AUTHORIZATION (HTTPS://BLOG.OAUTH.IO/CATEGORY/AUTHORIZATION/), OAUTH2 (HTTPS://BLOG.OAUTH.IO/CATEGORY/OAUTH2/), OAUTH2 FLOWS (HTTPS://BLOG.OAUTH.IO/CATEGORY/OAUTH2-FLOWS/), SECURITY (HTTPS://BLOG.OAUTH.IO/CATEGORY/SECURITY/), SSO (HTTPS://BLOG.OAUTH.IO/CATEGORY/SSO/)

OAuth2 Introduction Through Flow Diagrams in 5-minutes

September 6, 2018 by oauthio Leave a comment (https://blog.oauth.io/introduction-oauth2-flow-diagrams/#respond)

Introduction to OAuth2

OAuth2 is a standard for streamlining the process of enabling a user to grant authorization to a web service or application to access her data or perform something on her behalf on another web service (OAuth provider). There are 4 different OAuth2 flows (https://oauth.net/2/grant-types/), and to understand which best suit your needs, refer to this (https://blog.oauth.io/choose-oauth2-flow-grant-types-for-app/). In this article, we want to create a simple introduction that enables engineers, managers, and investors to understand the high level flow of each OAuth2 grant type quickly at a glance through OAuth2 flow diagrams.

In our previous article (https://blog.oauth.io/understand-oauth2-grant-types-by-spotting-the-difference/), we introduced the four OAuth2 grant types by comparing them from the perspective of security, implementation difficulty, and use cases. In the same vein here, we will compare the four OAuth2 grant types side-by-side, but from the perspective of data flow between the parties (user, web browser/native app, web service, OAuth provider) in each OAuth2 grant type.

Similarity between all OAuth2 Grant Type Flow Diagrams

Every OAuth2 grant type flow has the same goal:

- To obtain authorization key/access token, which represents a set of permissions, from the user, ***and*** perform something on her behalf

Achieving this goal is a 2-part flow:

1. Get Access Token
    - Acquire the authorization key/access token for the user from the OAuth provider, e.g., Twitter
2. Use Access Token
    - Use the authorization key/access token to perform something by calling a protected API endpoint on behalf of the user, e.g., post a tweet

