



Introduction to Verifiable Public Registry

Fabrice Rochette

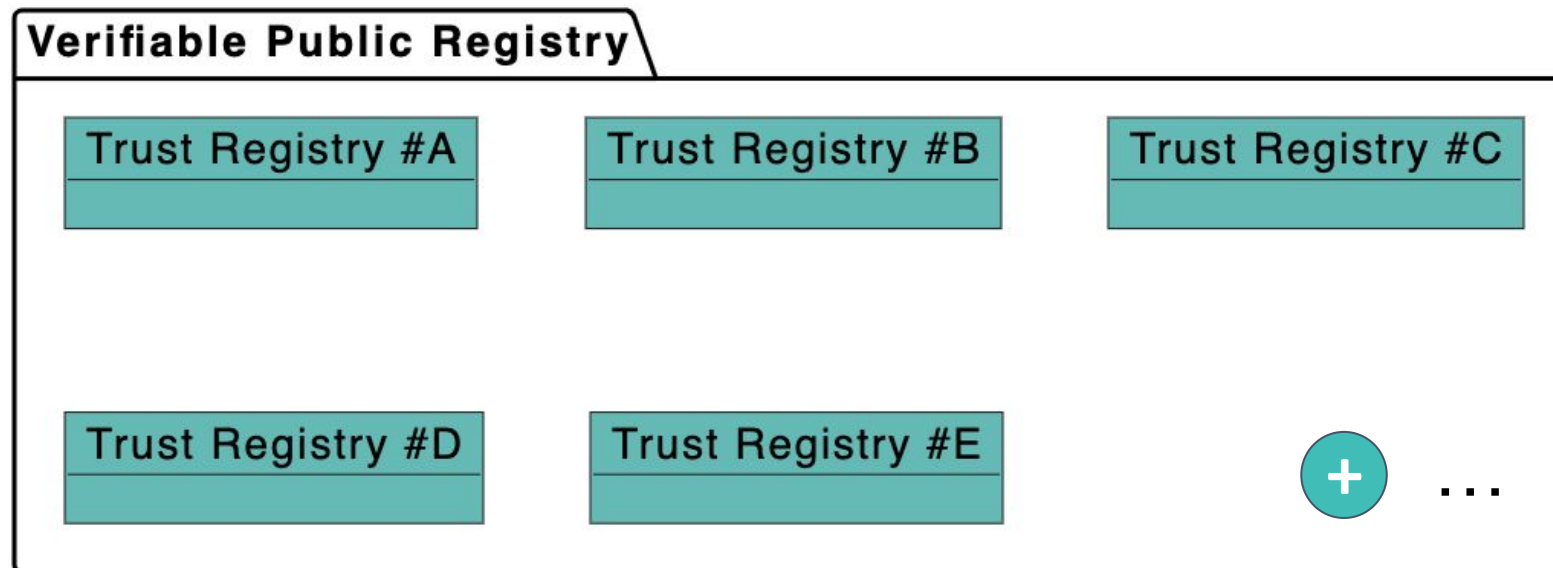
<https://www.linkedin.com/in/fabricerochette/>



What is a Verifiable Public Registry?

A VPR is a public Registry of Trust Registries

Any Ecosystem can create a Trust Registry in a VPR.



Trust Registries

Each Ecosystem **Trust Registry** is identified by a **resolvable DID**, and provides, at least:

- **Governance Framework** document(s).
- Zero or more **Credential Schemas**.

Trust Registry
did schemas governance framework docs

A VPR doesn't care about the DID methods used because trust resolution is performed outside the VPR.

In a VPR, you can use **any DID method**.

Credential Schemas

Credential Schemas

They are created and controlled by **Trust Registries**.

Credential Schemas include:

- Configuration information;
- A **Json Schema** of the Credential Schema.

A **Credential Schema** is immutable.

