University of Wollongong 

TOURISM ECOSYSTEM

A Report submitted as a part of
the Assignment for the subject CSCI927

Service-Oriented Software Engineering

from

UNIVERSITY OF WOLLONGONG

by

Karan Goel - 7836685
Banin Sensha Shrestha - 8447196
Dipesh Baral - 8712785
Sudarshan Khadka - 8793694
Dang Tuan Nguyen - 8450535
Gaurav Santosh - 7032663
Muhammad Ateeb Aslam - 8262810

School of Computing and Information Technology
Faculty of Engineering and Information Sciences

2024

Tourism Ecosystem

Service-Oriented Software Engineering

School of Computing and Information Technology
University of Wollongong

ABSTRACT

The goal of this project is to create a holistic tourist ecosystem that integrates cutting-edge technology like artificial intelligence (AI) and mobile applications to transform the travel and tourism sector. Our goal is to establish a service-oriented software architecture that unifies multiple services offered by different tourist sector players, resulting in a smooth and personalised travel experience.

At least five recognised service providers, each providing a variety of services that address various facets of the traveler experience, shall be represented in the tourist ecosystem. These services include visitor information, reservations for eating and local attractions, as well as bookings for lodging and transportation. We will develop and construct a scalable platform that enables various services to interact cohesively and offers end users a unified and enhanced experience by utilising the concepts of service-oriented software engineering.

Our methodology entails defining and characterising services, creating the enterprise architecture, simulating business operations, and deploying specific services through the use of microservices architecture. In order to improve the system using data from actual user usage, we will also carry out service and process analytics. A fully working, executable system that showcases our design and development efforts will be the ultimate product, along with an extensive report and presentation.

MEMBER CONTRIBUTION

Table 1: Contribution Table

Name	Student Number	Contribution
Karan Goel	7836685	Contributed
Banin Sensha Shrestha	8447196	Contributed
Dipesh Baral	8712785	Contributed
Sudarshan Khadka	8793694	Contributed
Dang Tuan Nguyen	8450535	Contributed
Gaurav Santosh	7032663	Contributed
Muhammad Ateeb Aslam	8262810	Contributed

Table 2: Group contribution agreement

Name	**Signature**
Karan Goel	images/signature_karan.jpg
Banin Sensha Shrestha	images/signature_banin.jpeg
Dipesh Baral	images/signature_dipesh.jpeg
Sudarshan Khadka	images/sugnature_sudarshan.jpeg
Dang Tuan Nguyen	images/signature_dang.jpeg

Table of Contents

List of Tables

List of Figures

Part 1

Service Identification and Specification

In Part 1, we outline the identification of key external services that form the foundation of our tourism ecosystem.

We identified five key service providers, each offering three core services, which have been represented in the Business Service Reference Layer (BSRL). These service providers and their associated services are critical to fulfilling the diverse needs of travelers across various stages of their journey.

1.1 Service Providers

To identify the service providers, we focused on the needs of travelers at various touchpoints, ensuring a comprehensive tourism experience. A complete tourism ecosystem must support travelers at every step of their journey, which includes:

- Booking tickets for flights
- Reserving accommodations
- Providing dining options at restaurants
- Offering relevant information about local attractions

The roles of the service providers within the tourism ecosystem are outlined in Table 1.1.

The identified service providers and the core services they offer are presented in Table 1.2.

Service Provider	Role
Accommodation and Flights	Facilitate travel logistics and provide essential lodging options.
Local Attractions and Museums	Enhance cultural experiences and provide recreational activities.
Tourist Information Centers	Offer guidance and support to tourists for a better experience.
Restaurants and Cafes	Promote local cuisine and provide dining experiences.
Local Transportation Services	Enable convenient travel and exploration within the destination.

Table 1.1: Roles of Service Providers in the Tourism Ecosystem

Service Provider	Core Services
Accommodation and Flights	Flight Booking Room Booking Loyalty Programs
Local Attractions and Museums	Ticket Booking Guided Tours Educational Content Delivery
Tourist Information Centers	Visitor Information Tour Bookings Event Notifications
Restaurants and Cafes	Table Reservations Local Cuisine Offers Online Ordering
Local Transportation Services	Vehicle Rentals Route Planning Real-Time Traffic Information

Table 1.2: Key Services Provided by External Service Providers

1.2 BSRL For Services

Accommodation and Flights

Flight Booking

```
Goal: This service allows the customer to check for flight ticket availabilities
   and reserve one.
Preconditions:
    The customer accesses the system.
Postconditions:
```

```
    The system deducts one ticket from the inventory and the booking information
       displays in the customer booking dashboard.
Assumptions:
    The customer must have an active phone number and an email address.
Inputs:
    The customer enters the following information:
    - From <string>: The departure airport
    - To <string>: The destination airport
    - Depart <date>: The date of departure
Outputs:
    The customer receives an email confirming the reservation.
Resources:
    Third-party services for flight information, secure payment gateway.
Key Steps:
    1. Customer enters input information.
    2. The system displays the best flights based on several factors such as prices
       , number of connecting flights, time, etc.
    3. The customer chooses the best offer.
    4. The customer selects seats, luggage bags, insurance, etc.
    5. The customer enters personal information: name, email, phone number.
    6. The customer is redirected to the payment page.
    7. The customer enters payment information such as credit card details.
    8. A confirmation email is sent or an error message is displayed if anything
       goes wrong.
QoS (Quality of Service):
    - Available 24/7.
    - Loading time when users first visit must be less than 1 second.
    - Return search results within 1-3 seconds.
    - Secure payment transaction.
    - Conversion rate (the number of customers who booked a ticket successfully per
        the number of customers who visit the site) is above 30%.
Penalties:
```

```
    - Once the booking is confirmed, the customer is charged 30% of the ticket fees
      if they want to cancel. Cancellation must be made at least 2 weeks before
      the departure/return date.
    - The customer is charged 10% of the ticket fees for changes to the departure/
      return date. The adjustment must be made at least 7 days before departure.
      New fees apply based on current ticket prices.
    - If the airline cancels the flight, 80% of the ticket fees are refunded and a
      20% discount is given for the next booking.
```

Room Booking

```
Goal:
    This service allows customers to search for available rooms and book
       accommodation.
Preconditions:
    The customer accesses the system and provides necessary information.
Postconditions:
    The system deducts one room from the inventory, and the booking information is
       displayed in the customers dashboard.
Assumptions:
    The customer must have an active phone number and an email address.
Inputs:
    - Location <string>: The city or area where the customer wants to book a room.
    - Check-in <date>: The date of arrival.
    - Check-out <date>: The date of departure.
    - Room type <string>: The type of room (e.g., single, double, suite).
    - Number of guests <int>: The number of people staying.
Outputs:
    The customer receives an email confirming the reservation and the details are
       shown in the dashboard
Resources:
    Integration with hotel databases, secure payment gateway, and booking
       management systems.
```

```
Key Steps:
    1. Customer enters input information.
    2. The system displays available rooms based on filters such as price,
       amenities, and location.
    3. The customer selects a room and additional services (e.g., breakfast,
       airport pickup).
    4. The customer is redirected to the payment page.
    5. The customer enters payment information and confirms the booking.
    6. A confirmation email is sent, or an error message is displayed if an issue
       occurs.
QoS (Quality of Service):
    - Available 24/7.
    - Initial page load time should be under 1 second.
    - Search results should be returned within 1-3 seconds.
    - Secure payment transactions.
    - Conversion rate above 25%.
Penalty:
    - A 30% fee applies for cancellations made at least 7 days before the check-in
      date.
    - A 10% fee applies for changes to the booking made at least 3 days before the
      check-in date. Additional fees apply based on new room rates.
```

Loyalty Programs

```
Goal:
    This service allows customers to enroll in a loyalty program, earn points for
       stays, and rewards for those points.
Preconditions:
    The customer has an account on the system and is logged in.
Postconditions:
    Loyalty points are updated in the customers account after a stay or service
       is completed.
Assumptions:
```

```
    The customer provides accurate personal information.
Inputs:
    - Customer ID <string>: Unique identifier for the customer.
    - Points accrued <int>: Number of points earned per booking or stay.
    - Rewards redeemed <string>: The type of reward redeemed (e.g., free night,
       discount).
Outputs:
    The customer receives notifications about points earned and rewards redeemed
       via email and dashboard updates.
Resources:
    Loyalty program database, integration with room booking and other service
       systems.
Key Steps:
    1. The customer enrolls in the loyalty program.
    2. Points are automatically added after eligible transactions.
    3. The customer views points and available rewards in their account.
    4. The customer redeems points for rewards or services.
    5. The system confirms the reward and adjusts the point balance accordingly.
QoS (Quality of Service):
    - Available 24/7.
    - Points update immediately after eligible transactions.
    - Secure handling of customer data.
    - User-friendly interface for managing loyalty points.

Penalty:
    - No penalties for customers, but points may expire if not used within a
       specified time frame (e.g., 2 years).
```

Local Attractions and Museums

Ticket Booking

```
Goal:
```

```
    This service allows customer to check for flight ticket availabilities and
       reserve one.
Preconditions:
    The customer accesses to the system.

Postconditions:
    The system deducts one ticket from inventory and the booking information
       displays in the customer booking dashboard.
Assumptions:
    The customer must have an active phone number and an email address.
Inputs:
    - From <string>: The departure airport.
    - To: <string>: The destination airport.
    - Depart <date>: The date of departure.
Outputs:
    The customer receives an email confirms the reservation.
Resources:
    Third-party services for flight information, secure payment gateway.
Key Steps:
    1. Customer enters input information.
    The system displays the best flights based on several factors such as prices,
       number of connecting flights, time, etc.
    2. The customer chooses the best offer.
    3. The customer chooses seats, luggage bags, insurance, etc.
    4. The customer enter personal information: name, email, phone number.
    5. The customer is redirected to payment page.
    6. The customers enter information such as credit card information.
    7. A successfully booked email is sent or displays an error message if anything
        goes wrong.

QoS (Quality of Service):
    - Available 24/7.
    - Loading time when the users first visit must less than 1s.
```

```
    - Return search results within 1-3s.
    - Secure payment transaction.
    - Conversion rate (The number of customers booked a ticket successfully per the
        number of customers visit the site) is above 30% Available 24/7.
    - Loading time when the users first visit must less than 1s.
    - Return search results within 1-3s.
    - Secure payment transaction.
    - Conversion rate (The number of customers booked a ticket successfully per the
        number of customers visit the site) is above 30%.
Penalty:
    - Once the booking is confirmed, the customer is charged 30% of the ticket fees
        if they want to cancel the ticket. The ticket cancelation needs to be made
       at least 2 weeks before departure/return date.
    - Once the booking is confirmed, the customer is charged 10% of the booked
       ticket fees if they want to change the departure/return date. The adjustment
        needs to be made at least 7 days before departure/return date. The new fees
        will apply according to the fee of the new ticket at the time purchase.

    - In the case of flight cancelation due to the airline company, 80% of the
       ticket fees will be refunded to the customer and they will get a discount of
        20% for the next booking.
```

Guided Tours

```
Goal:
    This service allows customers to explore tours in the local areas and book a
       tour if they are interested in.
Preconditions:
    The customer accesses the system and allows the application to access their
       location.
Postconditions:
    The system deducts one ticket from inventory and the booking information
       displays in the customer booking dashboard.
```

```
Assumptions:
    The customer must have an active phone number and an email address.
Inputs:
    - The system detects the customers location based on GPS.
Outputs:
    The customer receives an email confirming that they have booked the tour
       successfully and related posts on the system relating to the attractions
       they are going to visit.
Resources:
    Available tours in partnership with local travel companies.
Key Steps:
    1. The customer visits the tour booking page.
    2. The application asks for the permission to access the customers location.
    3. The application displays available tours in the area.
    4. The customer choose a tour.
    5. The customer filled in information such as phone number, email, etc.
    6. The customer is redirected to checkout page.
    7. The customer fills in credit card information.
    8. The system sends the confirmation email.
QoS (Quality of Service):
    - Available 24/7.
    - Loading time when the users first visit must less than 1s.
    - Secure payment transaction.
    - Conversion rate (The number of customers booked a ticket successfully per the
        number of customers visit the site) is above 30%.
Penalty:
    - Once the reservation is confirmed, the customer is charged 30% of the price
       of the tour if they want to cancel and needs to be announced at least 5 days
        before the tour occurs.
    - Once the reservation is confirmed, the customer is charged 10% of the booked
       tour fees if they want to change the date of the tour.
    - In case the tour needs to be cancelled due to inevitable factors such If the
       tour must be cancelled due to inevitable factors such as bad weather, the
```

```
        customer will get a discount of 30% for the next booking.
```

Educational Content Delivery

```
Goal:
    This feature allows customer to post a review about an attraction.
Preconditions:
    The review is stored in the system database and visible to other users.
Postconditions:
    The review is stored in the system database and visible to other users.
Assumptions:
    The customer has an account in the system and logged in.
Inputs:
    - The content of the review includes texts, images, videos.
Outputs:
    A full review of an attraction.
Resources:
    Existing users on the system.
Key Steps:
    1. The user logs into the system.
    2. The user writes their review.
    3. The review is stored in the system.
QoS (Quality of Service):
    - Available 24/7.
    - Content is stored in the system for a long period of time.
    - The UI is friendly.
```

Tourist Information Centers

Visitor Information

```
Goal:
```

```
    Provide appropriate and up-to-date information to tourists regarding local
       tourist destinations, various local activities, and general inquiries.
Preconditions:
    Customer accesses the system.
Postconditions:
    Customer receives up-to-date information and assistance regarding their queries
       .
Assumptions:
    Customer has access to the system preferably registered with a phone number or
       email address.
Inputs:
    - Query: <string> - The specific information the customer is seeking.
Outputs:
    Customer receives the requested information through the chosen channel.
Resources:
    - Up-to-date databases of local attractions and activities, customer query
       storing and retrieval database.
    - Trained staff for query addressing.
Key Steps:
    1. Customer accesses the system.
    2. Customer submits their query.
    3. System/staff retrieves the query and looks up to the relevant information
       from the database.
    4. Information is provided to the customer through system/email/or phone.
QoS (Quality of Service):
    - Available 24/7.
    - Response time to queries must be less than 1 minute.
    - Information accuracy rate above 99%.
Penalty:
    None
```

Tour Bookings

```
Goal:
    Provide the booking and reservation of various tours for tourists.
Preconditions:
    Customer accesses the booking system.
Postconditions:
    System confirms the booking and updates the booking database and page with the
        details.
Assumptions:
    Assumptions: Customer has internet access and can provide necessary booking
        details.
Inputs:
    - customer: <string> - details of the customer.
    - Tour_details: <string> - The tour the customer wishes to book.
    - Date: <date> - The date of the tour.
    - Number of participants: <int> - Number of people attending the tour/event.
Outputs:
    Customer receives a confirmation email or notification and booking details are
        updated in the system.
Resources:
    - Tour details database.
    - Secure payment gateway.
Key Steps:
    1. Customer selects a tour from the available options.
    2. Customer enters the necessary details and submits the booking request.
    3. System processes the booking and checks availability.
    4. Customer completes the payment process.
    5. System confirms the booking and sends a confirmation email.
QoS (Quality of Service):
    - Available 24/7.
    - Booking confirmation within 5 minutes.
    - Secure payment transactions.
    - Booking success rate above 95%.
    - Minimal booking cancelation.
```

```
Penalty:
    - 25% cancellation fee if canceled 3 days before the tour date.
    - 10% change fee if the tour date is modified 7 days before the tour date.
```

Event Notifications

```
Goal:
    Inform tourists about upcoming local events, festivals, and activities.
Preconditions:
    Customer has subscribed to the notification system through email newsletters,
       social media updates, or mobile apps.
Postconditions:
    Customer receives notifications about relevant events.
Assumptions:
    Customer has registered to the system via email address or phone numbers.
Inputs:
    - Subscription<boolean> - Customer signs up to receive notifications.
Outputs:
    - Notfication <string>- customer receives the notification.
Resources:
    - Event details database.
    - Subscription details database.
Key Steps:
    1. Customer subscribes to event notifications.
    2. System gathers upcoming event information.
    3. Notifications are sent out to subscribers.
    4. Customers receive and view event notifications.
QoS (Quality of Service):
    -  Available 24/7.
    - Notification delivery with minimal time.
    - Event information accuracy above 99%.
Penalty:
    None.
```

Restaurants and Cafes

Table Reservations

```
Goal:
    Provide customers with the ability to reserve tables at restaurants and cafes
       for specific dates and times.
Preconditions:
    Customer must have access to the reservation system via website or mobile app.
Postconditions:
    Reservation is confirmed, and a confirmation message is sent to the customer.
Assumptions:
    The restaurant has available tables for the selected date and time.
Inputs:
    - Customer Name <string>
    - Date <date>
    - Time <time>
    - Number of Guests <integer>
Outputs:
    - Reservation Confirmation <string>
Resources:
    - Reservation management system, database for table availability.
Key Steps:
    1. Customer selects date and time.
    2. System checks table availability.
    3. Reservation is confirmed and logged.
QoS (Quality of Service):
    - Availability <24/7>
    - Response Time <Immediate>
    - Reliability <99.9% uptime>
Penalty:
    None.
```

Local Cuisine Offers

```
Goal:
    Provide information on special local cuisine offers and promotions at
       restaurants and cafes.
Preconditions:
    Offers must be set up and active in the system.
Postconditions:
    Customer receives information on available offers.
Assumptions:
    Offers are current and accurate.
Inputs:
    - Location <string>
    - Customer Preferences <string>
Outputs:
    - List of Available Offers <string>
Resources:
    - Offer management system, local database.
Key Steps:
    1. System retrieves active offers.
    2. Offers are filtered based on location and customer preferences.
QoS (Quality of Service):
    - Customization <Based on customer profile>
    - Update Frequency <Real-time>
    - Accuracy <100% of active offers>
Penalty:
    None.
```

