

INDIGO-IAM: sviluppi futuri e federazioni OIDC

Roberta Miccoli
INFN-CNAF

Workshop sul Calcolo nell'I.N.F.N.
Loano (Savona) 22 - 26 maggio 2023



INDIGO Identity and Access Management Service

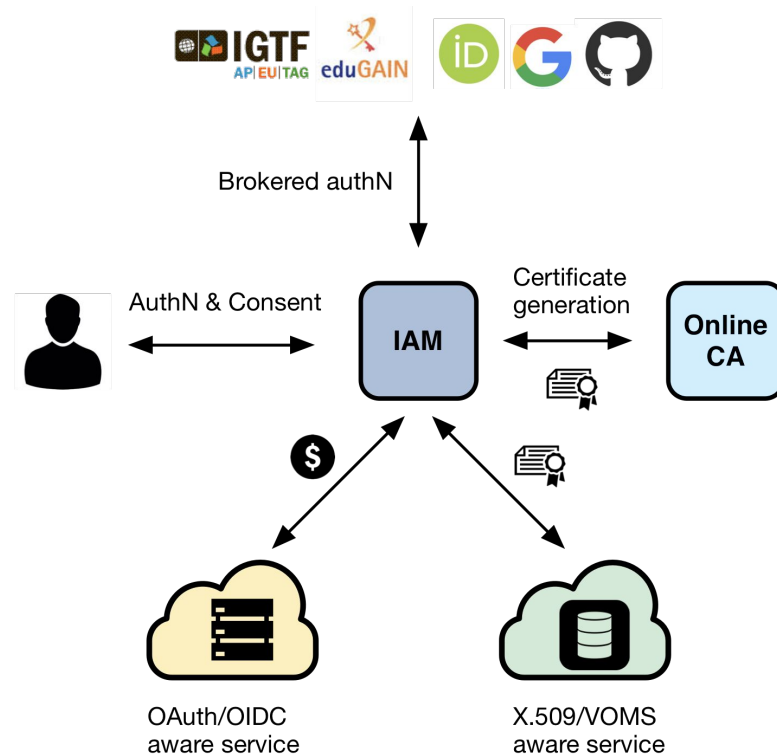
First developed in the context of the **H2020 INDIGO DataCloud** project

~7 years since 1st INDIGO IAM release v0.3.0 (2016-07-12)

Allows consistent authentication and authorization technologies/policies at all Cloud levels (IaaS, PaaS, SaaS) in the context of **INFN Datacloud**



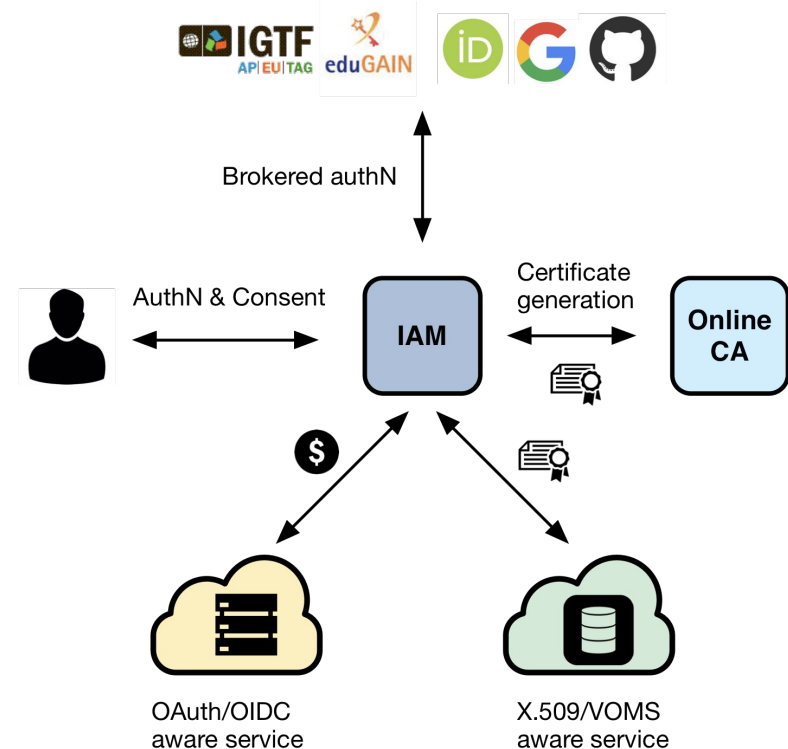
Selected by the WLCG management board to be the core of the future, token-based WLCG AAI



INDIGO Identity and Access Management Service

An authentication and authorization service that:

- supports **multiple authentication mechanisms**
- provides users with a **persistent, organization scoped** identifier
- exposes **identity information, attributes and capabilities** to services via **JWT** tokens and standard **OAuth & OpenID Connect** protocols
- can integrate existing **VOMS**-aware services
- supports **Web** and **non-Web** access, **delegation** and **token renewal**



Latest release: [IAM v1.8.1](#)

Released on: **2023-02-28**

Major highlights:

- Scopes management interface added to IAM dashboard
- Group Manager interface added to IAM dashboard
- Support for [AARC-G069](#) guideline (groups and roles membership information can be requested with the `entitlements` scope and appears in the `entitlements` claim of the access token) to increase conformance to AARC Blueprint Architecture

Scopes management interface

The screenshot displays the 'IAM for dev' dashboard. The left sidebar contains navigation options: Account Management, Home, Organization Management (Users: 16, Groups: 31, Requests, AUP, Clients, Tokens: 166), and Client management (MitreID Dashboard). The 'Scopes' menu item is highlighted with a red box and shows a count of 26. The main content area is titled 'Scopes' and features a '+ New Scope' button and a table of existing scopes.

Scope	Description	Actions
openid	log in using your identity	edit delete
profile	basic profile information	edit delete
email	email address	edit delete
address	physical address	edit delete
phone	telephone number	edit delete
offline_access	offline access	edit delete
scim:read	read access to SCIM user and groups	edit delete
scim:write	write access to SCIM user and groups	edit delete
registration:read	Grants read access to registration requests	edit delete
registration:write	Grants write access to registration requests	edit delete
scim	Authorizes access to IAM SCIM APIs	edit delete

This page replaces the functionality of the old *System Scopes* page of the MitreID dashboard and it is only visible by IAM Admins

Scopes management interface

The screenshot displays the 'IAM for dev' interface. The left sidebar contains navigation options: Account Management, Home, Organization Management, Users (16), Groups (31), Requests, AUP, Clients, Tokens (166), Scopes (26), Client management, and MitreD Dashboard. The main content area is titled 'Scopes' and features a '+ New Scope' button highlighted with a red box and a red arrow. A 'Scope creation form' modal is open, containing the following fields and options:

- Scope:** A text input field labeled 'Scope name' with a subtext 'Single string with no spaces'.
- Description:** A text input field labeled 'Description' with a subtext 'Human-readable text description'.
- Default Scope:** A checkbox labeled 'Default Scope' with the text 'Newly-created clients get this scope by default?'.
- Restricted:** A checkbox labeled 'Restricted' with the text 'Restricted scopes are only usable by system administrators and are unavailable to dynamically registered clients and protected resources'.

At the bottom of the modal are three buttons: 'Create' (blue), 'Reset Form' (yellow), and 'Cancel' (red). The background shows a list of existing scopes, each with a flag icon, a name, a description, and edit/delete icons.

