Regole tecniche OpenID Connect SPID/CIE

Regole tecniche OpenID Connect SPID/CIE

- SPID e CIE id sono i Sistemi Pubblici di Identità Digitale Italiani e adottano gli standard OpenID Connect Core, International Government Assurance Profile (iGov) for OpenID Connect 1.0 e OpenID Connect Federation 1.0.
- Grazie all'identità digitale, la Pubblica Amministrazione e i fornitori di servizi privati forniscono la chiave per accedere ai servizi online attraverso una credenziale unica.

Termini

Autorità di Federazione	Un'entità legale che gestisce la fiducia tra le parti coinvolte nella Federazione e norma il funzionamento e le modalità di registrazione e riconoscimento dei partecipanti.
Trust Anchor	Sistema gestito dalla Autorità di Federazione, che rappresenta la Federazione e la sua configurazione.
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) secondo le regole definite dall'Autorità di Federazione.
Foglia	Entità definita dal protocollo OpenID Connect come Relying Party e Provider OpenID. Può anche essere una Attribute Authority (OAuth2 Authorization Server e Resource Server).
Entità	Partecipante alla Federazione. Trust Anchor, Intermediario o Foglia.
Entity Configuration	Dichiarazione di un'entità, emessa per proprio conto, nella forma di JWT auto firmato **RFC 7515**[53] e contenente la sua configurazione. Contiene le chiavi pubbliche di Federazione, i 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 soggetto discendente (RP, OP, AA o Intermediario) in formato JWT firmato **RFC 7515**[54], contenente le chiavi pubbliche del soggetto discendente, i Trust Mark emessi per i quali è emettitore e la politica dei Metadata da applicare ai Metadata del soggetto.
Trust Mark	JWT firmato **RFC 7515**[55] dall'ente emettitore e relativo ad un partecipante. Attesta la conformità di questo ai profili riconoscibili all'interno Federazione (RP pubblico o privato, Soggetto Aggregatore Pubblico o Privato, etc.). La Foglia che acquisisce il marchio di fiducia durante il processo di onboarding DEVE includere questo nella sua Entity Configuration.
Metadata	Documento che descrive l'implementazione di una entità OpenID Connect o OAuth2. Le implementazioni di ogni Entità condividono i Metadata 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 parametro di Metadata.
Authority hint	Array di valori URL contenente gli 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 la Trust Chain deve essere aggiornata.
Onboarding	Procedura di registrazione di una nuova entità all'interno della Federazione SPID e CIE
Federation Endpoint	Endpoint definit in OIDC Federation 1.0, usati per prendere e risolvere gli statement delle entità, interrogare una lista di tutte le entità subordinate e verificare lo stato dei Trust Mark.

Acronimi

SPID	Sistema Pubblico di Identità Digitale italiano, la cui Authorità di Federazione è la AgID (Agenzia per l'Italia Digitale).
CIE id	Sistema Pubblico di Identità Digitale italiano basato sulla Carta d'Identità Elettronica (CIE), di cui il Ministero dell'Interno è l'Autorità di Federazione. La gestione tecnica e operativa è affidata all'Istituto Poligrafico e Zecca dello Stato (IPZS).
OIDC	OpenID Connect.
OIDC-FED	OIDC Federation 1.0[56].
FA	Autorità di Federazione (Federation Authority).
TA	OIDC Federation Trust Anchor.
AgID	Agenzia per l'Italia Digitale, FA/TA di SPID.
MinInterno	Ministero dell'Interno, FA/TA di CIE id.
OP	OpenID Provider (Entità Foglia).
RP	Relying Party (Entità Foglia).
SA	Soggetti Aggregatori. Entità Intermediarie che possono gestire tutti gli aspetti della Federazione di uno o più RP.
AA	Attribute Authority, Gestore degli Attributi qualificati (Entità Foglia).
TM	Trust Mark.
EC	Entity Configuration.
ES	Entity Statement.
URL	Uniform Resource Locator, corrispondente ad un indirizzo web.
JWT	Vedi **RFC 7519**[57] Jones, M., Bradley, J. and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015.
RS	OAuth2 Resource Server.
$JWT	Il valore di un JWT (JSON Web Token).

Le Federazioni e ID Italiane

Una Federazione delle Identità Digitali è una 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 sicur

Questa configurazione stabilisce i livelli di garanzia e di sicurezza adeguati affinchè un individuo possa autenticarsi presso un servizio web (Service Provider) mediante la propria identità digitale, rilasciata da un altro servizio web (Identity Provider). i che consentono l'interscambio dei dati.

I partecipanti (RP o OP), che si riconoscono all'interno della medesima Federazione, ottengono i Metadata gli uni degli altri. I Metadata contengono le chiavi pubbliche per le operazioni di firma digitale e criptazione e le definizioni necessarie all'interscambio delle informazioni.

I Metadata sono certificati da un parte fidata che all'interno della Federazione SPID è AgID, mentre all'interno della Federazione CIE è il Ministero dell'Interno. Questi corrispondono alla Autorità di Federazione.

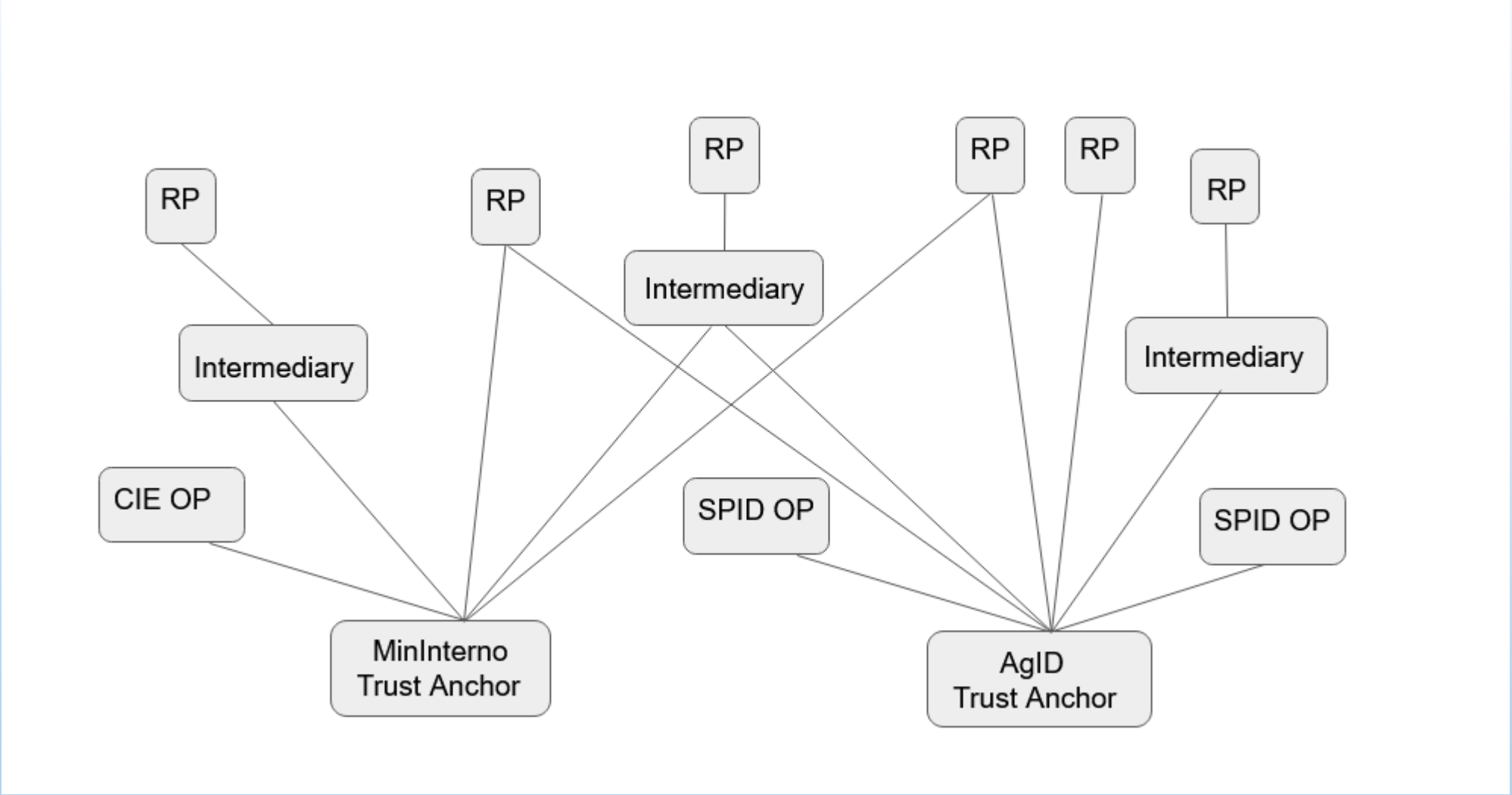
SPID e CIE id implementano OpenID Connect Federation 1.0 e ne estendono alcune funzionalità, realizzano una implementazione concreta e producono le buone pratiche per la sua adozione. Per approfondimenti allo standard si rimanda alle specifiche ufficiali OIDC-FED e alla sezione Differenze con OIDC Federation 1.0.

La Federazione OIDC produce una infrastruttura della fiducia che è:

Dinamica. La fiducia può essere stabilita dinamicamente durante la prima richiesta di autenticazione. Le Autorità della Federazione espongono un endpoint che fornisce "dichiarazioni" firmate riguardanti le entità discendenti. Queste contengono le chiavi pubbliche dei discendenti e la politica dei Metadata. Le Autorità della Federazione possono disabilitare un'entità nella Federazione in qualsiasi momento, semplicemente smettendo di emettere le dichiarazioni inerenti a questa.

Scalabile. Riduce significativamente i costi di onboarding, in accordo al principio di delega, con l'istituzione di entità intermediarie (SA).

Trasparente. Qualsiasi Entità coinvolta nella Federazione può in ogni momento costruire la fiducia autonomamente e in modo sicuro. Inoltre, la composizione della Federazione, in tutte le sue parti, diventa navigabile mediante la sua API, in tempo reale.



Schema ad albero con le Autorità di Federazione SPID e CIE id e, salendo, gli OP che non hanno Intermediari, gli RP e gli Intermediari che a loro volta Aggregano altri RP.

Configurazione della Federazione

La configurazione della Federazione è pubblicata dal Trust Anchor all'interno della sua Entity Configuration, disponibile 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 mantenerla aggiornata su base giornaliera. All'interno della configurazione della Federazione sono pubblicate le chiave pubbliche del Trust Anchor usate 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 Mark (trust_mark_issuers)

Modalità di partecipazione

Per aderire alle Federazioni SPID e CIE id un partecipante deve pubblicare la propria configurazione (Entity Configuration) presso il proprio web endpoint .well-known/openid-federation.

Gli incaricati tecnici ed amministrativi della Foglia completano la procedura amministrativa per la registrazione di una nuova Entità o l'aggiornamento di un'Entità preesistente definita dalla Autorità di Federazione o da un suo Intermediario (SA).

L'Autorità di Federazione o il 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).

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 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..

Entity Configuration

Un'**Entity Configuration (EC)** è un Metadata di Federazione in formato Jose e firmato da una Entità e riguardante se stessa, pubblicato presso il web endpoint **.well-known/openid-federation**.

Tutte le operazioni di verifica della firma relative agli ES (Entity State), EC (Entity Configuration) e TM (Trust Mark) sono eseguite con le chiavi pubbliche di Federazione.

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.

Algoritmi di firma supportati

In SPID e CIE id i seguenti algoritmi DEVONO essere supportati:

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

Entity Configuration - claim comuni

CLAIM	DESCRIZIONE	SUPPORTATO DA
iss	String. Identificativo dell'entità che lo emette.	sp:d CIE ID
sub	String. Identificativo del soggetto a cui è riferito.	sp:d CIE ID
iat	UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519	sp:d CIE ID
exp	UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519.	sp:d CIE ID
jwks	Un JSON Web Key Set (JWKS) RFC 7517 che rappresenta la parte pubblica delle chiavi di firma dell'entità interessata. Ogni JWK nel set JWK DEVE avere un ID di chiave (claim kid).	sp:d CIE ID
metadata	JSON Object. Ogni chiave dell'oggetto JSON rappresenta un identificatore del tipo di Metadata e ogni valore DEVE essere un oggetto JSON che rappresenta i Metadata secondo lo schema di Metadata di quel tipo. Una configurazione di entità PUÒ contenere più dichiarazioni di Metadata, ma solo una per ogni tipo di Metadata (<entity_type>). I tipi consentiti sono i seguenti: • openid_relying_party • openid_provider • federation_entity • oauth_authorization_server • oauth_resource	sp:d CIE ID

All'interno dell'EC i valori degli attributi **iss** e **sub** contengono il medesimo valore (URL).

Entity Configuration Foglia e intermediari

Gli EC delle entità Foglia e intermediari, in aggiunta ai claim precedentemente definiti, contengono anche i seguenti claim:

CLAIM	DESCRIZIONE	SUPPORTATO DA
authority_hints	Array di URL. Contiene una lista di URL delle entità superiori, quali TA o SA che POSSONO emettere un ES relativo a questo soggetto.	sp:d CIE ID
trust_marks	Un array JSON contenente i Trust Mark. Vedere la Sezione Trust Mark. Obbligatorio per tutti i partecipanti fatta esclusione del Trust Anchor.	sp:d CIE ID

Entity Configuration Trust Anchor

Gli EC di un TA, in aggiunta ai claim comuni a tutti i partecipanti, contengono anche i seguenti:

CLAIM	DESCRIZIONE	SUPPORTATO DA
constraints	JSON Object che descrive un insieme di vincoli della Trust Chain e che DEVE contenere l'attributo **max_path_length**. Rappresenta il numero massimo di SA tra una Foglia e il TA. PUÒ anche contenere il claim **allowed_leaf_entity_types**, che restringe i tipi di Entità riconoscibili come suoi discendenti.	spid CIE ID
trust_mark_issuers	JSON Array che indica quali autorità sono considerate attendibili nella Federazione per l'emissione di specifici TM, questi assegnati mediante il proprio identificativo univoco.	spid CIE ID

Entity Statement

Il componente basilare per costruire una Catena di Fiducia (Trust Chain) è l'**Entity Statement (ES)**, un JWT firmato che contiene la chiavi pubbliche dell' Entità discendente (subject) e ulteriori dati usati per controllare il processo di risoluzione della Trust Chain.

Una entità pubblica un **ES** relativo ad un suo discendente presso il proprio Fetch Endpoint. L'entità superiore PUÒ definire le policy sui metadata per un soggetto discendente e pubblicare i TM da lei emessi per questo.

Firma di Entity Statement

Si applicano le medesime considerazioni fatte per gli **EC**

Entity Statement – Claim comuni

CLAIM	DESCRIZIONE	SUPPORTATO DA
iss	Si rimanda alla specifica OIDC-FED Sezione 3.1 per i dettagli.	sp:d CIE ID
sub	Si rimanda alla specifica OIDC-FED Sezione 3.1 per i dettagli.	sp:d CIE ID
iat	Si rimanda alla specifica OIDC-FED Sezione 3.1 per i dettagli.	sp:d CIE ID
exp	Si rimanda alla specifica OIDC-FED Sezione 3.1 per i dettagli.	sp:d CIE ID
jwks	JWKS di Federazione dell'entità *sub*. Si rimanda alla specifica OIDC-FED Sezione 3.1 per i dettagli.	sp:d CIE ID
metadata_policy	JSON Object che descrive un criterio di Metadata. Ogni chiave dell'oggetto JSON rappresenta un identificatore del tipo di Metadata e ogni valore DEVE essere un oggetto JSON che rappresenta la politica dei Metadata in base allo schema di quel tipo di Metadata. Si rimanda alla specifica OIDC-FED Section 5.1 per i dettagli implementativi.	sp:d CIE ID
trust_marks	JSON Array contenente i Trust Mark emessi da se stesso per il soggetto discendente.	sp:d CIE ID
constraints	PUÒ contenere il claim **allowed_leaf_entity_types** per restringere i tipi di Entità riconoscibili per il suo discendente (esempio: solo RP).	sp:d CIE ID

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.

