An introduction to
Verifiable Trust

Fabrice Rochette
https://www.linkedin.com/in/fabricerochette/

IIWXXXIX Fall 2024

PRIVACY NOTICE

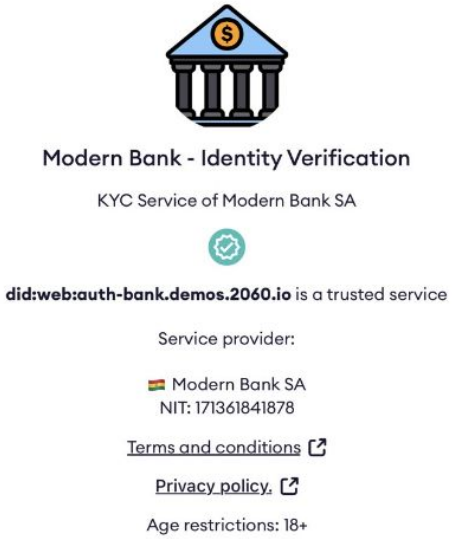
This presentation and all attachments found here constitute intellectual property of **2060 OÜ** and its partners solely, and contain confidential information intended for a specific addressee and purpose. The addressee shall not: (a) disclose, copy, distribute or take any action based on the contents hereof; (b) use the Confidential Information to compete with **2060 OÜ** and its partners; and 9c) acquire any rights (including any Intellectual Property Rights) using the Confidential Information of **2060 OÜ** and its partners included in this presentation.

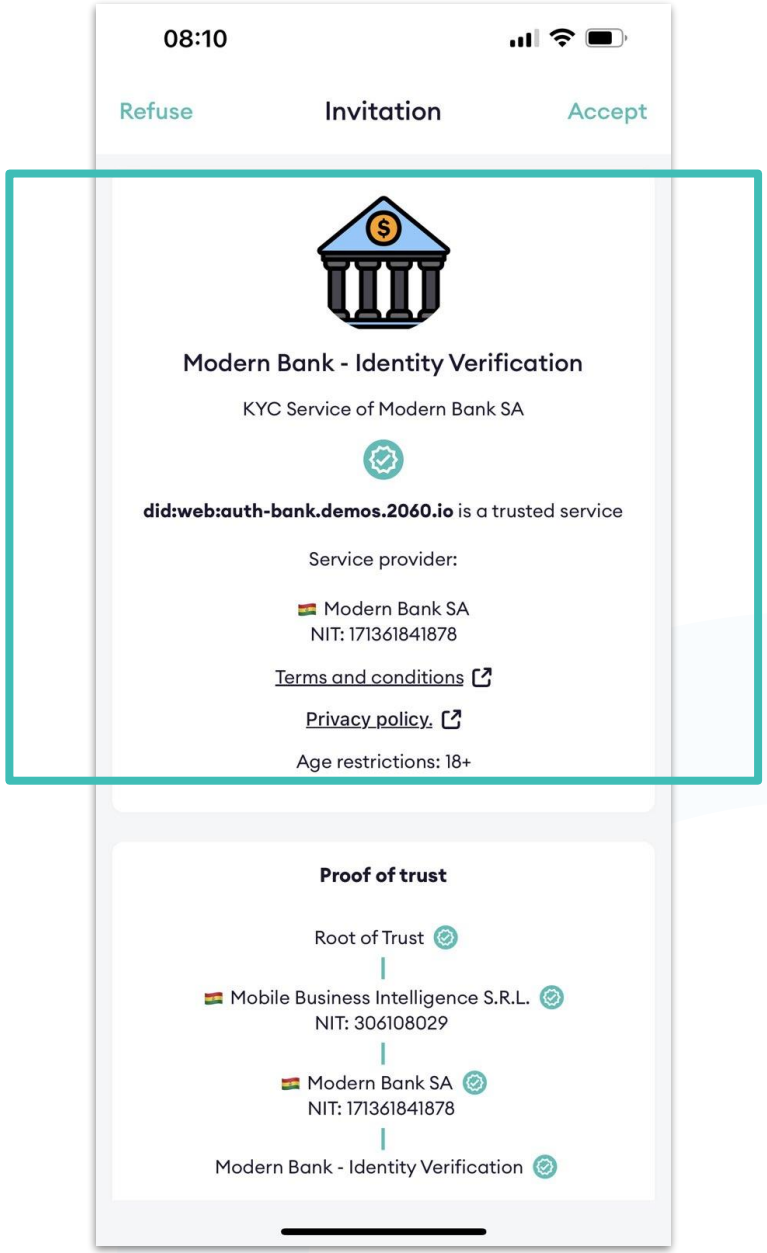
Any copying, publication or disclosure of the content of this presentation, or part hereof, in any form whatsoever, without the sender's express written consent, is prohibited.

Verifiable Service

A **VS** is a service that:

- is able to identify itself with **Verifiable Credential(s) before** connecting to it;
- Is capable of resolving trust of peers that connect to it (**VS** and/or **VUA**) and drop untrustable connections.

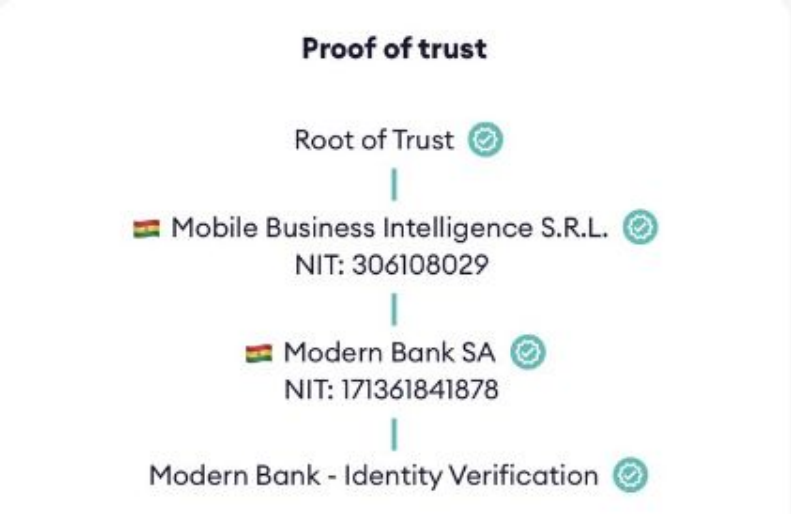


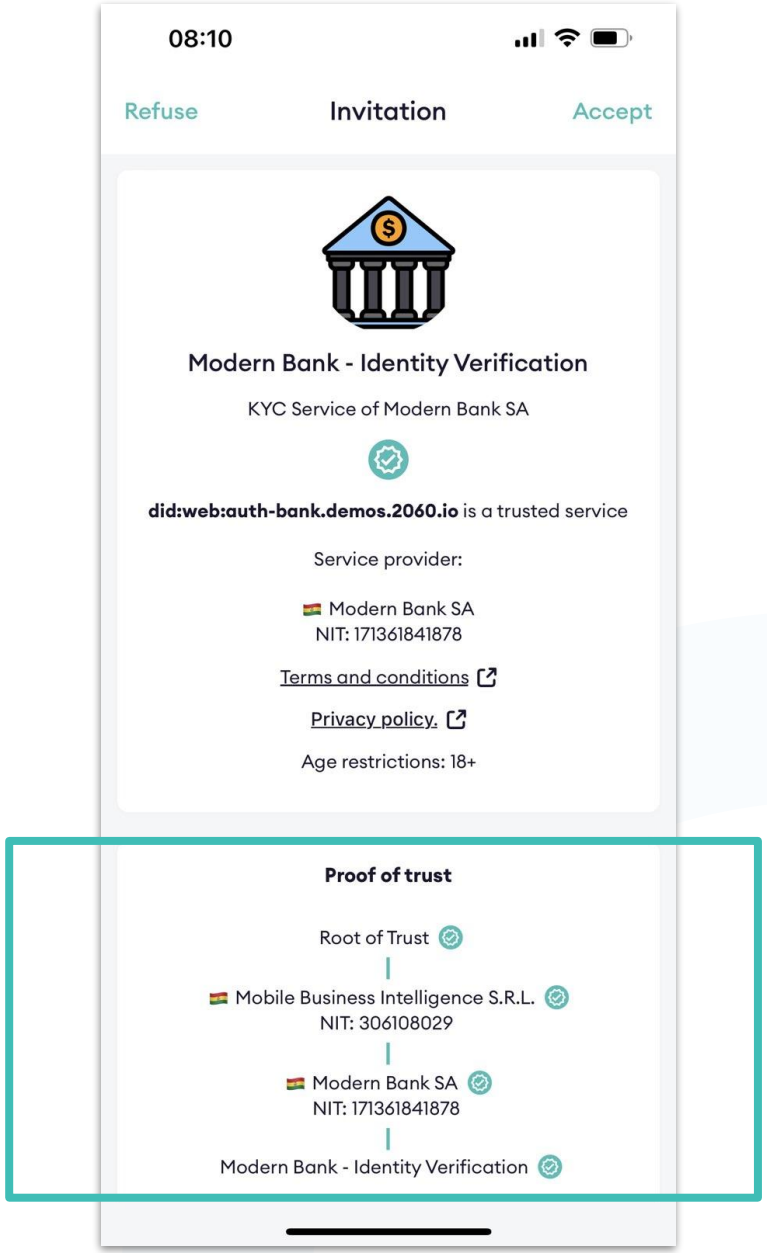


Verifiable - User Agent