Online Ordering

```
Goal:
    Allow customers to browse menus, place orders, and make payments online.
Preconditions:
    The restaurants online service system must be operational.
```

```
Postconditions:
    Orders are placed, and payments are processed.
Assumptions:
    Secure internet connection and payment gateway availability.
Inputs:
    - Customer Order <order details>
    - Payment Information <payment details>
Outputs:
    - Order Confirmation <string>
    - Payment Receipt <string>

Resources:
    - Online ordering system, secure payment gateway.
Key Steps:
    1. Customer browses menu.
    2. Customer places an order and provides payment details.
    3. Order is confirmed and payment is processed.
QoS (Quality of Service):
    - Security <Encryption>
    - User Interface <User-friendly>
    - Support <24/7>
Penalty:
    None.
```

Local Transportation Services

Vehicle Rentals

```
Goal:
    Provide customers with the ability to rent vehicles for local transport.
Preconditions:
     The customer must have access to the rental system via the website or mobile
        app.
```

```
Postconditions:
    Reservation is confirmed, and a confirmation message is sent to the customer.
Assumptions:
    The vehicle must be available for the selected date and time.
Inputs:
    - Customer Name <string>
    - Vehicle Type <string>
    - Start Date <date>
    - End Date <date>
    - Number of Passengers <int>
Outputs:
    - Reservation Confirmation <string>
Resources:
    - Vehicle inventory system, booking management system.
Key Steps:
    1. Customer selects vehicle type and rental dates.
    2. System checks vehicle availability.
    3. Reservation is confirmed and logged.
QoS (Quality of Service):
    - Available 24/7
    - Response time for booking confirmation within 5 minutes.
    - Reliability with 99.9% uptime.
Penalty:
    - A 30% fee applies for cancellations made at least 48 hours before the rental
       start date.
    - A 10% fee applies for changes made within 24 hours of the rental start time.
```

Route Planning

```
Goal:
     Provide users with optimized route planning for their journey.
Preconditions:
     User must enter starting and destination points.
```

```
Postconditions:
    The system displays the recommended route and estimated travel time.
Assumptions:
    The user has an internet connection and access to the system.
Inputs:
    - Start Location <string>
    - Destination <string>
    - Travel Date <date>
Outputs:
    - Recommended Route <string>
    - Estimated Travel Time <string>
Resources:
    - Mapping and routing service integration.
Key Steps:
    1. User inputs start and destination locations.
    2. System calculates optimal routes based on traffic data.
    3. Users are presented with route options.
QoS (Quality of Service):
    - Response time for route calculations should be within 1-3 seconds.
    - Accuracy of travel time estimates should be above 95%.
Penalty:
    None specified.
```

Real-Time Traffic Information

```
Goal:
     Provide users with real-time traffic updates to enhance route planning and
        avoid delays.
Preconditions:
     The user must have access to the internet and allow location services.
Postconditions:
    Users receive live traffic updates that affect their planned routes.
Assumptions:
```

```
     The system is connected to reliable traffic data sources.
Inputs:
    - Current Location <string>
    - Destination <string>
Outputs:
    - Traffic Updates <string>
    - Suggested Alternate Routes <string>
Resources:
    - Traffic monitoring systems, GPS integration.
Key Steps:
    1. User enters current location and destination.
    2. System retrieves current traffic data.
    3. User is informed of any traffic issues and alternate routes.
QoS (Quality of Service):
    - Response time for route calculations should be within 1-3 seconds.
    - Accuracy of travel time estimates should be above 95%.
Penalty:
    - Updates should be provided within seconds of traffic changes.
    - Accuracy of traffic data should be above 90%.
```

Part 2

Enterprise Architecture (EA) Design using ArchiMate

In Part 2, we delve into the design of Enterprise Architecture (EA) for the tourism ecosystem, utilizing the ArchiMate framework. This design aims to capture the relationships between various actors and services in the ecosystem, providing a clear blueprint for collaboration and service delivery.

Figure 2.1 illustrates the high-level architecture of the ecosystem, detailing the key components and their interactions.

2.1 Business Layer

The Business Layer represents the operational aspects of the tourism ecosystem, focusing on how external roles interact with business processes and services. It encompasses both the **External Business Services** provided by service providers and the internal **Business Processes** that support customer interactions.

External Roles and Actors

In this ecosystem, the primary actor is the **Customer**, who assumes the role of a **Traveller**. The Traveller engages with various service providers throughout their journey, from booking accommoda-

tions and transportation to seeking information about local attractions. The Customer's interactions are essential in driving the business processes and ensuring the seamless delivery of services within the tourism ecosystem.

External Business Services

The tourism ecosystem is supported by several **External Business Services** provided by various service providers, as mentioned in Section 1.1. These providers play a crucial role in delivering key services that enrich the Traveller's experience, ensuring seamless travel, accommodation, and entertainment options throughout their journey.

Business Processes

The core business processes within the ecosystem are designed to support both the operational needs of the service providers and the experience of the Traveller. These processes can be divided into two main categories:

1. **Booking Process**:
    - **Get Traveller Details**: The system gathers essential information about the Traveller, such as name, travel dates, and preferences.
    - **Create Booking**: Once the Traveller's details are confirmed, the booking is generated for services such as accommodation, transportation, or attractions.
    - **Take Payment**: The system processes payments from the Traveller to secure the booking, ensuring a smooth transaction flow.
2. **Information Provision Process**:
    - **Receive Query**: The system or agent receives a query from the Traveller, such as a request for information about local attractions or travel details.

- **Fetch Relevant Information**: Based on the query, the system retrieves accurate and up-to-date information from various sources.
- **Provide Information**: The system or agent delivers the requested information back to the Traveller, either through a mobile app, website, or direct customer service interaction.

These processes ensure that the needs of Travellers are efficiently met. The **Customer Service** function, whether fulfilled by an agent or an app, plays a central role in executing these processes, ensuring that every query or booking is handled smoothly.

2.2 Application Layer

The Application Layer comprises various application services, including their internal components, that support the underlying business processes.

External Application Services

The external services provided by the tourism ecosystem are realized through different apps, each catering to specific services. These apps implement the core business processes using a microservices architecture and provide a user-friendly frontend for seamless interaction.

The Table 2.1 outlines the services and their corresponding apps:

Table 2.1: Services and Corresponding Apps

Service	App
Accommodation and Flights	Flight Booking App, Accommodation Booking App
Local Attractions and Museums	Tour Booking App
Tourist Information Centers	Tourist Information Center App
Restaurants and Cafes	Restaurant Booking App
Local Transportation Services	Local Transport App

Additionally, all services are accessible through a web-based **Frontend** that acts as a central hub for Travellers to interact with the ecosystem.

Application Components

The application services rely on several internal components that support key functionalities and processes. These components work together to deliver a seamless experience for users.

- **User**: The primary component representing the Traveller who interacts with the system.
- **CRM System (Customer Relationship Management System)**: Manages user data, ensuring efficient tracking of customer interactions, preferences, and booking history.
- **Booking System**: An asynchronous booking system that uses a messaging service to process ticket bookings without delays, ensuring a smooth booking experience.
- **Transaction Management**: The combination of the Booking System and CRM System forms the **Transaction Admin**, which powers the overall booking process, handling customer interactions, payments, and confirmations.
- **CMS System (Content Management System)**: This system stores and manages information provided by the Information Center, ensuring that up-to-date content is available to Travellers.
- **Information Center**: Comprising functions to **Get** and **Save Information**, the Information Center is connected to the CMS System and serves as a hub for retrieving and storing relevant details for Travellers.
- **Information Management**: The CMS System and Information Center work together to power the **Information Process**, ensuring that Travellers receive accurate and timely information.

2.3 Technology Layer

The Technology Layer encompasses the infrastructure and services that support the application and business layers. It provides the foundational components that ensure scalability, reliability, and efficient operation.

Infrastructure Services

- **Pub/Sub Service**: This publish-subscribe service powers the asynchronous booking system, allowing for real-time messaging and communication between components without delays.
- **NFS (Network File System)**: The NFS is utilized to support the CMS system, enabling efficient storage and retrieval of content for the Information Center.

Infrastructure

The entire infrastructure is hosted on a **Cloud Platform**, ensuring scalability and high availability. It is deployed in a clustered environment to provide fault tolerance and continuous uptime, ensuring that services remain available even during high demand or system failures.

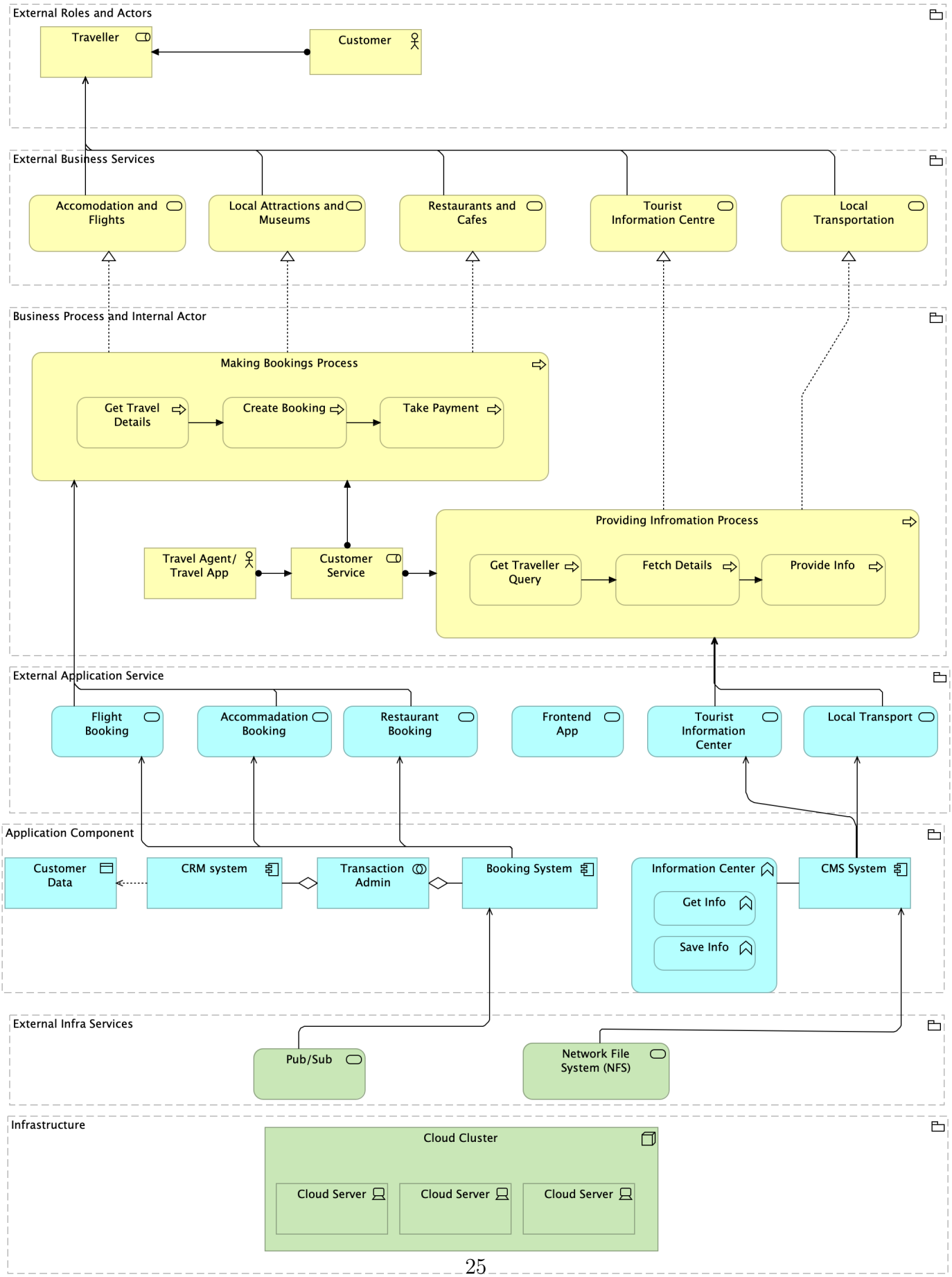


Figure 2.1: Enterprise Architecture Design for the Tourism Ecosystem

Part 3

Business Process Design and Analysis

In this part, we will explore the BPMN (Business Process Model and Notation) design for each service in detail, along with their semantic annotations.

3.1 BPMN

Booking Process BPMN

The Booking Process encompasses Flight Booking, Room Booking, and Tour Booking. The process begins by collecting necessary details from the user. Once the user information is gathered, the system interacts with specific APIs and suppliers. For instance, it sends a request to the Global Distribution System (GDS) to fetch relevant details, which are then displayed to the user.

The user is presented with various options and packages based on the retrieved data. After selecting their desired options, the user is directed to the payment page. If the payment fails, the user is redirected back to attempt the payment again. Upon successful payment, the system sends a confirmation email containing the ticket and invoice to the user.

The scenario labels and exclude sets for task T8 are as follows:

- T8 scenario (1): $\langle T1, T2, T3, \{T4, T5, T6\}, T7, T8 \rangle$

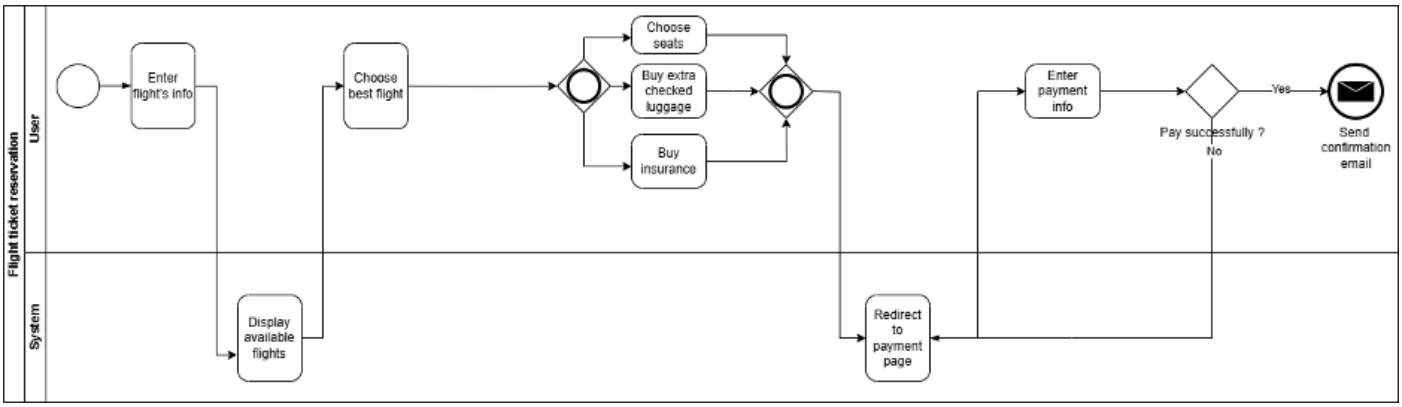


Figure 3.1: Flight Booking BPMN Diagram

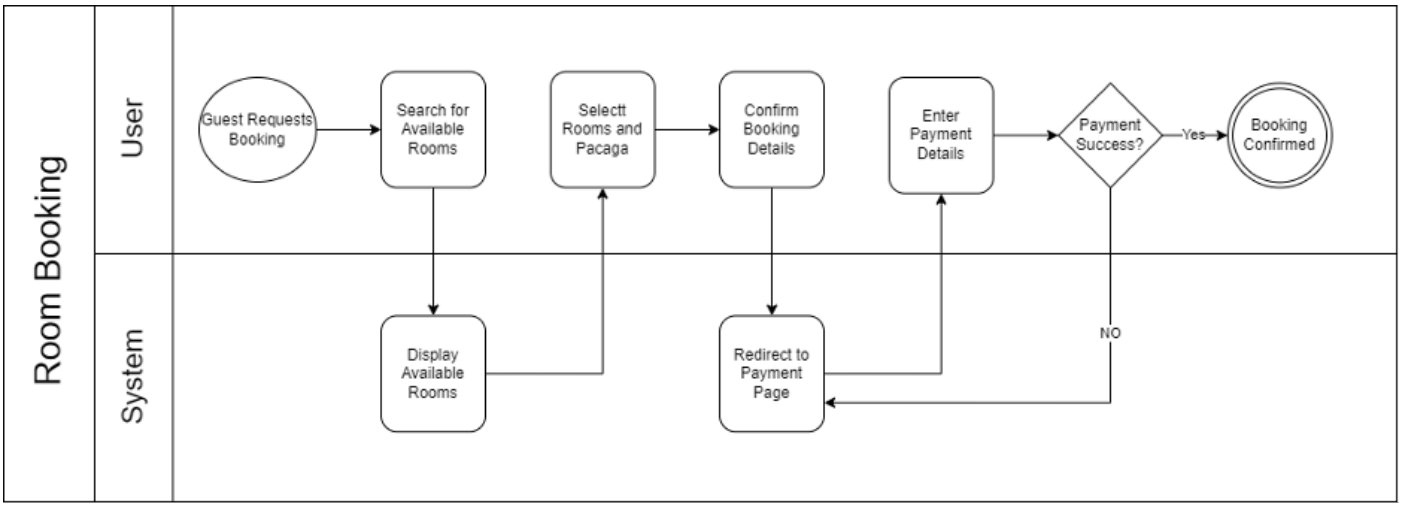


Figure 3.2: Room Booking BPMN Diagram

- T8 scenario (2): $\langle T1, T2, T3, \{T4, T5\}, T7, T8\rangle$
- T8 scenario (3): $\langle T1, T2, T3, \{T4, T6\}, T7, T8\rangle$
- T8 scenario (4): $\langle T1, T2, T3, \{T5, T6\}, T7, T8\rangle$
- T8 scenario (5): $\langle T1, T2, T3, \{T4\}, T7, T8\rangle$
- T8 scenario (6): $\langle T1, T2, T3, \{T5\}, T7, T8\rangle$
- T8 scenario (7): $\langle T1, T2, T3, \{T6\}, T7, T8\rangle$

