Linee Guida Attribute Authority – Allegato tecnico OAS3

Specifiche tecniche per la realizzazione dell'architettura Attribute Authority in SPID

Sommario

History of Changes

DATE	AUTORE	VERSIONE	MODIFICHE
31/05/2021	AgID	1.0	PRIMA VERSIONE
02/03/2022	AgID	1.1	REVISIONE
14/03/2022	AgID	1.2	REVISIONE FLUSSO PROTECTED
09/07/2022	AGID	1.3	REVISIONE DOCUMENTO E INSERIMENTO DEL DEMONSTRATING PROOF-OF-POSSESSION (DPOP)

Introduzione

Questo documento definisce i profili implementativi SAML2 e OpenID Connect dei Gestori degli Attributi Qualificati. I soggetti che intendono operare come Gestori degli Attributi Qualificati devono predisporre l'accesso ad una o più interfacce API RESTful. L'interfaccia API deve essere implementata in accordo con le specifiche riportate nelle presenti linee guida e in conformità con il MODI[1].

Termini normativi

Conformemente alle norme ISO/IEC Directives, Part 3 per la stesura dei documenti tecnici la presente linea guida utilizzerà le parole chiave «DEVE», «DEVONO», «NON DEVE», «NON DEVONO», «E' RICHIESTO», «DOVREBBE», «NON DOVREBBE», «RACCOMANDATO», «NON RACCOMANDATO» «PUÒ"» e «OPZIONALE», la cui interpretazione è descritta di seguito.

1. DEVE o DEVONO, indicano un requisito obbligatorio per rispettare la linea guida;
2. NON DEVE o NON DEVONO, indicano un assoluto divieto delle specifiche;
3. DOVREBBE o RACCOMANDATO o NON DOVREBBE o NON RACCOMANDATO, indicano che le implicazioni devono essere comprese e attentamente pesate prima di scegliere approcci alternativi;
4. PUÓ o POSSONO o l'aggettivo OPZIONALE, indica che il lettore può scegliere di applicare o meno senza alcun tipo di implicazione la specifica.

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

Termini e acronimi

I termini utilizzati per descrivere il contenuto dei documenti OpenAPI, quali Operation, Path, Tag, ... sono definiti nelle relative specifiche referenziate nel MODI[1].
Questo documento utilizza le specifiche definite in "Linee Guida OpenID Connect in SPID" e successivi Avvisi.
Di seguito si riportano gli acronimi e le abbreviazioni utilizzati nel presente Documento:

Termine	**significato**
PA	Pubblica Amministrazione.
AA	Attribute Authority, provider di attributi qualificati.

[1] *Modello di Interoperabilità - Linee guida sul Modello di Interoperabilità pubblicate da AgID e relative regole tecniche:* https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_interoperabilit_tecnica_pa.pdf

RS	Resource Server [RFC6749].
OAS3	OpenAPI Specification versione 3.0.
Metadata	Documento che descrive una implementazione di una entità OpenID Connect o SAML2. Le implementazioni di ogni Entità condividono i metadati per stabilire una base di fiducia e interoperabilità.
IDP	Identity Provider SAML2.
OIDC	OpenID Connect.
LG SPID OIDC	Linee Guida OpenID Connect in SPID.
OP	Identity Provider OpenID Connect.
SP	Service Provider SAML2.
Riautenticazione	Autenticazione dell'utente presso la AA, successiva alla prima già avvenuta presso il SP.
Assenso	Approvazione da parte dell'utente al rilascio dei propri dati ad una Entità riconoscibile all'interno della Federazione SPID.
RP	Relying Party OpenID Connect.
Entità	IDP/OP, SP/RP, AA. Indica generalmente un qualsiasi sistema riconoscibile all'interno di una Federazione delle Identità Digitali.
Onboarding	Procedura di registrazione di una nuova entità all'interno della Federazione.
JSON	JavaScript Object Notation [RFC8259].
JOSE	JavaScript Object Signing and Encryption [JOSE].
JWT	[RFC7519].
Nested JWT	[RFC7519#appendix-A.2].
JWE	JWT Criptato [RFC7516].
JWS	JWT Firmato [RFC7515].
OAuth2	Authorization Framework [RFC6749].
AS	Authorization Server OAuth2 [RFC6749].

Grant Token	Token di concessione per il rilascio delle autorizzazioni sulle risorse delle AA.
Federazione SPID	Implementazione SPID di OIDC Federation 1.0 o SAML2.
Trust Mark	Marchio di conformità secondo le specifiche di OIDC Federation 1.0.
Entity Configuration	Metadata di Federazione come definito in OIDC Federation 1.0.
SA	Soggetto Aggregatore.
Authorization Grant	[RFC6749].
Token Introspection endpoint	[RFC6749].

1. Architettura delle Attribute Authority

Le modalità di fruizione degli attributi sono classificate con i profili **public**, **protected** e **private**. I profili autorizzativi protected e private richiedono l'acquisizione dell'assenso da parte dell'utente al rilascio dei propri dati alle AA. Il profilo public non richiede alcun livello autorizzativo e consente l'utilizzo dei dati delle AA al pubblico.

2 Interfaccia API RESTful

L'interfaccia API che permette la comunicazione tra SP e AA deve essere implementata rispettando i criteri di progettazione definiti dal paradigma REST[2]. In particolare le operazioni consentite devono essere stateless, devono essere mappate su metodi HTTP in funzione della loro tipologia e devono essere orientate alla risorsa. Il dialogo DEVE utilizzare esclusivamente il protocollo HTTPS in conformità con le Raccomandazioni AGID - TLS e Cipher Suite [3]e tutte le indicazioni di sicurezza indicate nelle linee guida di riferimento. Poiché le minacce informatiche si evolvono rapidamente si raccomanda di seguire tutte le buone pratiche relative alle tecnologie di riferimento quali RFC8725 e successive modificazioni.

Per maggiori indicazioni in merito alle caratteristiche dello stile architetturale REST[3], si rimanda al MODI[1].

2.1 Estensioni OpenAPI

Il *documento OpenAPI* DEVE contenere l'elemento *"#/info/x-spid"* con i seguenti campi:

[2] *Fielding, Roy Thomas (2000). "Representational State Transfer (REST)". Architectural Styles and the Design of Network-based Software Architectures (PhD). University of California, Irvine*

[3] *Raccomandazioni Agid TLS e Cipher Suite (https://www.agid.gov.it/it/sicurezza/tls-e-cipher-suite)*

Campo	Tipo	Descrizione
aa-version	string	OBBLIGATORIO. Deve contenere il valore **1.0.0** quale riferimento alla versione delle linee guida SPID per le AA cui l'API fa riferimento. Tale valore non è da confondere con il valore nel campo openapi che specifica la versione OpenAPI, né con il valore nel campo version dell'oggetto info, che invece specifica la versione del documento.
aa-home	string (url)	OBBLIGATORIO. URI presso l'AA dove è esposto il presente *documento OpenAPI*
aa-registry	string (url)	OBBLIGATORIO. URI presso il registro SPID delle AA dove è esposto il presente *documento OpenAPI*
aa-required-attributes	array	OBBLIGATORIO. Array di stringhe corrispondenti agli attributi SPID necessari alla AA per le proprie finalità. Es.: ['fiscalNumber', 'email']
aa-lookup-attribute	string	OBBLIGATORIO. Stringa corrispondente all'attributo SPID utilizzato dalla AA come chiave di ricerca per l'utente. DEVE essere presente tra gli attributi indicati in in aa-required-attributes.

Tabella 1: OpenAPI - estensione x-spid

Viene di seguito riportato un esempio di documento OpenAPI con la specifica dell'elemento

```
openapi: 3.0.2
info:
 title: Esempio di Attribute Authority
 version: 1.1.0
 description: |-
   Descrizione delle informazioni restituite dall'Attribute Authority.
 termsOfService: https://api.aa.it/tos
 contact:
   name: Contatto di riferimento
   url: https://www.aa.it
 x-spid:
   aa-home: https://api.aa.it/v1/openapi.json
   aa-lookup-attribute: fiscalNumber
   aa-registry: https://registry.spid.gov.it/metadata/aa/ente.json
   aa-required-attributes:
   - fiscalNumber
   - email
   aa-version: 1.0.0
servers:
- url: https://api.aa.it/api/v1
   ...
```

Esempio 1: OpenAPI - estensione x-spid

Viene di seguito riportato un esempio di operation con un path associato, che mostra un attributo qualificato.

```
paths:
 /qualifica:
   get:
```

```
operationId: mostra_qualifica
summary: Mostra un attributo qualificato.
# Autenticazione di tipo UserConsent
security:
 - UserConsent: [ 'read:qualifica']
   ..
```

Esempio 2: Risorsa protetta da autorizzazione con scope read-qualifica.

2.2 Tipi di dato

L'AA descrive nel proprio *documento OpenAPI* il *tipo* di dato di ogni parametro accettato in input e di ogni dato restituito in output in conformità con quanto indicato nel MODI[1] e specificando, ove possibile, gli ulteriori vincoli di formato e sintassi associati ai campi. Le AA possono utilizzare il campo *externalDocs* per inserire il riferimento specifico ad un vocabolario controllato o ad una ontologia al fine di definire rispettivamente la sintassi o la semantica propria del dato.

2.4 Operazioni consentite

L' API DEVE esporre esclusivamente endpoint che realizzano operazioni di interrogazione verso l'AA al fine di ottenere gli attributi qualificati dell'utente o i riferimenti degli utenti che soddisfano i criteri di ricerca specificati (esclusivamente nei casi in cui non è previsto l'assenso). I formati dei nomi utilizzati per gli endpoint e i verbi HTTP sui quali sono esposte le operazioni DEVONO essere conformi agli standard HTTP e alle indicazioni contenute nel MODI[1].

2.5 Versioning

La specifica OpenAPI DEVE contenere nel campo #/info/version l'indicazione della versione secondo il formato: MAJOR.MINOR conformemente alle indicazioni del MODI[1].
I numeri MAJOR e MINOR sono da intendere secondo l'accezione *SemVer*, quindi le eventuali modifiche alle API devono permettere evoluzioni compatibili nel tempo.

```
openapi: 3.0.2
info:
 title: Esempio di Attribute Authority
 version: 1.4
```

Esempio 3: OpenAPI - versioning

Gli endpoint DEVONO supportare il versionamento sull'URL. Sarà possibile quindi effettuare una interrogazione verso una specifica versione dell'API indicando l'URL corrispondente, che DEVE essere formata nel seguente modo:

*https://**<HOST>**/api/v**<MAJOR>[.<MINOR>]/<operation>***

Nota: <HOST> è il dominio dell'AA
<MAJOR> è il numero MAJOR corrispondente alla versione dell'API

<MINOR> è il numero MINOR corrispondente alla versione dell'API
<operation> è il path della chiamata API

I numeri indicati tra parentesi quadre non sono obbligatori e DOVREBBERO essere omessi: questo perché l'API DOVREBBE essere sviluppata in modo che le modifiche siano retrocompatibili e che i client sviluppati per le versioni precedenti siano comunque in grado di funzionare correttamente con le nuove versioni. In tal caso verrà indirizzata la versione corrispondente al numero MINOR, rispettivamente, più alto disponibile. Sono di seguito riportati alcuni esempi di URI per versioni differenti della stessa chiamata API.

```
https://aa.it/api/v1/qualifica
https://aa.it/api/v1.1/qualifica
```

Esempio 4: OpenAPI - versioning URL

Per realizzare il versionamento sull'URL, l'attributo *url* di ognuno degli elementi presenti all'interno della sezione *"#/servers"* del *documento Open API*, DEVE includere il numero di versione coerentemente con quanto indicato nell'elemento *"#/info/version"*.

```
servers:
  - url: https://aa.it/v1
    description: Server di produzione
  - url: https://test.aa.it/v1
  - description: Server di test che eroga dati fittizi e statici.
```

Esempio 5: OpenAPI - versioning e servers

2.6 Documentazione API

L'AA deve esporre la documentazione API generata automaticamente sulla base del *documento OpenAPI* su un URL pubblica afferente al dominio della stessa AA espresso nella seguente forma, coerentemente con quanto già indicato nel § 2.5.

