



AlgoTutor

MASTER

SYSTEM DESIGN

IN JUST 21 DAYS



Day 1

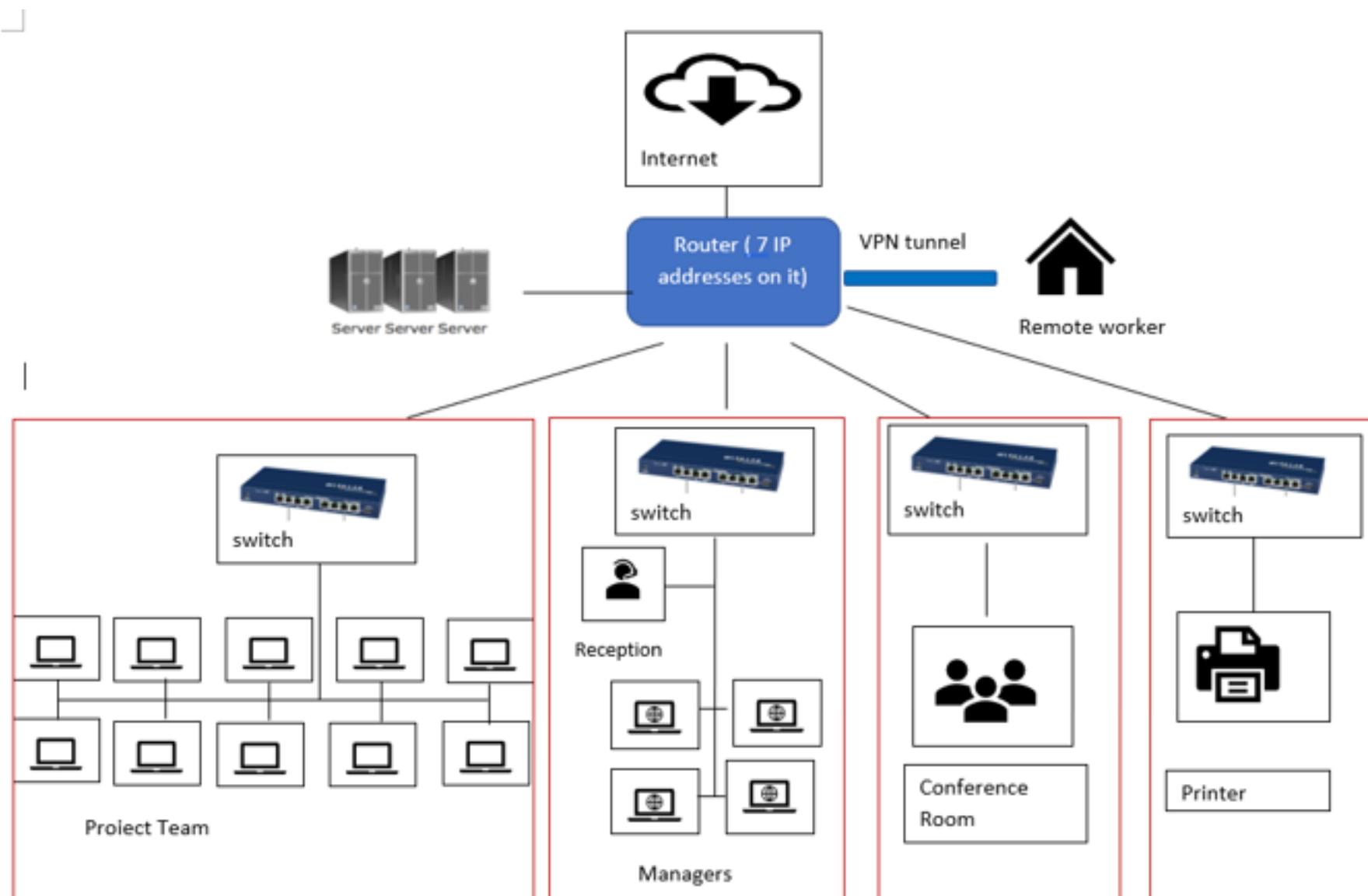
Introduction to System Design

- ◆ Understand the importance of system design in building scalable, reliable systems.
- ◆ Explore the key components of system design, such as load balancing, caching, and databases.

Day 2

Networking Basics

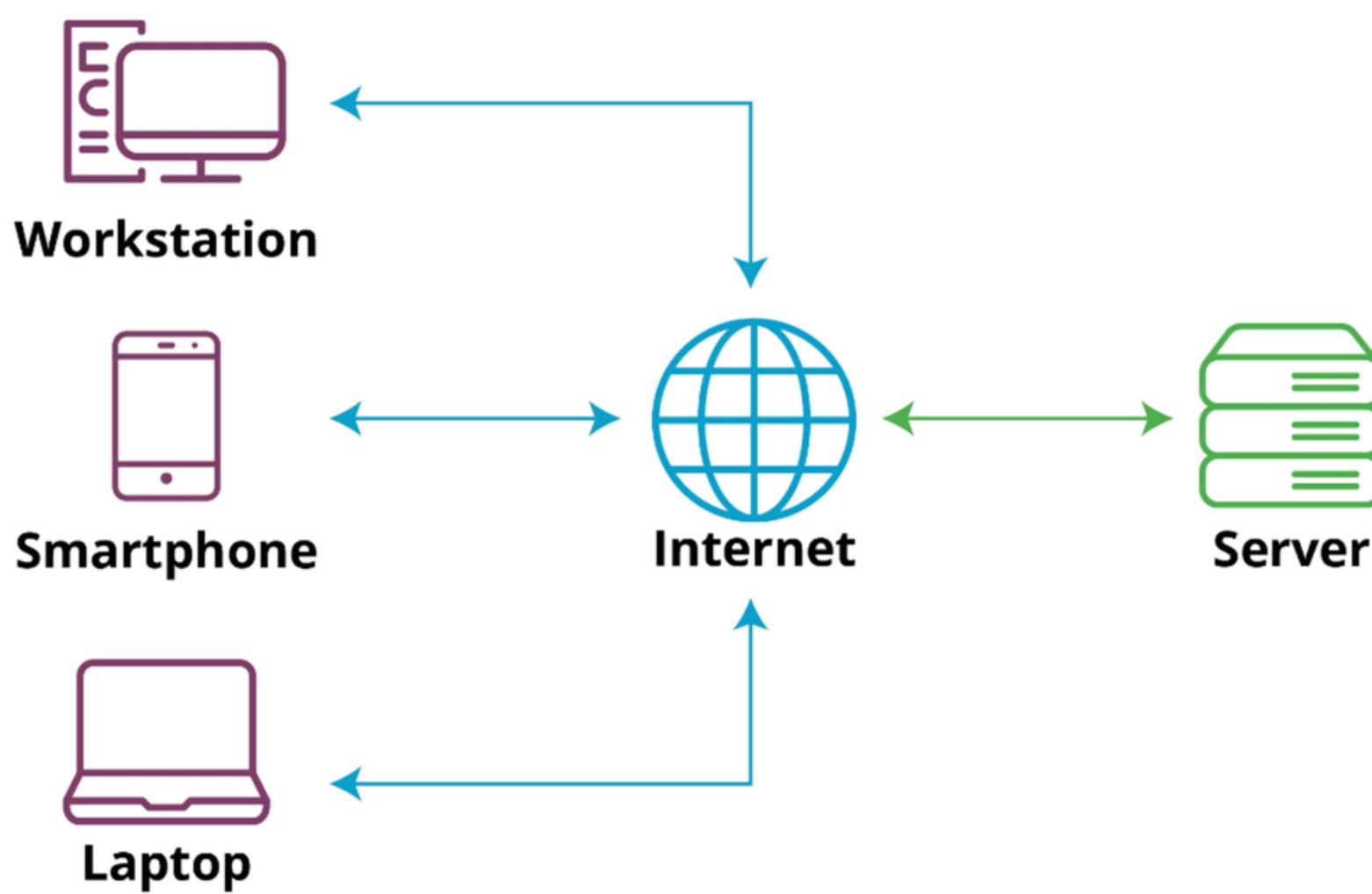
- ◆ Dive into networking concepts, including protocols, IP addressing, and routing.
- ◆ Learn about the OSI model and how it relates to system design.



Day 3

Understanding Client-Server Architecture

- ◆ Explore the client-server model and its relevance in system design.
- ◆ Study the roles and responsibilities of clients and servers in a distributed system.



Day 4

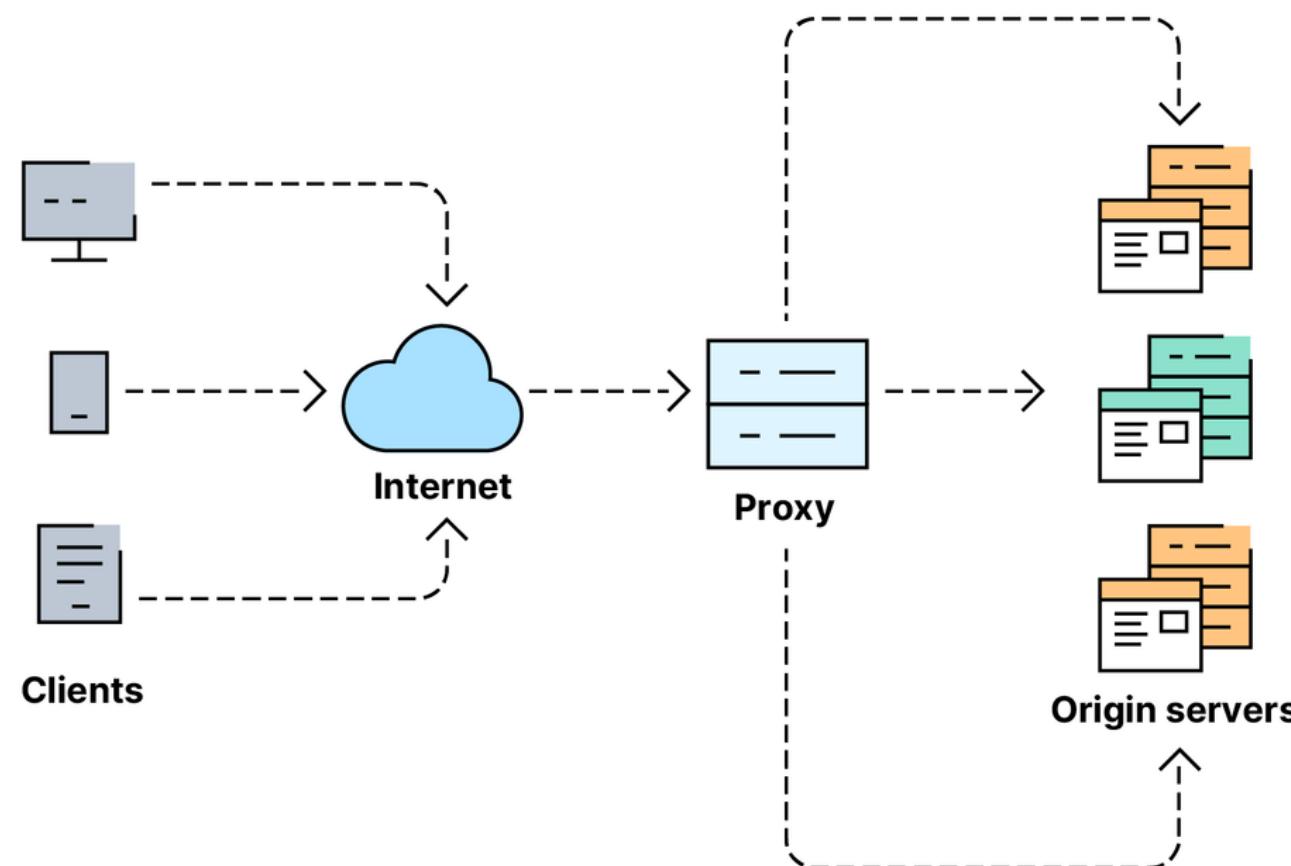
Scalability

- ◆ Delve into the principles of scalability.
- ◆ Understand horizontal and vertical scaling, and their use cases.

Day 5

Load Balancing

- ◆ Learn about load balancers and their role in distributing traffic.
- ◆ Study load balancing algorithms and strategies.



Day 6

Caching

- ◆ Explore the importance of caching in improving system performance.
- ◆ Study caching strategies and when to use them.

Day 7

Databases and Data Stores

- ◆ Understand different types of databases (SQL, NoSQL) and data stores.
- ◆ Study data modeling and database design.



Day 8

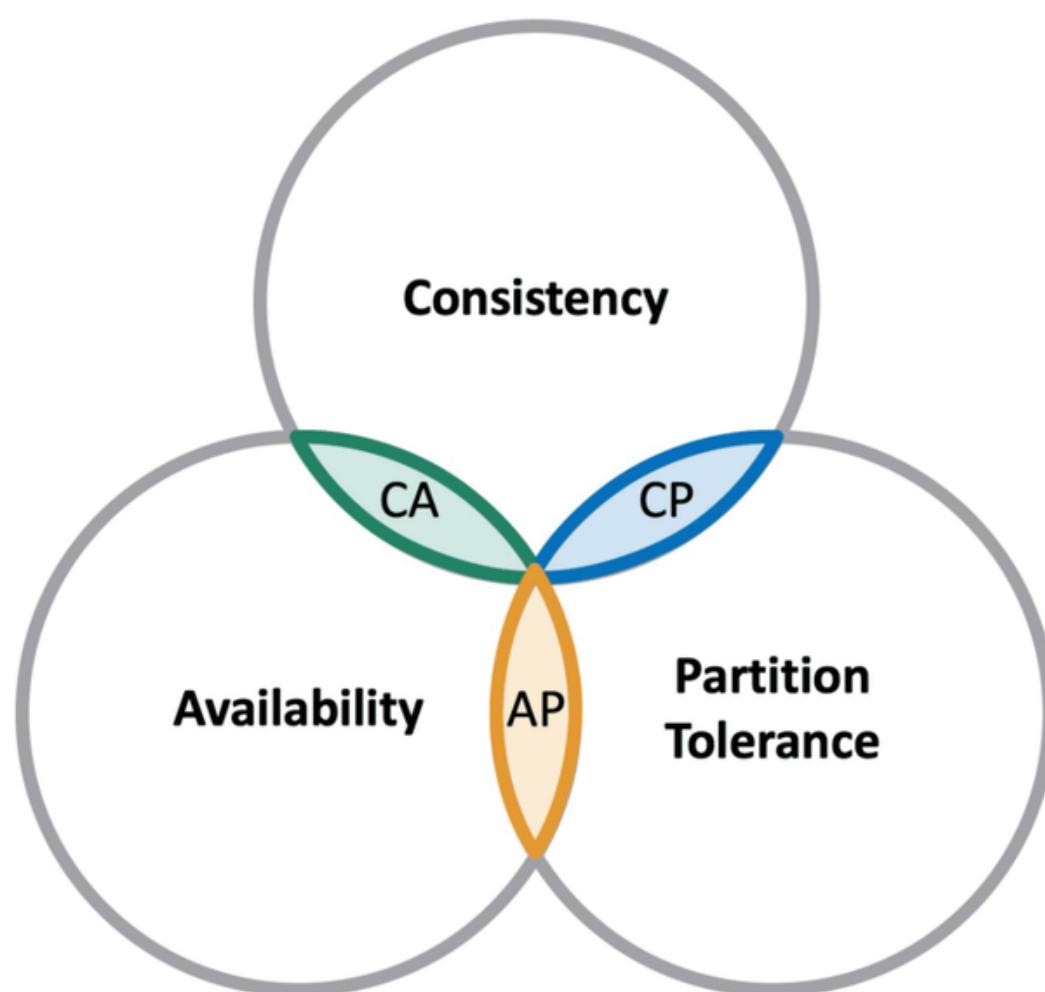
Distributed Systems Basics

- ◆ Dive into the fundamentals of distributed systems.
- ◆ Learn about distributed computing models and their challenges.

