Oracle APEX Social Login

Marc Sewtz

June 5th, 2019

Step Up to Modern Cloud Development

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Contents

1. **Authentication in Oracle APEX**
2. OAuth 2.0
3. Using Google Authentication
4. Using Facebook Authentication
5. Using LinkedIn Authentication
6. Authorization in Oracle APEX

APEX Authentication

Establishing User Identity Through Authentication

- Authentication establishes the identity of each user who accesses your application.
- Most authentication processes require that a user provide some type of credentials such as a user name and password.

APEX Authentication

Establishing User Identity Through Authentication

- When creating an application, you can choose to:
  - **Not require authentication**: Oracle Application Express does not check any user credentials. All pages of your application are accessible to all users.
  - **Select a built-in authentication scheme**: Create an authentication method based on available preconfigured authentication schemes.
  - **Create custom authentication scheme**: Provided complete control over the authentication interface. To implement this approach, you must provide a PL/SQL function the APEX engine executes before processing each page request. This function's Boolean return value determines whether the Application Express engine processes the page normally or displays a failure page.

APEX Authentication

Preconfigured Authentication Schemes

- **Application Express Accounts**: User accounts that are created within and managed in the Oracle Application Express user repository. When you use this method, your application is authenticated against these accounts.
- **Custom Authentication**: Creating a Custom Authentication scheme from scratch to have complete control over your authentication interface.
- **Database Accounts**: Database Account Credentials authentication utilizes database schema accounts to authenticate users.
- **HTTP Header Variable**: Authenticate users externally by storing the username in a HTTP Header variable set by the web server.
- **LDAP Directory Verification**: Authenticate a user and password with an authentication request to a LDAP server.

APEX Authentication

Preconfigured Authentication Schemes

- **No Authentication (using DAD)**: Adopts the current database user. This approach can be used in combination with a mod_plsql Database Access Descriptor (DAD) configuration that uses basic authentication to set the database session user.
- **Open Door Credentials**: Enable anyone to access your application using a built-in login page that captures a user name.
- **Oracle Single Sign-On** : Delegates authentication to the Oracle Application Server Single Sign-On (SSO) Server. To use this authentication scheme, your site must have been registered as a partner application with the SSO server.
- **Social Sign-In**: Supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.

APEX Authentication

Social Log In

- Supports authentication with Google, Facebook, and other social network that supports OpenID Connect or OAuth2 standards.
- Social Login for application was introduced in APEX 18.1
- Social Login for the Builder / APEX IDE is being added in APEX 19.1
- Social Login uses Web Credentials, which are stored on the Workspace level
- Web Credentials are not included in the export of an application
- When installing applications that use Social Login, you will be prompted to provide the Client ID (username) and Client Secret (password)

APEX Authentication

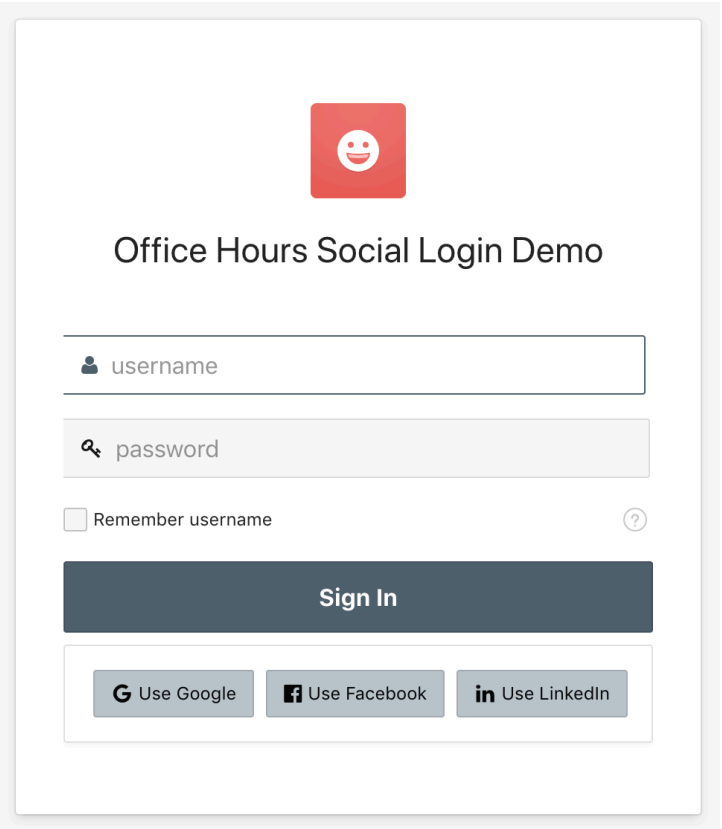
Deep Linking

- Applications that use authentication schemes support deep linking.
- Deep linking refers to the ability to link to an APEX page out of context, e.g. from a hyperlink in an email or workflow notification.
- It can be enabled in "***Session Management***" in the application's security attributes.
- When you link to a page out of context and the application requires the user be authenticated, the user is taken to the login page.
- After credentials verification, APEX automatically displays the page that was referenced in the original link.

APEX Authentication

Using multiple authentication schemes in one app

- Some applications may require more than one authentication scheme
- For example, if you wish to allow your users to login with a social network account of their choice (e.g. Google, Facebook, LinkedIn, etc.).
- To enable multiple authentication schemes:
  - Enable “Switch in Session”
    (see “login processing” in your authentication scheme)
  - If enabled, the current session's authentication scheme can be changed by passing *APEX_AUTHENTICATION=scheme name* in a URL's request parameter.



1 Authentication in Oracle APEX

2 **OAuth 2.0**

3 Using Google Authentication

4 Using Facebook Authentication

5 Using LinkedIn Authentication

6 Authorization in Oracle APEX

OAuth 2.0

Overview

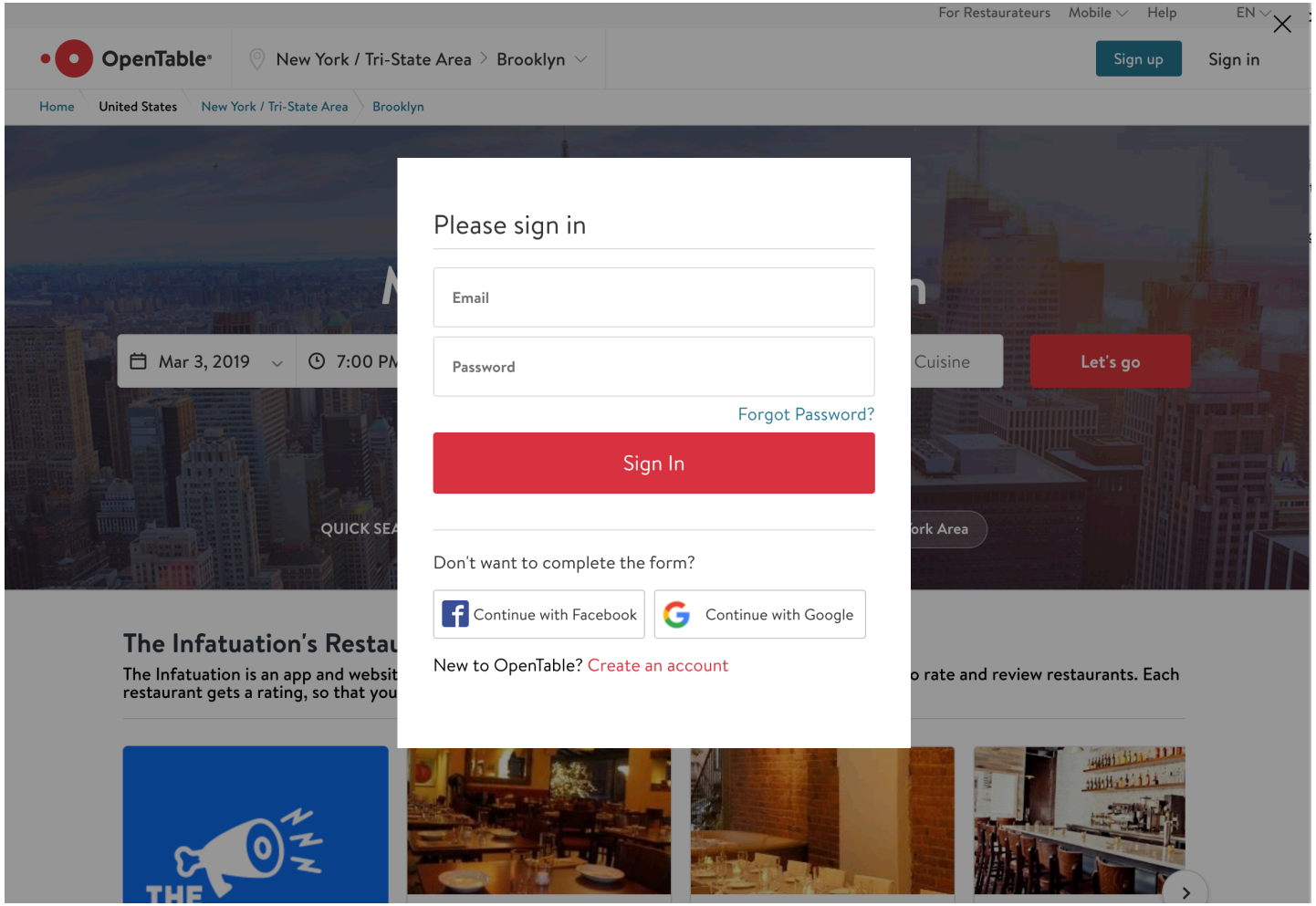
- OAuth 2.0 is an authorization framework that enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, Google, LinkedIn, Azure, Okta, etc.
- It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account.
- OAuth 2 provides authorization flows for web applications, desktop applications and mobile devices.

