Unidad 9
Aplicaciones

Sistemas Distribuidos

Universidad Nacional de Asunción, Facultad Politécnica

Ingeniería Informática

Ing. Fernando Mancía

OAuth2 *

- El marco de trabajo (framework) de autorización OAuth 2.0 permite que una aplicación de terceros obtenga acceso limitado a un servicio HTTP, ya sea en nombre del propietario de un recurso organizando una interacción de aprobación entre el propietario del recurso y el servicio HTTP, o permitiendo que la aplicación de terceros obtenga acceso en su propio nombre.

- Es un protocolo de autorización que permite a terceros (clientes) acceder a contenidos propiedad de un usuario (alojados en aplicaciones de confianza, servidor de recursos) sin que éstos tengan que manejar ni conocer las credenciales del usuario. Es decir, aplicaciones de terceros pueden acceder a contenidos propiedad del usuario, pero estas aplicaciones no conocen las credenciales de autenticación

* https://datatracker.ietf.org/doc/html/rfc6749
* https://oauth.net/2/

OAuth2 define 4 roles

Propietario del recurso
Una entidad capaz de otorgar acceso a un recurso protegido
Cuando el propietario del recurso es una persona, se lo denomina usuario final.

Servidor de recursos
El servidor que aloja los recursos protegidos, capaz de aceptar y responder a solicitudes de recursos protegidos mediante tokens de acceso.

Cliente
Una aplicación que realiza solicitudes de recursos protegidos en nombre del propietario del recurso y con su autorización.
El término "cliente" no implica ninguna característica de implementación particular (por ejemplo, si la aplicación se ejecuta en un servidor, un escritorio u otros dispositivos).

Servidor de autorización
El servidor que emite tokens de acceso al cliente después de autenticar con éxito al propietario del recurso y obtener la autorización.