Day 9

CAP Theorem

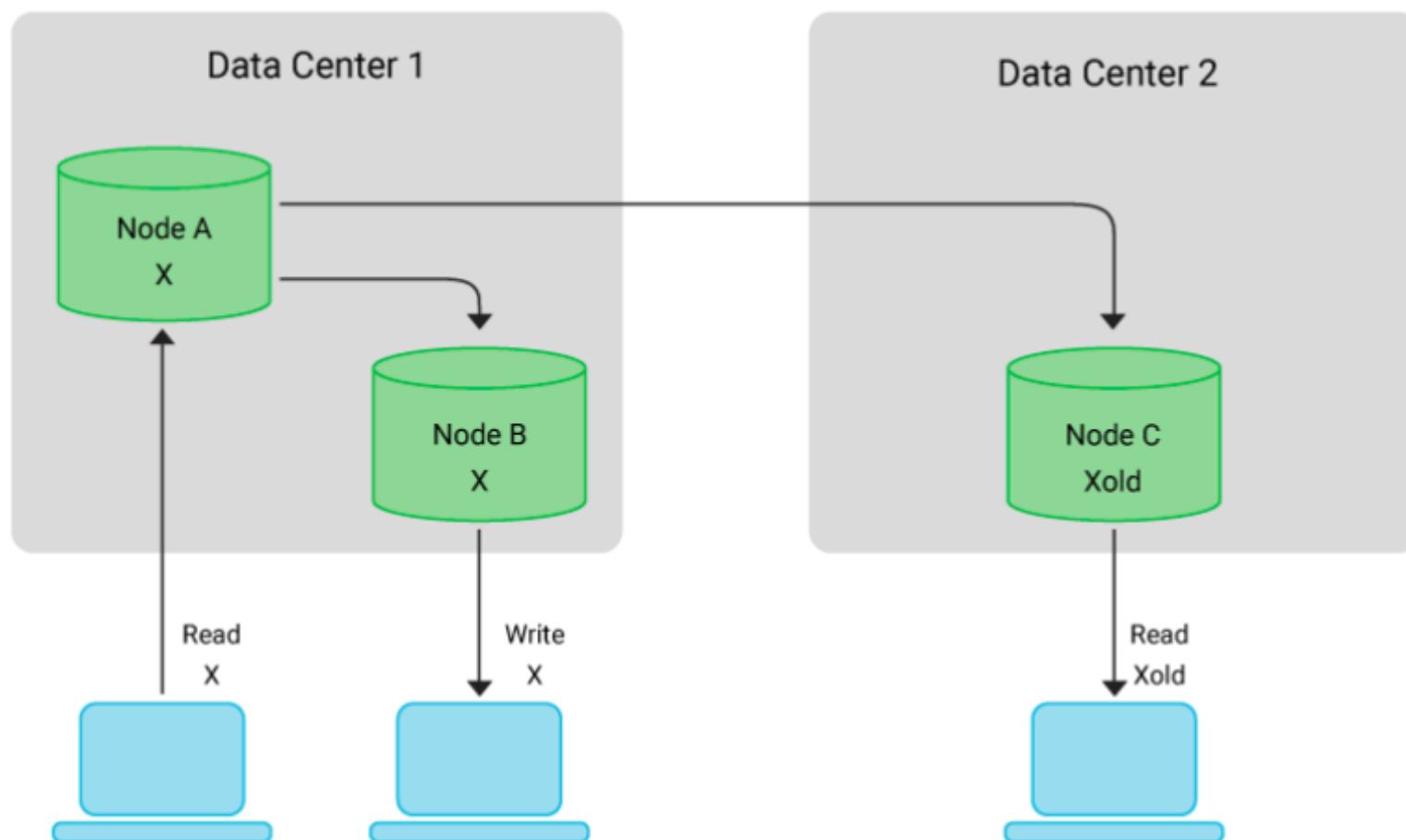
- ◆ Study the CAP theorem and its implications on distributed systems.
- ◆ Understand the trade-offs between Consistency, Availability, and Partition Tolerance.



Day 10

Eventual Consistency

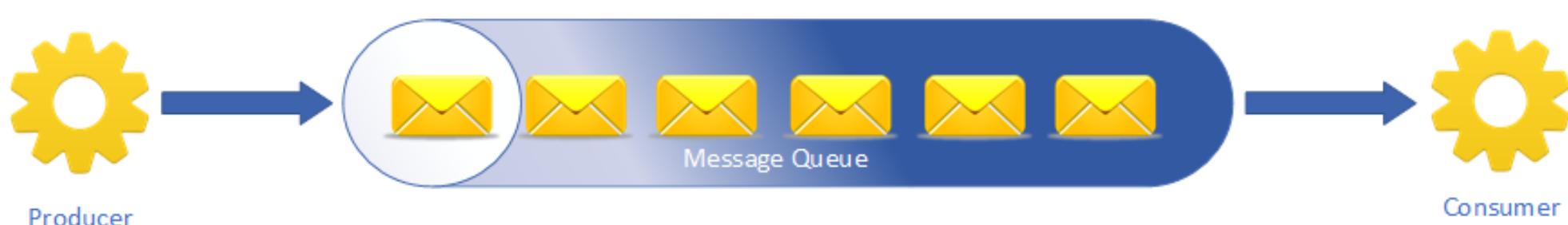
- ◆ Explore the concept of eventual consistency in distributed databases.
- ◆ Study how systems achieve consistency over time.



Day 11

Message Queues

- ◆ Learn about message queues and their use in building scalable systems.
- ◆ Study popular message queuing systems like Kafka and RabbitMQ.

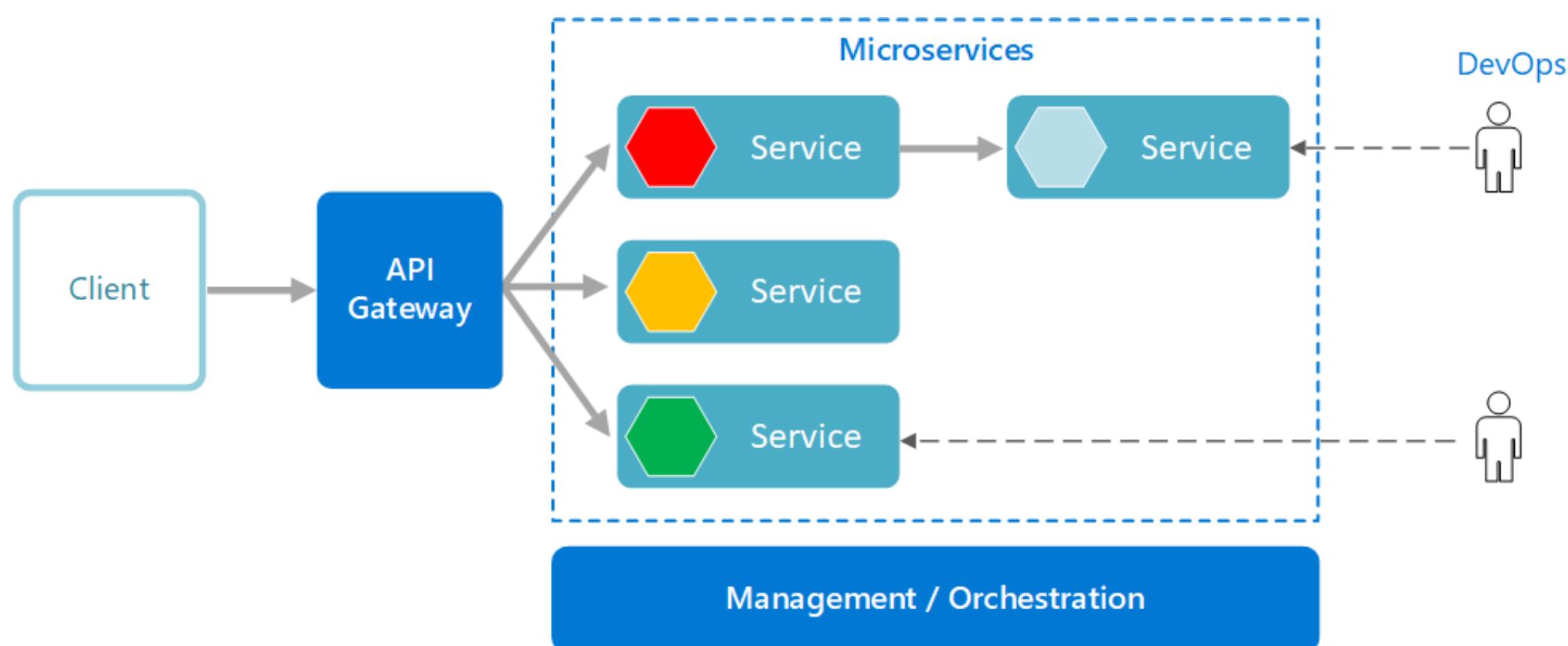




Day 12

Microservices Architecture

- ◆ Explore microservices architecture and its benefits.
- ◆ Learn about service discovery, communication, and orchestration.



Day 13

Security in System Design

- ◆ Understand security best practices in system design.
- ◆ Study authentication, authorization, and encryption.

Day 14

Implementing Infrastructure as Code (IaC)

- ◆ Learn about IaC tools like Terraform and Ansible.
- ◆ Study how to automate infrastructure provisioning.

Day 15

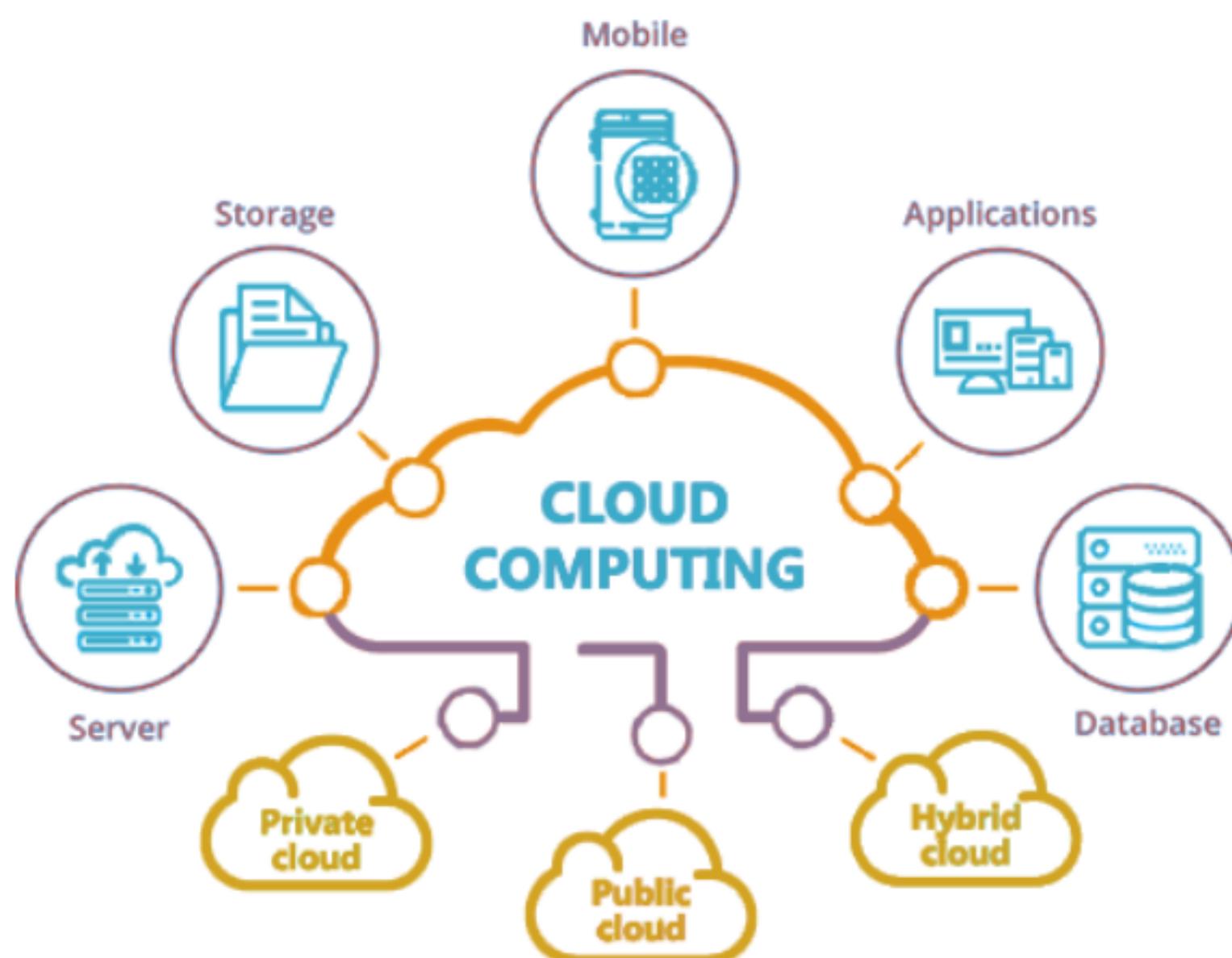
Case Studies

- ◆ Analyze real-world case studies of system design.
- ◆ Learn from successful system design implementations.

Day 16

Cloud Computing

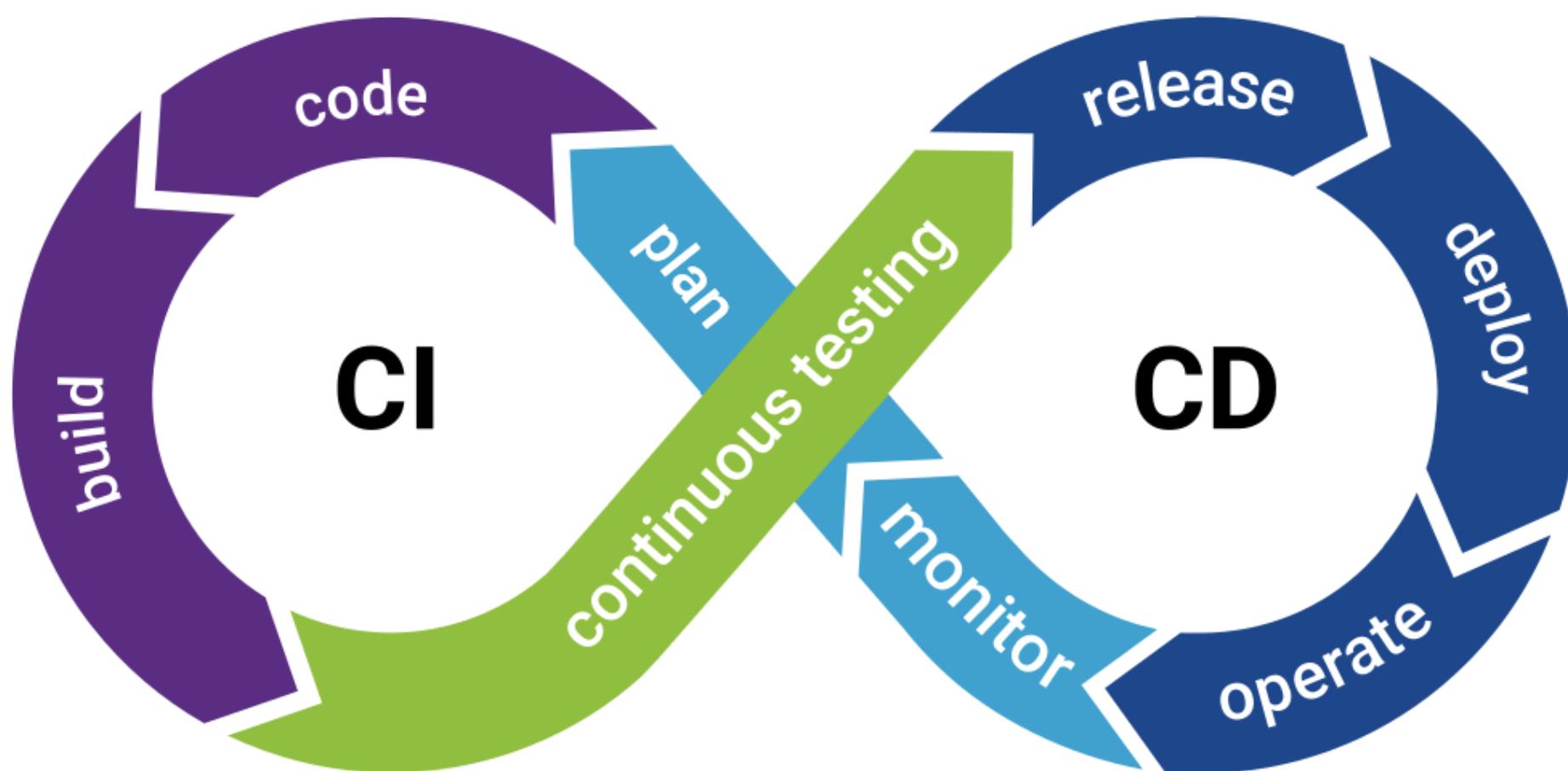
- ◆ Explore cloud services from providers like AWS, Azure, and GCP.
- ◆ Study how to design and deploy systems in the cloud.



Day 17

DevOps and Continuous Integration/Continuous Deployment (CI/CD)

- ◆ Learn about DevOps practices and CI/CD pipelines.
- ◆ Study how they are integrated into system design.



Day 18

Advanced Topics (Blockchain, IoT, etc.)

- ◆ Explore emerging technologies in system design.
- ◆ Study how blockchain and IoT influence system architecture.

!! Click To Download All Technical Notes !!



Notes

Download all technical notes for free & begin your interview preparations.





Day 19

Performance Optimization

- ◆ Learn about performance monitoring and optimization.
- ◆ Study profiling tools and techniques.

Day 20

Review and Practice

- ◆ Review key concepts from the past 20 days.
- ◆ Work on design exercises and case studies.

