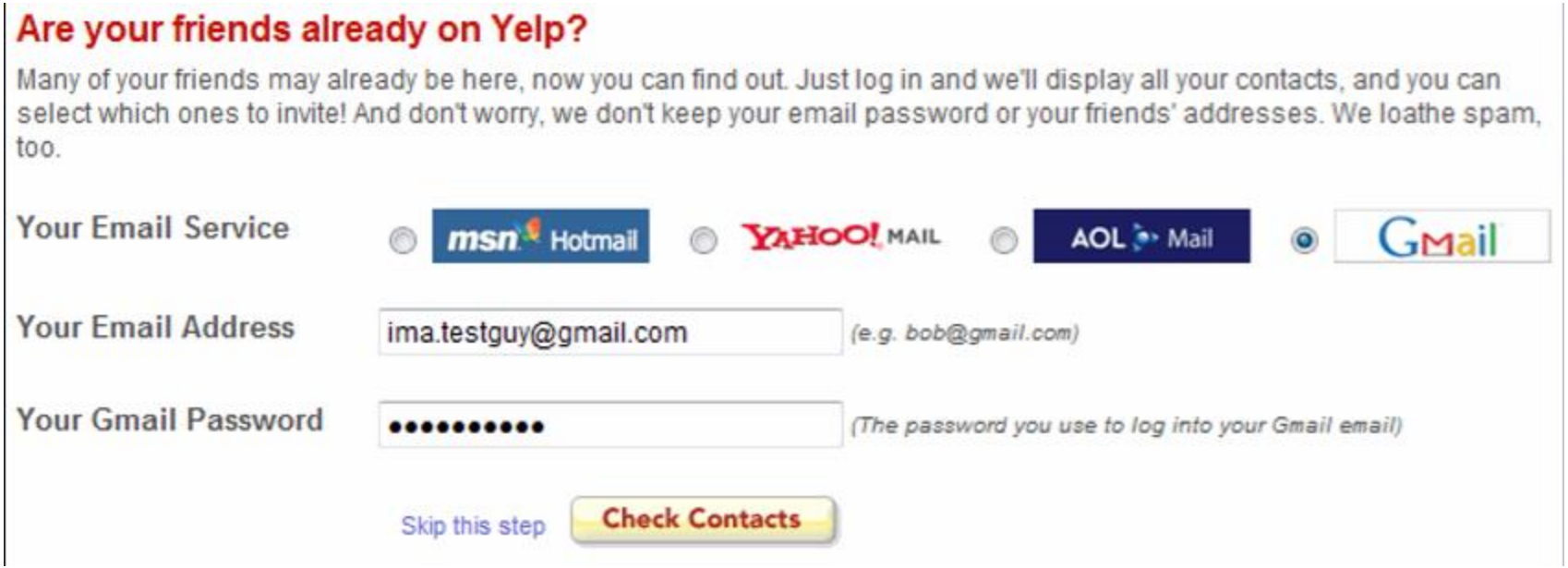
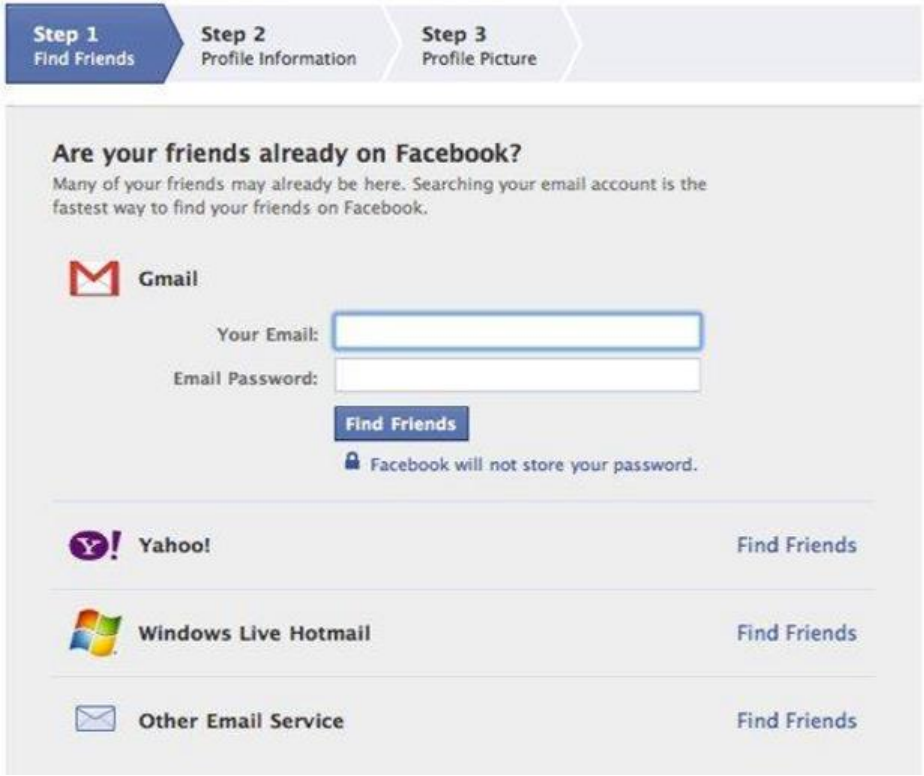

Are your friends already on Yelp?
Many of your friends may already be here, now you can find out. Just log in and we'll display all your contacts, and you can select which ones to invite! And don't worry, we don't keep your email password or your friends' addresses. We loathe spam, too.
Your Email Service
msn Hotmail
YAHOO! MAIL
AOL Mail
Gmail
Your Email Address
ima.testguy@gmail.com
(e.g. bob@gmail.com)
Your Gmail Password
(The password you use to log into your Gmail email)
Skip this step
Check Contacts

commit


Step 1
Find Friends
Step 2
Profile Information
Step 3
Profile Picture
Are your friends already on Facebook?
Many of your friends may already be here. Searching your email account is the fastest way to find your friends on Facebook.
Gmail
Your Email:
Email Password:
Find Friends
Facebook will not store your password.
Yahoo!
Find Friends
Windows Live Hotmail
Find Friends
Other Email Service
Find Friends

commit

103,00 €

Particulares / Private | Empresas

Por favor, asegúrese de usar los mismos datos de acceso de su banca online

MODO DE IDENTIFICACIÓN:

Documento

Tipo de documento:
NIF

Número de documento

Clave de acceso:
••••••••

Ich akzeptiere die Verwendung des Zahlungsinitiierungsdienstes oder des Kontoinformationsdienstes von Sofort, damit Sofort eine Zahlung initiieren oder auf mein Zahlungskonto zugreifen kann.

Se aplica nuestra Política de protección de datos personales

Continuar

commit

commit

Autorización, Tokens, Flujos, oAuth y OIDC para todo el mundo

Sobre todo developers.

TOKIOTA

Leonardo Micheloni
Developer

@leomicheloni

commit

Objetivo

Comprender los conceptos principales

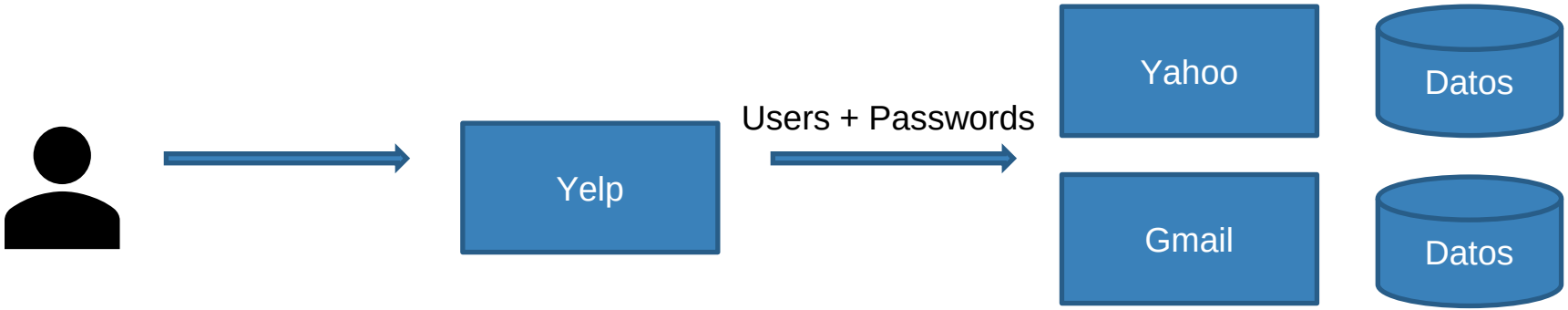
- Autorización
- Flujo
- Token
- Scope
- Client
- IDS
- Resource

Disclaimer

commit

commit

Modelo de acceso a recursos legacy



commit

Problemas

- Exponemos nuestra password
  - Y posiblemente la de otro sitio
- Damos acceso ilimitado a nuestros datos (recursos)
- Damos acceso por tiempo ilimitado
  - A menos que cambiemos nuestra password
  - Y el sitio web no lo haga por nosotros
- El Sistema opera en nuestro nombre

commit

oAuth2

Authentication is the process of ascertaining that somebody really is who they claim to be.

Authorization refers to rules that determine who is allowed to do what. E.g. Adam may be authorized to create and delete databases, while Usama is only authorised to read.

