OpenID Connect Intro e risultati GdL IDEM OIDC

Davide Vaghetti <davide.vaghetti@garr.it>

IDEM DAY 2021

24-26 Febbraio 2021

OpenID Connect è uno ***standard aperto*** *pubblicato dalla OpenID Foundation nel 2014.*

Principi di design

Keep Simple Things Simple

Make Complex Things Possible

OpenID Connect 1.0

"a simple identity layer on top of the OAuth 2.0 protocol"

Definisce l'identità dell'utente: identificatori e claims

REST-like

JWT (JSON Web Token)
JWK (JSON Web Key)
JWS (JSON Web Signature)
JWE (JSON Web Encryption)

Mobile friendly

(non) Facciamo confusione

SAML2	**OAuth2**	**OIDC**
Identity Provider	Authorization Server	OpenID Provider
Service Provider	Client	Relying-Party
XML	JSON	JSON
XMLSEC	JOSE (JWT/JWS/JWE/JWK)	JOSE (JWT/JWS/JWE/JWK)
Assertions, Attributes	Custom API, Claims	Standard API, Claims
Http GET/POST, SOAP, POAS	Http GET/POST	Http GET/POST
Metadata	Authorization Server Metadata (.well-known/oauth-authorization-server)	OpenID Connect Discovery 1.0 (.well-known/openid-configuration)

Attori

Utente

Accede ad una risorsa utilizzando uno User-Agent HTTP.

Relying Party o Client

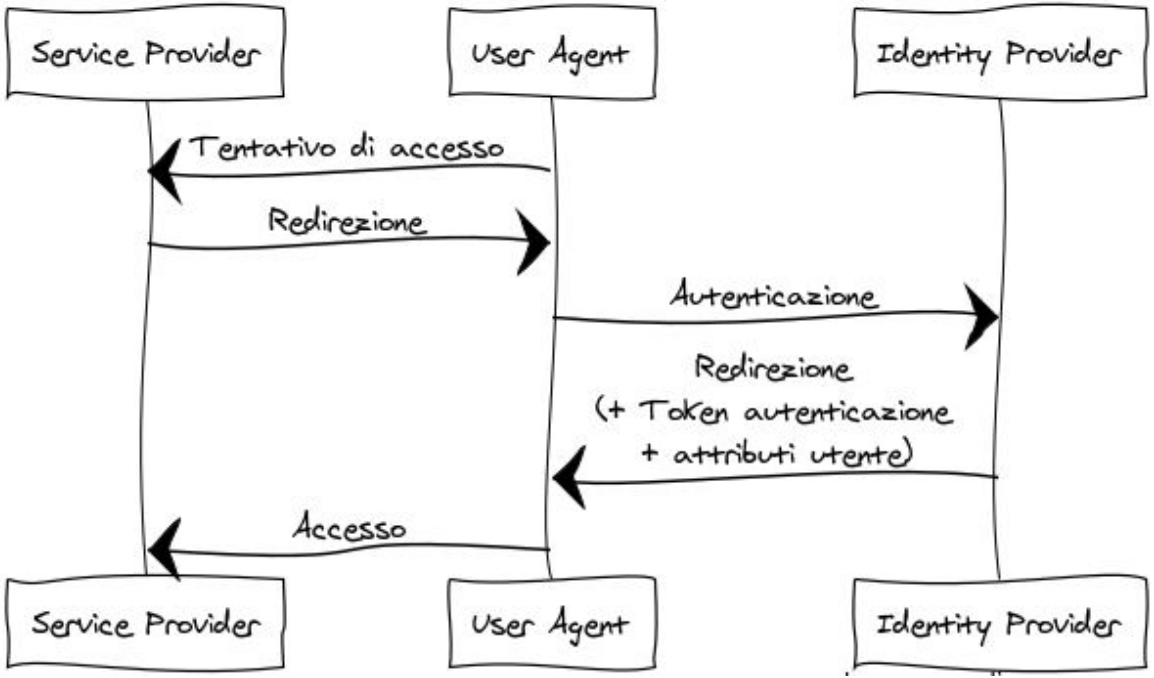
OAuth 2.0 Client application requiring End-User Authentication and Claims from an OpenID Provider.