*https://**<HOST>**/api/v**<MAJOR>[.<MINOR>]***

2.7 Raggruppamenti per tipologie di accesso ai dati

L'accesso ai dati, per ogni operazione, può essere classificato sulla base delle seguenti casistiche:

a. **public**: il dato è di pubblico dominio o liberamente accessibile. In tal caso l'accesso al dato non richiede l'acquisizione dell'assenso dell'utente da parte dell'AA;
b. **private**: l'accesso al dato è consentito solo previa acquisizione dell'assenso dell'utente da parte dell'AA, mediante una nuova autenticazione dell'utente presso di questa. Tale casistica si applica sempre nei casi di richiesta continuativa di attributi qualificati.
c. **protected**: Non è richiesta la riautenticazione dell'utente presso la AA, l'assenso viene acquisito secondo le modalità descritte nel seguito per il profilo **protected**.
   Per descrivere in maniera chiara le operazioni soggette all'acquisizione dell'assenso, il campo *"#/tags"* DOVREBBE contenere almeno i seguenti *Tag* utili a raggruppare le relative operazioni:

name	description
Public	Accesso pubblico
Protected	Accesso riservato a SP convenzionati. previa acquisizione dell'assenso dell'utente da parte dell'IdP per le AA
Private	Accesso previa acquisizione dell'assenso dell'utente da parte dell'AA

Tabella 2: OpenAPI - tags per tipologie di accesso ai dati

Detti tag, qualora siano presenti nel *documento OpenAPI*, conterranno il campo externalDocs con le seguenti informazioni:

description	url
Linee Guida Attribute Authority SPID	*Il riferimento presso il sito AgID dove sono pubblicate le Linee Guida Attribute Authority SPID*

Tabella 3: OpenAPI - externalDocs per tags predefiniti:

A solo titolo esemplificativo, viene di seguito riportata la parte del *documento OpenAPI* che definisce i suddetti Tag.

```
tags:
- description: Accesso pubblico
 externalDocs:
  description: Linee Guida Attribute Authority SPID
  url: https://www.agid.gov.it/...
 name: public
- description: Accesso riservato a SP convenzionati
 externalDocs:
  description: Linee Guida Attribute Authority SPID
  url: https://www.agid.gov.it/...
 name: protected
- description: |-
  Accesso previa acquisizione dell'assenso dell'utente da
  parte dell'AA
 externalDocs:
```

```
  description: Linee Guida Attribute Authority SPID
  url: https://www.agid.gov.it/...
name: private
```

Esempio 6: OpenAPI - tags per tipologie di accesso al dato

L'AA può indicare ulteriori raggruppamenti di operazioni tramite "*Tag*" per descrivere, ad esempio, quali operazioni sono consentite a determinate pubbliche amministrazioni o SP senza la necessità di acquisizione dell'assenso sulla base di una norma giuridica, regolamento o convenzione. In tal caso devono essere considerate le seguenti indicazioni:

- il valore del campo *name* del Tag deve riportare come prefisso *"protected-" seguito* dal nome identificativo del Tag, senza spazi;
- nell'elemento *externalDocs* devono essere specificati i campi *description* e *url* in riferimento alla specifica norma, regolamento o convenzione che giustifica la politica di accesso.

```
tags:
...
- name: protected-convenzione-caf
 description: Accesso pubblico per CAF convenzionati
 externalDocs:
  description: Convenzione che regola l'accesso ai dati per i CAF
  url: "https://... "
```

Esempio 7: OpenAPI - tags per ulteriori raggruppamenti

2.8 Documentazione operazioni

Il campo *summary* di ogni oggetto *Operation* DEVE contenere una breve descrizione degli attributi qualificati che l'operazione permette di recuperare.

Il campo *description* di ogni oggetto *Operation* PUÒ indicare le modalità richieste per l'accesso al dato ed in particolare:

- se l'attributo richiede la raccolta esplicita dell'assenso dell'utente o meno;
- se è richiesta un'autenticazione SPID di un livello minimo;
- se sono consentite richieste continuative e la durata massima consentita.

È prevista per l'oggetto *Operation*, l'estensione *x-spid-operation*, con i campi indicati nella seguente tabella, per descrivere in maniera strutturata le caratteristiche elencate sopra.

campo	tipo	descrizione
consentRequired	boolean	Indica se la risorsa richiede l'assenso dell'utente per accedere alle sue informazioni. DEVE essere **True** or **False**
spidLevel	string	ACR del livello SPID minimo richiesto

offlineAccessExpiresIn	integer	se > 0 sono consentite richieste continuative, se 0 non sono consentire richieste continuative. num. max di secondi consentiti per richieste continuative. Esempio: 10 giorni = 864000.

Tabella 4: OpenAPI - estensione x-spid-operation

2.9 Autenticazione API

Gli endpoint delle API per la richiesta di attributi qualificati, ad eccezione degli endpoint ad accesso pubblico (Tipologia di accesso "public"[4]), devono essere protetti tramite autenticazione.

Tutti i JWT generati nelle fasi di richiesta o risposta contengono almeno gli attributi di seguito indicati.

Nell'header, DEVONO essere presenti i seguenti parametri:

Parametro		Descrizione
typ	Richiesto	Tipologia del Token, come normato dalle LG SPID OIDC e successivi avvisi, se non specializzata assume il valore di "JWT".
alg	Richiesto	Come normato dalle LG SPID OIDC e successivi avvisi.
kid	Richiesto	Come normato dalle LG SPID OIDC e successivi avvisi.

Tabella 5: claim della intestazione del JWT

Nell'header, PUÓ essere presente il seguente parametro:

Parametro		Descrizione
x5c	Opzionale	valorizzato con il certificato o la catena dei certificati, in formato X.509. Il certificato o la catena dei certificati sono codificati come array JSON di stringhe corrispondenti ai valori dei certificati in formato DER. Ogni stringa nell'array è codificata in Base64. Il certificato contenente la chiave pubblica utilizzata per firmare il token deve essere la prima stringa dell'array.

Tabella 6: claim della intestazione del JWT

Nel payload in formato JWT, DEVONO essere presenti almeno i claim:

Claim	Descrizione
sub	identificativo univoco del soggetto titolare del token.
iss	identificativo univoco del soggetto che ha emesso il token.
aud	identificativi univoci dei soggetti ai quali il token è destinato (audience).
iat	Data/ora di emissione del token in formato UNIX Timestamp
exp	Data/ora di scadenza del token in formato UNIX Timestamp.

[4] *Vedi paragrafo 2.7 Raggruppamenti per tipologie di accesso ai dati*

jti	Identificatore unico del token che è possibile utilizzare per prevenirne il riuso illecito. Deve essere di difficile individuazione da parte di un attaccante e composto da una stringa casuale.

Tabella 7: claim del payload del JWT

Lo schema di autenticazione predefinito, valido per tutti gli endpoint, deve essere dichiarato nell'elemento *"#/components/securitySchemes"* e referenziato nell'elemento *security* dell'*Operation*. A titolo esemplificativo, viene di seguito riportata la parte del *documento OpenAPI* che definisce lo schema di autenticazione predefinito.

```
openapi: 3.0.2
components:
 securitySchemes:
   DefaultSecurity:
    type: http
    scheme: Bearer
    bearerFormat: JWT
    description: |-
     Autenticazione basata su un JWT. L'implementazione
     DEVE essere conforme al MODI e alle indicazioni di sicurezza
     contenute in RFC8725 e s.s.m.
security:
     ● DefaultSecurityScheme: []
paths:
 /attributo-a:
   get:
    .. # Questo usa DefaultSecurity perché non il campo security non è definito
 /codice-fiscale:
   get:
    security:
     ● AnotherSecurity: []  # Questo non usa DefaultSecurity
 ..
```

Esempio 8: OpenAPI – autenticazione

I vari tipi di profilo di accesso utilizzano differenti modalità di autenticazione e autorizzazione (profilo private e protected).

2.10 Response

Le Response alle richieste di attributi qualificati DEVONO usare uno o più dei seguenti *Media Type*:

- **application/json**, indicando lo schema dell'oggetto restituito;
- **application/jwt**, referenziando nella documentazione lo schema del payload del JSON Web Token utilizzato e che estende lo schema utilizzato nel caso application/json.

```
responses:
    "200":
     description: |-
      Attributo qualificato restituito con successo."
```

```
    content:
      application/json:
        schema:
          $ref: "#/components/schemas/Qualifica"
      application/jwt:
        schema:
          type: string
          pattern: ^[a-zA-Z\_\-0-9]+\.[a-zA-Z\_\-0-9]+\.[a-zA-Z\_\-0-9]+$
          description: |-
            Un oggetto il cui payload è definito in
            #/components/schemas/QualificaJWTPayload.
            Indicare se la codifica è JWE o JWS.
```

Esempio 9: OpenAPI - response MediaType

2.11 Errori

Se la richiesta di attributi qualificati non può essere soddisfatta, l'AA DEVE restituire una Response, secondo le indicazioni riportate nelle Response, conforme al MODI[1] e basate sull' RFC7807[5]:

Proprietà		Descrizione
status	Richiesto	HTTP status code
type	Opzionale	URI che rimanda al codice specifico dell'errore, definito dall'AA
title	Richiesto	Descrizione standard dell'errore
detail	Opzionale	Messaggio specifico dell'errore

Tabella 8: OpenAPI - errori Response

Gli elementi type e detail possono essere utilizzati dall'AA per specificare ulteriormente l'errore al fine di evidenziare casistiche di propria pertinenza. Nel caso in cui non esista una ulteriore specificazione essi potranno essere omessi. Sono di seguito riportati alcuni esempi di risposta di errore.

```
HTTP/1.1 401 Unauthorized
Content-Type: application/problem+json

{
  "type": "https://aa.it/error/auth-failed",
  "status": 401,
  "title": "Autenticazione fallita"
}
```

Esempio 10: OpenAPI - errore di Autenticazione fallita

```
HTTP/1.1 500 Internal Server Error
```

[5] *RFC7807 - Problem Details for HTTP APIs (https://tools.ietf.org/rfc/rfc7807.txt)*

```
Content-Type: application/problem+json
Retry-After: 300

{
  "status": 500,
  "title": "Internal Server Error",
  "detail": "Riprova più tardi"
}
```

Esempio 11: OpenAPI - errore 500 Personalizzato

3 Richiesta di attributi (Attribute Request)

Sono di seguito presentati i flussi che descrivono i possibili scenari relativi alla richiesta di attributi.

3.1 Richiesta senza necessità dell'assenso - "public"

Viene di seguito descritto il flusso valido nel caso in cui non sia necessaria l'acquisizione dell'assenso dell'utente alla trasmissione dei dati. Questo flusso rientra nella casistica standard del modello di interoperabilità.

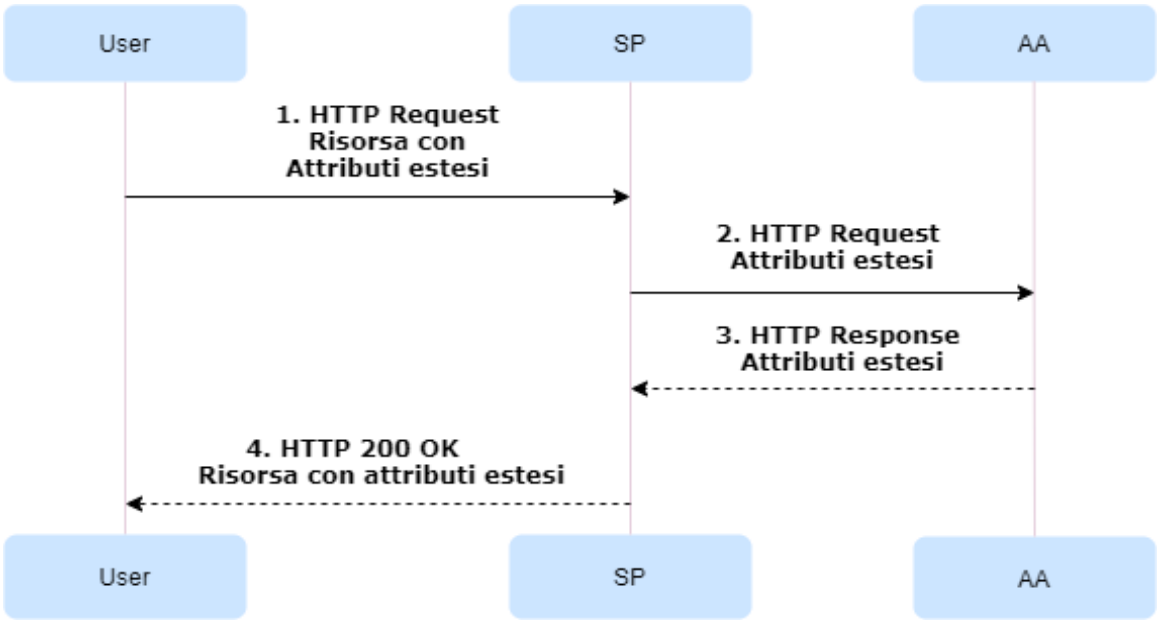

Figura 1: Flusso - richiesta senza necessità dell'assenso per risorsa di tipo pubblico

1. L'utente chiede di utilizzare un servizio presso il SP (Service Request) per il quale è necessario conoscere uno o più attributi qualificati.
2. Il SP, avendo necessità di conoscere gli attributi qualificati per erogare il servizio esegue una chiamata API alla specifica AA che ha la facoltà di attestare e restituire tali attributi qualificati.