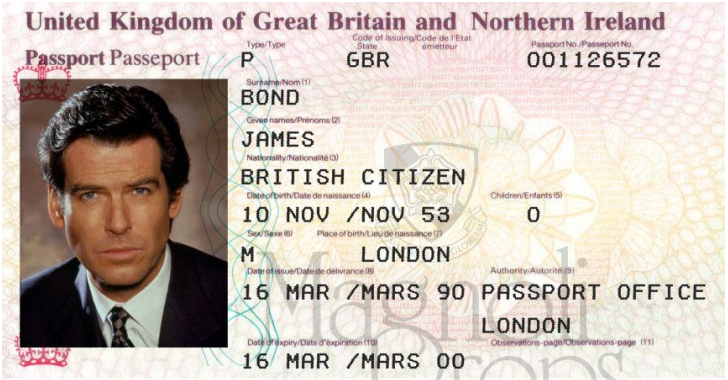
OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.[1] This mechanism is used by companies such as Amazon,[2] Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites.

Generally, OAuth provides to clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without sharing their credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server.[3]

https://oauth.net/2/

commit

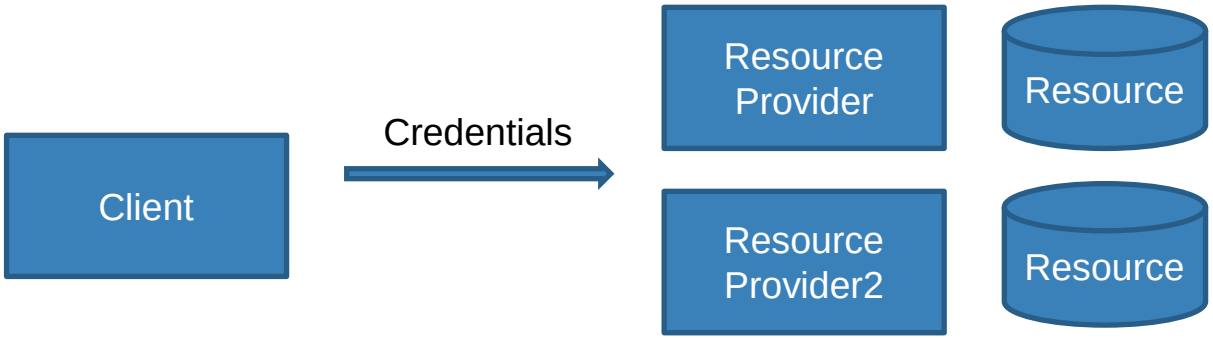
Autorización



Autenticación

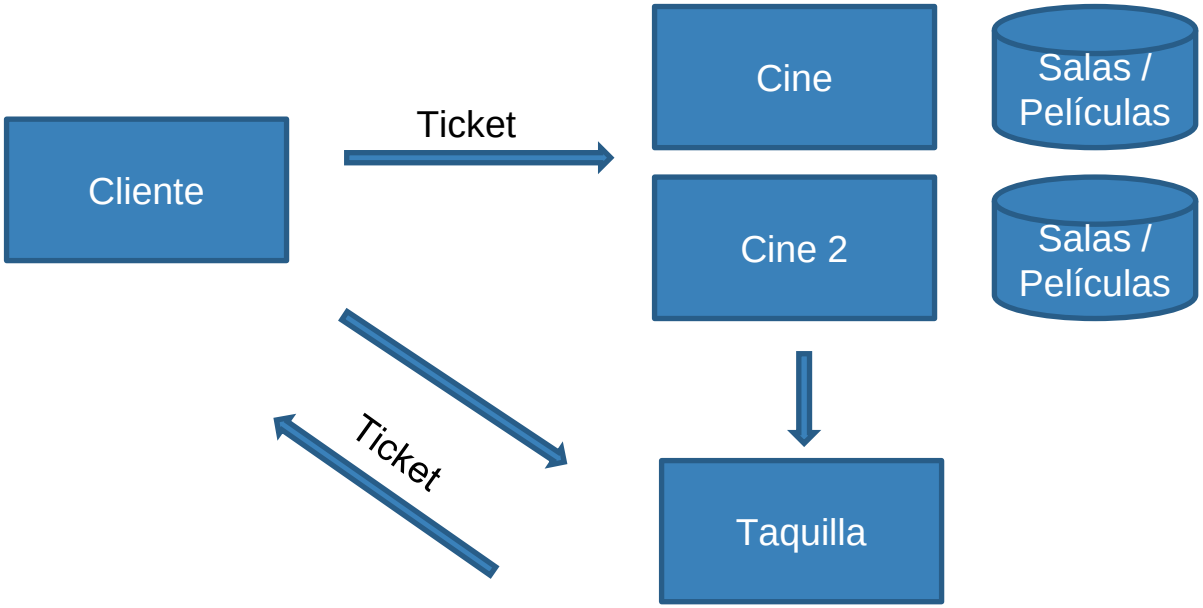
commit

Usando terminología oAuth



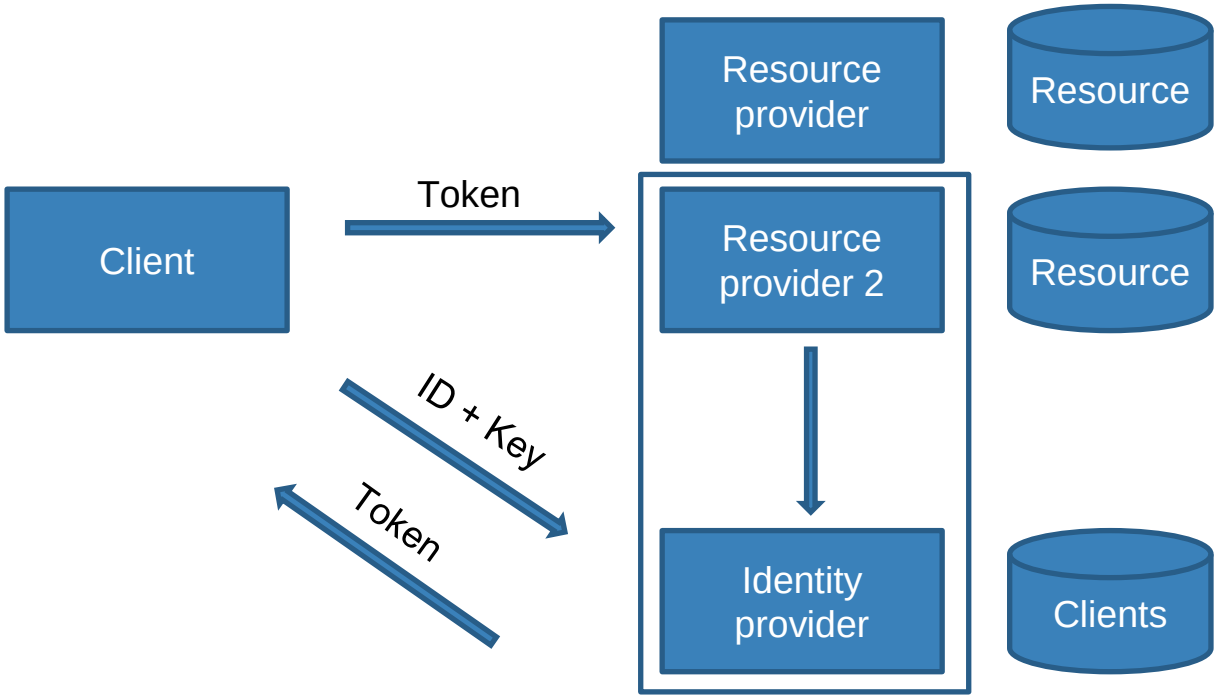
commit

Mejora basada tickets



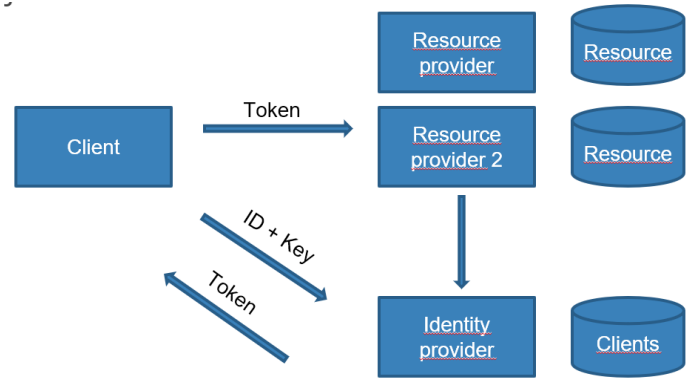
commit

Mejora basada en oAuth



commit

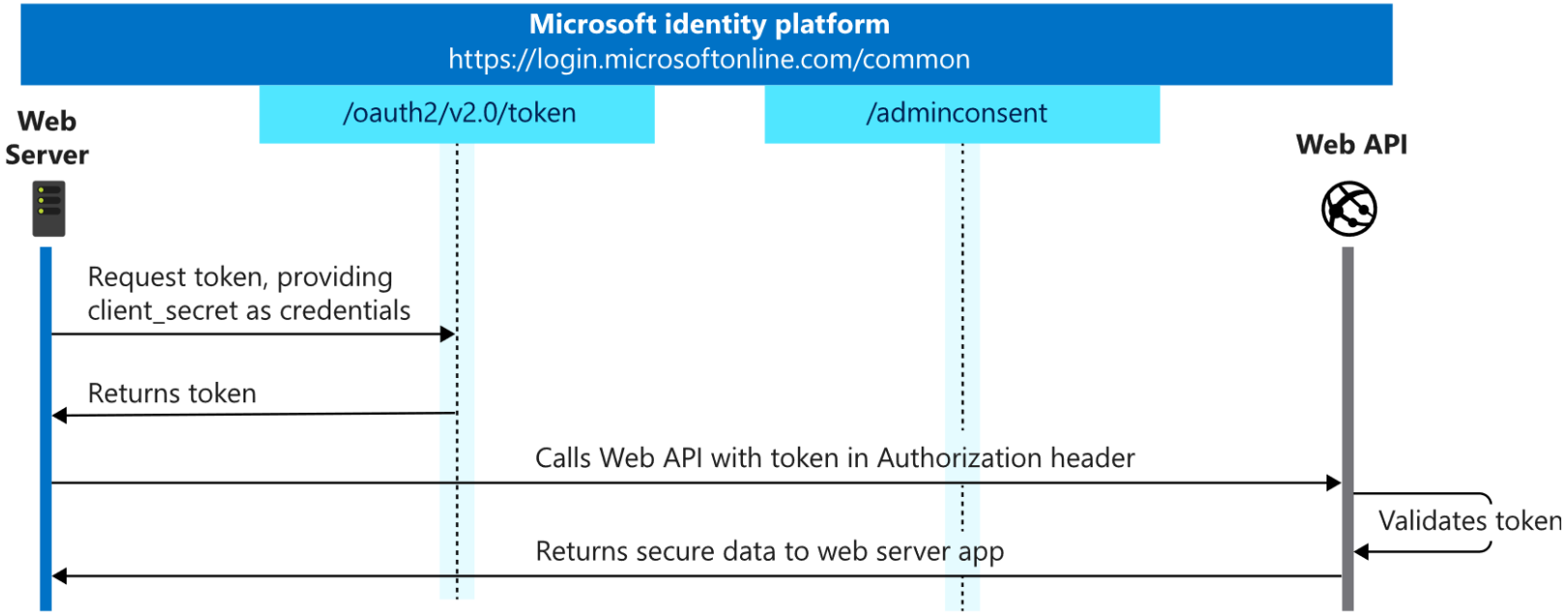
¿Qué conseguimos con esto?



- No exponer nuestras credenciales
- Otorgar un ticket (token) de acceso
- Limitar el acceso a un recurso (scope)
- Otorgar acceso por un tiempo limitado
- Separar responsabilidades
- El IDP (auth server) conoce a los clients
- Los resource providers conocen al IDP
- Centralizar el otorgamiento de accesos

commit

oAuth Client Credentials grant



commit

JWT

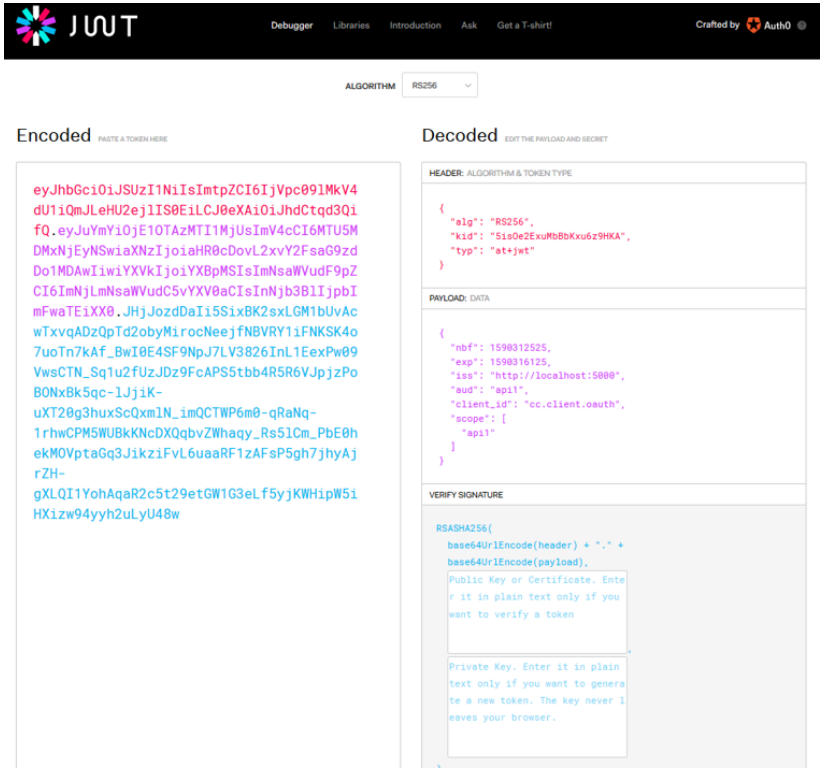
JSON Web Token

commit

@leomicheloni

El token JWT

- ¿Qué es?
  - Un conjunto de datos de texto
  - Formato JSON
- ¿Qué tiene dentro?
  - Header
  - Payload
  - Firma
- ¿Puedo leerlo?
  - Sí, el contenido está URLencoded
- ¿Se puede alterar?
  - No, porque está firmado
- ¿Cómo se valida?
  - Se verifica la firma (self signed)
  - Se pregunta al Identity provider



commit

¿Cómo funciona la firma?

commit

El token


```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICItVURQRXpOUVZPd0ptbU10VFFHbnRNa0N2Z2NDWEpiQnpSZ2Q1bk56M3NBIn0.