A **VUA** is a Mobile App, Browser, Wallet… that:

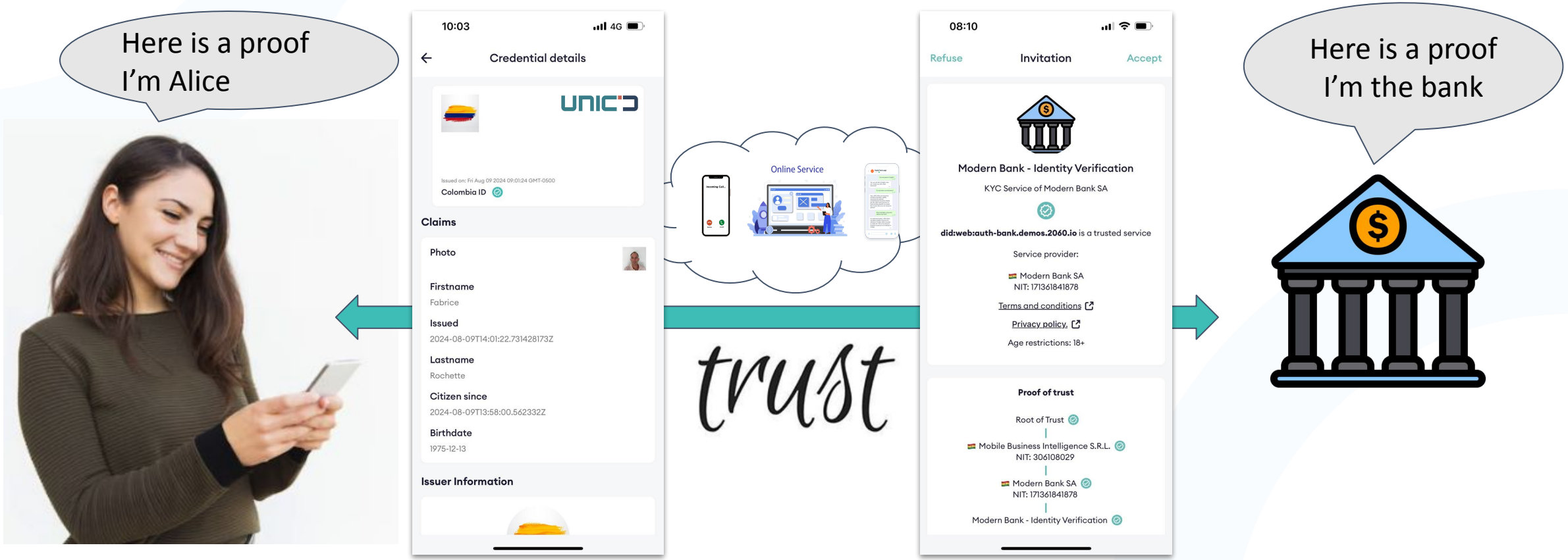
- is able to perform, when user wants to connect to a **VS/VUA**, a **Trust Resolution** and display a **Proof of Trust** of the peer **VS/VUA** to the user, so that user can **decide to connect or not**;
- is able to authenticate itself to peers (**VS** and/or **VUA**).





