* Basic requirement in order to communicate between applications is identity and permissions (Authentication and Authorization)
* OAuth is the **Authentication and Authorization standard** which allows an application to gain access to user’s data within other application without knowing the user id and password for second application.
* If application A wants information in application B, instead of giving user id and password to application A, OAuth will redirect you to application B and prompts for authentication and gives back a security token to application A. By this these two applications will communicate with each other. This process of logging in and getting authorized is covered in this OAuth standard.
* This process of one application using other application to login or Authenticate is called **Federated Authentication.**
* **Delegated Authorization**: One application delegate the process of authorization to other application. User grants an action to an application to performs actions on the user’s behalf and the application only performs those actions which user authorizes it to do.
* Without OAuth :

A picture containing electronics

Description automatically generated

* Gives web applications to get secure way of getting access tokens.

**OAuth Work Flow:**

**A picture containing electronics

Description automatically generated** A screenshot of a cell phone

Description automatically generated

**A screenshot of a cell phone

Description automatically generated**

* If you want to comment here it will redirect you to google api page with above link. It has client id which says google API you are the requested client, State is the unique id which is the current session state of the user and the redirected URL.
* The user logs in and google authenticates and redirects with state of the user and Unique Authorization code generated by google for exchange.
* Now, java world send this code in http request to google, in exchange it will get a security token from google which helps to communicates between applications.
* By using this token application can make any number of calls to API to get the User data.

A screenshot of a cell phone

Description automatically generated

**Roles**:

1. **Resource Server**: server which hosts the user resource, which is protected by OAuth.
2. **Resource owner**: end user who wants to communicate.
3. **Client**: application which wants to access the resources of end user on protected resource server.
4. **Authorization Server**: which generated the authorization code, typically resource server and Authorization server or on the same machine.

A screenshot of a cell phone

Description automatically generated

Grant types:

* **Authorization code grant** type is used between web servers.

A screenshot of a cell phone

Description automatically generated

OAuth Implementation using Spring Boot:

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated