DEI
DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
TÉCNICO LISBOA

LAAS

THE LOYALTY-AS-A-SERVICE

Project for Enterprise Integration 2025 course

Your IE project locates in the innovative concept of **Loyalty-As-A-Service** (LaaS) that not only fosters customer loyalty within an individual supermarket but also **explores the feasibility of enabling direct cross-selling opportunities between different supermarkets**. LaaS is a development of the established system of loyalty programs. The LaaS point of difference is that companies use the **LaaS providers** to create, supervise and manage their respective loyalty programs instead of developing and managing their loyalty programs. LaaS providers offer various services, from customer engagement and data analysis to product design and management.

Therefore, your project' goal is to develop an information system for a LaaS provider.

The LaaS is designed to offer unique benefits, personalized promotions, and reward mechanisms to enhance customer retention. Additionally, the LaaS serve as a potential avenue for cross-collaborations, allowing customers to access exclusive promotions or discounts when shopping at partner supermarkets, thereby exploring new dimensions of customer engagement and inter-store cooperation.

The Core Concepts definition

A **loyalty card** works as a customer retention tool, systematically enrolling participants through online or in-store registration, providing them with **a unique identification-bearing card**. During transactions, customers present this card, linking purchases to their loyalty card. Through the accumulation of points or rewards based on transaction values, the program incentivizes repeated sponsorship, often incorporating bonus points for specific items or promotional periods. Utilizing data analytics, supermarkets tailor offers and promotions accordingly with individual shopping patterns, enhancing customer engagement. Patrons redeem accumulated points for benefits such as discounts or cashback, heightening the perceived value of participation. Exclusive invitations to events and promotions create a sense of exclusivity, fostering customer loyalty. Leveraging amassed data, supermarkets gain insights for strategic inventory management and targeted marketing, establishing a structured conduit for enduring and mutually beneficial engagement with their clientele.

Supermarket cross-selling refers to the practice of promoting and selling additional products or services to customers based on their current purchase, with the objective of maximizing sales opportunities and enhancing overall revenue. This strategy involves offering complementary or related items during the customer's shopping experience, leveraging the initial purchase to encourage additional transactions and broaden the range of products or services acquired by the customer.

Cross-selling between different supermarkets, where one sells its products within another, is not a conventional practice. However, supermarkets may engage in collaborations, joint promotions, or shared marketing efforts to mutually benefit from their respective customer bases.

The two business problems to be solved by your project

a) **Managing multiple loyalty cards can be problematic for consumers**: due to issues such as wallet clutter, the mental load of remembering various program details, potential privacy concerns, complexities in redeeming rewards, inconsistent user experiences, limited storage options, and the likelihood of becoming inactive members in some programs. The lack of uniformity across programs may create confusion and frustration for consumers, impacting the overall effectiveness of loyalty programs. While some consumers may appreciate the variety of rewards, streamlining loyalty programs through consolidated or universal solutions could mitigate these challenges and enhance the efficiency of managing multiple loyalty cards.

b) **Difficulties implementing direct cross-selling between separate supermarkets faces logistical complexity in coordinating inventory and pricing, compounded by concerns about data privacy and security**. The competitive dynamics of the retail environment, technological integration challenges, and regulatory compliance issues further complicate collaboration. Consumer resistance, operational misalignment, and the risk of cannibalization pose additional hurdles, requiring careful consideration and strategic approaches. Standardizing technology across the

retail sector and effectively communicating the value proposition of cross-selling initiatives are crucial for overcoming these challenges and establishing successful collaborations between distinct supermarket entities.