OAuth 2.0

OAuth 2.0 Roles

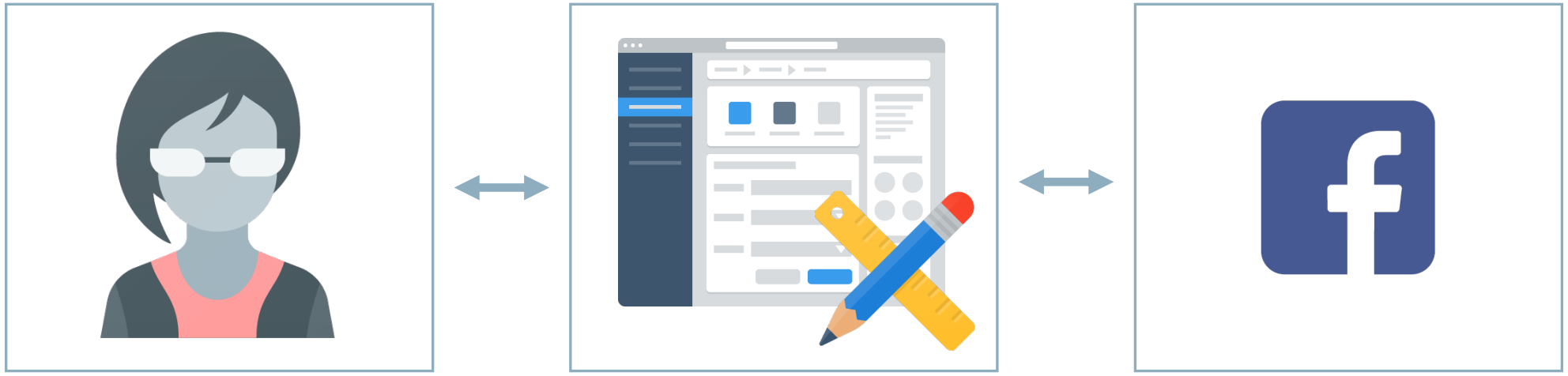
OAuth 2.0

OAuth 2.0 Roles



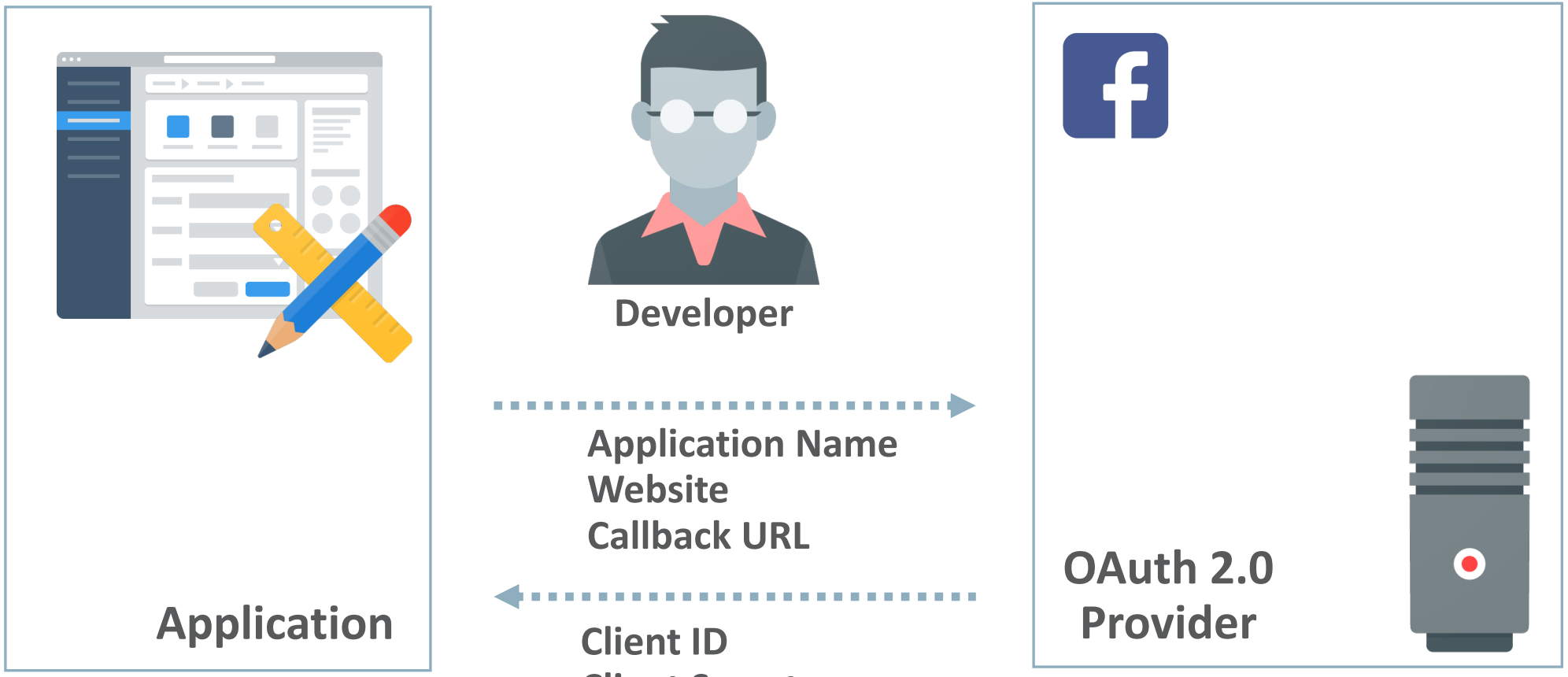
OAuth 2.0

OAuth 2.0 Roles



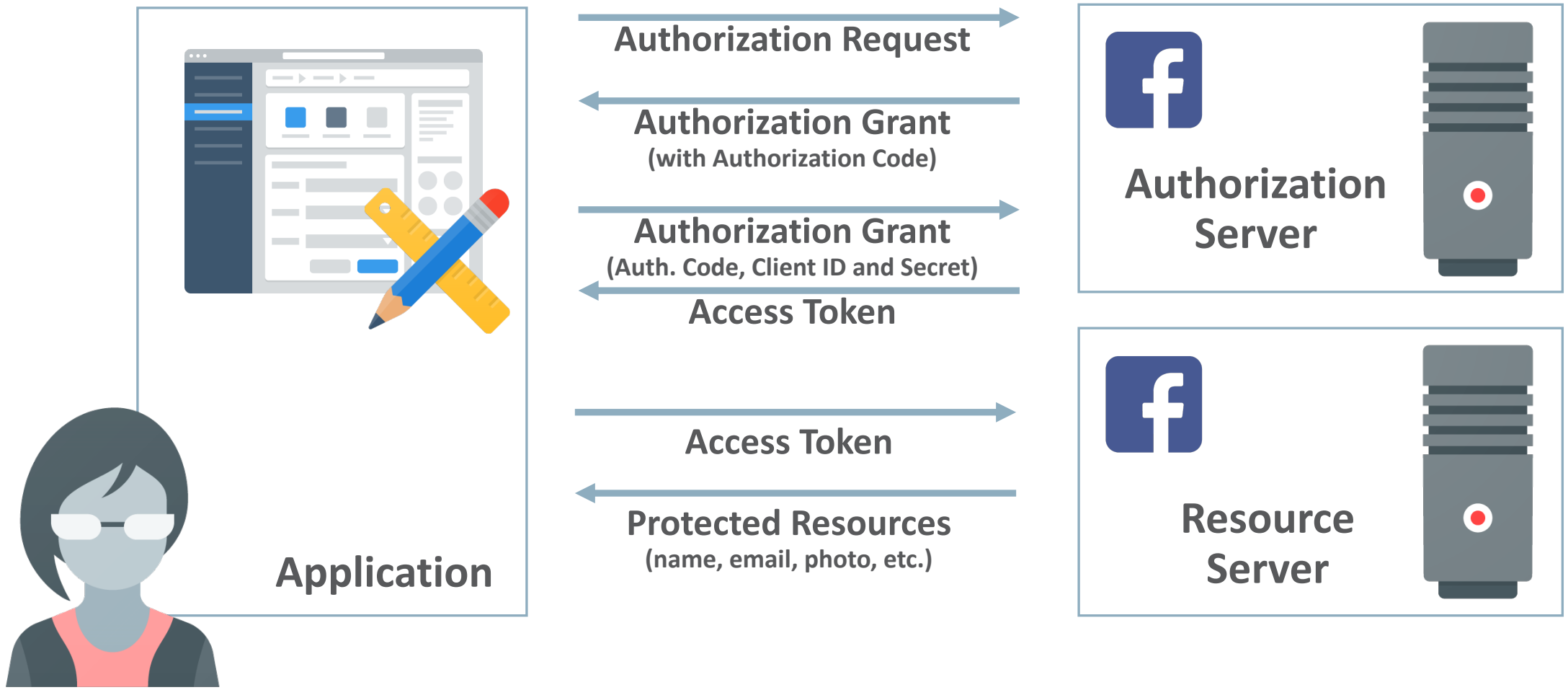
OAuth 2.0

Register Application with OAuth 2.0 provider



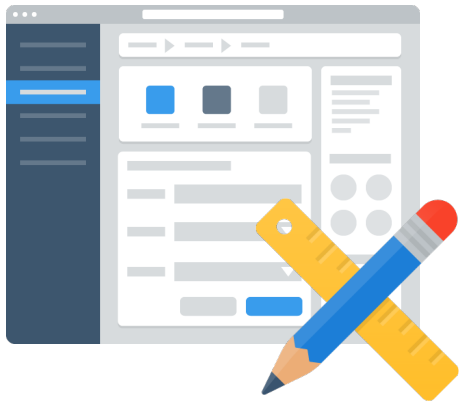
OAuth 2.0

OAuth 2.0 Flow



Social Login Authentication Flow

Register Application with OAuth 2.0 provider



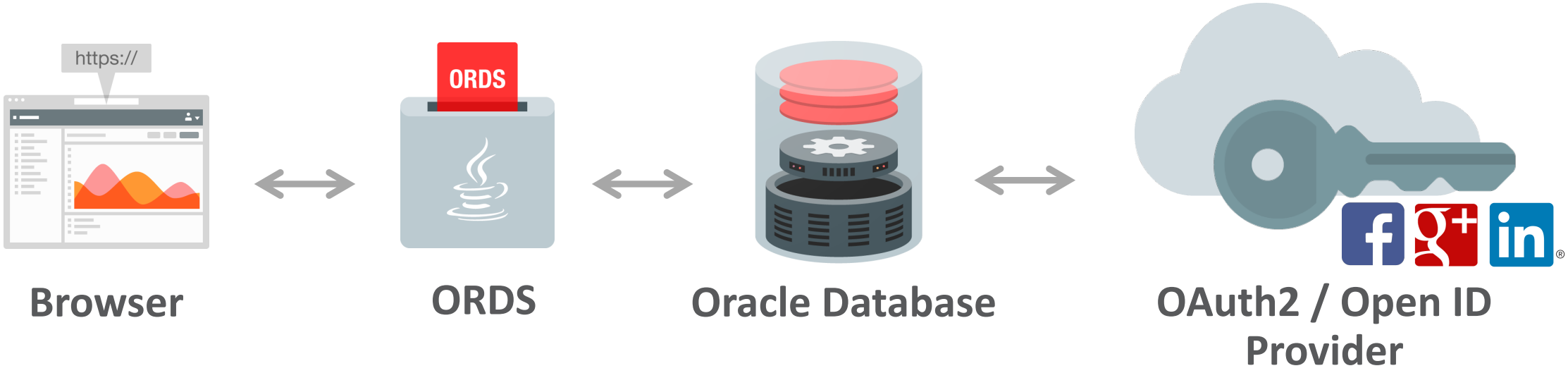
APEX

OAuth2 / Open ID Provider

- APEX app is registered as OAuth2 client with the provider (Google, Facebook, etc.).
- Provider generates Client ID (username) and client secret (password), which are stored as “Web Credentials” on APEX workspace level.

Social Login Authentication Flow

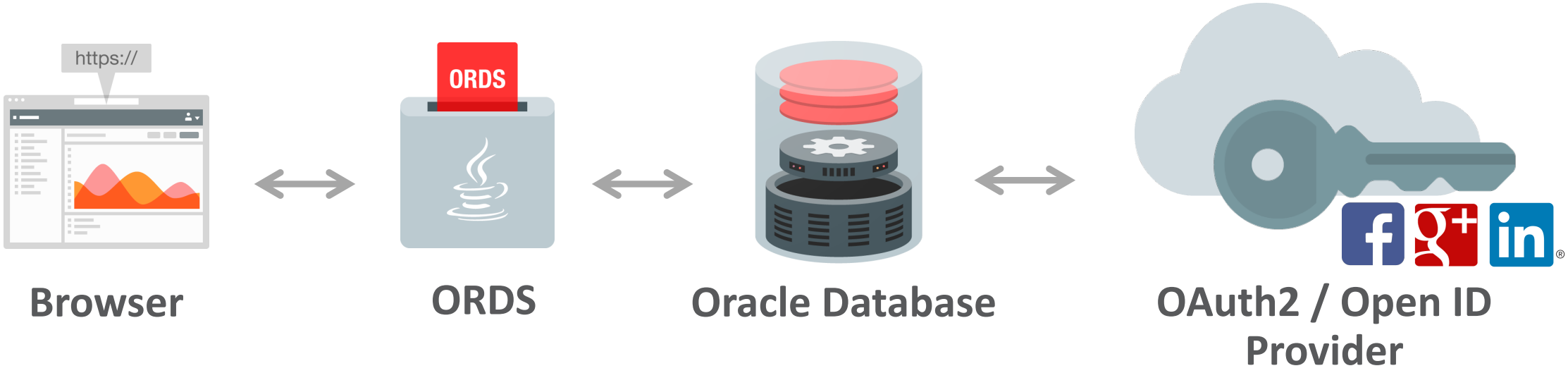
Communication Flow in APEX



- The user opens APEX app in browser:
  e.g. https://apexea.oracle.com/ords/f?p=111:1

Social Login Authentication Flow

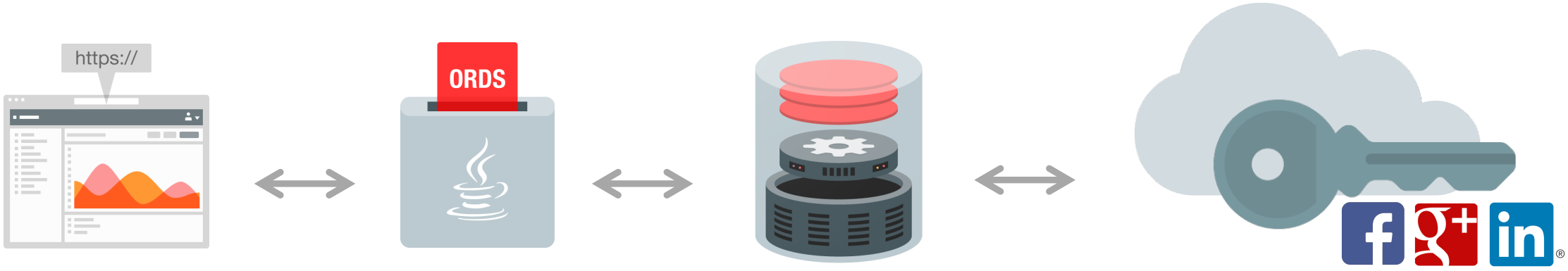
Communication Flow in APEX



- APEX creates a session for “nobody” and checks if the page requires authentication.
- If authentication is required, details of the current authentication scheme are retrieved from the metadata.

