

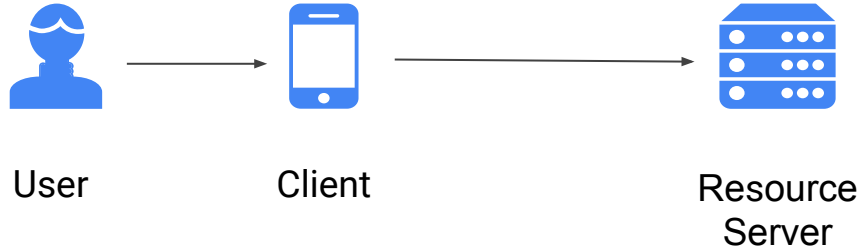


OAuth - Implicit Grant Type

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?



Implicit Grant - Actors



User

Implicit Grant - Actors



User



User
Agent

Implicit Grant - Actors



User



User
Agent



Client

Implicit Grant - Actors



User



User
Agent



Client



Apigee
(Authorization
Server)

Implicit Grant - Actors



User



User
Agent



Client



Apigee
(Authorization
Server)



Authentication
Server
(Login App)

Implicit Grant - Actors



User



User
Agent



Client



Apigee
(Authorization
Server)

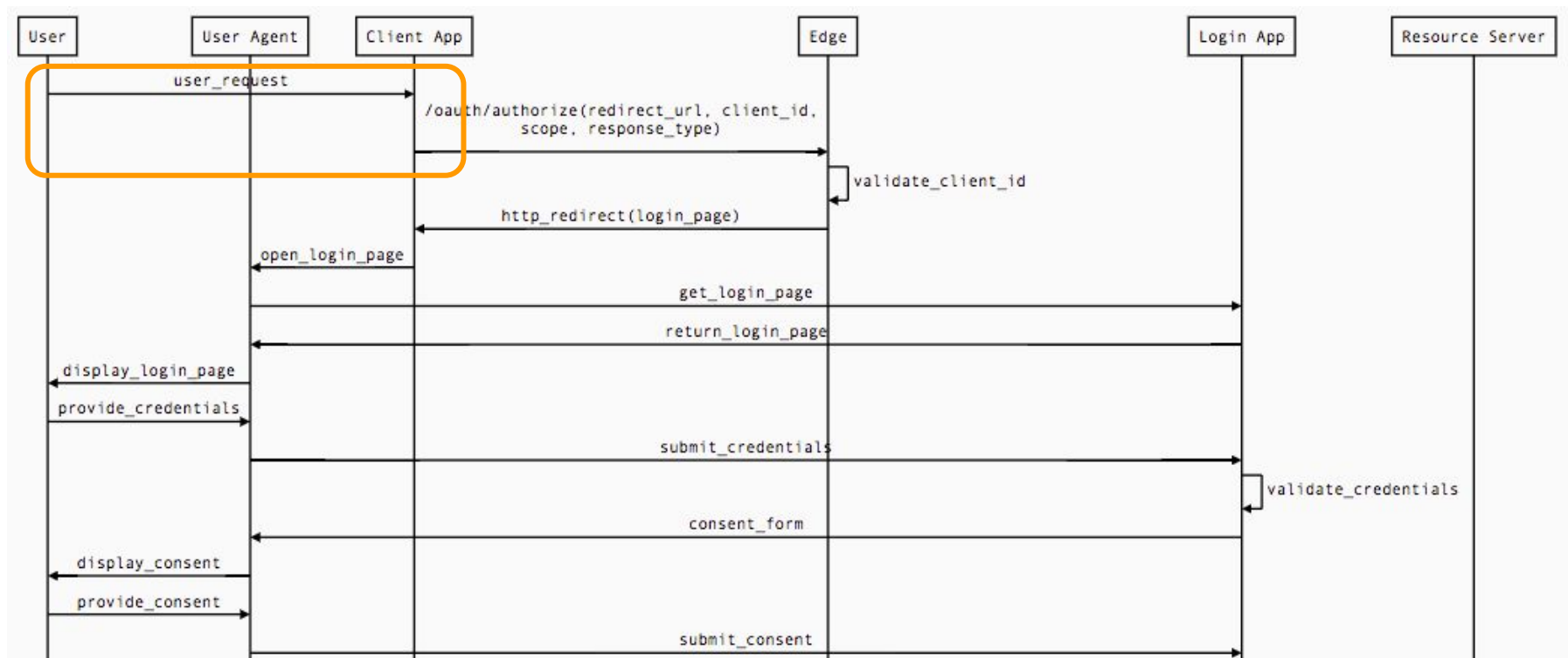


Authentication
Server
(Login App)



Resource
Server

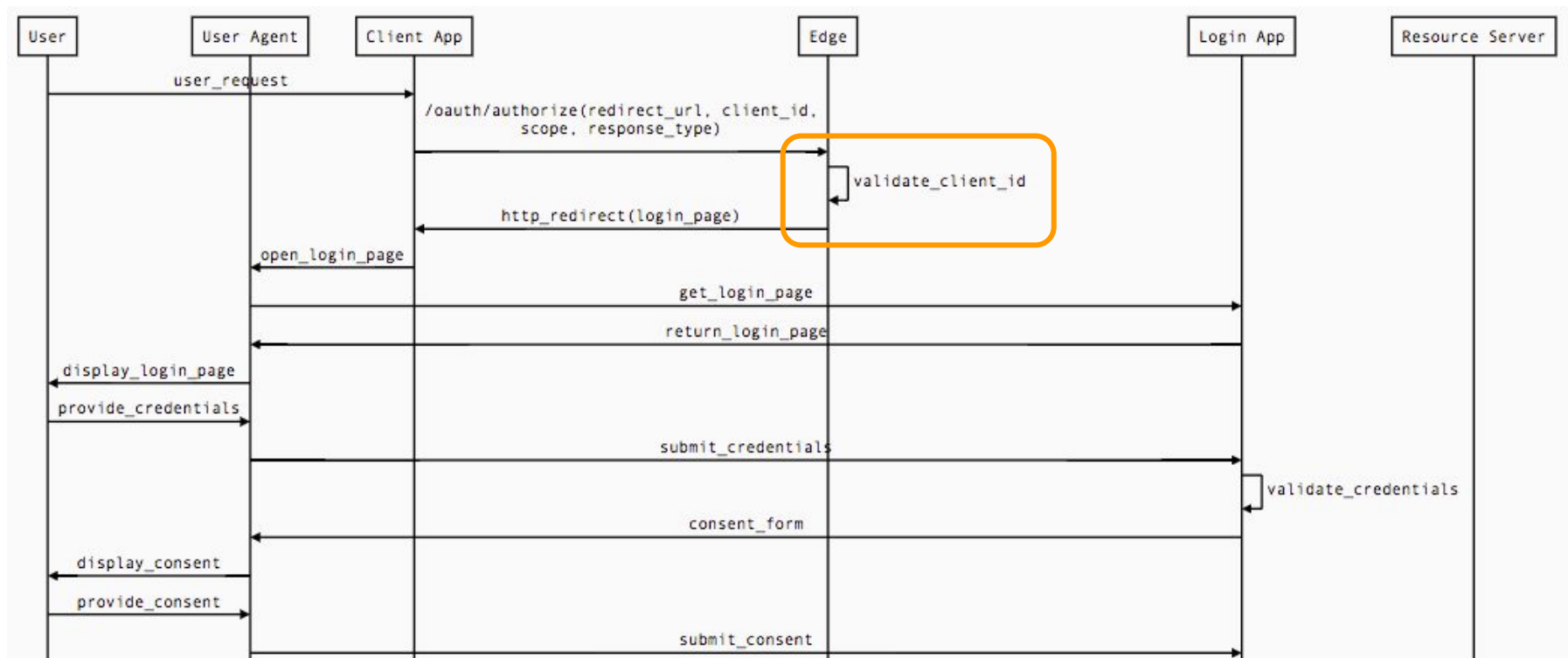
Implicit - Sequence diagram



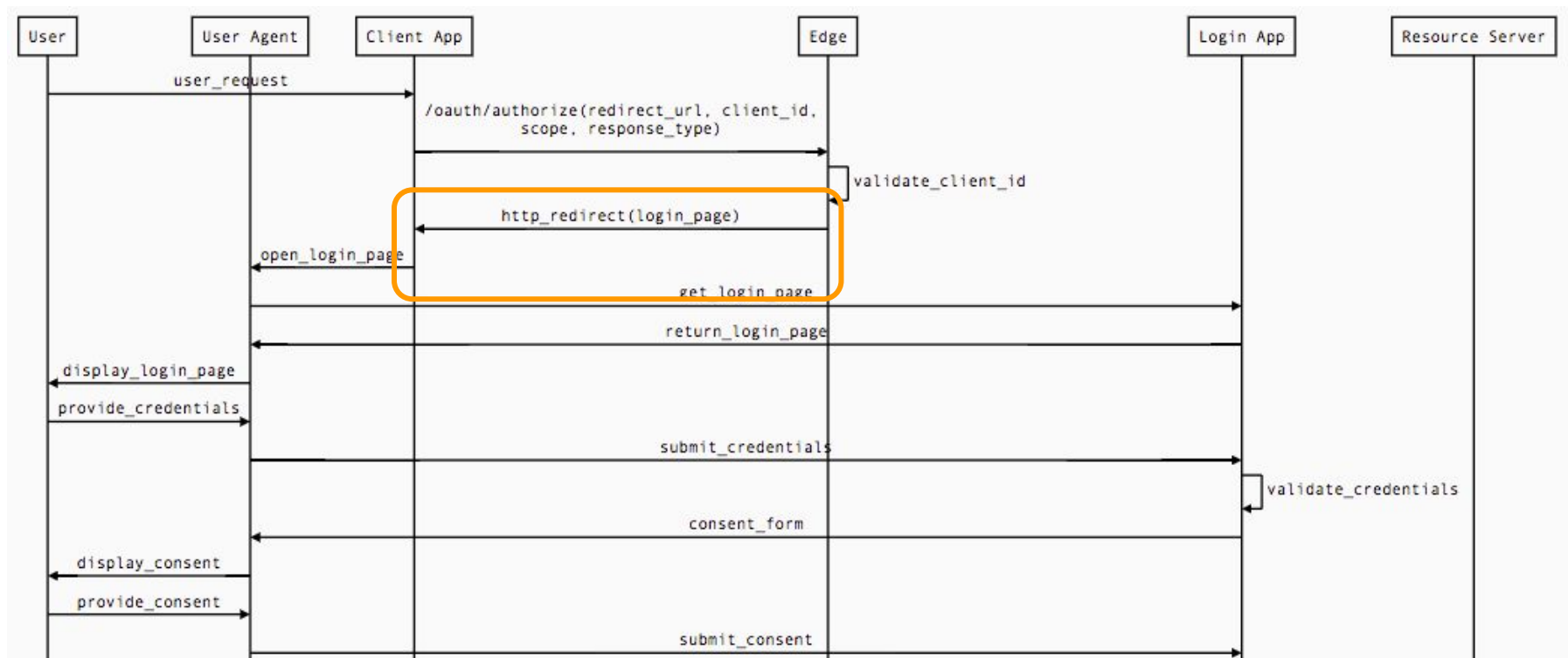
Implicit - Sequence diagram



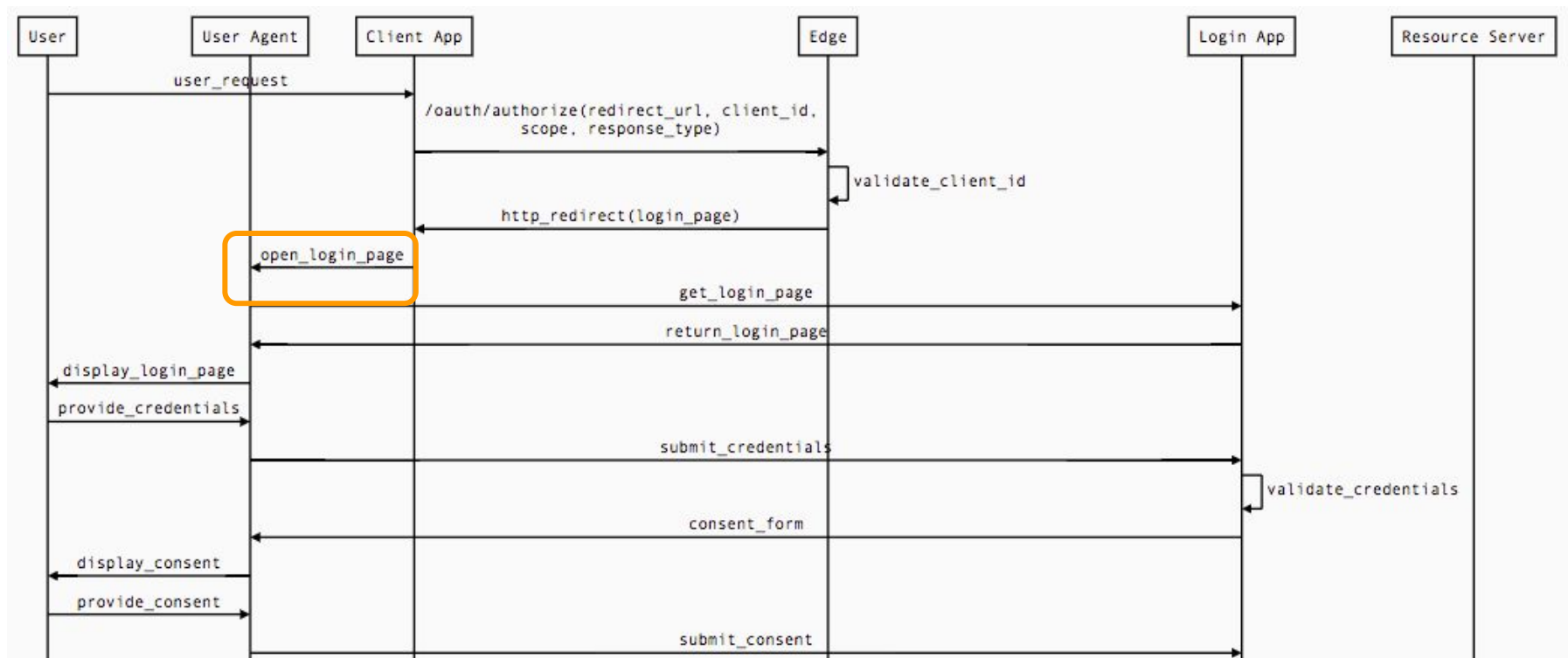
Implicit - Sequence diagram



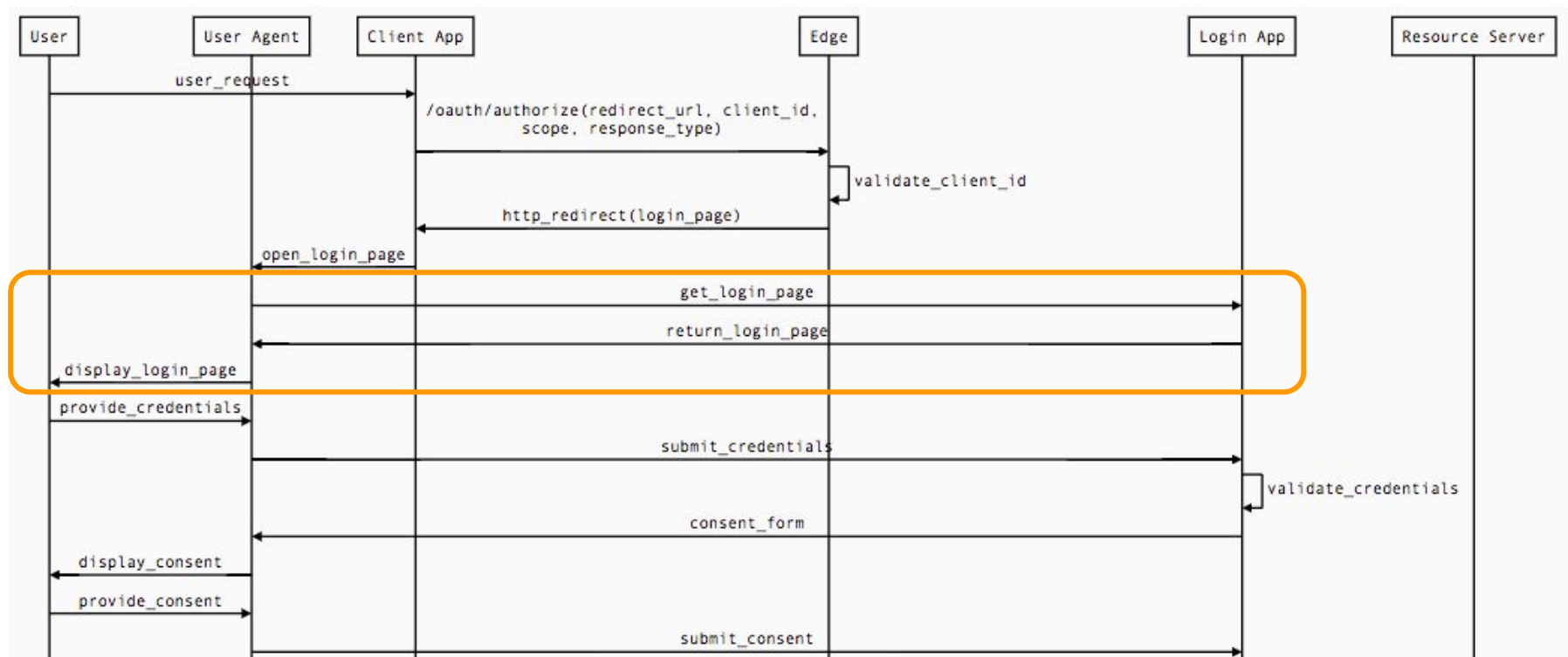
