Theory Assignment Report

Only for course Teacher					
	Needs Improvement	**Developing**	**Sufficient**	**Above Average**	**Total Mark**
Allocate mark & Percentage	25%	50%	75%	100%	5
Clarity	1				
Content Quality	2				
Spelling & Grammar	1				
Organization and Formatting	1				
				Total obtained mark	
Comments					

Semester: Summer 2025

Student Name: Afzal Hossain Raju

Student ID: 0242310005341031

Batch: 40 **Section:** D

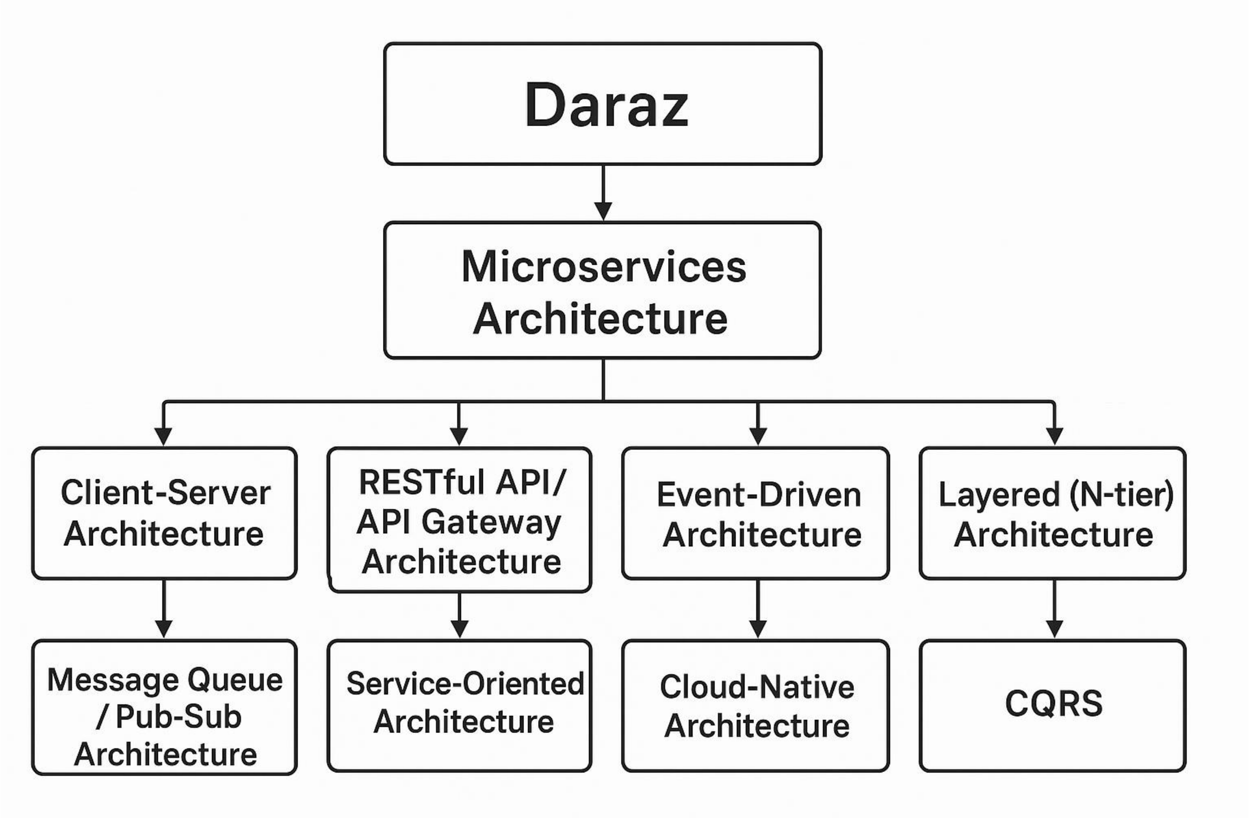
Course Code: SE323 **Course Name:** Software Architecture Design, and Analysis

Course Teacher Name: Fazla Rabby Raihan

Designation: Lecturer

Submission Date: 12/8/2025

Uses of Microservice Architecture in Daraz App


Daraz
Microservices Architecture
Client-Server Architecture
RESTful API/ API Gateway Architecture
Event-Driven Architecture
Layered (N-tier) Architecture
Message Queue / Pub-Sub Architecture
Service-Oriented Architecture
Cloud-Native Architecture
CQRS

About Daraz App Software Architecture Model: The Daraz app is built on a microservices architecture, which means the application is divided into many small, independent services. Each service handles a specific function—such as product catalog, payments, orders, or user accounts. This makes the app scalable, easier to update, and more reliable.

Alongside microservices, Daraz also uses other supporting architectures:

Client-Server architecture: where the mobile app communicates with backend servers.

API Gateway architecture: to connect clients with different microservices through a single entry point.

Event-driven architecture: to handle real-time features like notifications and order updates.

Layered architecture: for separating presentation, business logic, and data.

