OAuth Demystified

Harippriya S

TTL Session - Jan 2018

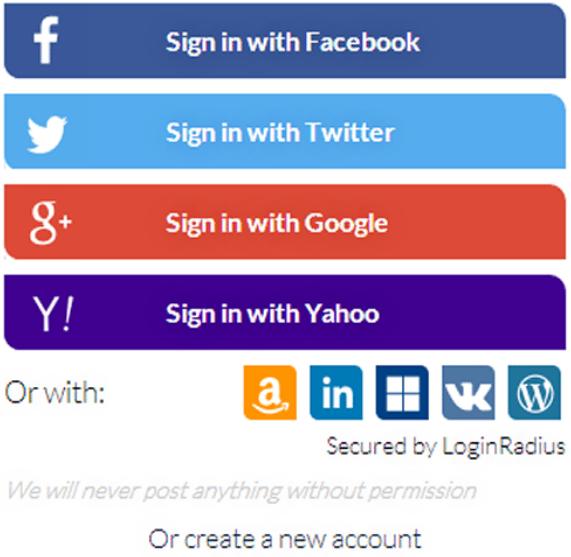
MICRO FOCUS

Agenda

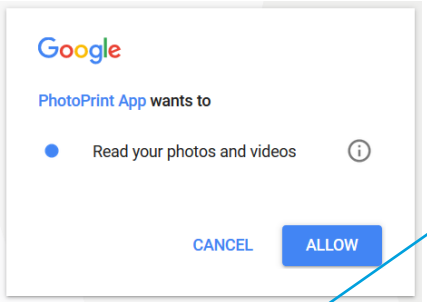
- OAuth 2.0 Overview
- Use Cases it solves
- Using OAuth

Social Login: Powered by OAuth and OpenID

OAuth powers the billions of social logins that happen every day!



Scenario: Photo printing application



What is OAuth 2.0

> The OAuth 2.0 authorization framework enables a third-party application (photo printing app) to obtain limited access to an HTTP service or data (Picasa APIs) on behalf of a resource owner (user).
>
> *– RFC 6749*

OAuth 2.0 defines how to:

- Delegate access
- Selective Access
- Revoke Access

OAuth does not define how to authenticate the user.

Why OAuth Matters

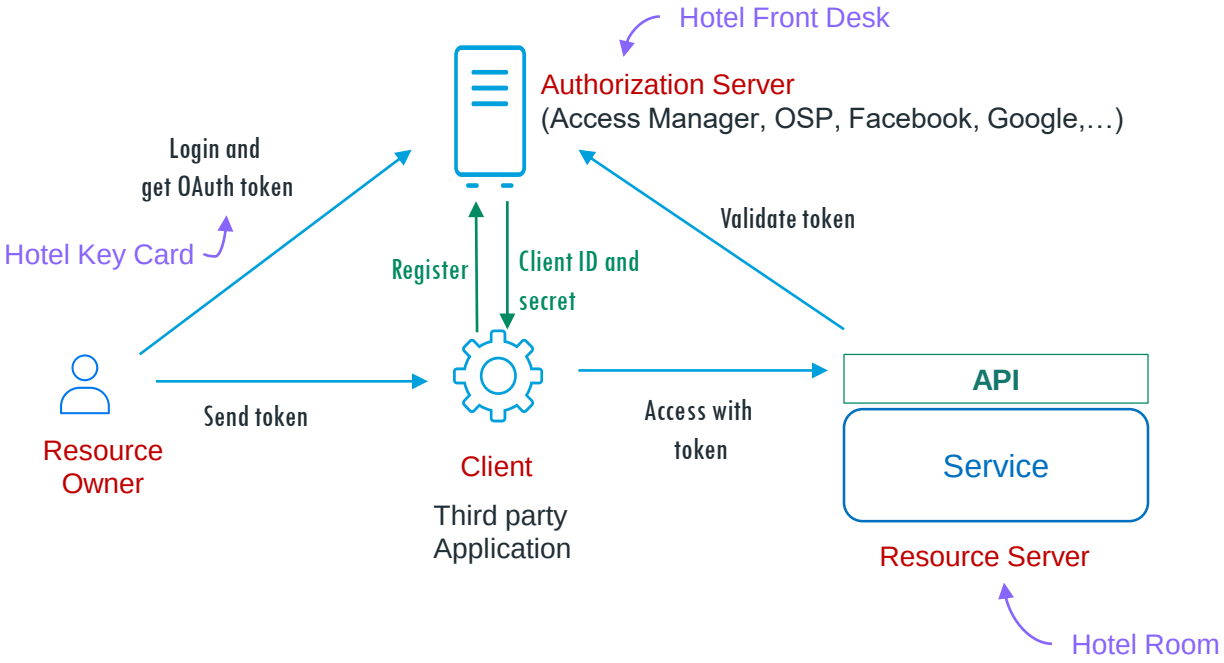
- API Market is on fire
  - 70% of US organizations actively use APIs.
  - 50% of B2B collaborations are through APIs.

