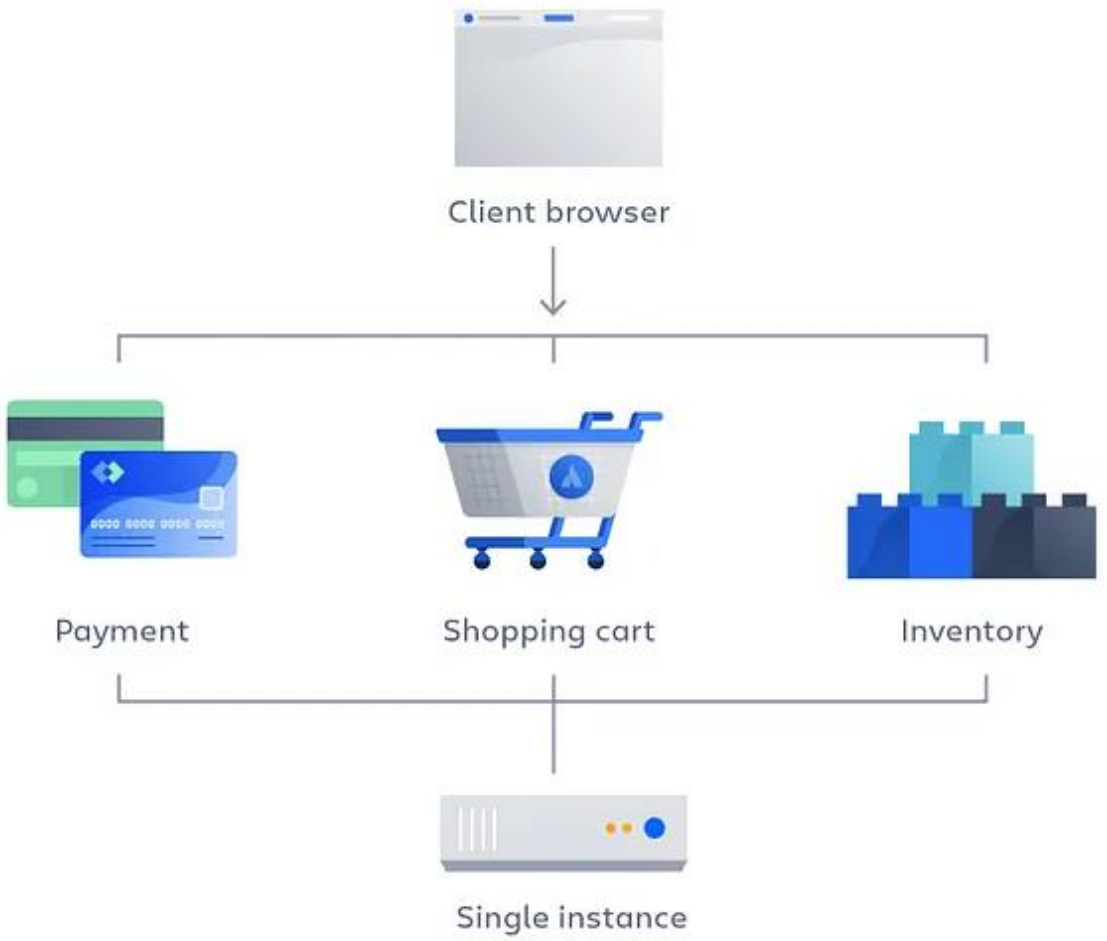
Software Architecture
Course's Code: CSE 483
Monolith and Microservice Architectures
(Chapter 9)

