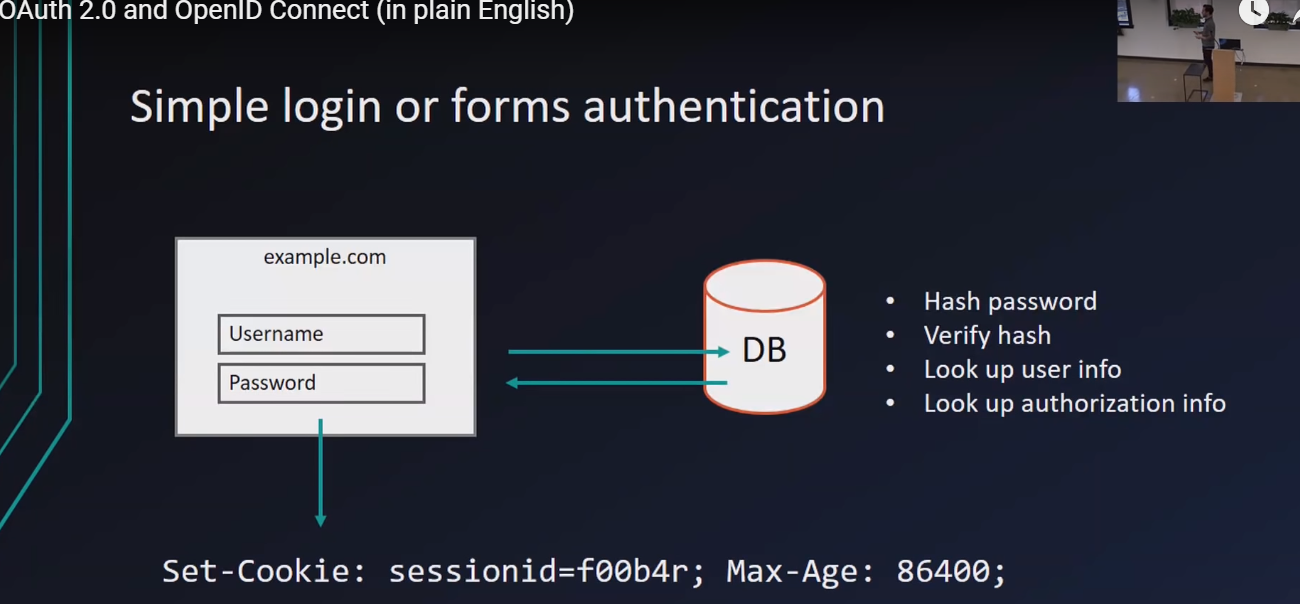
Old Login Approach::

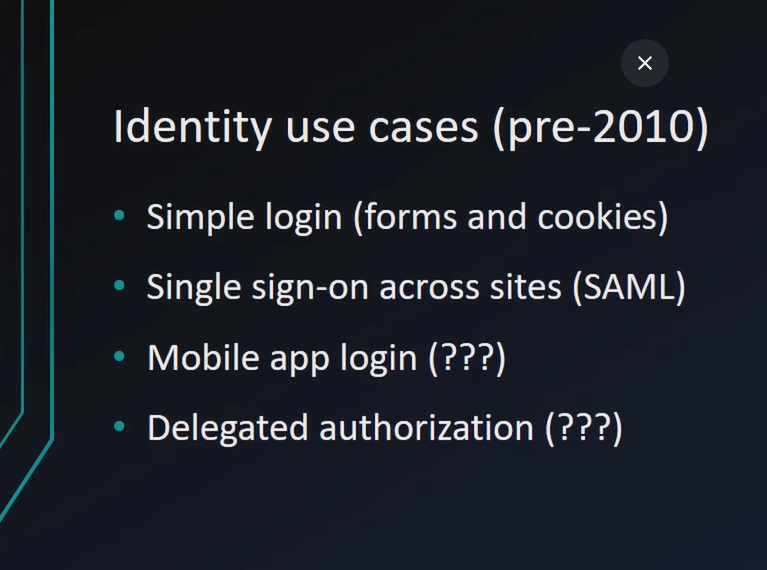


Downside of above approach is ::