Metadata Policy di un TA per un RP

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

CLAIM	OPERAZIONI / VALORI	SUPPORTATO DA
jwks	Operazioni: *value* Valori: DEVE contenere i JWKS del RP relativi alle operazioni di Core *essential = true*	sp:d CIE ID
grant_types	Operazioni: *subset_of, super_set* Valori: DEVE contenere *authorization_code* e *refresh_token* *essential = true*	sp:d CIE ID
id_token_signed_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	sp:d CIE ID
id_token_encrypted_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = false*	CIE ID
id_token_encrypted_response_enc	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = false*	CIE ID
userinfo_signed_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi	sp:d CIE ID
userinfo_encrypted_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	sp:d CIE ID
userinfo_encrypted_response_enc	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	sp:d CIE ID
token_endpoint_auth_method	Operazioni: *one_of* Valori: DEVE essere *private_key_jwt* *essential = true*	sp:d CIE ID
client_registration_types	Operazioni: *subset_of* Valori: DEVE essere *automatic* *essential = true*	sp:d CIE ID
redirect_uris	Operazioni: *essential = true*	sp:d CIE ID
client_id	Operazioni: *essential = true*	sp:d CIE ID
response_types	Operazioni: *value* Valori: DEVE essere *code* *essential = true*	sp:d CIE ID

Metadata Policy di un TA per un SA

Di seguito vengono riportati i claim che DEVONO essere considerati 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	SUPPORTATO DA
grant_types	Operazioni: *subset_of*, *superset_of* Valori: DEVE contenere *authorization_code* e *refresh_token* *essential = true*	spid cie
id_token_signed_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
id_token_encrypted_response_alg	Operazioni: *one_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = false*	cie
id_token_encrypted_response_enc	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = false*	cie
userinfo_signed_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
userinfo_encrypted_response_alg	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici	spid cie
userinfo_encrypted_response_enc	Operazioni: *one_of* Valori: DEVE contenere uno degli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
token_endpoint_auth_method	Operazioni: *one_of* Valori: DEVE essere *private_key_jwt* *essential = true*	spid cie
client_registration_types	Operazioni: *subset_of* Valori: DEVE essere *automatic* *essential = true*	spid cie
redirect_uris	Operazioni: *essential = true*	spid cie
client_id	Operazioni: *essential = true*	spid cie
response_types	Operazioni: *value* Valori: DEVE essere *code* *essential = true*	spid cie

Metadata Policy di un SA per una RP

Di seguito vengono riportati i claim che DEVONO essere considerati 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	SUPPORTATO DA
jwks	Operazioni: *value* Valori: DEVE contenere i JWKS del RP relativi alle operazioni di Core *essential = true*	sp:d CIE ID

Metadata Policy di un TA per un OP

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

CLAIM	OPERAZIONI / VALORI	SUPPORTATO DA
jwks	Operazioni: *value* Valori: DEVE contenere i JWKS del OP relativi alle operazioni di Core *essential = true*	sp:d CIE ID
revocation_endpoint_auth_methods_supported	Operazioni: *subset_of* Valori: DEVE essere *private_key_jwt* *essential = true*	sp:d CIE ID
code_challenge_methods_supported	Operazioni: *subset_of* Valori: DEVE essere *S256* *essential = true*	sp:d CIE ID
scopes_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere *openid, offline_access*. Per CIE id PUÒ contenere anche *profile, email*. *essential = true*	sp:d CIE ID
response_types_supported	Operazioni: *subset_of* Valori: DEVE essere *code*. *essential = true*	sp:d CIE ID
response_modes_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere *form_post, query*.	sp:d CIE ID
grant_types_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere *refresh_token, authorization_code*. *essential = true*	sp:d CIE ID
acr_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere *https://www.spid.gov.it/SpidL1, https://www.spid.gov.it/SpidL2, https://www.spid.gov.it/SpidL3*. *essential = true*	sp:d CIE ID
subject_types_supported	Operazioni: *subset_of* Valori: DEVE essere *pairwise*. *essential = true*	sp:d CIE ID
id_token_signing_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	sp:d CIE ID
id_token_encryption_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	CIE ID
id_token_encryption_enc_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	CIE ID

userinfo_signing_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
userinfo_encryption_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
userinfo_encryption_enc_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
token_endpoint_auth_methods_supported	Operazioni: *subset_of* Valori: DEVE essere *private_key_jwt* *essential = true*	spid cie
token_endpoint_auth_signing_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
claims_parameter_supported	Operazioni: *value* Valori: DEVE essere *true* *essential = true*	spid cie

request_parameter_supported	Operazioni: *value* Valori: DEVE essere *true* *essential = true*	spid cie
authorization_response_iss_parameter_supported	Operazioni: *value* Valori: DEVE essere *true* *essential = true*	spid cie
client_registration_types_supported	Operazioni: *subset_of* Valori: DEVE essere *automatic* *essential = true*	spid cie
request_authentication_methods_supported	Operazioni: *value* Valori: DEVE essere *request_object* *essential = true*	spid cie
request_authentication_signing_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie
request_object_signing_alg_values_supported	Operazioni: *subset_of, superset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione Algoritmi Crittografici *essential = true*	spid cie

issuer	Operazioni: *essential = true*	sp:d CIE ID
authorization_endpoint	Operazioni: *essential = true*	sp:d CIE ID
token_endpoint	Operazioni: *essential = true*	sp:d CIE ID
userinfo_endpoint	Operazioni: *essential = true*	sp:d CIE ID
introspection_endpoint	Operazioni: *essential = true*	sp:d CIE ID
revocation_endpoint	Operazioni: *essential = true*	sp:d CIE ID

Trust Mark

- I **Trust Mark (TM)**, letteralmente tradotti come *Marchi di Fiducia*, sono JWT firmati RFC 7515 e rappresentano la dichiarazione di conformità ad un insieme ben definito di requisiti di fiducia e/o di interoperabilità o un accordo tra le parti coinvolte all'interno della Federazione.
- Lo scopo principale dei TM 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 (Codice Fiscale, IPA Code, Partita IVA, VAT Number), i contatti istituzionali e altro, come definito in OIDC-FED. Ulteriori dati possono essere aggiunti dal soggetto che li emette.
- I TM sono emessi e firmati, durante il processo di registrazione di una nuova entità di tipo Foglia (Onboarding), dal (TA) o suoi Intermediari (SA) o da Gestori Qualificati di Attributi (AA), se definiti all'interno dell'attributo **trust_mark_issuers**, pubblicato all'interno dell'Entity Configuration del TA.

Trust Mark

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


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

Trust Mark

- Ogni entità partecipante DEVE esporre nella propria configurazione (EC) i TM rilasciati dalle autorità che li emettono.
- Nello scenario CIE / SPID, un TM viene firmato dal TA **MinInterno** / **Agid** o loro Intermediari (SA) o Gestori Qualificati di Attributi (AA).
- Il TA definisce i soggetti abilitati all'emissione dei TM riconoscibili all'interno della Federazione, mediante il claim **trust_mark_issuers**, presente all'interno del proprio Entity Configuration. Il valore dell'attributo **trust_mark_issuers** è composto da un oggetto JSON avente come chiavi gli identificativi dei TM e come valori la lista degli identificativi (URL) delle entità abilitate ad emetterli.
- 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.

Trust Mark

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_role> / [<trustmark_profile> /] [estensione /]

Alcuni esempi non normativi sono di seguito riportati:

- TM RP public: **https://registry.agid.gov.it/openid_relying_party/public/**
- TM SA private: **https://registry.agid.gov.it/intermediate/private/**
- TM AA: **https://registry.agid.gov.it/oauth_resource/public/**

Trust Mark

La tabella seguente definisce i <entity_role> riconoscibili all'interno delle Federazioni SPID e CIE id:

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 tabella seguente definisce i <trustmark_profile> riconoscibili all'interno delle Federazioni SPID e CIE id:

PROFILO	DESCRIZIONE	ENTITÀ
public	l'entità nel claim *sub* appartiene alla pubblica amministrazione italiana.	RP, OP, SA, AA
private	l'entità nel claim *sub* appartiene al settore privato.	RP, OP, SA, AA

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 la chiave pubblica dell'autorità che lo ha emesso (attributo iss), sulla base della corrispondenza dell'attributo sub con il medesimo attributo della Entity Configuration in cui è contenuto e sulla base del valore di scadenza (attributo exp).
2. Validazione dinamica. I partecipanti della Federazione possono interrogare l'endpoint trust mark status erogato dal suo emettitore (attributo iss) per la verifica in tempo reale dei TM da lui emessi.

Tutte le entità che rilasciano 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 solo ed esclusivamente dal soggetto che lo ha emesso. Ad esempio, 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 DEVE revocare il TM per il suo discendente.

Nota: 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.

Composizione dei Trust Mark

Gli attributi definiti all'interno dei TM aderiscono a quanto definito all'interno dello standard OIDC Federation 1.0 (OIDC-FED).

Attributi Trust Mark

CLAIM	DESCRIZIONE	SUPPORTATO DA
iss	String. URL che identifica univocamente l'Autorità che lo ha emesso.	spid CIE
sub	String. URL che identifica univocamente il soggetto per il quale il Trust Mark è stato emesso.	spid CIE
id	String. Identificativo univoco del Trust Mark. È un URL con la seguente struttura: **<TA domain>/<entity_type>/<trustmark_profile>/** es. non normativo: `https://registry.interno.gov.it/openid_relying_party/public/`	spid CIE
iat	UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE
logo_uri	String. Un URL che punta al logo rappresentante il Trust Mark.	spid CIE
exp	UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE
ref	String. URL che punta a informazioni presenti sul web relative a questo Trust Mark.	spid CIE
organization_type	String. Specifica se l'ente appartiene alla pubblica amministrazione italiana o al settore privato (**public** o **private**)	spid CIE
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*.	spid CIE
email	String. Email istituzionale o PEC dell'organizzazione.	spid CIE
organization_name	String. Il nome completo dell'entità che fornisce i servizi	spid CIE
sa_profile	String. RICHIESTO per SA. Specifica il profilo dell'Aggregatore, **full** o **light**.	spid CIE

Il valore contenuto nel parametro **exp** NON DEVE essere superiore alla durata delle convenzioni stipulate in fase di onboarding tra l'Entità che rilascia i Trust Mark e le organizzazioni che lo ricevono.

Soggetti Aggregatori

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).

I SA Light registrano RP preesistenti e conformi a OIDC-FED e pubblicano gli ES a questi riferiti.

I SA 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 la Entity Configuration del proprio discendente (aggregato);
- 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:

	MODALITÀ LIGHT	**MODALITÀ FULL**
client_id	https://www.rp.it/	https://www.sa.it/<id_code>/
redirect_uri	https://www.rp.it/callback/	https://www.sa.it/<id_code>/callback/
authorization endpoint	https://www.rp.it/authorization/	https://www.sa.it/<id_code>/authorization/
Entity Configuration	https://www.rp.it/.well-known/openid-federation	https://www.sa.it/<id_code>/.well-known/openid-federation

Acquisire i Metadata

modalità di mutuo riconoscimento dei partecipanti all'interno della medesima federazione, le modalità con le quali i partecipanti ottengono i metadata gli uni degli altri in maniera sicura.

Relying Party

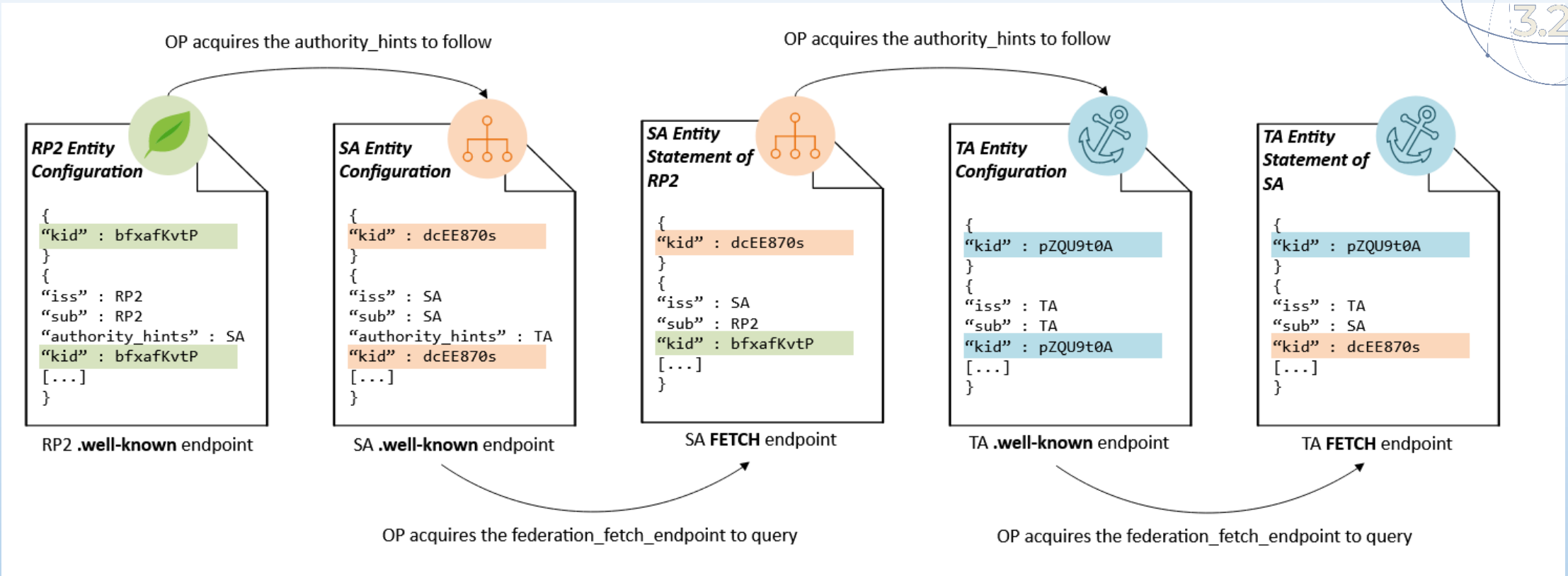
Il RP ottiene la lista degli OP in formato JSON interrogando l'endpoint list disponibile presso il Trust Anchor. Per ogni soggetto contenuto nella risposta dell'endpoint list e corrispondente ad un OP, il RP richiede ed ottiene l'Entity Configuration presso l'OP.

Per ogni EC degli OP, il RP verifica la firma del contenuto adoperando la chiave pubblica ottenuta dall'Entity Statement rilasciato dalla Trust Anchor per gli OP. Verifica la firma dell'Entity Configuration degli OP usando la chiave pubblica ottenuta dall'Entity Statement rilasciato dal TA.

Il RP applica infine le politiche pubblicate dal Trust Anchor sui Metadata dell'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 elementi che compongono la **Trust Chain**. Periodicamente il RP aggiorna i Metadata di tutti gli OP rinnovando la Trust Chain relativa a questi.

Ottenuti i Metadata finali di tutti i OpenID Connect Provider, il RP genera lo **SPID Button** o il **CIE id Button** e lo pubblica all'interno della pagina di autenticazione destinata agli utenti.

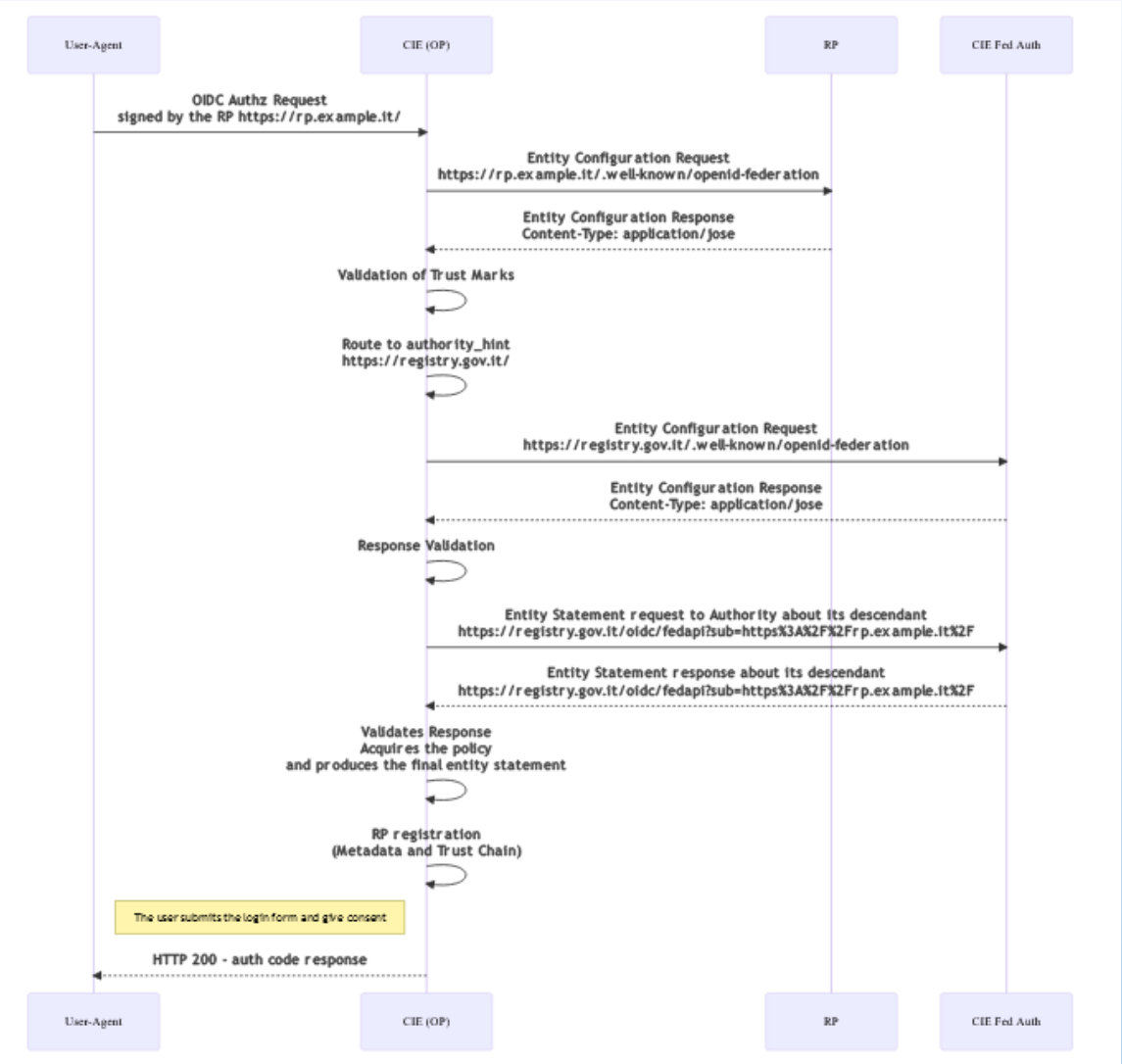
La procedura di Federation Entity Discovery risulta semplificata per i RP, perché all'interno della Federazione non è consentita l'esistenza di Intermediari tra gli OP ed il loro Trust Anchor.



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

Richiesta Autorizzazione

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.



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.