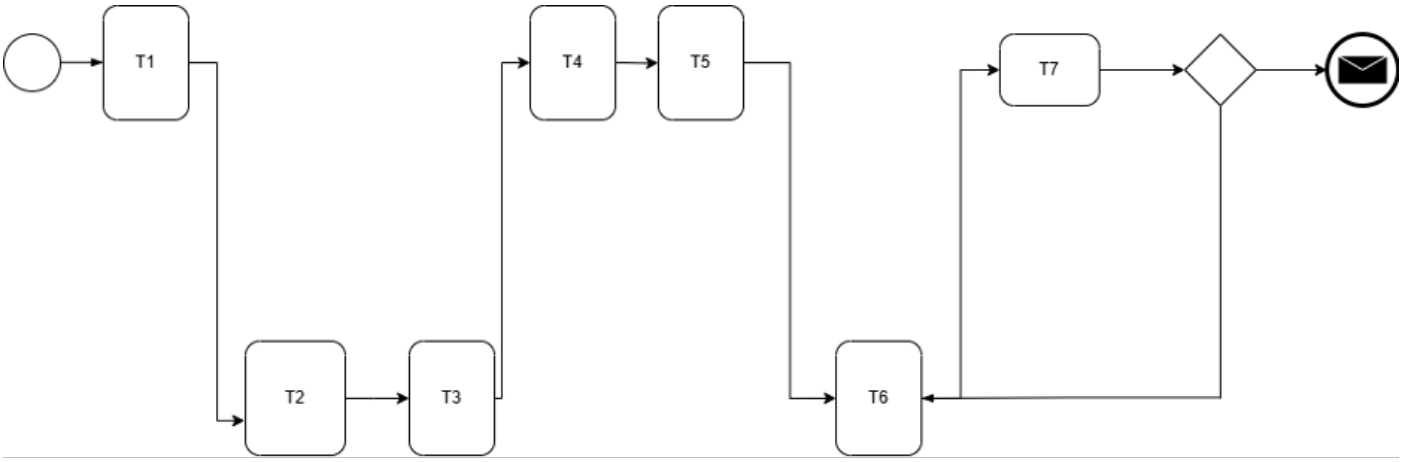


Figure 3.3: Booking Semantic Annotation

Visitor Information

The process begins when a user submits a query to the system. This query is stored in the Content Management System (CMS), and a background worker is assigned to retrieve the necessary information. If the retrieval is successful, the worker sends the response to the user immediately. If the system is unable to retrieve the information, the query is placed in a queue to be addressed by human staff.

Once a response is generated, either automatically or by human intervention, it is stored in the CMS for future reference. This ensures efficient handling of queries, with automation handling most cases and human support available for more complex queries.

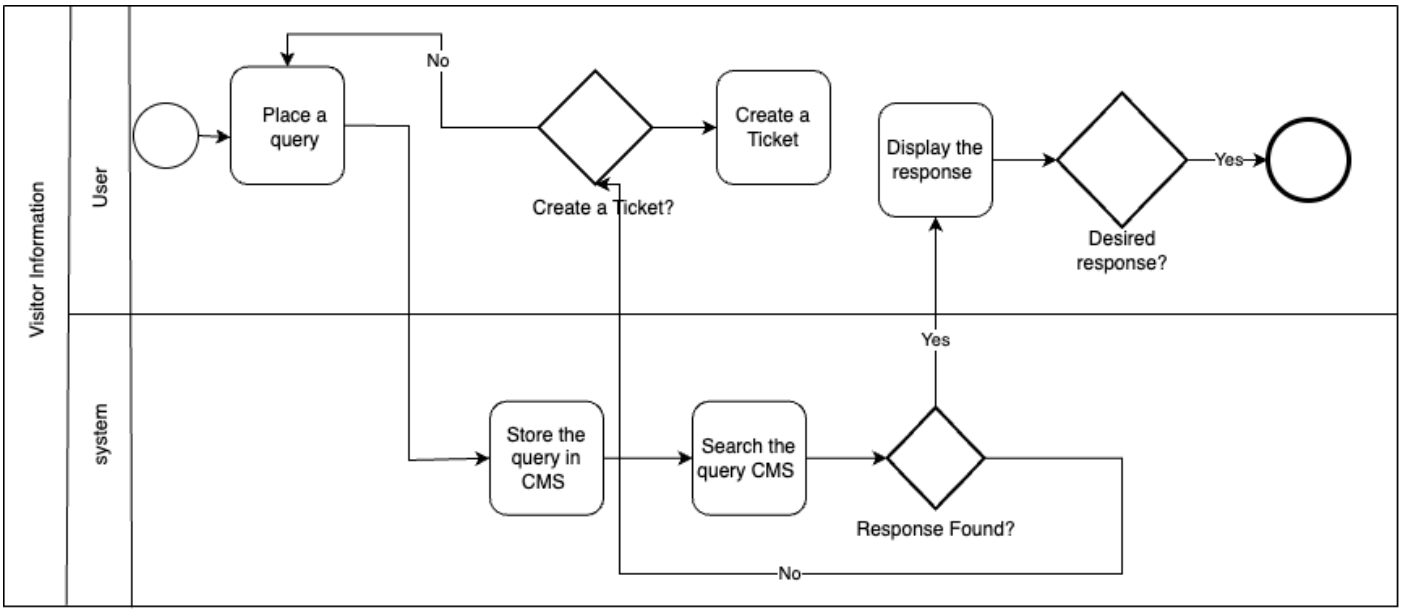


Figure 3.4: Visitor BPMN Diagram

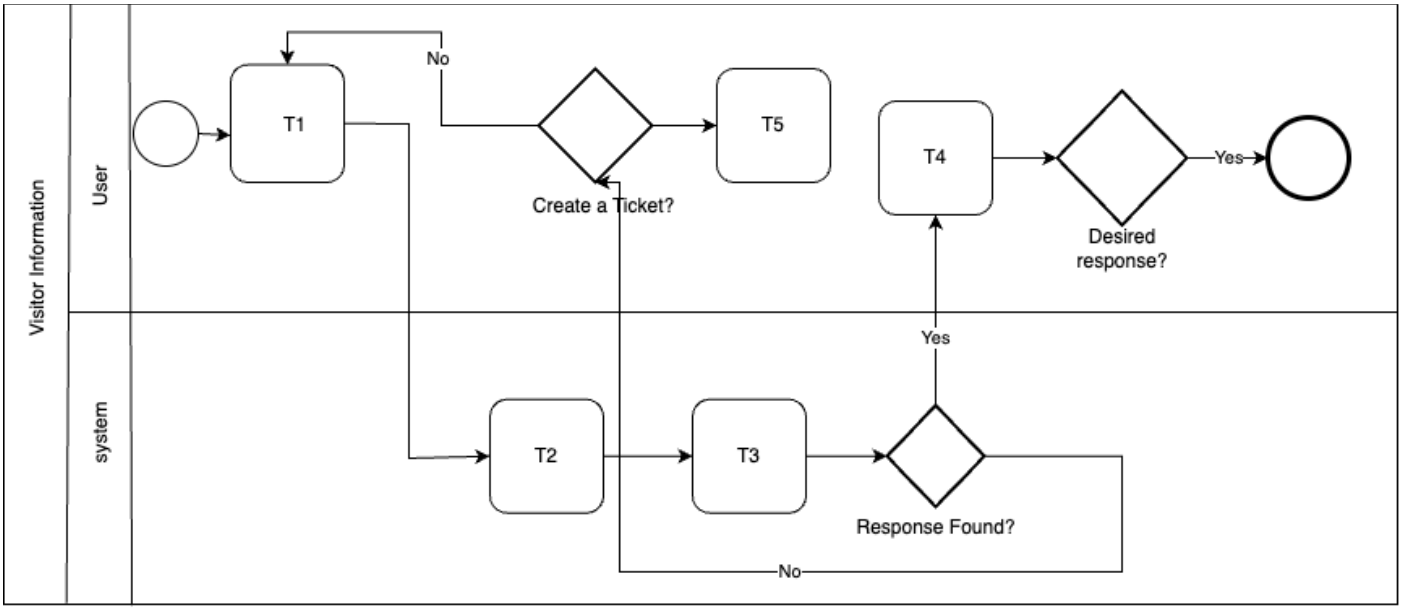


Figure 3.5: Visitor Semantic Annotation

The scenario labels and exclude sets for task T8 are as follows:

- T4 scenario (1): $\langle T1, \{T2\}, T3, T4, T5\rangle$
- T4 scenario (2): $\langle T1, \{T2, T3\}, T4, T5\rangle$
- T4 scenario (3): $\langle T1, T2, T3, T4\rangle$
- T4 scenario (4): $\langle T1, T2, \{T3, T5\}, T4\rangle$
- T4 scenario (5): $\langle T1, \{T2, T3, T5\}, T4\rangle$
- T4 scenario (6): $\langle T1, T2, \{T3\}, T4, T5\rangle$

Local Cuisine Offers BPMN

Local Cuisine provide two main processes for a Cuisine System: one for dine-in reservations and another for ordering items for delivery. The process starts with the customer opening the website and either selecting the dine-in option or placing an order for delivery. For dine-in, the customer selects the date, time, and number of guests, checks table availability, and, if a table is available, books it, receiving a confirmation email. For delivery orders, the customer selects items, adds them to the cart, checks for promotions or discounts, and the system calculates the total amount including tax and delivery fees. The customer then provides their delivery address and payment details, which the system verifies before placing the order and sending a confirmation message. Both processes involve key steps being communicated via email.

- T9: Scenario (1):

$$\langle t1, t2, t3, t4, t5, \{t6\}, t8, t9\rangle, \{\{t10, t11, t12, \ldots, t25\}\}$$

- T9: Scenario (2):

$$\langle t1, t2, t3, t4, t5, \{t7\}, t8, t9\rangle, \{\{t10, t11, t12, \ldots, t25\}\}$$

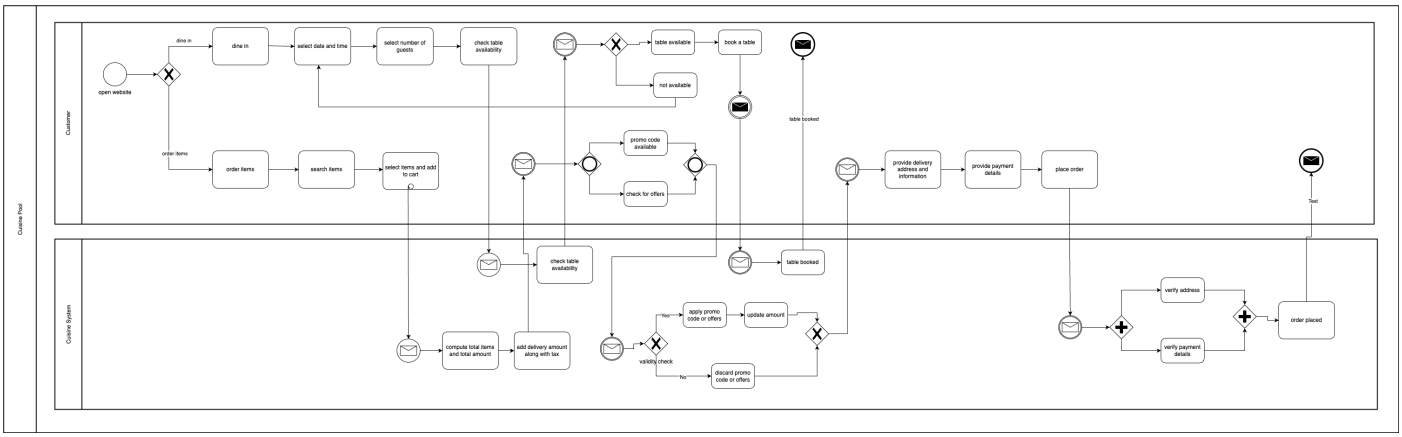


Figure 3.6: Cuisine BPMN Diagram

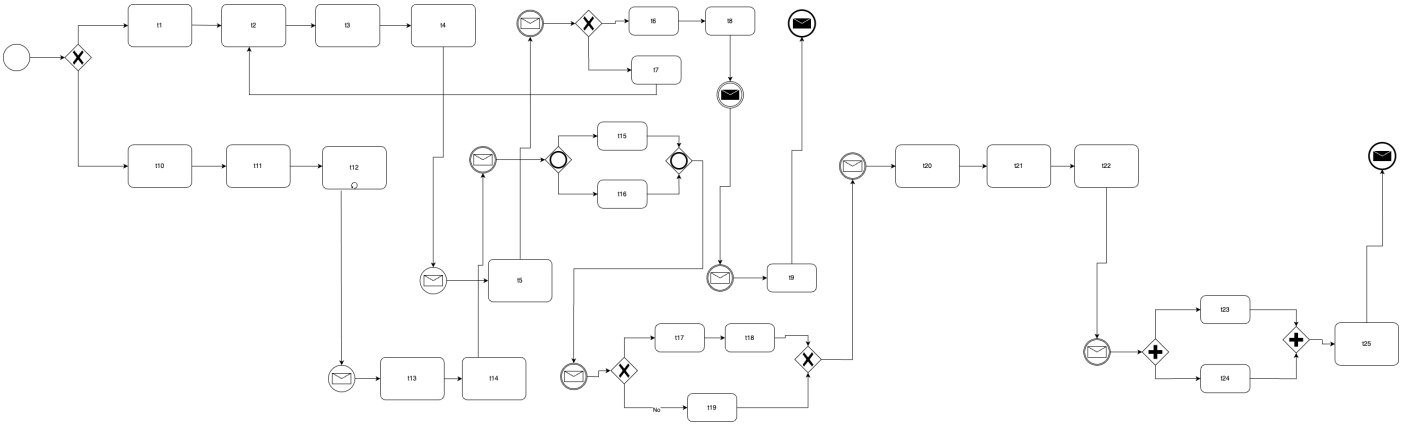


Figure 3.7: Cuisine Semantic Annotation

- T25: Scenario (1):

$$\langle t10, t11, t12, t13, t14, \{t15\}, \{t17, t18\}, t20, t21, t22, \{t23, t24\}, t25\rangle, \{\{t1, t2, t3, \ldots, t9\}\}$$

- T25: Scenario (2):

$$\langle t10, t11, t12, t13, t14, \{t16\}, \{t17, t18\}, t20, t21, t22, \{t23, t24\}, t25\rangle, \{\{t1, t2, t3, \ldots, t9\}\}$$

- T25: Scenario (3):

$$\langle t10, t11, t12, t13, t14, \{t15, t16\}, \{t17, t18\}, t20, t21, t22, \{t23, t24\}, t25\rangle, \{\{t1, t2, t3, \ldots, t9\}\}$$

- T25: Scenario (4):

$$\langle t10, t11, t12, t13, t14, \{t15\}, \{t19\}, t20, t21, t22, \{t23, t24\}, t25\rangle, \{\{t1, t2, t3, \ldots, t9\}\}$$

- T25: Scenario (5):

$$\langle t10, t11, t12, t13, t14, \{t16\}, \{t19\}, t20, t21, t22, \{t23, t24\}, t25\rangle, \{\{t1, t2, t3, \ldots, t9\}\}$$

- T25: Scenario (6):

$$\langle t10, t11, t12, t13, t14, \{t15, t16\}, \{t19\}, t20, t21, t22, \{t23, t24\}, t25\rangle, \{\{t1, t2, t3, \ldots, t9\}\}$$

Route Planning BPMN

In routing process a Traveller interacts with a system to find the best travel routes. The process begins with the customer entering the start and destination locations. The system calculates the optimal routes based on this input and may send a traffic update alert if necessary. The customer is then presented with route options, and they can either adjust the routes based on preferences or request additional information about the routes. Once the user finalizes their preferences, the route options are displayed, concluding the process.

Scenario Labels:

- Label 1: $\langle T1, T2, T3, T4\rangle$ (successful presentation of cost-effective options)
- Label 2: $\langle T1, T2, T3, T5\rangle$ (user notified of insufficient data)

Exclude Sets:

- For $T4$: $\{\langle T1, T2, T3, T5\rangle\}$
- For $T5$: $\{\langle T1, T2, T3, T4\rangle\}$

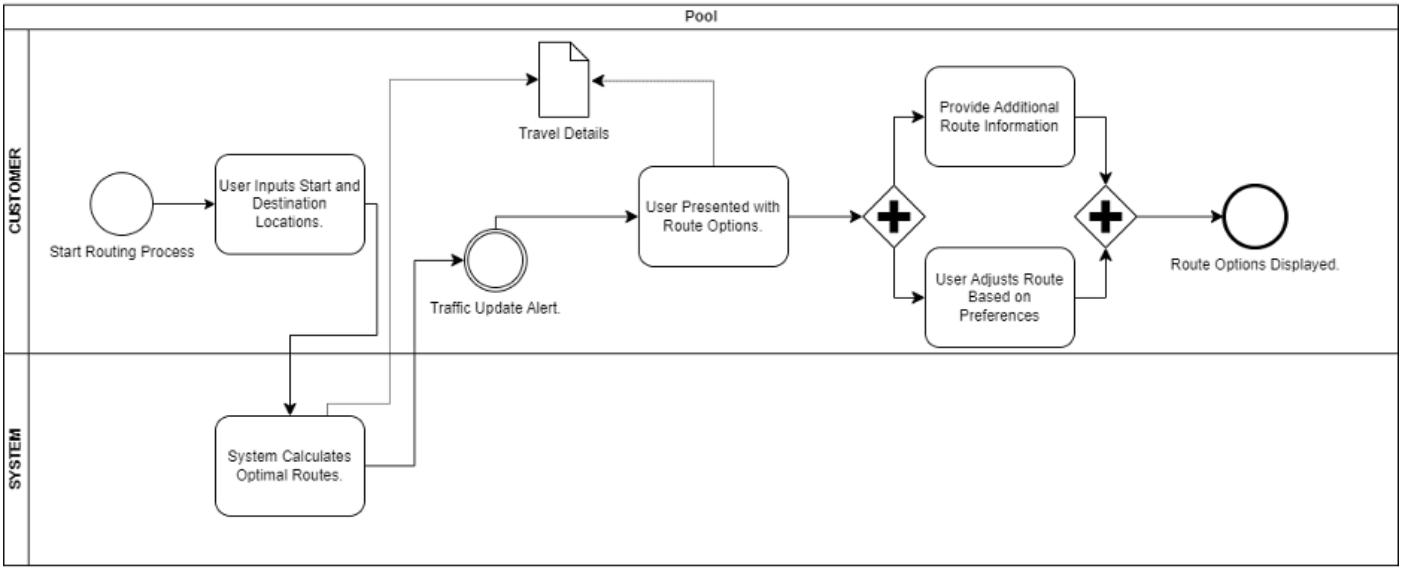


Figure 3.8: Route BPMN Diagram

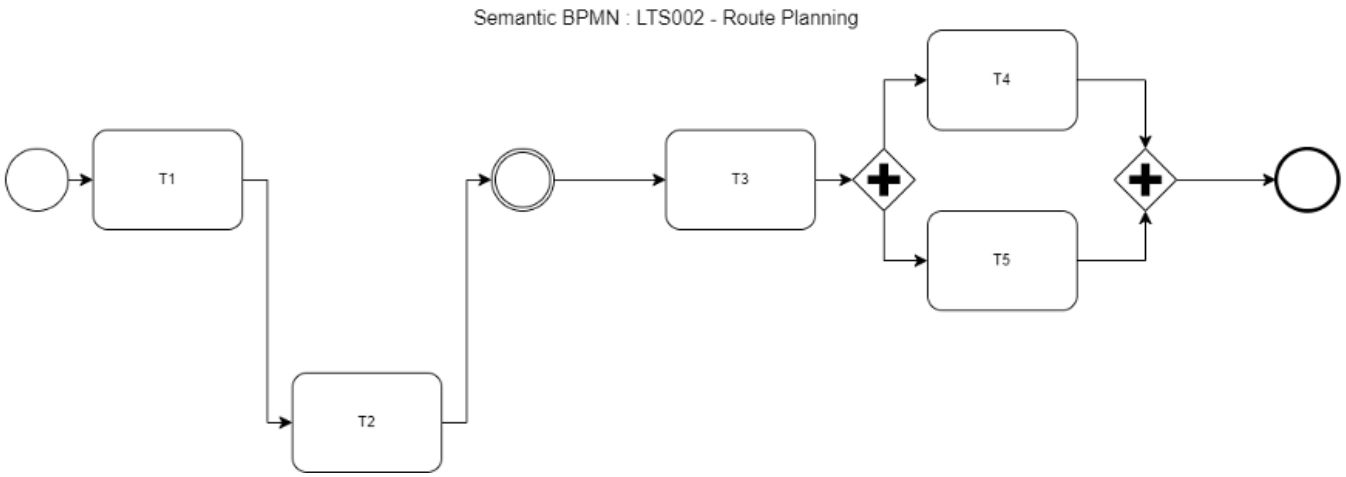


Figure 3.9: Route Semantic Annotation

Real-Time Traffic Monitor BPMN

For the Real-Time Traffic Monitor, the user inputs their current location and destination. The system retrieves traffic details from local sources in the current area and updates the traffic data accordingly. This updated information is then sent to the user. The traffic conditions are continuously analyzed and displayed in real-time, ensuring the user receives the most accurate and up-to-date information during their journey.

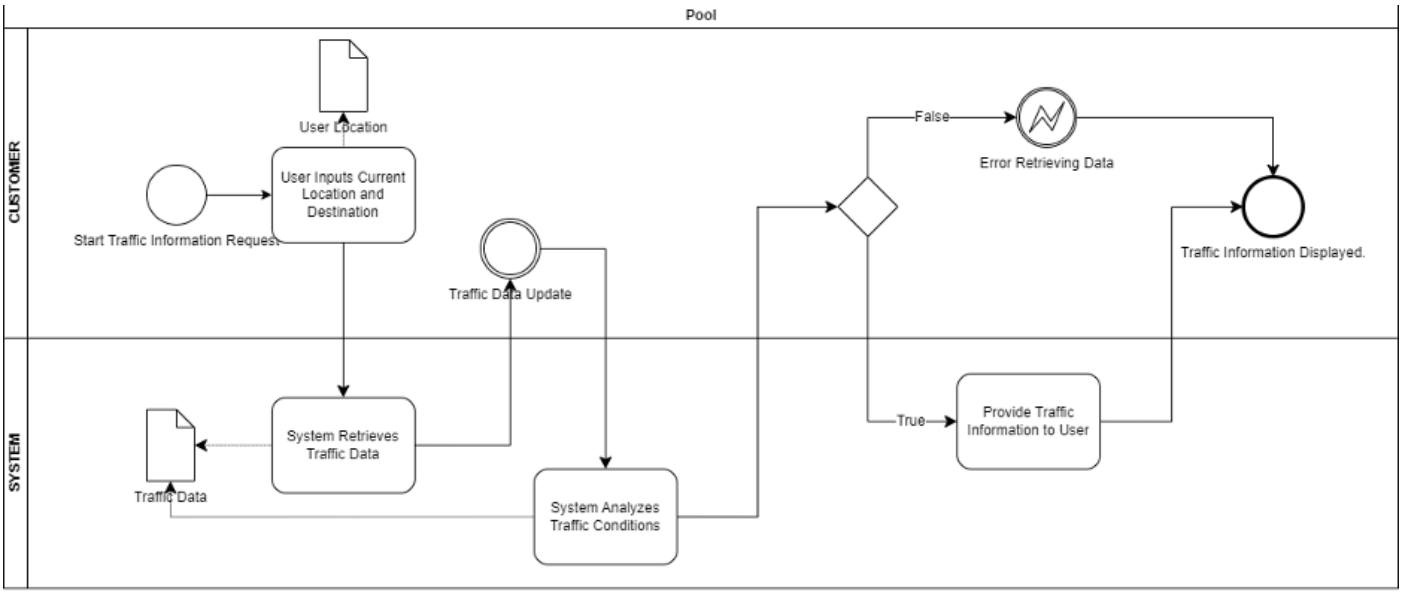


Figure 3.10: Traffic Monitor BPMN Diagram

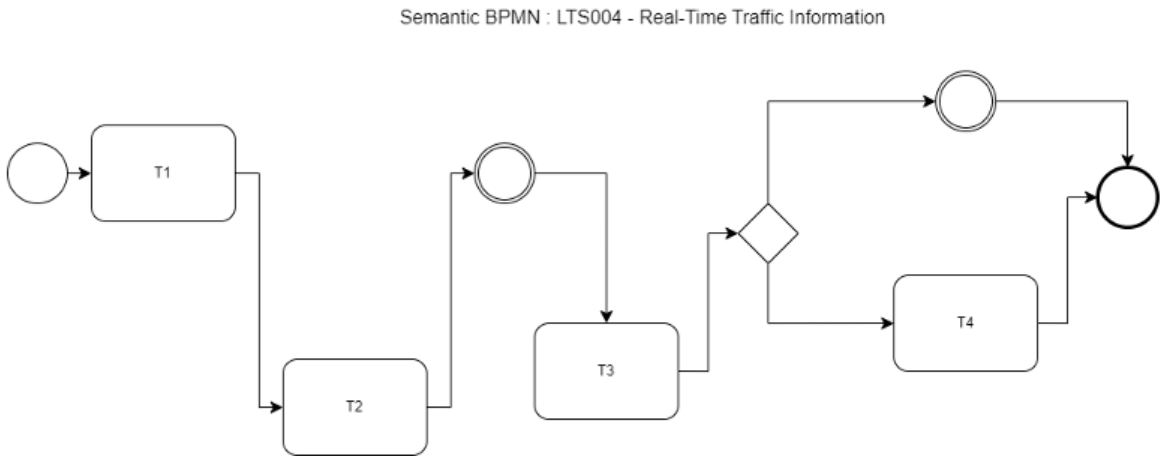


Figure 3.11: Real-Time Traffic Semantic Annotation

Scenario Labels:

- Label 1: $\langle T1, T2, T3, T4 \rangle$ (successful retrieval and display of traffic information)
- Label 2: $\langle T1, T2, T5 \rangle$ (error during data retrieval)

Exclude Sets:

- For $T4$: $\{\langle T1, T2, T5 \rangle\}$
- For $T5$: $\{\langle T1, T2, T3, T4 \rangle\}$

Part 4

Service Design and Analysis Using SoaML

This part focuses on service design and analysis by defining the Service-Oriented Architecture Modeling Language (SoaML) for microservices within our tourism ecosystem.

4.1 Service Architecture Overview

4.1.1 Booking Services

The Booking Service is a core component of the tourism ecosystem. Together with the Tour Service, it forms the foundation of the Customer Network Architecture. These services are depicted in Figures 4.1 and 4.2.

Service Architecture

The services mediate between the Customer (Traveller) and Airline (Partner) for flight booking, as well as between the Customer (Traveller) and Tour Provider for tour bookings.

Service Interface

The Customer is represented through the `TicketBuyer` and `Traveller` interfaces, which provide ports for ticket booking, tour booking, receipt generation, and review submission.

The Airline Provider is represented through the `TicketSeller` interface, which opens ports for processing ticket bookings and handling payments through the booking system.

The Tour Provider uses a generic service interface similar to the Airline Provider, utilizing the booking system and payment processor to manage tour bookings.

Service Contracts

The Booking Service and Payment Service serve as internal stakeholders, maintaining contracts with the Tour Service. The Tour Service also interacts with the Review Service, which provides review functionality.

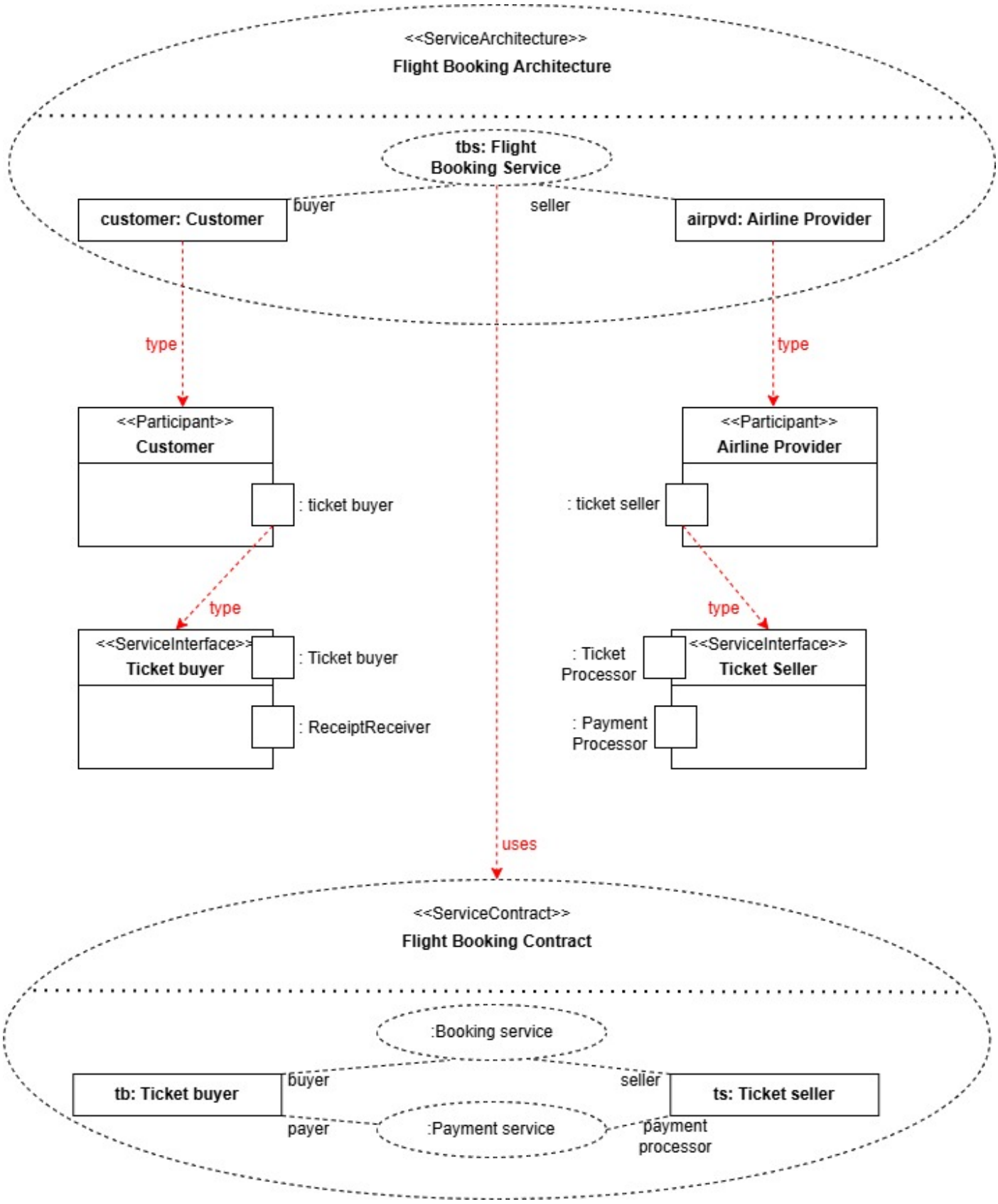


Figure 4.1: Booking Services SoaML Diagram

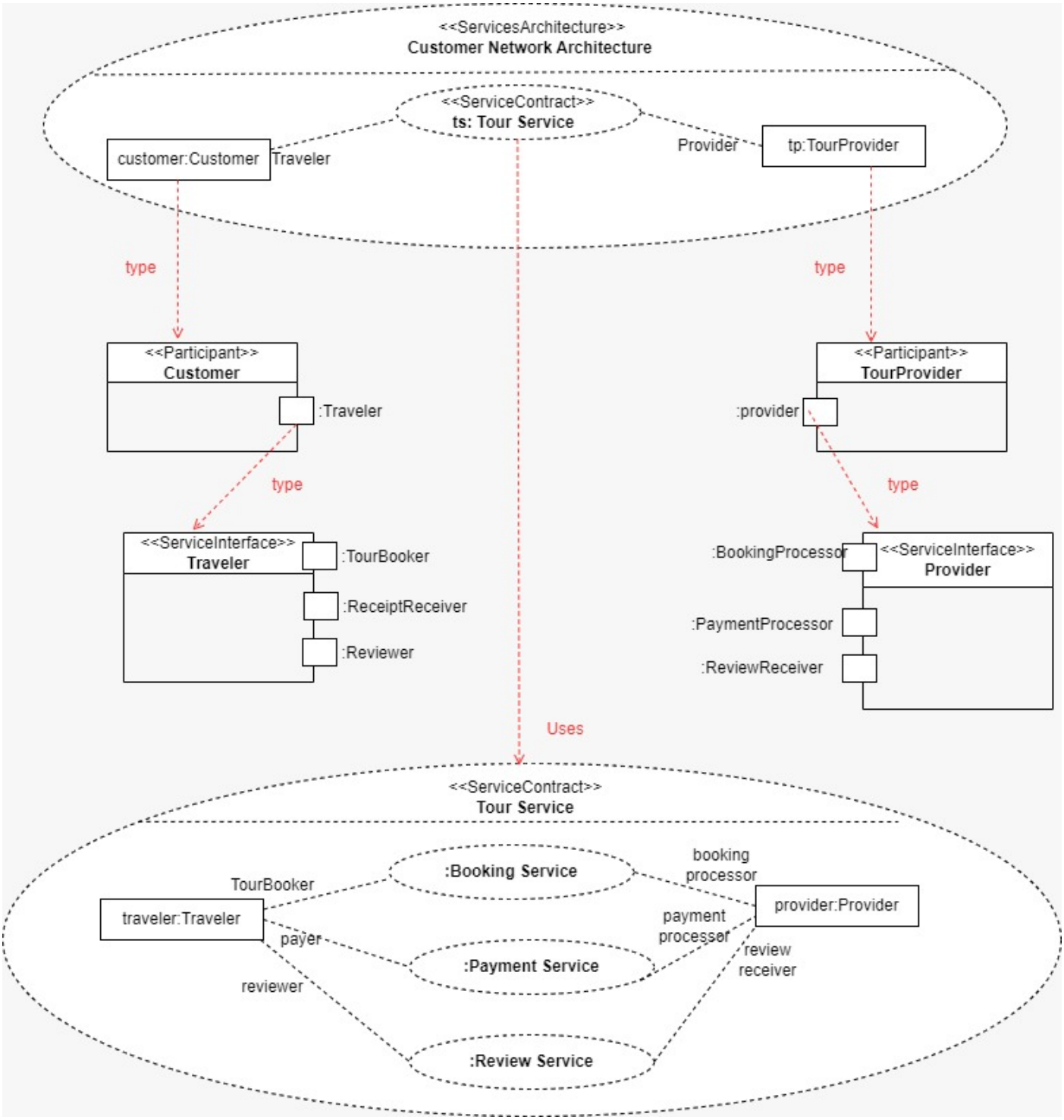


Figure 4.2: Tour Service SoaML Diagram

4.1.2 Restaurant Services

The Restaurant Service handles order and delivery processes for Local Cuisine. Together with the Billing Service, it forms the foundation of the Order Handling Architecture, as depicted in Figures 4.3 and 4.4.

