CODING CHALLENGE

NODEJS BACKEND CHALLENGE

JANUARY 2021

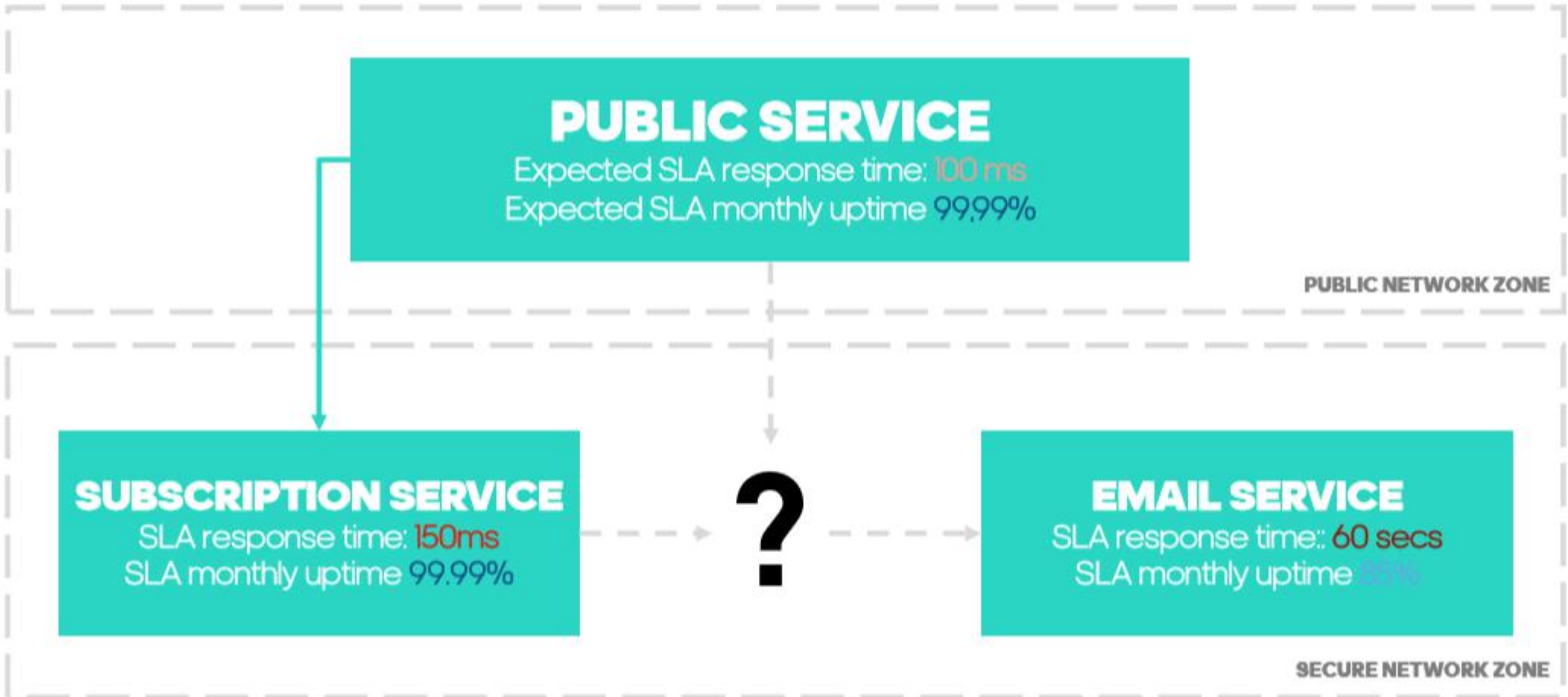
Architecture Principles

At adidas, we care about serving consumers (in the sense of applications making use of our data and services) by serving more (high volume), serving better (low latency) or serving more reliably (high resilience). In general, High Availability is very important, so designed solutions need to be scalable and resilient.

In a nutshell, the basic architecture principles are:

- Architect solutions in a way that makes them re-usable
- Components are designed to scale horizontally based on incoming load
- Always care about data & IT Security
- All solutions treat core operational database as a costly resource that needs to be used economically
- Design for failure, your solution has to assume and handle exception conditions

ARCHITECTURE LANDSCAPE AND REQUIREMENTS



Mission Statement

With the information given and your own additional assumptions, you have to develop a **SUBSCRIPTION SYSTEM**. System will be composed of three microservices:

- Public Service: Backend for Frontend microservice to be used by UI frontend.
- Subscription Service: microservice implementing subscription logic, including persistence of subscription data in a database and email notification to confirm process is completed.
- Email Service: microservice implementing email notifications. No need to implement real email send process, you can create a Mock interface.

Subscription System should provide these operations:

- Create new subscription
- Cancel existing subscription
- Get details of a subscription
- Get all subscriptions

Tips

- Subscriptions should contain this information: email, first name, gender, date of birth, flag for consent and newsletter Id corresponding to the campaign. Only gender and first name are optional values. All the services should receive the same parameters.
- System should return the ID of the created subscription.
- No UI is required
- Service must be secure
- This Frontend application can be used as an example of how Public Service will be used: https://www.adidas.co.uk/on/demandware.store/Sites-adidas-GB-Site/en_GB/Newsletter-Subscribe

What we expect from you!

- Develop this application with a microservice approach, using NodeJS and some framework of your choice (it can be express, koa, NestJS, loopback...). All services must run independently from each other.
- Run each microservice in a different docker container linking them with any docker technology.
- Create a README.md explaining how to build/run/use the app naming every meaningful framework/library you use and state what it does and why you used it.
- Anyone having NodeJS and some standard tools should be able to check out the code, build and run the app locally.
- Provide API documentation using Swagger, API Blueprint or similar.
- Take care of non–functional requirements:
  - Implement security for non-public services.
  - Proper exception handling and REST responses.
  - Unit testing for meaningful code (business logic / services).
  - Be explicit about which node version to target (10, 12, 14, ...).
- Please use English as documentation language.
- BONUS 1: Either sketch, draw or implement a CI/CD pipeline proposal for the app.
- BONUS 2: Create config files for deploying the microservices into a kubernetes cluster.
- When you're done check in your solution into any public GIT repository (github, bitbucket, etc) and send us the link and any other documentation created.