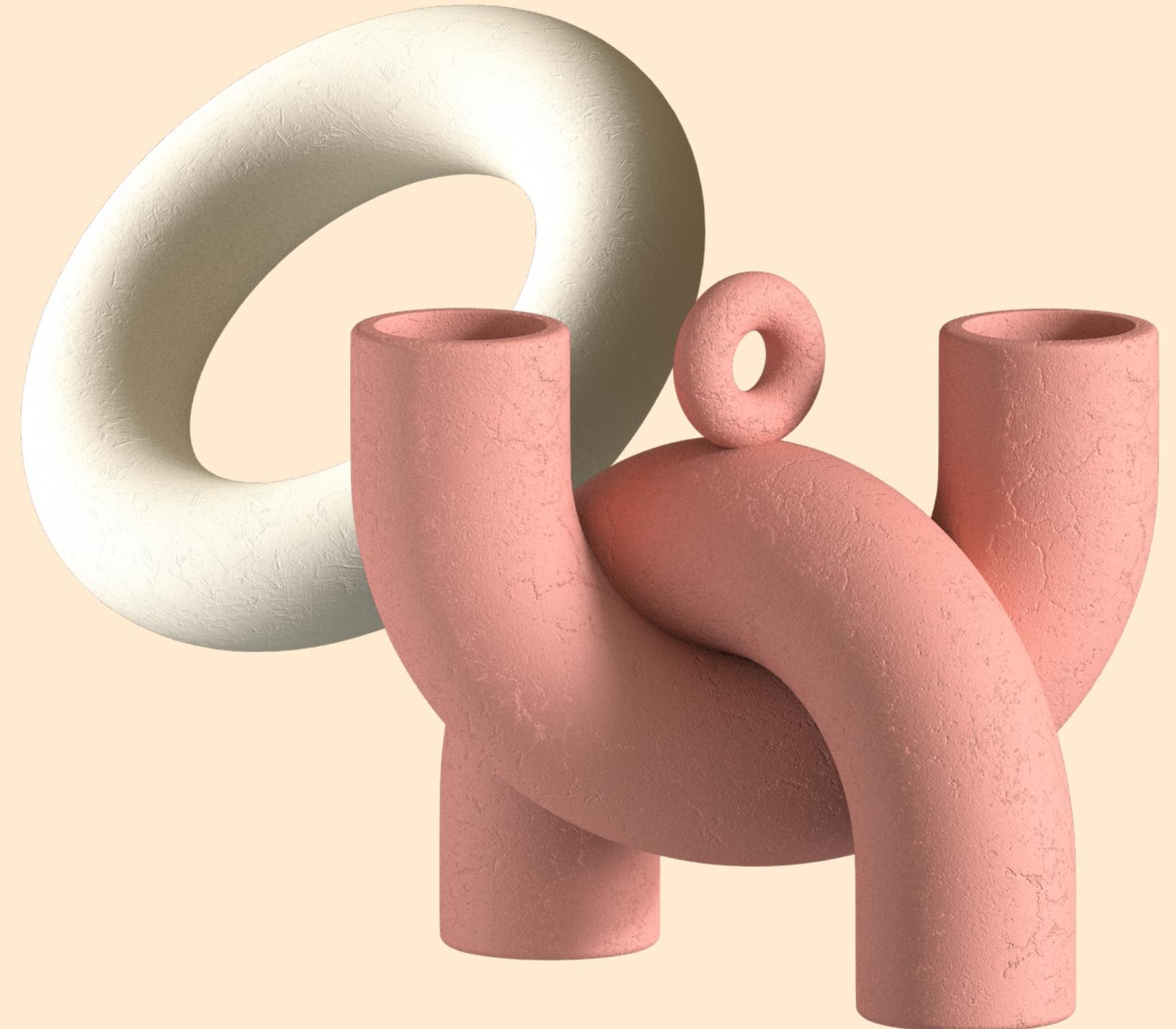


Multi Factor Authentication (Micro-okta)