Implicit - Sequence diagram



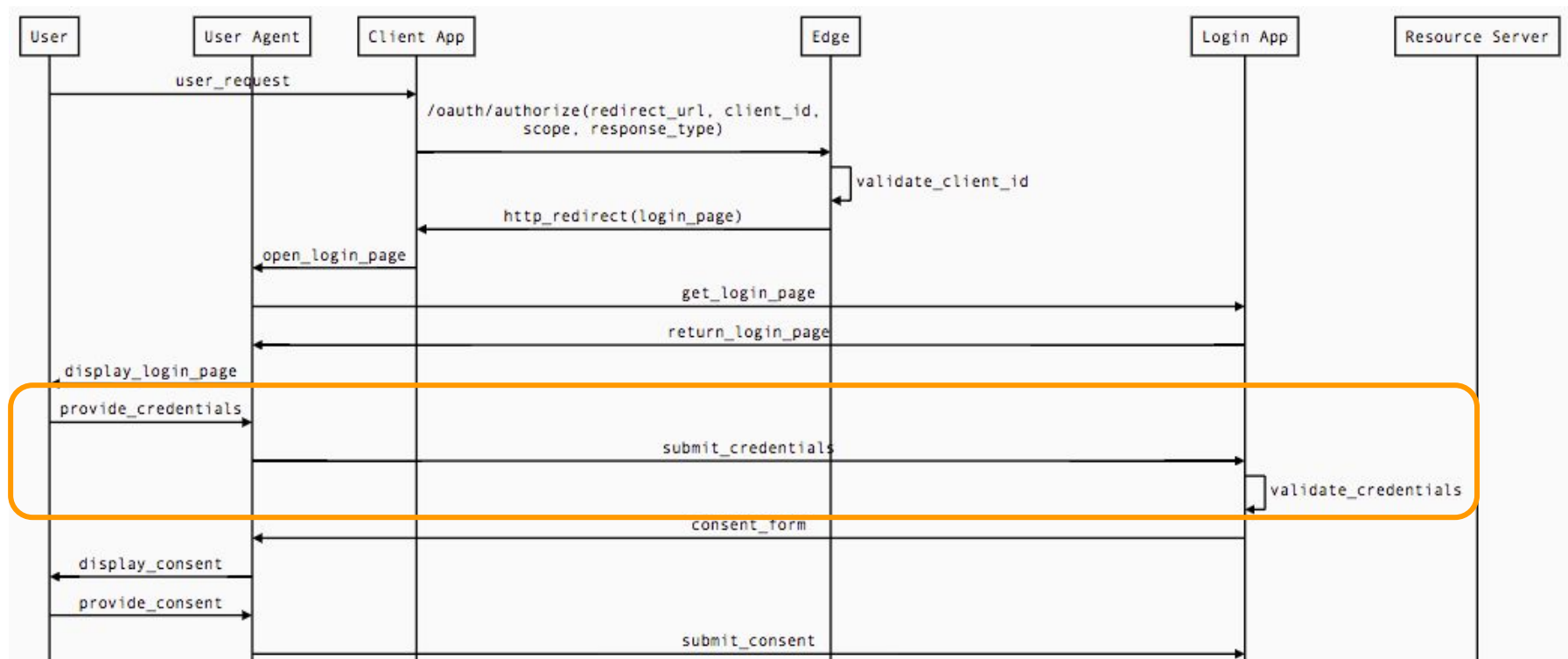
Implicit - Sequence diagram



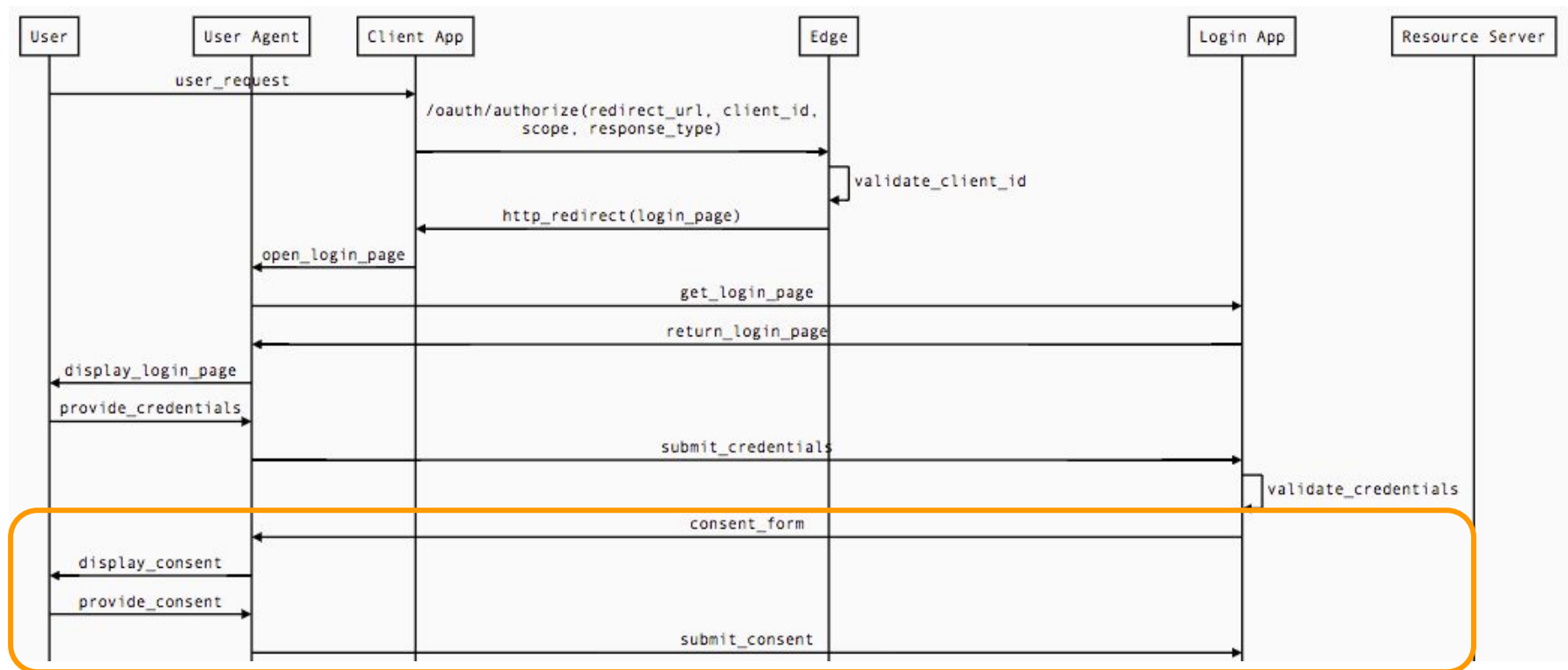
Implicit - Sequence diagram



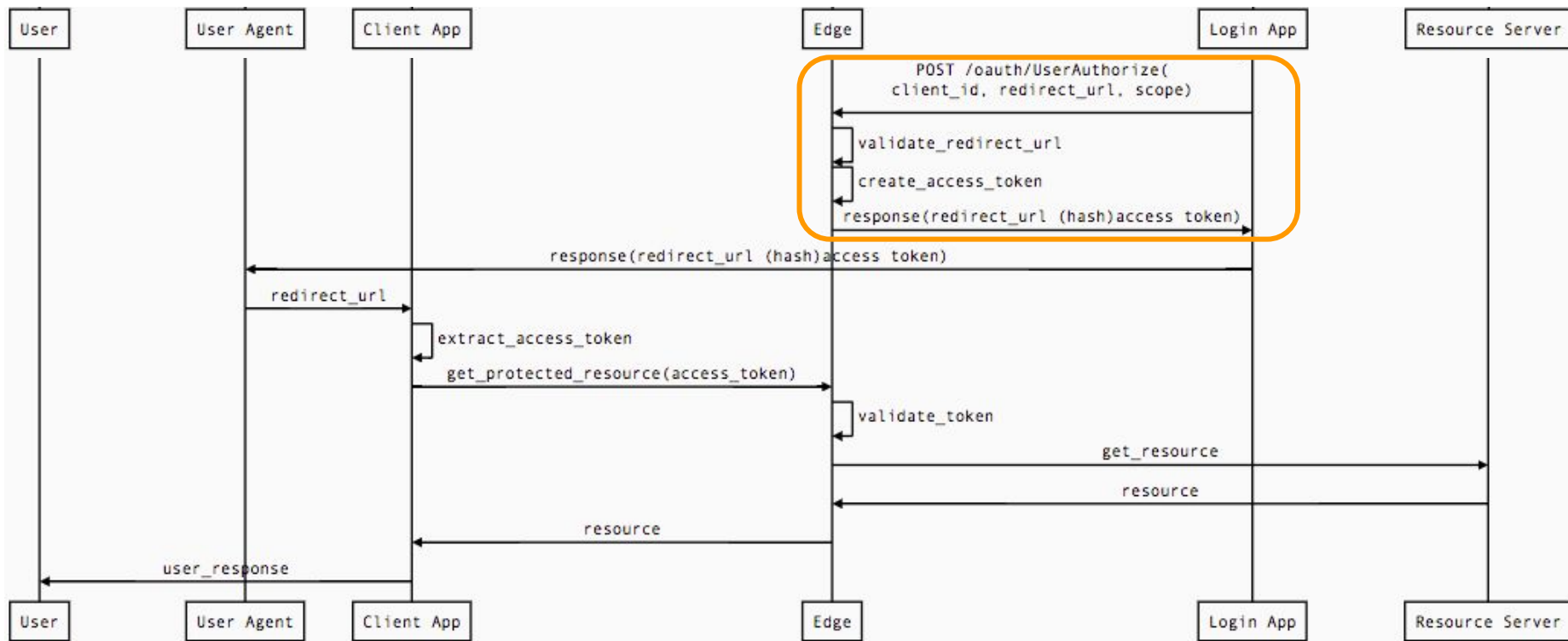
Implicit - Sequence diagram



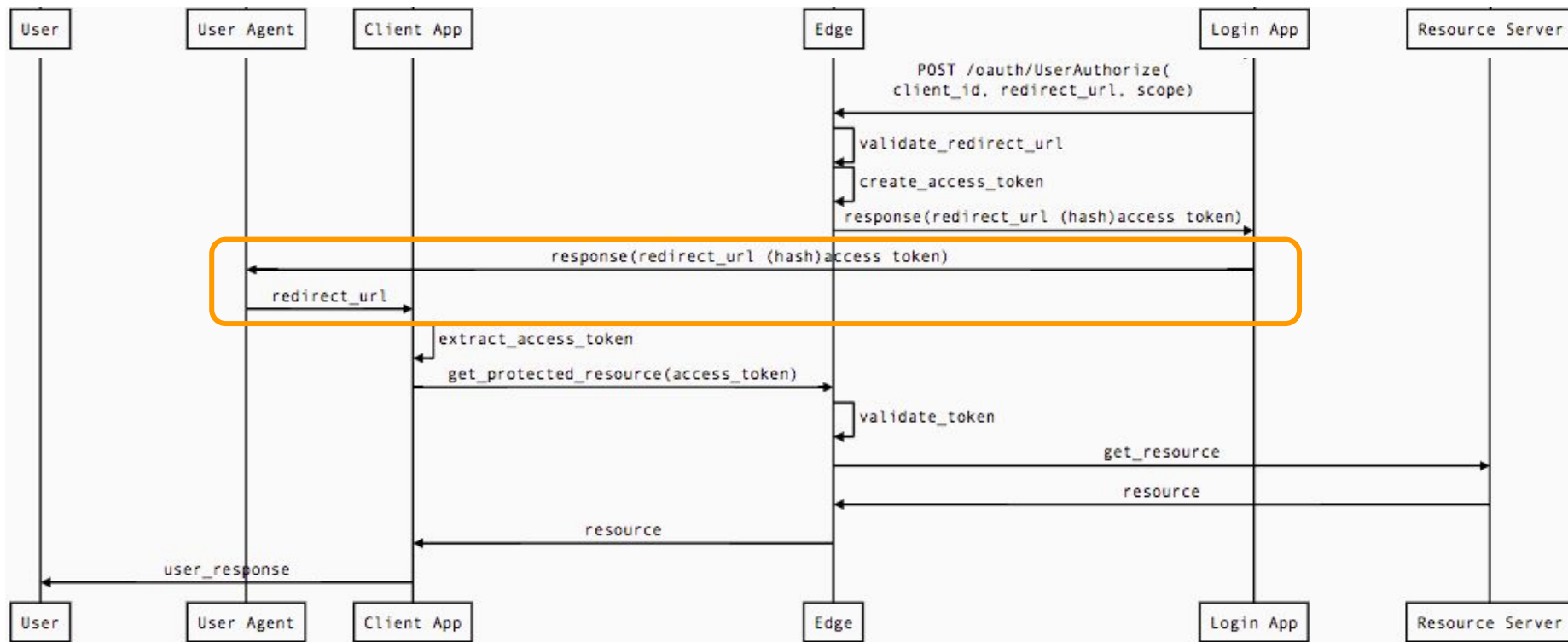
