AUTHLETE

OAuth 2.0

Client Credentials

Recap

OAuth 2.0 enable api clients parties to access protected resources in a domain with limited permissions

Recap - Authorization Server Endpoints

- Authorization
  - Allow the resource owner interact with the authorization server in order to grant permissions to client
- Token
  - Used by the client to retrieve access tokens using grants from authorization or refresh tokens

Recap

Client credentials is one of the grant types that can be used by clients and AS

Client Credentials

Client uses its credentials to request a token for accessing protected resources that client controls , i.e., there is no resource owner

Scenarios

App connecting to social network to access its resources, like: Facebook insight or read only access on Twitter

Scenarios

Client access resources on PaaS providers like Google Prediction API or Cloud Storage access or Azure Daemon apps accessing infrastructure

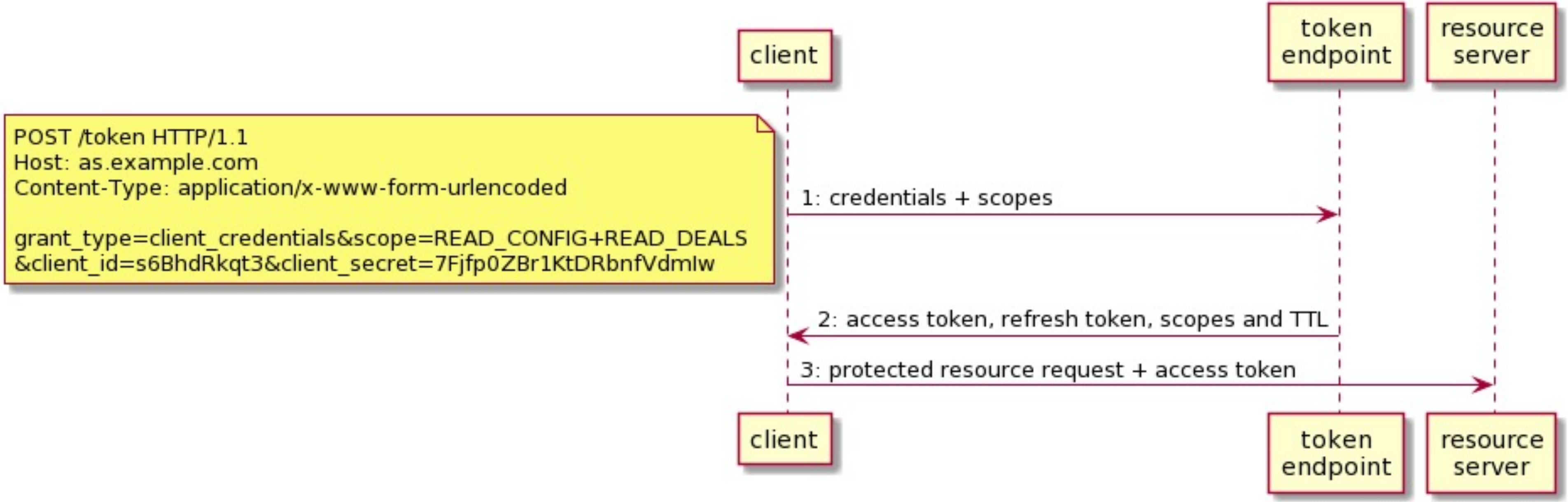
Scenarios

A terminal out of enterprise domain accessing services, like kiosk

Scenarios

Telemetry collection on a distributed plant

Client Credentials



```
POST /token HTTP/1.1
Host: as.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&scope=READ_CONFIG+READ_DEALS
&client_id=s6BhdRkqt3&client_secret=7Fjfp0ZBr1KtDRbnfVdmIw
```

Client Credentials

- Secret
  - As Basic auth on header
  - In the body
- JWT signed with secret
- JWT signed with private key (RSA, EC)
- TLS cert used by mtls
  - PKI
  - self signed

Caveats

Using client centered grant to customers profile is the that can be used for API Abuse