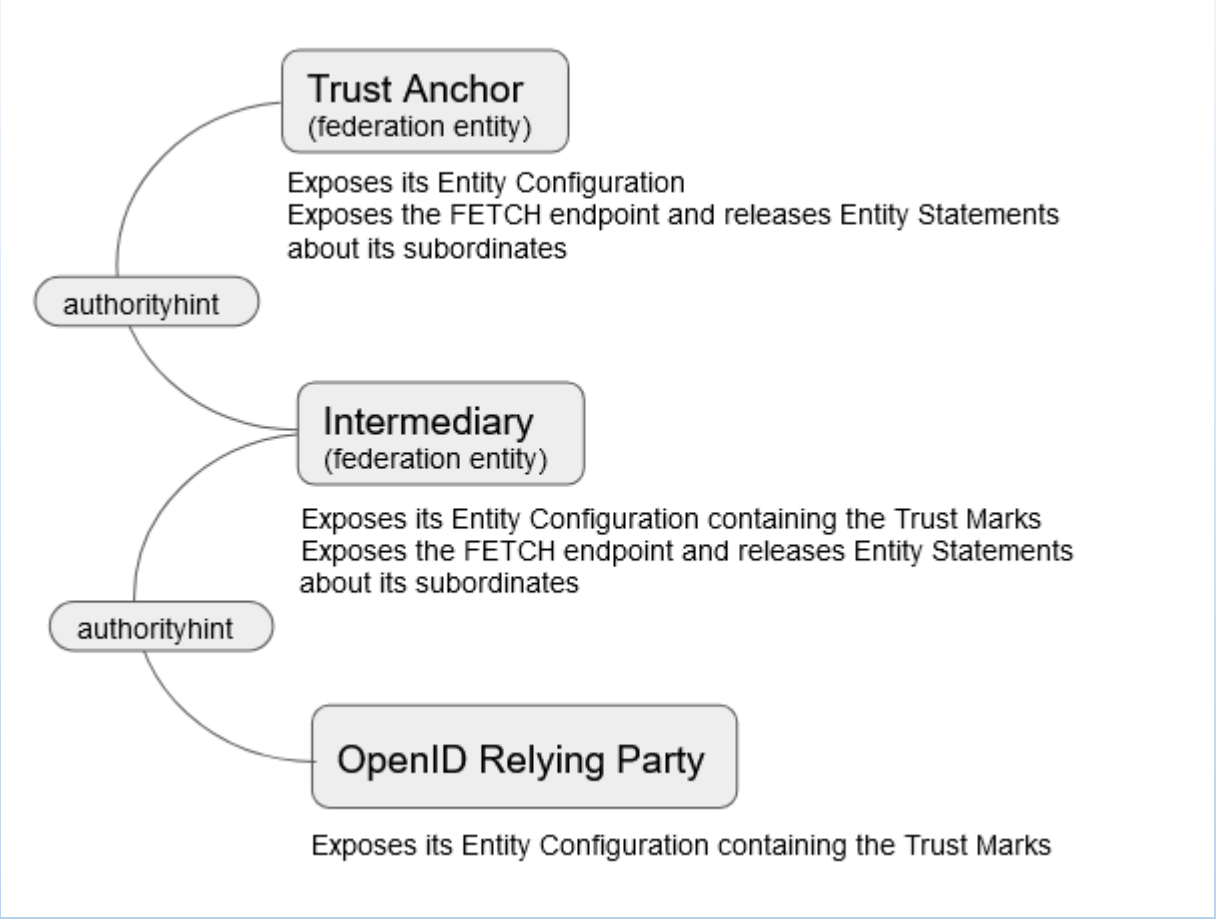
L'OP estrae l'identificativo univoco (**client_id**) dall'oggetto *request* contenuto all'interno della *Authorization Request* ed effettua una richiesta di Entity Configuration presso il RP. Ottiene l'Entity Configuration del RP e convalida la firma dei Trust Mark riconoscibili all'interno della Federazione

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

Durante il Federation Entity Discovery, il Provider richiede ad una o più entità superiori (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, sia questi SPID o CIE.) all'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.

Ottenuto il Metadata finale, il Provider valida la richiesta del RP

Nei casi in cui un RP avesse come entità superiore un SA e non direttamente il 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**, viene superata, l'OP blocca il processo di Federation Entity Discovery e rigetta la richiesta del RP.



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.

Accesso alla 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.

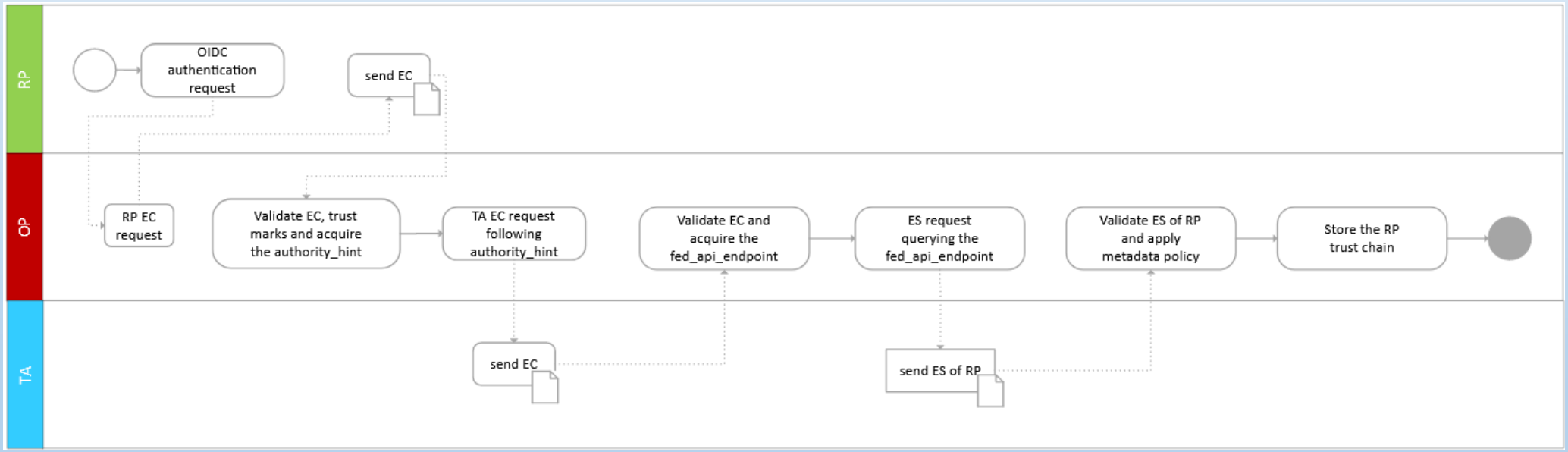
Esempi:

- 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 concatenare il web path della risorsa .well-known.

Una volta che un RP viene riconosciuto come parte della Federazione, ottiene il permesso di effettuare una Richiesta di Autenticazione. L'OP che non ha interagito prima d'ora con un RP che fa la richiesta, è in grado di risolvere la fiducia mediante l'API di federazione (Federation Entity Discovery e produzione della Trust Chain). L'OP inizia richiedendo la Entity Configuration del RP al .well-known endpoint del RP e, seguendo il percorso dato dall'*authority_hint*, raggiunge la radice del Trust, cioè il TA. In ogni passo della catena l'OP può eseguire tutti i controlli di sicurezza richiedendo le dichiarazioni di entità da ciascuna entità e convalidando i Trust Mark e le firme.

La figura che segue dà un esempio rappresentativo di come funziona la catena del Trust.



Endpoint di Federazione

Tutte le entità DEVONO contenere i seguenti endpoint:

/.well-known/openid-federation: fornisce l'Entity Configuration (per maggiori dettagli vedi OIDC-FED Section 6)

resolve entity statement endpoint: fornisce il metadata finale, la Trust Chain e i Trust Mark relativi ad un altro soggetto. Per maggiori dettagli vedi OIDC-FED Section 7.2.

Avvertimento

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.

Le Entità di tipo TA o SA DEVONO offrire i seguenti endpoint, in aggiunta agli endpoint di federazione sopra riportati:

- **fetch entity statement endpoint**: fornisce gli ES relativi ad un soggetto discendente diretto. Per ottenere un ES è necessario indicare almeno l'identificativo dell'entità di cui si vuole ottenere lo statement. (per maggiori dettagli consultare OIDC-FED Section 7.1)
- **trust mark status endpoint**: permette a un'entità di verificare se un TM è ancora attivo o no. La query DEVE essere inviata al soggetto che ha rilasciato quel TM. (per maggiori dettagli consultare OIDC-FED Section 7.4)
- **entity listing endpoint**: fornisce la lista delle entità discendenti registrate presso il TA o un SA (per maggiori dettagli consulatere OIDC-FED Section 7.3)

Un'entità di tipo AA, oltre agli endpoint di Federazione comuni a tutte le entità, DEVE riportare anche il trust mark status endpoint per consentire la validazione dinamica dei TM rilasciati dall'AA.

I webpath degli endpoint di Federazione DEVONO essere definiti nel modo seguente:

- */.well-known/openid-federation
- */fetch
- */resolve
- */trust_mark_status
- */list

Gestione degli errori di federazione

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

CLAIM	DESCRIZIONE	SUPPORTATO DA
Errore	Vedi Codici di errori	spid CIE
Descrizione dell'errore	Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging.	spid CIE

Codici di errore di Federation

ERRORE	DESCRIZIONE	CODICE HTTP	SUPPORTATO DA
temporarily_unavailable	Uno degli endpoint di well-known o di Federation non è raggiungibile.	*302 Found* or *400 Bad Request*	spid CIE
invalid_client	Il Client non è autorizzato perchè la validazione della Trust Chain fallisce.	*302 Found*	spid CIE
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.	*302 Found*	spid CIE
invalid_request	La richiesta non è completa o non è conforme a quanto definito dalle presenti specifiche tecniche.	*400 Bad Request*	spid CIE
not_found	La risorsa richiesta non è stata trovata.	*404 Not Found*	spid CIE

Metadata

OIDC-FED utilizza ed estende i claim dei Metadata così come definiti all'interno delle specifiche di OpenID Connect Discovery 1.0 (OpenID.Discovery) e OpenID Connect Dynamic Client Registration 1.0 (OpenID.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 (OP)
- OpenID Connect Relying Party Metadata (RP)
- Metadata di Trust Anchor (TA) e Intermediari (SA)
- Metadata Attribute Authority

OpenID Connect Provider Metadata (OP)

Un OP DEVE pubblicare all'interno del suo EC un Metadata da *federation_entity* e uno da *openid_provider* come riportato nell'esempio di EC per OP: https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/esempi.html#esempio-en1-2

Il metadata di tipo **"federation_entity"** e contenere almeno i seguenti parametri obbligatori

Il metadata di tipo **«openid_provider"** e contenere almeno i seguenti parametri obbligatori

Elenco parametri obbligatori:

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_oidc_op.html

OpenID Connect Relying Party Metadata (RP)

Un RP DEVE pubblicare all'interno del suo EC un Metadata di tipo *federation_entity* e uno di tipo *openid_relying_party* come riportato nell'esempio di EC per OP: https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/esempi.html#esempio-en1-2

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

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_oidc_rp.html

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

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_oidc_rp.html

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

Un TA e un SA DEVONO pubblicare all'interno del loro EC un Metadata da *federation_entity* come riportato nell'esempio https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/esempi.html#esempio-en1-4 e https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/esempi.html#esempio-en1-3

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

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_oidc_ta_sa.html

Metadata Attribute Authority

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*.

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

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_aa.html

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

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_aa.html

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

https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/metadata_aa.html

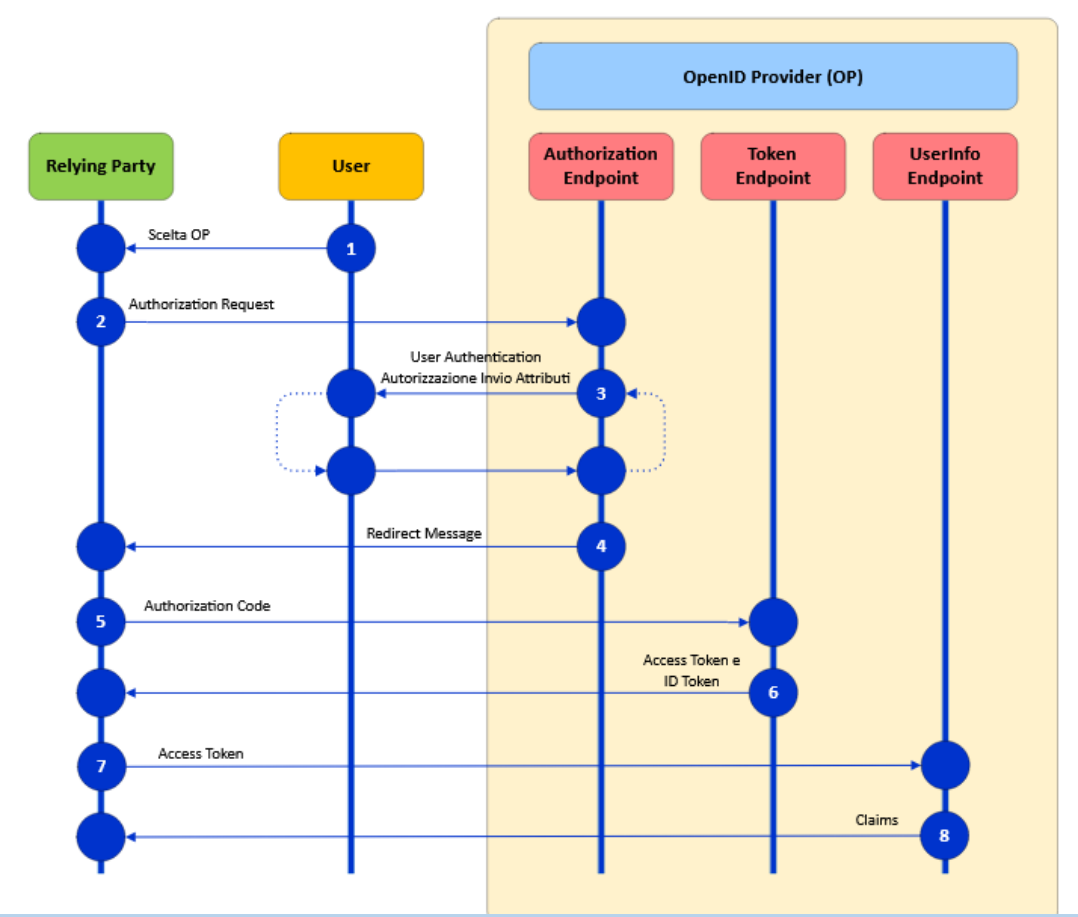
Flusso di autenticazione

Gli schemi di autenticazioni **"Entra con SPID"** e **"Entra con CIE"** implementano il flusso **OpenID Connect Authorization Code Flow** con l'estensione **PKCE** (Proof Key for Code Exchange, RFC 7636). Questo flusso restituisce un **Authorization Code** che può essere utilizzato per ottenere un **ID Token** e un **Access Token** e se possibile anche un **Refresh Token**. L'**Authorization Code Flow** ottiene l'**Authorization Code** dall'*Authorization Endpoint* dell'OpenID Provider e tutti i token sono restituiti dal **Token Endpoint**.

Flusso di autenticazione

PKCE è un'estensione del protocollo *OAuth 2.0* prevista anche nel profilo *iGov* (International Government Assurance Profile for OAuth 2.0) e finalizzata ad evitare un potenziale attacco attuato con l'intercettazione dell'*authorization code*. Consiste nella generazione di un codice (**code verifier**) e del suo hash (**code challenge**). Il **code challenge** viene inviato all'OP nella richiesta di autenticazione.

Quando il RP contatta il *Token Endpoint* al termine del flusso di autenticazione, invia il **code verifier** originariamente creato, in modo che l'OP possa confrontare che il suo hash corrisponda con quello acquisito nella richiesta di autenticazione.



1. L'Utente, nella pagina di accesso del Relying Party (RP):
   - Seleziona il pulsante "Entra con SPID" o "Entra con CIE";
   - Nel caso SPID, seleziona l'OP con cui autenticarsi.
2. Il RP prepara una Richiesta di Autorizzazione con i parametri necessari previsti da PKCE e la invia all'Authorization Endpoint dell'OP.
3. L'OP autentica l'utente mediante l'inserimento delle credenziali e ottiene il consenso per l'accesso agli attributi dell'utente da parte del RP.
4. L'OP reindirizza l'utente all'URL contenuto nel parametro redirect_uri specificato dal RP, passando un Authorization Code nell'Authorization Response.
5. Il RP invia l'Authorization Code ricevuto al Token Endpoint dell'OP.
6. Il Token Endpoint dell'OP rilascia un ID Token, un Access Token e se previsto un Refresh Token.
7. Il RP riceve e valida l'Access Token e l'ID Token. Per chiedere gli attributi che erano stati autorizzati dall'utente al punto 3, invia una richiesta all'UserInfo Endpoint dell'OP utilizzando l'Access Token per l'autenticazione all'interno della intestazione HTTP Authorization.
8. Lo UserInfo Endpoint dell'OP verifica la validità dell'Access Token e rilascia gli attributi richiesti al RP.

Authorization endpoint (Authentication)

Request

Per avviare il processo di autenticazione, il RP reindirizza l'utente all'Authorization Endpoint dell'OP selezionato, inviando una richiesta HTTP contenente il parametro request in formato JWT firmato e contenente l'Authorization Request firmata dal RP.

Per veicolare la richiesta, il RP PUÒ utilizzare i metodi POST e GET. Mediante il metodo POST i parametri DEVONO essere trasmessi utilizzando la Form Serialization. Mediante il metodo GET i parametri DEVONO essere trasmessi utilizzando la Query String Serialization. Per maggiori dettagli vedi OpenID.Core#Serializations.

Nota:

Il parametro scope DEVE essere trasmesso sia come parametro nella chiamata HTTP sia all'interno dell'oggetto request e i loro valori DEVONO corrispondere.

I parametri client_id e response_type DOVREBBERO essere trasmessi sia come parametri sulla chiamata HTTP sia all'interno dell'oggetto request.

I parametri client_id e response_type DEVONO essere trasmessi sia come parametri sulla chiamata HTTP sia all'interno dell'oggetto request e i loro valori DEVONO corrispondere, in caso contrario solo i parametri all'interno dell'oggetto request DEVONO essere considerati.

Authorization Request


```
{
"alg": "RS256",
"kid": "2HnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
}
.
{
    "client_id": "https://rp.spid.agid.gov.it",
    "response_type": "code",
    "scope": "openid",
    "code_challenge": "qWJlMe0xdbXrKxTm72EpH659bUxAxw80",
    "code_challenge_method": "S256",
    "nonce": "MBzGqyf9QytD28eupyWhSqMj78WNqpc2",
    "prompt": "login",
    "redirect_uri": "https://rp.spid.agid.gov.it/callback1",
    "acr_values": {
      "https://www.spid.gov.it/SpidL1":null,
      "https://www.spid.gov.it/SpidL2":null
    },
    "claims": {
      "userinfo": {
          "given_name":null,
          "family_name":null
      }
    },
    "state": "fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd"
}
```

i parametri obbligatori nella richiesta di autenticazione *HTTP*.

