

# OpenID Federation 1.0

Shaping The Advanced Infrastructure of Trust

Giuseppe De Marco <demarcog83@gmail.com>

Mike Jones <michael\_b\_jones@hotmail.com>

John Bradley, Yubico <jbradley@yubico.com>

Roland Hedberg <roland@catalogix.se>

TODAY WE TALK ABOUT OPENID FEDERATION 1.0 WITH ...

- I'm **Giuseppe De Marco**,
- I work as **Open Source Project Leader** and **Digital Identity Expert** for the Department for digital transformation of the Italian Government,
- I have concrete experiences with LDAP, SAML2, OAuth 2.0 and OpenID Connect as **Software Developer**.
- I have designed the Italian infrastructure of trust for OpenID Connect based on OpenID Federation 1.0, with colleagues of other Italian institutions (AgID, IPZS) and contributing to IETF and OpenID specifications.
- I love to work with Roland Hedberg, Mike Jones, Vladimir Dzhuvinov, and John Bradley!



# The Technology to Build

- An **Infrastructure of Trust:**
  - **Public** and **Transparent**;
  - **Hierarchical** and **Decentralized** ... That scales!
- Verifiable **Digital Trust Relationships** between the participants of a digital ecosystem:
  - not repudiable over time (even when cryptographic keys changes!).

WE ESTABLISH THE TRUST WITH SIMPLE RULES

## Equation for a Relationship

PROOF of **IDENTITY**

+

PROOF of **COMPLIANCE** to requirements

=

**TRUST**

Periodically renewed,  
like human relationships!



## We Need to Periodically Evaluate

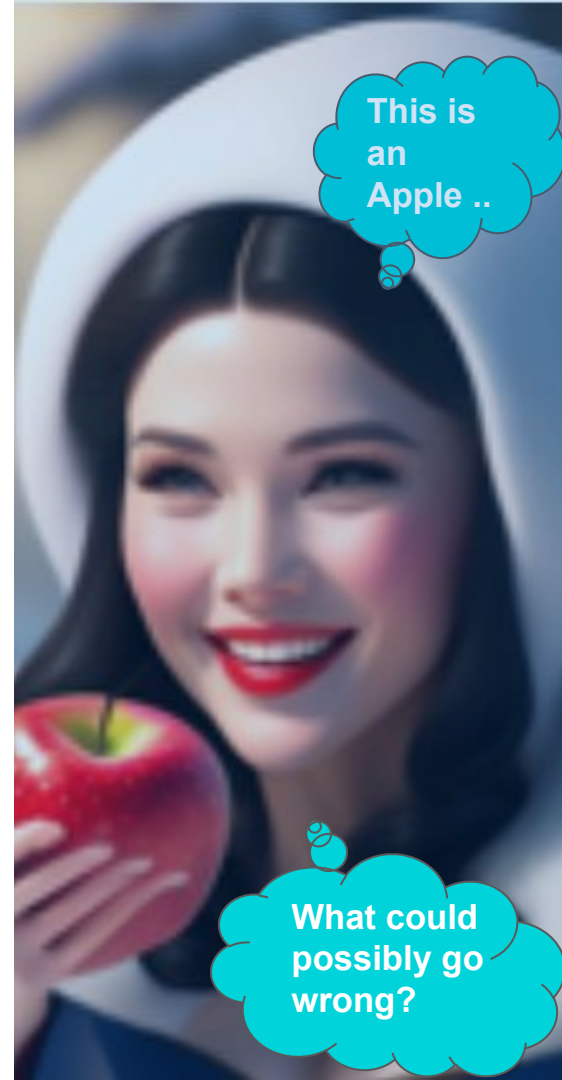
The party which we are interacting with

IDENTITY

The shared terms, conditions and rules.

COMPLIANCE

Identity, alone **(DNS+TLS)**, is not enough.



WHAT SHOULDN'T HAPPEN THAT OFTEN HAPPENS

# What If Something Goes Wrong? What if Someone Cheats?

Requirement of automated policy.  
Policy application **MUST** be enforced in a coercive manner.

HOW to do automatically this?  
OpenID Federation has a Policy Language.





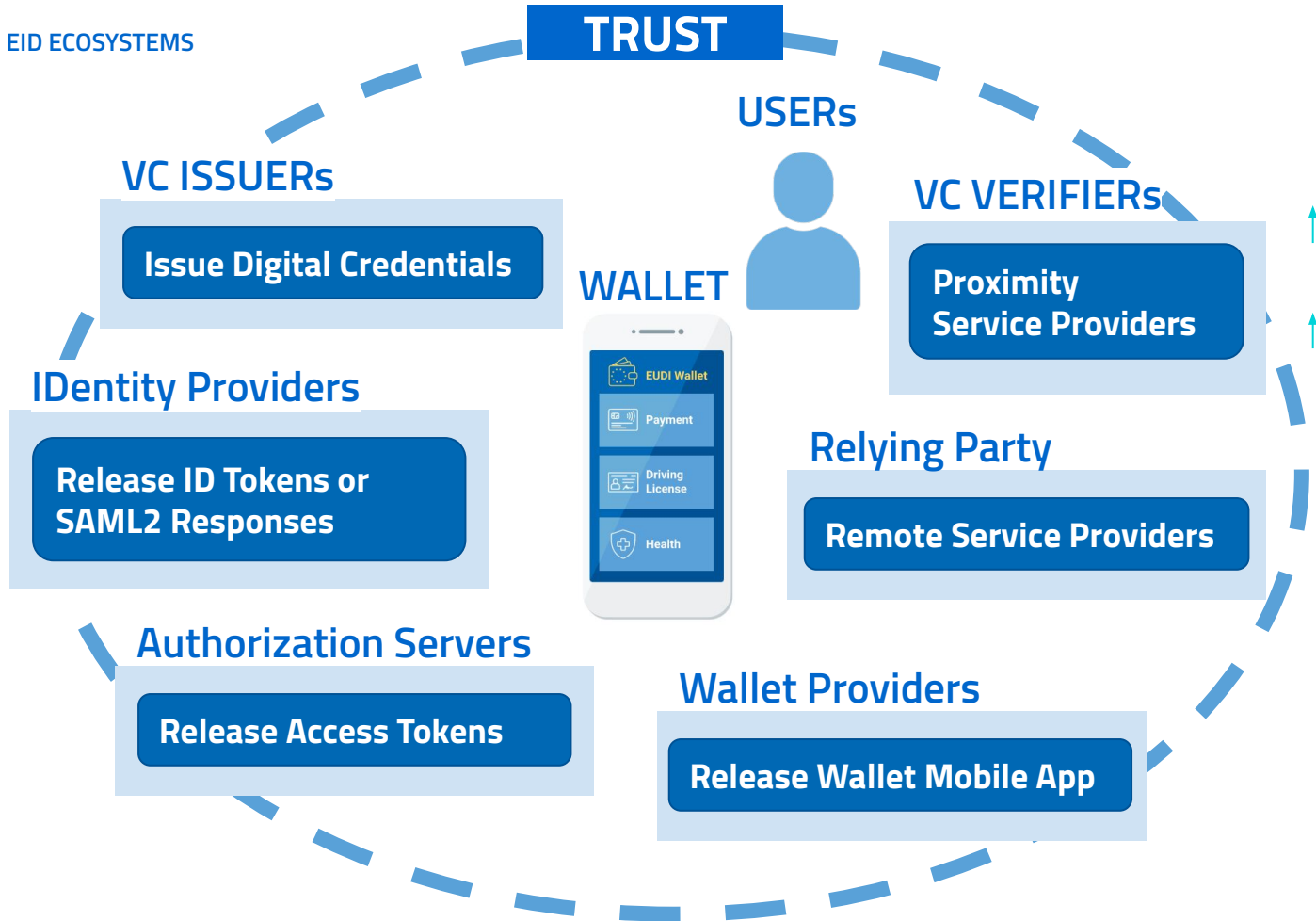
THE NEVER ENDING LIST OF WHAT SHOULDN'T HAPPEN THAT OFTEN HAPPENS

## Imagine an Ecosystem without automatic Policies evaluations ...

- The compromised/malicious **RP requests User personal information through social engineering.** Fake promises, fake prizes, phishing...
- The **bogus Issuer** issues Credentials for which it is **not authorized to issue**
- An RP would not know which Credential Issuer is eligible for issuing determined Credentials
- An RP would **obtain data from under-age Users without grants** for that
- A **fake Accreditation Body** would carry out onboarding systems and inspections without any authorization
- The coffee machine that requires personal data ... Do we have enough?

**Never tickle your adversary's imagination!**

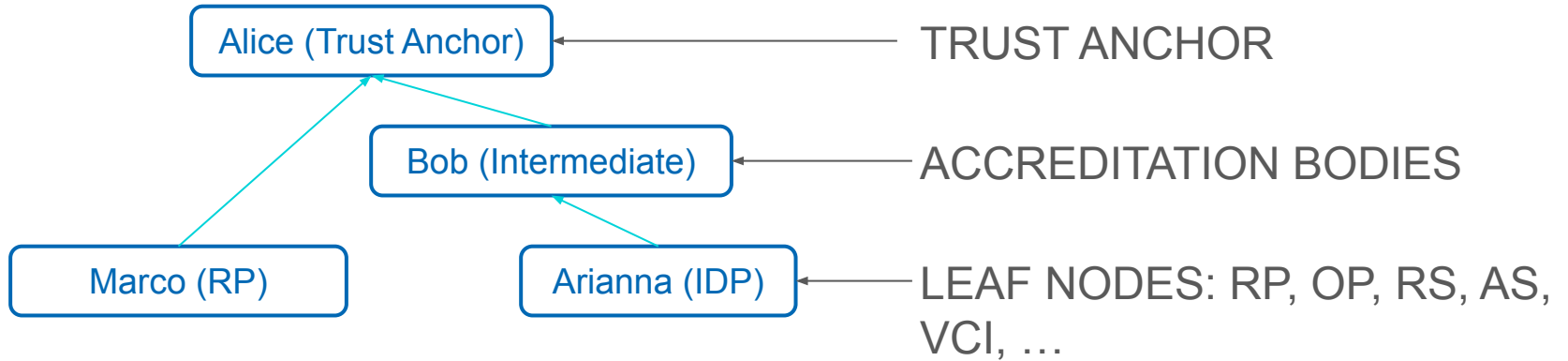
ROLES IN THE EID ECOSYSTEMS





## TRANSITIVE TRUST

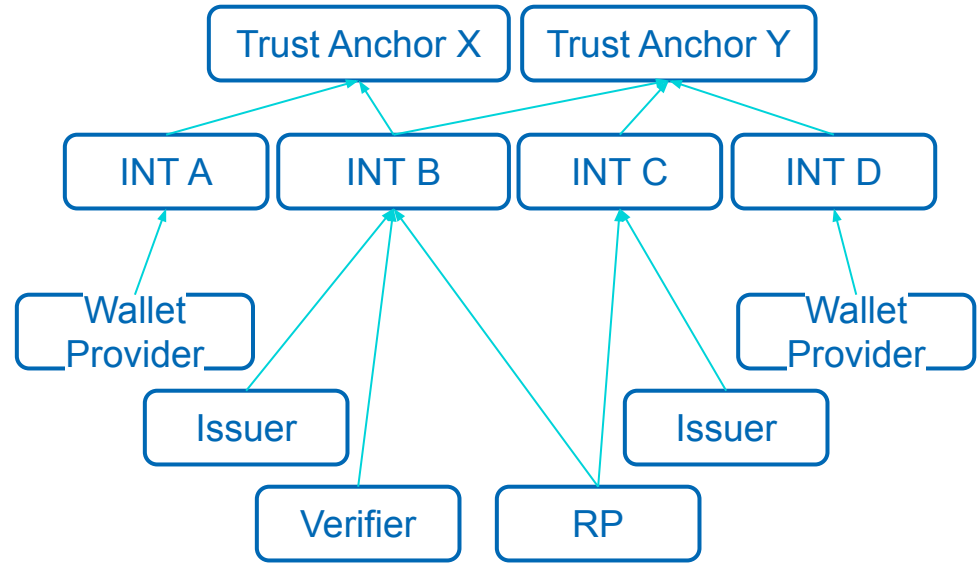
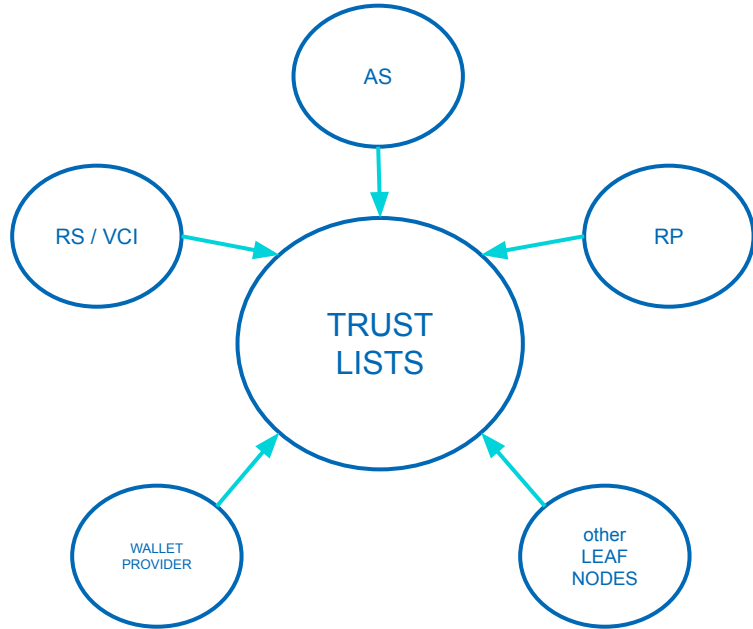
# How to go from Marco to Bob?



