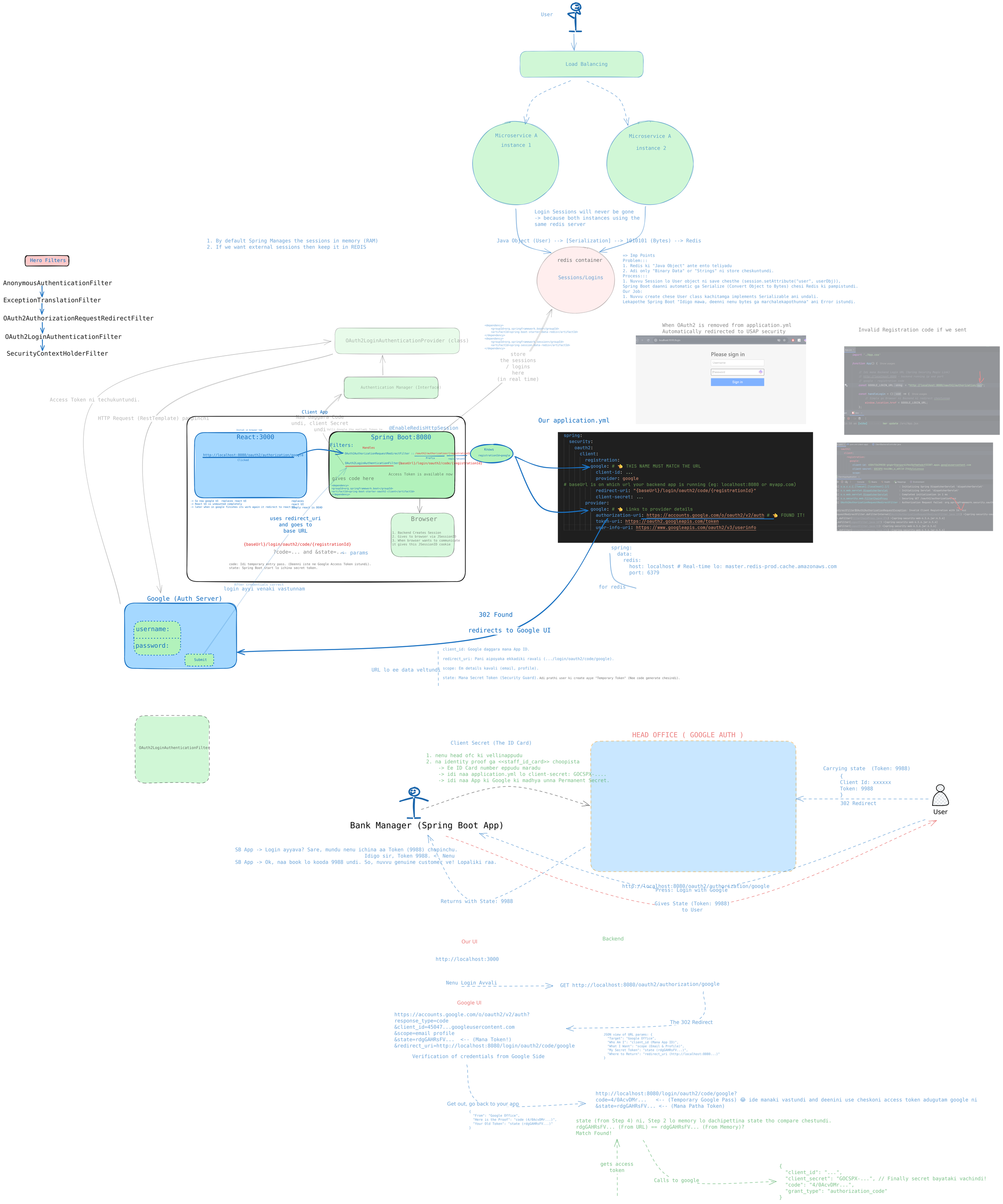
User
Load Balancing
Microservice A instance 1
Microservice A instance 2
Login Sessions will never be gone
-> because both instances using the same redis server
1. By default Spring Manages the sessions in memory (RAM)
2. If we want external sessions then keep it in REDIS
Java Object (User) --> [Serialization] --> 1010101 (Bytes) --> Redis
redis container
Sessions/Logins
=> Imp Points
Problem:::
1. Redis ki "Java Object" ante ento teliyadu
2. Adi only "Binary Data" or "Strings" ni store cheskuntundi.
Process:::
1. Nuvvu Session lo User object ni save chesthe (session.setAttribute("user", userObj)),
Spring Boot daanni automatic ga Serialize (Convert Object to Bytes) chesi Redis ki pampistundi.
