Appsfactory

Task: Application development "Insurance premium calculation"

The Pepperminzia insurance calculates the premium on the basis of the annual mileage, type class and regional class (county or state where the vehicle is registered).

The insurance company provides a CSV file for the data: postcode.csv. Here is the most important data from the CSV file:

- REGION1 Federal state
- REGION3 County
- REGION4 City
- POSTLEITZAHL Postal code
- LOCATION Town/Part of town

An application must be written which provides the functionality for calculating the insurance premium. The user entries and the result of the calculated insurance premium should be persisted. To calculate the premium, the user must enter the annual mileage and the postal code and select the vehicle manufacturer.

The formula is: Annual premium = factor annual mileage * factor type class * factor regional class

The following is specified for the factor annual mileage:

- 0 - 5.000 km - Factor 0.5
- 5.001 km - 10.000 km - Factor 1.0
- 10.001 km - 20.000 km - Factor 1.5
- ab 20.000km - Factor 2.0

The factor for regional class can be freely selected by state or county, as can the factor for type class.

Furthermore, there should be the possibility of integration, e.g. by a subsidiary of the insurance company. An API for calculating the insurance premium must be provided for this purpose.

Write an application with the following requirements:

- The entities must be persisted in a database. Which database and database type (SQL/NoSQL) do you use? Justify your decision.
- Create the corresponding microservices, at least two! Choose where to cut them based on customers domain requirements.

- Implement a testing framework and explain your approach to ensuring software quality.
- Which transactions/updates take place between the services?
- Create all assets and a documentation.
- (Optional) Provide a web UI to calculate the premium.

We're hanging up: *Domain Driven Design*, *Onion Application Architecture*, *Microservice-Architecture, use of principles to improve testability and maintainability*.