SPID OpenID Connect Federation

Regole tecniche per il funzionamento della Federazione SPID OpenID Connect

Sommario

Scopo 3
La Federazione OpenID Connect SPID 3
Entità della Federazione 4
Termini normativi 5
Acronimi 5
Termini e definizioni 5
Configurazione della Federazione SPID 7
Modalità di partecipazione alla Federazione 7
Modalità di riconoscimento e instaurazione della fiducia tra le parti 8
Relying Party 9
OpenID Provider 10
Modalità di accesso alla Entity Configuration 13
Trust Mark 13
Composizione dei Trust Mark 15
Federation_entity Trust Mark 17
Oauth_resource Trust Mark 18
Validazione dei Trust Mark 18
Revoca dei Trust Mark 18
Pubblicazione dei Trust Marks 19
Entity Statement e Configuration 19
Firma 20
Attributi (claim) 20
Metadata 22
OpenID Connect Provider Metadata 22
OpenID Connect Relying Party Metadata 25
Metadata di Trust Anchor (TA) e Intermediari (SA) 26
Attribute Authority Metadata 28
Metadata Policy 31
Metadata Policy di un TA per un RP 31

Metadata Policy di un TA per un SA 32
Metadata Policy di un SA per un RP 32
Metadata Policy di un TA per un OP 32
Soggetti Aggregatori 34
Endpoint di Federazione 35
Endpoint comuni a tutti 36
.well-known/openid-federation 36
Resolve Entity Statement endpoint 36
Endpoint per Trust Anchor ed Intermediari 37
Fetch entity statement endpoint 37
Trust mark status endpoint 37
Entity Listing endpoint 37
Gestione degli errori di federazione 37
Codici di errore di Federation 38
Algoritmi crittografici 38
Retention Policy 40
Gestione dei Log di un OP e di un RP 40
Registro storico delle chiavi pubbliche di Federazione 40
Differenze con OIDC Federation 1.0 41
Client Registration 41
Trust Mark 41
Claim non supportati negli Entity Statement 41
Considerazioni di Sicurezza 41
Trust Mark come deterrente contro gli abusi 42
Numero Massimo di authority_hints 42
Resolve Entity Statement 42
Buone Pratiche 42
Specializzare le chiavi pubbliche OpenID Core e Federation 43
Modalità di aggiornamento dei Metadata OpenID Core 43
Periodo di grazia per le Trust Chain scadute 43
Riferimenti Tecnici agli Standard 43

Storico modifiche e versioni

Date	*Autore*	*Versione*	*Modifiche*
14/09/2022	*AGID*	*1.0*	
31/03/2025	*AGID*	*2.0*	*Inserimento paragrafo "Federazione OpenID Connect SPID".* *Sostituzione di Metadata Discovery con Federation Entity Discover.* *Aggiornamento sezioni:* • *Revoca dei Trust Mark,* • *Entity Statement e Configuration, Metadata.* *Riscrittura completa della sezione Metadata Policy.* *Inserimento delle sezioni:* • *Federation_entity Trust Mark,* • *Oauth_resource Trust Mark,* • *Gestione degli errori di federazione,* • *Algoritmi crittografici,* • *Retention Policy.* *Eliminazione degli esempi finali.*

Scopo

Al fine di adempiere a quanto previsto nel Capitolo 3 delle Linee Guida OpenID Connect in SPID, per "mantenere e distribuire i metadata dal Registry SPID a tutti i soggetti della federazione" e consentire la configurazione dei rispettivi sistemi, questo documento definisce le regole di funzionamento della Federazione OpenID Connect SPID per Fornitori di Servizio pubblici e privati (RP), Identity Provider (OP), Attribute Authority (AA) e Soggetti Aggregatori (SA), e definisce gli schemi dei metadati di RP, OP, SA e AA in contesto Federativo, le modalità di registrazione dei RP presso gli OP, le risorse e gli endpoint a supporto della Federazione.

La Federazione OpenID Connect SPID

La Federazione OpenID Connect SPID è un'infrastruttura all'interno della quale tante organizzazioni, afferenti a domini differenti, aderiscono ad un medesimo quadro regolatorio per costruire un meccanismo di fiducia sia amministrativo (mediante la stipula di convenzioni e accreditamento presso una o più autorità super partes) che tecnologico (mediante l'adozione di standard di interoperabilità sicuri che consentono l'interscambio dei dati)

La Federazione stabilisce i livelli di garanzia e di sicurezza adeguati affinché un individuo possa autenticarsi presso un servizio web (Service Provider) mediante la propria identità digitale, rilasciata da un altro servizio web (Identity Provider).
Questo documento integra alcune parti delle LL.GG. OIDC SPID.
Affinché le parti si riconoscano all'interno della medesima Federazione delle identità è necessario che ognuna di queste ottenga la prova della reciproca aderenza ad un medesimo quadro regolatorio.
Le parti ottengono i metadati gli uni degli altri, contenenti le chiavi pubbliche per le operazioni di firma digitale e criptazione e le definizioni necessarie all'interscambio delle informazioni, secondo le regole prestabilite.
SPID adotta le specifiche di OpenID Connect (OIDC) Federation 1.0 [OIDC-FED] che definiscono come le entità, intese come partecipanti ad una Federazione, possono riconoscersi ed ottenere i metadati di Federazione e i metadati per il protocollo OpenID Connect [OpenID.Core].
I metadati sono certificabili da una parte fidata che all'interno della Federazione SPID è AgID e corrisponde alla Autorità di Federazione.
SPID implementa OpenID Connect Federation 1.0 ed estende alcune funzionalità dello standard, ne realizza una implementazione concreta e produce le buone pratiche per la sua adozione. Per approfondimenti allo standard si rimanda alle specifiche ufficiali [OIDC-FED][1] e alla sezione "*Differenze con OIDC Federation 1.0*".

Entità della Federazione

Le parti coinvolte all'interno di una Federazione OpenID Connect sono le seguenti:

Autorità di Federazione	Agenzia per l'Italia Digitale (AgID). Norma il funzionamento e le modalità di registrazione e riconoscimento dei partecipanti.
Trust Anchor	Sistema gestito dalla AgID il cui compito è quello di pubblicare la configurazione della Federazione e le affermazioni di riconoscimento delle parti che afferiscono alla Federazione. Il Trust Anchor corrisponde alla Autorità di Federazione e rappresenta la Federazione stessa.
Intermediario	Soggetto Aggregatore (SA), facilita l'ingresso nella Federazione e PUÒ gestire le funzionalità per conto di un suo discendente (Aggregato), pubblica la propria configurazione all'interno della Federazione e le affermazioni di riconoscimento delle parti sue discendenti (Aggregati) in conformità alle regole definite dalla AgID.
Foglia	Entità definita dal protocollo OIDC come Relying Party e Provider OpenID.
Entità	Partecipante alla Federazione. Trust Anchor, Intermediario o Foglia.

[1] https://openid.net/specs/openid-connect-federation-1_0.html

Termini normativi

Le parole chiave "DEVE" e "DEVONO", "NON DEVE" e "NON DEVONO", "RICHIEDE" e "RICHIESTO", "NON DEVE", "DOVREBBE", "NON DOVREBBE", "RACCOMANDATO", "PUÒ" e "OPZIONALE" nel presente documento devono essere interpretate come descritte nel BCP 14 [RFC2119] [RFC8174] quando e solo quando appaiono in maiuscolo.
Le notazioni [...] e ... indicano che il testo è stato troncato per esigenze editoriali.

Acronimi

In questa sezione sono definiti tutti gli acronimi utilizzati all'interno del testo.

AdF	Autorità di Federazione, che è AgID.
OIDC	OpenID Connect
OIDC-FED	OIDC Federation 1.0.
IOF	Italian OIDC Federation 1.0.
SPID	Sistema Pubblico per la gestione dell'Identità Digitale
AgID	Agenzia per l'Italia Digitale.
SA	Soggetti Aggregatori.
TA	OIDC Federation Trust Anchor.
OP	OpenID Provider.
RP	Relying Party.
AA	Attribute Authority, OAuth Resource Server, Gestore degli Attributi qualificati.
TM	Trust Mark.
EC	Entity Configuration.
ES	Entity Statement.
URL	Uniform Resource Locator, corrispondente ad un indirizzo web.
JWT	Vedi [RFC7519].

