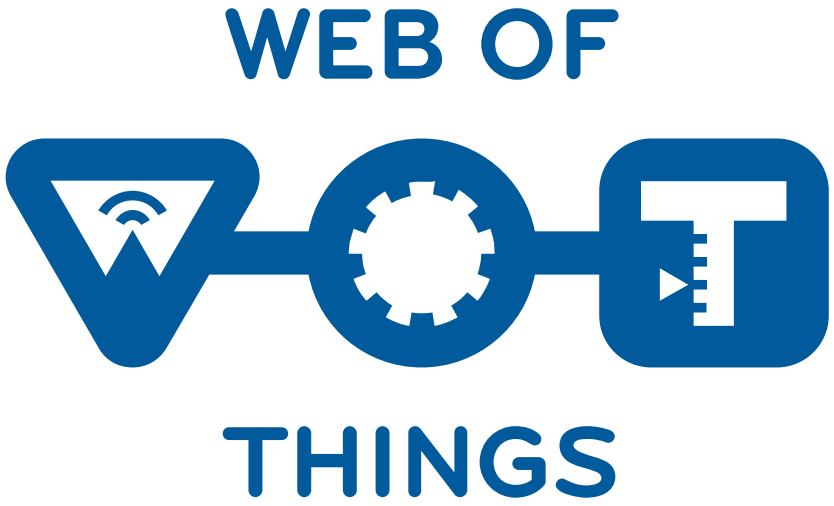


WoT and OAuth 2.0

Cristiano Aguzzi

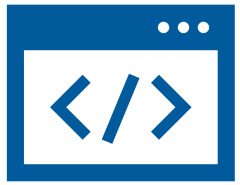
20/10/2020

Outline

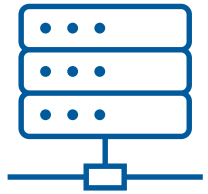
- OAuth 2.0 overview
- OAuth 2.0 flows
  - Code flow
    - Code – WoT scenarios
  - Device flow
    - Device – WoT scenarios
  - Client flow
    - Client – WoT scenarios
- Open points & discussion

OAuth 2.0

OAuth 2.0 is an **authorization protocol** widely known for its usage across several web services. It enables third-party applications to obtain **limited access** to HTTP(s) services on behalf of the **resource owner** or of itself.



Client



Authorization server

Resource server

Resource owner

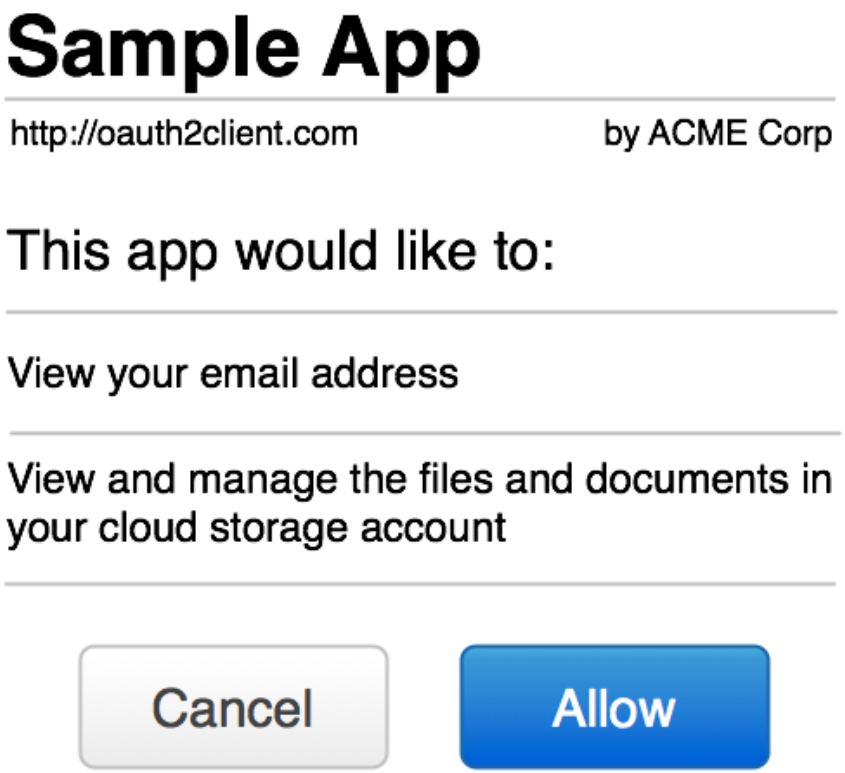
OAuth 2.0

```
+--------+                               +---------------+
|        |--(A)- Authorization Request ->|   Resource    |
|        |                               |     Owner     |
|        |<-(B)-- Authorization Grant ---|               |
|        |                               +---------------+
|        |
|        |                               +---------------+
|        |--(C)-- Authorization Grant -->| Authorization |
| Client |                               |     Server    |
|        |<-(D)----- Access Token -------|               |
|        |                               +---------------+
|        |
|        |                               +---------------+
|        |--(E)----- Access Token ------>|    Resource   |
|        |                               |     Server    |
|        |<-(F)--- Protected Resource ---|               |
+--------+                               +---------------+
```

OAuth 2.0 Scopes

Scope is a mechanism in OAuth 2.0 to limit an application's access to a user's account.