The application integration reference architecture

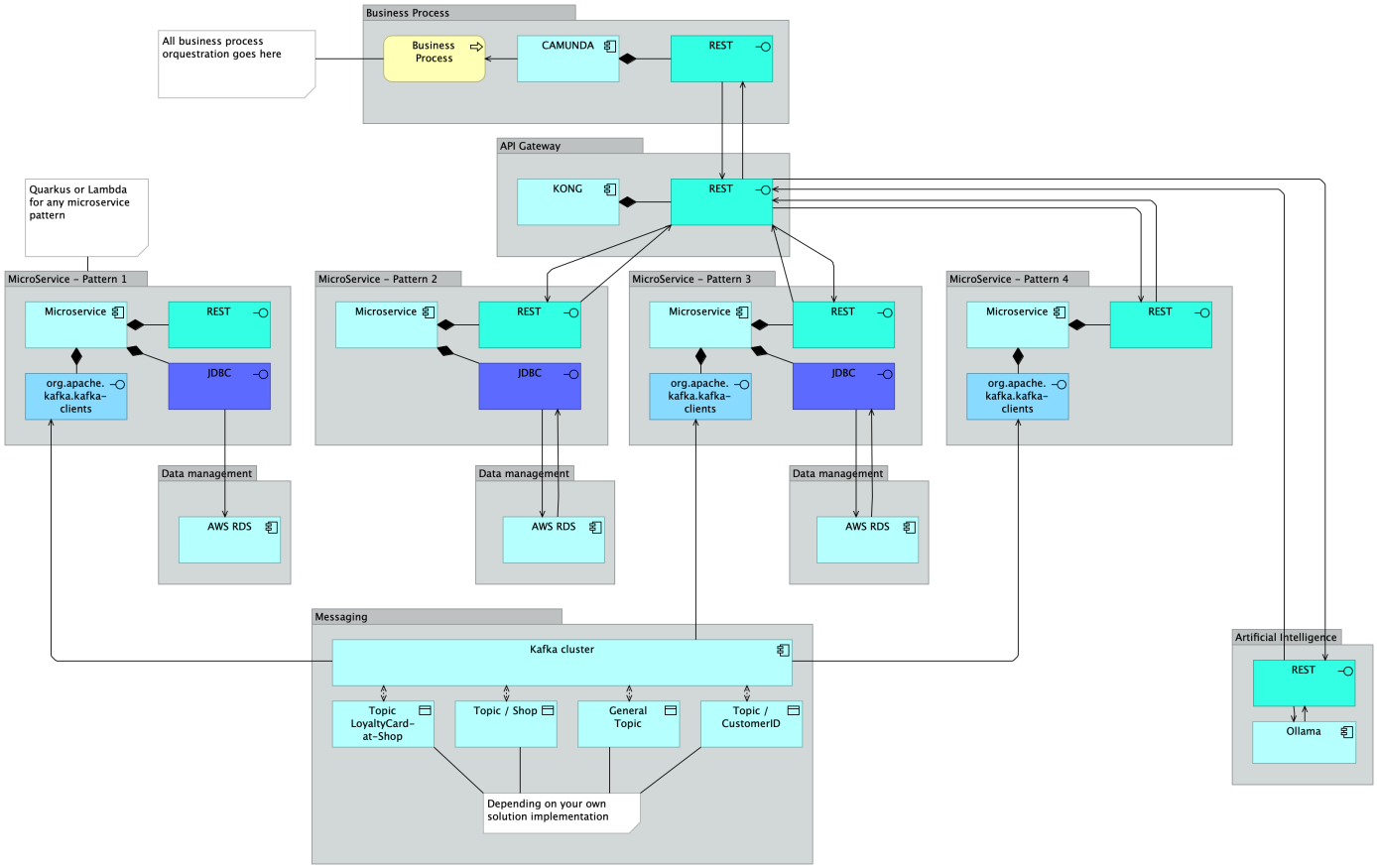

Figure 1. The microservice reference architecture patterns in ArchiMate. Each pattern is customized according to the needs of the requirements. From left to right. First pattern consumes from a Kafka cluster and stores the data entities in a database through a JBDC interface. Second, manages an entity stored in a database using REST and JDBC interfaces. Third, the most comprehensive, keeps a data entity consistent between a Kafka cluster and a database, and exposes that composed entity through REST, JDBC and org.apache.kafka.kafka-clients interface. Fourth pattern, produces entities to a Kafka cluster by exposing a REST interface and using the org.apache.kafka.kafka-clients interface. Finally, artificial intelligence exposes an API to API gateway offering a Ollama server that is deployed at EC2 to support the business processes decisions.

