledger).