OpenID Federation Overview

Michael B. Jones

Self-Issued Consulting

December 11, 2025

Structure of Proposed Discussion

- Quick Overview of OpenID Federation Background and Goals
- Dive into Spec Features Achieving Those Goals
- Next Steps

- *By all means, please interact and discuss*

OpenID Federation

- OpenID Federation Specification
  - https://openid.net/specs/openid-federation-1_0.html
  - Enables establishment and maintenance of multi-lateral federations
- Incorporates lessons learned from SAML-based federations
  - Defines hierarchical JSON-based metadata structures for federation participants
- Entities can be in multiple federations
- Federations can be in federations
- In 60-day review for Final Status
  - If you have issues, file them now
  - https://github.com/openid/federation/issues

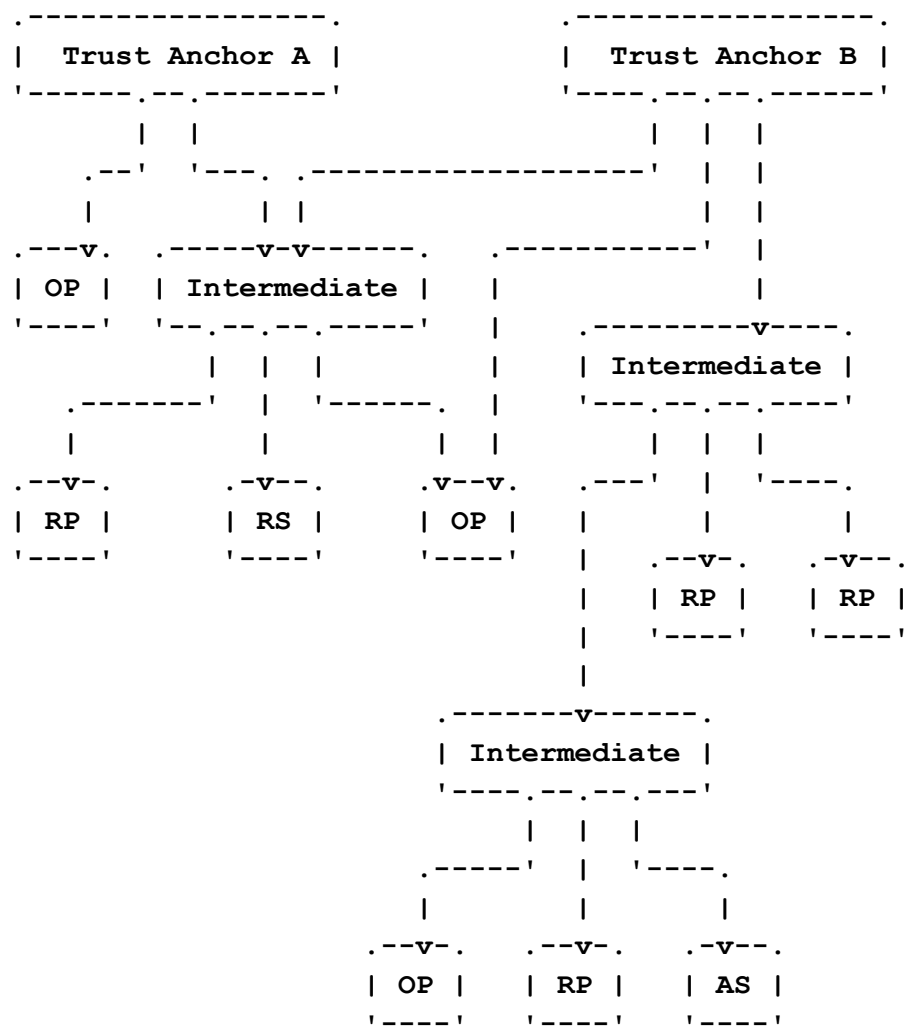
OpenID Federation Interest in R&E Sector

- CACTI *Profiling OpenID Federation for R&E (PORE)* WG
  - https://spaces.at.internet2.edu/spaces/NCTFWG/pages/303104442/Profiling+OpenID+Federation+for+Research+and+Education+Working+Group+Home
- eduGAIN OpenID Federation pilot
- Many 2025 TechEx presentations mention OpenID Federation
- Many dual-stack implementations of OpenID Connect & SAML

Establishing Trust within a Federation

- How do a Relying Party and an Identity Provider know that they're in the same federation?
  - Important for trust, liability, accountability, and reliability
- Shibboleth/SAML approach:
  - Federation Operator polls participants for their metadata, concatenates it into a huge flat file, and distributes it to all nightly *(written in 2020)*
  - In production use, but brittle and not scalable
    - SAML world developing Metadata Query protocol to try to move away from this
- OpenID Federation approach:
  - Hierarchical metadata, where organizations publish metadata about themselves and Federation Operators publish statements about orgs
  - Scalable, maintainable

Two Federations with Some Members in Common

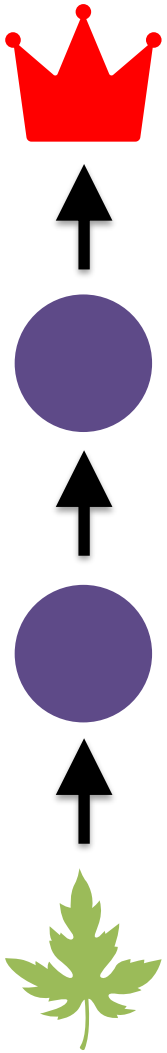


Use of Hierarchical Metadata

- Each leaf member publishes self-signed metadata about itself
  - Relying Parties
  - Identity Providers
  - Other Entity Types, such as those for wallet ecosystems
- Organizations publish signed metadata about the members that belong to them
- Federation operators publish signed metadata about orgs
- Inter-federations publish signed metadata about federations
- Hierarchical metadata is an online graph data structure

Trust Chains

- Participants follow metadata trust chains from leaves up to common roots, verifying signatures
- Both participants are members of a federation if a common Trust Anchor is found
- Participants can be members of multiple federations



Metadata Representation

- Each metadata statement is a signed JSON Web Token (JWT)
  - These are called Entity Statements
- They make statements about
  - The Entity itself
  - Keys used by the Entity
  - Policies of the Entity
  - Other entities up the trust chain that they are willing to trust
    - This is how trust chains can be followed to federation roots

Example Entity Statement

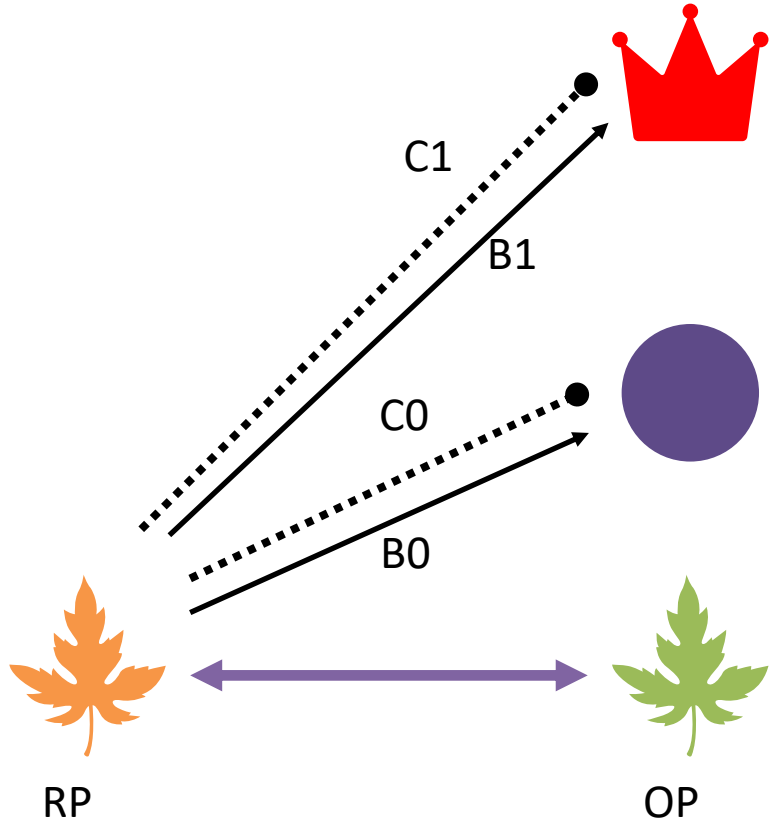

```
{
 "iss": "https://feide.no",
 "sub": "https://ntnu.no",
 "iat": 1516239022,
 "exp": 1516298022,
 "jti": "7l2lncFdY6SlhNia",
 "metadata_policy": {
    "openid_provider": {
      "issuer": {"value": "https://ntnu.no"},
      "organization_name": {"value": "NTNU"},
      "id_token_signing_alg_values_supported":
        {"subset_of": ["RS256", "RS384", "RS512"]},
    }
 },
  "jwks": {
    "keys": [
      {
        "e": "AQAB",
        "kid": "key1",
        "kty": "RSA",
        "n": "pnXBOusEANuug6ewezb9J_...",
        "use": "sig"
      }
    ]
  },
  "authority_hints": [
    "https://edugain.org/federation"
  ]
}
```

Collecting a Trust Chain

Self-signed Entity Statement. First Entity in trust chain.

1. From the claim *authority_hints*, pick superior Entity.
2. Grab superior's self-signed Entity Statement (using .well-known)
3. Request superior's view of subordinate (federation API). Add to the trust chain.
4. GOTO 1

Repeat until superior is a trusted trust anchor



SAML vs. OpenID Federation

SAML

- Appearing in a metadata file means you are part of a federation

OpenID Federation

- Entities with trust chains up to the same trust anchor belong to the same federation

SAML vs. OpenID Federation

SAML

- An entity's complete metadata must be accepted by the federation operator for the entity to be allowed into the federation

OpenID Federation

- The federation operator sets the boundaries of what is acceptable

Policy Language for Entity Statements

- subset_of
- one_of
- superset_of
- add
- value
- default
- essential
- Path length/name restrictions
- Trust/certification marks

Applying Metadata Policies

- Policies applied top-down from root to leaves of trust chain
- Policies higher in the chain override those lower in the chain

- For instance, a Federation Operator might specify that only a particular set of signing algorithms may be used
  - Policies are applied to all entities in the federation

SAML vs. OpenID Federation

SAML

- It is rare that an entity belongs to more then one federation. I believe that eduGAIN recommends that an entity only belong to one.

OpenID Federation

- There is no drawback to belonging to multiple federations

SAML vs. OpenID Federation

SAML

- The is no metadata negotiation

OpenID Federation

- The RP proposes and the OP decides, subject to applicable policies from the trust chain

OpenID Federation Spec Features

OpenID Federation Spec Features

- Entity Identifiers
- Entity Statements
- Trust Chains
- Metadata
- Entity Type Identifiers
- Metadata Policies
- Metadata Constraints
- Trust Marks
- Federation Endpoints
- /.well-known/openid-federation Resources
- OpenID Connect Client Registration

Entity Identifiers

- A globally unique string identifier that is bound to one Entity. They are URLs that use the `https` scheme, have a host component, and MAY contain port and path components.
  - https://op.umu.se
  - https://umu.se
  - https://openid.sunet.se
  - https://edugain.geant.org
- https://openid.net/specs/openid-federation-1_0.html#name-terminology

Entity Statements

- A signed JWT that contains the information needed for an Entity to participate in federation(s), including metadata about itself and policies that apply to other Entities for which it is authoritative.
  - Entity Configuration – Entity Statement about your own Entity
  - Subordinate Statement – Entity Statement about a Subordinate Entity
- https://openid.net/specs/openid-federation-1_0.html#name-entity-statement

Trust Chains

- A sequence of Entity Statements that represents a chain starting at an Entity Configuration that is the subject of the chain (typically of a Leaf Entity) and ending in a Trust Anchor.
- https://openid.net/specs/openid-federation-1_0.html#name-trust-chain

Metadata

- Information declared in an Entity Statement about an Entity
  - Superiors, endpoint URLs, algorithms, contact info, etc.
- https://openid.net/specs/openid-federation-1_0.html#name-metadata

Entity Type Identifiers

- The Entity Type Identifier uniquely identifies the Entity Type of a federation participant and the metadata format for that Entity Type.
  - federation_entity
  - openid_provider
  - openid_relying_party
  - openid_credential_verifier
- https://openid.net/specs/openid-federation-1_0.html#name-entity-type-identifiers

Metadata Policies

- Operators applied to metadata in a Trust Chain
  - `value`
  - `add`
  - `default`
  - `subset_of`
  - `one_of`
- Lets superiors influence metadata values used by Entities
- https://openid.net/specs/openid-federation-1_0.html#name-metadata-policy

Metadata Constraints

- Trust Anchors and Intermediate Entities MAY define constraining criteria that apply to their Subordinates.
  - `max_path_length`
  - `naming_constraints`
  - `allowed_entity_types`
- https://openid.net/specs/openid-federation-1_0.html#name-constraints

Trust Marks

- Trust Marks are statements of conformance to sets of criteria determined by an accreditation authority. Trust Marks are signed JWTs. Entity Statements MAY include Trust Marks.
- https://openid.net/specs/openid-federation-1_0.html#name-trust-marks

Federation Endpoints

- API endpoints providing Federation functionality
  - Fetch Endpoint
  - Subordinate Listing Endpoint
  - Resolve Endpoint
  - Trust Mark Status Endpoint
  - Historical Keys Endpoint
- https://openid.net/specs/openid-federation-1_0.html#name-federation-endpoints

/.well-known/openid-federation Resources

- Entity Identifier + `/.well-known/openid-federation` = URL of Entity Configuration
- Entities publish their Entity Configuration at this URL
- Used to retrieve Entity Configurations for Trust Chains
- https://openid.net/specs/openid-federation-1_0.html#name-obtaining-federation-entity

OpenID Connect Client Registration

- Spec defines two OpenID Connect Client Registration methods
  - Automatic Registration – No pre-registration performed
  - Explicit Registration – Pre-registration performed
- https://openid.net/specs/openid-federation-1_0.html#name-openid-connect-client-regis

Next Steps and Discussion

Next Steps for Specifications

- OpenID Federation 1.0 becomes final in February 2026
- Split into two specifications in non-breaking manner
  - OpenID Federation 1.1 – protocol-independent functionality
  - OpenID Connect Federation 1.1 – protocol-specific functionality
- Work on Federation extension specifications, such as
  - OpenID Federation Wallet Profile
  - OpenID Federation Extended Listing
- Eventually create OpenID Federation 2.0 incorporating extensions that are used in practice

Security Analysis

- OpenID Foundation performs security analysis of key specs
- Security analysis of last Implementer's Draft performed in 2024
  - Security Researchers from University of Stuttgart
  - Found actionable vulnerability in Federation, Connect, OAuth
    - https://openid.net/notice-of-a-security-vulnerability/
- Security analysis of OpenID Federation 1.0 planned
  - Will provide deployers and architects confidence in spec
  - Does cost money – approximately US$72,000
  - OpenID Foundation looking for parties interested in sponsoring part of that work

Upcoming Interop Event

- OpenID Federation Interop at TIIME Unconference
  - https://tiime-unconference.eu/, Feb 9-13, 2026, Amsterdam
  - Organized by Niels van Dijk of SURF and Davide Vaghetti of GARR

- Follows successful interop event at SUNET in April 2025
  - 30 participants, 14 implementations, 15 countries
  - Read about it at https://self-issued.info/?p=2697

What else is happening?

- What else related to OpenID Federation is happening that people may want to know about?

OpenID Federation Resources

- OpenID Federation Specification
  - https://openid.net/specs/openid-federation-1_0.html
- OpenID Connect Page
  - https://openid.net/connect/
- OpenID Connect Working Group Mailing List
  - https://lists.openid.net/mailman/listinfo/openid-specs-ab
- OpenID Blog
  - https://openid.net/
- Mike Jones' Blog
  - https://self-issued.info/

Backup Slides

What about Research and Education?

- Research and Education sector has numerous large-scale identity federations
  - Many national and regional federations
    - Such as SWAMID in Sweden and InCommon in the United States
    - Some have thousands of sites
  - Inter-federations among dozes of federations, such as eduGAIN
- These allow identities from any federation member to be used at relying parties from any federation member
  - For instance, using a University of Washington account at CERN
- BUT… today these are nearly all based on SAML 2
  - Many using Shibboleth software

OpenID Federation Past

- Second Implementer's Draft Approved in January 2020
- Spec refined from discussions at multiple federation events
  - NORDUnet 2017
  - SURFnet 2018
  - TNC/REFEDS 2019
  - Internet2/REFEDS 2019
  - Now OpenID Japan Workshop 2020
- Hackathon with interop among multiple implementations
  - Internet2/REFEDS 2019

OpenID Federation Future

- OpenID Foundation holding three interop events in 2020
  - Much like five interops were held for OpenID Connect
  - Interop results will be used to improve the specification
  - Contact Roland Hedberg roland@catalogix.se to participate
  - Join OpenID Federation Interop mailing list
    - https://groups.google.com/forum/#!forum/openid-federation-interop
- It's time for feedback from developers and early deployers
  - ***Will you be one?***
  - Please read (and implement!) the spec and give us your feedback!

Client Registration Methods

- Automatic
  - The client preforms no client registration. Instead, it sends an authorization request with *client_id* == *entity_id* and client authentication method *private_key_jwt.*
  - The OP fetches the RP's self-signed Entity Statement.
- Explicit
  - The client performs dynamic client registration. The OP responds with an Entity Statement about the RP with metadata policy.
  - The RP provides the OP with its self-signed Entity Statement in the body of the client registration request.