PARAMETRO	DESCRIZIONE	SUPPORTATO DA
scope	Riporta di valori di *scope* supportati dall'OP e definiti dal parametro **scopes_supported** nel Metadata OP. DEVE essere presente almeno il valore *openid*.	spid CIE
code_challenge	Vedi RFC 7636#section-4.2.	spid CIE
code_challenge_method	Come definito dal parametro **code_challenge_methods_supported** nel Metadata OP.	spid CIE
request	Vedi OpenID.Core#JWTRequests. DEVE essere un **JWT** firmato.	spid CIE

composizione dell'header del **JWT**

JOSE HEADER	DESCRIZIONE	SUPPORTATO DA
alg	Vedi RFC 7516#section-4.1.1. Vedi Algoritmi crittografici..	spid CIE
kid	Vedi RFC 7638#section_3.	spid CIE

Nota

Il parametro **typ** se omesso assume il valore implicito di **JWT**.

Response

Un'Authentication response è un messaggio di risposta di autorizzazione OAuth 2.0 restituito dall'authorization endpoint dell'OpenID Provider (OP) al termine del flusso di autenticazione. L'OP reindirizzerà l'utente all'url contenuto nel parametro redirect_uri specificato nella richiesta di autorizzazione, aggiungendo i parametri della risposta.

Se l'autenticazione è avvenuta con successo, l'OpenID Provider (OP), reindirizza l'utente aggiungendo i seguenti parametri obbligatori come query parameters al redirect_uri (come definito in OpenID.Core#AuthResponse):

CLAIM	DESCRIZIONE	SUPPORTATO DA
code	Codice univoco di autorizzazione (*Authorization Code*) che il client può passare al Token Endpoint per ottenere un ID Token e un Access Token. Questo ha il vantaggio di non esporre alcun token allo User Agent o a malware che controllano questo.	spid CIE
state	Valore state incluso nell'*Authentication Request*. Il client è tenuto a verificarne la corrispondenza. Deve essere lo stesso valore indicato dal client nella Authorization Request.	spid CIE
iss	Identificatore univoco dell'OP che ha creato l'Authentication Response. Il RP DEVE validare questo parametro e NON DEVE permettere a più OP di usare lo stesso identificatore.	CIE

```
https://rp.spid.agid.gov.it/resp?
  code=usDwMnEzJPpG5oaV8x3j&
  state=fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd
```

Gestione degli errori

In caso di errore, l'OP o il RP rappresentano i messaggi di anomalia relativi agli scambi OpenID Connect, come descritti nelle relative tabelle definite dalle Linee Guida UX SPID.

CLAIM	DESCRIZIONE	SUPPORTATO DA
Errore	Vedi Codici di errori	spid CIE
Descrizione dell'errore	Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging. Questo messaggio non è destinato ad essere visualizzato all'utente (a tal fine si faccia riferimento alle Linee Guida UX SPID)	spid CIE
state	Parametro obbligatorio solo nel caso di risposta di errore alla *Authentication Request* e DEVE essere uguale al valore *state* incluso nella *Authentication Request*. Il RP DEVE verificare che corrisponda a quello inviato nella *Authentication Request*.	spid CIE

Codici di errore

ERRORE	DESCRIZIONE	CODICE HTTP	SUPPORTATO DA
access_denied	L'OP ha negato l'accesso a causa di credenziali non valide o non adeguate al livello SPID richiesto (RFC 6749#section-4.1.2.1).	*302 Found*	spid CIE
unauthorized_client	Il client non è autorizzato a richiedere un authorization code (RFC 6749#section-4.1.2.1).	*302 Found*	spid CIE
invalid_request	La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri (RFC 6749#section-4.1.2.1).	*302 Found*	spid CIE
invalid_scope	Sono stati richiesti degli scope non validi (RFC 6749#section-4.1.2.1).	*302 Found*	spid CIE
server_error	L'OP ha riscontrato un problema interno (RFC 6749#section-4.1.2.1).	*302 Found*	spid CIE
temporarily_unavailable	L'OP ha riscontrato un problema interno temporaneo (RFC 6749#section-4.1.2.1).	*302 Found*	spid CIE
unsupported_response_type	Il response_type richiesto non è	*302 Found*	spid CIE

In caso di URI di reindirizzamento non valido, non corrispondente o mancante, l'OP restituisce *400 Bad Request* come codice HTTP.

Authorization endpoint (Authentication)

- https://docs.italia.it/italia/spid/spid-cie-oidc-docs/it/versione-corrente/authorization_endpoint.html

Token Endpoint

Al termine del flusso di autenticazione descritto precedentemente, il RP invia una richiesta al Token Endpoint inviando l'authorization code ricevuto dall'OP per ottenere un ID Token e un Access Token ed eventualmente un Refresh Token (se è stata effettuata una richiesta di autenticazione con scope=offline_access e prompt=consent)

Token Request

CLAIM	DESCRIZIONE	SUPPORTATO DA
client_id	Vedi OpenID.Registration. DEVE essere valorizzato con un HTTPS URL che identifica univocamente il RP.	sp:d CIE ID
client_assertion	JWT firmato con la chiave privata del Relying Party contenente i seguenti parametri: iss: DEVE corrispondere al valore *client_id* sub: DEVE corrispondere al valore *iss* aud: URL del Token Endpoint dell'OP iat: UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519. exp: UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519 jti: Identificatore univoco per questa richiesta di autenticazione, generato dal client. Ad esempio in formato *uuid4*.	sp:d CIE ID
client_assertion_type	Deve assumere il seguente valore: **urn:ietf:params:oauth:client-assertion-type:jwt-bearer**	sp:d CIE ID
client_assertion_type	Deve assumere il seguente valore: **urn:ietf:params:oauth:client-assertion-type:jwt-bearer**	sp:d CIE ID
code	Codice di autorizzazione restituito nell'Authentication response. Obbligatorio solo se **grant_type** è **authorization_code**	sp:d CIE ID
code_verifier	Codice di verifica del code_challenge. Obbligatorio solo se **grant_type** è **authorization_code**	sp:d CIE ID
grant_type	Tipo di credenziale presentata dal RP per la richiesta corrente. PUÒ assumere uno dei seguenti valori: • authorization_code • refresh_token	sp:d CIE ID
refresh_token	Obbligatorio solo se **grant_type** è **refresh_token**	sp:d CIE ID

Token Response

- 'OpenID Provider (OP) restituisce un ID Token e Access Token e un eventuale Refresh Token, in formato JWT firmato.
- L'Access Token deve essere formato secondo le indicazioni dello standard "International Government Assurance Profile (iGov) for OAuth 2.0 - Draft 03", section 3.2.1, "JWT Bearer Tokens".
- L'ID Token deve essere formato come indicato successivamente

La risposta DEVE contenere i seguenti claim.

CLAIM	DESCRIZIONE	SUPPORTATO DA
access_token	L'Access Token, in formato JWT firmato, consente l'accesso allo UserInfo endpoint per ottenere gli attributi.	spid CIE
token_type	Tipo di *Access Token* restituito. DEVE essere valorizzato sempre con **Bearer**	spid CIE
refresh_token	Disponibile sono nel caso di sessione lunga revocabile. Il *Refresh Token*, in formato JWT firmato, consente di chiamare nuovamente il Token Endpoint per ottenere un nuovo *Access Token* e un nuovo *ID Token*.	spid CIE
expires_in	Scadenza dell'*Access Token* in secondi.	spid CIE
id_token	ID Token in formato JWT (vedi paragrafo successivo)	spid CIE

HTTP/1.1 200 OK
Last-Modified: Wed, 22 Jul 2018 19:15:56 GMT
Content-Type: application/json

{
"access_token":"dC34Pf6kdG...",
"token_type":"Bearer",
"refresh_token":"wJ848BcyLP...",
"expires_in":1800,
"id_token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY..."
}

Access Token

L'Access Token è un JSON Web Token (JWT) firmato che consente l'accesso allo UserInfo endpoint per ottenere gli attributi dell'utente. Di seguito i claim che compongono l'Access Token.

CLAIM	DESCRIZIONE	SUPPORTATO DA
iss	DEVE essere valorizzato con un HTTPS URL che identifica univocamente l'OP. Il client DEVE verificare che questo valore corrisponda all'OP chiamato.	spid CIE
sub	Vedi OpenID.Core#SubjectIDTypes. DEVE essere di tipo *pairwise*.	spid CIE
client_id	DEVE essere valorizzato con un HTTPS URL che identifica univocamente il RP.	spid CIE
aud	DEVE contenere un elenco di Resource Server che consumano l'AT. DEVE contenere almeno lo *UserInfo Endpoint*.	spid CIE
scope	L'OP DOVREBBE inserire il parametro *scope* come previsto in RFC 9068 Sezione 2.2.3. DEVE coincidere con il valore presente in fase di richiesta di autenticazione.	spid CIE
iat	UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE
exp	UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE
jti	DEVE essere una Stringa in formato *uuid4*. Identificatore unico dell'ID Token che il RP PUÒ utilizzare per prevenirne il riuso, rifiutando l'ID Token se già processato.	spid CIE

ID Token

L'ID Token è un JSON Web Token (JWT) firmato che contiene informazioni sull'utente che ha eseguito l'autenticazione.
I RP DEVONO eseguire la validazione dell'ID Token.

Il RP PUÒ richiedere che L'ID Token sia cifrato (vedere il parametro id_token_encrypted_response_alg nel Metadata RP). Se il RP inserisce nel suo metadata il parametro id_token_encrypted_response_alg, l'OP DEVE restituire l'ID Token firmato e cifrato. L'ID Token in formato JWT DEVE contenere il parametro cty (Content-Type) nell'intestazione JOSE con il valore JWT (vedere RFC 7519#section-5.2).

claim disponibili nell'ID Token

iss	DEVE essere valorizzato con un HTTPS URL che identifica univocamente l'OP. Il client DEVE verificare che questo valore corrisponda all'OP chiamato.	spid CIE
sub	Vedi OpenID.Core#SubjectIDTypes. DEVE essere di tipo *pairwise*.	spid CIE
aud	DEVE coincidere con il valore *client_id*. Il RP DEVE verificare che questo valore corrisponda al proprio client ID.	spid CIE
acr	Livello di autenticazione effettivo. DEVE essere uguale o superiore a quello richiesto dal RP nella Authentication Request.	spid CIE
at_hash	Vedi OpenID.Core#CodeIDToken. Il suo valore è la codifica base64url della prima metà dell'hash calcolato sulla rappresentazione ASCII dell'*Access Token*, usando l'algoritmo di hashing indicato in **alg** nell'header dell'ID Token. Il client DEVE verificare che questo valore corrisponda applicando la medesima funzione all'*Access Token* restituito insieme all'ID Token.	spid CIE
iat	UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE
nbf	UNIX Timestamp. Istante di inizio validità del JWT in formato NumericDate, come indicato in RFC 7519. DEVE corrispondere con il valore di **iat**.	spid
exp	UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE
jti	DEVE essere una Stringa in formato *uuid4*. Identificatore unico dell'ID Token che il RP PUÒ utilizzare per prevenirne il riuso, rifiutando l'ID Token se già processato.	spid CIE
nonce	Vedi OpenID.Core#AuthRequest. DEVE essere una stringa casuale di almeno 32 caratteri alfanumerici. Questo valore DEVE coincidere con quello inviato dal RP nella richiesta di autenticazione.	spid CIE

Refresh Token

Il *Refresh Token* è un JWT che PUÒ essere rilasciato dall'OP e che PUÒ essere usato per ottenere un nuovo *Access Token* che abilita il RP ad accedere allo *UserInfo endpoint* senza interazione diretta dell'utente.

Il *Refresh Token* DEVE essere rilasciato in formato JWT, firmato, e contenere almeno i seguenti parametri.

CLAIM	DESCRIZIONE	SUPPORTATO DA
iss	DEVE essere valorizzato con un HTTPS URL che identifica univocamente l'OP. Il RP DEVE verificare che questo valore corrisponda all'OP chiamato.	spid CIE ID
client_id	DEVE coincidere con il valore *client_id*. Il RP DEVE verificare che questo valore corrisponda al proprio client ID.	spid CIE ID
aud	DEVE contenere il *Token Endpoint* dell'OP.	spid CIE ID
iat	UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE ID
exp	UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519	spid CIE ID
jti	DEVE essere una Stringa in formato *uuid4*. Identificatore unico del *Refresh Token* che il RP PUÒ utilizzare per prevenirne il riuso, rifiutando il *Refresh Token* se già processato.	spid CIE ID

Periodo di validità di un Refresh Token

Il *Refresh Token* NON DEVE avere una validità (differenza tra *iat* e *exp*) superiore a 30 giorni.

Se allo scadere del periodo di validità l'RP effettua una richiesta all'OP, quest'ultimo DEVE restituire un errore nella risposta

UserInfo Endpoint

Lo UserInfo Endpoint è una risorsa protetta che restituisce gli attributi dell'utente autenticato. Per ottenere gli attributi richiesti, il RP inoltra una richiesta allo UserInfo Endpoint utilizzando l'Access Token.

Request

Lo UserInfo Endpoint DEVE supportare l'uso del solo metodo HTTP GET e DEVE accettare e validare l'Access Token inviato all'interno del campo Authorization dell'Header, di tipo Bearer.

Lo UserInfo Endpoint DEVE supportare l'uso dei metodi HTTP GET e POST e DEVE accettare e validare l'Access Token inviato all'interno del campo Authorization dell'Header, di tipo Bearer

Response

La response dello UserInfo Endpoint DEVE specificare nel "Content-Type" il valore "application/jwt".

Il contenuto del corpo della Response DEVE essere un JWT firmato e cifrato..

L'header JOSE DEVE contenere il parametro *cty* (Content Type) valorizzato con *JWT* (vedi **RFC 7519#section-5.2**).

Lo UserInfo Endpoint restituisce gli attributi utente esplicitamente richiesti tramite il parametro **claims** o tramite l'utilizzo del parametro **scope** nella Authentication Request.

Esempio:

```
HTTP/1.1 200 OK
Last-Modified: Wed, 22 Jul 2018 19:15:56 GMT
Content-Type: application/jose

{
  "alg": "RSA-OAEP",
  "enc": "A256CBC-HS512",
  "kid": "HIvo33-Km7n03ZqKDJfWVnlFudsW28YhQZx5eaXtAKA",
  "cty": "JWT"
}
.
{
   "iss": "https://op.fornitore_identita.it",
   "aud": "https://rp.fornitore_servizio.it",
   "iat": 1519032969,
   "nbf": 1519032969,
   "exp": 1519033149,
   "sub": "OP-1234567890",
   "name": "Mario",
   "family_name": "Rossi",
   "https://attributes.spid.gov.it/fiscal_number": "MROXXXXXXXXXXXXX"
}
```

L'intestazione del JWE DEVE contenere i seguenti parametri:

CLAIM	DESCRIZIONE	SUPPORTATO DA
alg	String. Vedi Algoritmi crittografici..	spid CIE ID
kid	Vedi RFC 7638#section_3.	spid CIE ID
enc	String. Vedi Algoritmi crittografici..	spid CIE ID
cty	String. DEVE essere valorizzato con "JWT".	spid CIE ID

Il payload del JWE è un JWS contenente all'interno del suo payload i seguenti parametri:

CLAIM	DESCRIZIONE	SUPPORTATO DA
sub	String. Identificatore del soggetto, coincidente con quello già rilasciato nell'ID Token. Il RP DEVE verificare che il valore coincida con quello contenuto nell'ID Token.	sp:d CIE ID
iat	UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in RFC 7519.	sp:d CIE ID
exp	UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in RFC 7519.	sp:d CIE ID
aud	String. Identificatore del soggetto destinatario della response (RP). Il RP DEVE verificare che il valore coincida con il proprio client_id.	sp:d CIE ID
iss	String. URI che identifica univocamente l'OP.	sp:d CIE ID
<attributo>	I claim richiesti al momento dell'autenticazione.	sp:d CIE ID

Codici di errore
Come definiti per Token endpoint.

L'intestazione del JWS DEVE contenere i seguenti parametri:

CLAIM	DESCRIZIONE	SUPPORTATO DA
alg	String. Vedi Algoritmi crittografici..	sp:d CIE ID
kid	Vedi RFC 7638#section_3.	sp:d CIE ID
cty	String. DEVE essere valorizzato con "JWT".	sp:d CIE ID

Tabella attributi utente

La seguente tabella riporta l'elenco degli attributi utente supportati da SPID e/o CIE. La variable $PREFIX=https://attributes.eid.gov.it rappresenta il namespace

CLAIM	DESCRIZIONE	SUPPORTATO DA
$PREFIX/spid_code Categoria: anagrafica	Codice identificativo. String. Il codice identificativo è assegnato dal gestore dell'identità digitale e deve essere univoco. Il formato è il seguente: `<codice_Identificativo>=<cod_IdP> <nr.univoco>` Dove: <cod_IdP>: è un codice composto da 4 lettere univocamente assegnato al gestore delle identità; <nr.univoco>: è una stringa alfanumerica composta da 10 caratteri che il gestore delle identità genera in maniera univoca nell'ambito del proprio dominio. Esempio: `"$PREFIX/spid_code":"ABCD123456789A"`	spid

given_name
Categoria: anagrafica

Nome. String. Stringa composta da una sequenza di parole con carattere iniziale maiuscolo, intervallate da spazi singoli.

Esempio:

```
"given_name":"Giovanni Mario"
```

family_name
Categoria: anagrafica

Cognome. String. Stringa composta da una sequenza di parole con carattere iniziale maiuscolo, intervallate da spazi singoli.

