An introduction to Decentralized Trust

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

IIWXXXIX Fall 2024

PRIVACY NOTICE

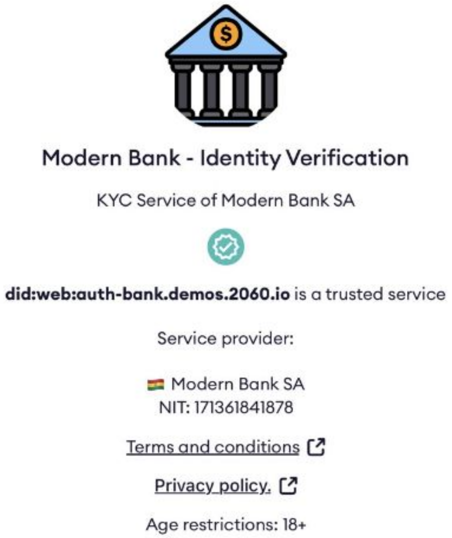
This presentation and all attachments found here constitute intellectual property of **2060 OÜ** and its partners solely, and contain confidential information intended for a specific addressee and purpose. The addressee shall not: (a) disclose, copy, distribute or take any action based on the contents 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.

Decentralized Trust - Service

A **DT-S** 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 (**DT-S** and/or **DT-UA**) and drop untrustable connections.



Modern Bank - Identity Verification

KYC Service of Modern Bank SA

did:web:auth-bank.demos.2060.io is a trusted service

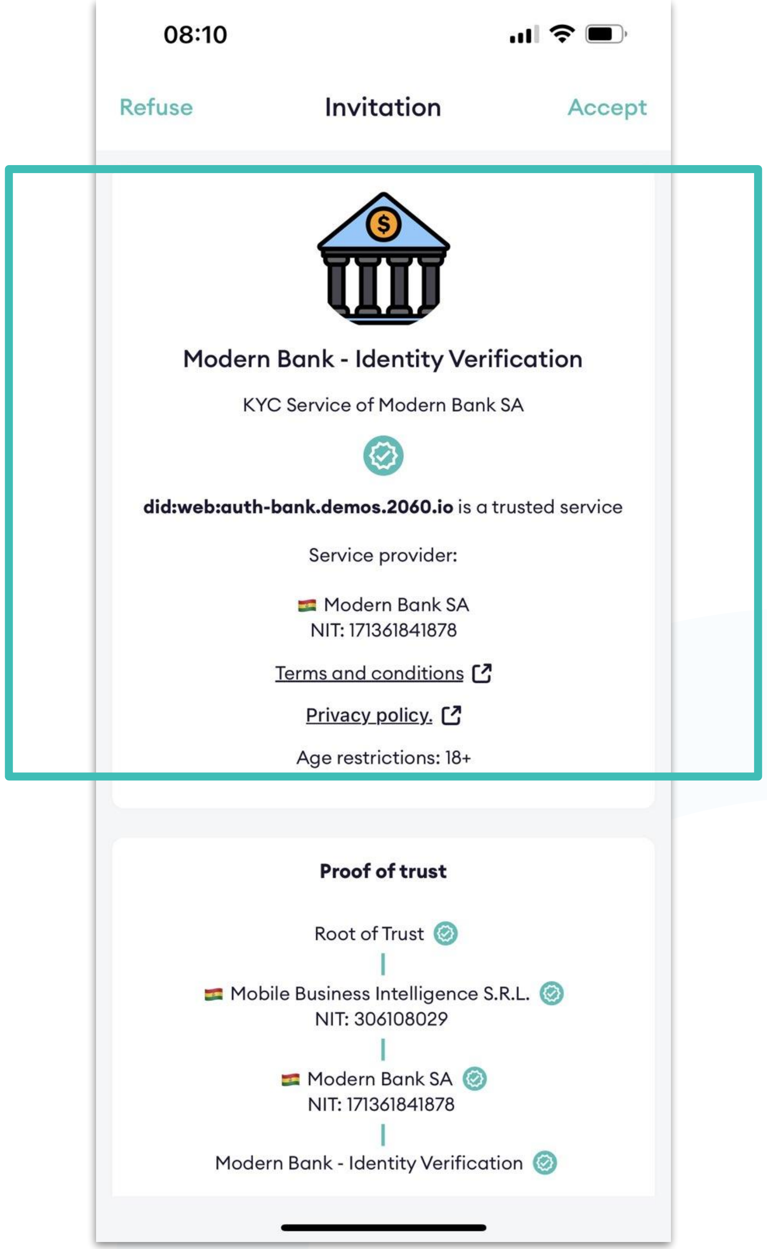
Service provider:

Modern Bank SA
NIT: 171361841878

Terms and conditions

Privacy policy.

Age restrictions: 18+



08:10

Refuse | Invitation | Accept

Modern Bank - Identity Verification

KYC Service of Modern Bank SA

did:web:auth-bank.demos.2060.io is a trusted service

Service provider:

Modern Bank SA
NIT: 171361841878

Terms and conditions

Privacy policy.

Age restrictions: 18+

Proof of trust

Root of Trust

Mobile Business Intelligence S.R.L.
NIT: 306108029

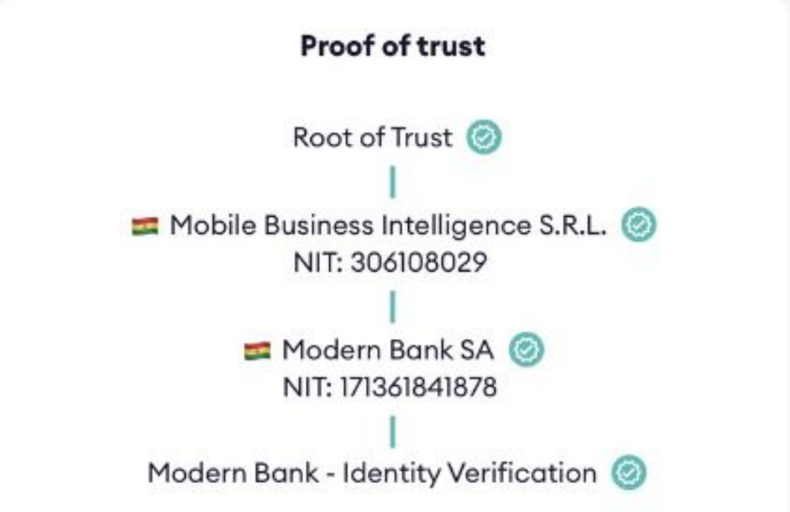
Modern Bank SA
NIT: 171361841878

Modern Bank - Identity Verification

Decentralized Trust - User Agent

A **DT-UA** is a Mobile App, Browser, Wallet… that:

