Before OAuth existed whenever users wanted to share data they had to send their user credentials to third party apps. For example, Let’s say a fitness app which a user uses stores user information and meal app wants to get user data from this fitness app because it can suggest the amount or type of meals based on user’s workout. Before the advent OAuth user has no choice but to share their username and password for the fitness app with the meal planning app, which would introduce a significant security risk. For instance, the meal planning app might store the user’s credentials in an insecure way, making them vulnerable in the event of a breach.

Enter OAuth….

**OAuth**:

* Resource Owner: It means a user that is ‘*granting’* third party of their data.
* Client: This is the third-party application that is requesting access to the resource owner’s data.
* Authorization Server: This acts as an intermediary between the client, the resource owner, and resource server by issuing *tokens* to the client after the resource owner successfully authorizes the request.
* Resource Server: This is the server that hosts the protected resources and it gives access to protected resources by accepting tokens.

A diagram of a flowchart

Description automatically generated

Let’s say that a user wants to login to TOI paper for posting some comments on articles or viewing their preferred news etc.,

A screenshot of a computer

Description automatically generated

When user clicks on ‘google’ signin , user enters username & password

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Then google will ask the user for consent to share their data like age, gender etc., to the client (TOI). When user agrees for sharing the data a token will be sent to client. This client can use this token to for accessing resources on the resource server.

More at : [What Is OAuth 2.0? | Postman Blog](https://blog.postman.com/what-is-oauth-2-0/)