Esempio:

```
"family_name":"Bianchi Verdi"
```

place_of_birth
Categoria: anagrafica

Luogo di nascita, Provincia di nascita. JSON Object:

"locality : Stringa corrispondente al codice catastale (Codice Belfiore) del Comune o della nazione estera di nascita (Es. "F205" per la città di Milano)

"region" : Stringa corrispondente alla sigla della provincia di nascita

Esempio:

```
"place_of_birth":{
    "region":"MI",
    "locality":"F205"
}
```

birthdate Categoria: anagrafica	Data di nascita. String. Secondo specifica ISO8601-2004 nel formato YYYY indica l'anno utilizzando 4 cifre MM indica il mese in (due) cifre DD indica il giorno in (due) cifre Esempio: `"birthdate":"2002-09-24"`	sp:d CIE ID
gender Categoria: anagrafica	Sesso. String. Valori ammessi: "female" per sesso femminile "male" per sesso maschile Esempio: `"gender":"female"`	sp:d CIE ID
$PREFIX/company_name Categoria: anagrafica	Ragione o denominazione sociale. String. Stringa composta da una sequenza di parole intervallate da spazi singoli. In maiuscolo le sottostringhe corrispondenti a nomi (es. "Agenzia per l'Italia Digitale") `"$PREFIX/company_name": "Agenzia per l'Italia Digitale"`	sp:d

$PREFIX/registered_office
Categoria: extra anagrafica

Sede legale. JSON Object: formatted, street_address, locality, region, postal_code, country, country_code. Json composto da una stringa composta da una sequenza di parole intervallate da spazi singoli rappresentanti:

- Tipologia(via, viale, piazza ...)
- Indirizzo
- Nr.civico
- CAP
- Luogo
- Provincia

la stringa è inserita nel claim "formatted" del JSON Object "address"

Esempio:

```
"$PREFIX/registered_office":{
    "formatted":"via Listz 21 00144 Roma"
}
```

sp:d

$PREFIX/fiscal_number Categoria: anagrafica	Codice fiscale della persona fisica. String. Per il formato si faccia riferimento alla codifica dell'attributo CF per i certificati, proposta nell'ambito del Draft ETSI EN 319 412-1, che nel caso specifico prevede la seguente composizione: TINIT-<CodiceFiscale> Esempio: `"$PREFIX/fiscal_number":"TINIT-ABCXYZ00W00Z000Z"`	sp:d CIE ID

$PREFIX/company_fiscal_number
Categoria: anagrafica
	Codice fiscale Persona Giuridica. String. Per il formato si faccia riferimento alla codifica dell'attributo CF per i certificati, proposta nell'ambito del Draft ETSI EN 319 412-1, che nel caso specifico prevede la seguente composizione:
	
	`TINIT-segue` `il` `codice` `fiscale`
	
	Esempio:
	
	`"$PREFIX/company_fiscal_number":"TINIT-ABCXYZ00W00Z000Z"`
		sp:d

$PREFIX/vat_number
Categoria: anagrafica
	Partita IVA. String. Per il formato si faccia riferimento alla codifica dell'attributo Partita IVA per i certificati, proposta nell'ambito del Draft ETSI EN 319 412-1, che nel caso specifico prevede la seguente composizione:
	
	`VATIT-<PartitaIVA>`
	
	Esempio:
	
	`"$PREFIX/vat_number":` `"VATIT-12345678901"`
		sp:d

document_details
Categoria: extra anagrafica

Documento d'identità. JSON Object (document):

Json contenente le proprietà che rappresentano:

- "type" : valori ammessi:
  - *cartaIdentita, passaporto, patenteGuida,*
  - *patenteNautica, librettoPensione,*
  - *patentinoImpTermici, portoArmi,*
  - *tesseraRiconoscimento;*
- "document_number" : Numero del documento;
- "issuer" : <ente emettitore> JSON Object:
  - "name" stringa ottenuta dalla concatenazione dei termini costituenti la denominazione dell'ente a meno di congiunzioni, articoli e preposizioni.

    Es. regioneLazio (Regione Lazio); provinciaCatania (Provincia di Catania); prefetturaRoma (Prefettura di Roma); MinisteroEconomiaFinanze (Ministero dell'Economia e delle Finanze);
- "date_of_issuance" : data di rilascio del documento;
- "date_of_expiry" : data di scadenza del documento;

Esempio:

```
"document_details":{
    "type":"cartaIdentita",
    "document_number":"AS09452389",
    "issuer":{
        "name":"ComuneRoma"
    },
    "date_of_issuance":"2013-01-02",
    "date_of_expiry":"2013-01-31"
}
```

phone_number Categoria: extra anagrafica	Numero di telefono mobile. String. Stringa numerica senza spazi intermedi Esempio: `"phone_number":"12345678901"`	spid CIE
phone_number_verified Categoria: extra anagrafica	Valore Booleano che indica se il numero di telefono mobile dell'utente è stato verificato dall'OP.	CIE
$PREFIX/landline_number Categoria: extra anagrafica	Numero di telefono fisso. String. Stringa numerica senza spazi intermedi Esempio: `"$PREFIX/landline_number":"12345678901"`	CIE
email Categoria: extra anagrafica	Indirizzo di posta elettronica. String. Formato standard indirizzo di posta elettronica Esempio: `"email":"name@domain.it"`	spid CIE
email_verified Categoria: extra anagrafica	Valore Booleano che indica se l'email dell'utente è stata verificata dall'OP.	CIE
$PREFIX/e_delivery_service Categoria: extra anagrafica	Domicilio digitale. Indirizzo casella PEC Esempio: `"$PREFIX/e_delivery_service":"nome@pecdomain.it"`	spid CIE
$PREFIX/eid_exp_date Categoria: extra anagrafica	Data di scadenza identità. Secondo specifica ISO8601-2004 nel formato "YYYY-MM-DD" dove YYYY indica l'anno utilizzando 4 cifre MM indica il mese in (due) cifre DD indica il giorno in (due) cifre Esempio: `"$PREFIX/eid_exp_date":"2002-09-24"`	spid

address

Categoria: extra anagrafica

JSON Object (address):

- "street_address": L'attributo contiene la tipologia (via, viale, piazza ...), l'indirizzo e il numero civico. Le tre informazioni sono preferibilmente ordinate come d'uso per lo specifico Stato.
- "postal_code": CAP
- "locality": Comune
- "region": Provincia
- "country_code": Nazione

Esempio:

```
"address":{
    "street_address":"Via Liszt 21",
    "postal_code":"00144",
    "locality":"Roma",
    "region":"RM",
    "country_code":"IT"
}
```

ESEMPI

Si riportano gli esempi che danno luogo alla composizione di un unico JSON Object da parte di più attributi ed in particolare i claim "place_of_birth", "address", "document_details", $PREFIX/registered_office.

Si riportano a titolo di esempio due indirizzi italiani

ATTRIBUTO	ESEMPIO CODIFICA OIDC

Indirizzo domicilio fisico CAP domicilio fisico Comune domicilio fisico Provincia domicilio fisico Nazione domicilio fisico	<pre>"address": {
	"street_address":"Via Liszt 21",
	"postal_code":"00144",
	"locality":"Roma",
	"region":"RM",
	"country_code":"IT"
	}</pre>

Indirizzo domicilio fisico CAP domicilio fisico Comune domicilio fisico Provincia domicilio fisico Nazione domicilio fisico	<pre>"address": {
	"street_address":"S.S. Salaria Km 23,800",
	"postal_code":"00015",
	"locality":"Monterotondo",
	"region":"RM",
	"country_code":"IT"
	}</pre>

Vi sono casi, come per gli Stati Uniti d'America, dove oltre alla nazione (US) esiste uno Stato. In tali casi lo Stato è indicato nel campo Provincia. Si riporta il seguente esempio:

ATTRIBUTO	ESEMPIO CODIFICA OIDC

Indirizzo domicilio fisico CAP domicilio fisico Comune domicilio fisico Provincia domicilio fisico Nazione domicilio fisico	<pre>"address":{
	"street_address":"503,Washington Avenue",
	"postal_code":"12401",
	"locality":"Kingston",
	"region":"New york",
	"country_code":"US"
	}</pre>

Introspection Endpoint (verifica validità token)

L'Introspection Endpoint esposto dall'OP consente ai RP di ottenere informazioni su un token in loro possesso, come ad esempio la sua validità.

Request

La richiesta all'Introspection Endpoint consiste nell'invio del token su cui si vogliono ottenere informazioni unitamente a una Client Assertion che consente di autenticare il RP che esegue la richiesta.

```
POST /introspection HTTP/1.1
Host: https://op.spid.agid.gov.it
Content-Type: application/x-www-form-urlencoded

client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM
ibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9q7oiXcYVIIqGWY0wWQlq
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
token=eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTQsImF1ZCI6WyJlNzFmYj
QwMDEtYmNiNy1lNjdjMmJjMDAzN2YiXSwiaXNzIjoiaHR0cHM6XC9cL2FzLXZhLmV4YW1
iLCJqdGkiOiIyMWIxNTk2ZC04NWQzLTQzN2MtYWQ4My1iM2YyY2UyNDcyNDQiLCJpYXQi
MTR9.FXDtEzDLbTHzFNroW7w27RLk5m0wprFfFH7h4bdFw5fR3pwiqejKmdfAbJvN3_yf
RJUbdjmFFfRRW23cMbpGQCIk7Nq4L012X_1J4IewOQXXMLTyWQQ_BcBMjcW3MtPrY1AoO
RkYtyVTLWlff6S5gKciYf3b0bAdjoQEHd_IvssIPH3xuBJkmtkrTlfWR0Q0pdpeyVePkM
h70xQLVzqwwl1P0-F_0JaDFMJFO1yl4IexfpoZZsB3HhF2vFdL6D_lLeHRyH2g2OzF59e
```

CLAIM	DESCRIZIONE	SUPPORTATO DA
client_assertion	JWT firmato con la chiave privata del Relying Party contenente gli stessi parametri documentati per le richieste al Token Endpoint. L'OP deve verificare la validità di tutti i campi presenti nel JWT, nonché la validità della sua firma in relazione al parametro **client_id**.	sp:d CIE
client_assertion_type	String. Valori ammessi: **urn:ietf:params:oauth:clientassertion-type:jwt-bearer**	sp:d CIE
client_id	URI che identifica univocamente il RP. L'OP deve verificare che il client_id sia noto all'interno della Federazione.	sp:d CIE
token	Il token su cui il RP vuole ottenere informazioni.	sp:d CIE

Response

L'Introspection Endpoint risponde con un oggetto JSON definito come segue.

Esempio:

```
{
    "active":true
}
```

CLAIM	DESCRIZIONE	SUPPORTATO DA
active	Valore booleano che indica la validità del token. Se il token è scaduto, è revocato o non è mai stato emesso per il client_id chiamante, l'Introspection Endpoint deve restituire false.	spid
scope	Lista degli scope richiesti al momento dell'Authorization Request.	spid
exp	Scadenza del token.	spid
sub	Identificatore del soggetto, coincidente con quello già rilasciato nell'ID Token. Il RP deve verificare che il valore coincida con quello contenuto nell'ID Token.	spid
client_id	URI che identifica univocamente il RP come da Registro SPID. Il RP deve verificare che il valore coincida con il proprio client_id.	spid
iss	Identificatore dell'OP che lo contraddistingue univocamente nella federazione nel formato Uniform Resource Locator (URL). Il client è tenuto a verificare che questo valore corrisponda all'OP chiamato.	spid
aud	Contiene il client ID. Il client è tenuto a verificare che questo valore corrisponda al proprio client ID.	spid

Codici di errore

Come definiti per [Token endpoint](#).

Revocation Endpoint

Un RP PUÒ chiedere la revoca di un Access Token o di un Refresh Token emesso da un OP. L'OP DEVE revocare il token

Request

La richiesta al Revocation Endpoint consiste nell'invio del token che si vuole revocare unitamente a una Client Assertion che consente di identificare il RP che esegue la richiesta.

Esempio:

```
POST /revoke HTTP/1.1
Host: https://op.spid.agid.gov.it
Content-Type: application/x-www-form-urlencoded

client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM
ibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9q7oiXcYVIIqGWY0wWQlq
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
token=eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTQsImF1ZCI6WyJlNzFmYj
QwMDEtYmNiNy1lNjdjMmJjMDAzN2YiXSwiaXNzIjoiaHR0cHM6XC9cL2FzLXZhLmV4YW1
iLCJqdGkiOiIyMWIxNTk2ZC04NWQzLTQzN2MtYWQ4My1iM2YyY2UyNDcyNDQiLCJpYXQi
MTR9.FXDtEzDLbTHzFNroW7w27RLk5m0wprFfFH7h4bdFw5fR3pwiqejKmdfAbJvN3_yf
RJUbdjmFFfRRW23cMbpGQCIk7Nq4L012X_1J4IewOQXXMLTyWQQ_BcBMjcW3MtPrY1AoO
RkYtyVTLWlff6S5gKciYf3b0bAdjoQEHd_IvssIPH3xuBJkmtkrTlfWR0Q0pdpeyVePkM
h70xQLVzqww11P0-F_0JaDFMJFO1yl4IexfpoZZsB3HhF2vFdL6D_lLeHRyH2g2OzF59e
```

CLAIM	DESCRIZIONE	SUPPORTATO DA
client_assertion	JWT firmato con la chiave privata OIDC del Relying Party contenente gli stessi parametri inseriti in fase di richiesta al Token Endpoint. L'OP deve verificare la validità di tutti i campi presenti nel JWT, nonché la validità della sua firma in relazione al parametro **client_id**.	spid CIE
client_assertion_type	String. **urn:ietf:params:oauth:clientassertion-type:jwt-bearer**	spid CIE
client_id	URL HTTPS che identifica univocamente il RP.	spid CIE
token	Il token che il RP chiede di revocare.	spid CIE

Response

Il Revocation Endpoint risponde con un codice HTTP 200, anche nel caso in cui il token indicato non esista o sia già stato revocato (in modo da non rilasciare informazioni).

Retention Policy
Gestione dei Log di un OP e di un RP

Gli OP e gli RP DEVONO mantenere:

Un registro delle transazioni contenente i log relativi ai messaggi scambiati. I messaggi memorizzati e mantenuti nel registro DEVONO essere almeno i seguenti:

- Trust Chain relativa all'Entità con la quale è avvenuta la transazione, composta da:
  - L'Entity Configuration del Entità con la quale è avvenuta la transazione.
  - [Solo per OP] L'Entity Statement del SA riferito al RP (se presente).
  - L'Entity Statement del TA riferito al suo discendente.
  - L'Entity Configuration del TA.
- AuthenticationRequest
- AuthenticationResponse relativa all'AuthenticationRequest
- TokenRequest relativa all'AuthenticationRequest
- TokenResponse relativa alla TokenRequest
- L'eventuale UserInfoRequest relativa alla TokenRequest
- L'eventuale UserInfoResponse relativa alla UserInfoRequest
- L'eventuale RevocationRequest relativa alla TokenRequest
- L'eventuale RevocationResponse relativa alla RevocationRequest

NOTA: Le informazioni contenute nei registri DEVONO essere mantenute e gestite per una durata non inferiore a 24 mesi nel pieno rispetto delle vigenti normative nazionali ed europee in materia di privacy. L'accesso ai dati DEVE essere riservato a personale incaricato. Al fine di garantire la confidenzialità DEVONO essere adottati meccanismi di cifratura dei dati o impiegati sistemi di basi di dati (DBMS) che realizzano la persistenza cifrata delle informazioni. Infine, nella memorizzazione dei dati DEVONO essere garantite le proprietà di integrità e non ripudio.

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-historical-jwks*.

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

Differenze tra SPID e CIE id

Metadata

Nei metadata OP e RP per CIE id sono presenti i parametri che abilitano la cifratura dell'ID Token (vedi le sezioni relative al Metadata OP e al Metadata RP). SPID non consente la cifratura dell'ID Token, dunque tali parametri non sono richiesti.

Inoltre, il metadata OP per CIE id richiede anche il parametro *revocation_endpoint_auth_methods_supported*, non richiesto da SPID.

Authorization Endpoint

SPID, al contrario di CIE id, prevede l'inserimento obbligatorio dei parametri *client_id* e *response_type* nella richiesta HTTP. Inoltre, CIE id prevede come obbligatorio il parametro *iss* nella response per mitigare gli attacchi di tipo mix-up I-D.ietf-OAuth-Security-BCP.

Parametri Scope e Claims

CIE id consente di richiedere gli attributi dell'utente sia tramite il parametro *claims* nella richiesta di autenticazione e sia tramite il parametro *scope*, abilitando in quest'ultimo i valori *profile* e *email*.

SPID non consente l'utilizzo di *profile* e *email* nel parametro *scope*.

Per ulteriori dettagli vedi la sezione Parametri Scope e claims.

ID Token

SPID non consente di rilasciare gli attributi dell'utente all'interno dell'ID Token. In CIE id gli attributi dell'utente sono disponibili sia nell'ID Token e sia nella UserInfo response. Inoltre, il CIE id supporta la criptazione dell'ID Token.

Refresh Token

SPID prevede l'utilizzo del Refresh Token per abilitare le sessioni lunghe rinnovabili così come definito nelle LL.GG. OpenID Connect in SPID e nell' Avviso n.41 . Consente, infatti, di ottenere, oltre all'Access Token, l'ID Token valido esclusivamente per SPID livello 1.

In CIE id il Refresh Token non consente di ottenere l'ID Token e non è utilizzabile dagli RP per ottenere una nuova autenticazione dell'utente con l'OP o rinnovare una sessione preesistente. In CIE id il Refresh Token è usato per ottenere dallo UserInfo endpoint esclusivamente il medesimo set di attributi dell'utente richiesti in fase di autenticazione iniziale, per il quale l'utente ha espresso il consenso esplicito. Per ulteriori dettagli si veda la sezione Refresh Token.

UserInfo Endpoint

CIE id supporta entrambi i metodi HTTP GET e HTTP POST per le richieste allo UserInfo endpoint. SPID consente solo l'utilizzo del metodo HTTP GET.

Introspection Endpoint

CIE id prevede il solo parametro *active* nella risposta dell'Introspection endpoint. SPID aggiunge ulteriori parametri come specificato nella sezione Introspection Endpoint.

Revocation Endpoint e Logout

Entrambi SPID e CIE id prevedono che il RP effettui una richiesta di revoca dell'Access Token in fase di logout dell'utente. In SPID la revoca di un Access Token implica anche la revoca dell'eventuale Refresh Token ancora attivo ad esso collegato e la scadenza della sessione di Single Sign-On se ancora attiva.

In CIE id, invece, la revoca di un Access Token non prevede la revoca del relativo Refresh Token, allo stesso tempo la richiesta di revoca di un Refresh Token determina anche la revoca di tutti i relativi token ancora attivi.