Termini e definizioni

In questa sezione descriviamo i termini utilizzati da [OIDC-FED#Section_1.2] e in questo documento

Entity configuration	Dichiarazione di una entità emessa per proprio conto, nella forma di JWT auto firmato [RFC7515] e contenente la configurazione di se stessa. Contiene le chiavi pubbliche di Federazione, il metadata OIDC, gli URL delle autorità sue superiori e i Trust Mark emessi da autorità riconoscibili nella Federazione che attestano l'aderenza del soggetto a determinati profili.
Entity statement	Dichiarazione di riconoscimento emessa da un'entità superiore (Trust Anchor o Intermediario) riguardante un'entità discendente (RP, OP o Intermediario) in formato JWT firmato [RFC7515], contenente la chiave pubblica del soggetto discendente, i Trust Mark emessi per i quali è emittente e la politica dei metadati da applicare ai metadati del soggetto.
Trust Mark	JWT firmato [RFC7515] dall'ente emittente e relativo ad un partecipante. Attesta la conformità di questo ai profili riconoscibili all'interno della Federazione (RP pubblico o privato, Soggetto Aggregatore Pubblico o Privato, etc.). La Foglia che acquisisce il marchio di fiducia durante la fase di onboarding deve includere questo nella sua Entity Configuration a mò di *Badge* di riconoscimento.
Metadata	Documento che descrive una implementazione di una entità OpenID Connect. Le implementazioni di ogni Entità condividono i metadati per stabilire una base di fiducia e interoperabilità.
Metadata policy	Il Trust Anchor pubblica le regole e le politiche da applicare sui metadata dei discendenti, specificando quali valori o sottoinsiemi di valori sono consentiti per un dato attributo di metadati.
Authority hint	Un *Array* di valori url corrispondenti agli identificativi delle entità superiori, Trust Anchor o Intermediario, che emettono un Entity Statement per i propri discendenti.
Federation Entity Discovery	Raccolta di Entity Configuration e Statement. Inizia da un'entità Foglia fino al raggiungimento del Trust Anchor.
Trust Chain	Procedura di validazione della sequenza di Entity Configuration e Statement raccolta mediante Federation Entity Discovery, il cui esito positivo è un metadata finale relativo ad una entità e la data di scadenza entro la quale questo deve essere aggiornato.
onboarding	Procedura di registrazione di una nuova entità all'interno della Federazione SPID

entity-type	Tipologie di metadata validi all'interno dell'Entity Configuration: • openid_relying_party • openid_provider • federation_entity • oauth_authorization_server • trust_mark_issuer • oauth_resource
entity-role	Tipologie di ruoli validi all'interno degli identificativi di Trust Mark: • openid_relying_party • openid_provider • intermediate • oauth_resource

Configurazione della Federazione SPID

La configurazione della Federazione SPID è pubblicata dal Trust Anchor all'interno della sua Entity Configuration[2], presso un web path ben noto e corrispondente a **.well-known/openid-federation**. Tutti i partecipanti DEVONO ottenere prima della fase di esercizio la configurazione della Federazione e mantenere quest'ultima aggiornata su base giornaliera. All'interno della Configurazione della Federazione è presente la chiave pubblica del Trust Anchor usata per le operazioni di firma, il numero massimo di Intermediari consentiti tra una Foglia e il Trust Anchor (**max_path length**) e le autorità abilitate all'emissione dei Trust Marks (**trust_mark_issuers**).
Si veda la Sezione dedicata alle Entity Configuration per ulteriori dettagli.

Modalità di partecipazione alla Federazione

Per aderire alla Federazione SPID una entità di tipo Foglia DEVE pubblicare la propria configurazione (Entity Configuration) presso il web endpoint **.well-known/openid-federation**[3].
Gli incaricati tecnici ed amministrativi della Foglia completano la procedura amministrativa per la registrazione di una nuova entità o l'aggiornamento di una preesistente definita dalla Autorità di Federazione o da un suo Intermediario (SA).
L'Autorità di Federazione o un suo Intermediario dopo aver effettuato tutti i controlli amministrativi e tecnici richiesti, registra le chiavi pubbliche della Foglia e rilascia una prova di adesione alla Federazione sotto forma di Trust Mark (TM).

[2] Esempio di Entity Configuration di un Trust Anchor
[3] Esempio di Entity Configuration request e response

La Foglia DEVE includere il TM all'interno della propria configurazione di Federazione (Entity Configuration) come prova del buon esito del processo di onboarding.
L'Autorità di Federazione o un suo Intermediario DEVE pubblicare la dichiarazione di riconoscimento della Foglia (Entity Statement) contenente le chiavi pubbliche di federazione della Foglia e i TM a questa rilasciati. L'Autorità di Federazione o un suo Intermediario PUÒ pubblicare una politica dei metadata[4] per forzare la modifica dei metadata OIDC della Foglia, nelle parti in cui questo sia necessario.

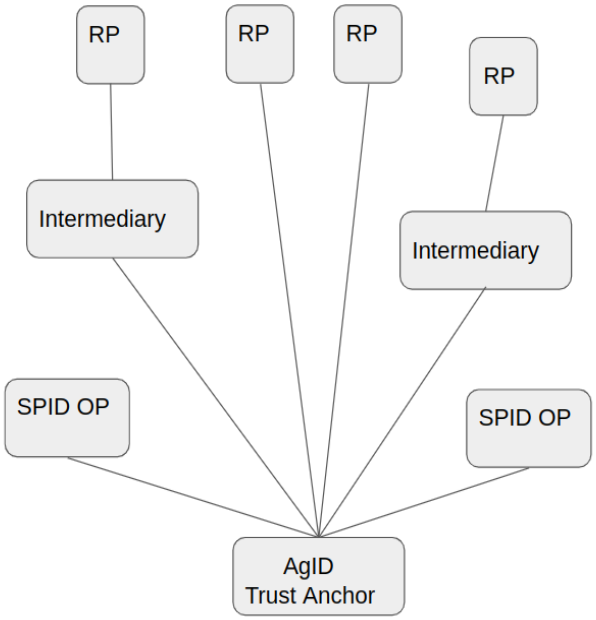

Figura 1: *Schema ad albero che rappresenta la struttura della Federazione. Alla Base l'Autorità di Federazione e a salire gli OP che non hanno intermediari, gli RP e gli Intermediari che a loro volta Aggregano altri RP.*

Modalità di riconoscimento e instaurazione della fiducia tra le parti

In questa sezione vi sono illustrate le modalità di mutuo riconoscimento tra RP e OP, le modalità con le quali le Foglie della Federazione SPID si riconoscono all'interno della medesima Federazione e ottengono le une i metadata delle altre.

[4] https://openid.net/specs/openid-connect-federation-1_0.html#rfc.section.5.1

Relying Party

Il RP ottiene la lista degli OP in formato JSON interrogando l'**endpoint list** disponibile presso il Trust Anchor[5]. Per ogni soggetto contenuto nella risposta[6] dell'**endpoint list** e corrispondente ad un OP, il RP richiede[7] ed ottiene l'Entity Configuration *self-signed* presso l'OP.
Per ogni EC degli OP, il RP verifica la firma del contenuto adoperando la chiave pubblica ottenuta dall'Entity Statement rilasciato dalla Trust Anchor.
Verificata la firma dell'Entity Configuration con la chiave pubblica pubblicata dalla Trust Anchor la fiducia è stabilita nei confronti del OP da parte del RP.
Il RP applica infine le politiche pubblicate dal Trust Anchor sui metadata del OP e salva il metadata finale associandolo ad una data di scadenza (claim **exp**). La data di scadenza corrisponde al valore di **exp** più basso ottenuto da tutti gli *statement* che compongono la **Trust Chain**. Periodicamente il RP aggiorna i metadati di tutti gli OP rinnovando la Trust Chain relativa a questi.
Ottenuti i metadati finali di tutti i Provider SPID, il RP genera lo SPID Button e lo pubblica all'interno della pagina di autenticazione destinata agli utenti.
La procedura di Federation Entity Discovery risulta semplificata per i RP SPID perché non è consentita all'interno della Federazione l'esistenza di Intermediari tra gli OP ed il loro Trust Anchor.

[5] Esempio di List endpoint request

[6] Esempio di List endpoint response

[7] Esempio di Entity Statement request

OpenID Provider

Quando un Provider (OP) riceve una richiesta di autorizzazione da parte di un RP non precedentemente riconosciuto avviene la procedura di **automatic client registration**. Sono di seguito descritte le operazioni compiute dal OP per registrare un RP dinamicamente.

