# SPID/CIE OIDC Regole tecniche

***Release version: latest***

**italia**

**07 feb 2023**

# Indice


**1 Indice dei contenuti** **3**

1.1 Riferimenti . . . 3
1.2 Normativa Nazionale ed Europea . . . 5
1.3 Termini e Acronimi . . . 6
1.4 Le Federazioni eID Italiane . . . 8
1.5 Entity Configuration . . . 10
1.6 Entity Statement . . . 12
1.7 Trust Mark . . . 17
1.8 Soggetti Aggregatori . . . 20
1.9 Acquisire i Metadata . . . 21
1.10 Endpoint di Federazione . . . 23
1.11 Gestione degli errori di federazione . . . 24
1.12 Metadata . . . 24
1.13 Flusso di autenticazione . . . 31
1.14 Authorization endpoint (Authentication) . . . 32
1.15 Token Endpoint . . . 38
1.16 UserInfo Endpoint . . . 46
1.17 Tabella attributi utente . . . 48
1.18 Introspection Endpoint (verifica validità token) . . . 50
1.19 Revocation Endpoint . . . 52
1.20 Logout . . . 53
1.21 Algoritmi crittografici . . . 54
1.22 Retention Policy . . . 55
1.23 Differenze tra SPID e CIE id . . . 56
1.24 Differenze con OIDC iGov . . . 58
1.25 Differenze con OIDC Federation . . . 59
1.26 Considerazioni di Sicurezza . . . 59
1.27 Buone Pratiche . . . 60
1.28 Esempi . . . 61
1.29 Diventa fornitore di servizi . . . 76
1.30 Come contribuire . . . 76

**Indice** **77**

SPID[3] e CIE id[4] sono i Sistemi Pubblici di Identità Digitale Italiani e adottano gli standard OpenID Connect Core[5], International Government Assurance Profile (iGov) for OpenID Connect 1.0[6] e OpenID Connect Federation 1.0[7].

Grazie all'identità digitale[8], la Pubblica Amministrazione e i fornitori di servizi privati forniscono la chiave per accedere ai servizi online attraverso una credenziale unica.

Questa documentazione contiene le specifiche tecniche consolidate, conformi alle Linee Guida Nazionali, per migliorare l'esperienza di integrazione alle Federazioni OIDC SPID e CIE id per i Fornitori di Servizio pubblici e privati (RP), Identity Providers (OP) e Soggetti Aggregatori (SA).

In questa documentazione trovi:

- Gli esempi pratici dei Metadata, delle richieste e delle risposte OpenID Connect.
- Come effettuare la registrazione automatica dei RP presso gli OpenID Provider.
- Come un OpenID Provider riconosce e registra dinamicamente un RP.
- Come utilizzare gli endpoint della API della Federazione.
- Come autenticare un utente a SPID e CIE ed ottenere i suoi attributi.

[3] https://www.spid.gov.it/
[4] https://www.cartaidentita.interno.gov.it/
[5] https://openid.net/specs/openid-connect-core-1_0.html
[6] https://openid.net/specs/openid-igov-openid-connect-1_0-03.html
[7] https://openid.net/specs/openid-connect-federation-1_0.html
[8] https://identitadigitale.gov.it/

CAPITOLO 1

Indice dei contenuti

## 1.1 Riferimenti

### 1.1.1 Standards

| | |
|---|---|
| OIDC-FED[9] | OpenID Connect Federation 1.0 |
| iGov.OIDC[10] | Varley, M., Grassi, P. "iGov Profile for OpenID Connect", October 2018. |
| OpenID.Core[11] | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B. and C. Mortimore, "OpenID Connect Core 1.0", August 2015. |
| OpenID.Registration[12] | Sakimura, N., Bradley, J., and M. Jones, "OpenID Connect Dynamic Client Registration 1.0," November 2014. |
| OpenID.Discovery[13] | Sakimura, N., Bradley, J., Jones, M., and E. Jay, "OpenID Connect Discovery 1.0," November 2014. |
| **RFC 2119**[14] | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997. |
| **RFC 2616**[15] | Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, June 1999. |
| **RFC 3339**[16] | Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002. |
| **RFC 3986**[17] | Uniform Resource Identifier (URI): Generic Syntax |
| **RFC 7009**[18] | Lodderstedt, T., Dronia, S., Scurtescu, M., "OAuth 2.0 Token Revocation," RFC7009, August 2013. |
| **RFC 7159**[19] | Bray, T., "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 7159, March 2014. |
| **RFC 7515**[20] | Jones, M., Bradley, J. and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015. |
| **RFC 7516**[21] | Jones, M., Hildebrand, J., "JSON Web Encryption (JWE)", May 2015. |
| **RFC 7517**[22] | Jones, M., "JSON Web Key (JWK)", RFC 7517, DOI 10.17487/RFC7517, May 2015. |
| **RFC 7518**[23] | Jones, M., "JSON Web Algorithms (JWA)", May 2015. |

Continua alla pagina successiva

Tabella 1.1 – continua dalla pagina precedente

| | |
|---|---|
| **RFC 7519**[24] | Jones, M., Bradley, J. and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015. |
| **RFC 7523**[25] | Jones, M., Campbell, B., Martimore, C., "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants", May 2015. |
| **RFC 7636**[26] | Sakimura, N., Bradley, J. and N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, DOI 10.17487/RFC7636, September 2015. |
| **RFC 7638**[27] | Jones, M., Sakimura, N., "JSON Web Key (JWK) Thumbprint,"RFC7638, September 2015. |
| **RFC 7662**[28] | Richer, J., "OAuth 2.0 Token Introspection", RFC 7662, DOI 10.17487/RFC7662, October 2015. |
| **RFC 7591**[29] | Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, "OAuth 2.0 Dynamic Client Registration Protocol", RFC 7591, DOI 10.17487/RFC7591, July 2015. |
| **RFC 7800**[30] | Jones, M., Bradley, J. and H. Tschofenig, "Proof-of-Possession Key Semantics for JSON Web Tokens (JWTs)", RFC 7800, DOI 10.17487/RFC7800, April 2016. |
| **RFC 8174**[31] | Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, DOI 10.17487/RFC8174, May 2017. |
| **RFC 8414**[32] | Jones, M., Sakimura, N., and J. Bradley, "OAuth 2.0 Authorization Server Metadata", RFC 8414, DOI 10.17487/RFC8414, June 2018. |
| **RFC 8725**[33] | Jones, M., D. Hardt, Sheffer, Y., "JSON Web Token Best Current Practices", February 2020. |
| **RFC 9068**[34] | Bertocci, V., "JSON Web Token (JWT) Profile for OAuth 2.0 Access Token," RFC9068, October 2021. |
| I-D.ietf-oauth-iss-auth-resp[35] | Selhausen, K. M. Z. and D. Fett, "OAuth 2.0 Authorization Server Issuer Identification", Work in Progress, Internet-Draft, Draft-5, January 2022. |
| I-D.ietf-OAuth-Security-BCP[36] | Lodderstedt, T., Bradley, J., Labunets, A., Fett, D., "OAuth 2.0 Security Best Current Practice", Draft-19, December 2021. |
| EN319-412-1[37] | Electronic Signatures and Infrastructures (ESI); Certificate Profiles; |
| E164[38] | International Telecommunication Union, "E.164: The international public telecommunication numbering plan," 2010. |
| ISO8601-2004[39] | International Organization for Standardization, "ISO 8601:2004. Data elements and interchange formats - Information interchange - Representation of dates and times," 2004. |
| ICAO-Doc9303[40] | INTERNATIONAL CIVIL AVIATION ORGANIZATION, "Machine Readable Travel Documents, Seventh Edition, 2015, Part 3: Specifications Common to all MRTDs", 2015 |
| ISO3166[41] | ISO, "ISO 3166-1:1997. Codes for the representation of names of countries and their subdivisions |

## 1.2 Normativa Nazionale ed Europea

| | |
|---|---|
| CAD[42] | DL 7 March 2005 n.82: "Codice dell'amministrazione digitale." (GU Serie Generale n.112 16-05-2005 - Suppl. Ordinario n. 93) |
| DL-SEMPLIFICAZIONI[43] | DL 16 July 2020 n.76: "Misure urgenti per la semplificazione e l'innovazione digitale." (20A04921) (GU Serie Generale n.228 14-09-2020 - Suppl. Ordinario n. 33) and its conversion into Law, with amendments, Law 11 September 2020 n. 120. |
| EIDAS[44] | Regulation (Eu) No 910/2014 of the European Parliament and of the Council 23 July 2014 "on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC." |

### 1.2.1 Riferimenti normativi SPID

L'avvio del **Sistema SPID**, per sua natura e complessità, può richiedere di intervenire su diversi aspetti con specificazioni, chiarimenti, note informative e casi esemplificativi, al fine di dare supporto ad una migliore applicazione e comprensione dei Regolamenti SPID già emanati dall'AgID in conformità con quanto prescritto dall'art.4 del DPCM 24 ottobre 2014.

Al fine di raccogliere organicamente tali interventi e attribuirvi un carattere cogente che ne comporti l'obbligo di applicazione da parte degli attori convolti nel Sistema SPID, siano essi pubblici che privati, è stata creata la presente sezione **"Avvisi SPID"** con l'obiettivo di assicurare un'uniforme interpretazione delle regole, degli aspetti tecnici e di quant'altro necessario per il corretto funzionamento del Sistema nel suo complesso.

[9] https://openid.net/specs/openid-connect-federation-1_0.html
[10] https://openid.net/specs/openid-igov-openid-connect-1_0-03.html
[11] https://openid.net/specs/openid-connect-core-1_0-27.html
[12] https://openid.net/specs/openid-connect-registration-1_0.html
[13] https://openid.net/specs/openid-connect-discovery-1_0.html
[14] https://tools.ietf.org/html/rfc2119.html
[15] https://tools.ietf.org/html/rfc2616.html
[16] https://tools.ietf.org/html/rfc3339.html
[17] https://tools.ietf.org/html/rfc3986.html
[18] https://tools.ietf.org/html/rfc7009.html
[19] https://tools.ietf.org/html/rfc7159.html
[20] https://tools.ietf.org/html/rfc7515.html
[21] https://tools.ietf.org/html/rfc7516.html
[22] https://tools.ietf.org/html/rfc7517.html
[23] https://tools.ietf.org/html/rfc7518.html
[24] https://tools.ietf.org/html/rfc7519.html
[25] https://tools.ietf.org/html/rfc7523.html
[26] https://tools.ietf.org/html/rfc7636.html
[27] https://tools.ietf.org/html/rfc7638.html
[28] https://tools.ietf.org/html/rfc7662.html
[29] https://tools.ietf.org/html/rfc7591.html
[30] https://tools.ietf.org/html/rfc7800.html
[31] https://tools.ietf.org/html/rfc8174.html
[32] https://tools.ietf.org/html/rfc8414.html
[33] https://tools.ietf.org/html/rfc8725.html
[34] https://tools.ietf.org/html/rfc9068.html
[35] https://www.ietf.org/archive/id/draft-ietf-oauth-iss-auth-resp-00.html
[36] https://www.ietf.org/archive/id/draft-ietf-oauth-security-topics-19.html
[37] https://www.etsi.org/deliver/etsi_en/319400_319499/31941202/02.02.01_60/en_31941202v020201p.pdf
[38] http://www.itu.int/rec/T-REC-E.164-201011-I/en
[39] https://www.iso.org/standard/40874.html
[40] https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf
[41] https://www.iso.org/iso-3166-country-codes.html
[42] https://www.gazzettaufficiale.it/eli/gu/2005/05/16/112/so/93/sg/pdf
[43] https://www.gazzettaufficiale.it/eli/id/2020/09/14/20G00139/sg
[44] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG

Le presenti regole tecniche implementano i seguenti avvisi SPID:

| Avviso | Riferimento | Data |
|---|---|---|
| LL.GG. OpenID Connect in SPID | LL.GG. OpenID Connect in SPID[45] | 24/11/2021 |
| Avviso n.41 | Avviso n.41 - Integrazione LL.GG. OpenID Connect in SPID.pdf[46] | 06/05/2022 |
| Tabella Attributi utente v1.3 | Tabella Attributi in SPID - Integrazione LL.GG. OpenID Connect in SPID.pdf[47] | 24/06/2022 |
| Determina SPID OpenID Connect Federation | Regole tecniche per il funzionamento della Federazione SPID OpenID Connect - Integrazione LL.GG. OpenID Connect in SPID.pdf -[48] | 14/09/2022 |
| Linee Guida Attribute Authority SPID | Linee guida recanti le regole tecniche dei gestori di attributi qualificati[49] | 18/07/2022 |

### 1.2.2 Riferimenti normativi CIE id

| | |
|---|---|
| DM-CIE[50] | DM 23 December 2015 n.210: "Modalità tecniche di emissione della Carta d'identità elettronica." (15A09809) (GU Serie Generale n.302 30-12-2015) |

## 1.3 Termini e Acronimi

### 1.3.1 Termini

Seguono i termini utilizzati da OIDC-FED#Section_1.2[51] e in questo documento.

[45] https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf
[46] https://www.agid.gov.it/sites/default/files/repository_files/spid-avviso-n41-integrazione_ll.gg_._openid_connect_in_spid.pdf
[47] https://www.agid.gov.it/sites/default/files/repository_files/tabella_attributi_v.1.3.pdf
[48] https://www.agid.gov.it/sites/default/files/repository_files/regolamento-spid_openid_connect_federation_1.0.pdf
[49] https://www.agid.gov.it/sites/default/files/repository_files/llgg_attribute_authorities_0.pdf
[50] https://www.gazzettaufficiale.it/eli/id/2015/12/30/15A09809/sg
[51] https://openid.net/specs/openid-connect-federation-1_0.html#Section-1.2

| | |
|---|---|
| **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**[52] 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**[53], 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**[54] 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. |

### 1.3.2 Acronimi

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

[52] https://tools.ietf.org/html/rfc7515.html
[53] https://tools.ietf.org/html/rfc7515.html
[54] https://tools.ietf.org/html/rfc7515.html

| | |
|---|---|
| **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[55]. |
| **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**[56] 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). |

### 1.3.3 Convenzioni e Termini normativi

Le parole chiave "DEVE" e "DEVONO", "NON DEVE" e "NON DEVONO", "RICHIEDE" e "RICHIESTO", "NON DEVE", "DOVREBBE", "NON DOVREBBE", "RACCOMANDATO", "PUÒ" e "OPZIONALE" nel presente documento devono essere interpretate come descritte nel BCP 14 **RFC 2119**[57] **RFC 8174**[58] quando e solo quando appaiono in maiuscolo.

Le notazioni [...] e ... indicano che il testo è stato troncato per esigenze editoriali.

*base64url* denota la codifica URL-safe base64 senza padding definita in **RFC 7515#section-2**[59].

Tutti gli esempi contenuti in questo documento sono da considerarsi come non normativi.

