

# YouShop Microservices — Full Understanding Guide

Educational Version

December 22, 2025

## Contents

1	1. What Is This Application?	2
2	2. Global Microservices Architecture	2
3	3. Why This Diagram Matters	2
4	4. API Gateway — Single Entry Point	3
5	5. Auth Service — Identity	3
6	6. Catalog Service — Products	3
7	7. Inventory Service — Stock	3
8	8. Orders Service — Business Logic	4
9	9. Event-Driven Communication	4
10	10. Full Order Flow	4
11	11. Why This Architecture Works	4

# 1. What Is This Application?

YouShop is the **backend** of an e-commerce platform.

- No UI / No frontend
- Only logic, data, security, and workflows

## Key Idea

The goal is to guarantee:

- Secure users
- Correct products
- Accurate stock
- Reliable orders

## 2. Global Microservices Architecture

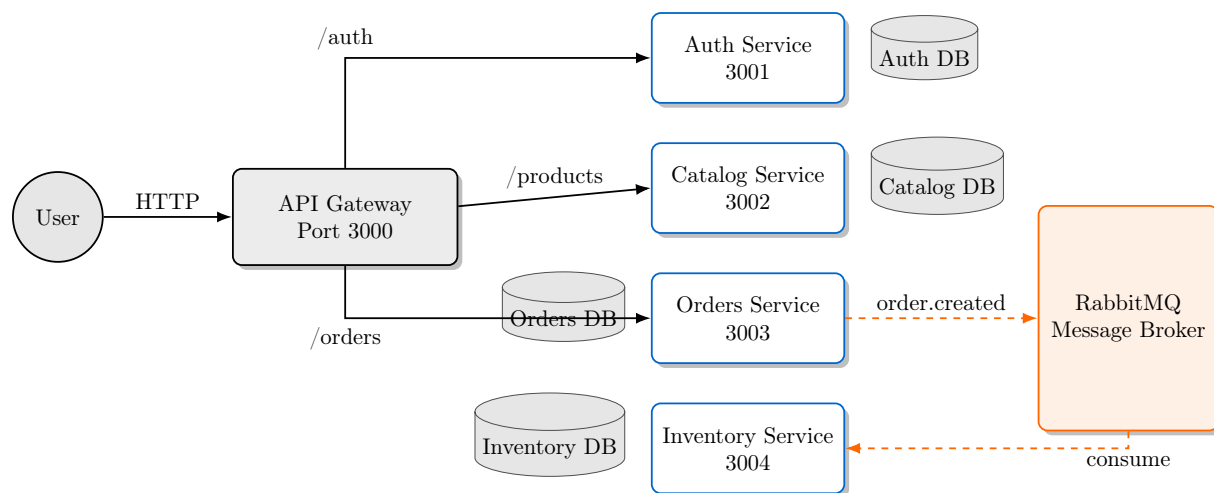


Figure 1: YouShop Microservices Architecture (API Gateway + Event-Driven Communication)

## 3. Why This Diagram Matters

### Key Idea

This diagram shows three important rules:

- Users talk **ONLY** to the Gateway
- Services never share databases
- Services communicate via events

### Important Rule

Orders Service never updates Inventory DB directly.

## 4 4. API Gateway — Single Entry Point

The Gateway:

- Receives all HTTP requests
- Validates JWT tokens
- Routes requests to services

### Real-Life Story

Like airport security: One entrance, many destinations.

## 5 5. Auth Service — Identity

Handles:

- Registration
- Login
- Password hashing
- JWT creation

Roles:

- Client
- Admin

## 6 6. Catalog Service — Products

Manages:

- Product information
- Categories
- Prices

### Important Rule

Catalog does NOT know stock quantities.

## 7 7. Inventory Service — Stock

Inventory tracks:

- Total stock
- Reserved stock

### Real-Life Story

Reserved stock = people paying.

## 8 8. Orders Service — Business Logic

Orders:

- Creates orders
- Tracks status
- Emits events

States:

- PENDING
- CONFIRMED
- CANCELLED

## 9 9. Event-Driven Communication

### Key Idea

Services never wait for each other.

Orders emits events, Inventory reacts.  
This ensures:

- Speed
- Loose coupling
- Failure isolation

## 10 10. Full Order Flow

1. User sends order request
2. Gateway verifies token
3. Order created (PENDING)
4. Event sent to RabbitMQ
5. Inventory reserves stock
6. Order confirmed or cancelled

## 11 11. Why This Architecture Works

- Independent services
- Safe data ownership
- Real-world scalability
- Clear responsibilities

**This is how YouShop works using Microservices.**