OpenID Provider

OAuth 2.0 Authorization Server that is capable of Authenticating the End-User and providing Claims to a Relying Party about the Authentication event and the End-User.

I Flussi: SAML vs OIDC

SAML 2.0 - HTTP Redirect Binding



1. L'utente tenta di accedere ad una applicazione protetta da un Service Provider.
2. Il Service Provider richiede l'autenticazione federata.
3. L'utente sceglie la propria organizzazione e viene rediretto al Identity Provider.
4. L'utente si autentica.
5. Il Identity Provider redirige l'utente al servizio con un token di autenticazione ed una serie di attributi.
6. L'utente accede al servizio.

Endpoint HTTP (OpenID Provider)

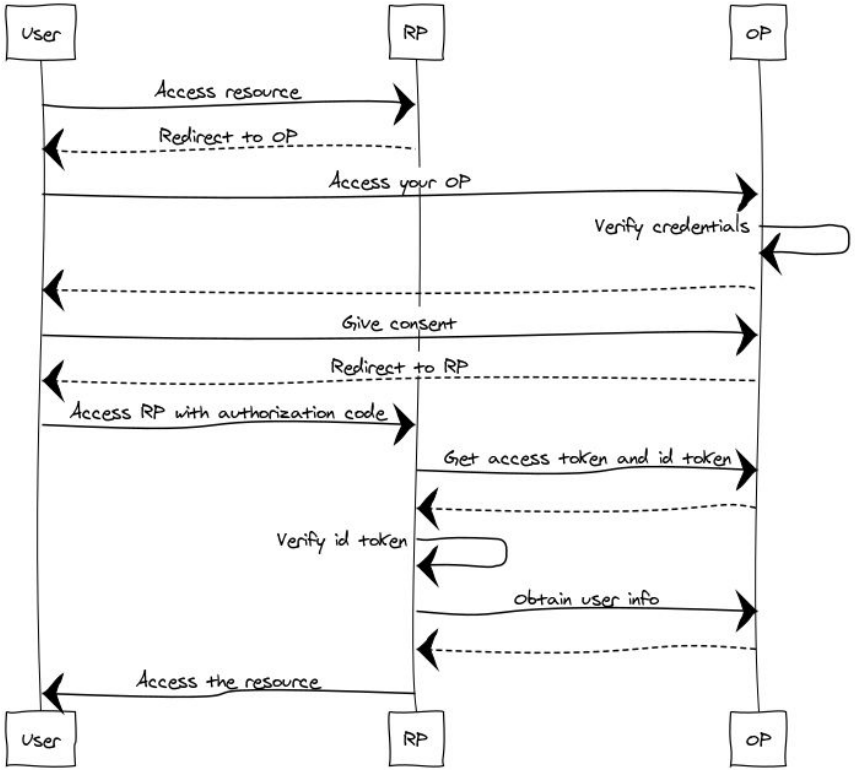
Obbligatori

- **Authorize endpoint**: dove avviene l'autenticazione dell'utente e dove si **autorizza** il Relying Party ad ottenere un token.
- **Token endpoint**: dove il Relying Party richiede un token per poter accedere all'identità dell'utente.
- **UserInfo endpoint**: dove il Relying Party richiede il profilo dell'utente.

Opzionali

- **Discovery**: (non un vero e proprio endpoint, ma una *well-known location*) dove l'OpenID Provider pubblica i metadata con tutti i parametri di configurazione (ad es. le URL degli endpoint).
- **Client Registration**: dove un Relying Party si può registrare dinamicamente presso un OpenID Provider.