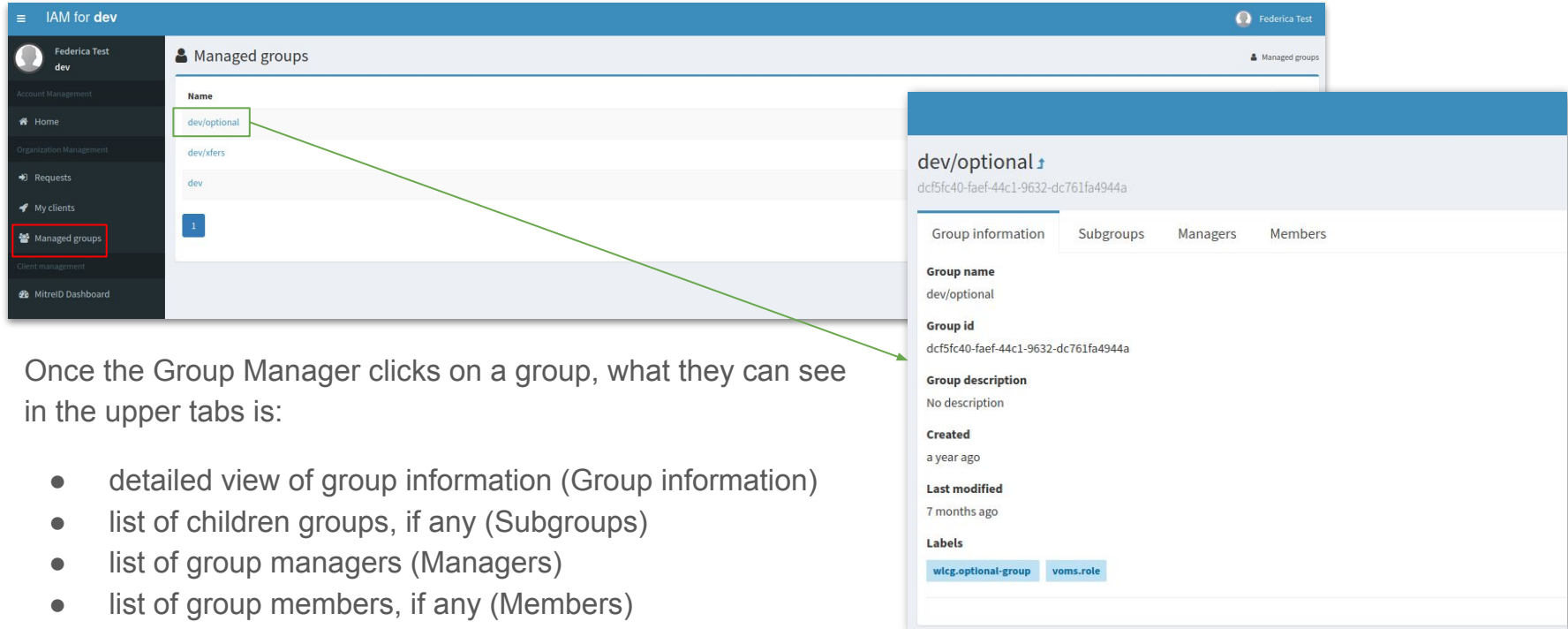
Scope Name	Description
openid	
profile	
email	
address	
phone	
offline_access	
scim:read	read access to SCIM user and groups
scim:write	write access to SCIM user and groups
registration:read	Grants read access to registration requests
registration:write	Grants write access to registration requests
scim	Authorizes access to IAM SCIM APIs

Scopes management interface

The screenshot displays the 'IAM for dev' interface with the 'Scopes' management section active. A modal dialog titled 'Edit Scope' is open, showing the configuration for the 'openid' scope. The dialog includes a 'Description' field with the text 'log in using your identity' and a 'Default Scope' checkbox which is checked. Below the dialog, a table lists various scopes with their descriptions and edit/delete actions.

Scope	Description	Actions
openid	log in using your identity	[Edit] [Delete]
profile		[Edit] [Delete]
email		[Edit] [Delete]
address		[Edit] [Delete]
phone		[Edit] [Delete]
offline_access		[Edit] [Delete]
scim:read	read access to SCIM user and groups	[Edit] [Delete]
scim:write	write access to SCIM user and groups	[Edit] [Delete]
registration:read	Grants read access to registration requests	[Edit] [Delete]
registration:write	Grants write access to registration requests	[Edit] [Delete]
scim	Authorizes access to IAM SCIM APIs	[Edit] [Delete]

Group Manager interface



The screenshot displays the IAM for dev Group Manager interface. On the left is a dark sidebar with navigation options: Home, Requests, My clients, Managed groups (highlighted with a red box), and MitreID Dashboard. The main content area shows a list of managed groups under the heading 'Managed groups'. The list includes 'dev/optional', 'dev/xfers', and 'dev'. A green box highlights 'dev/optional', and a green arrow points from it to a detailed view panel on the right. This panel shows the group name 'dev/optional', its ID 'dcf5fc40-faef-44c1-9632-dc761fa4944a', and tabs for Group information, Subgroups, Managers, and Members. The 'Group information' tab is active, displaying details such as Group name, Group id, Group description, Created, Last modified, and Labels.

Once the Group Manager clicks on a group, what they can see in the upper tabs is:

- detailed view of group information (Group information)
- list of children groups, if any (Subgroups)
- list of group managers (Managers)
- list of group members, if any (Members)

Group Manager interface

A Group Manager in IAM does not have the same privileges as the IAM Admin in managing groups. Currently, they can:

- approve/reject membership requests
- delete users from their managed groups

The Group Manager has also the possibility to click on group members, where a limited view of user information (including name, surname, uuid, username, email, status, created, updated, end time and labels) is shown.

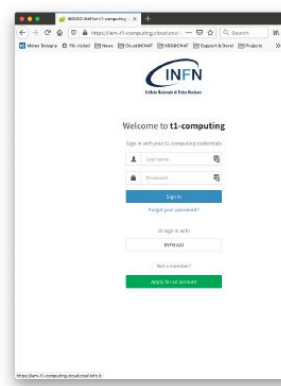
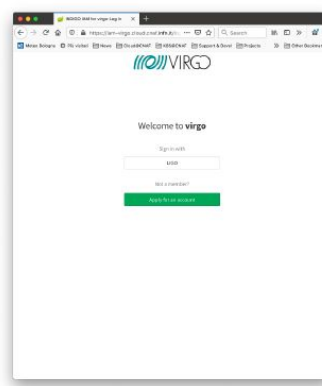
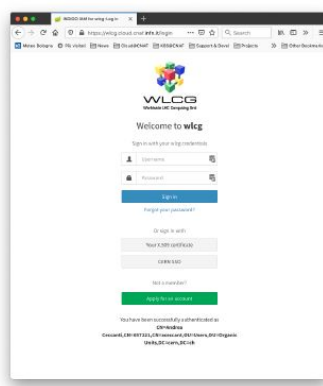
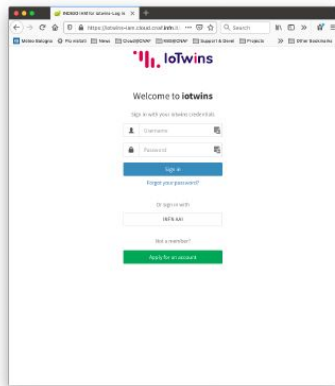
The left screenshot shows the 'IAM for dev' interface for the 'dev/optional' group. The 'Members' tab is active, showing a table with one member: Enrico Vianello. A green box highlights the name 'Enrico Vianello', and a green arrow points to the right screenshot.

The right screenshot shows the user profile for Enrico Vianello. The profile includes a profile picture, name, role (VO administrator), and email (vianellomod). Below this, a table lists user attributes:

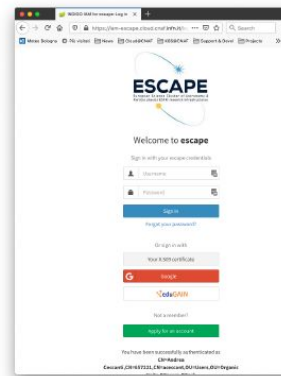
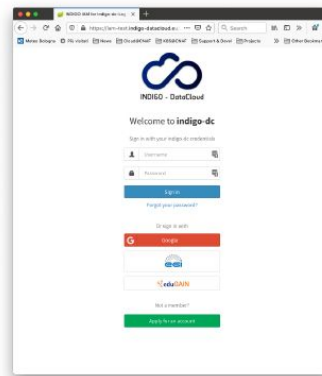
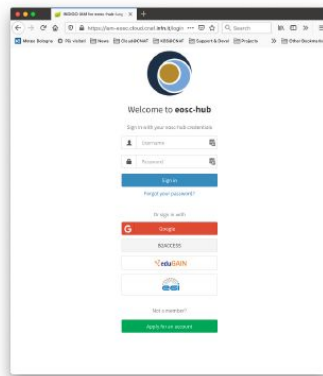
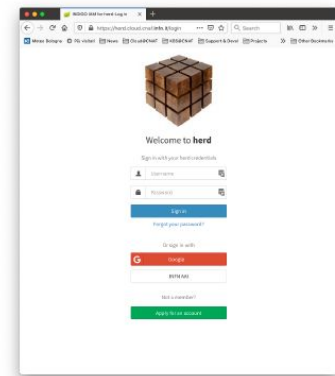
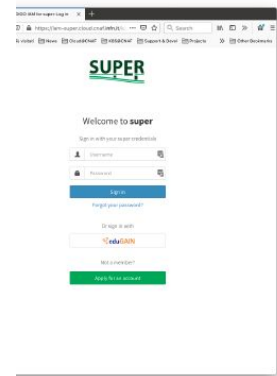
Email	enrico.vianello@cnaif.infn.it
Status	Active
Created	a year ago
Updated	3 months ago
End time	N/A
Signed AUP	a year ago
Labels	br.cern cern_person_id