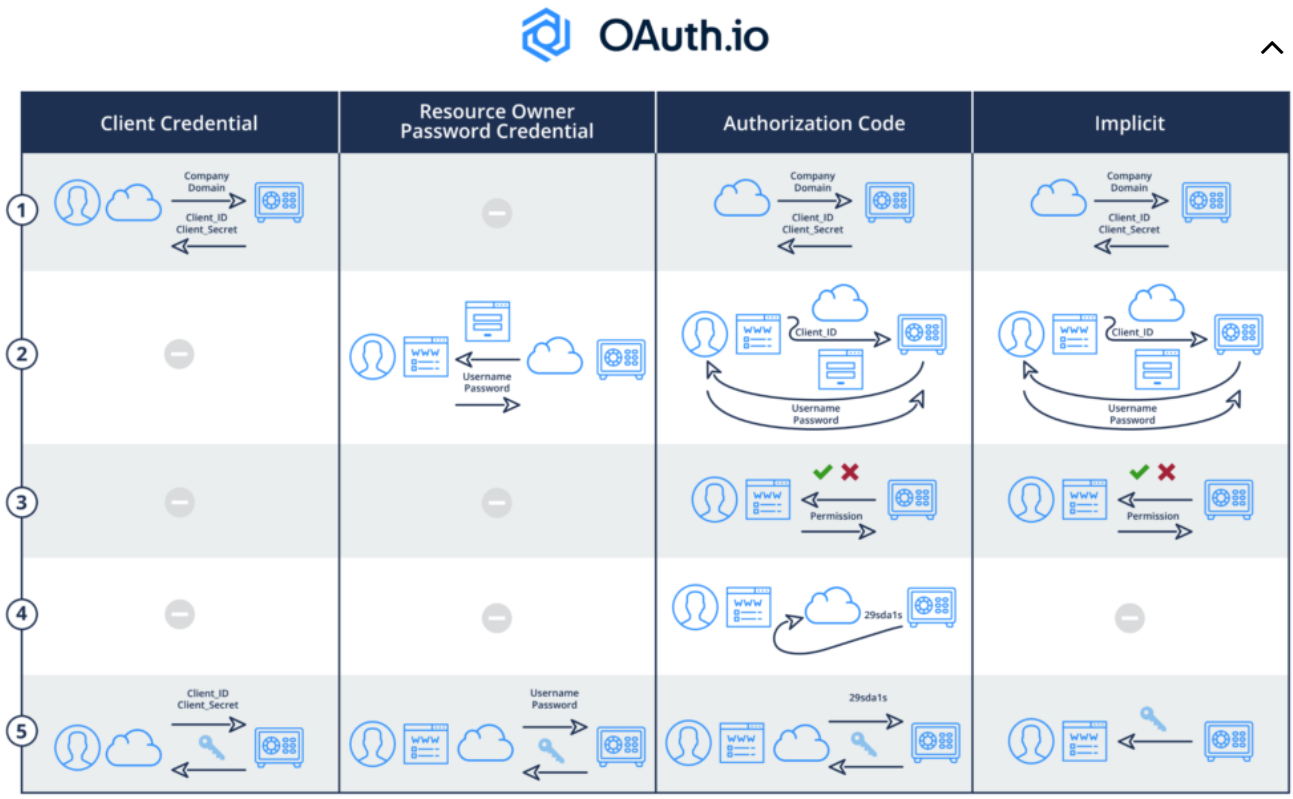
Differences Between OAuth2 Grant Type Flow Diagrams

Every OAuth2 grant type flow differs ***only*** in the first part of the main flow:

- Get Token Acquisition

In principle, the Get Access Token flow has 5 steps (as shown in the diagram below):

1. Pre-register Client (App) with OAuth Server to get Client ID/Client Secret
2. OAuth Server authenticates user when she clicks on the App's social login button, which is tagged with Client ID
3. OAuth Server solicits user permission to allow the App to perform something on her behalf
4. OAuth Server sends secret Code to App
5. App acquires Key/Access Token from OAuth Server by presenting secret Code and Client Secret



(https://blog.oauth.io/wp-content/uploads/2018/09/OAuth-Flow-Comparison.png)

NOTE:

- The 'cloud' icon represents the App
- The 'www' icon represents the User/Browser
- The 'safe' icon represents the OAuth Server
- 'N.A.' means 'not applicable', i.e., the step is not required in the grant type flow

Authorization Code Grant Type Flow

You can easily tell that Authorization Code (3rd from left) grant type flow is the most involved, i.e., it has all 5 steps, and it is also the most secure as the key/access token is only issued to the App (backend), which is well-guarded (Step #5), thus reducing the attack surface of the system.

Implicit Code Grant Type Flow

Implicit grant type flow (rightmost) is most similar to Authorization Code except Step #4 is not required, i.e., the OAuth server hands the key/access token directly back to the User/Browser This increases the attack surface of the system moderately since the key/access token in stored on the browser, which is more exposed to the internet than the App (backend). This is often mitigated by provisioning key/access token that cannot be renewed and has shorter expiration. This flow is necessary when the App does not have a backend, such as a single-page app (SPA) or a native mobile app.

Client Credential Grant Type Flow

Client Credential grant type flow (leftmost) is easy, having only 2 steps but it requires the User to be the same entity as the App since the User will use the client id/client secret of the App to identity herself when communicating with the OAuth Server in Step #5. Thus the use case for this grant type is limited, but secure as the key/access token is handed over to the App (backend), which is well-guarded.

