OAuth - Authorization Code

Google Cloud

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?

Authorization Code - Actors

User

Authorization Code - Actors

User

User Agent

Authorization Code - Actors

User

User Agent

Client

Authorization Code - Actors

User

User Agent

Client

Apigee (Authorization Server)

apigee

Authorization Code - Actors

User

User Agent

Client

Apigee (Authorization Server)

Authentication Server (Login App)

Authorization Code - Actors

User

User Agent

Client

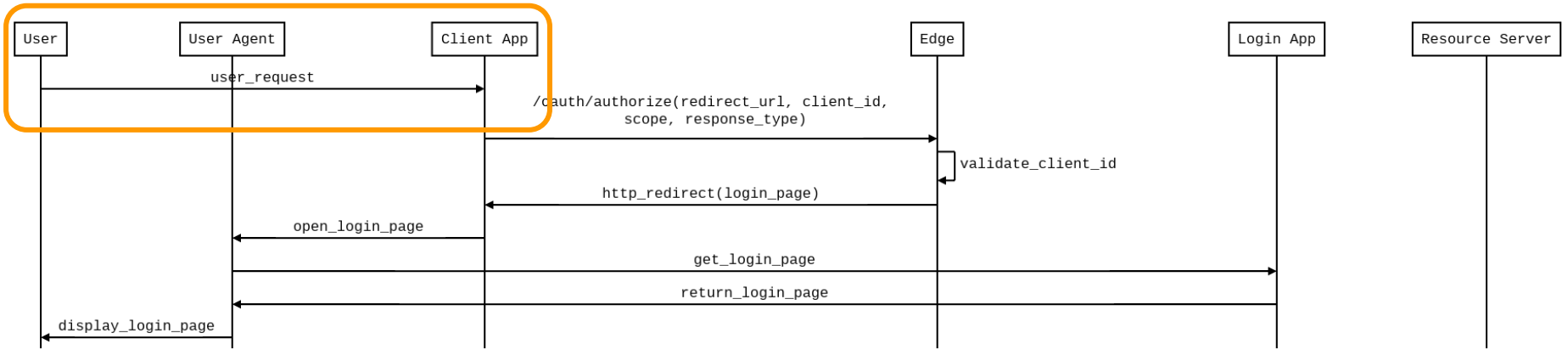
Apigee (Authorization Server)

Authentication Server (Login App)