Service Architecture

The services mediate between the Buyer and Product Supplier for order processing and delivery, as well as between the Buyer and Billing Handler for payment processing.

Service Interface

The Buyer is represented through the `OrderItem` interface, which provides ports for placing orders, tracking deliveries, and receiving billing information.

The Product Supplier is represented through the `DeliverItem` interface, which handles delivery requests and confirms order fulfillment.

The Billing Handler manages billing and payment processing via the `BillOrder` interface.

Service Contracts

The services maintain contracts with the Order Service and Billing Service. The Delivery Service interacts with the Product Supplier to ensure accurate and timely order fulfillment.

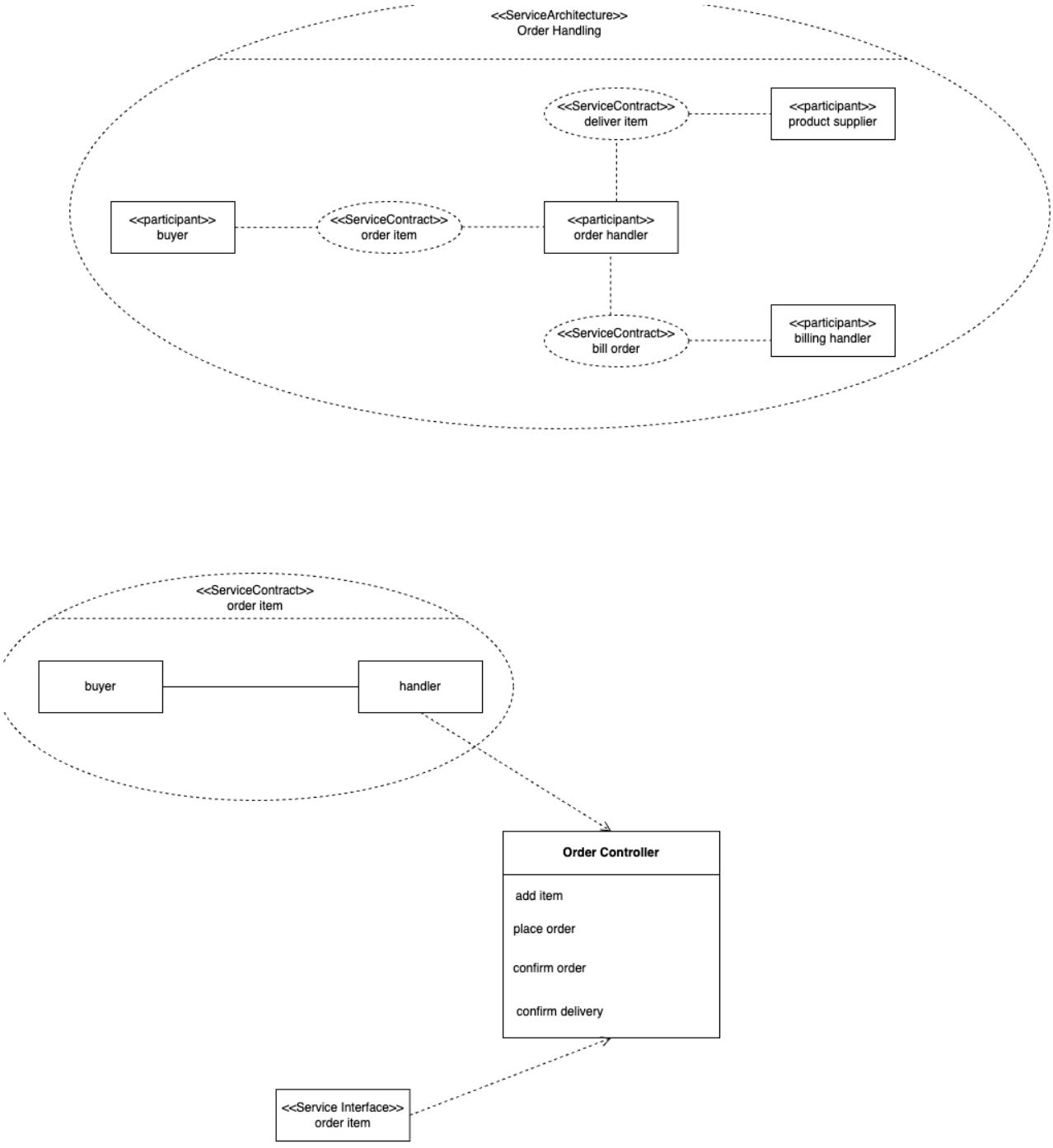


Figure 4.3: Cuisine Services SoaML Diagram

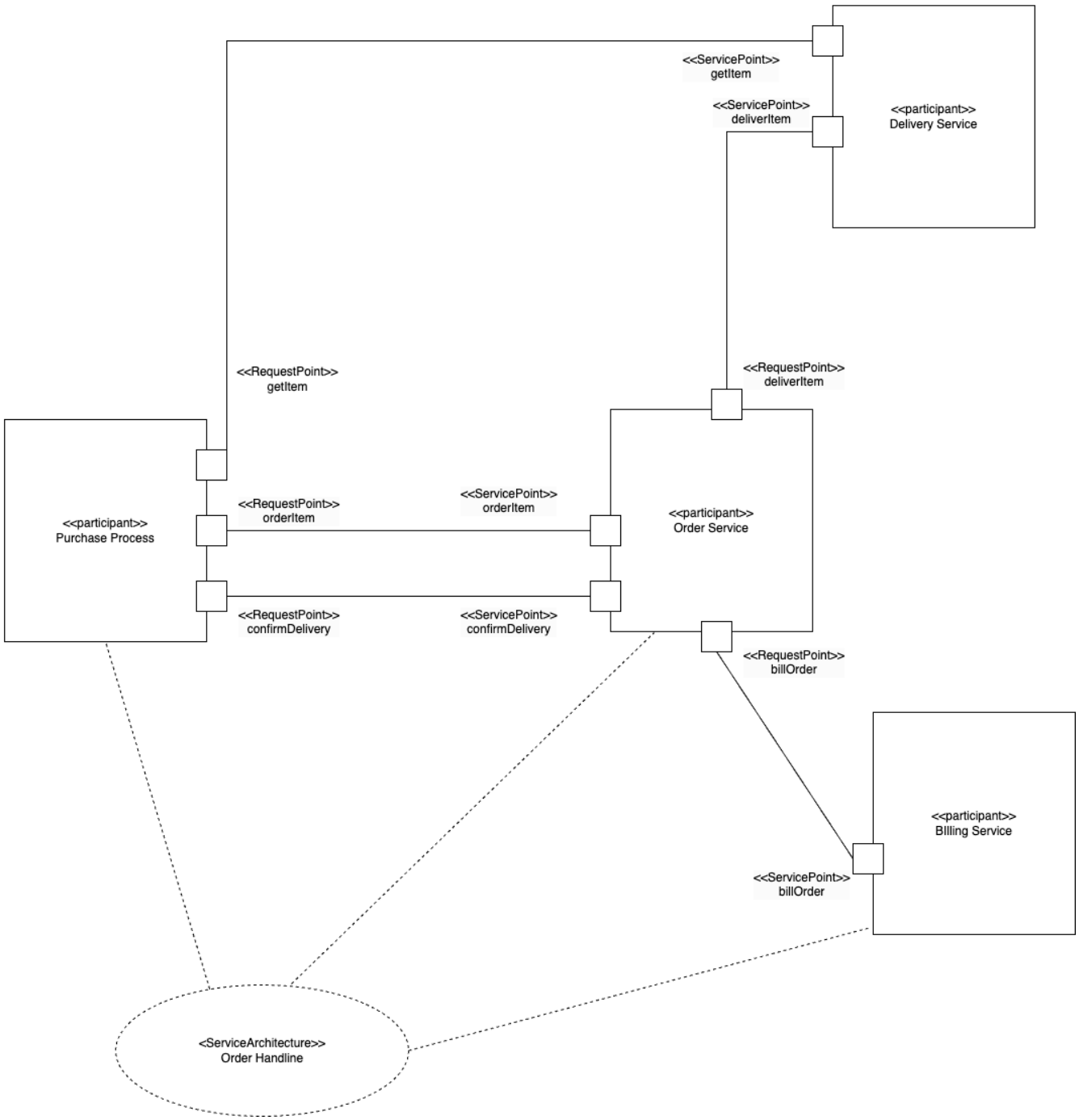


Figure 4.4: Cuisine Services SoaML Diagram

4.1.3 Transport Services

The Transport Service provides functionalities such as rental information, route planning, and real-time traffic updates.

Service Architecture

These services are request-based, processing information from various sources to offer optimal routes, traffic updates, and rental services to users.

Service Interface

- **Route Planning Service**:
  - The **Customer** is represented by the `Routing Request` interface, providing ports for route requests, availability checks, and notifications.
  - The **Routing System** is implemented via the `Routing Processing` interface, which handles route calculation and updates.
- **Real-Time Traffic Information**:
  - The **Customer** interacts via the `Traffic Request` interface, which allows requests for traffic updates and location checks.
  - The **Traffic Monitoring System** is represented by the `Traffic Monitoring` interface, which processes requests and provides alternative routes.
- **Transport Rentals**:
  - The **Renter** uses the `Rental Request` interface, with ports for submitting rental requests, checking vehicle availability, and receiving notifications.
  - The **Rental System** processes requests and payments through the `Rental Processing` interface.

Service Contracts

The services maintain contracts with the Route Planning Service, Traffic Monitoring Service, and Transport Rental Service. These services interact with external providers and monitoring systems to ensure accurate information and timely updates for users.