Social Login Authentication Flow

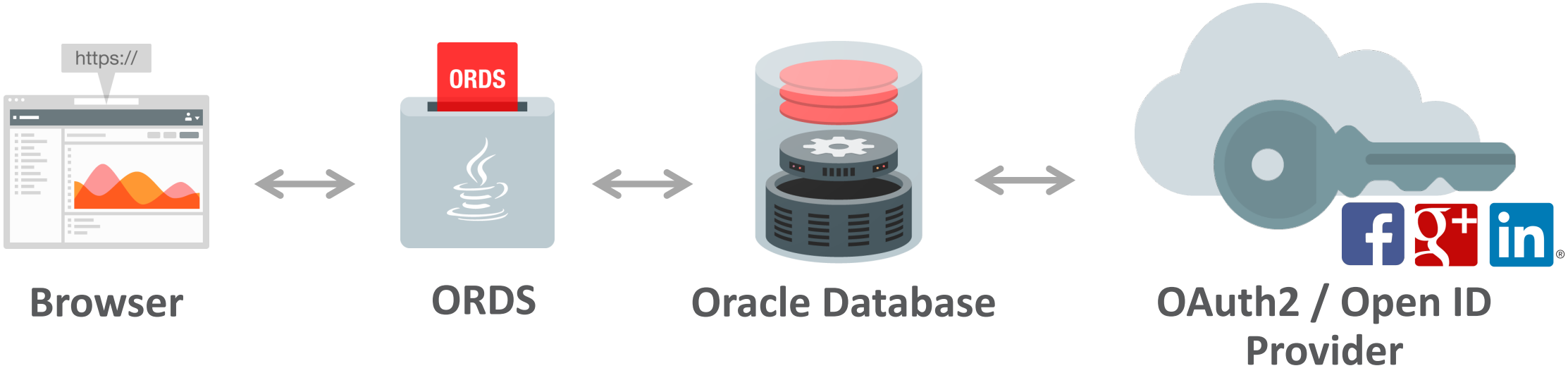
Communication Flow in APEX



- The user is redirected to the authorization endpoint URL.
  - When using generic OpenID, the discovery URL is used to retrieve the ***endpoint/token/user info URL*** with its parameters.
  - When using generic OAuth2 provider, these URLs have to be entered in the authentication scheme definition in APEX.
  - When using Facebook or Google, the URLs are pre-define in the authentication scheme.

Social Login Authentication Flow

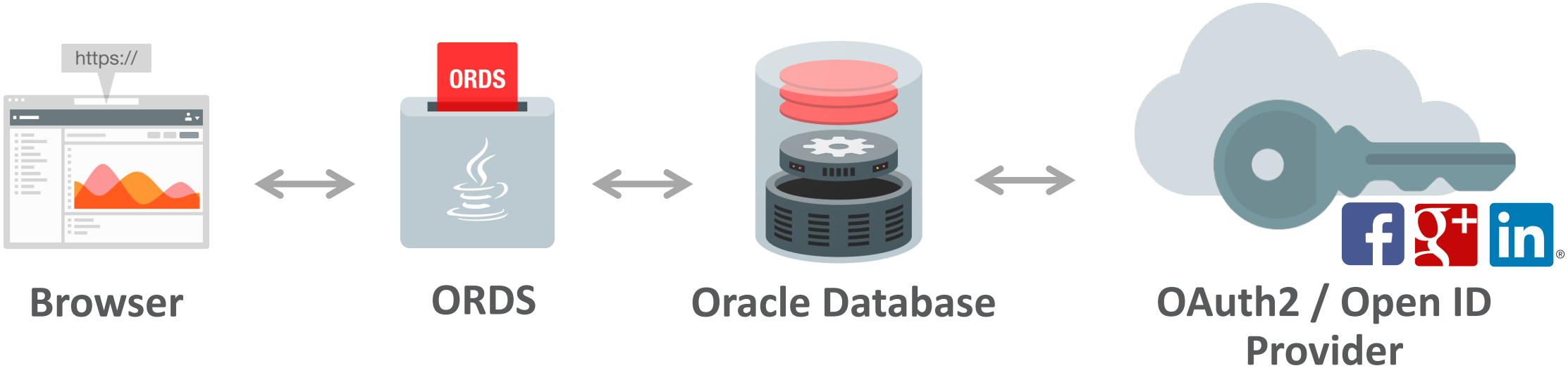
Communication Flow in APEX



- The user logs in through the providers login page, using **username** and **password**.
- The user may also need to **explicitly provide consent** to access to resources like email and photo.

Social Login Authentication Flow

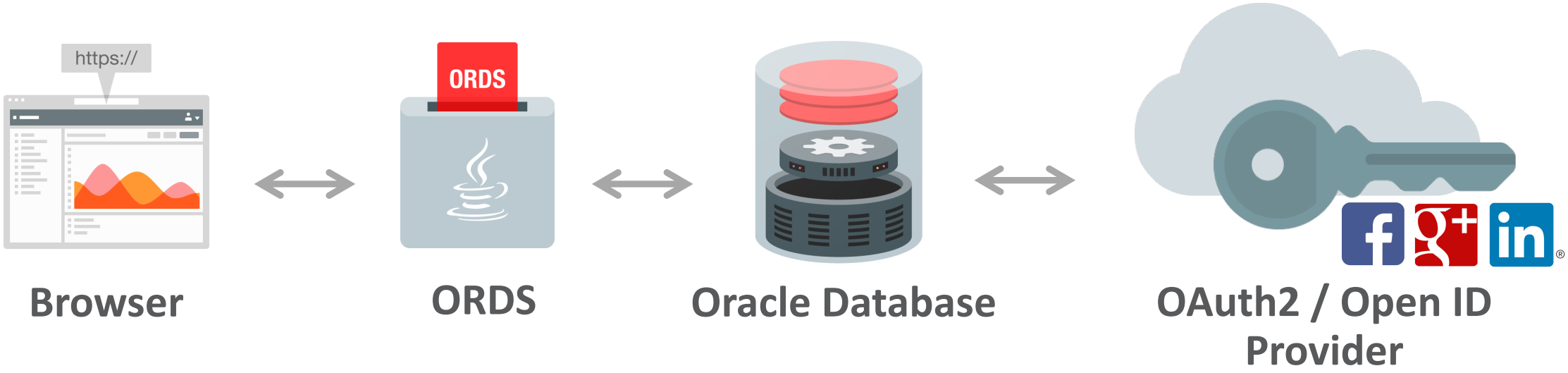
Communication Flow in APEX



- After successful authentication, the providers returns an **authorization code**.
- Browser is redirected, along with the authorization code, to the ***APEX callback URL***: https://apex.oracle.com/ords/apex_authentication.callback

Social Login Authentication Flow

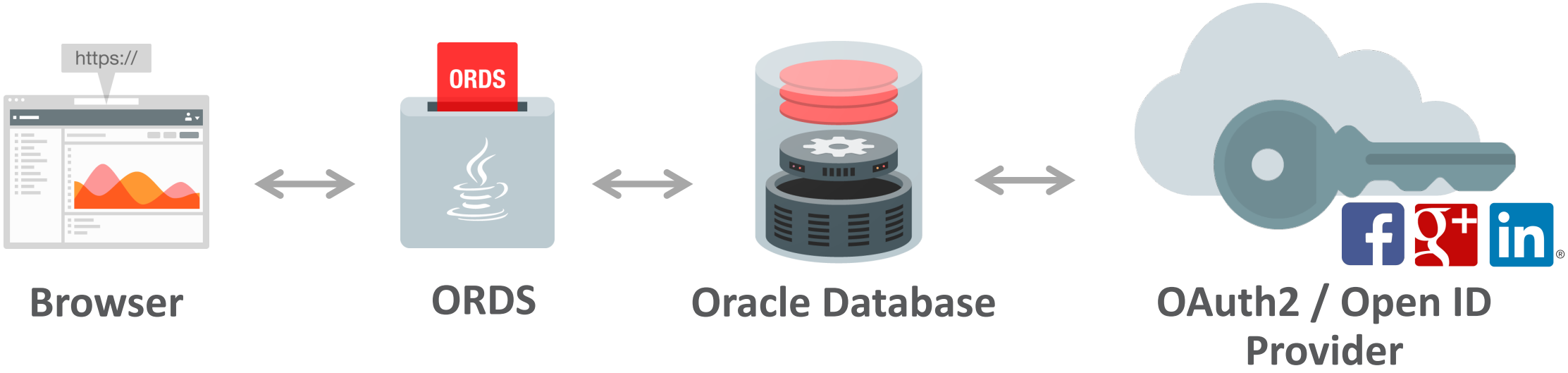
Communication Flow in APEX



- APEX requests an access token from the ***token endpoint URL,*** passing the authorization code, client ID and client secret.
- Provider returns an access token (plus refresh token and optional ID token).
- All tokens are stored in APEX.

Social Login Authentication Flow

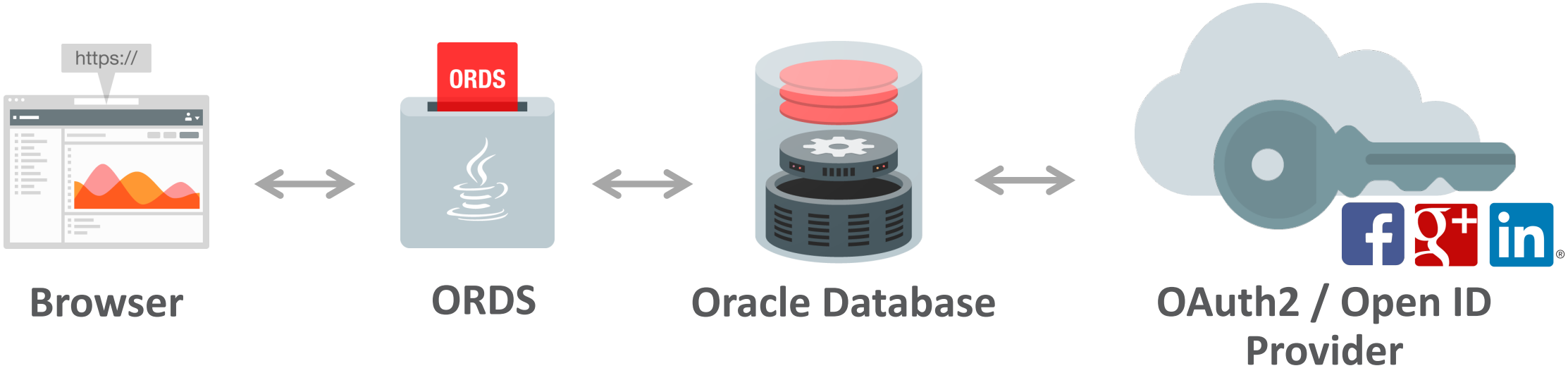
Communication Flow in APEX



- Based on the “scope” attribute defined in the authentication scheme, the user info endpoint URL is used to get additional information (e.g. photo) about the user.
- Information is returned as JSON.
- All attributes (incl. ID- & access token) are stored in APEX_JSON.G_VALUES.

Social Login Authentication Flow

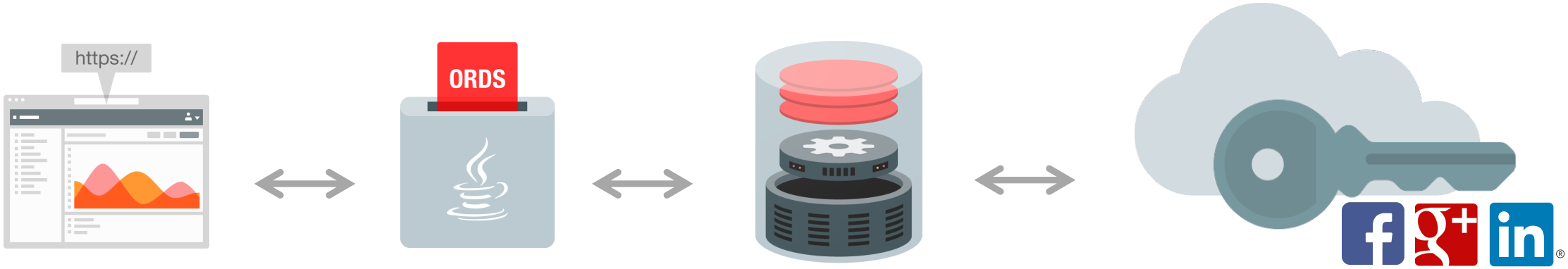
Communication Flow in APEX



- APEX retrieves the username from the tokens (access/ID).
- APEX authenticates user in the session and redirects to the originally requested page.

Social Login Authentication Flow

Communication Flow in APEX



- All subsequent browser requests for APEX pages include a session cookie, which APEX uses to identify the user.
- While the APEX session is valid, requested pages are loaded.
- The access token is not checked for expiration.
- The refresh token has no usage for the authentication process.

Demo

http://bit.ly/apex_social_login_demo

1 Authentication in Oracle APEX

2 OAuth 2.0

3 **Using Google Authentication**

4 Using Facebook Authentication

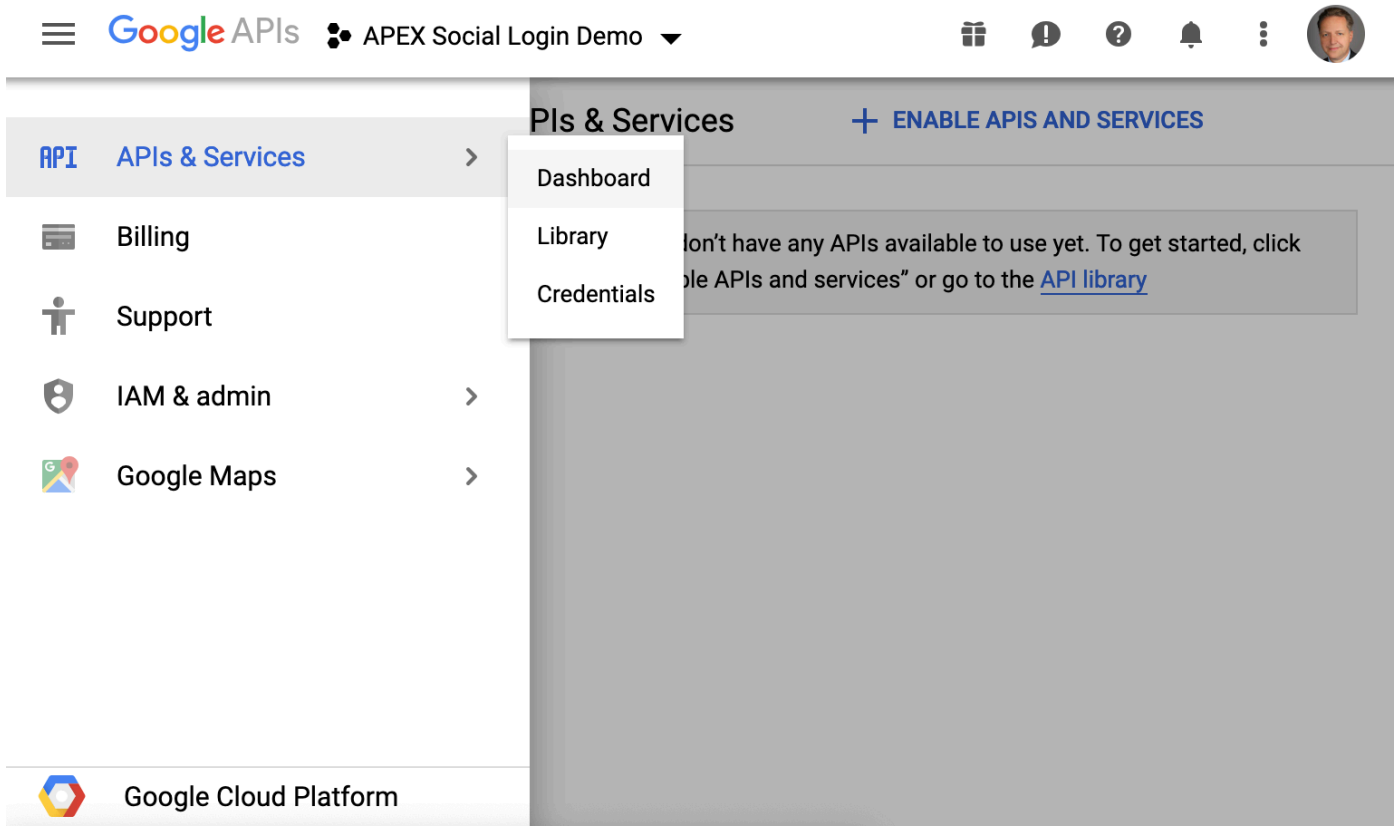
5 Using LinkedIn Authentication

6 Authorization in Oracle APEX

Using Google Authentication

Configure Google APIs "Project" to use Google Authentication

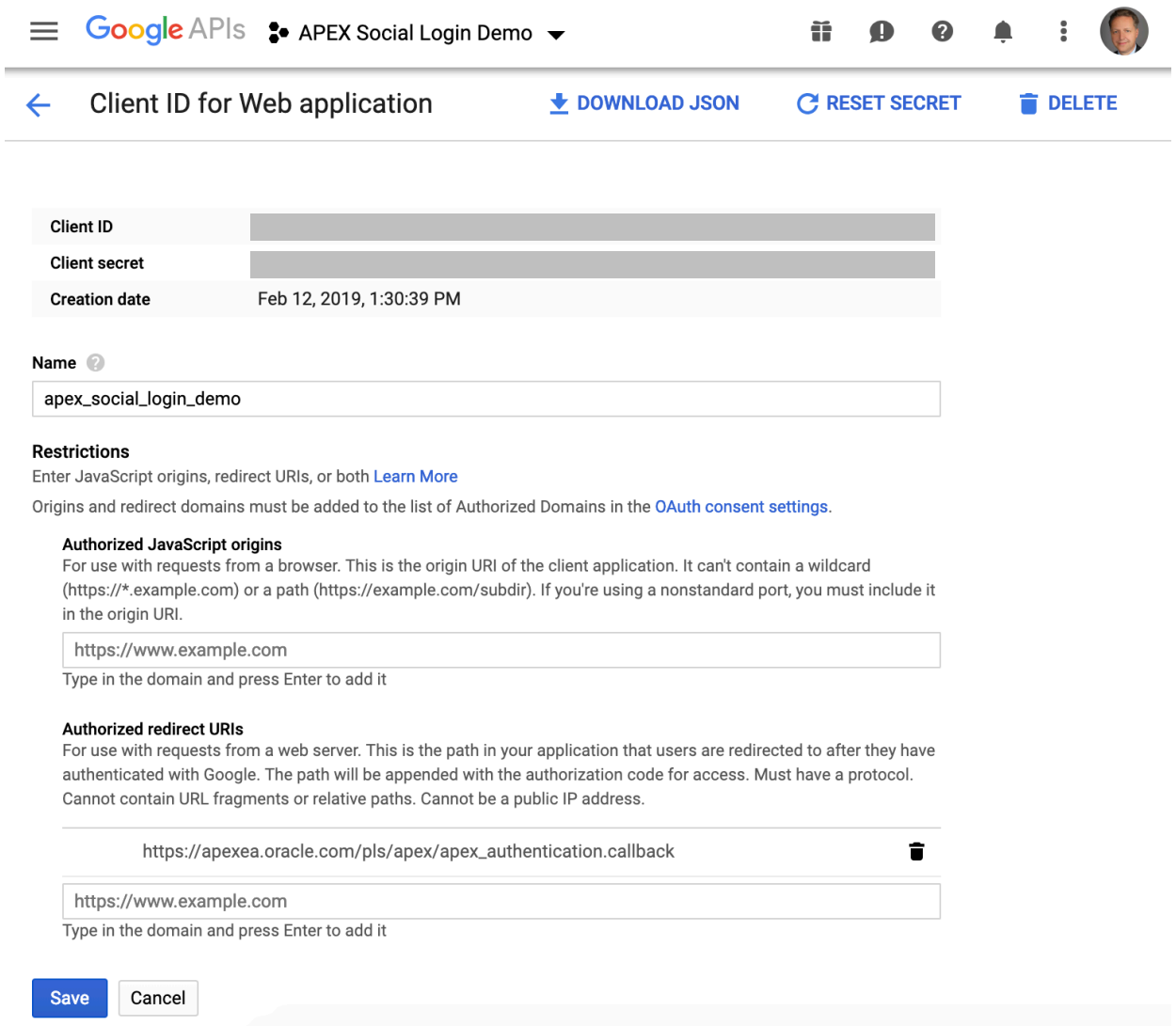
- Google APIs Developer Console: https://console.developers.google.com
- Create a "Project" from the drop-down on top of the page
- Select "Credentials" in menu on the left



Using Google Authentication

Client ID

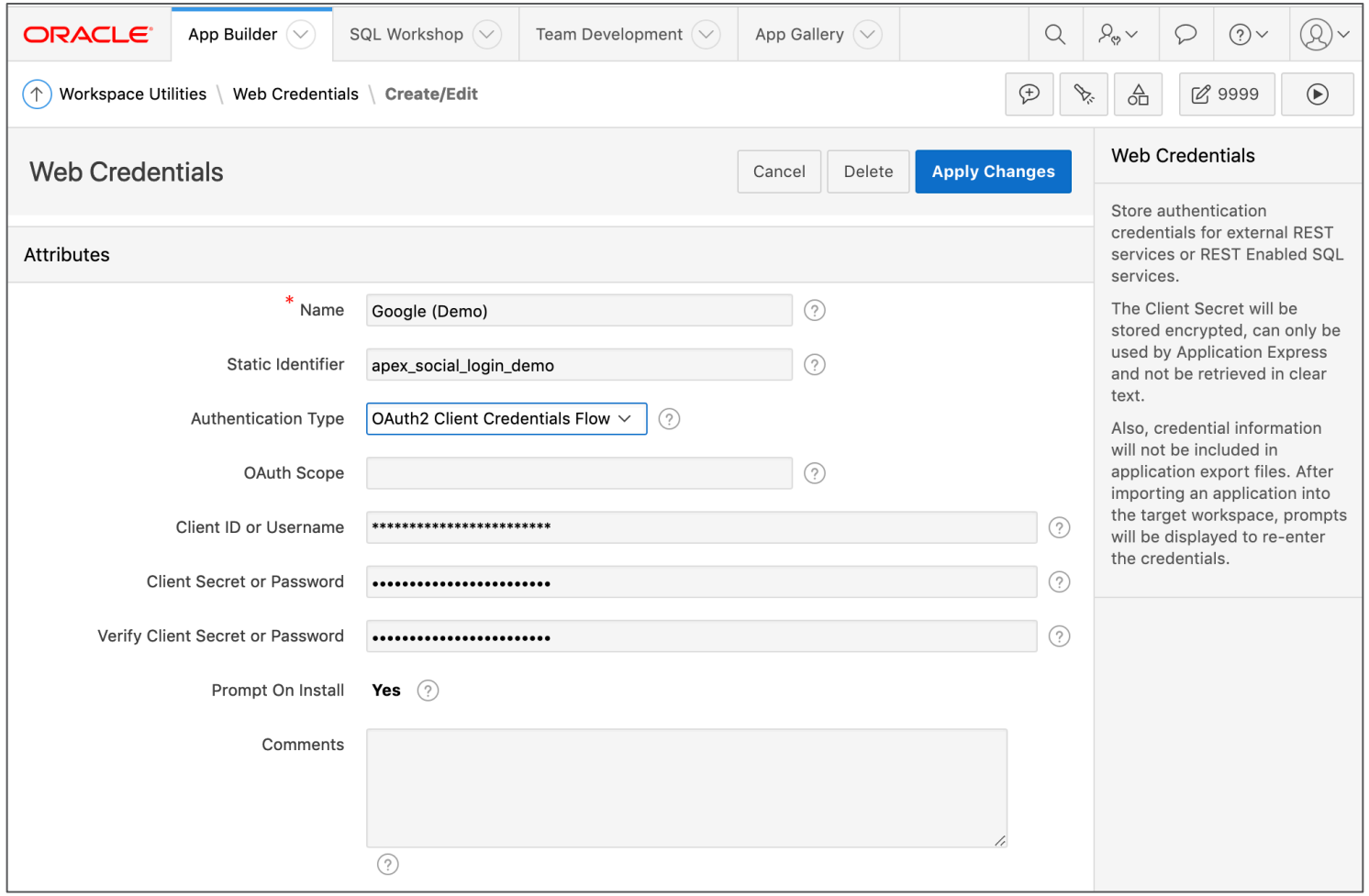
- Create OAuth 2.0 client IDs
- Requires name of OAuth Client ID and Authorized redirect URIs: *https://apexea.oracle.com/pls/apex/apex_authentication.callback*
- Provides Client ID and Client Secret, which are used when creating Web Credentials in APEX



