INFN-AAI Single Sign-On

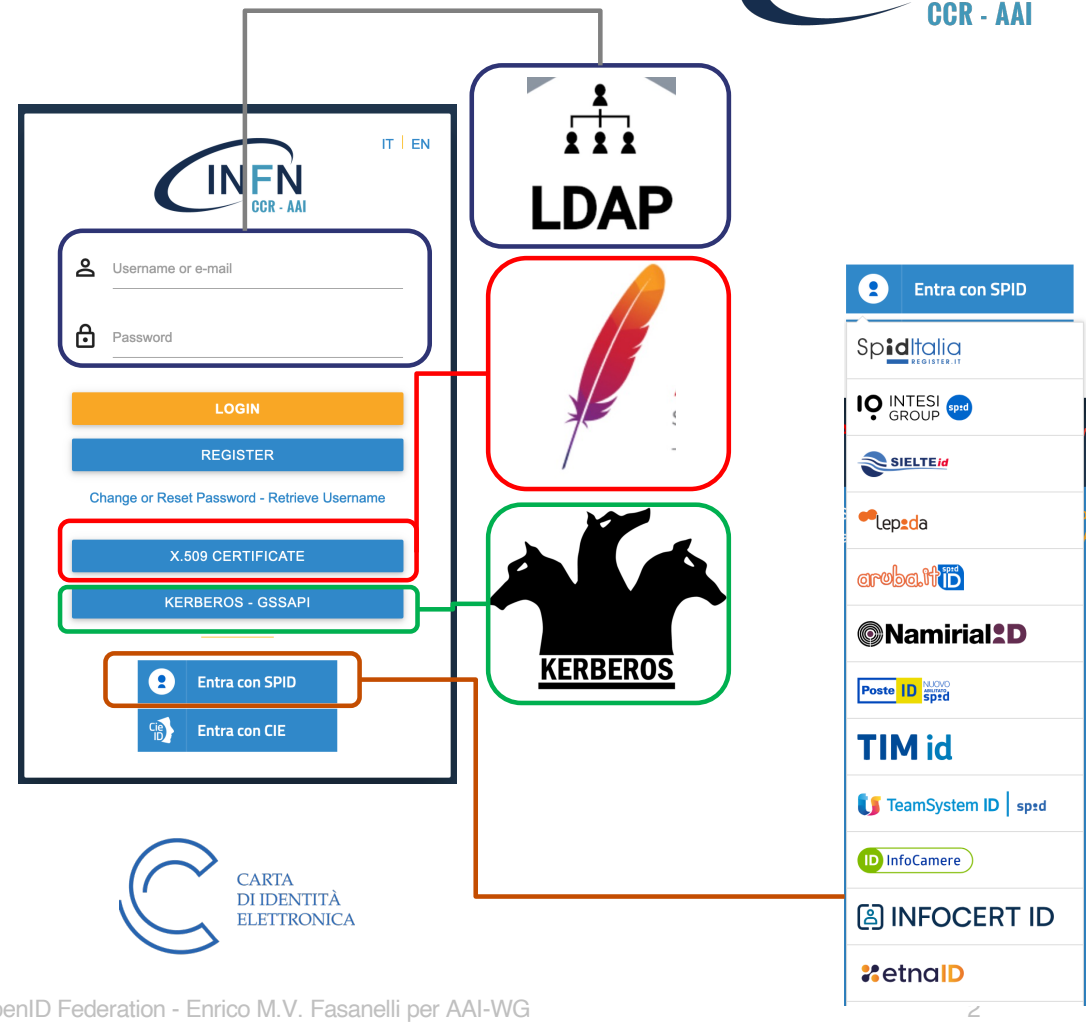
Da SAML e OAuth2/OIDC a OpenID Federation

Biodola - 30 maggio 2025

Enrico M.V. Fasanelli per AAI-WG

INFN-AAI in una slide

- Il sistema di Autenticazione ed Autorizzazione di INFN-AAI è stato disegnato per essere:
  - Single Sign-On compatibile con il «preinstallato» (Kerberos/AFS)
  - Multi-backend (inizialmente solo LDAP, X.509, Kerberos/AFS, poi anche SPiD/CIE)
  - Multi-protocollo (inizialmente solo SAML ed LDAP, poi anche OIDC/OAuth2)
  - Fault-tolerant



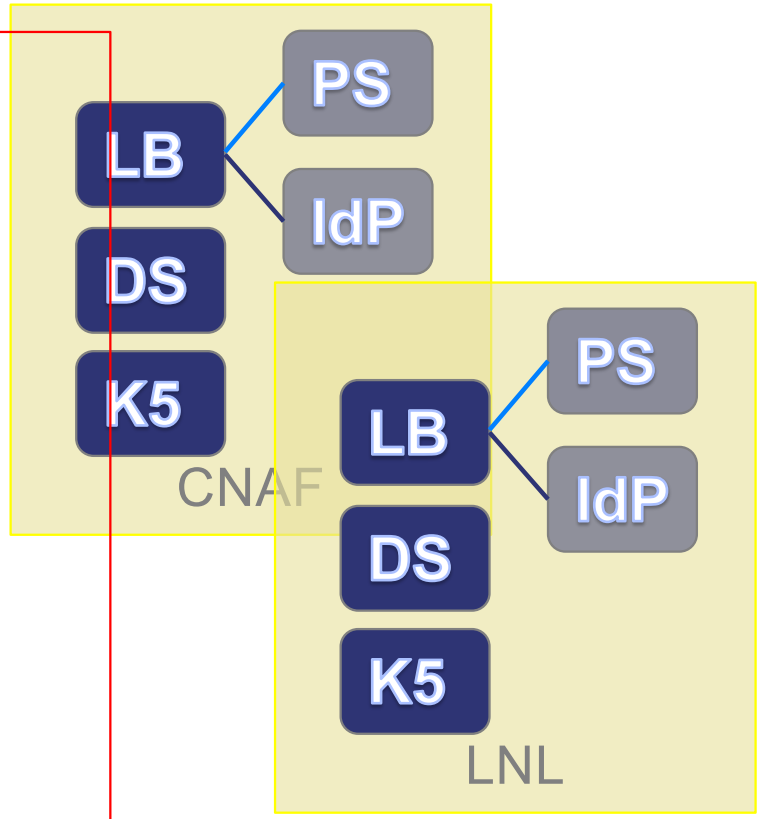
Off-topic: Man In The Middle?