Verifiable Communication Channel

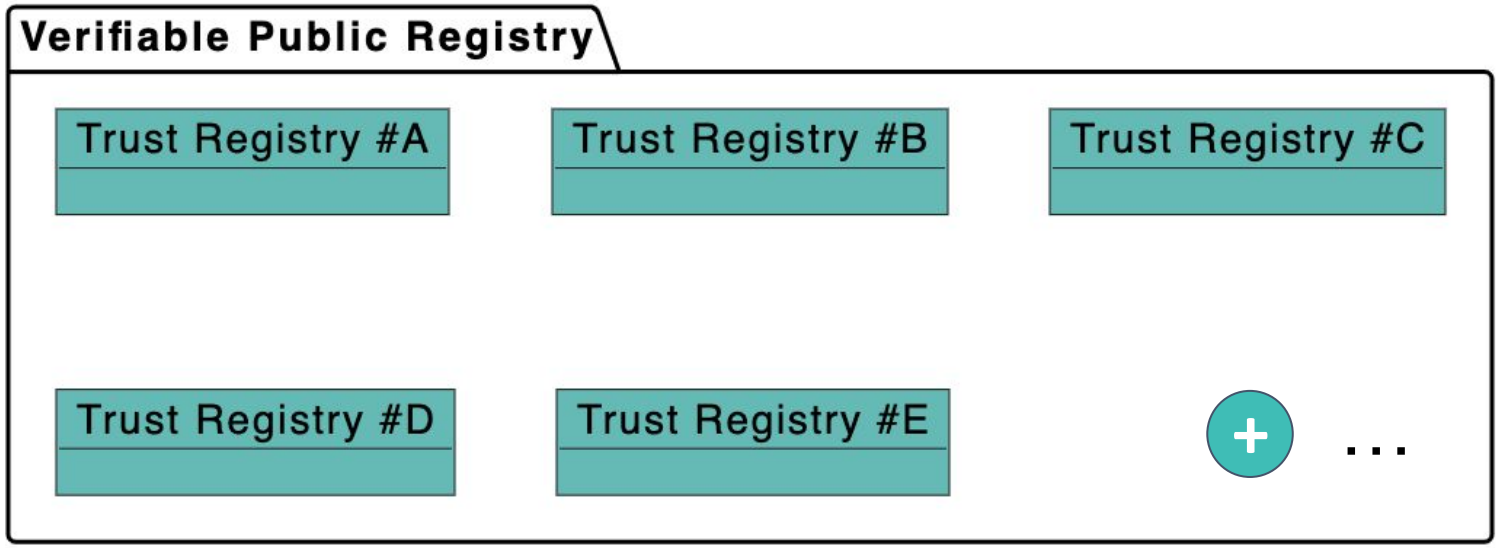
A persistent communication channel where **all participants** are **VS** and/or **VUA**.



Verifiable Public Registry 1/3

A VPR is a public RoR (Registry of Trust Registries)

Anyone can create a Trust Registry in a VPR.



