CC P2: Introduction to OAuth 2

Mario Ventura

Índice

1. Qué es OAuth 2
2. Propósito
3. Cómo funciona (roles)
4. En qué contextos se usa
5. Pros vs Cons

¿Qué es OAuth 2?

- Open Authentication
- Marco (estándar) de autorización
- Permite conceder acceso limitado sin necesidad de credenciales

Propósito

Pretende facilitar el acceso a recursos protegidos

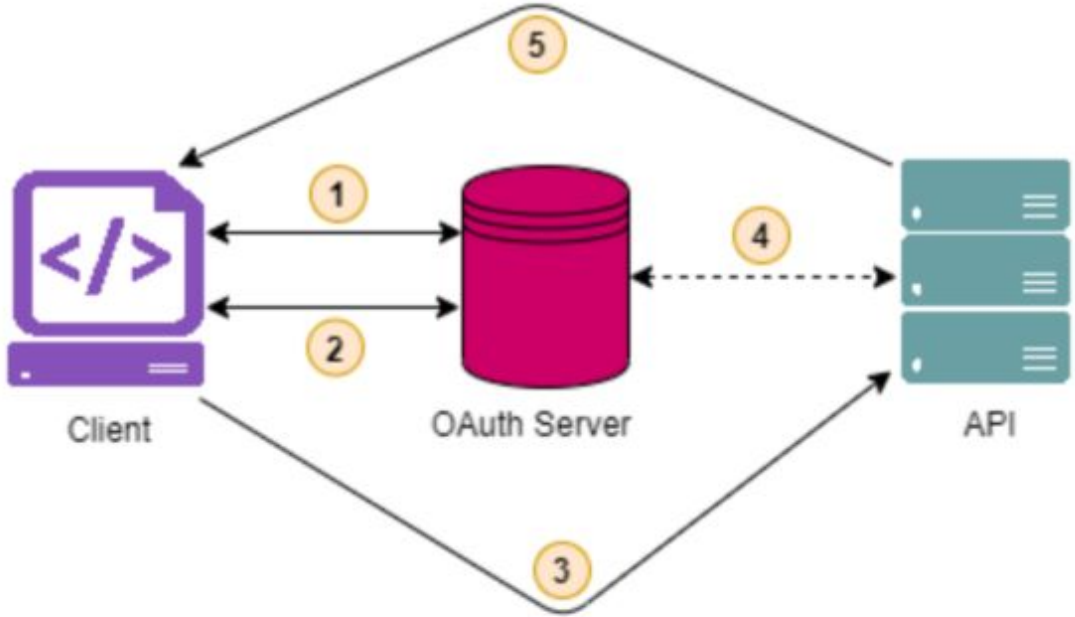
- Mejora de seguridad
- Mejora de UX

Cómo funciona

Sistema de 4 roles

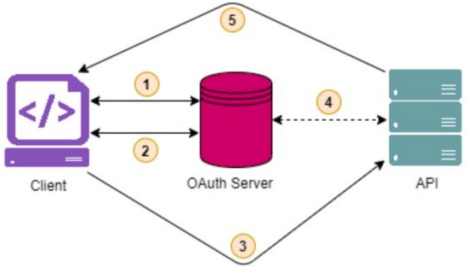
- **Propietario del recurso:** Posee el recurso
- **Cliente:** Solicita acceso a un recurso
- **Servidor del recurso:** Contiene el recurso protegido y recibe tokens de acceso
- **Servidor de autorización:** Autentica al cliente y otorga tokens de acceso

Cómo funciona



[1]

Cómo funciona



1. La solicitud de autorización se envía desde el cliente al servidor OAuth.
2. El token de acceso se devuelve al cliente.
3. El token de acceso se envía desde el cliente al servicio API (que actúa como servidor de recursos) en cada solicitud de acceso a un recurso protegido.
4. El servidor de recursos comprueba el token con el servidor OAuth, para confirmar que el cliente está autorizado a consumir dicho recurso.
5. El servidor responde con los recursos protegidos solicitados.

¿Por qué es seguro?

EJEMPLO

Cuando inicias sesión en una web usando "Iniciar sesión con Google", la web nunca ve tu contraseña.

¿Por qué es seguro?

EJEMPLO

Cuando inicias sesión en una web usando "Iniciar sesión con Google", la web nunca ve tu contraseña.

Esa”web”solo ve tu token de acceso, pero **nunca ve tus credenciales**

¿En qué contextos se usa?

Casos reales:

- Facebook
- Github
- Twitter
- Steam
- BitBucket
- etc.

[2]

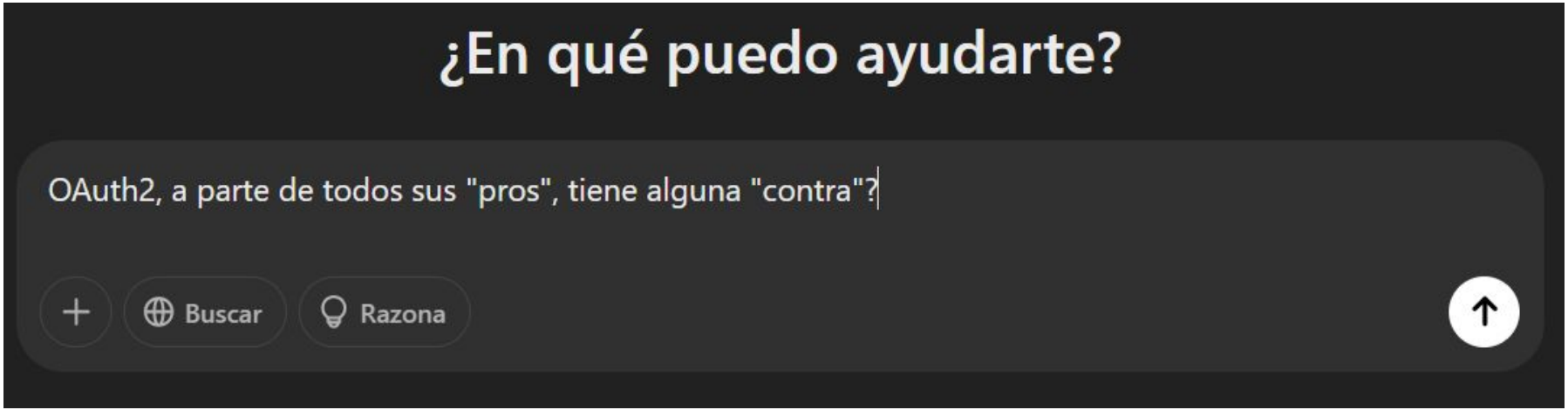
Beneficios

Beneficios

- Mejor seguridad
- Se mantiene el control de acceso

Contras

Hay algún “contra”?



Contras

Contras

- **Complejidad de Implementación**
- **Dependencia de servicios de terceros**

- Tokens con vida útil larga pueden suponer un riesgo si se filtran
- No autentica por sí mismo
- No es compatible con OAuth 1.0 (uso de firmas y claves privadas)

Gracias

Fuentes

[1] Autenticando servicios REST con OAuth2

[2] Explicación del protocolo OAuth 2

[3] Should You Use OAuth 2.0? Pros and Cons