- Why APIs?
  - Don’t reinvent the wheel.
  - Loosely coupled reuse → Quicker go to market.

- OAuth secures these APIs!

How It Works

Actors in OAuth 2.0



Tokens are the key.

- ✓ Tokens have expiration date.
- ✓ Tokens can be renewed.
- ✓ Tokens can be revoked.
- ✓ Tokens have scopes (permissions)

Token Types

Access Token

- Like a session. Will expire
- Contains permissions (scopes)

Refresh Token

- Like a password.
- Exchange for a new Access Token.
- Can be revoked.

ID Token

- Contains user details (claims)
- Part of OpenID Connect protocol.

OpenID Connect

- Identity protocol on top of OAuth 2.0 framework.
- Oauth Provider issues Access token and ID token.
- Helps solve some of the limitations of Oauth:
  - client can validate user and get user information.
  - client can ensure that the token was intended for it.

Short Quiz

1. How is OAuth different from older delegated access mechanisms?

   Ans: Credentials are not shared. Can control the level of access.

2. Uber grew to a $62B enterprise in 6 years. What made this growth possible?

   Ans: By API reuse

   Positioning via OS, route calculation and maps by MapKit and GoogleMaps, push notification by Cloud Messaging, payment by Braintree, receipts by Mandril, etc.

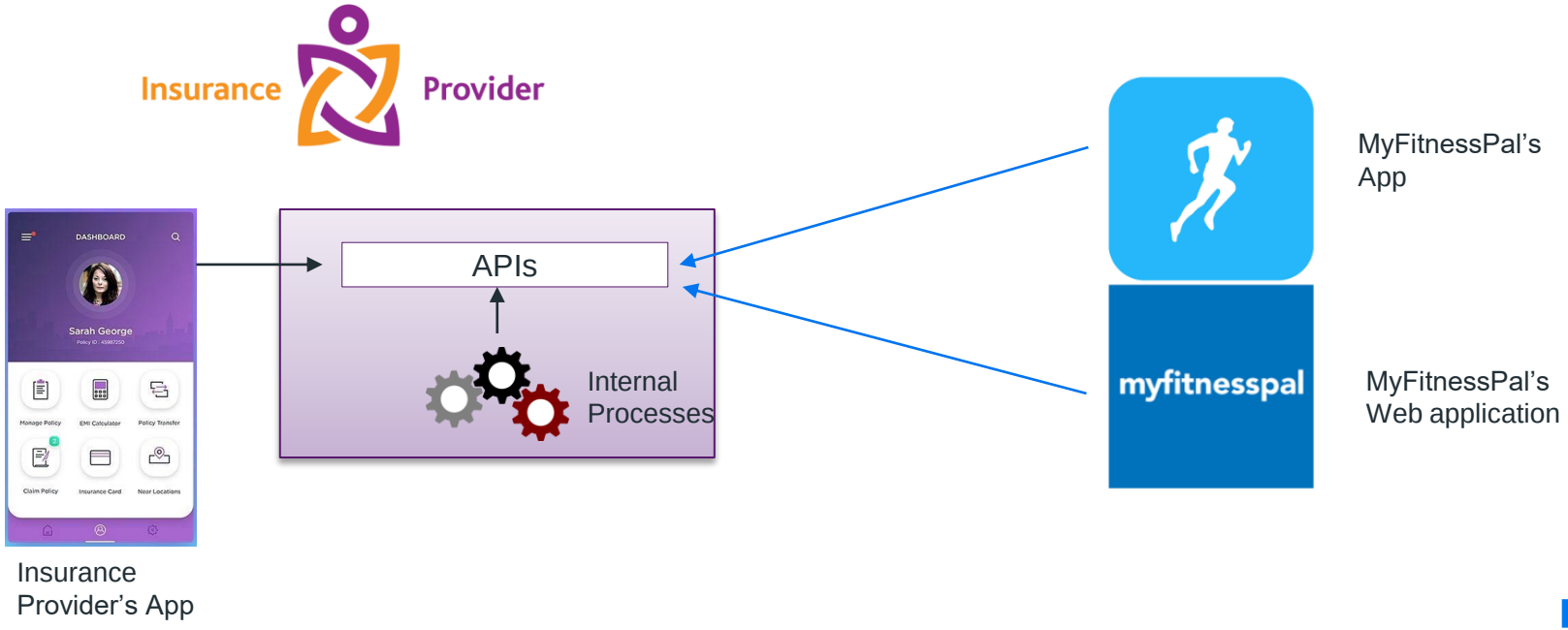
3. Which company started the idea of OAuth/OpenID?

