Decentralizing Trust in Identity Systems

Date: July 17, 2024

Author: Credential Trust Establishment Working Group, Decentralized Identity Foundation

Executive Summary

Recognizing the authorities within an ecosystem is critical to trust. Decentralized Identity and Verifiable Credentials rely on this trust when evaluating credential issuers. API driven approaches are common but suffer from privacy and cost issues. The Credential Trust Establishment Specification allows for efficient ecosystem governance without the drawbacks of other options.

Introduction and Background

Any ecosystem that involves the exchange of data between multiple parties must establish some mechanism for trust between the participants within the ecosystem. We will call such networks Trust Networks.

Some common examples of Trust Networks featuring different scales and architectures include the following:

- **Credit Card Networks**: When you use a credit card, you're participating in a trust network. Banks, merchants, and payment processors collaborate to validate transactions, ensuring that your card is legitimate and you have sufficient funds. This network's trust enables seamless and secure financial transactions.

- **Telecommunications**: Mobile networks operate on trust between service providers to enable roaming. When you travel abroad, your mobile provider relies on a network of agreements with foreign providers to ensure you can make calls, send texts, and use data.

- **Online Marketplaces**: Platforms like Amazon, eBay, and Etsy operate their own trust networks. These platforms verify the identities of sellers and buyers, facilitate payments, and manage reviews and ratings to ensure that transactions are reliable and trustworthy.

Each of these trust networks has an authority — often a singular organization. This organization may be a single company or an established consortium of companies. The authority both establishes and enforces the rules for the operation of the Trust Network. Participants within the

Trust Network refer to and follow the established rules when operating within the network to establish trust in participant interactions.

Application to Decentralized Identity

Verifiable Credentials enable something very powerful: portable trust. Evaluating the signatures of a presented credential can assure the verifying party that the data has not been tampered with after it was first issued by the credential issuer. This enables trust in the data as presented based on the trust placed on the issuer of the credential. Evaluating the credential issuer is therefore a critical part of accepting and trusting Verifiable Credentials.

Trust Networks within Decentralized Identity also have an authority that sets the rules of operation. With Verifiable Credentials, an authority must identify which types of credentials, often identified by a schema, are recognized within the network. The authority must also identify the participants in the network, principally, which network participants are recognized by the ecosystem authority as valid issuers of which types of credentials.

The recognition of valid issuers is critical to evaluating not only the cryptographic validity of the credential but also the authority of the credential issuer.

Trust Network Architectures

Trust Networks must have a mechanism for communicating the authority's decisions about whom to trust to all ecosystem participants. We will discuss some approaches to this technical need as well as recommend the approach we think is superior.

The Opportunity for Trust Foundations

As Verifiable Credentials are adopted, the need for Trust Networks increases along with the opportunity to be an authority of a trust network.

Authority Selection

All existing information ecosystems have an inherent authority — the organization that already sets the rules for trust within the ecosystem they lead. Inherent authorities almost always make the best leaders of trust networks. They understand the ecosystem and its function, and their leadership allows the ecosystem to adapt and grow. If the inherent authority does not exercise its opportunity to lead the ecosystem, another authority may step into the role.

A substitute authority may be used to coordinate a trust network. This substitute authority can be organized in several ways:It could be created as a consortium of companies or by an unrelated party filling a vacuum of authority.

Trust Network Size

A ‘network’ may sound like a very large thing, but it represents ecosystems of every size. A trust network may encompass the employees of a small company or an entire nation’s drivers license issuers.

Ecosystem Expansion

Effective Trust Networks encourage ecosystem growth through clear communication and defined processes. As with any ecosystem, growth in members and participation is often a goal, or it is a consequence of success. Potential partners should be able to understand how to join the ecosystem and the benefits that come from joining.

In addition to expanding the ecosystem by directly adding participants, expansion can also arrive in the form of agreements between Trust Networks. Bilateral or multilateral agreements leave the authorities in place and extend trust across multiple authorities. Larger partnerships can result in a federation model, where many parties enter into such agreements. This is a natural way to group smaller Trust Networks together. This umbrella style grouping is often referred to as a ‘Registry of Registries.’