> **Avvertimento:** Tutti gli esempi contenuti in questa documentazione sono da intendersi come non normativi

## 1.4 Le Federazioni eID 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 ammini-

[55] https://openid.net/specs/openid-connect-federation-1_0.html
[56] https://tools.ietf.org/html/rfc7519.html
[57] https://tools.ietf.org/html/rfc2119.html
[58] https://tools.ietf.org/html/rfc8174.html
[59] https://tools.ietf.org/html/rfc7515.html#section-2

strativo, mediante la stipula di convenzioni e accreditamento presso una o più autorità super partes, che tecnologico, mediante l'adozione di standard di interoperabilità sicuri che consentono l'interscambio dei dati.

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 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[60] e alla sezione *Differenze con OIDC Federation 1.0* (pagina 59).

### 1.4.1 OpenID Connect Federation

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.*

### 1.4.2 Configurazione della Federazione

La configurazione della Federazione è pubblicata dal Trust Anchor all'interno della sua *Entity Configuration* (pagina 12), 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_marks_issuers**).

Si veda qui un esempio non normativo di *Entity Configuration response Trust Anchor* (pagina 66)

Si veda la Sezione dedicata alle *Entity Configuration* (pagina 10) per ulteriori dettagli.

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

### 1.4.3 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* (pagina 61).

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.

## 1.5 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**.

### 1.5.1 Firma della Entity Configuration

Tutte le operazioni di verifica della firma relative agli ES, EC e TM sono eseguite con le chiavi pubbliche di Federazione. Per quanto riguarda gli algoritmi di firma supportati si veda la Sezione *Algoritmi Crittografici* (pagina 54).

> **Avvertimento:** 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.

### 1.5.2 Entity Configuration - claim comuni

| Claim | Descrizione | Supportato da |
|---|---|---|
| **iss** | String. Identificativo dell'entità che lo emette. | |
| **sub** | String. Identificativo del soggetto a cui è riferito. | |
| **iat** | UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[61] | |
| **exp** | UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[62]. | |
| **jwks** | Un JSON Web Key Set (JWKS) **RFC 7517**[63] 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). | |
| **metadata** | JSON Object. Ogni chiave dell'oggetto JSON rappresenta un identificatore del tipo di *Metadata* (pagina 24) e ogni valore DEVE essere un oggetto JSON che rappresenta i Metadata secondo lo schema di Metadata di quel tipo.<br>Una configurazione di entità PUÒ contenere più dichiarazioni di Metadata, ma solo una per ogni tipo di Metadata (**<entity_type>**).<br>I tipi consentiti sono i seguenti:<br>• openid_relying_party<br>• openid_provider<br>• federation_entity<br>• oauth_authorization_server<br>• oauth_resource | |

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

### 1.5.3 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. | |
| **trust_marks** | Un array JSON contenente i Trust Mark. Vedere la Sezione *Trust Mark* (pagina 17). Obbligatorio per tutti i partecipanti fatta esclusione del Trust Anchor. | |

**Vedi anche:**

- *Non-normative example of EC of an OP* (pagina 62)
- *Non-normative example of EC of a RP* (pagina 61)
- *Non-normative example of EC of a Federation Intermediary (SA)* (pagina 65)

[61] https://tools.ietf.org/html/rfc7519.html
[62] https://tools.ietf.org/html/rfc7519.html
[63] https://tools.ietf.org/html/rfc7517.html

### 1.5.4 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 riconoscobili come suoi discendenti. | |
| **trust_marks_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. | |

**Vedi anche:**

- *Esempio di EC di un TA* (pagina 66)

## 1.6 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* (pagina 23). L'entità superiore PUÒ definire le policy sui metadata per un soggetto discendente e pubblicare i TM da lei emessi per questo.

### 1.6.1 Firma di Entity Statement

Si applicano le medesime considerazioni fatte per gli **EC** e riportate nella sezione *Firma della Entity Configuration* (pagina 10).

### 1.6.2 Entity Statement

Gli ES emessi dal TA o da un suo Intermediario per i propri diretti discendenti, DEVONO contenere i seguenti attributi:

| Claim | Descrizione | Supportato da |
|---|---|---|
| **iss** | Si rimanda alla specifica OIDC-FED[64] Sezione 3.1 per i dettagli. | |
| **sub** | Si rimanda alla specifica OIDC-FED[65] Sezione 3.1 per i dettagli. | |
| **iat** | Si rimanda alla specifica OIDC-FED[66] Sezione 3.1 per i dettagli. | |
| **exp** | Si rimanda alla specifica OIDC-FED[67] Sezione 3.1 per i dettagli. | |
| **jwks** | JWKS di Federazione dell'entità *sub*. Si rimanda alla specifica OIDC-FED[68] Sezione 3.1 per i dettagli. | |
| **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[69] Section 5.1 per i dettagli implementativi. | |
| **trust_marks** | JSON Array contenente i Trust Mark emessi da se stesso per il soggetto discendente. | |
| **constraints** | PUÒ contenere il claim **allowed_leaf_entity_types** per restringere i tipi di Entità riconoscibili per il suo discendente (esempio: solo RP). | |

**Vedi anche:**

- OIDC-FED#Section_3.1[70]
- *Esempio non normativo di Entity Statement* (pagina 69)

### 1.6.3 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.

[64] https://openid.net/specs/openid-connect-federation-1_0.html
[65] https://openid.net/specs/openid-connect-federation-1_0.html
[66] https://openid.net/specs/openid-connect-federation-1_0.html
[67] https://openid.net/specs/openid-connect-federation-1_0.html
[68] https://openid.net/specs/openid-connect-federation-1_0.html
[69] https://openid.net/specs/openid-connect-federation-1_0.html
[70] https://openid.net/specs/openid-connect-federation-1_0.html#Section-3.1

| Claim | Operazioni / Valori | Supportato da |
|---|---|---|
| **jwks** | Operazioni: *subset_of* Valori: DEVE contenere i JWKS del RP relativi alle operazioni di Core | |
| **grant_types** | Operazioni: *subset_of* Valori: DEVE essere *authorization_code* e *refresh_token* | |
| **id_token_signed_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **id_token_encrypted_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **id_token_encrypted_response_enc** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_signed_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_encrypted_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_encrypted_response_enc** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **token_endpoint_auth_method** | Operazioni: *one_of* Valori: DEVE essere *private_key_jwt* | |
| **client_registration_types** | Operazioni: *one_of* Valori: DEVE essere *automatic* | |

### Metadata Policy di un TA per un SA

Di seguito vengono riportati i claim che DEVONO essere 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* Valori: DEVE essere *authorization_code* e *refresh_token* | |
| **id_token_signed_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **id_token_encrypted_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **id_token_encrypted_response_enc** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_signed_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_encrypted_response_alg** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_encrypted_response_enc** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **token_endpoint_auth_method** | Operazioni: *one_of* Valori: DEVE essere *private_key_jwt* | |
| **client_registration_types** | Operazioni: *one_of* Valori: DEVE essere *automatic* | |

### Metadata Policy di un SA per 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: *subset_of* Valori: DEVE contenere i JWKS del RP relativi alle operazioni di Core | |

### Metadata Policy di un TA per un OP

Di seguito vengono riportati i claim che DEVONO essere 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: *subset_of* Valori: DEVE contenere i JWKS del OP relativi alle operazioni di Core | |
| **revocation_endpoint_auth_methods_supported** | Operazioni: *one_of* Valori: DEVE essere *private_key_jwt* | |
| **code_challenge_methods_supported** | Operazioni: *subset_of* Valori: DEVE essere *S256* | |
| **scopes_supported** | Operazioni: *subset_of* Valori: DEVE contenere *openid*, *offline_access*. Per CIE id PUÒ contenere anche *profile*, *email*. | |
| **response_types_supported** | Operazioni: *one_of* Valori: DEVE essere *code*. | |
| **response_modes_supported** | Operazioni: *subset_of* Valori: DEVE essere *form_post*, *query*. | |
| **grant_types_supported** | Operazioni: *subset_of* Valori: DEVE essere *refresh_token*, *authorization_code*. | |
| **acr_values_supported** | Operazioni: *subset_of* Valori: DEVE essere *https://www.spid.gov.it/SpidL1*, *https://www.spid.gov.it/SpidL2*, *https://www.spid.gov.it/SpidL3*. | |
| **subject_types_supported** | Operazioni: *one_of* Valori: DEVE essere *pairwise*. | |
| **id_token_signing_alg_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **id_token_encryption_alg_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **id_token_encryption_enc_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_signing_alg_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_encryption_alg_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **userinfo_encryption_enc_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **token_endpoint_auth_methods_supported** | Operazioni: *one_of* Valori: DEVE essere *private_key_jwt* | |
| **token_endpoint_auth_signing_alg_values_supported** | Operazioni: *subset_of* Valori: DEVE contenere gli algoritmi definiti nella Sezione *Algoritmi Crittografici* (pagina 54) | |
| **claims_parameter_supported** | Operazioni: *one_of* Valori: DEVE essere *true* | |
| **request_parameter_supported** | Operazioni: *one_of* Valori: DEVE | |

**Vedi anche:**

- *Esempi non normativi di Metadata Policy* (pagina 72)

## 1.7 Trust Mark

I **Trust Mark (TM)**, letteralmente tradotti come *Marchi di Fiducia*, sono JWT firmati **RFC 7515**[71] 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[72]. 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.

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

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

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_marks_issuers**, presente all'interno del proprio Entity Configuration. Il valore dell'attributo **trust_marks_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.

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/**

[71] https://tools.ietf.org/html/rfc7515.html
[72] https://openid.net/specs/openid-connect-federation-1_0.html

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

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 |
| **intermediary** | 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 |

### 1.7.1 federation_entity Trust Mark

In aggiunta ai claim dei profili **public** e **private**, il profilo **intermediary** 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

**Vedi anche:**

Si veda Sezione *Soggetti aggregatori nel contesto Federativo* (pagina 20)

### 1.7.2 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. |

### 1.7.3 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* (pagina 23) 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**.

**Vedi anche:**

- OIDC-FED[73] Sezione .5.3.2.

### 1.7.4 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.

---

### 1.7.5 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[74]). Segue la lista.

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **iss** | String. URL che identifica univocamente l'Autorità che lo ha emesso. | |
| **sub** | String. URL che identifica univocamente il soggetto per il quale il Trust Mark è stato emesso. | |
| **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/` | |
| **iat** | UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[75] | |
| **logo_uri** | String. Un URL che punta al logo rappresentante il Trust Mark. | |
| **exp** | UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[76] | |
| **ref** | String. URL che punta a informazioni presenti sul web relative a questo Trust Mark. | |
| **organization_type** | String. Specifica se l'ente appartiene alla pubblica amministrazione italiana o al settore privato (**public** o **private**) | |
| **id_code** | Oggetto JSON. Contiene uno o più codici di identificazione dell'organizzazione. I claim disponibili sono: - **ipa_code**: OBBLIGATORIO nel caso di organizzazione pubblica. - **aoo_code**: OPZIONALE. - **uo_code**: OPZIONALE. - **vat_number**: OBBLIGATORIO per organizzazione privata se non presente *fiscal_number*. - **fiscal_number**: OBBLIGATORIO per organizzazione privata se non presente *vat_number*. | |
| **email** | String. Email istituzionale o PEC dell'organizzazione. | |
| **organization_name** | String. Il nome completo dell'entità che fornisce i servizi | |
| **sa_profile** | String. RICHIESTO per SA. Specifica il profilo dell'Aggregatore, **full** o **light**. | |

**Avvertimento:** 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.

**Vedi anche:**

- OIDC-FED[77] Sezione 5.3.1.
- Esempi non normativi: *Trust Mark issued by TA to a RP* (pagina 67), *Trust Mark issued by TA to a SA* (pagina 68), *Trust Mark issued by SA to a RP* (pagina 68),

## 1.8 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**).

[75] https://tools.ietf.org/html/rfc7519.html
[76] https://tools.ietf.org/html/rfc7519.html
[77] https://openid.net/specs/openid-connect-federation-1_0.html

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* (pagina 19)), 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 |

## 1.9 Acquisire i Metadata

In questa sezione sono illustrate le 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.

### 1.9.1 Relying Party

Il RP ottiene la lista degli OP in formato JSON interrogando l'*endpoint list* (pagina 23) disponibile presso il *Trust Anchor* (pagina 70). Per ogni soggetto contenuto nella *risposta* (pagina 70) dell'endpoint list e corrispondente ad un OP, il RP *richiede* (pagina 69) 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.*

### 1.9.2 OpenID Provider

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

*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* (pagina 61). Ottiene l'Entity Configuration del RP e convalida la firma dei Trust Mark riconoscibili all'interno della Federazione[1].

Se il RP non espone all'interno della sua configurazione nessun Trust Mark riconoscibile per il profilo di RP (vedi Sezione *Trust Mark* (pagina 17)) il Provider DEVE rifiutare l'autorizzazione con un messaggio di errore come definito nella Sezione *Gestione degli errori di Federazione* (pagina 24).

Se il Provider convalida con successo almeno un Trust Mark per il profilo RP contenuto all'interno della configurazione del RP richiedente, estrae le entità superiori contenute nel claim **authority_hints** ed avvia la fase di Federation Entity Discovery. Ne consegue il calcolo della **Trust Chain** e l'ottenimento del Metadata finale.

Durante il Federation Entity Discovery, il Provider richiede ad una o più entità superiori[2] l'Entity Statement relativo al RP e ottiene la chiave pubblica con la quale valida la configurazione del RP, fino a giungere al Trust Anchor. Infine applica la politica dei Metadata pubblicata dal Trust Anchor e salva il risultante Metadata finale del RP associandolo ad una data di scadenza, oltre la quale rinnoverà il Metadata secondo le modalità di rinnovo della Trust Chain.

Ottenuto il Metadata finale, il Provider valida la richiesta del RP secondo le modalità definite in questo documento.

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.*

### 1.9.3 Accesso alla Entity Configuration

In questa sezione viene descritto come individuare per un determinato soggetto l'URL **RFC 3986**[78] per il download della Entity Configuration.

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

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

[2] 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.

[78] https://tools.ietf.org/html/rfc3986.html

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.

*The Federation Entity Discovery process to build a Trust Chain and obtain the final Metadata.*

## 1.10 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[79] 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[80] 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 vedi OIDC-FED[81] 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 vedi OIDC-FED[82] Section 7.4)
- **entity listing endpoint**: fornisce la lista delle entità discendenti registrate presso il TA o un SA (per maggiori dettagli vedi OIDC-FED[83] 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.

[79] https://openid.net/specs/openid-connect-federation-1_0.html
[80] https://openid.net/specs/openid-connect-federation-1_0.html
[81] https://openid.net/specs/openid-connect-federation-1_0.html
[82] https://openid.net/specs/openid-connect-federation-1_0.html
[83] https://openid.net/specs/openid-connect-federation-1_0.html

## 1.11 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* (pagina 24) | |
| **Descrizione dell'errore** | Descrizione più dettagliata dell'errore, finalizzata ad aiutare lo sviluppatore per eventuale debugging. | |

### 1.11.1 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* | |
| *invalid_client* | Il Client non è autorizzato perchè la validazione della Trust Chain fallisce. | *302 Found* | |
| *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* | |
| *invalid_request* | La richiesta non è completa o non è conforme a quanto definito dalle presenti specifiche tecniche. | *400 Bad Request* | |
| *not_found* | La risorsa richiesta non è stata trovata. | *404 Not Found* | |

## 1.12 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[84]) e OpenID Connect Dynamic Client Registration 1.0 (OpenID.Registration[85]) 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.