Day 21

Final Project

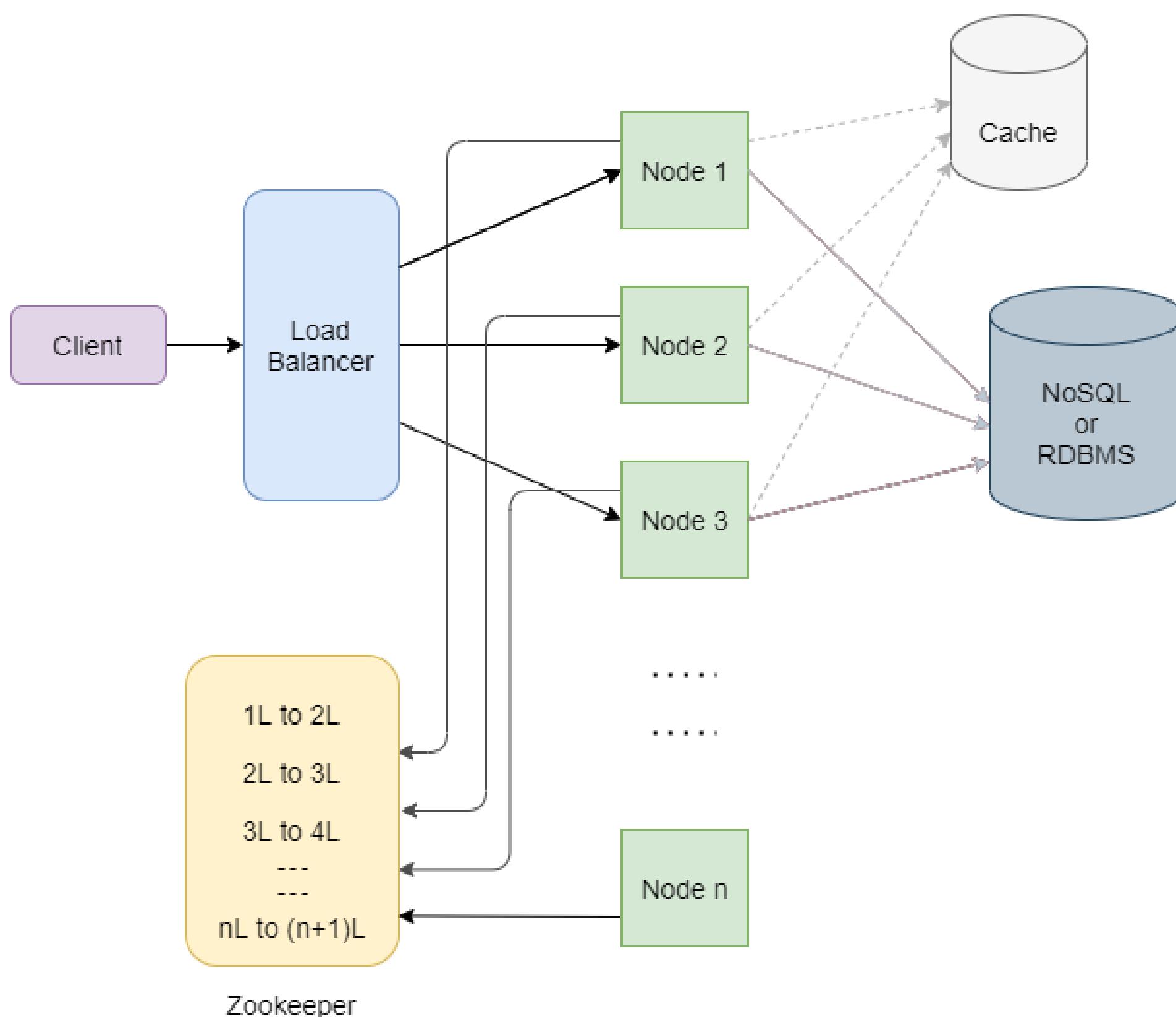
- ◆ Apply your knowledge to design a complete system.
- ◆ Present your design, and seek feedback from peers or mentors.



Important System Design Interview Questions

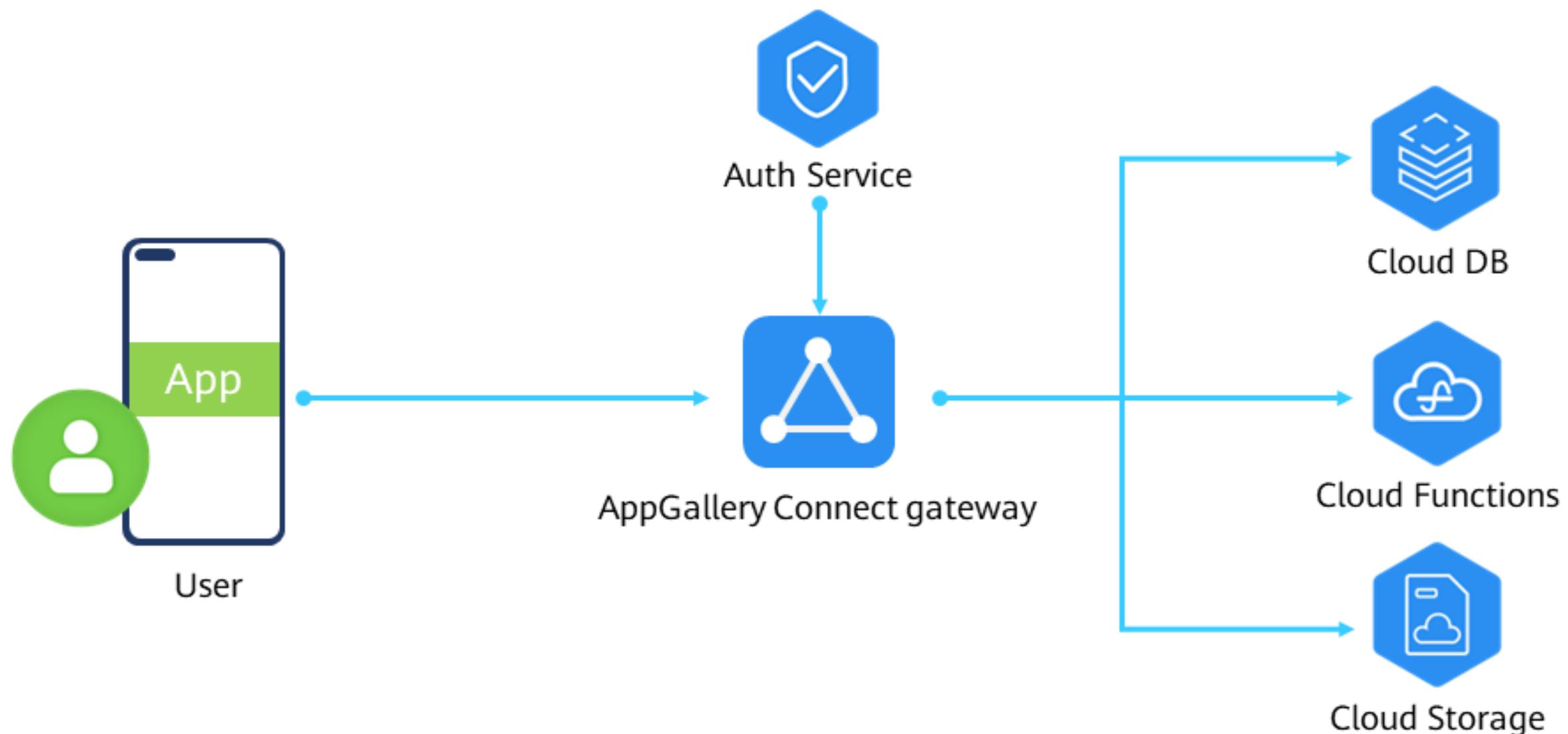
1. Design a URL shortening service like bit.ly:

- Design a service to shorten long URLs into short ones.
- Key Components: URL shortening algorithm, distributed storage, analytics.
- Additional: Customizable short URLs for branding.



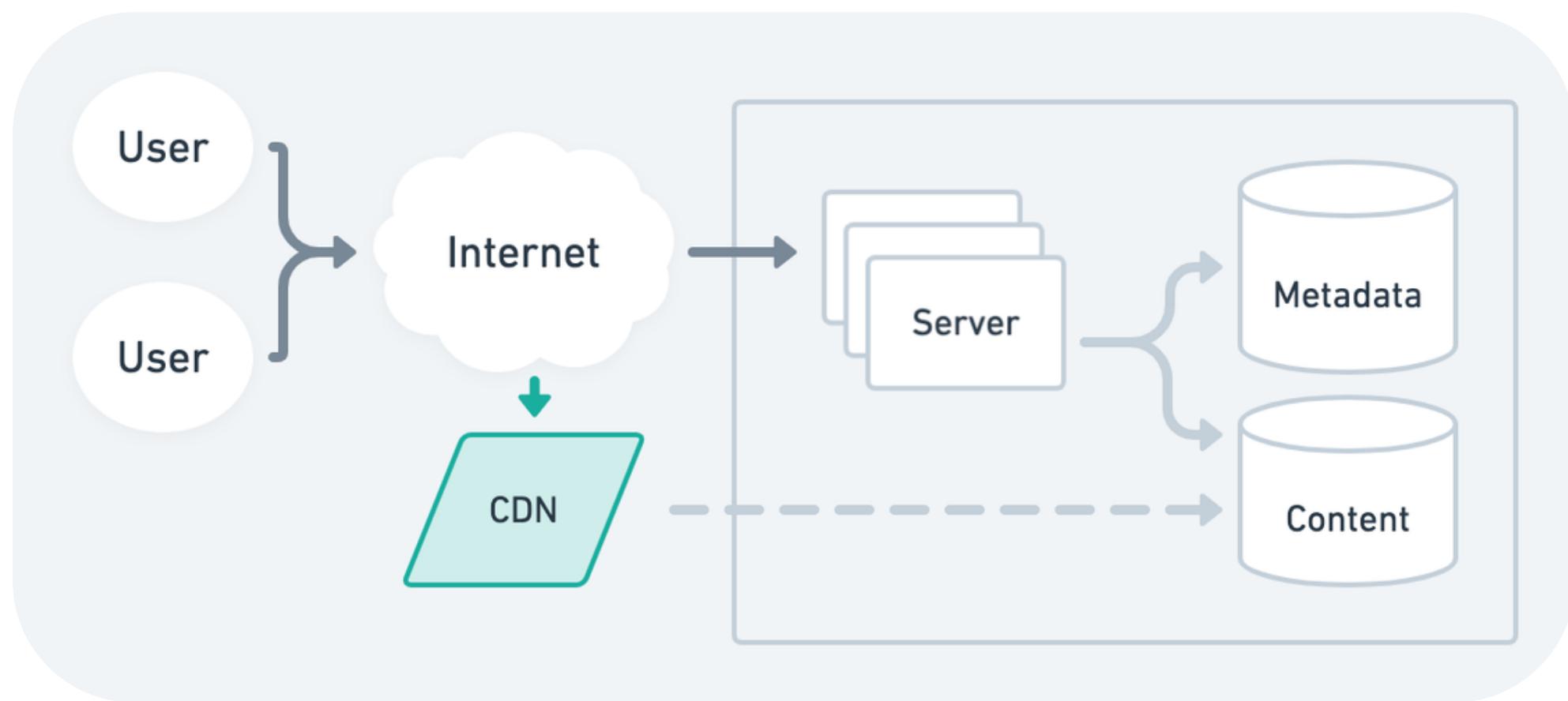
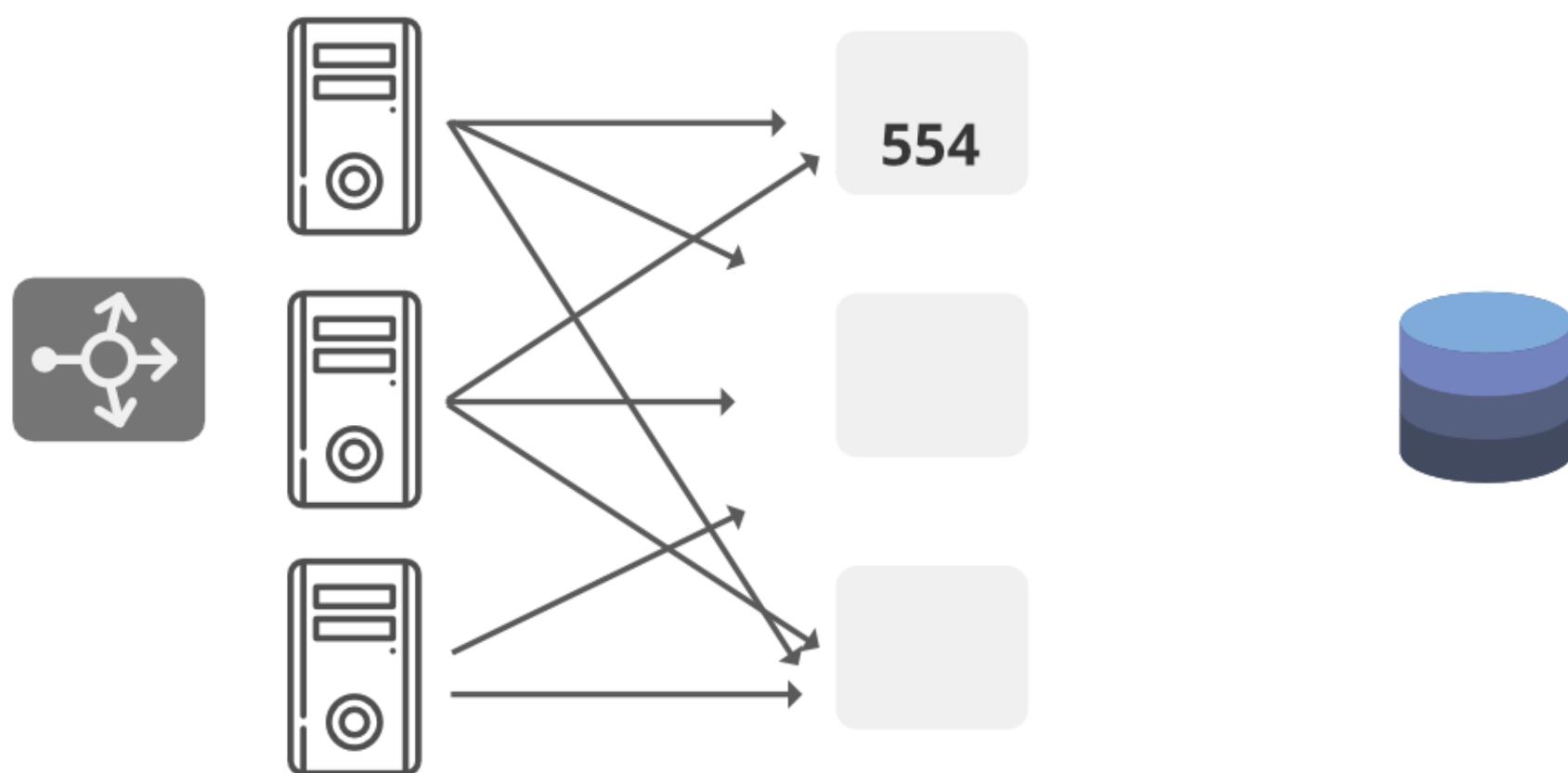
2. Design a Authentication Service:

- Design a centralized authentication service for secure user access.
- Key Components: User authentication, token generation, encryption.
- Additional: Multi-factor authentication for enhanced security.



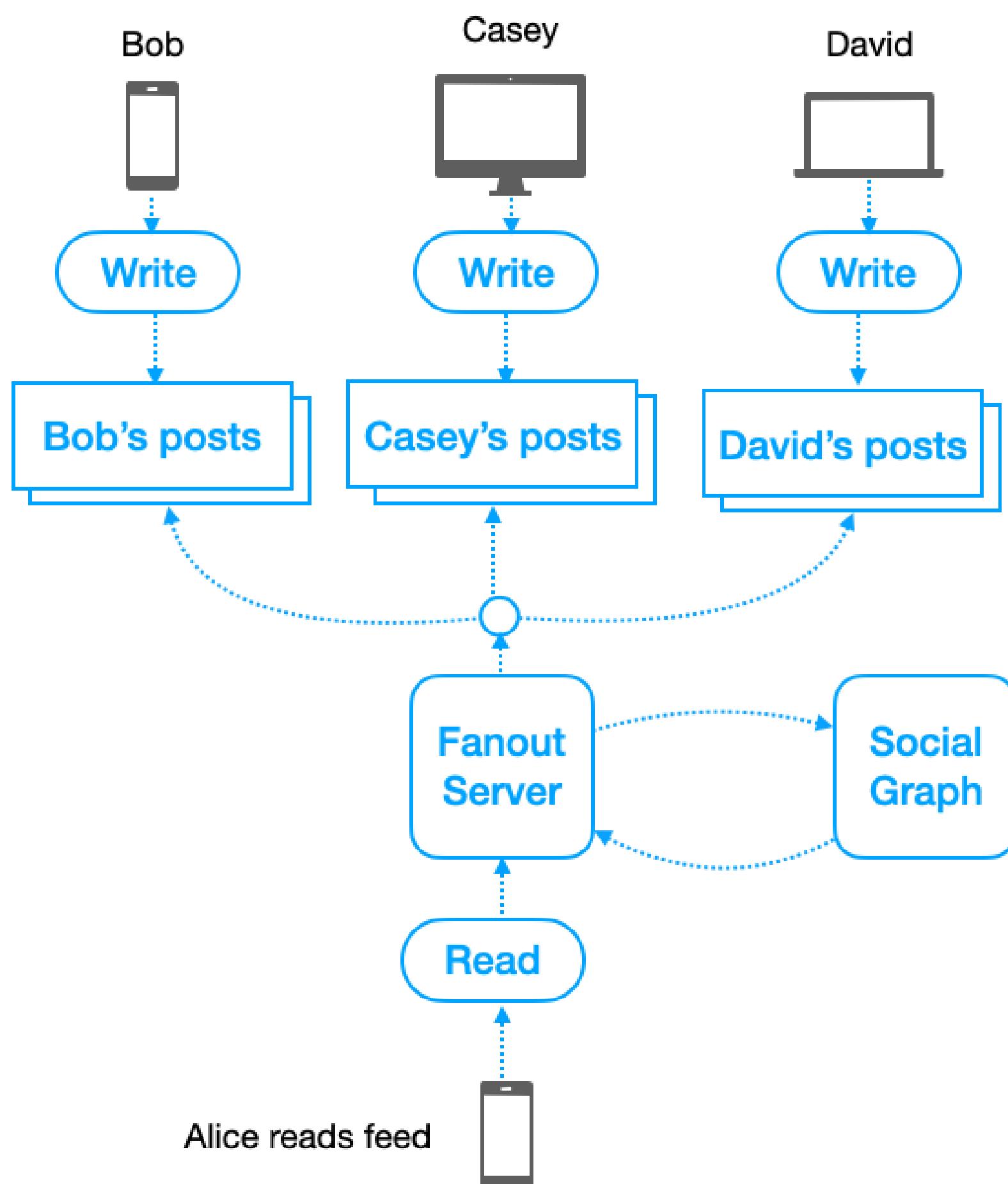
3. Design a Cache System:

- Create a caching system for frequently accessed data.
- Key Components: Cache eviction policies, cache coherence, distributed caching.
- Additional: Support for cache invalidation strategies.