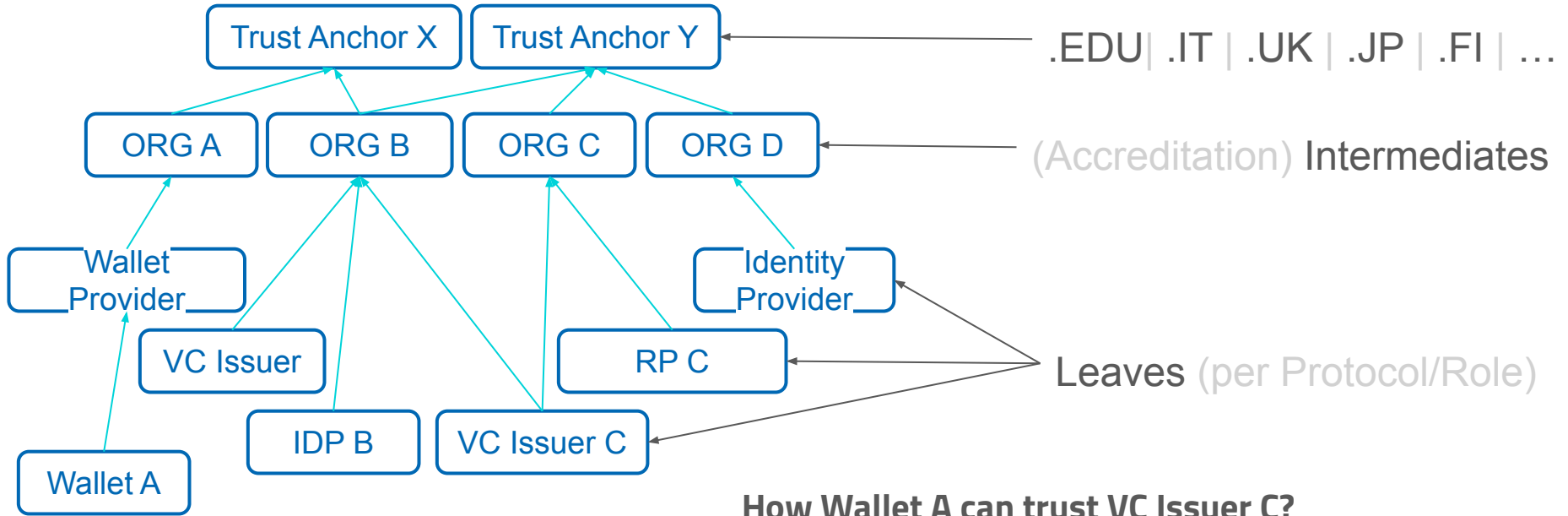
Problems to solve:

1. How to link entities that don't have any direct link?
2. How to establish if Marco and Alice comply with the same rules?
3. How to ensure that Marco and Bob apply Alice's rules?

# Flat vs. Decentralized and Hierarchical



# Real World Example With Domains



**How Wallet A can trust VC Issuer C?**

**How the RP C can trust IDP B?**

**-> They shares common rules under a Trust Anchor**

## Trust that scales

In small-scale and static deployments, it's possible to keep **a list of the trusted participants** and their metadata

However, **in large-scale and dynamic deployments, that doesn't scale**

This needs **multilateral relationships**, rather than **bilateral**



TODAY WE TALK ABOUT OPENID FEDERATION 1.0 WITH ...

- I'm **Roland Hedberg**,
- Author of `pysaml2`, `SATOSA`, `pyoidc`, `idpy-oidc`, `fedservice`, `idpy-sdjwt`, `openid4v`, `satosa-openid4vci`, ....
- Started with X.500 1988. Since then mainly done LDAP, SAML2 and OIDC/OAuth2.
- Authored the OpenID Connect certification software.



## How it all began ...

First official occurrence - IIW 21 October 2015

- Allow for semi-static client registration
- Allow for off-line verification of 'documents' that are sent around.

First presentation, TNC16, Prague

- Trust fabric
- Endforced metadata policies
- Independence of transport security
- New member without bringing in the FO



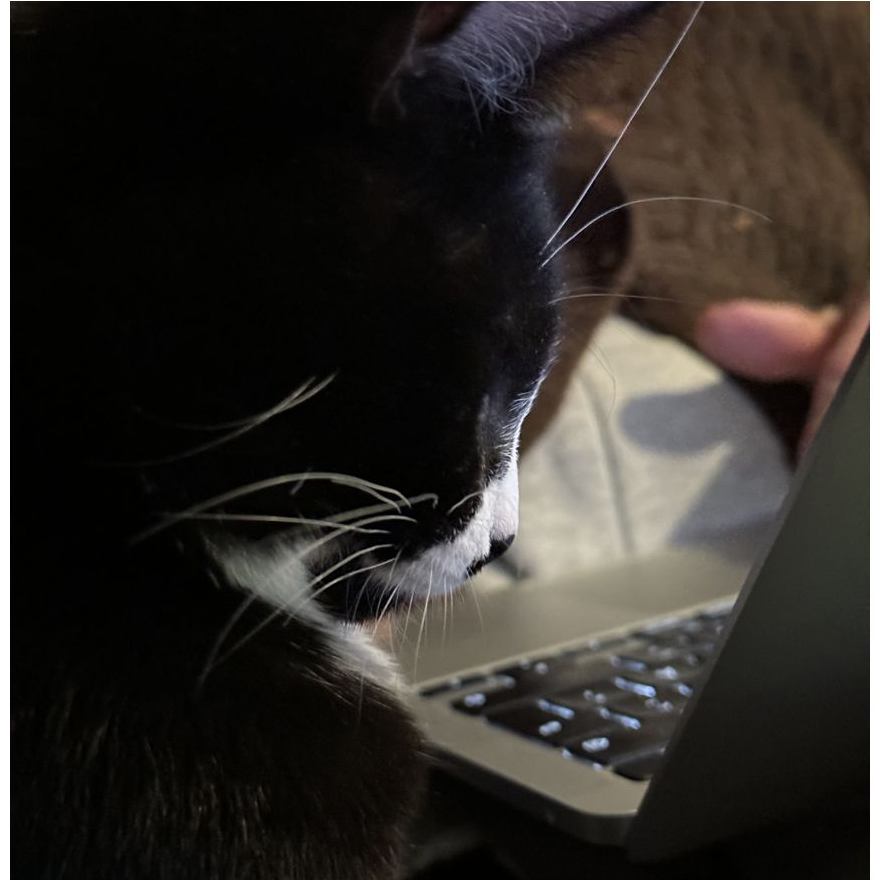
# What you want a federation to tell you as a member

- If another entity is a member of the same federation as you
- What type of entity an entity is
- An entity's metadata for the application protocol you want to use
- If another entity is special in any way



# What you as a federation operator wants from the federation

- Be able to define metadata policy
- Be able to set the basic layout of the federation
- Be able to say who can say what about whom






# Collecting a Trust Chain

## Step 1

- Fetch the Entity Configuration of the entity
  - The URL used is based on the entity's `entity_id`
    - `<entity_id>/well-known/openid-federation`
- Before this step the entity has gotten hold of the TA's public federation keys in some unspecified way.

Entity Configuration

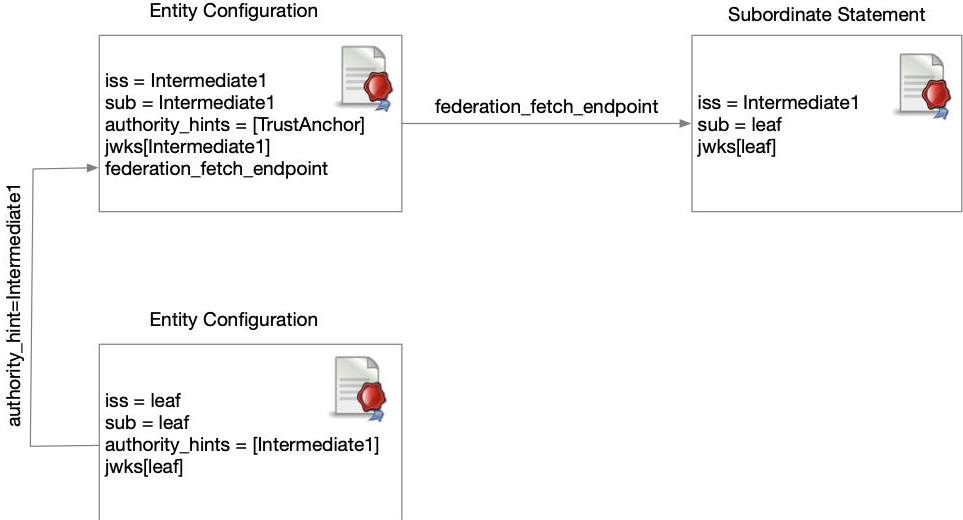
```
iss = leaf
sub = leaf
authority_hints = [Intermediate1]
jwks[leaf]
```

A small icon of a document with a red seal or stamp on it, located to the right of the text.

# Collecting a Trust Chain

## Step 2

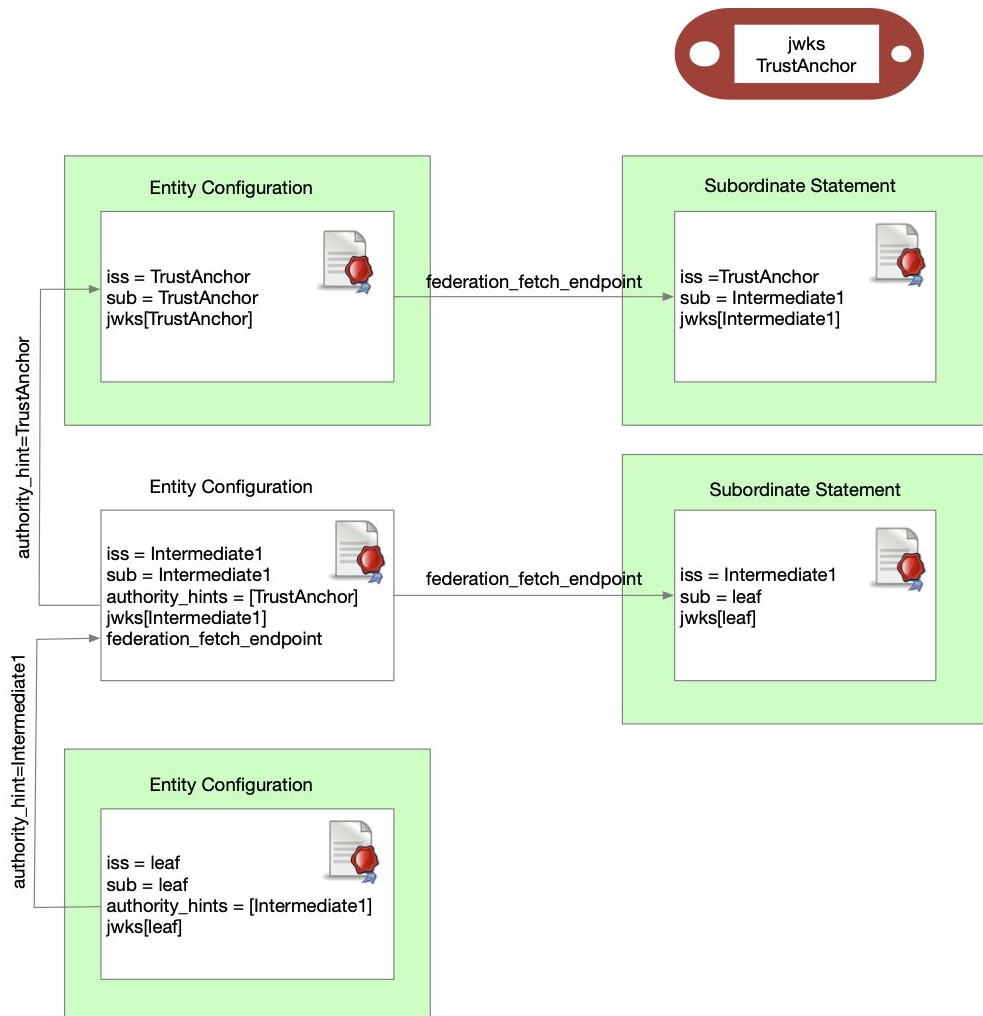
- 1. Pick one authority hint and fetch that entity's Entity Configuration
- 2. In the authority's Entity Configuration find the federation\_fetch\_endpoint URL
- 3. Request the authority's view of the subordinate



# Collecting a Trust Chain

## Step 2+

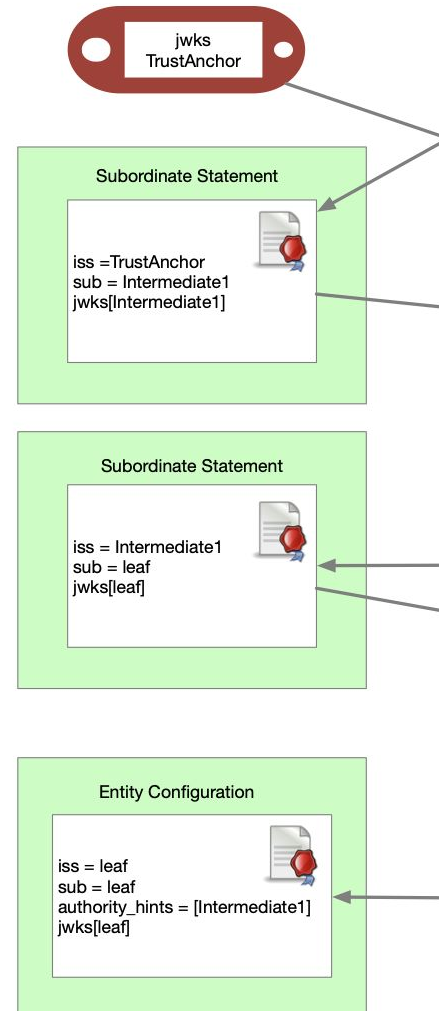
*Repeat until you have reach a Trust Anchor you want to use.*



# The Trust Chain

## Verifying correctness

- Start with the top most message and verify the signature on that using the TA's signing keys.
- If that verifies OK use the JWKS in the statement to verify the next message in the chain.
- Repeat until you have reached the end of the chain and the signature of the Entity Configuration has been verified



# The Trust Chain


## Applying policies

- An entity publishes its metadata in its Entity Configuration.
- Every superior can specify
  - **metadata** (to lock down specific values on specific claims)
  - **metadata\_policy** (to specify restrictions on which values can be used)
  - **constraints** (set limitation on the tree structure)

Entity Configuration (TA)

```
iss = TrustAnchor
sub = TrustAnchor
jwks[TrustAnchor]


metadata
```



Subordinate Statement

```
iss = TrustAnchor
sub = Intermediate1
jwks[Intermediate1]


metadata
metadata_policy
constraints
```



Subordinate Statement

```
iss = Intermediate1
sub = leaf
jwks[leaf]


metadata
metadata_policy
constraints
```



Entity Configuration

```
iss = leaf
sub = leaf
authority_hints = [Intermediate1]
jwks[leaf]

metadata
```



# Entity Configuration Example

```
{
  "iss": "https://op.umu.se",
  "sub": "https://op.umu.se",
  "exp": 1568397247,
  "iat": 1568310847,
  "metadata": {
    "federation_entity": {
      "federation_fetch_endpoint": "https://example.com/federation_fetch",
      "organization_name": "The example cooperation",
      "homepage_uri": "https://www.example.com"
    },
    "openid_provider": {
      "issuer": "https://op.umu.se/openid",
      "signed_jwks_uri": "https://op.umu.se/openid/signed_jwks.jose",
      "authorization_endpoint": "https://op.umu.se/openid/authorization",
      "token_endpoint": "https://op.umu.se/openid/token",
      "federation_registration_endpoint": "https://op.umu.se/openid/fedreg",
      "token_endpoint_auth_methods_supported": ["private_key_jwt"],
      "pushed_authorization_request_endpoint": "https://op.umu.se/openid/par"
    }
  },
  "authority_hints": [
    "https://umu.se"
  ],
  "jwks": {
    "keys": [{
      "e": "AQAB",
      "kid": "dEeTRj1zY3djcENuT01wOGxrZlkb3RIQVJlMTY0...",
      "kty": "RSA",
      "n": "x97YKqc9Cs-DNtFrQ7_vhXoH9bwkDWW6En2jJ044yH..."
    }]
  }
}
```

# metadata\_policy

openid\_provider

- value
- add
- default
- essential
- one\_of
- superset\_of
- subset\_of

```
{
  "grant_types": {
    "subset_of": [
      "authorization_code",
      "refresh_token",
      "implicit"
    ],
    "default": [
      "authorization_code",
      "refresh_token"
    ]
  },
  "token_endpoint_auth_method": {
    "one_of": [
      "client_secret_post",
      "client_secret_basic"
    ],
    "default": "client_secret_basic"
  },
  "contacts": {
    "add": "helpdesk@org.example.org"
  }
}
```

# Constraints

- `max_path_length`
  - The maximum number of Entity Statements between this Entity Statement and the last Entity Statement in the Trust Chain (the leaf Entity Configuration).
- `naming_constraints`
  - about `entity_ids`
- `allowed_leaf_entity_types`

```
{  
  "naming_constraints": {  
    "permitted": [  
      "https://.example.com"  
    ],  
    "excluded": [  
      "https://east.example.com"  
    ]  
  },  
  "max_path_length": 2,  
  "allowed_leaf_entity_types": [  
    "openid_provider",  
    "openid_relying_party"  
  ]  
}
```



# Trust Marks

- Expresses that someone has asserted something about someone.
- Note that a trust mark issuer may issue trust marks on delegation from the owner of the trust mark.



```
{  
  "id": "https://deleghedigitali.gov.it/openid_relying_party/sgd/",  
  "iss": "https://deleghedigitali.gov.it",  
  "sub": "https://rp.cie.id",  
  "iat": 1579621160,  
  "ref": "https://deleghedigitali.gov.it/documentation/manuale_operativo.pdf",  
  "logo_uri": "https://deleghedigitali.gov.it/sgd-cmyk-150dpi-90mm.svg",  
  "organization_type": "public",  
  "id_code": "123456",  
  "email": "info@rp.cie.id",  
  "organization_name#it": "Nome dell'organizzazzione",  
  "policy_uri#it": "https://rp.cie.id/privacy_policy",  
  "tos_uri#it": "https://rp.cie.id/info_policy",  
  "service_documentation": "https://rp.cie.id/api/v1/get/services"  
}
```

# Federation Endpoints

- **Fetch**
  - To get Entity Statements
- **Resolve**
  - Offloads collecting trust chains, verifying them and applying policies to the metadata
- **List**
  - Lists immediate subordinates
- **Trust mark**
  - Allows an entity to ask for a trust mark.
- **Trust mark status**
  - Returns the status of a trust mark. Whether it is active or not.
- **Trust marked entities listing**
  - Lists all entities that has received a trust mark of a special kind from this issuer.
- **Historical keys**
  - Lists all the keys that an entity has used.

## TODAY WE TALK ABOUT OPENID FEDERATION 1.0 WITH ...

- I'm **Mike Jones**
- An independent consultant in identity and security space
  - <https://self-issued.consulting/>
- Mission: Building the Internet's missing identity layer
- An author of many specs that actually get used, including OpenID Connect, JSON Web Token (JWT), multiple OAuth specs, WebAuthn/FIDO2
- Working on adding OpenID Federation to that set!

## REQUIREMENTS SATISFIED

1. Are they **who they say they are?** (ENTITY IDENTIFIERS)
2. Are they compliant with the **security and privacy requirements** required by the ecosystem (e.g., EUDI Wallet)? (COMPLIANCE)
3. Superiors sign statements about Subordinates in **Trust Chain** (KEY HIERARCHY)
4. Authorization to carry out intended actions (POLICIES)
  - a. in both **online** & **offline** interactions (ENCAPSULATED TRUST)
  - b. and **verifiable in the future** using retained historical keys (NON-REPUDIATION)
5. Use with any AuthN/AuthZ technology (PROTOCOL AGNOSTIC)



# OpenID Federation Functional Features

## Flexibility

- **Support for protocol-specific metadata**, policies, Trust Marks, and ecosystem roles. Tailored configurations meet specific needs, enhancing adaptability in diverse environments.
- **Multiple federations.** Entities can participate in multiple federations with different Trust Anchors.
- **Multiple roles:** Can a Trust Anchor in one Federation be an intermediate in another? With OpenID Federation YES.