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



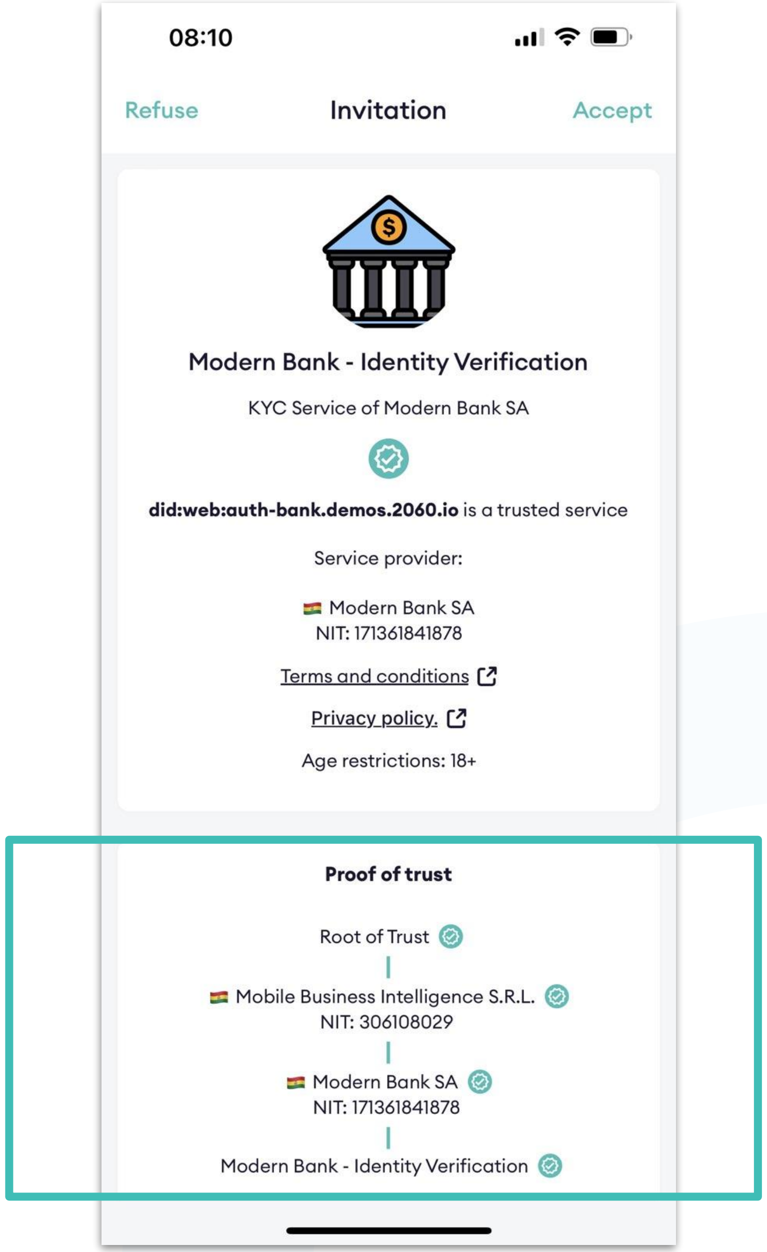
Proof of trust

Root of Trust ✓

Mobile Business Intelligence S.R.L. ✓
NIT: 306108029

Modern Bank SA ✓
NIT: 171361841878

Modern Bank - Identity Verification ✓



08:10

Modern Bank - Identity Verification

KYC Service of Modern Bank SA

✓

did:web:auth-bank.demos.2060.io is a trusted service

Service provider:

Modern Bank SA
NIT: 171361841878

Terms and conditions

Privacy policy.

Age restrictions: 18+

Proof of trust

Root of Trust ✓

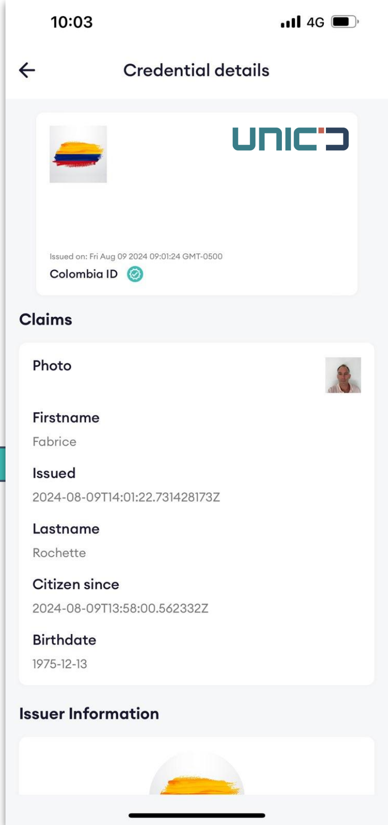
Mobile Business Intelligence S.R.L. ✓
NIT: 306108029

Modern Bank SA ✓
NIT: 171361841878

Modern Bank - Identity Verification ✓

Decentralized Trust - Trustable Communication Channel

A persistent communication channel where **all participants** are **DT-S** and/or **DT-UA**.



10:03 4G

Credential details

UNICO

Issued on: Fri Aug 09 2024 09:01:24 GMT-0500

Colombia ID

Claims

Photo

Firstname
Fabrice

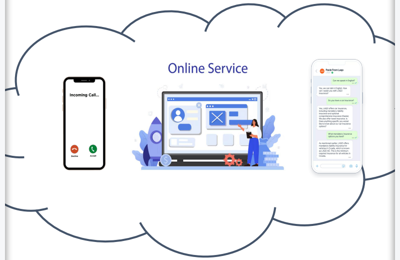
Issued
2024-08-09T14:01:22.731428173Z

Lastname
Rochette

Citizen since
2024-08-09T13:58:00.562332Z

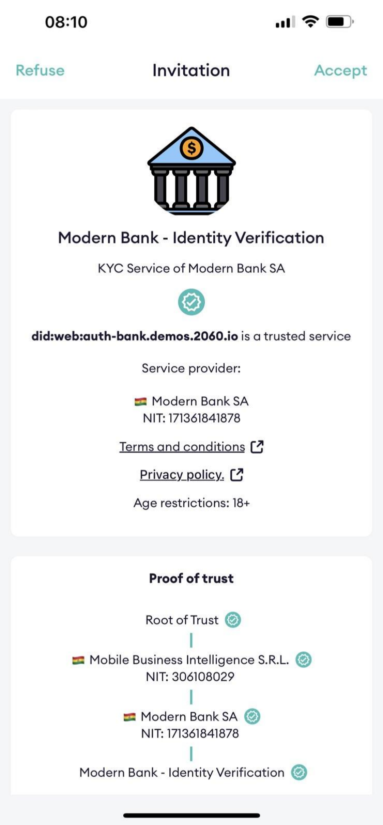
Birthdate
1975-12-13

Issuer Information



trust

DT-TCC



08:10

Refuse Invitation Accept

Modern Bank - Identity Verification

KYC Service of Modern Bank SA

did:web:auth-bank.demos.2060.io is a trusted service

Service provider:

Modern Bank SA
NIT: 171361841878

Terms and conditions

Privacy policy.