Using Google Authentication

Create Web Credentials in APEX

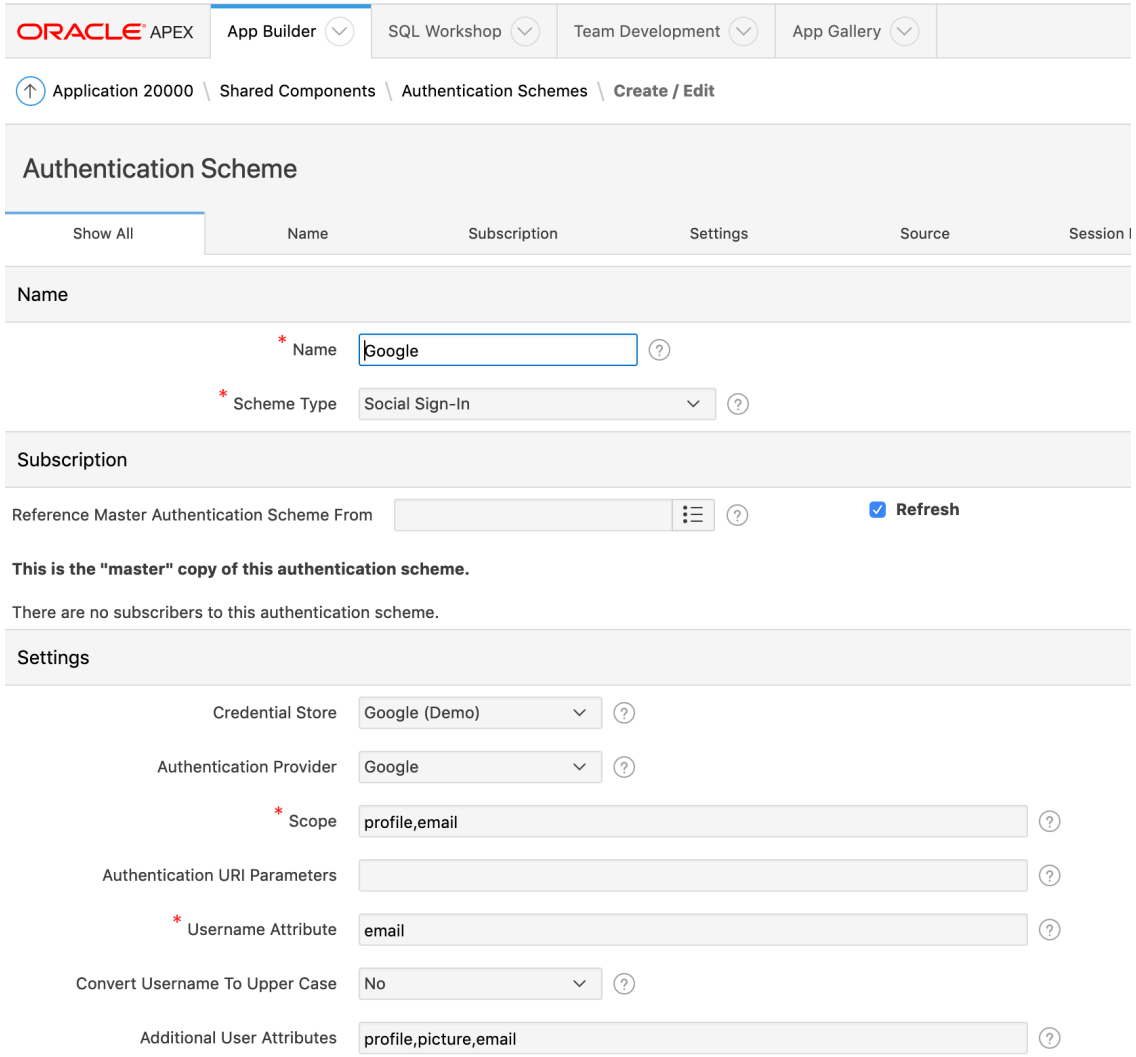
- OAuth2 Client ID and Secret are stored in APEX Web Credentials
- Web Credentials are managed on the Workspace level



Using Google Authentication

Google Authentication Scheme

- In your APEX app, create “Social Sign-In” authentication scheme
- Select your previously created Web Credentials
- Select “Google” as Authentication Provider
- Define Scope, Username Attribute and Additional User Attributes



Demo

Google Authentication

1 Authentication in Oracle APEX

2 OAuth 2.0

3 Using Google Authentication

4 **Using Facebook Authentication**

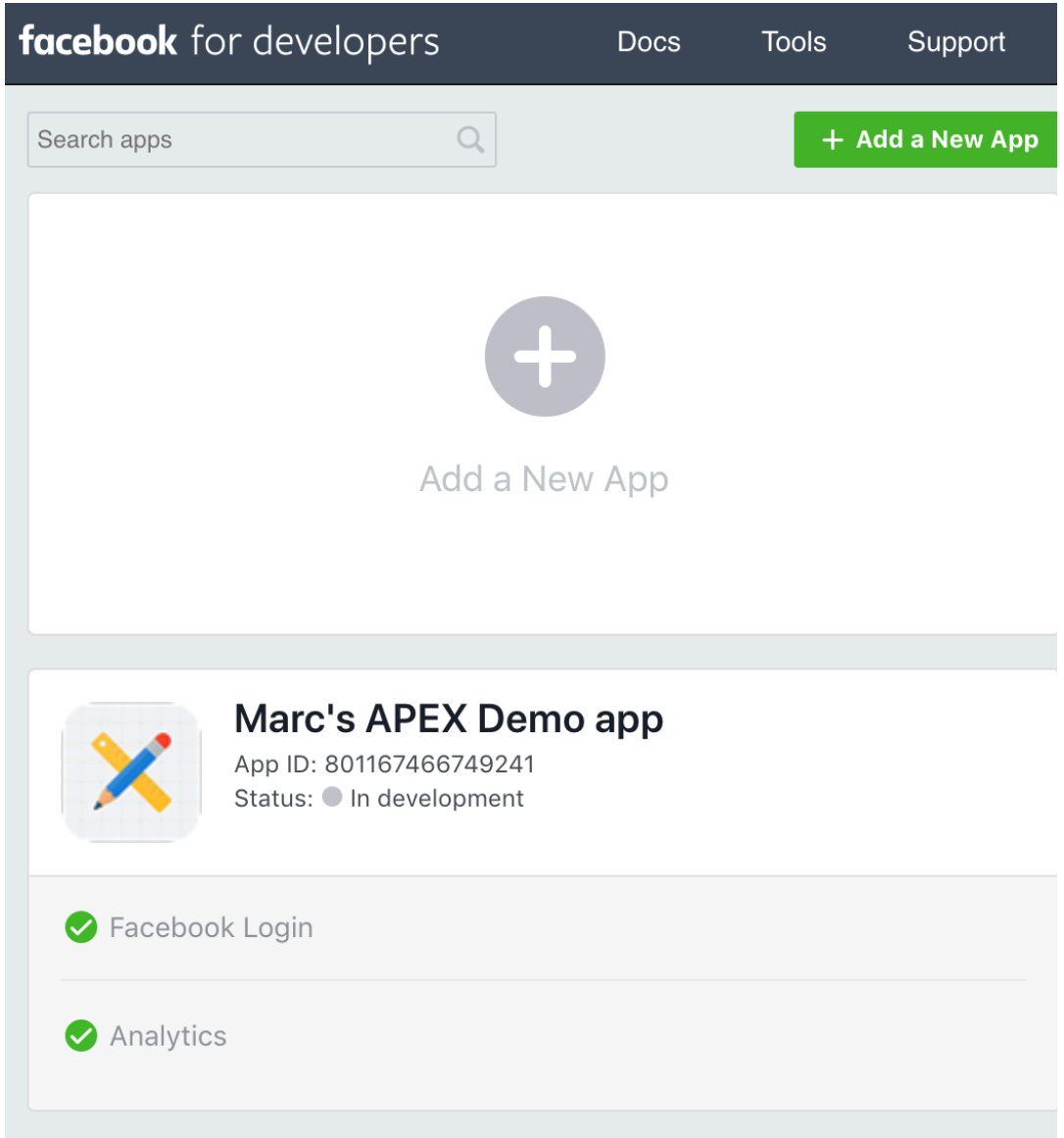
5 Using LinkedIn Authentication

6 Authorization in Oracle APEX

Using Facebook Authentication

Create App to use Facebook Authentication

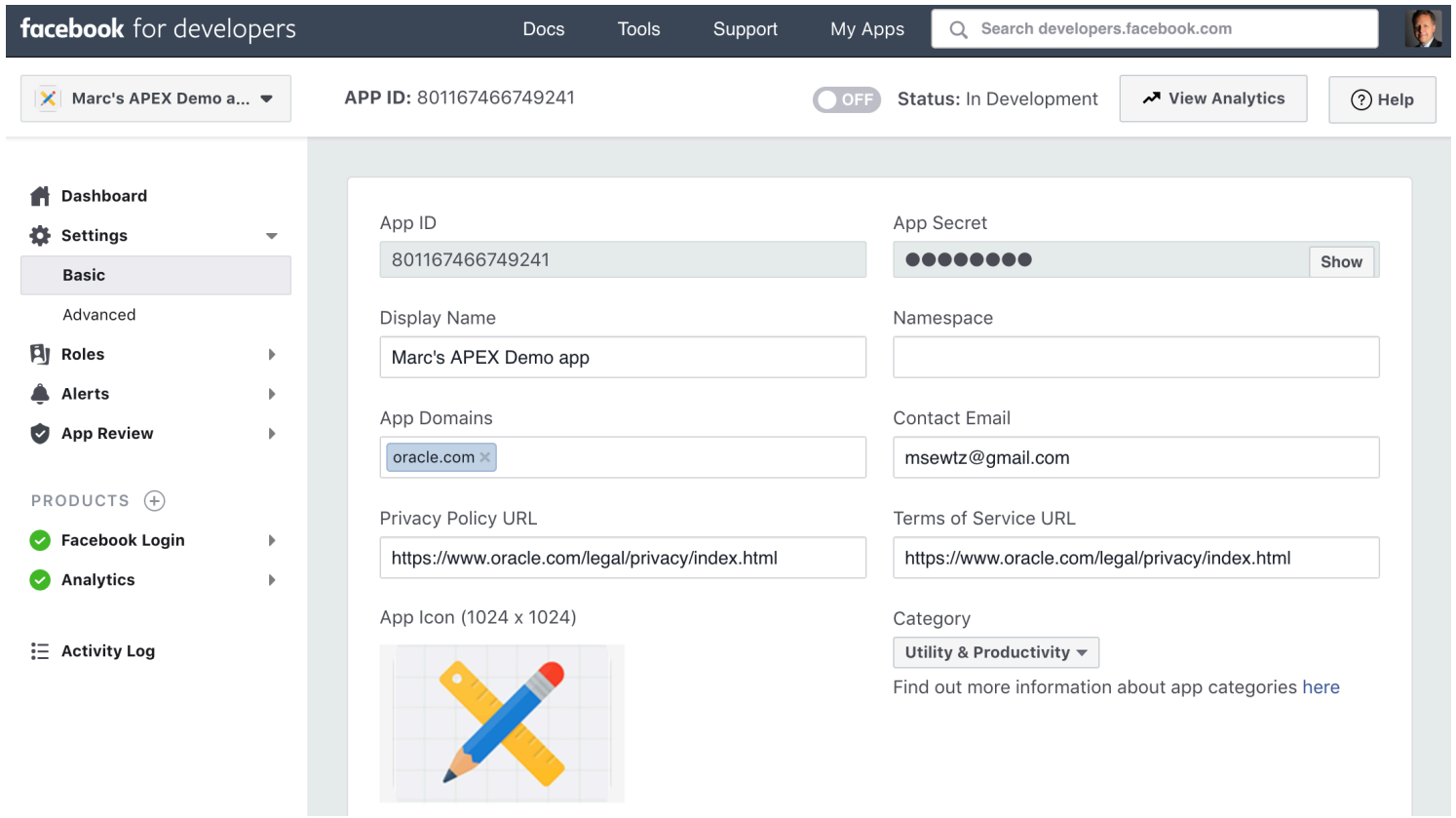
- Go to Facebook for Developers: https://developers.facebook.com/apps
- Select “Add a new App”