### 1.12.1 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 nel seguente esempio:

[84] https://openid.net/specs/openid-connect-discovery-1_0.html
[85] https://openid.net/specs/openid-connect-registration-1_0.html

```
{
   "metadata":{
     "federation_entity":{
       ...
     }
     "openid_provider":{
       ...
     }
   }
}
```

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **organization_name** | Vedi Sezione 4.8 di OIDC-FED[86] | |
| **homepage_uri** | Vedi Sezione 4.8 di OIDC-FED[87] | |
| **policy_uri** | Vedi Sezione 4.8 di OIDC-FED[88] | |
| **logo_uri** | URL del logo dell'entità; DEVE essere in formato SVG. Vedi Sezione 4.8 di OIDC-FED[89] | |
| **contacts** | PEC istituzionale dell'ente. Vedi Sezione 4.8 di OIDC-FED[90] | |
| **federation_resolve_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[91] Section 4.6 | |

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

[86] https://openid.net/specs/openid-connect-federation-1_0.html
[87] https://openid.net/specs/openid-connect-federation-1_0.html
[88] https://openid.net/specs/openid-connect-federation-1_0.html
[89] https://openid.net/specs/openid-connect-federation-1_0.html
[90] https://openid.net/specs/openid-connect-federation-1_0.html
[91] https://openid.net/specs/openid-connect-federation-1_0.html

| Claim | Descrizione | Supportato da |
|---|---|---|
| **issuer** | Vedi OpenID.Discovery#OP_Metadata[92]. DEVE essere valorizzato con un HTTPS URL che identifica univocamente l'OP. | |
| **authorization_endpoint** | Vedi OpenID.Discovery#OP_Metadata[93]. | |
| **token_endpoint** | Vedi OpenID.Discovery#OP_Metadata[94]. | |
| **userinfo_endpoint** | Vedi OpenID.Discovery#OP_Metadata[95]. | |
| **introspection_endpoint** | Vedi **RFC 8414#page-4**[96]. | |
| **revocation_endpoint** | Vedi **RFC 8414#page-4**[97]. | |
| **revocation_endpoint_auth_methods_supported** | Vedi **RFC 8414#page-4**[98]. Il valore supportato è **private_key_jwt** | |
| **code_challenge_methods_supported** | Vedi **RFC 8414#page-4**[99]. L'OP DEVE supportare S256 (vedi **RFC 7636#section-4.3**[100]). | |
| **scopes_supported** | Vedi OpenID.Discovery#OP_Metadata[101]. I valori supportati sono *openid* e *offline_access*. CIE id supporta anche *profile*, *email*. Per maggiori dettagli vedi *Sezione User Claims* (pagina 50). | |
| **response_types_supported** | Vedi OpenID.Discovery#OP_Metadata[102]. Il valore supportato è **code**. | |
| **response_modes_supported** | Vedi OpenID.Discovery#OP_Metadata[103]. I valori supportati sono *form_post* e *query*. | |
| **grant_types_supported** | Vedi OpenID.Discovery#OP_Metadata[104]. I valori supportati sono *refresh_token* e *authorization_code*. | |
| **acr_values_supported** | Vedi OpenID.Discovery#OP_Metadata[105]. I valori supportati sono:<br>`https://www.spid.gov.it/SpidL1`<br>`https://www.spid.gov.it/SpidL2`<br>`https://www.spid.gov.it/SpidL3` | |
| **subject_types_supported** | Vedi OpenID.Discovery#OP_Metadata[106]. Il valore supportato è **pairwise**. | |
| **id_token_signing_alg_values_supported** | Vedi OpenID.Discovery#OP_Metadata[107]. Vedi signature *Algoritmi crittografici* (pagina 54). | |
| **id_token_encryption_alg_values_supported** | Vedi OpenID.Discovery#OP_Metadata[108]. Vedi key encryption *Algoritmi crittografici* (pagina 54). | |
| **id_token_encryption_enc_values_supported** | Vedi OpenID.Discovery#OP_Metadata[109]. Vedi content encryption *Algoritmi crittografici* (pagina 54). | |
| **userinfo_signing_alg_values_supported** | Vedi OpenID.Discovery#OP_Metadata[110]. Vedi signature *Algoritmi crittografici* (pagina 54). | |
| **userinfo_encryption_alg_values_supported** | Vedi OpenID.Discovery#OP_Metadata[111]. Vedi key encryption *Algoritmi crittografici* (pagina 54). | |
| **userinfo_encryption_enc_values_supported** | Vedi OpenID.Discovery#OP_Metadata[112]. Vedi content encryption *Algoritmi crittografici* (pagina 54). | |
| **request_object_signing_alg_values_supported** | Vedi OpenID.Discovery#OP_Metadata[113]. Vedi signature *Algoritmi crittografici* (pagina 54). | |

**Avvertimento:** Il Metadata **"openid_provider"** DEVE adottare il parametro **jwks** o **signed_jwks_uri** come normato da OID-FED invece del parametro **jwks_uri** come richiesto in OpenID.Discovery#OP_Metadata[114].

**Vedi anche:**

- *Esempio di EC di un OP* (pagina 62)

### 1.12.2 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 nel seguente esempio:

```
{
   "metadata":{
     "federation_entity":{
        ...
     }
     "openid_relying_party":{
        ...
     }
   }
}
```

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **organization_name** | Vedi Sezione 4.8 di OIDC-FED[115] | |
| **homepage_uri** | Vedi Sezione 4.8 di OIDC-FED[116] | |
| **policy_uri** | Vedi Sezione 4.8 di OIDC-FED[117] | |
| **logo_uri** | (RACCOMANDATO) URL del logo dell'entità; DEVE essere in formato SVG. Vedi Sezione 4.8 di OIDC-FED[118] | |
| **contacts** | PEC istituzionale dell'ente. Vedi Sezione 4.8 di OIDC-FED[119] | |
| **federation_resolve_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[120] Section 4.6 | |

[92] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[93] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[94] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[95] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[96] https://tools.ietf.org/html/rfc8414.html#page-4
[97] https://tools.ietf.org/html/rfc8414.html#page-4
[98] https://tools.ietf.org/html/rfc8414.html#page-4
[99] https://tools.ietf.org/html/rfc8414.html#page-4
[100] https://tools.ietf.org/html/rfc7636.html#section-4.3
[101] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[102] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[103] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[104] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[105] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[106] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[107] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[108] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[109] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[110] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[111] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[112] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[113] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata
[114] https://openid.net/specs/openid-connect-discovery-1_0.html#ProviderMetadata

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **redirect_uris** | Vedi OpenID.Registration#ClientMetadata[121]. È obbligatorio l'uso dello schema HTTPS nel caso di client web-based. | |
| **grant_types** | Vedi OpenID.Registration#ClientMetadata[122]. I valori ammissibili **authorization_code** e **refresh_token**. | |
| **jwks** | Vedi OpenID.Registration#ClientMetadata[123] e JWK[124]. | |
| **id_token_signed_response_alg** | Vedi OpenID.Registration#ClientMetadata[125]. Vedi signature *Algoritmi crittografici* (pagina 54). | |
| **id_token_encrypted_response_alg** | Vedi OpenID.Registration#ClientMetadata[126]. Vedi key encryption *Algoritmi crittografici* (pagina 54). | |
| **id_token_encrypted_response_enc** | Vedi OpenID.Registration#ClientMetadata[127]. Obbligatorio solo nel caso sia presente anche il parametro *id_token_encrypted_response_alg*. Vedi content encryption *Algoritmi crittografici* (pagina 54). | |
| **userinfo_signed_response_alg** | Vedi OpenID.Registration#ClientMetadata[128]. Vedi signature *Algoritmi crittografici* (pagina 54). | |
| **userinfo_encrypted_response_alg** | Vedi OpenID.Registration#ClientMetadata[129]. Vedi key encryption *Algoritmi crittografici* (pagina 54). | |
| **userinfo_encrypted_response_enc** | Vedi OpenID.Registration#ClientMetadata[130]. Vedi content encryption *Algoritmi crittografici* (pagina 54). | |
| **token_endpoint_auth_method** | Vedi OpenID.Registration#ClientMetadata[131]. Il valore richiesto è **private_key_jwt**. | |
| **client_id** | Vedi OpenID.Registration[132]. DEVE essere valorizzato con un HTTPS URL che identifica univocamente il RP. | |
| **client_registration_types** | Vedi OIDC-FED[133] Section 4.1. Il valore richiesto è **automatic**. | |
| **response_types** | Array dei valori di response_type previsti da OAuth 2.0 che il RP userà nelle richieste di autenticazione. Deve contenere il valore **code**. | |

**Nota:** Gli URI presenti nel parametro **redirect_uris** POSSONO anche usare eventuali schemi custom (ad es. myapp://) al fine di supportare applicazioni mobili.

[115] https://openid.net/specs/openid-connect-federation-1_0.html
[116] https://openid.net/specs/openid-connect-federation-1_0.html
[117] https://openid.net/specs/openid-connect-federation-1_0.html
[118] https://openid.net/specs/openid-connect-federation-1_0.html
[119] https://openid.net/specs/openid-connect-federation-1_0.html
[120] https://openid.net/specs/openid-connect-federation-1_0.html
[121] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[122] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[123] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[124] https://datatracker.ietf.org/doc/html/draft-ietf-jose-json-web-key
[125] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[126] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[127] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[128] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[129] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[130] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[131] https://openid.net/specs/openid-connect-registration-1_0.html#ClientMetadata
[132] https://openid.net/specs/openid-connect-registration-1_0.html
[133] https://openid.net/specs/openid-connect-federation-1_0.html

### 1.12.3 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 nel seguente esempio:

```
{
   "metadata":{
     "federation_entity":{
       ...
     }
   }
}
```

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **organization_name** | Vedi Sezione 4.8 di OIDC-FED[134] | |
| **homepage_uri** | Vedi Sezione 4.8 di OIDC-FED[135] | |
| **policy_uri** | Vedi Sezione 4.8 di OIDC-FED[136] | |
| **logo_uri** | URL del logo dell'entità; DEVE essere in formato SVG. Vedi Sezione 4.8 di OIDC-FED[137] | |
| **contacts** | PEC istituzionale dell'ente. Vedi Sezione 4.8 di OIDC-FED[138] | |
| **federation_fetch_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[139] Section 4.8 | |
| **federation_list_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[140] Section 4.8 | |
| **federation_trust_mark_status** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[141] Section 4.8 | |
| **federation_resolve_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[142] Section 4.8 | |

**Vedi anche:**

- Esempio di EC di un *OP* (pagina 66) e di un SA *SA* (pagina 65)

### 1.12.4 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*.

```
{
   "metadata":{
     "federation_entity":{
       ...
```

(continues on next page)

[134] https://openid.net/specs/openid-connect-federation-1_0.html
[135] https://openid.net/specs/openid-connect-federation-1_0.html
[136] https://openid.net/specs/openid-connect-federation-1_0.html
[137] https://openid.net/specs/openid-connect-federation-1_0.html
[138] https://openid.net/specs/openid-connect-federation-1_0.html
[139] https://openid.net/specs/openid-connect-federation-1_0.html
[140] https://openid.net/specs/openid-connect-federation-1_0.html
[141] https://openid.net/specs/openid-connect-federation-1_0.html
[142] https://openid.net/specs/openid-connect-federation-1_0.html

(continua dalla pagina precedente)

```
    },
    "oauth_authorization_server":{
      ...
    },
    "oauth_resource":{
      ...
    }
  }
}
```

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **organization_name** | Vedi Sezione 4.8 di OIDC-FED[143] | |
| **homepage_uri** | Vedi Sezione 4.8 di OIDC-FED[144] | |
| **policy_uri** | Vedi Sezione 4.8 di OIDC-FED[145] | |
| **logo_uri** | URL del logo dell'entità; DEVE essere in formato SVG. Vedi Sezione 4.8 di OIDC-FED[146] | |
| **contacts** | PEC istituzionale dell'ente. Vedi Sezione 4.8 di OIDC-FED[147] | |
| **federation_trust_mark_status_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[148] Section 4.8 | |
| **federation_resolve_endpoint** | Vedi Sezione *Endpoint di Federazione* (pagina 23) e OIDC-FED[149] Section 4.8 | |

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **issuer** | Vedi **RFC 8414#page-4**[150]. DEVE essere valorizzato con un HTTPS URL che identifica univocamente l'AA. | |
| **authorization_endpoint** | Solo per Attribute Authority **private** flow. Vedi *LG-AA* and **RFC 8414#page-4**[151]. | |
| **token_endpoint** | Vedi **RFC 8414#page-4**[152]. | |
| **jwks** | Vedi JWK[153]. | |
| **scopes_supported** | Vedi **RFC 8414#page-4**[154]. | |
| **response_types_supported** | Vedi **RFC 8414#page-4**[155]. | |
| **grant_types_supported** | Vedi **RFC 8414#page-4**[156] e **RFC 8623**[157]. | |
| **token_endpoint_auth_methods_supported** | Vedi **RFC 8414#page-4**[158]. Il valore supportato è **private_key_jwt**. | |
| **token_endpoint_auth_signing_alg_values_supported** | Vedi **RFC 8414#page-4**[159]. Vedi signature *Algoritmi crittografici* (pagina 54). | |
| **op_policy_uri** | Vedi **RFC 8414#page-4**[160]. | |
| **op_tos_uri** | Vedi **RFC 8414#page-6**[161]. | |
| **dpop_signing_alg_values_supported** | Vedi OpenID DPoP[162]. Vedi signature *Algoritmi crittografici* (pagina 54). | |

[143] https://openid.net/specs/openid-connect-federation-1_0.html
[144] https://openid.net/specs/openid-connect-federation-1_0.html
[145] https://openid.net/specs/openid-connect-federation-1_0.html
[146] https://openid.net/specs/openid-connect-federation-1_0.html
[147] https://openid.net/specs/openid-connect-federation-1_0.html
[148] https://openid.net/specs/openid-connect-federation-1_0.html
[149] https://openid.net/specs/openid-connect-federation-1_0.html

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **resource** | Vedi OAuth-RS[163]. Una o più HTTPS URL che identificano gli endpoint delle risorse protette. | |

## 1.13 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**[164]). 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**.

Segue la descrizione dei passaggi, come da numerazione indicata in figura.

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.