Segregation of Scopes

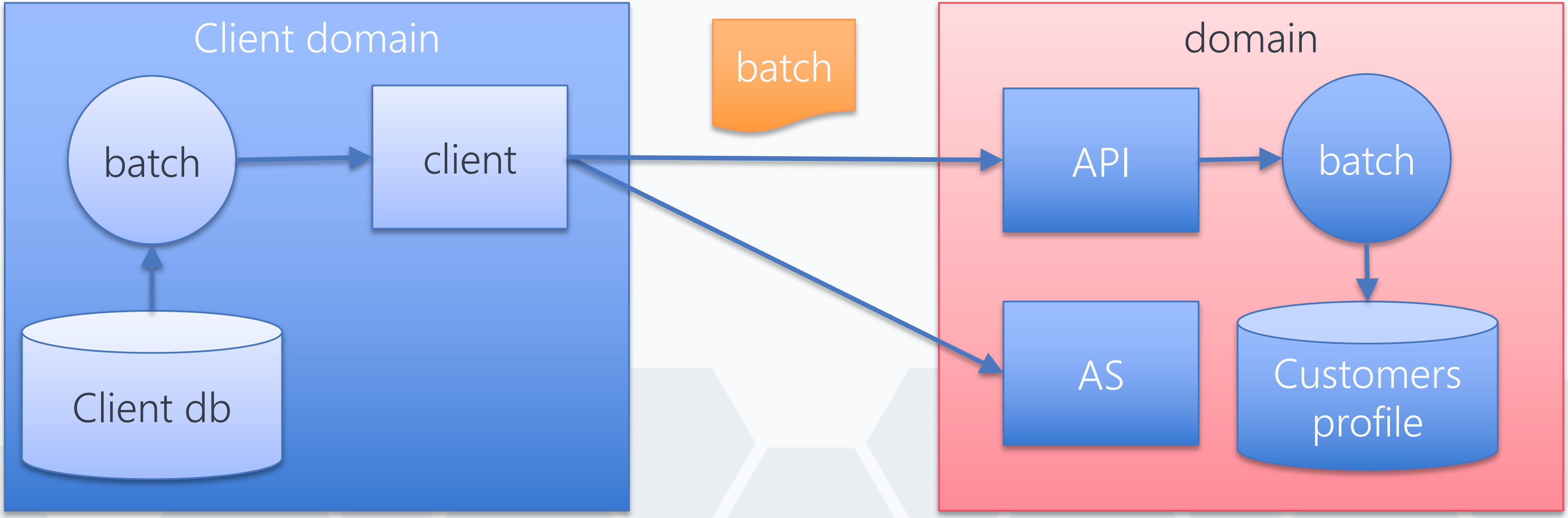
- User centered scopes
- Client centered scopes

- If the AS can't segregate the scopes, one can create different clients and enforce the segregation.

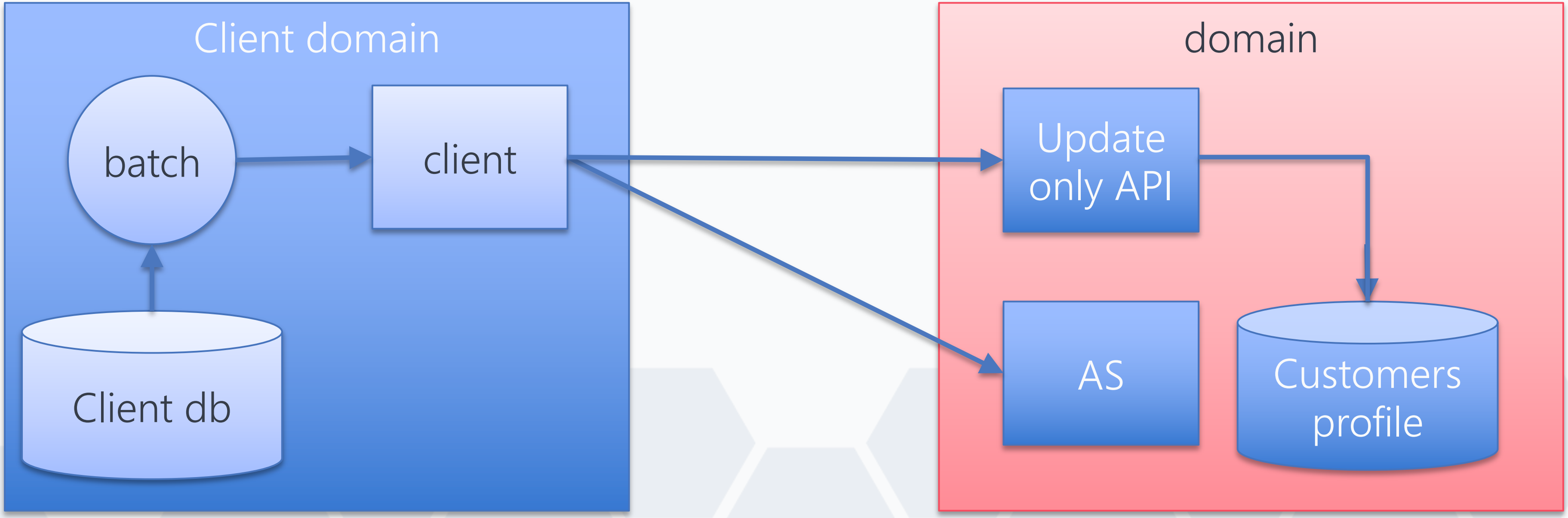
Batch Processing

Design suggestion is to split the batch processing between the 3rd party and in domain processing to access the customer's profile.