    eyJleHAiOjE2NjMwMTQ0ODYsImlhdCI6MTY2MzAxNDQyNiwianRpIjoiNWFhMzAzMzItZTI0ZC00OTIyLWFjZTYtZWFiMWZlYzY2ZjRlIiwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDg4L3JlYWxtcy9tYXN0ZXIiLCJhdWQiOiJhY2NvdW50Iiwic3ViIjoiMDcwMTY1MzctMWE1YS00N2MzLWFjOTMtYmQ2MWQ3ZWQ2ZDU2IiwidHlwIjoiQmVhcmVyIiwiYXpwIjoiY2MiLCJzZXNzaW9uX3N0YXRlIjoiZTM0ZDA1YTgtZTIzZS00ZTBlLWIyYzYtMGIyYTZkNjk5MTM1IiwiYWNyIjoiMSIsInJlYWxtX2FjY2VzcyI6eyJyb2xlcyI6WyJkZWZhdWx0LXJvbGVzLW1hc3RlciIsIm9mZmxpbmVfYWNjZXNzIiwidW1hX2F1dGhvcml6YXRpb24iXX0sInJlc291cmNlX2FjY2VzcyI6eyJhY2NvdW50Ijp7InJvbGVzIjpbIm1hbmFnZS1hY2NvdW50IiwibWFuYWdlLWFjY291bnQtbGlua3MiLCJ2aWV3LXByb2ZpbGUiXX19LCJzY29wZSI6ImFwaTEg
    b2ZmbGluZV9hY2Nlc3MgcHJvZmlsZSBlbWFpbCIsInNpZCI6ImUzNGQwNWE4LWUyM2UtNGUwZS1iMmM2LTBiMmE2ZDY5OTEzNSIsImNsaWVudEhvc3QiOiIxNzIuMjMuMC4xIiwiY2xpZW50SWQiOiJjYyIsIm
    VtYWlsX3ZlcmlmaWVkIjpmYWxzZSwicHJlZmVycmVkX3VzZXJuYW1lIjoic2VydmljZS1hY2NvdW50LWNjIiwiY2xpZW50QWRkcmVzcyI6IjE3Mi4yMy4wLjEifQ.
    jbIbix4SRF1OwSR0JYy-v2KhaQV2-MTBD_yJnyetUL7mTl8apt789tfKiXrtTk7wFT-2w0p4pLyRWLr2r7MwHnASa6X8RNjxxCbaipeNk9w95CCvXJpq1ZdSQv-K32U5yc5O78i1OZ9ywqTulCs30hB-ySXCAV
    7Ue120c9rqepbRkOCJDjkmKCjEG0EGJ9Lnh-NwVo6Kxn_7OUcy-8pKnqH2TGR-SjCIZYDRplu9V3wow1ydHyBDwNhm_a3lGK2uSirUPiCuJg4hnLgt_EOBZmcRkmUNaVdJPcj3cPMCOhhCzNLzSSKZpMeYIEZS
    aL_dsFj-YaIjp1mah6RN9WE3uA",
  "expires_in": 60,
  "refresh_expires_in": 0,
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI4MDE1MzkzMS00NzRmLTQwNzktODc5MS0yY2RjNWNmZTg1ODAifQ.
    eyJpYXQiOjE2NjMwMTQ0MjYsImp0aSI6ImM3YTg4NGQxLThkNGUtNDc3MS05YjgyLWYyYjA3YmVkYmQwYiIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODA4OC9yZWFsbXMvbWFzdGVyIiwiYXVkIjoiaHR0cD
    ovL2xvY2FsaG9zdDo4MDg4L3JlYWxtcy9tYXN0ZXIiLCJzdWIiOiIwNzAxNjUzNy0xYTVhLTQ3YzMtYWM5My1iZDYxZDdlZDZkNTYiLCJ0eXAiOiJPZmZsaW5lIiwiYXpwIjoiY2MiLCJzZXNzaW9uX3N0YXRl
    IjoiZTM0ZDA1YTgtZTIzZS00ZTBlLWIyYzYtMGIyYTZkNjk5MTM1Iiwic2NvcGUiOiJhcGkxIG9mZmxpbmVfYWNjZXNzIHByb2ZpbGUgZW1haWwiLCJzaWQiOiJlMzRkMDVhOC1lMjNlLTRlMGUtYjJjNi0wYj
    JhNmQ2OTkxMzUifQ.Ms3Y-SUhx4P_D4Lwq3Zd_kIQzgjc-qPdTsn-VOMgfUs",
  "token_type": "Bearer",
  "not-before-policy": 0,
  "session_state": "e34d05a8-e23e-4e0e-b2c6-0b2a6d699135",
  "scope": "api1 offline_access profile email"
}
```