An application can request **one or more scopes**, this information is then presented to the user in the **consent screen**, and the access token issued to the application will be **limited** to the scopes granted.



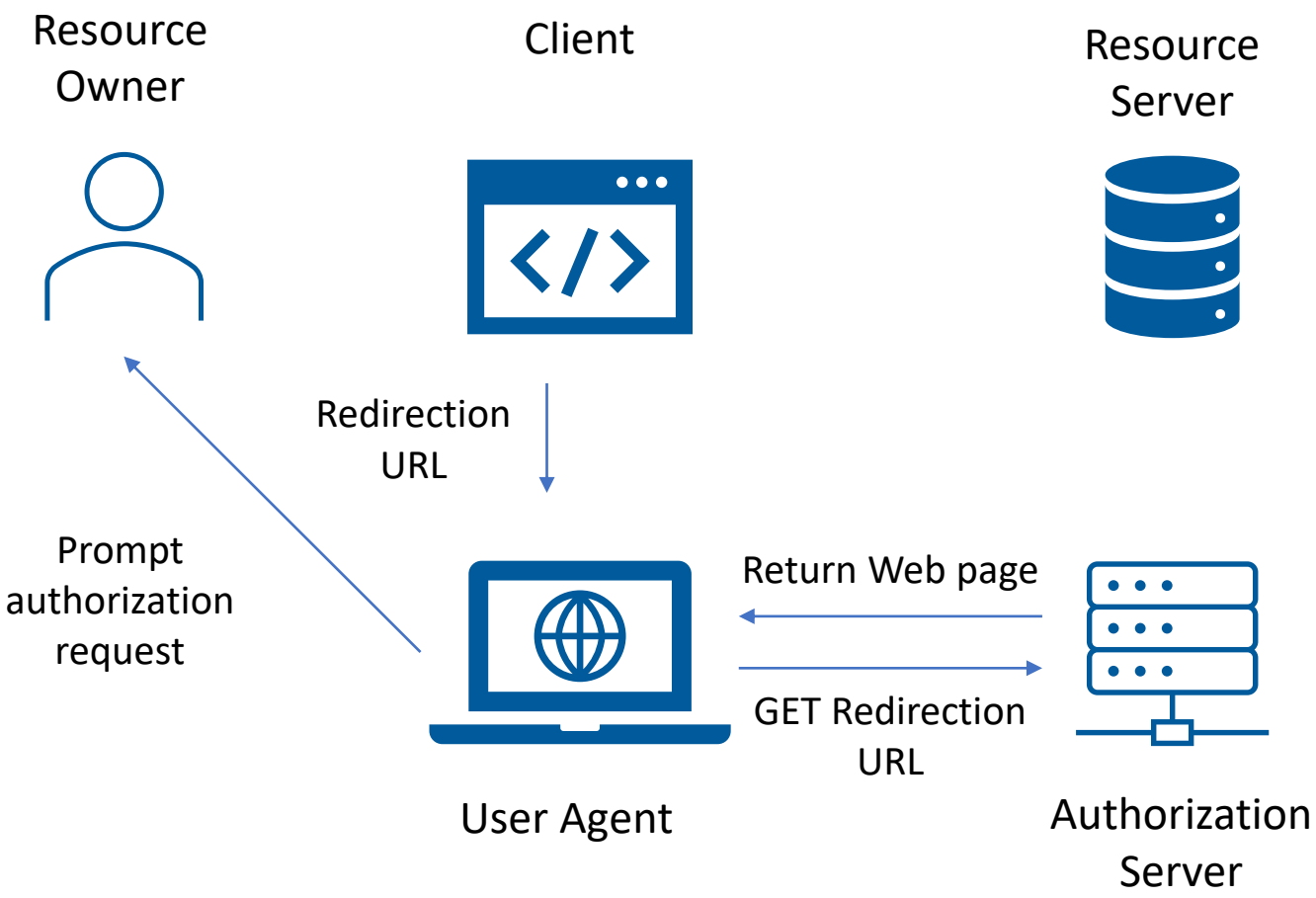
WoT and OAuth2.0

Is Oauth 2.0 a nice fit for IoT and Web of Thing?