Chapter 9. Monolith and Microservice Architectures

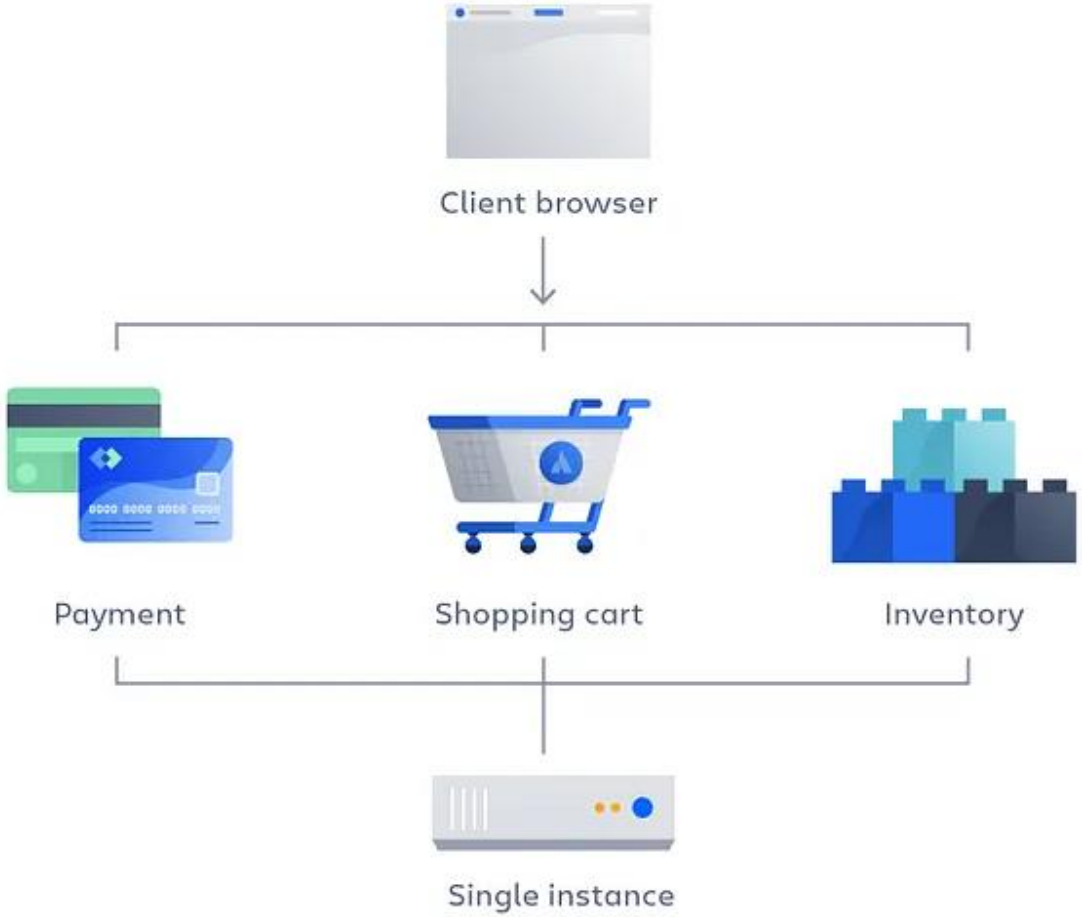
What is a Monolith?

Monolithic architecture is a software design approach where an entire application is built as a single, unified codebase. All the components, modules, and functionalities of the application are tightly integrated into a single executable package.