commit

El token


```
{
  "access_token": "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICI
      eyJleHAiOjE2NjMwMTQ0ODYsImlhdCI6MTY2MzAxNDQyNiwianRpIjoiNWFhMzAz
      N0ZXIiLCJhdWQiOiJhY2NvdW50Iiwic3ViIjoiMDcwMTY1MzctMWE1YS00N2MzLW
      ZTIzZS00ZTBlLWIyYzYtMGIyYTZkNjk5MTM1IiwiYWNyIjoiMSIsInJlYWxtX2Fj
      Rpb24iXX0sInJlc291cmNlX2FjY2VzcyI6eyJhY2NvdW50Ijp7InJvbGVzIjpbIm
      b2ZmbGluZV9hY2Nlc3MgcHJvZmlsZSBlbWFpbCIsInNpZCI6ImUzNGQwNWE4LWUy
      VtYWlsX3ZlcmlmaWVkIjpmYWxzZSwicHJlZmVycmVkX3VzZXJuYW1lIjoic2Vydm
      jbIbix4SRF1OwSR0JYy-v2KhaQV2-MTBD_yJnyetUL7mTl8apt789tfKiXrtTk7w
      7Ue120c9rqepbRkOCJDjkmKCjEG0EGJ9Lnh-NwVo6Kxn_7OUcy-8pKnqH2TGR-Sj
      aL_dsFj-YaIjp1mah6RN9WE3uA",
  "expires_in": 60,
  "refresh_expires_in": 0,
  "refresh_token": "eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6IC
      eyJpYXQiOjE2NjMwMTQ0MjYsImp0aSI6ImM3YTg4NGQxLThkNGUtNDc3MS05Yjgy
      ovL2xvY2FsaG9zdDo4MDg4L3JlYWxtcy9tYXN0ZXIiLCJzdWIiOiIwNzAxNjUzNy
      IjoiZTM0ZDA1YTgtZTIzZS00ZTBlLWIyYzYtMGIyYTZkNjk5MTM1Iiwic2NvcGUi
      JhNmQ2OTkxMzUifQ.Ms3Y-SUhx4P_D4Lwq3Zd_kIQzgjc-qPdTsn-VOMgfUs",
  "token_type": "Bearer",
  "not-before-policy": 0,
  "session_state": "e34d05a8-e23e-4e0e-b2c6-0b2a6d699135",
  "scope": "api1 offline_access profile email"
}
```


- Access token: Token de Acceso
- Expires: Segundos de expiración
- Scope: Accesos / recursos
- Token type: Cómo se envía el token

commit

Ejemplo Client Credentials

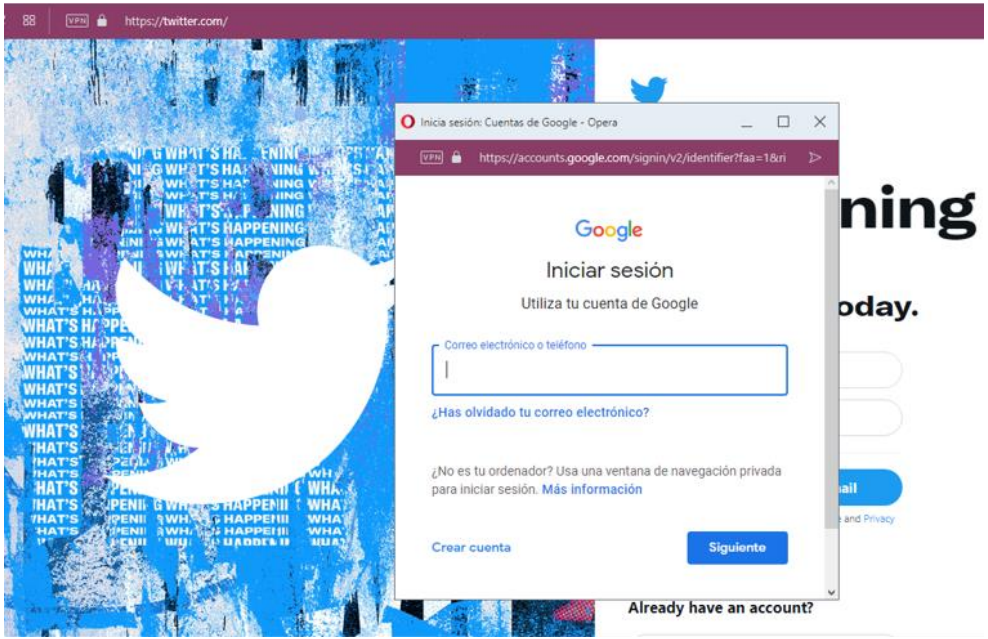
commit

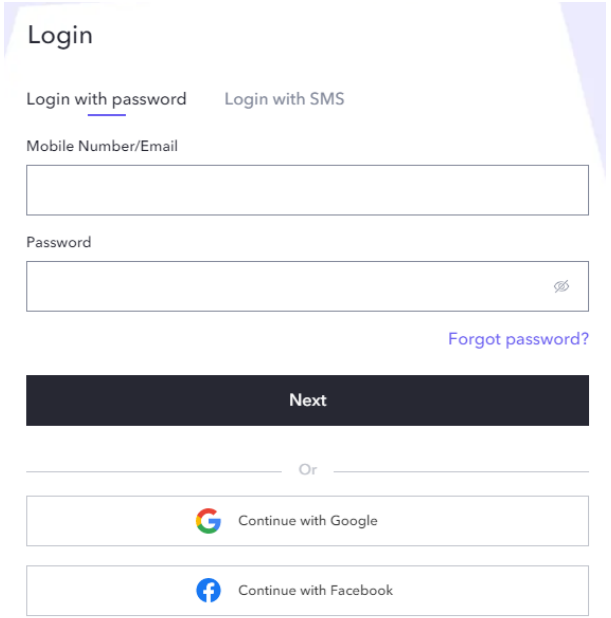
Flujos / Grants

Cómo interactúa el cliente para obtener un token

commit

¿Y cómo funciona la autorización con Google?





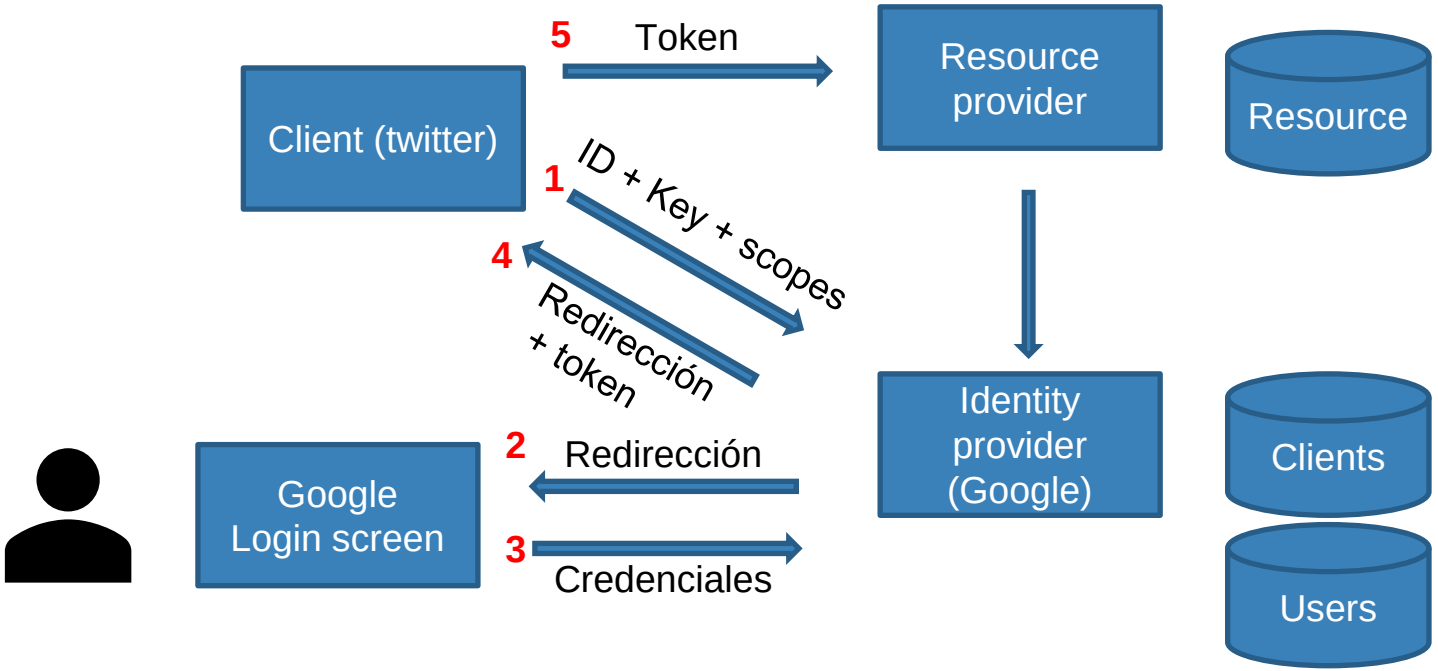
commit

¿Y cómo funciona la autorización con Google?