Resource Owner Password Credential Grant Type Flow

Resource Owner Password Credential grant type flow is also easy having only 2 steps but it requires the User to hand over username/password to the App in Step #2. Unless the App and the OAuth Server belong to the same entity, handing over the username/password to the App empowers the App with all the privileges as the User, which is a security risk.

Conclusion

Each OAuth2 grant type flow comprises 2 flows: get access token and use access token usage flow. The latter is the same for all OAuth2 grant types, while the former varies across grant types. Despite the variation, the former can still be generally broken down into 5 steps, with the variation arising from the parties involved in each step.

If you want to do not want to deal with OAuth complexity and prefer to effortlessly consume OAuth as a web service, become an OAuth provider as a platform, or use OAuth for Single-Sign On (SSO) please chat (https://oauth.io?utm_source=blog&utm_medium=post&utm_content=oauth-intro) or email (support@oauth.io) with the experts at OAuth.io or schedule a call (http://calendly.com/oauthio).

flow diagram (https://blog.oauth.io/tag/flow-diagram/)
grant types (https://blog.oauth.io/tag/grant-types/)
introduction (https://blog.oauth.io/tag/introduction/) oauth (https://blog.oauth.io/tag/oauth/)
oauth access token (https://blog.oauth.io/tag/oauth-access-token/)
oauth2 (https://blog.oauth.io/tag/oauth2/)

PREVIOUS POST (HTTPS://BLOG.OAUTH.IO/UNDERSTAND-OAUTH2-GRANT-TYPES-BY-SPOTTING-THE-DIFFERENCE/)

NEXT POST

You may also like

(https://blog.oauth.io/3-steps-to-migrate-google-plus-oauth2-sign-in-api-shutdown/)

3 Steps To Migrate Google+ (Google Plus) OAuth2 Sign-in & API Before Shutdown
(https://blog.oauth.io/3-steps-to-migrate-google-plus-oauth2-sign-in-api-shutdown/)
January 21, 2019

(https://blog.oauth.io/simplest-oauth-introduction-outcome-of-dedication-to-oauth/)

Simplest OAuth Introduction: The Outcome of Years of Dedication to OAuth
(https://blog.oauth.io/simplest-oauth-introduction-outcome-of-dedication-to-oauth/)
October 12, 2018

Leave a Reply

Your email address will not be published. Required fields are marked *

Name*

Email*

Website

POST COMMENT

OAuth.io

Welcome to our blog, where we'll be talking all things social logins, authorizations, user management, and more 🙂

Stay tuned for some good content!

New Articles

(https://blog.oauth.io/3-steps-to-migrate-google-plus-oauth2-sign-in-api-shutdown/)
3 Steps To Migrate Google+ (Google Plus) OAuth2 Sign-in & API Before Shutdown
(https://blog.oauth.io/3-steps-to-migrate-google-plus-oauth2-sign-in-api-shutdown/)

(https://blog.oauth.io/simplest-oauth-introduction-outcome-of-dedication-to-oauth/)
Simplest OAuth Introduction: The Outcome of Years of Dedication to OAuth
(https://blog.oauth.io/simplest-oauth-introduction-outcome-of-dedication-to-oauth/)

(https://blog.oauth.io/introduction-oauth2-flow-diagrams/)
OAuth2 Introduction Through Flow Diagrams in 5-minutes
(https://blog.oauth.io/introduction-oauth2-flow-diagrams/)

(https://blog.oauth.io/understand-oauth2-grant-types-by-spotting-the-difference/)
Introduction to OAuth2 Grant Types in 2-Minutes With Pictures
(https://blog.oauth.io/understand-oauth2-grant-types-by-spotting-the-difference/)

(https://blog.oauth.io/oauth2-flow-grant-types-in-pictures/)
The OAuth2 Grant Type/Flow Introduction – In Simplified Pictures
(https://blog.oauth.io/oauth2-flow-grant-types-in-pictures/)