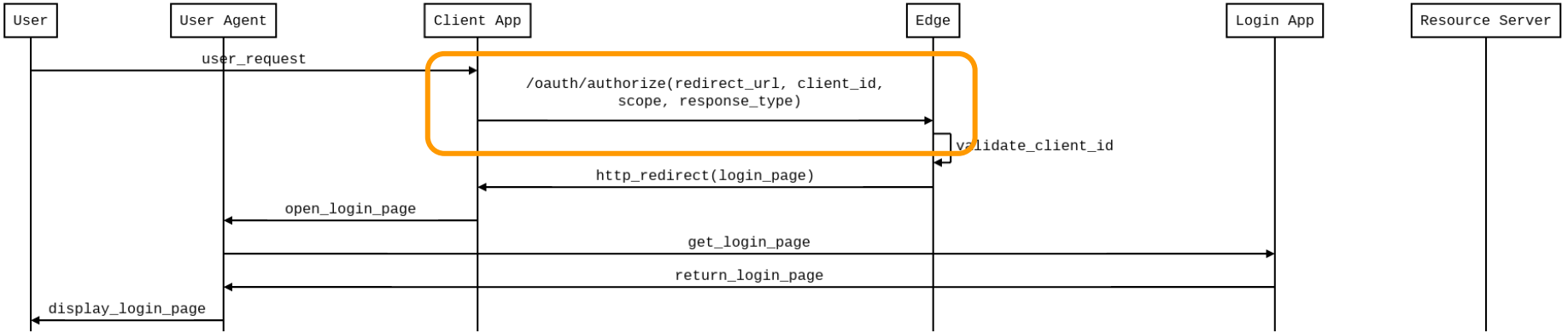
Resource Server

apigee

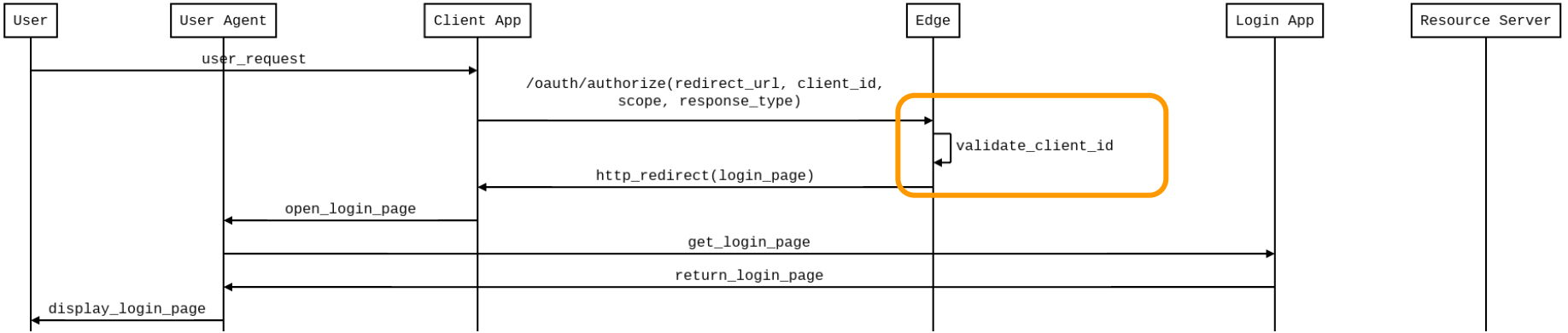
Authorization code - Sequence diagram



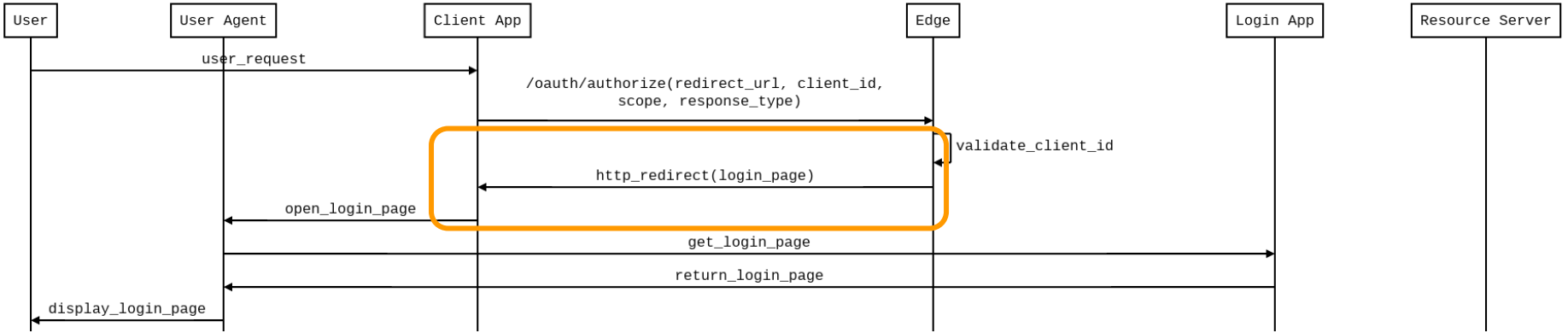