Using Facebook Authentication

App Definition

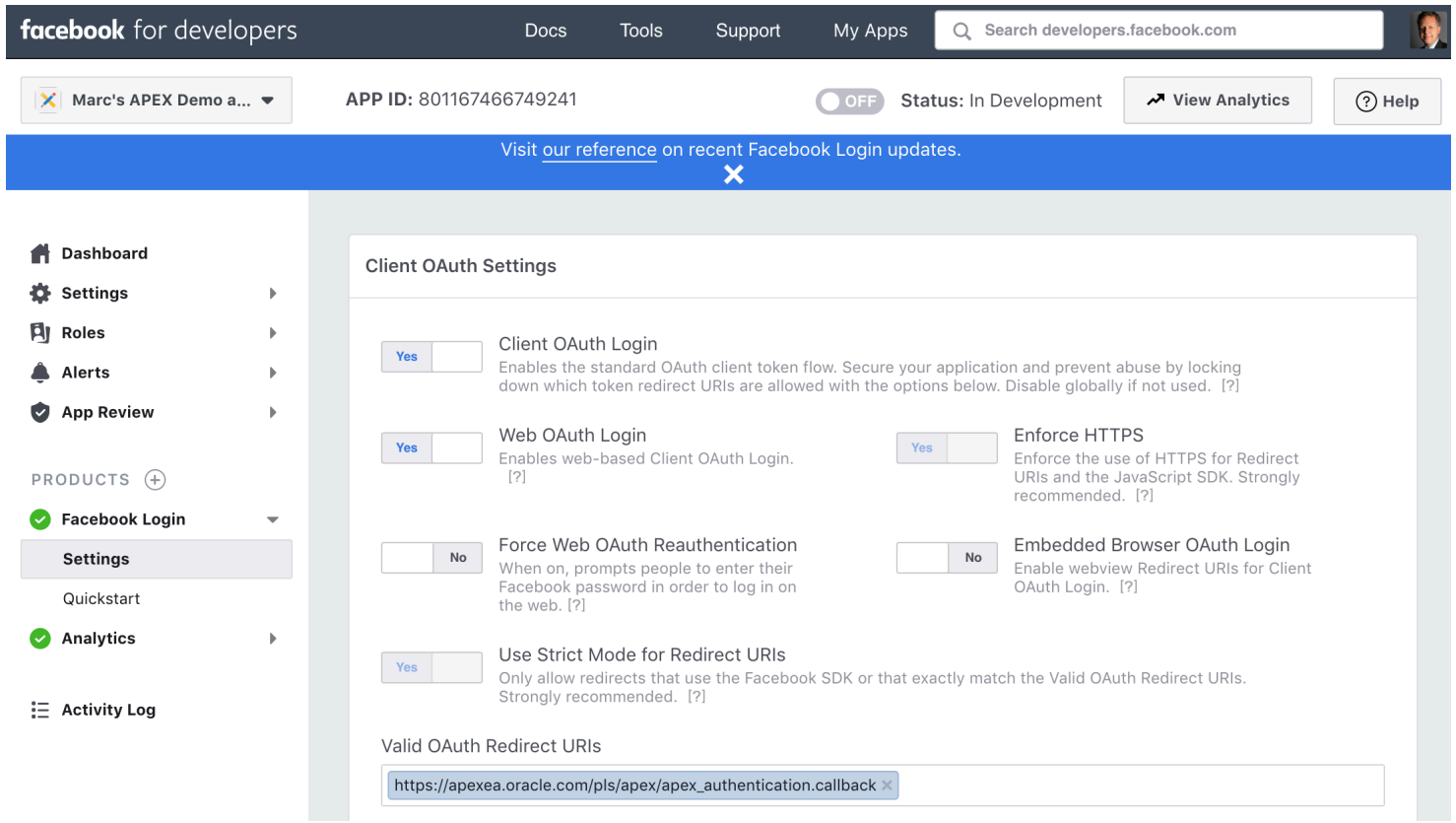
- Define App name, icon, domains, contact email, privacy and terms of service URLs
- Copy App ID and App Secret, those are used in APEX Web Credentials for Client ID and Client Secret



Using Facebook Authentication

App Definition

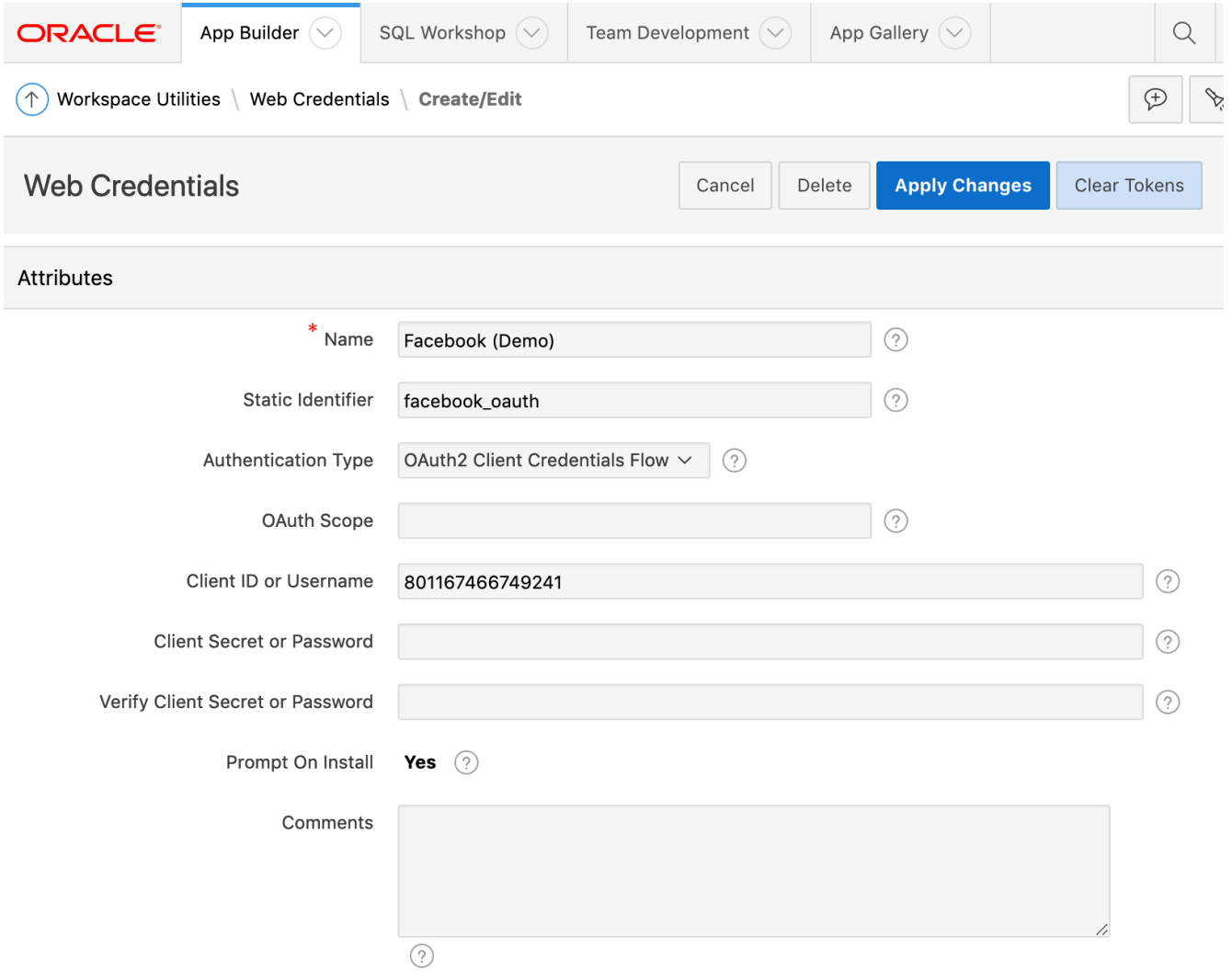
- New to also define “Facebook Login” Settings (see products)
- This is where the APEX redirect URL is stored: *https://apexea.oracle.com/pls/apex/apex_authentication.callback*



Using Facebook Authentication

Create Web Credentials in APEX

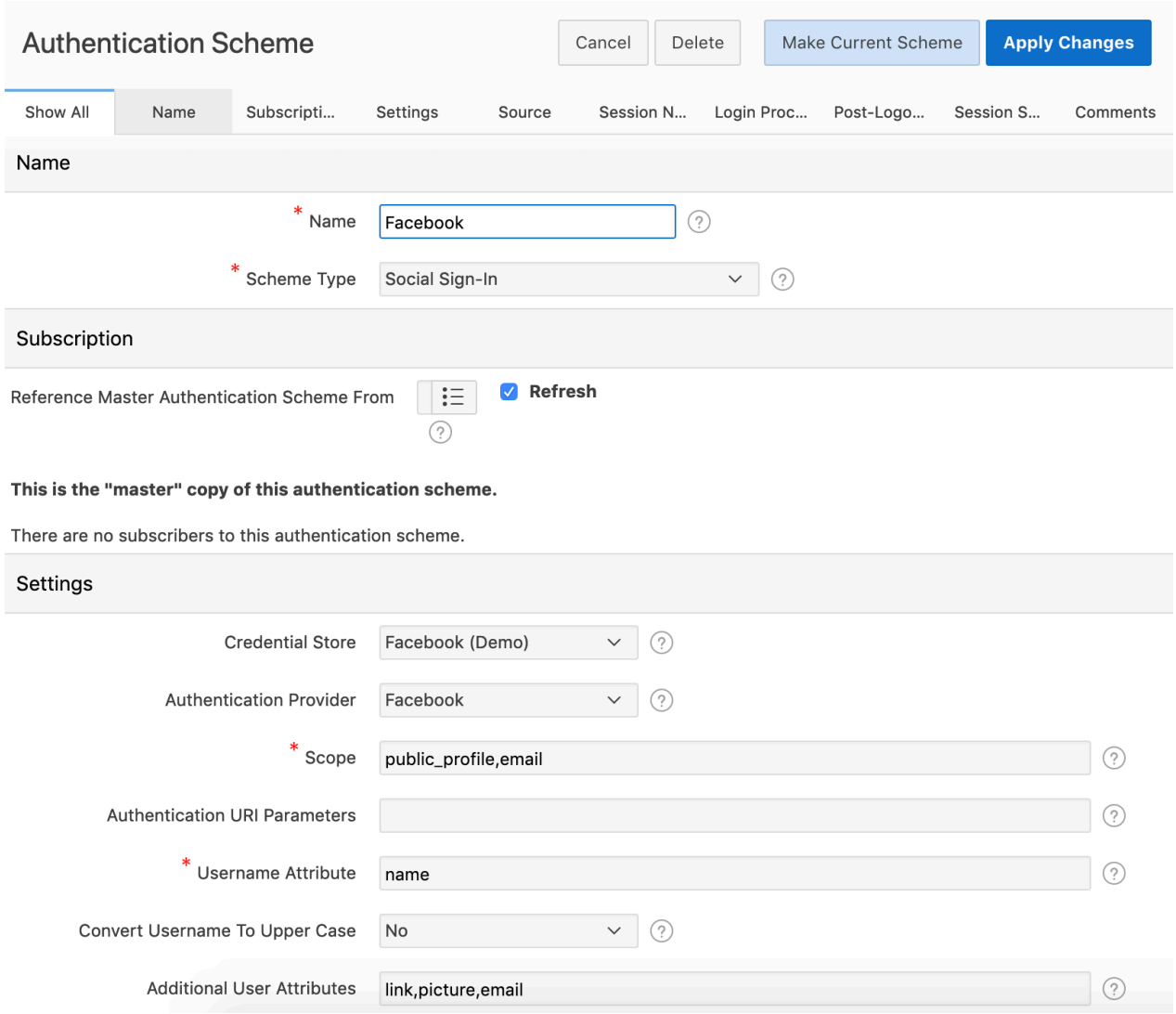
- Facebook App ID and App Secret are stored as Client ID and Client Secret in APEX Web Credentials
- Web Credentials are managed on the Workspace level