The data integration architecture

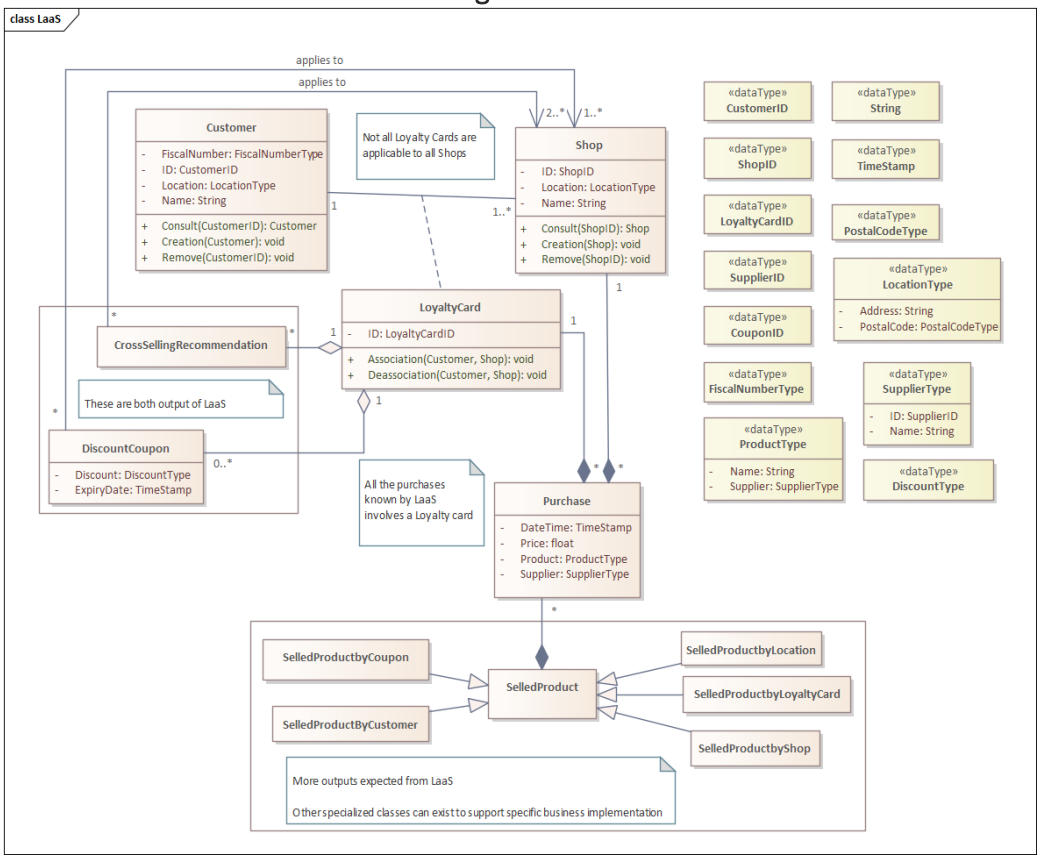

Figure 2. The UML domain model of LaaS system containing the most relevant classes and relationships.

The integration landscape architecture

Each one of classes depicted in Figure 2 represents one key concept taken from the LaaS domain that need to be managed by the LaaS system. To obtain a decoupled architecture it is recommended that each concept is modelled, and implemented, using the concept of a microservice. From Figure 3 to Figure 8, the role of each microservice is depicted in the scope of the required application integrations.

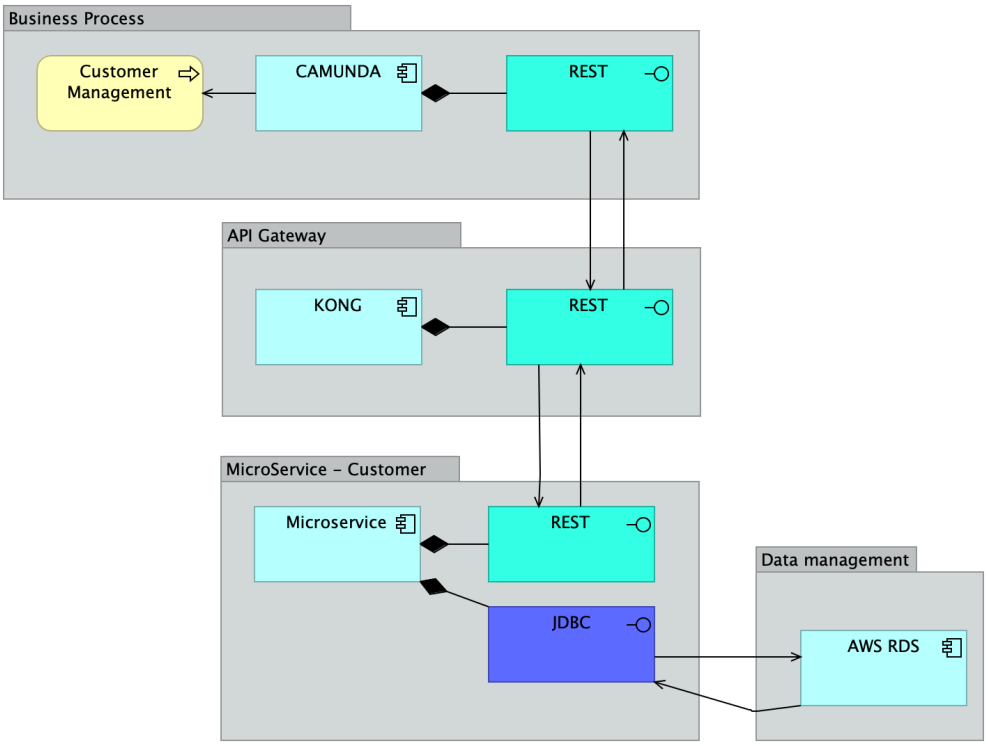

Figure 3. Customer microservice integration landscape architecture.

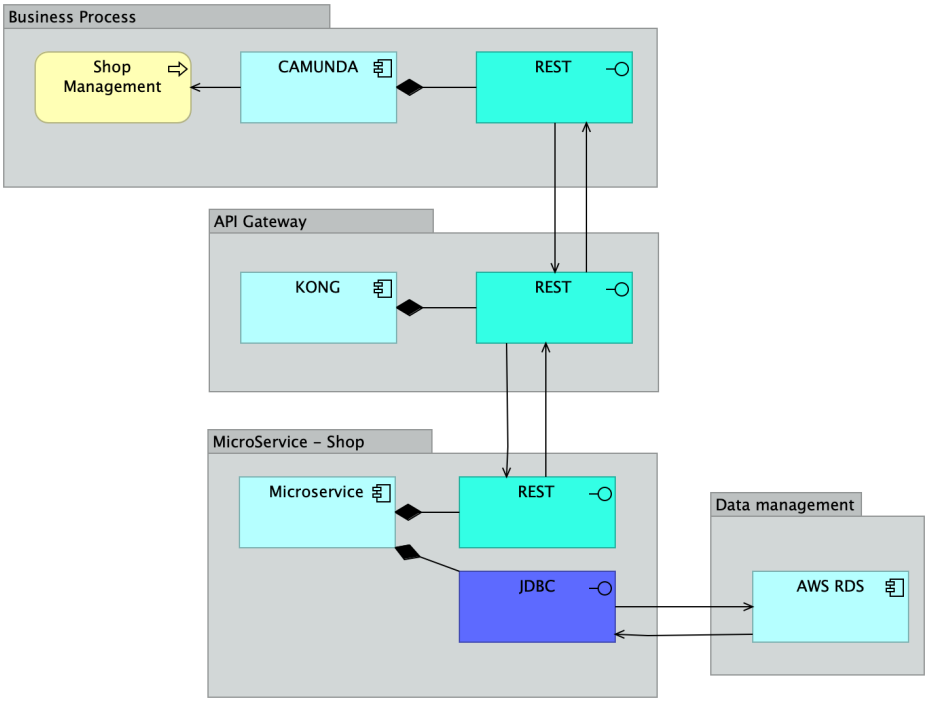

Figure 4. Shop microservice integration landscape architecture.

