Example Address Token

Suggest Edits

Example Address-Bound JWT

In practice, a wallet that wants to receive and share the simpler address-bound VCs simply needs to be able to handle (and verify signatures on) JWTs that look like this:

JSON { "sub": "did:pkh:eip155:1:0xb9ff5450db13a154d3cc40eb57a619e59bb7d8b9", "nbf": 1660857957, "iss": "did:web:assets.circle.com", "exp": 1663449957, "vc": { "@context": ["https://www.w3.org/2018/credentials/v1", "https://verite.id/identity"], "type": ["VerifiableCredential", "KYBPAMLAttestation", "EntityAccInvAttestation"], "id": "d49b865e-86f1-4954-a2c3-b94f206ca668", "credentialSubject": [{ "type": "EntityAccInvAttestation", "approvalDate": "2022-08-18T21:25:57.285Z", "process": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-entity-accinv-all-checks", "type": "EntityAccInvAttestation" },{ "type": "KYBPAMLAttestation", "approvalDate": "2022-08-18T21:25:57.285Z", "process": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-legal_person", "type": "KYBPAMLAttestation" }], "credentialStatus": { "id": "https://issuer-smokebox.circle.com/api/v1/issuance/status/0#1161", "type": "StatusList2021Entry", "statusPurpose": "revocation", "statusListIndex": "1161", "statusListCredential": "https://issuer-smokebox.circle.com/api/v1/issuance/status/0" }, } }

Example Address-Bound Verifiable Credential (Decoded)

The mandatory properties of a JWT that are redundant to mandatory properties in the Verifiable Credentials data model get converted deterministically (and can, of course, be round-tripped). Note that sub gets moved into the id property inside [each] credentialSubject object rather than staying at the top level.

JSON { "issuer": "did:web:assets.circle.com", "issuanceDate": "2022-08-18T23:25:57Z", "expirationDate": "2022-09-17T23:25:57Z", "vc": { "@context": ["https://www.w3.org/2018/credentials/v1", "https://verite.id/identity"], "type": ["VerifiableCredential", "KYBPAMLAttestation", "EntityAccInvAttestation"], "id": "d49b865e-86f1-4954-a2c3-b94f206ca668", "credentialSubject": [{ "id": "did:pkh:eip155:1:0xb9ff5450db13a154d3cc40eb57a619e59bb7d8b9", "type": "EntityAccInvAttestation", "approvalDate": "2022-08-18T21:25:57.285Z", "process": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-entity-accinv-all-checks", "type": "EntityAccInvAttestation" },{ "id": "did:pkh:eip155:1:0xb9ff5450db13a154d3cc40eb57a619e59bb7d8b9", "type": "KYBPAMLAttestation", "approvalDate": "2022-08-18T21:25:57.285Z", "process": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-legal_person", "type": "KYBPAMLAttestation" }], "credentialStatus": { "id": "https://issuer-smokebox.circle.com/api/v1/issuance/status/0#1161", "type": "StatusList2021Entry", "statusPurpose": "revocation", "statusListIndex": "1161", "statusListCredential": "https://issuer-smokebox.circle.com/api/v1/issuance/status/0" } } }

Example Address-Bound Verifiable Credential (Annotated)

JSON { "issuer": "did:web:assets.circle.com", We recommend handling this property carefully in your UX, since users generally know more about "Circle.com" than they understand anything else you could display to them about a credential. See below on how to fetch and derive the public key needed for verifying signatures on the JWT from this string.

JSON "issuanceDate": "2022-08-18T23:25:57Z", "expirationDate": "2022-09-17T23:25:57Z", It's up to wallet how to display or handle expired credentials, but an expired credential can still be valid (if the public key fetched from the issuer still verifies the signature). We recommend hiding expired credentials by default but allowing them to be found with toggles or settings.

JSON "vc": { "@context": ["https://www.w3.org/2018/credentials/v1", "https://verite.id/identity"], "type": ["VerifiableCredential", "KYBPAMLAttestation", "EntityAccInvAttestation"], In many use-cases, a token actually contains multiple credential/claim types, each expressed as a distinct (and fully-formed) credentialSubject object with a type property in this list. The semantics for these types is explained inthis blog post .

JSON "id": "d49b865e-86f1-4954-a2c3-b94f206ca668", Verite mandates a unique identifier for each credential to facilitate cross-organization (and even forensic) audit trails. We recommend a standard UUID, which can simplify how wallets handle multiple such tokens.

JSON "credentialSubject": [{ Note that multiple credentials are put into the VC as multiple co-equal credentialSubject objects-- each with its own id and type . See theVC data model spec for guidance.

JSON "id": "did:pkh:eip155:1:0xb9ff5450db13a154d3cc40eb57a619e59bb7d8b9", Note that an explanatory prefix before the address, specified in the cross-chaindid:pkh specification based on theCAIP-10 address URN scheme, shows this to be an Ethereum (eip155) Mainnet (1) address. This same convention is used for CACAO receipts of offchain signature-based authentication event, i.e. "web3 wallet connections".

JSON "type": "EntityAccInvAttestation", "approvalDate": "2022-08-18T21:25:57.285Z", "process": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-entity-accinv-all-checks", "type": "EntityAccInvAttestation" },{

"type": "KYBPAMLAttestation", "approvalDate": "2022-08-18T21:25:57.285Z", "process": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-legal_person", "type": "KYBPAMLAttestation" }], "credentialStatus": { "id": "https://issuer-smokebox.circle.com/api/v1/issuance/status/0#1161", This URL can simplify on-going verification-- rather than ask the wallet to re-present this credential at each future transaction or event, a verifier can simply re-check this URL at each transaction taking place between the first verification and the expiration of the credential.

JSON "type": "StatusList2021Entry", "statusPurpose": "revocation", "statusListIndex": "1161", "statusListCredential": "https://issuer-smokebox.circle.com/api/v1/issuance/status/0" } } }

Verifying a

did:web Credential

The did:web: prefix lets you know you're dealing with a DNS-based DID publication mechanism conformant with the DID-Web Method Specification . The rest of the iss / issuer string after the prefix provides a domain (and, optionally, a query string), the controller of which is also the controller and subject of a DID Document. At any given time, you can get the current DID document (containing a public key for verifying signatures) from {the domain}/.well-known/did.json. In the example above, that means https://assets.circle.com/.well-known/did.json , which you can confirm in your browser. Updated5 months ago

* Table of Contents * * Example Address-Bound JWT * * Example Address-Bound Verifiable Credential (Decoded) * * Example Address-Bound Verifiable Credential (Annotated) * * Verifying adid:web Credential