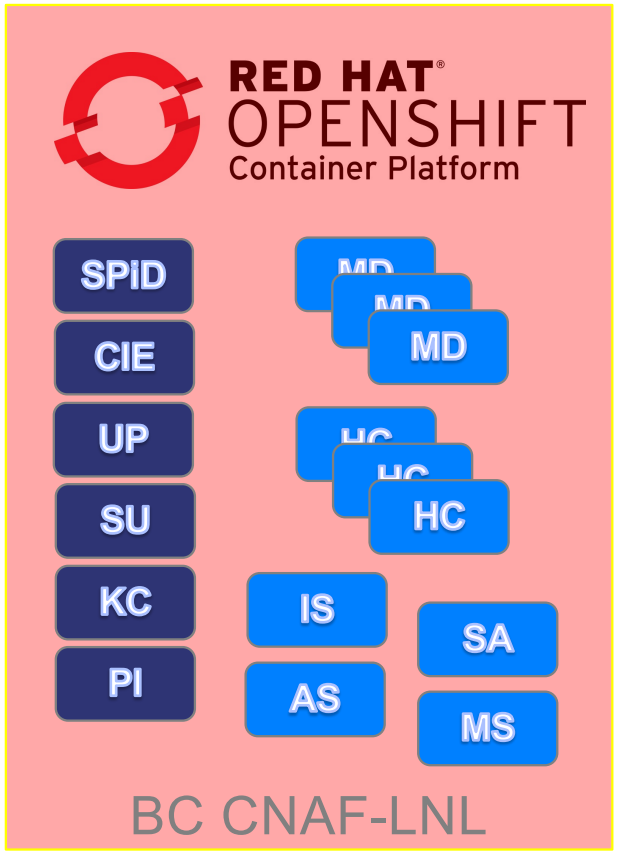
- Il back-end di autenticazione LDAP (coppia username-password) permette un attacco MITM che superi anche il secondo fattore, come quello che ci ha mostrato Vincenzo.
- X.509, SPiD e CIE sono da verificare, ma non credo che funzioni
  - X.509: il server remoto può permettere l'autenticazione, ma poi non ha i dati (username) per poter effettuare la chiamata alla API di privacyIDEA che si aspetta la coppia username-OTP
  - SPiD/CIE: il server dovrebbe poter impersonare il nostro proxy-SP SAML, ma per poterlo fare deve avere accesso alla chiave privata del certificato con cui l'SP firma l'asserzione SAML da inviare agli IdP di SPiD e CIE
- Con Kerberos-GSSAPI l'attacco MITM non può funzionare
  - Il browser deve essere configurato per permettere GSSAPI verso *.infn.it e quindi qualunque dominio differente invalida la **mutua autenticazione** propria di Kerberos5
  - Si può fare, ma solo via DNS spoofing + chiave privata Kerberos dell'IdP

➔BONA PRATICA: ISTRUIRE I NOSTRI UTENTI AD USARE GSSAPI ←

Fault tolerant: VM & POD

- LB: Load Balancer
- DS: Directory Server
- K5: KDC Kerberos5
- IdP: Identity Provider SAML
- PS: ProtoServ
- SPiD: SP SAML proxy
- CIE: SP SAML proxy
- UP: UserPortal
- SU: SignUp
- KK: Keycloak
- PI: PrivacyIDEA
- IS: Identity Service
- AS: Account Service
- MD Mongo DataBase
- HC: Hazelcast
- SA: Site Access
- MS: Mail Service





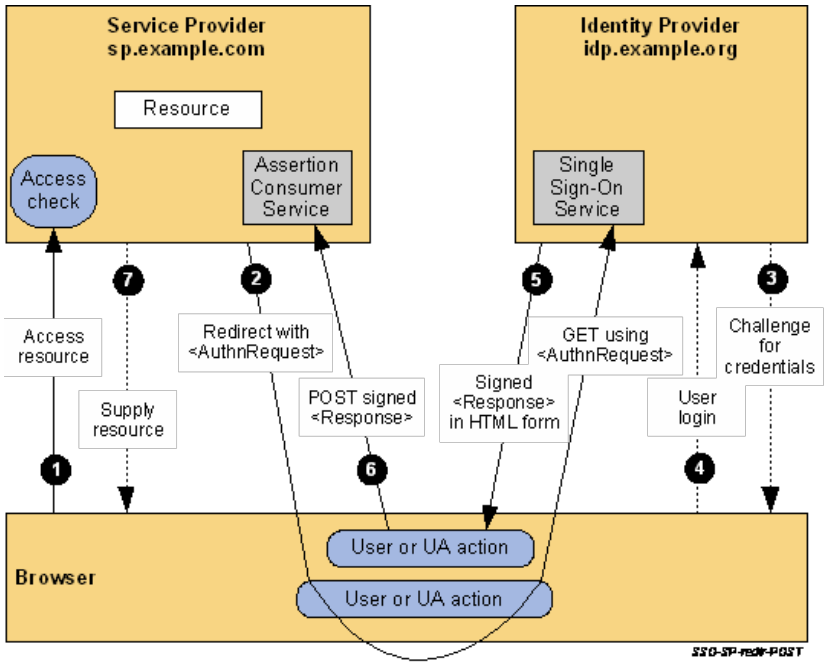
Gli attori in SAML2.0

- Web browser dell'utente
- IdP – Identity Provider
- SP – Service Provider

- Il flusso funziona se SP ed IdP si riconoscono
- l'IdP deve conoscere i metadati dell'SP e vice-versa
  - L'IdP deve conoscere almeno l'endpoint ACS
  - L'SP deve conoscere almeno l'endpoint del SSO Service ed il certificato dell'IdP
- AuthnRequest da SP ad IdP può essere cifrata con il certificato dell'IdP
- La risposta SAML è sempre almeno firmata dall'IdP

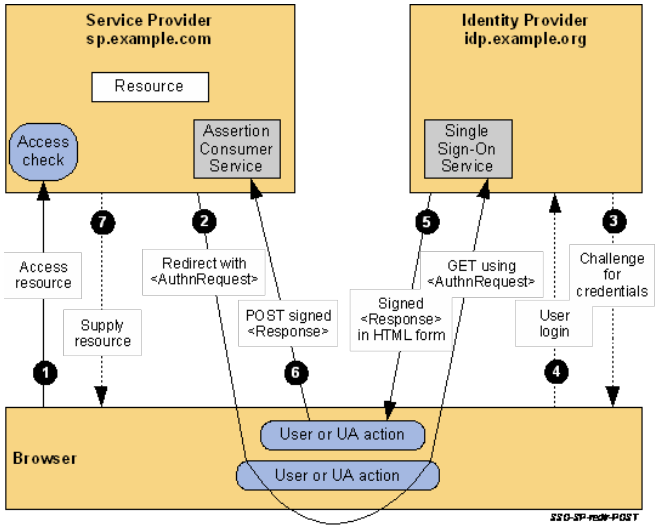
- **L'assertion SAML nella risposta può essere cifrata con il certificato dell'SP**