Authorization code - Sequence diagram



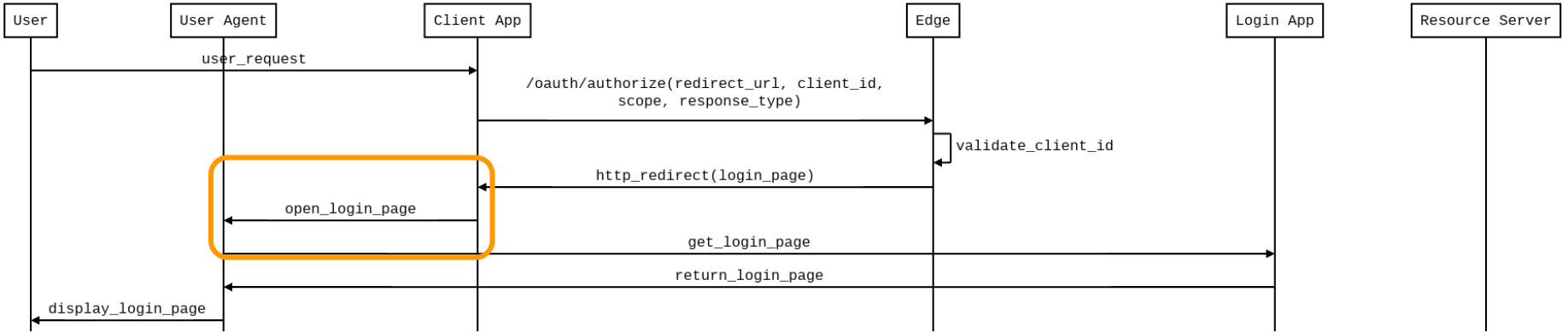
Authorization code - Sequence diagram



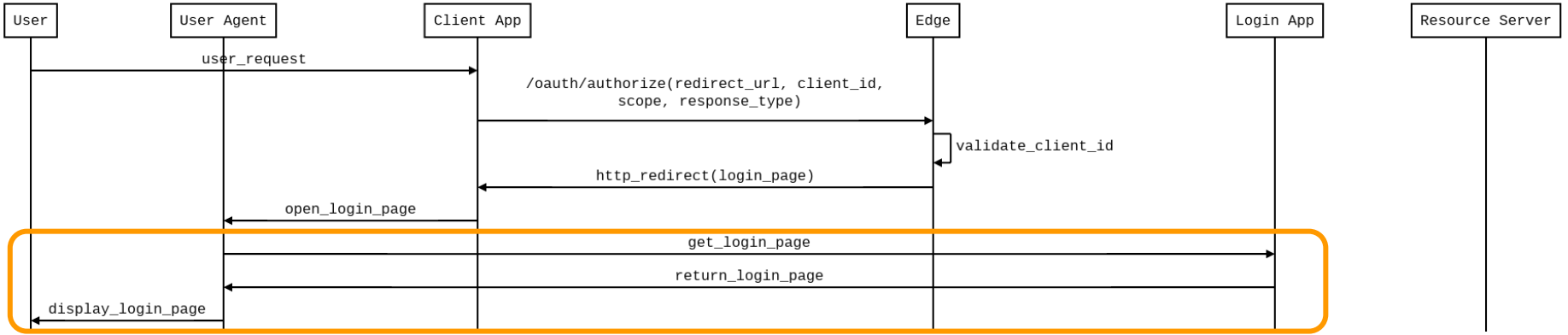
Authorization code - Sequence diagram



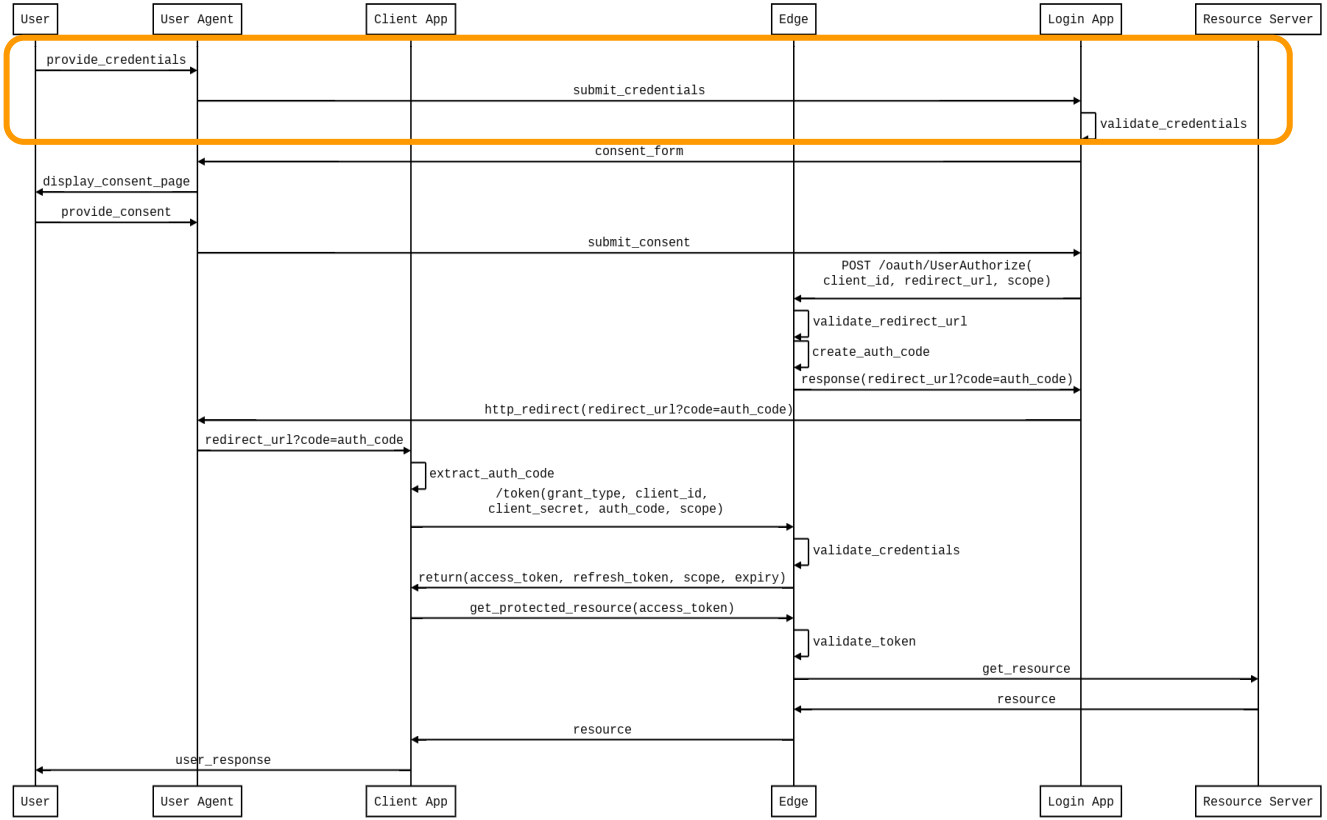