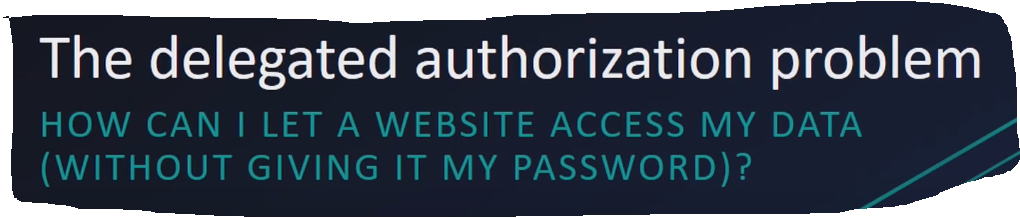
1.**Maintaince**

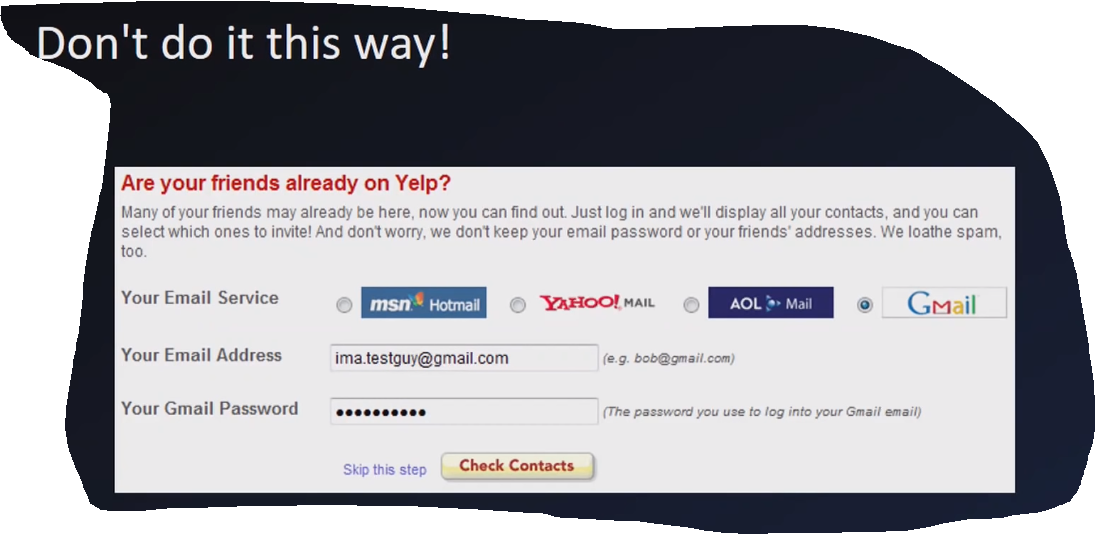
**2.Security**

So OpenId & OAuth2.0 are best practice for solving these problems.



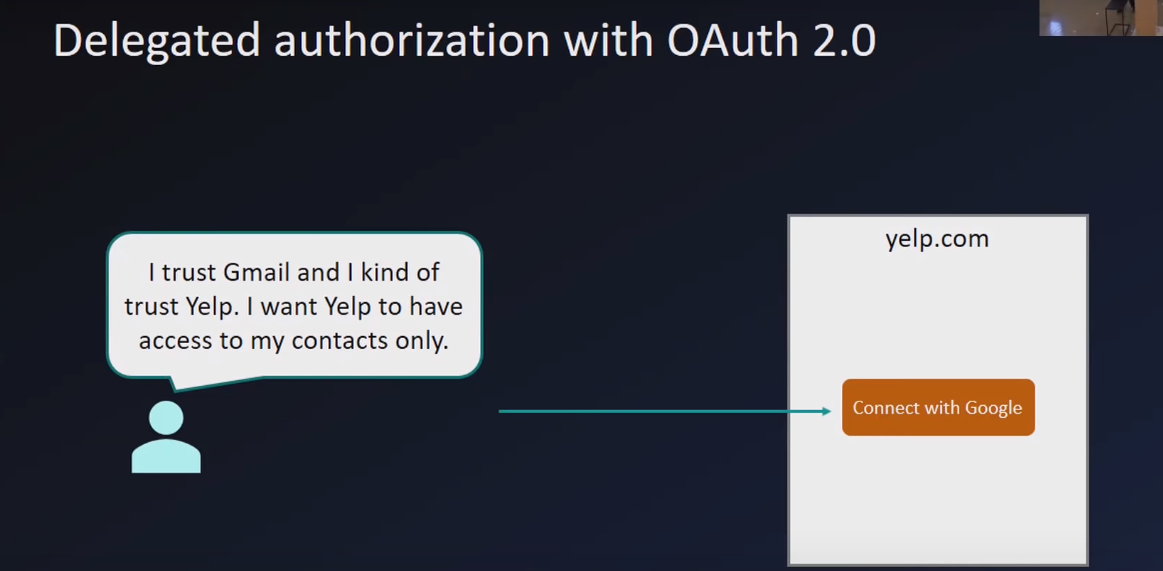
\***SAML** helps to login with same credentials with different systems.