➔ENFORCING CIFRATURA ASSERZIONE SAML?←

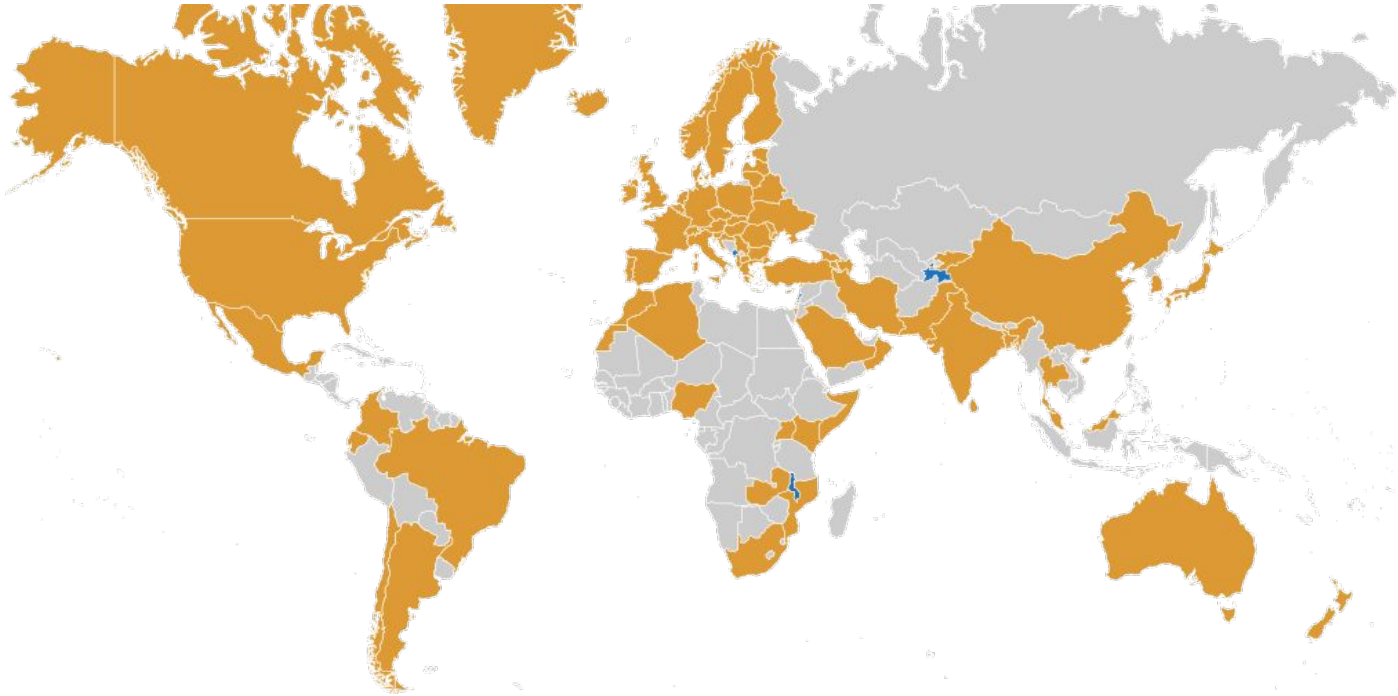


Federazioni & SAML

- Il concetto di federazione non è nelle specifiche SAML2.0
- La federazione (come IDEM) è essenzialmente un «terzo fidato» (terzo rispetto IdP ed SP) che distribuisce metadati, che sono da esso «certificati»
- L'endpoint «access check» dell'SP viene sostituito da un servizio di discovery (WAYF - Where Are You From) che permette all'utente di scegliere l'IdP tra quelli che aderiscono alla federazione
- L'IdP, che ha precedentemente ottenuto dalla federazione i metadati dell'SP, permette l'autenticazione e conosce l'endpoint ACS dell'SP.
- Estendendo questo schema ad un ulteriore livello, si hanno le federazioni di federazioni (come eduGAIN)



eduGAIN Global Coverage



80 Federations

9536 Entities

5754 Identity Providers

3800 Service Providers

Last update November 6th 2024

Davide Vaghetti - GARR - davide.vaghetti@garr.it

eduGAIN Technological Profiles: SAML 2.0

An open standard

Extremely successful and adopted

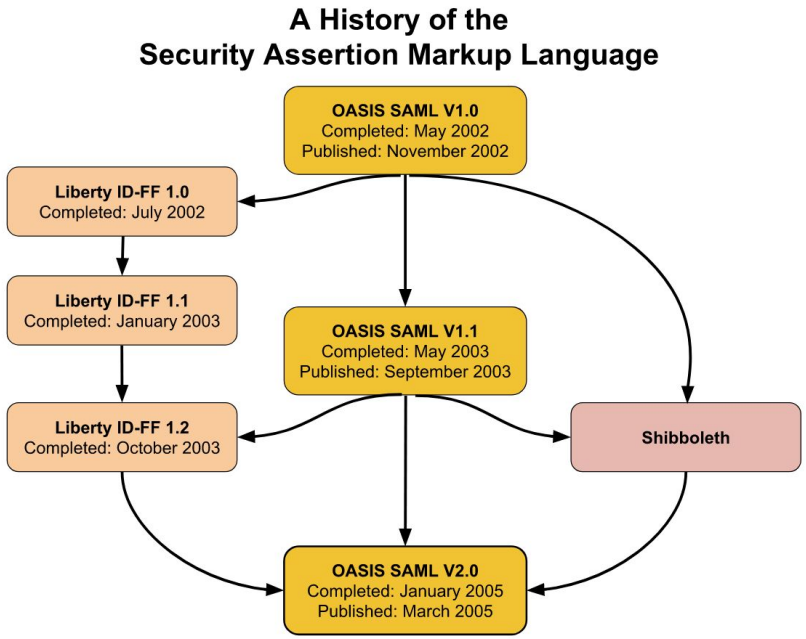
87 R&E Federations + eduGAIN

Legacy Protocol: no new devs in the last 5 years

No support for Mobile App, REST/API flows, etc.

Decentralized identity and Verifiable Credentials?

Post-quantum cryptography support?



Altro «rosso»

- Sul terzo fidato i partecipanti alla federazione devono avere una «fiducia cieca».
- SP e IdP non hanno nessun meccanismo di controllo sui propri metadati.
- Il gestore di federazione può modificare i metadati dell'SP (tranne solo l'endpoint ACS ed eventualmente il certificato) a piacimento senza che l'IdP possa stabilire se ciò sia stato fatto.
- Stesso dicasi per i metadati dell'IdP (tranne solo per l'endpoint SSO ed il certificato) che vanno a finire nel servizio di discovery
- Noi modifichiamo i metadati ➔

```
$idemAuthProc = [
    'class' => 'authorize:Authorize',
    'isMemberOf' => [
        '/^i:infn:.*/',
    ],
    'reject_msg' => [
        'en' => 'You are not authorized accessing to requested resource.',
        'it' => 'Non sei autorizzato ad accedere alla risorsa richiesta.',
    ]
];
```

- OpenID Federation 1.0 risolve questo problema

OpenID Connect 1.0

- Protocollo che aggiunge il livello di Autenticazione al protocollo OAuth2.0 (che tratta solo Autorizzazione)
- Supporto per mobile e REST/API (microservizi)