- Solicita nuestros datos (email, nombre) y nos registra en su sistema
- Yo doy acceso a sistema externo (cliente)
- El cliente es conocido previamente por el IDP
- El cliente establece previamente el scope (en este caso nuestro email, nombre, etc.)
- Nosotros somos resource owner (interacción del usuario)
- Siempre es necesario un navegador

commit

Flow con interacción del usuario



commit

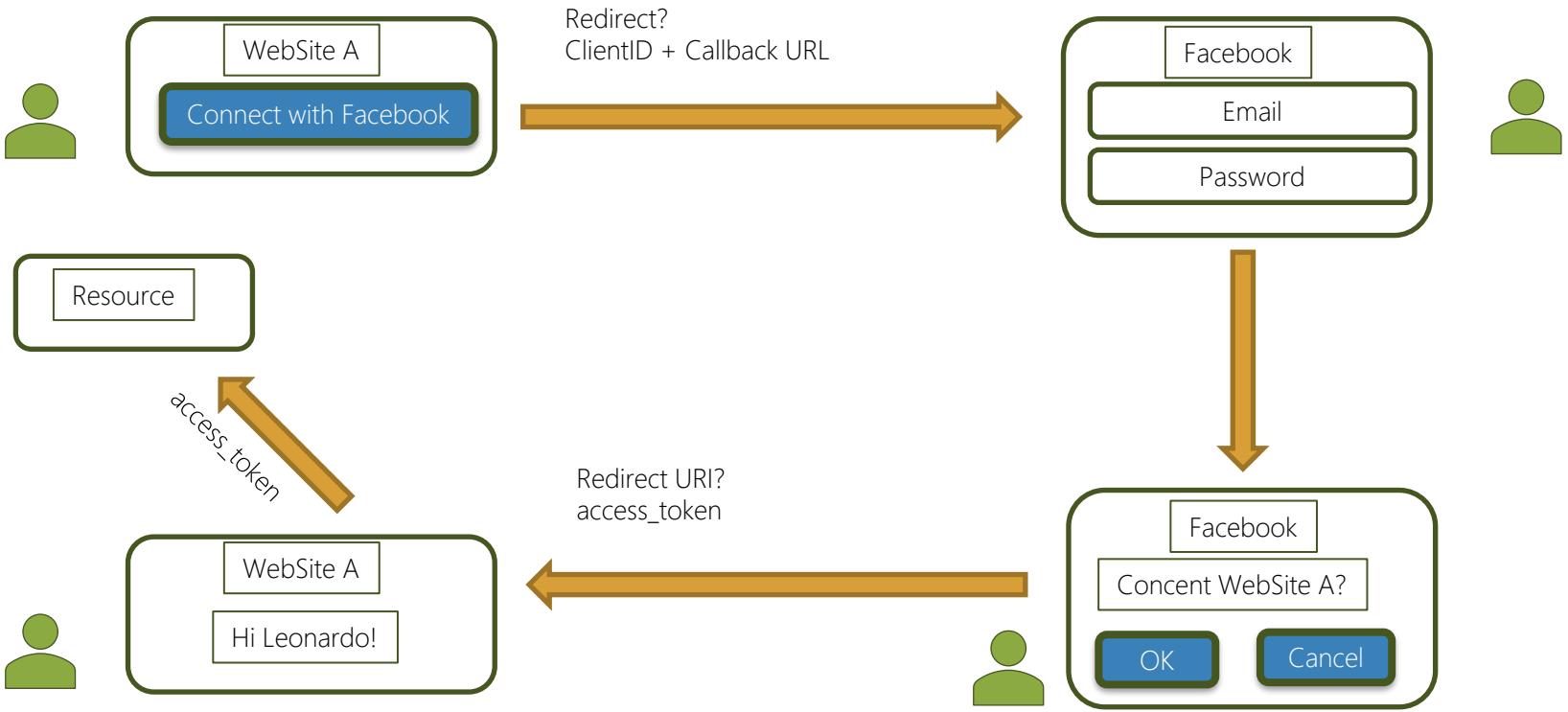
oAuth flow / grants

“Los flujos o grants son las formas en que los clientes interactúan con el IDP para gestionar tokens”

- Client credentials – Robots
- Implicit – Web SPA
- Code [PKCE] – Web
- Device – Devices
- Resource owner – Legacy
- Hybrid

