Authentication: Who you are ? [Login Process]

Knowledge based [user , pwd]

Possession based [user, pwd + mfa or opt etc mobile or device verification that belongs to you]

Authorization : what you are allowed to be ?

Principal: Currently Logged User.

Authority: What are privileges or things allow to done , granular read or write

Roles: Group of Authority , if you belong to associate you have some permission instead of giving person by person

First Component in spring is Filter

Delegating Filter Proxy : Every Request is coming to application will be intercepted by this filter [Like front end]

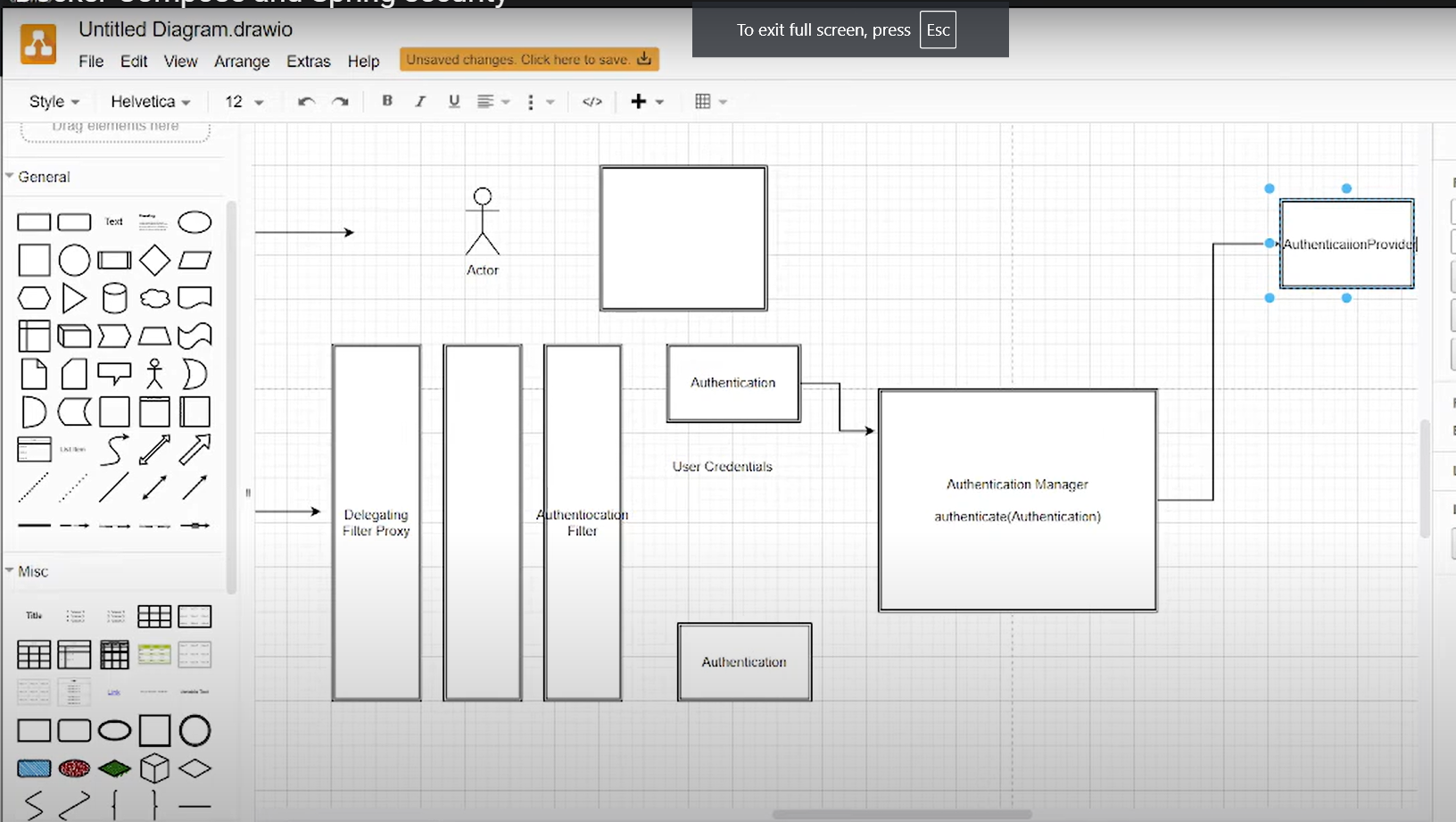
You can assume like Dispatcher servlet [ filter are servlet technology like api gateWay].

