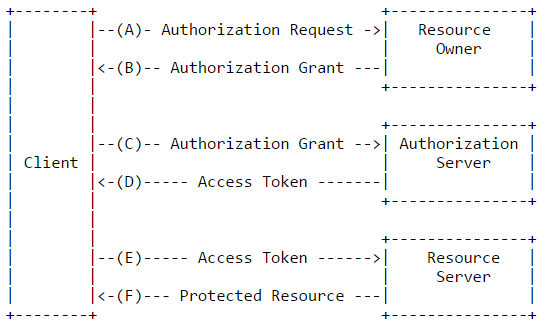
Oauth2 details based on <http://tools.ietf.org/html/rfc6749>

Abstract protocol flow (section 1.2):



Grant types (section 1.3):

* Authorization Code (full stack flow; login page jumps for the 1st time from authentication service – redirected by client, token obtained after name/password validation for further usage; login via Facebook, for example)
* Implicit (in-browser javascript apps; no client authentication)
* Resource Owner Password Credentials (similar to authorization code but client uses owner’s credentials to obtain the token, which used for further requests; access token returned directly instead of authorization code; public clients only)
* Client Credentials (client is a resource owner or accessing protected resources based on an authorization previously arranged with the authorization server; for trusted clients only)

Clients (section 2, 3.2.1):

* Before initiating the protocol, the client registers with the authorization server. This process does not require a direct interaction between the client and the authorization server
* Client can be confidential (capable of maintaining the confidentiality of their credentials – by restricting access to the client credentials or using other means) and public. Confidential clients or other clients issued client credentials MUST authenticate with the authorization server as described in Section 2.3 when making requests to the token endpoint.

Good links:

Spec: <http://tools.ietf.org/html/rfc6749>

Grants description: <http://alexbilbie.com/2013/02/a-guide-to-oauth-2-grants/>

<https://aaronparecki.com/articles/2012/07/29/1/oauth2-simplified>

<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>

<http://www.quora.com/How-does-OAuth-2-0-work> - useful links in comments

<http://www.beingjavaguys.com/2014/10/spring-security-oauth2-integration.html>

<https://github.com/raonirenosto/silverauth>