TEAM : MA-216350

Authentication simplified



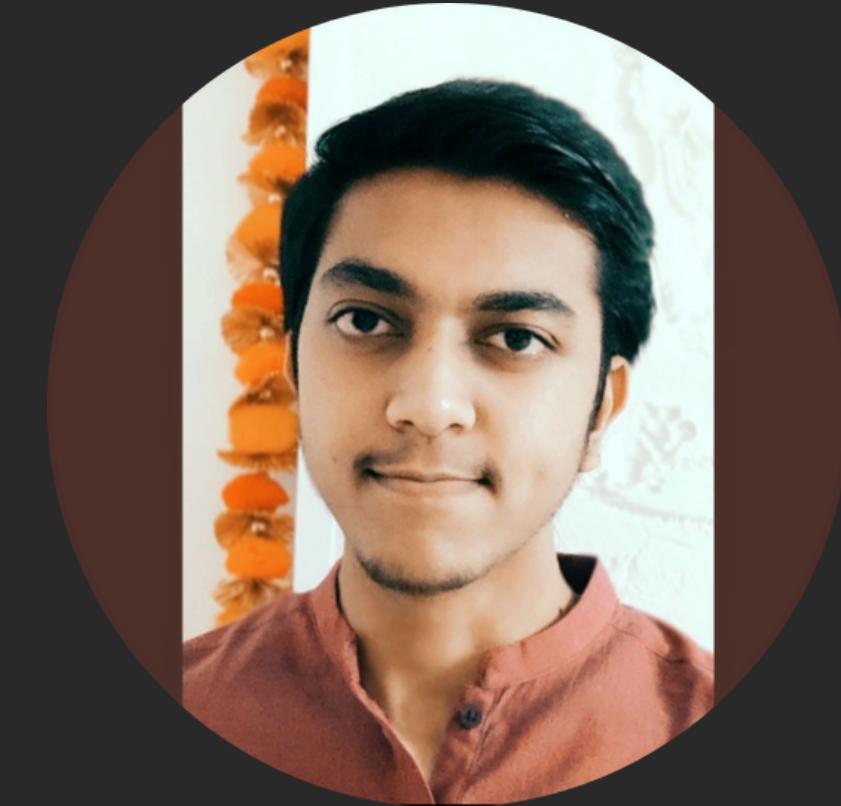
TEAM



MRITYUNJAY
VYAS



HARSH
SINGHVI



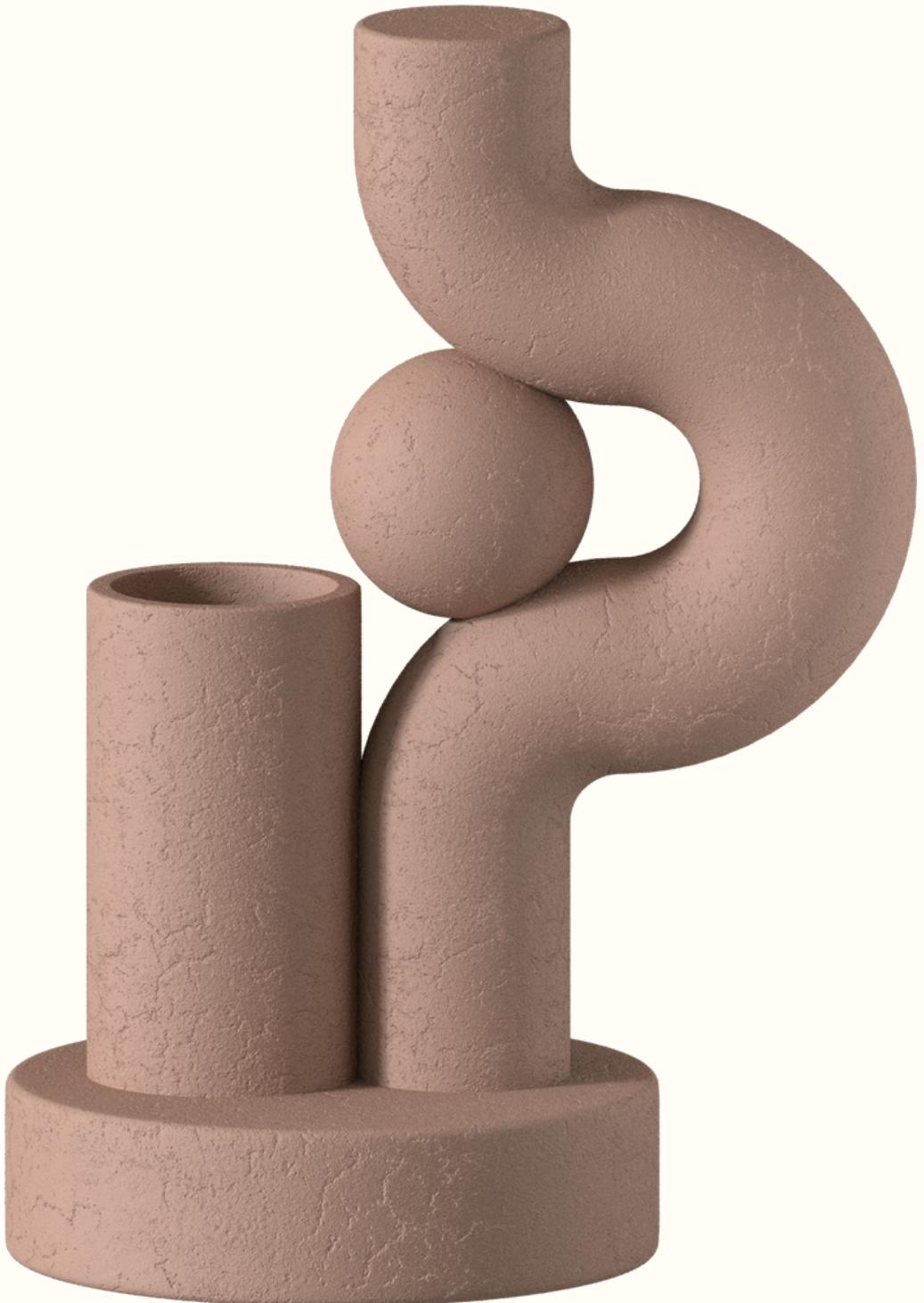
SHREYANSH
AGARWAL



GARVIT
CHOUHAN



Problem



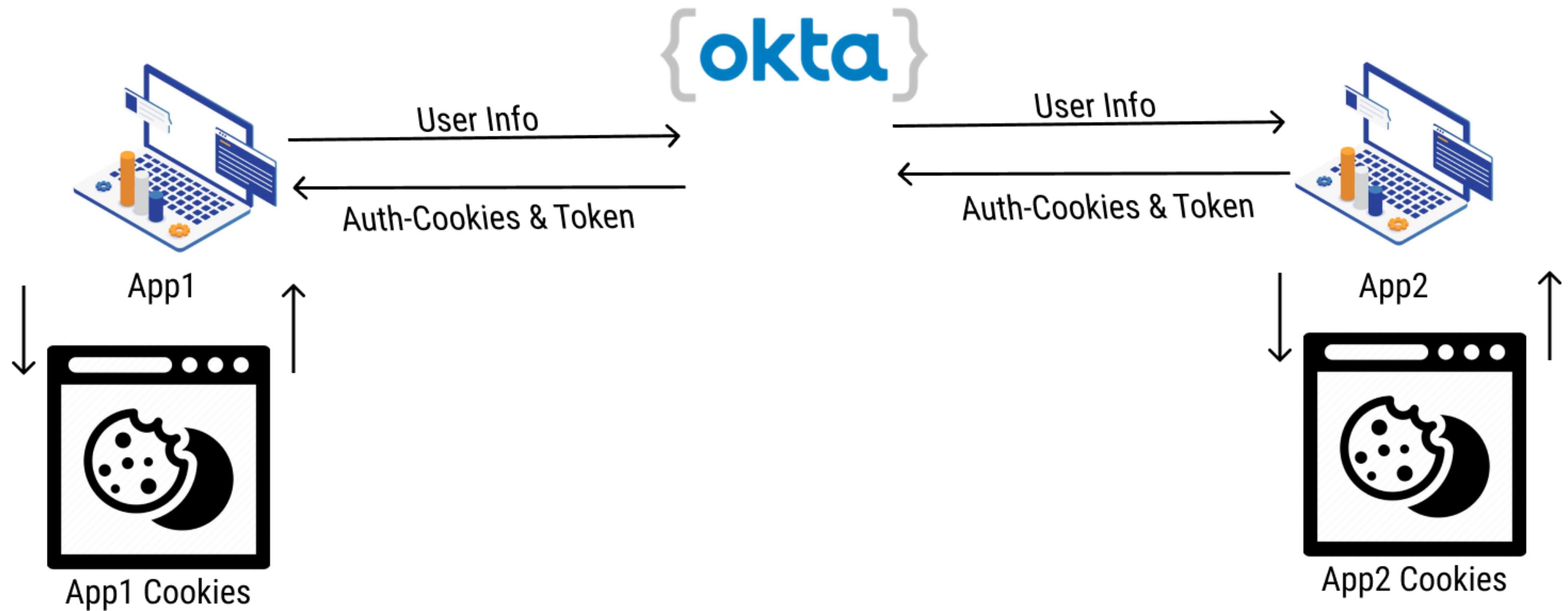
- a Organization with multiple web portals for its clients.
- This organization uses OKTA to verify users for authentication.
- Design a microservice that would talk to OKTA and return a valid session to the calling application.
- Common Login should persist over those web portal