IAM deployment, performance and HA

IAM deployments at CNAF

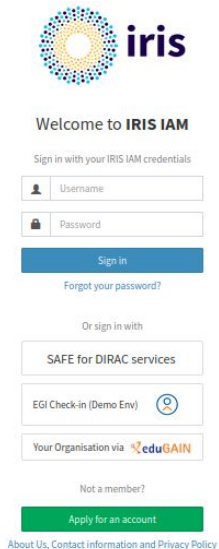


~ 20 IAM instances



IAM deployments outside CNAF

~ 10 IAM instances



The screenshot shows the login interface for the iris IAM system. It features the iris logo at the top, followed by the text "Welcome to IRIS IAM". Below this, there is a prompt to "Sign in with your IRIS IAM credentials" and two input fields for "Username" and "Password". A blue "Sign in" button is positioned below the password field. A link for "Forgot your password?" is located below the sign-in button. An alternative sign-in method is provided: "Or sign in with", followed by three buttons: "SAFE for DIRAC services", "EGI Check-in (Demo Env)", and "Your Organisation via eduGAIN". At the bottom, there is a "Not a member?" link and a green "Apply for an account" button. A footer link for "About Us, Contact information and Privacy Policy" is also present.

iris-iam.stfc.ac.uk



Welcome to **atlas**

Sign in with

CERN SSO

Not a member?

Apply for an account

atlas-auth.web.cern.ch



Welcome to **cms**

Sign in with

CERN SSO

Not a member?

Apply for an account

cms-auth.web.cern.ch



Welcome to **lhcb**

Sign in with

CERN SSO

Not a member?

Apply for an account

lhcb-auth.web.cern.ch



Welcome to **ALICE**

Sign in with

CERN SSO

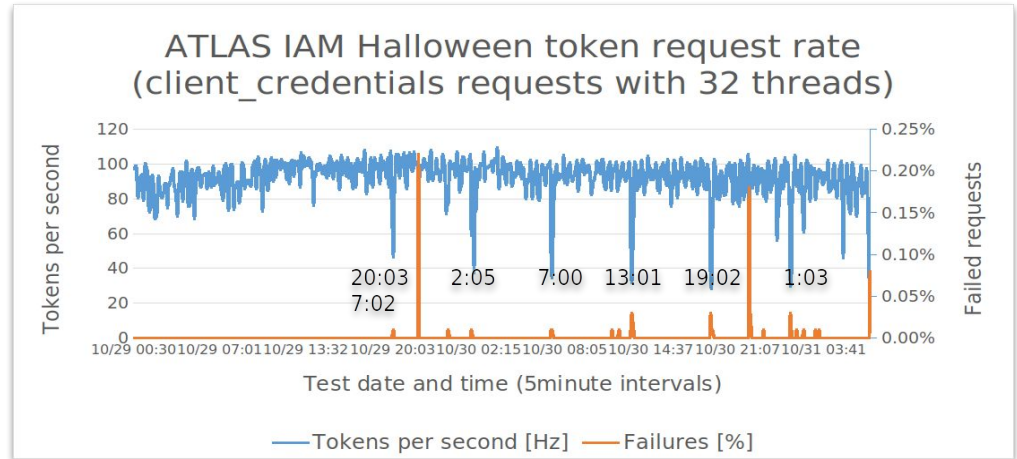
Not a member?

Apply for an account

alice-auth.web.cern.ch

IAM performance: a goal to be achieved

- Unannounced stress tests have been performed on the Atlas IAM instance hosted at CERN
 - `vegeta attack` with 100 Hz token request rate using client credentials grant
- ~100 Hz sustained for more than two days (300 ms response time, 0% error rate)
- then, IAM showed some degradation and it became unavailable afterwards
 - the k8s liveness probe took IAM down because it was not responding to the `/health` endpoint within the timeout
- **Scalability and performance tests are planned for the next [IAM Hackathon](#)**



IAM core technologies

IAM is a **Spring Boot** application

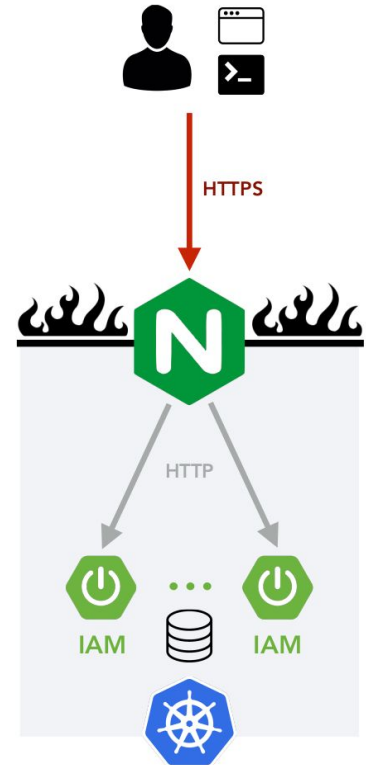
- currently based on the [MitreID Connect](#)
- deployed behind an **NGINX**
- stores data in a **MariaDB/MySQL** database

Horizontally scalable

- all state persisted in the database

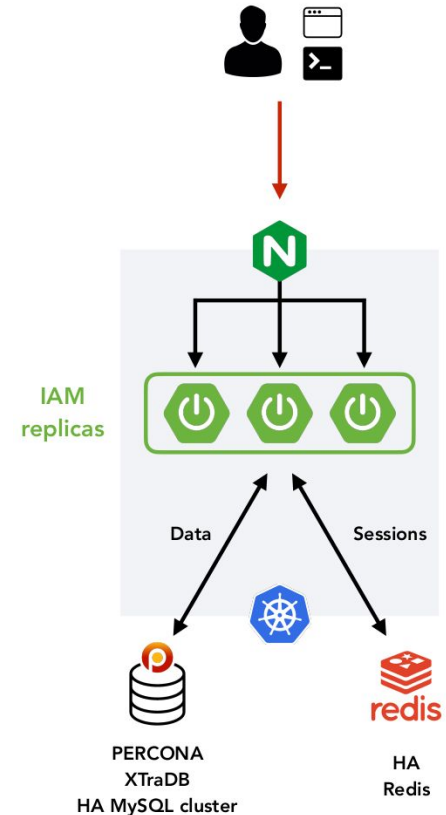
We deploy IAM as a **containerized** service on top of **Kubernetes**

- autoscaling, zero downtime rolling updates



IAM in High Availability

- Starting from version 1.8.0, the IAM service can be deployed in **High Availability** mode
 - IAM supports session data externalization
 - IAM becomes a completely stateless application
- About externalized sessions: IAM relies on redis as external component used to store session data
- Tests in progress: IAM has been deployed with 3 replicas on the dev IAM instance (at CNAF)
 - we faced some cluster limits
 - we planned to use a testbed hosted at CERN



Future IAM developments

Planned release: [IAM v1.8.2](#)

Done:

- Spring dependency version update
- CERN HR suspended status synchronization [#530](#)
- Invalid request error when the AuthZ request is modified during the user approval step [#554](#)

and other minor fixes

In progress:

- Any token created by IAM admin have full access to IAM API [#543](#)
- Token column in database is a limited index and can lead to a "Duplicate entry" error [#579](#)

We are almost ready for a release; it will be available soon.

