DEVSU TECHNICAL CHALLENGE
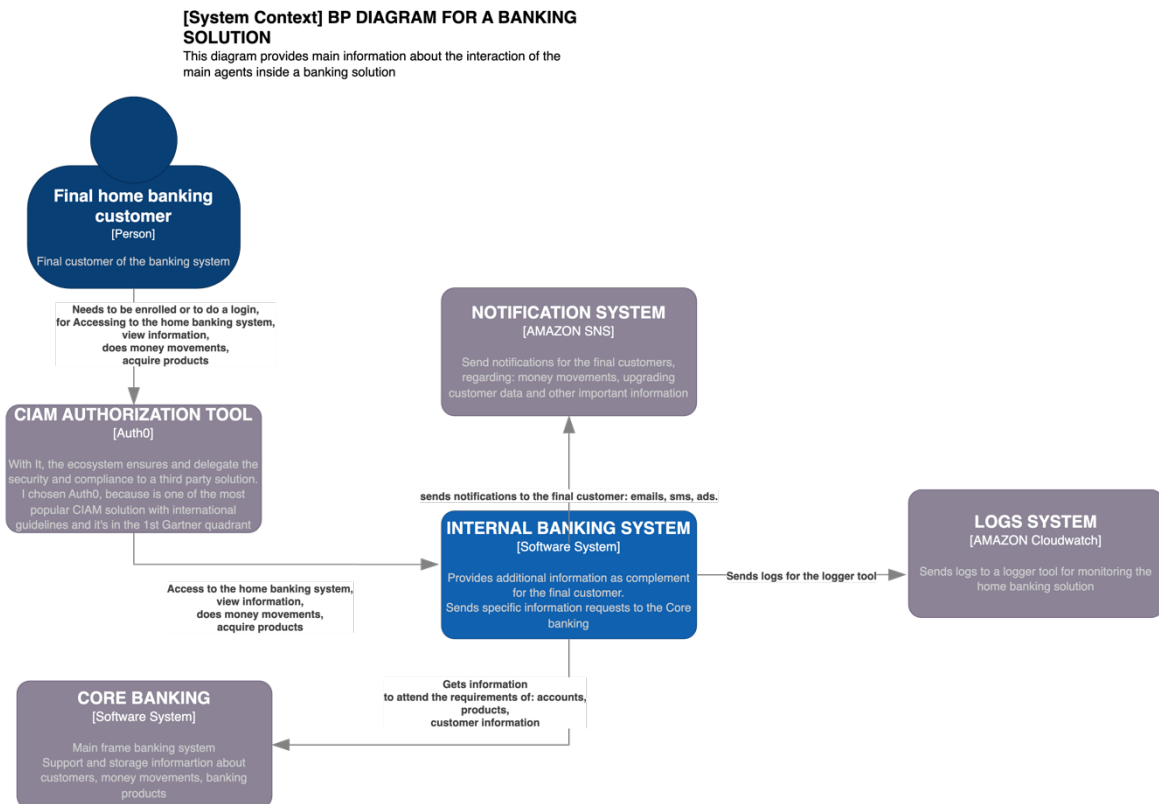By Isaac Coronado

DESCRIPTION OF THE SOLUTION

MODEL SOLUTION

In this aspect I create 3 types of diagram:

Context diagram:
This diagram has 4 systems that have interaction with the final customer
- Internal banking system
- Notification System
- CIAM Authorization tool
- Core banking
- Log system