   Ans: Twitter

Use Cases and OAuth Flows

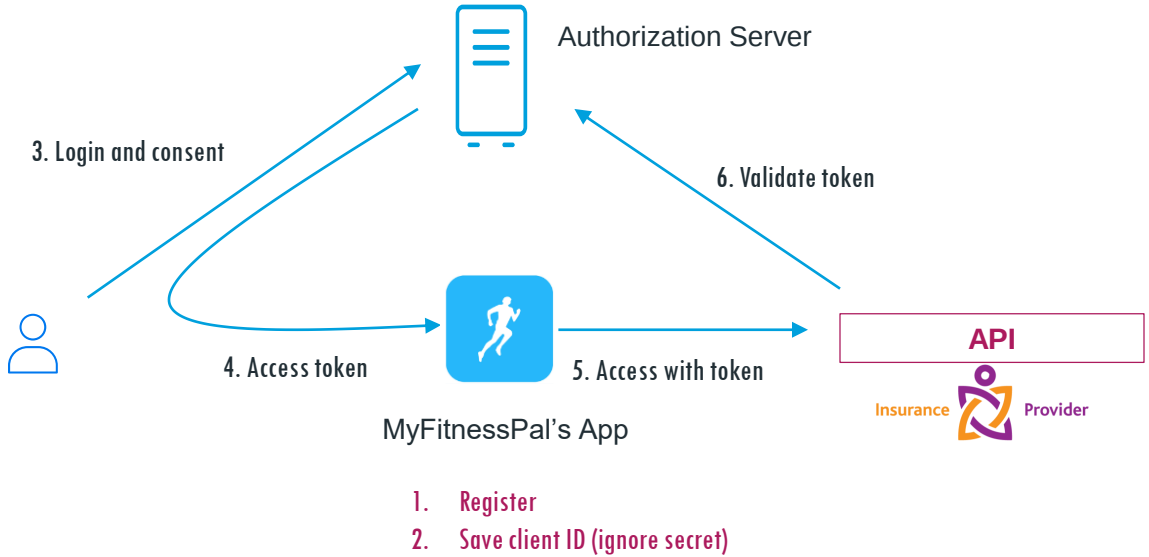
OAuth Flows

Grant Types: Different ways to get Access token depending on the type of client.



#1 Public Clients

Public Client: A client running on user's device and hence cannot securely store the client secret.



Implicit Grant:

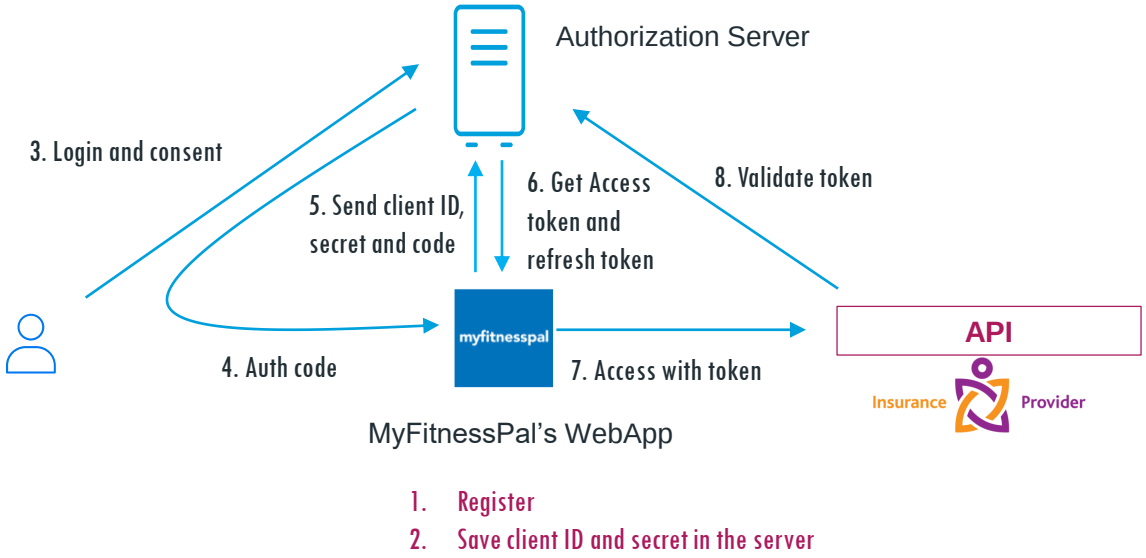
- ✓ Simple to implement.
- ✓ Access Token sent thru browser.
- ✓ No refresh token.

Suitable for:

- ✓ Browser based apps (like AngularJS)
- ✓ Mobile apps
- ✓ Desktop apps

#2 Confidential Clients

Confidential Client: Can safely store the client secret.



1. Register
2. Save client ID and secret in the server

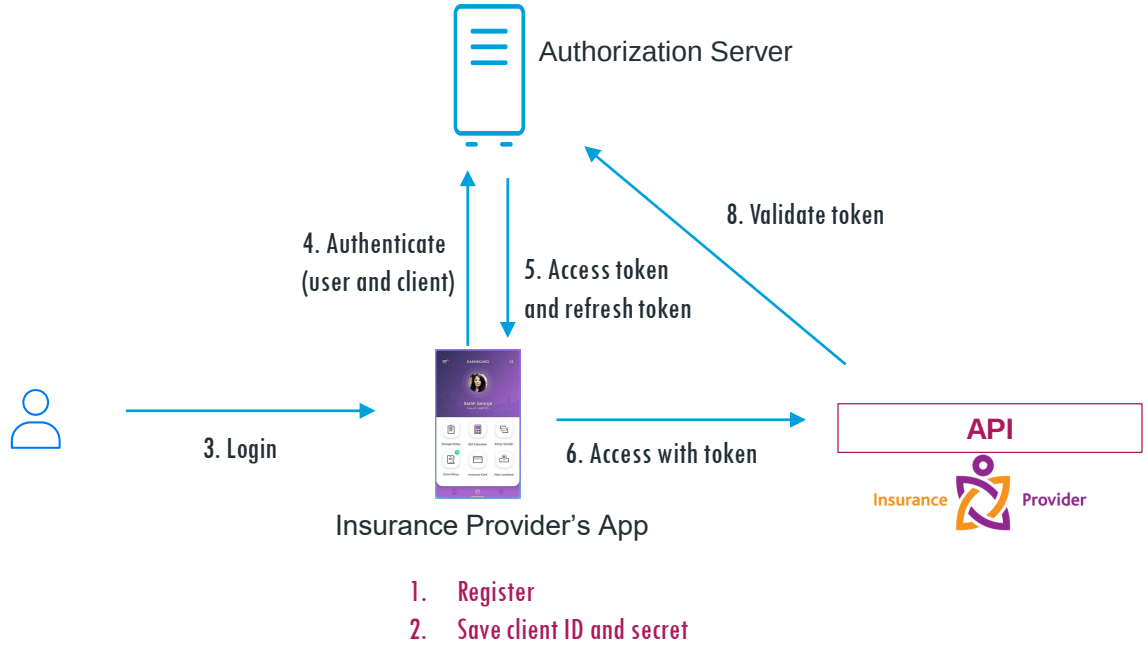
Authorization Code Grant:

- ✓ Most secure of all OAuth flows.
- ✓ Use Refresh token for background activities.
- ✓ Clients should handle browser redirects and HTTPS.