Our roadmap

In progress:

- Add scope policy management into IAM dashboard [#382](#)
- IAM username update blocked by case insensitive "is username available" check [#434](#)
- Case sensitivity confusion between MySQL unique fields and JPA equals comparisons [#550](#)

To do:

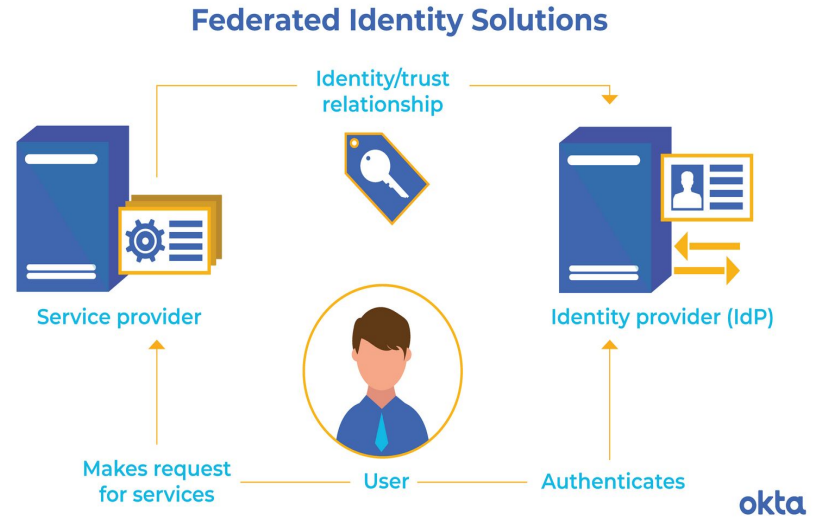
- Local accounts: check password quality [#544](#)
- Support for AARC guidelines [#467](#), [#466](#), [#469](#)
- Can't add certificate with same subject and different issuer [#454](#)
- Client problems due to unsupported response types [#601](#)
- IAM should allow users to request account removal [#362](#)
- Support for Multi-factor Authentication [#418](#)
- **Scalability/availability assessment**
- **Overall security assessment**
- **Support for OIDC Federation model**

...

Introduction to Federations

Identity Federations

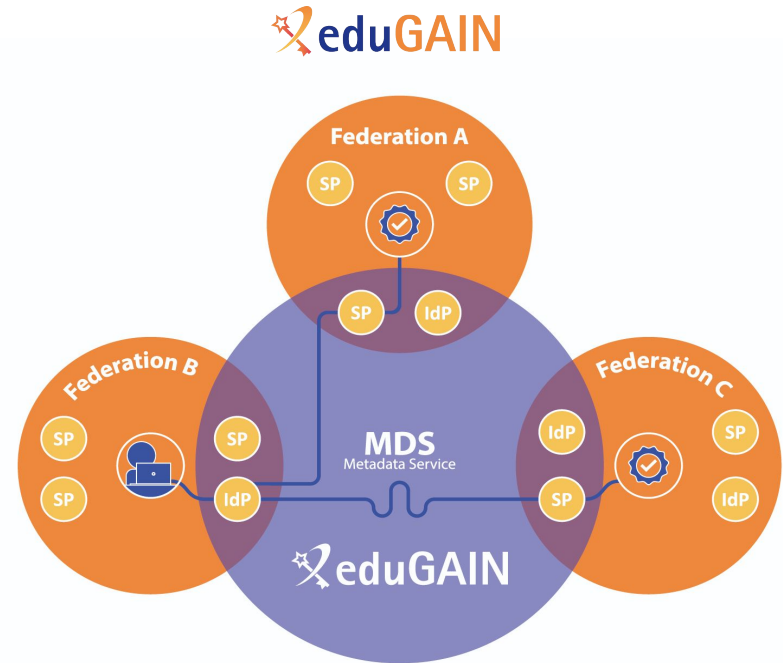
- A method of linking a user's identity across multiple separate identity management systems used by a group of institutions and organisations
- Extended range of services offered to the users
- A Web Single Sign On (SSO) is provided
- Only one account and password are required within the federation
- Common technologies used in federated identity management
 - SAML
 - OAuth/OpenID



SAML Federations

eduGAIN

- eduGAIN is an efficient, flexible way for participating federations, and their affiliated users and services, to interconnect at an international level
- eduGAIN technology involves a *metadata service* that aggregates all the Service and Identity Providers information, and makes this information available to federations
- eduGAIN coordinates necessary elements of the federations' technical infrastructure and provides a policy framework controlling the exchange of this information between Identity Federations
- An Indigo IAM instance can join a SAML federation (e.g. eduGAIN) as a Service Provider



Source: <https://edugain.org/>

EOSC AAI Federation

- [Check-in](#) acts as SAML Service Provider
- The [ESCAPE IAM](#) instance acts as SAML Identity Provider
 - since IAM can only act as SP, in order to integrate the ESCAPE IAM into the EOSC AAI federation an [OIDC-to-SAML proxy](#) has been deployed

From [Nicolas' presentation](#)

The screenshots show the following steps in the authentication process:

- Check-in Selection:** A user is prompted to choose an academic institution. 'ESCAPE IAM' is selected from a list.
- ESCAPE Login:** The user is redirected to the ESCAPE IAM login page, which includes a 'Welcome to escape' message and a sign-in form for username and password.
- Approval Required:** An 'Approval Required for *OIDC to SAML Proxy*' screen is shown, asking the user to authorize the connection. The 'Authorize' button is highlighted.
- SAML 2.0 SP Demo Example:** A 'SAML 2.0 SP Demo Example' screen displays the user's attributes, including:

Attribute Name	Value
urn:oid:1.1.4.1.1.25178.4.1.4	1aeb56fe-1773-4b83-83d9-e0a6e22defac@projectescape.eu
Display name	Nicolas Liampots
Given name	Nicolas
Mail	nliamp@met.gr
Surname	Liampots
Organization	escape
Unique, non-reassignable, persistent pseudonymous ID	1aeb56fe-1773-4b83-83d9-e0a6e22defac@projectescape.eu
User ID	nliampots18
Identity assurance profile	https://aii.eji.eu/LaRLow

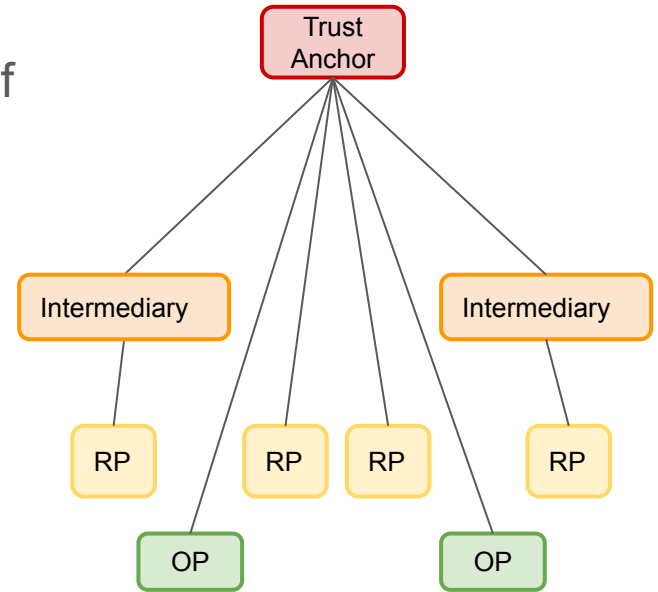
OpenID Connect Federation

OIDC Federation as a solution

- Issue: for participants in an Identity Federation, the onboarding process of the OpenID Connect (OIDC) standard is not sufficient to dynamically establish trust in the information exchanged
- Solution: the [OpenID Connect Federation 1.0](#) specification, being finalised, describes how two entities wishing to interact can **dynamically** retrieve and resolve trust and metadata for a given protocol using a third-party *Trust Anchor*

OIDC Federation entities

- **Trust Anchor (TA):** publishes the configuration of the Federation and the claims of recognition of the parties belonging to the Federation
- **Intermediary:** Soggetto Aggregatore (SA) in Spid, facilitates entry into the Federation, publishes its configuration within the Federation and claims of recognition by its descendants
- **Leaf:** Relying Party (RP) and OpenID Provider (OP)



SAML vs OIDC Federation

SAML	OIDC Federation
A participant in several federations must create ad hoc metadata for each federation	All federation participants publish their own federation metadata (Entity Configuration), which is the same for all federations to which the participant belongs; the final dynamically produced metadata is the result of the various policies acquired by the trust anchors applied to the Entity Configuration

SAML vs OIDC Federation

SAML

The image shows a portion of an Italian identity card. On the left, there are fields for personal data: Nome, nato il (date), Sesso (P for male, S for female), Cittadinanza, Residenza, Via, Stato civile, Professione, and Segni particolari. On the right, there is a large rectangular area labeled 'FOTOGRAFIA'. Below the photo area, there is a section for 'CONNOTATI E CONTRASSEGNI SALIENTI' with fields for Statura, Capelli, and Occhi. At the bottom, there is a section for 'IL SINDACO' with a field for 'Impronta del dito indice sinistro' and a circular stamp area.

OIDC Federation

DICHIARAZIONE SOSTITUTIVA DI ATTO NOTORIO (art. 19 e art. 47 D.P.R. 28 dicembre 2000 n. 445)

La/Il sottoscritto/a Mario Rossi.....
C.F. nata/o a (...)
il e residente a (...) in
via n. di cittadinanza
consapevole della responsabilità penale e delle con-seguenti sanzioni cui può andare incontro in
caso di dichiarazioni mendaci, falsità negli atti, uso di atti falsi, ai sensi dell'art. 76 del D.P.R. n.
445/2000 nonché della decadenza dai benefici eventualmente conseguiti in seguito a
provvedimenti emessi sulla base di dichiarazioni non veritiere, così come previsto dall'art. 75 del
D.P.R. n. 445/2000

DICHIARA
i seguenti stati, qualità personali o fatti¹

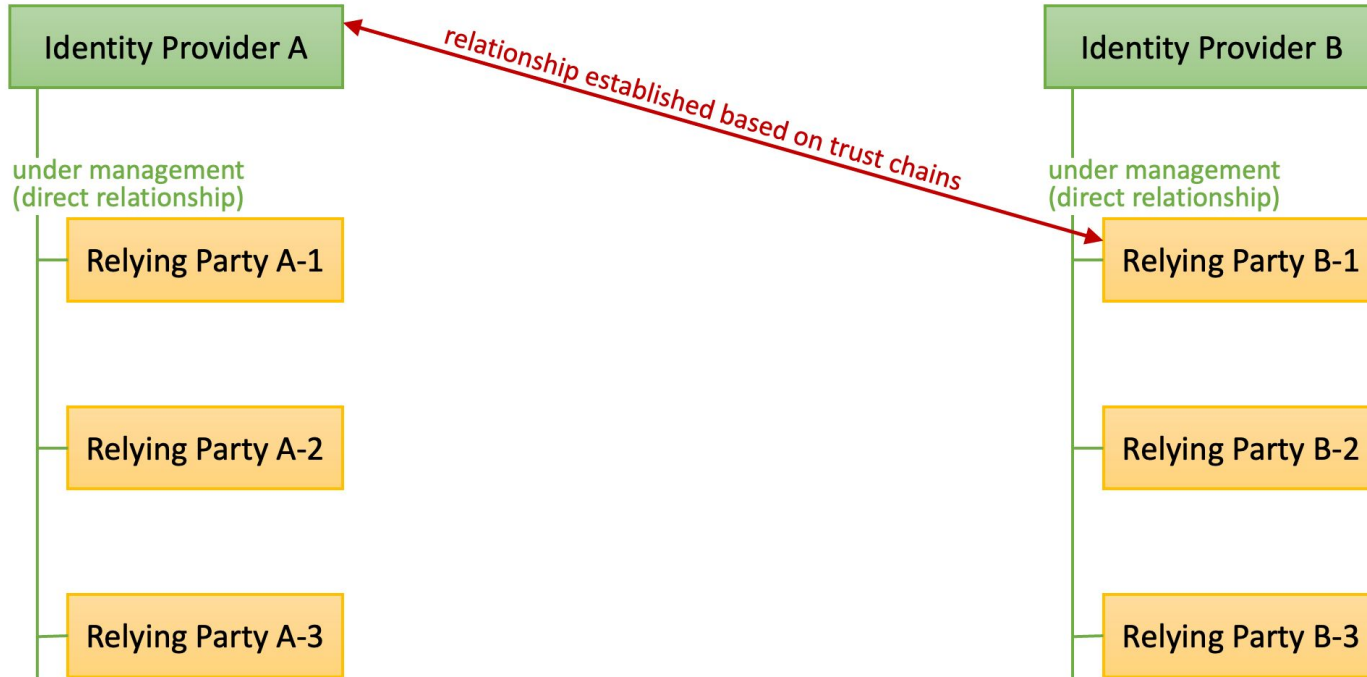
- **SPID-SAML**

- the SAML metadata can be compared to the identity card of a Service Provider (SP)
- the characteristic information of a service is certified by AgID (Agenzia per l'Italia Digitale)

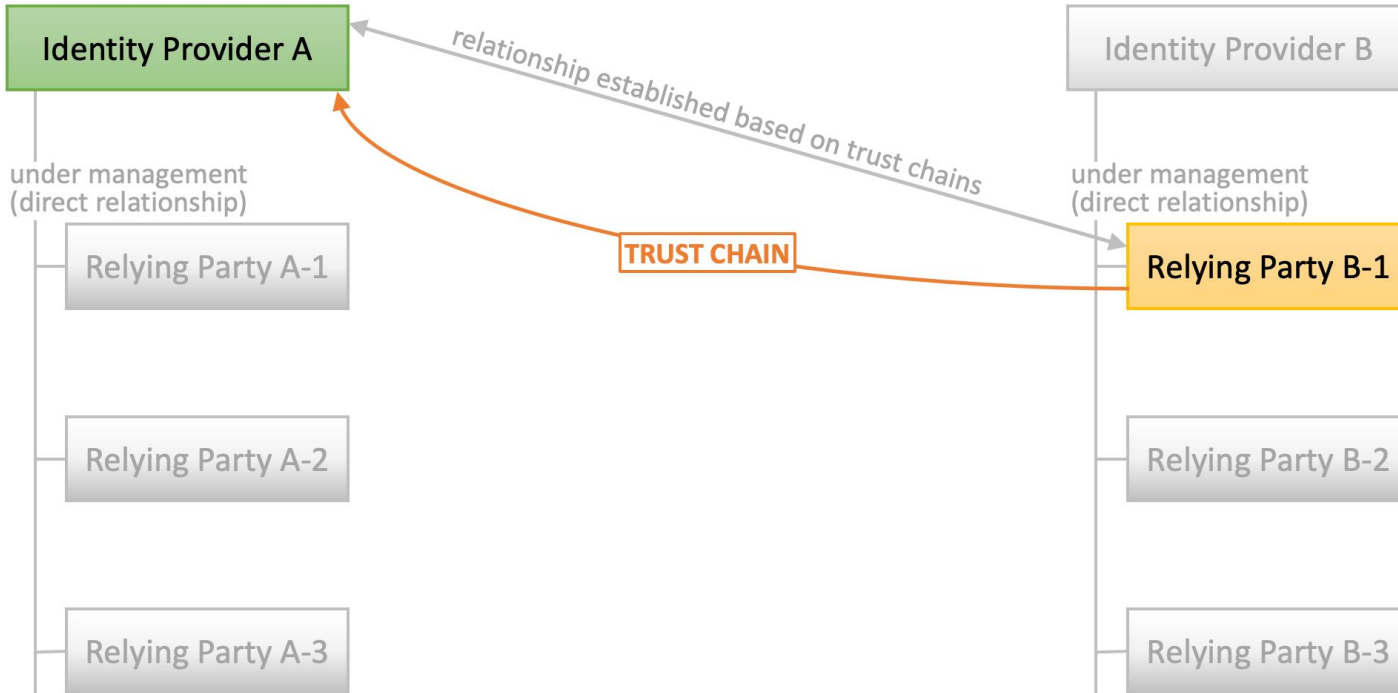
- **SPID-OIDC Federation**

- the Trust Anchor guarantees the identity of the federation members
- federation member declares their characteristics
- e.g. in the declaration in lieu of affidavit, Mario Rossi declares and signs his characteristics