[System Context] BP DIAGRAM FOR A BANKING SOLUTION
This diagram provides main information about the interaction of the main agents inside a banking solution
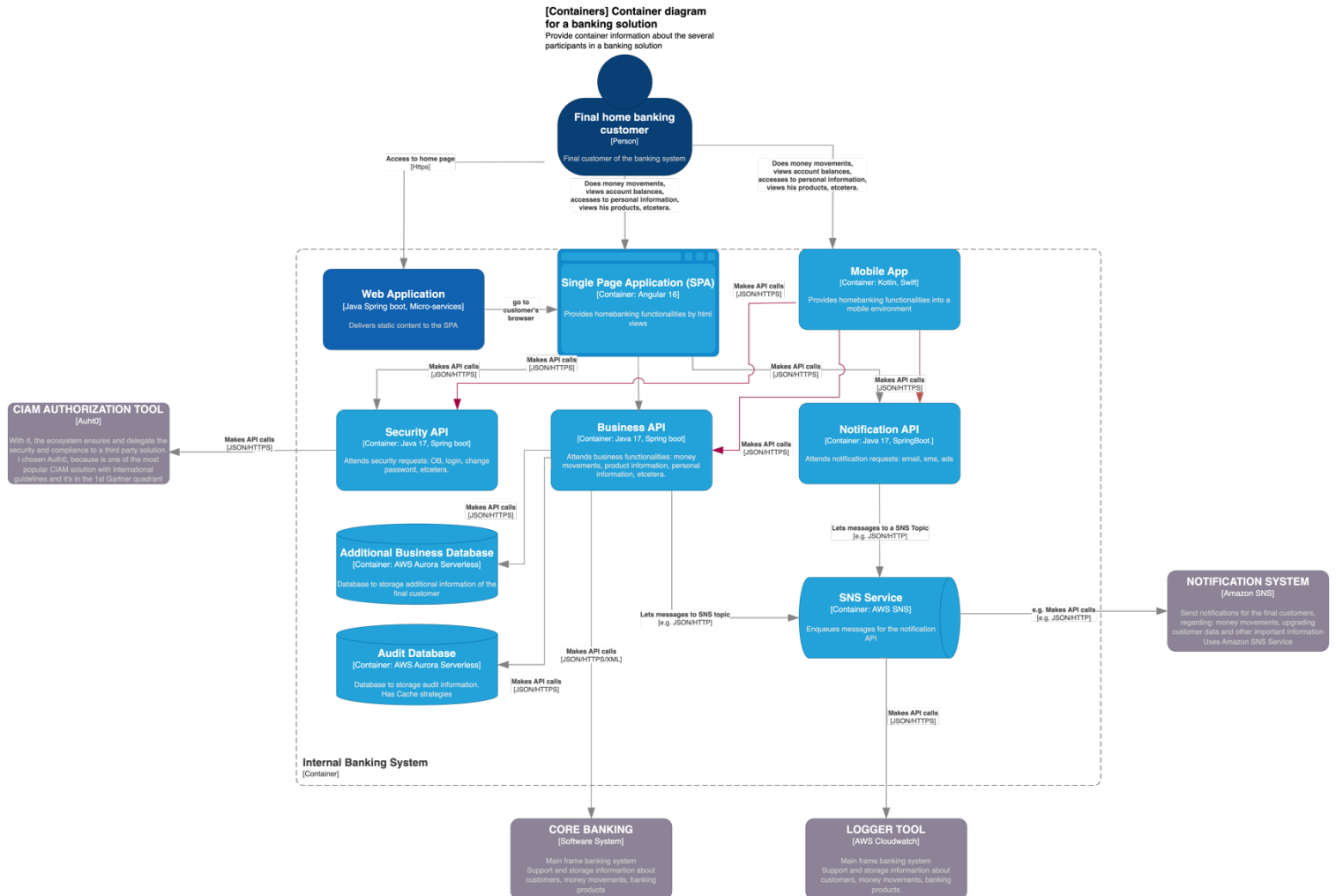
Container Diagram
This diagram has the several artifacts to interact into the banking environment, such us:
- Web Application: Serves static content.
- Single Page Application
- Mobile App
- I have 3 Apis:
    o Security API: Has the responsibility to interact with Auht0 solution

- o Business API: Has the responsibility to interact with the Core banking and Databases that not are part of the Core solution.
- o Notification Api: Has the responsibility to sends Notifications to the final customers, uses Amazon SNS to comply this.