Suitable for:

- ✓ Apps running on the server
- ✓ Native apps (with PKCE)

#3 Trusted Clients



Resource Owner Password Credentials Grant:

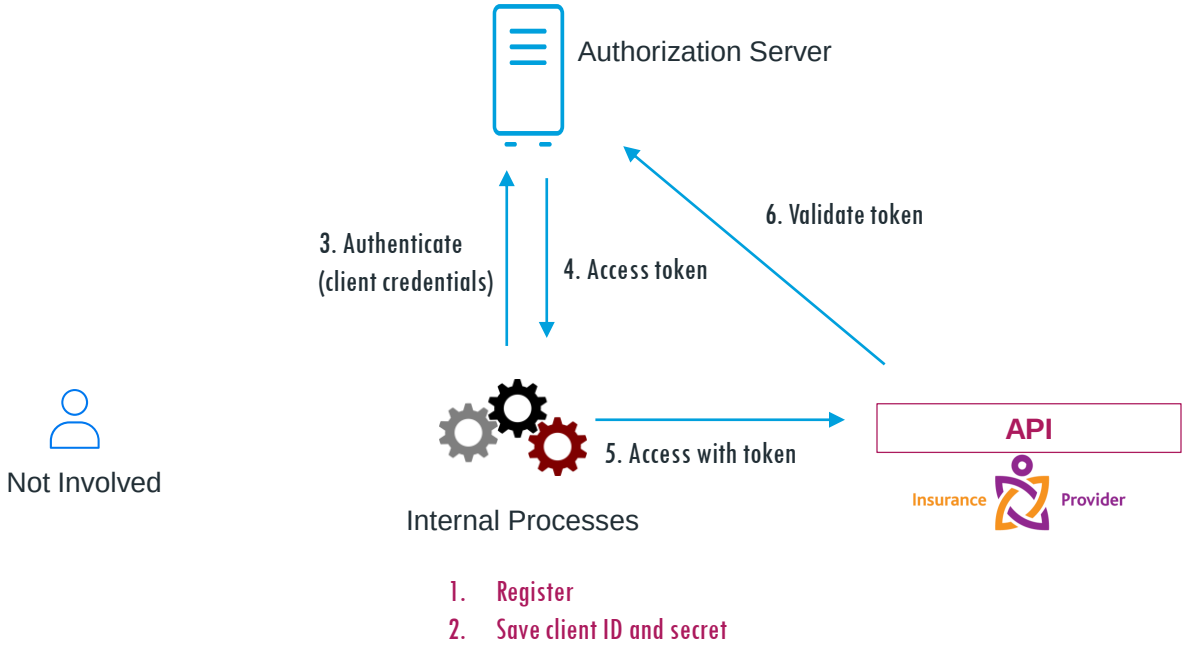
- ✓ Client has access to credentials.
- ✓ RFC: Use it as last option.
- ✓ Refresh token supported.

Suitable for:

- ✓ Highly trusted apps
- ✓ Legacy apps migrating to OAuth

MICRO FOCUS

#4 Inter-service Communication



Client Credentials Grant:

- ✓ No user involved.
- ✓ Authenticate with client credentials.
- ✓ No refresh token.

Suitable for:

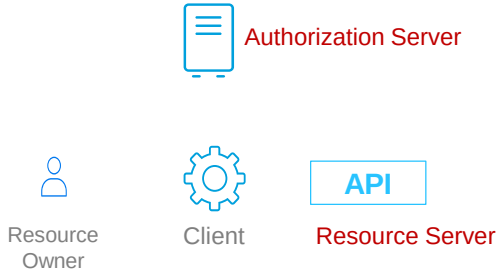
- ✓ Headless clients (game consoles, printers, etc)
- ✓ Batch processing scripts