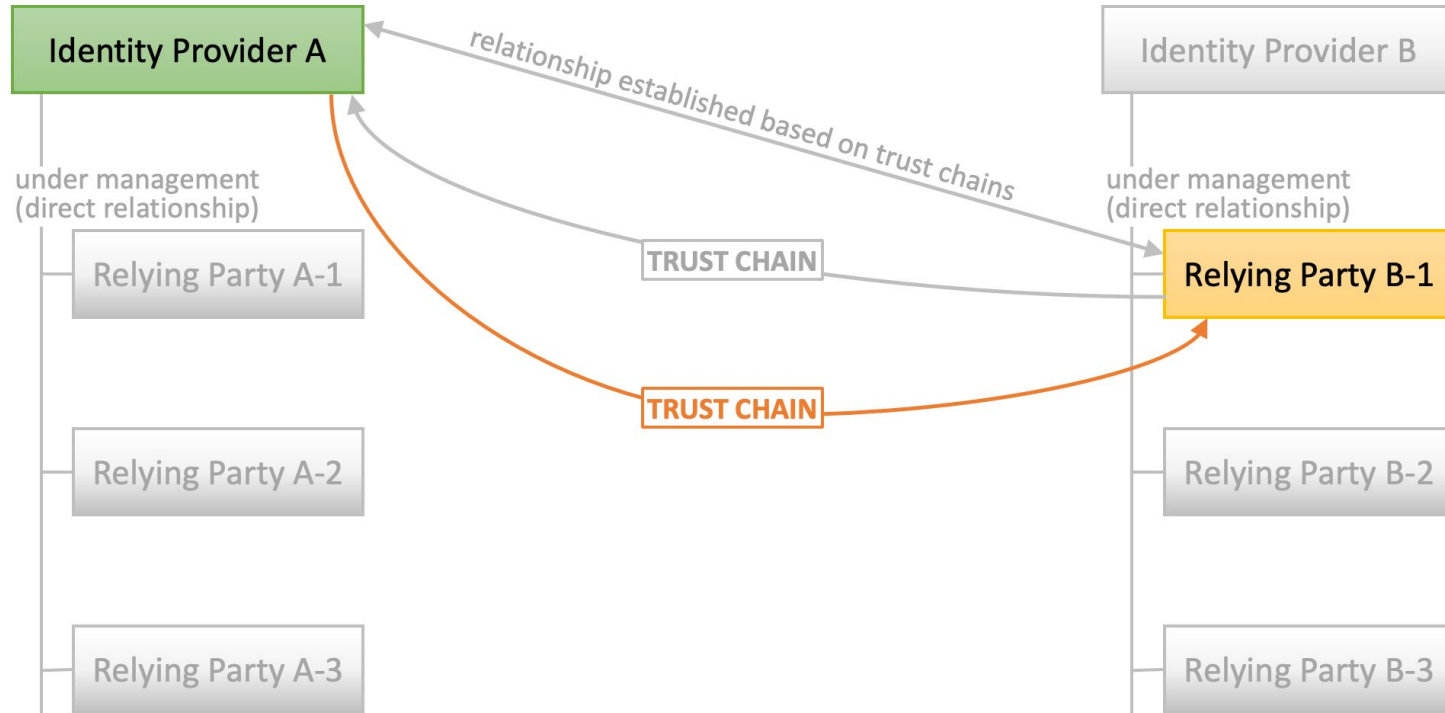
OIDC Federation overview



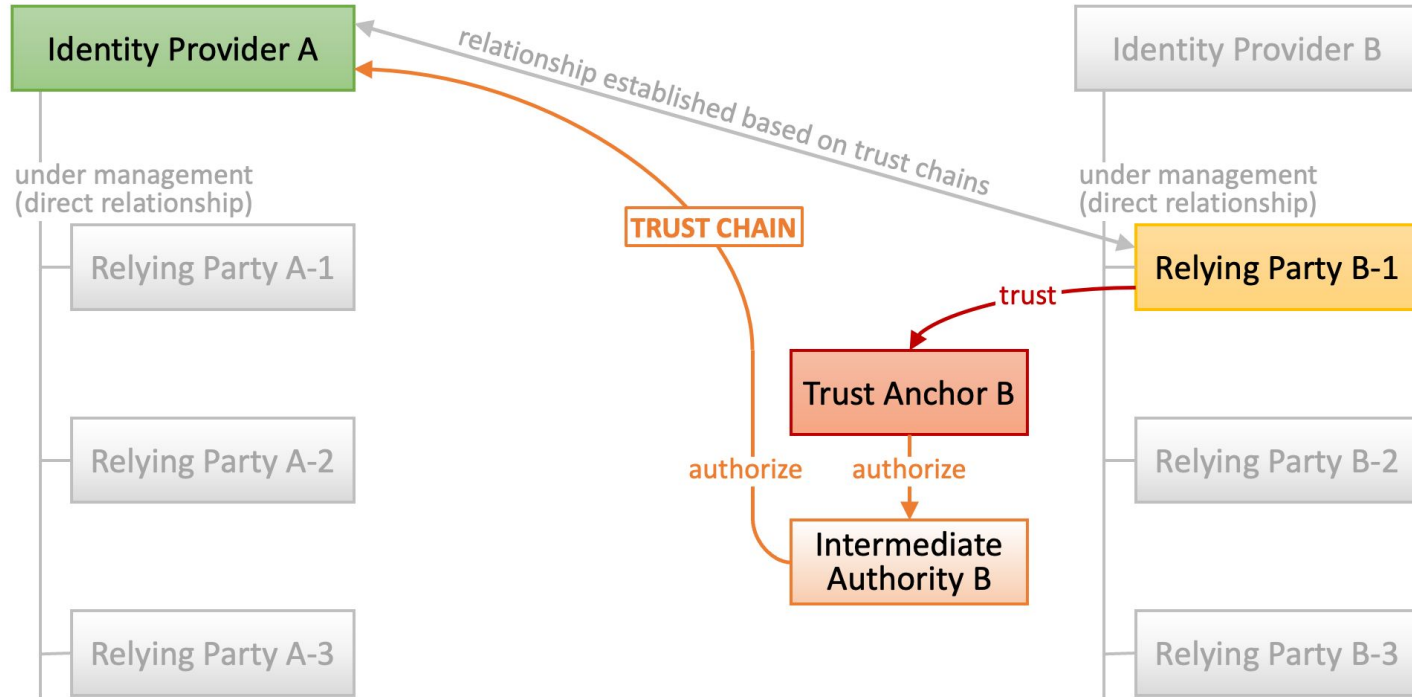
OIDC Federation overview



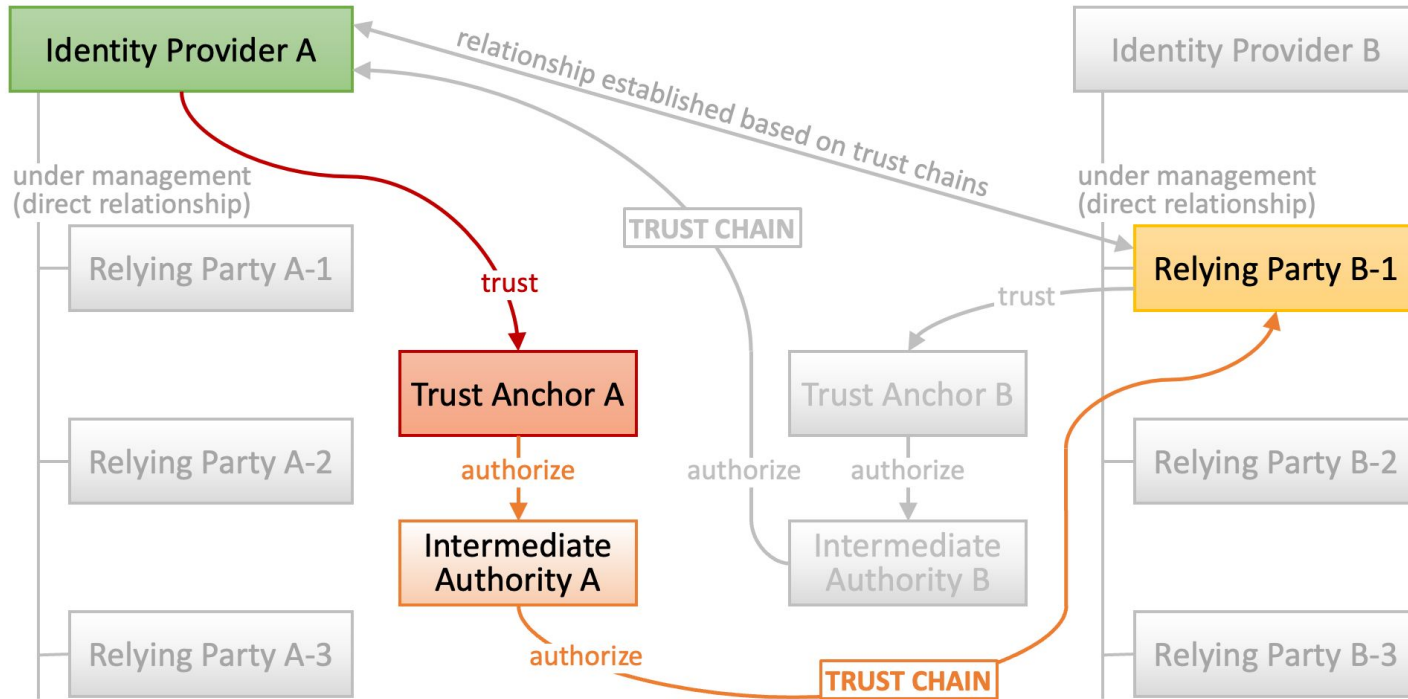
OIDC Federation overview



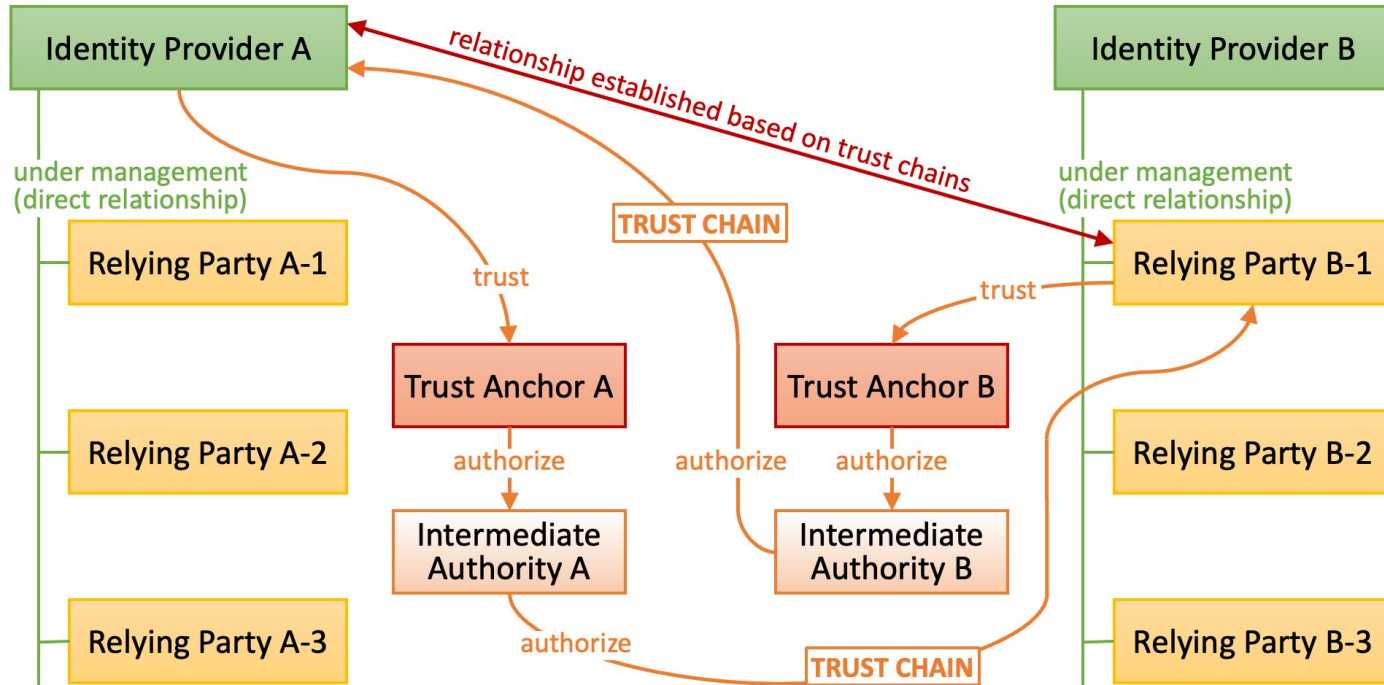
OIDC Federation overview



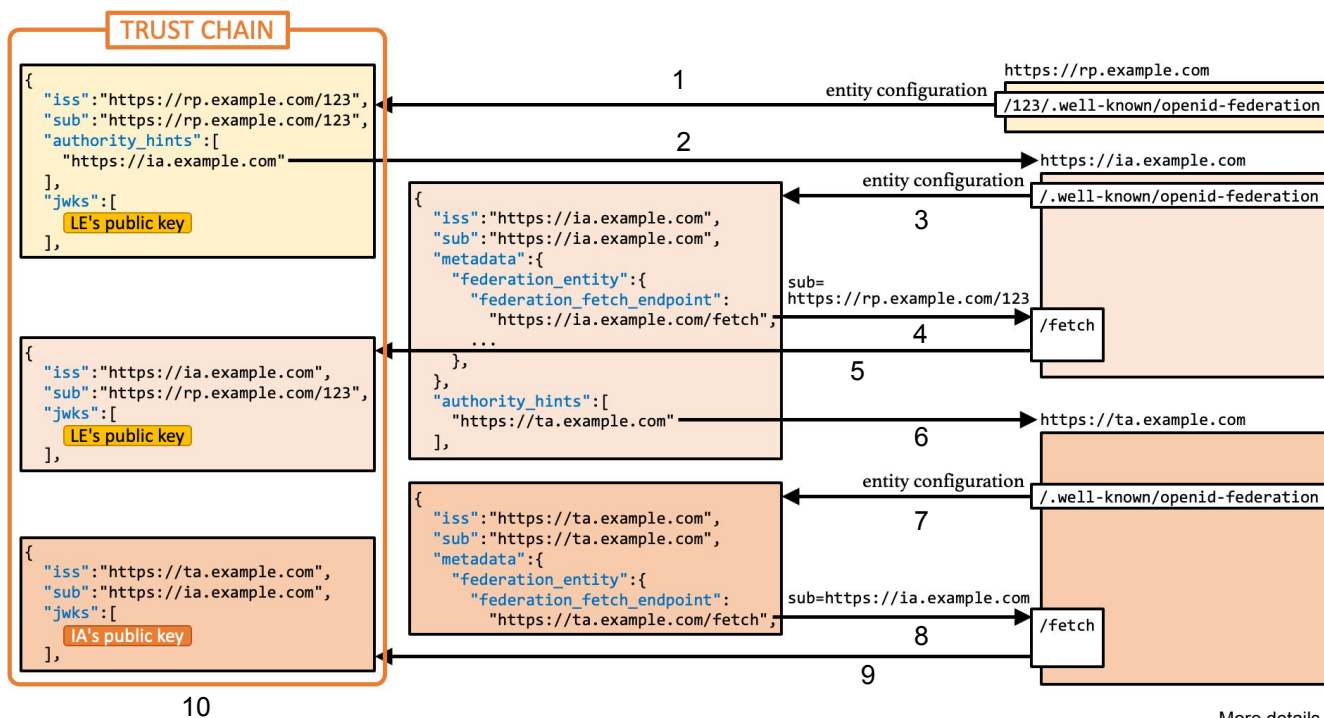
OIDC Federation overview



OIDC Federation overview



Trust Chain resolution flow



More details in the backup slides

Thanks for your attention!



Useful references

IAM on GitHub: <https://github.com/indigo-iam/iam>

IAM documentation: <https://indigo-iam.github.io/docs>

IAM in action video: <https://www.youtube.com/watch?v=1rZlvJADOnY>

For general information:

- OAuth 2.0: <https://oauth.net/2/> and OAuth 2.1: <https://oauth.net/2.1/>
- OpenID Connect: <https://openid.net/connect/>
- OpenID Connect Federation: https://openid.net/specs/openid-connect-federation-1_0.html

Contacts:

- iam-support@lists.infn.it

Questions?

Backup slides

Terminology

- **Entity Statement:** a signed JWT issued by a superior entity (TA or SA) concerning a descendant entity (RP, OP or SA) and containing the descendant's public key, the Trust Marks issued and the metadata policy to be applied to the subject metadata
- **Entity Configuration:** an Entity Statement issued by an Entity about itself, in self-signed JWT format; it contains the Entity's signing keys and further data used to control the Trust Chain resolution process, such as authority hints
- **Trust Mark:** statement of conformance to a well-scoped set of trust and/or interoperability requirements as determined by an accreditation authority, in signed JWT format; the Leaf that acquires the trust mark during the onboarding phase must include this in its EC as a recognition badge
- **Metadata:** document describing an implementation of an OpenID Connect entity

Terminology

- **Metadata policy:** the Trust Anchor publishes the rules and policies to be applied on descendant metadata
- **Authority hint:** an array of url values corresponding to the identifiers of the superior entities (TA or SA) issuing an ES for their descendants
- **Federation Entity Discovery:** collection of EC and ES; starts from a Leaf entity until the TA is reached
- **Trust Chain:** validation procedure of the EC and ES sequence collected through Federation Entity Discovery, the successful outcome of which is a final metadata related to an entity and the expiry date by which it must be updated