Authorization code - Sequence diagram



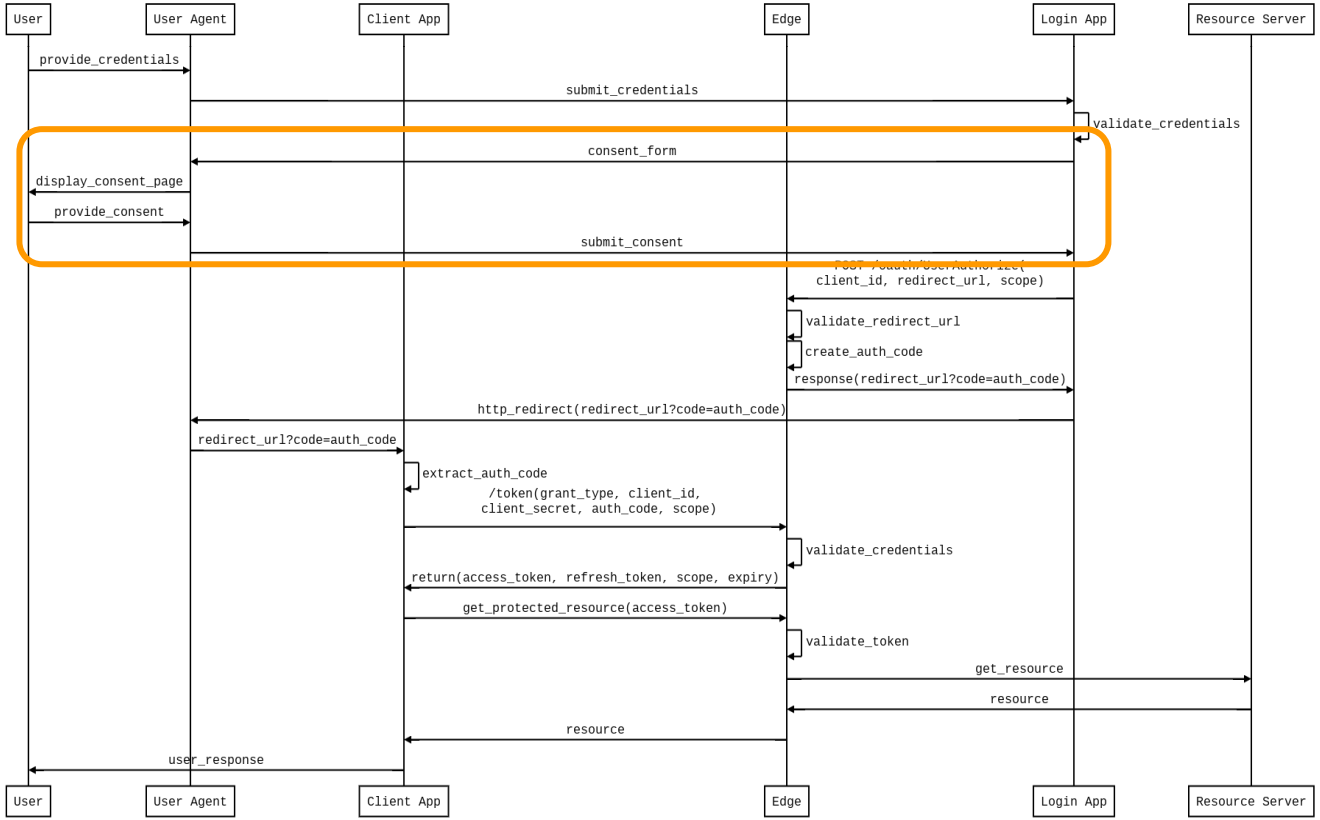
Authorization code - Sequence diagram



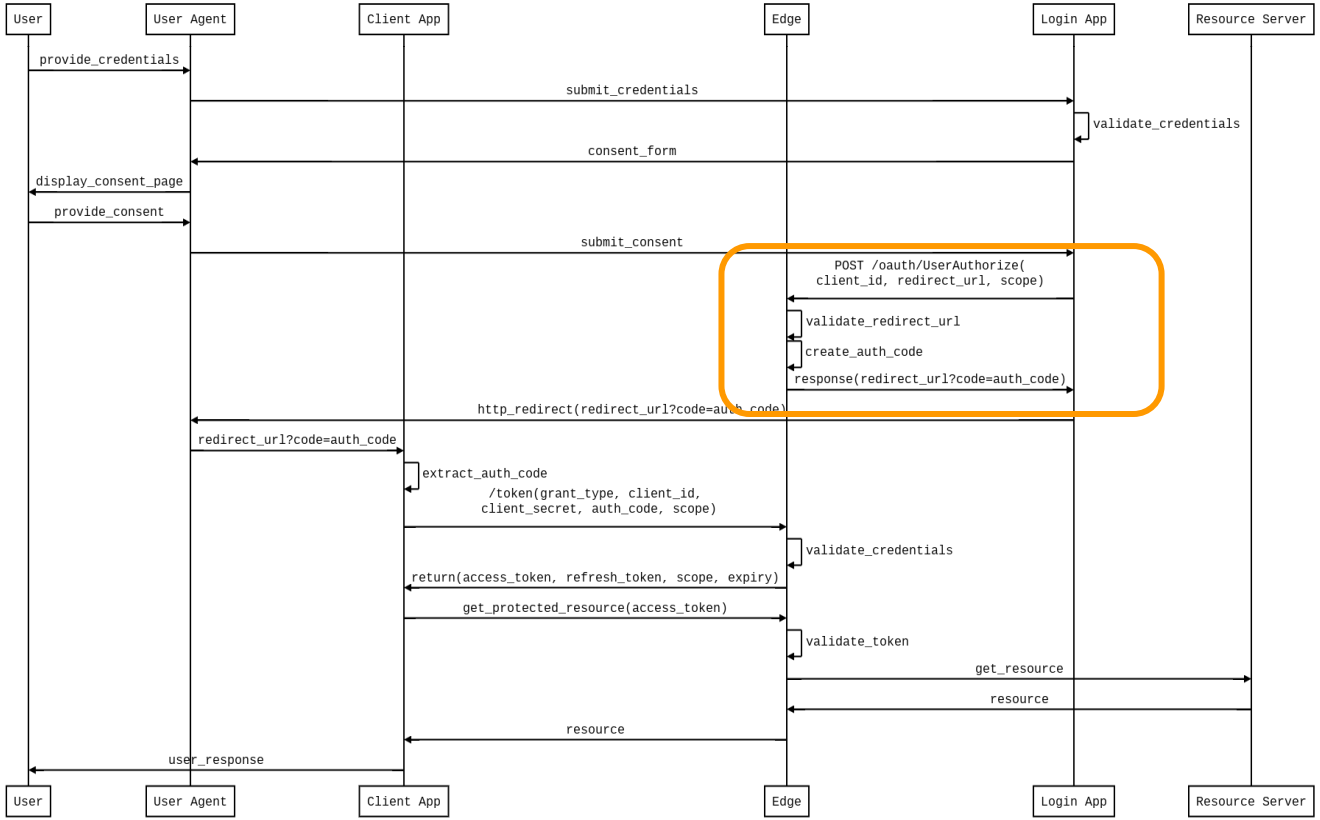
Authorization code - Sequence diagram