Micro-Okta

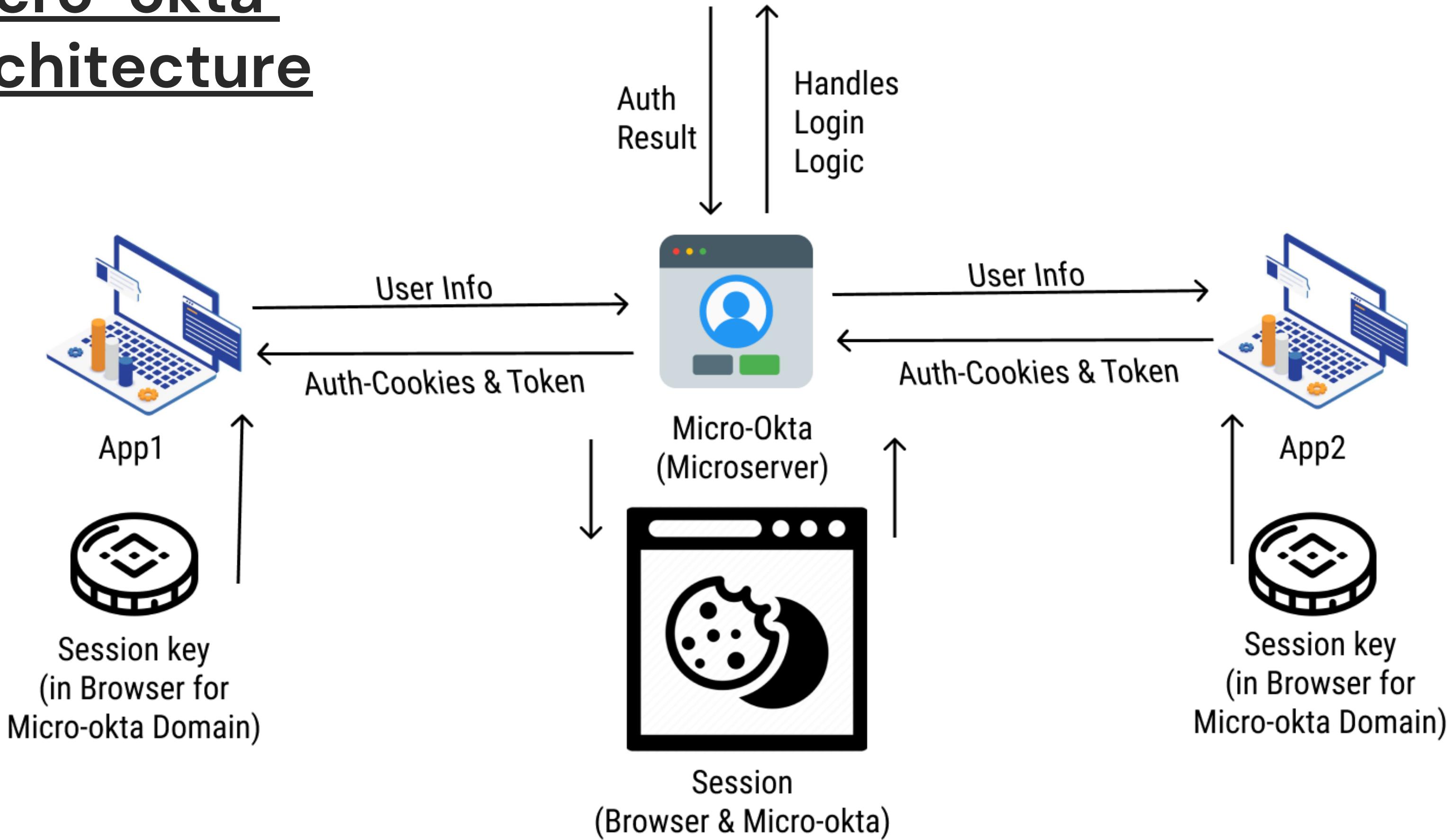
Unified Authentication Micro service for increasing scalability, efficiency of rolling out updates & Maintaining multiple web portal's Authentication System.