Verifiable Public Registry 2/3

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

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

Trust Registry
did credential schemas governance framework docs

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

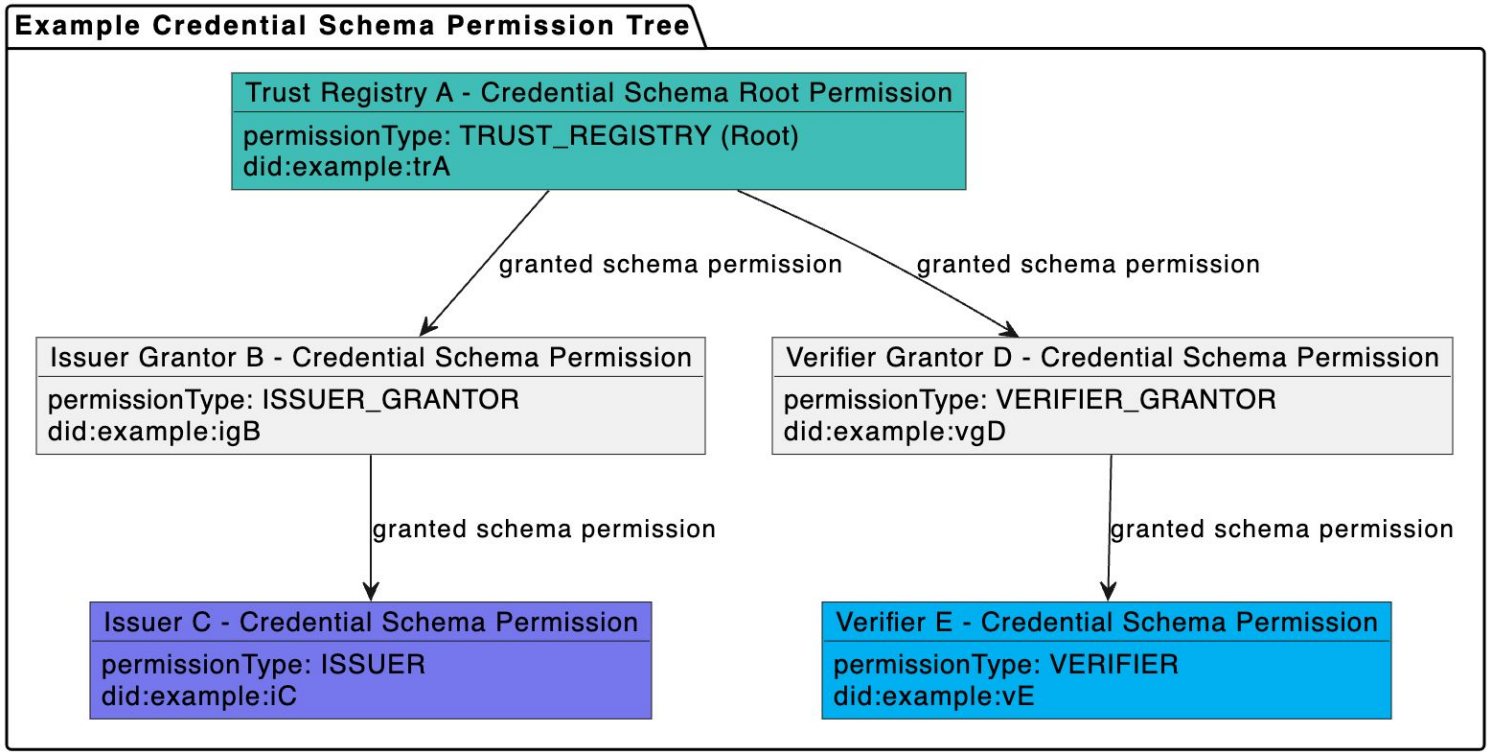
In a **VPR**, you can use **any DID method**.

Note: VPR may be presented in a separate session

Verifiable Public Registry 3/3

Each Credential Schema has its own Credential Schema Permission (CSP) 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

Essential Credential Schema

To resolve basic Trust using the **Verifiable Trust** paradigm, we simply need a **Trust Registry** with **4 basic credential schemas**: the **ECSs.**