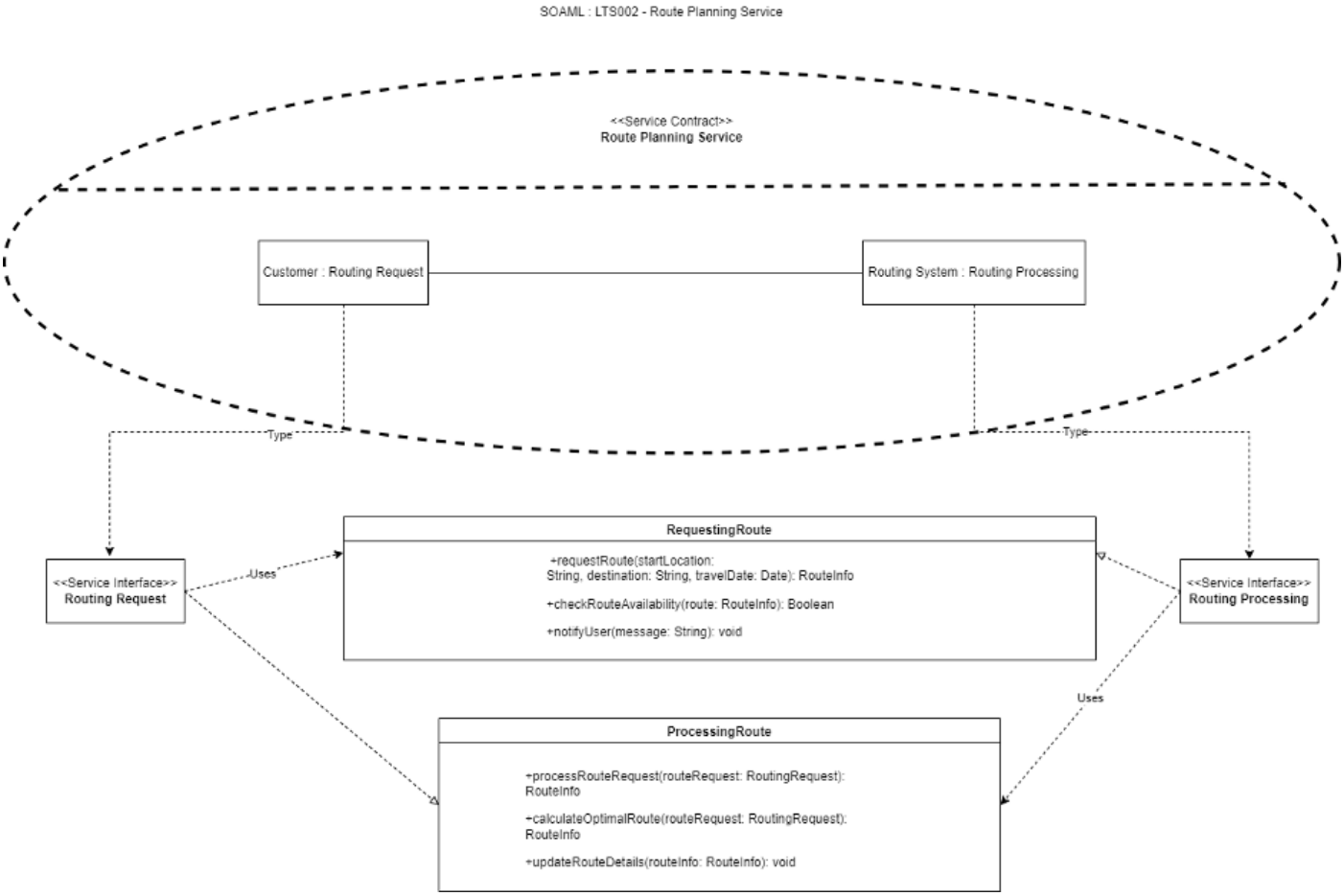


Figure 4.5: Route Planning SoaML Diagram

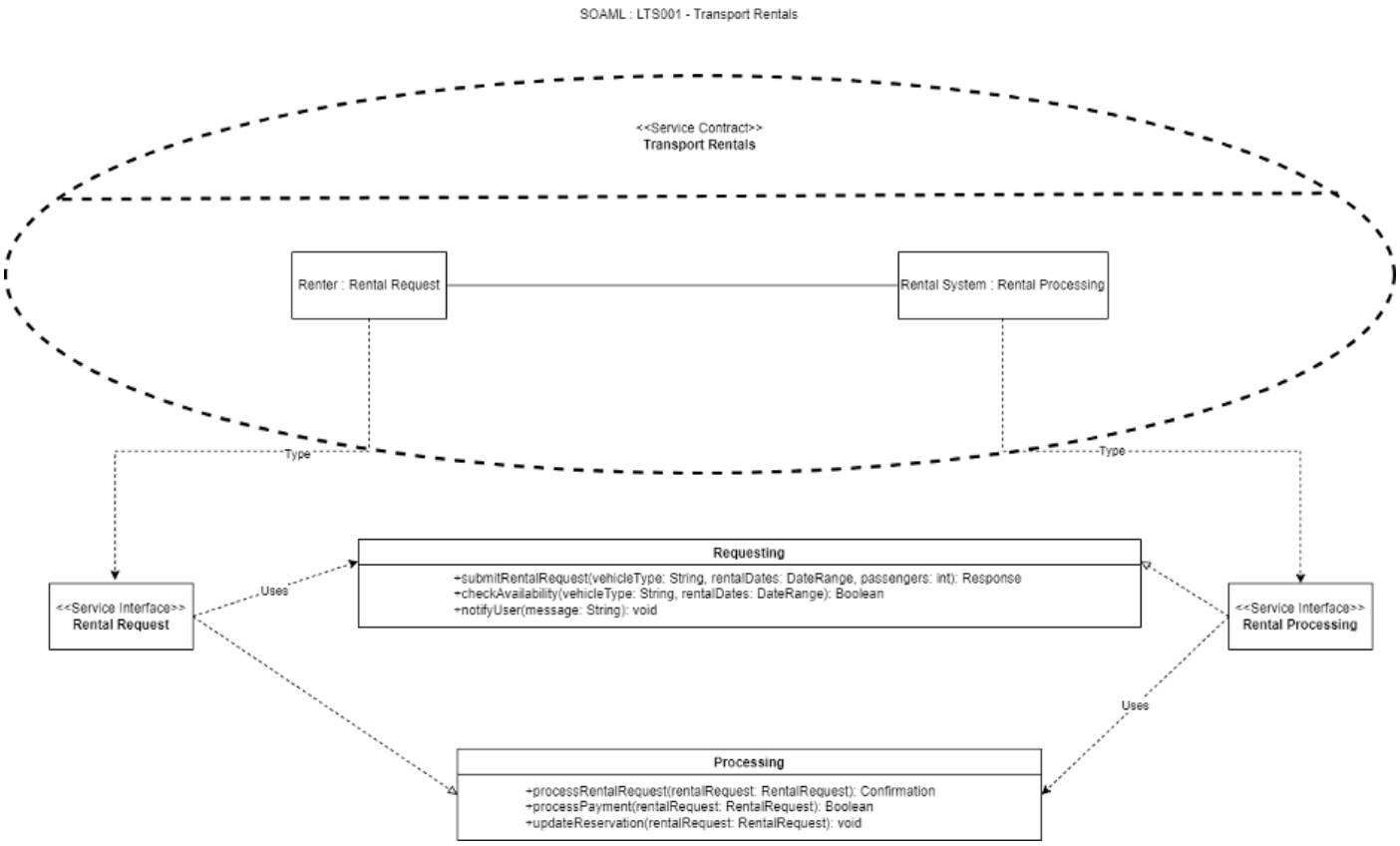


Figure 4.6: Rental Services SoaML Diagram

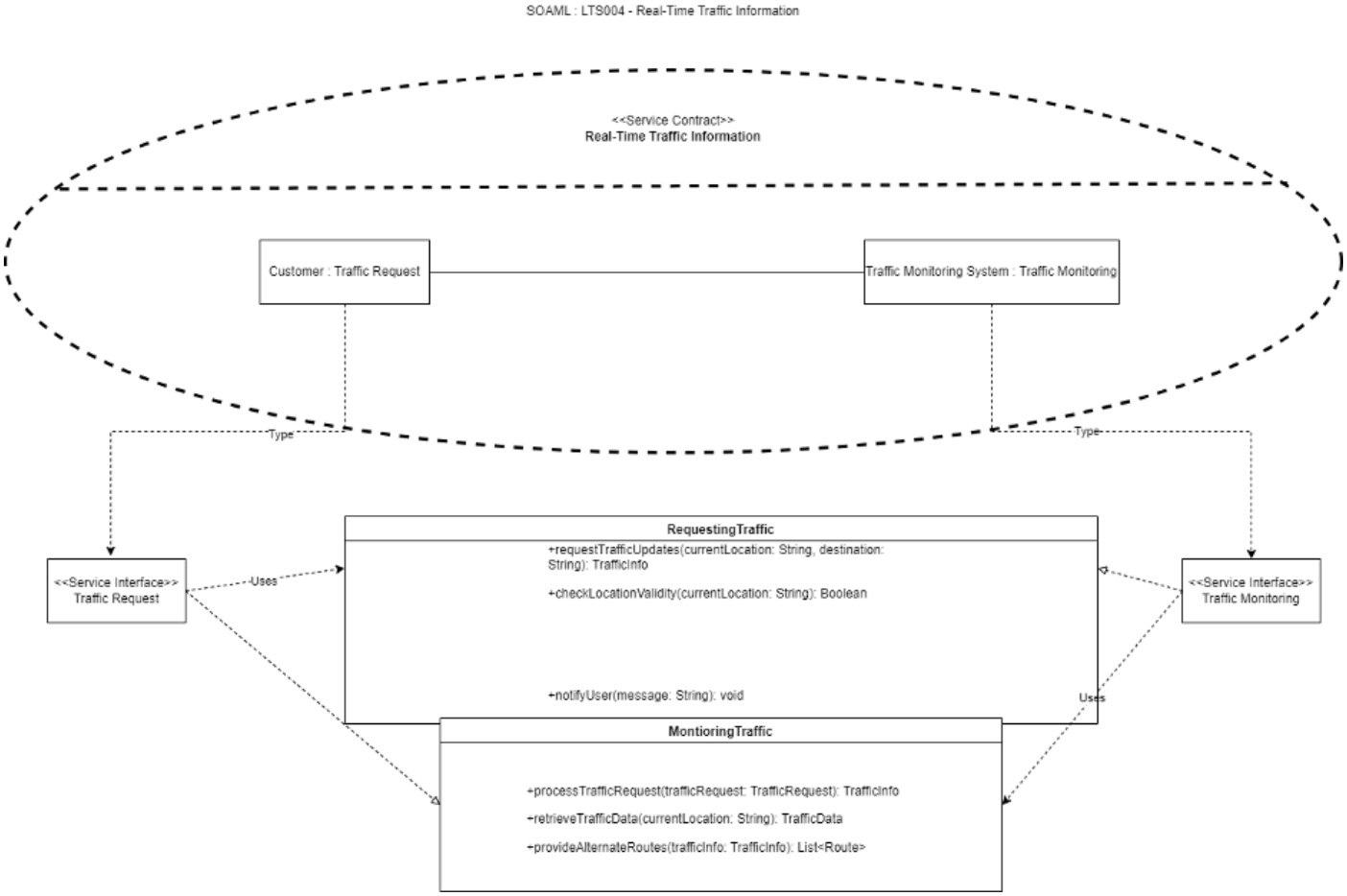


Figure 4.7: Real-Time Traffic SoaML Diagram

4.1.4 Information Services

Information Services provide key functionalities for tourists and users through structured inquiry and notification systems, offering destination data and real-time alerts.

Service Architecture

These services are both inquiry-based (for tourists) and request-based (for notification users), managing incoming queries to generate reports and provide notifications, ensuring timely delivery of information.

Service Interface

- **Tourist Inquiry Service**:
  - The **Tourist** uses the `Requestor` interface to submit inquiries and request reports.
  - The **Request Handling System** is implemented via the `Inquirer` interface, processing requests and retrieving necessary data.
- **Information Provider Service**:
  - The **Information Provider** handles queries from tourists via the `Provider` interface, generating reports based on the information requested.
  - The **Information System** processes tourist queries through the `QueryProcessor` interface, delivering relevant information.
- **Notification Subscription Service**:
  - The **User** subscribes to notifications through the `Subscriber` interface, receiving alerts via the system.
  - The **Subscription System** processes subscriptions using the `NotificationSubscriber` interface.
- **Event Notification Service**:

- The **Event Notifier** interacts via the `Notifier` interface, processing notifications for subscribed users.
- The **Notification System** is implemented via the `SubscriptionHandler` and `NotificationGenera` interfaces, generating event-based alerts.

Service Contracts

The Information Services maintain contracts with the Tourist Inquiry, Information Provider, Notification Subscription, and Event Notification Services. These work together to ensure accurate responses and timely notifications for tourists and users.

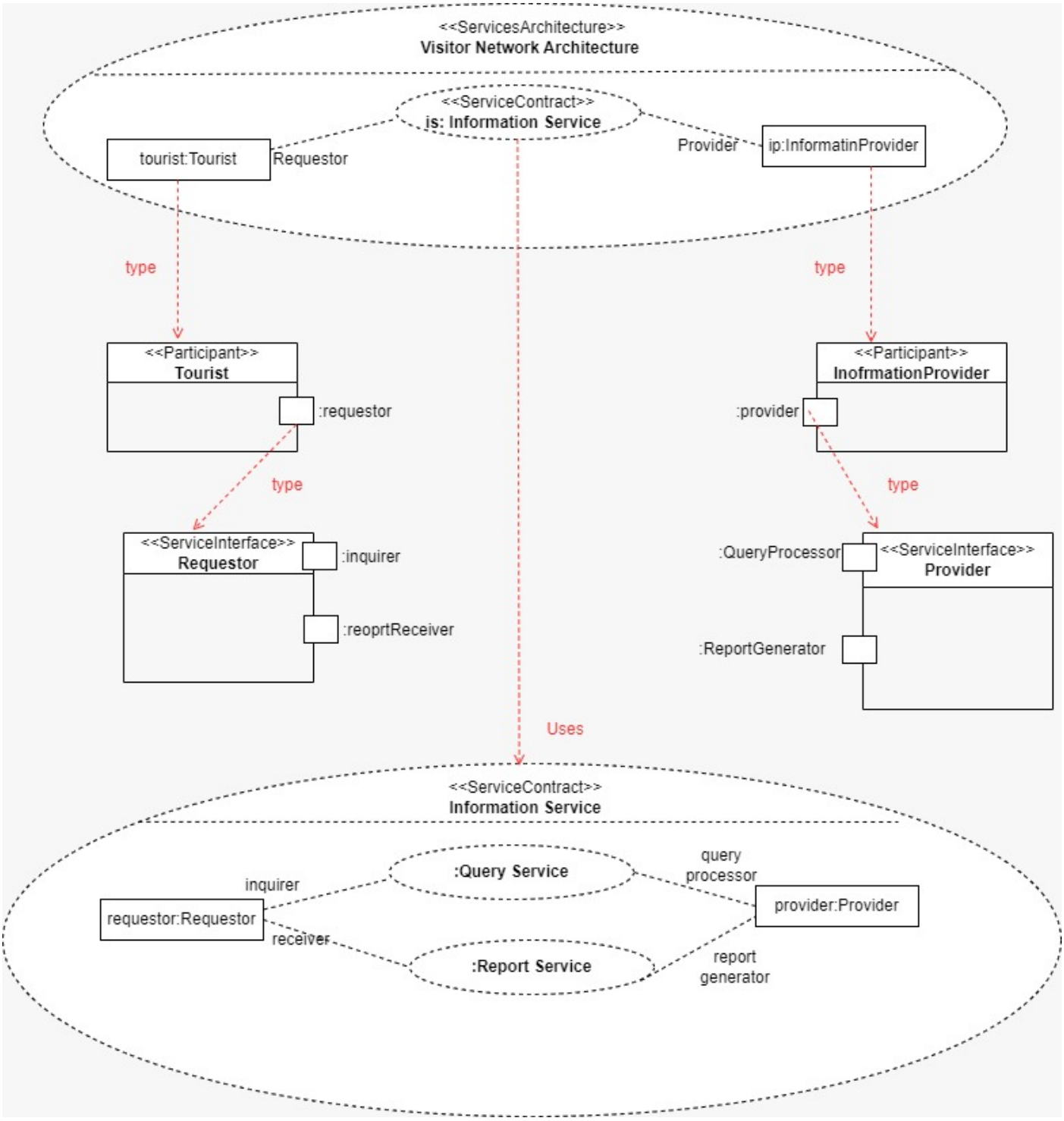


Figure 4.8: Visitor Network SoaML Diagram

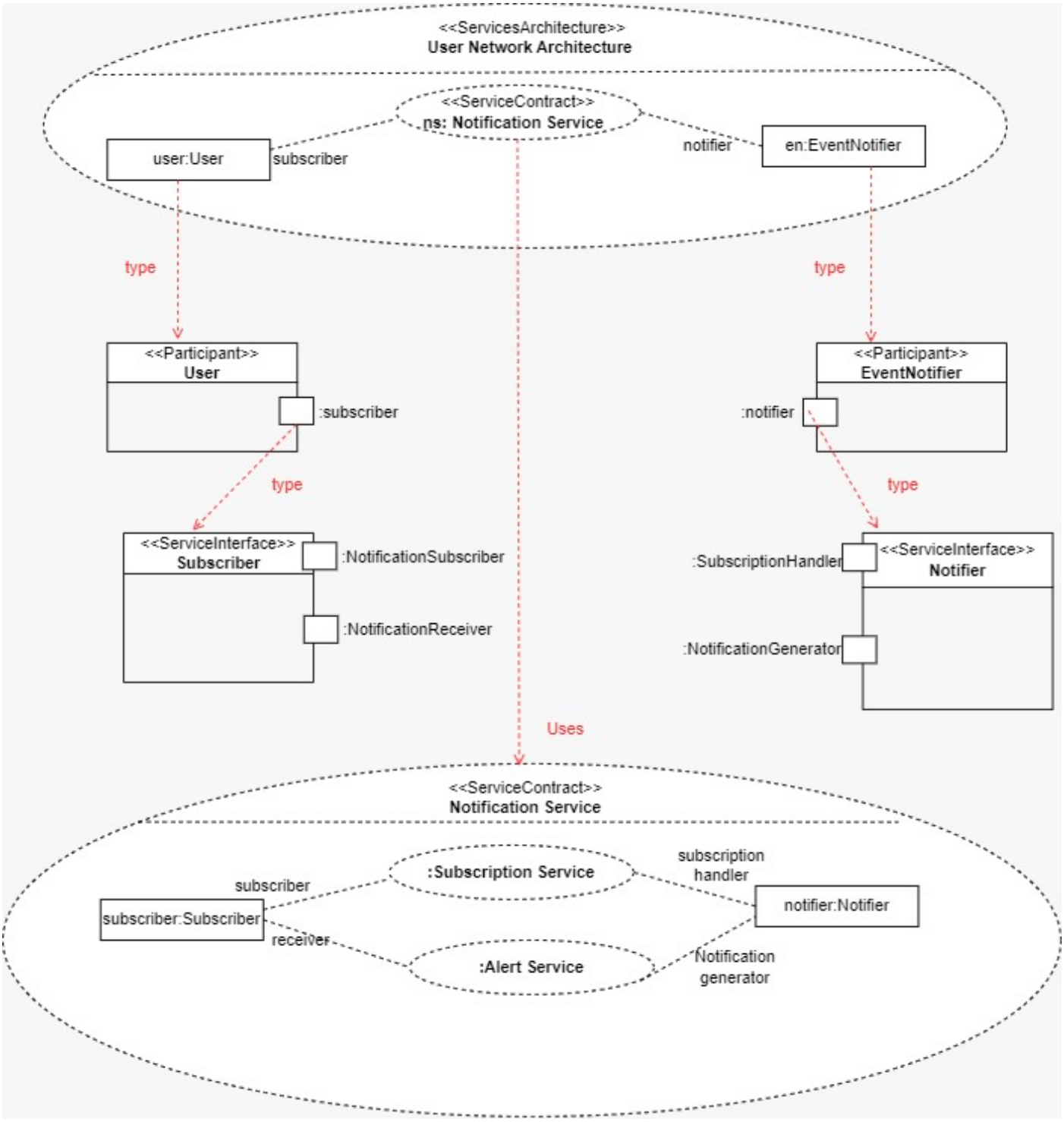


Figure 4.9: Notification Service SoaML Diagram

Part 5

Microservices Design and Implementation

In this section, we will discuss the design and implementation of our microservices for the tourism ecosystem.

5.1 Service Selection

For the purpose of this report, we have selected the following services from the booking system: **Flight Booking**, **Room Booking**, **Tour Booking**, and **Cuisines**.

The rationale behind this selection lies in the fact that these services represent the core components of the tourism system, serving as essential touchpoints for travelers. Additionally, they form the foundation of the platform's revenue model, making them critical to the overall success of the tourism ecosystem.

5.2 Service Design

The services are designed using a microservices architecture, with each service operating independently and implemented through a RESTful architecture. This design promotes modularity, scalability, and ease of integration.

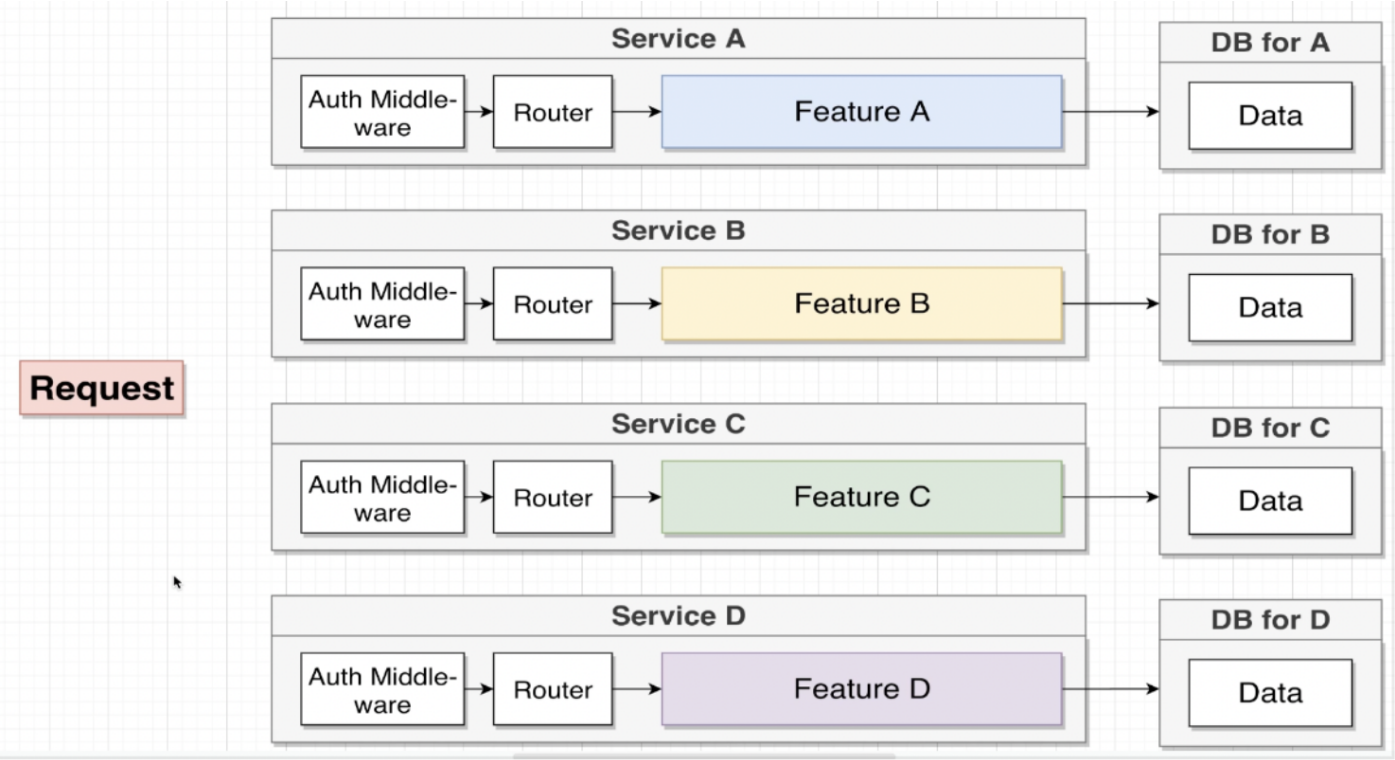


Figure 5.1: Service Design

Database Design

Each microservice is designed with a dedicated database to ensure data separation and service independence. Below are the database schemas we have implemented for the respective services:

- **Accounts and Tour Database** - Figure 5.2 - **Flight Booking Database** - Figure 5.3 - **Accommodation Database** - Figure 5.4

5.3 Service Implementation

5.3.1 Technology Stack

We have selected **C#**, **Java**, and **TypeScript** for the development of our services.