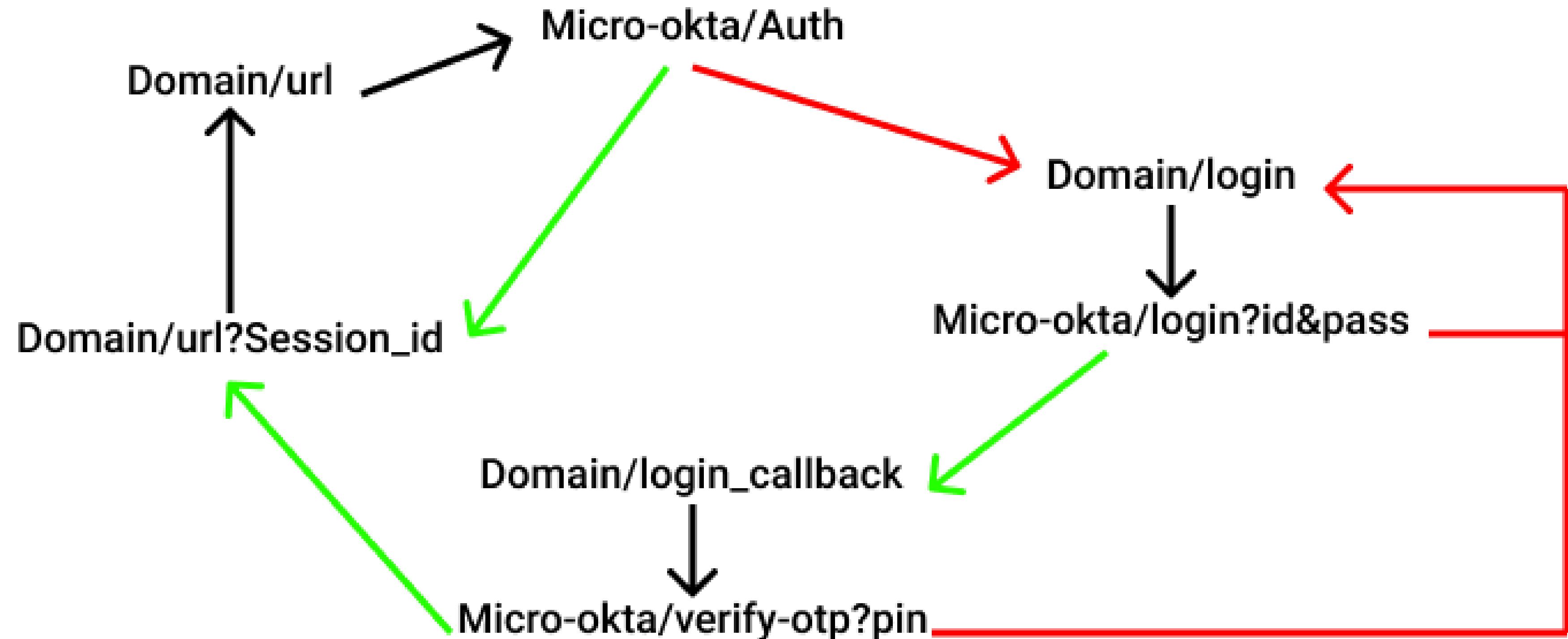
Existing okta Architecture



Micro-okta Architecture



Overall Flow



Integrating Micro-Okta with your web app

App Should Have Two login routes

/login

- Takes email & password from user.
- Redirects to micro-okta.herokuapp.com/login?id={email}&pass={password}

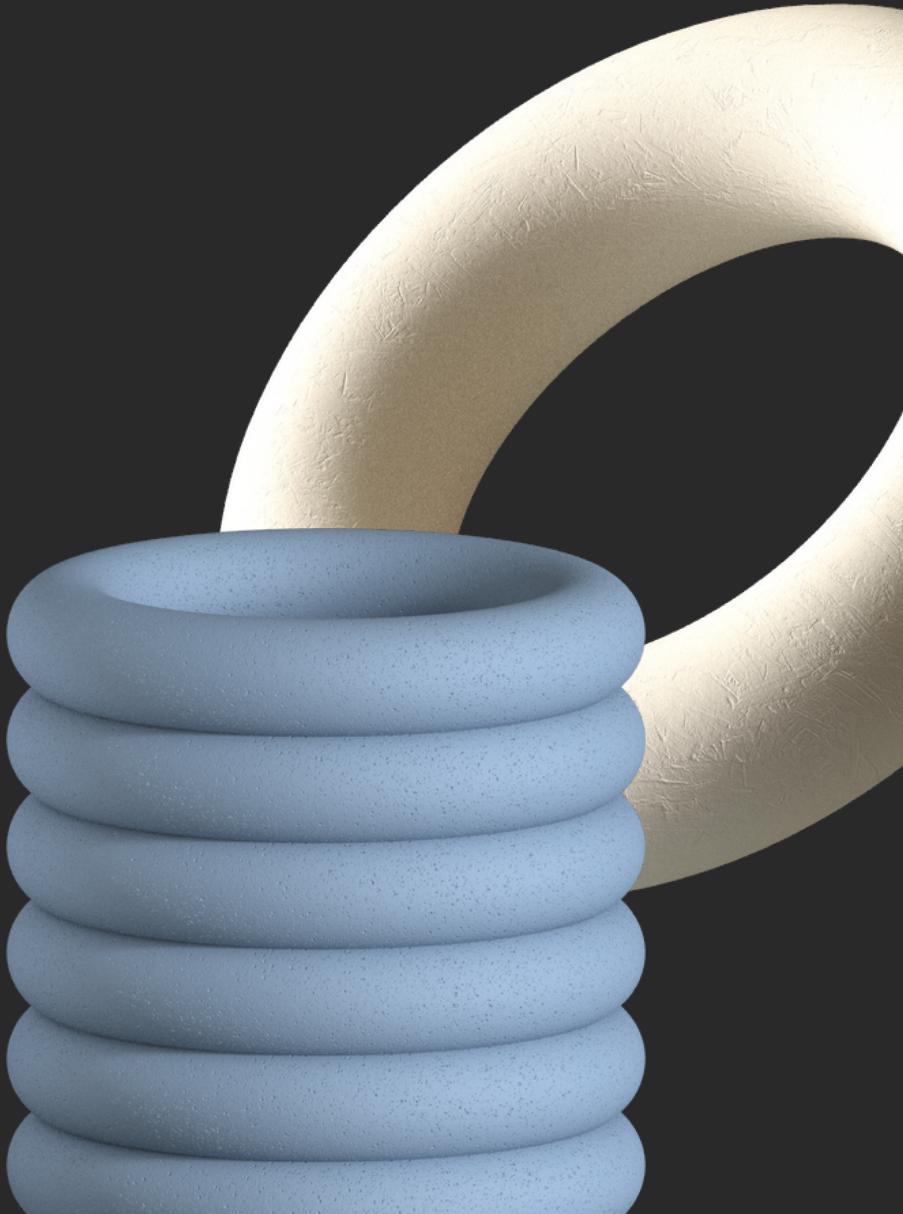
/login_callback

- Takes OTP from user.
- Redirects to micro-okta.herokuapp.com/verify-otp?pin={OTP}

