Identifier Methods

Suggest Edits

Verifiable Credentials allow flexible methods for identifying the issuer, the subject, and holder in any given situation. The identifier data type is a URI, which can be a web page, Solid profile, a decentralized identifier (DID), etc. Most of the example code in the documentation and sample implementation use DIDs, but this is exemplary and equivalent functionality using other portable and user-centric identifier systems could be entirely conformant.

The Verite sample implementations usedid:key (a scheme for representing offline/offchain key material in a DID-compatible format based onthe multicodec registry), anddid:web (a simplifed resolution mechanism for using a well-known DNS domain as root of trust for self-publishing DID documents) for convenience and simplicity. You should use the identifier method that is appropriate for your requirements, however, assessing distinctly the privacy, durability, and interoperability/intelligibility requirements separately for issuers, holders, and verifiers.

On the subject/holder side, we chose to model holder-side cryptography in the sample implementation using did:key because it requires no gas/onchain fees nor assumes any blockchain dependencies. It's a purely generative method, enabling resolution offline, without any registry lookups (i.e., the DID documents can be derived from the DID itself deterministically). This enables identity wallet applications to mimic the behavior of existing crypto wallets.

At the same time, did:key does not support key rotation, which is sometimes a desired DID method characteristic, especially for issuers. A simple option for issuers with long-standing web domains and established processes for pushing updates is to use did:web . For a hands-on tutorial on did:web , seedid:web in minutes on the SpruceID didkit developer documentation.

Evaluating DID Methods

There are a large number of DID methods, some of which are based on blockchains, some based on web pages, and some purely generative. Some may support a complete range of DID operations like update, and others (like did:key) may not.

While implementors may choose any method they like, some common factors include:

- technical feasibility
- availability of multiple open source implementations (demonstrating interoperability)
- no/low fee
- the ability for issuers, verifiers, and holder wallets to resolve DIDs without relying on a specific tenant or chain

In general, DID method options will be influenced by a broader set of criteria relevant to all roles in the VC ecosystem, a comprehensive consideration of which is available in theDID Method Rubric .

Wallet-based versus Address-based Holder Identification Schemes

Depending on the trust model and architecture, and crucially on the liability and control structure around a given wallet, Verite issuers may prefer to issue claims (and downstream dapps interpret those claims) either about:

1. a specific blockchain address, or
2. a wallet controller (i.e. legal or natural person) via an internal representation of that wallet's control structure identified by a full-featured, fit-for-purpose DID.

For example, an identity-verification process that provides identity assurance over a multi-address or multi-chain wallet is better served by a full-featured user-controlled on-chain DID (e.g.did:ion).

Conversely, in a situation where identity-verification and wallet-binding are better done separately for each address controlled by an entity (or where relying parties prefer to identify counterparties exclusively by on-chain addresses), the blockchain address itself can be the subject of claims, expressed as a generativedid:pkh for maximum compatibility across contexts and interoperability with other Verite implementations.

Note that this crucial distinction has ramifications throughout the implementation for how credentials are presented and verified, and each favors a different trust model for wallets and intermediaries. TheVerification flow page includes examples of both. Updated3 months ago * Table of Contents * * Evaluating DID Methods * * Wallet-based versus Address-based Holder Identification Schemes