Our Job:
1. Nuvvu create chese User class kachitamga implements Serializable ani undali.
Lekapothe Spring Boot "Idigo mawa, deenni nenu bytes ga marchalekapothunna" ani Error istundi.
Hero Filters
AnonymousAuthenticationFilter
ExceptionTranslationFilter
OAuth2AuthorizationRequestRedirectFilter
OAuth2LoginAuthenticationFilter
SecurityContextHolderFilter
OAuth2LoginAuthenticationProvider (class)
Authentication Manager (Interface)
store the sessions / logins here (in real time)
When OAuth2 is removed from application.yml
Automatically redirected to USAP security
Please sign in
Sign in
Invalid Registration code if we sent
Access Token ni techukuntundi.
HTTP Request (RestTemplate) pampinchi
Client App
Naa daggara Code undi, client Secret undi
React:3000
http://localhost:8080/oauth2/authorization/google
Clicked
Spring Boot:8080
@EnableRedisHttpSession
Filters:
OAuth2AuthorizationRequestRedirectFilter
OAuth2LoginAuthenticationFilter{baseUrl}/login/oauth2/code/{registrationId}
Access Token is available now
gives code here
Knows registrationId=google
Our application.yml
spring:
security:
oauth2:
client:
registration:
google: # 👈 THIS NAME MUST MATCH THE URL
client-id: ...
provider: google
baseUrl is on which url your backend app is running (eg: localhost:8080 or myapp.com)
redirect-uri: "{baseUrl}/login/oauth2/code/{registrationId}"
client-secret: ...
provider:
google: # 👈 Links to provider details
authorization-uri: https://accounts.google.com/o/oauth2/v2/auth # 👈 FOUND IT!
token-uri: https://oauth2.googleapis.com/token
user-info-uri: https://www.googleapis.com/oauth2/v3/userinfo
Browser
1. Backend Creates Session
2. Gives to browser via JSessionID
3. When browser wants to communicate it gives this JSessionID cookie
uses redirect_uri and goes to base URL
{baseUrl}/login/oauth2/code/{registrationId}
?code=... and &state=... <- params
spring:
data:
redis:
host: localhost # Real-time lo: master.redis-prod.cache.amazonaws.com
port: 6379
for redis
Google (Auth Server)
username:
password:
Submit
302 Found
redirects to Google UI
client_id: Google daggara mana App ID.
redirect_uri: Pani aipoyaka ekkadiki ravali (.../login/oauth2/code/google).
scope: Em details kavali (email, profile).
state: Mana Secret Token (Security Guard).
URL lo ee data veltundi
OAuth2LoginAuthenticationFilter
HEAD OFFICE (GOOGLE AUTH)
Client Secret (The ID Card)
1. nenu head ofc ki vellinappudu
2. na identity proof ga <<staff_id_card>> choopista
-> Ee ID Card number eppudu maradu
-> idi naa application.yml lo client-secret: GOCSPX-....
-> idi naa App ki Google ki madhya unna Permanent Secret.
Carrying state (Token: 9988)
{
Client Id: xxxxxx
Token: 9988
}
302 Redirect
User
Bank Manager (Spring Boot App)
SB App -> Login ayyava? Sare, mundu nenu ichina aa Token (9988) chupinchu.
Idigo sir, Token 9988. <- Nenu
SB App -> Ok, naa book lo kooda 9988 undi. So, nuvvu genuine customer ve! Lopaliki raa.
Returns with State: 9988
http://localhost:8080/oauth2/authorization/google
Press: Login with Google
Gives State (Token: 9988) to User
Our UI
http://localhost:3000
Backend
Nenu Login Avvali
GET http://localhost:8080/oauth2/authorization/google
Google UI
https://accounts.google.com/o/oauth2/v2/auth?
response_type=code
&client_id=45047...googleusercontent.com
&scope=email profile
&state=rdgGAHRsFV... <-- (Mana Token!)
&redirect_uri=http://localhost:8080/login/oauth2/code/google
The 302 Redirect
Verification of credentials from Google Side
Get out, go back to your app
http://localhost:8080/login/oauth2/code/google?
code=4/0AcvDMr... <-- (Temporary Google Pass) 🤔 ide manaki vastundi and deenini use cheskoni access token adugutam google ni
&state=rdgGAHRsFV... <-- (Mana Patha Token)
state (from Step 4) ni, Step 2 lo memory lo dachipettina state tho compare chestundi.
rdgGAHRsFV... (From URL) == rdgGAHRsFV... (From Memory)?
Match Found!
gets access token
Calls to google
"client_id": "...",
"client_secret": "GOCSPX-...", // Finally secret bayataki vachindi!
"code": "4/0AcvDMr...",
"grant_type": "authorization_code"