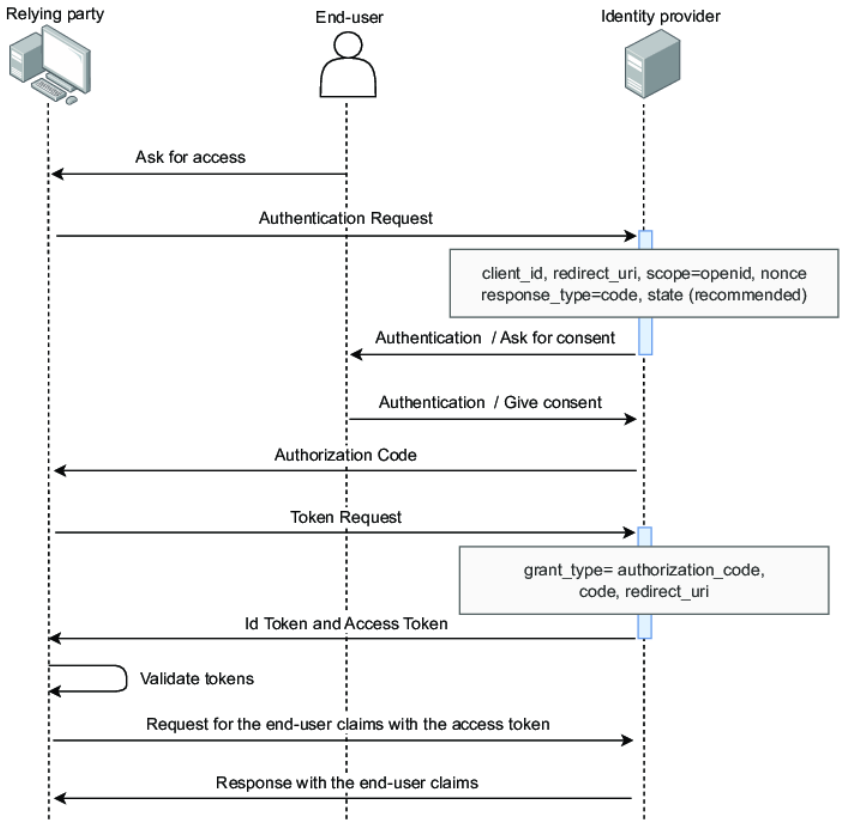
- Protagonisti

Ruolo	Nome in OIDC	Descrizione
Utente	End-User	La persona che vuole autenticarsi
Identity Provider	OpenID Provider (OP)	Il servizio che autentica l'utente
Service Provider	Relying Party (RP)	L'app che richiede l'identità dell'utente

OIDC Authorization code flow

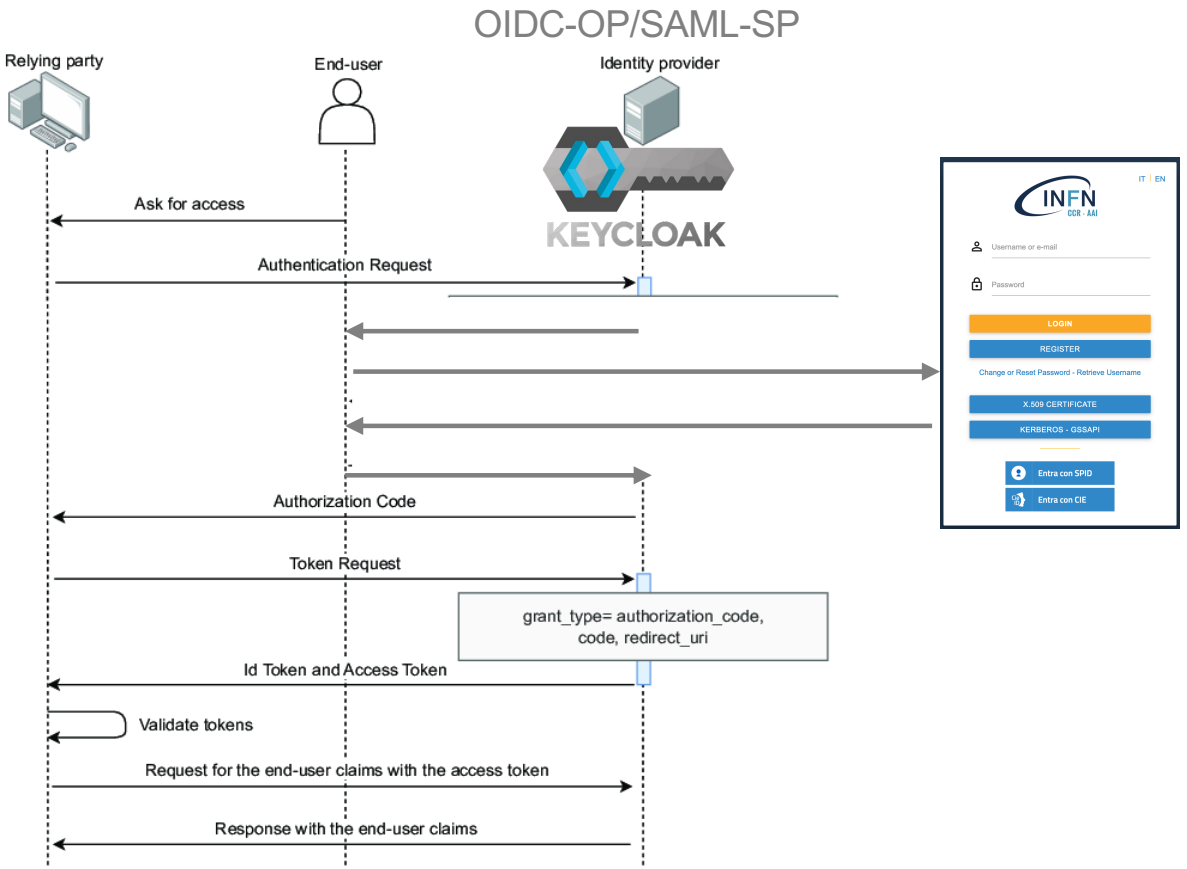
- L'utente prova ad accedere al RP che lo rimanda all'IdP
- L'utente si autentica sull'IdP che gli fornisce un **`authorization_code`** e lo redirige al RP

- Il RP parla direttamente con l'IdP usando l' **`authorization_code`** e richiede i token
  - **`id_token`**
  - **`access_token`**
- L'IdP fornisce i token all'RP che li valida
- L'RP chiede all'IdP i **`claim`** (attributi) dell'utente usando l'**`access_token`**