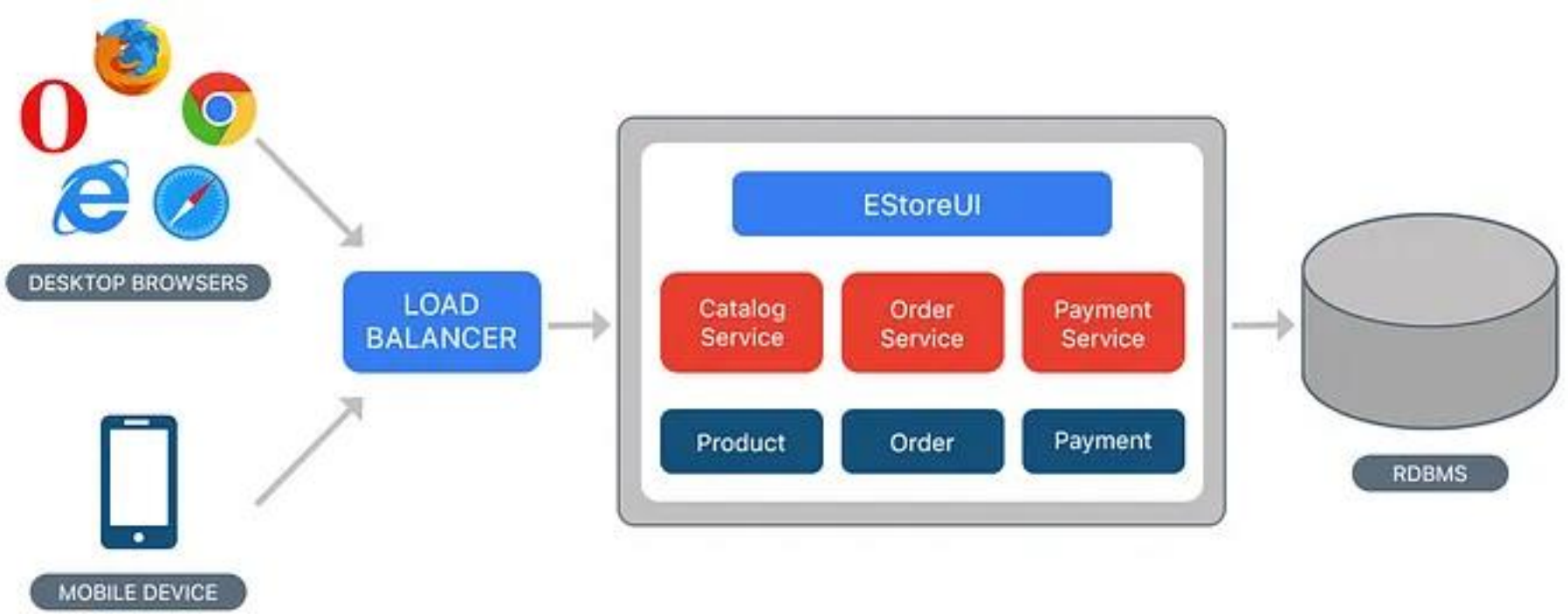
What is a Monolith?

- Self-contained and packaged together
- Package as WAR/EAR
- Typical application



What is a Monolith?

- Deployed to webserver
- Run behind load-balancer
- Can be any structure



What is a Monolith?

Advantages

- **Easy debugging** : With all code located in one place, it's easier to follow a request and find an issue.
- **Simplified testing**: Since a monolithic application is a single, centralized unit, end-to-end testing can be performed faster than with a distributed application.
- **Simplicity**: Monoliths are relatively straightforward to develop and deploy since all the code resides in one place. This simplicity can be advantageous for small to medium-sized projects with limited complexity.

What is a Monolith?

Disadvantages

- **Slower development speed**: A large, monolithic application makes development more complex and slower.
- **Scalability**: You can't scale individual components.
- **Reliability**: If there's an error in any module, it could affect the entire application's availability.

