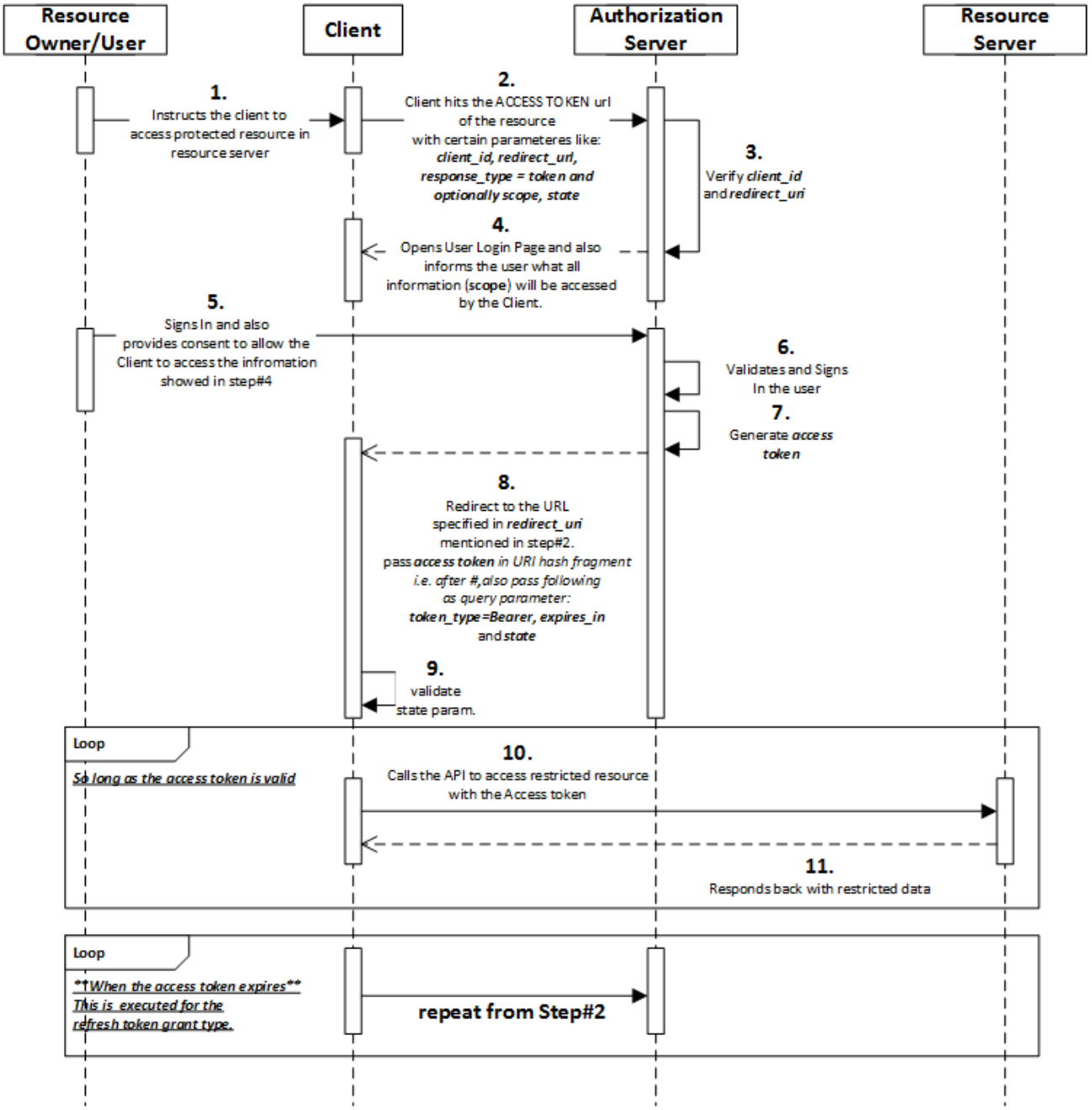

1) Resource Owner/User instructs the Client to access its protected resource in the Resource Server.

2) Upon getting the user instruction to access a protected resource, the Client sends a request to the access token end point of with following parameters:

- **client_id** – this is the unique identification number of the client.
- **redirect_uri** – This is the URL where the server will redirect the user after s/he completes the authorization process.
- **state** – Client sends a request to access token end-point with certain value in this parameter. When the authorization server sends a response back to redirect_uri, it sends back the value of state parameter unchanged. Then the Client checks this

value to ensure that if it's indeed same value it sent as part of token request. Thus, client can ensure that it receives the response for the correct request.

- **scope** – For which the Client is requesting authorization. These are space delimited list of scope string; for example – profile, email, location etc.
- **response_type** – It's defaulted to **token**. This denotes that the request is for obtaining the access token.

3) The Authorization server verifies the client_id in the request with the stored client_id. It may also optionally validate based on the redirect_uri.

4) Thereafter, it opens the sign in web-page and also mention what all information(scope) will be accessed by the Client.

5) User or the Resource Owner provides credentials and the consent that the Client can access the list of scope.

6) If the provider credential is correct, user is signed in and consents are recorded. If the user doesn't provide the consent, the entire process halts and exception process kicks in.

7) After successful signing in and record of user consent, Authorization server generates access token.

8) Redirect the user's browser to the URL specified in redirect_uri and this response contains the access token. Authorization server also passes the state parameter without change along with following information:

- **access_token** – Client uses this is the token in further calls to the actual resource. This token is passed in URI hash fragment (#). It's to note that, this parameter will not be passed as query parameter.
- **token_type** – with value Bearer
- **expires_in** – This integer value is the TTL of the access_token
- **state** – The same value that was passed in Step#2

9) Client will compare the value of state parameter to ensure that it's the same value that client sent in Step 2. This is ti avoid any CSRF attack.

10/11) The Client uses the access token to hit the protected resource URL and accesses the protected data. When the access token expires, begin from Step 2.