Batch Processing (1)



Batch Processing (2)



OAuth 2.0

Client Credentials

AUTHLETE

OAuth 2.0

Implicit Flow

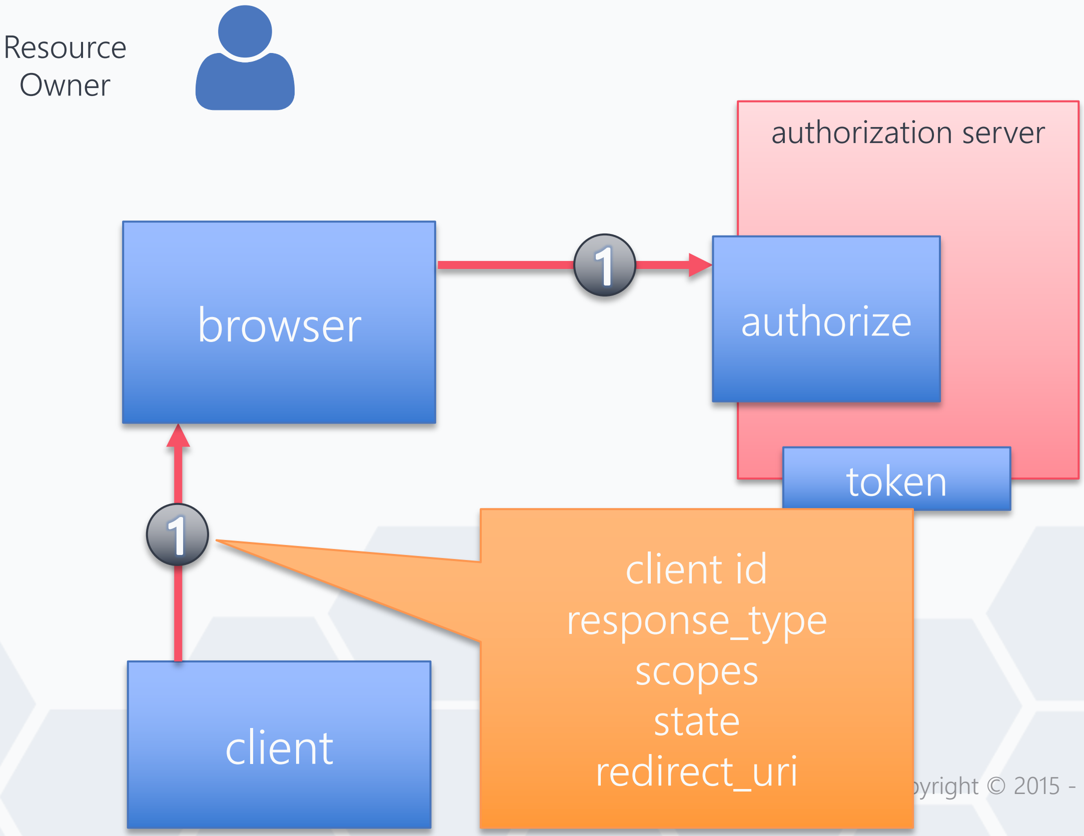
Implicit flow

It is a grant type focused on browser running clients

Implicit flow

The client receives the access token on the redirect uri, so it does not interact with the token endpoint