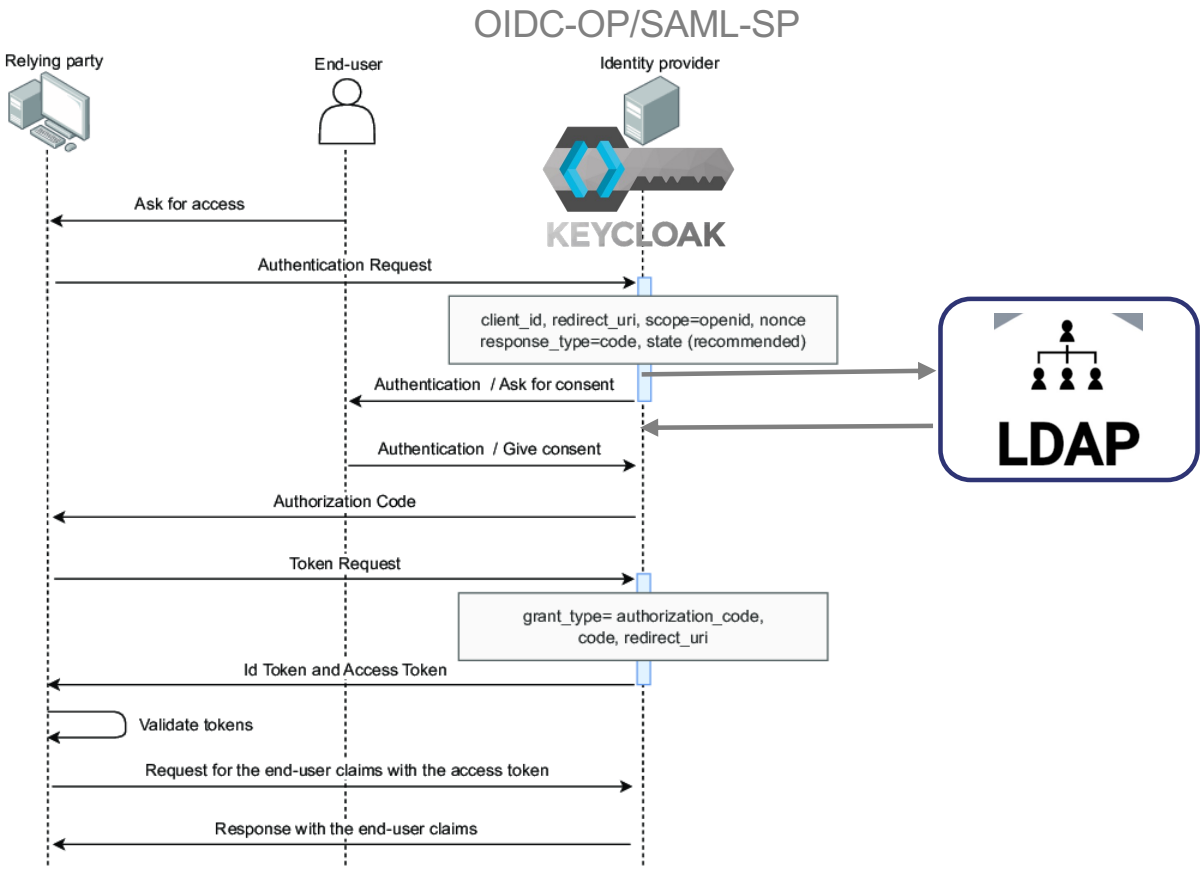
INFN-AAI OIDC Web

- Implementato con keycloak configurato come Identity Broker (proxy)
- Il keycloak funge da OP OIDC e SP SAML
- L'autenticazione è demandata all'IdP SAML attraverso redirect proprie di SAML
- Confidential: i RP OIDC vengono registrati in Keycloak e condividono un segreto con l'OP
- Multiple valid redirect URI per singolo RP
- Custom scope «aai» e custom claims



INFN-AAI OIDC CLI

- Implementato con keycloak in ulteriore REALM configurato in federazione con LDAP



Schema a blocchi e versioni

- Situazione attuale

Aggiornamento infrastruttura INFN-AAI

- La manutenzione di keycloak, usato solo come Identity Broker, ha un rapporto costi benefici molto sfavorevole
- Nuovo modulo OIDC per SimpleSAMLphp v. 2.3 con supporto per OpenID Federation 1.0
  - Ben supportato e già verificato
  - Possiamo integrare OIDC direttamente nel nostro IdP
- Aggiornamento Identity Linking a PHP 8
- Abbandoneremo il modulo proxy SPiD/CIE per SSP1.19 passando a Shibboleth-sp
  - SPiD/CIE Livello 2
  - eIDAS

- SPiD/CIE Livello 2 forniranno, tra l'altro, un «backup» per il 2FA
- In questa fase dovremo riconfigurare tutti i RP OIDC e farlo puntare all'IdP

Schema a blocchi e versioni

- Situazione attuale

- Prossimo passo

To Do

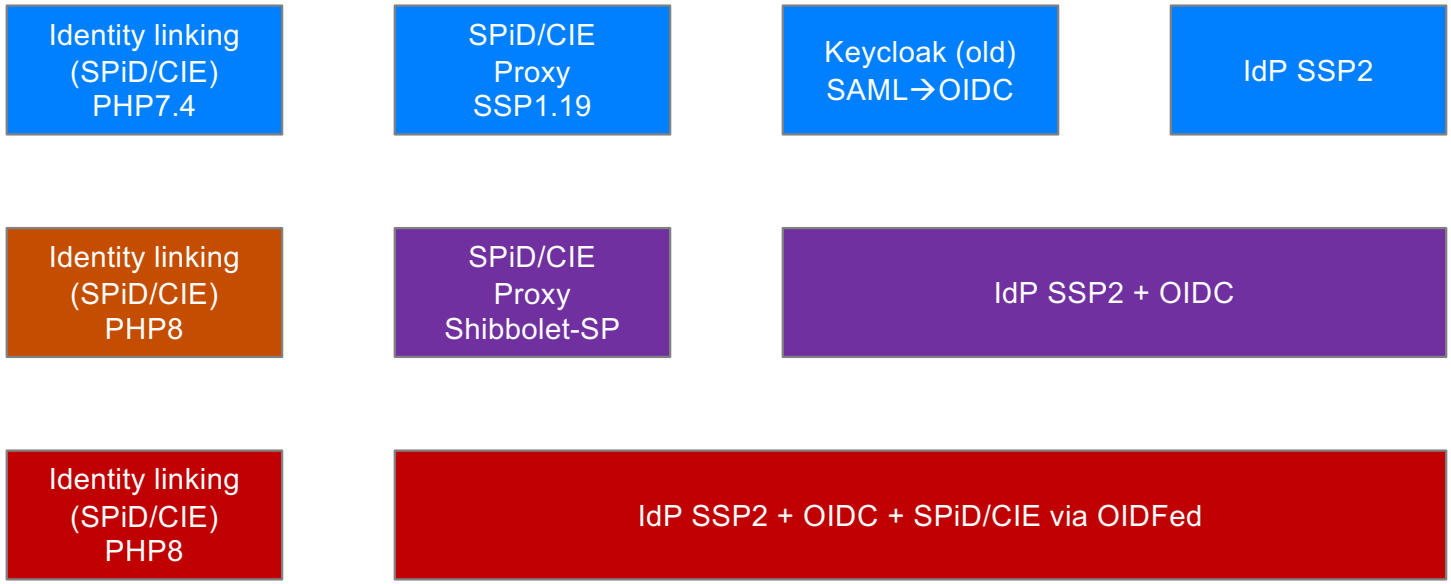
Ready in dev

OpenID Federation 1.0

- A differenza di quanto avviene per SAML, non è possibile «semplicemente» aggiungere metadati di SP via federazioni intese come «terzo fidato»
- OpenID Federation (OIDFed) è lo standard, definito in circa 140 pagine di specifiche, che aggiunge ad OIDC lo strato di federazione (come OIDC aggiunge lo strato di autenticazione ad OAuth2)
  - Gestione di relazioni di fiducia (trust chain, trust anchor)
  - Manifesti relativi alle caratteristiche dell'entità (entity statement)
  - Gestione delle deleghe e del consenso
- Maturo (draft 42, ma senza modifiche strutturali dal draft 22 in poi)
- Già in produzione in un paio di federazioni in Italia: AGiD (SPiD) e Ministero dell'Interno (CIE)
- Adozione in IDEM/eduGAIN: pilota previsto a partire dalla seconda metà di giugno 2025

