**Authentication**

The process of confirming user identity is Authentication.

**Oauth(Token authentication)**

<https://stormpath.com/blog/what-the-heck-is-oauth>

It is protocol running over HTTP and authorize Devices, APIS,Servers and applications using tokens

Ouath does not validate the user identities. Authentication service will do that.

**Oauth grant type**

* **The Authorization Code** (server-side)
  + Meant to be on the web server login by Facebook
  + User click log-in redirected to identifyId then facebook returns the Auth code
  + My app receives then POST call to identifyId to validate it again
  + If validated then can access the personal identification
* **The Implicit** (client-side)
  + Meant to be on the web server login by Facebook
  + User click log-in redirected to identify Id then facebook returns token
  + By token receives we can access user information - No POST call
* **The password (mobile)**
  + Simplify authorization is a good choice for Mobile
  + Asking only user name and password in our side - no redirect, 1 POST
  + We cannot redirect and use google to login to our app -
* **The Client Credential (application code)**
  + Non-user related tasks
  + We make application that does not need to interact with user in browser.
  + Application make a request to identity provider to get token
  + After token receives we can make API calls

**SSO**It is high level term to describe a user uses the same credential to access multiple domain

Difference between Oaut 1 and OAuth 2

**OAuth 1.0** requires client to send two security tokens **for**each API call, and use both to generate the signature.   
OAuth flow is 8 steps , but Oauth 2 is 6 steps

Difference between Oauth 1 and 2

1. Client application registers with provider, such as Twitter.
2. Twitter provides client with a “consumer secret” unique to that application.
3. Client app **signs** all OAuth requests to **Twitter with its unique “consumer secret.”**
4. If any of the OAuth request is malformed, missing data, or signed improperly, the request will be rejected.

**OAuth 2.0 Flow**

1. Client application registers with provider, such as Twitter.
2. Twitter provides client with a “client secret” unique to that application.
3. Client application **includes** **“client secret”** **with every request.**
4. If any of the OAuth request is malformed, missing data, or contains the wrong secret, the request will be rejected.

1. Basiec Auth

It is baseic, we endode user name and password in requestheader and then server-side will validate the cridential in filter chain or intercepter

2. Oauth 1

The protocol uses signatire wich often is HMAC to comboned token nonce and other request based information

provides 2 tokens : Access and Refresh token

Access token will expired soon but refresh token will get new access token

4. JWT

Jason web token; JWT has 3 sections :

Header

It describes the token vontain methadata about

payload (Claim)

Who this person is ?

What this person can access?

When the token is expired

Signature

Hashcode to verify integrity of the token

var headers = base64URLencode(myHeaders);

var claims = base64URLencode(myClaims);

var payload = header + "." + claims;

var signature = base64URLencode(HMACSHA256(payload, secret));

var encodedJWT = payload + "." + signature;

3. OAuth

It is framework it is not token format

It details how multiple different roles, users in your system, server side apps like an API, and clients such

as websites or native mobile apps, can authenticate with each other.

Roles :

Both applications and users can be one of the following:

Resource Owner

Resource Server

Client Application

Authorization Server

Client Types

A client is something that consumes your API. IT can be one of the following two types:

Confidential

Public

Client Profiles

There are also client profiles specified by the framework, that describe the kind of application type. They can be:

Web Application

User Agent

Native

Authorization Grants

An authorization grant is a set of permissions given by the resource owner to a client application. They can take the following forms:

Authorization Code

Implicit

Resource Owner Password Credentials

Client Credentials

Endpoints

In order for all of us this to work the following endpoints are required:

Authorization Endpoint

Token Endpoint

Redirection Endpoint

SAML

Exchanging authentication and authorization data between security domains called SAML is as part of SSO

Difference between Authentoicationin REST and SOAP :

WS-Security offers more protection than HTTPS would, and SOAP offers a richer API than REST