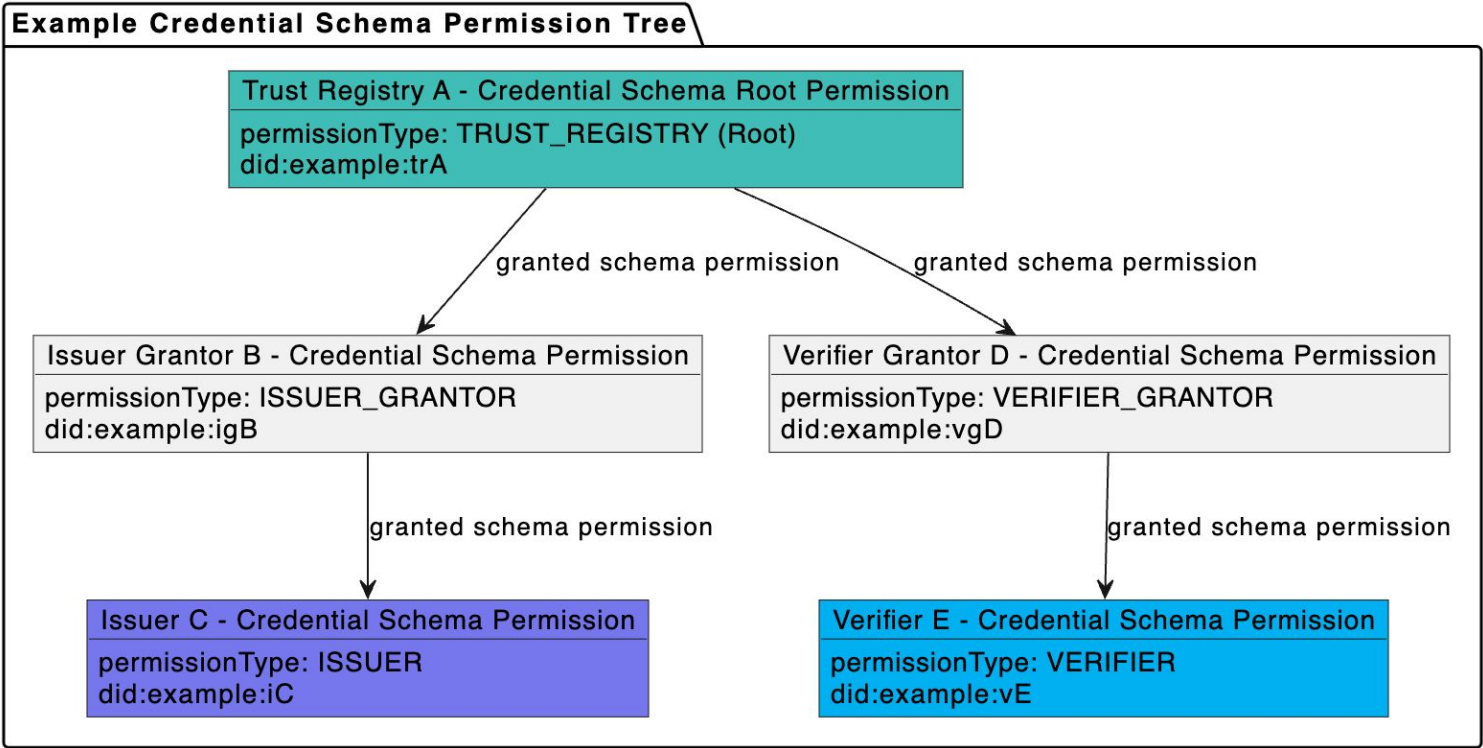
Credential Schema Permissions (CSPs) define who can perform actions related to the **Credential Schema**, such as onboarding **issuers** and **verifiers**, **issue** or **verify** credentials. CSPs define optional business rules.

```
{
  "$id": "vpr-mainnet:/vpr/v1/cs/js/VPR_CREDENTIAL_SCHEMA_ID",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "title": "SimpleExampleCredential",
  "description": "SimpleExampleCredential using JsonSchema",
  "type": "object",
  "properties": {
    "credentialSubject": {
      "type": "object",
      "properties": {
        "id": {
          "type": "string",
          "format": "uri"
        },
        "firstName": {
          "type": "string",
          "minLength": 0,
          "maxLength": 256
        },
        "lastName": {
          "type": "string",
          "minLength": 1,
          "maxLength": 256
        },
        "expirationDate": {
          "type": "string",
          "format": "date"
        },
        "countryOfResidence": {
          "type": "string",
          "minLength": 2,
          "maxLength": 2
        }
      }
    },
    "required": [
      "id",
      "lastName",
      "birthDate",
      "expirationDate",
      "countryOfResidence"
    ]
  }
}
```

