PHENIKAA UNIVERSITY
PHENIKAA SCHOOL OF COMPUTING

SOFTWARE ARCHITECTURE
Lab 4: Microservices Decomposition & Communication

Instructor: Vũ Quang Dũng
Course: Software Architecture – Class N02
Team Members: Group 11
Nguyễn Trọng Thái – 23010457
Nguyễn Minh Quang – 23010419

Tháng 12 năm 2025

Activity Practice 1: Decomposition by Business Capability

Goal: Identify independent microservices based on the logistics and shipping functions of QuickShip.

1. Identify Core Business Capabilities

We group the functional requirements into distinct domains: User Management, Logistics/Shipping, Real-time Tracking, Fleet Management, and Finance.

2. Define Microservices

Each capability is mapped to a dedicated service with its own database.

Business Capability	**Proposed Microservice**	**Data Owned (Entities)**
User Management	**User Service**	Customer Profiles, Courier Profiles, Auth Credentials.
Logistics Management	**Shipment Service**	Package Details, Weight, Sender/Receiver Addresses.
Real-time Tracking	**Tracking Service**	GPS Coordinates, Status History (Pending, In Transit, Delivered).
Fleet & Driver Mgmt	**Fleet Service**	Vehicle Details, Driver Availability, Route Assignments.
Billing & Pricing	**Billing Service**	Invoices, Shipping Rates, Payment Status.

3. Define External Dependencies

- **Payment Gateway API** (e.g., Stripe, VNPay): To process actual payments.
- **Map API** (e.g., Google Maps, MapBox): For distance calculation and route optimization.
- **SMS/Email Provider**: To send OTPs and shipment arrival notifications.

Activity Practice 2: Defining Service Contracts

Goal: Establish the public RESTful interfaces for interaction between services.
We focus on the interaction between the **Shipment Service** (Producer) and the **Tracking Service** (Consumer).

1. Shipment Service API Contract

Endpoint	**Method**	**Description**	**Data Returned**
/api/shipments/{id}	**GET**	Retrieve full details of a specific shipment.	Shipment object (Address, weight, type)
/api/shipments	**POST**	Create a new shipping request.	New Shipment object + Tracking ID
/api/shipments/{id}/status	**PATCH**	Update the shipment status (e.g., picked up).	Updated Status object

2. Service Interaction Requirement

When a user wants to see their package location, the **Tracking Service** needs basic info from the **Shipment Service**.

- **Constraint:** The Tracking Service **must not** access the Shipment Service's database directly (violates the database-per-service pattern). It must use the GET /api/shipments/{id} endpoint.

Activity Practice 3: C4 Model (Level 1: System Context)

Goal: Create a high-level map of the QuickShip ecosystem and its external interactions.