[Containers] Container diagram for a banking solution — Provide container information about the several participants in a banking solution
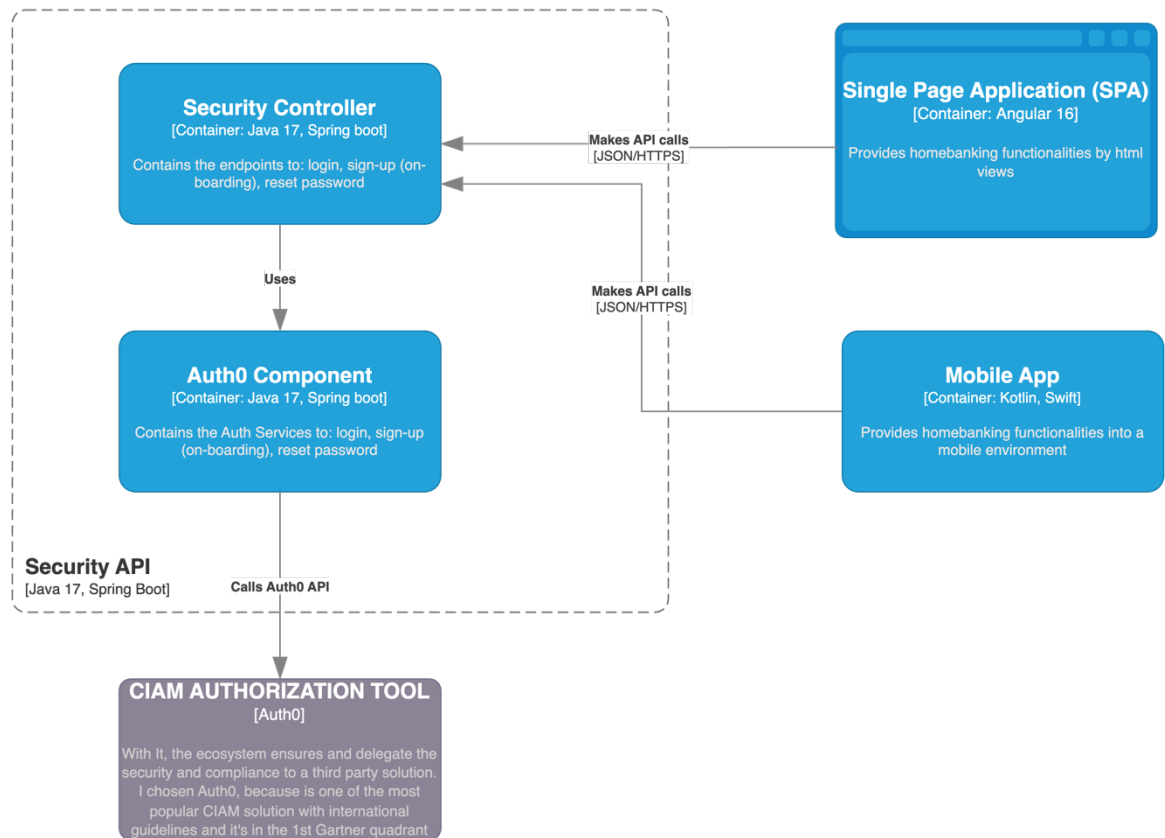
Component Diagrams

