Lab 4: Microservices Decomposition & Communication

This lab marks the transition from the monolithic **Layered Architecture** (Lab 3) to the **Microservices Architecture**. The goal is to logically break down the ShopSphere application, define the service boundaries, and establish how these new independent services will communicate.

Objectives

1. Understand the principles of **Microservice Decomposition** (Decomposition by Business Capability).
2. Define the clear **Service Contracts** (API Endpoints) for key services.
3. Design the system's **High-Level Communication Strategy** (Synchronous vs. Asynchronous).
4. Model the system using the **C4 Model (Level 1: System Context)**.

Technology & Tool Installation

Tool	Purpose	Installation/Setup Guide
draw.io (Diagrams.net)	Creating C4 Model and high-level communication diagrams.	Access online via a web browser (draw.io has a built-in C4 extension).
Documentation Tool (e.g., Markdown/Word)	Documenting the Service Contracts (API specifications).	Standard text editor or word processor.
No Coding Required	This lab is purely design and documentation.	N/A

Activity Practice 1: Decomposition by Business Capability

Goal: Identify clear, independent microservices based on the business functions of ShopSphere.

Step-by-Step Instructions

1. **Identify Core Business Capabilities:** Review the functional requirements from Lab 1 and group them into independent business domains.
   - **Action:** List the primary domains (e.g., "User Management," "Catalog Management," "Fulfillment").
2. **Define Microservices:** Map each capability to a dedicated microservice. Ensure services are **loosely coupled** (meaning they can operate and fail independently).

Business Capability	**Proposed Microservice**	**Data Owned (Entities)**
User Management	**User Service**	User Profile, Authentication Credentials.
Catalog Management	**Product Service**	Product Details, Price, Image URLs.
Fulfillment & Tracking	**Order Service**	Order Header, Line Items, Shipping Address.
Inventory	**Inventory Service**	Stock Levels, Warehouse Location.
Pre-Purchase	**Cart Service**	Cart Contents, Session State.

3. **Define External Dependencies:** Identify which external systems ShopSphere must interact with.
   - **Action:** List at least 3 external systems (e.g., Payment Processor, Email/SMS Provider, Shipping Carrier API).

Activity Practice 2: Defining Service Contracts

Goal: Establish the public interfaces (API Endpoints) that services will expose to the outside world or to other services.

Step-by-Step Instructions & Documentation Guide

Focus on the interaction between the **Product Service** (Producer) and the **Cart Service** (Consumer).

1. **Define the Product Service's API:** Document the RESTful contract for the Product Service.

Endpoint	**HTTP Method**	**Description**	**Data Returned**
/api/products/{id}	**GET**	Retrieve full details of a single product.	Product object (name, price, description)
/api/products	**GET**	Search/List products based on criteria.	List of Product objects (summary view)
/api/products	**POST**	Add a new product (Admin function).	Product object

2. **Define the Cart Service's Interaction:** Define what information the **Cart Service** needs from the **Product Service** when a user adds an item to their cart.
   - **Action:** The Cart Service needs the **Price** and **Name** of the product ID requested.
   - **Requirement:** The Cart Service *must not* access the Product Service's database directly (violates microservice boundary). It must use the defined API endpoint (GET /api/products/{id}).

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

Goal: Create a high-level visual map showing the boundaries of the new system and its external interactions.

Step-by-Step Instructions (Using draw.io with C4 Shapes)

1. **Draw the System:** Create a large component labeled **"[System] ShopSphere E-commerce Platform"**.
2. **Place Users and External Systems:** Draw the **"Web Customer"** and **"Administrator"** (actors) and the **"Payment Gateway"** (external system) outside the main system boundary.
3. **Define Communication:** Draw directional arrows to show interaction:
   - **Web Customer** $\rightarrow$ **ShopSphere:** "Browses and Places Orders."
   - **Administrator** $\rightarrow$ **ShopSphere:** "Manages Catalog and Fulfillment."
   - **ShopSphere** $\rightarrow$ **Payment Gateway:** "Submits Payment for Processing."

4. **Analyze Communication Strategy:** At this level, distinguish between synchronous and asynchronous communication needs.

Interaction	**Service/Component**	**Communication Type**	**Rationale**
Customer Lookup	API Gateway to User Service	**Synchronous (HTTP)**	Immediate response required for login/profile display.
Order Confirmation Email	Order Service to Email Provider	**Asynchronous (Message Queue)**	Email does not need to block the order completion process.
Inventory Check	Cart Service to Inventory Service	**Synchronous (HTTP)**	Need immediate stock verification before checkout.

C4 Model Sketch Guide (Level 1)

The completion of this lab provides the blueprint for building the microservices independently.