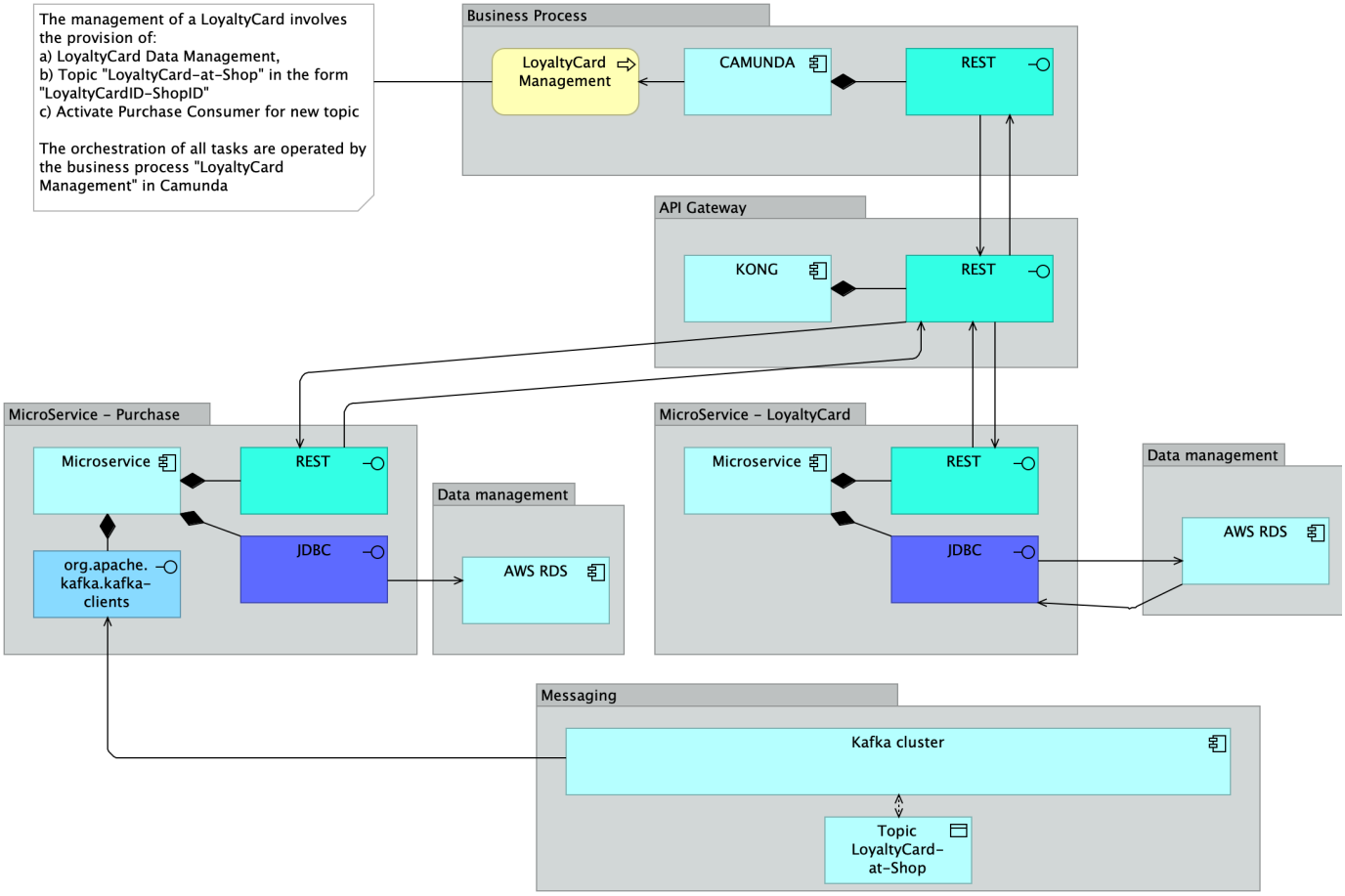

Figure 5. Loyalty Card microservice integration landscape architecture.