Regarding this point I did 3 diagrams to have more order and to understand better the reason and responsibility of each one.
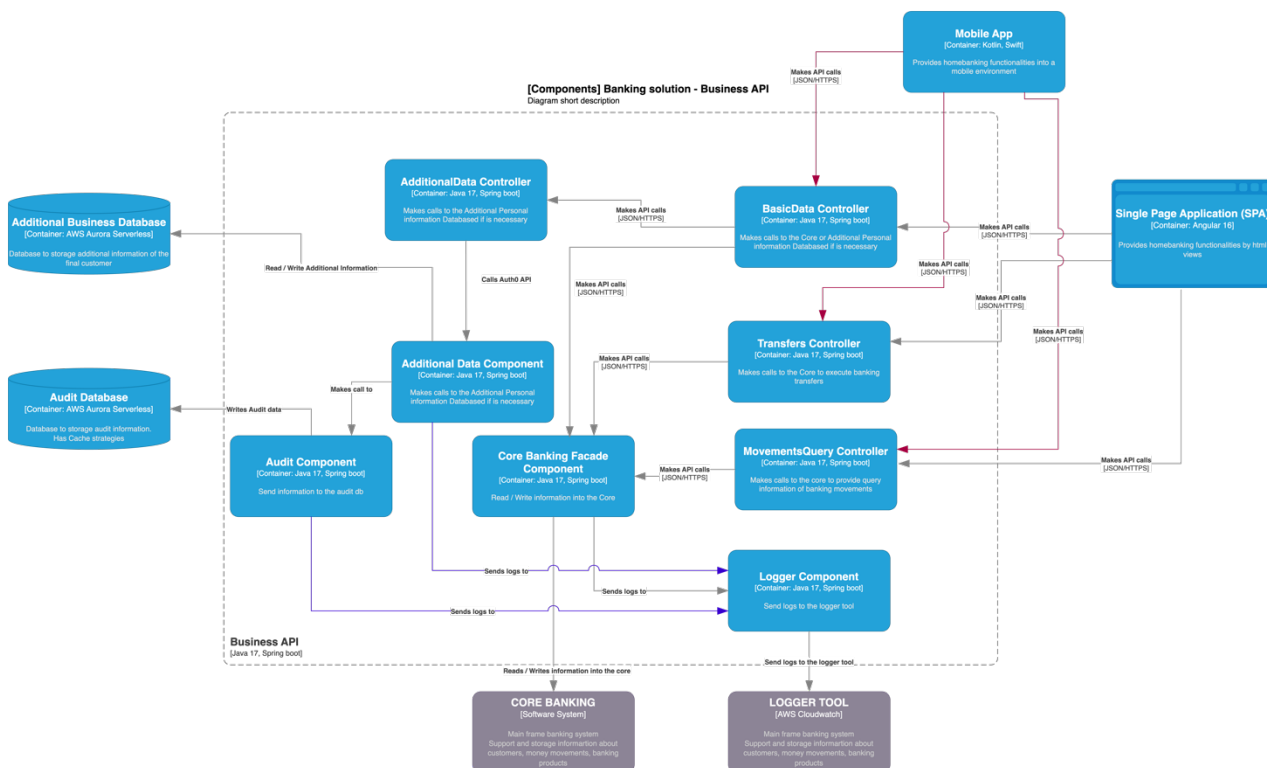
- - Security API: Shows the interaction with Auth0 Tool.
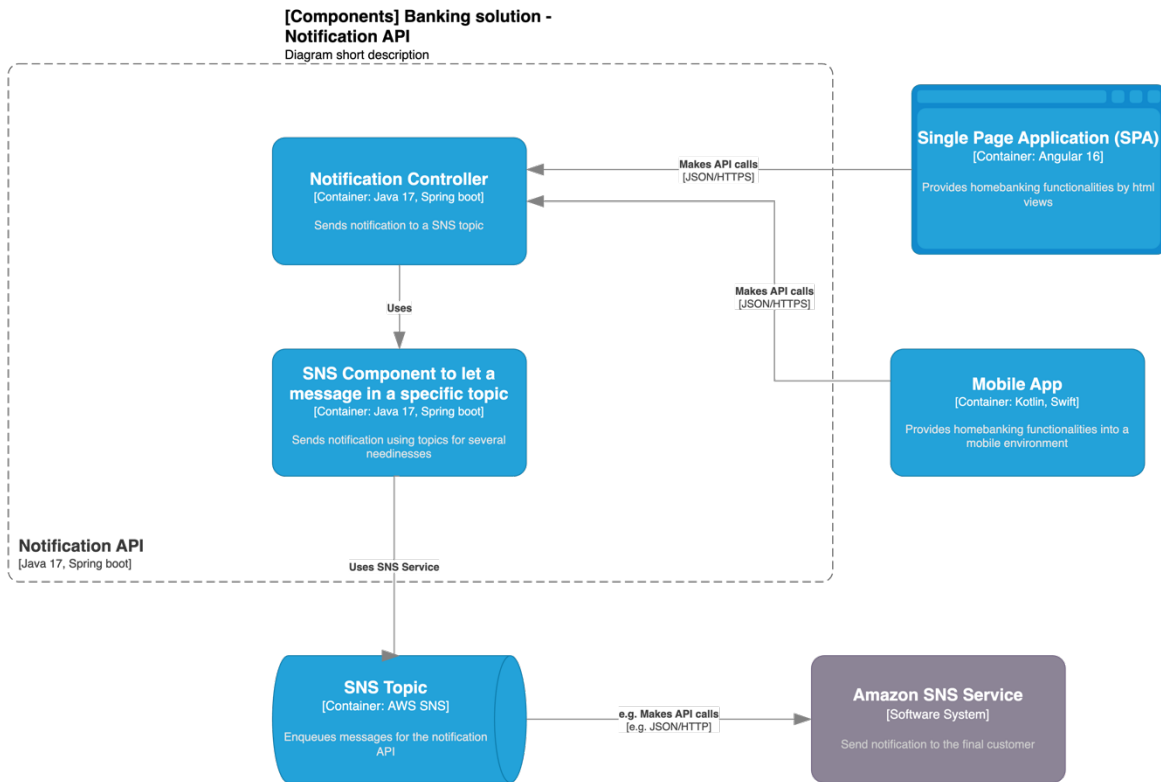
[Components] Banking Solution - Security API
Diagram short description

Security Controller
[Container: Java 17, Spring boot]