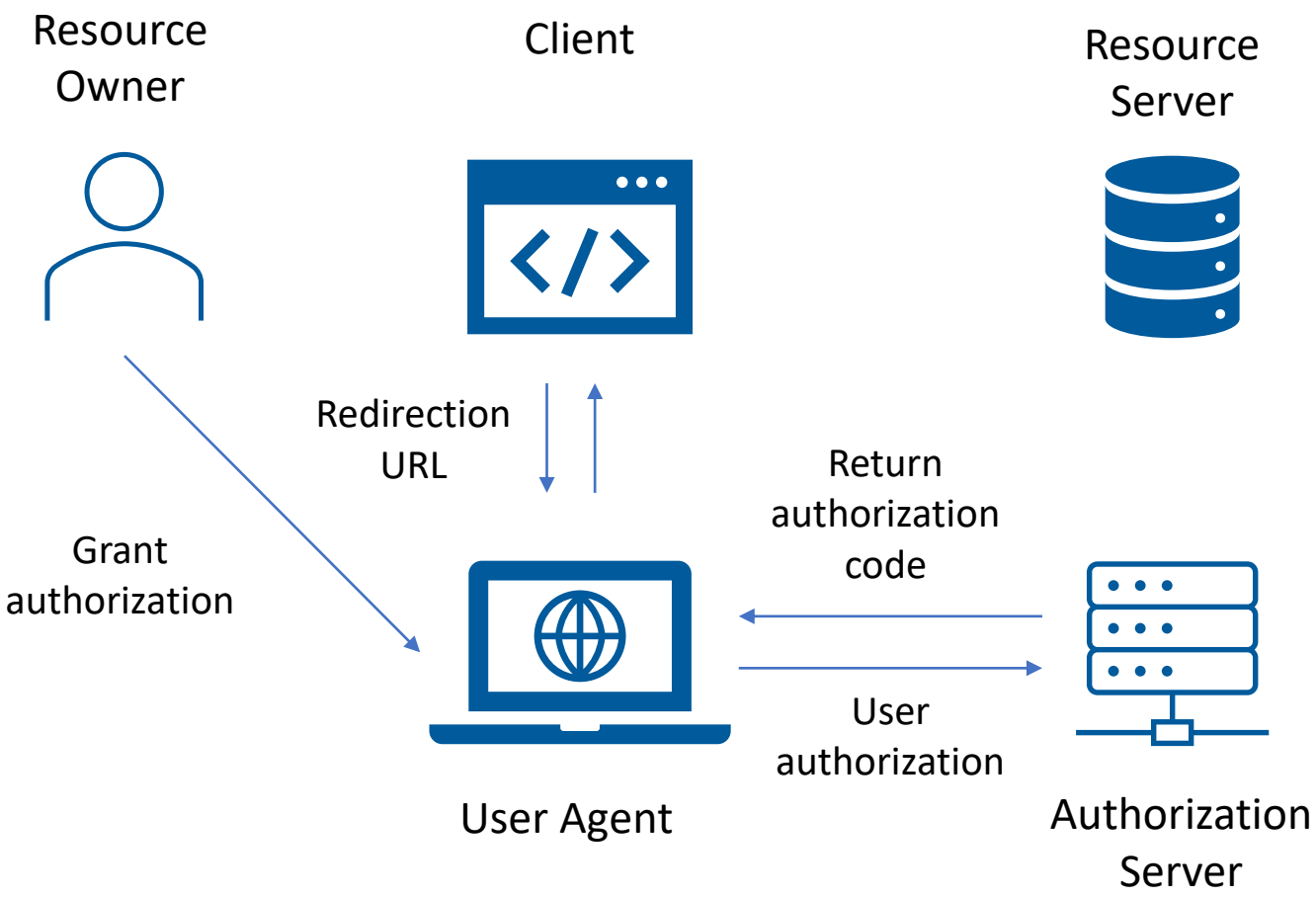
oAuth2.0 flows/grants

- Code
- Implicit 🚫
- Resource Owner Password Credentials 🚫
- Client Credentials
- Extension flows
  - Device