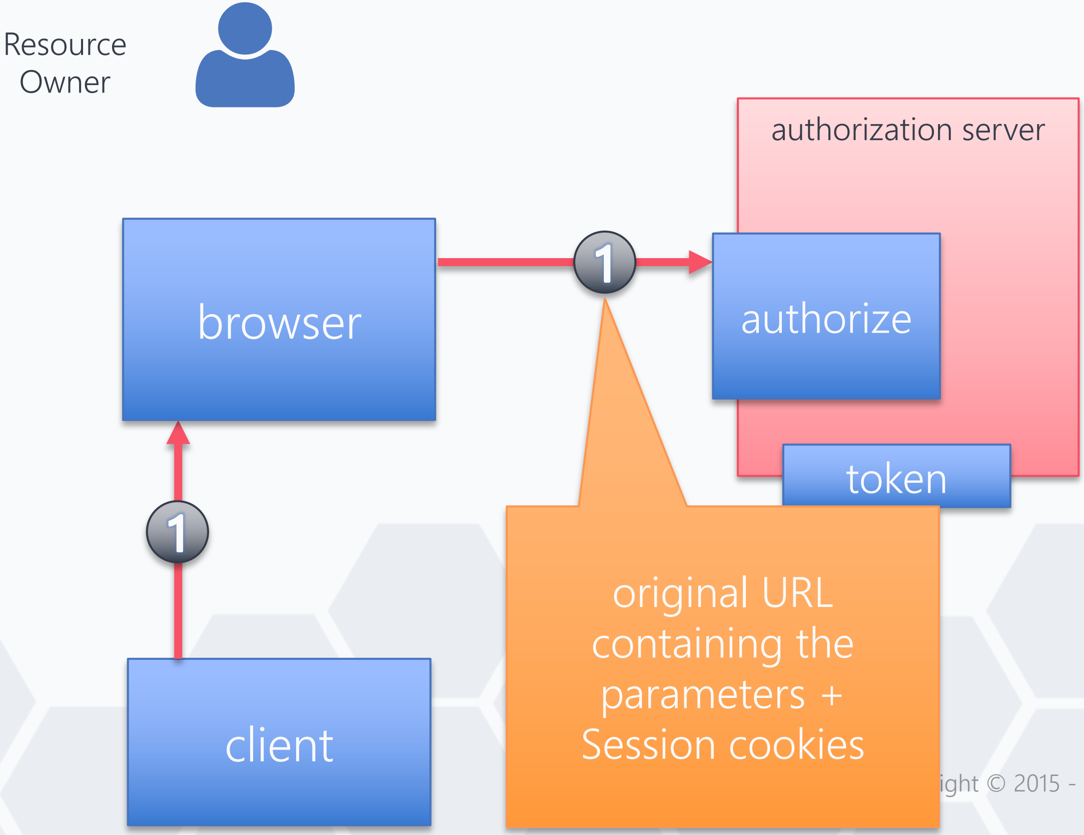
Implicit flow



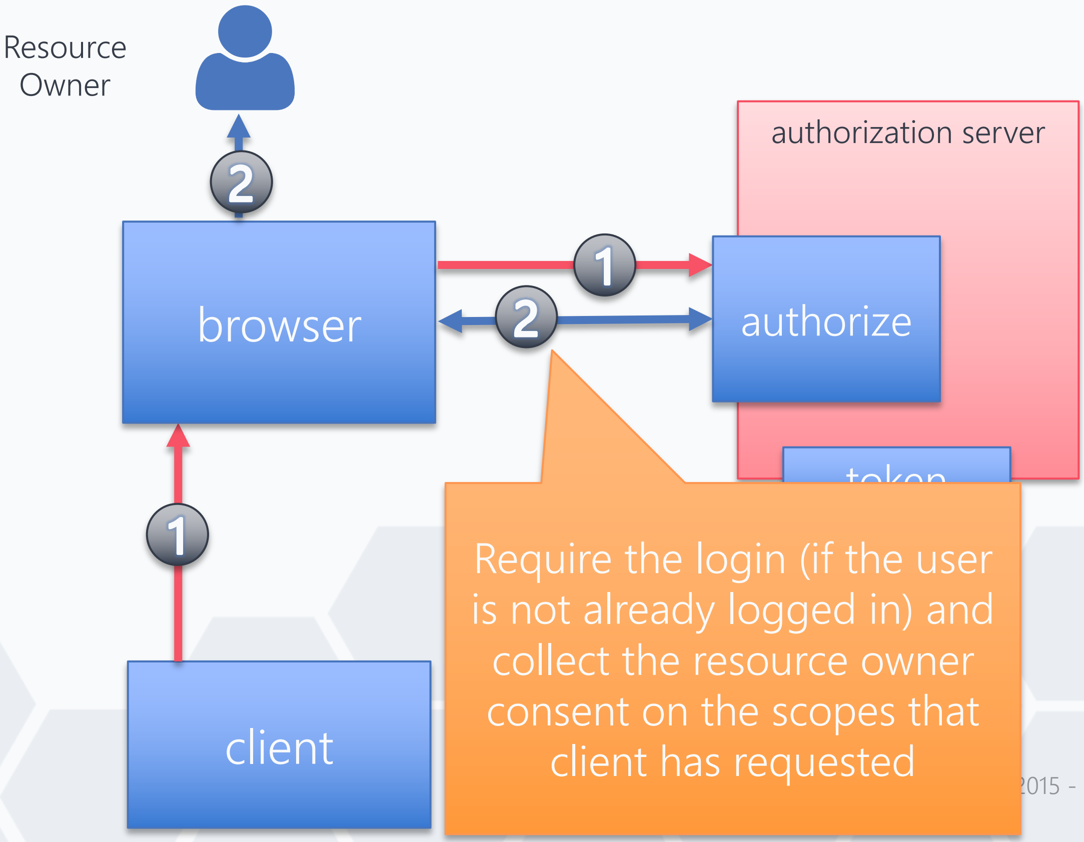
1 – Redirect to authorize endpoint with response_type=token