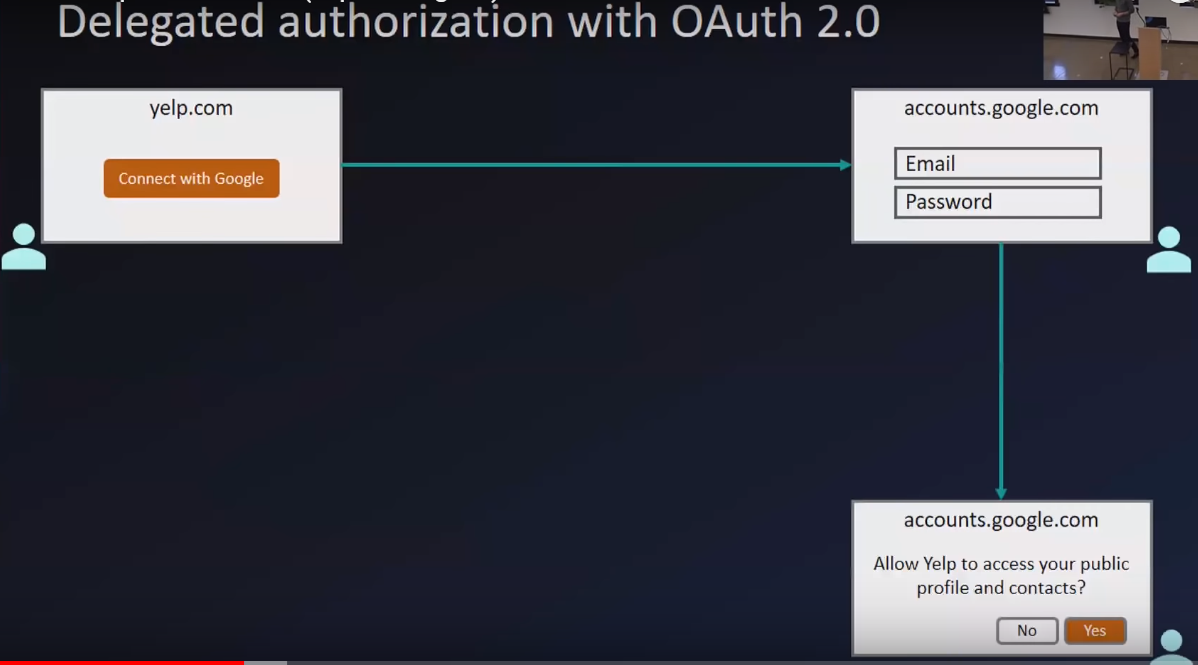




Above picture giving gmail id & pwd to YELP and they can do anything it’s all like giving 100% accessibility to other.