Using Facebook Authentication

Facebook Authentication Scheme

- In your APEX app, create “Social Sign-In” Auth. scheme
- Select your previously created Web Credentials
- Select “Facebook” as Authentication Provider
- Define Scope, Username Attribute and Additional User Attributes



Demo

Facebook Authentication

1 Authentication in Oracle APEX

2 OAuth 2.0

3 Using Google Authentication

4 Using Facebook Authentication

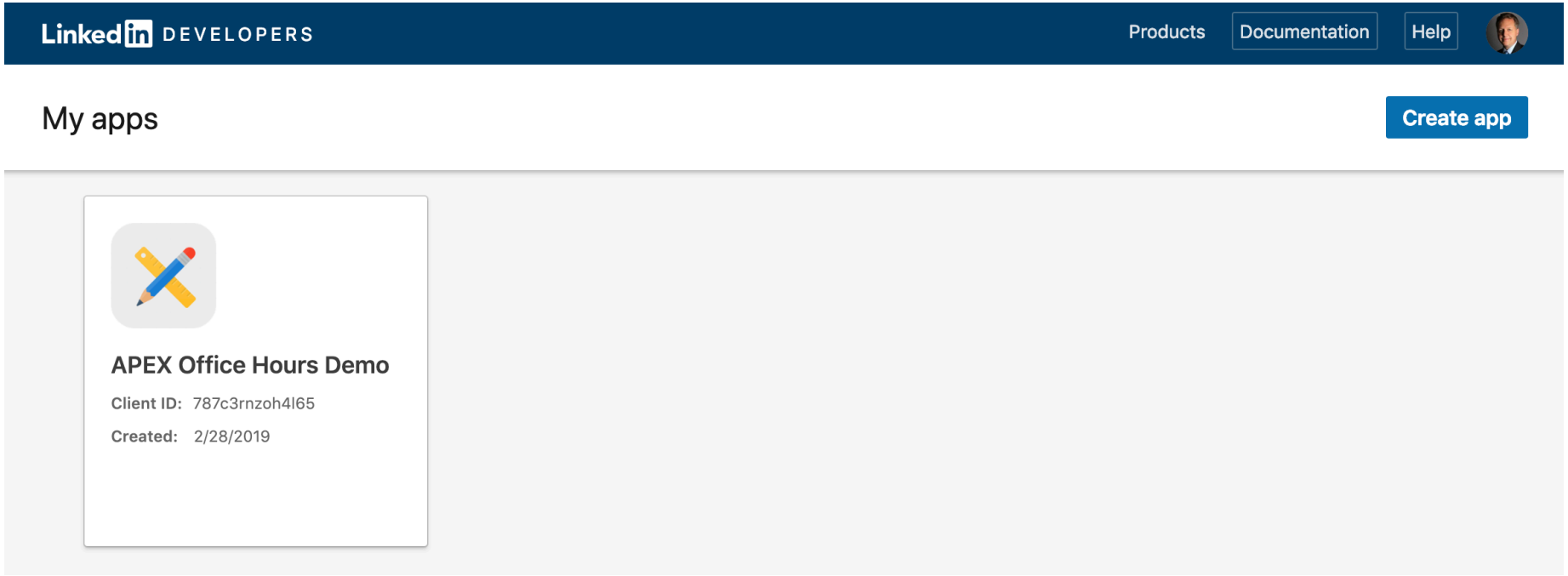
5 **Using LinkedIn Authentication**

6 Authorization in Oracle APEX

Using LinkedIn Authentication

Create App to use Facebook Authentication

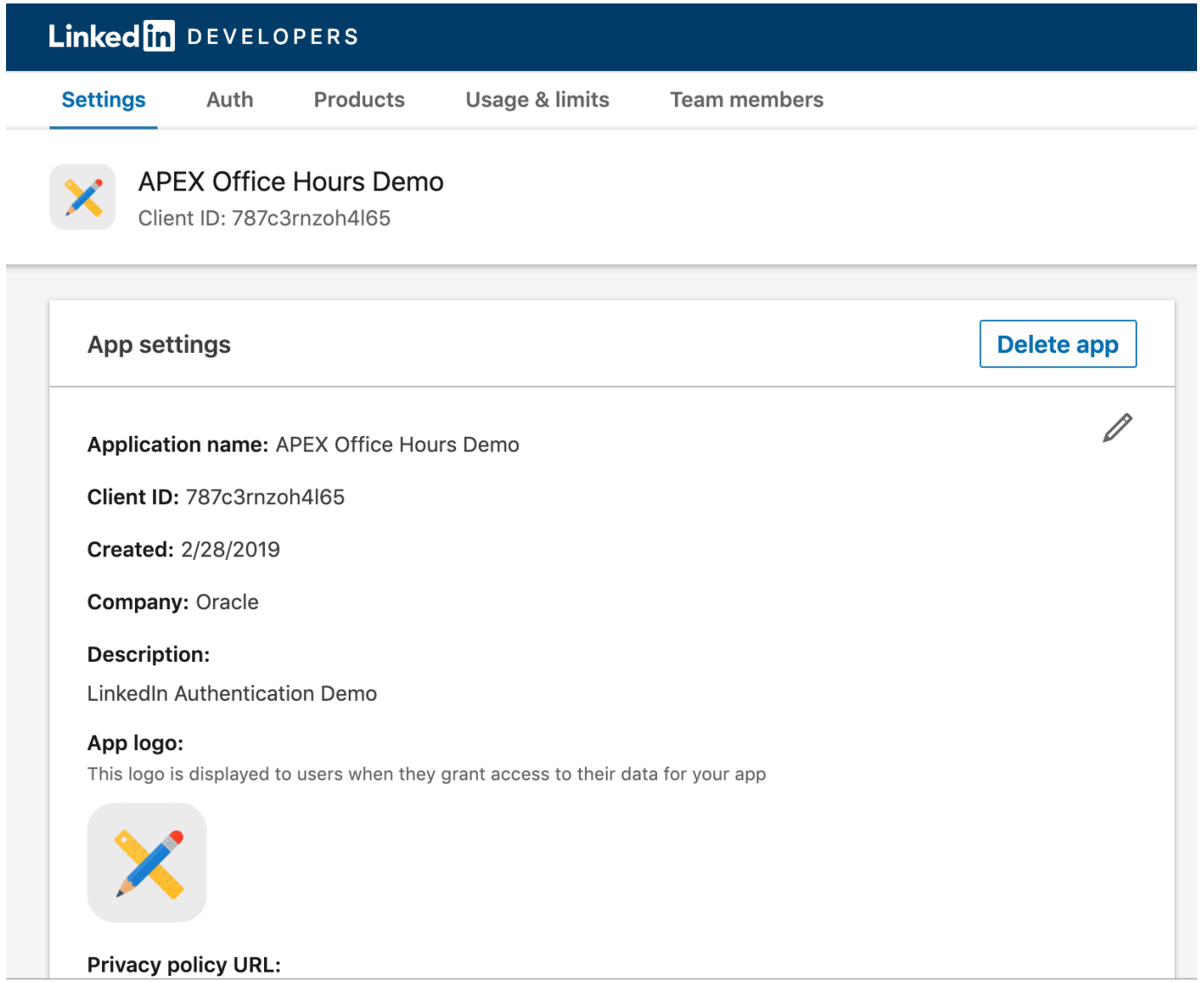
- Go to LinkedIn for Developers: **https://www.linkedin.com/developers/apps**
- Select “Create App”



Using LinkedIn Authentication

App Settings

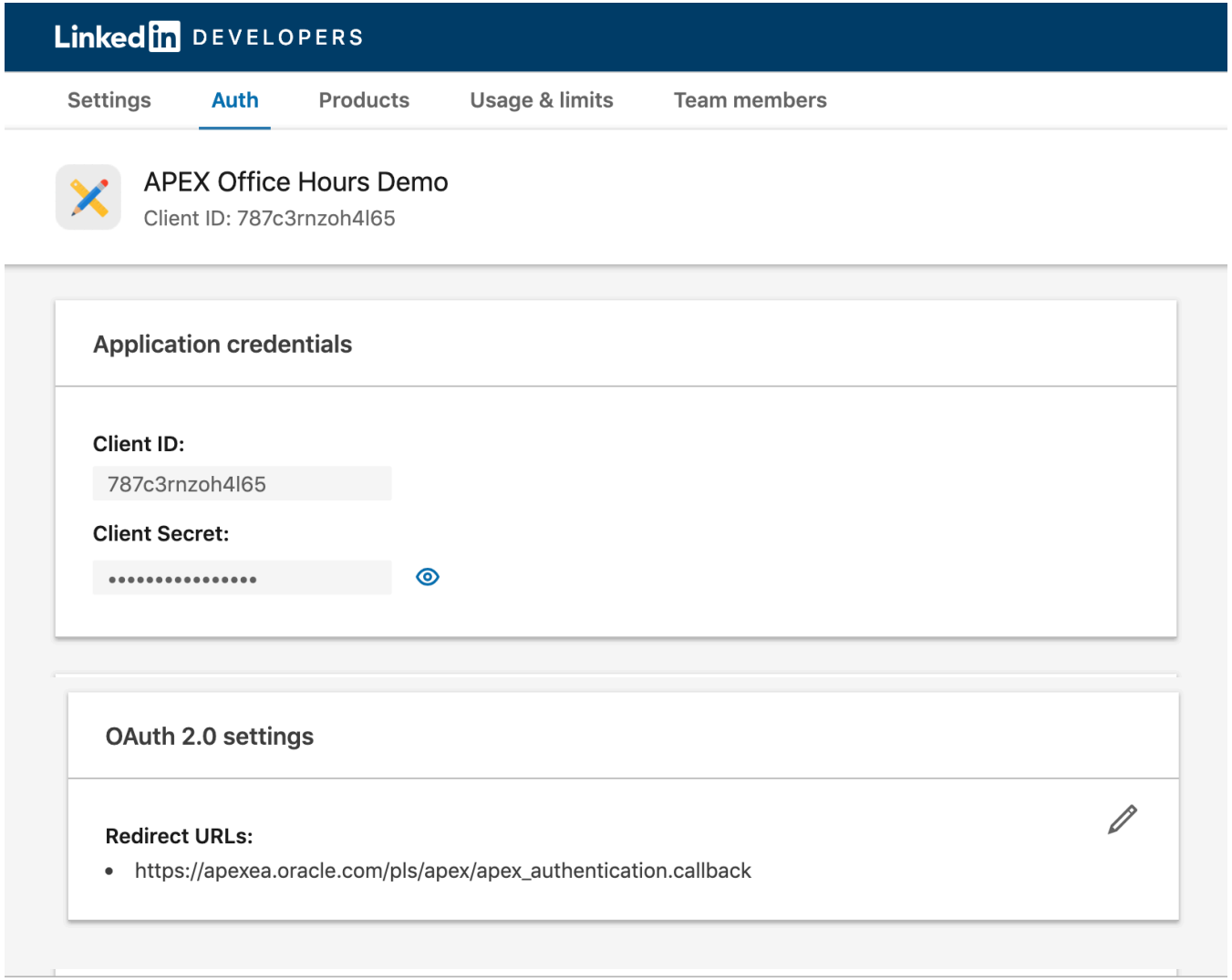
- Define Application Name, Logo, Description, Business Email, Privacy Policy and Terms