Implicit flow

1 – Redirect to authorize endpoint with response_type=token



Implicit flow

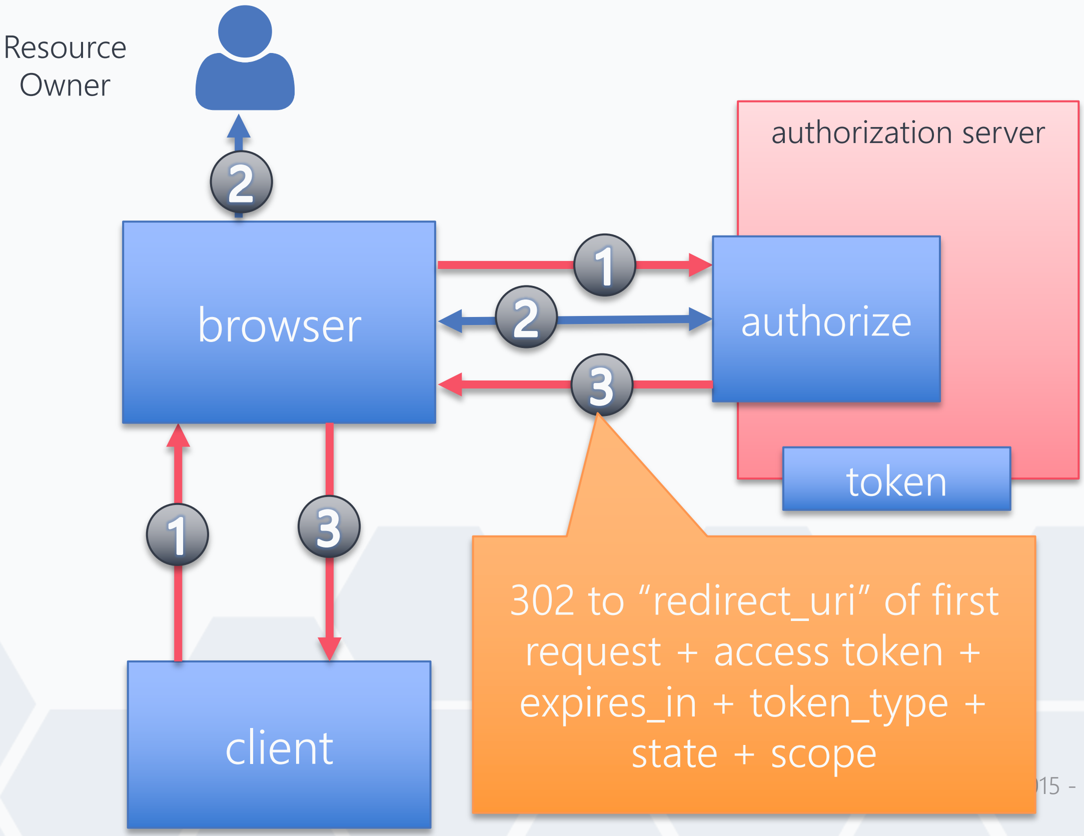


Require the login (if the user is not already logged in) and collect the resource owner consent on the scopes that client has requested

1 – Redirect to authorize endpoint with response_type=token

2 – The user login and grant the permission

Implicit flow



302 to "redirect_uri" of first request + access token + expires_in + token_type + state + scope

1 – Redirect to authorize endpoint with response_type=token

2 – The user login and grant the permission

3 – redirect to client with the access token and TTL

Nothing comes for free

- Client can’t be authenticated
- The effort is not to expose the access token
  - on transport
  - on logging
  - on history
  - on refer
  - Javascript tampered by overload

Implicit flow

- How the attack vector can be prevented make it a very restricted option.