4. Design a Social Media Feed:

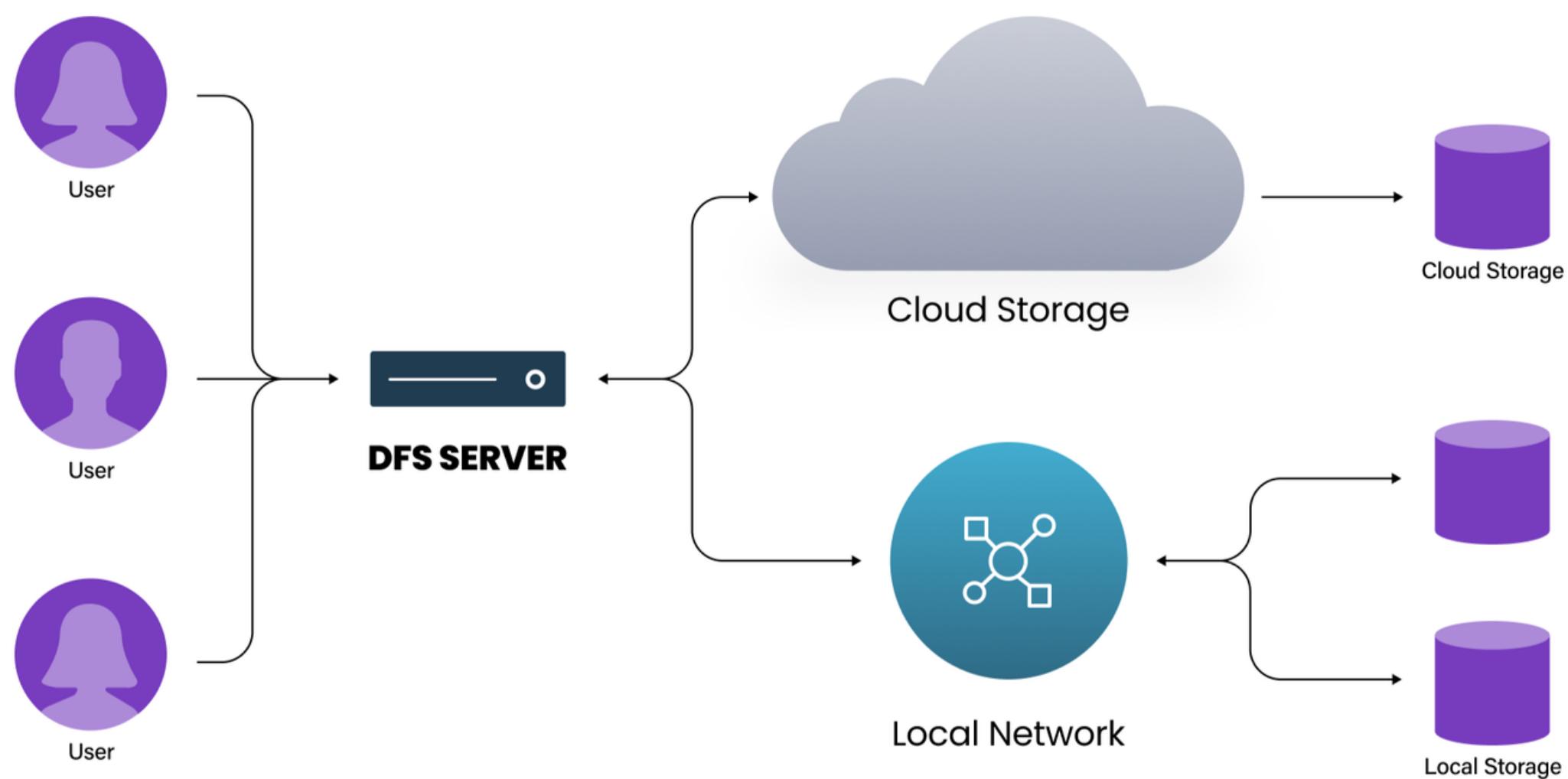
- Design a scalable social media feed system.
- Key Components: Feed generation, user timelines, content delivery.
- Additional: Real-time trending topics.





5. Design a Distributed File System:

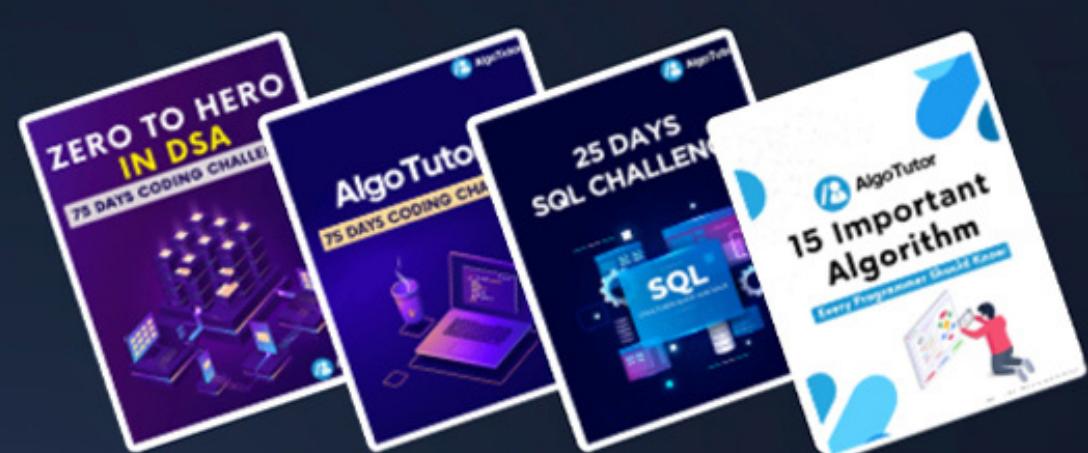
- Design a distributed file system like HDFS.
- Key Components: NameNode, DataNode, replication, fault tolerance.
- Additional: Support for large file streaming.



!! Click To Download All Technical Notes !!

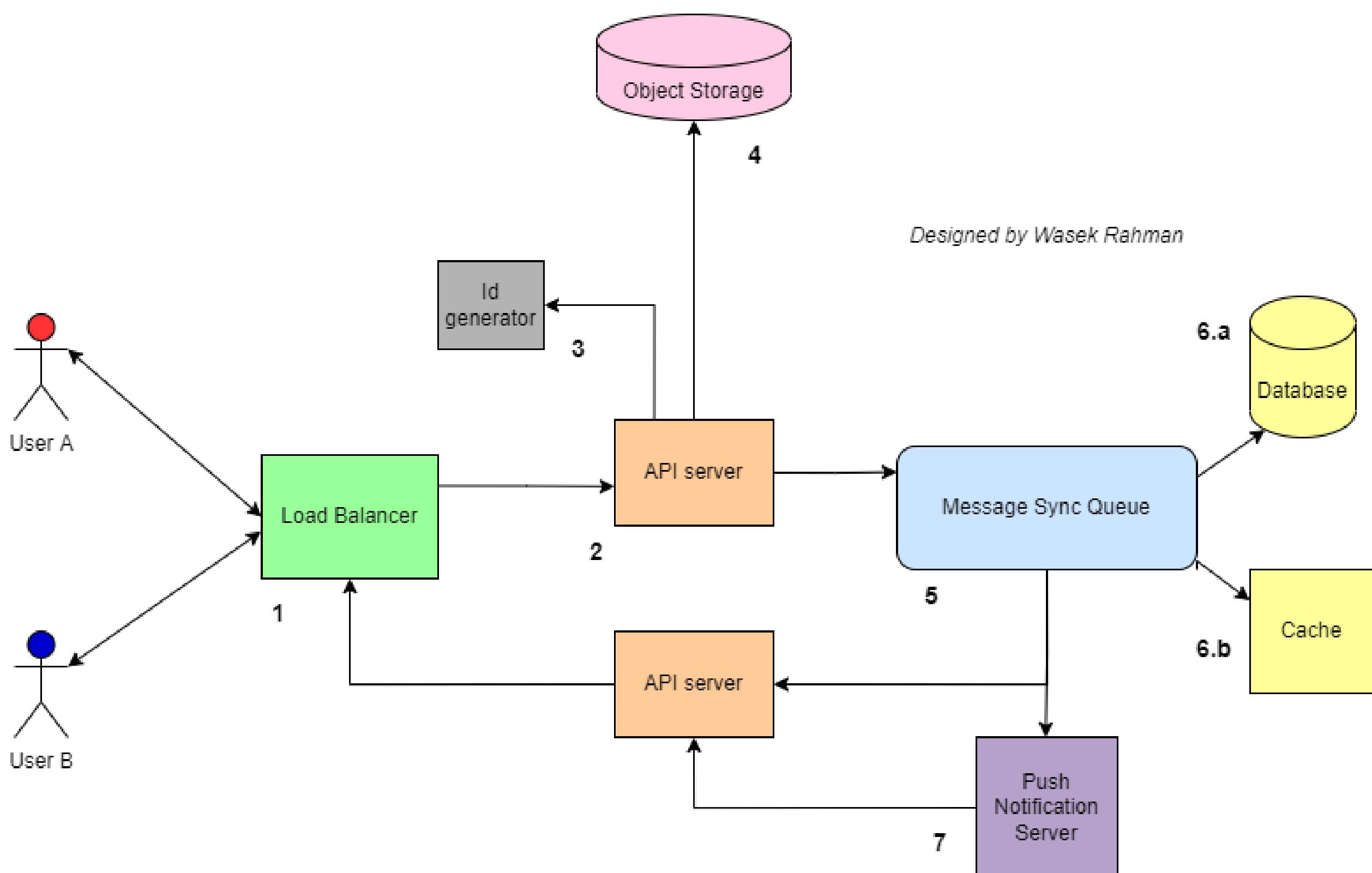
|Notes

Download all technical notes for free & begin your interview preparations.



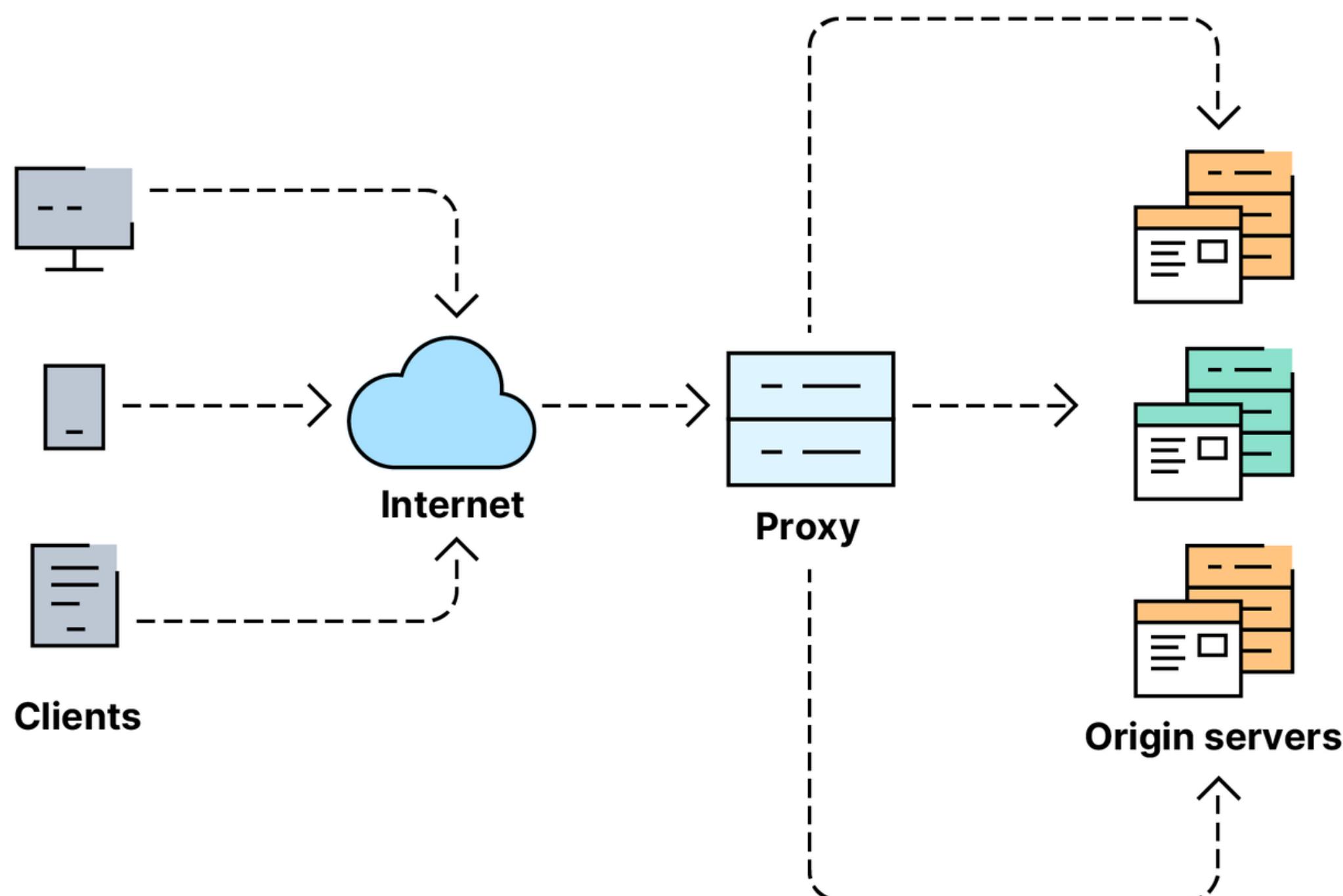
6. Design a Chat Application Design:

- Design a chat application for millions of users.
- Key Components: Messaging protocol, presence management, group chats.
- Additional: End-to-end encryption for security.



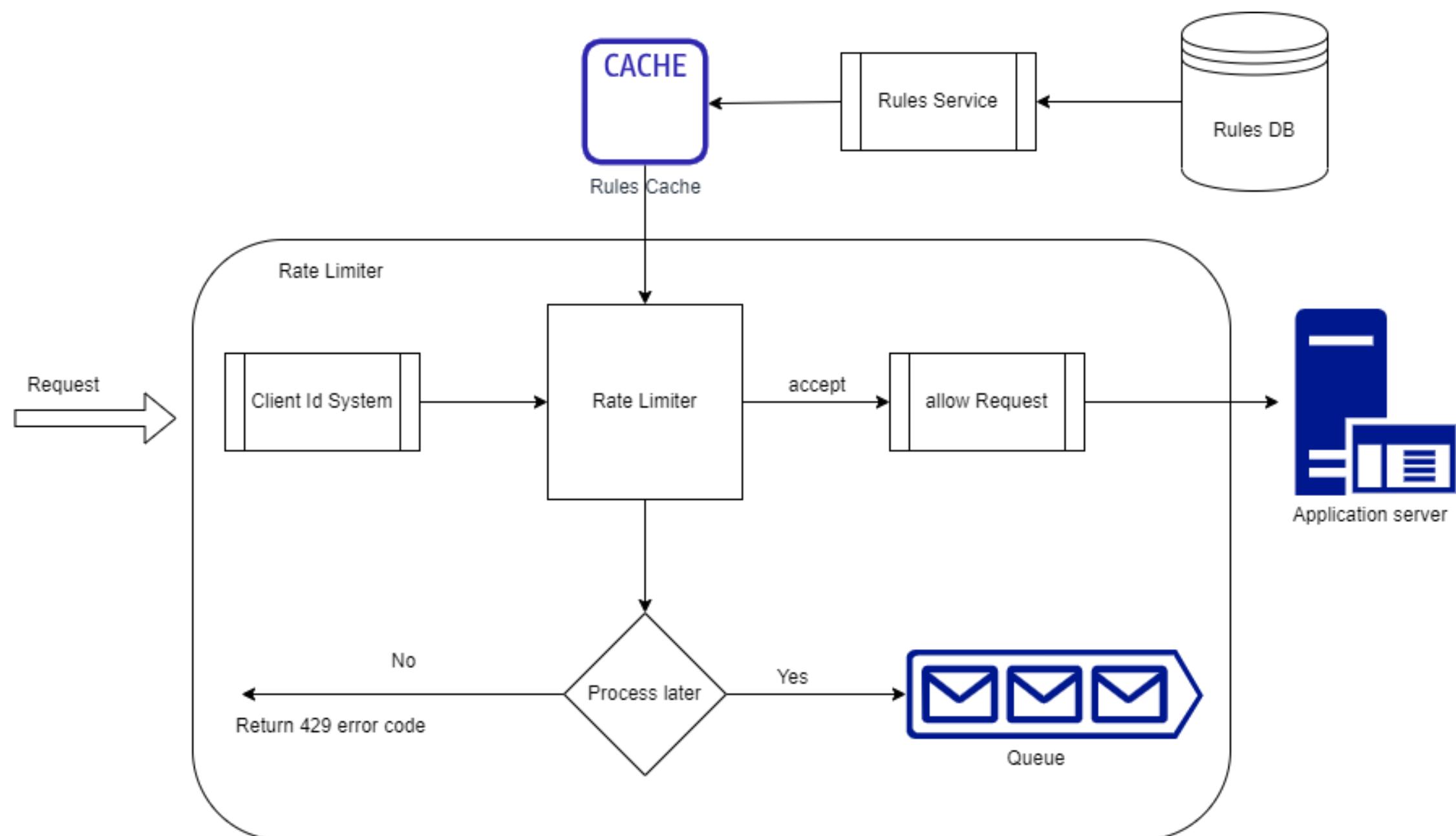
7. Design a Load Balancer:

- Design a load balancing system for a web application.
- Key Components: Load balancing algorithms, health checks, server scaling.
- Additional: Global load balancing for multi-region applications.