Age restrictions: 18+

Proof of trust

Root of Trust

Mobile Business Intelligence S.R.L.
NIT: 306108029

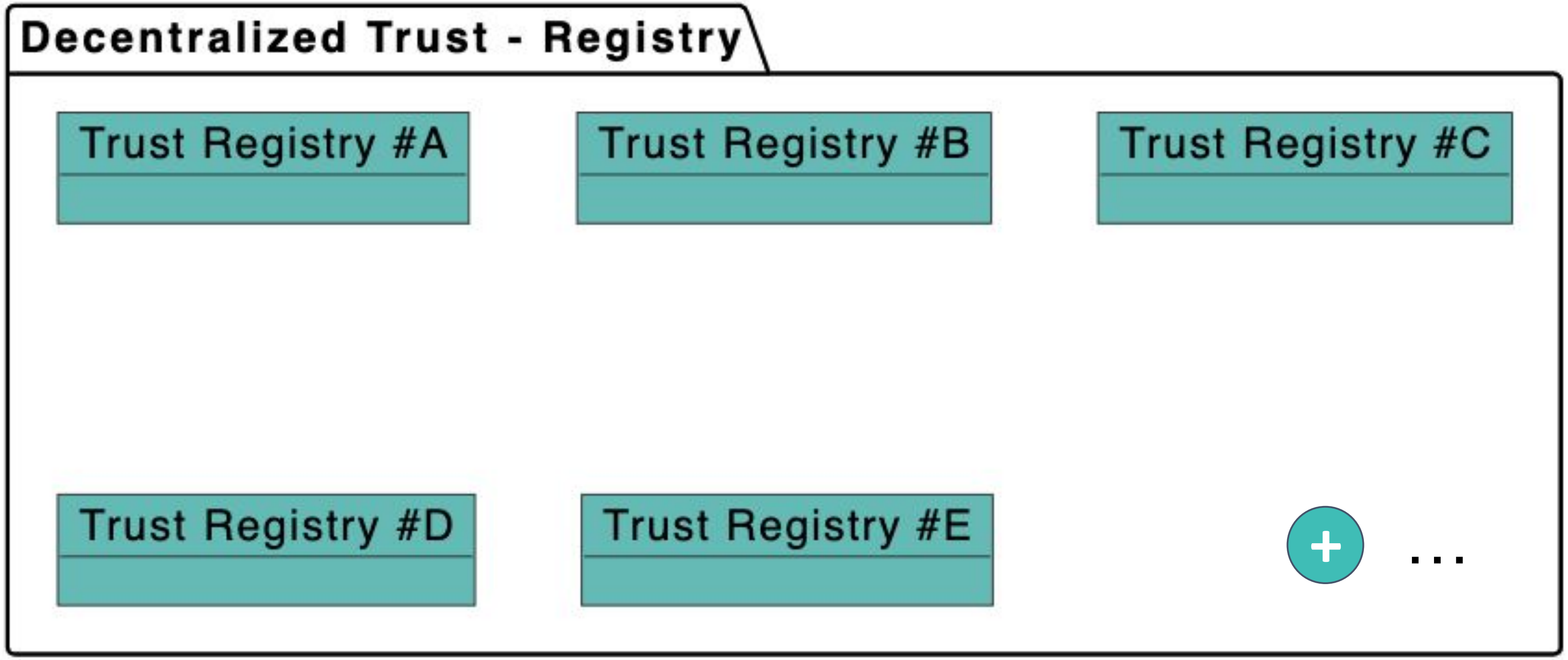
Modern Bank SA
NIT: 171361841878

Modern Bank - Identity Verification

Decentralized Trust - Registry 1/3

A DT-R is a public RoR (Registry of Registries)

Anyone can create a Trust Registry in a DT-R.

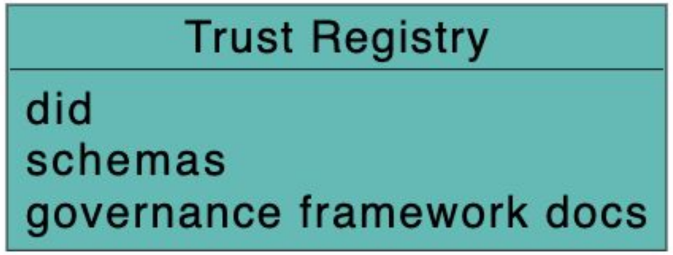


It's **Decentralized Trust - Registry**
Not **Decentralized - Trust Registry**

Decentralized Trust - Registry 2/3

In a **DT-R**, 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
schemas
governance framework docs

A **DT-R** doesn’t care about the DID methods used because DT resolution is performed outside the DT-R.

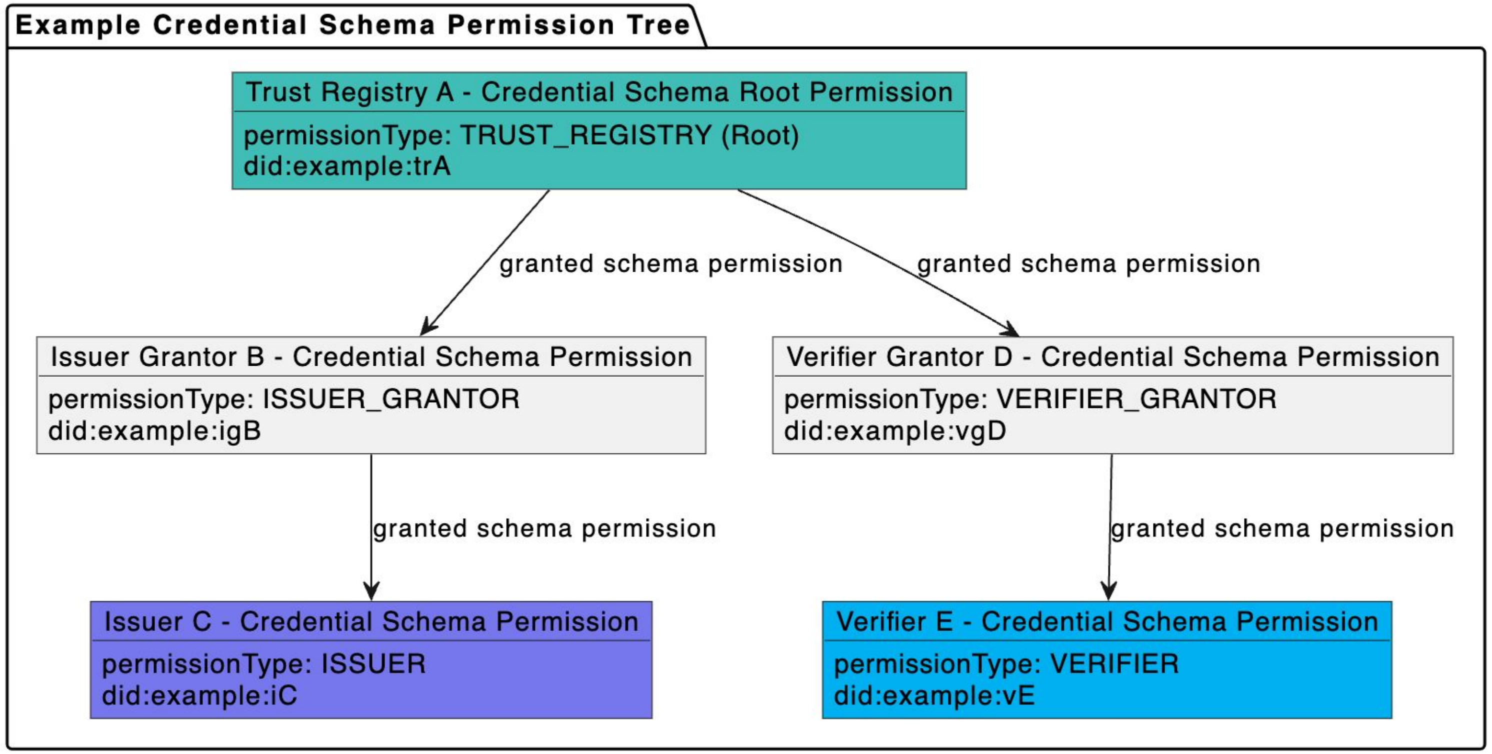
In a **DT-R**, you can use **any DID method**.

Note: DT-R may be presented in a separate session

Decentralized Trust - Registry 3/3

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

Credential Schema defines which Permission Types are allowed


Example Credential Schema Permission Tree
Trust Registry A - Credential Schema Root Permission
permissionType: TRUST_REGISTRY (Root)
did:example:trA
granted schema permission
granted schema permission
Issuer Grantor B - Credential Schema Permission
permissionType: ISSUER_GRANTOR
did:example:igB
Verifier Grantor D - Credential Schema Permission
permissionType: VERIFIER_GRANTOR
did:example:vgD
granted schema permission
granted schema permission
Issuer C - Credential Schema Permission
permissionType: ISSUER
did:example:iC
Verifier E - Credential Schema Permission
permissionType: VERIFIER
did:example:vE

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

Decentralized Trust - Essential Credential Schema

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

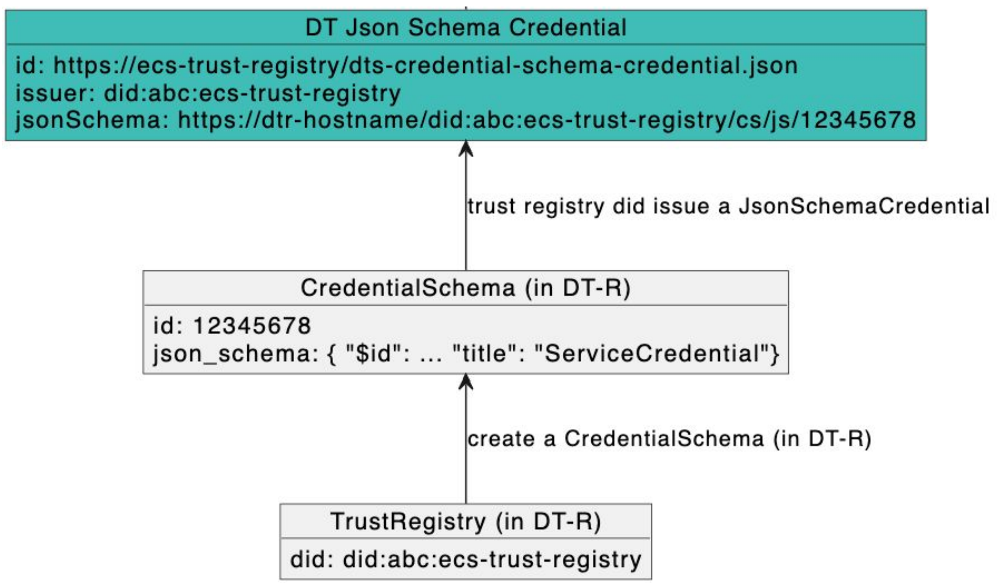
- Service
- Organization
- Person
- UserAgent

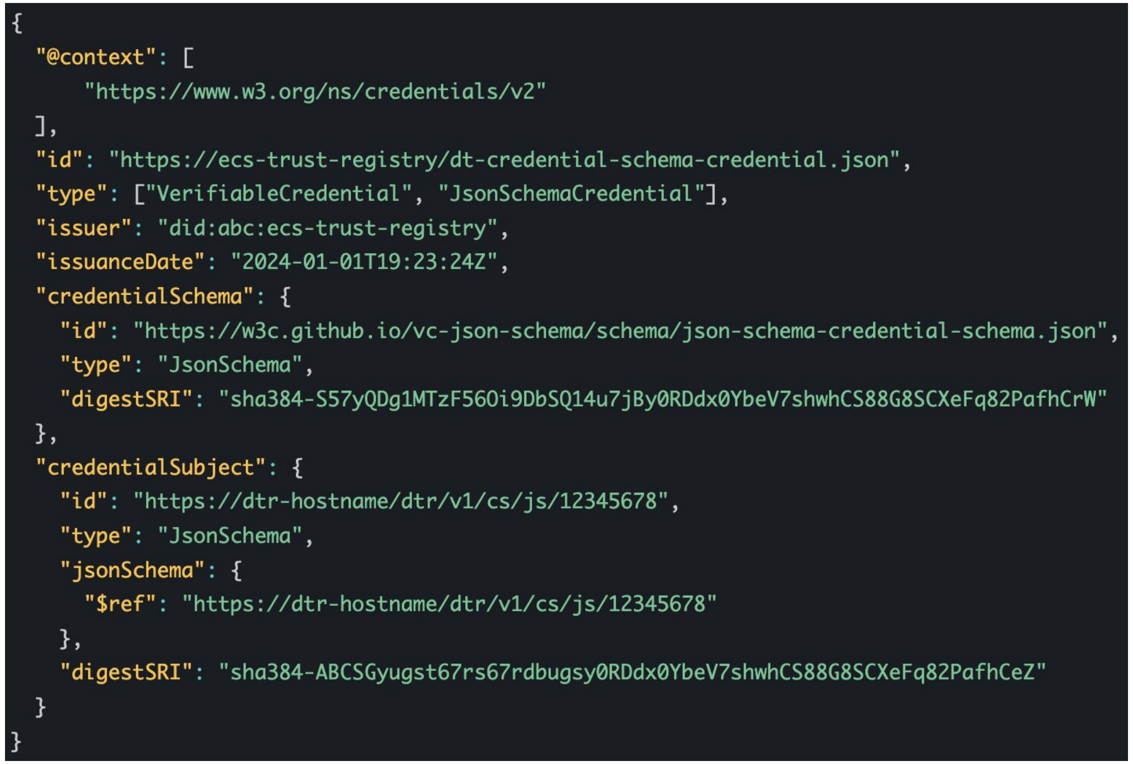
That’s enough to know **who is who** and perform **Trust Resolution**.

Decentralized Trust - Essential Credential Schema

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

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


DT Json Schema Credential
id: https://ecs-trust-registry/dts-credential-schema-credential.json
issuer: did:abc:ecs-trust-registry
jsonSchema: https://dtr-hostname/did:abc:ecs-trust-registry/cs/js/12345678
trust registry did issue a JsonSchemaCredential
CredentialSchema (in DT-R)
id: 12345678
json_schema: { "$id": ... "title": "ServiceCredential"}
create a CredentialSchema (in DT-R)
TrustRegistry (in DT-R)
did: did:abc:ecs-trust-registry