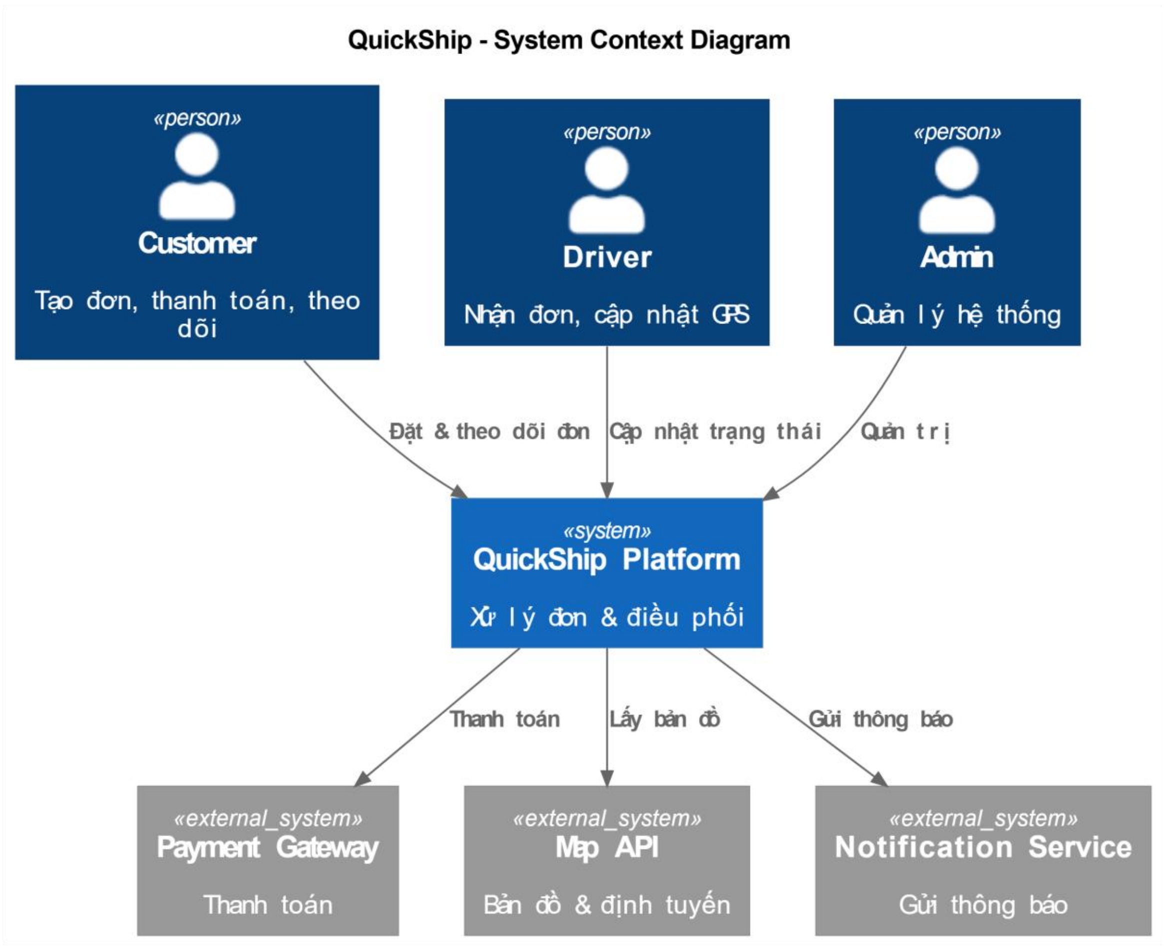
1. System Context Diagram Components

- **Actors:** Customer (Shipper), Courier (Driver), and Administrator.
- **System:** QuickShip Platform (The boundary containing all microservices).
- **External Systems:** Payment Gateway, Map API, SMS Gateway.

2. Communication Strategy Analysis

Interaction	**Service/Component**	**Communication Type**	**Rationale**
Tracking Lookup	App $\rightarrow$ Tracking Service	**Synchronous** (HTTP)	The user expects an immediate response to see the map location.
Calculate Fee	Shipment $\rightarrow$ Billing Service	**Synchronous** (HTTP)	The shipping cost must be shown before the user confirms the order.
Shipment Alert	Shipment $\rightarrow$ SMS Gateway	**Asynchronous** (Message Queue)	Sending a text message shouldn't block the order creation process.

Interaction	**Service/Component**	**Communication Type**	**Rationale**
Analytics Update	Billing $\rightarrow$ Admin Dashboard	**Asynchronous** (Event-driven)	Data for reports can be processed in the background without affecting performance.


QuickShip - System Context Diagram
«person»
Customer
Tạo đơn, thanh toán, theo dõi
«person»
Driver
Nhận đơn, cập nhật GPS
«person»
Admin
Quản lý hệ thống
Đặt & theo dõi đơn
Cập nhật trạng thái
Quản trị
«system»
QuickShip Platform
Xử lý đơn & điều phối
Thanh toán
Lấy bản đồ
Gửi thông báo
«external_system»
Payment Gateway
Thanh toán
«external_system»
Map API
Bản đồ & định tuyến
«external_system»
Notification Service
Gửi thông báo