Using LinkedIn Authentication

Auth

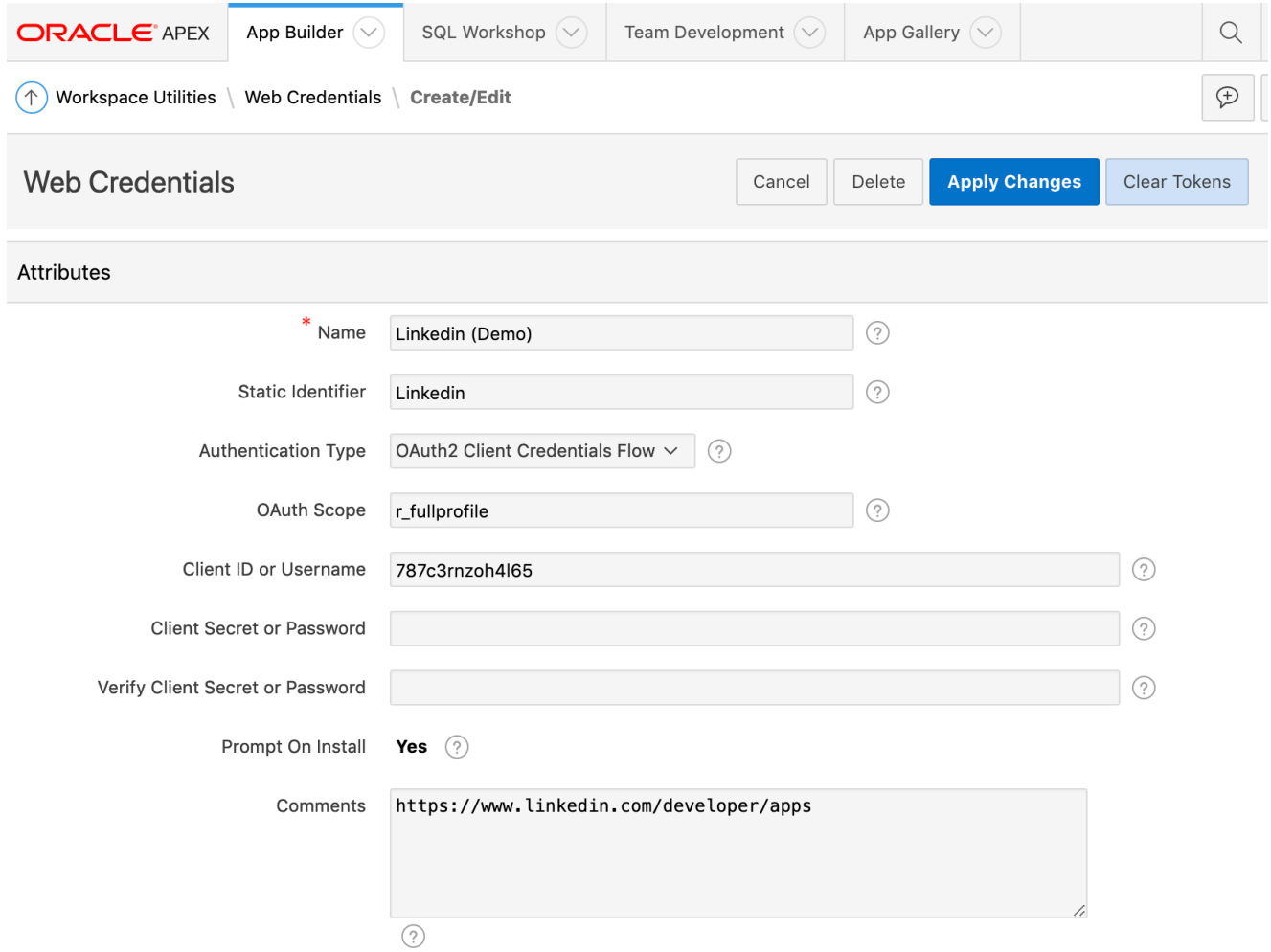
- Copy Client ID and Client Secret to be used for Web Credentials in APEX
- Enter Redirect URL at bottom of ”Auth” settings: *https://apexea.oracle.com/pls/apex/apex_authentication.callback*



Using LinkedIn Authentication

Create Web Credentials in APEX

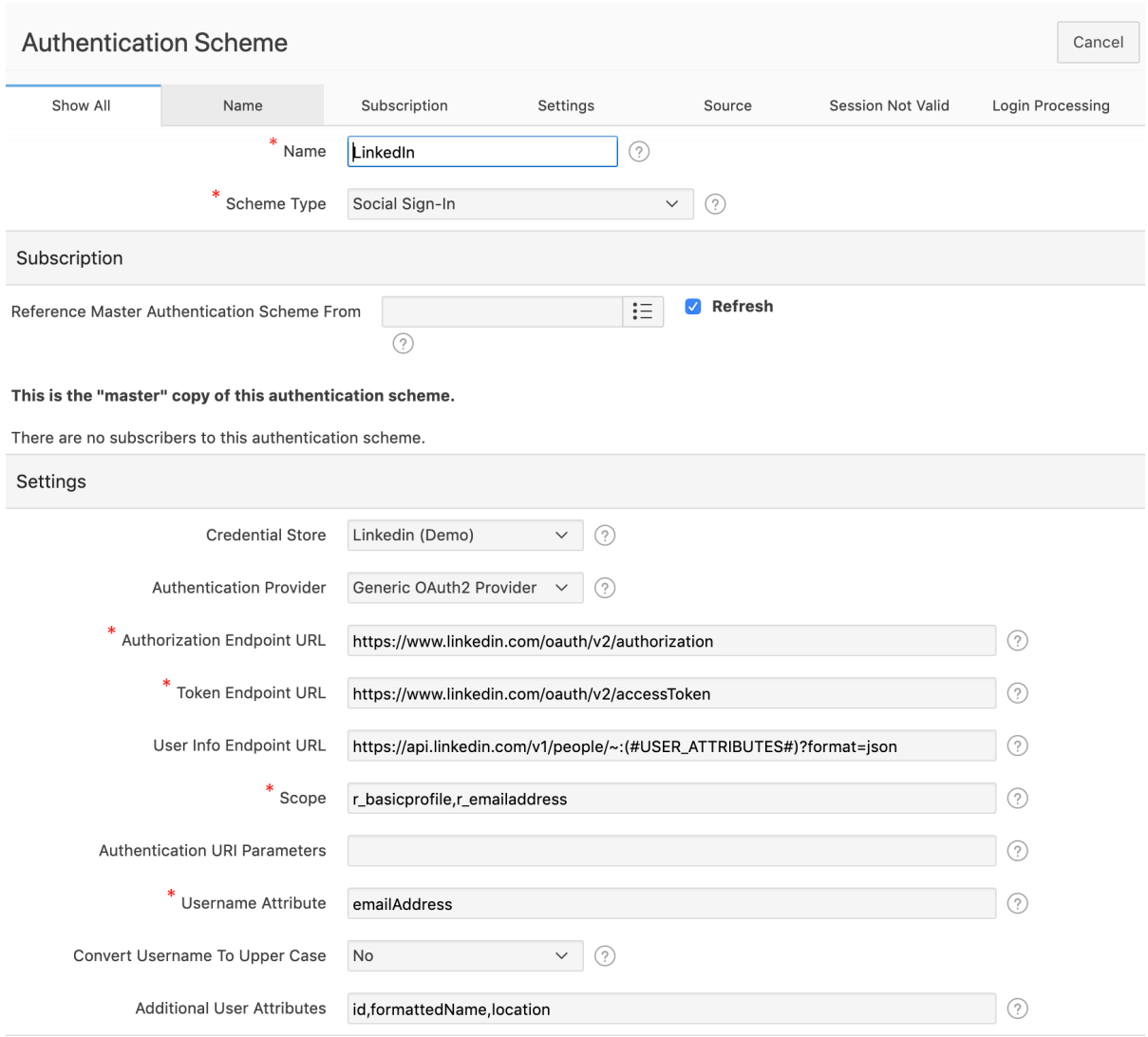
- LinkedIn Client ID and Client Secret are stored in APEX Web Credentials
- Web Credentials are managed on the Workspace level



Using LinkedIn Authentication

Facebook Authentication Scheme

- In your APEX app, create “Social Sign-In” Auth. scheme
- Select your previously created Web Credentials
- Select “Facebook” as Authentication Provider
- Define Scope, Username Attribute and Additional User Attributes



Demo

LinkedIn Authentication

1 Authentication in Oracle APEX

2 OAuth 2.0

3 Using Google Authentication

4 Using Facebook Authentication

5 Using LinkedIn Authentication

6 **Authorization in Oracle APEX**

APEX Authorization

Providing Security Through Authorization

- Extend the security of your application by creating an authorization scheme.
- While conditions control the rendering and processing of specific page controls or components, **authorization schemes control user access to specific controls or components**.
- You can specify an authorization scheme for an entire application, page, or specific control such as a region, item, or button.
- For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bars a user sees.
- When you define an authorization scheme, you give it a unique name. Once defined, you can attach it to any component or control in your application.

Access Control

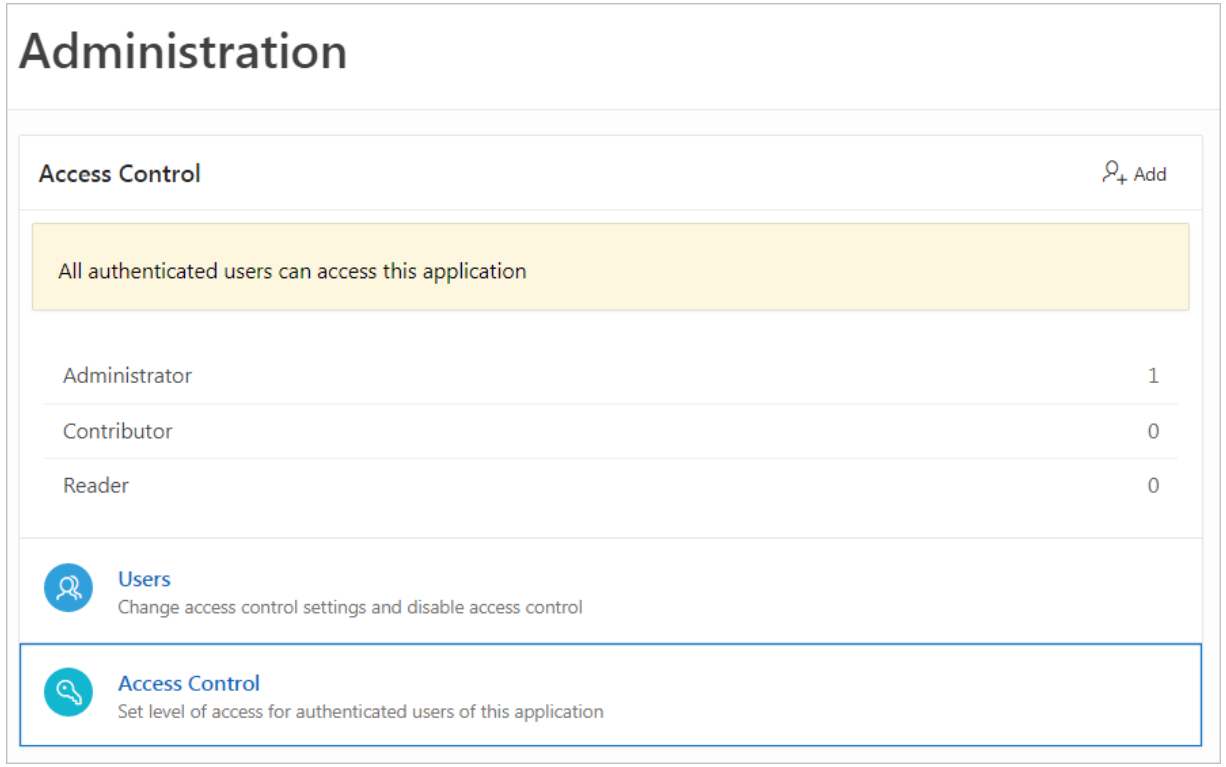
Controlling Access to Applications, Pages, and Page Components

- Adding the **Access Control feature** to an application, creates multiple pages and the following components:
  - Adds an Access Control region to the Administration page you specify.
  - Creates the access roles*: Administrator, Contributor,* and *Reader*.
  - Creates the authorization schemes: *Administration Rights, Contribution Rights,* and *Reader Rights*.
  - Creates the build option: *Feature: Access Control*.
  - Creates the Application Setting: *ACCESS_CONTROL_SCOPE*.

Access Control

Controlling Access to Applications, Pages, and Page Components

- Configure Access Control by running the app and accessing the Access Control region on the Administration page.
- Click **Users** to add new users, change a user's role, or disable access control by locking an account.
- Click **Access Control** to specify the behavior when authenticated users access the application.



Demo

Authorization

Q&A

Integrated Cloud

Applications & Platform Services