3. L'AA verifica l'integrità e l'autenticità della request (Attribute Request) e, dopo aver accertato la possibilità di rispondere alla richiesta senza che sia necessario ottenere l'assenso da parte dell'utente, restituisce al SP gli attributi richiesti.
4. Il SP verifica l'integrità e l'autenticità della response (Attribute Response) ottenuta dall'AA e utilizza gli attributi qualificati ricevuti per erogare il servizio all'utente (Service Response).

3.2 Richiesta con necessità dell'assenso - "protected"

Il dato di tipo Protected è riservato ai SP/RP (pubbliche amministrazioni o enti privati) che hanno una specifica convenzione con l'AA. Nel caso di RP OIDC, la convenzione è codificata tramite T*rust Mark* nell'E*ntity Configuration* del RP[6]. In SAML non è prevista la codifica della convenzione nel metadata SAML del SP.

Il flusso prevede le seguenti fasi:

1. Il SP seleziona le AA e le mostra all'utente, che conferma di voler procedere;
2. L'utente viene reindirizzato sull'IdP per l'autenticazione;
3. L'utente seleziona sull'IdP le AA per le quali dare l'assenso;
4. L'IdP genera la risposta di autenticazione e dei token autorizzativi firmati e cifrati denominati Grant Token (vedi § 3.2.3) per ognuna delle AA selezionate dall'utente;
5. Il SP chiede l'accesso alla AA utilizzando i Grant Token rilasciati dall'IdP;
6. Il SP ottiene il token dalla AA e lo usa per accedere agli attributi qualificati dell'utente.

Il SP che implementa il profilo **protected**:

1. DEVE individuare una lista di attributi qualificati con le relative AA (di seguito "lista di attributi");
2. DEVE mostrare tale lista all'utente in una pagina web informativa;
3. DEVE inoltrare l'utente presso l'IdP per l'autenticazione in modo da trasferire all'IdP la lista delle AA delle quali necessita;
4. DEVE usare il Grant token esclusivamente presso la AA per la quale è stato creato.

L'IdP che implementa il profilo **protected**:

1. DEVE acquisire dall'utente un singolo assenso per ogni AA;
2. DEVE trasmettere tale assenso al SP/RP in forma di Grant Token;
3. DEVE implementare un Token Introspection endpoint per permettere ai soggetti interessati di verificare i Grant Token emessi.

3.2.1 Obiettivi

L'implementazione del flusso Protected proposta e descritta nel presente documento intende raggiungere i seguenti obiettivi:

- L'utente deve poter esprimere l'assenso all'invio degli attributi verso la specifica AA.
- L'AA deve poter verificare che l'utente sia stato correttamente autenticato con il livello richiesto dall'AA e che tale autenticazione sia valida al momento della richiesta di attributo.

[6] *OpenID Connect Federation 1.0 - draft 18*

- Evitare all'utente di doversi riautenticare.
- Evitare l'invio all'AA di attributi identificativi eccedenti le finalità proprie dell'AA.
- Evitare l'invio al SP di attributi identificativi eccedenti le finalità proprie del SP.
- Garantire l'applicabilità del flusso sia nel protocollo SAML2 che nel protocollo OIDC.

3.2.2 Flusso Protected

Il SP mostra all'utente, prima di inoltrare l'utente presso l'IdP per la prima autenticazione, una informativa nella quale espone le AA che potrebbe interrogare e i relativi attributi qualificati di cui potrebbe aver bisogno per l'erogazione dei servizi.
Durante la prima autenticazione, l'IdP mostra all'utente la lista delle AA, dichiarate dallo stesso SP con la richiesta di autenticazione e contenenti quali AA il SP potrebbe interrogare ai fini della fruizione del servizio. L'assenso concesso dall'utente alla trasmissione dei propri dati identificativi viene quindi raccolto dall'IdP e trasmesso al SP in forma di token firmato e cifrato (Grant Token) che può essere speso dal SP esclusivamente presso la specifica AA, al fine di ottenere una autorizzazione con la quale recuperare l'attributo qualificato presso la medesima AA.
Il flusso prevede quindi complessivamente tre fasi:

1. preselezione delle AA sull'IdP da parte dell'utente;
2. autorizzazione del SP sull'AA;
3. accesso alla risorsa protetta per ottenere l'attributo qualificato.

3.2.2.1 Preselezione delle AA su IdP

1. L'utente accede sul SP e, dopo aver letto l'informativa sull'utilizzo dei propri dati e delle AA, seleziona l'IdP dal bottone "*Entra con SPID*".
2. Il SP redireziona l'utente sull'IdP con una richiesta di autenticazione contenente la lista delle AA.
3. L'IdP autentica l'utente sulla base del livello di autenticazione richiesto.
4. L'IdP mostra all'utente la lista delle AA che il SP ha dichiarato di poter interrogare e che risultano riconoscibili e valide. Per ognuna di queste l'IdP DEVE mostrare all'utente le seguenti informazioni:
    a. **URL del logo** in formato SVG, come disponibile dal metadata della AA;
    b. **Nome**, valorizzato con il valore di *organization_name* disponibile dal metadata della AA;
    c. **URL della pagina web** ospitata dalla AA che descrive il servizio offerto, se disponibile dal metadata (tos_uri);
    d. **Casella di selezione**, per la sottomissione dell'assenso.
5. L'utente seleziona o deseleziona le AA per le quali vuole fornire l'assenso.
6. L'IdP chiede all'utente il consenso alla trasmissione dei propri dati al SP.
7. L'IdP genera un *Grant Token* per ognuna delle AA selezionate, questo contiene la prova dell'avvenuta autenticazione e i dati minimi dell'utente necessari alla AA per autorizzare il SP ad accedere agli attributi qualificati dell'utente.
8. L'IdP redireziona l'utente sul SP con la risposta di autenticazione. In relazione allo specifico protocollo utilizzato (SAML2 o OIDC) l'IdP trasmette al SP, nelle modalità opportunamente descritte nel seguito, l'insieme dei Grant Token generati per ognuna delle AA selezionate.

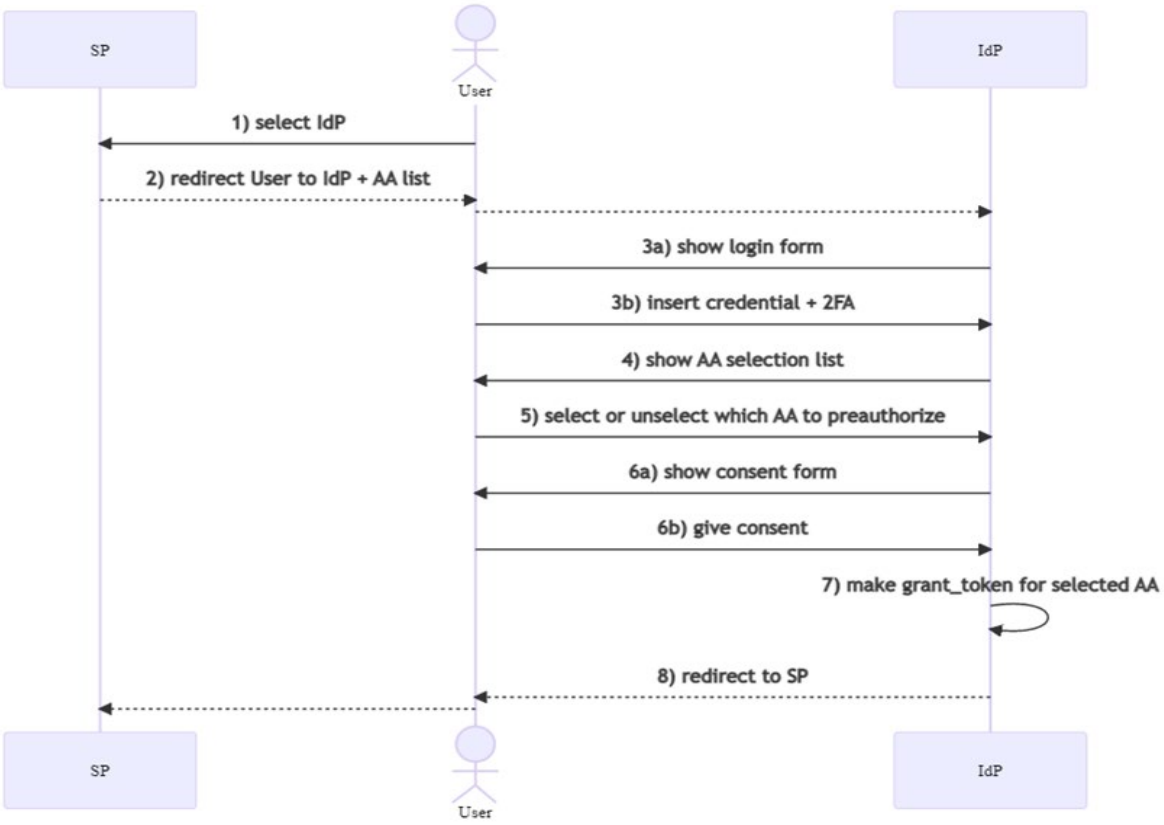

Figura 2: Sequence Diagram - Preselezione delle AA su IdP

3.2.2.2 Autorizzazione tramite Grant Token (Grant Token Exchange) e Attribute Request

Il SP usa il Grant Token ottenuto dall'IdP come *authorization grant* presso l'AS dell'AA. Il flusso di autorizzazione tramite Grant Token è il seguente:

1. Il SP invia all' AS dell'AA una richiesta di Token Exchange per scambiare il Grant Token con un Access Token.
2. L'AA decifra il Grant Token con la propria chiave privata.
3. L'AA verifica l'autenticità e l'integrità del Grant Token.
4. L'AA verifica l'associazione tra il Grant token e la risorsa richiesta. La AA verifica, in base alle informazioni contenute nel Grant Token, la validità della richiesta e del SP richiedente.
5. La AA PUÒ controllare lo stato del Grant Token usando l'I ntrospection endpoint dell'IdP.
6. L'AA crea un Access Token per il SP.
7. L'AA invia l'Access Token al SP.
8. Il SP utilizza l'Access Token per ottenere l'attributo qualificato.

