•JWT is a simple authentication protocol

**header.claims.signature**

•JWT Token is that it contains all necessary information about the token itself, user cannot change for example: any other claim, because this token is generated by the server with secret keyword.

Advantages of using JWT:

•The server side storage issues are gone.

•The client side code is easy.

Disadvantages:

•The JWT size could be larger than a session ID. It could affect network performance since it is included with each HTTP request.

•The data stored in the JWT is readable by the client. This may be an issue.

•The server side needs code to generate, validate, and read JWTs. It's not hard but there is a bit of a learning curve and security depends on it.

Anyone who gets a copy of the signing key can create JWTs. You might not know when this happens.

There was a bug in some libraries that accepted any JWT signed with the "none" algorithm so anyone could create JWTs that the server would trust.

•In order to revoke a JWT before it expires you need to use a revocation list. This gets you back to the server side storage issues you were trying to avoid.

OAuth

OAuth is used for authentication (identity), but it can be used to share other information like a content (I mean list of contents) the user has purchased. Oauth can also be used to grant access to write to data stored by the third party. You might **use OAuth to authenticate users and then use JWT for the session data**.

Advantages:

• No code for users to sign up or reset their password.

• No code to send an email with a validation link and then validate the address.

• Users do not need to learn or write-down another username and password.

Disadvantages:

• You depend on the third party in order for your users to use your service. If their service goes down or they discontinue it then you need to figure something else out.

Eg: how do you migrate the user's account data if their identity changes from "loo@z.com" to "sar@y.com"?

• In Oauth ,Usually you have to write code for each provider. Eg: Google, Facebook, Twitter.

• **You or your users might have privacy concerns. The providers know which of their users use your service.**

• You are trusting the provider. It is possible for a provider to issue tokens that are valid for one user to someone else. It could be for lawful purposesor not.