

## Assignment 2: Service Oriented Architectures

### Web services and REST

Recall Assignment 1 and the architecture diagram you designed. We will now use the following architecture as a reference:

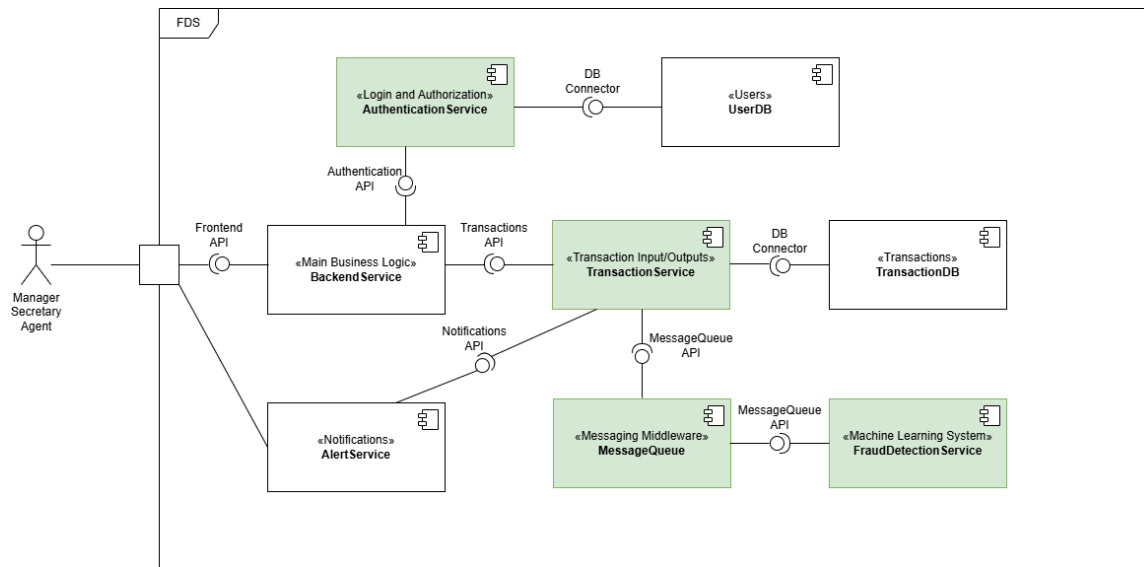


Figure 1: A possible architecture for our distributed system.

We now want to start implementing the system and develop some of its main services. For each service, you have to build only a part of the functionality, which is specified in the service description below. The main focus will be on **defining and implementing the interfaces** that each service exposes:

1. *Authentication service*: this service will be responsible for authenticating and logging users into the system. The following user groups (roles) are defined:
  - Administrator: can create and delete users and manage all types of data.
  - Secretary: can access only the customer data.
  - Agent: can receive fraud notification alerts, monitor and re-train the ML fraud detection system.

The service will expose an authentication API with the following functions:

- A function that accepts username/password and, if correct, emits a simple token (containing user role and a random base-64 string).
- A function for verifying that the token for a given user is still valid.

This service will be implemented with an in-memory cache (without persistent storage).

2. *Transaction service*: this service will be responsible for storing the metadata for the customers' transactions and manage at least two data tables in a persistent data store:
  - **transactions**: a table containing metadata information for each customer transaction (customer, timestamp, status (submitted, accepted, rejected), vendor-ID, amount).
  - **results**: a table containing metadata information for the predictions of the ML system for each transaction (transaction ID, timestamp, is-fraudulent, confidence).



The service will expose an API for importing new transactions, updating them and fetching the prediction results for each transaction. Note that only users of the user group *agents* and *administrators* are allowed to use this service. All other users and unauthenticated users will get an authorization error.

**Deadline:** 2024/04/22, 23:59 CET.

## Remarks

- You are free to use **any programming language** of your choice. You can even use different programming languages for different services, if you deem this appropriate.
- All files shall be submitted in a single zip file.
- A README.MD file will also be included in the submission with a short description of the submitted files.
- You may build a simple web UI for testing - since UI development is out of the scope of the course, it will not be graded.
- Every request performed by a client and all server responses must be logged with the following information: source, destination, headers (if applicable), metadata (if applicable), message body.
- You have free choice as to what web service/communication framework to use for each service: please explain your choice in the README.MD file.
- Don't spend much time on the business logic: please focus on setting up the services, APIs between services and logging.

## Assessment

Total: 15 points.

- All requirements are satisfied: 10 points.
- The documentation is concise and the choice on which web service/framework was used is well explained and technically correct: 5 points.