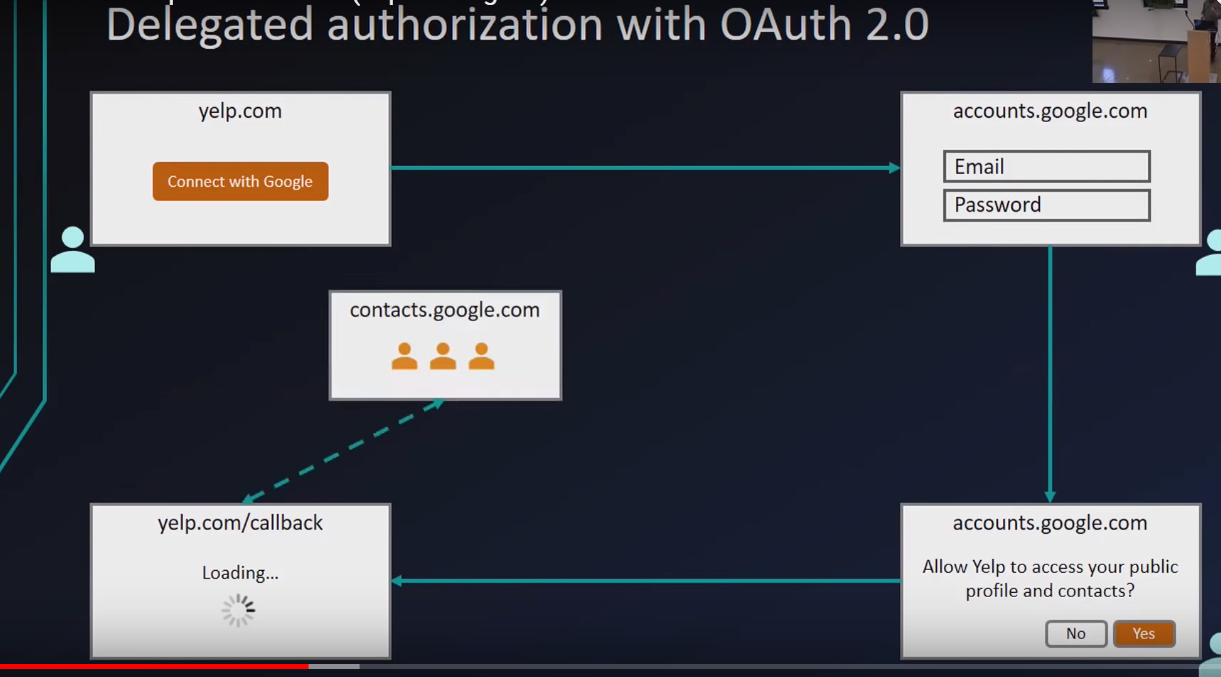
\*Bank still not addopted OAuth2.0 because we still enter ID and pwd for login.

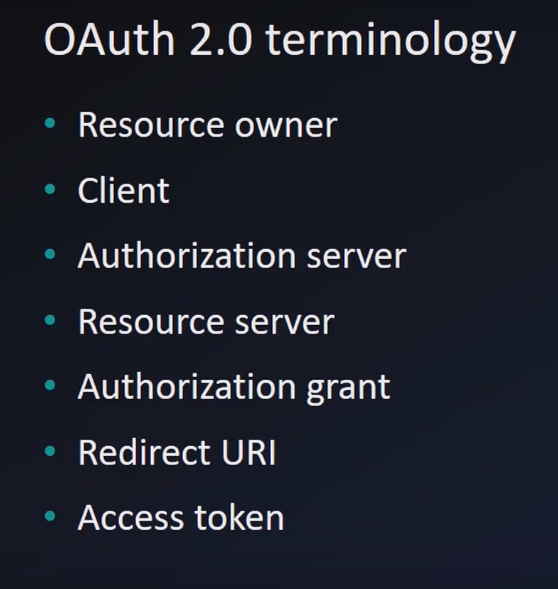




Now the way is different because we not providing id pwd to other app like YELP now we giving ID and PWD to google which I trust .

So after adapting OAUTH2.0 scenario got changed.





1.Resource Owner:: Myself is Resource Owner because holding info of my credentials.

2.Client :: is YEP.com

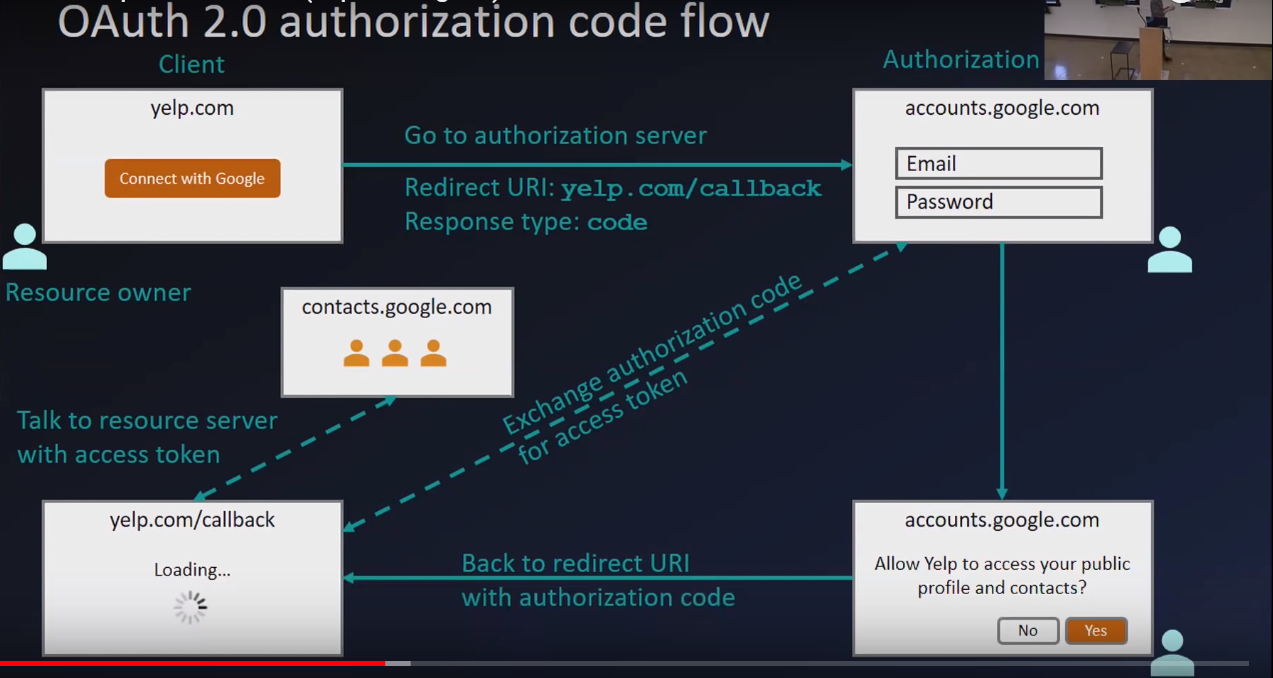
3.Authorization Server ::is a server which say yes I am giving permission to yes.