Trust Network Technical Architectures

Each Trust Network needs a mechanism by which the decisions of the ecosystem authority can be made known to ecosystem participants. How this is technically implemented will impact the functioning of the ecosystem.

It must be noted that our assertions here presume that some organized architecture for trust determination is in use. Making these decisions using bespoke mechanisms is brittle and not recommended. Not evaluating the trust basis of a credential at all is dangerous and will lead to system compromise and critical operational failures.

API-Oriented Architecture

The most common way for separate networked systems to communicate is through an API. A Trust Network using an API architecture places the information about the ecosystem, including valid issuers, into a system that offers an API to ecosystem participants. Ecosystem participants then query the API for trust foundation questions such as “Is Issuer A qualified to issue credential schema B?”

Advantages

- **Familiar**: The API approach is straightforward, familiar to developers, and allows easy integration into software systems.

- **Specific**: This approach handles very large collections of data, as only information about a specific query is returned via a single API call.

Disadvantages

- **Privacy Risks (logging):** Any API solution that returns granular answers will be in a position to monitor activity within the ecosystem. While policies against monitoring seem adequate, they are inevitably overridden by issues such as monitoring justified by devops concerns or even legal monitoring requirements. Common security procedures stipulate logging for network accesses and resulting logs become inputs to analytics systems. These requirements make privacy concerns a near certainty.
- **Cost:** An API requires computational resources to run that cost substantially more than hosted trust policy data. Charging fees for access to Trust Networks for validating credentials leads to increased costs for participants. This risks a "rent-seeking" behavior, where intermediaries extract value from the system without adding proportional benefits. Foundations that operate or control access to trust networks should be cautious about their potential *gatekeeper* role.
- **Scalability:** APIs require significantly more infrastructure to scale over data-oriented methods. In addition to the costs mentioned above, there is a significantly higher management load.
- **Offline Capability**: API-based trust networks require network connectivity to operate and cannot be accessed or queried offline. Overcoming this limitation requires the adoption of a data-oriented approach, described below, as extra effort and complexity.

Data-Oriented Architecture

A data-oriented architecture encodes the information for trust decisions into online resources. These resources are not computational by nature — they are simply static assets that are delivered using familiar web technologies. The assets are read by ecosystem participants, and then processed according to the rules and specifications in order to evaluate the basis of trust for a credential against the published data.

Advantages

- **Simple:** Publishing verifiable data is simpler than setting up and maintaining the systems required for an API infrastructure.
- **Cost:** The costs required to obtain Trust Network governance data via traditional file hosting and downloads is significantly less than those required to operate an API access network.
- **Scale**: Common hosting infrastructure can be used to host data with data-oriented architectures. Scaling the amount of data being downloaded or an increase in usage

incurs a minimal cost increase compared to those required for scaling API-based networks.
- **Offline Capability**: Supports offline processing for data that has been previously downloaded and cached. Since caching data is the standard operation for data-oriented architectures, offline access is inherent to the system. Periodic data refreshing is determined by the accessor.
- **Enhanced Privacy**: As trust data is downloaded as an aggregate data collection and queried locally, the provider of the trust data is unable to determine which trust parties (e.g., elements, individuals, etc.) are being sought. Since the hosting provider cannot see the individual queries, they are unable to identify specific targets, access frequency, or perform pattern analysis from requests alone.

Disadvantages

- **Local Processing**: Downloading trust governance data and processing it locally does require additional local processing; however, the size and nature of governance data (even in large ecosystems) is sufficiently finite that the local processing is viewed as negligible.
- **Local Storage**: While the local storage required is also viewed to be negligible, the data must be stored locally. This is a disadvantage compared to the on-demand accessibility of API-based systems; however, caching systems (e.g., web browsers) have been in use for decades and are very familiar to developers.
- **Refreshing**: Downloaded data should be periodically refreshed by local systems. While this is an additional step, it is also routinely performed by developers.

Recommended Solution: Credential Trust Establishment

In order to provide verifiable governance in Trust Networks and to assist participants in quickly evaluating which credentials, participants, and methods to trust, the Credential Trust Establishment (CTE) specification has been created. At a high level, a CTE is a document (or set of documents) that specifies all the governance roles, rights, credentials, cryptographic mechanisms, protocols, contact points, and identifiers that are required for the system being governed to operate verifiably and securely.