Differenze con OIDC iGov

CIE OpenID Connect e SPID OpenID Connect sono basati su iGov.OIDC con le seguenti differenze:

- La sezione 2.1 di iGov riporta vtr, acr_values e PKCE come OPZIONALI, sia in SPID che in CIE id PKCE e acr_values sono RICHIESTI. In entrambe le implementazioni di SPID e CIE, si è adottato acr_values al posto di vtr.
- L'Authentication Response nel flusso di autenticazione di CIE impone l'uso del claim iss per evitare l'attacco mix-up I-D.ietf-OAuth-Security-BCP. L'uso di questo claim è OPZIONALE in SPID.
- La sezione 2.4 di iGov stabilisce "Gli RP POSSONO opzionalmente mandare richieste all'Authorization Endpoint usando il parametro request." Sia in SPID che in CIE id, l'uso del parametro request è RICHIESTO.
- La sezione 3.1 di iGov stabilisce che "in caso di utilizzo di vtr nella richiesta di autenticazione, l'ID Token DEVE contenere i seguenti claim RICHIESTI, cioè: vot e vtm ". Considerando che vtr non è usato in SPID e CIE id, i claim appena citati non vengono inclusi all'interno dell'ID Token.
- La sezione 3.1 di iGov stabilisce che "il claim auth-time nell'ID Token è RACCOMANDATO". SPID e CIE id non adottano questo claim nell'ID Token.
- L'ID Token, sia in SPID che in CIE id, DEVE avere il claim acr RICHIESTO, mentre questo è opzionale nell'iGov draft iGov.
- L'ID Token, sia in SPID che in CIE id, ha il requisito del claim at_hash RICHIESTO. Questo è OPZIONALE in OIDC-CORE è assente in iGOV.
- Sia in SPID che in CIE id, l'identificatore del soggetto DEVE essere pairwised.
- La UserInfo Response, sia in SPID che in CIE id, DEVE essere un Nested JWT, firmato con la chiave privata dell'emettitore e cifrato con la chiave pubblica del RP.
- Il JWT firmato della UserInfo Response DEVE avere i claim iss, sub, aud, iat e exp.
- La sezione 3.4 di iGov stabilisce "Gli OpenID Provider POSSONO accettare oggetti request by reference usando il parametro request_uri". Questo parametro è intercambiabile con il parametro request. SPID e CIE id adottano solamente il parametro request.
- Sezione 3.8. La registrazione dinamica di iGOV specifica che la registrazione dinamica del client è obbligatoria. Sia in CIE id che in SPID, la registrazione automatica OIDC del client è OBBLIGATORIA, mentre la registrazione dinamica OIDC del client NON DOVREBBE essere supportata.
- Nella sezione 4.2 di iGOV gli scope openid, offline_access, profile e email vengono usati in CIE id OpenID Connect proposal e non considerano gli altri scope raccomandati nel profilo iGov, cioè: doc.
- Nella sezione 4.2 di iGOV gli scope openid, offline_access vengono usati in SPID OpenID Connect proposal e non considerano gli altri scope raccomandati nel profilo iGov, cioè: doc.
- La sezione 4.3 di iGov definisce la politica relativa all'oggetto userinfo del claim request. In CIE id, definiamo la politica per entrambi gli oggetti userinfo e ID Token.
- Nelle sezioni 3.7 e 2.5 di iGOV, i Metadata sia di SPID che di CIE id vengono distribuiti secondo le modalità definite nella sezione "3. Metadata".
- L'Access Token è un JWT firmato in conformità a RFC 9068.

Differenze con OIDC Federation

Differenze che intercorrono tra lo standard ufficiale e l'implementazione SPID e CIE.

Client Registration

SPID e CIE supportano esclusivamente **automatic_client_registration**. La modalità **explicit client registration** non è supportata.

Trust Mark

L'esposizione dei Trust Mark in SPID e CIE 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 e CIE non necessitano di alcun claim aggiuntivo in ambito federativo, non necessitano del 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.

Buone Pratiche

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.


JSON WEB TOKEN
JWT
SPIEGATO FACILE

OpenID Connect in SPID

SPID

SPID (Sistema Pubblico di Identità Digitale) è la chiave di accesso a tutti i servizi digitali della pubblica amministrazione ed a svariati del settore privato. Consiste in un singolo set di credenziali (username e password) che rappresenta l'identità digitale e personale dei cittadini italiani, consentendo loro di essere riconosciuti e di accedere in modo sicuro e personalizzato ai servizi digitali.

Il sistema di identificazione attraverso SPID si basa su tre diversi livelli di sicurezza progressivamente più rigorosi, richiesti dai servizi durante la fase di accesso e correlati al tipo di attività che l'utente sta per svolgere. Per ottenere le credenziali SPID esistono diversi IdP, ad esempio TIM id, POSTE id, aruba.it id, e ognuno di essi può garantire al cliente un certo livello di sicurezza raggiungibile.

I tre livelli sono:

1. Accesso con semplice nome utente e password (SPIDL1).
2. Stesso del punto 1 pi`u un codice temporaneo, un OTP (One Time Password), come ad esempio un SMS o il supporto offerto dall'applicazione mobile (per smartphone e tablet) (SPIDL2).
3. Richiede altri tipi di soluzioni di sicurezza, come un dispositivo fisico (ad esempio una smart card) rilasciato dall'identity provider (SPIDL3).

OpenID Connect in SPID

Le caratteristiche di OpenID Connect rispetto allo standard attualmente utilizzato da Spid (Saml - Security Assertion Markup Language) sono maggiore sicurezza; maggiore facilità di integrazione in sistemi eterogenei (single-page app, web, backend, mobile, IoT); migliore integrazione di componenti di terze parti in modalità sicura, interoperabile e scalabile.

Termini e definizioni

Essendo le funzionalità simili, ritroviamo gli stessi concetti di SAML 2.0 anche in OpenID Connect:

SAML 2.0	**OpenID Connect**
Assertion	*ID Token*
Attribute query	*UserInfo Endpoint*
Authentication request	*Authentication request*
ForceAuthn	*prompt=login*
Identity Provider (IdP)	*OpenID Provider (OP)*
IdP metadata	*OpenID Provider metadata*
Issuer	*Issuer*
Logout	*Revoke*
NameID policy	*Subject identifier type*
Passive Authentication	*prompt=none*
Service Provider (SP)	*Relying Party (RP)*
SP metadata	*Client metadata*
Subject	*Subject Identifier*
Attributes	*Claims*

OpenID Provider e Relyng Party

Per OpenID Provider (OP) e *Relying Party (RP)* si intendono rispettivamente i Gestori dell'identità digitale (Identity Provider - IdP) e i Fornitori di servizi (Service Provider - SP) di cui al DPCM 24 ottobre 2014, "Definizione delle caratteristiche del sistema pubblico per la gestione dell'identità digitale di cittadini e imprese (SPID), nonché dei tempi e delle modalità di adozione del sistema SPID da parte delle pubbliche amministrazioni e delle imprese."

Metadata

I metadata sono strutture dati contenenti le informazioni di OpenID Provider (OP) e di Relying Party (RP), mantenute e distribuite dal Registro SPID a tutti i soggetti della federazione, secondo le modalità definite dall'Agenzia per l'Italia Digitale, al fine di consentirne la configurazione nei rispettivi sistemi.

OpenID Provider (OP) Metadata Esempio


```
{
  "issuer": "https://op.fornitore_identita.it",
  "authorization_endpoint": "https://op.fornitore_identita.it/auth
  "token_endpoint": "https://op.fornitore_identita.it/token",
  "userinfo_endpoint": "https://op.fornitore_identita.it/userinfo"
  "introspection_endpoint": "https://op.fornitore_identita.it/intr
  "revocation_endpoint": "https://op.fornitore_identita.it/revoke"
  "end_session_endpoint": "https://op.fornitore_identita.it/logout
  "jwks_uri": "https://registry.spid.gov.it/...",
  "id_token_encryption_alg_values_supported": [
      "..."
  ],
  "userinfo_signing_alg_values_supported": [
      "..."
  ],
  "request_object_encryption_enc_values_supported": [
      "..."
  ],
  "token_endpoint_auth_methods_supported": ["private_key_jwt"],
  "userinfo_encryption_alg_values_supported": [
      "..."
  ],
  "claims_supported":[
    "https://attributes.spid.gov.it/spidCode",
    "https://attributes.spid.gov.it/name",
    "https://attributes.spid.gov.it/familyName",
    "https://attributes.spid.gov.it/placeOfBirth",
    "https://attributes.spid.gov.it/countyOfBirth",
    "https://attributes.spid.gov.it/dateOfBirth",
    "https://attributes.spid.gov.it/gender",
    "https://attributes.spid.gov.it/companyName",
    "https://attributes.spid.gov.it/registeredOffice",
    "https://attributes.spid.gov.it/fiscalNumber",
    "https://attributes.spid.gov.it/ivaCode",
    "https://attributes.spid.gov.it/idCard",
    "https://attributes.spid.gov.it/mobilePhone",
    "https://attributes.spid.gov.it/email",
    "https://attributes.spid.gov.it/address",
    "https://attributes.spid.gov.it/expirationDate",
    "https://attributes.spid.gov.it/digitalAddress"
  ],
  "acr_values_supported":[
    "https://www.spid.gov.it/SpidL1",
```

Elementi nei metadata OP

- **issuer**: identificativo per l'OP (con schema HTTPS), tipicamente il bae URL. Deve corrispondere al valore di iss nel token ID emesso dall'OP. Corrisponde all'attributo entityID in SAML e rappresenta la chiave unica per identificare l'IdP.
- **Authorization_endpoint**: URL per l'endpoint di autorizzazione, al quale il client verrà reindirizzato per avviare il flusso di utenticazione.
- **token_endpoint**: URL per l'endpoint del token che la RP utilizzer`a per scambiare il codice ricevuto alla fine del processo di autenticazione con un access token.
- **Userinfo_endpoint**: URL per lo user info endpoint che la RP pu`o invocare per ottenere gli attributi autorizzati dall'utente.
- **Introspection_endpoint**: URL per l'introspection endpoint che restituisce informazioni su un token.
- **Revocation_endpoint**: URL per l'endpoint di revoca che revoca un refresh token o access token precedentemente emesso per la RP richiedente.
- **Jwks_uri**: URL per il jwks che è un json contenente i seguenti parametri:
  - kty: famiglia dell'algoritmo crittografico adottato.
  - alg: algoritmo adottato
  - use: uso previsto per la chiave pubblica, signature (sig) o encryption (enc).
  - kid: identificatore univoco della chiave.
  - n: modulo (pem standard).
  - e: esponente (pem standard)
- ***Provider_name***: nome del provider OpenID.
- ***Provider_url***: URL del provider OpenID.
- altri campi contenenti gli algoritmi supportati.
- ***Acr_values supported***: array contenente i livelli SPID supportati dall'OP. Uno o pi`u tra:
  - *https://www.spid.gov.it/SpidL1*
  - *https://www.spid.gov.it/SpidL2*
  - *https://www.spid.gov.it/SpidL3*

Relying Party Metadata Esempio


```
{
  "client_id": "https://rp.spid.agid.gov.it",
  "redirect_uris": [
    "https://rp.spid.agid.gov.it/callback1/",
    "https://rp.spid.agid.gov.it/callback2/"
  ],
  "jwks_uri": "https://registry.spid.gov.it/...",
  "jwks": {
    "keys": [
     {
      "kty": "RSA",
      "alg": "RS256",
      "use": "sig",
      "kid": "e27671d73a2605ccd454413c4c94e25b3f66cdea",
      "n": "vmyoDT6ND_YJa1ItdvULuTJr2pw4MvN3Z5kmSiJBm9glVoakcDEBGF
      "e": "ABAB"
     }
    ]
  },
  "response_types": ["code"],
  "grant_types": ["authorization_code", "refresh_token"],
  "client_name": "Agenzia per l'Italia Digitale",
  "client_name#en": "Agency for Digital Italy"
}
```

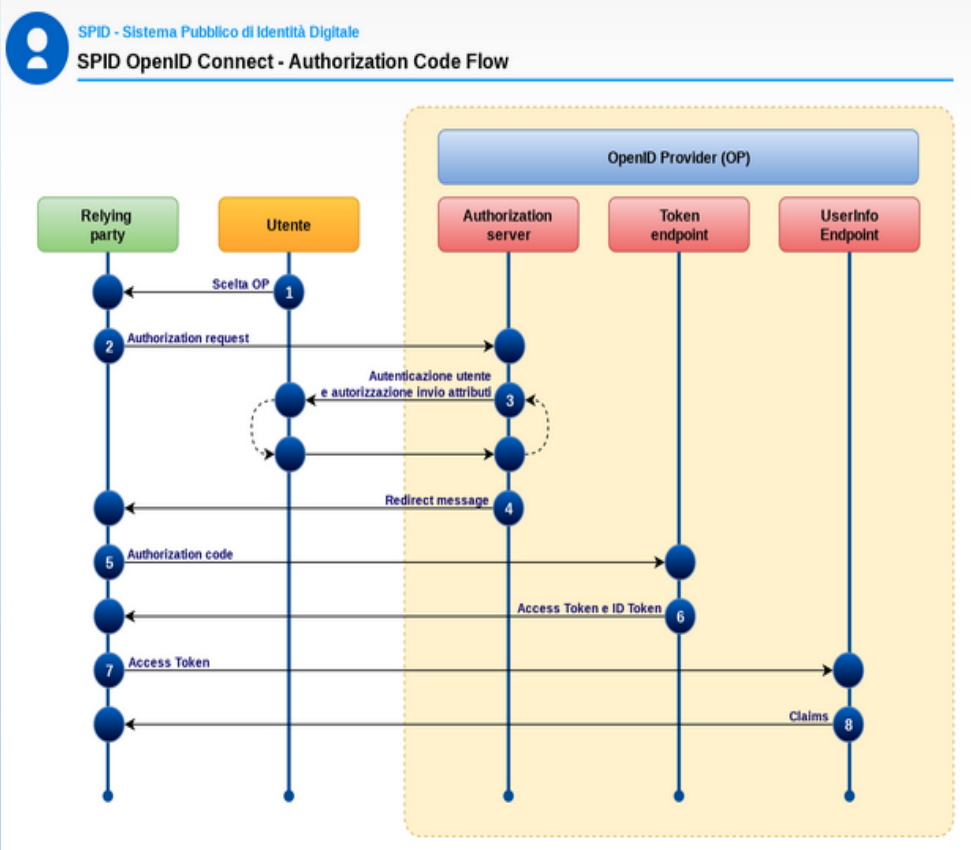
Elementi nei metadati dell'RP

- **client_id**: URI per identificare univocamente la RP, come da registro SPID.
- **redirect_uris**: elenco di URI di call-back utilizzati dalla RP. Il protocollo HTTPS è obbligatorio. Uno di essi deve essere presente nella richiesta di autenticazione.
- **jwks_uri**: stesso formato di quello presente nei metadati dell'OP.
- **client_name**: nome della RP da mostrare nelle pagine di autenticazione e consenso.
- **response_types**: deve contenere solo il valore code code
- **grant_types**: deve contenere solo i valori authorization code e refresh toke

Flusso

Il modello di flusso è l' «**OpenID Connect Authorization Code Flow**» che è infatti l'unico flusso previsto da iGov.

L'Authorization code flow restituisce un codice di autorizzazione che può essere scambiato per un ID token e/o un access token. Questo flusso è anche la soluzione ideale per sessioni lunghe o aggiornabili attraverso l'uso del refresh token. L'Authorization code flow ottiene l'authorization code dall'authorization endpoint dell'OpenID Provider e tutti i token sono restituiti dal token endpoint.



#	Da	A	Azione
1	**Utente**	**RP**	L'Utente, nella pagina di accesso del Relying Party (RP), seleziona, sul pulsante SPID, l'OpenID Provider (OP) con cui autenticarsi
2	**RP**	**OP Authorization server**	Il Relying Party (RP) prepara un'authenticati on request e la invia all'Authorizati on Endpoint dell'OpendID Provider selezionato dall'utente
3	**OP Authorization Server**	**Utente**	L'OpendID Provider (OP) richiede all'utente l'inserimento delle credenziali, secondo il livello SPID richiesto dal Relying Party (RP), all'utente a cui chiede, una volta autenticato, di autorizzare gli attributi richiesti dal Relying Party (RP)
4	**OP Authorization Server**	**RP**	L'OpenID Provider reindirizza l'utente verso il Redirect URI specificato dal RP, passando un authorization code
5	**RP**	**OP Token endpoint**	L'RP invia l'authorization code ricevuto al Token endpoint dell'OP
6	**OP Token endpoint**	**RP**	L'OP Token endpoint rilascia un ID Token, un Access token, e se richiesto un Refresh token
7	**RP**	**UserInfo endpoint**	L'RP valida l'ID token e registra nella propria sessione tutti i token ricevuti. Per chiedere gli attributi che erano stati autorizzati dall'utente al punto 3, invia l'Access token allo UserInfo endpoint dell'OP
8	**OP Userinfo endpoint**	**RP**	L'OP rilascia gli attributi richiesti

Authorization Endpoint (Authentication Request)

Per avviare il processo di autenticazione, il RP manda l'utente all'Authorization Endpoint dell'OP selezionato passando in POST o GET una richiesta in formato JWT.

Tale richiesta DEVE essere firmata e cifrata, secondo le modalità definite dall'Agenzia per l'Italia Digitale.