OIDC - Authorization code grant



1. L'utente tenta l'accesso ad una risorsa.
2. L'RP redirige lo User-Agent al OP trasmettendo la Authentication Request come parametri HTTP.
3. L'OP autentica l'utente e ne chiede l'autorizzazione per trasmettere la sua identità al RP.
4. L'OP redirige l'utente al RP trasmettendo la Authentication Response e l'Authorization code.
5. L'RP accede al Token Endpoint scambiando l'Authorization code.
6. L'RP riceve l'ID token e l'access token, li valida e accede allo User Info endpoint per ottenere il profilo dell'utente.

Token

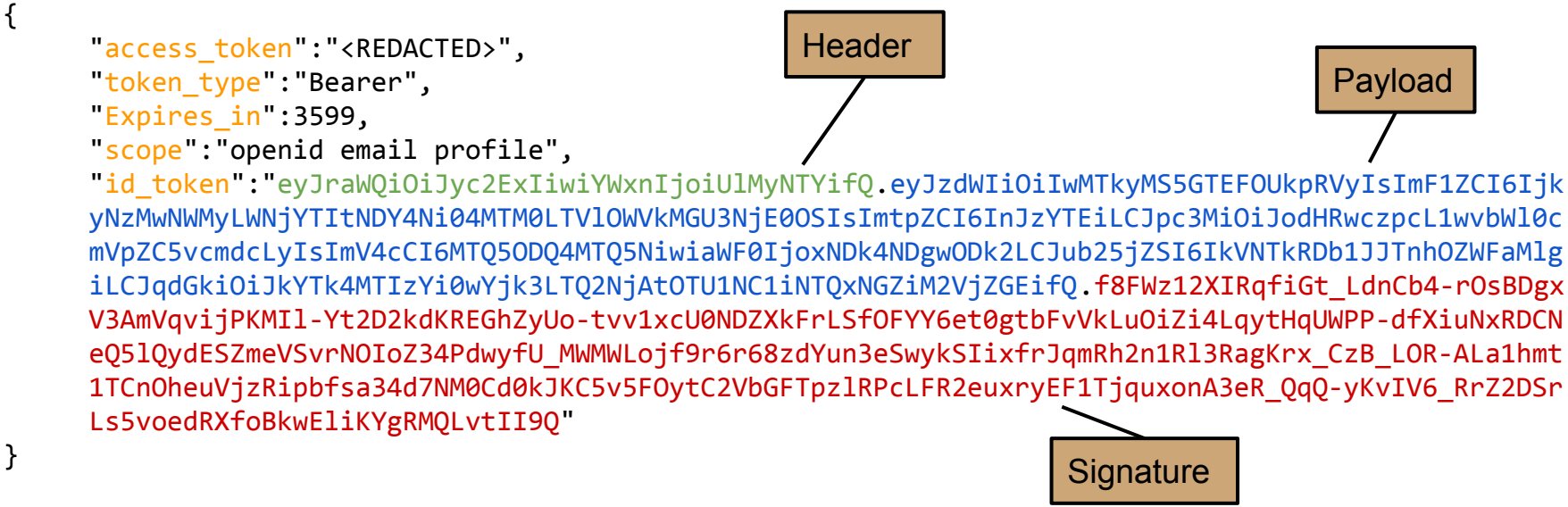
Access token: una stringa che rappresenta l'autorizzazione concessa ad un Relying Party.

Refresh token: un token che permette di ottenere altri access token quando questi scadono o diventano invalidi.

ID token: un **JSON Web Token** firmato e potenzialmente criptato che contiene le informazioni relative al processo di autenticazione e rappresenta la **credenziale** per accedere alle risorse del Relying Party.