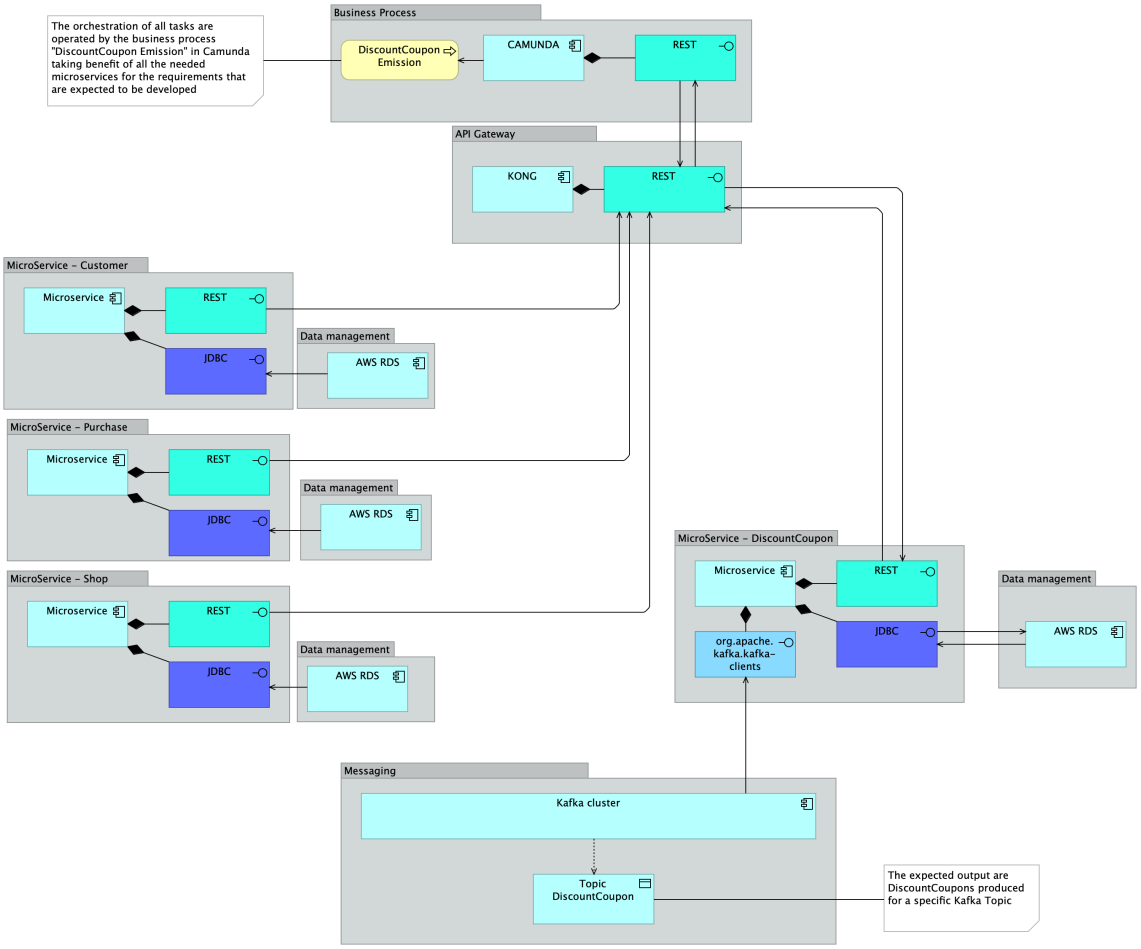

Figure 6. Discount Coupon microservice integration landscape architecture.