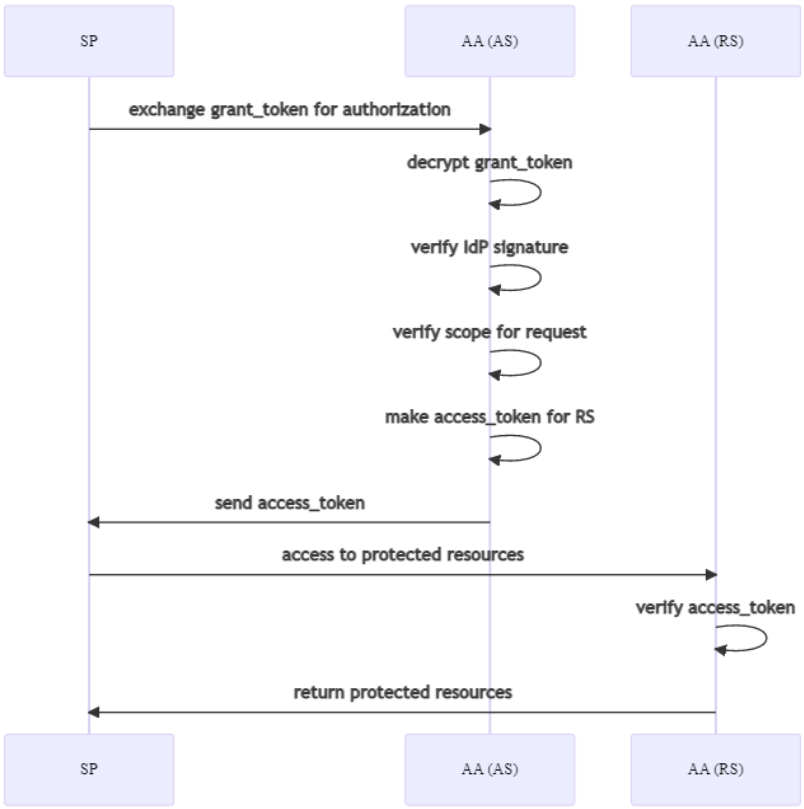

Figura 3: Sequence Diagram - Grant Token Exchange

3.2.3 Grant Token

Il Grant Token è un Nested JWT firmato con la chiave privata dell'IdP e cifrato con la chiave pubblica dell'AA a cui è destinato.

L'intestazione Jose del Grant Token specifica i seguenti claim:

Claim	Descrizione
typ	aa-grant+jwt
alg	[RFC7516#section-4.1.1] e LG SPID OIDC e successivi avvisi.
enc	[RFC7516#section-4.1.2] e LG SPID OIDC e successivi avvisi.
kid	LG SPID OIDC e successivi avvisi.

Tabella 9: intestazione Jose del Grant Token

Il payload del Grant Token DEVE contenere i seguenti claim:

Claim	Type	Description	Reference
iat	JSON Integer	OBBLIGATORIO. Timestamp di emissione del token, espresso in UNIX Timestamp.	RFC7519#section-4.1.6
exp	JSON Integer	OBBLIGATORIO. Timestamp di scadenza del token, espresso in UNIX Timestamp.	RFC7519#section-4.1.4
nbf	JSON Integer	OPZIONALE. Timestamp di inizio validità del token, espresso in UNIX Timestamp.	RFC7519#section-4.1.5
jti	JSON String	OPZIONALE. Identificativo univoco del token.	RFC7519#section-4.1.7
aud	JSON String	OBBLIGATORIO. DEVE corrispondere all'identificativo univoco della AA (URL).	RFC7519, Section 4.1.3
sid	JSON String	OBBLIGATORIO. ResponseID SAML2 (con prefisso "saml:") o jti dell'ID Token OIDC (con prefisso "oidc:"), per collegare il Grant Token all'autenticazione.	OpenID Connect Front-Channel Logout 1.0, Section 3
acr	JSON String	OBBLIGATORIO. Identifica il livello di autenticazione SPID.	Linee Guida OpenID Connect in SPID
iss	JSON String	OBBLIGATORIO. Identificativo dell'IdP che ha emesso il token.	RFC7519, Section 4.1.1
sub	JSON String	OBBLIGATORIO. Identificativo univoco dell'utente di tipo *public*, come definito in *OpenID Connect Core 1.0, Sezione 8*. il valore è da interpretare in relazione all'attributo di lookup (**aa-lookup-attribute**) dichiarato nel documento OpenAPI.	RFC7519, Section 4.1.2
act	JSON Object	OBBLIGATORIO. Oggetto JSON che contiene il claim **sub** valorizzato con l'identificativo del SP (entityID) / RP (client_id) al quale viene restituito il Grant Token. Esempio: "act": { "sub": "https://rp.spid.gov.it" }	RFC8693, Section 4.1

Tabella 10: claim del payload del Grant Token

Il Grant Token contiene inoltre, per la specifica AA, l'insieme degli attributi dell'utente necessari per l'erogazione del servizio, dichiarati nel documento OpenAPI (aa-required-attributes).

3.2.4 Preselezione delle AA - SAML

3.2.4.1 AuthnRequest

L'AuthnRequest SAML, inviata dal SP all'IdP, contiene l'Extension:

- **RequiredAttributeAuthority**: (una occorrenza) con namespace *https://spid.gov.it/saml-extensions*, contenente uno o più elementi:
  - **Location**: (obbligatorio, una o più occorrenze) avente come valore l'identificativo dell'Authorization Server dell'AA richiesta, corrispondente al valore del parametro *issuer* dichiarato nel metadata dell'Authorization Server.

```
<samlp:Extensions>
  <spid:RequiredAttributeAuthority
      xmlns:spid="https://spid.gov.it/saml-extensions">
    <Location>https://as.aa1.it</Location>
    <Location>https://as.aa2.it</Location>
    <Location>https://as.aa3.it</Location>
  </spid:RequiredAttributeAuthority>
</samlp:Extensions>
```

Esempio 12: AuthnRequest SAML - RequiredAttributeAuthority

3.2.4.2 Verifica dell'AuthnRequest

Alla ricezione della AuthnRequest l'IdP verifica l'eventuale presenza dell'Extension AttributeAuthority, questa deve essere opportunamente formata e contenere uno o più elementi Location. Per ogni elemento Location, l'IdP verifica l'esistenza del relativo valore all'interno della Federazione SPID e recupera il nome descrittivo dell'AA dal parametro *organization_name* del Metadata dell'Authorization Server presso l'AA. A tale scopo, PUÒ utilizzare il servizio di Registry API / Entity Listing reso disponibile dal Registro SPID. Se il valore di uno o più degli elementi *Location* non è valido all'interno della Federazione o se l'Extension non è presente o è malformata, l'OP DEVE procedere comunque all'autenticazione SPID, considerando esclusivamente le sole AA verificabili. Dopo aver autenticato l'utente, l'IdP gli mostra l'elenco delle AA consentendogli di selezionare o deselezionare quelle per le quali fornire l'assenso.

3.2.4.3 Response

La Response SAML contiene l'Extension:

- **GrantedAttributeAuthority**: (una occorrenza) con namespace *https://spid.gov.it/saml-extensions*, contenente uno o più elementi:
    - **GrantToken**: (obbligatorio, una o più occorrenze) avente l'attributo *Destination* valorizzato con l'identificativo della AA cui il token è destinato (corrispondente al claim *aud* del token) e come valore il Grant Token generato, firmato e cifrato.

```
<samlp:Extensions>
  <spid:GrantedAttributeAuthority
      xmlns:spid="https://spid.gov.it/saml-extensions">
    <GrantToken Destination="https://as.aa1.it">eyJhbGciOiJS...</GrantToken>
    <GrantToken Destination="https://as.aa2.it">eyJhbGciOiJS...</GrantToken >
    <GrantToken Destination="https://as.aa3.it">eyJhbGciOiJS...</GrantToken > </spid:GrantedAttributeAuthority
>
</samlp:Extensions>
```

Esempio 13: Response SAML - GrantedAttributeAuthority

La Response SAML2 DEVE contenere l'elemento Signature, valorizzato con la firma dell'intero messaggio SAML2 di Response, comprendente sia l'elemento Assertion, a sua volta firmato, sia l'elemento Extensions.

3.2.4.4 Verifica della Response

Alla ricezione della Response, dopo averne verificato la firma, il SP verifica l'eventuale presenza dell'Extension *GrantedAttributeAuthority* ed estrae i Grant Token contenuti nella Extension per le rispettive AA indicate nell'attributo *Destination*.

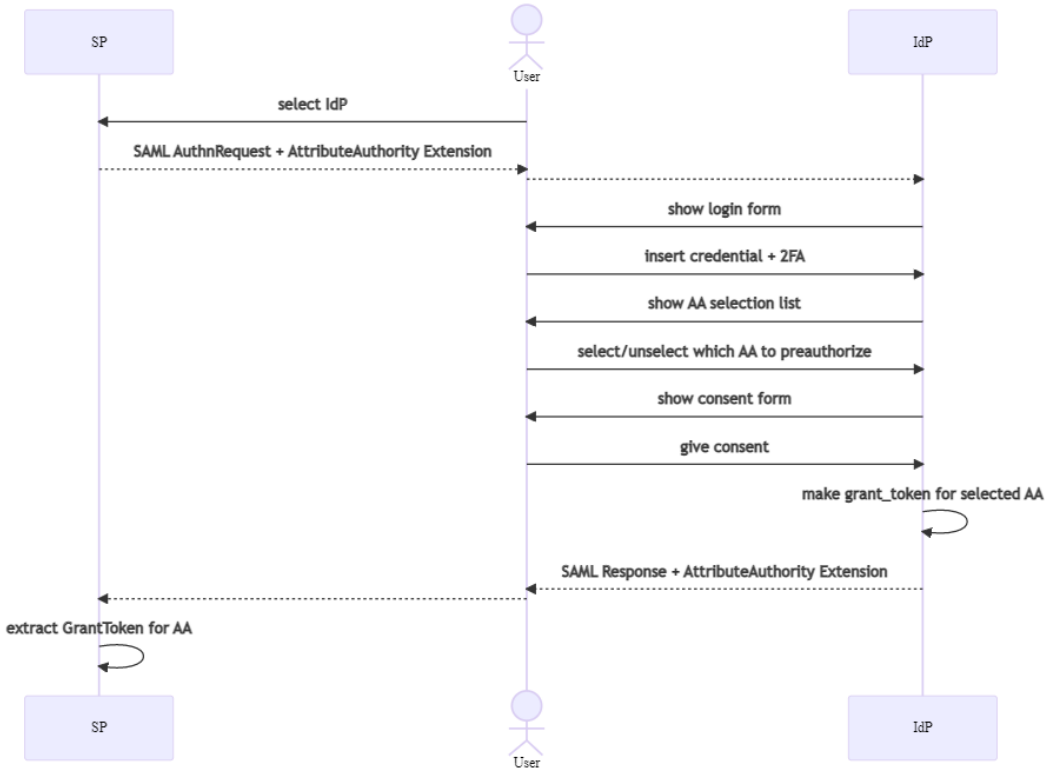

Figura 4: Sequence Diagram - Verifica della Response SAML

3.2.5 Preselezione delle AA - OIDC

3.2.5.1 Authentication Request

L'Authentication Request, inviata dal RP all'Authorization Endpoint dell'OP, è conforme alla specifica Rich Authentication Request (RAR)[7] e contiene nell'oggetto *request*, anche il parametro:

- **authorization_details**: array JSON di oggetti contenente almeno un oggetto costituito dai seguenti attributi:
  - **type**: (obbligatorio) identificativo univoco che permette di riconoscere il tipo di richiesta, deve essere valorizzato con "*https://spid.gov.it/attribute-authority/required-aa";*
  - **locations**: (obbligatorio) array JSON contenente lista degli AS delle AA richieste, corrispondenti al valore dell'attributo *issuer* definito nel Metadata dell'AA.

```
{
  "client_id": "https://rp.spid.agid.gov.it",
  "response_type": "code",
  "scope": "openid",
```

[7] *RAR – OAuth2 Rich Authentication Request (https://datatracker.ietf.org/doc/html/draft-ietf-oauth-rar)*

```
  "code_challenge": "qWJlMe0xdbXrKxTm72EpH659bUxAxw80",
  "code_challende_method": "S256",
  "nonce": "MBzGqyf9QytD28eupyWhSqMj78WNqpc2",
  "prompt": "login",
  "redirect_uri": "https://rp.spid.agid.gov.it/callback/",
  "acr_values": "https://spid.gov.it/SpidL2",
  "claims": {
    "userinfo": {
      "https://attributes.spid.gov.it/name": {"essential": true},
      "https://attributes.spid.gov.it/familyName": {"essential": true}
    }
  },
  "state": "fyZiOL9Lf2CeKuNT2JzxiLRDink0uPcd",
  "authorization_details": [
    {
      "type": "https://spid.gov.it/attribute-authority/required-aa",
      "locations": [
        "https://as.aa1.it",
        "https://as.aa2.it",
        "https://as.aa3.it"
      ]
    }
  ]
}
```


Esempio 14: Authentication Request OIDC – esempio non normativo di authorization_details

3.2.5.2 Verifica dell'Authentication Request

L'OP verifica se l'Authorization Request contiene il parametro **authorization_details** e se questo contiene un elemento di tipo *https://spid.gov.it/attribute-authority/required-aa*. Per ogni location indicata e corrispondente all' identificativo univoco di una AA (URL) l'OP DEVE:

1. recuperare l'Entity Configuration dell'AA dal relativo endpoint di */.well-known/openid-federation*, se non già fatto in precedenza e all'interno delle soglie di validità temporale delle Trust Chain;
2. deve verificare che l'AA sia un soggetto riconosciuto all'interno della Federazione SPID OIDC tramite:
    a. verifica del Trust Mark nell'Entity Configuration dell'AA, corrispondente all'id **https://spid.gov.it/attribute-authority**;
    b. costruzione della Trust Chain;
3. recuperare dal Trust Mark della AA la lista di attributi dell'utente da inserire (**claims**), successivamente, all'interno del Grant Token.

Nel caso in cui il Trust Mark sia rilasciato da un SA, l'OP DEVE, inoltre, verificare la Trust Chain relativa al SA per stabilire la fiducia tra tutte le parti coinvolte.

Se il valore di uno o più degli elementi *Location* non è valido all'interno della Federazione o se l'Extension non è presente o è malformata, l'OP DEVE procedere comunque all'autenticazione SPID, considerando esclusivamente le sole AA verificabili.

3.2.5.3 Token Response

L'ID Token restituito al RP con la Response del Token Endpoint contiene anche il seguente ulteriore parametro:

- **tokens**: array JSON di oggetti contenenti i seguenti parametri:
  - **type**: *https://spid.gov.it/attribute-authority/grant-token*
  - **aud**: l'identificativo della AA cui il token è destinato;
    - **token**: Grant Token generato, firmato e cifrato.

```
{
  iss: "https://op.spid.agid.gov.it/",
  sub: "OP-1234567890",
  aud: "https://rp.spid.agid.gov.it/auth",
  acr: "https://www.spid.gov.it/SpidL2",
  at_hash: "qiyh4XPJGsOZ2MEAyLkfWqeQ",
  iat: 1519032969,
  nbf: 1519032969,
  exp: 1519033149,
  jti: "nw4J0zMwRk4kRbQ53G7z",
  nonce: "MBzGqyf9QytD28eupyWhSqMj78WNqpc2",
  tokens: [
    {
      type: "https://spid.gov.it/attribute-authority/grant-token",
      aud: "https://as.aa1.it",
      token: "eyJhbGciOiJS ... "
    },
    {
      type: "https://spid.gov.it/attribute-authority/grant-token",
      aud: "https://as.aa2.it",
      token: "eyJhbGciOiJS ... "
    },
    {
      type: "https://spid.gov.it/attribute-authority/grant-token",
      aud: "https://as.aa3.it",
      token: "eyJhbGciOiJS ... "
    }
  ]
}
```

Esempio 15: Token Response OIDC – esempio non normativo di ID Token contenente nel parametro tokens i Grant Token.

3.2.5.4 Verifica della Token Response

Alla ricezione della Response dal Token Endpoint, il RP recupera l'ID Token, ne verifica la firma, ed estrae i Grant Token eventualmente contenuti nel parametro *tokens* per le ispettive AA indicate nel corrispondente parametro *aud*.

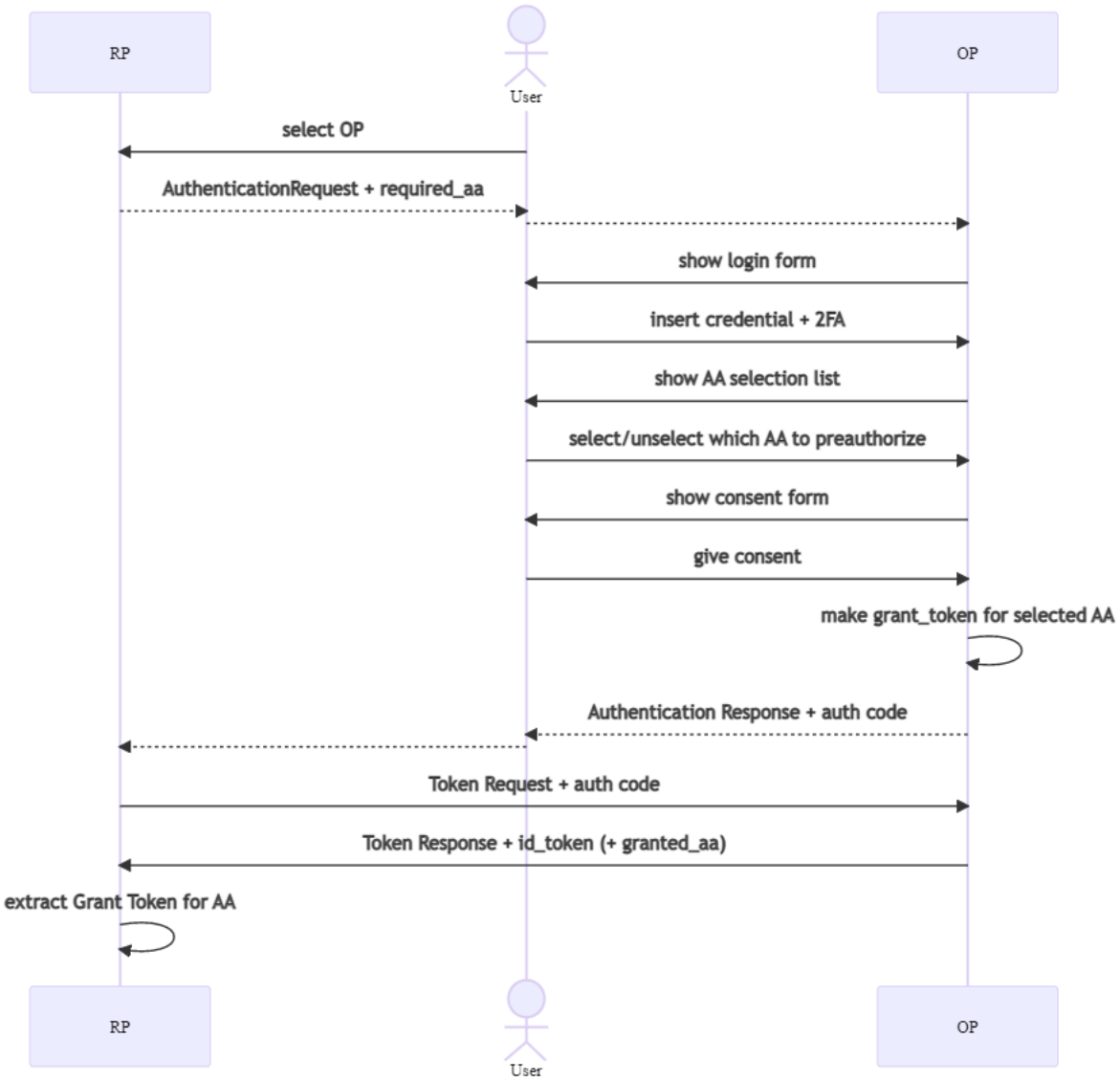

Figura 5: Sequence Diagram - Verifica della Token Response OIDC

3.2.6 Grant Token Exchange

3.2.6.1 Security Token Service

La AA DEVE implementare un endpoint di STS (Security Token Service) per supportare la richiesta di Token Exchange da parte del SP / RP. La richiesta DEVE contenere il Grant Token generato per la specifica AA e restituire in risposta, in caso di esito positivo, un Access Token per il consumo delle risorse protette dalla AA.

3.2.6.2 Token Exchange Request

Il SP / RP invia all'endpoint di token exchange una richiesta HTTP POST con autenticazione private_key_jwt[8] contenente I seguenti parametri:

- **grant_type**: (obbligatorio) "*urn:ietf:params:oauth:grant-type:token-exchange*";
- **resource**[9]: (opzionale) URL assoluta corrispondente all'endpoint presso l'AA che permette di recuperare l'attributo qualificato e per il quale si vuole ottenere l'access_token. È possibile utilizzare un array di *resource* per indicare che l'access_token emesso è destinato a essere utilizzato presso le risorse multiple elencate;
- **scope**: (opzionale) lista di stringhe, delimitata da spazi, che identifica l'ambito di utilizzo dell'autorizzazione (esempio: *get-qualifica*);
- **requested_token_type**: (obbligatorio) "*urn:ietf:params:oauth:token-type:access_token*";
- **subject_token:** (obbligatorio) il Grant Token ottenuto precedentemente per la AA;
- **subject_token_type:** (obbligatorio) "*https://spid.gov.it/attribute-authority/grant-token*";
- **client_assertion**: (obbligatorio) JWT firmato con la chiave privata del RP[10];
- **client_assertion_type**: (obbligatorio) "*urn:ietf:params:oauth:client-assertion-type:jwt-bearer*".

```
POST /token HTTP/1.1
Host: as.aa1.it
Content-Type: application/x-www-form-urlencoded

client_assertion=eyJhbGciOiJIUzI1NiIs …
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwtbearer
&grant_type=urn:ietf:params:oauth:grant-type:token-exchange
&resource=https://rs.aa1.it/attributes
&scope=read:qualifica
&requested_token_type=urn:ietf:params:oauth:token-type:access_token
&subject_token=eyJhbGciOiJS ...
&subject_token_type=https://spid.gov.it/attribute-authority/grant-token
```

Esempio 16: Token Exchange Request

3.2.6.3 Verifica della Token Exchange Request

Alla ricezione di una Token Exchange Request l'AA verifica che:

- la request contenga i parametri obbligatori;
- i valori per *grant_type*, *requested_token_type*, *subject_token_type*, *client_assertion_type* siano uguali ai valori consentiti;
- il JWT di *client_assertion* sia correttamente firmato, verificando la firma con la chiave pubblica del RP;
- l'eventuale URL indicata in *resource* appartenga alla AA e sia un indirizzo di risorsa valido;
- lo *scope, se presente,* sia un ambito riconosciuto valido all'interno della AA;
- il Grant Token inviato nel parametro *subject_token* venga decifrato correttamente utilizzando la chiave privata dell'authorization server della AA;

[8] *https://openid.net/specs/openid-connect-core-1_0.html#ClientAuthentication*

[9] *https://datatracker.ietf.org/doc/html/rfc8693/#section-2.1*

[10] *vedi paragrafo 7.1 delle Linee Guida OpenID Connect in SPID (agid.gov.it)*

- il Grant Token decifrato sia firmato correttamente, verificando la firma con il certificato pubblico dell'OP che lo ha emesso;
- il valore del parametro *aud* del Grant Token corrisponda all'issuer dell'AS della AA;
- il Grant Token non risulti scaduto (valore del parametro *exp* all'interno del Grant Token).

Se la verifica non ha esito positivo, l'AS restituisce una risposta di errore OAuth2 con codice di stato HTTP 400 e codice di errore *invalid_request*.

3.2.6.4 Autorizzazione per l'accesso alla risorsa

L'AA valuta se è possibile rilasciare un Access Token sulla base delle seguenti informazioni:

- **Ambito di utilizzo**: parametro *scope* contenuto nella Token Exchange Request.
- **Endpoint da interrogare**: parametro *resource* contenuto nella Token Exchange Request.
- **Fornitore del servizio**: parametro *act* contenuto nel Grant Token.
- **Livello di autenticazione SPID**: parametro *acr* contenuto nel Grant Token.
- **Identificativo dell'utente**: valore del claim sub del Grant Token.
- *Ulteriori attributi dell'utente ricevuti nel Grant Token.*

La AA PUÒ definire le policy in relazione ai suddetti parametri sulla base delle quali rilasciare o negare l'accesso alla specifica risorsa. Nel caso in cui l'autorizzazione non possa essere rilasciata, l'AS restituisce una risposta di errore OAuth2 con codice di stato HTTP 400 e codice di errore *unauthorized_client*.

3.2.6.5 Token Exchange Response

Nel caso in cui l'autorizzazione possa essere rilasciata l'endpoint di Token Exchange restituisce una Response con media type "application/json" e codice di stato HTTP 200 contenente nel body I seguenti parametri:

access_token: token di accesso con il quale il SP / RP può effettuare la richiesta di attributo;
issued_token_type: "urn:ietf:params:oauth:token-type:access_token";
token_type: "Bearer oppure *DPoP* ";
expires_in: tempo di validità del token in secondi.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  access_token: "eyJhbGciOiJFUzI1NiIsImtp…",
  issued_token_type: "urn:ietf:params:oauth:token-type:access_token",
  token_type: "Bearer",
  expires_in: 1800
}
```

Esempio 17: Token Exchange Response

3.2.6.6 Token Exchange Response per richieste continuative

Se la richiesta di Token Exchange contiene nel parametro scope anche il valore offline_access, e se la AA supporta il rinnovo dei token, nella risposta sarà presente anche il parametro

refresh_token con il quale il SP PUÒ ottenere un nuovo Access Token per effettuare richieste continuative in maniera asincrona.

```
POST /token HTTP/1.1
HOST as.aa1.it
Content-Type: application/x-www-form-urlencoded

client_assertion=eyJhbGciOiJIUzI1NiIs…
&client_assertion_type=urn%3Aietf%3Aparams%3Aoauth%3Aclient-assertion-type%3Ajwtbearer
&grant_type=urn:ietf:params:oauth:grant-type:token-exchange
&resource=https://rs.aa1.it/attributes
&scope=offline_access read:qualifica
&requested_token_type=urn:ietf:params:oauth:token-type:access_token
&subject_token=eyJhbGciOiJS ...
&subject_token_type=https://spid.gov.it/attribute-authority/grant-token
```

Esempio 18: Token Exchange Request per richieste continuative

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  access_token: "eyJhbGciOiJFUzI1NiIsImtp …",
  issued_token_type: "urn:ietf:params:oauth:token-type:access_token",
  token_type: "Bearer",
  expires_in: 1800,
  refresh_token: "eyJhbGciOiJFUzI1NiIsImtp …"
}
```

Esempio 19: Token Exchange Response con refresh_token per richieste continuative

3.3 Richiesta con necessità dell'assenso e riautenticazione – "private"

Viene di seguito descritto il flusso nel caso in cui l'acquisizione dell'assenso dell'utente alla trasmissione dei dati sia necessaria. Tale acquisizione avviene, da parte della AA, tramite autenticazione dell'utente presso l'IdP SPID.

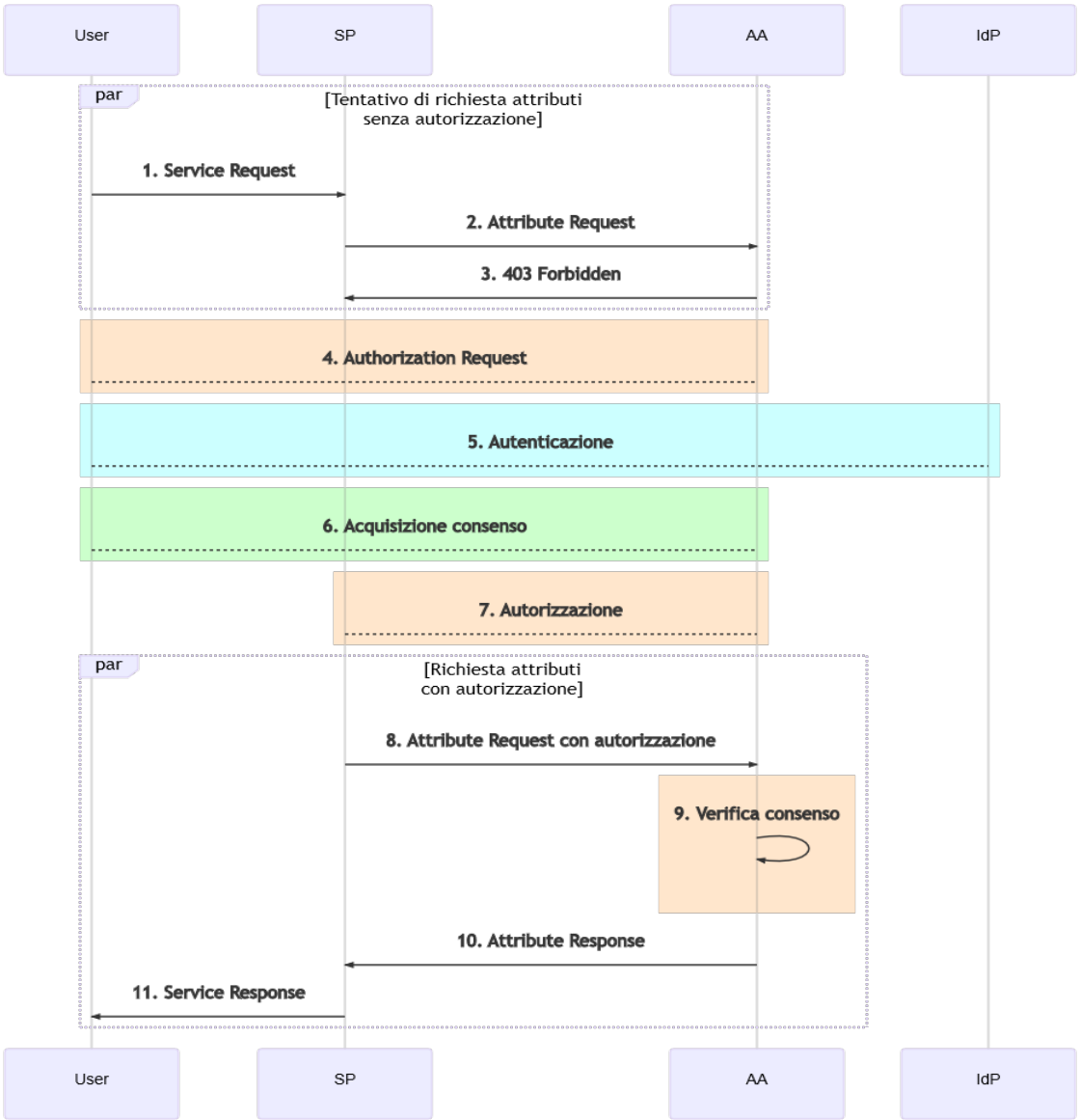

Figura 6: Flusso - richiesta con necessità dell'assenso e riautenticazione (semplificato)

Il rilascio dell'autorizzazione dell'utente al SP per effettuare le richieste di attributi è implementato tramite il flusso standard di *Authorization Code Flow* definito dai protocolli OAuth2.0[11] e OIDC[12]. Il flusso mostrato nello schema seguente riprende parte del flusso della

[11] *The OAuth 2.0 Authorization Framework (https://tools.ietf.org/rfc/rfc6749.txt)*

[12] *OIDC Core 1.0 (https://openid.net/specs/openid-connect-core-1_0.html)*

Figura 7. Per continuità di lettura, lo schema seguente riprende anche la stessa numerazione, specificando nel dettaglio i passi relativi ai processi di autorizzazione, autenticazione e acquisizione dell'assenso. A differenza del precedente schema, l'AA, intesa come Resource Server (RS) è stavolta divisa rispetto all'**Authorization Server (AS)**.

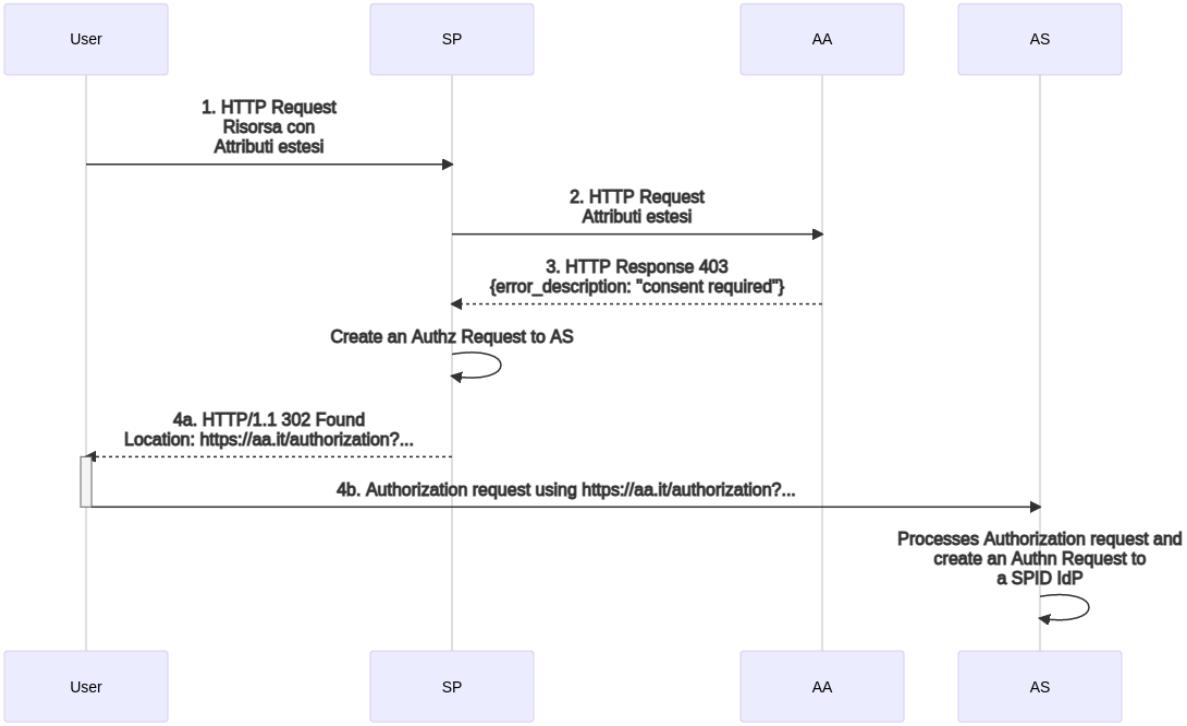

Figura 7: Flusso - richiesta con necessità dell'assenso e riautenticazione

1. L'utente chiede di utilizzare un servizio presso il SP (**Service Request**) per il quale è necessario conoscere uno o più attributi qualificati.
2. Il SP avendo necessità di conoscere gli attributi qualificati per erogare il servizio esegue una chiamata API alla specifica AA che ha la facoltà di attestare e restituire tali attributi qualificati.
3. L'AA verifica l'integrità e l'autenticità della request (**Attribute Request**) e, avendo necessità di ottenere l'assenso per restituire gli attributi richiesti, restituisce al SP un messaggio del tipo 403 Forbidden, contenente un oggetto JSON con i parametri **error_description** valorizzato a "Consent Required".
4.
    a. Il SP valida la response ricevuta dalla AA e costruisce una **Authorization Request**
    b. Il SP reindirizza l'utente sull'AS con l'URL di **Authorization Request**