[150] https://tools.ietf.org/html/rfc8414.html#page-4
[151] https://tools.ietf.org/html/rfc8414.html#page-4
[152] https://tools.ietf.org/html/rfc8414.html#page-4
[153] https://datatracker.ietf.org/doc/html/draft-ietf-jose-json-web-key
[154] https://tools.ietf.org/html/rfc8414.html#page-4
[155] https://tools.ietf.org/html/rfc8414.html#page-4
[156] https://tools.ietf.org/html/rfc8414.html#page-4
[157] https://tools.ietf.org/html/rfc8623.html
[158] https://tools.ietf.org/html/rfc8414.html#page-4
[159] https://tools.ietf.org/html/rfc8414.html#page-4
[160] https://tools.ietf.org/html/rfc8414.html#page-4
[161] https://tools.ietf.org/html/rfc8414.html#page-6
[162] https://datatracker.ietf.org/doc/html/draft-ietf-oauth-dpop-10
[163] https://datatracker.ietf.org/doc/html/draft-jones-oauth-resource-metadata
[164] https://tools.ietf.org/html/rfc7636.html

## 1.14 Authorization endpoint (Authentication)

### 1.14.1 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[165].

> **Avvertimento:** 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.

**Vedi anche:**

- *Esempio di Authorization Request* (pagina 72)

Di seguito 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* (pagina 24). DEVE essere presente almeno il valore *openid*. | |
| **code_challenge** | Vedi **RFC 7636#section-4.2**[166]. | |
| **code_challenge_method** | Come definito dal parametro **code_challenge_methods_supported** nel *Metadata OP* (pagina 24). | |
| **request** | Vedi OpenID.Core#JWTRequests[167]. DEVE essere un **JWT** firmato. | |

Di seguito una tabella che riporta la composizione dell'header del **JWT**.

| Jose Header | Descrizione | Supportato da |
|---|---|---|
| **alg** | Vedi **RFC 7516#section-4.1.1**[168]. Vedi *Algoritmi crittografici* (pagina 54).. | |
| **kid** | Vedi **RFC 7638#section_3**[169]. | |

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

[165] https://openid.net/specs/openid-connect-core-1_0.html#Serializations
[166] https://tools.ietf.org/html/rfc7636.html#section-4.2
[167] https://openid.net/specs/openid-connect-core-1_0.html#JWTRequests
[168] https://tools.ietf.org/html/rfc7516.html#section-4.1.1
[169] https://tools.ietf.org/html/rfc7638.html#section_3

Il payload del **JWT** contiene i seguenti parametri obbligatori.

| Claim | Descrizione | Supportato da |
|---|---|---|
| **client_id** | Vedi OpenID.Registration[170]. DEVE essere valorizzato con un HTTPS URL che identifica univocamente il RP. | |
| **code_challenge** | Come definito nella Tabella dei parametri HTTP. | |
| **code_challenge_method** | Come definito nella Tabella dei parametri HTTP. | |
| **nonce** | Vedi OpenID.Core#AuthRequest[171]. DEVE essere una stringa casuale di almeno 32 caratteri alfanumerici. 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. | |
| **prompt** | Vedi OpenID.Core#AuthRequest[172]. I valori consentiti sono:<br>**consent**: Se non è già attiva una sessione di Single Sign-On, l'OP fa una richiesta di autenticazione all'utente. Quindi chiede il consenso al trasferimento degli attributi.<br>**consent login**: l'OP forza una richiesta di autenticazione all'utente. Quindi chiede il consenso al trasferimento degli attributi. | |
| **redirect_uri** | Vedi OpenID.Core#AuthRequest[173]. DEVE essere una URL indicata nel *Metadata RP* (pagina 27). | |
| **response_type** | Vedi OpenID.Core#AuthRequest[174]. Come definito dal parametro **response_types_supported** nel *Metadata OP* (pagina 24). | |
| **scope** | Come definito nella Tabella dei parametri HTTP. | |
| **acr_values** | Vedi OpenID.Core#AuthRequest[175]. Come definito dal parametro **acr_values_supported** nel *Metadata OP* (pagina 24). Valori di riferimento della classe di contesto dell'Authentication Request. DEVE essere una stringa separata da uno spazio, che specifica i valori "acr" richiesti in ordine di preferenza. L'OP PUÒ utilizzare un'autenticazione ad un livello più alto di quanto richiesto. Tale scelta non DEVE comportare un esito negativo della richiesta. | |
| **claims** | Vedi OpenID.Core#ClaimsRequestParameter[176]. Vedi Sezione "Parametri scope e claims". | |
| **state** | Vedi OpenID.Core#AuthRequest[177]. DEVE essere una stringa casuale di almeno 32 caratteri alfanumerici. Identificativo univoco della sessione lato RP. Questo valore verrà restituito al client nella risposta al termine dell'autenticazione. | |
| **exp** | UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[178] | |
| **iat** | UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[179] | |
| **iss** | DEVE corrispondere al *client_id*. | |
| **aud** | DEVE corrispondere all'identificativo del OP (parametro *issuer* presente nel *Metadata OP* (pagina 24).) | |
| **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. | |

**Nota:** **PKCE** è un'estensione del protocollo *OAuth 2.0* prevista anche nel profilo *iGov* (International Government Assurance Profile for OAuth 2.0[180]) 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.

Di seguito un script Python di esempio per generare i parametri richiesti.

```
import hashlib
import base64
import re

def get_pkce(code_challenge_method: str = "S256", code_challenge_length: int = 64):
    hashers = {"S256": hashlib.sha256}

    code_verifier = base64.urlsafe_b64encode(os.urandom(40)).decode("utf-8")
    code_verifier = re.sub("[^a-zA-Z0-9]+", "", code_verifier)

    code_challenge = hashers.get(code_challenge_method)(
        code_verifier.encode("utf-8")
    ).digest()
    code_challenge = base64.urlsafe_b64encode(code_challenge).decode("utf-8")
    code_challenge = code_challenge.replace("=", "")

    return {
        "code_verifier": code_verifier,
        "code_challenge": code_challenge,
        "code_challenge_method": code_challenge_method,
    }
```

### Parametri scope e claims

Gli attributi dell'utente POSSONO essere richiesti dal RP nell'Authorization Request usando il parametro **claims**.

Non è possibile richiedere attributi SPID nell' ID Token. Gli attributi dell'utente sono disponibili all'interno della response dello UserInfo endpoint.

[170] https://openid.net/specs/openid-connect-registration-1_0.html
[171] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
[172] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
[173] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
[174] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
[175] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
[176] https://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter
[177] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest
[178] https://tools.ietf.org/html/rfc7519.html
[179] https://tools.ietf.org/html/rfc7519.html
[180] https://openid.net/specs/openid-igov-oauth2-1_0-03.html#Section-3.1.7

Gli attributi dell'utente POSSONO essere richiesti dal RP nell'Authorization Request usando i parametri **scope** o **claims**.

Nel caso di utilizzo del parametro **scope** i seguenti valori sono supportati:

- **profile**: usando questo valore è possibile ottenere il profilo utente di default che corrisponde al Minimum Dataset eIDAS:
  - *family_name*,
  - *given_name*,
  - *birthdate*,
  - *https://attributes.eid.gov.it/fiscal_number* (National Unique Identifier).
- **email**: questo valore permette di ottenere, se resi disponibili dall'utente, i seguenti attributi:
  - *email*,
  - *email_verified*.

Il parametro **scope** PUÒ contenere uno o più valori separati da uno spazio. Ad esempio l'utilizzo congiunto di *profile* e *email* permette di ottenere l'unione degli insiemi degli attributi (Minimum Dataset eIDAS e l'email). Nel caso di richiesta di singoli attributi dell'utente o specifiche combinazioni di essi, Il RP DOVREBBE usare il parametro **claims**.

Gli attributi richiesti tramite il parametro **scope** sono disponibili sia nell'ID Token e sia nella risposta allo *userinfo endpoint*.

**Avvertimento:** Quando il parametro **scope** contiene solo il valore **openid** e il parametro **claims** non è presente oppure non è valorizzato, la response dello userinfo endpoint NON DEVE contenere nessun attributo utente ma soltanto il claim *sub*.

---

Per la definizione del parametro **claims** e la modalità di utilizzo per la richiesta degli attributi dell'utente si può fare riferimento a OpenID.Core#ClaimsParameter[181].

### 1.14.2 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.

**Vedi anche:**

- https://tools.ietf.org/html/rfc6749#section-4.1.2
- https://openid.net/specs/openid-connect-core-1_0.html#AuthRequestValidation

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[182]):

[181] https://openid.net/specs/openid-connect-core-1_0.html#ClaimsParameter
[182] https://openid.net/specs/openid-connect-core-1_0.html#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. | |
| **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. | |
| **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. | |

Esempio di Authorization Response dell'OP:

```
http://rp-test.it/oidc/rp/callback/?
↪code=a032faf23d986353019ff8eda96cadce2ea1c368f04bf4c5e1759d559dda1c08056c7c4d4e8058cb002a0c8fa
↪state=2Ujz3tbBHWQEL4XPFSJ5ANSjkhd7IlfC&iss=http%3A%2F%2Fop-test%2Foidc%2Fop
↪%2F
```

### 1.14.3 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[183].

| Claim | Descrizione | Supportato da |
|---|---|---|
| **Errore** | Vedi *Codici di errori* (pagina 37) | |
| **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[184]) | |
| **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*. | |

[183] https://www.spid.gov.it/wp-content/uploads/2021/07/agid-spid-lg-interfacce-informazioni-idp-sp.pdf
[184] https://www.spid.gov.it/wp-content/uploads/2021/07/agid-spid-lg-interfacce-informazioni-idp-sp.pdf

### 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**[185]). | *302 Found* | |
| *unauthorized_client* | Il client non è autorizzato a richiedere un authorization code (**RFC 6749#section-4.1.2.1**[186]). | *302 Found* | |
| *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**[187]). | *302 Found* | |
| *invalid_scope* | Sono stati richiesti degli scope non validi (**RFC 6749#section-4.1.2.1**[188]). | *302 Found* | |
| *server_error* | L'OP ha riscontrato un problema interno (**RFC 6749#section-4.1.2.1**[189]). | *302 Found* | |
| *temporarily_unavailable* | L'OP ha riscontrato un problema interno temporaneo (**RFC 6749#section-4.1.2.1**[190]). | *302 Found* | |
| *unsupported_response_type* | Il response_type richiesto non è supportato (**RFC 6749#section-4.1.2.1**[191]). | *302 Found* | |
| *login_required* | L'OP richiede l'autenticazione da parte dell'utente (OpenID.Core#AuthError[192]). | *302 Found* | |
| *consent_required* | L'OP richiede il consenso esplicito da parte dell'utente (OpenID.Core#AuthError[193]). | *302 Found* | |
| *request_uri_not_supported* | L'OP non supporta l'uso del parametro *request_uri* (OpenID.Core#AuthError[194]). | *302 Found* | |
| *registration_not_supported* | L'OP non supporta l'uso del parametro *registration* (OpenID.Core#AuthError[195]). | *302 Found* | |
| *invalid_request_object* | Il parametro *request* contiene un *Request Object* non valido (OpenID.Core#AuthError[196]). | *302 Found* | |

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

## 1.15 Token Endpoint

Al termine del flusso di autenticazione descritto nel paragrafo precedente, 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*. Vedi la Sezione *Refresh Token* (pagina 43)).

I token restituiti devono essere JWT firmati.

---

In presenza di una sessione lunga revocabile[197], il RP PUÒ chiamare il Token Endpoint inviando il *Refresh Token* in suo possesso per ottenere un nuovo *Access Token* e *ID Token*.

---

**Nota:** Il metodo di autenticazione del RP presso il token endpoint è il **private_key_jwt** (OpenID.Core#ClientAuthentication[198]).

---

**Vedi anche:**

- https://tools.ietf.org/html/rfc6749#section-3.2
- https://openid.net/specs/openid-connect-core-1_0.html#TokenEndpoint
- https://openid.net/specs/openid-igov-oauth2-1_0-03.html#Section-2.1.2
- https://openid.net/specs/openid-igov-openid-connect-1_0-03.html#Section-2.2

### 1.15.1 Request

Di seguito i claim che DEVONO essere inseriti nella *Token Request*.

**Esempio di richiesta con authorization code (caso 1)**

[185] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[186] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[187] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[188] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[189] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[190] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[191] https://tools.ietf.org/html/rfc6749.html#section-4.1.2.1
[192] https://openid.net/specs/openid-connect-core-1_0.html#AuthError
[193] https://openid.net/specs/openid-connect-core-1_0.html#AuthError
[194] https://openid.net/specs/openid-connect-core-1_0.html#AuthError
[195] https://openid.net/specs/openid-connect-core-1_0.html#AuthError
[196] https://openid.net/specs/openid-connect-core-1_0.html#AuthError
[197] https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf#page=47
[198] https://openid.net/specs/openid-connect-core-1_0.html#ClientAuthentication

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

client_id=https://rp.spid.agid.gov.it&
client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
 ↪eyJzdWIiOiIxMjM0NTY3ODkwIiw
ibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.
 ↪LVyRDPVJm0S9q7oiXcYVIIqGWY0wWQlqxvFGYswL...&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type
 ↪%3Ajwtbearer&
code=usDwMnEzJPpG5oaV8x3j&
code_verifier=9g8S40MozM3NSqjHnhi7OnsE38jklFv2&
grant_type=authorization_code
```

**Vedi anche:**

- https://openid.net/specs/openid-connect-core-1_0.html#RPAuthentication

**Esempio di richiesta con Refresh Token (caso 2):**

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

client_id=https://rp.spid.agid.gov.it&
client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
 ↪eyJzdWIiOiIxMjM0NTY3ODkwIiw
ibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.
 ↪LVyRDPVJm0S9q7oiXcYVIIqGWY0wWQlqxvFGYswL...&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type
 ↪%3Ajwtbearer&
grant_type=refresh_token&
refresh_token=8xLOxBtZp8
```

| Claim | Descrizione | Supportato da |
|---|---|---|
| **client_id** | Vedi OpenID.Registration[199]. DEVE essere valorizzato con un HTTPS URL che identifica univocamente il RP. | |
| **client_assertion** | JWT firmato con la chiave privata del Relying Party contenente i seguenti parametri:<br>**iss**: DEVE corrispondere al valore *client_id*<br>**sub**: DEVE corrispondere al valore *iss*<br>**aud**: URL del Token Endpoint dell'OP<br>**iat**: UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[200].<br>**exp**: UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[201]<br>**jti**: Identificatore univoco per questa richiesta di autenticazione, generato dal client. Ad esempio in formato *uuid4*. | |
| **client_assertion_type** | Deve assumere il seguente valore: **urn:ietf:params:oauth:client-assertion-type:jwtbearer** | |
| **code** | Codice di autorizzazione restituito nell'Authentication response. Obbligatorio solo se **grant_type** è **authorization_code** | |
| **code_verifier** | Codice di verifica del code_challenge. Obbligatorio solo se **grant_type** è **authorization_code** | |
| **grant_type** | Tipo di credenziale presentata dal RP per la richiesta corrente. PUÒ assumere uno dei seguenti valori:<br>• **authorization_code**<br>• **refresh_token** | |
| **refresh_token** | Obbligatorio solo se **grant_type** è **refresh_token** | |

### 1.15.2 Response

L'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"[202].

L'ID Token deve essere formato secondo le indicazioni del paragrafo successivo.

La risposta DEVE contenere i seguenti claim.

**Esempio di risposta:**

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

(continues on next page)

[199] https://openid.net/specs/openid-connect-registration-1_0.html
[200] https://tools.ietf.org/html/rfc7519.html
[201] https://tools.ietf.org/html/rfc7519.html
[202] https://openid.net/specs/openid-igov-oauth2-1_0-03.html#Section-3.2.1

(continua dalla pagina precedente)

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

| Claim | Descrizione | Supportato da |
|---|---|---|
| **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** | Disponibile sono nel caso di sessione lunga revocabile[203]. 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*. | |
| **expires_in** | Scadenza dell'*Access Token* in secondi. | |
| **id_token** | ID Token in formato JWT (vedi paragrafo successivo) | |

### 1.15.3 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.

**Esempio del contenuto di intestazione di payload di un Access Token:**

```
{
  "alg": "RS256",
  "kid": "dB67gL7ck3TFiIAf7N6_7SHvqk0MDYMEQcoGGlkUAAw",
  "typ": "at+jwt"
}
.
{
  "iss":"https://op.spid.agid.gov.it/",
  "sub": "9sd798asd98asui23hiuds89y798sfyg",
  "aud": [
  "https://rp.spid.example.it"
  ],
  "client_id": "https://rp.spid.example.it",
  "scope": "openid",
  "jti": "9ea42af0-594c-4486-9602-8a1f8dde42d3",
  "exp": 1656859559,
  "iat": 1656857579
}
```

[203] https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf#page=47

| 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. | |
| **sub** | Vedi OpenID.Core#SubjectIDTypes[204]. DEVE essere di tipo *pairwise*. | |
| **client_id** | DEVE essere valorizzato con un HTTPS URL che identifica univocamente il RP. | |
| **aud** | DEVE coincidere con il valore *client_id*. Il RP DEVE verificare che questo valore corrisponda al proprio client ID. | |
| **scope** | L'OP DOVREBBE inserire il parametro *scope* come previsto in **RFC 9068**[205] Sezione 2.2.3. DEVE coincidere con il valore presente in fase di richiesta di autenticazione. | |
| **iat** | UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[206] | |
| **exp** | UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[207] | |
| **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. | |

### 1.15.4 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* (pagina 27) ). 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**[208]).

---

Di seguito i claim disponibili nell'ID Token.

**Esempio del contenuto di intestazione e di payload di un ID Token:**

```
{
  "alg": "RS256",
  "kid": "dB67gL7ck3TFiIAf7N6_7SHvqk0MDYMEQcoGGlkUAAw"
}
.
{
    "iss":"https://op.spid.agid.gov.it/",
    "sub":"9sd798asd98asui23hiuds89y798sfyg",
    "aud":"https://rp.spid.example.it/auth",
    "acr":"https://www.spid.gov.it/SpidL2",
```

(continues on next page)

[204] https://openid.net/specs/openid-connect-core-1_0.html#SubjectIDTypes
[205] https://tools.ietf.org/html/rfc9068.html
[206] https://tools.ietf.org/html/rfc7519.html
[207] https://tools.ietf.org/html/rfc7519.html
[208] https://tools.ietf.org/html/rfc7519.html#section-5.2

(continua dalla pagina precedente)

```
    "at_hash":"qiyh4XPJGsOZ2MEAyLkfWqeQ",
    "iat":1519032969,
    "nbf":1519032969,
    "exp":1519033149,
    "jti":"nw4J0zMwRk4kRbQ53G7z",
    "nonce":"MBzGqyf9QytD28eupyWhSqMj78WNqpc2"
}
```

| 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. | |
| **sub** | Vedi OpenID.Core#SubjectIDTypes[209]. DEVE essere di tipo *pairwise*. | |
| **aud** | DEVE coincidere con il valore *client_id*. Il RP DEVE verificare che questo valore corrisponda al proprio client ID. | |
| **acr** | Livello di autenticazione effettivo. PUÒ essere uguale o superiore a quello richiesto dal RP nella Authentication Request. | |
| **at_hash** | Vedi OpenID.Core#CodeIDToken[210]. Il client DEVE verificare che questo valore corrisponda all'*Access Token* restituito insieme all'ID Token. | |
| **iat** | UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[211] | |
| **nbf** | UNIX Timestamp. Istante di inizio validità del JWT in formato NumericDate, come indicato in **RFC 7519**[212]. DEVE corrispondere con il valore di **iat**. | |
| **exp** | UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[213] | |
| **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. | |
| **nonce** | Vedi OpenID.Core#AuthRequest[214]. DEVE essere una stringa casuale di almeno 32 caratteri alfanumerici. Questo valore DEVE coincidere con quello inviato dal RP nella richiesta di autenticazione. | |

**Vedi anche:**

- https://openid.net/specs/openid-connect-core-1_0.html#IDToken
- https://openid.net/specs/openid-igov-openid-connect-1_0-03.html#Section-3.1

### 1.15.5 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.

[209] https://openid.net/specs/openid-connect-core-1_0.html#SubjectIDTypes
[210] https://openid.net/specs/openid-connect-core-1_0.html#CodeIDToken
[211] https://tools.ietf.org/html/rfc7519.html
[212] https://tools.ietf.org/html/rfc7519.html
[213] https://tools.ietf.org/html/rfc7519.html
[214] https://openid.net/specs/openid-connect-core-1_0.html#AuthRequest

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. | |
| **aud** | DEVE coincidere con il valore *client_id*. Il RP DEVE verificare che questo valore corrisponda al proprio client ID. | |
| **iat** | UNIX Timestamp con l'istante di generazione del JWT, codificato come NumericDate come indicato in **RFC 7519**[215] | |
| **exp** | UNIX Timestamp con l'istante di scadenza del JWT, codificato come NumericDate come indicato in **RFC 7519**[216] | |
| **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. | |

---

Per ottenere un *Refresh Token*, il RP DEVE includere nel parametro *scope* della richiesta di autenticazione il valore *offline_access* e nel parametro *prompt* il valore *consent*. L'utilizzo di questo *scope* può essere utile in scenari nei quali un RP ha la necessità di verificare che l'identità digitale di un utente finale sia ancora valida o vuole mantenere aggiornati gli attributi che ha precedentemente raccolto durante la fase di autenticazione, ad esempio per l'invio di notifiche all'utente finale successive all'autenticazione dello stesso. **Il Refresh Token NON DEVE consentire al RP richiedente di ottenere un ID Token, nè quello precedentemente rilasciato in fase di autenticazione nè un nuovo ID Token. L'utilizzo del Refresh Token NON DEVE essere utilizzato dagli RP per ottenere una nuova autenticazione dell'utente con l'OP o rinnovare una sessione preesistente, ma PUÒ essere utilizzato come meccanismo per ottenere dallo UserInfo endpoint esclusivamente il medesimo set di attributi dell'utente richiesti in fase di autenticazione iniziale e per il quale l'utente ha espresso il consenso esplicito.** Tale consenso DEVE essere raccolto dall'OP in fase autenticazione dell'utente finale nella pagina di consenso. L'utente finale DEVE avere la possibilità di abilitare o disabilitare questa opzione prima di inviare il consenso che PUÒ essere soggetto ad un periodo di validità se definito dall'OP in base alle policy sul trattamento dei dati personali.

L'OP che riceve una richiesta di un nuovo *Access Token* tramite un *Refresh Token* PUÒ inviare una notifica all'utente tramite uno dei recapiti digitali disponibili (email, sms, notifica mobile app). L'utente che non riconosce legittima questa operazione o che vuole disabilitare questa opzione PUÒ richiedere all'OP una revoca del consenso dato (e quindi dei token emessi a seguito dello stesso) secondo le modalità rese note all'interno della pagina di raccolta del consenso. La notifica DEVE avere solo carattere informativo e non autorizzativo. All'interno della notifica DEVE essere reso noto all'utente le modalità di revoca del consenso dato. L'OP DEVE consentire all'utente di disabilitare in qualsiasi momento questa opzione tramite apposita funzionalità messa a disposizione dall'OP stesso.

Per ragioni di sicurezza, un OP DEVE restituire, insieme ad un nuovo *Access Token*, anche un nuovo *Refresh Token*, invalidando tutti i token precedentemente rilasciati (*refresh token rotation*) al RP e in relazione al soggetto interessato (utente finale). Il nuovo *Refresh Token* DEVE avere il parametro *exp* non superiore alla durata prevista.

---

Per applicazioni mobili in cui il RP intenda offrire un'esperienza utente che non richieda il reinserimento delle credenziali SPID ad ogni utilizzo dell'applicazione, si POSSONO utilizzare le sessioni lunghe revocabili utilizzando il Refresh Token come normato nelle LL.GG. OpenID Connect in SPID[217] e nell' Avviso n.41[218] . Il *Token endpoint*

[215] https://tools.ietf.org/html/rfc7519.html
[216] https://tools.ietf.org/html/rfc7519.html
[217] https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf
[218] https://www.agid.gov.it/sites/default/files/repository_files/spid-avviso-n41-integrazione_ll.gg_._openid_connect_in_spid.pdf

verifica la validità del Refresh Token e, se nella richiesta di autenticazione originaria era presente nell' *acr_values* il valore *https://www.spid.gov.it/SpidL1*, rilascia un nuovo *ID Token* valido esclusivamente per il livello 1 SPID. Per maggiori dettagli sull'utilizzo del Refresh Token nel contesto SPID, si vedano i seguenti documenti normativi:

- LL.GG. OpenID Connect in SPID[219]
- Avviso n.41 - Integrazione LL.GG. OpenID Connect in SPID[220]

### 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 (Vedi *Codici di Errore* (pagina 37)).

Fermo restando la validità del token, l'OP PUÒ fissare un periodo di validità relativo al consenso che l'utente ha fornito all'utilizzo dello *scope offline_access* e del *Refresh Token*. In prossimità del termine di validità del consenso, qualora tale termine sia previsto nelle policy dell'OP, il valore di *exp* DEVE essere calcolato come il valore minimo tra la durata di validità del token e quella del consenso.

**Nota:** Al fine di chiarire il meccanismo di rotazione si riporta di seguito un esempio non normativo dove si descrive l'emissione e il lifecyle del *Refresh Token* con validità di 30 giorni.

- t1: un RP effettua un autenticazione con *scope=offline_access*, quindi ottiene *Refresh Token* RT1 (validità 30gg)
- t2 = t1 + 4gg: l'RP fa richiesta al *Token endpoint* presentando RT1. L'OP riconosce che la richiesta proviene dallo stesso RP e rilascia un nuovo *Access Token* e nuovo *Refresh Token* RT2 con validità 30gg a partire da t2
- t3 = t1 + 32gg: dopo 28gg da t2 l'RP fa richiesta al *Token endpoint* presentando RT2. L'OP riconosce che la richiesta proviene dallo stesso RP e rilascia un nuovo *Access Token* e nuovo *Refresh Token* RT3 con validità 30gg da t3
- t4 = t1 + 64gg: dopo 32gg da t3 l'RP fa richiesta al *Token endpoint* presentando RT3. Questa volta l'OP rifiuta la richiesta con un errore perchè RT3 risulta non più valido.

[219] https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf
[220] https://www.agid.gov.it/sites/default/files/repository_files/spid-avviso-n41-integrazione_ll.gg_._openid_connect_in_spid.pdf

### 1.15.6 Codici di errore

| Claim | Descrizione | Codice HTTP | Supportato da |
|---|---|---|---|
| *invalid_client* | Problemi durante la client authentication (ad esempio, il client_id è conosciuto, non è fornita l'autenticazione del client o il metodo di autenticazione non è supportato) (**RFC 6749#section-5.2**[221]). | *401 Unauthorized* | |
| *unsupported_grant_type* | Il parametro grant_type contiene un valore non corretto (**RFC 6749#section-5.2**[222]). | *400 Bad Request* | |
| *invalid_grant* | I parametri grant_type, code, code_verifier, access_token non sono validi (**RFC 6749#section-5.2**[223]). | *400 Bad Request* | |
| *invalid_request* | La richiesta non è valida a causa della mancanza o della non correttezza di uno o più parametri (**RFC 6749#section-5.2**[224]). | *400 Bad Request* | |
| *server_error* | L'OP ha riscontrato un problema interno (**RFC 6749#section-5.2**[225]). | *400 Bad Request* | |
| *temporarily_unavailable* | L'OP ha riscontrato un problema interno temporaneo (**RFC 6749#section-5.2**[226]). | *400 Bad Request* | |

## 1.16 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.

### 1.16.1 Request

[221] https://tools.ietf.org/html/rfc6749.html#section-5.2
[222] https://tools.ietf.org/html/rfc6749.html#section-5.2
[223] https://tools.ietf.org/html/rfc6749.html#section-5.2
[224] https://tools.ietf.org/html/rfc6749.html#section-5.2
[225] https://tools.ietf.org/html/rfc6749.html#section-5.2
[226] https://tools.ietf.org/html/rfc6749.html#section-5.2

Lo UserInfo Endpoint DEVE supportare l'uso del solo metodo HTTP GET **RFC 2616**[227] e DEVE accettare e validare l'Access Token inviato all'interno del campo Authorization dell'Header, di tipo Bearer **RFC 6750**[228].

Lo UserInfo Endpoint DEVE supportare l'uso dei metodi HTTP GET e POST **RFC 2616**[229] e DEVE accettare e validare l'Access Token inviato all'interno del campo Authorization dell'Header, di tipo Bearer **RFC 6750**[230].

```
GET https://op.spid.agid.gov.it/userinfo
Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6ImRCNjdnTDdja ...
```

**Vedi anche:**

- https://openid.net/specs/openid-connect-core-1_0.html#UserInfo
- https://openid.net/specs/openid-igov-openid-connect-1_0-03.html#Section-4

### 1.16.2 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.[231].

L'header JOSE DEVE contenere il parametro *cty* (Content Type) valorizzato con *JWT* (vedi **RFC 7519#section-5.2**[232]).

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,
```

(continues on next page)

[227] https://tools.ietf.org/html/rfc2616.html
[228] https://tools.ietf.org/html/rfc6750.html
[229] https://tools.ietf.org/html/rfc2616.html
[230] https://tools.ietf.org/html/rfc6750.html
[231] https://openid.net/specs/openid-connect-core-1_0.html#UserInfoResponse
[232] https://tools.ietf.org/html/rfc7519.html#section-5.2

(continua dalla pagina precedente)

```
   "nbf": 1519032969,
   "exp": 1519033149,
   "sub": "OP-1234567890",
   "name": "Mario",
   "family_name": "Rossi",
   "https://attributes.spid.gov.it/fiscalNumber": "MROXXXXXXXXXXXXX"
}
```

Il payload del JWT è un JSON contenente 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. | |
| **aud** | String. Identificatore del soggetto destinatario della response (RP). Il RP DEVE verificare che il valore coincida con il proprio client_id. | |
| **iss** | String. URI che identifica univocamente l'OP. | |
| **<attributo>** | I claim richiesti al momento dell'autenticazione. | |

### 1.16.3 Codici di errore

Come definiti per *Token endpoint* (pagina 46).

## 1.17 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 |
|---|---|---|
| **$PREFIX/spid_code** Categoria: anagrafica | Codice identificativo. String. Il codice identificativo è assegnato dal gestore dell'identità digitale e deve essere univoco.<br>Il formato è il seguente: `<codice_Identificativo>=<cod_IdP><nr. univoco>`<br>Dove:<br>**<cod_IdP>**: è un codice composto da 4 lettere univocamente assegnato al gestore delle identità;<br>**<nr.univoco>**: è una stringa alfanumerica composta da 10 caratteri che il gestore delle identità genera in maniera univoca nell'ambito del proprio dominio.<br>Esempio:<br>`"$PREFIX/spid_code":"ABCD123456789A"` | |
| **given_name** Categoria: anagrafica | Nome. String. Stringa composta da una sequenza di parole con carattere iniziale maiuscolo, intervallate da spazi singoli.<br>Esempio:<br>`"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.<br>Esempio:<br>`"family_name":"Bianchi Verdi"` | |
| **place_of_birth** Categoria: anagrafica | Luogo di nascita, Provincia di nascita. JSON Object:<br>"**locality** : Stringa corrispondente al codice catastale (Codice Belfiore) del Comune o della nazione estera di nascita (Es. "F205" per la città di Milano)<br>"**region**" : Stringa corrispondente alla sigla della provincia di nascita<br>Esempio:<br>`"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:<br>`"birthdate":"2002-09-24"` | |
| **gender** Categoria: anagrafica | Sesso. String. Valori ammessi: "female" per sesso femminile "male" per sesso maschile Esempio:<br>`"gender":"female"` | |
| **$PREFIX/company_name** Categoria: anagrafica | Ragione o denominazione sociale. String. Stringa composta da una sequenza di parole intervallate da spazi singoli. In maiuscolo le sottostringhe corrispondenti a nomi (es. "Agenzia per l'Italia Digitale")<br>`"$PREFIX/company_name": "Agenzia per l'Italia Digitale ↪"` | |
| **$PREFIX/registered_office** Categoria: extra anagrafica | Sede legale. JSON Object: formatted, street_address, locality, region, postal_code, country, country_code. Json composto da una stringa composta da una sequenza di parole intervallate da spazi singoli rappresentanti:<br>• Tipologia( via, viale, piazza . . . )<br>• Indirizzo<br>• Nr.civico<br>• CAP<br>• Luogo<br>• Provincia<br>la stringa è inserita nel claim "formatted" del JSON Object "address"<br>Esempio:<br>`"$PREFIX/registered_office":{ "formatted":"via Listz 21 00144 Roma"` | |

### 1.17.1 Esempi

Si riportano per comodità 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 | `"address": { "street_address":"Via Liszt 21", "postal_code":"00144", "locality":"Roma", "region":"RM", "country_code":"IT" }` |
| Indirizzo domicilio fisico CAP domicilio fisico Comune domicilio fisico Provincia domicilio fisico Nazione domicilio fisico | `"address": { "street_address":"S.S. Salaria Km 23,800", "postal_code":"00015", "locality":"Monterotondo", "region":"RM", "country_code":"IT" }` |

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 | `"address":{ "street_address":"503,Washington Avenue", "postal_code":"12401", "locality":"Kingston", "region":"New york", "country_code":"US" }` |

## 1.18 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.

**Vedi anche:**

- https://tools.ietf.org/html/rfc7662
- https://openid.net/specs/openid-igov-oauth2-1_0-03.html#Section-3.2.2

### 1.18.1 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.

**Esempio:**

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

client_assertion=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiw
ibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9q7oiXcYVIIqGWY0wWQlqxvFGYswLF88 ... &
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwtbearer&
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
token=eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTQsImF1ZCI6WyJlNzFmYjcyYS05NzRmLT
QwMDEtYmNiNy1lNjdjMmJjMDAzN2YiXSwiaXNzIjoiaHR0cHM6XC9cL2FzLXZhLmV4YW1wbGUuY29tXC8
iLCJqdGkiOiIyMWIxNTk2ZC04NWQzLTQzN2MtYWQ4My1iM2YyY2UyNDcyNDQiLCJpYXQiOjE0MTg2OTg4
MTR9.FXDtEzDLbTHzFNroW7w27RLk5m0wprFfFH7h4bdFw5fR3pwiqejKmdfAbJvN3_yfAokBv06we5RA
RJUbdjmFFfRRW23cMbpGQCIk7Nq4L012X_1J4IewOQXXMLTyWQQ_BcBMjcW3MtPrY1AoOcfBOJPx1k2jw
RkYtyVTLWlff6S5gKciYf3b0bAdjoQEHd_
↪IvssIPH3xuBJkmtkrTlfWR0Q0pdpeyVePkMSI28XZvDaGnxA4j7QI5loZYeyzGR9
h70xQLVzqwwl1P0-F_0JaDFMJFO1yl4IexfpoZZsB3HhF2vFdL6D_lLeHRyH2g2OzF59eMIsM_
↪Ccs4G47862w...
```

| 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**. | |
| **client_assertion_type** | String. Valori ammessi: **urn:ietf:params:oauth:clientassertion-type:jwt-bearer** | |
| **client_id** | URI che identifica univocamente il RP. L'OP deve verificare che il client_id sia noto all'interno della Federazione. | |
| **token** | Il token su cui il RP vuole ottenere informazioni. | |

### 1.18.2 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. | |
| **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. | |
| **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. | |
| **aud** | Contiene il client ID. Il client è tenuto a verificare che questo valore corrisponda al proprio client ID. | |

### 1.18.3 Codici di errore

Come definiti per *Token endpoint* (pagina 46).

## 1.19 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 specificato nella richiesta.

---

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 richiedere la revoca dell'Access Token e dell'eventuale Refresh Token in suo possesso, se questi non fossero già scaduti.

---

**Nota:** La revoca di un Access Token comporta la revoca di tutti i Refresh Token non ancora scaduti a questo collegati.

---

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

---

---

La revoca di un Access Token NON DEVE comportare la revoca di tutti i Refresh Token a questo collegati.

La revoca di un Refresh Token DEVE comportare la revoca di tutti gli Access Token a questo collegati.

---

---

**Nota:** Il metodo di autenticazione del RP presso il *revocation endpoint* DEVE essere **private_key_jwt** (vedi il parametro *revocation_endpoint_auth_methods_supported* nella Sezione *Metadata OP* (pagina 24))

---

**Vedi anche:**

- https://tools.ietf.org/html/rfc7009

### 1.19.1 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.eyJzdWIiOiIxMjM0NTY3ODkwIiw
ibmFtZSI6IlNQSUQiLCJhZG1pbiI6dHJ1ZX0.LVyRDPVJm0S9q7oiXcYVIIqGWY0wWQlqxvFGYswLF88&
client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwtbearer&
client_id=https%3A%2F%2Frp.spid.agid.gov.it&
token=eyJhbGciOiJSUzI1NiJ9.eyJleHAiOjE0MTg3MDI0MTQsImF1ZCI6WyJlNzFmYjcyYS05NzRmLT
QwMDEtYmNiNy1lNjdjMmJjMDAzN2YiXSwiaXNzIjoiaHR0cHM6XC9cL2FzLXZhLmV4YW1wbGUuY29tXC8
iLCJqdGkiOiIyMWIxNTk2ZC04NWQzLTQzN2MtYWQ4My1iM2YyY2UyNDcyNDQiLCJpYXQiOjE0MTg2OTg4
MTR9.FXDtEzDLbTHzFNroW7w27RLk5m0wprFfFH7h4bdFw5fR3pwiqejKmdfAbJvN3_yfAokBv06we5RA
RJUbdjmFFfRRW23cMbpGQCIk7Nq4L012X_1J4IewOQXXMLTyWQQ_BcBMjcW3MtPrY1AoOcfBOJPx1k2jw
RkYtyVTLWlff6S5gKciYf3b0bAdjoQEHd_
↪IvssIPH3xuBJkmtkrTlfWR0Q0pdpeyVePkMSI28XZvDaGnxA4j7QI5loZYeyzGR9
h70xQLVzqwwl1P0-F_0JaDFMJFO1yl4IexfpoZZsB3HhF2vFdL6D_lLeHRyH2g2OzF59eMIsM_Ccs4G47862w
```

| 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**. | |
| **client_assertion_type** | String. **urn:ietf:params:oauth:clientassertion-type:jwt-bearer** | |
| **client_id** | URL HTTPS che identifica univocamente il RP. | |
| **token** | Il token su cui il RP vuole ottenere informazioni. | |

### 1.19.2 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).

### 1.19.3 Codici di errore

Come definiti per *Token endpoint* (pagina 46).

## 1.20 Logout

I RP POSSONO instaurare sessioni individuali relative agli utenti autenticati. Nei casi in cui tali sessioni individuali vengano instaurate dai RP, questi ultimi DEVONO fornire agli utenti una funzionalità di logout con lo scopo di eliminare la sessione individuale instaurata. Durante la fase di logout i RP DEVONO revocare tutti gli Access Token ancora attivi e collegati all'autenticazione degli utenti, tramite l'utilizzo del revocation endpoint (*Revocation Endpoint* (pagina 52)).

---

**Nota:** Nel caso sia supportato dall'OP un meccanismo di *offline_access* tramite *Refresh Token*, quest'ultimo NON DEVE essere revocato a seguito di un logout.

---

## 1.21 Algoritmi crittografici

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

---

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

---

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

| Algoritmi | Operazioni | Riferimento | Supportato da |
|---|---|---|---|
| **RS256** | Signature | OpenID.Core[233] and RFC7518[234]. | |
| **RS512** | Signature | RFC7518[235] | |
| **RSA-OAEP** | Key Encryption | RFC7518[236]. | |
| **RSA-OAEP-256** | Key Encryption | RFC7516[237]. | |
| **A128CBC-HS256** | Content Encryption | RFC7516[238]. | |
| **A256CBC-HS512** | Content Encryption | RFC7516[239]. | |

In SPID e CIE id è RACCOMANDATO il supporto per i seguenti algoritmi:

| Algoritmi | Operazioni | Riferimento | Applicabile a |
|---|---|---|---|
| **ES256** | Signature | OpenID.Core[240] and RFC7518[241]. | |
| **ES512** | Signature | RFC7518[242]. | |
| **PS256** | Signature | RFC7518[243]. | |
| **PS512** | Signature | RFC7518[244]. | |
| **ECDH-ES** | Key Encryption | RFC7518[245]. | |
| **ECDH-ES+A128KW** | Key Encryption | RFC7518[246]. | |
| **ECDH-ES+A256KW** | Key Encryption | RFC7518[247]. | |

[233] https://openid.net/specs/openid-connect-core-1_0-27.html
[234] https://www.rfc-editor.org/rfc/rfc7518
[235] https://www.rfc-editor.org/rfc/rfc7518
[236] https://www.rfc-editor.org/rfc/rfc7518
[237] https://www.rfc-editor.org/rfc/rfc7516
[238] https://www.rfc-editor.org/rfc/rfc7516
[239] https://www.rfc-editor.org/rfc/rfc7516

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

| Algoritmi | Operazioni | Riferimenti | Applicabile a |
|---|---|---|---|
| **none** | Signature | RFC7518[248]. | |
| **RSA_1_5** | Key Encryption | RFC7516[249]. | |
| **HS256** | Signature | RFC7518[250]. | |
| **HS384** | Signature | RFC7518[251]. | |
| **HS512** | Signature | RFC7518[252]. | |

## 1.22 Retention Policy

### 1.22.1 Gestione dei Log di un OP e di un RP

Gli OP e gli RP DEVONO mantenere:

1. 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:
     1. L'**Entity Configuration** del Entità con la quale è avvenuta la transazione.
     2. [Solo per OP] L'**Entity Statement** del SA riferito al RP (se presente).
     3. L'**Entity Statement** del TA riferito al suo discendente.
     4. 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*

---

Per ogni messaggio POSSONO essere indicizzate, ai fini di ricerca e consultazione, le seguenti informazioni:

[240] https://openid.net/specs/openid-connect-core-1_0-27.html
[241] https://www.rfc-editor.org/rfc/rfc7518
[242] https://www.rfc-editor.org/rfc/rfc7518
[243] https://www.rfc-editor.org/rfc/rfc7518
[244] https://www.rfc-editor.org/rfc/rfc7518
[245] https://www.rfc-editor.org/rfc/rfc7518
[246] https://www.rfc-editor.org/rfc/rfc7518
[247] https://www.rfc-editor.org/rfc/rfc7518
[248] https://www.rfc-editor.org/rfc/rfc7518
[249] https://www.rfc-editor.org/rfc/rfc7516
[250] https://www.rfc-editor.org/rfc/rfc7518
[251] https://www.rfc-editor.org/rfc/rfc7518
[252] https://www.rfc-editor.org/rfc/rfc7518

- authorization code
- client_id
- jti
- iss
- sub
- iat
- exp

**Avvertimento:** 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 adottai 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.

### 1.22.2 Registro storico delle chiavi pubbliche di Federazione

Al fine di consentire la verifica dei messaggi scambiati dalle Entità che partecipano alla federazione e delle relative Trust Chain, il TA DEVE pubblicare lo storico delle proprie chiavi pubbliche (JWKS) di federazione all'interno di un registro reso disponibile a tutti i partecipanti tramite l'endpoint */.well-known/openid-federation-jwks*. Per ulteriori dettagli tecnici si rimanda alla Sezione 7.5 di OIDC-FED[253].

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

## 1.23 Differenze tra SPID e CIE id

In questa sezione sono riportate le principali differenze tra i profili implementativi SPID e CIE id.

### 1.23.1 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.

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

### 1.23.2 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[254].

### 1.23.3 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* (pagina 34).

### 1.23.4 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.

### 1.23.5 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[255] e nell' Avviso n.41[256] . 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* (pagina 43).

### 1.23.6 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.

### 1.23.7 Introspection Endpoint

CIE id prevede il solo parametro *active* nella risposta dell'Introspection endpoint. SPID aggiunge ulteriori parametri come specificato nella sezione *Introspection Endpoint* (pagina 50).

[254] https://www.ietf.org/archive/id/draft-ietf-oauth-security-topics-19.html
[255] https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf
[256] https://www.agid.gov.it/sites/default/files/repository_files/spid-avviso-n41-integrazione_ll.gg_._openid_connect_in_spid.pdf

### 1.23.8 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.

## 1.24 Differenze con OIDC iGov

CIE OpenID Connect e SPID OpenID Connect sono basati su iGov.OIDC[257] 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[258]. 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**.

[257] https://openid.net/specs/openid-igov-openid-connect-1_0-03.html
[258] https://www.ietf.org/archive/id/draft-ietf-oauth-security-topics-19.html

- 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**[259].

## 1.25 Differenze con OIDC Federation

In questa sezione sono elencate le differenze che intercorrono tra lo standard ufficiale e l'implementazione SPID e CIE.

### 1.25.1 Client Registration

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

### 1.25.2 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* (pagina 59).

### 1.25.3 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.

## 1.26 Considerazioni di Sicurezza

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

### 1.26.1 Trust Mark come deterrente contro gli abusi

L'implementazione dei Trust Mark e il filtro su questi in fase di Federation Entity Discovery risulta necessario contro gli attacchi destinati al consumo delle risorse. Un OP attaccato con un numero ingente di connessioni presso il suo endpoint di *authorization*, contenenti **client_id** e **authority_hints** fasulli, produrrebbe svariate connessioni verso sistemi di terze parti nel tentativo di trovare un percorso verso la TA e instaurare la fiducia con il richiedente.

L'OP DEVE validare staticamente il TM oppure DEVE escludere a priori la richiesta ove il TM non risultasse presente, in caso di assenza o non validità di un TM la procedura di Federation Entity Discovery NON DEVE essere avviata e NON DEVE creare di conseguenza connessioni verso sistemi di terze parti.

[259] https://tools.ietf.org/html/rfc9068.html

### 1.26.2 Numero Massimo di authority_hints

All'interno di una Federazione il Trust Anchor decide quante intermediazioni consentire tra di lui e le Foglie, mediante la constraint denominata **max_path_lenght**. Questo tipo di relazione è di tipo verticale, dalla Foglia alla radice. Questo attributo se valorizzato ad esempio con un valore numerico intero pari a 1 indica che soltanto un SA è consentito tra una Foglia e il TA.

Ogni Foglia DEVE pubblicare i suoi superiori all'interno della lista contenuta nel claim **authority_hints**. Una Foglia all'interno della Federazione PUÒ avere superiori afferenti a diverse Federazioni. L'analisi dei superiori disponibili introduce un modello di navigazione orizzontale, ad esempio un OP tenta di trovare il percorso più breve verso il Trust Anchor attraverso tutti gli URL contenuti all'interno dell'array **authority_hints** prima di fare un ulteriore movimento verticale, a salire, verso uno degli Intermediari presenti in questo array.

La soglia **max_path_lenght** si applica per la navigazione verticale e superata questa soglia senza aver trovato il TA, la procedura di Federation Entity Discovery DEVE essere interrotta. Si faccia l'esempio di un RP discendente di un SA che a sua volta è discendente di un altro SA, essendo il valore di **max_path_lenght** pari a 1 e, superata questa soglia senza aver trovato il Trust Anchor, la procedura DEVE essere interrotta.

Allo stesso tempo la specifica OIDC Federation 1.0 non definisce un limite per il numero di **authority_hints**, questo perché nessun Trust Anchor può limitare il numero di Federazioni alle quali un partecipante può aderire. Per questa ragione è utile che gli implementatori adottino un limite massimo del numero di elementi consentiti all'interno dell'Array authority_hint. Questo per evitare che un numero esagerato di URL contenuti nella lista di **authority_hints**, dovuto ad una cattiva configurazione di una Foglia, produca un consumo di risorse eccessivo.

### 1.26.3 Resolve endpoint

Questo endpoint DEVE rilasciare i Metadata, i Trust Mark e la Trust Chain già precedentemente elaborata e NON DEVE innescare una procedura di Federation Entity Discovery ad ogni richiesta pervenuta, a meno che questo endpoint non venga protetto con un meccanismo di autenticazione dei client, come ad esempio *private_key_jwt* [OIDC-CORE]. In caso di utilizzo di *private_key_jwt* il valore presente nel parametro *sub* del *private_key_jwt* DEVE coincidere con quello presente nella richiesta al Resolve endpoint.

## 1.27 Buone Pratiche

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

### 1.27.1 Specializzare le chiavi pubbliche OpenID Core e Federation

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

### 1.27.2 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.

## 1.28 Esempi

In questa sezione sono raccolti tutti gli esempi non normativi delle richieste e delle risposte agli endpoint di Federazione definiti all'interno di questo documento.

Tutte le response di tipo jose sono state decodificate e rappresentate insieme alle loro intestazioni per migliorare la lettura.

### 1.28.1 EN 1. Entity Configuration Request