MICRO FOCUS

Demo: OAuth Behind the Scenes

Using OAuth

Securing REST APIs



Setup Authorization Server

Install Access Manager or OSP or Spring OAuth AS or Auth0

Create a web application containing the REST APIs

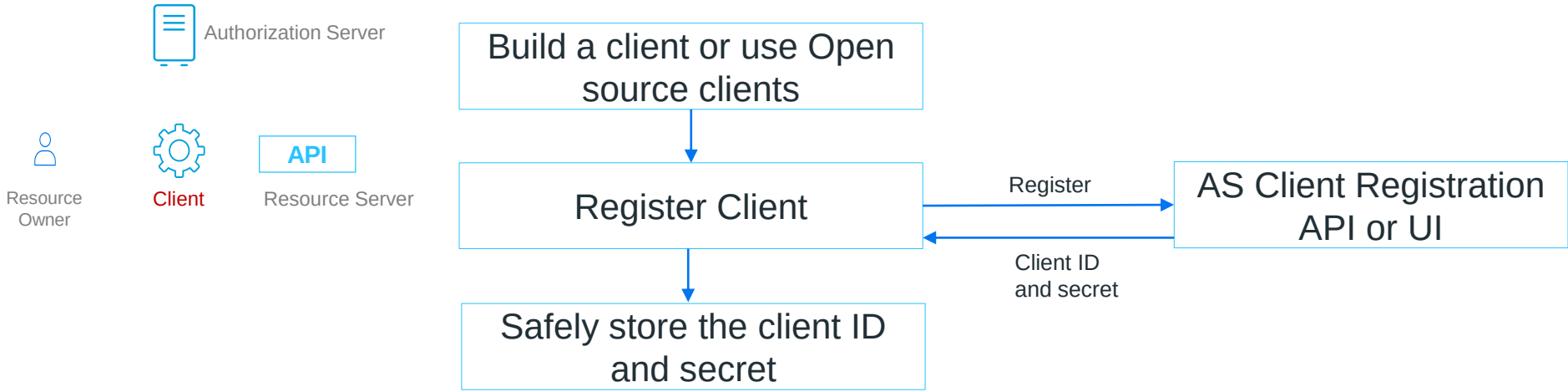
Setup Resource Server

Install web server like tomcat. Deploy and protect the API.

Enable the application to handle OAuth

Define the scopes required for each API

Setting up Oauth Client



Accessing REST APIs from OAuth Client

Pre-requisite: User must have been logged in.

1. Request for Access Token

```
https://authserver.netiq.com/nidp/oauth/nam/authz?
response_type=token+id_token&client_id=4e4ae330-1215-4fc8-9aa7-
79df8325&redirect_uri=https://client.example.com/callback&scope=email+OpenID&state=s1234&nonce=n123
```

2. Parse the response to extract the token

```
https://client.example.com/callback#token_type=bearer&access_token=/wEBAAUFACAjDfPtn
d/zlOWPpN/kV1Jtt3nxCPtzHyUH~&expires_in=3600&id_token=eyJhbGciOiJSUzI1NiJ9.eyJp
c3MiOiJo&scope=email&state=s1234
```

3. Access the REST API

Send the access token in the header of the API request

```
GET /sample/api/profile HTTP/1.1
Host: www.resourceserver.com:8443
Accept: */*
Authorization: Bearer =/wEBAAUFACAjDfPtnd/zlOWPpN/kV1Jtt3nxCPtzHyUH~
```

MICRO FOCUS

Summary

- Oauth 2.0 enables third party apps to access on behalf of users.
- OpenID Connect enables to include user information.
- Different grants types (flows) to support different OAuth clients.
  - Public client (Browser UIs) – Implicit Grant
  - Confidential clients (Web apps) – Authorization Code Grant
  - Trusted / Legacy clients – Resource Owner Credentials Grant
  - Machine to machine – Client Credentials Grant
- All interactions with Authorization Server are through its endpoints.
- Use Google Playground or Auth0 to get a feel for Oauth.
- Use MicroFocus products like Access Manager or OSP or Spring Oauth to try out.

Resources

https://www.oauth.com/oauth2-servers/background/

MicroFocus OAuth Whitepaper

https://www.netiq.com/docrep/documents/rsqj13lwgf/oauth_securing_apis_mobile_and_beyond_wp.pdf

Spring REST API + OAuth2 + AngularJS - http://www.baeldung.com/rest-api-spring-oauth2-angularjs

Oauth2 Reference for Access Manager (includes sample application, NAM Oauth playground) –

https://www.netiq.com/communities/cool-solutions/oauth2-reference-for-access-manager/

When to use which Oauth flow –

https://medium.com/@robert.broeckelmann/when-to-use-which-oauth2-grants-and-oidc-flows-ec6a5c00d864

www.microfocus.com