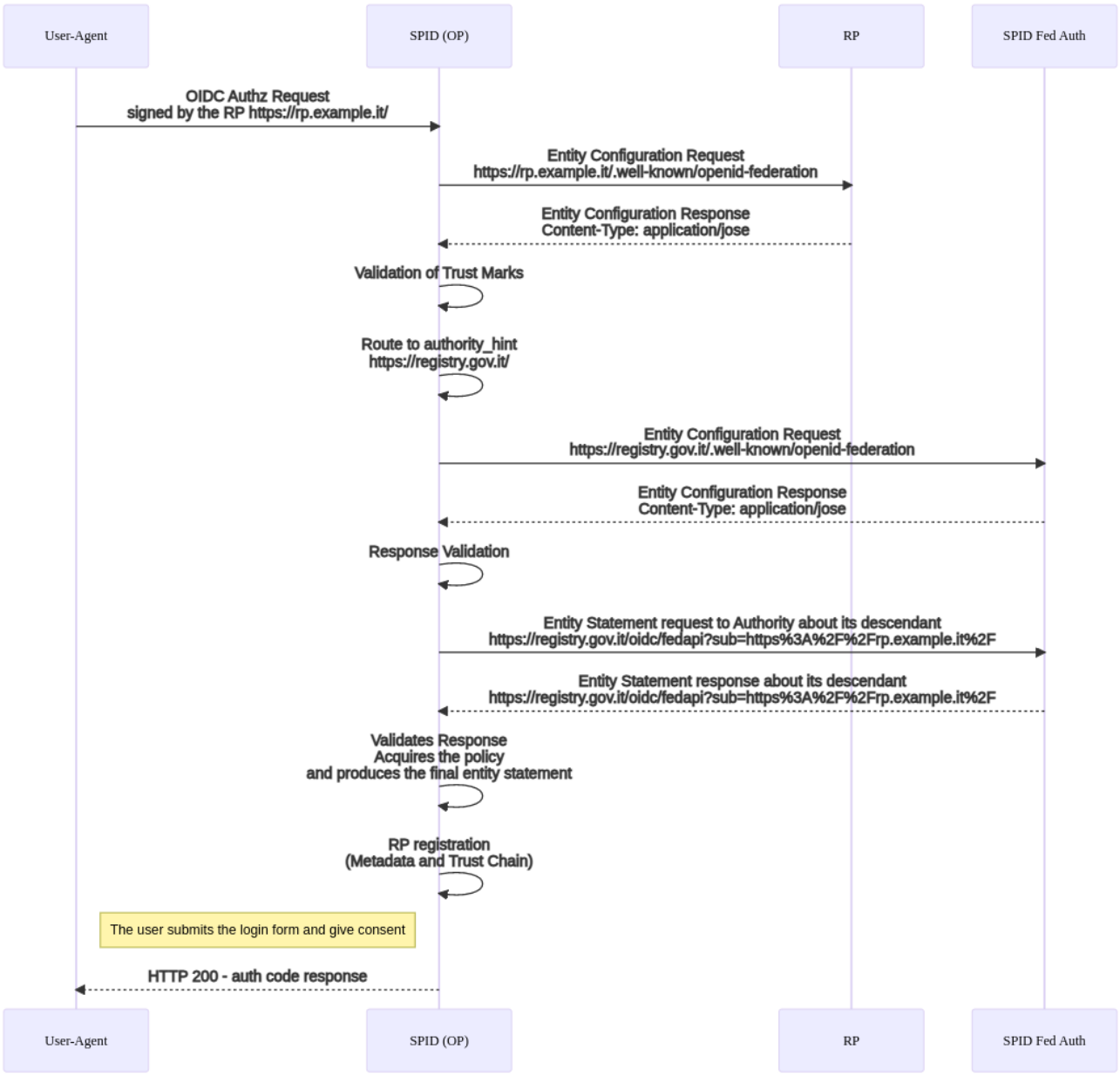

Figura 2: La registrazione di un RP dalla prospettiva di un OP che per la prima volta riceve una richiesta di autorizzazione dal RP e avvia il processo di Federation Entity Discovery e salvataggio della Trust Chain.

L'OP estrae l'identificativo univoco (**client_id**) dall'oggetto *request* contenuto all'interno della *Authorization Request* ed effettua una richiesta di Entity Configuration presso il RP[8]. Ottiene la configurazione *self-signed* del RP e convalida la firma dei Trust Marks riconoscibili all'interno della Federazione[9].
Se il RP non espone all'interno della sua configurazione nessun Trust Mark riconoscibile per il profilo di RP (vedi Sezione Trust Mark) il Provider DEVE rifiutare l'autorizzazione con un messaggio di errore di tipo *unauthorized_client* conforme alla Linee Guida OpenID Connect SPID.
Se il Provider convalida con successo almeno un Trust Mark per il profilo RP contenuto all'interno della configurazione del RP richiedente, estrae le entità superiori contenute nel claim **authority_hints** ed avvia la fase di Federation Entity Discovery. Ne consegue il calcolo della **Trust Chain** e l'ottenimento del metadata finale.
Durante il Federation Entity Discovery, il Provider richiede ad una o più entità superiore[10] l'Entity Statement relativo al RP e ottiene la chiave pubblica con la quale valida la configurazione del RP, fino a giungere al Trust Anchor.
Infine, applica la politica dei metadata pubblicata dal Trust Anchor e salva il risultante metadata finale del RP associandolo ad una data di scadenza, oltre la quale rinnoverà il metadata secondo le modalità di rinnovo della Trust Chain.

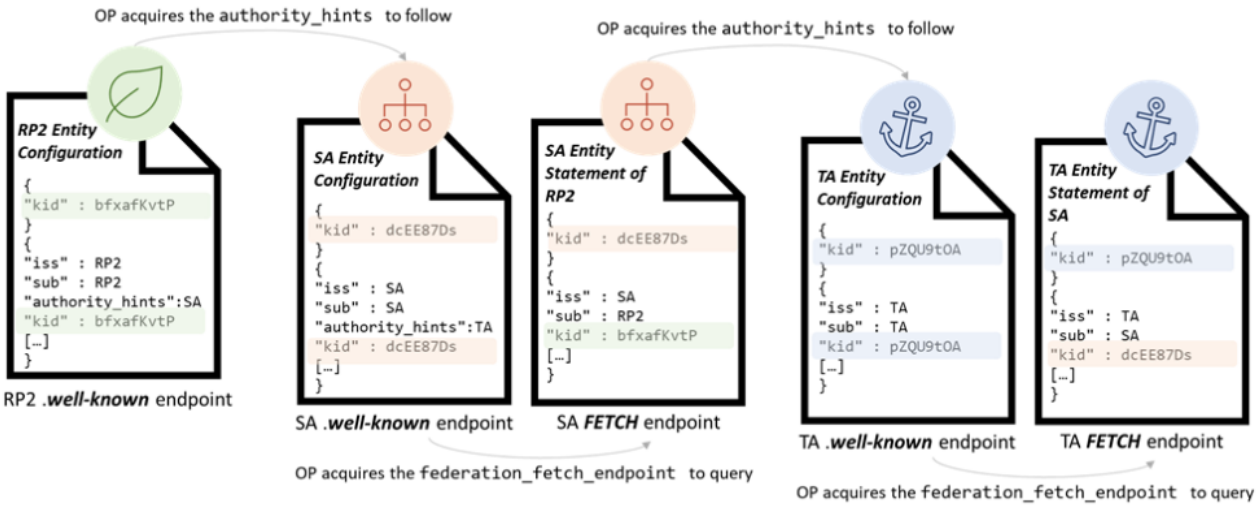

Figura 3: La procedura di Federation Entity Discovery a partire dalla Foglia fino al Trust Anchor. Si noti come dall'Entity Statement rilasciato da un superiore si ottiene la chiave pubblica per la validazione dell'Entity Configuration dell'entità discendente.

Ottenuto il metadata finale, il Provider valida la richiesta del RP secondo le modalità definite all'interno delle Linee Guida OpenID Connect SPID.

[8] Entity Configuration di un RP

[9] I Trust Mark di Federazione sono configurati nel claim **trust_mark_issuers** e contenuti nell'Entity Configuration del Trust Anchor.

[10] Un RP può esporre più di una entità superiore all'interno del proprio claim di **authority_hints**. Si pensi ad un RP che partecipa sia alla Federazione SPID che a quella CIE. Inoltre un RP può risultare come aggregato di molteplici intermediari, se questi SPID o CIE.

Nei casi in cui un RP avesse come entità superiore un SA e non direttamente la TA, la procedura di acquisizione e validazione dell'Entity Configuration del RP avviene mediante l'Entity Statement pubblicato dal SA nei confronti del RP e mediante la convalida dell'Entity Configuration del SA con l'Entity Statement emesso dalla TA in relazione al SA.
Se la soglia del massimo numero di intermediari verticali, definita dal valore di **max_path_length,** venisse superata, l'OP blocca il processo di Federation Entity Discovery e rigetta la richiesta del RP.

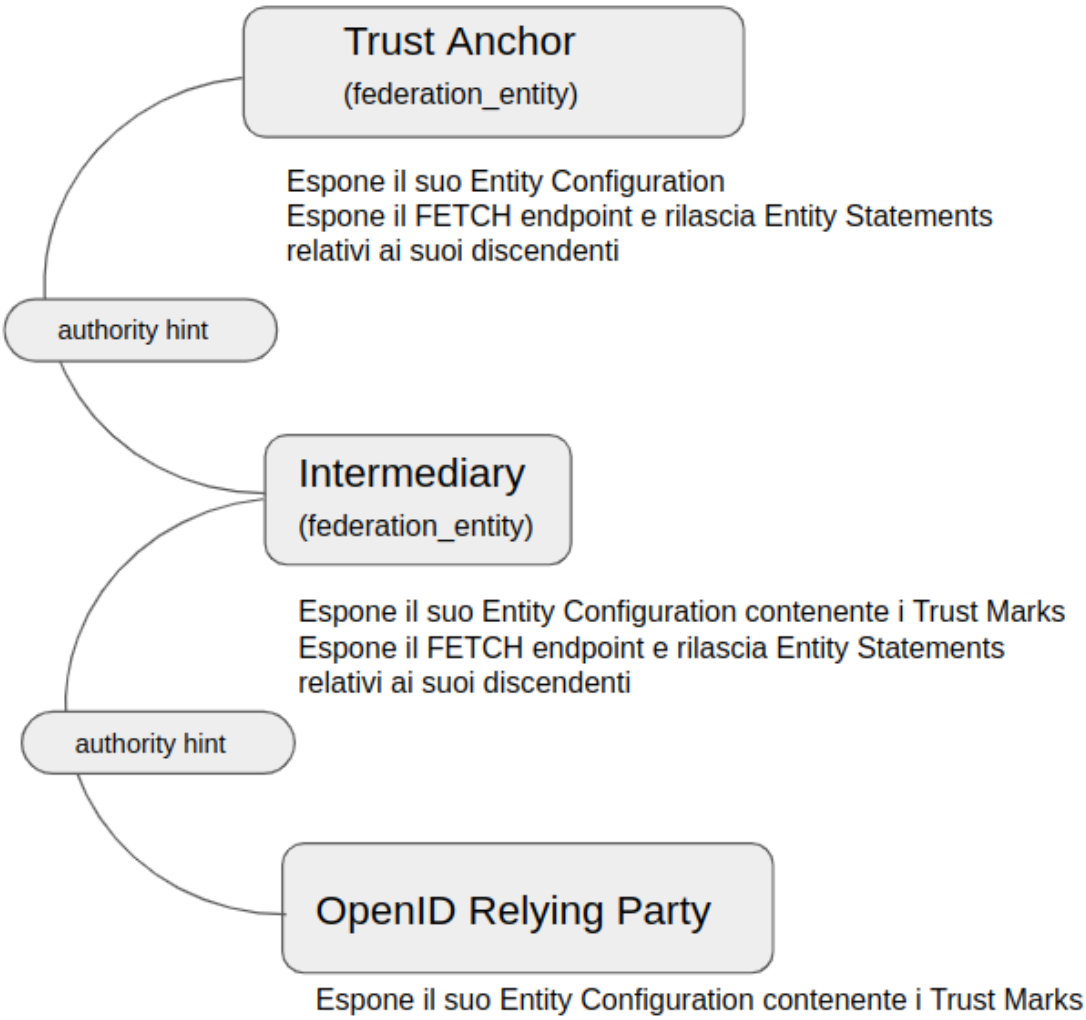

Figura 4: Ogni partecipante espone la propria configurazione e i propri Trust Mark. Il collegamento tra una Foglia e il Trust Anchor avviene in maniera diretta oppure mediante un Intermediario (Soggetto Aggregatore) come in Figura.