Every Protected Page Should have these Scripts in Head

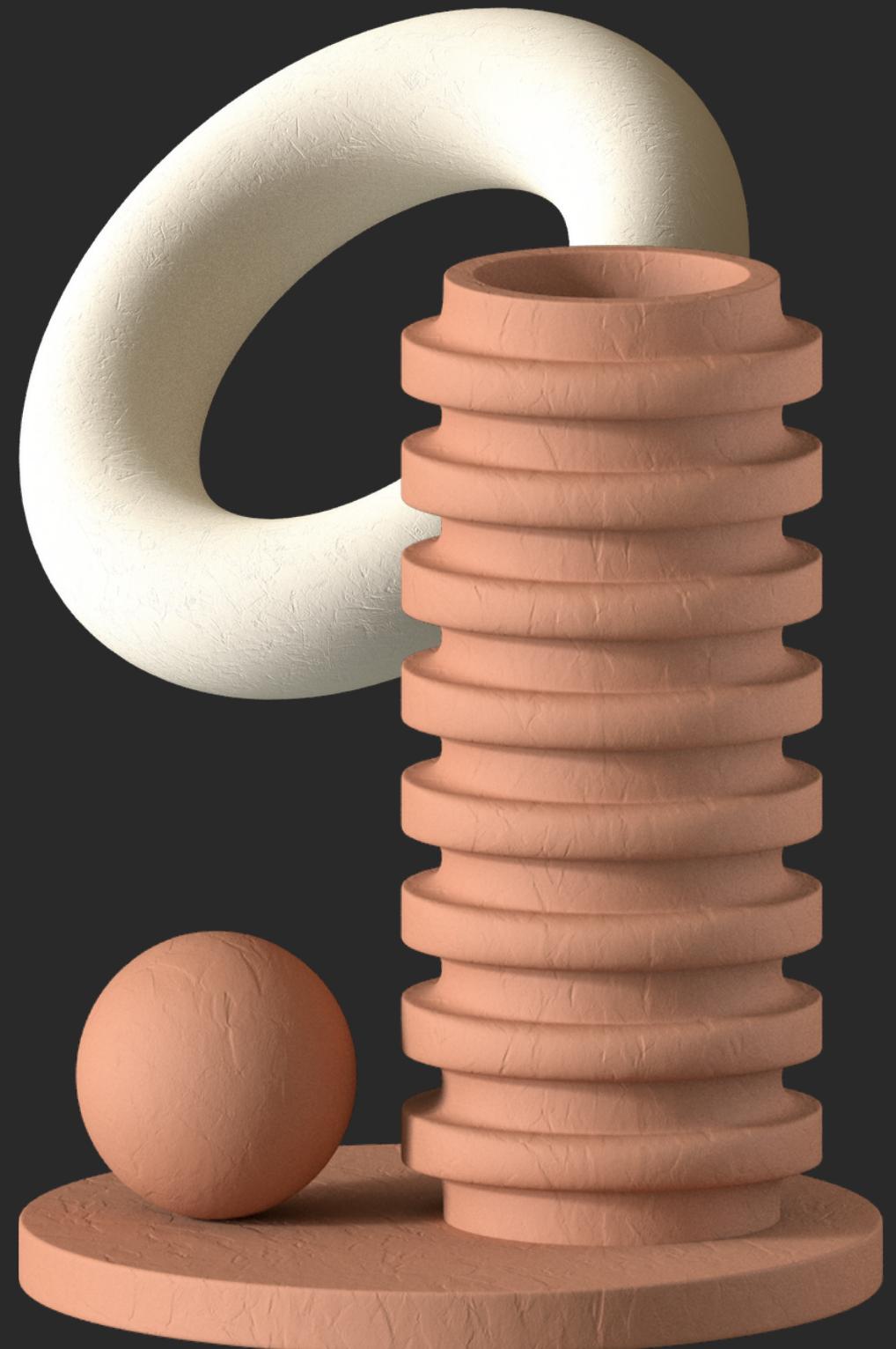
<https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js>