## Transparency, online access to all Trust relationships:

- Federation API makes Trust relationships transparent and consultable online by providing a standardized way to share and access trust information, enabling **real-time navigation of the federation**

## Dynamic Relationships among Entities

- Adding entities to and removing entities from organizations and federations is a local operation
- Only requires superior to issue an updated Subordinate Statement about entity (or to stop issuing it)
- Short entity statement timeouts can enable timely addition, revocation, and metadata updates

## Constraints ensure compliance with policies

- Can prevent malicious parties and misconfigurations from performing actions they're ineligible for

## Offline Usage enabled through cached trust chains

- `trust_chain`JWS header parameter can occur within requests, responses, attestations, and digital credentials

## TODAY WE TALK ABOUT OPENID FEDERATION 1.0 WITH

- I'm **John Bradley**
- Sr. Distinguished Architect at Yubico
- A co-author of many specs, including OpenID Connect, JSON Web Token (JWT), multiple OAuth specs, WebAuthn/FIDO2
- I started in 1987 with x.25, LDAP and Certificate Authorities in 1995. Moving to SAML, InfoCard, and OpenID in 2005



# Can We do all these Things with an X.509 PKI?

- Multilateral Federation Multiple roots of trust for a single entity?
  - Requires PKI Bridges and cross signing.
  - Support for validation requires special software.
- Delegation?
  - In a limited way, but not in use in actual software.
- Revocation?
  - Requires CRL or OCSP (doesn't scale)

# X.509 vs. OpenID Federation

	X.509	OpenID Federation
<b>Born</b>	1988	2016
<b>Format</b>	ASN.1/DER	JWT
<b>RESTful API</b>	–	Yes
<b>Revocations</b>	CRL, OCSP	Using the RESTful API
<b>Attestation artifact</b>	Public key certificate	Entity Statement (ES)
<b>Attestation content</b>	1 Identity (FQDN) 1 Public Key	1 Identity (FQDN) N Public Keys Protocol specific capabilities Grants
<b>Chain name</b>	Certificate chain	Trust Chain
<b>Chain payload</b>	Identity, Public key, constraints, custom extensions.	Identity, Public key <sup>S</sup> , constraints, <b>protocol specific metadata, Trust Marks, policies, X.509 certificates.</b>
<b>Cryptographic Keys specialized per scope</b>	–	Federation Keys != Protocol specific keys



Concluding

# MIKE

RECAP, VISION, MILESTONES, STATUS

## Frequently Asked Questions

### What is the status of OpenID Federation?

- Core components have been stable since draft 20 (now at draft 32). Useful (optional) backwards-compatible features added. Last Implementer's Draft is imminent. Final specification anticipated for later this year.

### Testing and Certification

- Test plans critical to mitigate the risk of misconfigurations and bugs in large-scale deployments. In Italy, test plans were applied to the production deployments.
- General-purpose federation test plans and certification criteria by OpenID Foundation planned for mid-year.

### Is OpenID Federation a PKI?

- It's more than a PKI, incorporating dynamic multilateral trust establishment and policy evaluation mechanisms.

### Availability of Libraries and Support

- **Java Nimbus OAuth 2.0 SDK**. Italian implementations in: **Java, Python, JS, PHP, .NET** ([github.com/italia](https://github.com/italia)). Roland Hedberg's Python impl (<https://github.com/rohe/fedservice>). Ongoing work in the Research and Scholarship community, using GoLang. All Open Source.

### What are the incentives for supporting both X.509 and OpenID Federation?

- X.509 certificates can be used for their traditional purpose of establishing a basic level of trust based on the certificate chain, while OpenID Federation can handle the more advanced trust management tasks, using Trust Marks and automatic policy processing. The Federation API can be used to check the revocation of X.509 certificates within Federation Trust Chains, without requiring OCSP and CRL lists.

### Can Federation be used with ISO/IEC 18013-5?

- A federation Trust Chain can be added to the COSE\_Sign1 Document (MSO). [This paves the way for standardization](#). The Federation API can provide an equivalence of the "Verified Issuer Certificate Authority List" (VICAL).

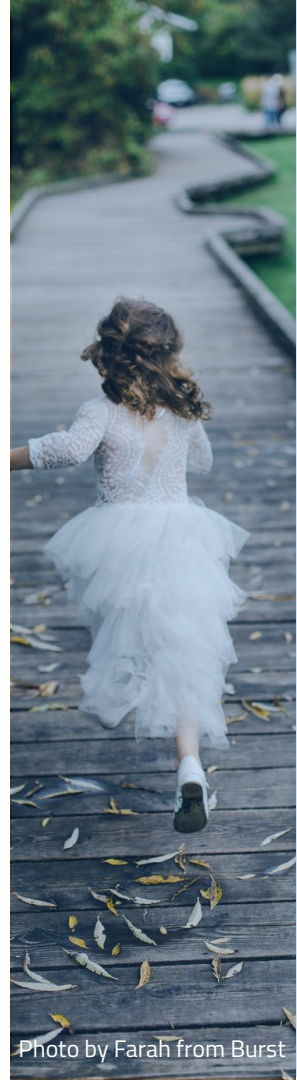


Photo by Farah from Burst

# Thank You For Your Attention!

For further clarifications, ideas, proposals, or discussions, contact us

- Our e-mail addresses are on the first slide

Other recent OpenID Federation presentations:

- Discover the Italian Federation, [TNC 2022](#) by Roland Hedberg
- How Do You Know Who to Trust?, [EIC 2023](#) by Mike Jones
- Do you Trust the Wallet?, [FBK Trust and Identity Summit](#) by Francesco Marino (IPZS)
- The Key is Not Enough, [OAuth Security Workshop 2023](#) by Vladimir Dzhubinov
- eIDAS Expert Group - [EUDI Wallet with OpenID Federation 1.0](#)

