Overview

Suggest Edits

Landscape

Today's identity providers, identity verification services, banks, and other core services that perform some degree of Know-Your-Customer (KYC) and/or Know-Your-Business-Partner (KYBP) services on their customers operate in a predomintrustantly issuer-centric, phone-home world: today's web [2.0] and mobile-app ecosystems are almost 100% powered by API phone-homes and delegation tokens. This is gradually changing, though: third party cookies are being replaced by more nebulous (and platform-centric) tracking models, reducing the centrality of identity providers in the web advertising ecosystem. In parallel, user-centric approaches are increasingly putting cryptocurrency private keys and sensitive identity data directly in user-controlled wallets. The role of an identity provider is shifting, along with their phone-home mechanisms, user experience expectations, and business models. The next big thing is portable identity, user-controlled sharing, and trustless, universally-verifiable tokens. Verite provides a standards-driven approach for empowering your end-users without assuming new risks and liabilities.

If you want your end-users to be able to present a [revocable, tightly-controlled] "badge" that proves them to be your customers anywhere such a badge grants them access to better products and validated-customer prices, you want to be issuing them Verite credentials. Which exact form that takes depends on the answers to a few key questions:

1. What use-cases do you want to start with?Overview of use-cases
2. What liability are you comfortable holding for the credentials you issue?
3. Liability and Auditing Considerations
4. What kind of wallets is it strategic for you to support?Wallet Overview
5. Is the issuance/wallet-interaction something you want to "build or buy"?
6. Architectural options

Use-Cases

In the short-term, the following use-cases are going live in at least one product offering in 2022:

1. VC-based gating of KYB status and domicile (US y/n) for a "company wallet"
2. (i.e., Compliant and Auditable Institutional DeFi)
3. VC-based gating of Investor Accreditation for a "company wallet" (US-domiciled wallets)

The following are in research and design phase, being co-developed by participants and adopters:

1. KYC'd individual wallet (custodial and self-custodial)
2. Non-US KYB status (including interoperability with GLEIF and FincID credentialing)
3. FATF-compliant reporting for custodial-to-custodial transactions (incl
4. custodial-to-noncustodial transactions)
5. Currency controls/FX reporting
6. Verifiable credit-assessment and forensics-sourcing

Liability and Auditing Considerations

Each Verite credential type represents a different liability surface and lifecycle. When designing your Verite engagement, consider the following variables:

- What are the "semantics" (functional content) of each credential you are considering issuing?* KYB credentials basically say "I know this wallet to be controlled by a
  - company that I have KYB'd according to the linked standard", no more and
  - no less
  - Accredited Investor credentials are essentially the same, linking to a
  - different process definition standard
  - FATF reporting requires legal entity information to be verifiable and
  - anchored in auditing and/or registration authorities; the liability
  - considerations are more complex forrelying on

    - these credentials than
-
    - issuing self-attested (non-repudiable) ones
- Which customers will you issue these portable credentials to?
- What can you safely assume about the wallets you are issuing to?
- How do you want to link your logging and record-keeping for these credentials
- to your core identity systems and business model? We recommend storing a copy
- of every VC you ever issue, in a way that can be easily queried at scale at
- least by UUID or other unique per-credential key.

A note on "Uptime" : particularly if you are issuing credentials that may need to be revoked quickly, you should consider whether you are operationally equipped to maintain (24/7, 365) monitoring of real-world data sources like OFAC and PEP lists. Most IDV companies have some kind of realtime monitoring that triggers "push" notifications to clients when statuses change, but with portable, self-certifying credentials like Verite, you don't know whom to push that notification to-- instead, you have to comply with the low latency-tolerance of publishing credential status updates to "revocation lists".

Wallet Overview

Take a minute to ask yourself some difficult strategic questions:

- Depending on your business model, you may be more interested in supporting
- "identity wallets" (applications for signing contracts, handling sensitive
- identity documents, providing verifiable consent, etc) or in supporting
- "cryptocurrency wallets" (that authorize transactions on cryptocurrency
- blockchains and "web3" applications).
- You might be interested in supporting only cryptocurrency wallets with full
- "identity wallet" functionality, or interested in separating the two concerns
- in two distinct pieces of software. (Our sample implementation may provide a
- useful starting point for this latter approach! A browser-extension for
- identity functionality to complement a cryptocurrency wallet is coming soon).
- Retail wallets tend to have long, slow upgrade cycles and governance
- processes. Conversely, many companies contract out to wallet firms to provide
- highly-customized "provisioned wallets" to their employees for managing
- company funds. As Verite capabilities are standardized and rolled out as
- common APIs, these may be a better match for "testing the waters"
- Depending on which exact credentials you issue and your risk tolerance, you
- might have different requirements for identity-assurance,
- sybil-resistance/uniqueness, deduplication, or liveness/biometric binding.
- I.e., if your use-case requires you to be certain that the authorized employee
- is authorizing each transaction of a company wallet, you may want to limit
- your support to wallets with built-in per-transaction or per-session
- biometrics, etc.

Architectural options

At present, the two main options to consider are whether you want to attest to the controller of anblockchain address or to the controller of a specificwallet , which may control multiple addresses in addition to a DID (wallet identifier). For more information, read theidentifier scheme considerations and compare theaddress-bound credential exchange flow and thewallet-bound credential exchange flow .

For address-bound flows, please seeWallet Connect issuance for implementation guidance and code examples, andWallet Connect request presentation for reference.

For wallet-bound flows, please see the pages onwallet-bound issuance mechanics andwallet-bound issuance service setup for implementation guidance and code examples, andwallet-bound credential exchange flow for reference.

Once you have clear your use-cases and your high-level architecture, you arrive at the build-or-buy decision. If you want to issue the credentials you will be responsible for yourself, there are tutorials and documents to guide you through the process in the"For Developers" section of thissite . Updated5 months ago *Table of Contents * * Landscape * * Use-Cases * * Liability and Auditing Considerations * * Wallet Overview * * Architectural options