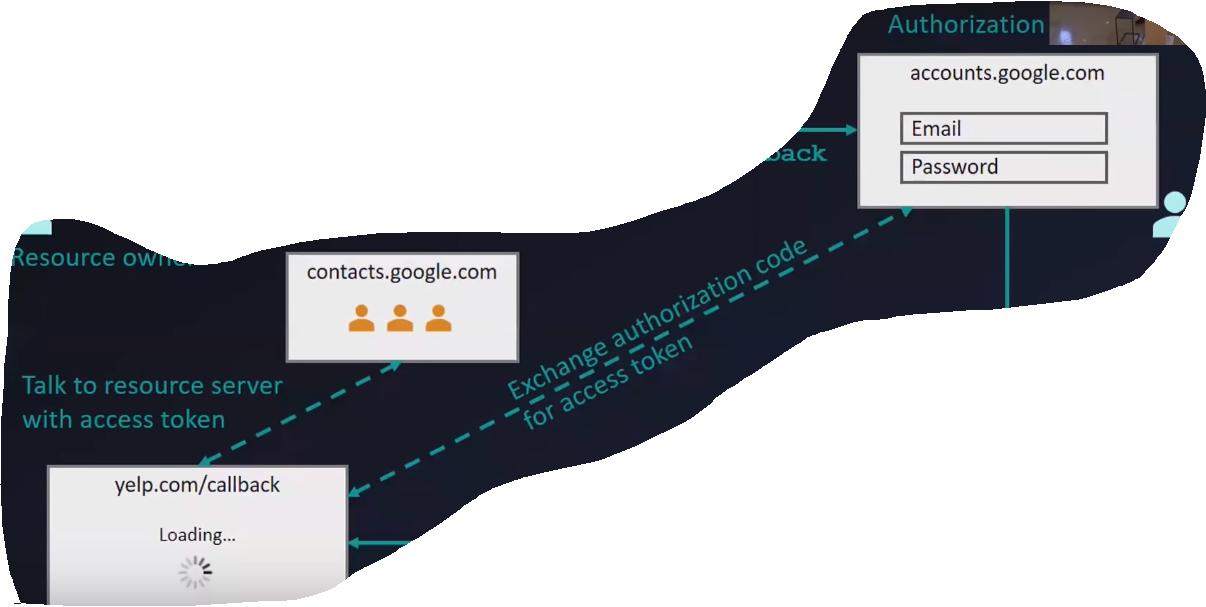
4.Resourse Server:: is a server which hold our Data as a user & pwd and example:: Google.com Authorization server and Resource Server is same and mainly some time they are different.

5.Authorizatio Grant:: say yes I am giving permission to Recourse to use me.

6.Redirect URI::After Authorization Grant user gets URI where they requested for access.

7.Access Token::





Above picture Yelp.com ask for access google contacts but google says who are you don’t know so now Yelp.com will receive access tokens then google will prove and they will allows to login access, now goggle ill know you have correct token id but with limited access you can’t get access for anything so .

OAuth2.0 termonology::

1.concent

2.Scope

Authorization server having list of sources .like context.read, context.write, email.read,email.delete