```
GET /.well-known/openid-federation HTTP/1.1
Host: rp.example.it
```

### 1.28.2 EN 1.1. Entity Configuration Response Relying Party

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

{
    "alg": "RS256",
    "kid": "2HnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs",
    "typ": "entity-statement+jwt"
}
.
{
    "exp": 1649590602,
    "iat": 1649417862,
    "iss": "https://rp.example.it/",
    "sub": "https://rp.example.it/",
    "jwks": {
        "keys": [
            {
                "kty": "RSA",
                "n": "5s4qi ...",
                "e": "AQAB",
                "kid": "2HnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
            }
        ]
    },
    "metadata": {
        "openid_relying_party": {
            "application_type": "web",
            "client_id": "https://rp.example.it/",
            "client_registration_types": [
                "automatic"
            ],
            "jwks": {
```

(continues on next page)

(continua dalla pagina precedente)

```
                "keys": [
                    {
                        "kty": "RSA",
                        "use": "sig",
                        "n": "1Ta-sE ...",
                        "e": "AQAB",
                        "kid": "YhNFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
                    }
                ]
            },
            "client_name": "Name of an example organization",
            "contacts": [
                "ops@rp.example.it"
            ],
            "grant_types": [
                "refresh_token",
                "authorization_code"
            ],
            "redirect_uris": [
                "https://rp.example.it/oidc/rp/callback/"
            ],
            "response_types": [
                "code"
            ],
            "subject_type": "pairwise"
        },
        "federation_entity": {
            "federation_resolve_endpoint": "https://rp.example.it/resolve/"
        }
    },
    "trust_marks": [
        {
            "id": "https://registry.agid.gov.it/openid_relying_party/public/",
            "trust_mark": "eyJh ..."
        }
    ],
    "authority_hints": [
        "https://registry.agid.gov.it/"
    ]
}
```

### 1.28.3 EN 1.2. Entity Configuration Response Openid Provider

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

{
    "alg": "RS256",
    "kid": "dB67gL7ck3TFiIAf7N6_7SHvqk0MDYMEQcoGGlkUAAw",
    "typ": "entity-statement+jwt"
}
.
{
    "exp": 1649610249,
```

(continues on next page)

(continua dalla pagina precedente)


```
    "iat": 1649437449,
    "iss": "https://openid.provider.it/",
    "sub": "https://openid.provider.it/",
    "jwks": {
        "keys": [
            {
                "kty": "RSA",
                "e": "AQAB",
                "n": "01_4a ...",
                "kid": "dB67gL7ck3TFiIAf7N6_7SHvqk0MDYMEQcoGGlkUAAw"
            }
        ]
    },
    "metadata": {
        "openid_provider": {
            "authorization_endpoint": "https://openid.provider.it/authorization",
            "revocation_endpoint": "https://openid.provider.it/revocation/",
            "id_token_encryption_alg_values_supported": [
                "RSA-OAEP"
            ],
            "id_token_encryption_enc_values_supported": [
                "A128CBC-HS256"
            ],
            "token_endpoint": "https://openid.provider.it/token/",
            "userinfo_endpoint": "https://openid.provider.it/userinfo/",
            "introspection_endpoint": "https://openid.provider.it/introspection/",
            "claims_parameter_supported":true,
            "contacts": [
                "ops@https://idp.it"
            ],
            "client_registration_types_supported": [
                "automatic"
            ],
            "code_challenge_methods_supported": [
                "S256"
            ],
            "request_authentication_methods_supported": {
                "ar": [
                    "request_object"
                ]
            },
            "acr_values_supported": [
                "https://www.spid.gov.it/SpidL1",
                "https://www.spid.gov.it/SpidL2",
                "https://www.spid.gov.it/SpidL3"
            ],
            "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",
```


(continues on next page)

(continua dalla pagina precedente)

```
                "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"
            ],
            "grant_types_supported": [
                "authorization_code",
                "refresh_token"
            ],
            "id_token_signing_alg_values_supported": [
                "RS256",
                "ES256"
            ],
            "issuer": "https://openid.provider.it/",
            "jwks": {
                "keys": [
                    {
                        "kty": "RSA",
                        "use": "sig",
                        "n": "1Ta-sE ...",
                        "e": "AQAB",
                        "kid": "FANFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
                    }
                ]
            },
            "scopes_supported": [
                "openid",
                "offline_access"
            ],
            "logo_uri": "https://openid.provider.it/static/svg/spid-logo-c-lb.svg",
            "organization_name": "SPID OIDC identity provider",
            "op_policy_uri": "https://openid.provider.it/it/website/legal-information/
↪",
            "request_parameter_supported":true,
            "request_uri_parameter_supported":true,
            "require_request_uri_registration":true,
            "response_types_supported": [
                "code"
            ],
            "subject_types_supported": [
                "pairwise",
                "public"
            ],
            "token_endpoint_auth_methods_supported": [
                "private_key_jwt"
            ],
            "token_endpoint_auth_signing_alg_values_supported": [
                "RS256",
                "RS384",
                "RS512",
                "ES256",
                "ES384",
                "ES512"
            ],
```

(continues on next page)

(continua dalla pagina precedente)

```
            "userinfo_encryption_alg_values_supported": [
                "RSA-OAEP",
                "RSA-OAEP-256"
            ],
            "userinfo_encryption_enc_values_supported": [
                "A128CBC-HS256",
                "A192CBC-HS384",
                "A256CBC-HS512",
                "A128GCM",
                "A192GCM",
                "A256GCM"
            ],
            "userinfo_signing_alg_values_supported": [
                "RS256",
                "RS384",
                "RS512",
                "ES256",
                "ES384",
                "ES512"
            ],
```

### 1.28.4 EN 1.3. Entity Configuration Response Intermediary

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

{
    "alg": "RS256",
    "kid": "em3cmnZgHIYFsQ090N6B3Op7LAAqj8rghMhxGmJstqg",
    "typ": "entity-statement+jwt"
}
.
{
    "exp": 1649631824,
    "iat": 1649459024,
    "iss": "https://aggregatore.it/",
    "sub": "https://aggregatore.it/",
    "jwks": {
        "keys": [
            {
                "kty": "RSA",
                "e": "AQAB",
                "n": "14aW ...",
                "kid": "em3cmnZgHIYFsQ090N6B3Op7LAAqj8rghMhxGmJstqg"
            }
        ]
    },

    "metadata": {
        "federation_entity": {
            "contacts": [
                "soggetto@aggregatore.it"
            ],
            "federation_fetch_endpoint": "https://aggregatore.it/fetch/",
```

(continues on next page)

(continua dalla pagina precedente)

```
            "federation_resolve_endpoint": "https://aggregatore.it/resolve/",
            "federation_list_endpoint": "https://aggregatore.it/list/",
            "homepage_uri": "https://soggetto.aggregatore.it",
            "name": "Soggetto Aggregatore di esempio"
        },
        "trust_mark_issuer": {
            "federation_status_endpoint": "https://aggregatore.it/trust_mark_status/",

        }
    },
    "trust_marks": [
        {
            "id": "https://registry.gov.it/intermediate/private/full/",
            "trust_mark": "eyJh ..."
        }
    ],
    "authority_hints": [
        "https://registry.agid.gov.it/"
    ]
}
```

### 1.28.5 EN 1.4. Entity Configuration Response Trust Anchor

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

{
    "alg": "RS256",
    "kid": "FifYx03bnosD8m6gYQIfNHNP9cM_Sam9Tc5nLloIIrc",
    "typ": "entity-statement+jwt"
}
.
{
    "exp": 1649375259,
    "iat": 1649373279,
    "iss": "https://registry.agid.gov.it/",
    "sub": "https://registry.agid.gov.it/",
    "jwks": {
        "keys": [
            {
                "kty": "RSA",
                "n": "3i5vV-_ ...",
                "e": "AQAB",
                "kid": "FifYx03bnosD8m6gYQIfNHNP9cM_Sam9Tc5nLloIIrc"
            }
        ]
    },
    "metadata": {
        "federation_entity": {

            "organization_name": "example TA"
            "contacts":[
                "spid.tech@agid.gov.it"
            ],
```

(continues on next page)

(continua dalla pagina precedente)

```
            "homepage_uri": "https://registry.agid.gov.it/",
            "logo_uri":"https://registry.agid.gov.it/static/svg/logo.svg",
            "federation_fetch_endpoint": "https://registry.agid.gov.it/fetch/",
            "federation_resolve_endpoint": "https://registry.agid.gov.it/resolve/",
            "federation_list_endpoint": "https://registry.agid.gov.it/list/",
            "federation_trust_mark_status_endpoint": "https://registry.agid.gov.it/
↪trust_mark_status/"
        }
    },
    "trust_marks_issuers": {
        "https://registry.agid.gov.it/openid_relying_party/public/": [
            "https://registry.spid.agid.gov.it/",
            "https://public.intermediary.spid.it/"
        ],
        "https://registry.agid.gov.it/openid_relying_party/private/": [
            "https://registry.spid.agid.gov.it/",
            "https://private.other.intermediary.it/"
        ]
    },
    "constraints": {
        "max_path_length": 1
    }
}
```

### 1.28.6 EN 1.5. Trust Mark issued by TA to a RP

```
{
    "trust_marks": [
        {
            "id": "https://registry.interno.gov.it/openid_relying_party/public/",
            "iss": "https://registry.interno.gov.it/",
            "trust_mark": "$JWT"
        }
    ]
}
```

Where the $JWT payload is:

```
{
    "id": "https://registry.interno.gov.it/openid_relying_party/public/",
    "iss": "https://sa.esempio.it/",
    "sub": "https://rp.esempio.it/",
    "iat": 1579621160,
    "organization_type": "public",
    "id_code": {
       "ipa_code": "123456",
       "aoo_code": "Uff_protocollo"
    }
    "email": "email_or_pec@rp.it",
    "organization_name#it": "Denominazione del RP",
    "ref": "https://documentazione_di_riferimento.it/"
}
```

### 1.28.7 EN 1.6. Trust Mark issued by TA to a SA

```
{
    "trust_marks": [
        {
            "id": "https://registry.interno.gov.it/intermediate/private/full/",
            "iss": "https://registry.interno.gov.it/",
            "trust_mark": "$JWT"
        }
    ]
}
```

Where the $JWT payload is:

```
{
    "id": "https://registry.interno.gov.it/intermediate/private/full/",
    "iss": "https://registry.interno.gov.it/",
    "sub": "https://sa.esempio.it/",
    "iat": 1579621160,
    "organization_type": "private",
    "id_code": {
       "fiscal_number": "1234567890"
    }
    "email": "email_or_pec@intermediate.it",
    "organization_name#it": "Denominazione del SA",
    "sa_profile": "full",
    "ref": "https://documentazione_di_riferimento.it/"
}
```

### 1.28.8 EN 1.7. Trust Mark issued by SA to a RP