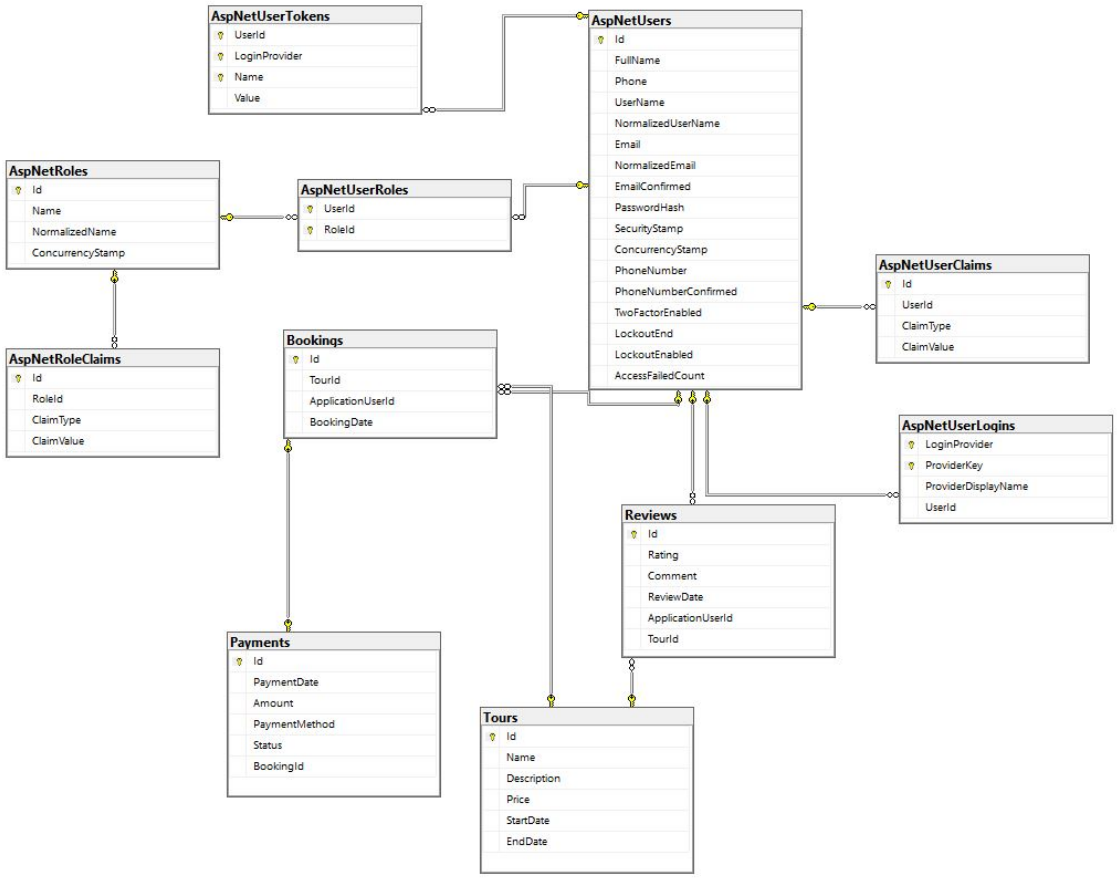


Figure 5.2: Accounts and Tour Database Schema

5.3.2 Service Architecture

The architectures of the Booking and Cuisine services are illustrated in Figures 5.5a and 5.5b, respectively. Each service interacts with a payment service, after which a message is sent to a queue. This message is subsequently processed by the Booking/Order service, which handles the necessary external calls.

5.3.3 Service Communication

The services will communicate using REST protocols. The API gateway will serve as a layer between the frontend and the backend, providing control over service exposure and effectively routing requests.

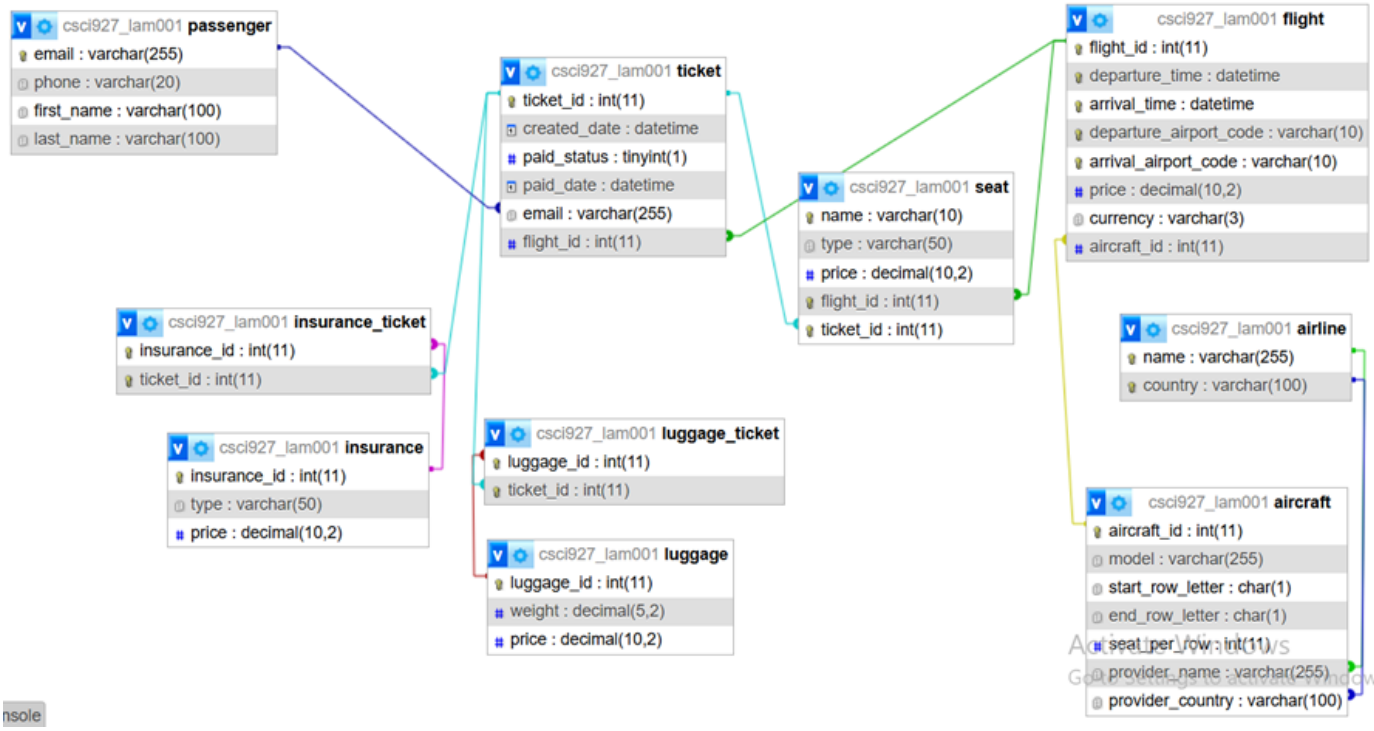


Figure 5.3: Flight Booking Database Schema

5.3.4 Deployment Strategy

The deployment strategy involves utilizing Google Cloud Run for hosting our microservices. This managed platform allows us to automatically scale our applications based on demand, ensuring efficient resource management and seamless updates.

5.3.5 Failures, Circuit Breakers, and Logging

In a microservices architecture deployed on Google Cloud Run, managing failures is crucial for maintaining high availability and resilience. To address this, we implement circuit breakers that monitor service interactions. When a service call fails repeatedly, the circuit breaker trips, preventing further requests to the failing service and allowing it time to recover.

Google Cloud Run's auto-scaling capabilities complement this strategy by automatically adjusting the number of service instances based on incoming requests. In the event of a failure, Cloud Run can seamlessly redirect traffic to healthy instances, minimizing downtime.

Additionally, fallback mechanisms provide alternative responses or cached data when a service is

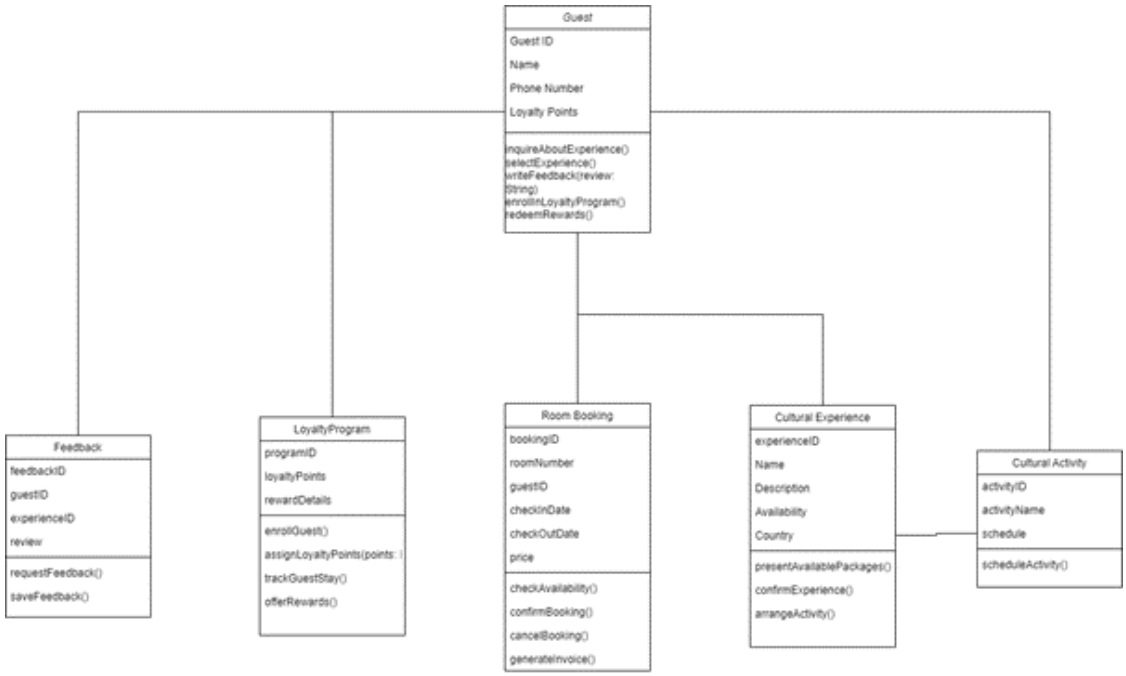
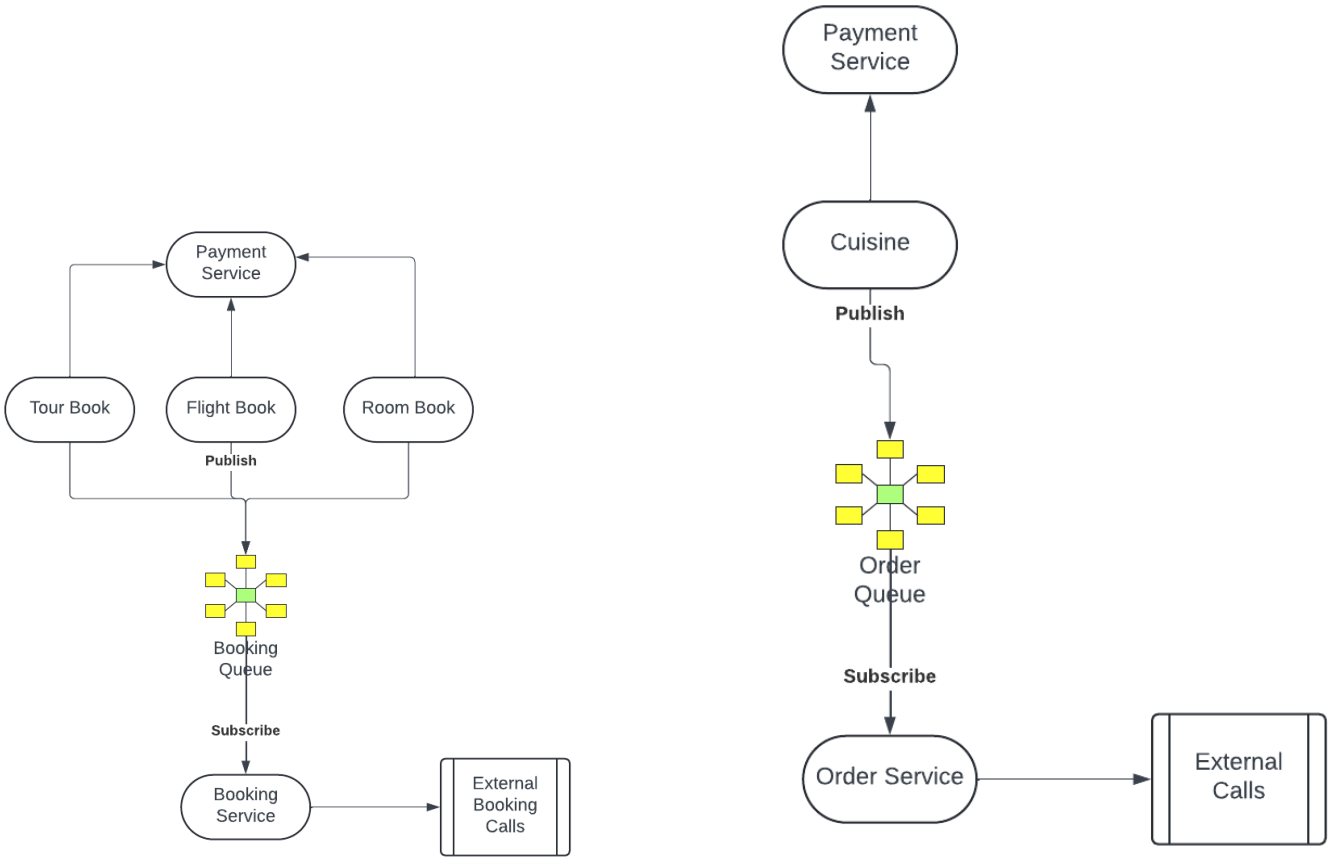


Figure 5.4: Accommodation Database Schema

unavailable. This ensures that even during outages, users can still access essential functionalities, enhancing the overall user experience. Leveraging Google Cloud's monitoring and logging tools allows us to analyze failure patterns, enabling continuous improvement of our service reliability.



(a) Booking Architecture (b) Cuisine Architecture

Figure 5.5: Architectures of Booking and Cuisine Services

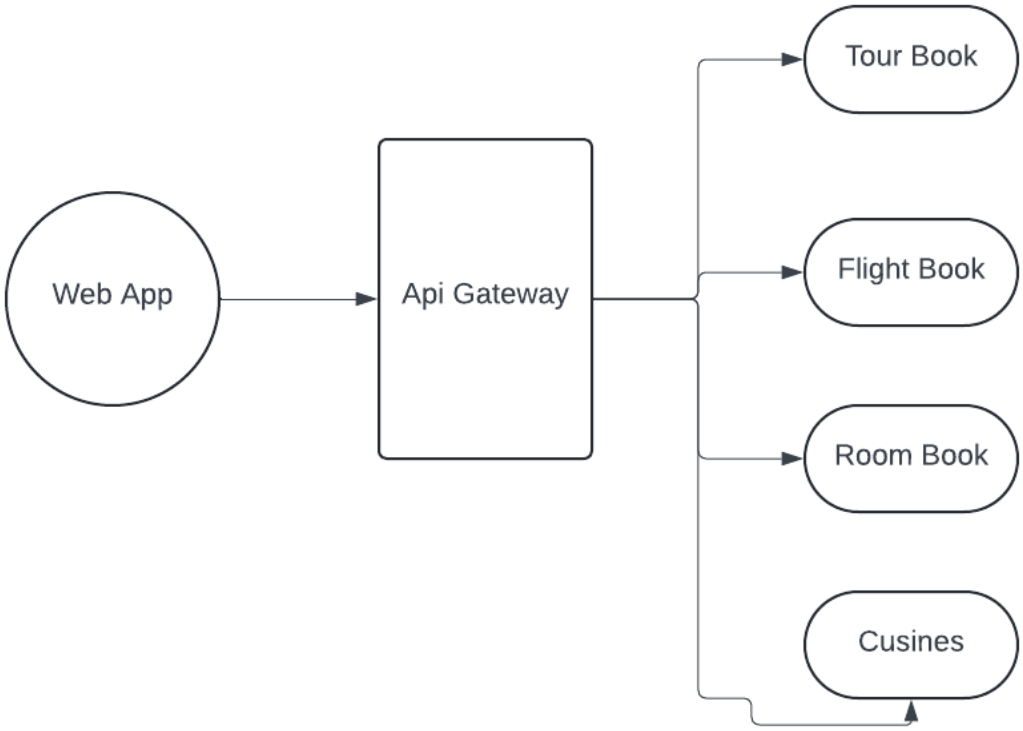


Figure 5.6: API Gateway

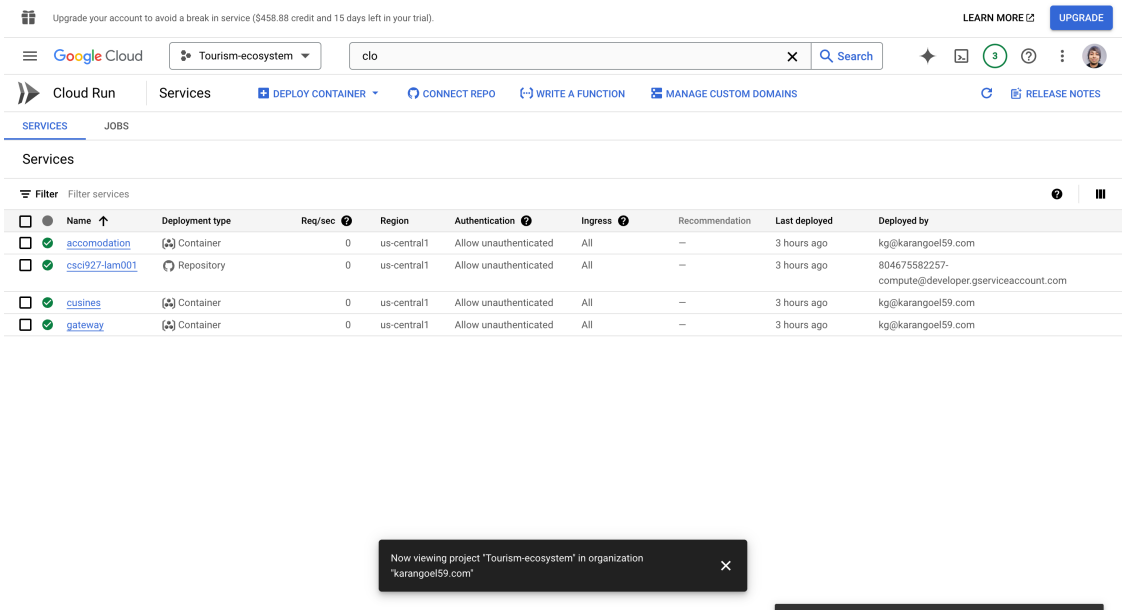


Figure 5.7: Deployment

Part 6

Service/process analytics

6.1 Busienss Processes

To begin the process of mining and analyzing business operations, we first generated event logs by simulating the execution of three key business processes within the service platform:

- Flight Booking
- Room Booking
- Tour Booking

These processes were chosen because they are in the middle of the tourism platform, they interface with other internal and external systems, and because it was believed that results from these processes would give insightful information on the performance and dependability of the entire system.

