Google Cloud

Edge Security

OAuth - Resource Owner Password

OAuth grants

An OAuth Grant is a credential representing the resource owner’s authorization. More often than not, we tend to think of grants in terms of the process used to obtain an access token.

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, and potentially insecure as well

Resource owner password grant type

Resources participating in password Grant Type

- End User
- Mobile Application
- Apigee Generating and Validating Token
- Authentication Server To Validate the Credentials
- Backend API Resource

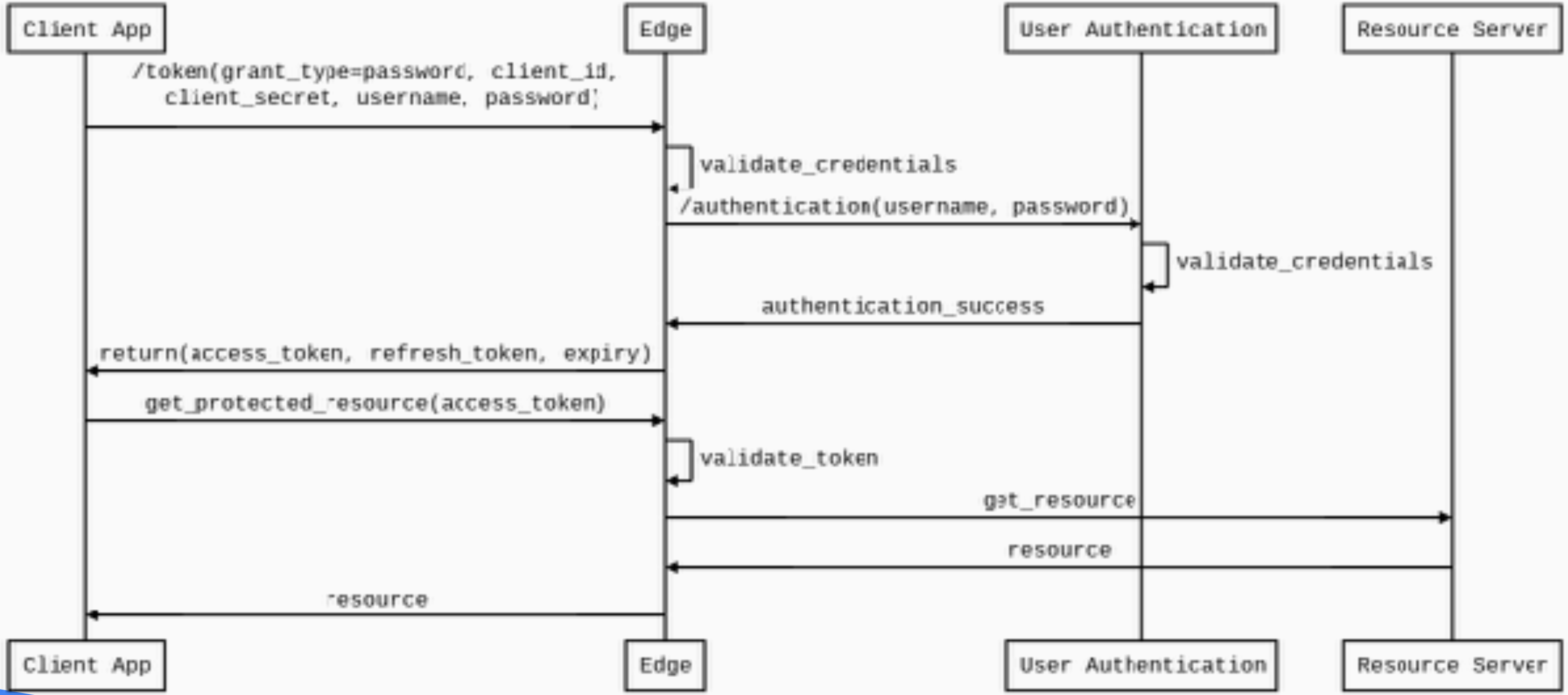
Resource Owner is involved, Typically owned by User and Requesting Application is Trusted

Slightly More Complex to Implement Than Client Credentials Grant Type

More Secure Than Client Credentials Grant Type

Refresh Token Generated Along with Bearer Token

Resource owner password - Sequence diagram



Client App

Edge

User Authentication

Resource Server

/token(grant_type=password, client_id, client_secret, username, password)

validate_credentials

/authentication(username, password)

validate_credentials

authentication_success

return(access_token, refresh_token, expiry)

get_protected_resource(access_token)

validate_token

get_resource

resource

resource

Client App

Edge

User Authentication

Resource Server

Generate access token policy

The resource owner password grant is similar to the client credentials grant type, but with an extra step to validate the user credentials.
Edge does not validate user credentials, so an authentication service should be used.
Create this endpoint and use the OAuthV2 policy to generate an 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>password</GrantType>
    </SupportedGrantTypes>
    <GrantType>request.queryparam.grant_type</GrantType>
    <UserName>request.formparam.username</UserName>
    <PassWord>request.formparam.password</PassWord>
    <GenerateResponse/>
</OAuthV2>
```

Responses: Client credentials vs. Password grant types

Response:

```
{
  "issued_at" : "1407513671919",
  "application_name" : "26c855a9-c485-4318-accc-7e3f533a154c",
  "scope" : "",
  "status" : "approved",
  "api_product_list" :
"[Certification_OAuthClientCredentialsWeather]",
  "expires_in" : "3599",
  "developer.email" : "certifieddev@apigee.com",
  "organization_id" : "0",
  "token_type" : "BearerToken",
  "client_id" : "vn0zG4cnSWaWIzdwBZgnREI1NGORDXXz",
  "access_token" : "2CsgxkPqfNtCSAZ5qGEI9x5dGdvV",
  "organization_name" : "chrisv-cs",
  "refresh_token_expires_in" : "0",
  "refresh_count" : "0"
}
```

Response:

```
{
  "issued_at" : "1407513709051",
  "scope" : "",
  "application_name" : "26c855a9-c485-4318-accc-7e3f533a154c",
  "refresh_token_issued_at" : "1407513709051",
  "status" : "approved",
  "refresh_token_status" : "approved",
  "api_product_list" :
"[Certification_OAuthClientCredentialsWeather]",
  "expires_in" : "3599",
  "developer.email" : "certifieddev@apigee.com",
  "organization_id" : "0",
  "token_type" : "BearerToken",
  "refresh_token" : "HsnXmyIQqmJJQrFVdevmVztGGASUfBfz",
  "client_id" : "vn0zG4cnSWaWIzdwBZgnREI1NGORDXXz",
  "access_token" : "GRQAJcgSFZcklbIUxfoUaYFW2ROd",
  "organization_name" : "chrisv-cs",
  "refresh_token_expires_in" : "0",
  "refresh_count" : "0"
}
```

Verify OAuth token policy

The OAuthV2 policy’s *VerifyAccessToken* operation will validate the access token for subsequent requests for all grant types.

```
<?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>
```

Set the access token as the bearer token in the authorization header of the http request.

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

THANK YOU