OAuth2 - Flujo

1.2. Protocol Flow

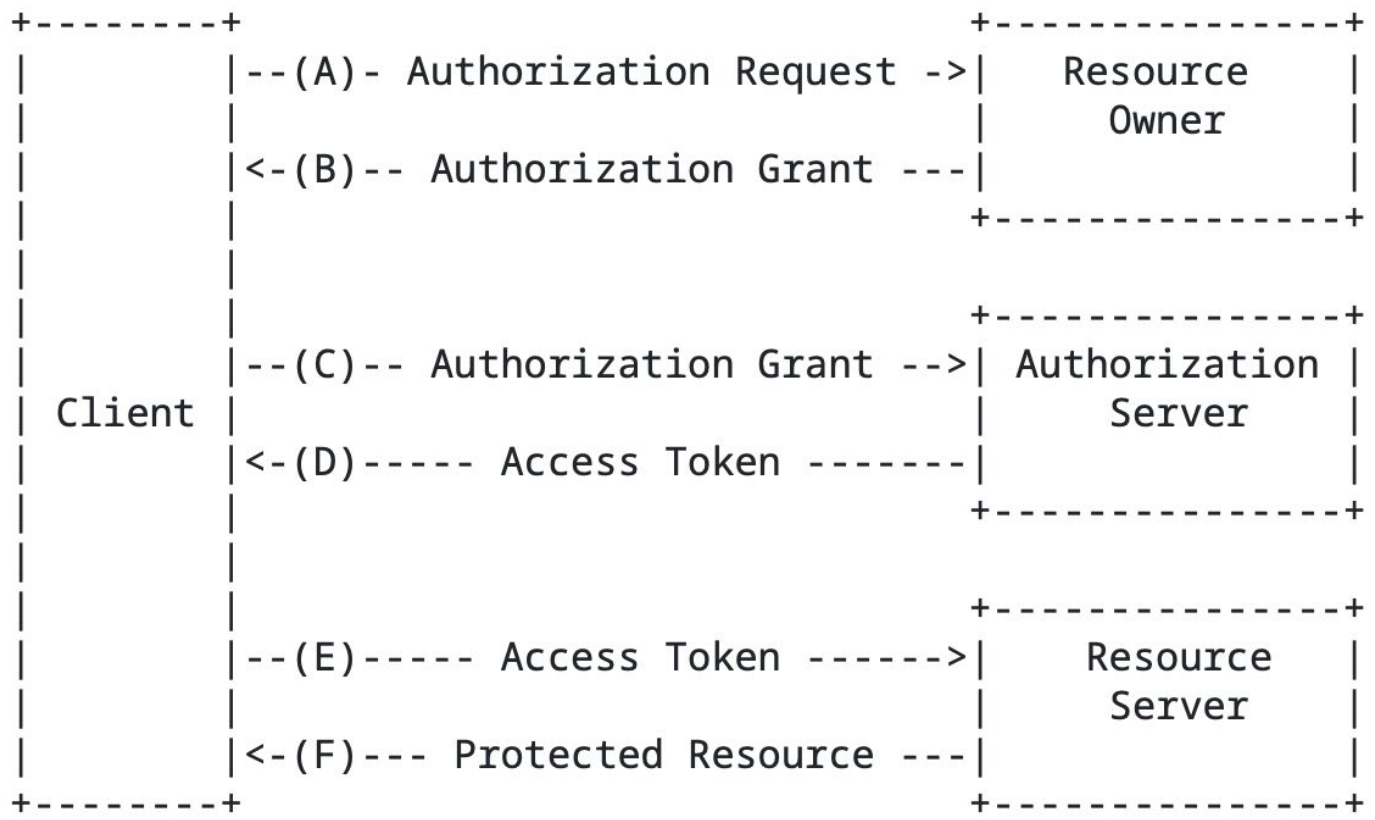


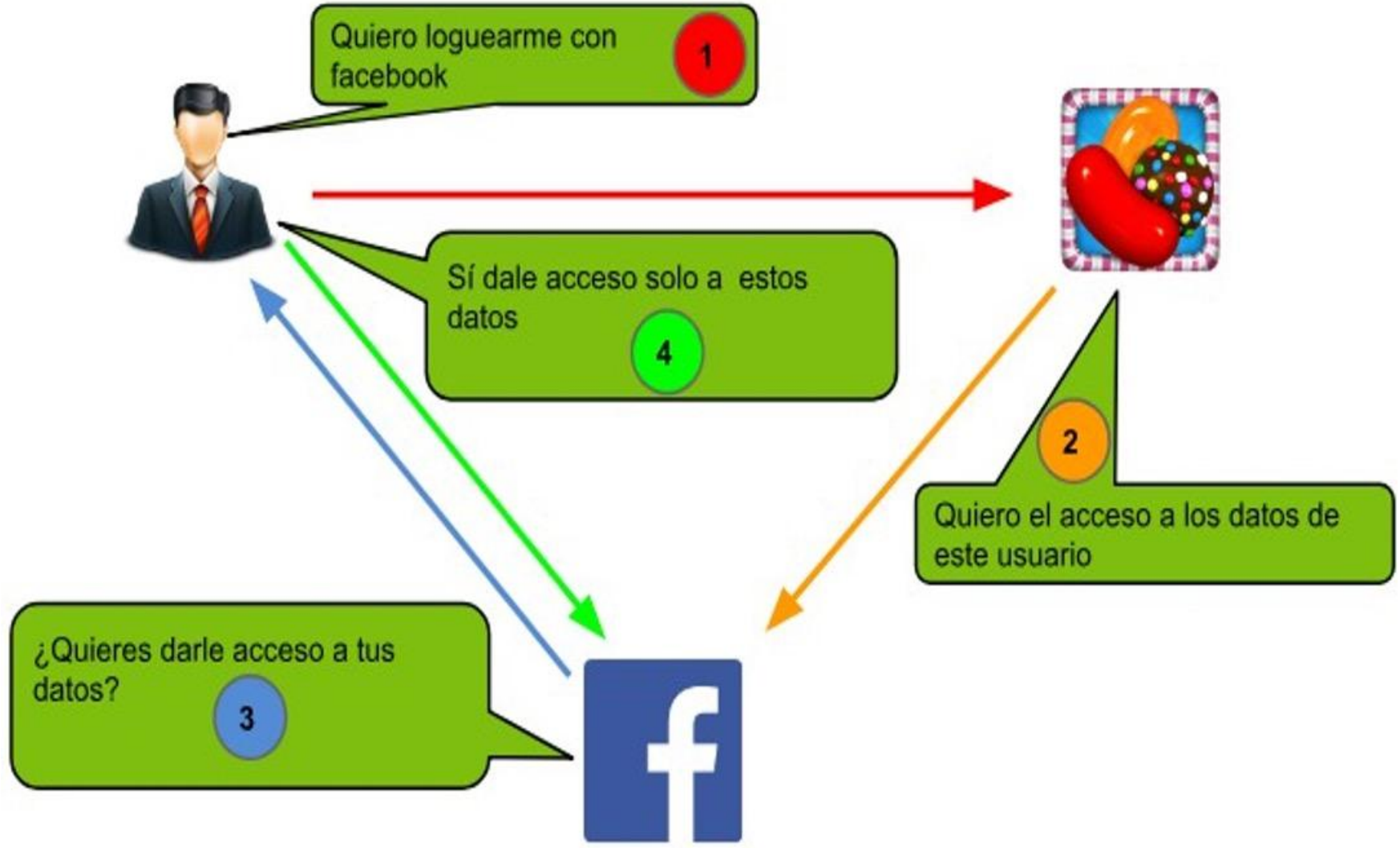
Figure 1: Abstract Protocol Flow

La interacción entre el servidor de autorización y el servidor de recursos está fuera del alcance de esta especificación.

El servidor de autorización puede ser el mismo servidor que el servidor de recursos o una entidad separada.

Un solo servidor de autorización puede emitir tokens de acceso aceptados por varios servidores de recursos.

OAuth2 - Flujo



JSON Web Token (JWT)

JSON Web Token (JWT) es un medio compacto y seguro para URL de representar reclamaciones que se transferirán entre dos partes. Las reclamaciones en un JWT se codifican como un objeto JSON que se utiliza como carga útil de una estructura de Firma Web JSON (JWS) o como texto sin formato de una estructura de Cifrado Web JSON (JWE), lo que permite que las reclamaciones se firmen digitalmente o se proteja su integridad con un Código de Autenticación de Mensajes (MAC) y/o se cifren.

Los JSON Web Token son un método abierto, estándar de la industria RFC 7519 para representar reclamaciones de forma segura entre dos partes.

https://datatracker.ietf.org/doc/html/rfc7519

JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens

Esta especificación define un perfil para emitir tokens de acceso OAuth 2.0 en formato JSON Web Token (JWT). Los servidores de autorización y los servidores de recursos de diferentes proveedores pueden aprovechar este perfil para emitir y consumir tokens de acceso de manera interoperable

https://datatracker.ietf.org/doc/html/rfc9068

JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens

```
GET /as/authorization.oauth2?response_type=code
        &client_id=s6BhdRkqt3
        &state=xyz
        &scope=openid%20profile%20reademail
        &redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
        &resource=https%3A%2F%2Frs.example.com%2F HTTP/1.1
     Host: authorization-server.example.com
```

Figure 1: Authorization Request with Resource and Scope Parameters

Once redeemed, the code obtained from the request above will result in a JWT access token in the form shown below:

Header:

```
{"typ":"at+JWT","alg":"RS256","kid":"RjEwOwOA"}
```

Claims:

```
{
  "iss": "https://authorization-server.example.com/",
  "sub": "5ba552d67",
  "aud":   "https://rs.example.com/",
  "exp": 1639528912,
  "iat": 1618354090,
  "jti" : "dbe39bf3a3ba4238a513f51d6e1691c4",
  "client_id": "s6BhdRkqt3",
  "scope": "openid profile reademail"
}
```

Figure 2: The Header and JWT Claims Set of a JWT Access Token