ID Token

L'ID Token è sempre relativo ad una sessione attiva.




```
token = base64urlEncoding(header) + '.' + base64urlEncoding(payload) + '.' + base64urlEncoding(signature)
```

ID Token payload


```
{
    "sub":"01921.FLANRJQW",
    "aud":"927305c2-cca2-4686-8134-5e9ed0e76149",
    "kid":"rsa1",
    "iss":"https:\/\/mitreid.org\/",
    "exp":1498481496,
    "iat":1498480896,
    "nonce":"EMNDCoRINxNeaZ2X",
    "jti":"da98123b-0b97-4660-9554-b5414fb3ecda"
}
```

Claims e scope

Scope	Claims
email	email, email_verified
phone	phone_number, phone_number_verified
profile	name, family_name, given_name, middle_name, nickname, preferred_username, profile, picture, website, gender, birthdate, zoneinfo, locale, updated_at
address	address

Reference

OpenID Connect Federations specification

http://openid.net/specs/openid-connect-federation-1_0.html

OpenID Foundation Research and Education Working Group

https://openid.net/wg/rande/

Gruppo di lavoro OIDC
IDEM CTS

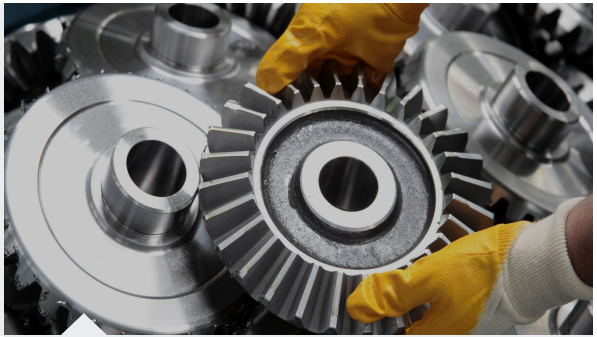
Il Gruppo

Giuseppe De Marco - coordinatore (UniCal fino a 01-2021)
Maurizio Festi (UniTrento)
Michele D'Amico (Agid)
Marco Pirovano (UniBocconi)
Angelo Rossini (Cineca)
Dario Pilori (Inrim)
Antonio Florio (Agid)
Stefano Zanelli (UniTrento)
Davide Vaghetti (Garr)
Nunzio Napolitano (UniParthenope)

Applicazioni considerate*

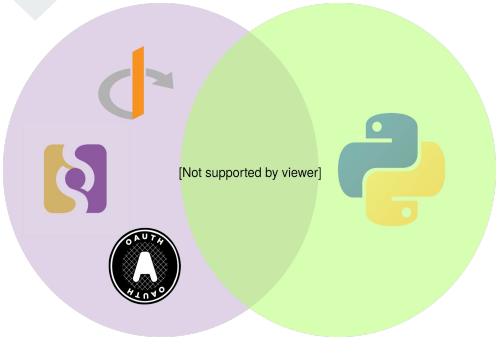
oidcendpoint (jwtconnect.io)

Identity Python community
Roland Hedberg (lead) anche editor di Fed 1.0
Giuseppe De Marco (contributore)
e sviluppatore di django-oidc-op



GÉANT OIDC-Plugin for Shibboleth IdP 1.0.0

Plugin ufficiale **Shibboleth**
Henri Mikkonen (lead)
Soluzione più popolare in **IDEM**



Altro

Ansible
Linux Container (LXC)

**Giuseppe De Marco - WS GARR 2020*

Risultati

https://wiki.idem.garr.it/wiki/Gruppo_di_Lavoro_OIDC

Documenti

- Description of an OpenID Connect Session (idpy jwtconnect.io)
- Shibboleth CSCfi OIDC - Configuration experiences

Artefatti

- django-oidc-op
- Ansible playbook (Shib IdP and SP SAML2 and CSCfi OIDC)

Test

- OIDC Federation 1.0 PoC (fedservice)

Risorse di terze parti

- White Paper for implementation of mappings between SAML 2.0 and OpenID Connect in Research and Education

Grazie

Davide Vaghetti (davide.vaghetti@garr.it)