Esempio (chiamata HTTP):

https://op.spid.agid.gov.it/auth?
request=eyJhbGciOiJSUzI1NiIs
ImtpZCI6ImsyYmRjIn0.ew0KICJpc3MiOiAiczZCaGRSa3F0MyIsDQogI
mF1ZCI6ICJod
HRwczovL3NlcnZlci5leGFtcGxlLmNvbSIsDQogInJlc3BvbnNlX3R5cGUi
OiAiY29kZS
BpZF90b2tlbiIsDQogImNsaWVudF9pZCI6ICJzNkJoZFJrcXQzIiwNCiAic
mVkaXJlY3R
fdXJpIjogImh0dHBzOi8vY2xpZW50LmV4YW1wbGUub3JnL2NiIiwNCi
Aic2NvcGUiOiAi
b3BlbmlkIiwNCiAic3RhdGUiOiAiYWYwaWZqc2xka2oiLA0KICJub25jZ
SI6ICJuLTBTN
l9XekEyTWoiLA0KICJtYXhfYWdlIjogODY0MDAsDQogImNsYWltcyI6IA
0KICB7DQogIC
AidXNlcmluZm8iOiANCiAgICB7DQogICAgICJnaXZlbl9uYW1lIjogeyJlc
3NlbnRpYWw iOiB0cnVlfSwNCiAgICAgIm5p

Esempio (contenuto del JWT**):**

{
client_id=https%3A%2F%2Frp.spid.agid.gov.it
code_challenge=qWJlMe0xdbXrKxTm72EpH659bUxAxw80
code_challenge_method=S256
nonce=MBzGqyf9QytD28eupyWhSqMj78WNqpc2
prompt=login
redirect_uri=https%3A%2F%2Frp.spid.agid.gov.it%2Fcallback1%2F
response_type=code
scope=openid
acr_values=https://www.spid.gov.it/SpidL1 https://www.spid.gov.it/SpidL2
claims={
"id_token":{
"nbf": { essential: true},
"jti": { essential: true}
},
"userinfo":{
"https://attributes.spid.gov.it/name": null,
"https://attributes.spid.gov.it/familyName": null
},
}
state=fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd
}

Elementi Authentication Request

Parametro	Descrizione	Valori ammessi	Obbligatorio
client_id	URI che identifica univocamente il RP come da Registro SPID.	Deve corrispondere ad un valore nel Registro SPID.	SI
code_challenge	Un challenge per PKCE da riportare anche nella successiva richiesta al Token endpoint.	V. paragrafo 6.1 «Generazione del code_challenge per PKCE»	SI
code_challenge_method	Metodo di costru zione del challen ge PKCE.	È obbligatorio specificare il valore **S256**	SI
nonce	Valore che serve ad evitare attacchi Reply, generato casualmente e non prevedibile da terzi. Questo valore sarà restituito nell'ID Token fornito dal Token Endpoint, in modo da consentire al client di verificare che sia uguale a quello inviato nella richiesta di autenticazione.	Stringa di almeno 32 caratteri alfanumerici.	SI
prompt	Definisce se l'OP deve occuparsi di eseguire una richiesta di autenticazione all'utente o meno.	**consent**: l'OP chiederà le credenziali di autenticazione all'utente (ma solo se non è già attiva una sessione di Single Sign-On) e successivamente chiederà il consenso al trasferimento degli attributi (valore consigliato). **consent login:** l'OP chiederà sempre le credenziali di autenticazione all'utente e successivamente chiederà il consenso al trasferimento degli attributi (valore da utilizzarsi limitatamente ai casi in cui si vuole forzare la riautenticazion e)	SI

Proof Key for Code Exchange è un'estensione per OAuth 2.0 al fine di evitare possibili attacchi messi in atto intercettando il codice di autorizzazione.

Elementi Authentication Request

redirect_uri	URL dove l'OP reindirizzerà l'utente al termine del processo di autenticazione.	Deve essere uno degli URL indicati nel client metadata (v. paragrafo 3.2).	SI
response_type	Il tipo di credenziali che deve restituire l'OP.	**code**	SI
Scope	Lista degli scope richiesti.	**openid** (obbligatorio). **offline_access** se specificato, l'OP rilascerà oltre all'*access token* anche un *refresh token* necessario per instaurare sessioni lunghe revocabili. L'uso di questo valore è consentito solo se il client è un'applicazione per dispositivi mobili che intenda offrire all'utente una sessione lunga revocabile.	SI

Elementi Authentication Request

acr_values	Valori di riferimento della classe di contesto dell'autenticazione richiesta. Stringa separata da uno spazio, che specifica i valori "acr" richiesti al server di autorizzazione per l'elaborazione della richiesta di autenticazione, con i valori visualizzati in ordine di preferenza.	https://www.spid.gov.it/SpidL1 https://www.spid.gov.it/SpidL2 https://www.spid.gov.it/SpidL3	SI
Claims	Lista dei claims (attributi) che un RP intende richiedere per il servizio e livello SPID richiesto.	v. paragrafo 5.1	SI
State	Valore univoco utilizzato per mantenere lo stato tra la request e il callback. Questo valore verrà restituito al client nella risposta al termine dell'autenticaz ione. Il valore deve essere significativo esclusivamente per il RP e non deve essere intellegibile ad altri.	Stringa di almeno 32 caratteri alfanumerici.	SI
response_mode	Definisce la modalità di risposta del Form*	form_post	SI
ui_locales	Lingue preferibili per visualizzare le pagine dell'OP. L'OP può ignorare questo parametro se non dispone di nessuna delle lingue indicate.	Lista di codici RFC5646 separati da spazi.	NO

Claims

Il parametro claims definisce gli attributi e il livello SPID richiesti. all'interno dell'elemento «userinfo» si elencano gli attributi, da richiedere come chiavi di oggetti JSON, i cui valori devono essere null. Gli attributi elencati sotto «userinfo» sono disponibili al momento della chiamata allo UserInfo Endpoint.

```
{
 "userinfo":
  {
   "https://attributes.spid.gov.it/familyName":
    {
     "essential": true
    }
  },
}
```

Se il Relying Party è privato, gli OpenID Provider devono controllare che gli attributi richiesti rientrino tra quelli che essi, in base alla convenzione, possono utilizzare.

Generazione del code_challenge per PKCE

PKCE (Proof Key for Code Exchange, RFC7636) è un'estensione del protocollo OAuth 2.0 finalizzata ad evitare un potenziale attacco attuato con l'intercettazione dell'authorization code, soprattutto nel caso di applicazioni per dispositivi mobili. Consiste nella generazione di un codice (*code verifier*) e del suo hash (*code challenge*). Il *code challenge* viene inviato all'OP nella richiesta di autenticazione.

Quando il client contatta il Token Endpoint al termine del flusso di autenticazione, invia il *code verifier* originariamente creato, in modo che l'OP possa confrontare che il suo hash corrisponda con quello acquisito nella richiesta di autenticazione.

Il *code verifier* deve avere una lunghezza compresa tra 43 e 128 caratteri e deve essere generato con un algoritmo crittografico ad alta entropia.

Il *code challenge* deve essere generato con algoritmo SHA256.

Authentication response

Un'Authentication response è un messaggio di risposta di autorizzazione OAuth 2.0 restituito dall'authorization endpoint dell'OpenID Provider (OP) al termine del flusso di autenticazione. L'OP reindirizzerà l'utente al redirect_uri specificato nella richiesta di autorizzazione, aggiungendo nella post i parametri in risposta.

https://op.spid.agid.gov.it/resp?
code=usDwMnEzJPpG5oaV8x3j&
state=fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd

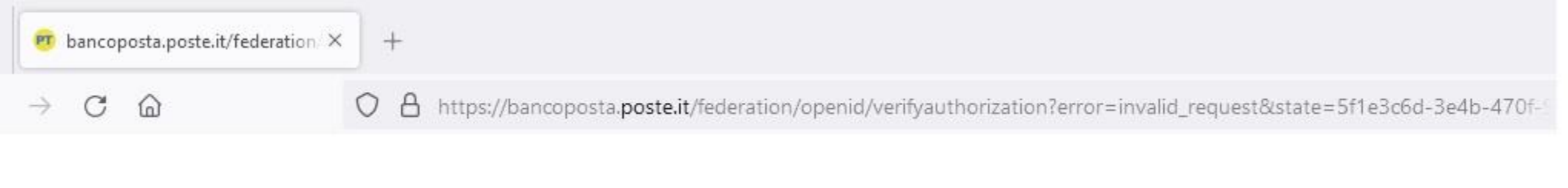
Parametro	Descrizione	Valori ammessi
code	Codice univoco di autorizzazione (*authorization code*) che il client poi passerà al Token Endpoint, secondo le modalità definite dall'Agenzia per l'Italia Digitale.	
state	Valore state incluso nell'Authentication request. Il client è tenuto a verificarne la corrispondenza.	Deve essere lo stesso valore indicato dal client nella Authorization Request.

Errori

In caso di errore, l'OP visualizza i messaggi definiti dalle Linee Guida UX SPID. Nei casi in cui tali linee guida prescrivono un redirect dell'utente verso il RP, l'OP effettua il redirect verso l'URL indicata nel parametro **redirect_uri** della richiesta (solo se valido, ovvero presente nel client metadata), con i seguenti parametri.

https://op.spid.agid.gov.it/resp?
error=invalid_request&
error_description=request%20malformata&
state=fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd

Parametro	Descrizione	Valori ammessi
error	Codice dell'errore	
error_description	Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging. Questo messaggio non è destinato ad essere visualizzato all'utente (a tal fine si faccia riferimento alle Linee Guida UX SPID).	
state	Valore *state* incluso nell'Authentication Request.	Il client è tenuto a verificare che corrisponda a quello inviato nella Authentication Request.



Codici Errori

Scenario	Codice errore
L'OP ha negato l'accesso a causa di credenziali non valide o non adeguate al livello SPID richiesto.	**access_denied**
Il client_id indicato nella richiesta non è riconosciuto.	**invalid_client**
La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri.	**invalid_request**
Sono stati richiesti degli scope non validi.	**invalid_scope**
L'OP ha riscontrato un problema interno.	**server_error**
L'OP ha riscontrato un problema interno temporaneo.	**temporarily_unavailable**

Token Endpoint (richiesta token)

Il Token Endpoint rilascia *access token*, *ID Token* e *refresh token*, vi sono due scenari distinti in cui il client chiama il Token Endpoint:

1. al termine del flusso di autenticazione descritto nel paragrafo precedente, il Client chiama il Token Endpoint inviando l'Authorization code ricevuto dall'OP (code=usDwMnEzJPpG5oaV8x3j) per ottenere un *ID Token* e un *access token* (necessario per poi chiedere gli attributi/claim allo UserInfo Endpoint) ed eventualmente un refresh token (se è stata avviata una sessione lunga revocabile);
2. in presenza di una sessione lunga revocabile, il Client chiama il Token Endpoint inviando il *refresh token* in suo possesso per ottenere un nuovo *access token*.

Request

Esempio di richiesta con authorization code (caso 1):

POST https://op.spid.agid.gov.it/token?
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWI
iOiIxMjM0NTY3ODkwIiwibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9
q7oiXcYVIIqGWY0wWQlqxvFGYswLF88&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-as sertion-type%3Ajwt-bearer&
code=usDwMnEzJPpG5oaV8x3j&
code_verifier=9g8S40MozM3NSqjHnhi7OnsE38jklFv2&
grant_type=authorization_code

Esempio di richiesta con refresh token (caso 2):

POST https://op.spid.agid.gov.it/token?
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWI
iOiIxMjM0NTY3ODkwIiwibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9
q7oiXcYVIIqGWY0wWQlqxvFGYswLF88&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-as sertion-type%3Ajwt-bearer&
grant_type=refresh_token&
refresh_token=8xLOxBtZp8

Elementi chiave nella richiesta del token

Parametro	Descrizione	Valori ammessi	Obbligatorio
client_id	URI che identifica univocamente il RP come da Registro SPID.		SI
client_assert ion	JWT firmato con la chiave privata del Relying Party contenente i seguenti parametri: **iss**: Identificatore del RP registrato presso gli OP e che contraddistingu e univocamente l'entità nella federazione nel formato Uniform Resource Locater (URL); corrisponde al client_id usato nella richiesta di autenticazione **sub**: uguale al parametro **iss** **aud**: URL del Token Endpoint dell'OP **iat**: data/ora in cui è stato rilasciato il JWT in formato UTC **exp**: data/ora di scadenza della request in formato UTC. **jti**: Identificatore univoco per questa richiesta di autenticazione, generato dal client casualmente con almeno 128bit di entropia.	**iat**: secondo le modalità definite dall'Agenzia per l'Italia Digitale. **exp**: secondo le modalità definite dall'Agenzia per l'Italia Digitale.	SI

Client_assert ion_type		Deve assumere il seguente valore: **urn:ietf:para ms:oauth:client - assertion-type :jwt-bearer**	SI
Code	Codice di autorizzazione restituito nell'Authentica tion response.		Solo se **grant_type** è **authorization _code**
code_verifier	Codice di verifica del code_challenge (v paragrafo 5.2)		Solo se **grant_type** è **authorization _code**
grant_type	Tipo di credenziale presentata dal Client per la richiesta corrente.	Può assumere uno dei seguenti valori: **authorization _code refresh _token**	SI
refresh_token			Solo se **grant_type** è **refresh_token**

Esempio di token response

Dopo avere ricevuto e validato la Token request dal client, il Token endpoint dell'OpenID Provider (OP) restituisce una response che include ID Token e Access Token e un eventuale Refresh Token, in formato JWT e firmati secondo le modalità definite dall'Agenzia per l'Italia Digitale.

Access Token e ID Token devono essere formati secondo le indicazioni dello standard "International Government Assurance Profile (iGov) for OAuth 2.0 - Draft 03, paragrafo 3.2.1, "JWT Bearer Tokens".


```
{
 "access_token": "dC34Pf6kdG...",
 "token_type": "Bearer",
 "refresh_token": "wJ848BcyLP...",
 "expires_in": 1800,
 "id_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY..."
}
```

Elementi chiave nella token response

Parametro	Descrizione	Valori ammessi
access_token	L'access token, in formato JWT firmato, consente l'accesso allo UserInfo endpoint per ottenere gli attributi.	
token_type	Tipo di *access token* restituito.	Deve essere valorizzato sempre con **Bearer**
refresh_token	Il *refresh token*, in formato JWT firmato, consente di chiamare nuovamente il Token Endpoint per ottenere un nuovo *access token* e quindi recuperare una sessione lunga revocabile.	
expires_in	Scadenza dell'*access token*, in secondi.	Secondo le modalità definite dall'Agenzia per l'Italia Digitale.
id_token	ID Token in formato JWT, firmato e cifrato.	

ID Token

L’ID Token è un JSON Web Token (JWT) che contiene informazioni sull’utente che ha eseguito l’autenticazione. I Client devono eseguire la validazione dell’ID Token.

Esempio


```
{
 "iss": "https://rp.spid.agid.gov.it/",
 "sub": "OP-1234567890",
 "aud": "https://op.spid.agid.gov.it/auth",
 "acr": "https://www.spid.gov.it/SpidL2",
 "at_hash": "qiyh4XPJGsOZ2MEAyLkfWqeQ",
 "iat": 1519032969,
 "nbf": 1519032969,
 "exp": 1519033149,
 "jti": "nw4J0zMwRk4kRbQ53G7z",
 "nonce": "MBzGqyf9QytD28eupyWhSqMj78WNqpc2"
}
```

Elementi chiave dell'ID token

Parametro	Descrizione	Validazione
Iss	Identificatore dell'OP che lo contraddistingue univocamente nella federazione nel formato Uniform Resource Locator (URL).	Il client è tenuto a verificare che questo valore corrisponda all'OP chiamato.
Sub	Per il valore di questo parametro fare riferimento allo standard "OpenID Connect Core 1.0", paragrafo 8.1. "Pairwise Identifier Algorithm".	
Aud	Contiene il client ID.	Il client è tenuto a verificare che questo valore corrisponda al proprio client ID.
Acr	Livello di autenticazione effettivo. Può essere uguale o superiore a quello richiesto dal client nella Authentication Request.	

at_hash	Hash dell'Access Token; il suo valore è la codifica base64url della prima metà dell'hash del valore access_token, usando l'algoritmo di hashing indicato in **alg** nell'header dell'ID Token.	Il client è tenuto a verificare che questo valore corrisponda all'*access token* restituito insieme all'ID Token.
Iat	Data/ora di emissione del token in formato UTC.	

Nbf	Data/ora di inizio validità del token in formato UTC. Deve corrispondere con il valore di **iat**.	{
		userinfo: {…}
		id_token: {
		acr: {…},
		nbf: { essential: true},
		jti: { essential: true }
		}
		}

Exp	Data/ora di scadenza del token in formato UTC, secondo le modalità definite dall'Agenzia per l'Italia Digitale.	
Jti	Identificatore unico dell'ID Token che il client più utilizzare per prevenirne il riuso, rifiutando l'ID Token se già processato. Deve essere di difficile individuazione da parte di un attaccante e composto da una stringa casuale.	
Nonce	Stringa casuale generata dal Client per ciascuna sessione utente ed inviata nell'Authentication Request (parametro nonce), finalizzata a mitigare attacchi replay.	Il client è tenuto a verificare che coincida con quella inviata nell'Authentication Request.

Errori

In caso di errore, l'OP restituisce un codice HTTP 401 con un JSON nel body avente gli elementi di seguito indicati.

Esempio:

```
{
  "error": "invalid_client",
  "error_description: "client_id non riconosciuto."
}
```

Parametro	Descrizione	Valori ammessi
Error	Codice dell'errore	
error_description	Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging. Questo messaggio non è destinato ad essere visualizzato all'utente (a tal fine si faccia riferimento alle Linee Guida UX SPID).	

Codici Errore

Scenario	Codice errore
Il client_id indicato nella richiesta non è riconosciuto.	**invalid_client**
Il parametro **grant_type** contiene un valore non corretto.	**unsupported_grant_type**
I parametri **grant_type**, **code**, **code_verifier**, **access_token** non sono validi.	**invalid_grant**
La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri.	**invalid_request**
L'OP ha riscontrato un problema interno.	**server_error**
L'OP ha riscontrato un problema interno temporaneo.	**temporarily_unavailable**

UserInfo Endpoint (attributi)

Lo UserInfo Endpoint è una risorsa protetta OAuth 2.0 che restituisce attributi dell'utente autenticato. Per ottenere gli attributi richiesti dal Relying Party, il client inoltra una richiesta allo UserInfo endpoint utilizzando l'Access token. Il risultato è presentato in JSON e contiene una raccolta di coppie nome e valore.

Lo UserInfo Endpoint deve supportare l'uso dei metodi HTTP GET e HTTP POST definiti in RFC 2616 [RFC2616], accettare i token di accesso come utilizzo di token bearer OAuth 2.0 [RFC6750] e supportare l'uso di Cross Origin Resource Sharing (CORS) e/o altri metodi appropriati per consentire ai client Java Script di accedere all'endpoint.

