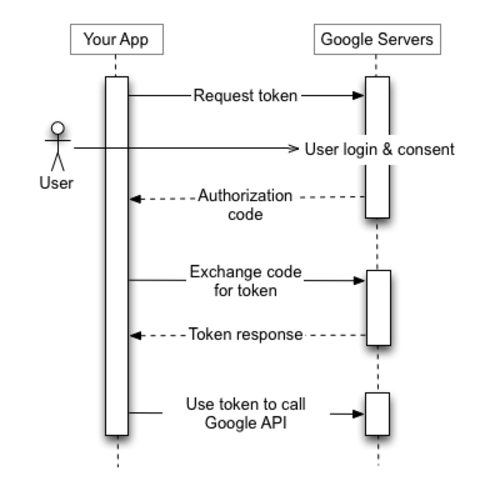
# Wat is OAuth?

OAuth is een protocol voor het inloggen bij een website. Sommige websites staan ook toe dat andere websites hun login gebruiken voor de autorisatie. OAuth is een veilige manier om data te delen met andere bedrijven, want je hoeft je wachtwoord niet te delen. Zo voorkomen gebruikers dat bedrijven hun wachtwoorden op een onveilige manier opslaan.



# Welke OAuth providers zijn er?

Veel bedrijven ondersteunen OAuth, waaronder:

* Google
* Github
* Microsoft
* Facebook
* Amazon
* Apple

Van deze OAuth providers gebruiken wij Google en Github. Deze zijn al standaard geïmplementeerd binnen Spring boot. De OAuth implementatie tussen deze providers is vrijwel hetzelfde. De data die je teruggestuurd krijgt verschilt wel, zoals bij GitHub is de username ‘login’ en bij Google is het ‘name’.

# Gebruikte tools

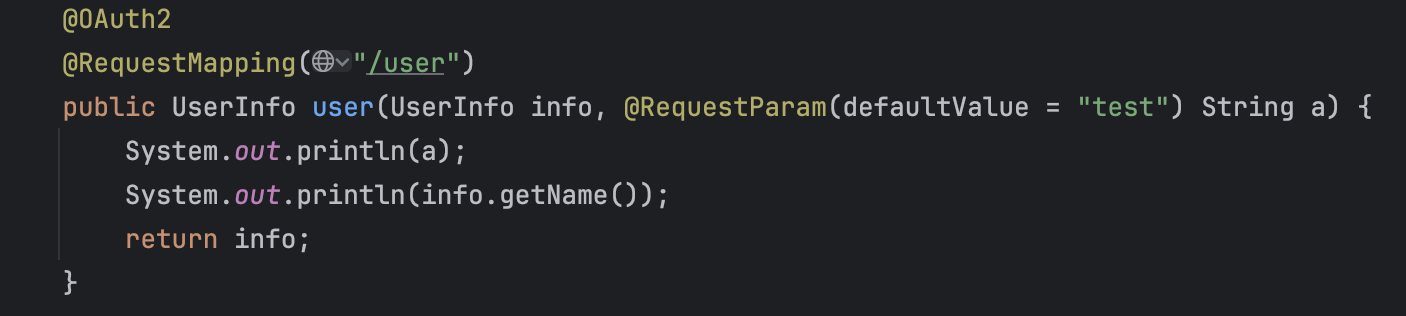
* Spring security
  + Handelt OAuth2 aanvragen af en geeft de informatie vanuit de services terug aan onze code.
  + Voordelen:
    - Is al ingebouwd in spring en is dus makkelijk toe te passen binnen onze stack.
    - Security experts hebben nagedacht over de veiligheid.
    - Scheelt tijd in implementatie.
  + Nadelen:
    - Je hebt geen controle over de flow
    - Je moet de dependency versie up-to-date houden, voor als één van de services hun API aanpast
    - Foutmeldingen zijn niet duidelijk
* Spring AOP
  + Zorgt ervoor dat de OAuth2 implementatie in controllers zo simpel mogelijk is met een annotatie.
  + Voordelen:
    - Is al ingebouwd in spring en is dus makkelijk toe te passen binnen onze stack.
    - Maakt het simpel voor ons om OAuth te implementeren binnen onze controllers, door er alleen een annotatie boven te zetten.
  + Nadelen:
    - De controller methodes worden hierdoor onderschept wat kan leiden tot een langzamere applicatie