Schema a blocchi e versioni

- Situazione attuale
- Prossimo passo
- To Do
- Ready in dev
- Step finale



The End

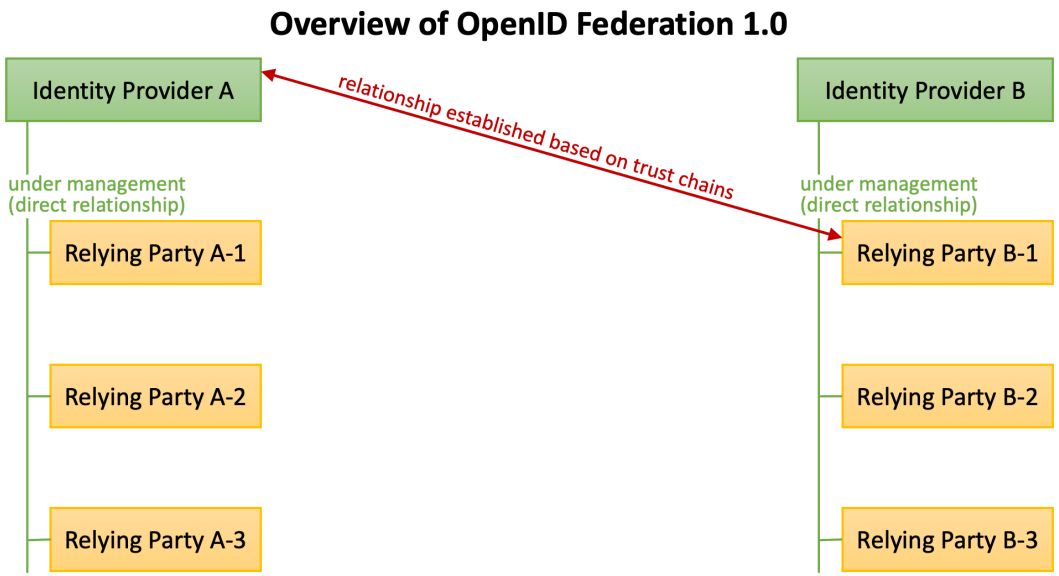
Grazie

Domande?

Catene di fiducia (Trust-Chain) -1

- Grazie alle *trust chain* identity provider / authorization server e un relying party (client) che non hanno una relazione diretta, possono fidarsi l'uno dell'altro.

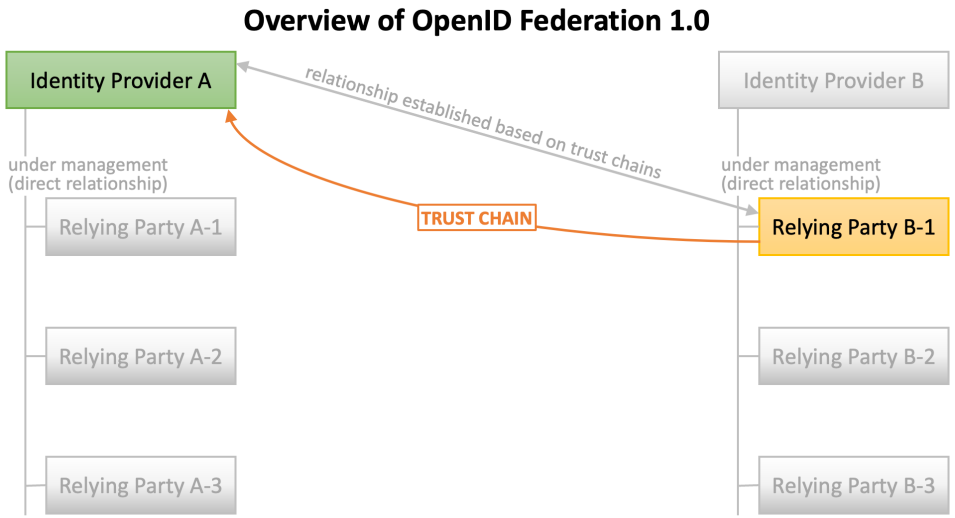
- L'identity provider accetta richieste OAuth/OIDC da parte del relying party senza che sia necessaria una registrazione preventiva di quest'ultimo.



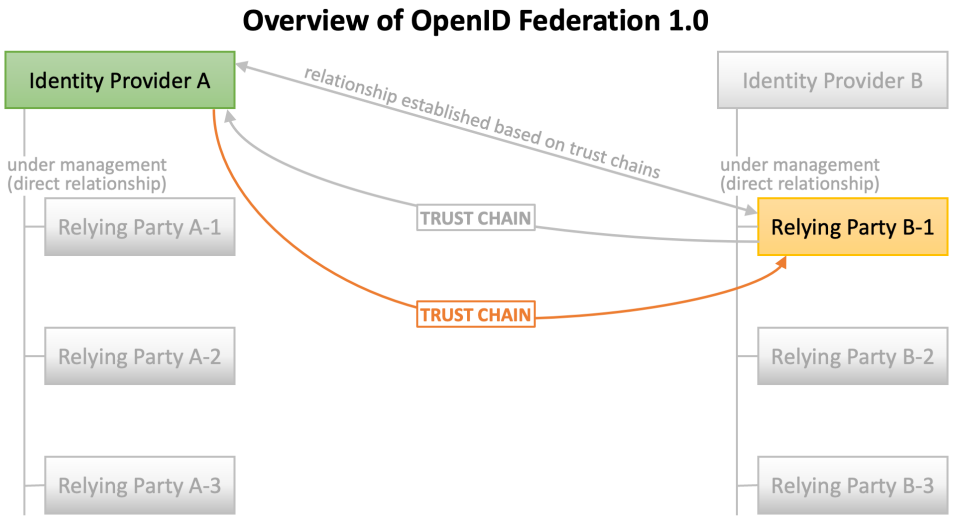
Catene di fiducia (Trust-Chain) - 2

- Le alle *trust chain* sono mono-direzionali. Questo significa che devono esser costruite due trust chain affinché un IdP ed un RP si fidino uno dell'altro.