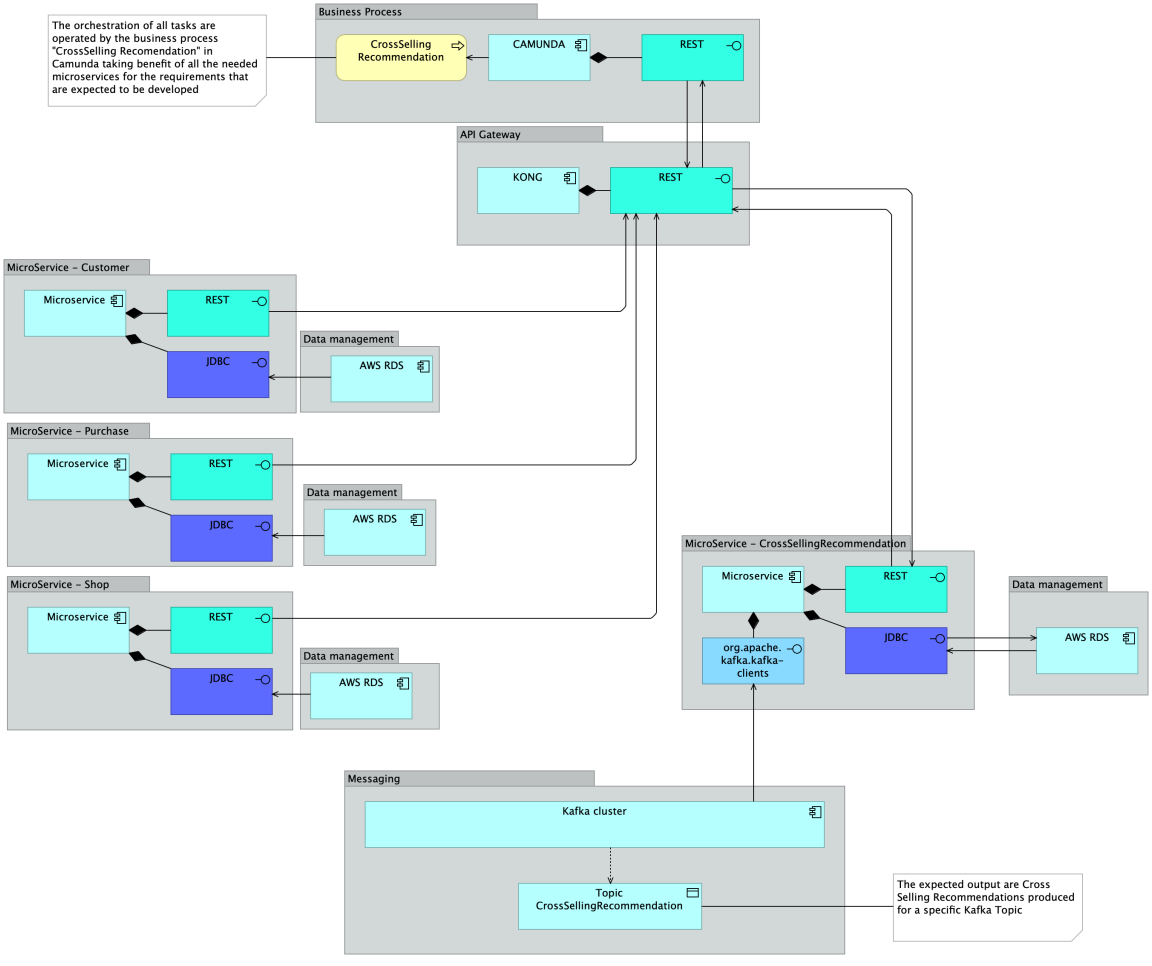

Figure 7. Cross Selling Recommendation microservice integration landscape architecture.

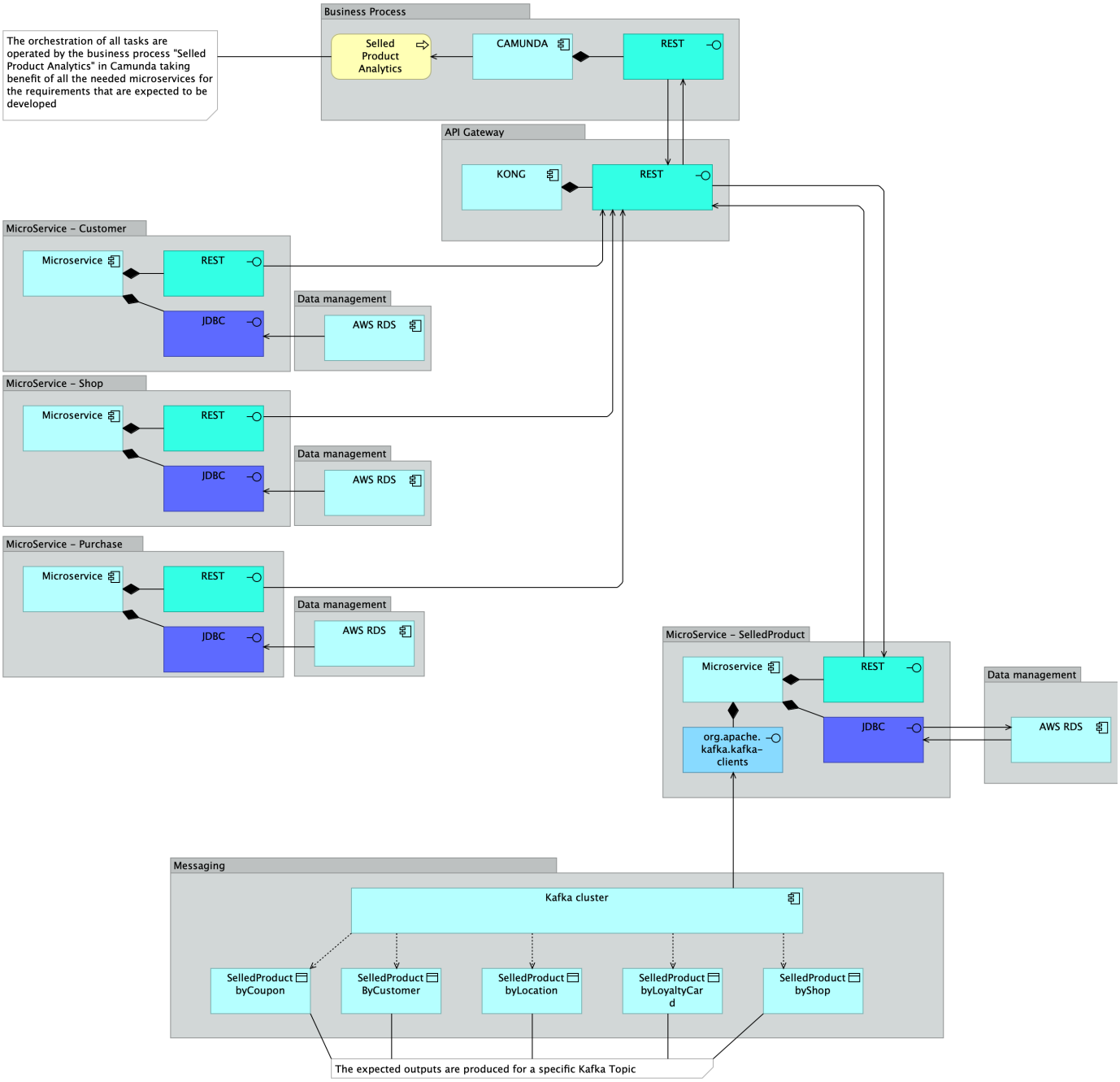

Figure 8. Selled Product microservice integration landscape architecture.

