ToIP Trust Registry Protocol Specification

This the home page for a draft of the **ToIP Trust Registry Protocol Specification**, a proposed Draft Deliverable of the TSWG developed by the Trust Registry Task Force (TRTF).

Status and Links

Second Generation

The second generation specification is nearing Implementers Draft status. Here is a link to the current Working Draft.

First Generation

A first generation specification was developed by the first iteration of the TRTF:

- Here is the Google doc version.
- Here is the GitHub Markdown version.

This first generation version was published for public review in September 2021. It was not further advanced by the TRTF at that time as there was a consensus that other related specifications needed to be advanced first.

Contributors

To comply with the intellectual property rights protections in the charter of the ToIP Foundation (as required by all Joint Development Foundation projects hosted the Linux Foundation), all contributors to this Pre-Draft Deliverable **MUST be current members of the ToIP Foundation**. The following contributors have certified that they meet this requirement:

- Darrell O'Donnell
- Antti Kettunen
- Andor Kesselman
- Drummond Reed
- Samuel Rinnetmäki, Findynet

Other contributors MUST also add their name and membership affiliation to the Working Draft of the specification as it proceeds through development.

Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL", when appearing in ALL CAPITALS, are to be interpreted as described in RFC 2119.

All other terms in **bold** will be defined in one or more of the ToIP glossaries in a process being defined by the ToIP Concepts and Terminology Working Group.

Purpose

The purpose of this specification is to specify a ToIP Layer 3 **trust task** protocol for standardized interactions with any ToIP-compliant **trust registry**. For more about ToIP Layer 3 trust tasks, see the ToIP Technology Architecture Specification V1.0.

Motivations

A core component of ToIP architecture is a **trust registry**. This is a network service that enables a **governing authority** for a **trust community** to describe the requirements of its **governance framework** and specify what **governed parties** are authorized to perform what **actions** under the **governance framework**. For example:

1. What **issuers** are authorized to issue what types of **verifiable credentials**.
2. What **verifiers** are authorized to request what types of **verifiable presentations**.
3. What other **trust registries** are trusted by this **trust registry**.

As with all layers of the ToIP stack, the purpose of a ToIP specification is to enable the technical interoperability necessary to support transitive trust across different trust communities implementing the ToIP stack. In this case, the desired interoperability outcome is a common protocol that works between any number of decentralized trust registries operated by independent governing authorities representing multiple legal and business **jurisdictions**. One example of a trust community with this need is the digital trust ecosystem defined by the Interoperability Working Group for Good Health Pass (GHP). The GHP Trust Registries Drafting Group produced an extensive set of recommended requirements for a GHP-compliant trust registry.

Trust Registry Protocol v2 Scoping

The v1 trust registry protocol provides answers for three main questions

1. Is this Issuer Authoritative to issue a particular credential type under a governance framework?
2. Is a Verifier Authorized to request a presentation under a governance framework?
3. Does the answering Trust Registry acknowledge another Trust Registry under a governance framework?

The v2 efforts are exploring the following areas to be included in the v2 protocol:

- Approved Wallets - defining the wallet (applications) approved for use in a particular ecosystem.
- Key Parameters Required (early candidate list):
    - DID Methods - listing the DID Methods supported by a governed ecosystem.
    - Credential Types - List the types of credentials that are in use. This includes:
        - Credential Formats - what formats (W3C JWT/JSON-LD/BBS+/etc/., AnonCreds, etc.)
        - Credential Schema - provides the data structure expected in the credentials.
        - Credential Definitions - provides information about each credential type.
        - Revocation Information - provides information about the approach to revocation, where relevant.
    - Roles - lists, in a simple string array, the formal roles that are active in an ecosystem.
    - EGFURI - URI for the Ecosystem Governance Framework, nominally a DID.
    - Assurance Levels - list the assurance levels defined in the EGF for a governed ecosystem.
    - more to come...
- Requirements Capture - a loose capture: Trust Registry Protocol v2 - Loose Capture
- Early input deck: https://docs.google.com/presentation/d/1qQiYTzFrLE4xMFQmgTMM_K6Op-cD6TsjDvUtiDqVmr8/edit?usp=sharing

DISCUSSION AREA

This area is for posting and discussing topics relevant to this Task Force.

EU TRAIN Project

TRAIN stands for "TRust mAnagement INfrastructure". It is a subproject run by Fraunhofer-Gesellschaft within the EU eSSIF-Labs Project. This quote is from a recent post to the W3C Credentials Community Group (CCG) mailing list by David Chadwick:

> In the TRAIN project we have devised an alternative strategy for trusting VC issuers, based on the existing concept of Trust Lists as standardised by ETSI. It works like this.
>
> Every VC that is issued by any issuer, contains a Terms of Use containing the trust scheme(s) that the issuer is a member of (specified as DNS names). These can be true or false statements, it does not matter.
>
> Any DNS owner can create their own trust scheme and decide which VC issuers are members of it and therefore are trusted to issue VCs of a certain type (with a certain schema). The DNS owner adds a URL RR to their DNS entry containing a TRAIN formatted URL.
>
> The TRAIN open source code will read this RR, dereference the URL and expect to find an ETSI trust list published at this https URL. It will then check if the VC Issuer is listed in this DNS named trust list, and if so will tell the Verifier that the issuer is a trusted member of this trust scheme operated by this "DNS name". In this way it does not matter whether the issuer was telling the truth or not. The TRAIN API and the DNS Owner tells the verifier the truth.
>
> Each entry in the trust list has a Service Type Identifier. This is a URL and the web page pointed to should contain the JSON schema (and @context) for the VCs that are issued for this Service Type. In this way the verifier can also find out which attributes the issuer is trusted to issue.
>
> All the verifier has to do is configure the DNS names of the trust scheme owners that it trusts. When it receives a VC, it extracts the asserted trust schemes made by the issuer in the ToU property, and calls the TRAIN API, passing it the ID of the issuer and the trust scheme it is a member of. The TRAIN API will then check if the VC Issuer is a member of this trust list, and if so return the Service Type URL to the Verifier, so that the verifier can validate that the attributes in the received VC match the schema for the Service Type.
>
> Note that the TRAIN source code can be run by anyone, so there can be multiple distributed copies of this service running on the Internet, and verifiers only need to keep pointers to one or more of them to provide them with backup services (much like the dozen or so root DNS name servers).

Credential Chaining

In the same W3C CCG thread, Daniel Hardman made this point:

> I feel like decentralization is running into a difficult tension here: we want to democratize issuance (anyone can do it), but we want to trust a limited set of issuers (or at least, a limited set on any given topic). Anybody can create a COVID test result credential, but we only want to accept them if they were issued by a lab that we have reason to trust. Etc...
>
> One solution to this problem is registries: list trusted sources and have your software check whether the issuer is on approved/accredited list by querying. Of course this re-centralizes around the oracle.
>
> Another solution is **chaining**: have an accreditation authority issue a VC to issuers, attesting to the issuer's bona fides; verification = verify proximate VC + VC that makes proximate issuer trustworthy. Possibly repeat through several levels of indirection. This still centralizes trust (in the ultimate accreditation parties), but at least it decentralizes the verification.
>
> Right now I am more interested in the chaining model, because I think it is more flexible and it reuses our core primitive (VC verification) instead of substituting a different type of lookup. I also think it scales better. But there can be political reasons to use a registry instead.