Innopolis University, B24
Software Architecture

Autumn 2024

Hands-on:
Distributed services

In this task you need to implement the service-based pattern from [FOSA]. The implementation should be based on principles described in [FOSA]. You can use any procedural or OOP programming language.

The final application should consist of 3-4 services. Services should be separated based on business context, not technical details (domain vs technical partitioning). Notice, that any API gateway or other orchestrators are not considered services.

Business requirements

- You need to build a twitter-like system where users can post short messages (<400 characters), read them and like them.
- Users can't post messages if they are not registered. Registered users have a username and nothing else (not even password).
- Messages can be read by anyone as a feed. Feed should display last 10 messages.

What not to build:

- We do not require you to build an application that can handle millions of requests per second. Though you need to think which parts of the system should scale and handle large amount of clients better than the others.
- We do not require you to build WebUI with React or a mobile app for it - even CLI is fine. But we are not limiting you. Just remember that some problems can be solved much simpler.
- We do not require you to implement reverse proxies or load balancers by hand. You can use Nginx or any other proxy of your choosing if you need to.

Submission artifacts

- Git repository URL
- Video demo uploaded to YouTube or Google Drive

Grading criteria

- Compliance of implementation with service-based architectural style principles and quality of code.
  - **7/10 points**
- Quality of services separation based on scalability, capacity requirements, and business contexts.
  - **2/10 points**
- Quality of codebase organisation: naming of files and folders, clear separation of concerns, helpful README.md. Services should be clearly separated into folders.
  - **1/10 point**

[FOSA]: *Fundamentals of Software Architecture: An Engineering Approach by Mark Richards, Neal Ford*