commit

Implicit flow

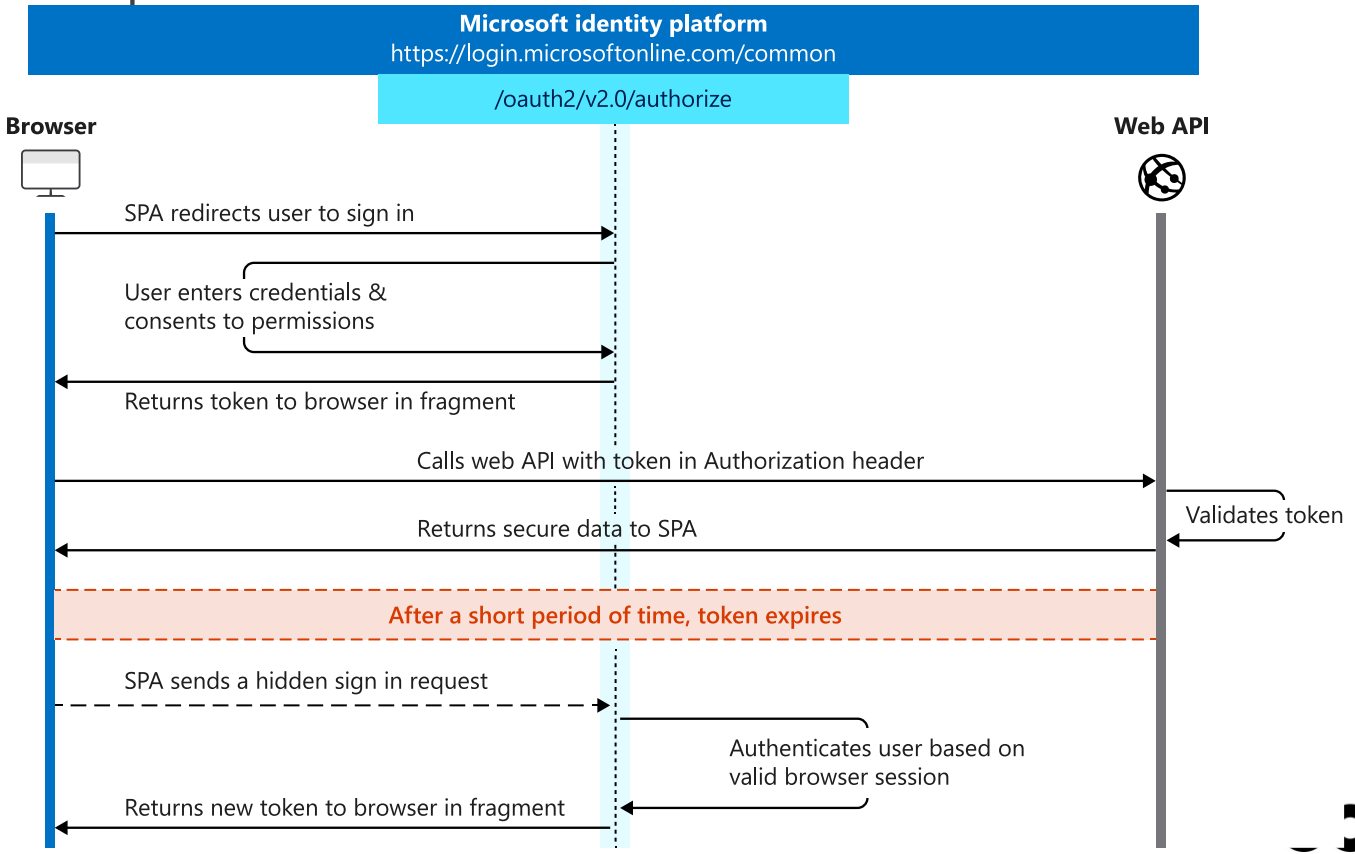


commit

Ejemplo Implicit flow

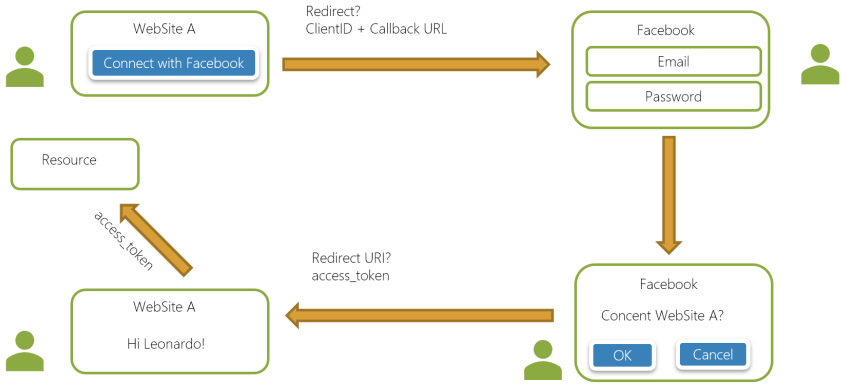
commit

oAuth Implicit flow



Características

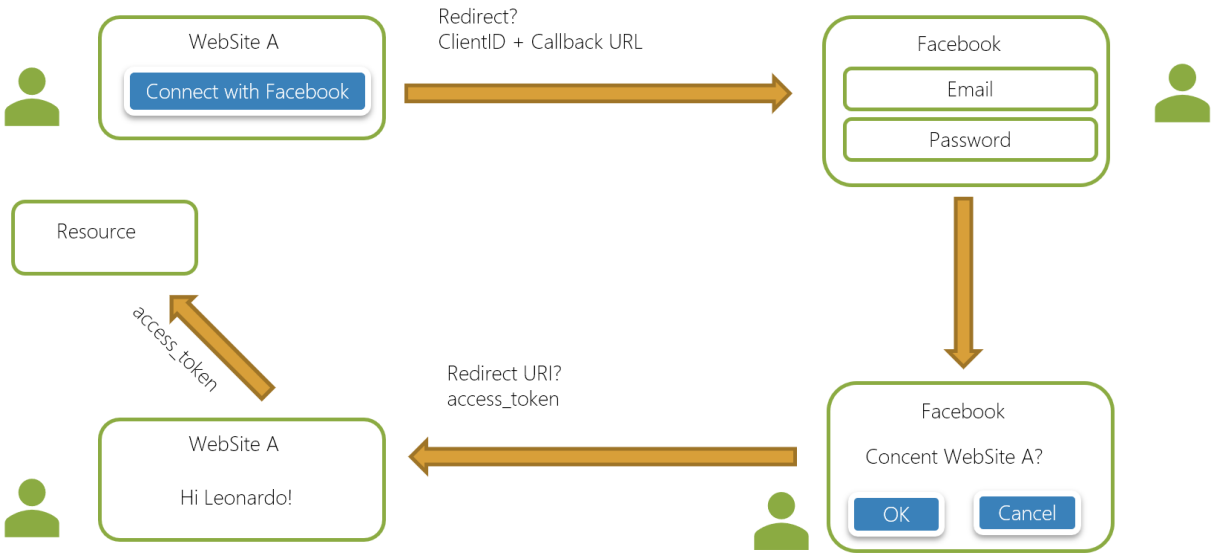
- El usuario ingresa sus credenciales en el sitio de confianza
- El cliente se encuentra registrado
- Se puede mostrar un “consent” para autorizar los acceso a scopes
- Se utilizan redirecciones (web)
- callback_url es un dato muy importante
- Todo ocurre en el frontend



commit

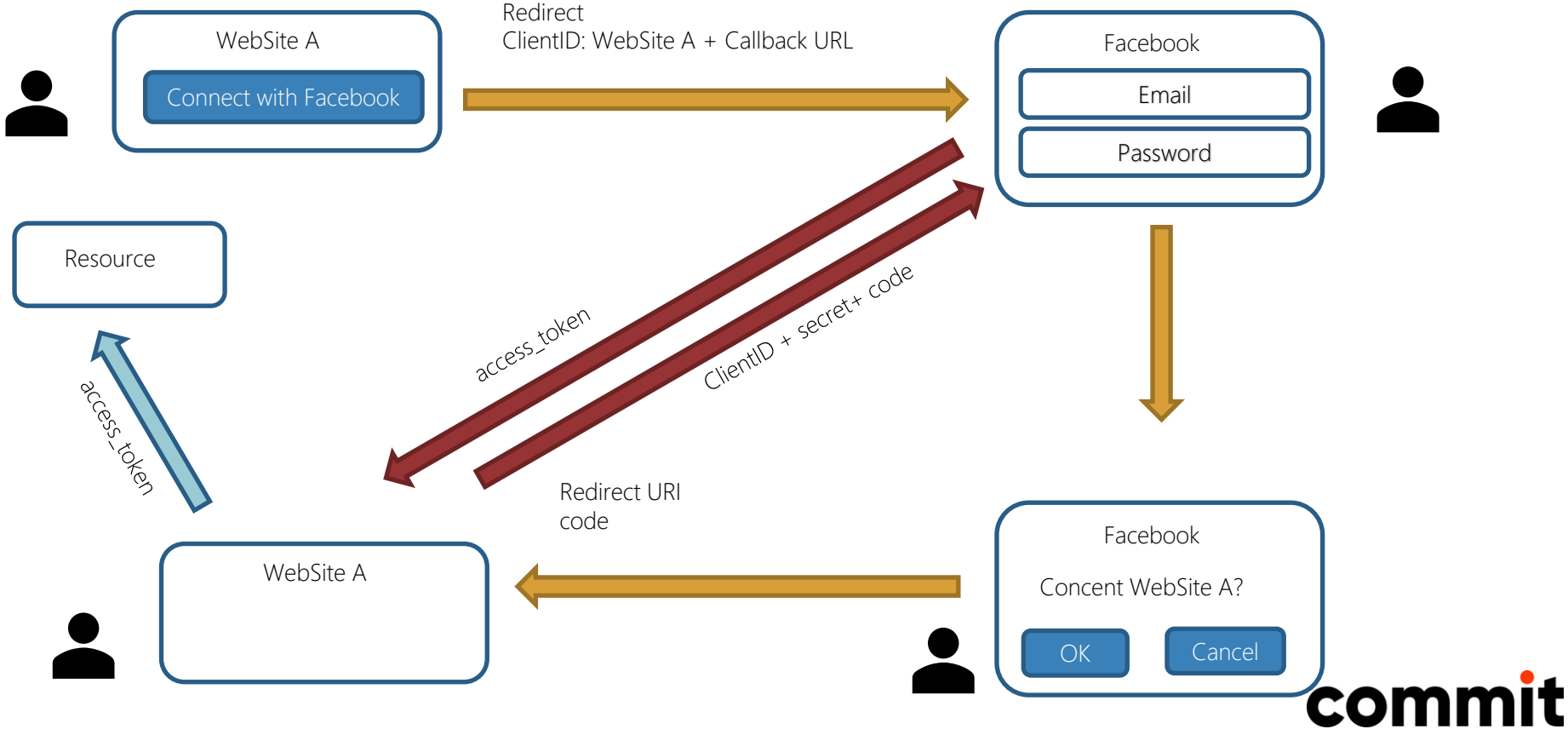
Problemas

- Es posible interceptar el token
- El Token puede quedar almacenado en el browser