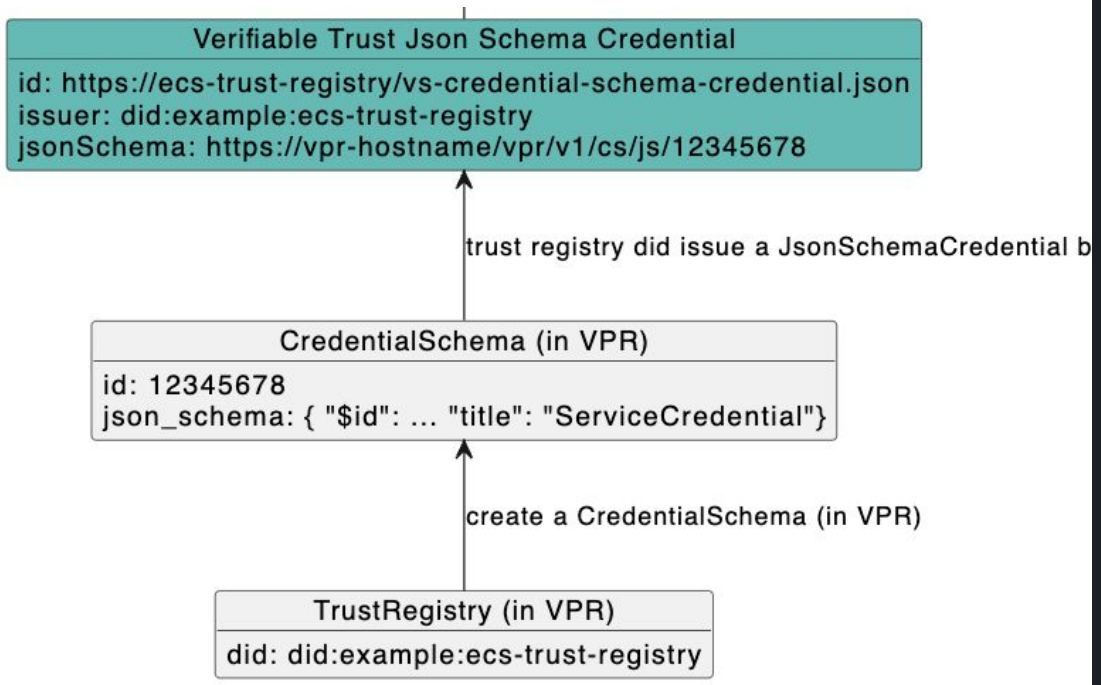
- Service
- Organization
- Person
- UserAgent

That's enough to know **who is who** and perform **Trust Resolution**.

Essential Credential Schema

From Json Schema to linked-vp of a Json Schema Credential

1. **ECSs** are created by **Trust Registry** did:abc:ecs-trust-registry as a **Json Schema** in a **VPR**.
2. For each schema, **Trust Registry DID** issues a **Json Schema Credential** that point to the **Json Schema URI**.




```
{
  "@context": [
      "https://www.w3.org/ns/credentials/v2"
  ],
  "id": "https://ecs-trust-registry/vt-credential-schema-credential.json",
  "type": ["VerifiableCredential", "JsonSchemaCredential"],
  "issuer": "did:example:ecs-trust-registry",
  "issuanceDate": "2024-01-01T19:23:24Z",
  "credentialSchema": {
    "id": "https://w3c.github.io/vc-json-schema/schema/json-schema-credential-schema.json",
    "type": "JsonSchema",
    "digestSRI": "sha384-S57yQDg1MTzF56Oi9DbSQ14u7jBy0RDdx0YbeV7shwhCS88G8SCXeFq82PafhCrW"
  },
  "credentialSubject": {
    "id": "https://vpr-hostname/vpr/v1/cs/js/12345678",
    "type": "JsonSchema",
    "jsonSchema": {
      "$ref": "https://vpr-hostname/vpr/v1/cs/js/12345678"
    },
    "digestSRI": "sha384-ABCSGyugst67rs67rdbugsy0RDdx0YbeV7shwhCS88G8SCXeFq82PafhCeZ"
  }
}
```

Essential Credential Schema

Trust Registry DID Document

3. **Trust Registry**
did:abc:ecs-trust-registry publishes the DT-ECS Json Schema Credentials as **linked-vps** in its **DID Document** as well as a **VPR service entry**.


```
"service": [
  {
    "id": "did:example:ecs-trust-registry#vpr-essential-schemas-service-credential-schema-credentia
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/service-credential-schema-presentation.json"]
  },
  {
    "id": "did:example:ecs-trust-registry#vpr-essential-schemas-organization-credential-schema-crede
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/org-credential-schema-presentation.json"]
  },
  {
    "id": "did:example:ecs-trust-registry#vpr-essential-schemas-person-credential-schema-credential
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/person-credential-schema-presentation.json"]
  },
  {
    "id": "did:example:ecs-trust-registry#vpr-essential-schemas-user-agent-credential-schema-creden
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/user-agent-credential-schema-presentation.json"]
  },
  {
    "id": "did:example:ecs-trust-registry#vpr-essential-schemas-trust-registry-789041745",
    "type": "VerifiablePublicRegistry",
    "version": "1.0",
    "serviceEndpoint": ["https://vpr-hostname/vpr/v1/"]
  }

  ...
]
```