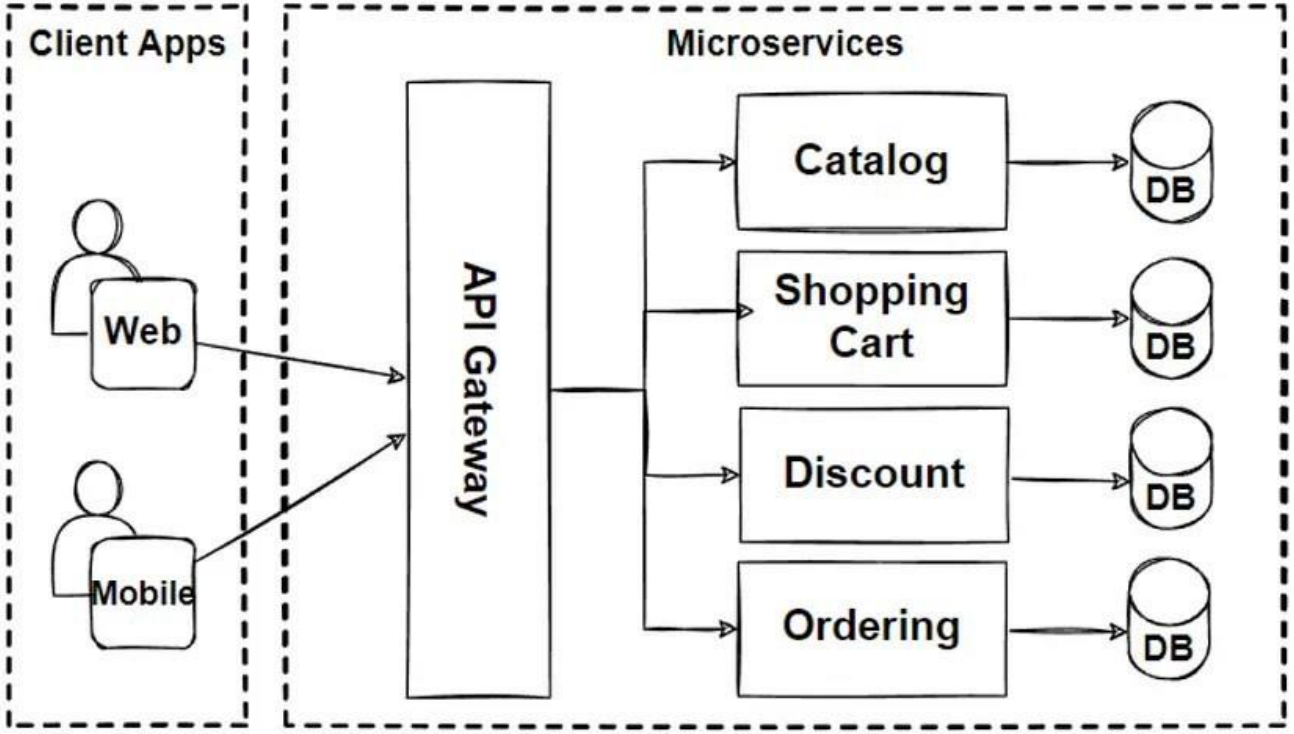
What is a Monolith?

Disadvantages

- **Barrier to technology adoption:** Any changes in the framework or language affects the entire application, making changes often expensive and time-consuming.
- **Lack of flexibility:** A monolith is constrained by the technologies already used in the monolith.
- **Deployment:** A small change to a monolithic application requires the redeployment of the entire monolith.

What is Microservice?

- Hot topic
- Distributed and Scalable
- Small and handle one concern
- Decompose into small services



Microservices - Design characteristics

- **Autonomy**: The independence and decoupling of microservices allow teams to organize themselves independently and focus on a smaller domain with a smaller change surface. This allows the company to quickly change and evolve services according to business needs without having to rewrite the entire application.

- **Unique components:** Services are designed and deployed as individual components working together to accomplish a specific function or address a specific requirement.

- **Decentralized:** Unique microservices components have few if any dependencies, although loose coupling requires frequent and extensive communication between components.

Microservices - Design characteristics

- **Resilient:** Services are designed for maximum fault tolerance. A single service failure shouldn't disable an entire application.
- **API-based:** A microservices architecture relies on APIs (**application programming interfaces**) and API gateways to facilitate communication between components and other applications.
- **Data separation:** Each service accesses its own database or storage volume.

What is a Microservice?

Advantages

- Easy to understand and manage
- Quicker deployment
- One microservice per team
- UI divorced from back-end development

What is a Microservice?

Advantages

- Increased fault tolerance
- Scaling does not break service
- Change technology stack easily

What is a Microservice?

Disadvantages

- Microservice spawn
- Maintenance becomes harder
- Requires investment in deployment and maintenance
- Cross-cutting changes costly to implement

What is a Microservice?

Disadvantages

- Duplicated code
- Increased unreliability and complexity of communication

Monolith vs. Microservice?

- Monoliths work well for small applications
- Monolith easy to develop and test
- Microservices easier to understand
- Continuous deployment and scalability