Steps Taken:

Simulated Execution:

Following the description of the three business processes, a custom script was then written in order to mimic the end-to-end running of the three processes. Each execution represents advanced usage scenarios with the system, and included, for example interactions with Payment gateways and tour management third-party systems.

Logged Details:

Each time the process was executed the script generated a structured log that included the following information:

Case ID:

Special reference number for every occurrence of the business process. This was useful for monitor and evaluate case flows to the different stations for the purposes of analysis.

Start Time and Complete Time:

These stamps actually pointed out the time taken to perform any activity. Time data was required for analyzing the bottlenecks and performance issues turns into non value added activities.

Activity:

The job role that is involved in performing each of the step in the process like "Check Token Validity", "Reservation Success", "Flight booking" etc. Resource: This logged the system or service that was supposed to perform the task. For instance, the resources could be the flight booking system, payment gateway or the reservation system.

Role:

This field drew the line between the activities that involve user interactions and interactions that are system driven, where the system generates an email as a confirmation of a booking, or to alert the user of payment confirmation among others.

The participant-side logs offered a valuable data source around 300 entries, which covered multiple occurrences of every business process. This vast dataset ranging from various users' activities to their interactions with the system formed the input for the subsequent process mining.

Thus, the generated log data were in CSV format for easy compatibility with process mining tools such as Disco. It was also possible to consider different logs that are not necessarily completely identical and thus reflect different actions by different users in different system conditions, such as increased load or simultaneous transactions.

6.2 Process Mining

The next step involved process mining of the generated event logs to find the real flow of activities, the measures of performance, and various issues relating to efficiency or otherwise of the applied process. The purpose of this step was to establish a working knowledge as to how the system performed the business processes and more importantly, areas where improvement could be made concerning the efficiency and effectiveness of the system as it used key organisational resources.

Steps Taken:

Importing Logs into Disco:

Disco is a tools specifically designed for process mining, and we uploaded the 300-event log CSV file into this tool. With Disco in place, the log data was analyzed automatically as well as real models illustrating the sequence of the business processes in each case were generated.

6.2.1 The analysis of the process of Room Booking Service

Case ID: The identification of every experience of room booking is assigned a case ID tag, for example, Booking 1, to refer to a particular attempt.

Task IDs: These are the unique ones They include T2- select room and T6-confirm payment.

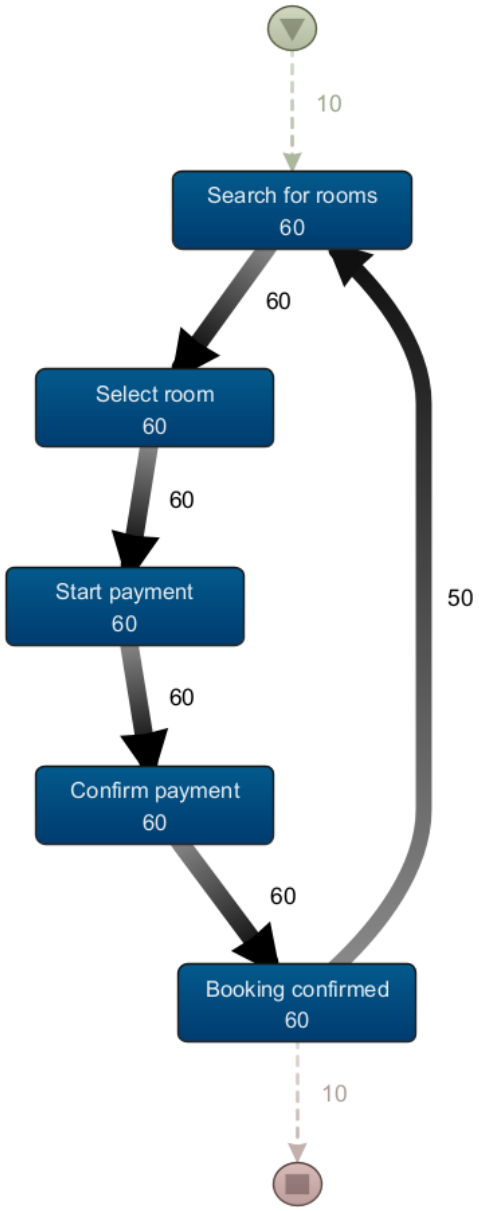


Figure 6.1: Room Booking Service

Time Stamping and Resource Utilisation

Timestamps: On logs, we are able to measure the time each task took, thus, be able to determine any time wastages or phase constrains. For example, such a task as room search or payment processing can be performed in different time periods depending on the load on the system.

Resource Utilization: This shows who of the systems or personnel are to handle which tasks. For instance, there is a query to external data sources when searching for a room, while payment confirmation will require an interaction with the payment system.

Work Choice Incorporation

Using a process map of the room booking service, the actual steps of the booking process are demonstrated. Several discrepancies from the initial BPMN model have been observed:

Multiple Login Attempts: The lots of users reported sessions expired or failed the first time, and they lost accounts even when they signed out.

Repeated Room Availability Checks Where there were errors in checking room availability or the system could not provide results, the operation was carried out several times.

Error Handling

Managing error is become important as it relates directly to the proper operational processes of room bookings. The system captures and manages various failure scenarios to prevent interruptions in user experience:

Session Timeout: If the guest is idle for sometime, then the system logs out the guest and asks for re-login.

Room Unavailability: In case a selected room is booked, the system informs the guest and provides other choices to the reservation.

Payment Gateway Issues: For the payment, the guest is prompted in case of a failure of the payment gateway, and the system then attempts to charge the guest again or cancel the booking.

The Bottleneck Analysis

The most significant bottlenecks identified in the room booking process include:

Room Availability Check: This task takes time especially during busy traffic periods when, for instance, external room databases may be unresponsive or sometimes, they may not give results at all.
Solution: Some general ideas like improving caching strategies for room availability data might help to mitigate these lags – at the very least, make them less critical.
Database Connection Issues: It common to experience delay especially at the room confirmation stage due to failures in accessing the payment database.
Solution: These delays might be eliminated and system reliability improved by stronger database system foundations for example through connection pooling.

Process Performance Predictive Analysis

Using predictive analytics, we can forecast potential outcomes based on past event logs:
Booking Time Predictions: The availability of system load data and the previous booking logs can be used to make an estimate of how long the current booking will take. For instance, during surge time, some searches like room search might take long, and the system may alert the users of expected delay.
Failure Likelihood: It means that identification of situations, when the probability of session timeouts or payment gateway problems is high will allow for the anticipation of the corresponding changes in the allocation of resources to minimize failures.

6.2.2 The analysis of the process of Ticket Booking Service

Case ID: Each instance of the ticket booking process is identified by a unique case ID, such as Booking 3, referring to a specific ticket reservation attempt.
Task IDs: These represent individual steps in the booking process, such as T2 - Select Flight and T6 - Confirm Payment.

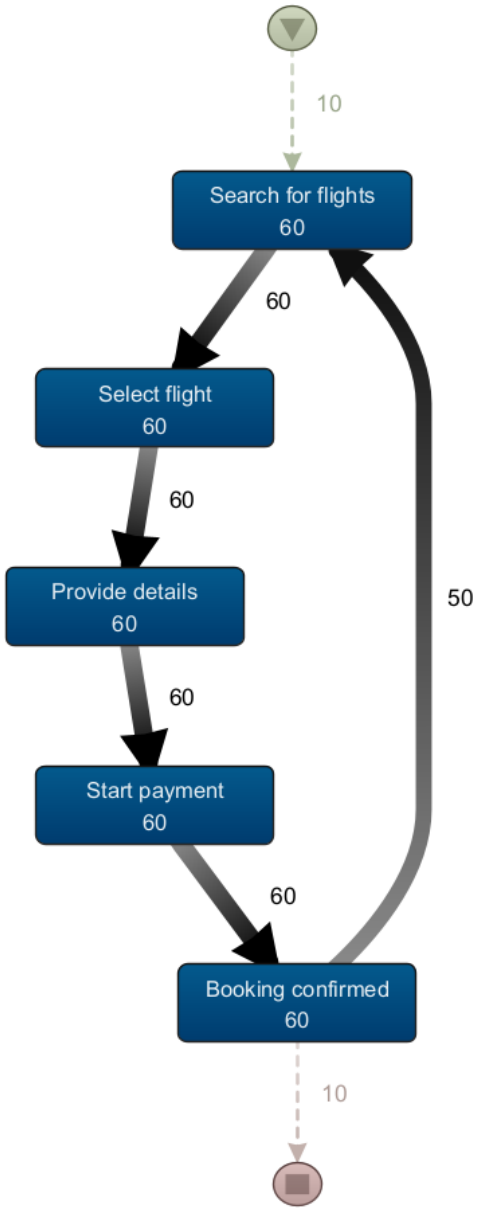


Figure 6.2: Ticket Booking Services

Timestamp and Resource Utilization

Timestamps: The logs include timestamps that allow us to measure the time taken for each task, crucial for identifying potential delays or bottlenecks in the process. For example, steps like "Search for Flights" and "Confirm Payment" might experience variation in duration depending on system load and resource availability.

Resource Utilization: This tracks which systems or personnel were responsible for each task. For instance, the "Search for Flights" task may query external flight databases, while "Confirm Payment" involves interaction with a payment gateway, which might sometimes delay due to network issues.

Visualizing Real Process Flow

The real-world execution of the ticket booking process revealed some deviations from the expected BPMN model:

Multiple Flight Search Attempts: In cases where the initial flight search failed, the task was repeated multiple times.

Repeated Payment Attempts: When the payment gateway encountered an issue, some users had to reattempt payment, causing delays in the overall process.

Error Handling

Error handling plays a critical role in ensuring the smooth operation of the ticket booking system, particularly during high-traffic periods. Common system errors include:

Invalid Login Credentials: If a user's login token is invalid or expired, the system prompts the user to re-login.

Flight Unavailability: When a selected flight is unavailable, the system informs the user and suggests alternative options or returns a failure notification.

Payment Gateway Failures: Payment processing failures lead to error messages and retries, causing users to go through the payment flow again.

Bottleneck Identification

Key bottlenecks in the ticket booking process include:

T8-Search for Flights: This task often experiences delays, particularly during peak travel periods when external flight databases may respond slowly or be unavailable.

Solution: Implementing efficient caching of flight data or improving API query efficiency could help reduce the occurrence of delays during flight search.

Payment Processing Delays: External payment gateways frequently cause delays, especially during high-demand periods, when the system is unable to handle requests efficiently.

Solution: Improving the reliability of the payment gateway connection or using multiple gateways to distribute the load can mitigate these bottlenecks.

Prescriptive Analytics and Process Improvement

Based on the insights gathered from predictive and diagnostic analytics, the following recommendations can improve the ticket booking process:

Optimize Login Systems: Implementing a more robust login system, such as extending token expiration times or allowing more flexible session handling, could reduce the number of failed login attempts.

Enhance Flight Search Efficiency: Optimizing flight availability checks, potentially by integrating better caching mechanisms and handling errors more gracefully, will minimize repeated checks and reduce system load.

Payment Gateway Optimization: Strengthening the infrastructure for payment processing, possibly by using multiple payment gateways and distributing load across them during high-demand periods, will reduce delays in payment confirmation.

6.2.3 Tour Booking Service Process Analytics

Case ID and Task Flow

Case ID: Each instance of the tour booking process is identified by a unique case ID, such as Booking 1, representing a specific tour reservation attempt.

Task IDs: These represent individual steps in the booking process, such as T2 - Select Tour and T6 - Confirm Payment.

Timestamp and Resource Utilization

Timestamps: The tour booking event logs contain timestamps that allow us to track the duration of

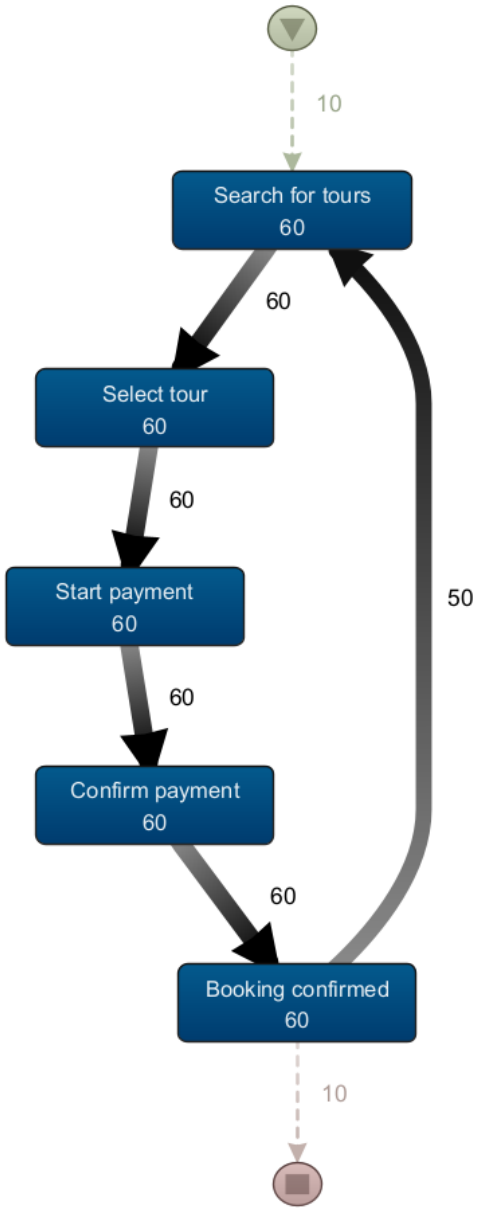


Figure 6.3: Tour Booking Services

each task, helping to identify potential delays or inefficiencies in the process. For example, tasks like "Search for Tours" and "Confirm Payment" might vary in time depending on system availability.

Resource Utilization: This measures the system or personnel responsible for completing each task. For instance, checking tour availability requires querying an external system, while confirming the booking involves interaction with the payment gateway, which may slow down during peak periods.

Visualizing Real Process Flow

A process map of the **tour booking service** revealed some deviations from the initial BPMN model:

Multiple Tour Search Attempts: Guests often had to search for available tours multiple times due to system delays or unavailability.

Repeated Payment Attempts: When the system encountered issues with the payment gateway, guests were required to restart the payment process.

Error Handling Effective error handling is crucial for a smooth tour booking experience. Common errors encountered include:

Invalid Login Tokens: When a guest's login credentials expire or fail, an error message is shown, and the guest is asked to re-login.

Tour Unavailability: If the selected tour is fully booked, the system informs the guest and provides alternatives or a failure message.

Bottleneck Identification

Key bottlenecks identified in the tour booking process include:

T8-Search for Tours: The task of searching for available tours often experiences delays, especially during peak tourist seasons or when external resources are slow to respond.

Solution: Implementing a more efficient querying mechanism or caching frequently searched tour availability could help reduce delays.

Payment Gateway Issues: Payment processing can often be delayed, particularly when external payment services experience high demand or network failures. **Solution:** Strengthening the payment infrastructure by utilizing multiple gateways and load-balancing during peak times can mitigate these bottlenecks.

Predictive Analytics for Process Performance

Using predictive analytics, the system can forecast potential outcomes based on historical event logs:

Booking Time Predictions: The system can estimate how long a booking will take based on the guest's actions and current system load. For instance, if the system is experiencing high traffic, the booking time might be longer.

Failure Likelihood: Predicting the probability of login errors or tour unavailability allows the system to preemptively adjust resources and avoid process failures.