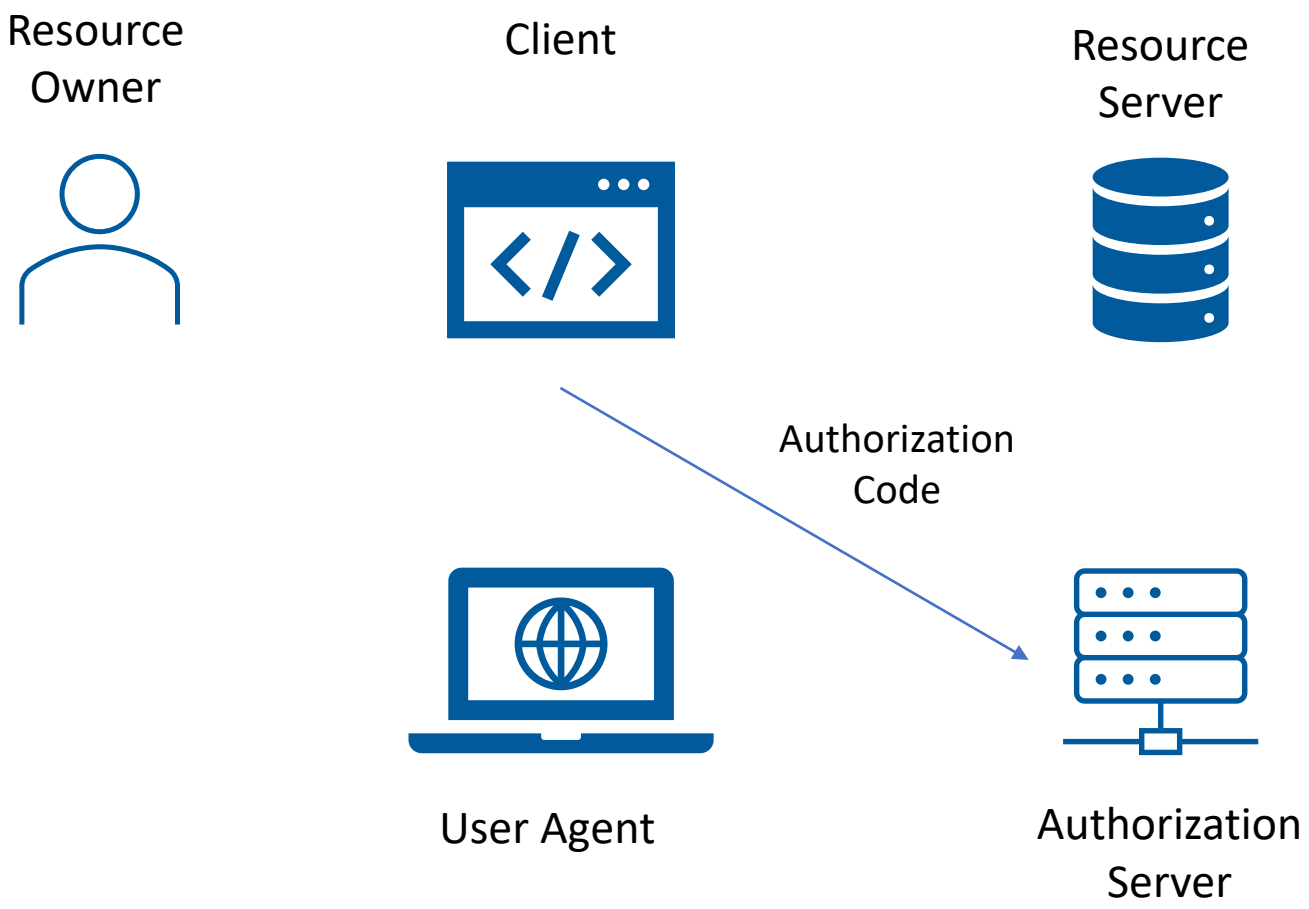
Code


Resource Owner
Client
Resource Server
Redirection URL
Prompt authorization request
Return Web page
GET Redirection URL
User Agent
Authorization Server

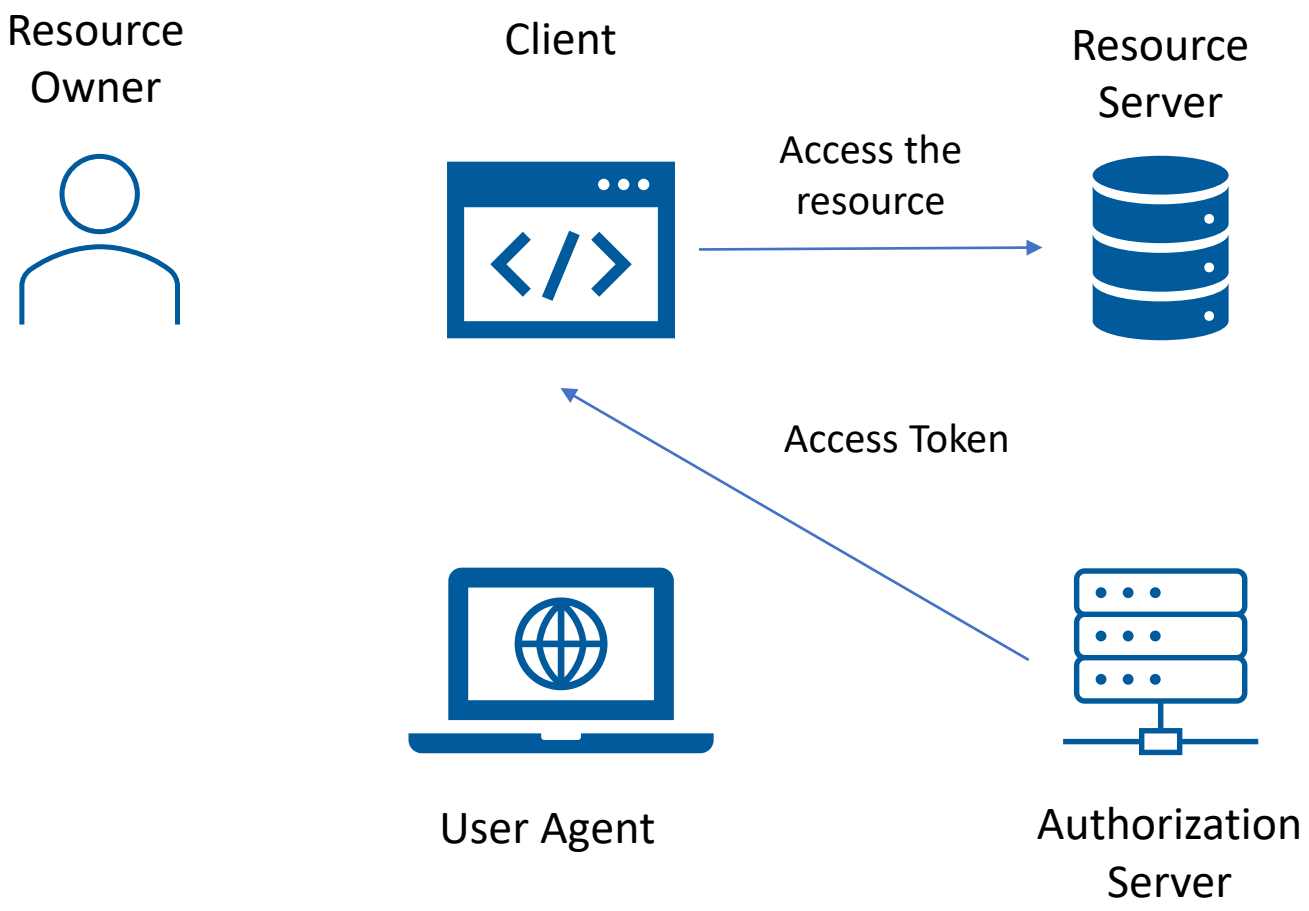
Code


Resource Owner
Client
Resource Server
Redirection URL
Return authorization code
Grant authorization
User authorization
User Agent
Authorization Server