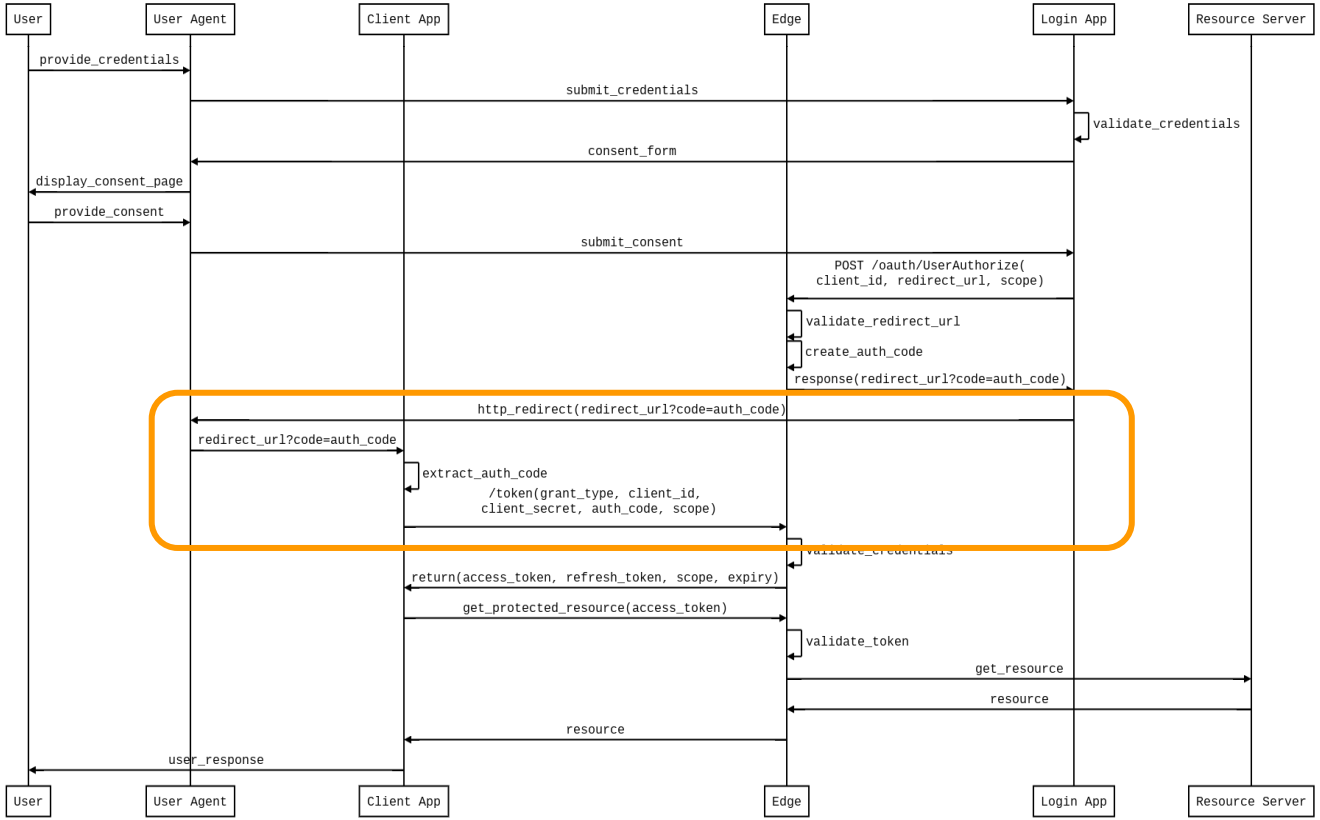
Authorization code - Sequence diagram



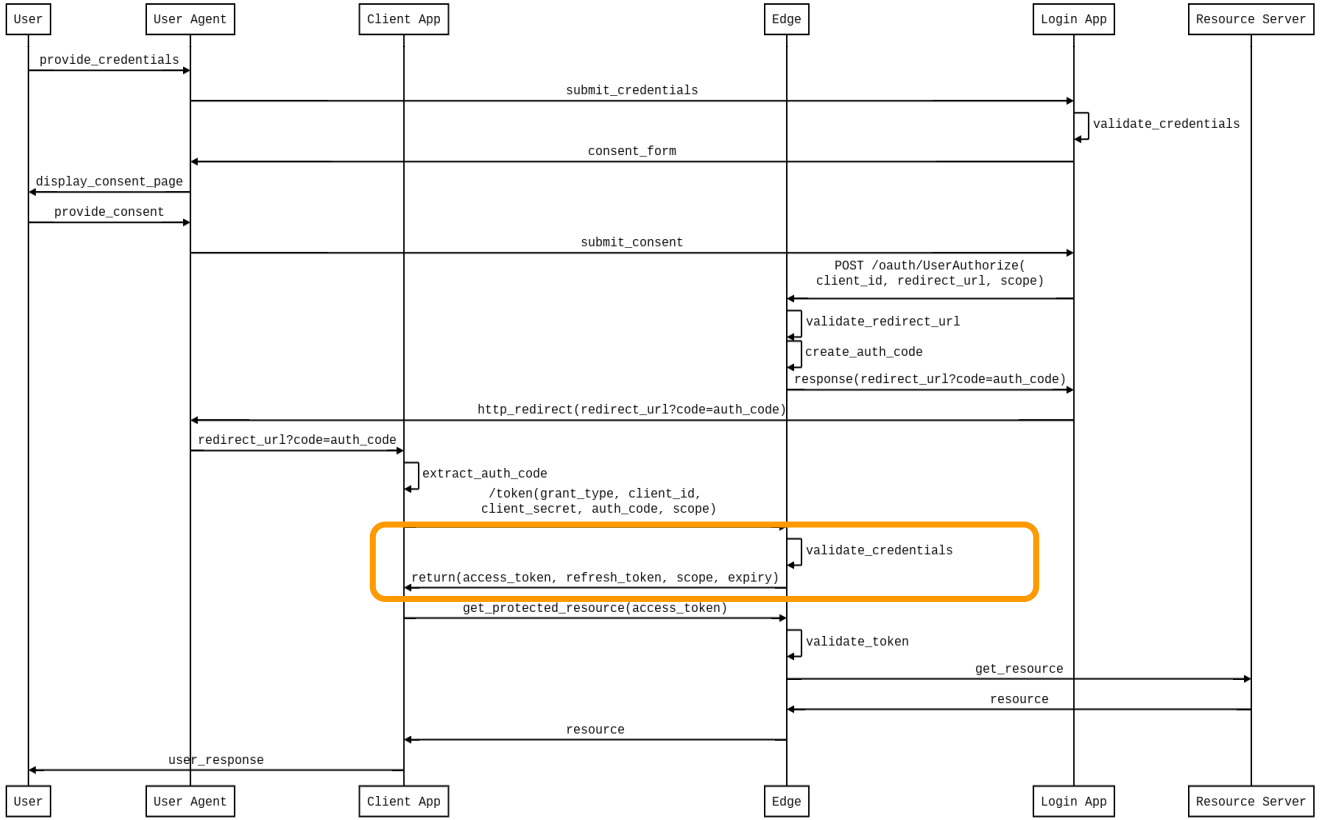
Authorization code - Sequence diagram



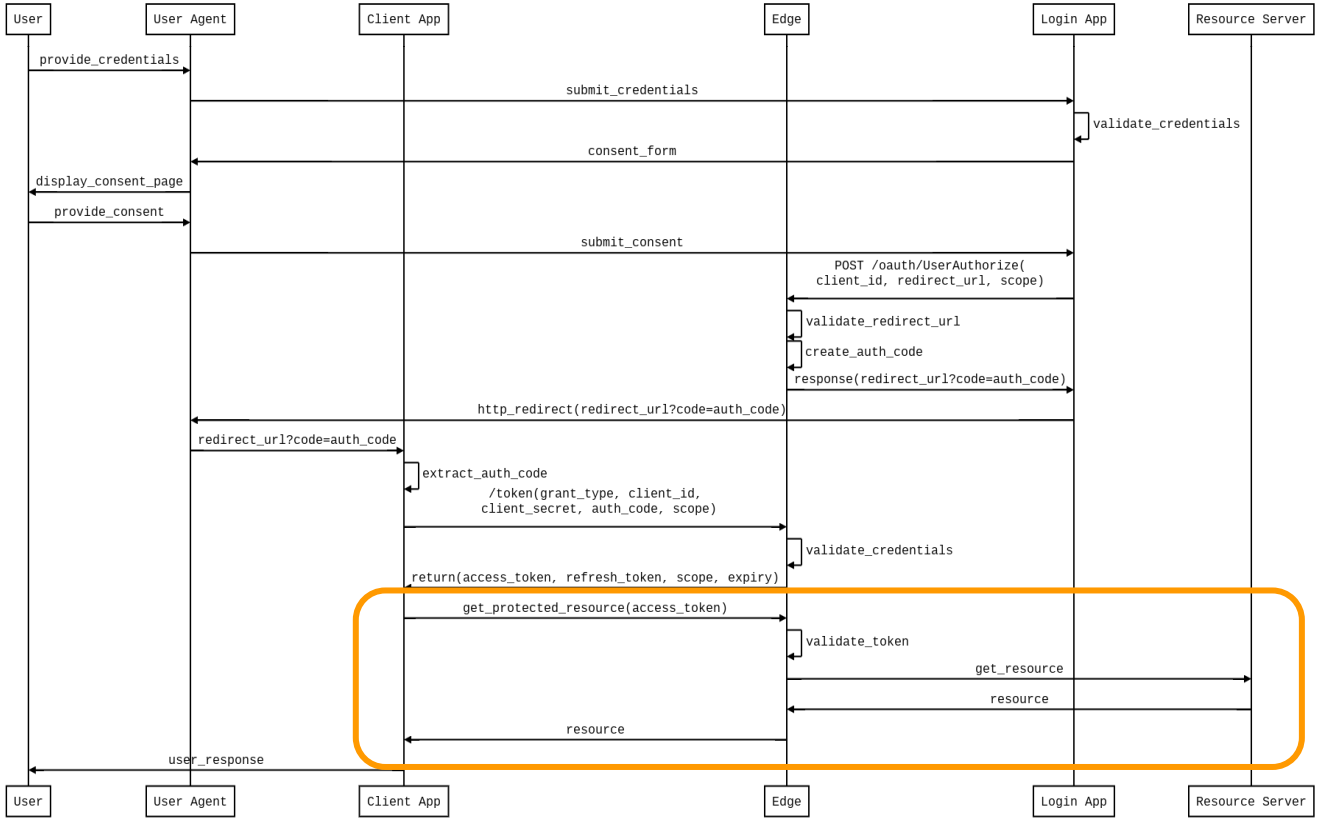
Authorization code - Sequence diagram



Authorization code - Sequence diagram



Authorization code - Sequence diagram



Generate Authorization Code

```
<OAuthV2 async="false" continueOnError="false" enabled="true" name="GetAuthCode">
    <DisplayName>GetAuthCode</DisplayName>
    <Operation>GenerateAuthorizationCode</Operation>
    <ExpiresIn>600000</ExpiresIn>
    <GenerateResponse/>
</OAuthV2>
```