Message queues: for managing high traffic smoothly.

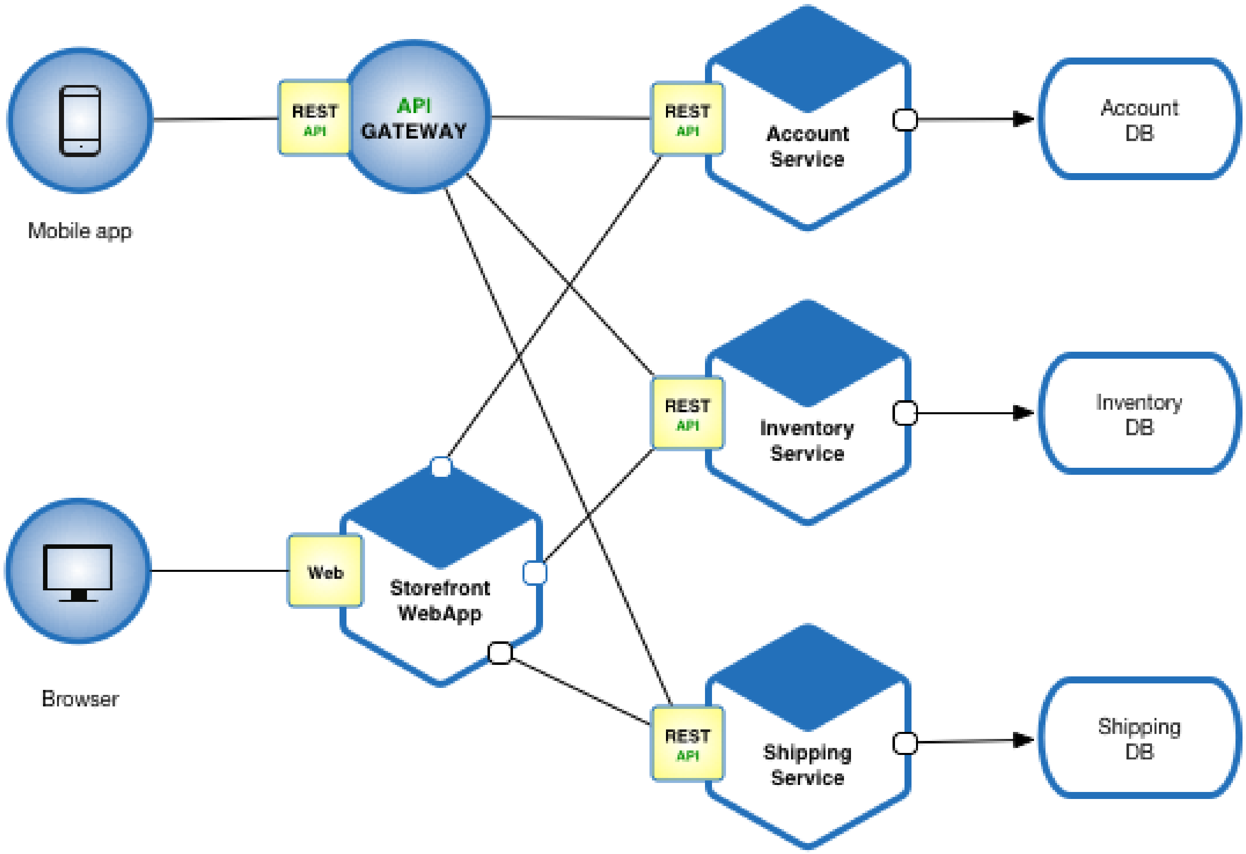
All of these work together to make Daraz fast, scalable, and capable of handling millions of users.

About Microservice Architecture:

Microservices Architecture is a way of designing software where an application is divided into small, independent services.

Advantages of Microservice Architecture:

- Each service handles one function (e.g., payments, orders, notifications).
- Services run independently but communicate through APIs.
- Easy to scale, update, and maintain.
- If one service fails, the rest of the system still works.