# Implementatie en obstakels

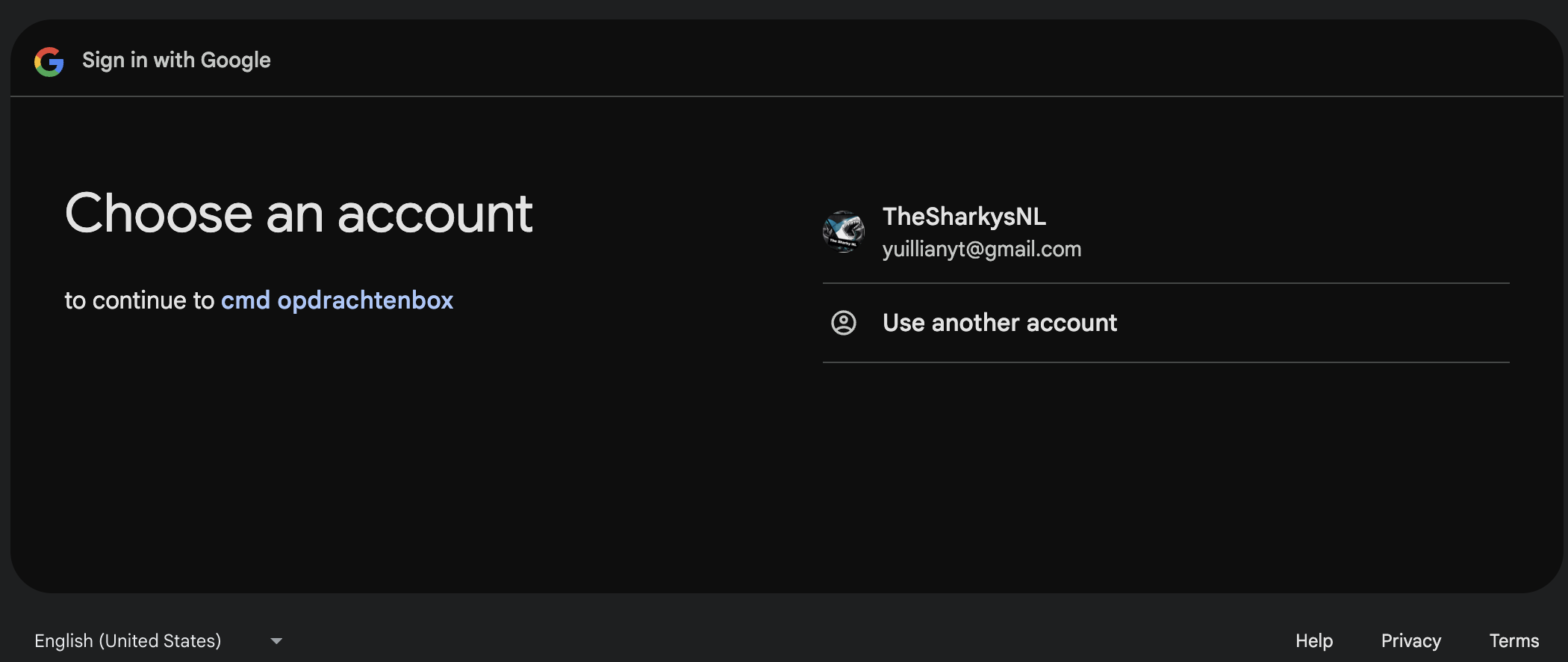
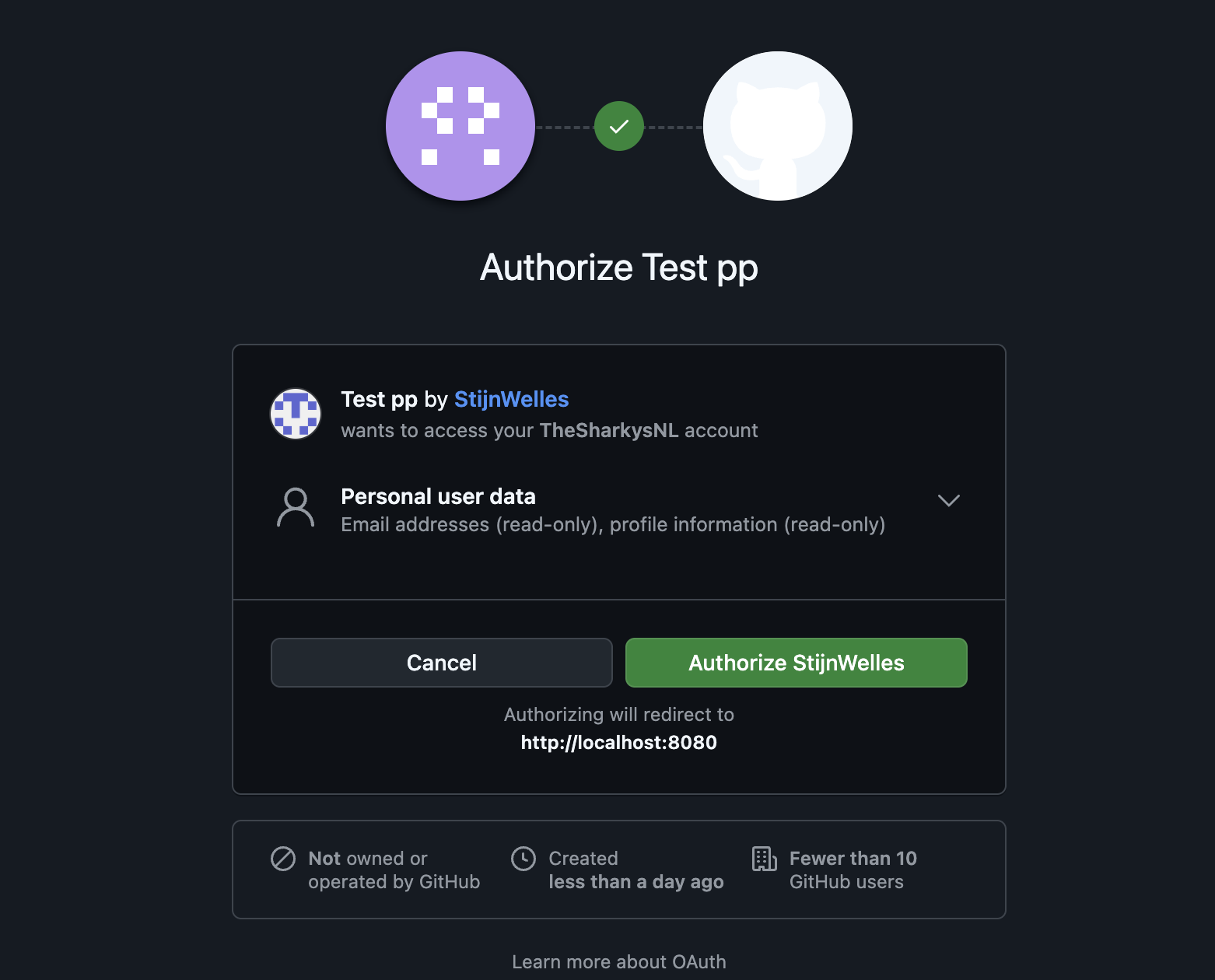
Wij als team verwachten dat het 6 uur zal duren om dit te implementeren. We hebben door de demo al een goed idee hoe de implementatie er uit zal zien, het grootste deel van de tijd zal dus gaan naar het opslaan van de user in de database en het implementeren in alle endpoints.