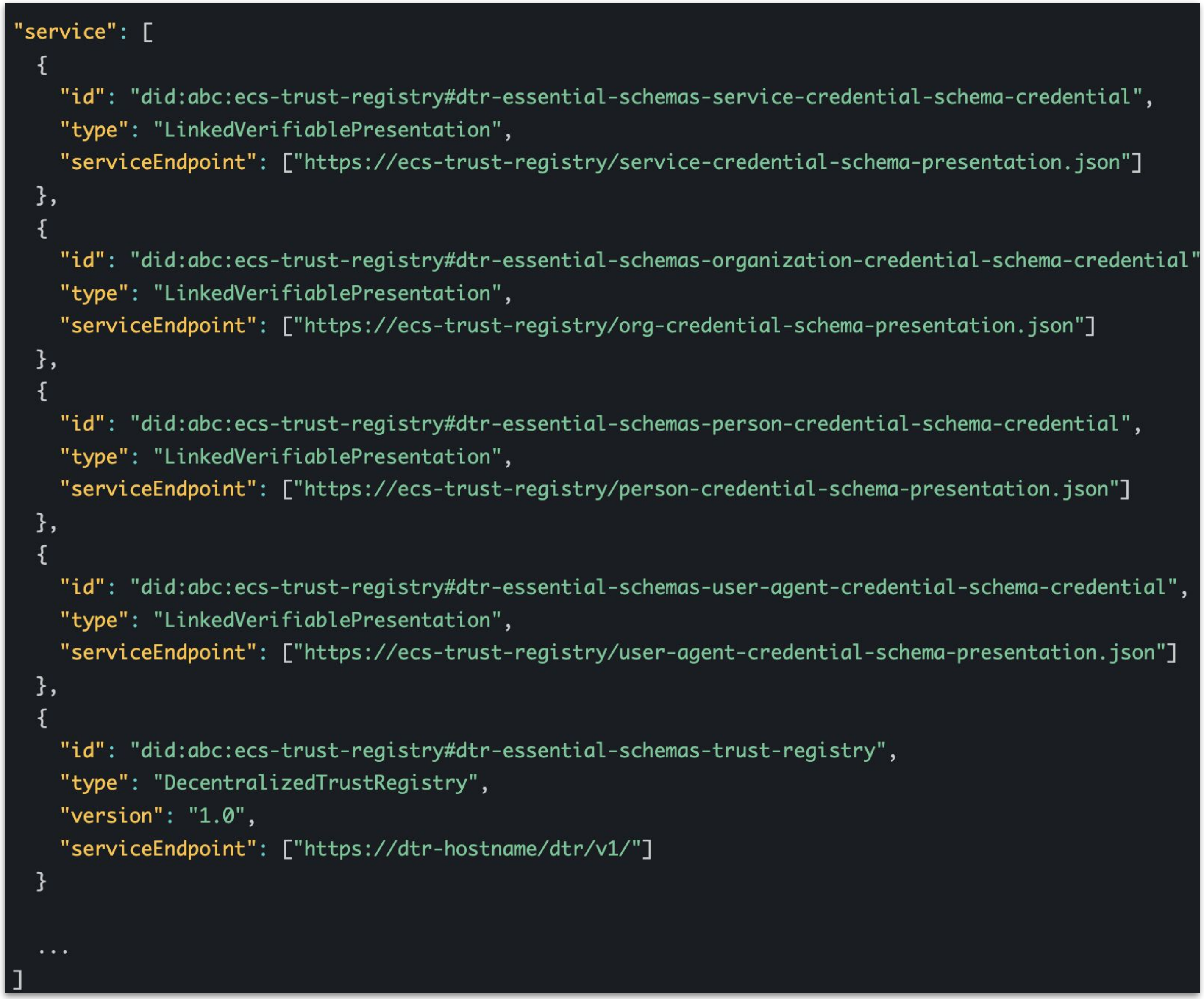


```
{
  "@context": [
      "https://www.w3.org/ns/credentials/v2"
  ],
  "id": "https://ecs-trust-registry/dt-credential-schema-credential.json",
  "type": ["VerifiableCredential", "JsonSchemaCredential"],
  "issuer": "did:abc:ecs-trust-registry",
  "issuanceDate": "2024-01-01T19:23:24Z",
  "credentialSchema":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

Decentralized Trust - 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 **DT-R service entry**.

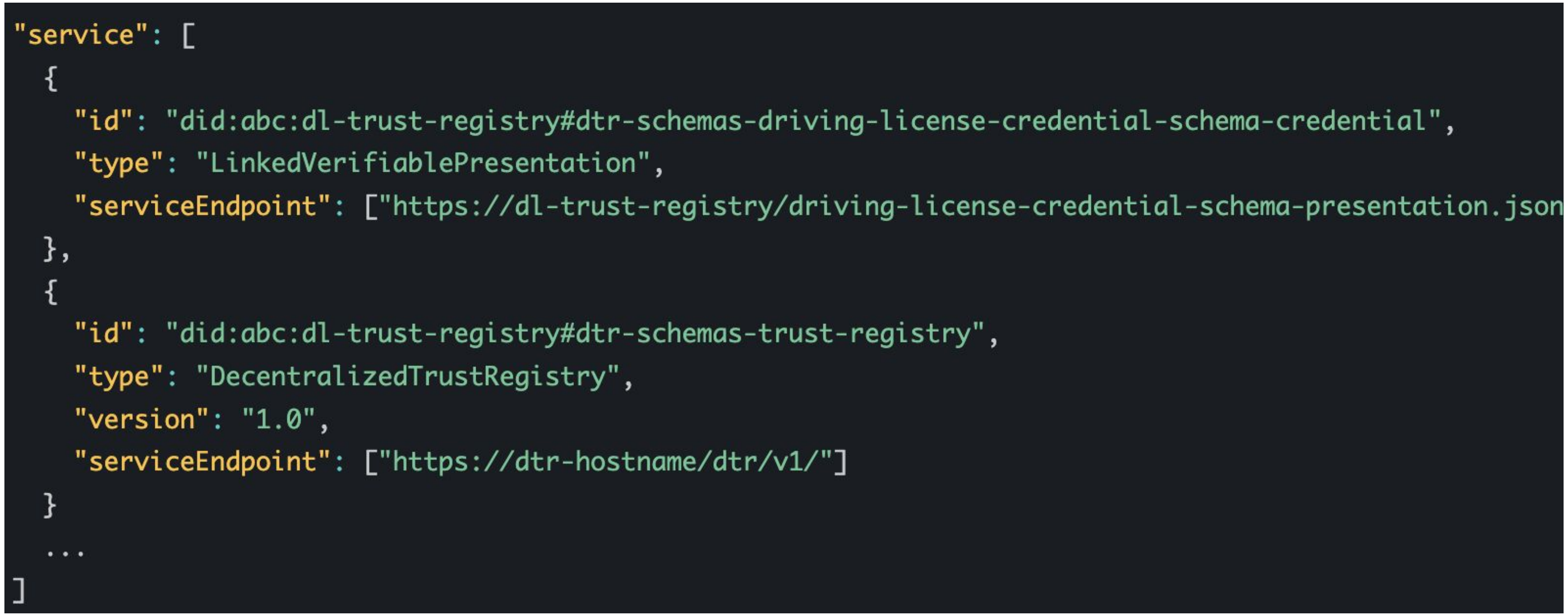


```
"service": [
  {
    "id": "did:abc:ecs-trust-registry#dtr-essential-schemas-service-credential-schema-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/service-credential-schema-presentation.json"]
  },
  {
    "id": "did:abc:ecs-trust-registry#dtr-essential-schemas-organization-credential-schema-credential"
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/org-credential-schema-presentation.json"]
  },
  {
    "id": "did:abc:ecs-trust-registry#dtr-essential-schemas-person-credential-schema-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/person-credential-schema-presentation.json"]
  },
  {
    "id": "did:abc:ecs-trust-registry#dtr-essential-schemas-user-agent-credential-schema-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://ecs-trust-registry/user-agent-credential-schema-presentation.json"]
  },
  {
    "id": "did:abc:ecs-trust-registry#dtr-essential-schemas-trust-registry",
    "type": "DecentralizedTrustRegistry",
    "version": "1.0",
    "serviceEndpoint": ["https://dtr-hostname/dtr/v1/"]
  }