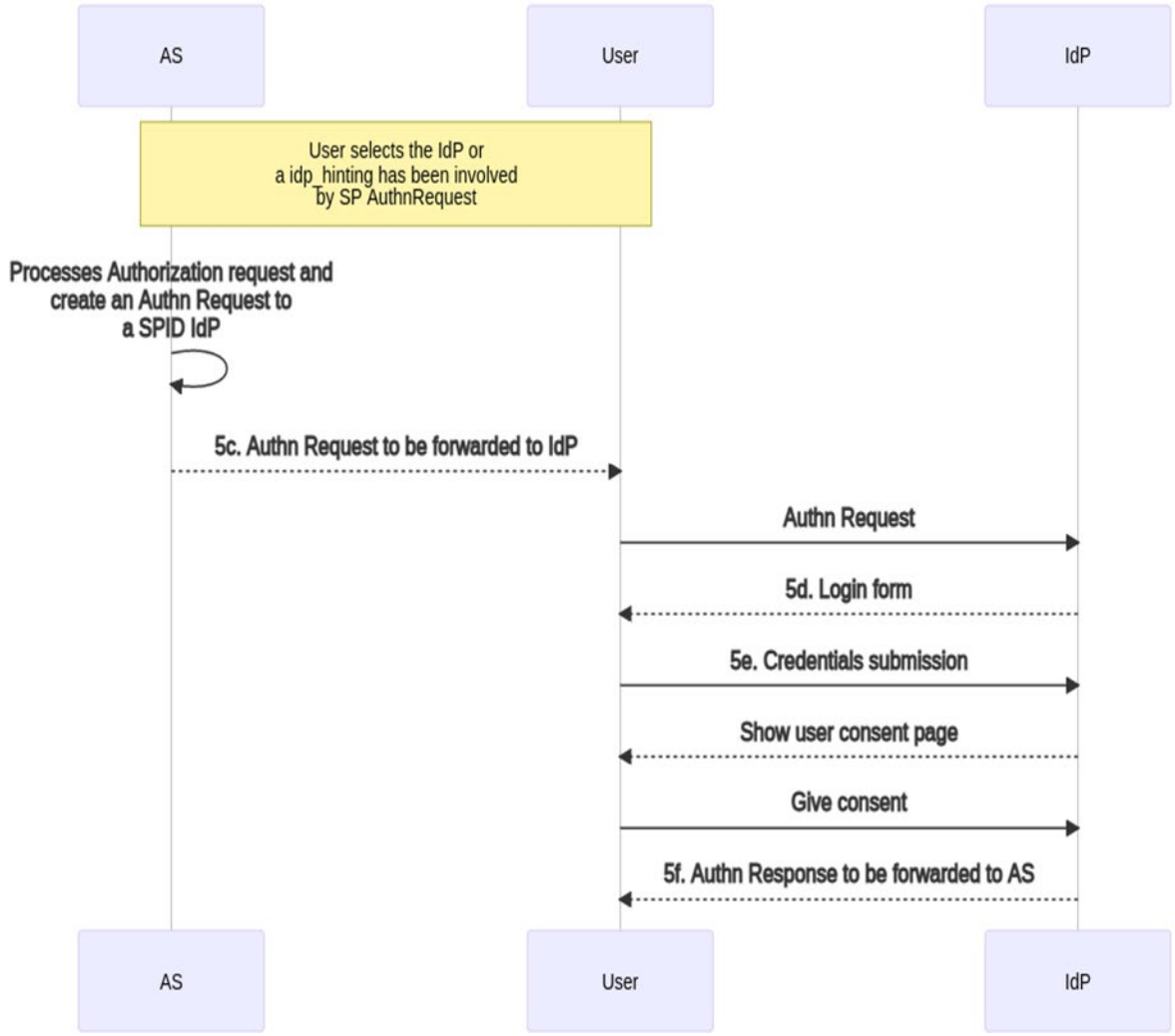

Figura 8: Flusso - Autenticazione richiesta dalla AS presso l'IdP SPID per conto dell'utente da identificare

5.

a. L'AS presenta all'utente la scelta dell'IdP presso il quale effettuare l'autenticazione ai fini dell'acquisizione l'assenso (se IdP hinting fosse assente)
b. L'utente seleziona l'IdP (se IdP hinting fosse assente)
c. L'AS reindirizza l'utente presso l'IdP selezionato con la richiesta di autenticazione.
d. L'IdP presenta all'utente il form di login
e. L'utente si autentica presso l'IdP
f. L'IdP reindirizza il flusso utente verso L'AS con la Response contenente gli attributi dell'utente necessari a verificare se è possibile concedere l'autorizzazione alla richiesta di attributi qualificati

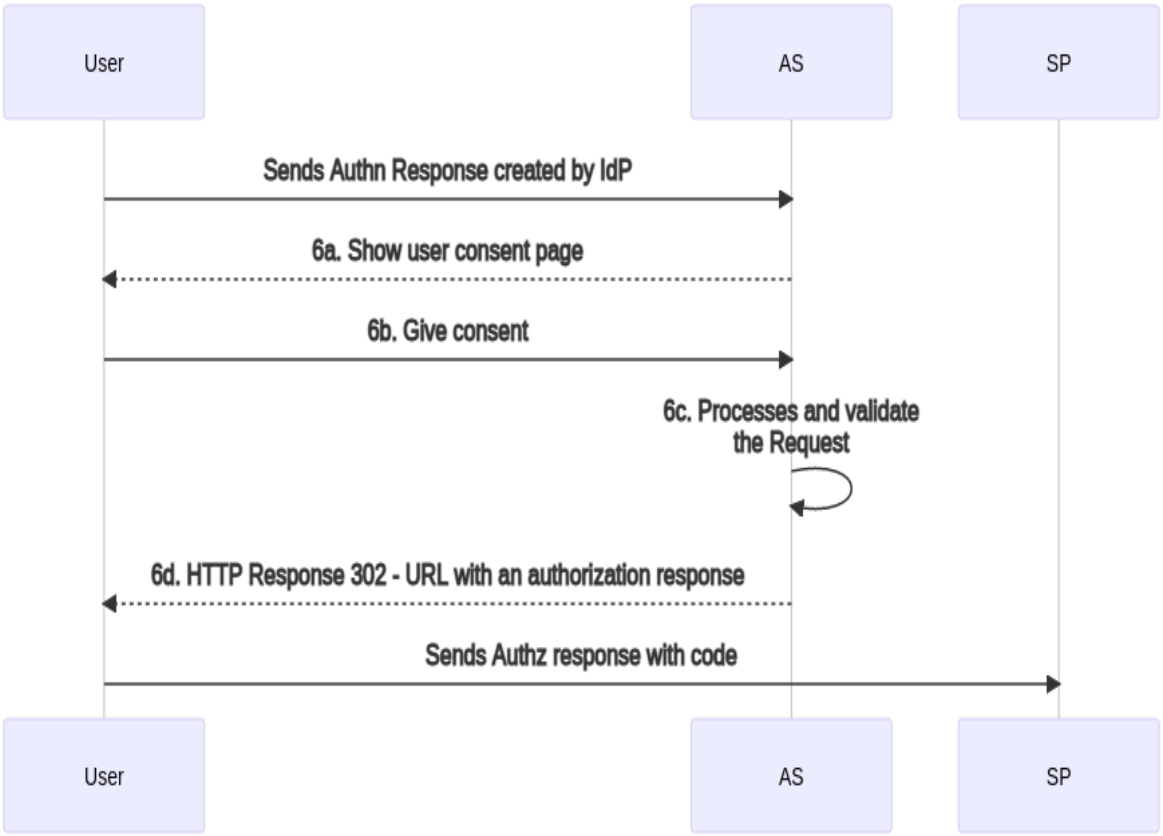

Figura 9: Flusso -Identificazione dell'utente presso l'AS e rilascio del codice di autorizzazione al SP

6.

a. L'AS chiede all'utente l'assenso alla trasmissione degli attributi qualificati al SP.
b. L'utente fornisce l'assenso alla trasmissione degli attributi qualificati al SP.
c. L'AS verifica I parametri forniti dal SP (richiesta API, attributi richiesti, identità del SP, contesto o finalità dichiarata dal SP) con l'**Authorization Request** (punto 4), gli attributi dell'utente rilasciati dall'IdP (soggetto interessato) e se l'utente ha fornito l'assenso al rilascio degli attributi qualificati (punto 6).
d. L'AS reindirizza l'utente presso il SP con l'**Authorization Code.**

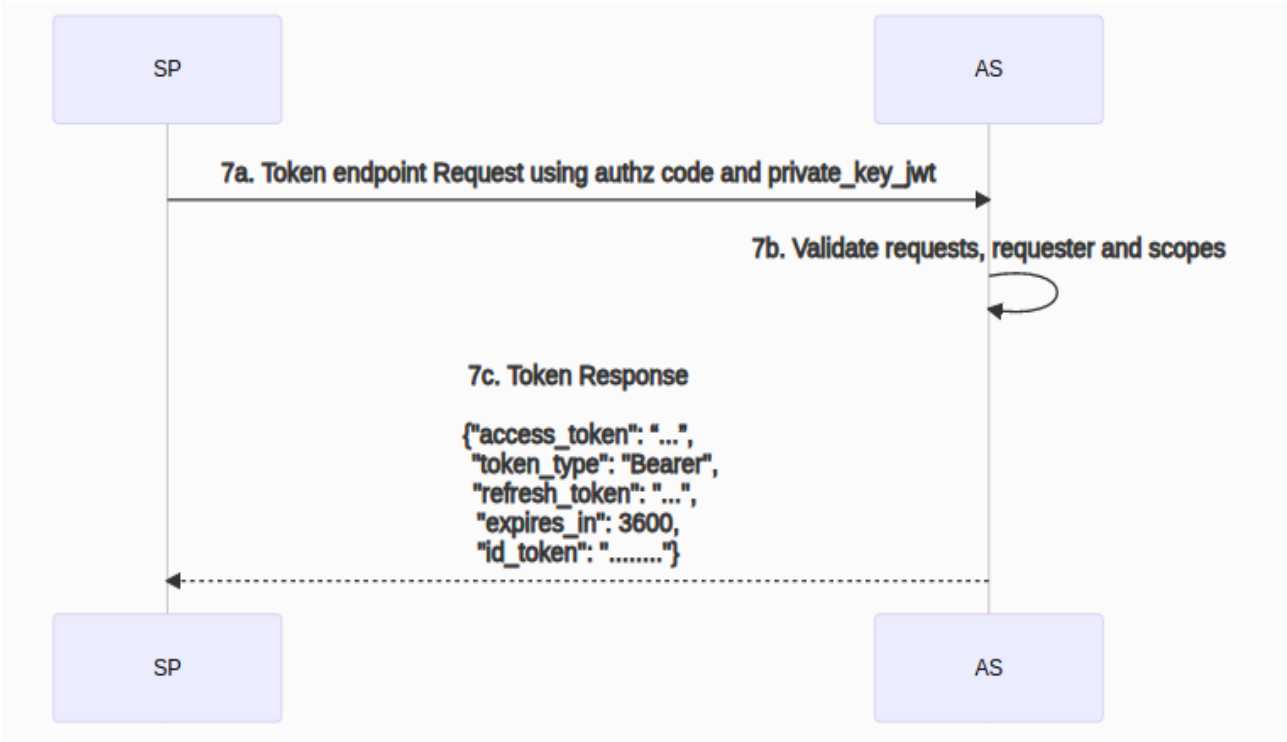

Figura 10: Flusso - Rilascio dell'Access Token al SP per il consumo delle risorse di AA

7.
   a. Il SP effettua una richiesta verso l'endpoint /token presso l'AS presentando l'**Authorization Code** al fine di ottenere un **Access Token** valido.
   b. L'AS verifica la validità dell'Authorization Code.
   c. L'AS fornisce un Access Token valido.

8. Il SP effettua nuovamente la chiamata API (**Attribute Request**) alla AA per ottenere gli attributi qualificati autenticandosi tramite l'**Access Token** ottenuto dall'AS (punto 7) a seguito dell'autorizzazione.

9.
   a. La AA verifica la firma e la validità dell'Access Token, infine per quali scopi questo sia stato rilasciato.
   b. La AA restituisce al SP gli attributi richiesti.

10. Il SP verifica l'integrità e l'autenticità della response (**Attribute Response**) ottenuta dalla AA e utilizza gli attributi qualificati ricevuti per erogare il servizio all'utente (**Service Response**).