8. Design a Rate Limiter:

- Design a rate limiter to prevent service abuse.
- Key Components: Token bucket algorithm, distributed rate limiting.
- Additional: Dynamic rate limiting based on user behavior.



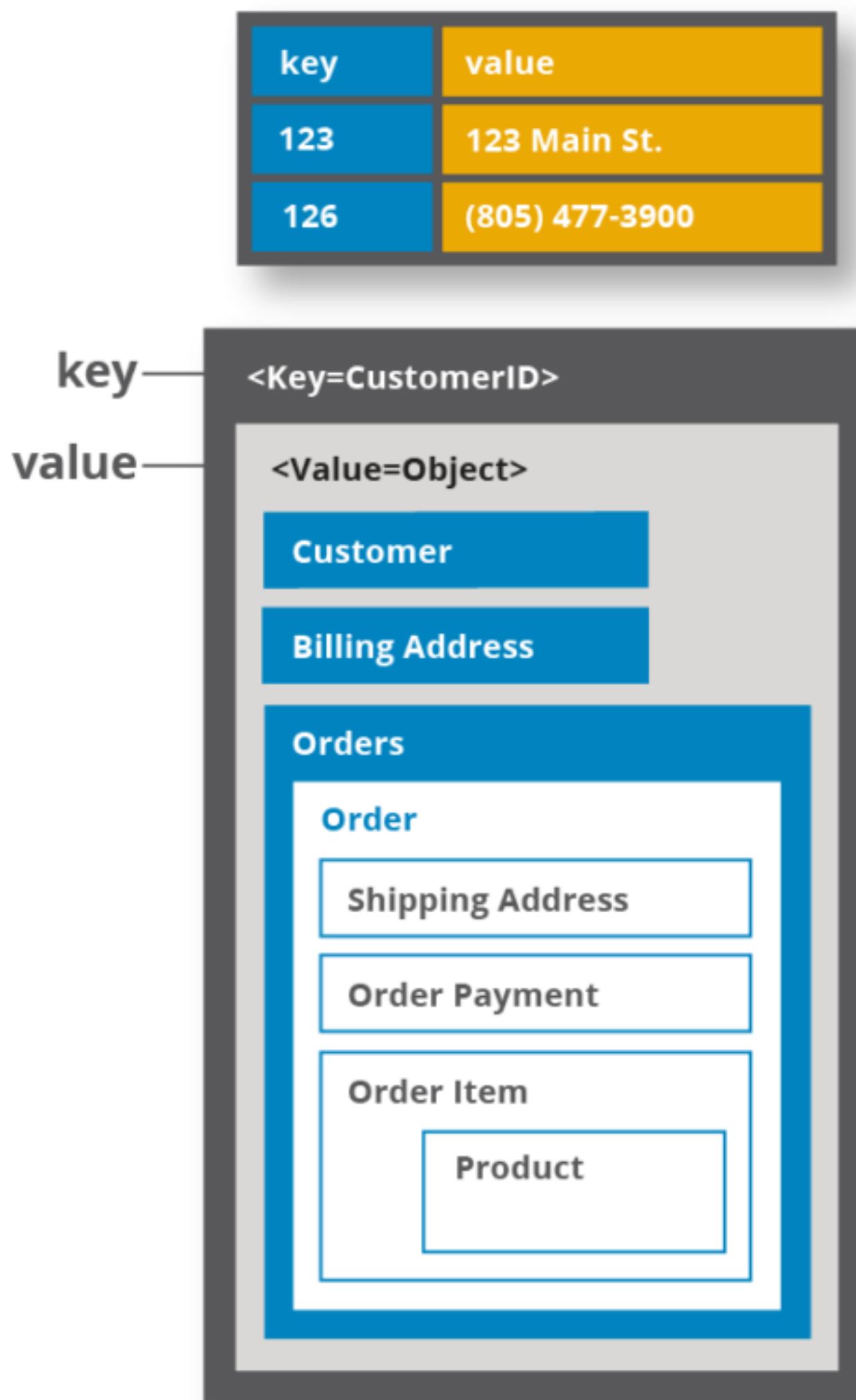
For Admission Enquiry

+91-7260058093

info@algotutor.io

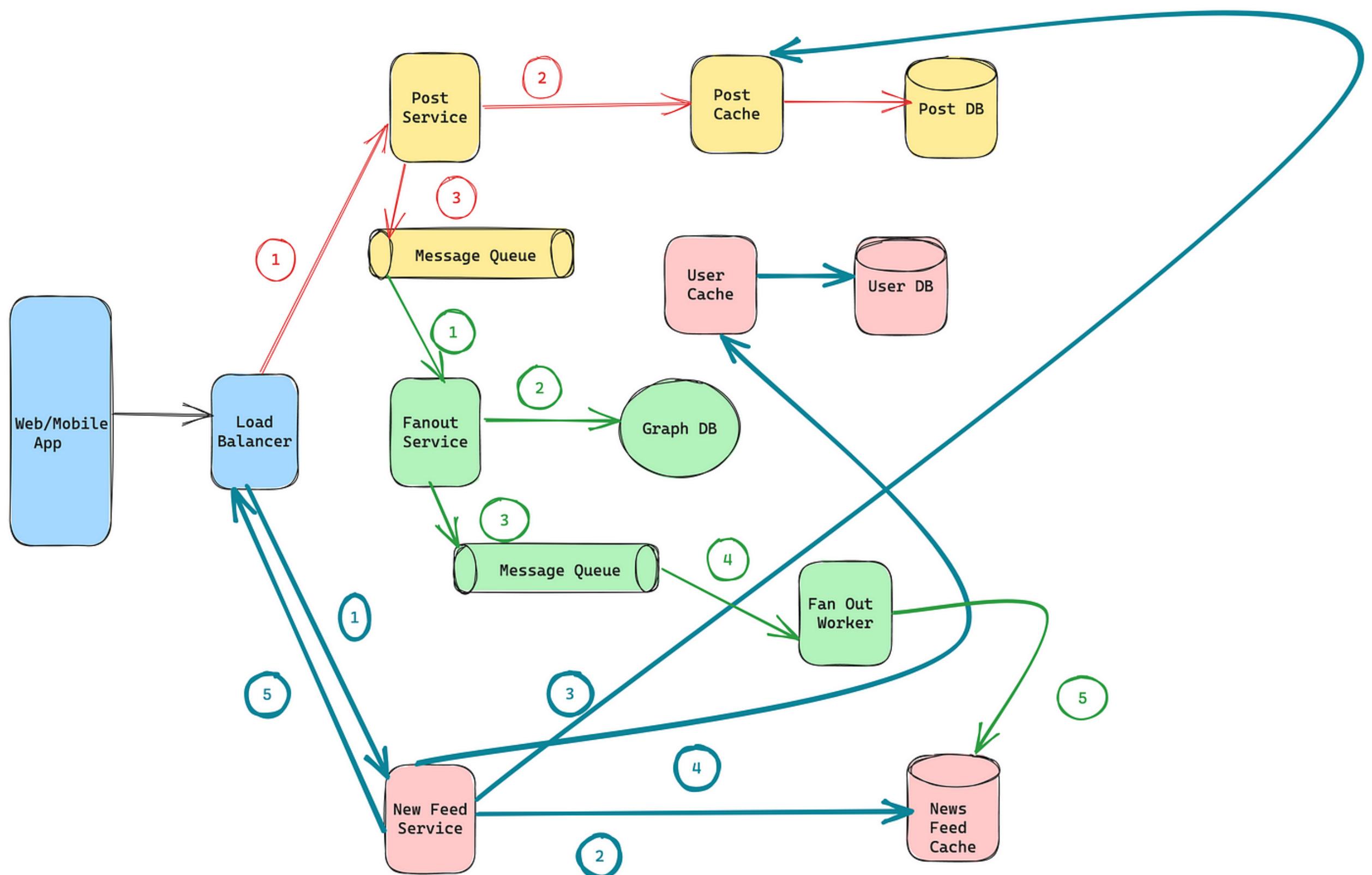
9. Design a Key-Value Store:

- Design a distributed key-value store like Redis.
- Key Components: Partitioning, replication, consistency, fault tolerance.
- Additional: Support for secondary indexes.



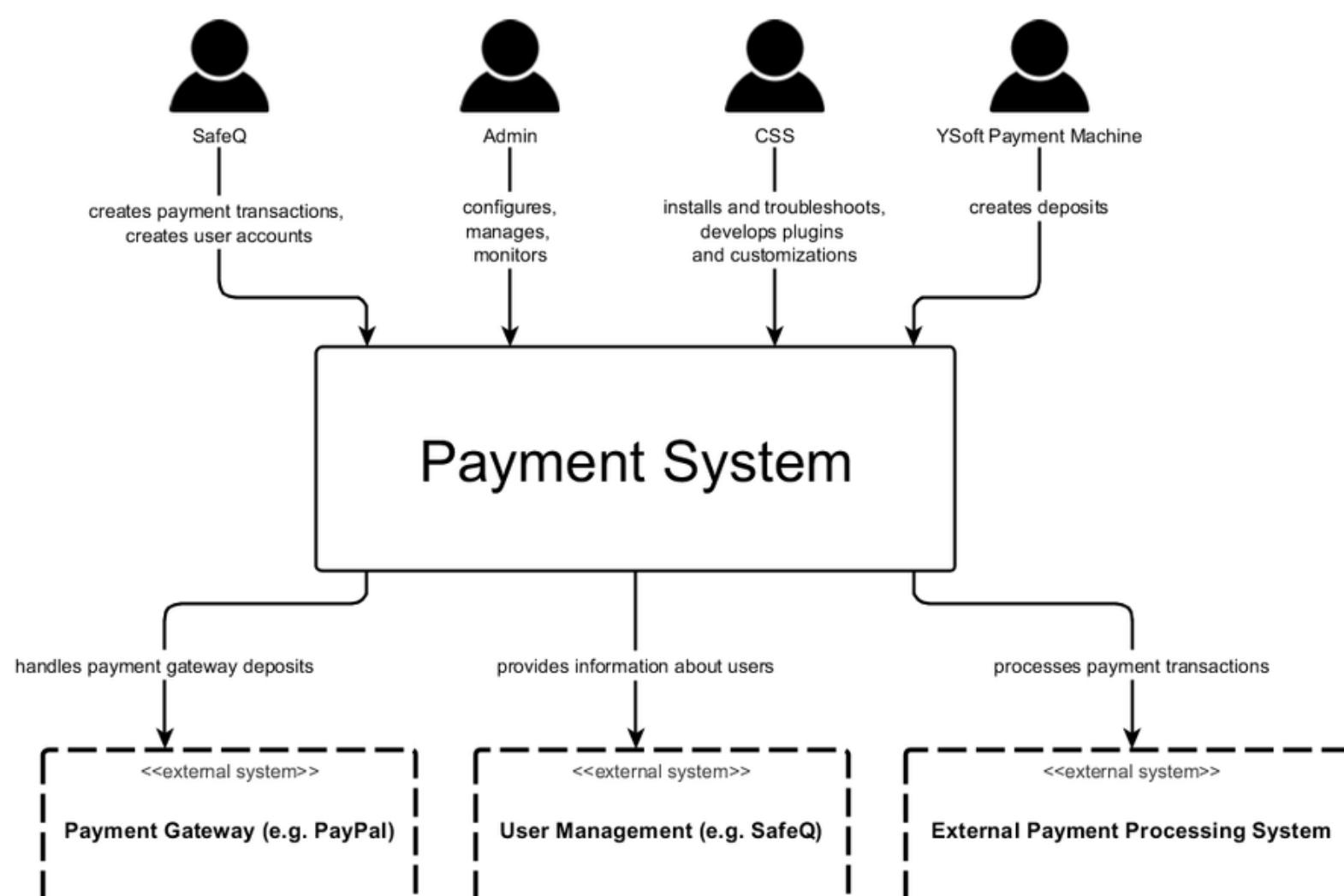
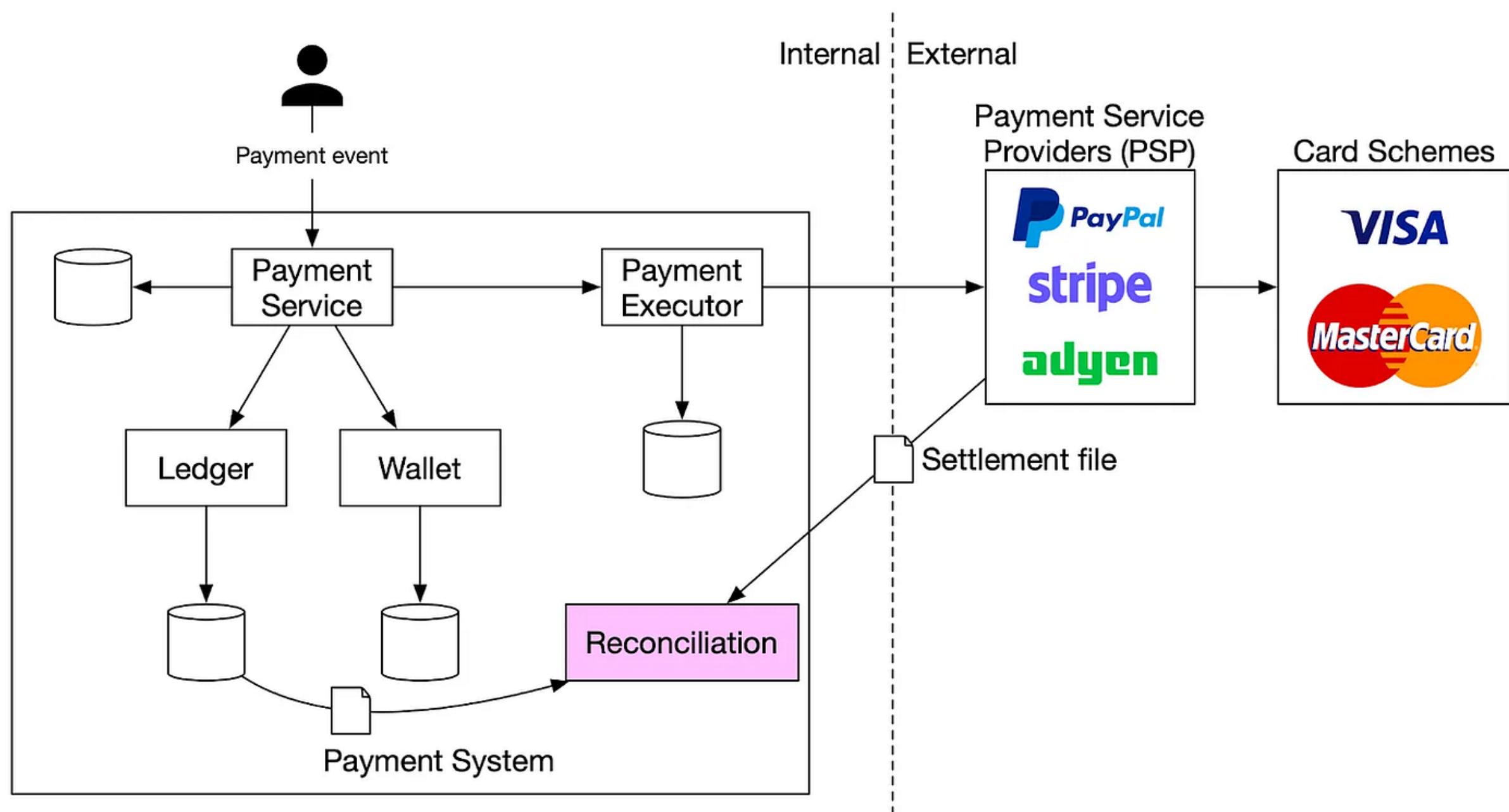
10. Design a scalable news feed system (like Facebook's news feed):

- Design a system for generating personalized news feeds.
- Key Components: Content aggregation, user preferences, ranking algorithm.
- Additional: Personalized notifications for breaking news.