  ...
]
```

Decentralized Trust - Non Essential Schemas

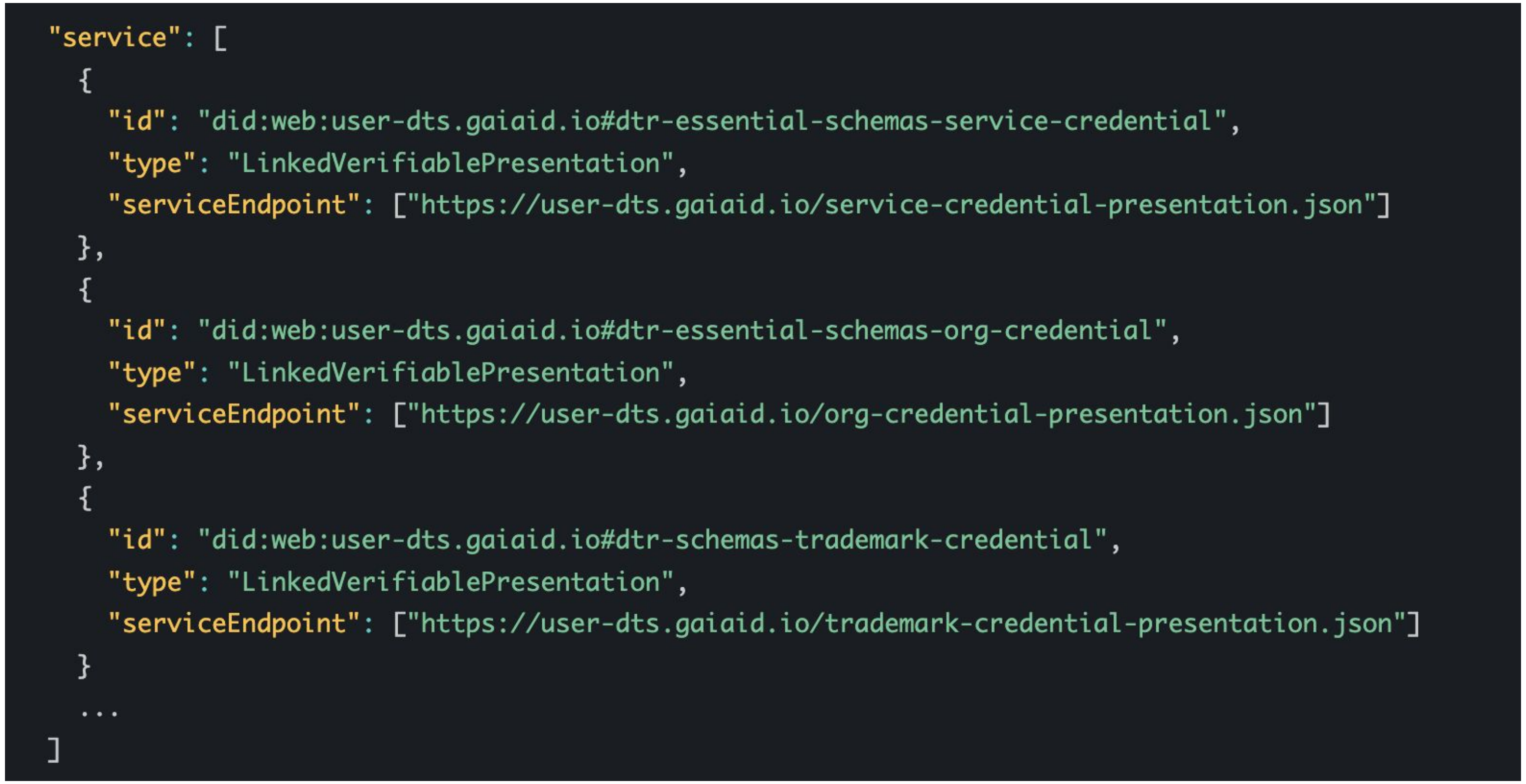
Of course Trust Resolution is not limited to **DT-ECS**. Anyone can create another **Trust Registry**, this one created a schema for DLs:



```
"service": [
  {
    "id": "did:abc:dl-trust-registry#dtr-schemas-driving-license-credential-schema-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://dl-trust-registry/driving-license-credential-schema-presentation.json
  },
  {
    "id": "did:abc:dl-trust-registry#dtr-schemas-trust-registry",
    "type": "DecentralizedTrustRegistry",
    "version": "1.0",
    "serviceEndpoint": ["https://dtr-hostname/dtr/v1/"]
  }
  ...
]
```

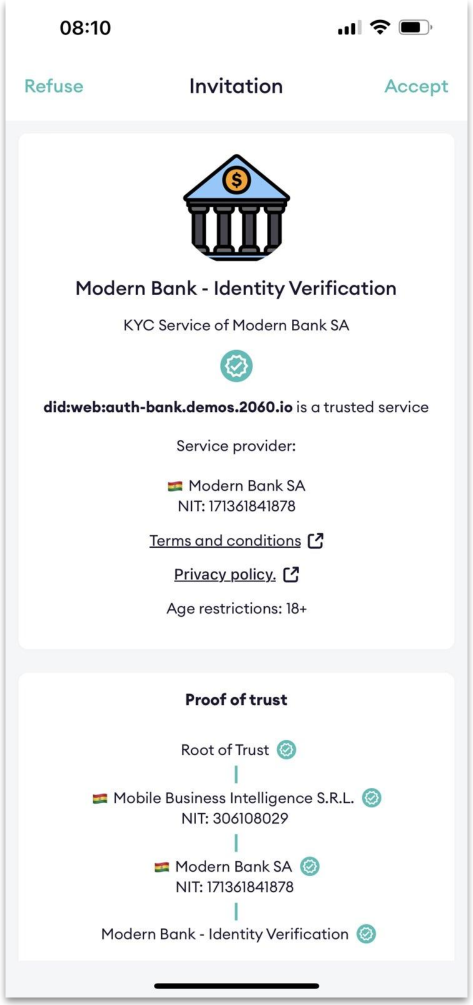
Decentralized Trust - DT Service

Now we have our ECSs, how a DT-S DID Document looks like?