3.3.1 Servizio di acquisizione dell'assenso (Authorization Service)

Nel *documento OpenAPI* DEVE essere presente in *securitySchemes* l'oggetto *UserConsent*.

```
components:
  securitySchemes:
    UserConsent:
     type: openIdConnect
     openIdConnectUrl: https://aa.it/.well-known/...
     description: |-
       Questa sezione esplicita il riferimento all'URL di configurazione
       di OIDC relativo al provider. Deve contenere le informazioni indicate in:
       https://openid.net/specs/openid-connect-discovery-1_0.html
     indicate in: https://openid.net/specs/openid-connect-discovery-1_0.html
```

Esempio 20: La specifica del meccanismo di autenticazione basato su OIDC e denonimato UserConsent

Il Metadata fornito dalla URL specificata in *openIdConnectUrl* DEVE contenere i riferimenti agli endpoint *authorization_endpoint* e *token_endpoint* per il flusso **private** e soltanto *token_endpoint* per il profilo **protected**. Affinché l'AA possa ottenere l'assenso dell'utente al rilascio dei suoi dati ad un RP/SP in modalità private, quest'ultimo deve redirezionare l'utente presso l'authorization_endpoint della AA. Di seguito un contenuto di esempio dei metadati OIDC usati per ottenere l'assenso dell'utente per il flusso private:

```
{
  "version": "3.0",
  "issuer": "https://as.aa.it",
  "organization_name": "Example Attribute Authority",
  "logo_uri": "https://as.aa.it/static/logo.svg",
  "token_endpoint_auth_methods_supported": [
    "private_key_jwt"
  ],
  "claims_parameter_supported": true,
  "grant_types_supported": [
    "authorization_code",
    "urn:ietf:params:oauth:grant-type:jwt-bearer",
    "refresh_token"
  ],
  "subject_types_supported": [
  "public",
  "pairwise"
  ],
  "introspection_endpoint": "https://as.aa.it/introspection",
  "response_types_supported": [
    "code",
  ],
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "request_object_signing_alg_values_supported": [
    "RS256"
  ],
  "request_object_encryption_alg_values_supported": [
    "RSA-OAEP",
```

```
    "RSA-OAEP-256"
  ],
  "request_object_encryption_enc_values_supported": [
    "A128GCM",
    "A192GCM",
    "A256GCM"
  ],
  "claim_types_supported": [
    "normal",
    "aggregated",
    "distributed"
  ],
  "authorization_endpoint": "https://as.aa.it/authorization",
  "token_endpoint_auth_signing_alg_values_supported": [
    "RS256"
  ],
  "token_endpoint": "https://as.aa.it/token",
  "userinfo_signing_alg_values_supported": [
    "RS256"
  ],
  "end_session_endpoint": "https://as.aa.it/session",
  "acr_values_supported": [
    "https://www.spid.gov.it/SpidL1",
    "https://www.spid.gov.it/SpidL2"
  ],
  "jwks_uri": "https://as.aa.it/static/jwks.json",
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "id_token_encryption_alg_values_supported": [
    "RSA-OAEP",
    "RSA-OAEP-256"
  ],
  "id_token_encryption_enc_values_supported": [
    "A128GCM",
    "A192GCM",
    "A256GCM"
  ],
  "scopes_supported": [
    "offline_access",
    "openid",
    "read:qualifica",
  ],
  "claims_supported": [
    "https://attributes.eid.gov.it/spid_code",
    "given_name",
    "https://attributes.eid.gov.it/fiscal_number",
    "email"
  ]
}
```


Esempio 21: il documento openid-configuration che descrive le caratteristiche dell'AS per il profilo private

Se una richiesta di attributi qualificati, nel profilo **private**, non può essere soddisfatta perché sprovvista della prova di assenso dell'utente, l'AA DEVE ritornare una risposta con codice di stato *HTTP 400 Bad Request* e un contenuto di tipo json con l'indicazione descrittiva "*Authorization required*" all'interno dell'attributo **error_description** [RFC6749#section-5.2].

Nella Figura di seguito vengono illustrate le interazioni tra le Entità coinvolte per il profilo **private**.

È importante considerare che nel seguente esempio una AA e un AS rappresentano endpoint distinti.

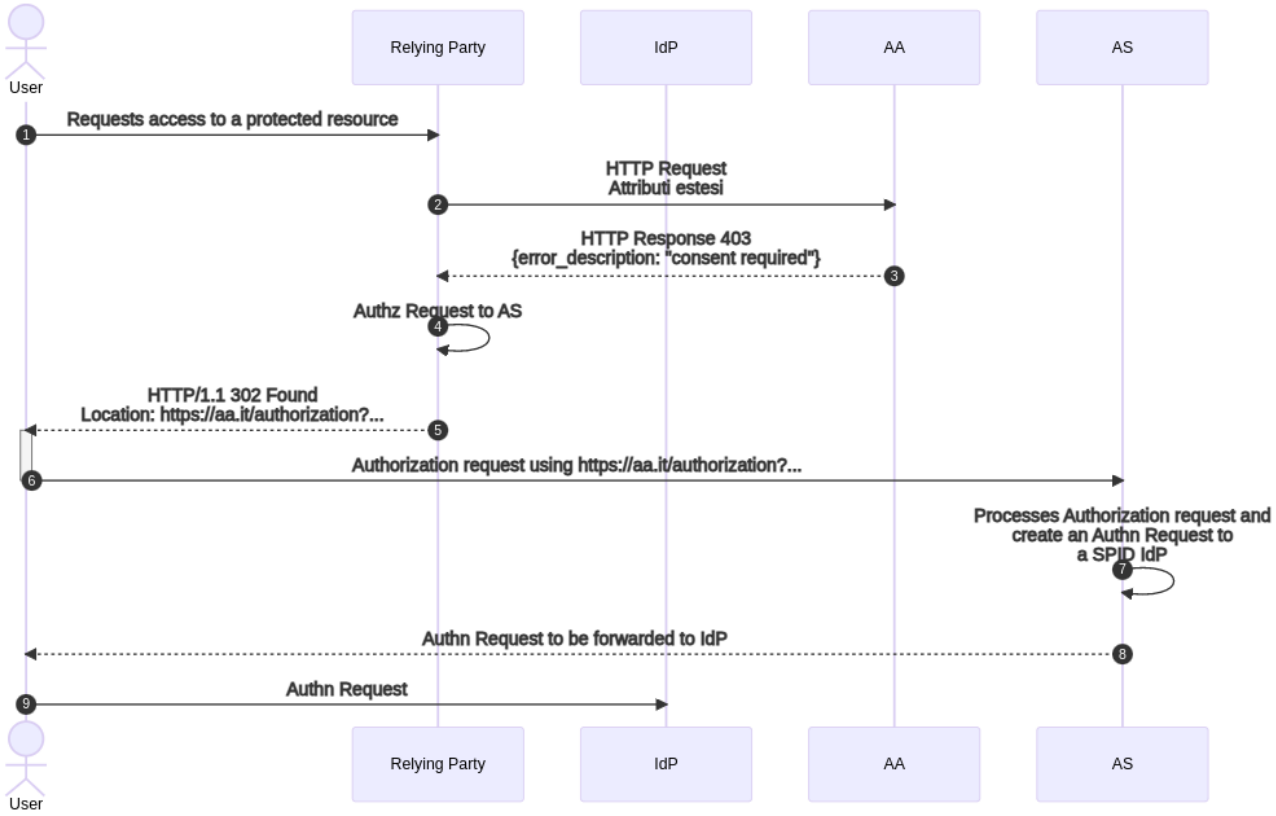

Figura 11: Sessione di richiesta Attributi con richiesta di autenticazione. Gli endpoint di AA e AS vengono erogati da entità diverse

3.3.2 Suggerimento dell'IdP da utilizzare (IdP hinting)

L'IdP hinting è un meccanismo che permette al SP/RP di indicare alla AA quale IdP utilizzare per autenticare l'utente. Gli SP e le AA DOVREBBERO implementare l'IdP hinting, perché questo meccanismo semplifica il processo di autenticazione evitando all'utente di selezionare nuovamente l'IdP (bottone "Entra con SPID"). Un SP/RP che supporta IdP hinting:

1. DEVE includere nella redirect url OIDC contenuta nella authorization request diretta alla AA l'URL parameter *idphint*;
2. il parametro *idphint* DEVE corrispondere all'identificativo univoco dell'IdP presso il quale l'utente è stato precedentemente autenticato dall'SP e DEVE essere codificato in percent encoding [RFC3986].

Quando un SP non invia il parametro *idphint*, una AA che supporta IdP hinting DEVE comunque mostrare la schermata "*Entra con SPID*".

Esempio: per indicare l'IdP con identificativo https://idp.spid.it il parametro idphint sarà valorizzato come *idphint=https%3A%2F%2Fspid.idp.it*.

3.5 Attribute Request

Il SP invia una richiesta di attributi qualificati all'AA che detiene il dato secondo la documentazione OpenAPI.

1. Ove richiesto dalla specifica la richiesta DEVE autenticarsi tramite un Access Token JWT di tipo *Bearer* o *DPoP* secondo le indicazioni contenute nei paragrafi *Autenticazione API*.

```
GET /v1/attribute HTTP/1.1
Host: aa.it
Authorization: Bearer eyJ0eXAiOiJKT1NFIiwiYWxnIjoiSFMyNTYiLCJlbmMiOiJIUzI1NiIsIng1YyI6IkNlc ...
```

Esempio 22: Intestazione di una Attribute Request con credenziali di autenticazione di tipo Bearer

3.5.1 Demonstrating Proof-of-Possession (DPoP[13])

Le AA POSSONO adottare il meccanismo di sicurezza denominato *Demonstrating Proof-of-Possession*[14] (DPoP) di conseguenza i RP DEVONO implementarlo.

3.6 Attribute Response

In risposta ad una Attribute Request l'AA ritorna un messaggio che PUÒ contenere gli attributi qualificati richiesti oppure l'indicazione dell'errore riscontrato per il quale non è possibile trasmettere gli attributi richiesti. I possibili dati contenuti nella risposta o le specifiche dell'errore sono definiti nella documentazione OpenAPI dell'AA.

La risposta DOVREBBE avere Media-Type *application/jwt* in modo da poter firmare e cifrare il contenuto, ma PUÒ anche avere Media-Type *application/json*.

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "id_documento": "AA12345678" }
```

Esempio 23: Attribute Response con JWT firmato (JWS)

```
HTTP/1.1 200 OK
Content-Type: application/jwt

eyJ0eXAiOiJKT1NFIiwiYWxnIjoiSFMyNTYiLCJlbmMiOiJIUzI1NiIsIng1YyI6IkNlcnRpZmljYXRvL2NvZGlmaWNhd ...
```

Esempio 24: Attribute Response con JWT firmato (JWS)

[13] https://datatracker.ietf.org/doc/html/draft-ietf-oauth-dpop

Indice delle tabelle

Indice delle figure

Bibliografia

Fielding, Roy Thomas (2000). "Representational State Transfer (REST)". *Architectural Styles and the Design of Network-based Software Architectures* (PhD). University of California, Irvine
Modello di interoperabilità per la Pubblica Amministrazione. (https://www.agid.gov.it/it/infrastrutture/sistema-pubblico-connettivita/il-nuovo-modello-interoperabilita)
Semantic Version Specification. (https://semver.org/)
AARC-G049 - A specification for IdP hinting. (https://aarc-project.eu/wp-content/uploads/2019/04/AARC-G049-A_specification_for_IdP_hinting-v6.pdf)
Open API Specification v3 - (https://swagger.io/specification)
OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer (DPoP) (https://datatracker.ietf.org/doc/html/draft-ietf-oauth-dpop)
[EN319-412-1] - Electronic Signatures and Infrastructures (ESI); Certificate Profiles;
RFC3986 - Uniform Resource Identifier (URI): Generic Syntax (https://datatracker.ietf.org/doc/html/rfc3986)
RFC6749 - The OAuth 2.0 Authorization Framework (https://tools.ietf.org/rfc/rfc6749.txt)
RFC6750 - Bearer Token Authentication - (https://tools.ietf.org/html/rfc6750)
RFC7515 - JSON Web Signature (JWS) (https://datatracker.ietf.org/doc/html/rfc7515)
RFC7516 - JSON Web Encryption (JWE) (https://datatracker.ietf.org/doc/html/rfc7516)
RFC7519 - JSON Web Token (JWT) - (https://tools.ietf.org/html/rfc7519)
RFC7523 (private_key_jwt) - JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants (https://datatracker.ietf.org/doc/html/rfc7523)
RFC7807 - Problem Details for HTTP APIs (https://tools.ietf.org/rfc/rfc7807.txt)
OIDC-CORE - OpenID Connect Core 1.0 (https://openid.net/specs/openid-connect-core-1_0.html)
RFC8259 - The JavaScript Object Notation (JSON) Data Interchange Format (https://datatracker.ietf.org/doc/html/rfc8259)
RFC8693 - OAuth 2.0 Token Exchange (https://datatracker.ietf.org/doc/html/RFC8693)
RFC8725 - JSON Web Token Best Current Practices (https://tools.ietf.org/html/rfc8725)
Linee Guida SPID OIDC
- (https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_openid_connect_in_spid.pdf)
OpenID Federation 1.0 - (https://openid.bitbucket.io/connect/openid-connect-federation-1_0.html)