Credential Schema Permissions

Each Credential Schema has its own Permission tree

Credential Schema defines which Permission Types are allowed



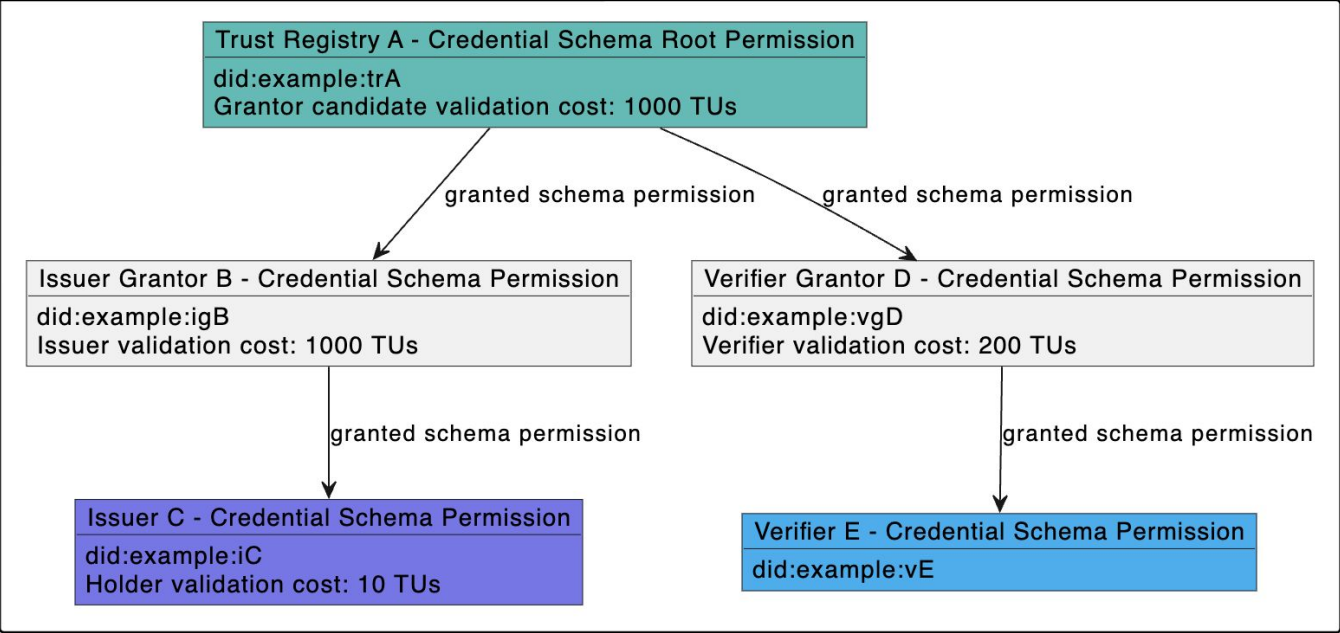
Permission Type	Description
Trust Registry	Create and control Credential Schemas. Grant other roles.
Issuer Grantor	Grant Issuer permissions to candidate issuers
Verifier Grantor	Grant Verifier permissions to candidate verifiers
Issuer	Can issue credentials of this schema
Verifier	Can request presentation of credentials of this schema

Validation Process: to create new CSPs and/or issue VCs

To get granted a CSP, an Applicant must run a Validation process

Applicant starts the Validation Process by selecting a Validator CSP. Validator CSP defines required fees.

Pay per candidate validation Fee Structure



Code	Description	Rate
TD	Trust Deposit	20%

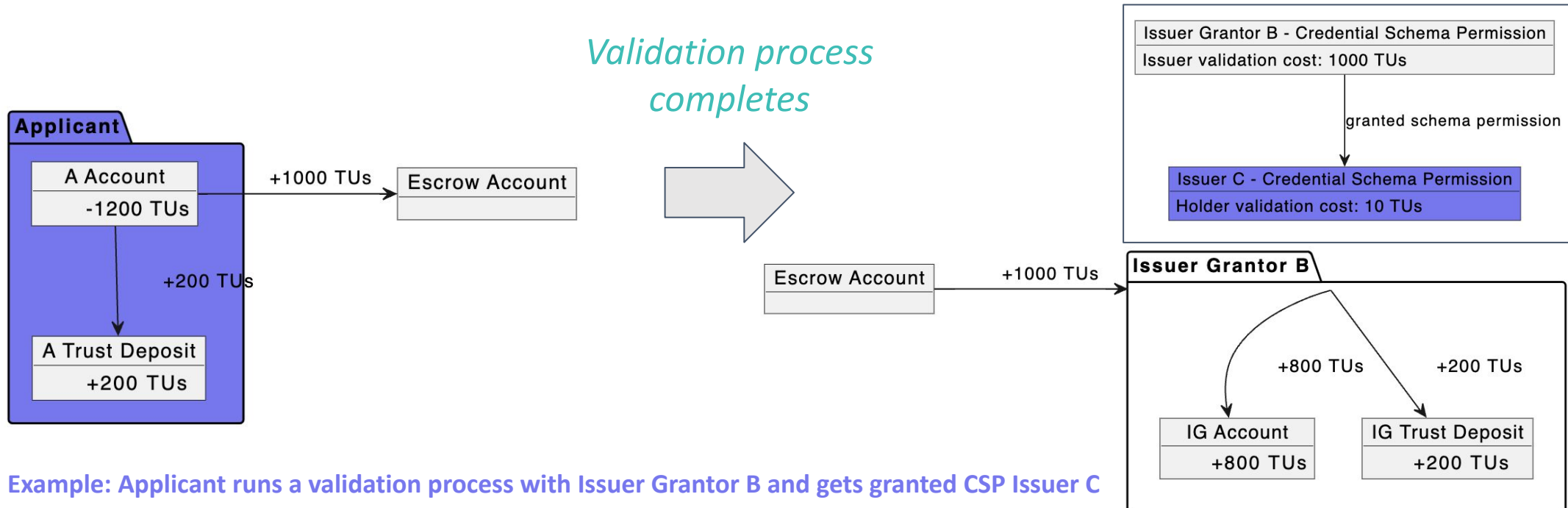
In this example:

- An Applicant will need to pay $1,000 * (1 + TD) = 1,200$ TUs to run a validation process with **Issuer Grantor B** and get granted an **ISSUER CSP Issuer C** for this **Credential Schema of Trust Registry A**

Validation Process: to create new CSPs and/or issue VCs

Validation Process runs, and optional fees are distributed.

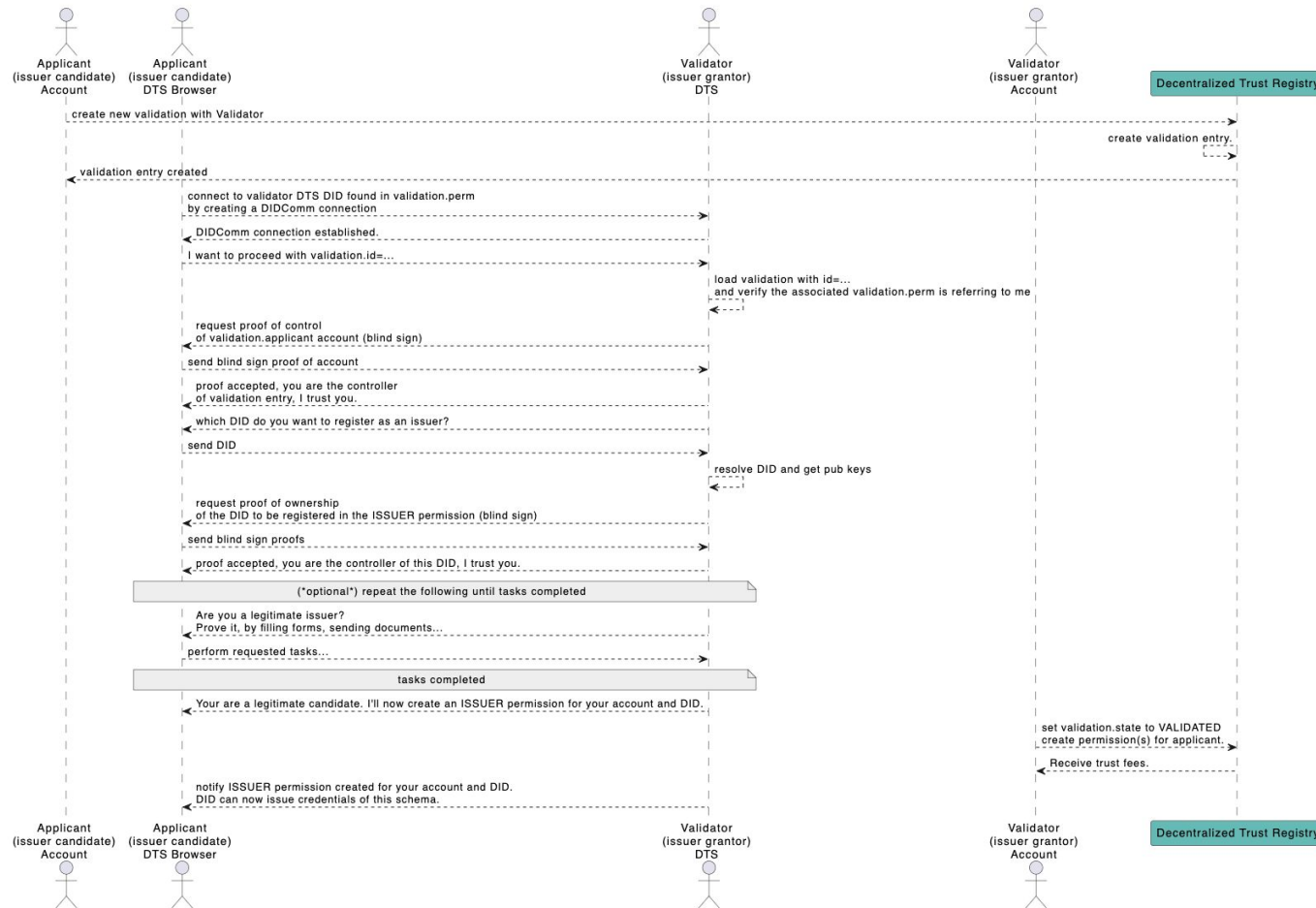
- **Validation** is started, fees paid by **Applicant** are escrowed
- **Applicant** connects to the **DT-Service (DTS)** provided by **Validator** (the **DID** registered in the **Validator's** CSP). They exchange information for completing the **Validation** process.
- When **Validation** process completes, Applicant CSP is created (and/or a credential is issued), then fees are distributed.