```
"service": [
  {
    "id": "did:web:user-dts.gaiaid.io#dtr-essential-schemas-service-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://user-dts.gaiaid.io/service-credential-presentation.json"]
  },
  {
    "id": "did:web:user-dts.gaiaid.io#dtr-essential-schemas-org-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://user-dts.gaiaid.io/org-credential-presentation.json"]
  },
  {
    "id": "did:web:user-dts.gaiaid.io#dtr-schemas-trademark-credential",
    "type": "LinkedVerifiablePresentation",
    "serviceEndpoint": ["https://user-dts.gaiaid.io/trademark-credential-presentation.json"]
  }
  ...
]
```

Something similar applies for DT-UAs.



Modern Bank - Identity Verification

KYC Service of Modern Bank SA

did:web:auth-bank.demos.2060.io is a trusted service

Service provider:

🇧🇴 Modern Bank SA
NIT: 171361841878

Terms and conditions

Privacy policy.

Age restrictions: 18+

Proof of trust

Root of Trust ✓

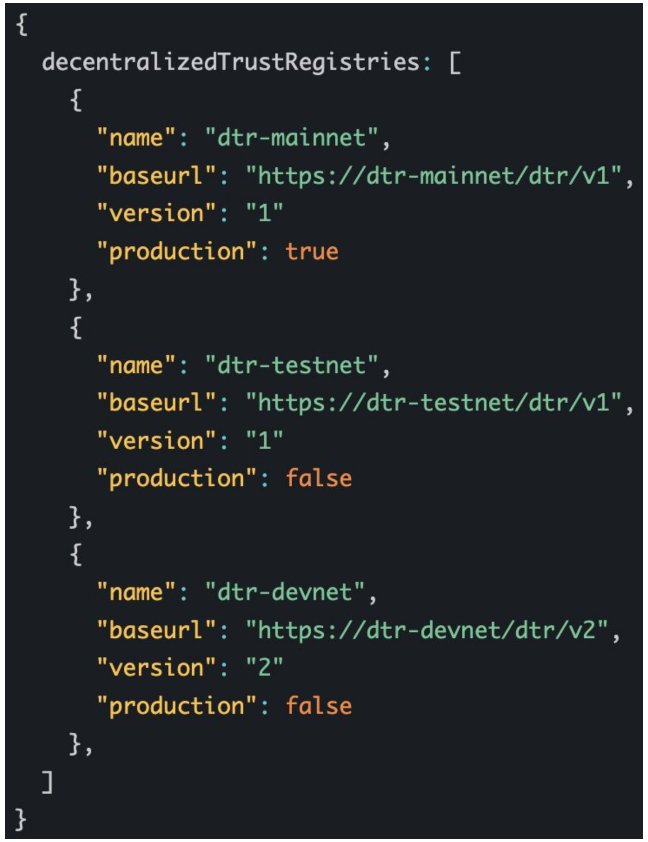
🇧🇴 Mobile Business Intelligence S.R.L. ✓
NIT: 306108029

🇧🇴 Modern Bank SA ✓
NIT: 171361841878

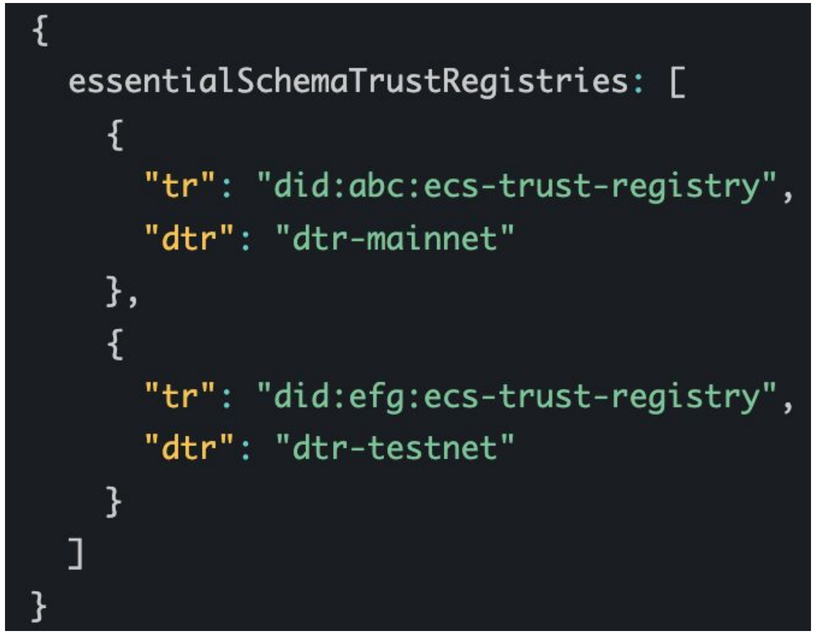
Modern Bank - Identity Verification ✓

Decentralized Trust - Trust Registry lists

Compliant DT-Ss and DT-UAs maintain a list of trusted DT-Rs



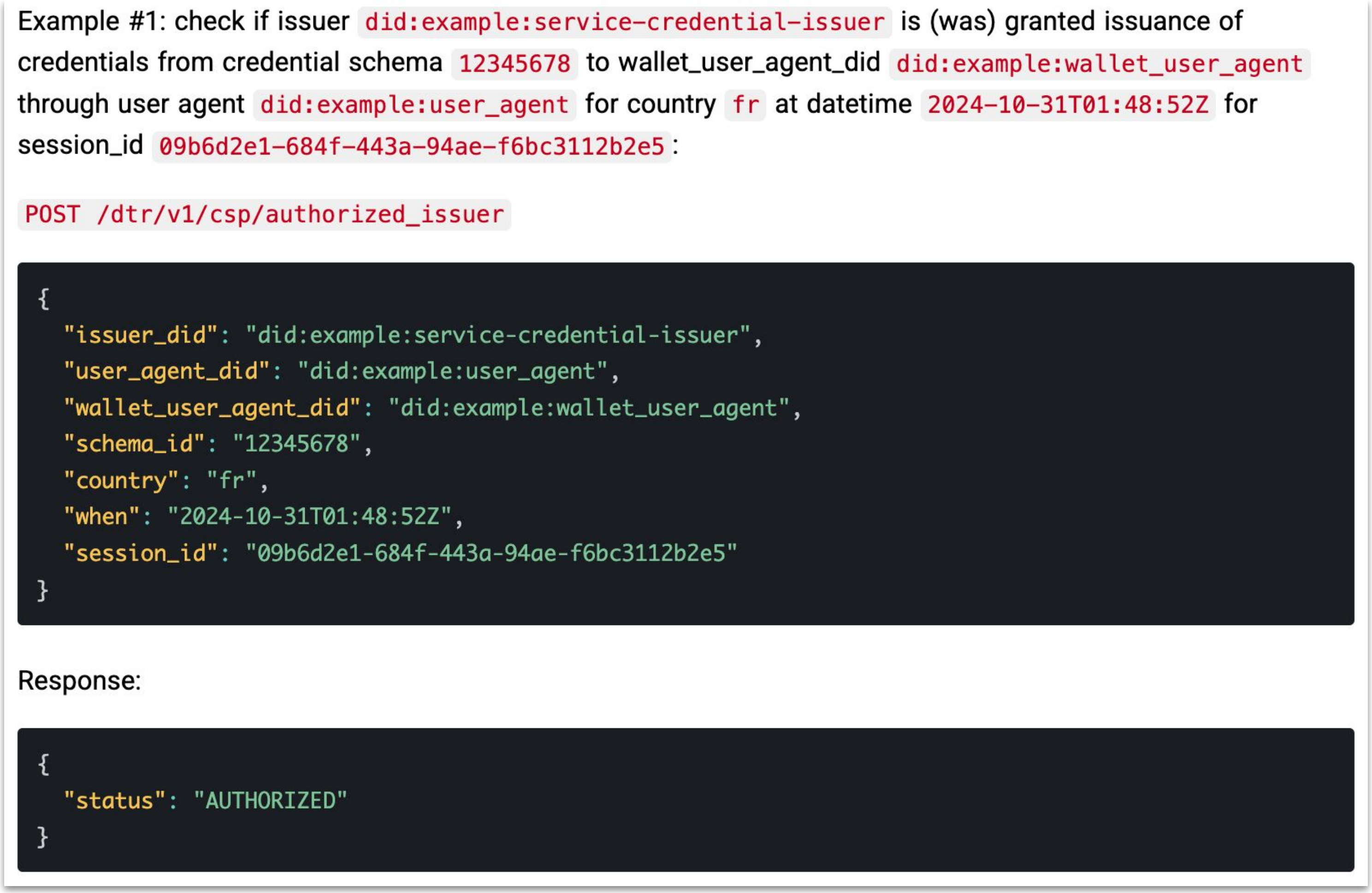
```
{
  decentralizedTrustRegistries: [
    {
      "name": "dtr-mainnet",
      "baseurl": "https://dtr-mainnet/dtr/v1",
      "version": "1"
      "production": true
    },
    {
      "name": "dtr-testnet",
      "baseurl": "https://dtr-testnet/dtr/v1",
      "version": "1"
      "production": false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