- Il RP si fida dell'IdP

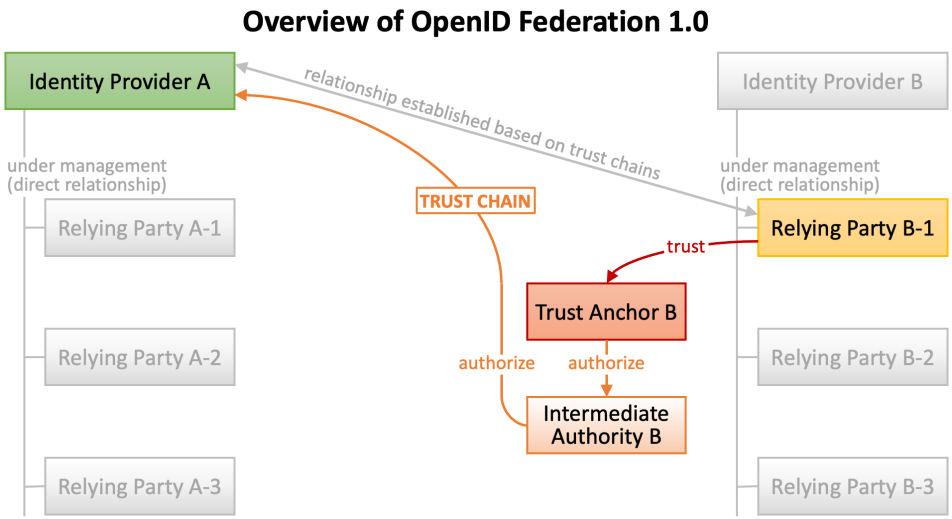


- L'IdP si fida dell'RP

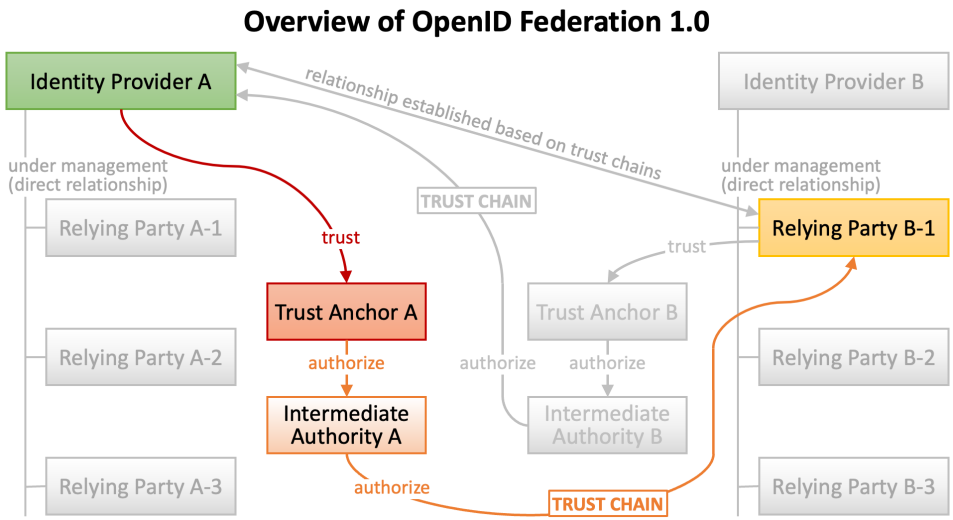


Catene di fiducia (Trust-Chain) - 3

- Un RP si fida direttamente di una o più *trust anchor*
- Se da una di queste è possibile arrivare ad un IdP allora l'RP può fidarsi dell'IdP

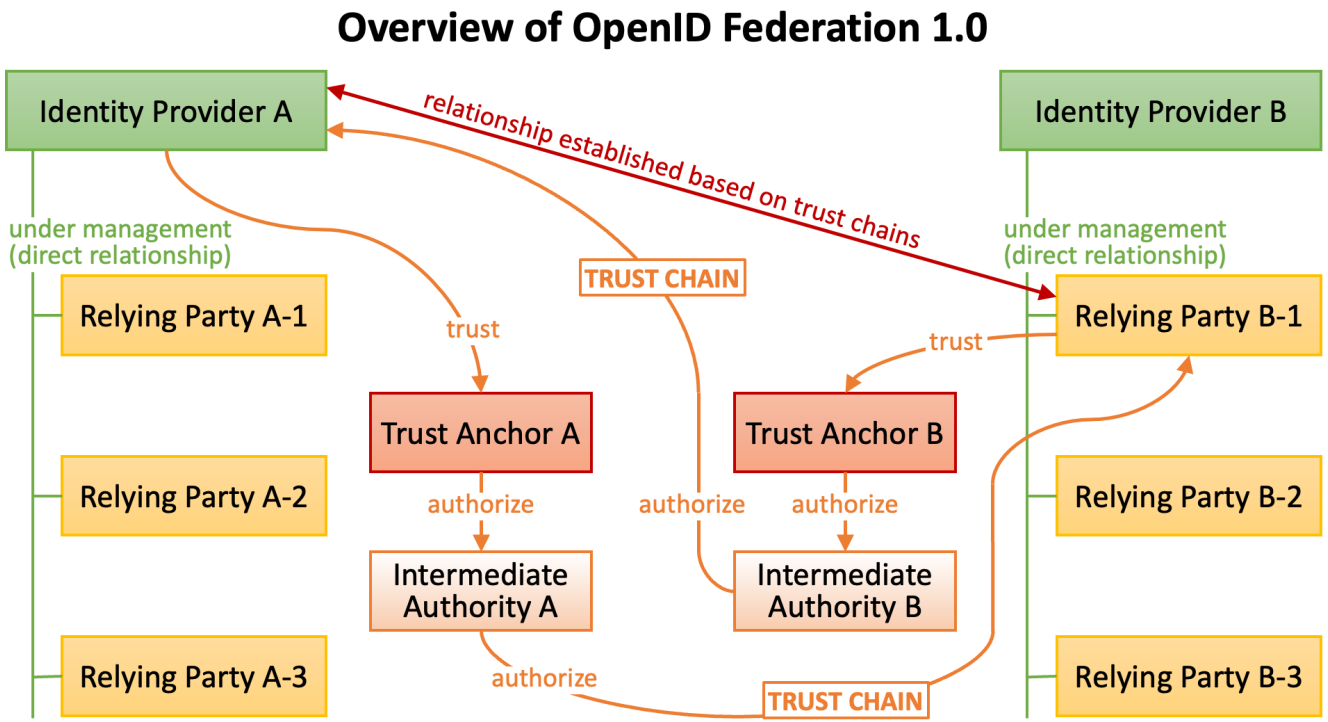


- L'IdP si fida direttamente di una o più *trust anchor*
- Se da una di queste è possibile arrivare ad un RP, allora l'Idp può fidarsi dell'RP