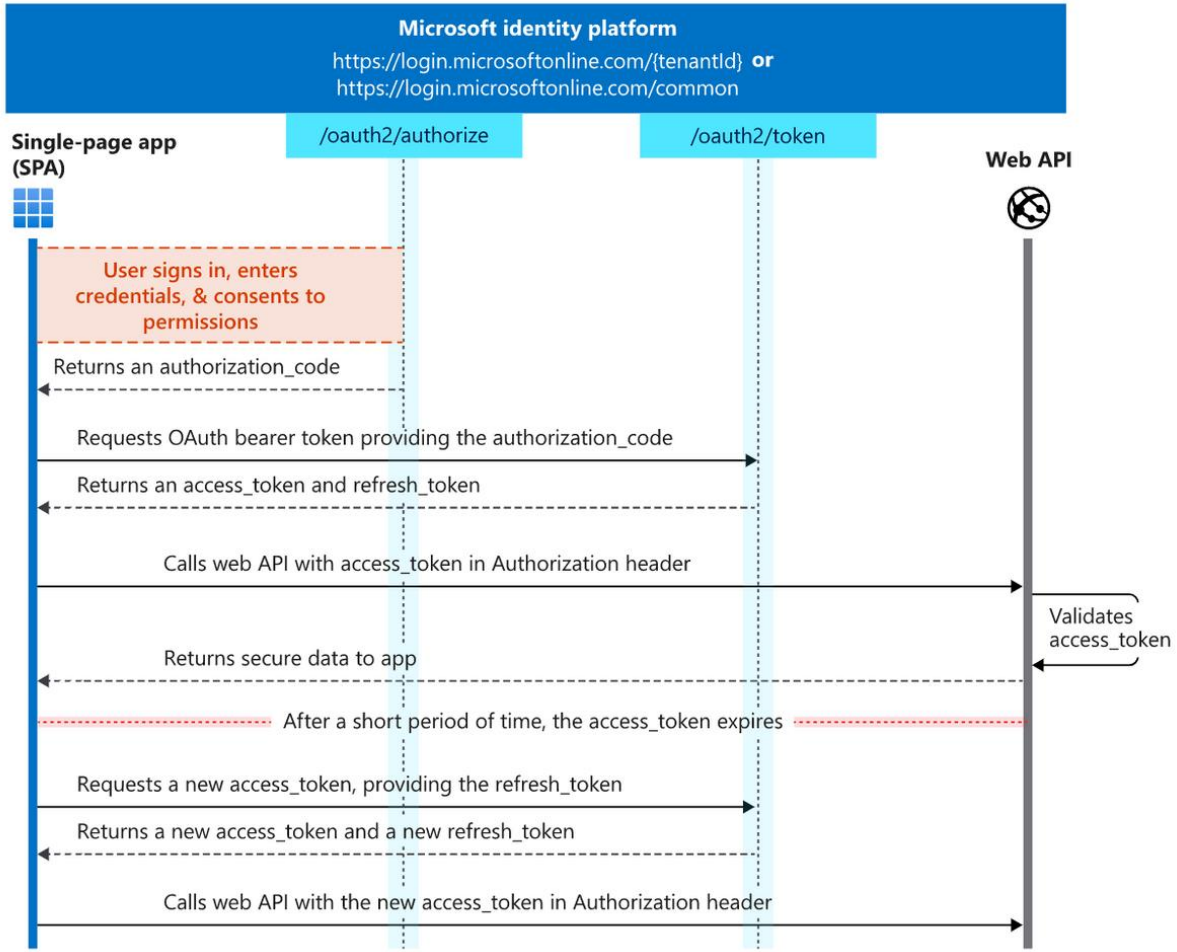


commit

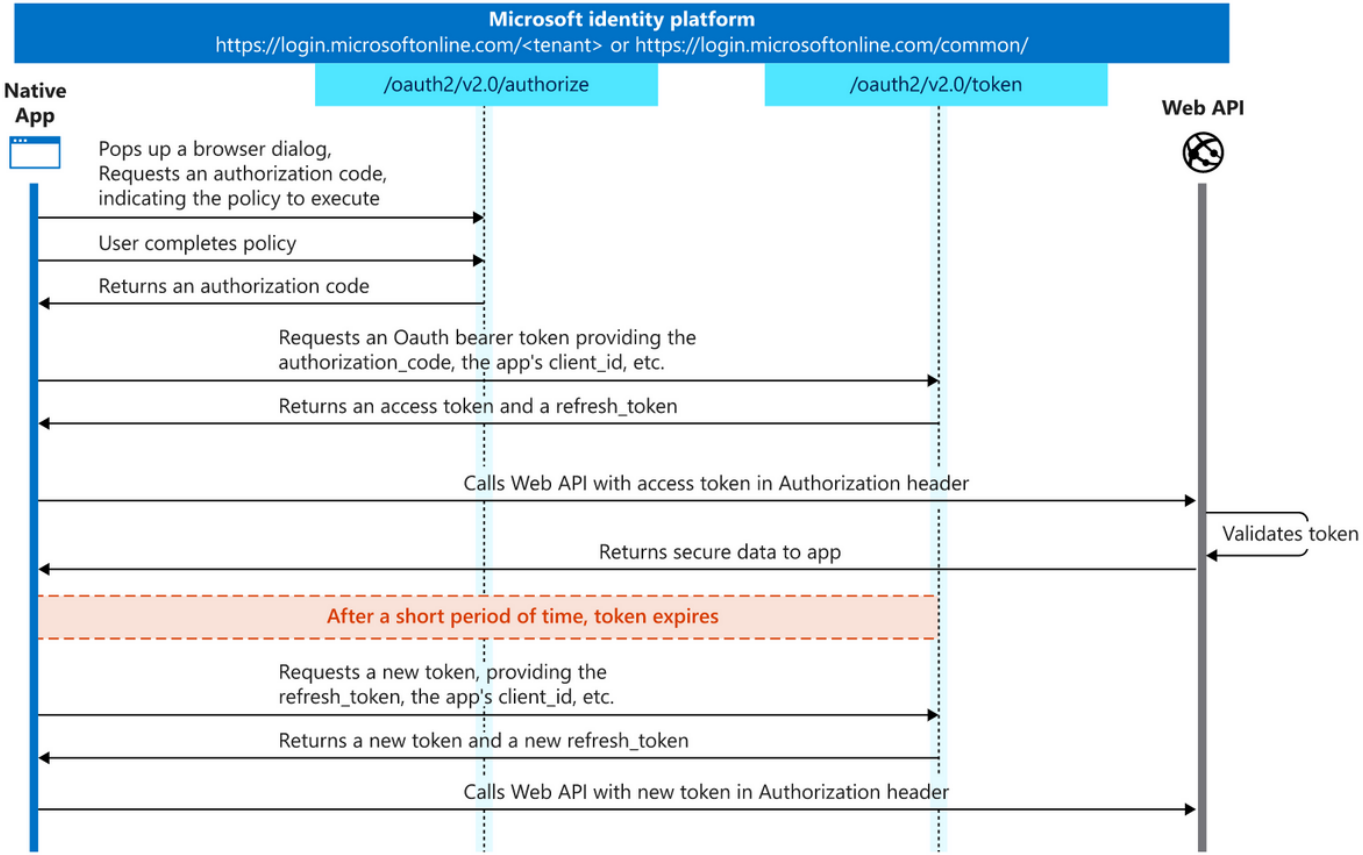
Solucionando problemas del implicit Flow: code flow



commit

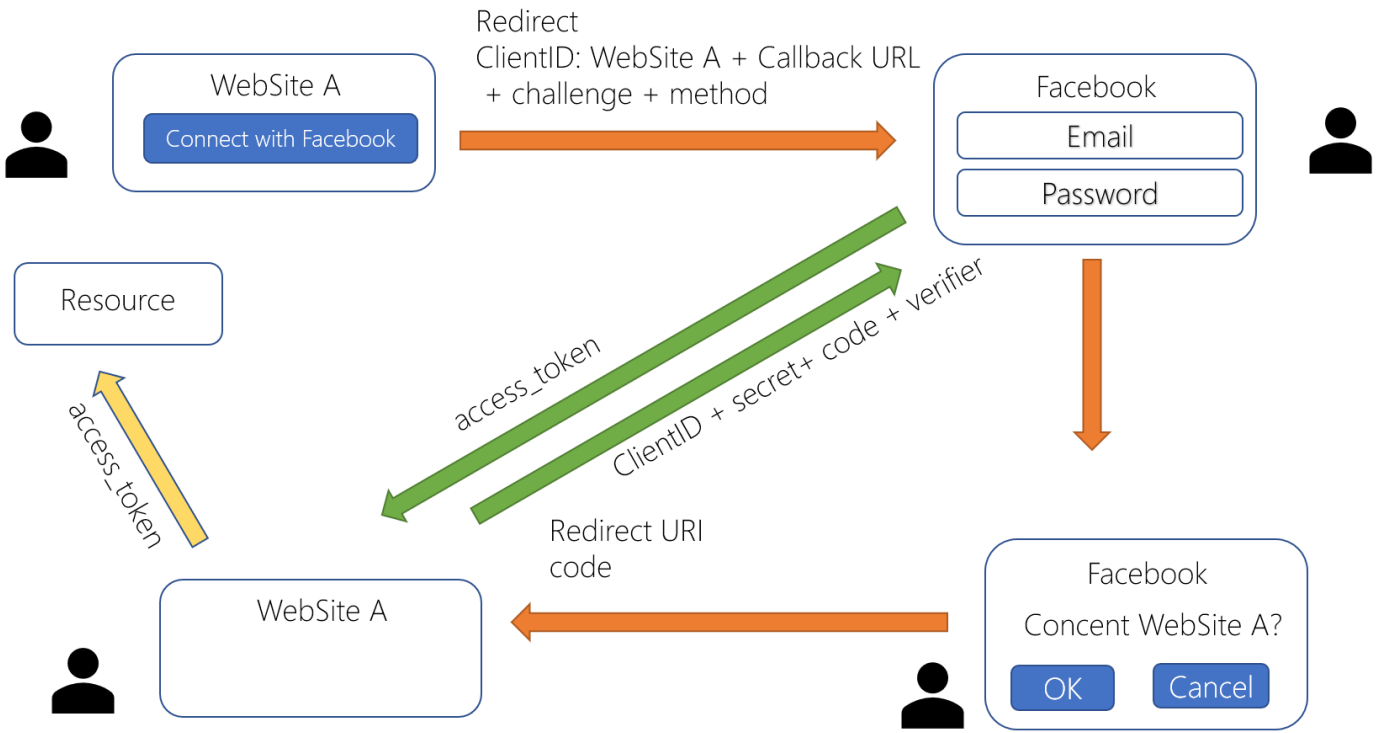

Microsoft identity platform
https://login.microsoftonline.com/{tenantId} or
https://login.microsoftonline.com/common
/oauth2/authorize
/oauth2/token
Single-page app (SPA)
Web API
User signs in, enters credentials, & consents to permissions
Returns an authorization_code
Requests OAuth bearer token providing the authorization_code
Returns an access_token and refresh_token
Calls web API with access_token in Authorization header
Validates access_token
Returns secure data to app
After a short period of time, the access_token expires
Requests a new access_token, providing the refresh_token
Returns a new access_token and a new refresh_token
Calls web API with the new access_token in Authorization header

commit


Microsoft identity platform
https://login.microsoftonline.com/<tenant> or https://login.microsoftonline.com/common/
/oauth2/v2.0/authorize
/oauth2/v2.0/token
Native App
Web API
Pops up a browser dialog,
Requests an authorization code,
indicating the policy to execute
User completes policy
Returns an authorization code
Requests an Oauth bearer token providing the
authorization_code, the app's client_id, etc.
Returns an access token and a refresh_token
Calls Web API with access token in Authorization header
Validates token
Returns secure data to app
After a short period of time, token expires
Requests a new token, providing the
refresh_token, the app's client_id, etc.
Returns a new token and a new refresh_token
Calls Web API with new token in Authorization header

commit

PCKE: Más seguridad a authorization code flow



commit

Open ID Connect

Agregando autenticación a oAuth

commit

@leomicheloni

OIDC

OpenID Connect (**OIDC**) is an authentication layer on top of OAuth 2.0, an authorization framework.[1] The standard is controlled by the OpenID Foundation.

OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol, which allows computing clients to verify the identity of an end-user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner. In technical terms, OpenID Connect specifies a RESTful HTTP API, using JSON as a data format.

commit

¿Cómo funciona OIDC?

Ben pocas palabras, utilizando oAuth como base agregar scopes que representan elementos de la identidad del usuario

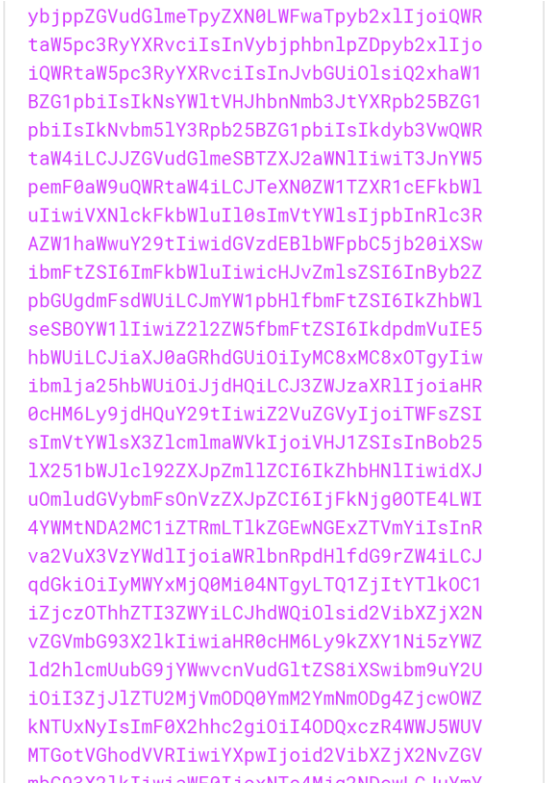
- Name
- Surname
- Addess
- Etc.

Además define un nuevo endpoint (user_info) para obtener un nuevo token (id_token) con esta información de identidad

Y algunas cosas…

commit

El Id_token



```
ybjppZGVudGlmeTpyZXN0LWFwaTpyb2xlIjoiQWR
taW5pc3RyYXRvciIsInVybjphbnlpZDpyb2xlIjo
iQWRtaW5pc3RyYXRvciIsInJvbGUiOlsiQ2xhaW1
BZG1pbiIsIkNsYWltVHJhbnNmb3JtYXRpb25BZG1
pbiIsIkNvbm5lY3Rpb25BZG1pbiIsIkdyb3VwQWR
taW4iLCJJZGVudGlmeSBTZXJ2aWNlIiwiT3JnYW5
pemF0aW9uQWRtaW4iLCJTeXN0ZW1TZXR1cEFkbWl
uIiwiVXNlckFkbWluIl0sImVtYWlsIjpbInRlc3R
AZW1haWwuY29tIiwidGVzdEBlbWFpbC5jb20iXSw
ibmFtZSI6ImFkbWluIiwicHJvZmlsZSI6InByb2Z
pbGUgdmFsdWUiLCJmYW1pbHlfbmFtZSI6IkZhbWl
seSBOYW1lIiwiZ2l2ZW5fbmFtZSI6IkdpdmVuIE5
hbWUiLCJiaXJ0aGRhdGUiOiIyMC8xMC8xOTgyIiw
ibmlja25hbWUiOiJjdHQiLCJ3ZWJzaXRlIjoiaHR
0cHM6Ly9jdHQuY29tIiwiZ2VuZGVyIjoiTWFsZSI
sImVtYWlsX3ZlcmlmaWVkIjoiVHJ1ZSIsInBob25
lX251bWJlcl92ZXJpZmllZCI6IkZhbHNlIiwidXJ
uOmludGVybmFsOnVzZXJpZCI6IjFkNjg0OTE4LWI
4YWMtNDA2MC1iZTRmLTlkZGEwNGExZTVmYiIsInR
va2VuX3VzYWdlIjoiaWRlbnRpdHlfdG9rZW4iLCJ
qdGkiOiIyMWYxMjQ0Mi04NTgyLTQ1ZjItYTlkOC1
iZjczOThhZTI3ZWYiLCJhdWQiOlsid2VibXZjX2N
vZGVmbG93X2lkIiwiaHR0cHM6Ly9kZXY1Ni5zYWZ
ld2hlcmUubG9jYWwvcnVudGltZS8iXSwibm9uY2U
iOiI3ZjJlZTU2MjVmODQ0YmM2YmNmODg4ZjcwOWZ
kNTUxNyIsImF0X2hhc2giOiI4ODQxczR4WWJ5WUV
MTGotVGhodVVRIiwiYXpwIjoid2VibXZjX2NvZGV
```


```
{
  "sub": "admin",
  "uiId": "427a74fb-7e47-4bed-b92e-0d09e6f1b479",
  "unique_name": "admin",
  "auth_time": 1578286467,
  "urn:identify:rest-api:role": "Administrator",
  "urn:anyid:role": "Administrator",
  "role": [
    "ClaimAdmin",
    "ClaimTransformationAdmin",
    "ConnectionAdmin",
    "GroupAdmin",
    "Identify Service",
    "OrganizationAdmin",
    "SystemSetupAdmin",
    "UserAdmin"
  ],
  "email": [
    "test@email.com",
    "test@email.com"
  ],
  "name": "admin",
  "profile": "profile value",
  "family_name": "Family Name",
  "given_name": "Given Name",
  "birthdate": "20/10/1982",
  "nickname": "ctt",
  "website": "https://ctt.com",
  "gender": "Male",
  "email_verified": "True",
  "phone_number_verified": "False",
  "urn:internal:userid": "1d684918-b8ac-4060-be4f-
9dda04a1e5fb",
  "token_usage": "identity_token",
  "jti": "21f12442-8582-45f2-a9d8-bf7398ae27ef",
  "aud": [
    "webmvc_codeflow_id",
    "https://dev56.safewhere.local/runtime/"
```

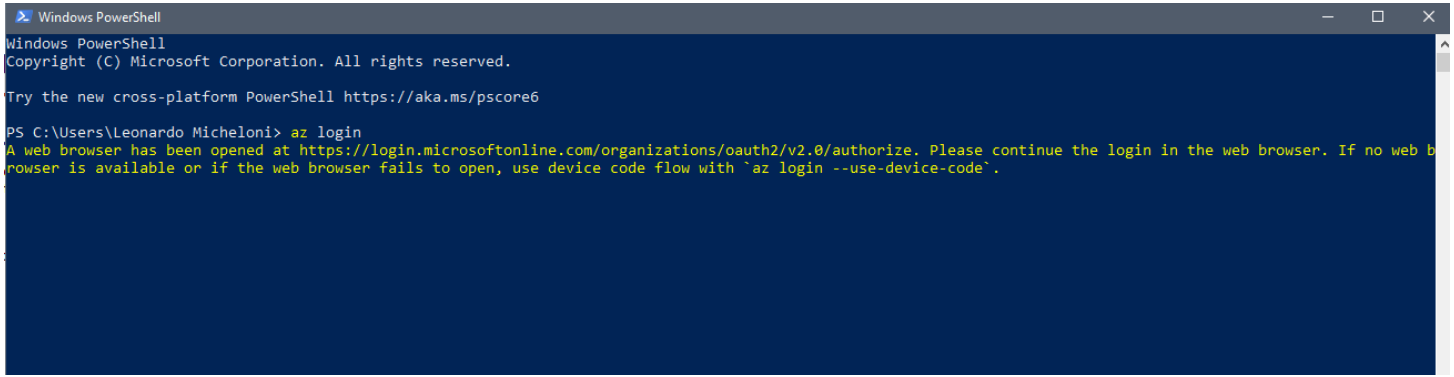

commit

Ejemplo OIDC

commit

Otros flujos

- Resource owner: Para aplicaciones legacy o sin acceso a un navegador (user y password se envían al Auth Server)
- Device: Para dispositivos donde necesitamos que el usuario haga login pero el dispositivo no tiene acceso a un browser (ej. az login)



commit

¿Preguntas?

Referencias

- [JWT.io](https://jwt.io/)
- [oAuth](https://oauth.net)
- [Keycloack](https://www.keycloak.org/)
- [oAuth Playground](https://www.oauth.com/playground)

commit

¡Gracias!

commit