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

Decentralized Trust - Trust Resolution

DT-UAs and DT-S query the DT-R to verify authorizations: Issuer



Example #1: check if issuer `did:example:service-credential-issuer` is (was) granted issuance of credentials from credential schema `12345678` to wallet_user_agent_did `did:example:wallet_user_agent` through user agent `did:example:user_agent` for country `fr` at datetime `2024-10-31T01:48:52Z` for session_id `09b6d2e1-684f-443a-94ae-f6bc3112b2e5`:

`POST /dtr/v1/csp/authorized_issuer`

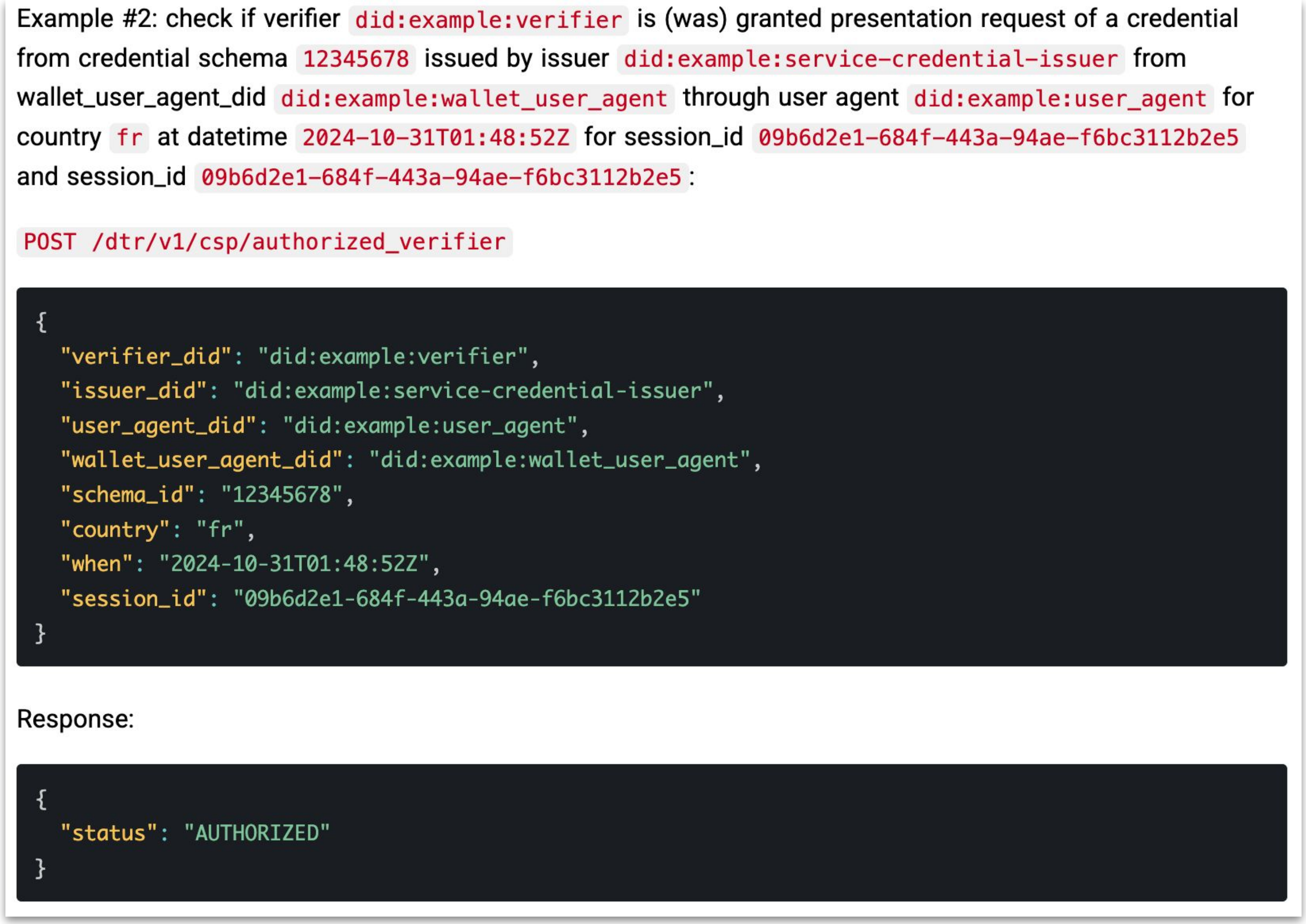
```
{
  "issuer_did": "did:example:service-credential-issuer",
  "user_agent_did": "did:example:user_agent",
  "wallet_user_agent_did": "did:example:wallet_user_agent",
  "schema_id": "12345678",
  "country": "fr",
  "when": "2024-10-31T01:48:52Z",
  "session_id": "09b6d2e1-684f-443a-94ae-f6bc3112b2e5"
}
```

Response:

```
{
  "status": "AUTHORIZED"
}
```

Decentralized Trust - Trust Resolution

DT-UAs and DT-S query the DT-R to verify authorizations: Verifier



Example #2: check if verifier `did:example:verifier` is (was) granted presentation request of a credential from credential schema `12345678` issued by issuer `did:example:service-credential-issuer` from wallet_user_agent_did `did:example:wallet_user_agent` through user agent `did:example:user_agent` for country `fr` at datetime `2024-10-31T01:48:52Z` for session_id `09b6d2e1-684f-443a-94ae-f6bc3112b2e5` and session_id `09b6d2e1-684f-443a-94ae-f6bc3112b2e5`:

`POST /dtr/v1/csp/authorized_verifier`

```
{
  "verifier_did": "did:example:verifier",
  "issuer_did": "did:example:service-credential-issuer",
  "user_agent_did": "did:example:user_agent",
  "wallet_user_agent_did": "did:example:wallet_user_agent",
  "schema_id": "12345678",
  "country": "fr",
  "when": "2024-10-31T01:48:52Z",
  "session_id": "09b6d2e1-684f-443a-94ae-f6bc3112b2e5"
}
```

Response:

```
{
  "status": "AUTHORIZED"
}
```

Decentralized Trust - Spec

Contributions? Discussions?

https://github.com/verana-labs/decentralized-trust-spec

2060

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