Contains the endpoints to: login, sign-up (on-boarding), reset password

Makes API calls
[JSON/HTTPS]

Single Page Application (SPA)
[Container: Angular 16]

Provides homebanking functionalities by html views

Uses

Makes API calls
[JSON/HTTPS]

Auth0 Component
[Container: Java 17, Spring boot]

Contains the Auth Services to: login, sign-up (on-boarding), reset password

Mobile App
[Container: Kotlin, Swift]

Provides homebanking functionalities into a mobile environment

Security API
[Java 17, Spring Boot]

Calls Auth0 API

CIAM AUTHORIZATION TOOL
[Auth0]

With It, the ecosystem ensures and delegate the security and compliance to a third party solution. I chosen Auth0, because is one of the most popular CIAM solution with international guidelines and it's in the 1st Gartner quadrant

- Business API: Shows the interaction of Business Components.

Mobile App
[Container: Kotlin, Swift]
Provides homebanking functionalities into a mobile environment

[Components] Banking solution - Business API
Diagram short description

Makes API calls
[JSON/HTTPS]

AdditionalData Controller
[Container: Java 17, Spring boot]
Makes calls to the Additional Personal information Databased if is necessary

Makes API calls
[JSON/HTTPS]

BasicData Controller
[Container: Java 17, Spring boot]
Makes calls to the Core or Additional Personal information Databased if is necessary

Makes API calls
[JSON/HTTPS]

Single Page Application (SPA)
[Container: Angular 16]
Provides homebanking functionalities by html views

Additional Business Database
[Container: AWS Aurora Serverless]
Database to storage additional information of the final customer

Read / Write Additional Information

Calls Auth0 API

Makes API calls
[JSON/HTTPS]

Makes API calls
[JSON/HTTPS]

Makes API calls
[JSON/HTTPS]

Additional Data Component
[Container: Java 17, Spring boot]
Makes calls to the Additional Personal information Databased if is necessary

Makes API calls
[JSON/HTTPS]

Transfers Controller
[Container: Java 17, Spring boot]
Makes calls to the Core to execute banking transfers

Makes API calls
[JSON/HTTPS]

Audit Database
[Container: AWS Aurora Serverless]
Database to storage audit information. Has Cache strategies

Makes call to

Writes Audit data

Audit Component
[Container: Java 17, Spring boot]
Send information to the audit db

Core Banking Facade Component
[Container: Java 17, Spring boot]
Read / Write information into the Core

Makes API calls
[JSON/HTTPS]

MovementsQuery Controller
[Container: Java 17, Spring boot]
Makes calls to the core to provide query information of banking movements

Makes API calls
[JSON/HTTPS]

Sends logs to

Sends logs to

Sends logs to

Logger Component
[Container: Java 17, Spring boot]
Send logs to the logger tool

Business API
[Java 17, Spring boot]

Reads / Writes information into the core

Send logs to the logger tool

CORE BANKING
[Software System]
Main frame banking system Support and storage information about customers, money movements, banking products

LOGGER TOOL
[AWS Cloudwatch]
Main frame banking system Support and storage information about customers, money movements, banking products

- Notification API: Shows the interaction of Notification Components.

[Components] Banking solution - Notification API
Diagram short description

Notification Controller
[Container: Java 17, Spring boot]

Sends notification to a SNS topic

Single Page Application (SPA)
[Container: Angular 16]

Provides homebanking functionalities by html views

Makes API calls
[JSON/HTTPS]

Uses

Makes API calls
[JSON/HTTPS]

SNS Component to let a message in a specific topic
[Container: Java 17, Spring boot]

Sends notification using topics for several needinesses

Mobile App
[Container: Kotlin, Swift]

Provides homebanking functionalities into a mobile environment

Notification API
[Java 17, Spring boot]

Uses SNS Service

SNS Topic
[Container: AWS SNS]

Enqueues messages for the notification API

e.g. Makes API calls
[e.g. JSON/HTTP]

Amazon SNS Service
[Software System]