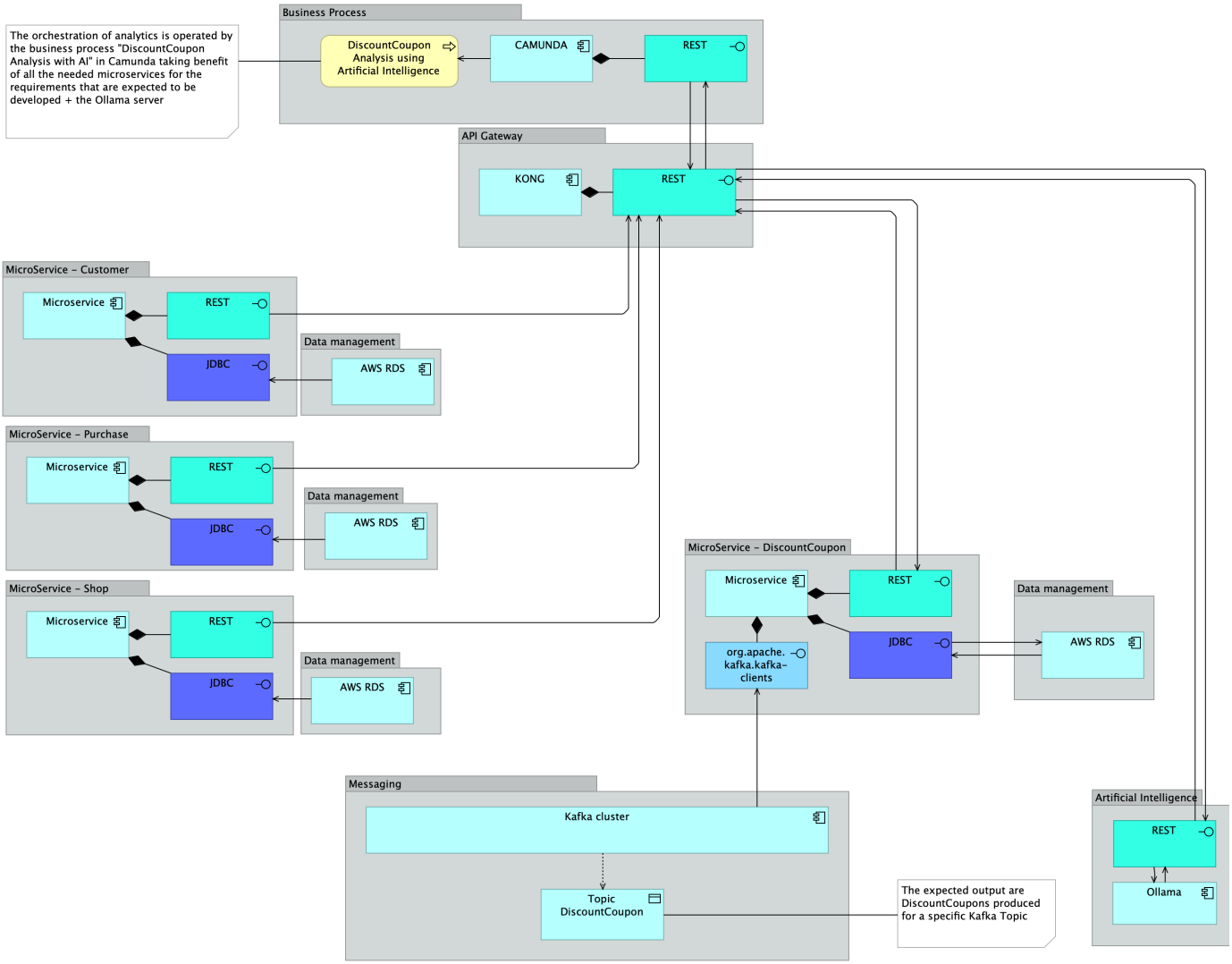

Figure 9. DiscountCouponAnalysis with Artificial Intelligence

Business processes to be developed in Camunda

a) ***Customer management*** – consists in the management of a customer, encompassing the creation, the decommission, the update, and the consultation of customers. Refer to Figure 3 for the implementation requirements, and for the required integrations.

b) ***Shop management*** – consists in the management of a shop, encompassing the creation, the decommission, the update, and the consultation of shops. Refer to Figure 4 for the implementation requirements, and for the required integrations.