Code


Resource Owner
Client
Resource Server
Authorization Code
User Agent
Authorization Server

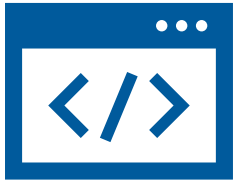
Code


Resource Owner
Client
Resource Server
Access the resource
Access Token
User Agent
Authorization Server

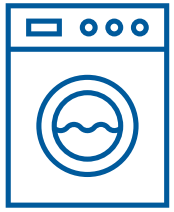
Code – WoT Scenarios

Device owner

Smart Home Dashboard Application



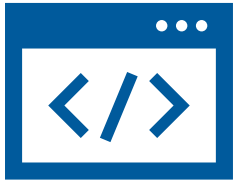
WoT Washing machine



Code – WoT Scenarios

Device owner

Smart Home Dashboard Application

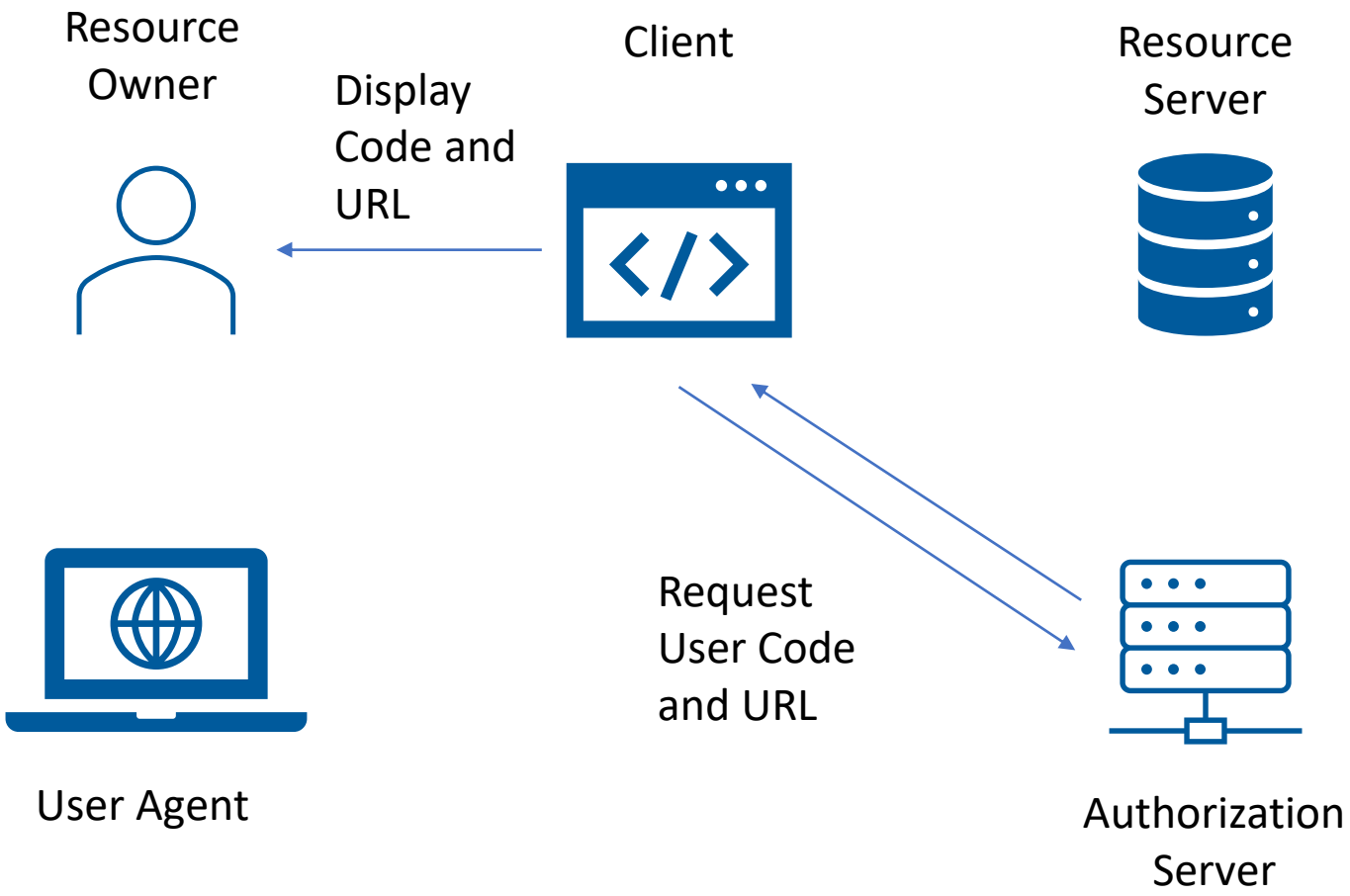


Thing Description Directory

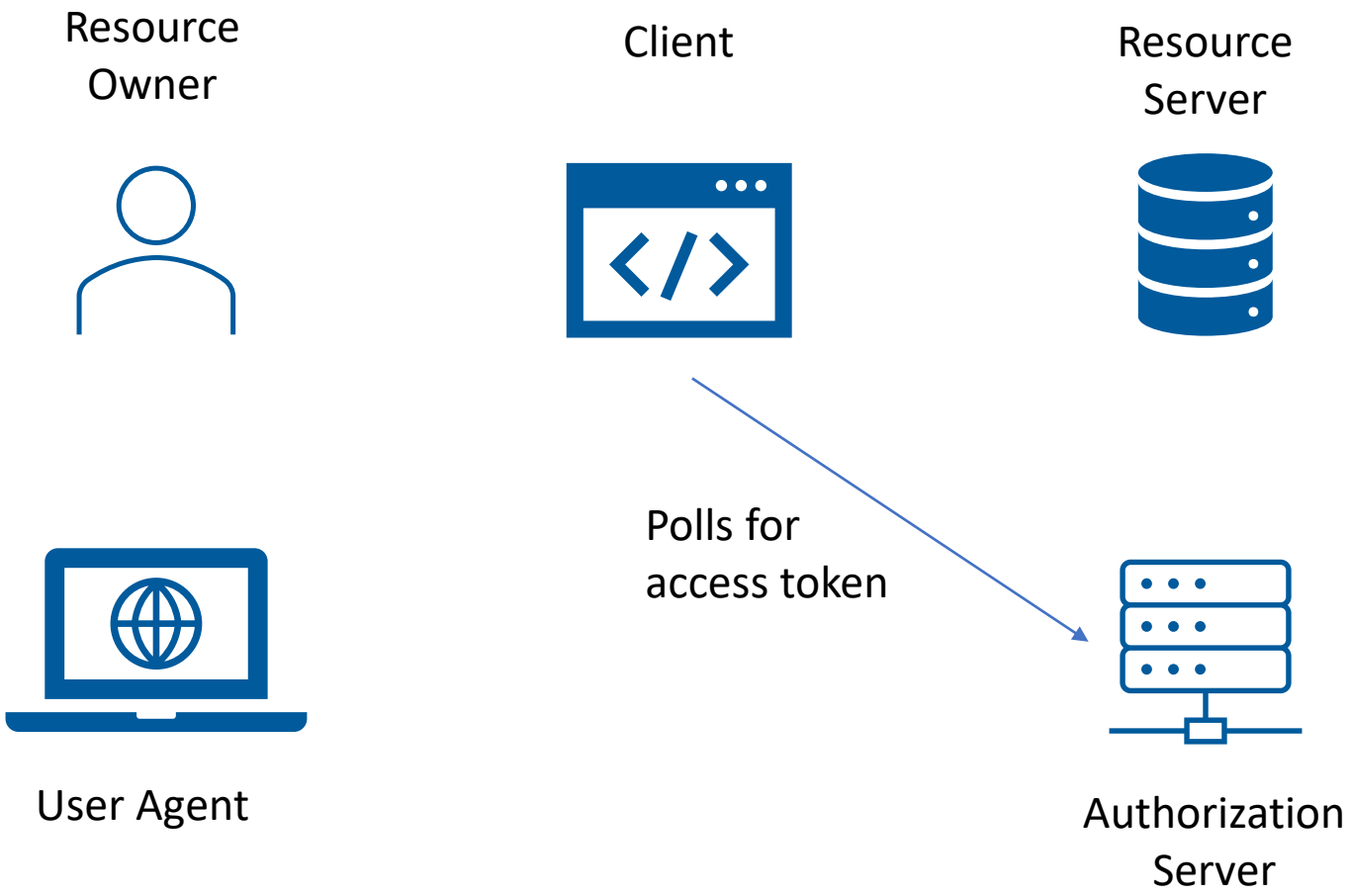
Device

Device flow was born to cope with **limited input** devices. In particular, if a device was not able to prompt a full fledge browser it couldn't use OAuth 2.0. Typically, device flow is used in Smart TVs, Home Assistant speakers, and wearables.

