

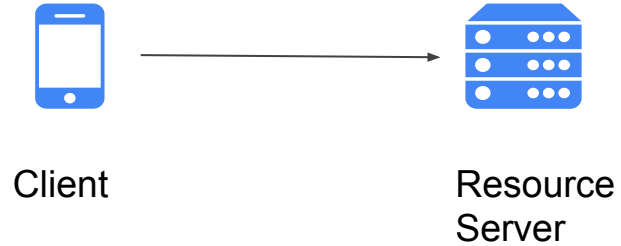


OAuth - Client Credentials

OAuth Grant Types

Grant Type	Typical Use Case	Complex?
No specific resource owner is involved		
Client Credentials	Business system interactions, where resources being operated on are owned by the partner, not a particular user	No
A specific resource owner is involved		
Resource Owner Password Credentials	Resources are owned by a particular user and the requesting application is trusted	A bit
Authorization Code	Resources are owned by a particular user and the requesting application is untrusted	Very
Implicit	Resources are owned by a particular user, and the requesting application is an untrusted browser-based app written in a scripting language such as JavaScript	Very complex, and less secure than Auth Code

What's the main idea?



Client Credentials - Actors



Client

Client Credentials - Actors



Client



Apigee
(Authorization
Server)

Client Credentials - Actors



Client

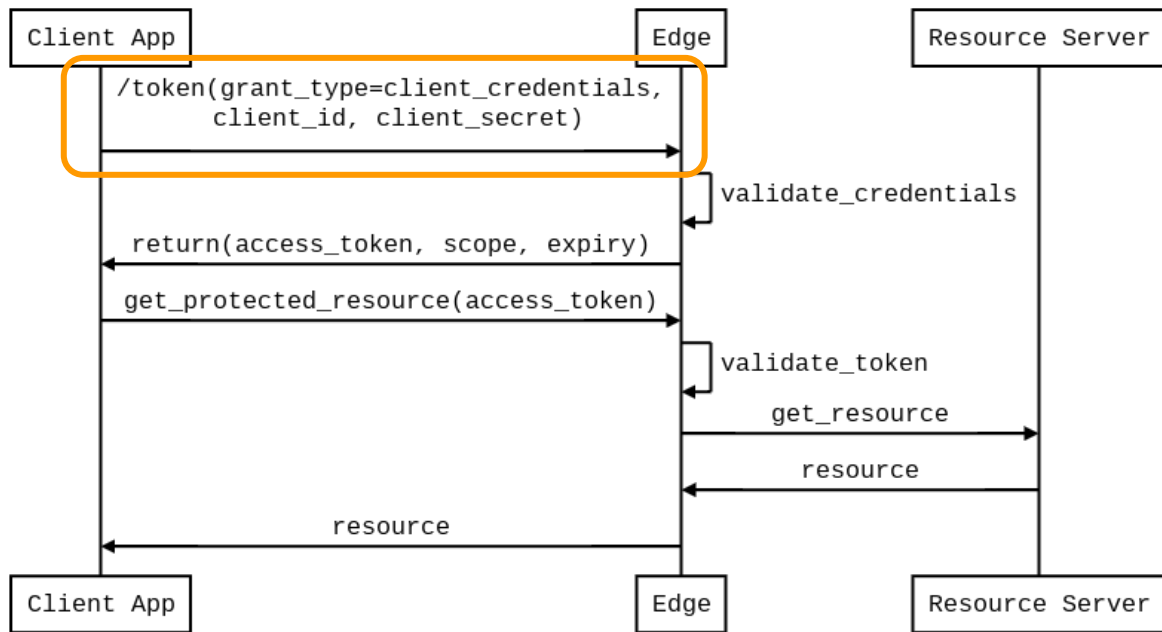


Apigee
(Authorization
Server)

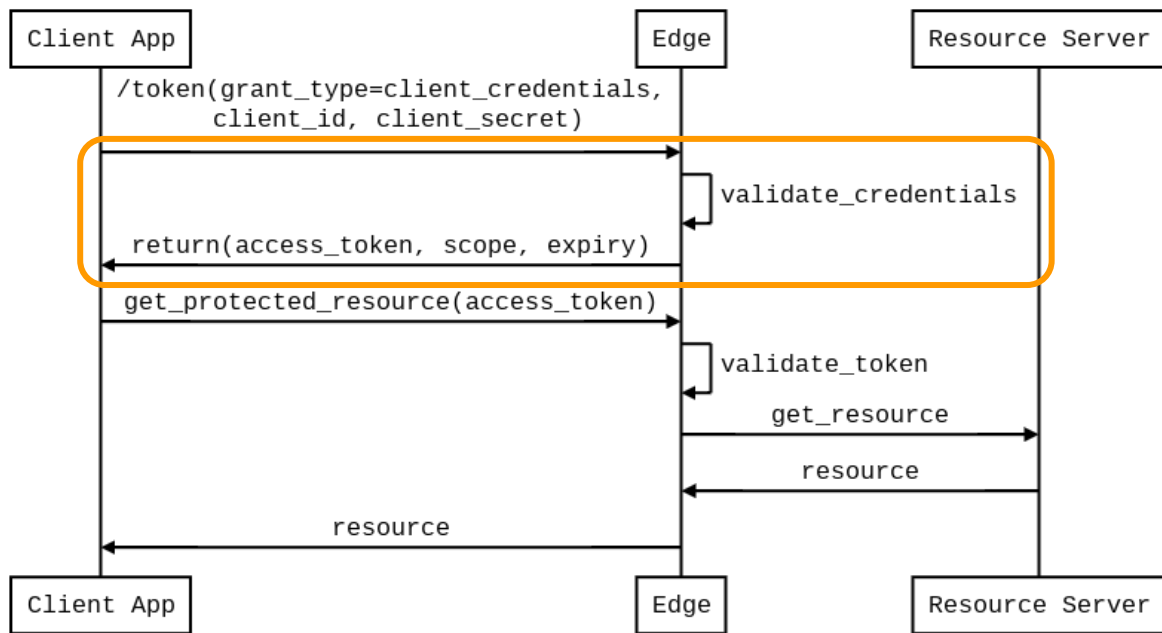


Resource
Server

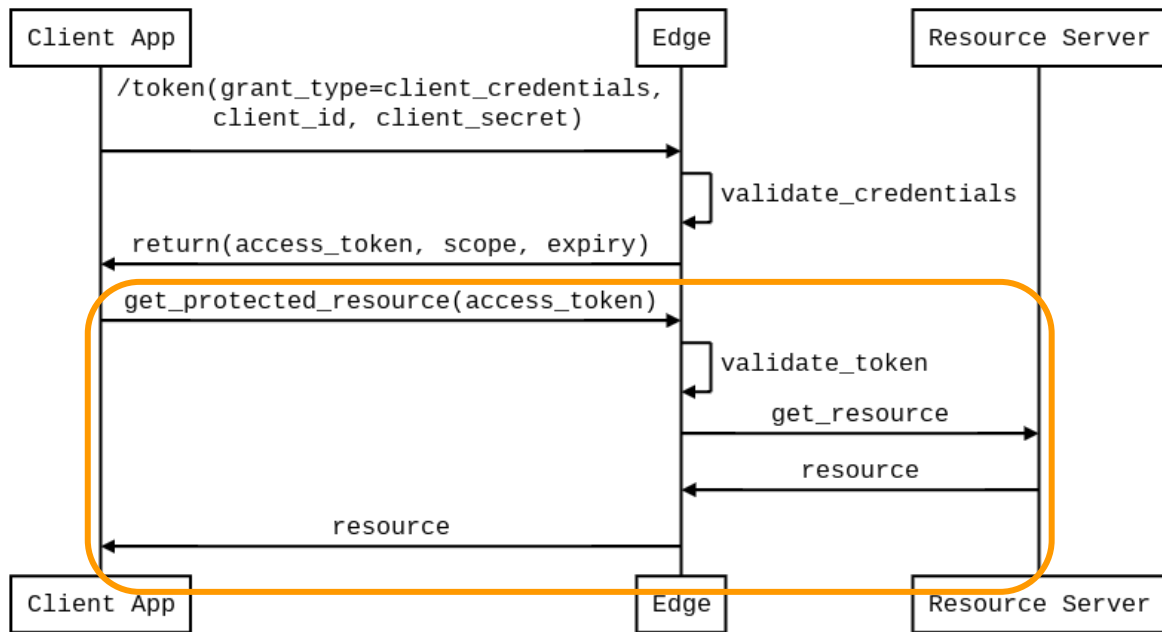
Client credentials - Sequence diagram



Client credentials - Sequence diagram



Client credentials - Sequence diagram



Generate Access Token

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<OAuthV2 async="false" continueOnError="false"
enabled="true" name="oauth-generate-token">
  <DisplayName>OAuth Generate Token</DisplayName>
  <Operation>GenerateAccessToken</Operation>
  <ExpiresIn>86400000</ExpiresIn>
  <SupportedGrantTypes>
    <GrantType>client_credentials</GrantType>
  </SupportedGrantTypes>
  <GrantType>request.queryparam.grant_type</GrantType>
  <GenerateResponse/>
</OAuthV2>
```

Generate Access Token

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAuthV2 async="false" continueOnError="false"
enabled="true" name="oauth-generate-token">
  <DisplayName>OAuth Generate Token</DisplayName>
  <Operation>GenerateAccessToken</Operation>
  <ExpiresIn>86400000</ExpiresIn>
  <SupportedGrantTypes>
    <GrantType>client_credentials</GrantType>
  </SupportedGrantTypes>
  <GrantType>request.queryparam.grant_type</GrantType>
  <GenerateResponse/>
</OAuthV2>
```

Verify OAuth Token Policy

```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<OAuthV2 async="false" continueOnError="false"
enabled="true" name="VerifyOAuthToken">
  <DisplayName>OAuth Verify Token</DisplayName>
  <Operation>VerifyAccessToken</Operation>
</OAuthV2>
```

Verify OAuth Token Policy

```
curl -H "Authorization: Bearer {access_token}"  
http://myorg-test.apigee.net/v1/cc/oauth_cc_weather  
/forecastrss?w=12797282
```



Thank You