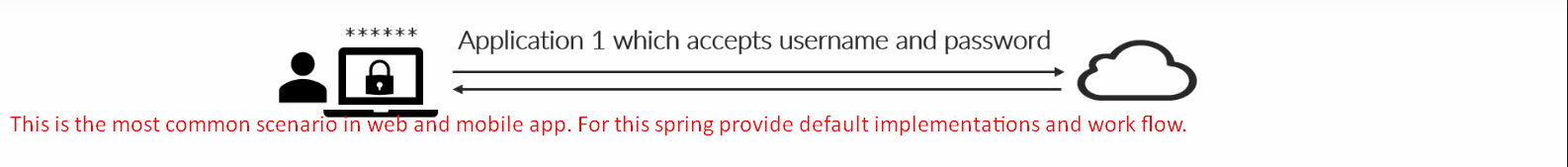
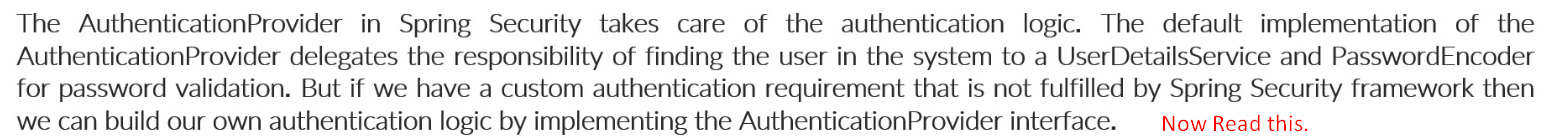
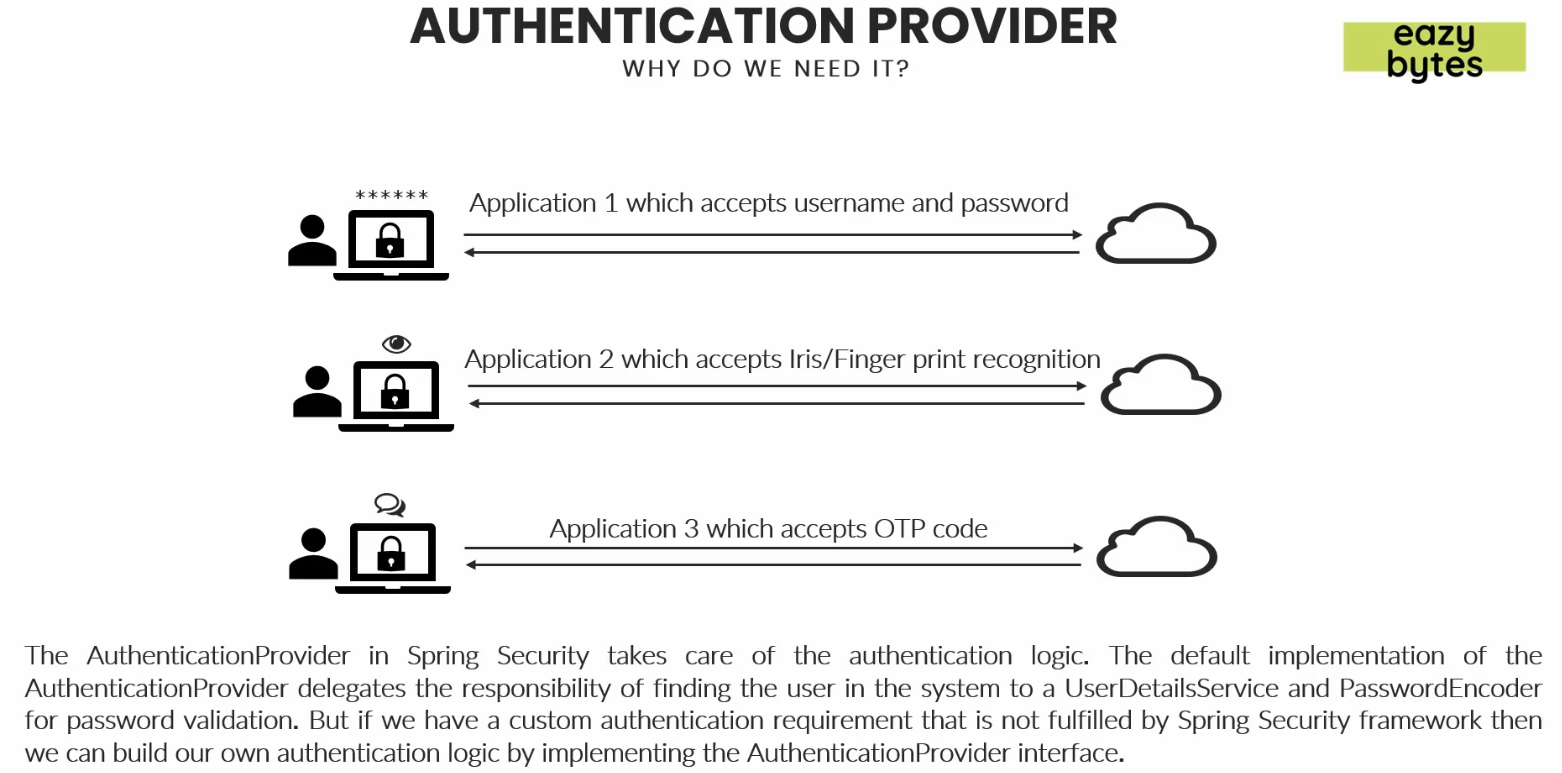
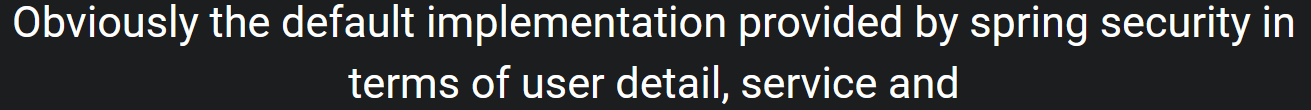
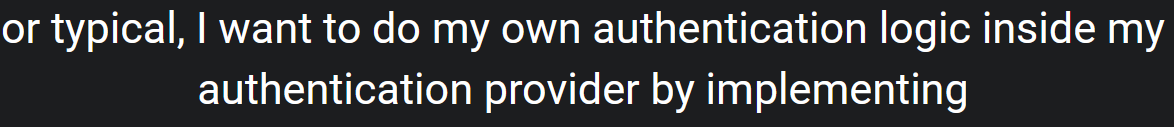
1. Let’s understand the scenario where we don’t want to leverage the UserDetailsService and PaswordEncoder but we want to give our own implementation for AuthenticationProvider.
2. Consider the following 3 applications.
3. 
4. 
5. Now suppose we have another kind of app where authentication is done with face scanning, figure prints or **iris** reorganization.
6. Another app where authentication is done with OTP sent over mobile or email.  
   
7.  password encoder is not good enough because in the scenarios of fingerprint scanning or face recognition 
8. **Good News**: Spring Security allows you to configure as many Providers as you like.
9. So, in the above scenario, we can configure 3 different kinds of providers in the same app.
10. Based on the nature of logging in mechanism, the AuthenticationManager would check for compatible authentication providers out of all 3 authentication provider.
11. The selected authentication provider would be responsible to authenticate the logging in user.