apigee

Generate Authorization Code

```
<OAuthV2 async="false" continueOnError="false" enabled="true" name="GetAuthCode">
    <DisplayName>GetAuthCode</DisplayName>
    <Operation>GenerateAuthorizationCode</Operation>
    <ExpiresIn>600000</ExpiresIn>
    <GenerateResponse/>
</OAuthV2>
```

Exchange Authorization Code for Access Token

```
<OAuthV2 name="GetAccessToken">
    <Operation>GenerateAccessToken</Operation>
    <ExpiresIn>360000000</ExpiresIn>
    <SupportedGrantTypes>
        <GrantType>authorization_code</GrantType>
    </SupportedGrantTypes>
    <GrantType>request.queryparam.grant_type</GrantType>
    <GenerateResponse/>
</OAuthV2>
```

apigee

Exchange Authorization Code for Access Token

```
<OAuthV2 name="GetAccessToken">
    <Operation>GenerateAccessToken</Operation>
    <ExpiresIn>360000000</ExpiresIn>
    <SupportedGrantTypes>
        <GrantType>authorization_code</GrantType>
    </SupportedGrantTypes>
    <GrantType>request.queryparam.grant_type</GrantType>
    <GenerateResponse/>
</OAuthV2>
```

apigee

Exchange Authorization Code for Access Token

```
<OAuthV2 name="GetAccessToken">
    <Operation>GenerateAccessToken</Operation>
    <ExpiresIn>360000000</ExpiresIn>
    <SupportedGrantTypes>
        <GrantType>authorization_code</GrantType>
    </SupportedGrantTypes>
    <GrantType>request.queryparam.grant_type</GrantType>
    <GenerateResponse/>
</OAuthV2>
```

apigee

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>
```

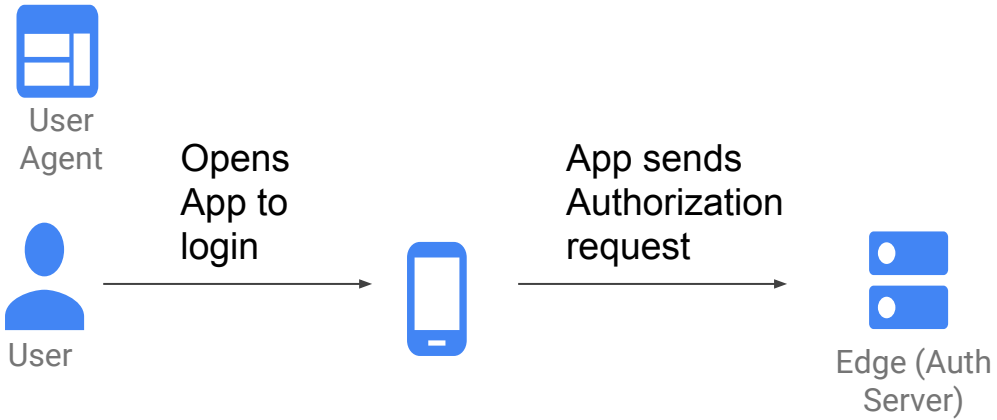
Verify OAuth Token Policy

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

Thank You

Google Cloud

Authorization code - Sequence diagram



Authorization code - Sequence diagram

User	Client Application	Apigee Edge Authorization Server
Opens mobile application	Mobile application sends authorization request to Apigee Edge	Validates the client ID and the redirect URI in the request

apigee

Authorization code - Sequence diagram

User prompted to login

Opens the login page

Accepts response and opens a web browser

Responds with a redirect request

apigee

User | Authentication Server | User Agent

Login

Authenticate user credentials and respond with consent web page

Displays Consent Page

apigee

Authorization code - Sequence diagram

User	Authentication Server	Apigee Edge
Provides consent/allows application to access their data	Saves consent and sends authorize request to Apigee Edge	Validates the client ID and the redirect URI in the request and generates authorization code

Authorization code - Sequence diagram

Client Application	User Agent	Authentication Server	Apigee Edge
Extract the authorization code	Forwards request to Mobile application	Forwards redirect to user's agent	Responds with auth code included in URI




apigee

Authorization code - Sequence diagram

Client Application

Apigee Edge

Sends request for access token and include auth code

Validate auth code and generate access token

apigee

Authorization code - Sequence diagram