c) ***Loyalty Card management*** - consists in the management of a loyaltycard. Each loyaltycard is an association between one customer and multiple shops. The core idea is for a customer to have a single card that can be used in many shops. The business process encompasses the association, the decommission, the update, and the consultation of loyalty cards. Refer to Figure 5 for the implementation requirements, and for the required integrations. As noticed in this Figure, the association involves the provision of:

a) LoyaltyCard Data Management,
b) Kafka Topic "LoyaltyCard-at-Shop" in the form "LoyaltyCardID-ShopID"
c) Activate Purchase Consumer for this new topic

d) ***Discount Coupon emission*** - consists in the emission of a discount coupon, encompassing the analysis of purchases of a given customer, that was associated with a loyalty card, to all shops. All the rules for the discount coupon emission are up to be decided by each working group. The resulting coupon is produced to one specific Kafka topic with all required elements to identify it. Refer to Figure 6 for the implementation requirements, and for the required integrations.

e) ***Cross Selling Recommendation*** – consists in the recommendation for cross selling of a given customer between different shops involving a single loyalty card, e.g., in collaborations, joint promotions, or shared marketing efforts. All the rules for the cross-selling recommendations are up to be decided by each working group. The resulting cross selling recommendation is produced to one specific Kafka topic with all required elements to identify it. Refer to Figure 7 for the implementation requirements, and for the required integrations.

f) ***Selled Product Analytics*** – consists in the quantitative analysis of the purchases. All the rules for analytics are up to be decided by each working group, the suggestions presented in Figure 2 can be extended to other metrics. The resulting analytics is produced to one specific Kafka topic with all required elements to identify it. Refer to Figure 8 for the implementation requirements, and for the required integrations.

g) **Discount Coupon analysis using Artificial Intelligence** – consists in the quantitative analysis of the purchases that used the produced discount coupons. This is a business assessment whether the customers valuate the produced discount coupons by using them. Refer to Figure 9 for the implementation requirements, and for the required integrations as depicted Figure 1 by Artificial Intelligence.

The event producer tool

The event producer tool acts as a simulator of purchases using the loyalty cards. The tool starts by discovering all the topic available in the kafka cluster (each topic is a different LoyaltyCardID-Shop) and then randomize messages for all that discovered topics.

The discovery assumes that each topic (*the pair: loyaltyCard-ShopName*) has a name accordingly with the following format name:

```
loyaltycardID-ShopName

For example: 560987123-ArcoCegoLisbon
```

One example of a JSON produced message is given below:

```
{
 "Purchase_Event":
          {
     "TimeStamp":"2024-02-09 10:44:07.748",
     "LoyaltyCard_ID":"560987123",
     "Price":"3.21",
     "Product":"Pie",
     "Supplier":"Feathered Friends Pet Haven",
     "Shop":"ArcoCegoLisbon"
          }
}
```

The tool is particularly useful to validate your solution. Start the tool accordingly with your Kafka configuration, and then, consume the produced messages from your architecture. To test performance, you can also increase the message throughput.
For further details, including the options available, please refer to https://github.com/Enterprise-Integration-IST-2025/LaaS-EventProducer. **You are free to customize the simulator accordingly with your project' purposes.** If so, you can also create a pull request to change this repository.

Moreover, Scenario 3 offers a baseline for the development of the core components of your project. See document of LaaS Proof-of-Concept (PoC) at *S3-Integration-Scenario.pdf*

Deliverables

1ST SPRINT

The first sprint consists in the definition of the informational flows, in the creation of a kafka cluster, in the development of the microservices (that could be based on the already available 4 microservices), and the testing of all the development components.

In specific the following deliverables are expected:

a. Expected deliverables

1. Design your information flow interpretation considering the overall architecture of LaaS
   a. We suggest you defining the Kafka topics and partitions, as well as the microservices to be used.

2. Implementation of the Kafka Cluster using TERRAFORM
   Suggestion: for testing purposes use the LaaS producer (https://github.com/Enterprise-Integration-IST-2024/LaaS-EventProducer) and create the needed topics by command line

3. The implementation of the following microservices, using ((Quarkus or AWSLambda) and AWS RDS) and TERRAFORM (and excluding the Camunda and Kong – business process implementation is a deliverable for 2nd sprint):

   - Customer
   - Shop
   - Loyalty Card
   - Purchase
   - DiscountCoupon
   - Cross Selling Recommendation
   - Selled Product
   - Discount Coupon analysis using Artificial Intelligence

4. Documentation of tests considering all the previous deliverables

5. Documentation for the source code, terraform scripts files, installation procedures, and parametrizations.

b. Submission

The submissions will be made **via Fénix**, in a single ZIP file containing the report PDF and the developed code, and scripts, artifacts.

c. Deadline

17/5/2025 12:00 – submission of all the sprint 1 deliverables

a. Expected deliverables

This second sprint reuses the microservices from previous sprint, and finalizes with:

1. The development of the support business processes using Camunda platform, Kong, and all the required integrations. All the developed microservices should now be used by the business processes:

    - Customer management
    - Shop management
    - Loyalty Card management
    - Discount Coupon emission
    - Cross Selling Recommendation
    - Selled Product Analytics
    - Discount Coupon analysis using Artificial Intelligence

2. Documentation of tests considering all the previous deliverables.

3. Documentation for the source code, script files, BPMN files, installation procedures, and parametrizations.

b. Submission

The submissions will be made **via Fénix**, in a single ZIP file containing the report PDF and the developed code, and scripts, artifacts.

c. Deadline

14/6/2025 12:00 – submission of all the sprint 2 deliverables