<https://cdn.jsdelivr.net/gh/MFA9000/micro-okta/protect.js>



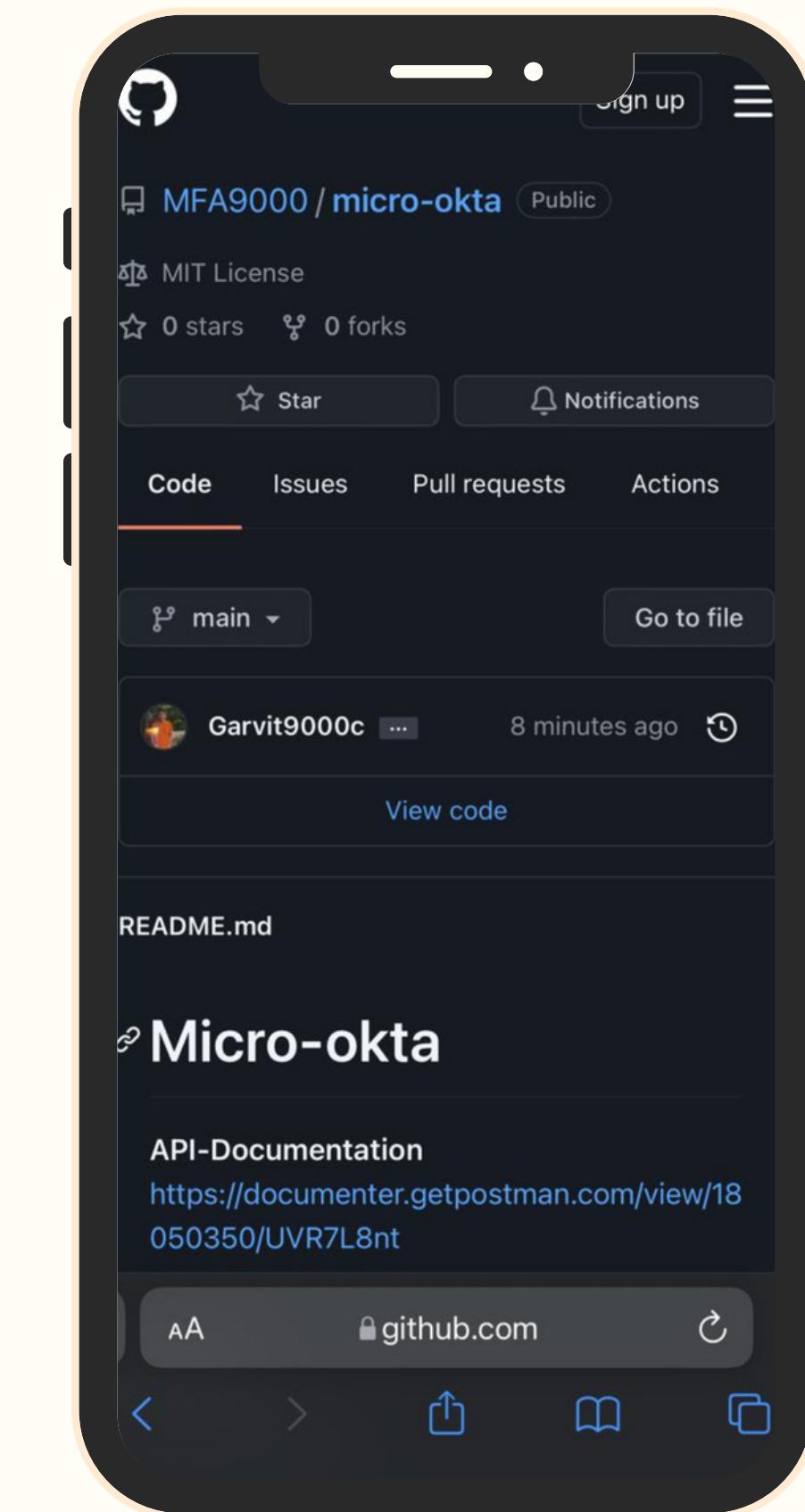
Documentation

<https://documenter.getpostman.com/view/18050350/UVR7L8nt>



Github-Repo

[https://github.com
/MFA9000/micro-
okta](https://github.com/MFA9000/micro-okta)