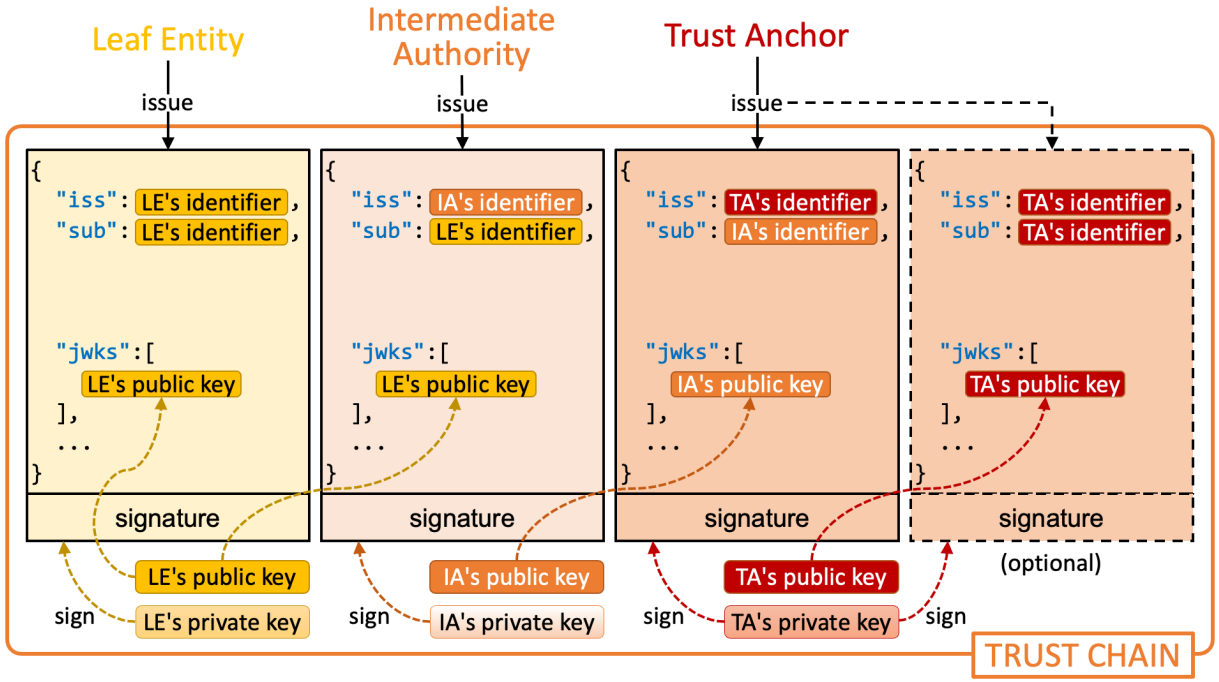
Catene di fiducia (Trust-Chain) - 4

- Mettendo insieme i due percorsi



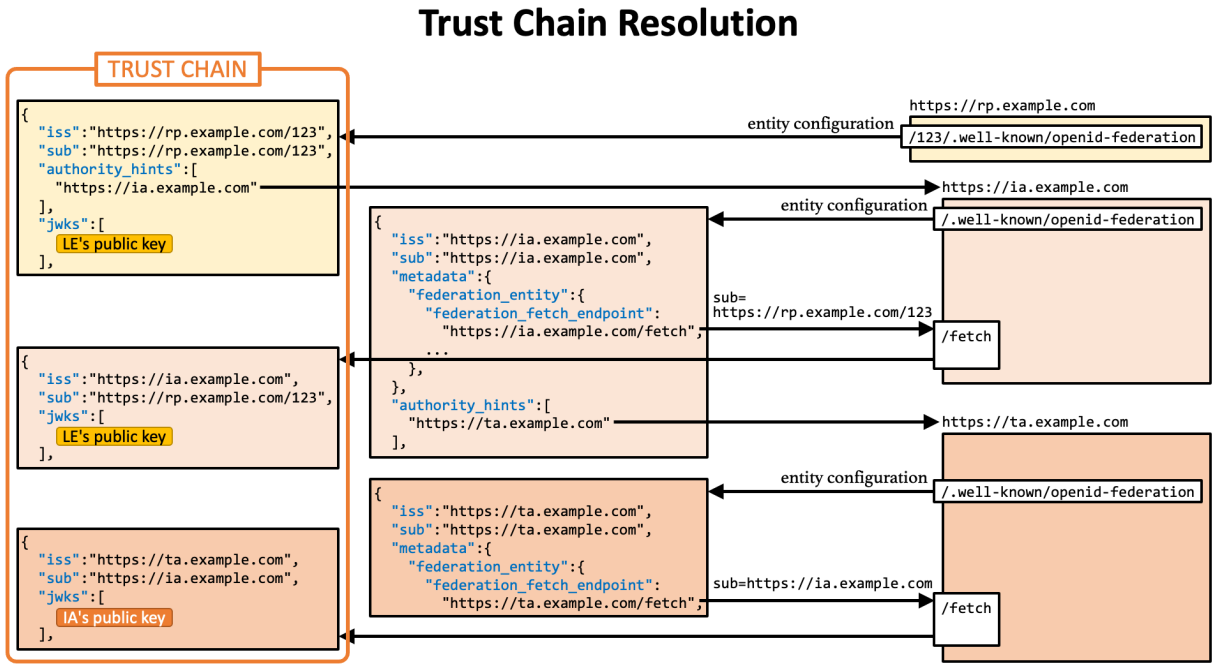
Dentro le trust chain

- Una trust chain è essenzialmente una sequenza di JWT che vanno dalla «Foglia» all' «ancora di fiducia» analogamente alla struttura delle catene di certification authorities di un certificato X.509
- Opzionale il JWT emesso dalla trust anchor, riferita a sè stessa e self-signed



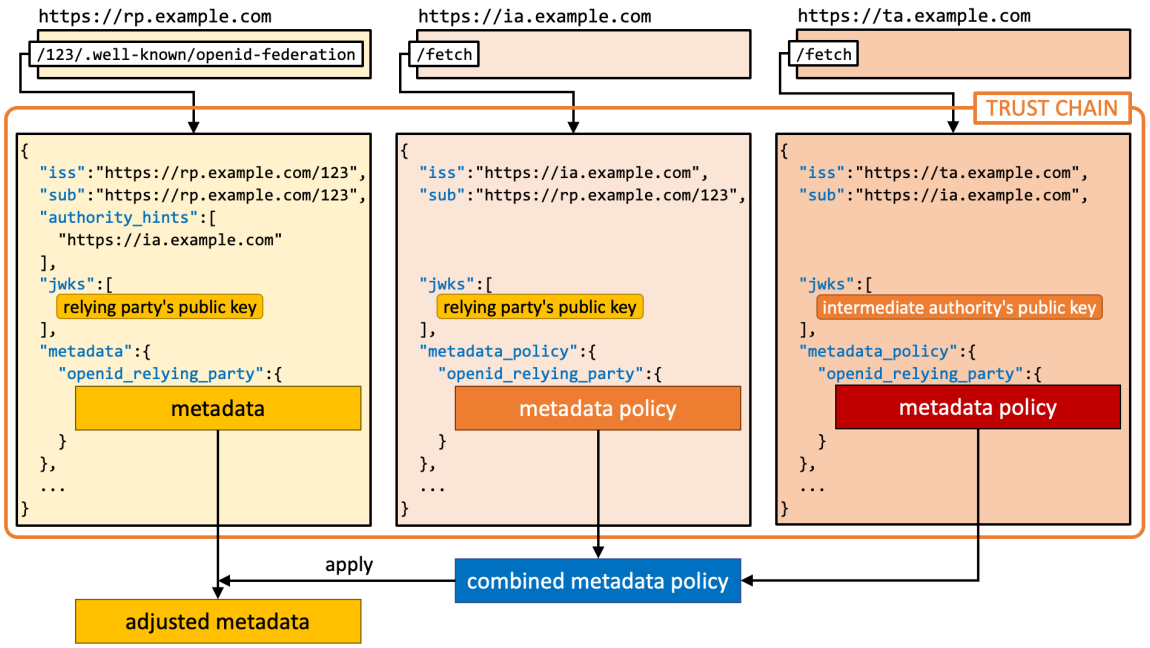
Costruzione di trust chain: resolution flow

- L'entity configuration della foglia (in questo caso un RP) espone l'indirizzo del'autorità intermedia via autority_hints
- L'autorità intermedia certifica il RP nei suoi metadati ed espone l'indirizzo della trust anchor
- La Trust anchor certifica l'autorità intermedia nei suoi metadati



Metadata policy

- Nell'entity configuration della leaf entity (il punto di partenza di una trust chain) c'è il claim metadata che contiene i metadati dell'entità
- I JWT emessi dalle autorità possono contenere metadata policies, all'interno del claim **`metadata_policy`**
- Tutte le policy sono combinate tra di loro prima del loro utilizzo
- Le policy sono combinate dovranno essere applicate ai metadati della leaf entity



Use case: Automatic client registration

- Se nei metatati dell'RP, rettificati con le metadata policy di intermediate authority/ies e trust anchor, è prevista la registrazione automatica