Send notification to the final customer

INFRAESTRUCTURE DIAGRAM

I used AWS, because I have more knowledge and experience working with their services.
I have 2 knowledge programs with AWS: Aws Cloud Technical Essentials, Aw cloud solutions architect
The diagram has notes that describe the services that I used.
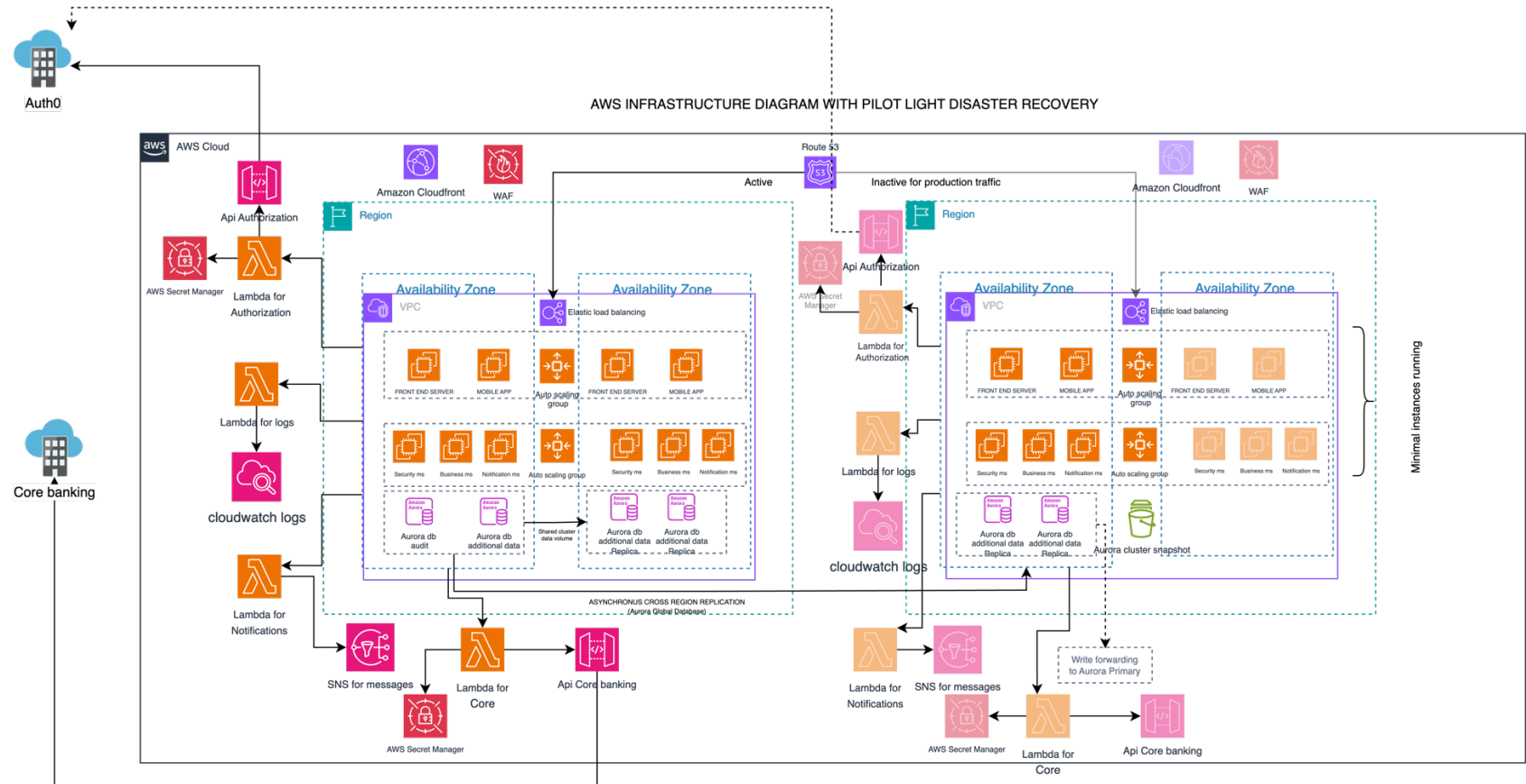
Notes
- For disaster recovery I chosen Pilot Light.
It's a disaster recover strategy with low cost.
- Region of the right side is inactive, It's part of the DR strategy
- I use Aws secret Manager with the flows that have interactions with third party solutions (Auth0, Core banking)
- Regarding notifications services, I use Amazon SNS, with DR Strategy and for It,
isn't necessary to have another notification service.
- For logs I use Amazon Cloudwatch
- For Authentication and Authorization, I used Auth0
 that has several features: On-boarding, biometric features, MFA, Cryptographic features, Push Notification, Soft Token, OAuth 2.0, etcetera.

Annexes

- [Link to Google drive](#) that has the diagrams part of this documents to have more visibility.
- [Link to Github repository](#) that has this PDF. Document.