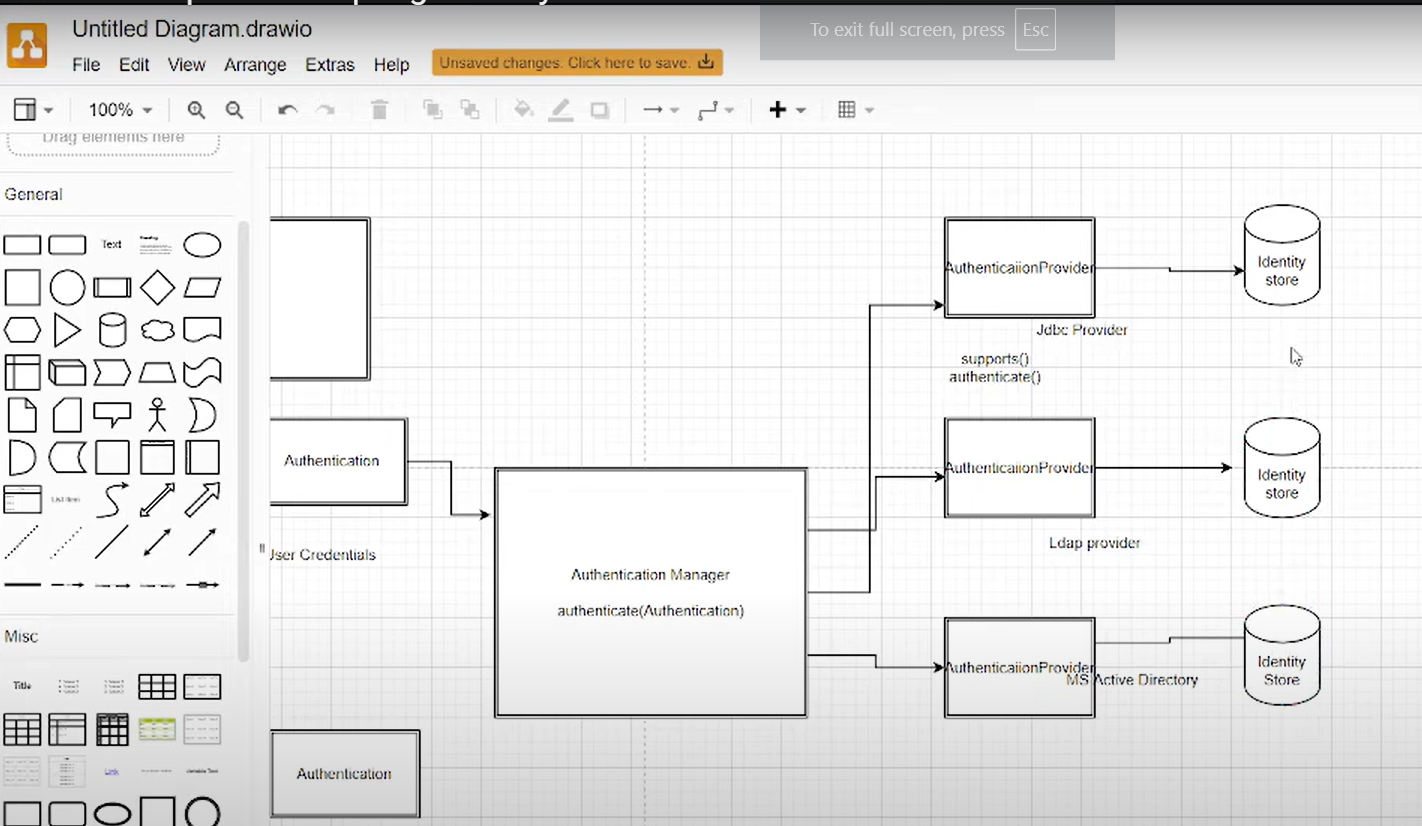
It will take request passes to subsequent filters

There are lot of filters, one of filters is Authentication filter

It will take userName and password will put into Authentication Object [User Credentials ]