Modalità di accesso alla Entity Configuration

In questa sezione viene descritto come individuare per un determinato soggetto l'URL [RFC3986] per il download della Entity Configuration.

La risorsa attraverso la quale un partecipante pubblica la sua configurazione (Entity Configuration) corrisponde al webpath **.well-known/openid-federation** e DEVE essere appesa all'URL che identifica il soggetto.
Esempi:

- con identificativo del soggetto pari a **https://rp.example.it** il risultante URL di Entity Configuration è **https://rp.example.it/.well-known/oidc-federation**.
- con identificativo del soggetto pari **https://rp.servizi-spid.it/oidc/** il risultante URL di Entity Configuration è **https://rp.servizi-spid.it/oidc/.well-known/oidc-federation**.

Se l'URL che identifica il soggetto non presenta il simbolo di slash finale ("/") è necessario aggiungerlo prima di appendere il web path della risorsa **well-known**.

Trust Mark

I Trust Mark, letteralmente tradotti come marchi di fiducia, sono oggetti JSON firmati in formato Jose [RFC7515] e rappresentano la dichiarazione di conformità a un insieme ben definito di requisiti di fiducia e/o di interoperabilità o un accordo tra le parti coinvolte all'interno della Federazione. I Trust Marks sono rilasciati principalmente durante il processo di registrazione di una nuova entità di tipo Foglia (onboarding) dal Trust Anchor o suoi Intermediari.
Lo scopo principale di questi marchi di fiducia è quello di esporre alcune informazioni non richieste dal protocollo OpenID Connect Core ma che risultano utili in contesto Federativo. Esempi tipici includono il codice di identificazione nazionale o internazionale dell'entità[11], i contatti istituzionali e altro. Ulteriori dati possono essere aggiunti dall'emittente se resi comprensibili all'interno della Federazione.
I Trust Marks riconoscibili all'interno della Federazione SPID sono emessi e firmati dalla AgID (TA) o suoi intermediari (SA) o dai Gestori di attributi qualificati (AA) se definiti all'interno del claim **trust_mark_issuers** pubblicato all'interno dell'Entity Configuration del TA. Ogni partecipante DEVE esporre nella propria configurazione (EC) i Trust Mark rilasciati dalle autorità emittenti.
I Trust Mark rappresentano il primo filtro per l'instaurazione della fiducia tra le parti, sono elementi indispensabili per avviare la risoluzione dei metadati. In loro assenza una entità non è riconoscibile come partecipante all'interno della Federazione SPID.
All'interno della Federazione SPID i Trust Mark presentano degli identificativi univoci (claim **id**) in formato URL che adottano la seguente struttura:

https:// <domain> / <entity_type> / [<trustmark_profile> /] [estensione /][12]

Alcuni esempi non normativi sono di seguito riportati:

- profilo RP public: *https://registry.spid.gov.it/openid_relying_party/public/*
- profilo SA private di tipo full o light: *https://registry.spid.gov.it/federation_entity/private/*
- profilo AA: *https://registry.spid.gov.it/oauth_resource/public/*

[11] Codice Fiscale, IPA Code, Partita IVA, VAT Number, etc.

[12] La notazione indica che l'elemento è opzionale e non obbligatorio.

Composizione dei Trust Mark

I claim definiti all'interno dei Trust Marks aderiscono a quanto definito all'interno dello standard **OIDC Federation 1.0**. Questi sono di seguito riportati.

Claim	Tipo	Descrizione
iss	String	RICHIESTO. URL che identifica univocamente l'Autorità che lo ha emesso.
sub	String	RICHIESTO. URL che identifica univocamente il soggetto per il quale il Trust Mark è stato emesso.
id	String	RICHIESTO. Identificativo univoco del Trust Mark.
iat	UTC Timestamp	RICHIESTO. Quando è stato emesso questo marchio di fiducia. Espresso come "*Seconds Since the Epoch*" [RFC7519].
logo_uri	String	OPZIONALE. Un URL che punta al logo rappresentante il Trust Mark.
exp	UTC Timestamp	RICHIESTO. Momento oltre il quale non sarà più valido. Espresso come "*Seconds Since the Epoch*" [RFC7519]
ref	String	OPZIONALE. URL che punta a informazioni presenti sul web relative a questo marchio di fiducia

La tabella seguente definisce gli <entity_role> riconoscibili all'interno delle Federazioni SPID:

Tipo	Descrizione	Entità
openid_relying_party	l'entità nel claim *sub* è un RP.	RP
openid_provider	l'entità nel claim *sub* è un OP.	OP
intermediate	l'entità nel claim *sub* è un Soggetto Aggregatore.	SA
oauth_resource	l'entità nel claim *sub* è una Attribute Authority.	AA

La seguente tabella riassume i profili di TM supportati all'interno della Federazione SPID e identificati dal claim "id".

Tipo	Descrizione	Entità
public	Indica che il RP l'entity statement afferisce ad una Pubblica Amministrazione.	RP, OP, SA, AA
private	Indica che il RP l'entity statement afferisce ad una organizzazione privata.	RP, OP, SA, AA

Agli attributi dei TM definiti nella tabella precedente, i Trust Mark SPID aggiungono i seguenti.

Claims	Description
organization_type	RICHIESTO. Specifica se l'ente appartiene alla pubblica amministrazione italiana o al settore privato (private or public).
id_code	Oggetto JSON. Contiene uno o più codici di identificazione dell'organizzazione. I claim disponibili sono: - **ipa_code**: OBBLIGATORIO nel caso di organizzazione pubblica. - **aoo_code**: OPZIONALE. - **uo_code**: OPZIONALE. - **vat_number**: OBBLIGATORIO per organizzazione privata se non presente fiscal_number. - **fiscal_number**: OBBLIGATORIO per organizzazione privata se non presente vat_number.
email	RICHIESTO. Email istituzionale o PEC dell'organizzazione.
organization_name	RICHIESTO. Il nome completo dell'entità che fornisce i servizi.
sa_profile	RICHIESTO per SA. Specifica il profilo dell'Aggregatore, esempio: **full** o **light**.

Quello che segue è un esempio non normativo di un marchio di fiducia emesso da AgID per un SA privato di tipo full.

```
"trust_marks": [
 {
  "id":"https://registry.spid.gov.it/federation_entity/private/",
  "trust_mark": "…"
 }
]
```

Dove il contenuto del JWT firmato all'interno del claim **trust_mark** corrisponde a:

```
{
  "id":"https://registry.spid.gov.it/federation_entity/private/",
  "iss": "https://registry.spid.gov.it",
  "sub": "https://intermediate.example.it",
  "iat": 1579621160,
  "organization_type": "private",
  "sa_profile": "full",
  "id_code": "12345678900",
  "email": "email_or_pec@example.it",
  "organization_name": "Full name of the SA",
  "ref": "https://reference_to_some_documentation.example.it/"
}
```

Un'entità intermediaria (SA) è riconoscibile come emittente di Trust Mark. Quello che segue è un esempio non normativo di un Trust Mark emesso da un Soggetto Aggregatore a favore di un RP suo discendente.

```
"trust_marks": [
 {
  "id":"https://registry.spid.gov.it/openid_relying_party/public/",
  "trust_mark": …
  }
]
```

Dove il contenuto del JWT firmato all'interno del claim **trust_mark** corrisponde al seguente esempio non normativo.

```
{
  "id":"https://registry.spid.gov.it/openid_relying_party/public/",
  "iss": "https://intermediate.example.it",
  "sub": "https://rp.example.it",
  "iat": 1579621160,
  "organization_type": "public",
  "id_code": "123456",
  "email": "email_or_pec@rp.it",
  "organization_name": "Full name of the RP",
  "ref": "https://reference_to_some_documentation.it/"
}
```

Federation_entity Trust Mark

In aggiunta ai claim dei profili public e private, il profilo intermediate individua i SA e aggiunge le estensioni full e light all'interno del claim sa_profile, a seconda della modalità con cui operano rispetto ai Soggetti Aggregati.

Oauth_resource Trust Mark

In aggiunta ai claim dei profili public e private, il profilo oauth_resource individua le AA e aggiunge i seguenti claim obbligatori:

CLAIM	DESCRIZIONE
policy_uri	URL dove è disponibile la privacy policy dell'AA.
tos_uri	URL dove è disponibile la info policy dell'AA.
claims	Lista di JSON Object che definiscono gli attributi dell'utente richiesti dall'AA. Esempio: {"https://attributes.eid.gov.it/fiscal_number":{"essential":true}, "email":{"essential":true},}
service_documentation	URL dove è disponibile il documento OAS3 che descrive il funzionamento dei servizi dell'AA.

Validazione dei Trust Mark

Esistono due modi per validare un Trust Mark:

1. Validazione **statica**. Il Trust Mark viene validato mediante il certificato pubblico dell'autorità che lo ha emesso (claim **iss**), sulla base della corrispondenza del claim **sub** con il medesimo claim della Entity Configuration in cui è contenuto e sulla base del valore di scadenza (claim **exp**).
2. Validazione **dinamica**. I partecipanti della federazione possono interrogare l'endpoint *trust mark status* erogato dal suo emittente (claim **iss**) per la verifica in tempo reale dei TM da lui emessi.