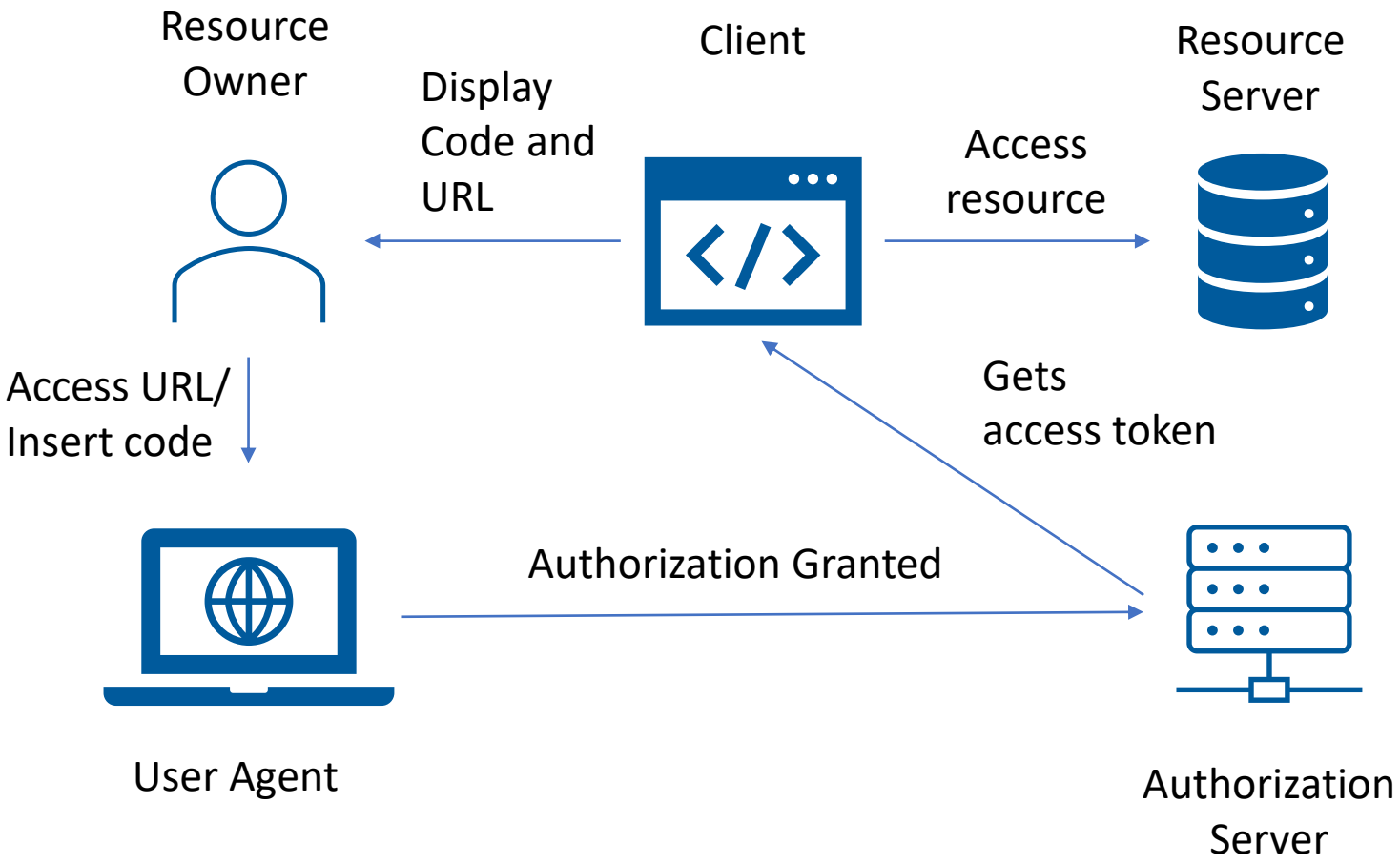
Device


Resource Owner
Display Code and URL
Client
Resource Server
Request User Code and URL
User Agent
Authorization Server

Device


Resource Owner
Client
Resource Server
Polls for access token
User Agent
Authorization Server

Device

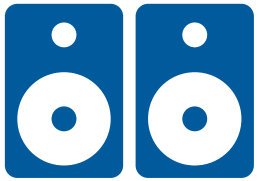

Resource Owner
Client
Resource Server
Display Code and URL
Access resource
Access URL/ Insert code
Gets access token
Authorization Granted
User Agent
Authorization Server