- It is not considered for OAuth 2.1

AUTHLETE

OAuth 2.0

Implicit Flow

OAuth 2.0

Resource Owner Flow

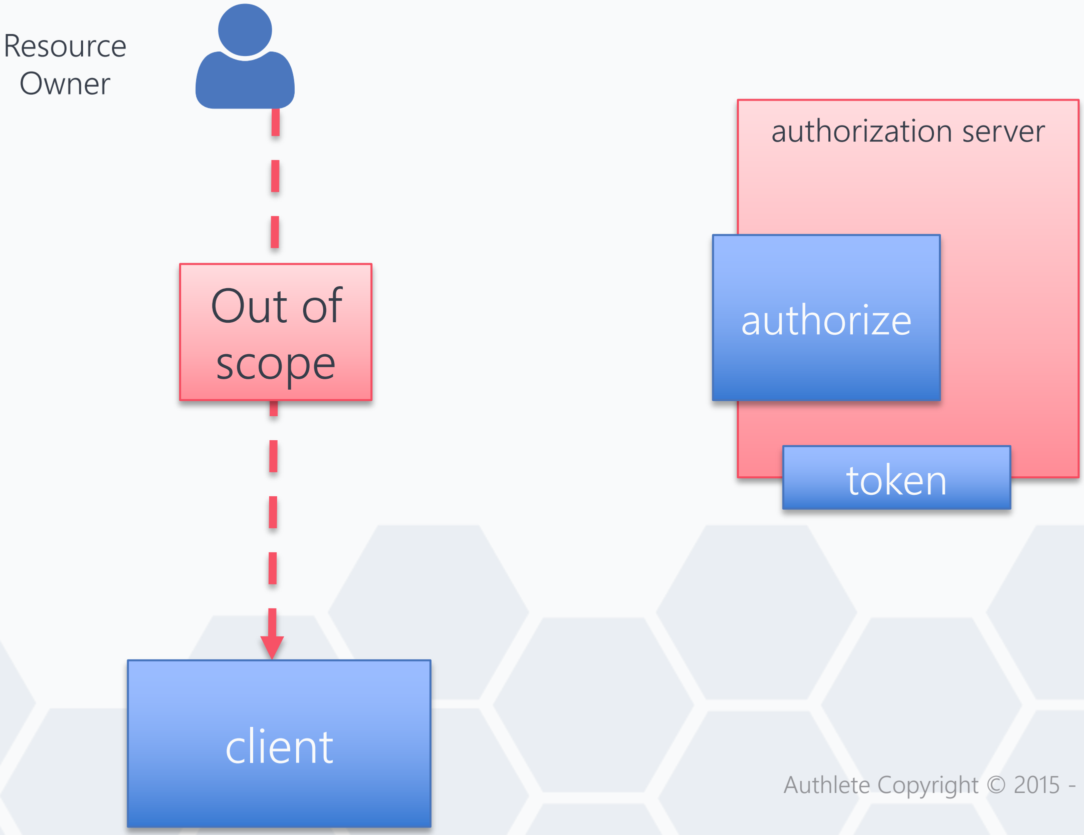
Resource Owner flow

It is a grant type designed on migrating the infrastructure to access token usage