Federation API Endpoints

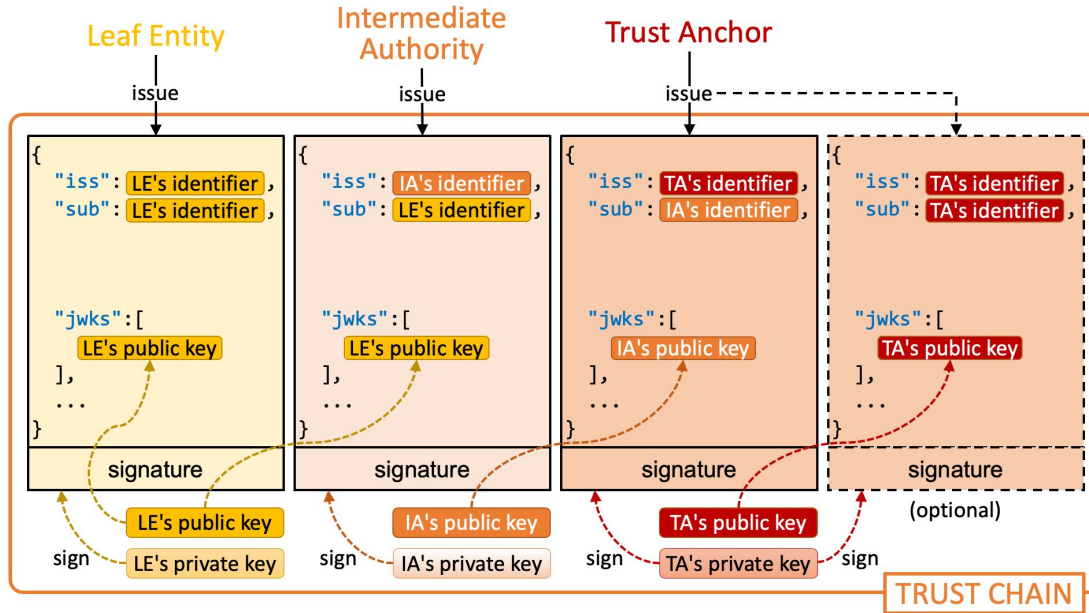
All participants:

- **.well-known/openid-federation**: well known web path where the EC is located
- **federation_resolve_endpoint**: url where the ES can be obtained with pre-processed Trust Chains

Trust Anchor and intermediaries:

- **federation_listing_endpoint**: url where the list of descendants can be obtained in JSON format
- **federation_fetch_endpoint**: url where the ES in JWT format of the descendants are published
- **federation_trust_mark_status_endpoint**: url where it is possible to check whether a Trust Mark is still active or not
- **federation-historical-jwks**: url where the list of historicised TA keys can be obtained

Trust Chain



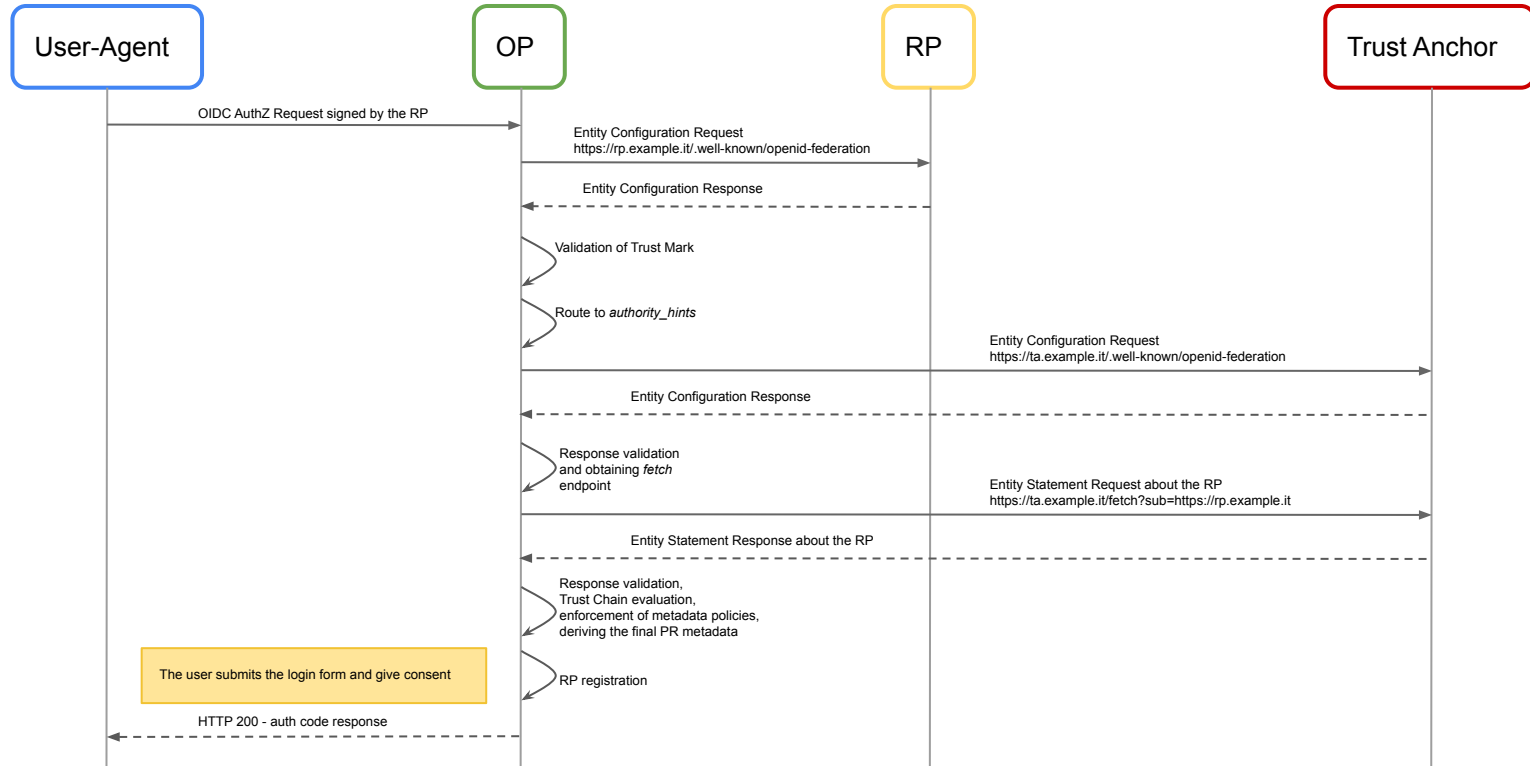
The **Trust Chain** is a sequence of JWTs that are issued by a leaf entity, zero or more intermediate authorities, and a trust anchor.

The Trust Chain linking the declarations to each other can be verified by signing each declaration. Once verified, the metadata policy is applied and the resulting final leaf metadata is saved with an expiry date

Trust Chain resolution flow

1. The first step is to get the EC of a leaf entity by querying its *.well-known/openid-federation* endpoint
2. The EC contains the *authority_hints* claim, a JSON array listing entity IDs of intermediate authorities or trust anchors
3. The EC of the intermediate authority is needed to know the URL of its federation fetch endpoint
4. An HTTP request with the *sub* request parameter is sent to the federation fetch endpoint (e.g. <https://ia.example.com/fetch?sub=https://rp.example.com>)
5. The federation fetch endpoint returns a JWT (ES) that indicates that the intermediate authority authorizes the leaf entity
6. The upper authorities of the intermediate authority are listed in the *authority_hints* claim in the EC of the intermediate authority
7. The EC of the trust anchor is needed to know the URL of its federation fetch endpoint
8. An HTTP request with the *sub* request parameter is sent to the federation fetch endpoint (e.g. <https://ta.example.com/fetch?sub=https://ia.example.com>)
9. The federation fetch endpoint returns a JWT (ES) that indicates that the trust anchor authorizes the intermediate authority
10. The entity configuration of the leaf entity and the JWTs issued from the federation fetch endpoints consist of a Trust Chain

The perspective of an OpenID Provider



OpenID Connect communication

- In a typical implementation of identity provider, identifiers of relying parties (clients) are under the management of the identity provider
 - The identifiers are unique only in the realm of the identity provider
- In the OIDC Federation world, every federation entity has a globally unique identifier
 - The globally unique identifiers, i.e., entity IDs, can be used as a client ID in OAuth/OIDC requests
 - An authorization request like below can be made

```
https://idp.example.com/authorize?request_uri=...& client_id=https://rp.example.com
```

OpenID Connect communication

There are two alternative approaches to establish trust between a RP and an OP

- **Automatic Registration:** enables a RP to make Authentication Requests without a prior registration step with the OP; the OP resolves the RP's EC from the Client ID in the Authentication Request, following the process defined in the previous slide
- **Explicit Registration:** involves performing an explicit registration step for a new client before the RP interacts with an OP for the first time, similar to the process specified by [OpenID Connect Dynamic Client Registration 1.0](#), but where the client registration request contains the Entity Configuration or an entire Trust Chain