Implicit - Sequence diagram



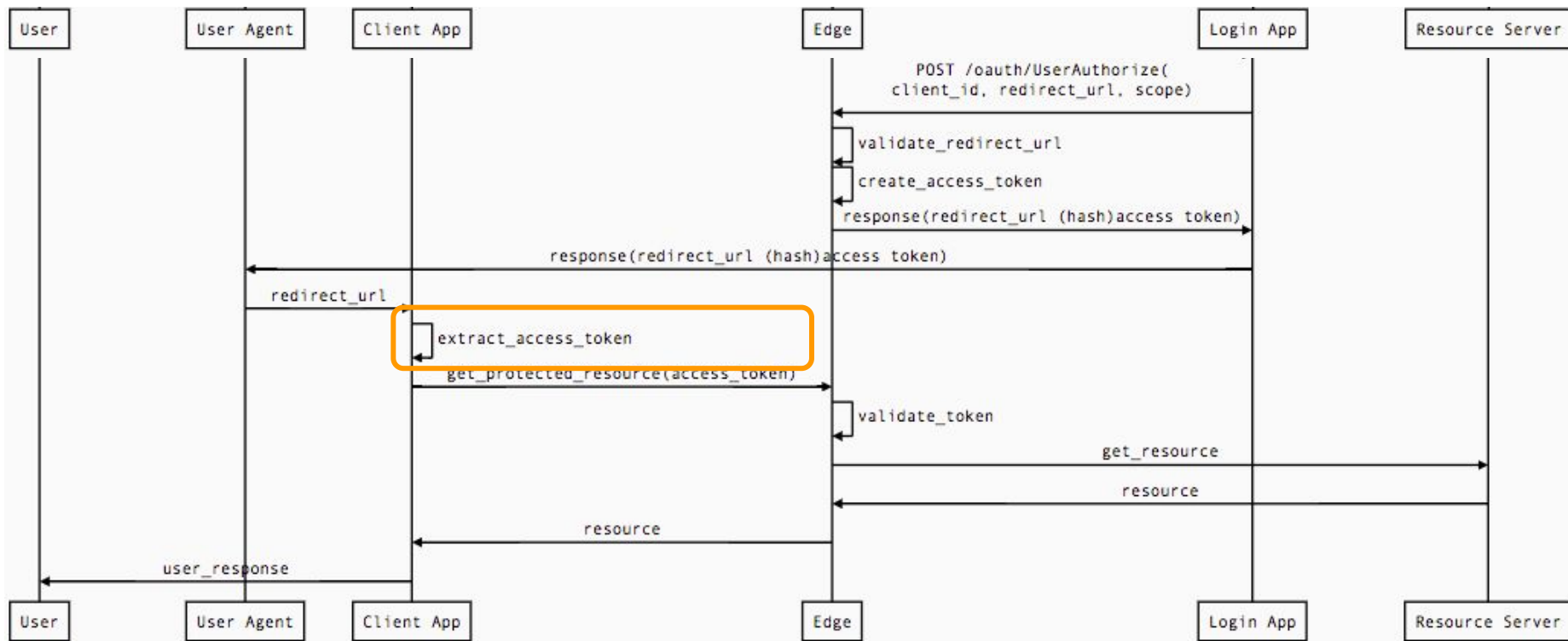
Implicit - Sequence diagram



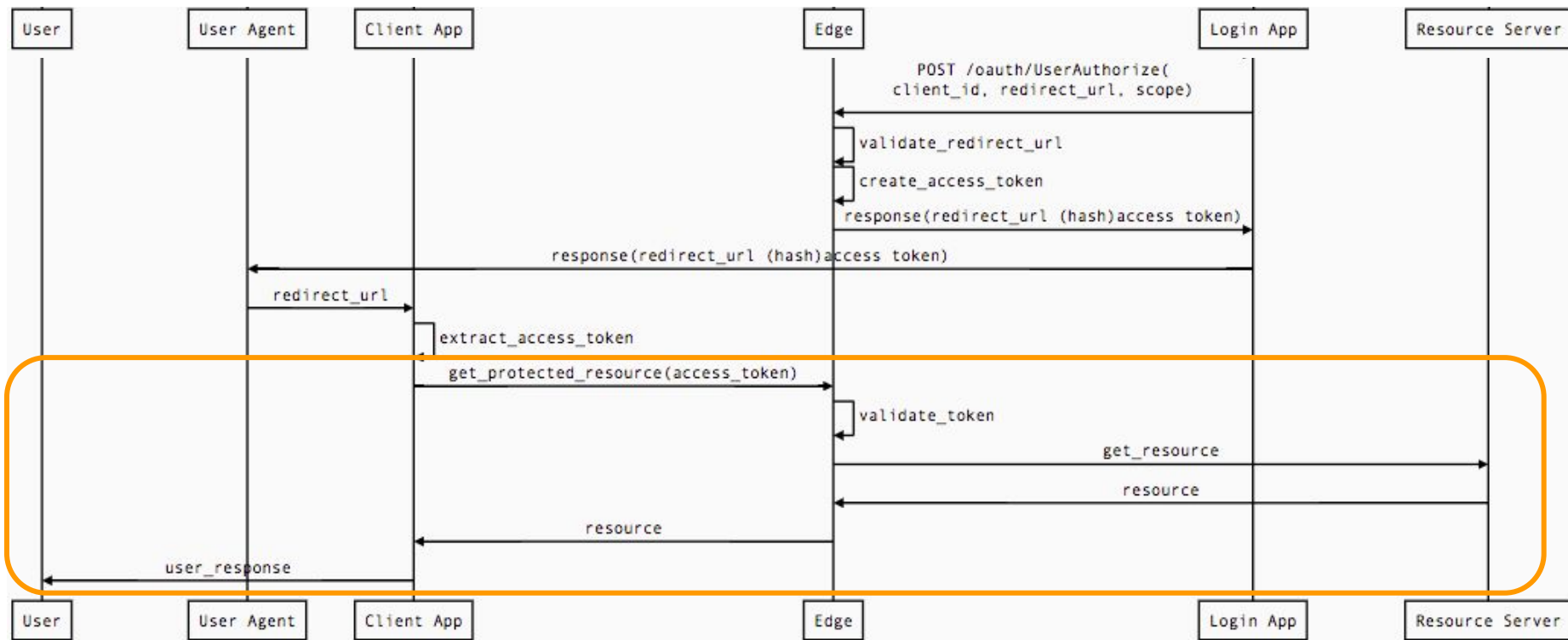
Implicit - Sequence diagram



Implicit - Sequence diagram



Implicit - Sequence diagram



Generate implicit access token

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

Generate implicit access token

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

Verify Access 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>
```


Request to Apigee

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



Thank You

Implicit grant type

- A simplified version of authorization code
- Resources participating in authorization code grant type
 - End user
 - Client application
 - User agent
 - Edge - generating and validating token
 - Authentication server - to validate the credentials
 - Backend API resource
- Used when the app resides on the client
 - For example, the app's code is implemented in a browser using JavaScript
- Authorization server returns an access token directly when the user is authenticated, rather than issuing an authorization code first
- Can improve app responsiveness in some cases
- This advantage needs to be weighed against possible security implications as described in the IETF specification

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>VerifyOAuthToken</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
```

Implicit - Sequence diagram