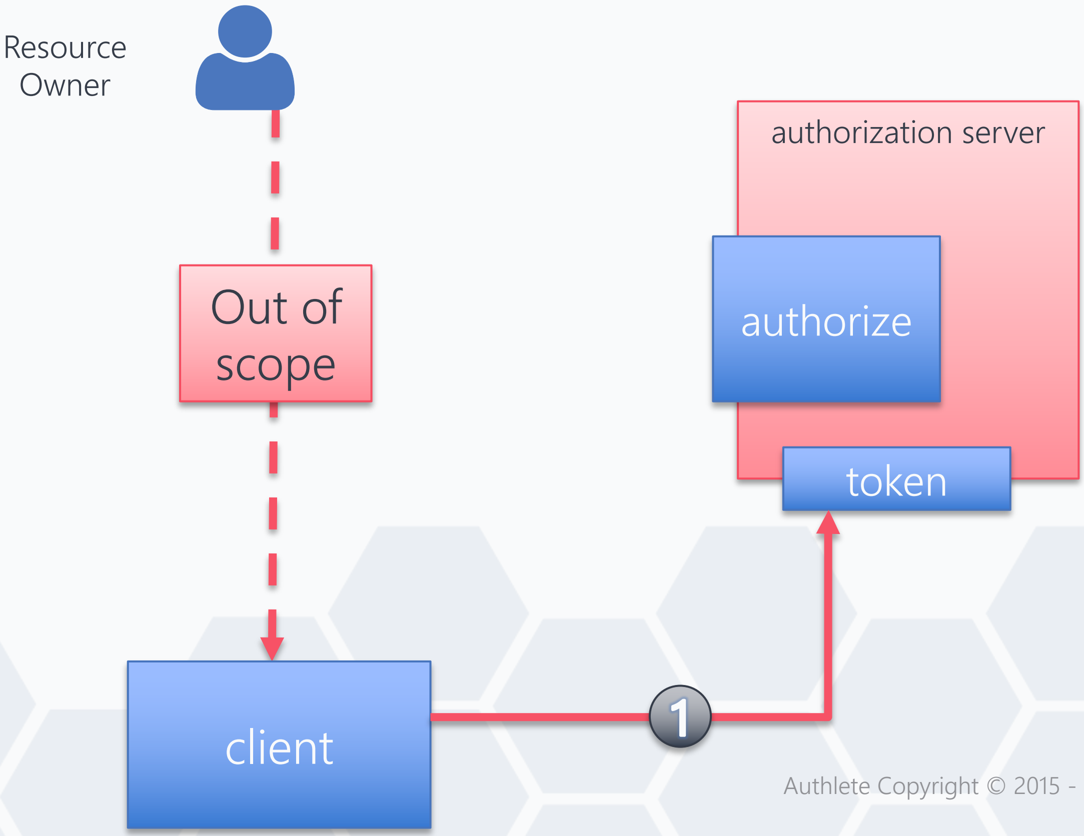
Resource Owner flow

The resource owner does not control authorization process: the user credentials are handled to client and client creates an access token