Device - WoT Scenarios

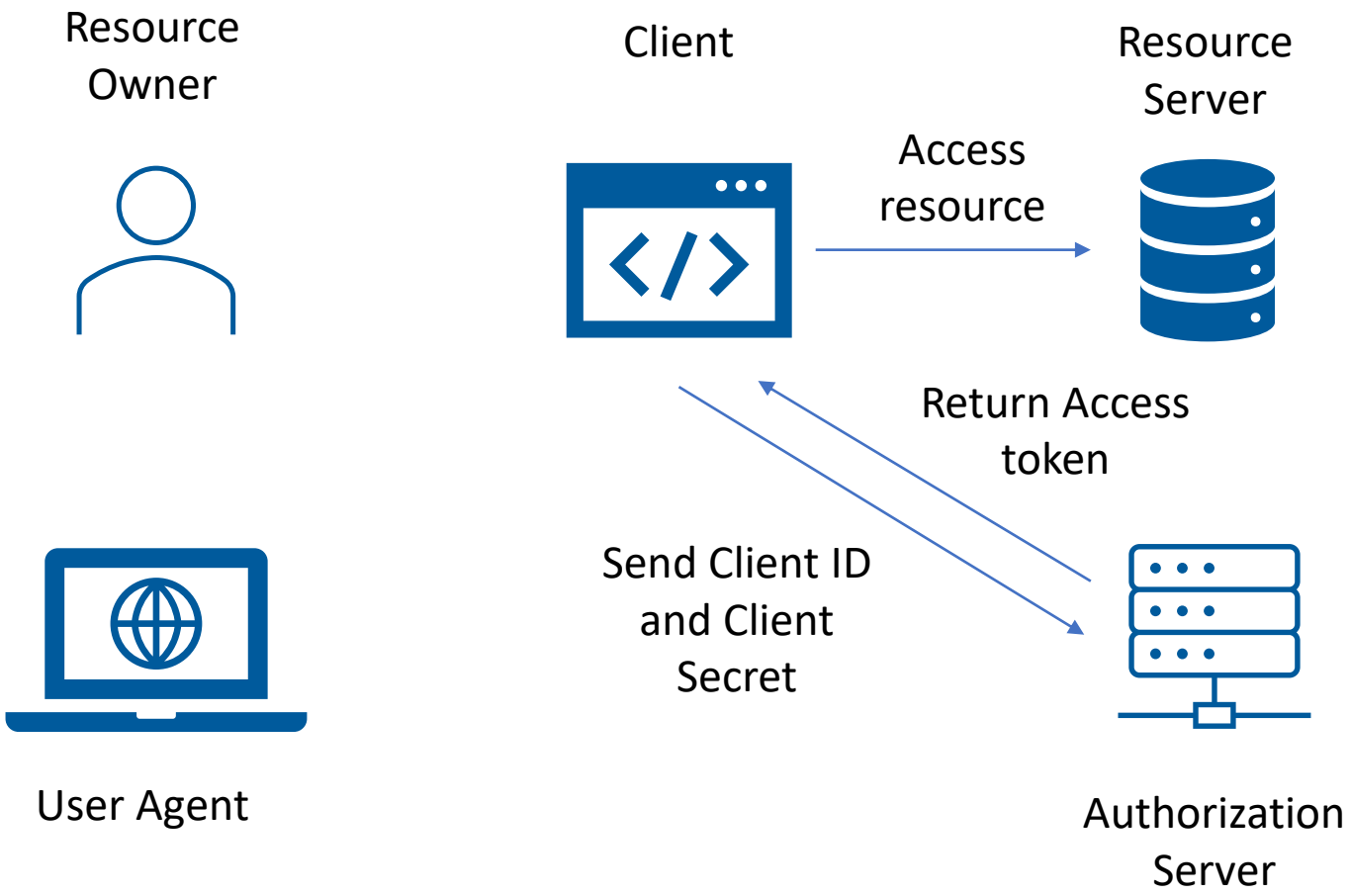
Device owner

Smart Radio

WoT speakers

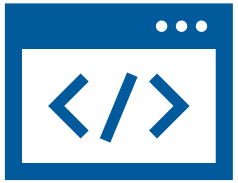


Client

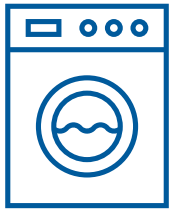

Resource Owner
Client
Resource Server
Access resource
Return Access token
Send Client ID and Client Secret
User Agent
Authorization Server

Client - Scenarios

Proprietary companion application



WoT Washing machine



Client - Scenarios

The Client Credentials grant type is used by clients to obtain an access token **outside** of the context of an **end-user**. From RFC 6749:

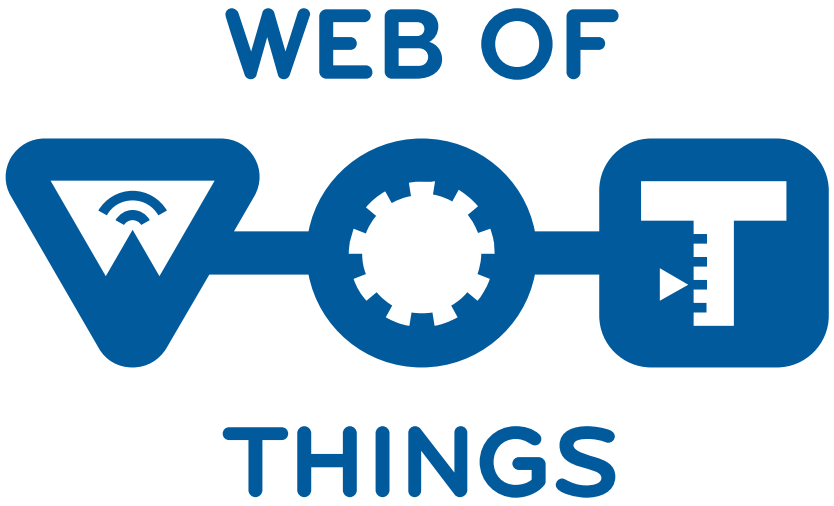
The client can request an access token using only its client credentials (or other supported means of authentication) when the the client is requesting access to the protected resources under its control, or ***those of another resource owner that has been previously arranged with the authorization server*** *(the method of which is beyond the scope of this specification).*

Therefore the client credential grant might be used:

- When the resource owner is a **public authority**. For example, in a smart city context, the authority provides a web service where to register an application id.
- Industrial IoT. Consider a smart factory where the devices or services are provisioned with client credentials.

Open points

- Provide examples about how to use deprecated flows
- Multiple OAuth 2.0 flows in security definitions
  - https://github.com/w3c/wot-thing-description/issues/929
- Address implementations variability
  - https://github.com/w3c/wot-thing-description/issues/923
- Node-WoT implementation
  - https://github.com/eclipse/thingweb.node-wot/issues/325
- How WoT scripts can handle OAuth 2.0 code gracefully
  - https://github.com/w3c/wot-scripting-api/issues/214



WoT and OAuth 2.0

Cristiano Aguzzi

20/10/2020