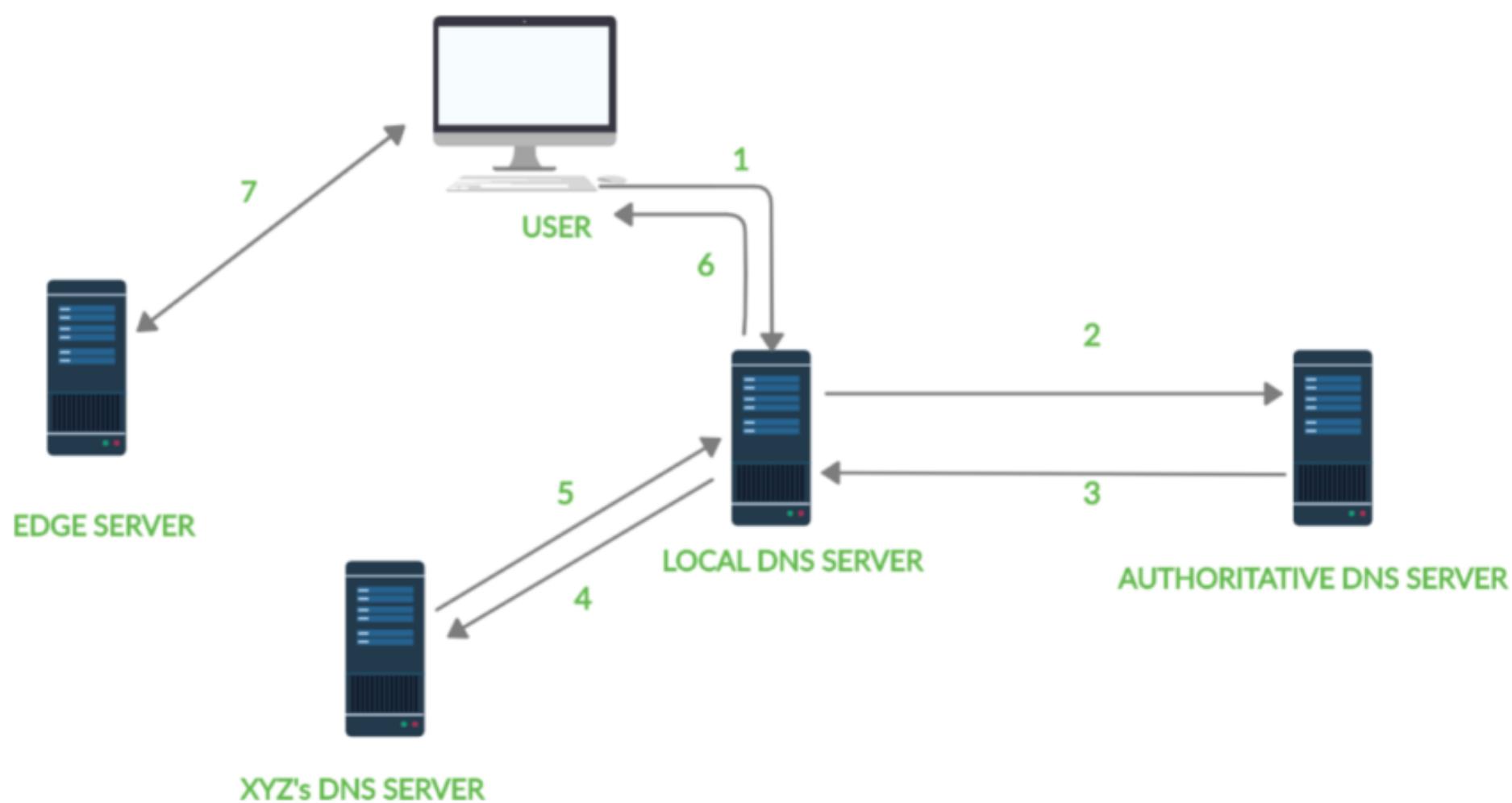
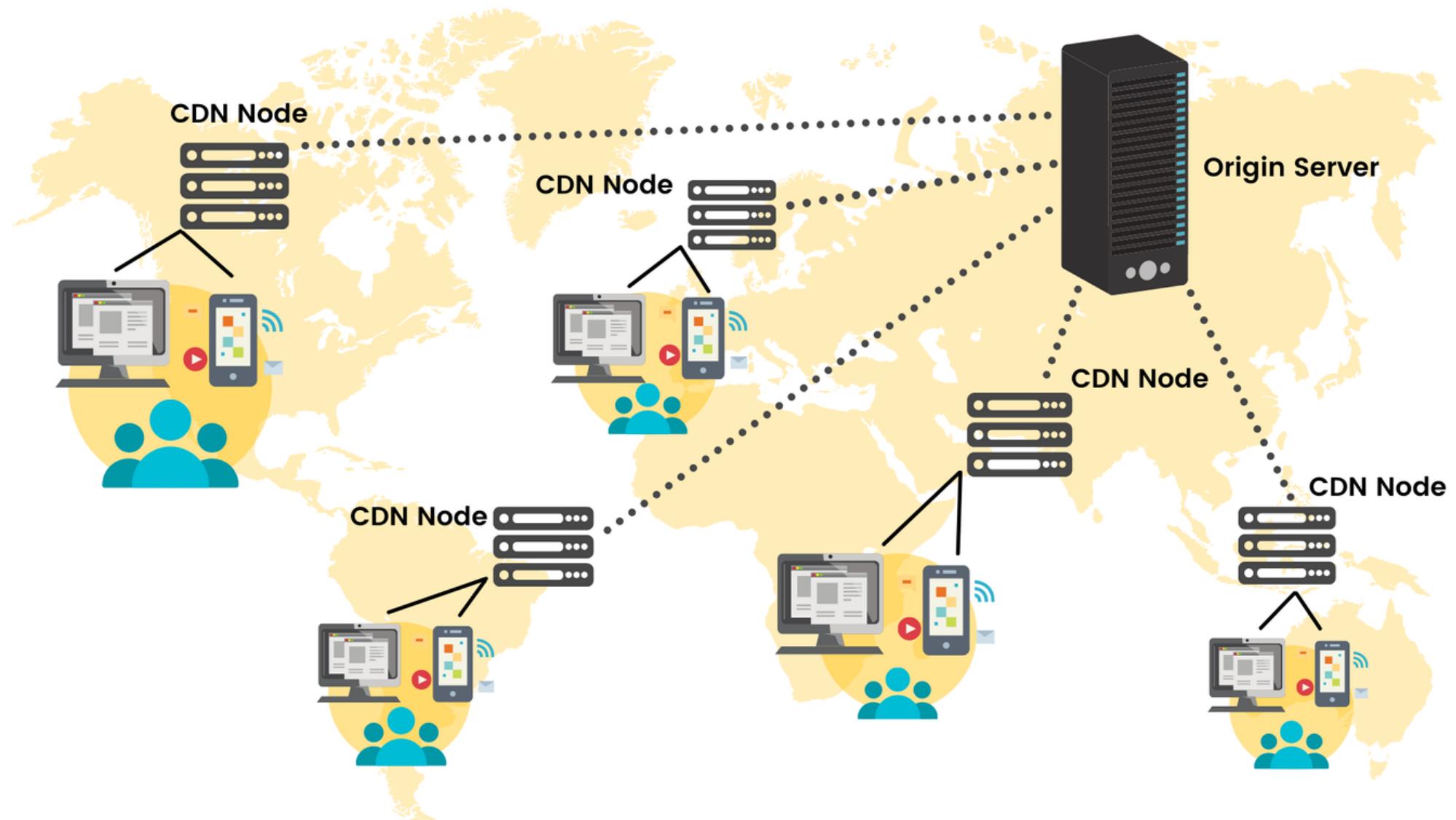
11. Design a Payment Gateway System:

- Design a payment gateway for online transactions.
- Key Components: Payment processing, security, fraud detection.
- Additional: Two-factor authentication for added security.



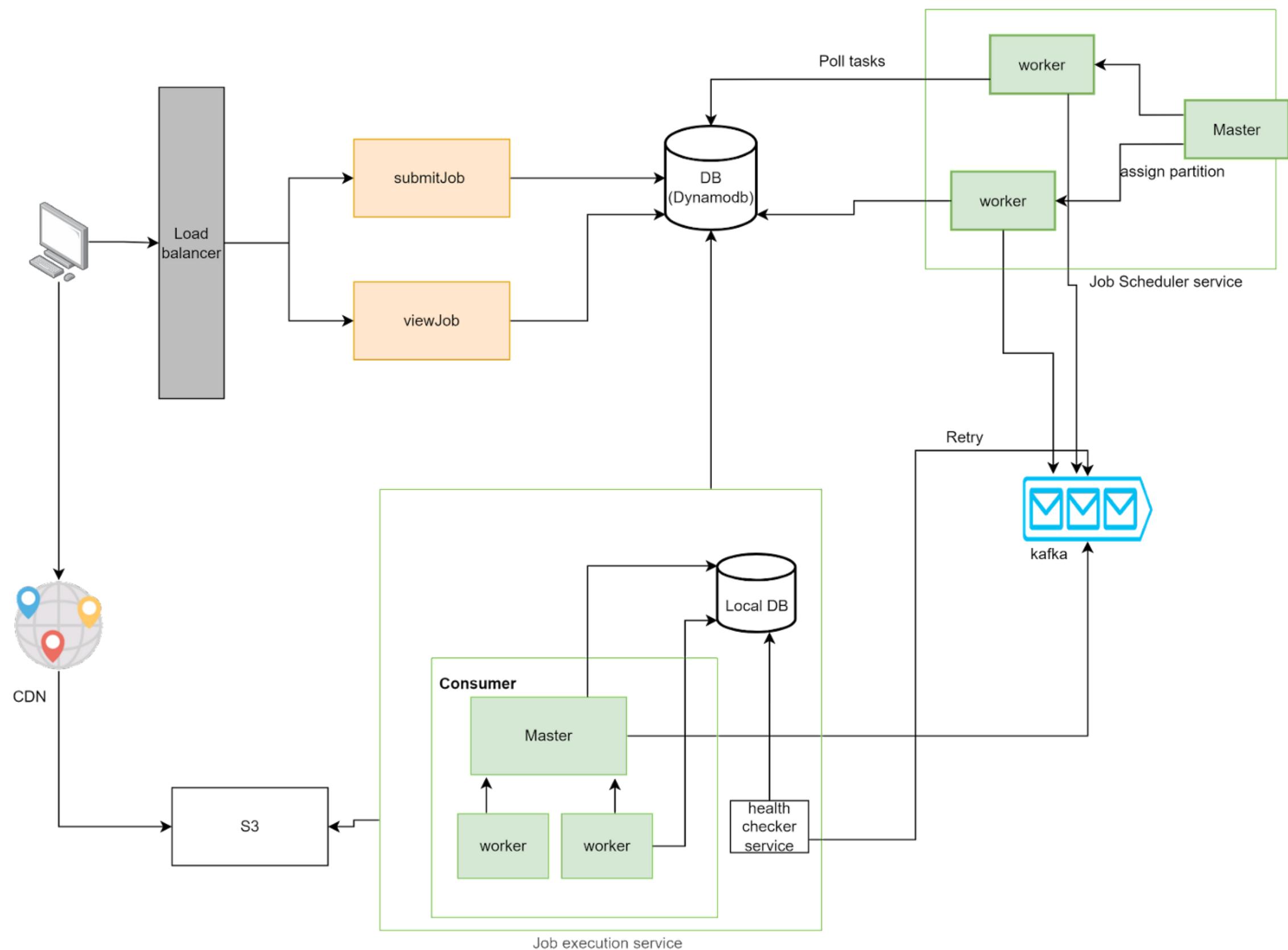
12. Design a Content Delivery Network (CDN):

- Design a CDN for efficient content distribution.
- Key Components: Edge servers, caching, content routing.
- Additional: Dynamic content caching for frequently updated content.



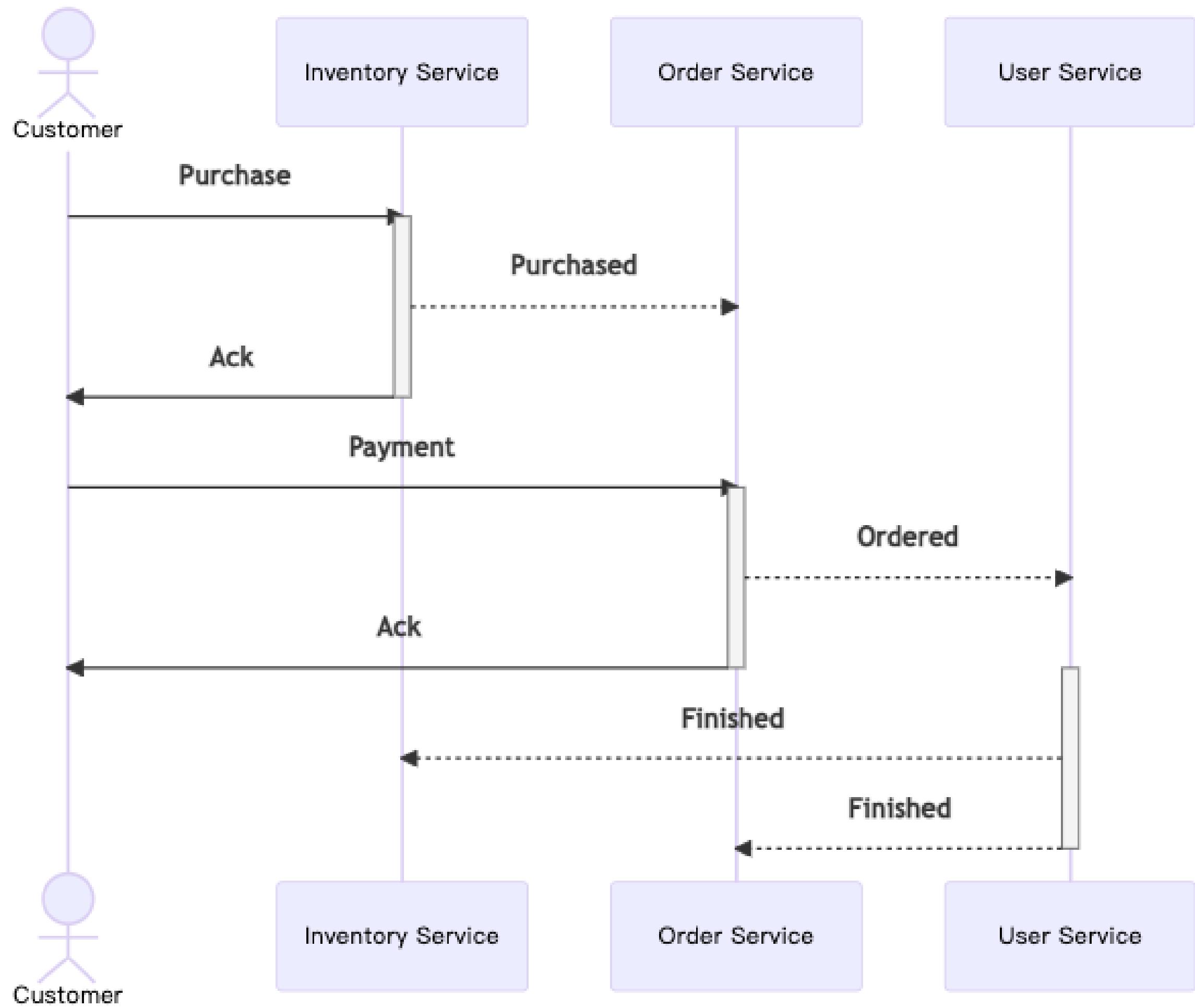
13. Design a Job Scheduler:

- Design a job scheduler for a distributed computing environment.
- Key Components: Job queue, scheduling algorithms, fault tolerance.
- Additional: Prioritization of critical jobs.