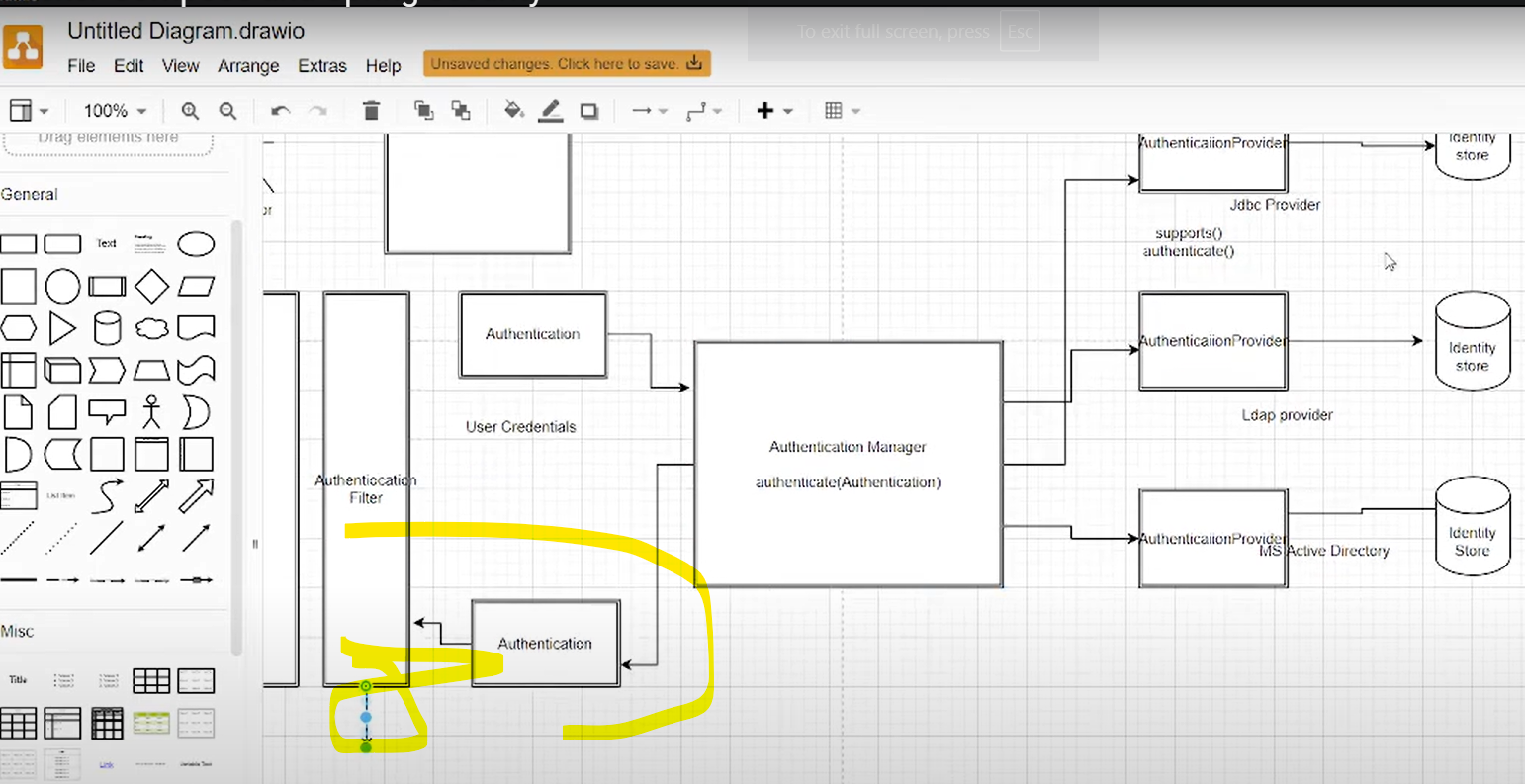
Authentication Object will pass into Auth manager contains Authenticate Method





Auth Manager calls support whether it can support jdbc or ldap or ms etc ,if it’s support then pull data compare data and will do auth , I mean your user and password correct or not

If it’s return true or false that is called principle authenticated user [currently logged user].