Tutti gli emittenti di Trust Mark DEVONO esporre un endpoint di *trust mark status* per consentire la validazione **dinamica**.

Revoca dei Trust Mark

Un Trust Mark può essere revocato in qualsiasi momento. In caso di esclusione di un Soggetto Aggregato da parte della Autorità di Federazione, questa comunica al Soggetto Aggregatore l'esclusione dell'Aggregato. Di conseguenza il SA revoca il TM per il suo discendente.
Nel caso di revoca di un TM, la validazione dinamica darà esito negativo, mentre la validazione statica continuerà a dare esito positivo, a meno di rotazioni delle chiavi crittografiche di firma del soggetto che ha rilasciato il TM.

Pubblicazione dei Trust Marks

La TA definisce i TM e gli emittenti di questi abilitati nella Federazione mediante il claim **trust_mark_issuers**, presente all'interno del proprio Entity Configuration. Il valore del claim **trust_mark_issuers** è composto da un oggetto JSON, avente come chiavi gli id dei TM e come valori la lista degli emittenti abilitati.

Di seguito un esempio non normativo dell'oggetto **trust_mark_issuers** all'interno della Entity Configuration del TA.

```
"trust_mark_issuers":{
    "https://registry.spid.gov.it/openid_relying_party/public/":[
      "https://registry.spid.gov.it/",
      "https://public.intermediate.spid.it/"
    ],
    "https://registry.spid.gov.it/openid_relying_party/private/":[
      "https://registry.spid.gov.it/",
      "https://private.other.intermediate.it/"
    ],
    “https://deleghedigitali.gov.it/openid_relying_party/sgd/”: [
      “https://deleghedigitali.gov.it”
    ]
  }
```

I TM emessi per le Foglie DEVONO essere pubblicati dalle stesse all'interno della propria Entity Configuration, all'interno del claim **trust_marks**. Questo è composto da lista di oggetti JSON, ognuno di questi DEVE contenere almeno i **claim id** e **trust_mark**, il primo identifica il TM, il secondo contiene il JWT firmato del TM.

Entity Statement e Configuration

Un Entity Configuration (EC) è un Metadata di federazione in formato Jose e firmato dal soggetto che lo emette e riguardante se stesso, all'interno del quale i valori dei claim **iss** e **sub** contengono il medesimo valore (URL).
Un Entity Statement (ES) è un documento di riconoscimento che una Autorità di Federazione o suo Intermediario emette per uno specifico soggetto, suo discendente, individuato all'interno del claim **sub**.
L'Entity Statement, un JWT firmato che contiene le chiavi pubbliche delle entità discendenti e ulteriori dati usati per controllare il processo di risoluzione della Trust Chain.

Una entità rende pubblico un ES relativo ad un suo discendente presso il proprio Fetch Endpoint. L'entità superiore PUÒ definire le policy sui metadata dei soggetti discendenti e pubblicare i TM da lei emessi per questi.

Firma

Tutte le operazioni di firma relative agli Entity Statements, Entity Configuration e Trust Mark sono condotte con le chiavi pubbliche di Federazione[13].
Per quanto riguarda gli algoritmi di firma supportati si veda la Sezione Algoritmi Crittografici.
Distinguiamo le chiavi di Federazione da quelle di OIDC Core. Queste ultime risiedono nei Metadata OIDC. Un EC contiene sia le chiavi pubbliche di Federazione che i Metadata OIDC. Le chiavi di Federazione DOVREBBERO essere diverse da quelle di OIDC Core.

Attributi (claim)

Entity Configuration e Statement presentano i seguenti claim comuni:

Nome	**tipo**	**descrizione**
iss	String	RICHIESTO. Identificativo dell'entità che lo emette.
sub	String	RICHIESTO. Identificativo del soggetto a cui è riferito.
iat	Unix Timestamp	RICHIESTO. Data di emissione.
exp	Unix Timestamp	RICHIESTO. Data di scadenza.
jwks	JWKS	RICHIESTO. Un JSON Web Key Set (JWKS) [RFC7517] che rappresenta la parte pubblica delle chiavi di firma dell'entità interessata. Ogni JWK nel set JWK DEVE avere un ID (claim **kid**).
metadata	JSON object	RICHIESTO. Ogni chiave dell'oggetto JSON rappresenta un identificatore del tipo di metadati e ogni valore DEVE essere un oggetto JSON che rappresenta i metadati secondo lo schema di metadati di quel tipo. Una configurazione di entità PUÒ contenere più dichiarazioni di metadati, ma solo una per ogni tipo di metadati (<**entity_type**>).

Gli oggetti Entity Configuration delle Entità di tipo Foglia contengono in aggiunta ai claim comuni anche i seguenti:

[13] Distinguiamo le chiavi di Federazione da quelle di OIDC Core, questi ultimi risiedono nei Metadata OIDC. Un Entity Configuration contiene sia le chiavi pubbliche di Federazione che i Metadata OIDC.

Nome	tipo	descrizione
authority_hints	Array di URLs	RICHIESTO. Contiene una lista di URL delle entità superiori, quali TA o SA che POSSONO emettere un Entity Statement relativo a questo soggetto.
trust_marks	JSON array	RICHIESTO per tutti i partecipanti fatta esclusione del Trust Anchor. Un array JSON contenente i Trust Mark. Vedere la Sezione Trust Mark.

Gli oggetti Entity Configuration della Federation Authority che è AgID, contiene in aggiunta ai claim comuni anche i seguenti:

Nome	tipo	descrizione
constraints	JSON object	RICHIESTO e include l'elemento **max_path_length** al quale viene assegnato un valore Integer. Indica il numero massimo di intermediari consentiti tra una Foglia e il suo Trust Anchor. PUÒ anche contenere il claim **allowed_leaf_entity_types**, che restringe i tipi di Entità riconoscibili come suoi discendenti.
trust_mark_issuers	JSON array	RICHIESTO. Indica quali autorità sono considerate attendibili nella federazione per l'emissione di specifici Trust Mark, questi assegnati mediante il proprio identificativo univoco.

Gli Entity Statement emessi dal Trust Ancor o suo Intermediario per i propri diretti discendenti, contengono in aggiunta ai claim comuni anche i seguenti:

Nome	tipo	descrizione
metadata_policy	JSON object	OPZIONALE. Oggetto JSON che descrive un criterio di metadati. Ogni chiave dell'oggetto JSON rappresenta un identificatore del tipo di metadati e ogni valore DEVE essere un oggetto JSON che rappresenta la politica dei metadati in base allo schema di quel tipo di metadati. Si rimanda alla specifica [OIDC-FED#Section.5.1] per i dettagli implementativi.

trust_marks	JSON array	RICHIESTO. Un array JSON contenente i Trust Mark emessi da se stesso per il soggetto discendente.

Metadata

OIDC-FED utilizza i claim dei Metadata così come definiti all'interno delle specifiche di OpenID Connect Discovery 1.0 e OpenID Connect Dynamic Client Registration 1.0 [OpenID.DiscoveryOpenID.Registration] rispettivamente per OP e RP.

In OIDC-FED il Metadata OIDC relativo a RP e OP viene definito all'interno del claim "**metadata**" e del suo sotto claim "<**entity_type**>", all'interno dell'Entity Configuration, come oggetto JSON.

OpenID Connect Provider Metadata

Un OP DEVE pubblicare all'interno del suo EC un Metadata da federation_entity e uno da openid_provider come riportato nel seguente esempio:

```
   "metadata":{
      "openid_provider": { … },
      "federation_entity": { … }

}
```

L'EC di un OP DEVE configurare un metadata di tipo "federation_entity" e contenere almeno i seguenti parametri obbligatori:

Nome	**Descrizione**	**valore**
organization_name	String	OBBLIGATORIO. Nome dell'organizzazione
homepage_uri		Sezione 4.8 di OIDC-FED
policy_uri		Sezione 4.8 di OIDC-FED
logo_uri	URL del logo dell'entità; DEVE essere in formato SVG	Sezione 4.8 di OIDC-FED
contacts	PEC istituzionale dell'ente	Sezione 4.8 di OIDC-FED

federation_resolve_endpoint		Sezione Endpoint di Federazione e Sezione 4.8 di OIDC-FED

Oltre ai parametri previsti dalle LLGG OIDC SPID il metadata di tipo "openid_provider" DEVE contenere anche i seguenti parametri obbligatori:

CLAIM	**DESCRIZIONE**
revocation_endpoint_auth_methods_supported	Vedi **RFC 8414#page-4**. Il valore supportato è **private_key_jwt**
code_challenge_methods_supported	Vedi **RFC 8414#page-4**. L'OP DEVE supportare S256 (vedi **RFC 7636#section-4.3**).
scopes_supported	Vedi OpenID.Discovery#OP Metadata. I valori supportati sono *openid*, *offline_access*, Per maggiori dettagli vedi Sezione User Claims.
response_types_supported	Vedi OpenID.Discovery#OP Metadata. Il valore supportato è **code**.
response_modes_supported	Vedi OpenID.Discovery#OP Metadata. I valori supportati sono *form_post* e *query*.
grant_types_supported	Vedi OpenID.Discovery#OP Metadata. I valori supportati sono *refresh_token* e *authorization_code* .
request_object_signing_alg_values_supported	Vedi OpenID.Discovery#OP Metadata. Vedi signature Algoritmi crittografici.
claims_supported	Vedi OpenID.Discovery#OP Metadata. Vedi Attributi Utente per maggiori dettagli.
authorization_response_iss_parameter_supported	Vedi **RFC 9207#section-3**. Deve valere *true*.
client_registration_types_supported	Vedi OIDC-FED Section 4.2. Il valore supportato è **automatic**.
request_authentication_methods_supported	Vedi OIDC-FED Section 4.2. Il valore supportato è **request_object**.