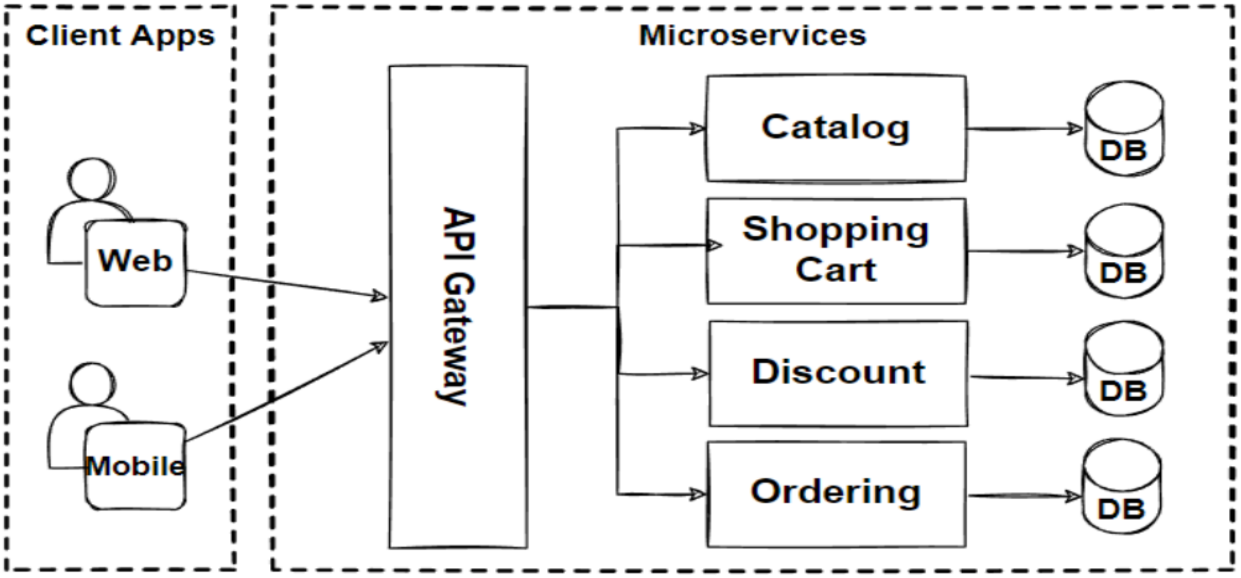


(Diagram Of Microservice Architecture)

Why Microservice Architecture in Daraz App:

They are used in the Daraz App to overcome the challenges of running a large-scale e-commerce platform, such as:

- **Fast feature development:** Daraz frequently adds new shopping features, payment methods, discounts, and campaigns. With microservices, different teams can build and release updates independently without waiting for the whole system.
- **Better scalability**: During big campaigns like 11.11 sales, traffic spikes massively. Microservices allow Daraz to scale specific services (e.g., checkout, payment, search) instead of scaling the entire app.
- **Reduced system failure risks:** If the payment service has a bug, it won't crash product browsing or order tracking. This isolation ensures the app stays available even if one part faces issues.
- **Continuous deployment & flexibility:** Microservices make it easier for Daraz to roll out updates, fix bugs, and deploy new features without redeploying the whole system.
- **Handling complex operations:** Daraz manages millions of users, sellers, products, and logistics. Microservices break this complexity into smaller, manageable services (e.g., product service, order service, delivery service, notification service).



(Use case of Microservice Architecture in Daraz App)

Alternative Architecture that could have been used:

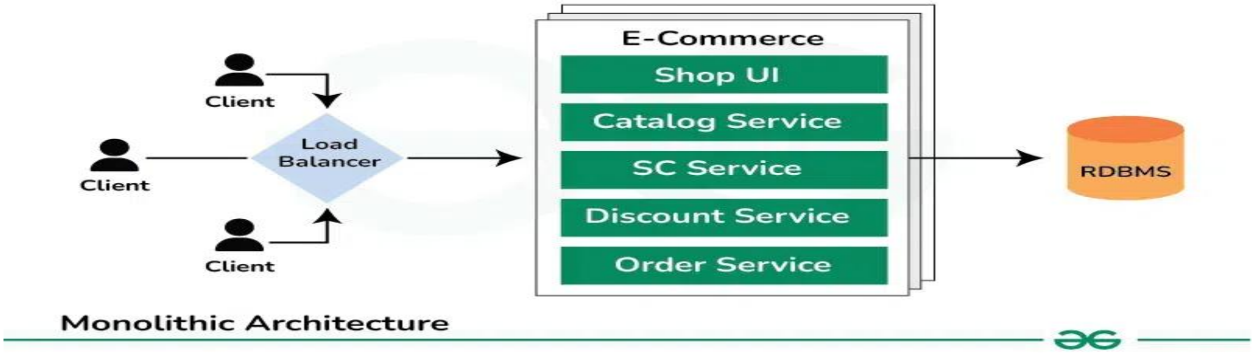
If Daraz hadn’t used Microservices, there are a few alternative architectures they could have considered, but each comes with trade-offs. Here are the main ones:

1. **Monolithic Architecture**

   Description: The entire app (products, orders, payments, logistics) is built as a single unit.

   Why it could be used: Easier to develop initially, lower cost, simple to deploy.

   Limitations for Daraz: Difficult to scale during campaigns (11.11, Black Friday), as a single bug could crash the entire system, and feature updates are very slow.



2. **Service-Oriented Architecture (SOA)**

   Description: Similar to microservices, but services are larger and often depend on a centralized Enterprise Service Bus (ESB) for communication.

   Why it could be used: Better modularity than a monolith, easier integration with external systems.

   Limitation for Daraz: The ESB can become a bottleneck, making it less scalable than microservices.

01
PEOPLE
Empower decision Makers
02
PROCESS
Align IT with business operations
SOA
Service Oriented Architecture
03
PLATFORM
Increase Operational efficiency
04
PRACTICE
Employ best practice methology