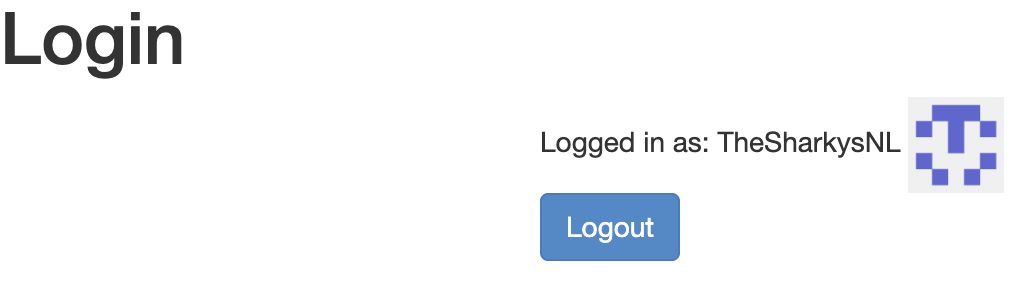
Als hier ook nog OAuth2 van een andere provider bij moet komen zal dit waarschijnlijk 48 uur duren en zal hier een nieuw onderzoek voor gestart moeten worden. Binnen dit onderzoek zal er gekeken worden hoe OAuth2 voor deze provider werkt, hoe dit te implementeren is binnen spring boot en of dit samen werkt met de standaard implementatie van spring boot. De dependency die we gebruiken voor OAuth2 ondersteunt geen andere providers en we zouden het dan handmatig moeten implementeren door middel van de flow die eerder in dit document is beschreven. Dit kost veel tijd.

# Demo



Dit is functie voor het ophalen van de gebruikers data uit de OAuth2. Hier wordt de zelfgemaakte OAuth2 annotatie gebruikt om de ‘UserInfo’ mee te geven aan de functie. Dit wordt dan teruggegeven aan de gebruiker.

Bij het klikken op de login knop wordt de gebruiker verstuurd naar de autorisatie pagina van GitHub of Google. Hier moet de gebruiker inloggen met zijn account.



Na het inloggen kan de gebruiker zien dat hij/zij is ingelogd. Op de backend wordt gezorgd dat Google en GitHub authenticatie hetzelfde verloopt.

Bronnen:

* <https://developers.google.com/identity/protocols/oauth2>
* <https://medium.com/@shubhkotnala/implementing-custom-annotation-in-springboot-b776eb2e77ae>
* <https://spring.io/guides/tutorials/spring-boot-oauth2>
* <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens>
* <https://en.wikipedia.org/wiki/List_of_OAuth_providers>