Se un OP non dispone all'interno dei propri Metadata dei claim **client_registration_types_supported** e/o **request_authentication_methods_supported** i valori da intendersi come impliciti sono i seguenti.

Nome	**tipo**	**valore**
client_registration_types_supported	String	"automatic"
request_authentication_methods_supported	JSON Object	{ "authorization_endpoint":["request_object"] }

Per la composizione dei Metadata SPID si rimanda alle LLGG OIDC SPID e successive integrazioni.

OpenID Connect Relying Party Metadata

Un RP DEVE pubblicare all'interno del suo EC un Metadata di tipo federation_entity e uno di tipo openid_relying_party come riportato nel seguente esempio:

```
   "metadata":{
      "openid_relying_party": { … },
      "federation_entity": { … }
}
```

Il Metadata di tipo "federation_entity" DEVE contenere almeno i seguenti parametri obbligatori:

Nome	**Descrizione**	**valore**
organization_name	String	OBBLIGATORIO. Nome dell'organizzazione
homepage_uri		Vedi Sezione 4.8 di OIDC-FED
policy_uri		Vedi Sezione 4.8 di OIDC-FED
logo_uri	URL del logo dell'entità; DEVE essere in formato SVG	Vedi Sezione 4.8 di OIDC-FED
contacts	PEC istituzionale dell'ente	Vedi Sezione 4.8 di OIDC-FED
federation_resolve_endpoint		Vedi Sezione Endpoint di Federazione e Sezione 4.8 di OIDC-FED

Oltre ai parametri previsti dalle LLGG OIDC SPID il metadata di tipo " openid_relying_party" DEVE contenere anche i seguenti parametri obbligatori:

CLAIM	**DESCRIZIONE**
id_token_signed_response_alg	Vedi OpenID.Registration#ClientMetadata. Vedi signature Algoritmi crittografici.
userinfo_signed_response_alg	Vedi OpenID.Registration#ClientMetadata. Vedi signature Algoritmi crittografici.
userinfo_encrypted_response_alg	Vedi OpenID.Registration#ClientMetadata. Vedi key encryption Algoritmi crittografici.

userinfo_encrypted_response_enc	Vedi OpenID.Registration#ClientMetadata. Vedi content encryption Algoritmi crittografici.
token_endpoint_auth_method	Vedi OpenID.Registration#ClientMetadata. Il valore richiesto è **private_key_jwt**.
client_registration_types	Vedi OIDC-FED Section 4.1. Il valore richiesto è **automatic**.

Se un RP non dispone all'interno dei propri Metadata dei claim **client_registration_types** i valori da intendersi come impliciti sono i seguenti.

Nome	**tipo**	**valore**
client_registration_types	String	"automatic"

Per la composizione dei Metadata SPID si rimanda alle LLGG OIDC SPID e successivi avvisi.

Metadata di Trust Anchor (TA) e Intermediari (SA)

È il Metadata che il Trust Anchor, o suo Intermediario, pubblica con l'identificativo **federation_entity**. Questa tipologia caratterizza il TA e i suoi Intermediari. Di seguito la struttura del Metadata di federation_entity.

```
   "metadata":{
      "federation_entity": { … }
}
```

Il Metadata di tipo "federation_entity" DEVE contenere almeno i seguenti parametri obbligatori:

Nome	**tipo**	**descrizione**
organization_name	String	OBBLIGATORIO. Nome dell'organizzazione
federation_fetch_endpoint	URL	RICHIESTO, url presso il quale sono pubblicati gli Entity Statements in formato JWT dei soggetti discendenti.
federation_list_endpoint	URL	RICHIESTO, url presso il quale è possibile ottenere la lista dei discendenti in formato JSON.

federation_resolve_endpoint	URL	RICHIESTO, url presso il quale è possibile ottenere i trust mark validati,il Metadata finale e la Trust Chain, relativamente ad un soggetto.
federation_trust_mark_status	URL	RICHIESTO, url presso il quale è possibile validare l'assegnazione di un Trust Mark ad uno specifico soggetto.
homepage_uri	URL	url della pagina web del Trust Anchor o SA.
policy_uri		Vedi Sezione 4.8 di OIDC-FED
logo_uri		URL del logo dell'entità; DEVE essere in formato SVG. Vedi Sezione 4.8 di OIDC-FED
contacts		PEC istituzionale dell'ente. DEVE essere in formato SVG. Vedi Sezione 4.8 di OIDC-FED.

Attribute Authority Metadata

Una AA DEVE pubblicare, all'interno del suo EC, un Metadata federation_entity e un Metadata oauth_resource e, se le risorse sono protette, DEVE anche pubblicare un Metadata oauth_authorization_server.

```
   "metadata":{
      "oauth_authorization_server": { … },
      "oauth_resource": { … },
      "federation_entity": { … }
}
```

Oltre ai claim comuni, il Metadata di tipo "federation_entity" DEVE contenere almeno i seguenti parametri obbligatori:

Nome	Tipo	Descrizione	Standard di riferimento
homepage_uri			Vedi Sezione 4.8 di OIDC-FED
policy_uri			Vedi Sezione 4.8 di OIDC-FED
logo_uri		URL del logo dell'entità; DEVE essere in formato SVG.	Vedi Sezione 4.8 di OIDC-FED
contacts		PEC istituzionale dell'ente.	Vedi Sezione 4.8 di OIDC-FED
federation_trust_mark_status_endpoint			Vedi Sezione Endpoint di Federazione e Sezione 4.8 di OIDC-FED
federation_resolve_endpoint			Vedi Sezione 4.8 di OIDC-FED

Di seguito i claim del Metadata di tipo oauth_authorization_server.

Nome	Tipo	Descrizione	Standard di riferimento
issuer	Stringa	Identificativo univoco della Attribute Authority.	[RFC8414]
authorization_endpoint			[RFC8414]
token_endpoint	Stringa	URL che il RP deve chiamare per scambiare un *Grant Token* con un *Access Token* (token exchange).	[RFC8414]
jwks	Oggetto JSON	Un JSON Web Key Set (JWKS) che rappresenta la parte pubblica delle chiavi di firma dell'entità interessata. Ogni JWK nel set JWK DEVE avere un ID (claim **kid**).	[RFC7517]
scopes_supported	JSON array	Lista di OAuth 2.0 **scope** che la AS supporta.	[RFC8414]
response_types_supported	JSON array	Lista dei valori "response_type" supportati dalla AA. Nel contesto di token exchange (profilo AA protected) viene supportato solo il valore token.	[RFC8414]
grant_types_supported	JSON array	Lista di OAuth 2.0 grant type che la AS supporta.	[RFC8414] [RFC8693]
token_endpoint_auth_methods_supported	JSON array	Array contenente i metodi di autenticazione supportati dal Token Endpoint. Deve essere presente solo il valore **private_key_jwt**	[RFC8414]
token_endpoint_auth_signing_alg_values_supported	JSON array	Array contenente l'elenco degli algoritmi di firma JWS supportati dal Token Endpoint per la firma del JWT utilizzato	[RFC8414]

		nell'autenticazione **private_key_jwt**	
op_policy_uri	Stringa	URL dove è disponibile la privacy policy del servizio AA. Può essere presente più di una occorrenza opportunamente localizzata in più lingue.	[RFC8414]
op_tos_uri			[RFC8414]
dpop_signing_alg_values_supported	JSON array	Array contenente l'elenco degli algoritmi di firma JWS supportati dalla AA per la DPoP proof.	draft-ietf-oauth-dpop-03 [OAuth-DPoP]

Il Metadata di "oauth_resource" contiene i seguenti parametri.

Nome	Tipo	Descrizione	Standard di riferimento
resource	JSON array	OBBLIGATORIO. Una o più URL che identificano gli endpoint delle risorse protette.	Draft-jones-oauth-resource-metadata-01 [Oauth-RS]

Metadata Policy

Trust Anchors e Intermediari (SA) DEVONO pubblicare una policy relativa ai rispettivi discendenti nell'Entity Statement ad essi riferito. La Metadata Policy si DEVE applicare a cascata su tutti i discendenti.

Di seguito vengono indicati i claim necessari che DEVONO essere sempre presenti nei diversi Metadata Policy, il TA può indicare ulteriori claim.

Metadata Policy di un TA per un RP

Di seguito vengono riportati i claim che DEVONO essere presenti nel parametro metadata di tipo **openid_relying_party** all'interno della policy che il TA stabilisce per un RP suo discendente diretto.

CLAIM	OPERAZIONI / VALORI
jwks	Operazioni: *subset_of* Valori: DEVE contenere i JWKS del RP relativi alle operazioni di Core
grant_types	Operazioni: *subset_of* Valori: DEVE essere *authorization_code* e *refresh_token*
id_token_signed_response_alg	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_signed_response_alg	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_encrypted_response_alg	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_encrypted_response_enc	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
token_endpoint_auth_method	Operazioni: *one_of* Valori: DEVE essere *private_key_jwt*
client_registration_types	Operazioni: *one_of* Valori: DEVE essere *automatic*

Metadata Policy di un TA per un SA

Di seguito vengono riportati i claim che DEVONO essere presenti nel parametro metadata di tipo **openid_relying_party** all'interno della policy che il TA stabilisce per un SA. Questa policy DEVE essere applicata a cascata ai metadata dei RP discendenti diretti (aggregati) del SA.