GET https://op.spid.agid.gov.it/userinfo
Authorization: Bearer dC34Pf6kd

Response

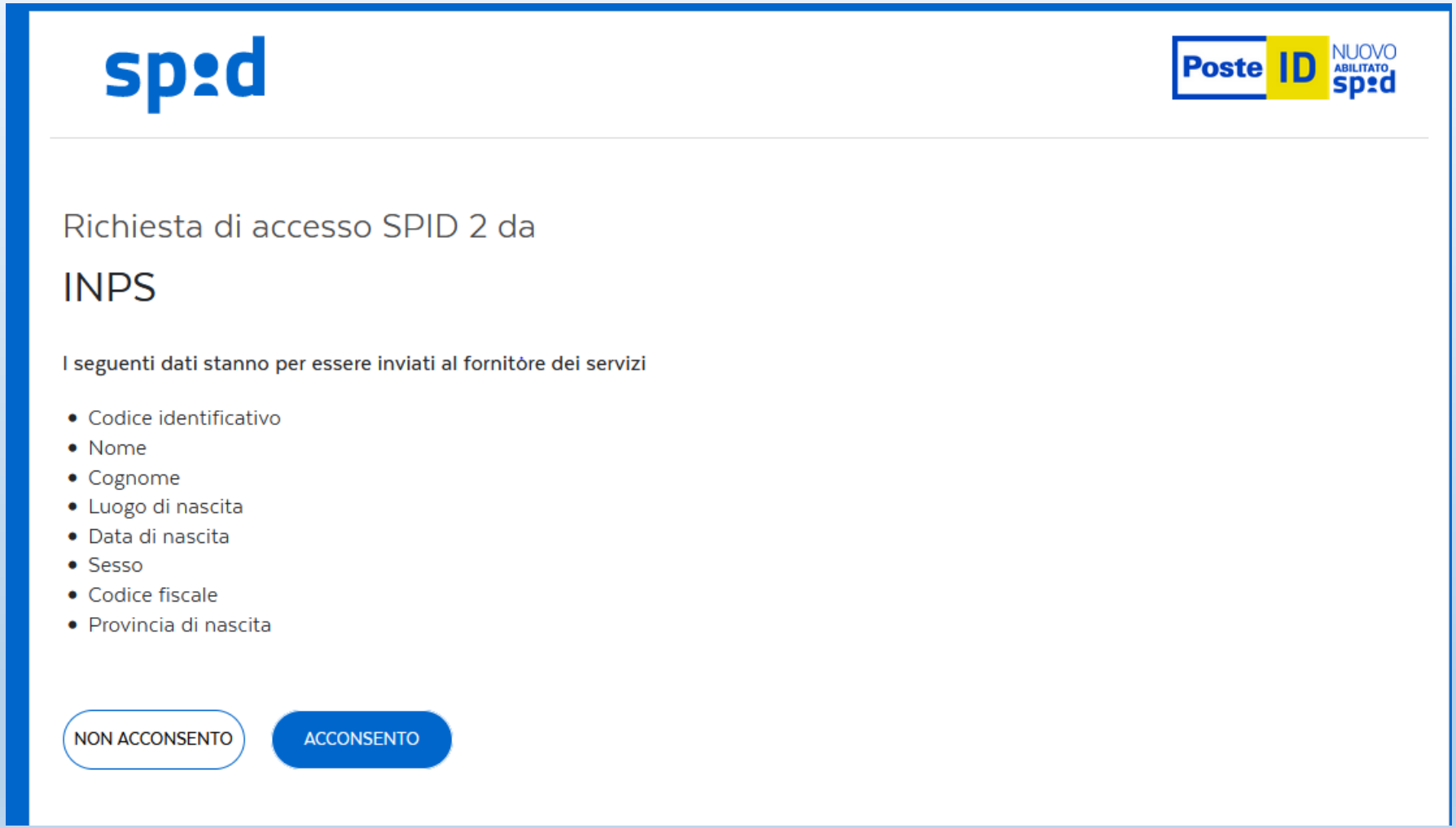
La response dello UserInfo Endpoint deve essere firmata e cifrata secondo le modalità definite dall'Agenzia per l'Italia Digitale. Lo UserInfo Endpoint restituisce i claim autorizzati nella Authentication Request.

Esempio:


```
{
  "iss": "https://op.fornitore_identita.it",
  "aud": "https://rp.fornitore_servizio.it",
  "iat": 1519032969,
  "nbf": 1519032969,
  "exp": 1519033149,
  "sub": "OP-1234567890",
  "https://attributes.spid.gov.it/name": "Mario",
  "https://attributes.spid.gov.it/familyName": "Rossi",
  "https://attributes.spid.gov.it/fiscalNumber": "MROXXXXXXXXXXXXX"
}
```


Parametro	Descrizione	Valori ammessi
sub	Identificatore del soggetto, coincidente con quello già rilasciato nell'ID Token.	Il RP deve verificare che il valore coincida con quello contenuto nell'ID Token.
aud	Identificatore del soggetto destinatario della response	
iss	URI che identifica univocamente il RP come da Registro SPID (client_id).	Il RP deve verificare che il valore coincida con il proprio client_id.
<attributo>	I claim richiesti al momento dell'autenticazione	

In caso di errore di autenticazione, lo UserInfo Endpoint restituisce un errore "HTTP 401".


spid
Poste ID NUOVO ABILITATO spid
Richiesta di accesso SPID 2 da
INPS
I seguenti dati stanno per essere inviati al fornitore dei servizi
Codice identificativo
Nome
Cognome
Luogo di nascita
Data di nascita
Sesso
Codice fiscale
Provincia di nascita
NON ACCONSENTO
ACCONSENTO

Introspection Endpoint (verifica validità token)

L'Introspection Endpoint esposto dall'OP consente ai RP di ottenere informazioni su un token in loro possesso, come ad esempio la sua validità.

La richiesta all'Introspection Endpoint consiste nell'invio del token su cui si vogliono ottenere informazioni unitamente ad una Client Assertion che consente di identificare il RP che esegue la richiesta.

Esempio:
POST https://op.spid.agid.gov.it/introspection?
client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWI
iOiIxMjM0NTY3ODkwIiwibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9 q7oiXcYVIIqGWY0wWQlqxvFGYswLF88&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-as sertion-type%3Ajwt-bearer&
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
token=eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTQsImF1ZCI6WyJl
NzFmYjcyYS05NzRmLTQwMDEtYmNiNy1lNjdjMmJjMDAzN2YiXSwiaXNzIjoiaHR0cHM6X
C9cL2FzLXZhLmV4YW1wbGUuY29tXC8iLCJqdGkiOiIyMWIxNTk2ZC04NWQzLTQzN2MtYW
Q4My1iM2YyY2UyNDcyNDQiLCJpYXQiOjE0MTg2OTg4MTR9.FXDtEzDLbTHzFNroW7w27R
Lk5m0wprFfFH7h4bdFw5fR3pwiqejKmdfAbJvN3_yfAokBv06we5RARJUbdjmFFfRRW23
cMbpGQCIk7Nq4L012X_1J4IewOQXXMLTyWQQ_BcBMjcW3MtPrY1AoOcfBOJPx1k2jwRkY tyVTLWlff6S5gK-
ciYf3b0bAdjoQEHd_IvssIPH3xuBJkmtkrTlfWR0Q0pdpeyVePkMSI 28XZvDaGnxA4j7QI5loZYeyzGR9h70xQLVzqwwl1P0-
F_0JaDFMJFO1yl4IexfpoZZsB3 HhF2vFdL6D_lLeHRy-H2g2OzF59eMIsM_Ccs4G47862w

Introspection Endpoint – Elementi chiave

Parametro	Descrizione	Valori ammessi
client_assertion	JWT firmato con la chiave privata del Relying Party contenente gli stessi parametri documentati per le richieste al Token Endpoint.	L'OP deve verificare la validità di tutti i campi presenti nel JWT, nonché la validità della sua firma in relazione al parametro **client_id**.
client_assertion_ty pe		**urn:ietf:params:oau th:client-assertion-t ype:jwt-bearer**
client_id	URI che identifica univocamente il RP come da Registro SPID.	L'OP deve verificare che il client_id sia noto.
token	Il token su cui il RP vuole ottenere informazioni.	

Introspection Endpoint - Response

L'Introspection Endpoint risponde con un oggetto JSON definito come segue.

Esempio:

```
{
  "active": true,
  "scope": "foo bar",
  "exp": 1519033149,
  "sub": "OP-1234567890",
  "client_id": "https://rp.agid.gov.it/"
}
```

Parametro	Descrizione	Valori ammessi
active	Valore booleano che indica la validità del token. Se il token è scaduto, è revocato o non è mai stato emesso per il client_id chiamante, l'Introspection Endpoint deve restituire **false**.	
scope	Lista degli scope richiesti al momento dell'Authorization Request.	
exp	Scadenza del token.	
sub	Identificatore del soggetto, coincidente con quello già rilasciato nell'ID Token.	Il RP deve verificare che il valore coincida con quello contenuto nell'ID Token.
client_id	URI che identifica univocamente il RP come da Registro SPID.	Il RP deve verificare che il valore coincida con il proprio client_id.

Introspection Endpoint - Errori

In caso di errore, l'OP restituisce un codice HTTP 401 con un JSON nel body avente gli elementi di seguito indicati.

Parametro	Descrizione
Error	Codice dell'errore (v. tabella sotto)
error_description	Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging. Questo messaggio non è destinato ad essere visualizzato all'utente (a tal fine si faccia riferimento alle Linee Guida UX SPID).

Esempio:

```
{
 "error": "invalid_client",
 "error_description: "client_id non riconosciuto."
}
```

Scenario	Codice errore
Il client_id indicato nella richiesta non è riconosciuto.	**invalid_client**
La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri.	**invalid_request**
L'OP ha riscontrato un problema interno.	**server_error**
L'OP ha riscontrato un problema interno temporaneo.	**temporarily_unavailable**

Revocation Endpoint (logout)

Il Revocation Endpoint consente al RP di chiedere la revoca di un *access token* o di un *refresh token* in suo possesso. Lo stato del token può essere verificato inviandolo al introspection endpoint

Quando l'utente esegue il logout, o quando la sua sessione presso il RP scade (in base alle policy decise da quest'ultimo), il RP deve chiamare questo endpoint per revocare l'*access token* e l'eventuale *refresh token* in suo possesso.

L'OP dovrà revocare il token specificato nella richiesta e dovrà terminare la sessione di Single Sign-On se ancora attiva. Eventuali altri token attivi per l'utente dovranno invece essere mantenuti validi.

```
https://idp-poste.poste.it/jod-idp-retail/federation/openid-logout?
iss=agent_idp_ppay&sid=_iTn4JADNOdMwe4BttSoWBedusPvCBQnz3w7yceKlpAMMzyHoUwCPzAGt4tghXcKuJaXLIyb6vXywRdLulgVd5B8HVyvnSAH9bABOcKstIM&sigalg=ht
tp%3A%2F%2Fwww.w3.org%2F2001%2F04%2Fxmldsig-more%23rsa-sha256&sig=GBRR7DUPVlEr9LOhNQWg66oIuCz
%2Firo4Bed6frrwfnk30Xa4ckpKojdtdtZTwdt4Thnx3uiWy%2Bir%0A3vP4PJquAPRpibFa6eUEv5yLEnDq3vyVIuTj%2BJyogCTYcn8MBQGeAiv9Bi8k0pbX4Pa0urYTnzRT
%0Ak39lRY%2B6fdv4NsJSuXo92Lr4o5bcIjxmd1Mee5OQ08BFbOMwL%2BeFfvoCUtFuDKB5bN7VwUJ65J2z%0A5MpS%2BiCqz0iB40hYMDsJ4BSdjbViu6D
%2BL3%2FTo2GvSNHpjs23vgigFGIN1lKrQ%2BbWbHxeOf4Pvigg%0A0Jltrzn7Drp9CYaCQGRjHHr%2BgVBE9fwiyBUFeA%3D%3D
```

Revocation Endpoint - Request

La richiesta al Revocation Endpoint consiste nell'invio del token che si vuole revocare unitamente ad una Client Assertion che consente di identificare il RP che esegue la richiesta.

Esempio:
POST https://op.spid.agid.gov.it/revoke?
client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9q7 oiXcYVIIqGWY0wWQlqxvFGYswLF88&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-asser tion-type%3Ajwt-bearer&
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
token=eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTQsImF1ZCI6WyJl
NzFmYjcyYS05NzRmLTQwMDEtYmNiNy1lNjdjMmJjMDAzN2YiXSwiaXNzIjoiaHR0cHM6
XC9cL2FzLXZhLmV4YW1wbGUuY29tXC8iLCJqdGkiOiIyMWIxNTk2ZC04NWQzLTQzN2Mt
YWQ4My1iM2YyY2UyNDcyNDQiLCJpYXQiOjE0MTg2OTg4MTR9.FXDtEzDLbTHzFNroW7w
27RLk5m0wprFfFH7h4bdFw5fR3pwiqejKmdfAbJvN3_yfAokBv06we5
RARJUbdjmFFfR
RW23cMbpGQCIk7Nq4L012X_1J4IewOQXXMLTyWQQ_BcBMjcW3Mt
PrY1AoOcfBOJPx1k2 jwRkYtyVTLWlff6S5gK-
ciYf3b0bAdjoQEHd_IvssIPH3xuBJkmtkrTlfWR0Q0pdpeyV
ePkMSI28XZvDaGnxA4j7QI5loZYeyzGR9h70xQLVzqwwl1P0-
F_0JaDFMJFO1yl4Iexf poZZsB3HhF2vFdL6D_lLeHRy-
H2g2OzF59eMlsM_Ccs4G47862w

Parametro	Descrizione	Valori ammessi
client_assertion	JWT firmato con la chiave privata del Relying Party contenente gli stessi parametri documentati per le richieste al Token Endpoint.	L'OP deve verificare la validità di tutti i campi presenti nel JWT, nonché la validità della sua firma in relazione al parametro **client_id**.
client_assertion_ty pe		**urn:ietf:params:oauth:client-assertion-type:jwt-bearer**
client_id	URI che identifica univocamente il RP come da Registro SPID.	L'OP deve verificare che il client_id sia noto.
Token	Il token su cui il RP vuole ottenere informazioni.	

Revocation Endpoint - Response

Il Revocation Endpoint risponde con un codice HTTP 200, anche nel caso in cui il token indicato non esista o sia già stato revocato (in modo da non rilasciare informazioni).

Sessioni lunghe revocabili

Per applicazioni mobili in cui l'RP intenda offrire un'esperienza utente che non passi per il reinserimento delle credenziali SPID ad ogni avvio, è possibile beneficiare di sessioni lunghe revocabili. Per adottare questo tipo di sessione, la RP deve impostare nella authentication request lo scope *offline access* per ottenere successivamente anche un refresh token dopo il consenso esplicito dell'utente

Sessioni lunghe revocabili - Ambiti e limiti di utilizzo

1. Al primo avvio dell'applicazione l'Utente deve essere informato della possibilità di utilizzare la sessione lunga revocabile, per mantenere un'autenticazione di SPID di livello 1 che consenta all'applicazione di ricevere notifiche o effettuare azioni richieste dalla RP, anche quando l'Utente "non sia presente";
2. Le applicazioni mobili che fanno uso di sessioni lunghe revocabili sono tenute a richiedere all'utente, ad ogni avvio o attivazione, un PIN locale oppure un fattore biometrico.
3. In fase di installazione o di prima configurazione, l'applicazione chiede all'utente di registrare il fattore di autenticazione da utilizzare per ogni avvio successivo al primo.
4. Quando l'Utente avvia nuovamente l'applicazione, questa deve richiedere all'Utente il fattore di autenticazione scelto in fase di installazione o di prima configurazione e consentire l'accesso alle funzioni del RP fruibili con il Livello 1 di SPID.
5. Nel caso in cui sia necessario accedere all'applicazione con un livello superiore a SPID di Livello 1, occorre effettuare una nuova autenticazione SPID in base al livello richiesto.

Infine, l'OP deve includere una pagina raggiungibile dall'utente che mostri loro le attuali longterm session attive con la possibilit`a di revocarle. In caso di modifica della password, l'OP deve inoltre fornire la possibilit`a di revocare tutte le attuali long-term session attive dell'utente

Sessioni lunghe revocabili - Request

Per poter utilizzare le sessioni lunghe revocabili, l'RP include nella Authentication Request:

- lo scope "offline_access", al fine di ottenere un refresh token utilizzabile dietro espressa consenso dell'utente;
- il parametro "acr_values" contenente una delle seguenti opzioni:
  - il livello SPID 1;
  - il livello SPID 2 + il livello SPID 1.
  - il livello SPID 3 + il livello SPID 1.

Gestione delle sessioni

- Al fine di poter gestire le sessioni lunghe revocabili e poter rilasciare un refresh token per il Livello 1 di SPID anche a seguito di un'autenticazione di Livello 2 o 3 di SPID, è ammessa l'instaurazione, per ogni livello di SPID, di una sessione di autenticazione associata ad un determinato utente titolare di identità digitale, mantenuta dal gestore dell'identità digitale.
- Gli OP devono includere all'interno della "Pagina di gestione dell'identità SPID", descritta nelle Linee Guida UX SPID, un'interfaccia per visualizzare le sessioni lunghe revocabili attive, dove l'utente possa revocarle singolarmente o in massa.
- In caso di modifica della password richiesta dall'utente, l'OP deve prevedere la possibilità di revocare tutte le sessioni lunghe attive.

Gestione dei log

OpenID Provider e Relying party devono conservare i log di ogni autenticazione e devono essere mantenuti per un tempo pari a 24 mesi.

In particolare devono essere conservate le evidenze di:

rilascio di ID e access token a fronte di autenticazione;

rilascio di refresh token a fronte di autenticazione;

rilascio di ID e access token a fronte di utilizzo del refresh token.

Per ogni rilascio devono essere conservati JWT costituenti richiesta e risposta, occorre, inoltre, tracciare le chiamate e le relative risposte effettuate verso ogni endpoint.

Le tracciature devono essere mantenute nel rispetto del codice della privacy sotto la responsabilità dell'OpenID Provider o del Relying Party e l'accesso ai dati di tracciatura deve essere riservato a personale incaricato.

Al fine di garantire la confidenzialità potrebbero essere adottati meccanismi di cifratura dei dati o impiegati sistemi di basi di dati (DBMS) che realizzano la persistenza cifrata delle informazioni.

Per il mantenimento devono essere messi in atto meccanismi che garantiscono l'integrità e il non ripudio.