```
{
    "trust_marks": [
        {
            "id": "https://registry.interno.gov.it/openid_relying_party/public/",
            "iss": "https://sa.esempio.it",
            "trust_mark": "$JWT"
        }
    ]
}
```

Where the $JWT payload is:

```
{
    "id": "https://registry.interno.gov.it/openid_relying_party/public/",
    "iss": "https://sa.esempio.it/",
    "sub": "https://rp.esempio.it/",
    "iat": 1579621160,
    "organization_type": "public",
    "id_code": {
       "ipa_code": "987654",
    }
    "email": "email_or_pec@rp.it",
    "organization_name#it": "Denominazione del RP",
    "ref": "https://documentazione_di_riferimento.it/"
}
```

### 1.28.9 EN 2. Entity Statement Request

```
GET /fetch?sub=https://rp.example.it/
HTTP/1.1
Host: registry.agid.gov.it
```

### 1.28.10 EN 2.1 Entity Statement Response

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

{
    "alg": "RS256",
    "kid": "FifYx03bnosD8m6gYQIfNHNP9cM_Sam9Tc5nLloIIrc",
    "typ": "entity-statement+jwt"
}
.
{
    "exp": 1649623546,
    "iat": 1649450746,
    "iss": "https://registry.agid.gov.it/",
    "sub": "https://rp.example.it/",
    "jwks": {
        "keys": [
            {
                "kty": "RSA",
                "n": "5s4qi ...",
                "e": "AQAB",
                "kid": "2HnoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
            }
        ]
    },
    "metadata_policy": {
        "openid_relying_party": {
            "scope": {
                "superset_of": [
                    "openid"
                ],
                "subset_of": [
                    "openid",
                    "offline_access"
                ]
            },
            "contacts": {
                "add": [
                    "tech@example.it"
                ]
            }
        }
    },
    "trust_marks": [
        {
            "id": "https://registry.agid.gov.it/openid_relying_party/public/",
            "trust_mark": "eyJhb ..."
```

(continues on next page)

(continua dalla pagina precedente)

```
        }
    ]
}
```

### 1.28.11 EN 3. Entity List Request

```
GET /list?entity_type=openid_provider
HTTP/1.1
Host: registry.agid.gov.it
```

### 1.28.12 EN 3.1. Entity List Response

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

["https://openid-provider.it/", "https://spid.provider.it", ... ]
```

### 1.28.13 EN 4. Resolve Entity Statement Endpoint Request

```
GET /resolve/?sub=https://openid.provider.it/&anchor=https://registry.agid.gov.it/
HTTP/1.1
Host: registry.agid.gov.it
```

### 1.28.14 EN 4.1. Resolve Entity Statement Endpoint Response

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

{
    "alg": "RS256",
    "kid": "FifYx03bnosD8m6gYQIfNHNP9cM_Sam9Tc5nLloIIrc",
    "typ": "entity-statement+jwt"
}
.
{
    "iss": "https://registry.agid.gov.it/",
    "sub": "https://rp.example.it/",
    "iat": 1649355587,
    "exp": 1649410329,
    "trust_marks": [
        {
            "id": "https://registry.agid.gov.it/openid_relying_party/public/",
            "trust_mark": "eyJh ..."
        }
    ],
    "metadata": {
```

(continues on next page)

(continua dalla pagina precedente)

```
        "openid_relying_party": {
            "application_type": "web",
            "client_id": "https://rp.example.it/",
            "client_registration_types": [
                "automatic"
            ],
            "jwks": {
                "keys": [
                    {
                        "kty": "RSA",
                        "use": "sig",
                        "n": "...",
                        "e": "AQAB",
                        "kid": "5NNNoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
                    }
                ]
            },
            "client_name": "Name of an example organization",
            "contacts": [
                "ops@rp.example.it"
            ],
            "grant_types": [
                "refresh_token",
                "authorization_code"
            ],
            "redirect_uris": [
                "https://rp.example.it/oidc/rp/callback/"
            ],
            "response_types": [
                "code"
            ],
            "subject_type": "pairwise"
        }
    },
    "trust_chain": [
        "eyJhbGciOiJSUzI1NiIsImtpZCI6Ims1NEhRdERpYnlHY3M5WldWTWZ2aUhm ...",
        "eyJhbGciOiJSUzI1NiIsImtpZCI6IkJYdmZybG5oQU11SFIwN2FqVW1BY0JS ...",
        "eyJhbGciOiJSUzI1NiIsImtpZCI6IkJYdmZybG5oQU11SFIwN2FqVW1BY0JS ..."
    ]
}
```

### 1.28.15 EN 5. Trust Mark Status Request

```
POST /trust_mark_status HTTP/1.1
Host: registry.agid.gov.it
Content-Type: application/x-www-form-urlencoded

id=https%3A%2F%2registry.agid.gov.it%2Fopenid_relying_party%2Fpublic%2F
&sub=https%3A%2F%2rp.example.it%2F
```

### 1.28.16 EN 5.1. Trust Mark Status Response

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

{"active": true}
```

### 1.28.17 EN 6. Authorization Request

**Example (HTTP request):**

```
GET /auth?client_id=https://rp.spid.agid.gov.it&
response_type=code&scope=openid& code_challenge=qWJlMe0xdbXrKxTm72EpH659bUxAxw80&
code_challenge_method=S256&request=eyJhbGciOiJSUzI1NiIsImtpZCI6ImsyYmRjIn0.ew0KIC
Jpc3MiOiAiczZCaGRSa3F0MyIsDQogImF1ZCI6ICJodHRwczovL3NlcnZlci5leGFtcGxlLmNvbSIsDQo
gInJlc3BvbnNlX3R5cGUiOiAiY29kZSBpZF90b2tlbiIsDQogImNsaWVudF9pZCI6ICJzNkJoZFJrcXQz
IiwNCiAicmVkaXJlY3RfdXJpIjogImh0dHBzOi8vY2xpZW50LmV4YW1wbGUub3JnL2NiIiwNCiAic2Nvc
GUiOiAib3BlbmlkIiwNCiAic3RhdGUiOiAiYWYwaWZqc2xka2oiLA0KICJub25jZSI6ICJuLTBTNl9Xek
EyTWoiLA0KICJtYXhfYWdlIjogODY0MDAsDQogImNsYWltcyI6IA0KICB7DQogICAidXNlcmluZm8iOiA
NCiAgICB7DQogICAgICJnaXZlbl9uYW1lIjogeyJlc3NlbnRpYWwiOiB0cnVlfSwNCiAgICAgI...

Host: https://op.spid.agid.gov.it
HTTP/1.1
```

**Example of JWT payload:**

```
{
    "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": {
          "https://attributes.spid.gov.it/name":null,
          "https://attributes.spid.gov.it/familyName":null
      }
    },
    "state": "fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd"
}
```

### 1.28.18 EN 7. Metadata Policy

The following example shows a Metadata policy in the Entity Statement provided by a TA and related to an RP

```
"metadata_policy": {
    "openid_relying_party": {
        "jwks": {
          "keys": [{
            "subset_of": [{
                "kty": "RSA",
                "use": "sig",
                "n": "...",
                "e": "AQAB",
                "kid": "5NNNoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
            }]
          }]
        },
        "grant_types": {
            "subset_of": ["authorization_code", "refresh_token"]
        },
        "id_token_signed_response_alg": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "id_token_encrypted_response_alg": {
            "one_of": ["RSA-OAEP", "RSA-OAEP-256", "ECDH-ES", "ECDH-ES+A128KW", "ECDH-
↪ES+A256KW"]
        },
        "id_token_encrypted_response_enc": {
            "one_of": ["A128CBC-HS256", "A256CBC-HS512"]
        },
        "userinfo_signed_response_alg": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "userinfo_encrypted_response_alg": {
            "one_of": ["RSA-OAEP", "RSA-OAEP-256", "ECDH-ES", "ECDH-ES+A128KW", "ECDH-
↪ES+A256KW"]
        },
        "userinfo_encrypted_response_enc": {
            "one_of": ["A128CBC-HS256", "A256CBC-HS512"]
        },
        "token_endpoint_auth_method": {
            "one_of": ["private_key_jwt"]
        },
        "client_registration_types": {
            "one_of": ["automatic"]
        }
    }
}
```

The following example shows a Metadata policy in the Entity Statement provided by a TA and related to an SA

```
"metadata_policy": {
    "openid_relying_party": {
        "grant_types": {
            "subset_of": ["authorization_code", "refresh_token"]
        }
        "id_token_signed_response_alg": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "id_token_encrypted_response_alg": {
            "one_of": ["RSA-OAEP", "RSA-OAEP-256", "ECDH-ES", "ECDH-ES+A128KW", "ECDH-
↪ES+A256KW"]
```

(continues on next page)

(continua dalla pagina precedente)

```
        },
        "id_token_encrypted_response_enc": {
            "one_of": ["A128CBC-HS256", "A256CBC-HS512"]
        },
        "userinfo_signed_response_alg": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "userinfo_encrypted_response_alg": {
            "one_of": ["RSA-OAEP", "RSA-OAEP-256", "ECDH-ES", "ECDH-ES+A128KW", "ECDH-
↪ES+A256KW"]
        },
        "userinfo_encrypted_response_enc": {
            "one_of": ["A128CBC-HS256", "A256CBC-HS512"]
        },
        "token_endpoint_auth_method": {
            "one_of": ["private_key_jwt"]
        },
        "client_registration_types": {
            "one_of": ["automatic"]
        }
    }
}
```

The following example shows a Metadata policy in the Entity Statement provided by a SA and related to an RP

```
"metadata_policy": {
    "openid_relying_party": {
        "jwks": {
            "subset_of": [{
                "kty": "RSA",
                "use": "sig",
                "n": "...",
                "e": "AQAB",
                "kid": "5NNNoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
            }]
        }
    }
}
```

The following example shows a Metadata policy in the Entity Statement provided by a TA and related to an OP.

```
"metadata_policy": {
    "openid_relying_party": {
        "jwks": {
            "subset_of": [{
                "kty": "RSA",
                "use": "sig",
                "n": "...",
                "e": "AQAB",
                "kid": "5NNNoFS3YnC9tjiCaivhWLVUJ3AxwGGz_98uRFaqMEEs"
            }]
        },
        "revocation_endpoint_auth_methods_supported": {
            "one_of": ["private_key_jwt"]
        },
        "code_challenge_methods_supported": {
            "subset_of": ["authorization_code", "refresh_token"]
```

(continues on next page)

(continua dalla pagina precedente)

```
        },
        "scopes_supported": {
            "subset_of": ["openid", "offline_access", "profile", "email"]
        },
        "response_types_supported": {
            "one_of": ["code"]
        },
        "response_modes_supported": {
            "subset_of": ["form_post", "query"]
        },
        "grant_types_supported": {
            "subset_of": ["authorization_code", "refresh_token"]
        },
        "acr_values_supported": {
            "subset_of": ["https://www.spid.gov.it/SpidL1", "https://www.spid.gov.it/
↪SpidL2", "https://www.spid.gov.it/SpidL3"]
        },
        "subject_types_supported": {
            "one_of": ["pairwise"]
        },
        "id_token_signing_alg_values_supported": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "id_token_encryption_alg_values_supported": {
            "one_of": ["RSA-OAEP", "RSA-OAEP-256", "ECDH-ES", "ECDH-ES+A128KW", "ECDH-
↪ES+A256KW"]
        },
        "id_token_encryption_enc_values_supported": {
            "one_of": ["A128CBC-HS256", "A256CBC-HS512"]
        },
        "userinfo_signing_alg_values_supported": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "userinfo_encryption_alg_values_supported": {
            "one_of": ["RSA-OAEP", "RSA-OAEP-256", "ECDH-ES", "ECDH-ES+A128KW", "ECDH-
↪ES+A256KW"]
        },
        "userinfo_encryption_enc_values_supported": {
            "one_of": ["A128CBC-HS256", "A256CBC-HS512"]
        },
        "token_endpoint_auth_methods_supported": {
            "one_of": ["private_key_jwt"]
        },
        "token_endpoint_auth_signing_alg_values_supported": {
            "one_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        },
        "claims_parameter_supported": {
            "one_of": ["true"]
        },
        "request_parameter_supported": {
            "one_of": ["true"]
        },
        "authorization_response_iss_parameter_supported": {
            "one_of": ["true"]
        },
        "client_registration_types_supported": {
            "one_of": ["automatic"]
```

(continues on next page)

(continua dalla pagina precedente)

```
        },
        "request_authentication_methods_supported": {
            "one_of": ["request_object"]
        },
        "request_authentication_signing_alg_values_supported": {
            "subset_of": ["RS256", "RS512", "ES256", "ES512", "PS256", "PS512"]
        }
    }
}
```

## 1.29 Diventa fornitore di servizi

Qui di seguito riportiamo gli indirizzi di riferimento per le procedure di "onboarding" di SPID e CIE, cioè per diventare fornitori di servizi.

- Come diventare fornitori di servizi SPID[260]
- Come diventare fornitori di servizi CIE[261]

## 1.30 Come contribuire

Per contribuire clicca in alto a destra sulla icona di GitHub, alla voce "Sorgente" e accedi al repository pubblico.

Se trovi una inesattezza o desideri risolvere un dubbio o semplicemente notificare qualcosa per migliorare questa documentazione, apri una nuova Issue.

A seguito dell'apertura della Issue e dei riscontri ottenuti dalla comunità di Developers italia potrai aprire una nuova Pull Request contenente la modifica o la correzione da te proposta.

[260] https://www.spid.gov.it/cos-e-spid/diventa-fornitore-di-servizi/
[261] https://www.cartaidentita.interno.gov.it/esercenti/come-attivare-entra-con-cie/

# Indice

## R


RFC
    RFC 2119, 3, 8
    RFC 2616, 3, 47
    RFC 3339, 3
    RFC 3986, 3, 22
    RFC 6749#section-4.1.2.1, 37
    RFC 6749#section-5.2, 46
    RFC 6750, 47
    RFC 7009, 3
    RFC 7159, 3
    RFC 7515, 3, 7, 17
    RFC 7515#section-2, 8
    RFC 7516, 3
    RFC 7516#section-4.1.1, 32
    RFC 7517, 3, 11
    RFC 7518, 3
    RFC 7519, 4, 8, 11, 20, 33, 40, 42–44
    RFC 7519#section-5.2, 42, 47
    RFC 7523, 4
    RFC 7591, 4
    RFC 7636, 4, 31
    RFC 7636#section-4.2, 32
    RFC 7636#section-4.3, 26
    RFC 7638, 4
    RFC 7638#section_3, 32
    RFC 7662, 4
    RFC 7800, 4
    RFC 8174, 4, 8
    RFC 8414, 4
    RFC 8414#page-4, 26, 30
    RFC 8414#page-6, 30
    RFC 8623, 30
    RFC 8725, 4
    RFC 9068, 4, 42, 59