Demo-app

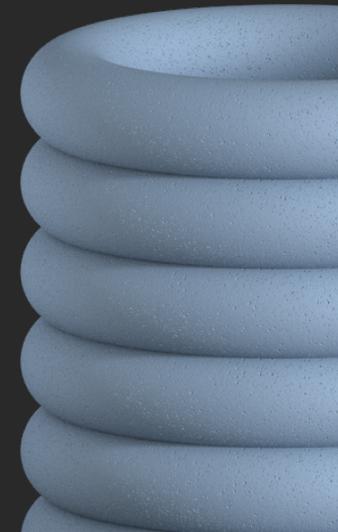


App1 - Flask

<https://demoplix1.herokuapp.com/>

App2 - Node/express

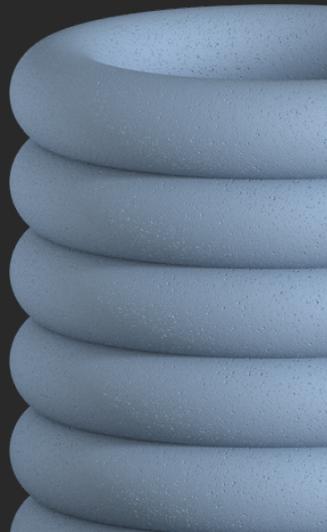
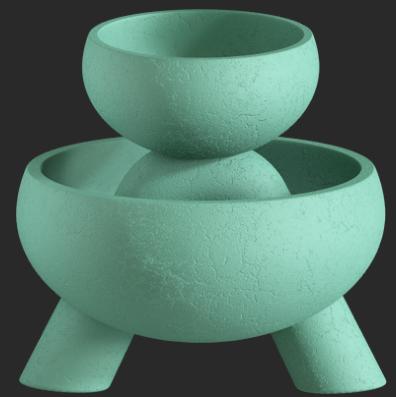
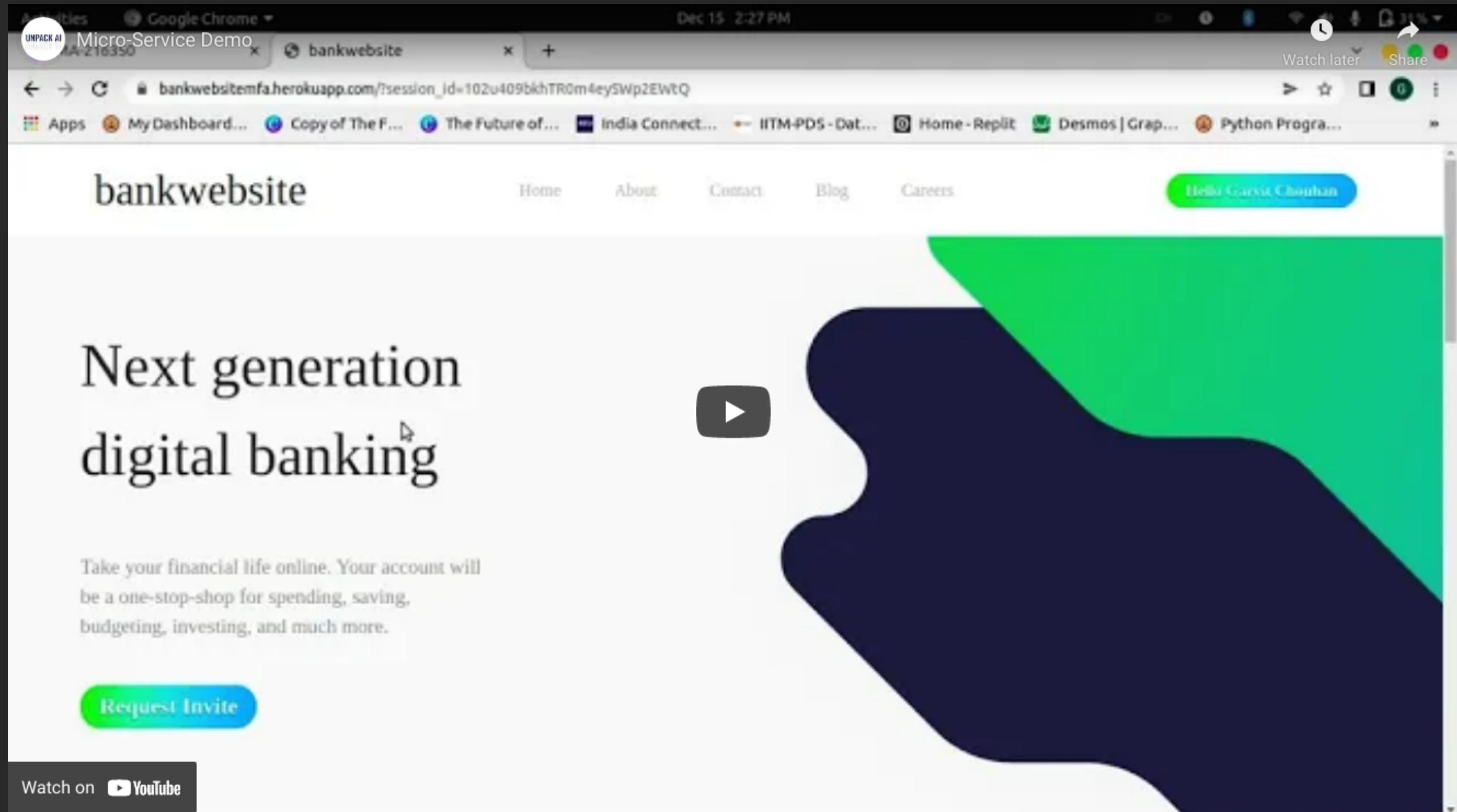
<https://bankwebsitemfa.herokuapp.com/>



Tech-Stack

- Python/Flask
- Node/ExpressJS
- Okta

Demo-Video



Thank You

