Primer: DIDs, VCs, VPs

Primer: DIDs, Verifiable Credentials, Verifiable Presentations Suggest Edits

Verite is based on several foundational standards to achieve a safer, more robust, more transparent solution for compliance verifications and record-keeping that does not leak identity data, is not private or permissioned, does not rely on metadata mappings or central registries, works with existing identity solutions rather than reinventing, and supports key DeFi use cases.

Decentralized Identifiers (DIDs)

A Decentralized Identifier (DID) is a unique reference that resolves to a DID document. A DID document contains sparse information: A set of one or more public keys used for interacting with the subject of the identifier and a set of optional service endpoints defining how further to interact with the subject of the DID.

DID Documents do not have claims or credentials, no PII, and nothing deemed to be correlatable or assignable to the subject or to any other real world entity. Loosely similar to Bitcoin addresses, which are encoded versions of multi-hashed public keys, a DID contains public keys plus optional service URLs. This enables them to be public, shareable, and portable, and provide references for further identity interaction, while preserving privacy.

DID Document Structure

DID Example

JSON { "@context": ["https://w3id.org/did/v1", "https://identity.foundation/EcdsaSecp256k1RecoverySignature2020#"],

"id": "did:method:123", "authentication": [// implies these keys are used when a verifier authenticates a subject // references the key defined in verificationMethod below "#key-1",

// example of an eth key
"did:ethr:0x0279be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798#controller",

// example of a key defined inline
{
  "type": "Secp256k1SignatureAuthentication2018",
  "publicKey": "did:web:veramo.dev#0405aa19bb98a5fd29c15a730cb5064ca128dea19247b896b1a7bdad0b4bccccda9b47366cd1359e740d938e5a47d7bed0501150e8a1623805ac47c489421b1506"
}

],

"assertionMethod": [// implies keys are used when a verifier verifies claims about this DID "#key-1"], "service": [// optional, but can be used to bootstrap private communication { "id": "did:method:@peerId", "type": "Verifiable Messaging", "serviceEndpoint": "libp2pPeerId", "description": "encrypted p2p messaging between DID service endpoints" }], "verificationMethod": [{ "id": "#key-1", "controller": "", "type": "EcdsaSecp256k1VerificationKey2019", "publicKeyJwk": { "kty": "EC", "crv": "secp256k1", "x": "XpqpVOdyo9WK2M-mESW9-zIg6WvjiHBWrBz8vb-aqWQ", "y": "FgGSe0eLL8FFp4PxqxOv6XWuc2Rt0fEftu0fxQ6ECAs" } }] } DID Anchoring to Public Blockchains

DIDs can be anchored to public blockchains through a number of approaches. Anchoring is the thing that enables DIDs to be portable.

The resolution of a DID from underlying storage and chains, and CRUD operations on DIDs, are handled by DID Resolvers and Controllers, and the method component of a DID URL will determine what kind of resolver the DID needs and expects. Different resolvers use different anchoring mechanisms.

For example,ION (from the DIF and Microsoft) is a resolver that periodically anchors thousands of DID CRUD ops into a single bitcoin transaction, while uPort's ethereum DID resolver (ethr-did) maps one eth account to one DID and leverages a deployed smart contract for managing state changes and metadata such as service attributes.Veramo provides a plugin architecture to utilize several resolvers at once. Any or all of these at once can be used for Verite.

DID Spec

https://www.w3.org/TR/did-core/

Verifiable Credentials

DID may serve as an identifier for both the issuer and recipient ("subject") of Verifiable Credentials (VCs). VCs can represent any number of claims, including proof of KYC and proof of ownership of a blockchain account.

Verite's demo wallet generates and manages DIDs and associated key pairs on behalf of subjects. The wallet sends DIDs to issuers to serve as a subject identifier of a Verifiable Credential. Upon issuance, the wallet holds multiple Verifiable Credentials about its addresses and the real world entities who own those addresses. This allows wallet holders to prove control over the DIDs embedded in the credentials, which in turn enables identity binding for the owner of a wallet.

In a typical decentralized identity case, a verifier would request a Verifiable Credential from an identity holder. That requires discovery or knowledge of the holder first(simple analogy: someone asks you to prove you are vaccinated) .

But Circle's use cases can also involve no knowledge of a holder endpoint at first, in which verifiers query the peer network to find one or more DIDs that can satisfy the claim in the query using a particular Verifiable Credential(simple analogy: someone posts a notice in a public place asking to be privately and anonymously messaged by anyone who chooses to prove they are vaccinated) .

VC Structure

VC Example

JSON { "@context": ["https://www.w3.org/2018/credentials/v1"], "type": ["VerifiableCredential", "BlockchainAccountOwnerCredential" // example credential], "issuer": { "id": "did:example:123" }, "issuanceDate": "2021-01-01T19:73:24Z", "credentialSubject": { "id": "did:example:user", "blockchainAccountOwner": { "type": "EcdsaSecp256k1RecoveryMethod2020", "blockchainAccountId": "0xab16a96d359ec26a11e2c2b3d8f8b8942d5bfcdb@eip155:1" } }, "proof": { "type": "RsaSignature2018", "created": "2017-06-18T21:19:10Z", "proofPurpose": "assertionMethod", "verificationMethod": "https://example.edu/issuers/keys/1", "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..TCYt5XsITJX1CxPCT8yAV-TVkIEq_PbChOMqsLfRoPsnsgw5WEuts01mq-pQy7UJiN5mgRxD-WUcX16dUEMGlv50aqzpqh4Qktb3rk-BuQy72IFLOqV0G_zS245-kronKb78cPN25DGlcTwLtjPAYuNzVBAh4vGHSrQyHUdBBPM" } } VC Spec

https://www.w3.org/TR/vc-data-model/

Verifiable Presentations (VP)

Verifiable Presentations (VP) provide a thin wrapper around one or more Verifiable Credentials. They enable useful features, such as dynamically creating a new VC derived from other VCs without revealing the data contained in the original VCs.

One of their most practical purposes is to prevent replay attacks: They enable proof that the current holder of the Verifiable Credential has the same identity as the subject of the Verifiable Credential whenever the credential was originally issued.

VP Structure

VP Example

JSON { "@context": ["https://www.w3.org/2018/credentials/v1", "https://www.w3.org/2018/credentials/examples/v1"], "type": "VerifiablePresentation", "verifiableCredential": [{ // cut for brevity, this is a full VC as in the above noted structure }], "proof": { "type": "RsaSignature2018", "created": "2018-09-14T21:19:10Z", "proofPurpose": "authentication", "verificationMethod": "did:example:ebfeb1f712ebc6f1c276e12ec21#keys-1", "challenge": "1f44d55f-f161-4938-a659-f8026467f126", "domain": "4jt78h47fh47", "jws": "eyJhbGciOiJSUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..kTCYt5XsITJX1CxPCT8yAV-TVIw5WEuts01mq-pQy7UJiN5mgREEMGlv50aqzpqh4Qq_PbChOMqsLfRoPsnsgxD-WUcX16dUOqV0G_zS245-kronKb78cPktb3rk-BuQy72IFLN25DYuNzVBAh" } } Updated5 months ago * Table of Contents * * Decentralized Identifiers (DIDs) * * Verifiable Credentials * * Verifiable Presentations (VP)