Example: Applicant runs a validation process with Issuer Grantor B and gets granted CSP Issuer C

Validation Process: to create new CSPs and/or issue VCs

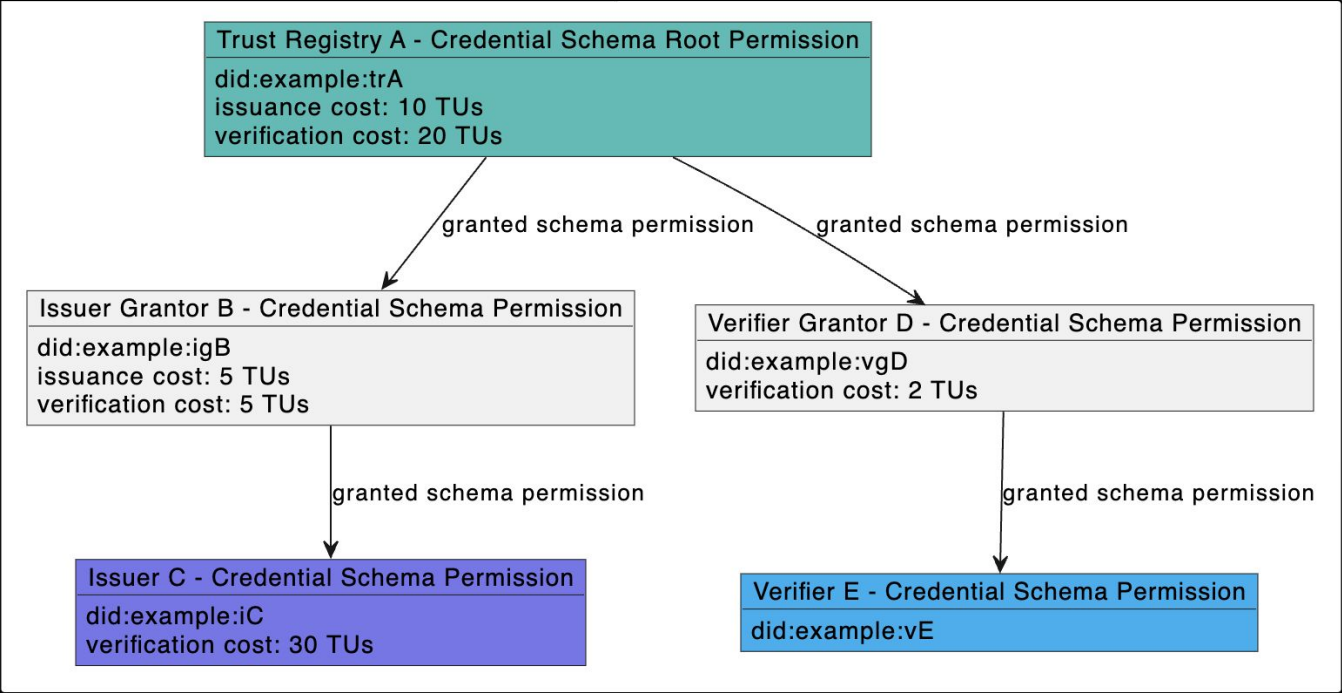
Validation Process example



Pay per issuance/verification: applying CSP rules

CSPs provide a flexible pay per issuance/verification model

Pay per issuance/verification Fee Structure



Code	Description	Rate
UAR	User Agent Rate , for rewarding Apps and Browser and Services that enforce the trust layer	10%
WUAR	Wallet User Agent Rate , for rewarding Wallets and Services that enforce the trust layer	10%
TD	Trust Deposit	20%

In this example:

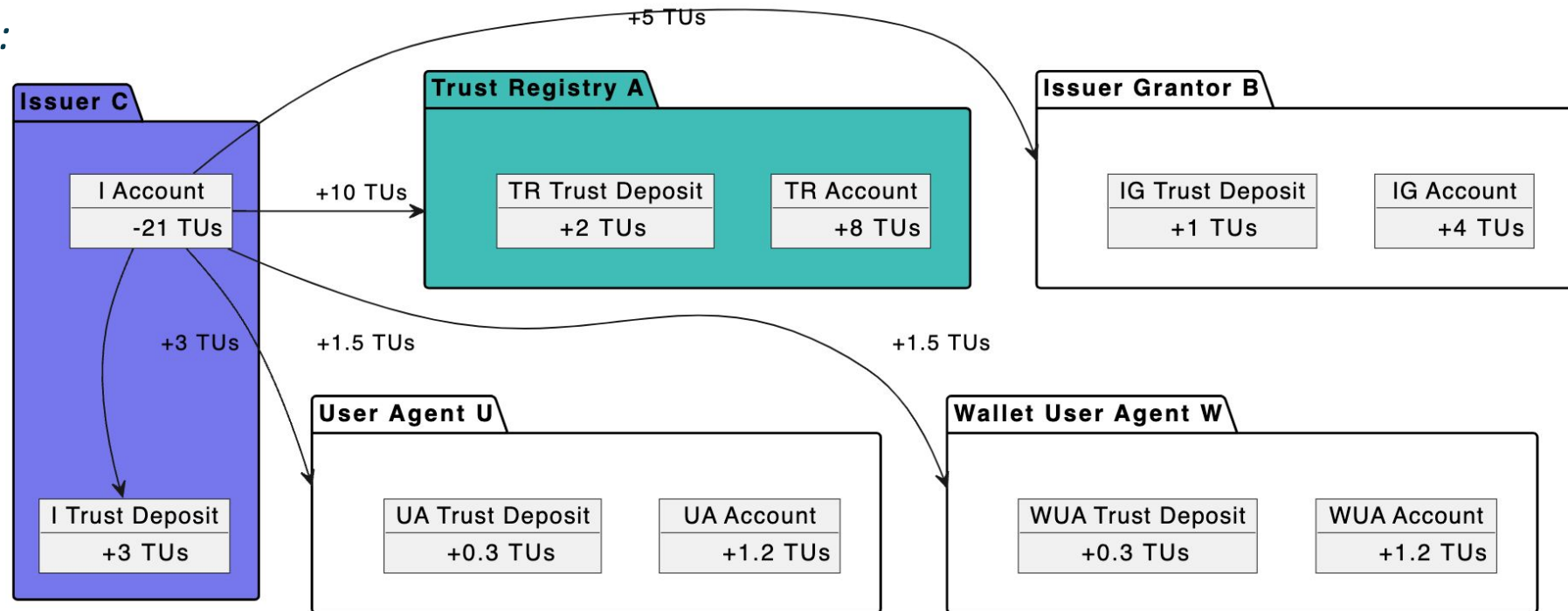
- Total paid by **Issuer C** for issuing a credential: $(10 + 5) * (1 + \text{UAR} + \text{WUAR} + \text{TD}) = \mathbf{21\ TUs}$
- Total paid by **Verifier E** for verifying a credential: $(20 + 5 + 2 + 30) * (1 + \text{UAR} + \text{WUAR} + \text{TD}) = \mathbf{79.8\ TUs}$

Pay per issuance/verification: applying CSP rules

A flexible pay per issuance/verification model that rewards all participants

- If fees > 0, Issuer must create a transaction else **DT compliant wallet** will not accept the credential
- Fee distribution is automatically handled by VPR
- **Privacy Preserving** for Holder

Fees distribution:

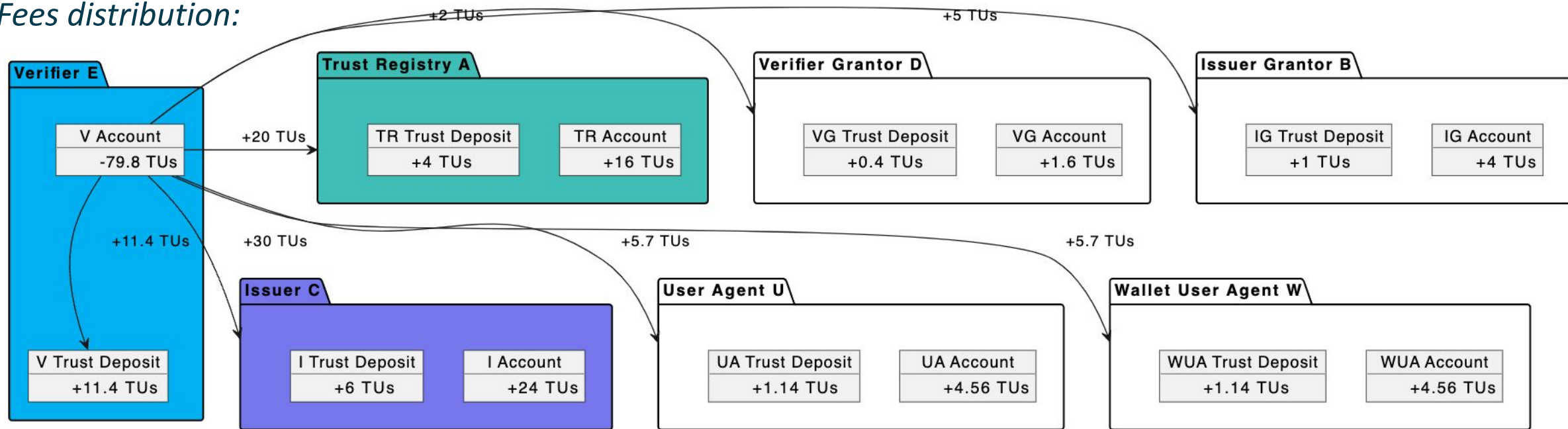


Pay per issuance/verification: applying CSP rules

A flexible pay per issuance/verification model that rewards all participants

- If fees > 0, Verifier must create a transaction else **DT compliant wallet** will not accept the presentation request
- Fee distribution is automatically handled by DTR network
- **Privacy Preserving** for Holder