14. Design a Online Marketplace:

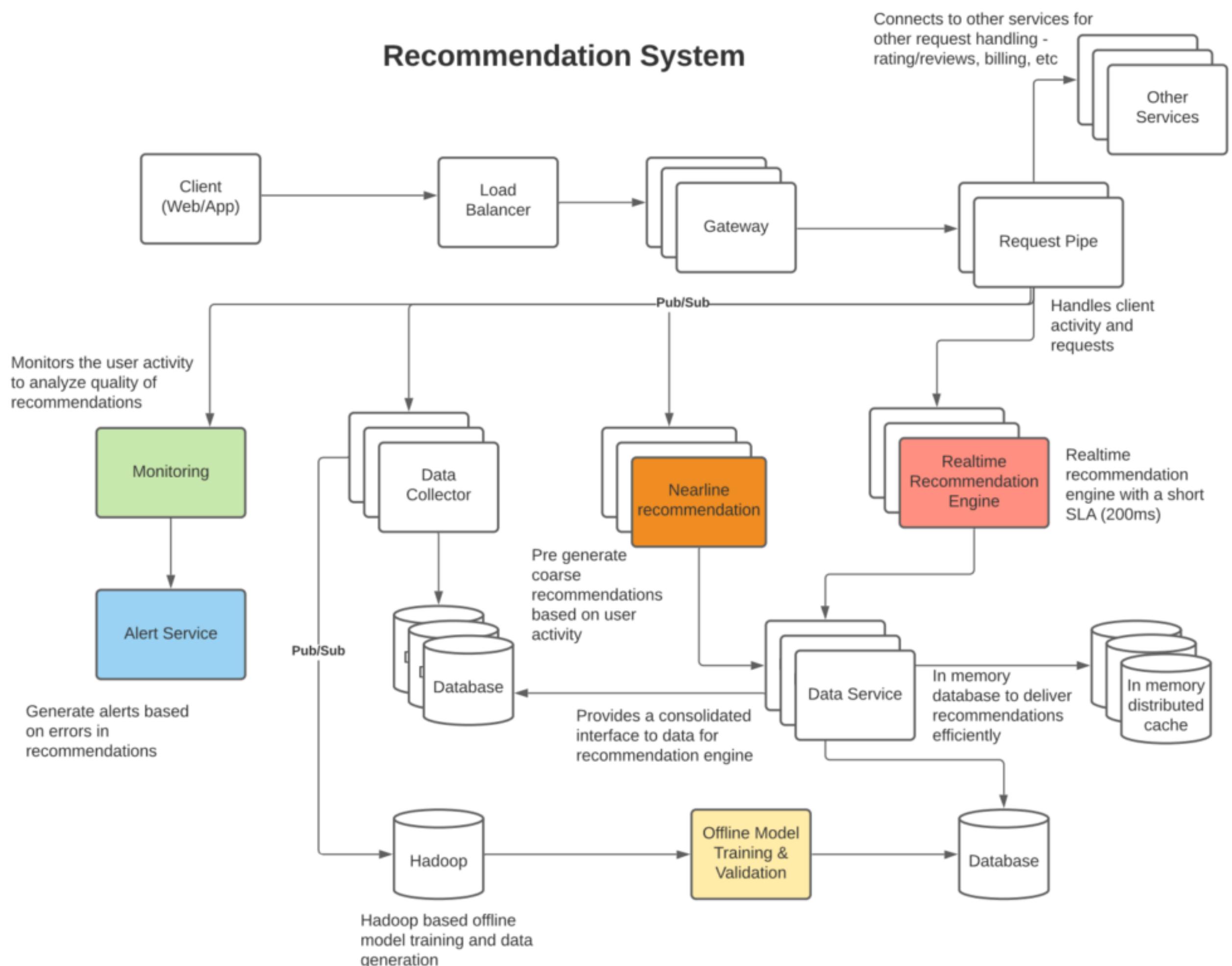
- Design an online marketplace for buying and selling goods.
- Key Components: Listings, search, user reviews, transaction processing.
- Additional: Integrated customer support and dispute resolution.





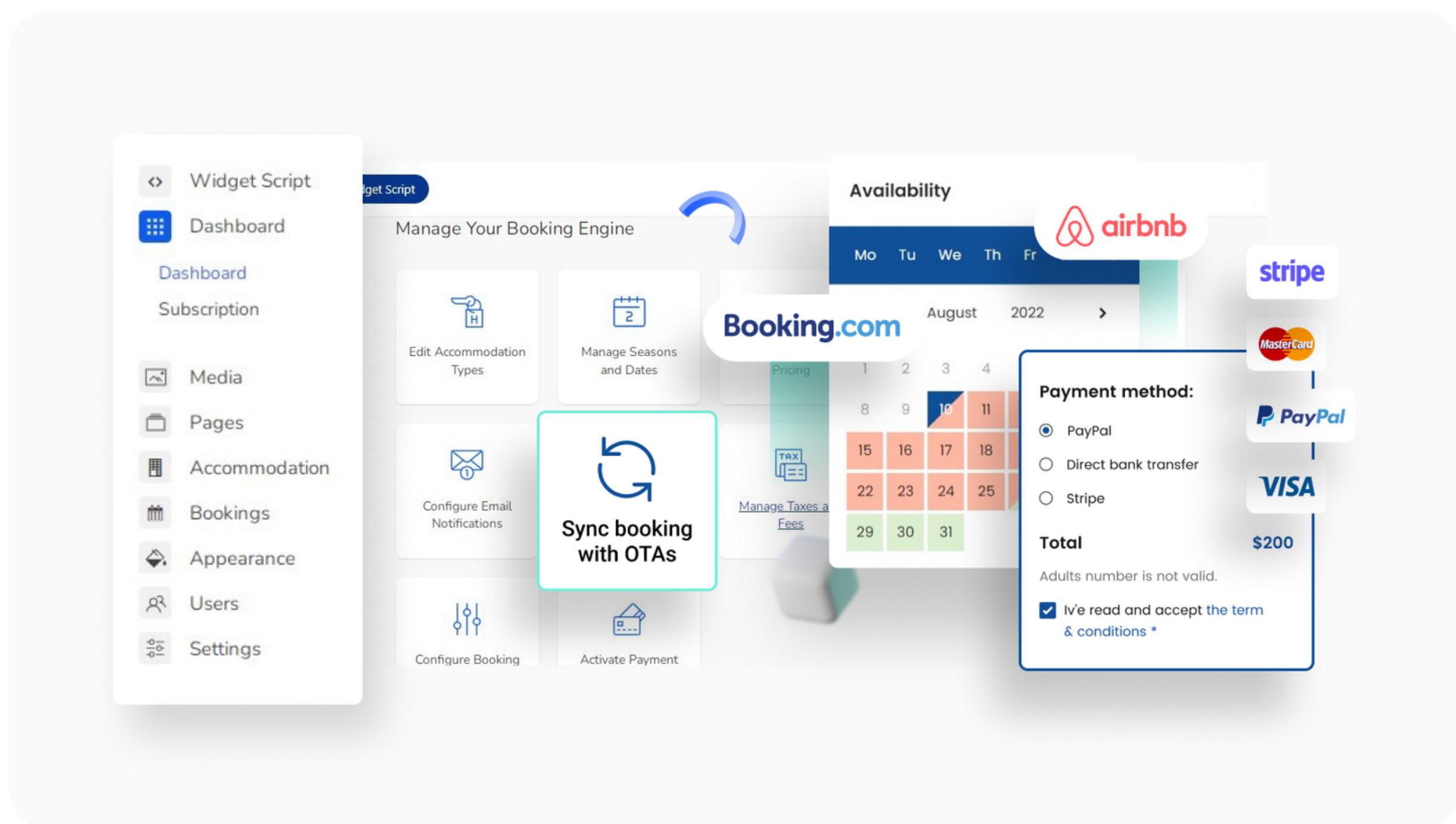
15. Design a Recommendation System:

- Design a recommendation system for a streaming service.
- Key Components: Collaborative filtering, content-based filtering, personalization.
- Additional: Continuous learning for adapting to user preferences.



16. Design a Hotel Reservation System:

- Design a system for hotel room reservations.
- Key Components: Booking engine, inventory management, reservation system.
- Additional: Integration with external booking platforms.



!! Click To Download All Technical Notes !!

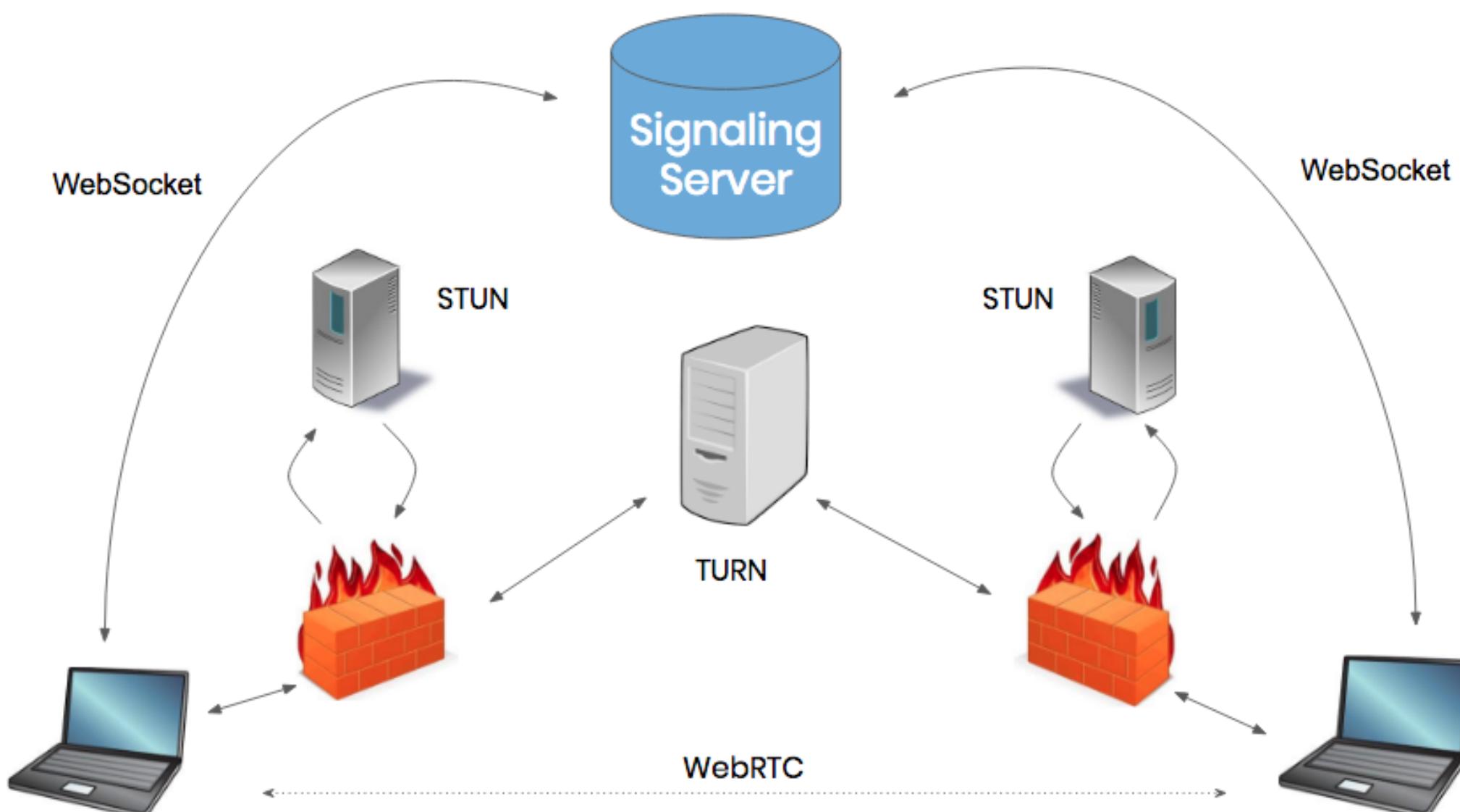
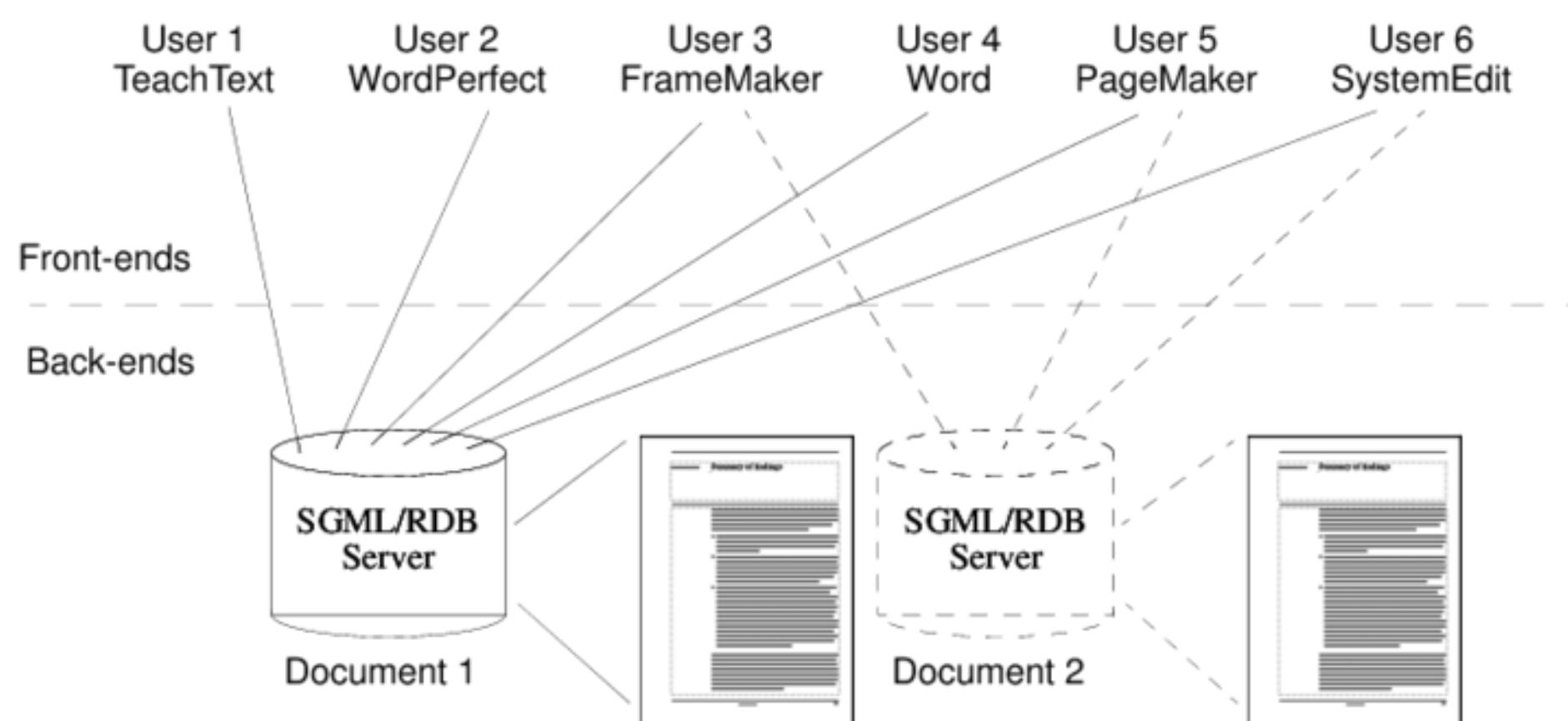
|Notes

Download all technical notes for free & begin your interview preparations.



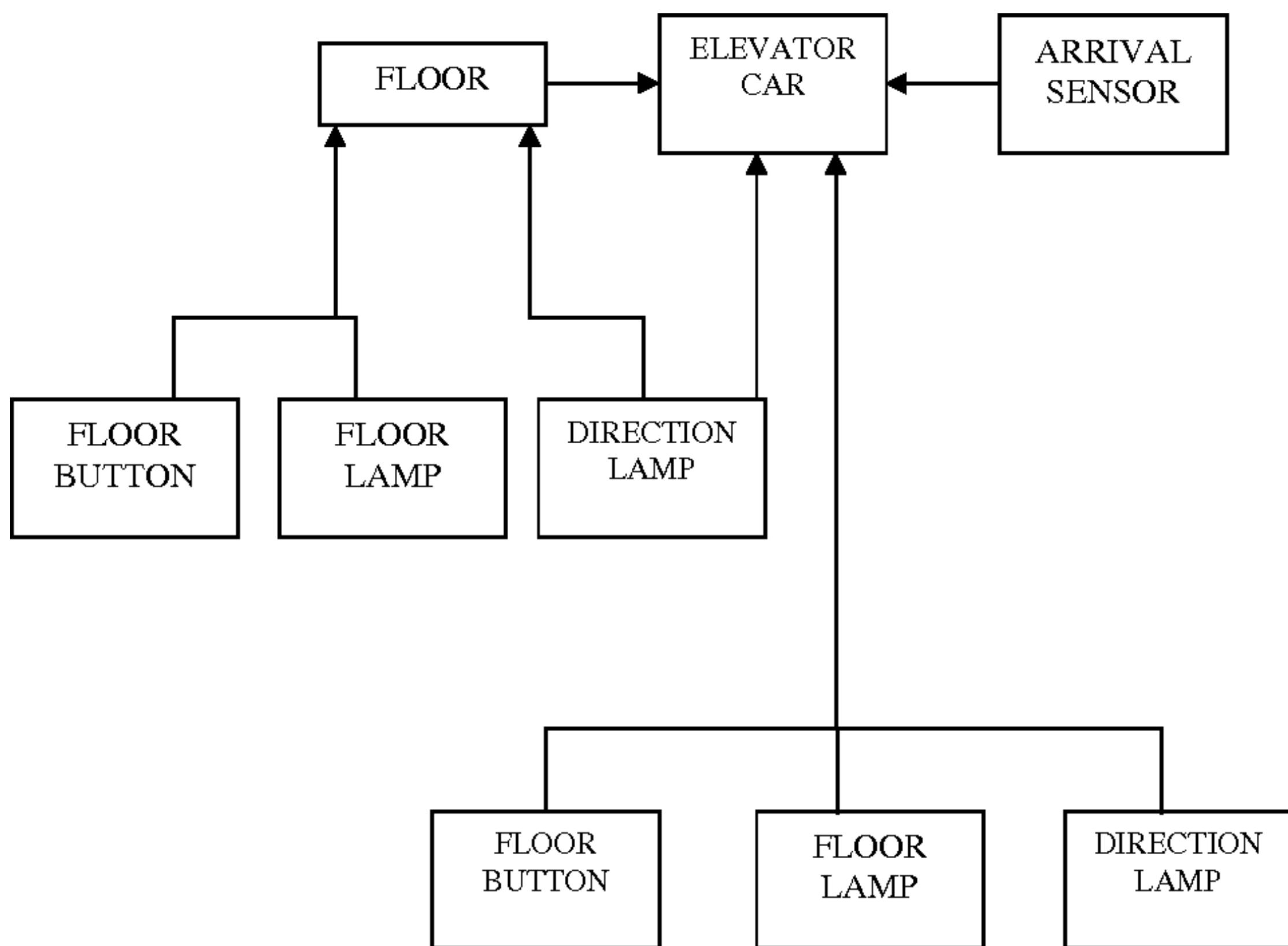
17. Design a Collaborative Editing System:

- Design a collaborative text editing system.
- Key Components: Operational transformation, real-time synchronization.
- Additional: Version control for collaborative edits.