CLAIM	OPERAZIONI / VALORI
grant_types	Operazioni: *subset_of* Valori: DEVE essere *authorization_code* e *refresh_token*
id_token_signed_response_alg	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_signed_response_alg	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_encrypted_response_alg	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_encrypted_response_enc	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
token_endpoint_auth_method	Operazioni: *one_of* Valori: DEVE essere *private_key_jwt*
client_registration_types	Operazioni: *one_of* Valori: DEVE essere *automatic*

Metadata Policy di un SA per un RP

Di seguito vengono riportati i claim che DEVONO essere presenti nel parametro metadata di tipo **openid_relying_party** all'interno della policy che il SA stabilisce per un RP suo discendente diretto (Aggregato).

CLAIM	OPERAZIONI / VALORI
jwks	Operazioni: *subset_of* Valori: DEVE contenere i JWKS del RP relativi alle operazioni di Core

Metadata Policy di un TA per un OP

Di seguito vengono riportati i claim che DEVONO essere presenti nel parametro metadata di tipo **openid_relying_party** all'interno della policy che il TA stabilisce per un RP suo discendente diretto.

CLAIM	OPERAZIONI / VALORI
jwks	Operazioni: *subset_of* Valori: DEVE contenere i JWKS del OP relativi alle operazioni di Core
revocation_endpoint_auth_methods_supported	Operazioni: *one_of* Valori: DEVE essere *private_key_jwt*
code_challenge_methods_supported	Operazioni: *subset_of* Valori: DEVE essere *S256*
scopes_supported	Operazioni: *subset_of* Valori: DEVE essere *openid*, PUO' contenere *offline_access*
response_types_supported	Operazioni: *one_of* Valori: DEVE essere *code*.
response_modes_supported	Operazioni: *subset_of* Valori: DEVE essere *form_post*, *query*.
grant_types_supported	Operazioni: *subset_of* Valori: DEVE essere *refresh_token*, *authorization_code*.
acr_values_supported	Operazioni: *subset_of* Valori: DEVE essere *https://www.spid.gov.it/SpidL1*, *https://www.spid.gov.it/SpidL2*, *https://www.spid.gov.it/SpidL3*.
subject_types_supported	Operazioni: *one_of* Valori: DEVE essere *pairwise*.
id_token_signing_alg_values_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_signing_alg_values_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
userinfo_encryption_alg_values_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi

	definiti nella Sezione Algoritmi Crittografici
userinfo_encryption_enc_values_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
token_endpoint_auth_methods_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
token_endpoint_auth_signing_alg_values_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici
claims_parameter_supported	Operazioni: *one_of* Valori: DEVE essere *true*
request_parameter_supported	Operazioni: *one_of* Valori: DEVE essere *true*
authorization_response_iss_parameter_supported	Operazioni: *one_of* Valori: DEVE essere *true*
client_registration_types_supported	Operazioni: *one_of* Valori: DEVE essere *automatic*
request_authentication_methods_supported	Operazioni: *one_of* Valori: DEVE essere *request_object*
request_authentication_signing_alg_values_supported	Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici

Soggetti Aggregatori

In questa sezione sono specificate le modalità implementative dei Soggetti Aggregatori in contesto Federativo.

Un SA o Intermediario di Federazione è un soggetto che provvede alla registrazione di Foglie di tipo Relying Party e per le quali emette dei Trust Mark riconoscibili dalla AgID e all'interno della Federazione.
Un SA può registrare RP preesistenti e già conformi allo standard OIDC-FED, afferenti a domini esterni al proprio oppure mascherare dietro di sé i propri discendenti. Nel primo caso il SA è di tipo Trasparente (Aggregatore Light) mentre nel secondo caso è di tipo Proxy (Aggregatore Full).
Gli Aggregatori Light registrano RP preesistenti e conformi a OIDC-FED e pubblicano gli entity statement a questi riferiti.
Gli Aggregatori Full provvedono a costruire una interfaccia di autenticazione e federazione per conto dei propri aggregati, mediante risorse web solitamente esposte all'interno del proprio dominio. Questa tipologia di Aggregatore espone le seguenti risorse per ogni suo aggregato:

- **.well-known/openid-federation**, contenente un subject identifier del RP univoco;
- Authorization callback endpoint per l'acquisizione dell'auth code da parte del OP (**redirect_uri**).

Il SA di tipo **Full** DEVE aggiungere almeno uno dei codici identificativi presenti nell'**id_code** (così come definito nella Sezione Composizione dei Trust Mark), all'interno del web path che compone il client_id, questo identifica univocamente all'interno della federazione l'aggregato <SA_domain>/<id_code>/.
Se sono disponibili più di un codice identificativo, il SA PUÒ riportarli nel web path come nel seguente esempio: <SA_domain>/ipa_code/aoo_code/.
Nella seguente tabella sono presenti alcuni esempi non normativi per evidenziare le differenze tra gli aggregati Light e Full, dove per l'aggregato Full si usa la variabile $IDCODE ad identificare il soggetto aggregato.

	Light mode	**Full mode**
client_id	https://www.rp.it/	https://spid.sa.it/$IDCODE/
redirect_uri	https://www.rp.it/callback/	https://spid.sa.it/$IDCODE/callback/
authorization endpoint	https://www.rp.it/authorize/	https://spid.sa.it/$IDCODE/authorize/
entity configuration	https://www.rp.it/.well-known/openid-federation	https://spid.sa.it/$IDCODE/.well-known/openid-federation

Endpoint di Federazione

In questa sezione sono descritti gli endpoint di federazione che ogni entità, in base al tipo, deve esporre.

Endpoint comuni a tutti

Tutti i partecipanti all'interno della Federazione DEVONO esporre i seguenti *endpoint* web:

.well-known/openid-federation

Risorsa pubblica attraverso la quale un partecipante pubblica la sua configurazione (Entity Configuration).

Resolve Entity Statement endpoint

Risorsa pubblica attraverso la quale un partecipante rende noto il Metadata finale calcolato su una Trust Chain precedentemente elaborata e relativa ad un altro soggetto.
Il resolve entity statement endpoint non DEVE restituire alcuna informazione relativa ad un soggetto del quale non ha precedentemente raccolto gli statement e calcolato la Trust Chain. Nel caso in cui i TM non siano più validi al momento della richiesta, questi non DEVONO essere inclusi nella risposta.
L'Entità che espone questo endpoint rende noti i Trust Marks, i metadati e la Trust Chain, relativi alle Entità da esso riconosciute.
Un RP che espone questo endpoint rende noti i metadati degli OP da esso riconosciuti e viceversa.
Questo endpoint DEVE essere esposto da tutti i partecipanti della Federazione per rendere trasparenti le operazioni di analisi delle problematiche dovute al disallineamento dei metadati tra le Entità.
Questo endpoint richiede obbligatoriamente i seguenti parametri in fase di HTTP Request:

Claim	**tipo**	**descrizione**
sub	URL	Identificativo dell'Entità per la quale si chiede di ottenere i Trust Mark e i Metadata.
anchor	URL	Identificativo del TA.

Endpoint per Trust Anchor ed Intermediari

Il Trust Anchor e i suoi Intermediari (federation_entity) DEVONO in aggiunta esporre al pubblico i seguenti *endpoint*:

Fetch entity statement endpoint

Il recupero degli Entity Statement viene effettuato presso questo endpoint secondo le modalità definite all'interno di OIDC-FED "7.1. Fetching Entity Statements".

Trust mark status endpoint

L'assegnazione di un Trust Mark ad un soggetto viene effettuato presso questo endpoint secondo le modalità definite all'interno di OIDC-FED "7.4. Trust Mark Status".

Entity Listing endpoint

Per ottenere la lista dei discendenti registrati presso la TA o un suo Intermediario è possibile interrogare questo endpoint secondo le modalità descritte in OIDC-FED "7.3. Entity Listings". Ai parametri esistenti già definiti nella specifica, si aggiunge per SPID il parametro **entity_type** come filtro sul tipo di entità dei discendenti (<**entity-type**>)**.**

Gestione degli errori di federazione

Se l'OP non riesce a stabilire un rapporto di trust con l'RP o rileva che i metadati dell'RP non sono validi o sono in conflitto con la policy dei metadati, DEVE considerare l'URI di reindirizzamento come non valido e non eseguire il reindirizzamento previsto.

In caso di errore durante le operazioni di federazione, le entità DEVONO rappresentare i messaggi di anomalia come descritto di seguito.

CLAIM	**DESCRIZIONE**
error	Vedi Codici di errori
error_description	Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging.

Codici di errore di Federation

In caso di errore durante le operazioni di federazione, le entità DEVONO rappresentare i messaggi di anomalia come descritto di seguito.

ERRORE	**DESCRIZIONE**	**CODICE HTTP**
temporarily_unavailable	Uno degli endpoint di well-known o di Federation non è raggiungibile.	*503 Service Unavailable*
invalid_client	Il Client non è autorizzato perchè la validazione della Trust Chain fallisce.	*401 Unauthorized*
unauthorized_client	L'applicazione del metadata policy produce un metadata non conforme o nessun Trust Mark valido per il profilo richiesto è presente all'interno della configurazione.	*401 Unauthorized*
invalid_request	La richiesta non è completa o non è conforme a quanto definito dalle presenti specifiche tecniche.	*400 Bad Request*
not_found	La risorsa richiesta non è stata trovata.	*404 Not Found*

Algoritmi crittografici

Tutti i partecipanti devono pubblicare gli algoritmi supportati di criptazione e firma all'interno dei propri metadata. Tali agoritmi sono utilizzati per tutte le operazioni di cifratura e firma previsti da Federation.

La lunghezza delle chiavi RSA deve essere pari o superiore a 2048 bit. Si raccomanda una lunghezza di 4096 bit.

I seguenti algoritmi DEVONO essere supportati:

ALGORITMI	OPERAZIONI	RIFERIMENTO
RS256	Signature	OpenID.Core and RFC7518.
RS512	Signature	RFC7518
RSA-OAEP	Key Encryption	RFC7518.
RSA-OAEP-256	Key Encryption	RFC7516.
A128CBC-HS256	Content Encryption	RFC7516.
A256CBC-HS512	Content Encryption	RFC7516.