Resource Owner flow



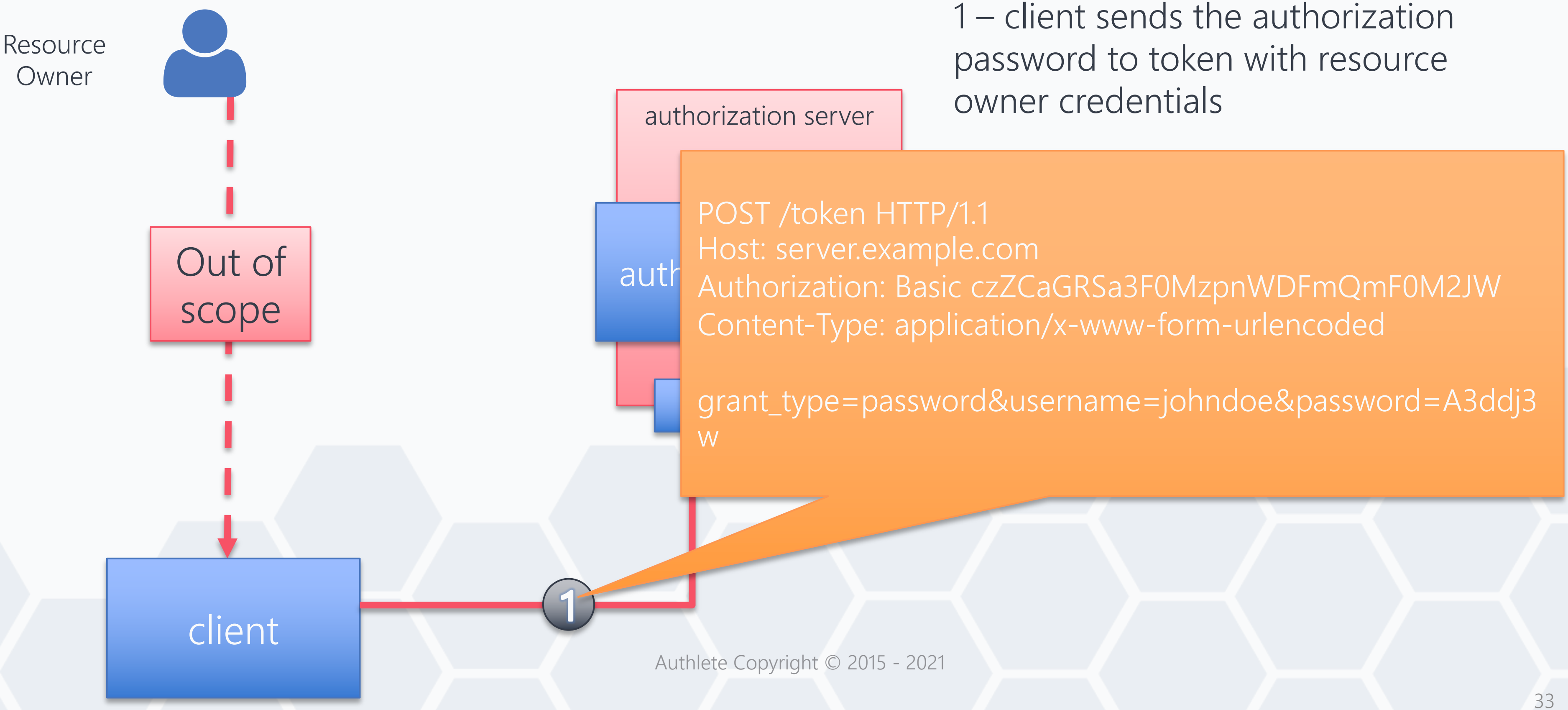
Resource Owner flow



1 – client sends the authorization password to token with resource owner credentials

Resource Owner flow

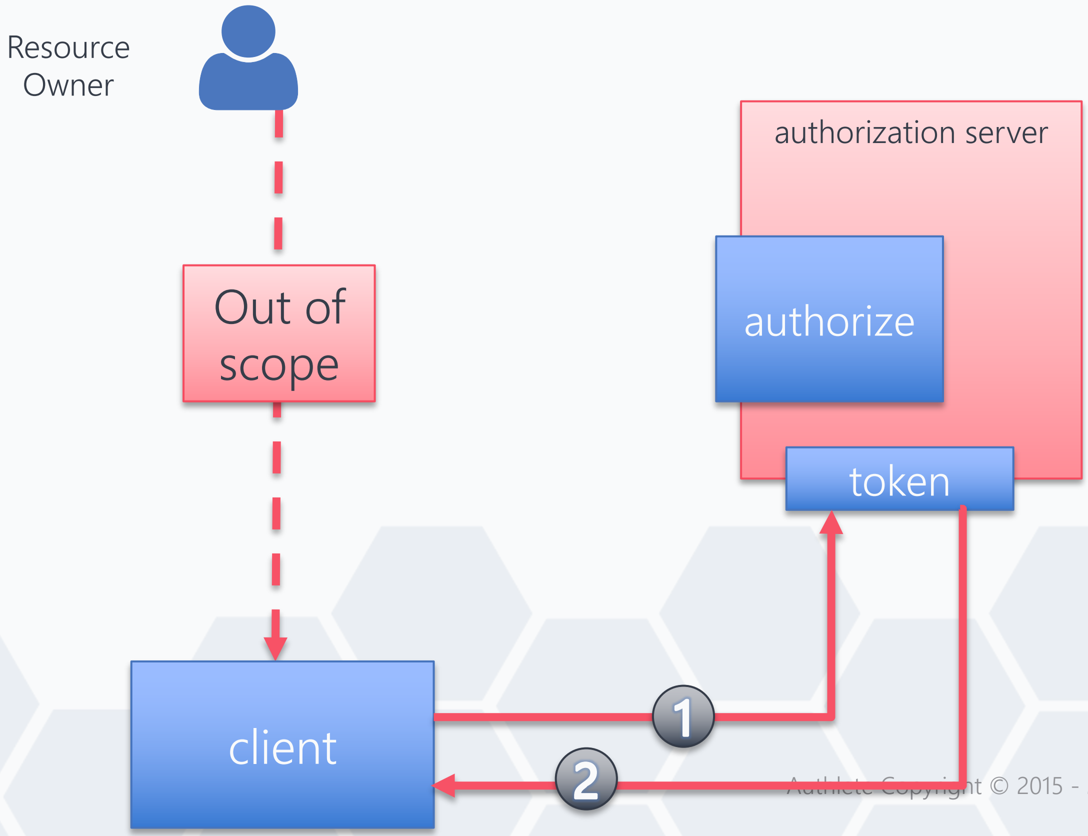
1 – client sends the authorization password to token with resource owner credentials



```
POST /token HTTP/1.1
Host: server.example.com
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
Content-Type: application/x-www-form-urlencoded

grant_type=password&username=johndoe&password=A3ddj3
w
```

Resource Owner flow



1 – client sends the authorization password to token with resource owner credentials

2 – AS returns the access token, refresh token, granted scopes and time to live of the token

Resource Owner flow

- Client collects the credentials
- Multi factor authentication is not addressed
- Not considerer for OAuth 2.1

AUTHLETE

OAuth 2.0

Resource Owner Flow