(Note: Since this document is intended to compare and contrast API-based solutions with a Data-Oriented Architectural approach, the specific details of the internal workings of the CTE documents have been omitted, but are available in the CTE specification.)

A CTE document will, typically, package this content using the familiar JSON encoding format, although some systems may opt for customized encodings. This lightweight, file-based format is cryptographically-signed by the governing authority and made available for easy download and verification. While API-based solutions are more significantly impacted by traditional network outages and attacks, as well as, the notable infrastructure costs associated with such high traffic systems, the CTE processes are intended to provide a much more cost effective and resilient solution. This architectural approach will simplify governance system design and deployment for implementers as well as avoid many of the disadvantages of other architectures.

Another benefit of CTE is the versatility and ease by which it can be adapted to service existing systems that don't have trust ecosystems explicitly in place and enhance existing trust networks that need augmentation. It has no opinion on who is eligible to be an issuer or what roles they may take. Further, it references — but does not prescribe — the governance file format.

Who is CTE for?

A CTE-published document can be used by the following parties:

- **Ecosystem Authorities:** Organizations or entities responsible for establishing governance within an ecosystem.
- **Credential Issuers:** Entities that issue verifiable credentials.
- **Ecosystem Participants:** Any individual or organization that needs to verify the trustworthiness of credentials within the ecosystem.

How it works

CTE can scale up or down depending on requirements. The CTE specification includes large-scale and small-scale examples, which are elaborated here.

The general steps for building and releasing a CTE registry are:

1. **Determine the trust context**: Determine the context of the use case and the relevant inherent authorities.
2. **Define the credential schemas**: Identify the credential schemas that will occur in tracing the authenticity of the credential.
3. **Identify the participants**: In addition to inherent authorities, the set of participants includes credential issuers and intermediate authorities.
4. **Establish roles**: Link participants to the credentials they issue and receive.
5. **Finalize metadata:** This includes critical information for ecosystem participants, including versioning information.
6. **Sign and publish the document**: Signing the document allows consumers to confirm the authenticity and integrity of the trust registry itself. The CTE specification describes use of interoperability profiles, which includes cryptographic signature selections.
7. **Share the URI**: Distribute the URI of the published document to ecosystem participants.

The parties within the ecosystem are then configured with the URIs of each CTE document they decide to use as a foundation of trust. Upon interaction with a party or a credential within the ecosystem, they can evaluate the party or issuer against the document to assist in making trust decisions.

The specification contains examples of use in a variety of contexts, and these can be adapted to any use case.

Advanced Topics

CTE usage can begin very simply; however, the specification's advanced features allow it to expand in powerful ways..

CTE Roles by Credential

The CTE Specification contains a mechanism by which participants can be linked to roles within the document by presentation of a credential approved within the document. This provides a number of benefits:

- **Deep Scalability:** Governing very large and layered ecosystems with a small but well organized basis of trust within the CTE document.
- **Full Offline Functionality**: Allowing credentials to be both issued and presented offline without compromise of trust.
- **Authority Delegation**: Placing ecosystem trust decisions within the hands of delegated authorities for efficient operation and reduced administrative bottlenecks.

Scaling with Linked Governance

Each document is capable of linking to other documents. This mechanism can form groups of governance, either as peer governance with links between peers or bundled into a Registry of Registries mechanism with links both down to sub documents and up to the umbrella governance. A single CTE document can be involved in multiple peer and umbrella associations at the same time.

Conclusion

A well-structured system for determining the foundations of trust is critical for a healthy ecosystem. While API-based solutions might, initially, seem appealing, they come with significant drawbacks, including increased operations costs, privacy risks resulting from 'phoning home,' uncertain transparency, and usage fees, all of which can limit and damage ecosystems. The increased operational burden, including technical resource and management costs, disproportionately impacts smaller ecosystems and introduces unnecessary barriers to entry.

Conversely, a data-oriented approach is simpler, faster, cost-effective, and supports privacy-by-design. It easily scales to ecosystems of any size, offline ecosystems, and flexible peer and umbrella relationships with other ecosystems.