Fees distribution:

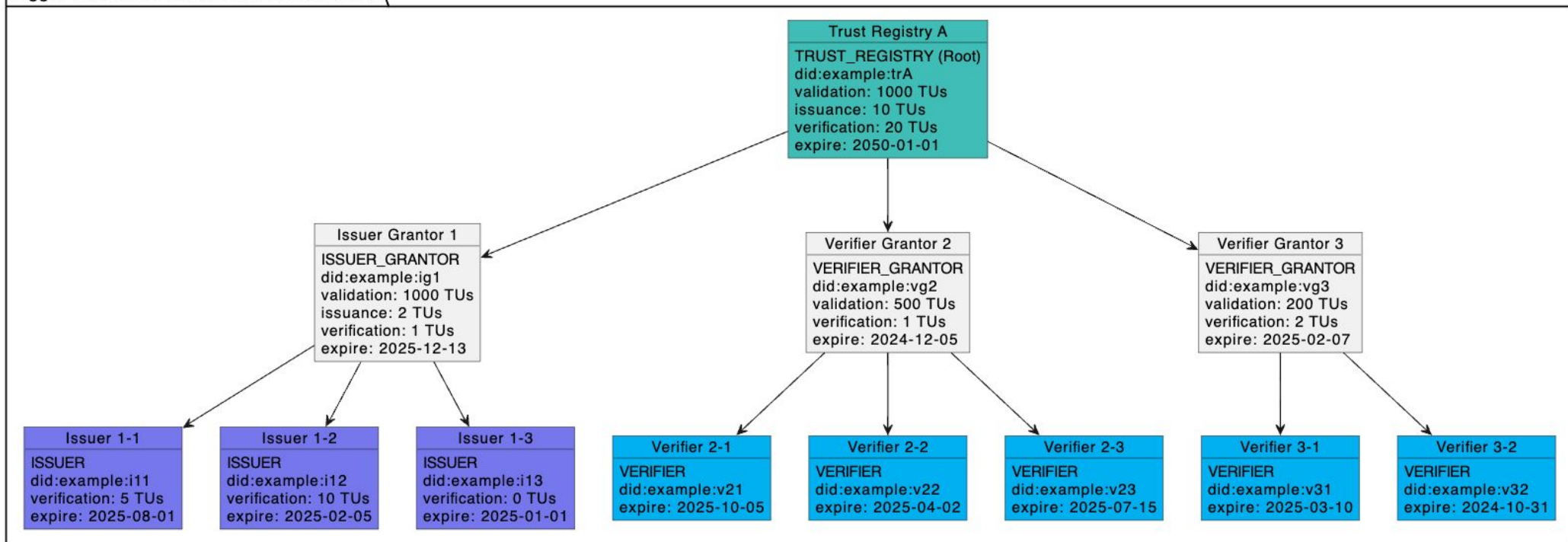


Credential Schema Permissions: Summary

A decentralized way of controlling permissions

- All participants must comply with the Trust Registry Governance Framework
- **Validation** processes are needed for creating and maintaining **CSPs**
- For issuing (resp. verifying) credentials, **Issuer** (resp. **Verifier**) may have to **pay fees**.