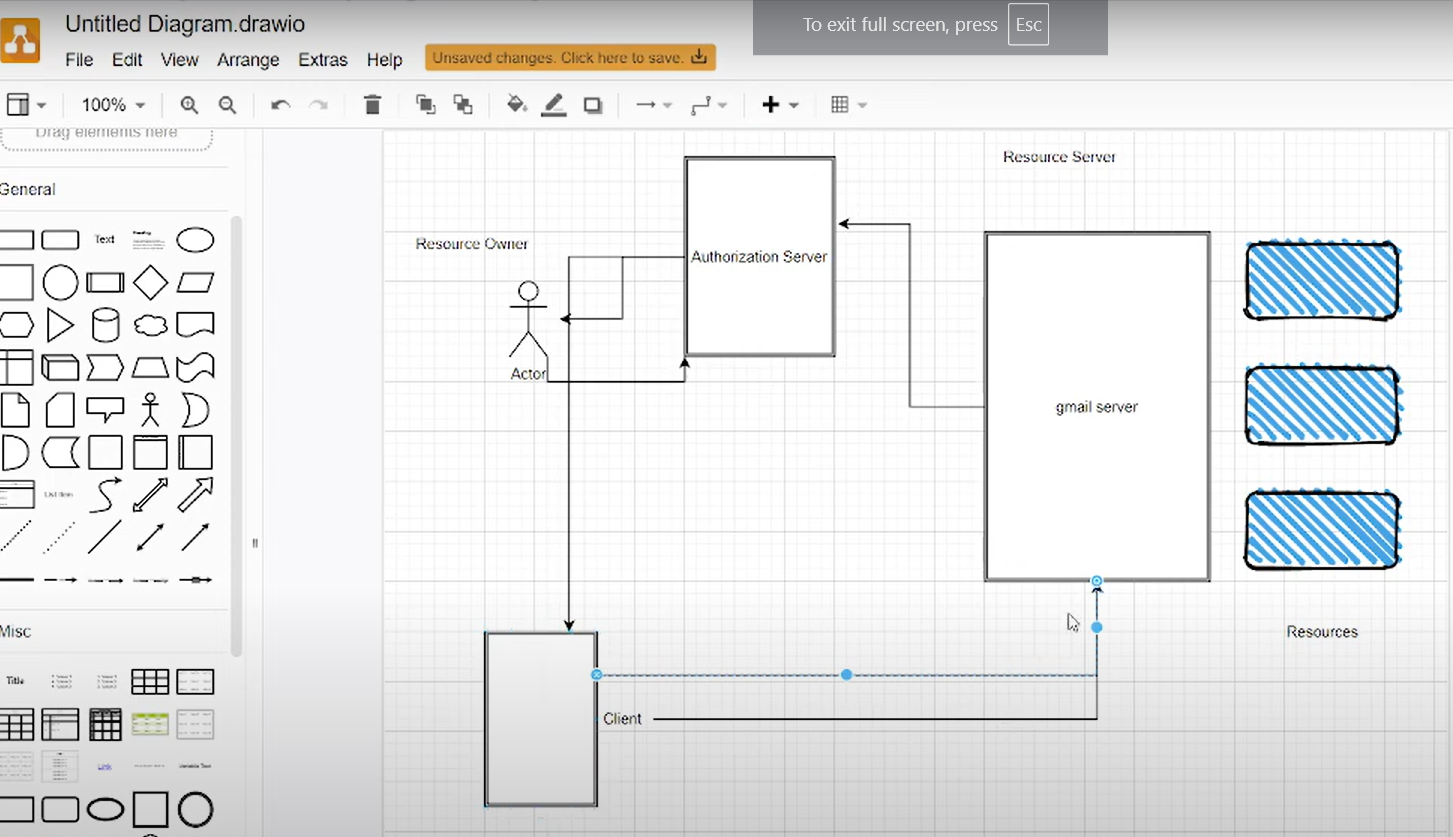


If login success , if any request with same user I won’t pass subsequent filters [currently logged user]

<https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/core/Authentication.html>

<https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/authentication/AuthenticationManager.html#authenticate(org.springframework.security.core.Authentication)>

**OAuth:**  one micro will call second micro , how will you verify ? everytime you can’t username and password that scenario is called OAuth open Authorization

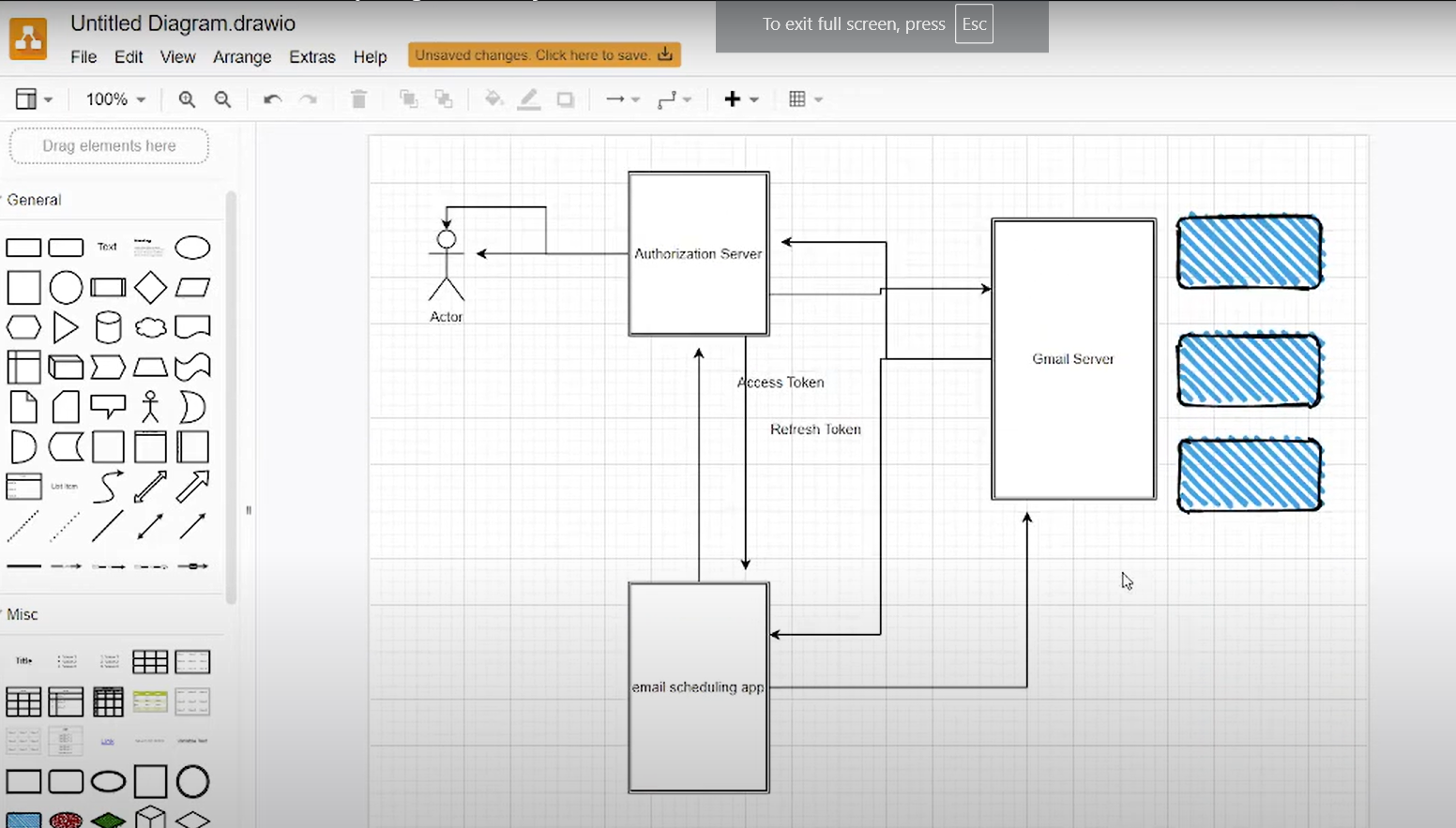


Client [Third party app] request access your gmail send notifications need to auth.

Auth sever send to user, if user agreed and given credentials then it’s it will given token instead of credentials .

Note: still your credentials also stored in auth server there is no concept of giving your password to everywhere

Client 🡪 Auth Server 🡪 User 🡪 Auth 🡪 Token 🡪 Client



Encode: <https://jwt.io/>

Decode:  
 <https://www.jstoolset.com/jwt>

<https://www.fusionauth.io/dev-tools/jwt-decoder>

Do not pass any sensitive information in webtoken



Do not share webtoken to anyone, they take this as impersonate user behalf of you

Auth : generate , validate

Micro: go auth