E' RACCOMANDATO il supporto per i seguenti algoritmi:

ALGORITMI	OPERAZIONI	RIFERIMENTO
ES256	Signature	OpenID.Core and RFC7518.
ES512	Signature	RFC7518.
PS256	Signature	RFC7518.
PS512	Signature	RFC7518.

I seguenti algoritmi NON DEVONO essere supportati:

ALGORITMI	OPERAZIONI	RIFERIMENTI
none	Signature	RFC7518.
RSA_1_5	Key Encryption	RFC7516.

HS256	Signature	RFC7518.
HS384	Signature	RFC7518.
HS512	Signature	RFC7518.

Retention Policy

Gestione dei Log di un OP e di un RP

Gli OP e gli RP, oltre a quanto già previsto nelle LLGG OIDC SPID, DEVONO registrare nel registro delle transazioni la **Trust Chain** relativa all'Entità con la quale è avvenuta la transazione, composta da:

a. L'**Entity Configuration** del Entità con la quale è avvenuta la transazione.
b. [Solo per OP] L'**Entity Statement** del SA riferito al RP (se presente).
c. L'**Entity Statement** del TA riferito al suo discendente.
d. L'**Entity Configuration** del TA.

Registro storico delle chiavi pubbliche di Federazione

Al fine di consentire la verifica dei messaggi scambiati dalle Entità che partecipano alla federazione e delle relative Trust Chain, il TA DEVE pubblicare lo storico delle proprie chiavi pubbliche (JWKS) di federazione all'interno di un registro reso disponibile a tutti i partecipanti tramite l'endpoint /.well-known/openid-federation-jwks. Per ulteriori dettagli tecnici si rimanda alla Sezione 7.5 di OIDC-FED

Le chiavi che non sono più attive da più di 24 mesi POSSONO essere rimosse dal registro a discrezione del TA.

Differenze con OIDC Federation 1.0

In questa sezione sono elencate le principali peculiarità in ambito SPID.

Client Registration

SPID supporta esclusivamente **automatic_client_registration**. La modalità **implicit** è da intendersi come non supportata.

Trust Mark

In OIDC-FED l'uso dei Trust Mark non è obbligatorio. In SPID l'esposizione dei Trust Mark è obbligatoria. Per approfondimenti sulla ragione dell'obbligo dei Trust Mark si rimanda alla sezione "Considerazioni di Sicurezza".

Claim non supportati negli Entity Statement

Poiché SPID non necessita di alcun claim aggiuntivo in ambito federativo, non necessita dei claim **crit**. Inoltre non sono supportati i claim **aud**, **naming_constraints**, **policy_language_crit** e **trust_anchor_id**. L'eventuale presenza di questi claim non presenta alcuna implicazione, questi verranno semplicemente ignorati fino ad ulteriori avvisi che li normino.

Considerazioni di Sicurezza

In questa sezione descriviamo alcune considerazioni di sicurezza in ambito OIDC Federation.

Trust Mark come deterrente contro gli abusi

L'implementazione dei Trust Mark e il filtro su questi in fase di Federation Entity Discovery risulta necessario contro gli attacchi destinati al consumo delle risorse. Un OP attaccato con un numero ingente di connessioni presso il suo endpoint di *authorization*, contenenti **client_id** e **authority_hints** fasulli, produrrebbe svariate connessioni verso sistemi di terze parti nel tentativo di trovare un percorso verso la TA e instaurare la fiducia con il richiedente.
L'OP DEVE validare staticamente il TM oppure DEVE escludere a priori la richiesta ove il TM non risultasse presente, in caso di assenza o non validità di un TM la procedura di Federation Entity Discovery NON DEVE essere avviata e NON DEVE creare di conseguenza connessioni verso sistemi di terze parti.

Numero Massimo di authority_hints

All'interno di una Federazione il Trust Anchor decide quante intermediazioni consentire tra di lui e le Foglie, mediante la *constraint* denominata **max_path_lenght**. Questo tipo di relazione è di tipo verticale, dalla foglia alla radice. Questo attributo se valorizzato ad esempio con un valore numerico intero pari a 1 indica che soltanto un SA è consentito tra una Foglia e il TA.
Ogni Foglia DEVE pubblicare i suoi superiori all'interno della lista contenuta nel claim **authority_hints**. Una Foglia all'interno della Federazione PUÒ avere superiori afferenti a diverse Federazioni, si pensi a CIE id per esempio. L'analisi dei superiori disponibili introduce un modello di navigazione orizzontale, ad esempio un OP tenta di trovare il percorso più breve verso il Trust Anchor attraverso tutti gli URL contenuti all'interno dell'array **authority_hints** prima di fare un ulteriore movimento verticale, a salire, verso uno degli Intermediari presenti in questo array.
La soglia **max_path_lenght** si applica per la navigazione verticale e superata questa soglia senza aver trovato il TA la procedura di Federation Entity Discovery DEVE essere interrotta. Si faccia l'esempio di un RP discendente di un 1 SA che quest'ultimo a sua volta è discendente di un altro SA, essendo il valore di **max_path_lenght** pari a uno e superata questa soglia senza aver trovato il Trust Anchor, la procedura DEVE essere interrotta.
Allo stesso tempo la specifica OIDC Federation 1.0 non definisce un limite per il numero di **authority_hints**, questo perché nessun Trust Anchor può limitare il numero di Federazioni alle quali un partecipante può aderire. Per questa ragione è utile che gli implementatori adottino un limite massimo del numero di elementi consentiti all'interno dell'Array **authority_hint**. Questo per evitare che un numero esagerato di URL contenuti nella lista di **authority_hints**, dovuto ad una cattiva configurazione di una Foglia, produca un consumo di risorse eccessivo.

Resolve Entity Statement

Questo endpoint DEVE rilasciare i Metadata, i Trust Marks e la Trust Chain già precedentemente elaborati e NON DEVE innescare una procedura di Federation Entity Discovery ad ogni richiesta pervenuta, a meno che questo endpoint non venga protetto con un meccanismo di autenticazione dei client, come ad esempio **private_key_jwt** [OIDC-CORE].

Buone Pratiche

In questa sezione descriviamo alcune buone pratiche per ottenere la massima resa dalle entità di Federazione.

Specializzare le chiavi pubbliche OpenID Core e Federation

È buona pratica usare chiavi pubbliche specializzate per i due tipi di operazioni, Core e Federation.

Modalità di aggiornamento dei Metadata OpenID Core

L'interoperabilità tra i partecipanti funziona mediante i Metadata ottenuti dal calcolo e dalla conservazione delle Trust Chain. Questo significa che se un OP al tempo T calcola la Trust Chain per un RP e questo al tempo T+n modifica i propri Metadata, l'OP di conseguenza potrebbe incorrere in problematiche di validazione delle richieste di autorizzazione del RP, fino a quando non avrà aggiornato la Trust Chain relativa a questo.

La buona pratica per evitare le interruzioni di servizio relative alle operazioni di OIDC Core è quella di aggiungere le nuove chiavi pubbliche all'interno degli oggetti *jwks* senza rimuovere i valori preesistenti. Oppure, ad esempio, i nuovi *redirect_uri*.
In questa maniera dopo il limite massimo di durata delle Trust Chain, definito con il claim **exp** e pubblicato nella Entity Configuration della TA, si ha la certezza che tutti i partecipanti abbiano rinnovato le loro Trust Chain, e sarà possibile agli amministratori della Foglia rimuovere le vecchie definizioni in cima alla lista.

Periodo di grazia per le Trust Chain scadute

In una Federazione distribuita come quella di OIDC-FED è possibile che al tempo T+x un OP necessiti di aggiornare alcune Trust Chain, relative a diversi RP, prossime alla scadenza. Si faccia l'esempio che parte di questi RP risultino aggregati da una SA e i servizi di questo risultino temporaneamente non raggiungibili.
In questi casi, ove vi fosse l'impossibilità di aggiornare una Trust Chain a causa di irraggiungibilità dei servizi web di federazione, è possibile continuare ad utilizzare le Trust Chain scadute fino ad un massimo di 24 ore successive al primo tentativo di aggiornamento. All'interno di questo intervallo temporale "di grazia" sono comunque necessari periodici tentativi di aggiornamento.

Riferimenti Tecnici agli Standard

[SPID-OIDC-CORE]	Linee Guida per OpenID Connect in SPID

[OIDC-FED]	OpenID Connect Federation 1.0
[LG-AA]	Attribute Authority Guidelines – "Linee Guida Attribute Authority SPID"
[OpenID.Core]	Sakimura, N., Bradley, J., Jones, M., de Medeiros, B. and C. Mortimore, "OpenID Connect Core 1.0", August 2015.
[OpenID.Registration]	Sakimura, N., Bradley, J., and M. Jones, "OpenID Connect Dynamic Client Registration 1.0," November 2014.
[OpenID.Discovery]	Sakimura, N., Bradley, J., Jones, M., and E. Jay, "OpenID Connect Discovery 1.0," November 2014.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997.
[RFC7515]	Jones, M., Bradley, J. and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015.
[RFC7517]	Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015.
[RFC7519]	Jones, M., Bradley, J. and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015.
[RFC8174]	Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, DOI 10.17487/RFC8174, May 2017.
[RFC3339]	Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002.
[RFC8414]	Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, DOI 10.17487/RFC8414, June 2018.
[RFC7591]	Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, DOI 10.17487/RFC7591, July 2015.
[RFC3986]	Uniform Resource Identifier (URI): Generic Syntax
[EN319-412-1]	Electronic Signatures and Infrastructures (ESI); Certificate Profiles;