Bigger Credential Schema Permission Tree



Trust Resolution

VUAs and VS query the VPR to verify authorizations

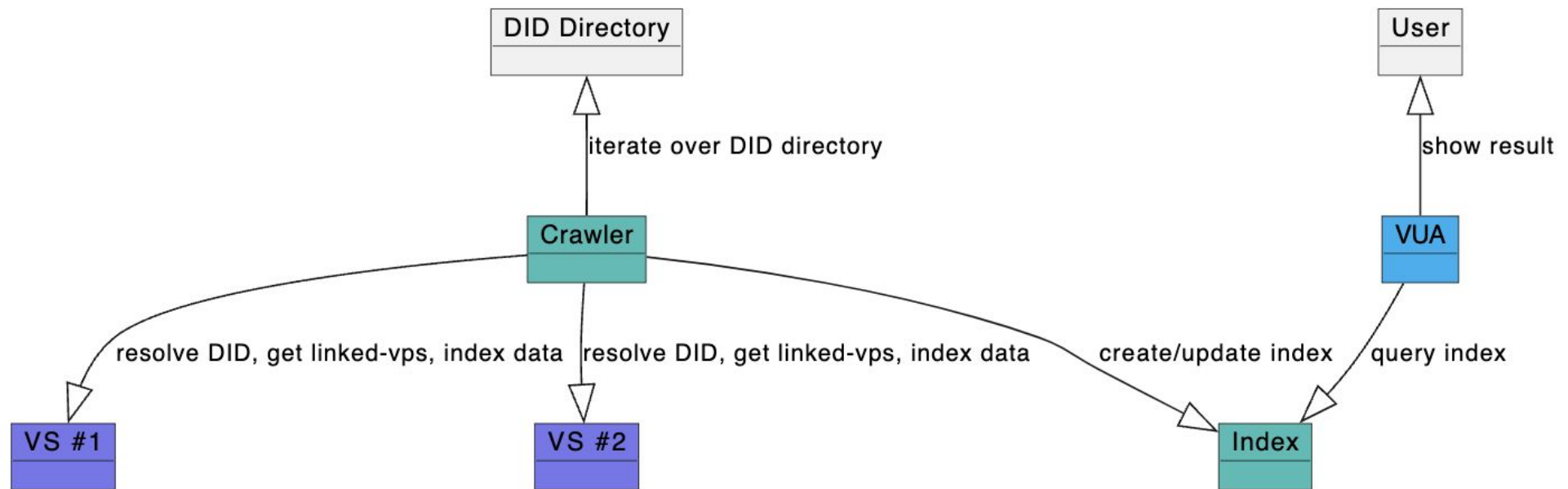
- Is **Issuer** did:example:issuerabc authorized to issue credentials of schema 1234? If payment is required, did the issuer pay?
- Is **Verifier** did:example:issuerabc authorized to verify credentials of schema 1234? If payment is required, did the verifier pay?

If payment is required, a token is sent to the peer and the payer execute a transaction in the VPR, linked to this token, to pay.

DID Directory

A Directory of verifiable services

The DID directory is a **public database of DIDs** that can be used by **crawlers** to build an index of Decentralized Trust - Services (DTSSs). Crawlers simply need to iterate over the DID Directory, and for each DID, try to resolve its DID Document, and dereference all interesting information as explained in the **Verifiable Trust Specification**.

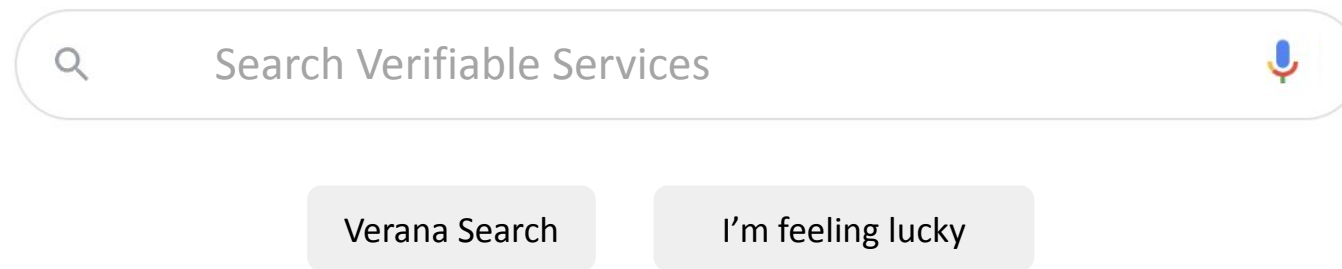


Any participant can register a DID in the DID directory.

DID Directory

Used by search engines, apps...

By using the index built by crawlers, a search engine can be provided to users and they can search for **verifiable services** by querying **verifiable metadata**.



A search interface with a rounded rectangular input field. Inside the field, on the left, is a magnifying glass icon. In the center is the text "Search Verifiable Services". On the right is a microphone icon. Below the input field are two buttons: "Verana Search" and "I'm feeling lucky".

Apps can use an index to let user search only for content/services they support. **Example:** a **Social Network App** could work by indexing **Social Channel Verifiable Services** that present a specific credential only, ie a credential issued by the social network app owner. That's another business model: purchase a credential to appear in a service.

Verifiable Public Registry - Spec

Contributions? Discussions?



<https://verana-labs.github.io/verifiable-trust-vpr-spec/>

Verana Foundation





Building The Missing Trust Layer

Location

- 📍 Ahtri tn 12
10151 Tallinn, Estonia
- 📍 Cra. 13A #86A—42
Bogotá DC, Colombia
- 📍 Paseo de Recoletos 27-41
Madrid, 28004, Spain
- ✉️ f@2060.io