Non Essential Schemas

Of course Trust Resolution is not limited to **DT-ECS**. Anyone can create another **Trust Registry**, this one created a schema for Trademark Creds:

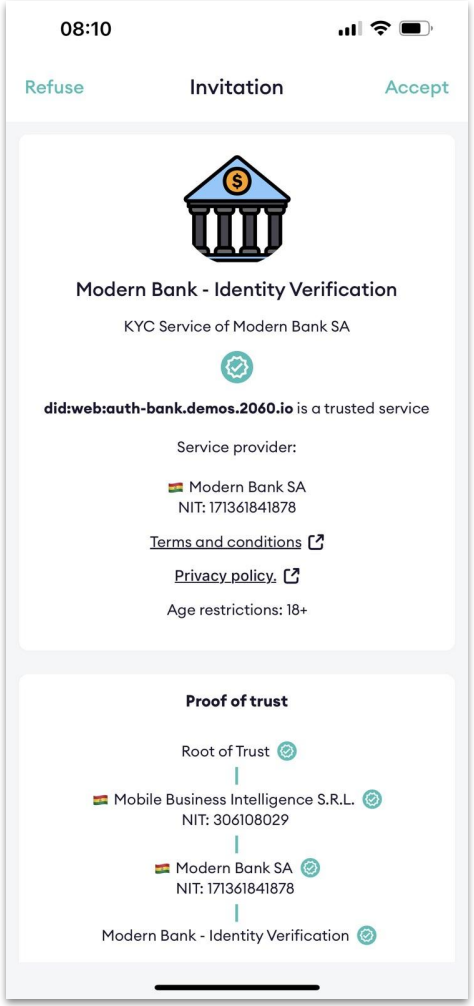
```
...
"service": [
  {
    "id": "did:example:trademark-trust-registry#vpr-schemas-trademark-credential-schema-credential"
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://trademark.abc/credentials/TrademarkJsonSchemaCredential"]
  },
  {
    "id": "did:example:trademark-trust-registry#vpr-schemas-trust-registry-7890",
    "type": "VerifiablePublicRegistry",
    "version": "1.0",
    "serviceEndpoint": ["https://vpr-hostname/vpr/v1/"]
  }

  ...
]
...
```

Verifiable Service

Now we have our ECSs, how a VS DID Document looks like?

```
"service": [
  {
    "id": "did:example:user-vs.example.com#vpr-essential-schemas-service-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://user-vs.example.com/service-credential-presentation.json"]
  },
  {
    "id": "did:example:user-vs.example.com#vpr-essential-schemas-org-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://user-vs.example.com/org-credential-presentation.json"]
  },
  {
    "id": "did:example:user-vs.example.com#vpr-schemas-trademark-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://user-vs.example.com/trademark-credential-presentation.json"]
  }
  ...
]
```



Something similar applies for VUAs.

Trust Registry lists

Compliant VSs and VUAs maintain a list of trusted VPRs

```
{
  verifiablePublicRegistries: [
    {
      "name": "vpr-mainnet",
      "baseurl": "https://vpr-mainnet/vpr/v1",
      "version": "1"
      "production": true
    },
    {
      "name": "vpr-testnet",
      "baseurl": "https://vpr-testnet/vpr/v1",
      "version": "1"
      "production": false
    },
    {
      "name": "vpr-devnet",
      "baseurl": "https://vpr-devnet/vpr/v2",
      "version": "2"
      "production": false
    },
  ]
}
```

```
{
  essentialSchemaTrustRegistries: [
    {
      "tr": "did:example:ecs-trust-registry",
      "vpr": "vpr-mainnet"
    },
    {
      "tr": "did:efg:ecs-trust-registry",
      "vpr": "vpr-testnet"
    }
  ]
}
```

Verana Foundation

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.

Verifiable Trust - Spec

Contributions? Discussions?

https://verana-labs.github.io/verifiable-trust-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