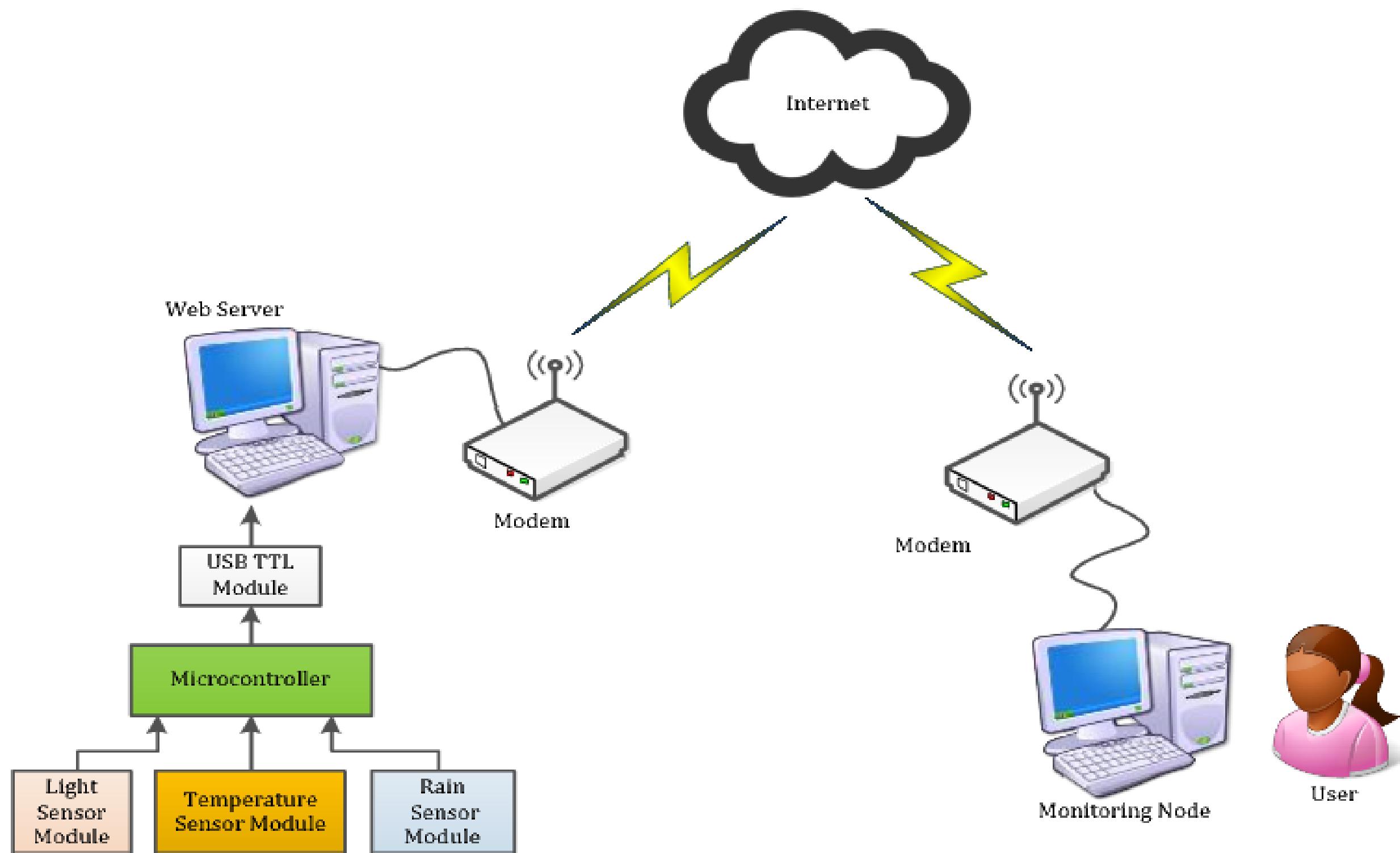
18. Design a Elevator System:

- Design a control system for a multi-floor building's elevators.
- Key Components: Scheduling algorithm, elevator control, fault tolerance.
- Additional: Emergency evacuation protocols.



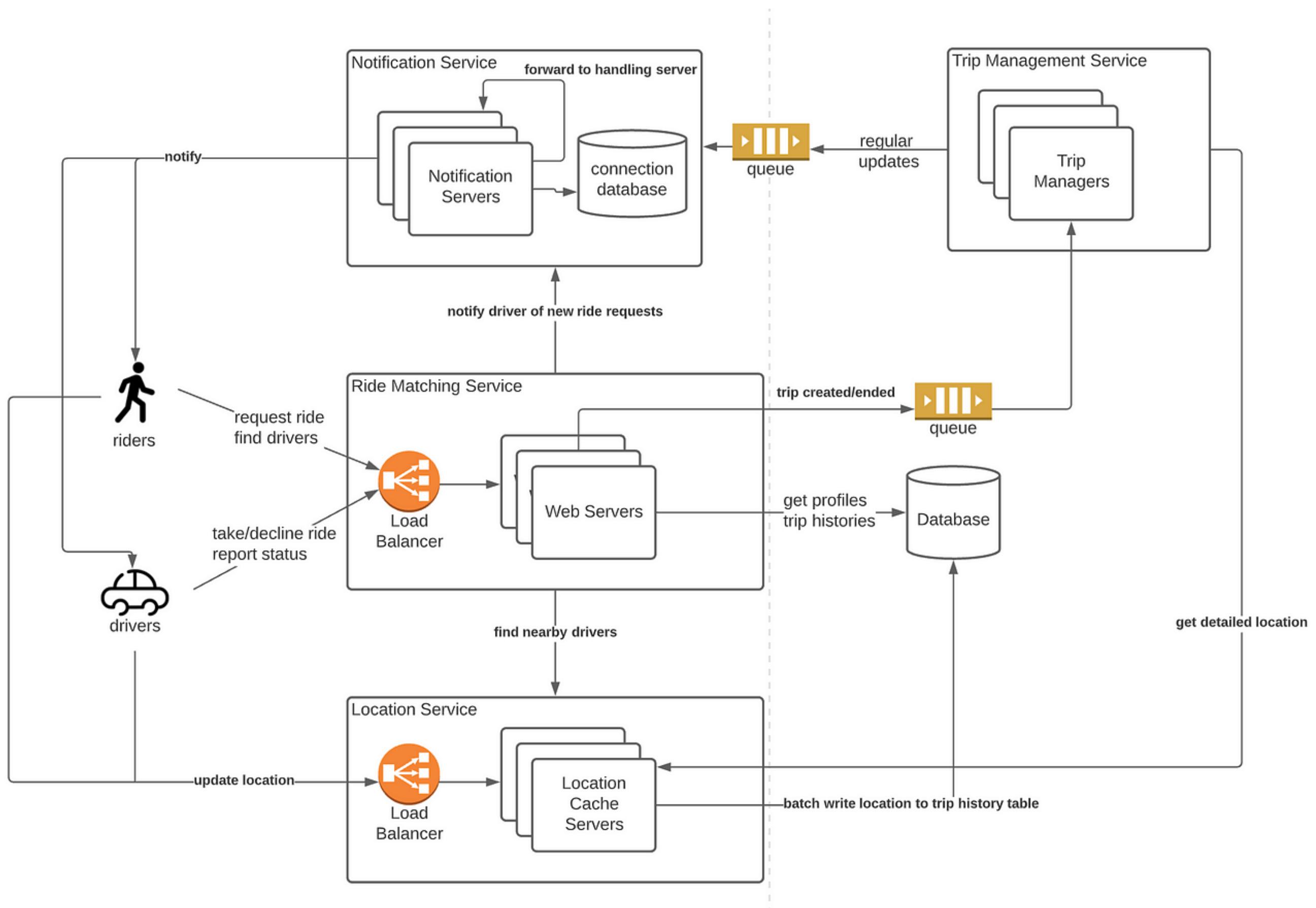
19. Design a Weather Service:

- Design a system for providing weather information.
- Key Components: Data acquisition, storage, API, real-time updates.
- Additional: Historical weather data storage and retrieval.



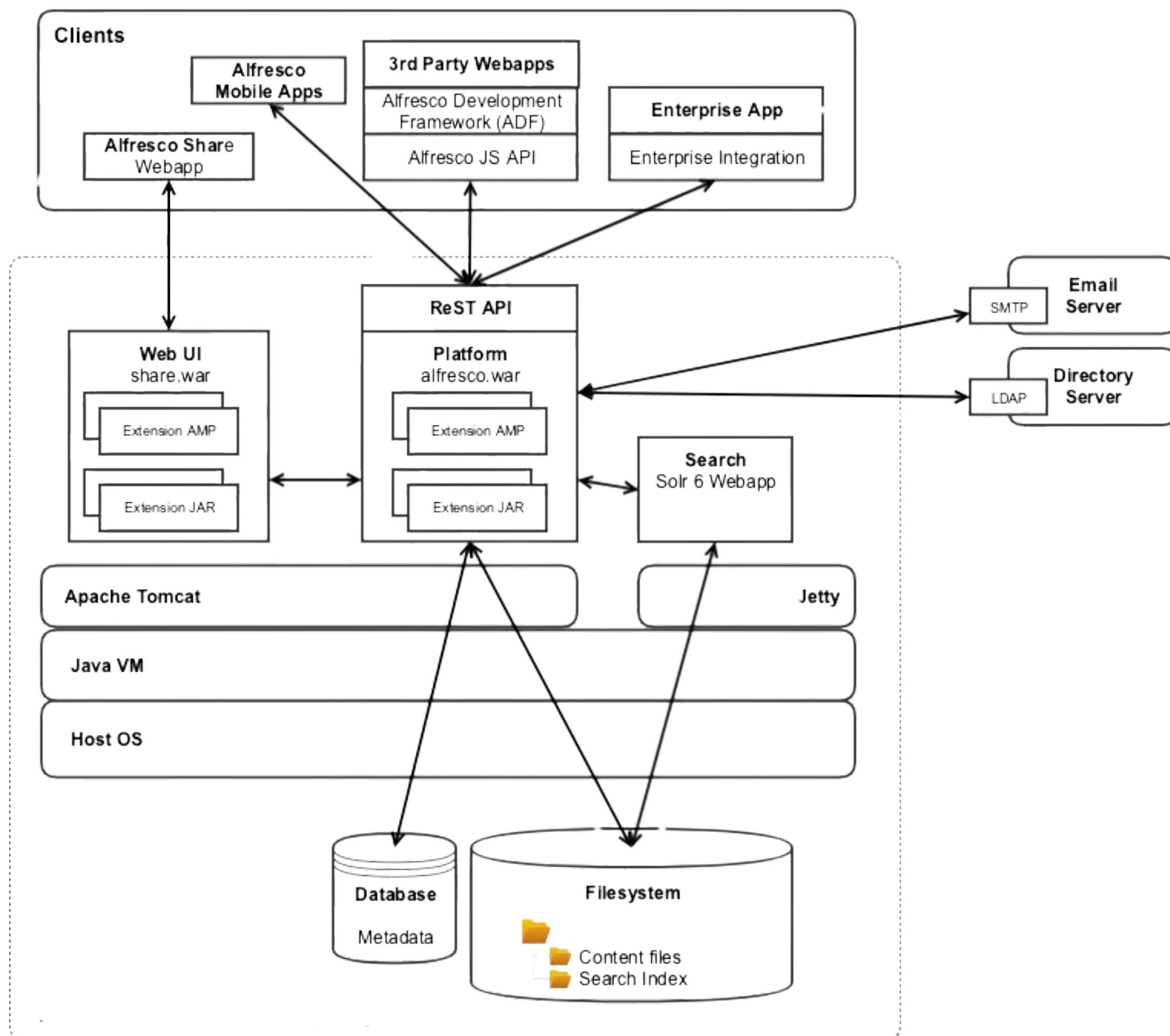
20. Design a Ride-Sharing System:

- Design a system for a ride-sharing service like Uber.
- Key Components: Matching algorithm, user tracking, pricing.
- Additional: Dynamic pricing based on demand and supply.



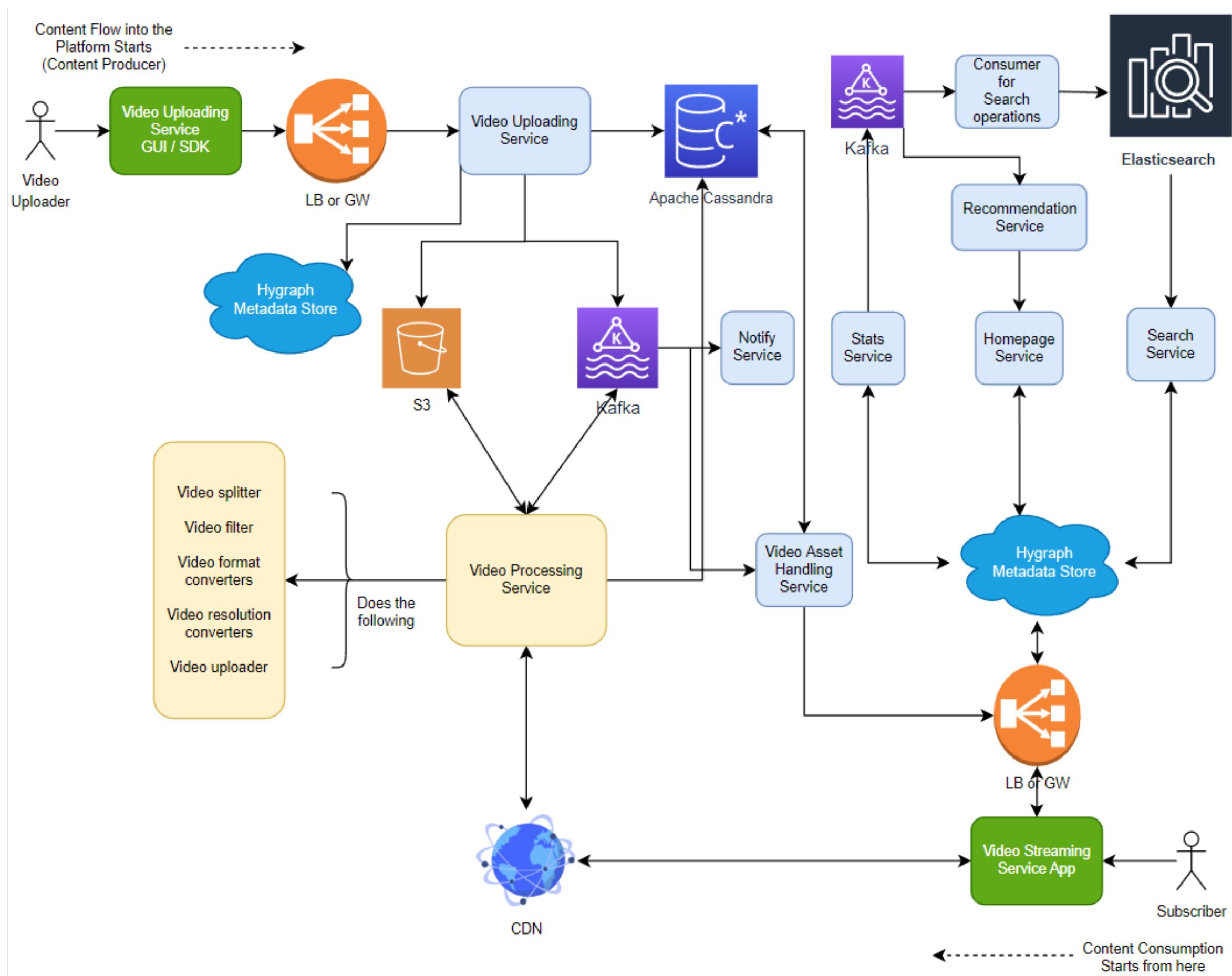
21. Design a Document Management System:

- Design a system for storing, retrieving, and managing documents.
- Key Components: Document storage, indexing, access control.
- Additional: Versioning for document revisions.



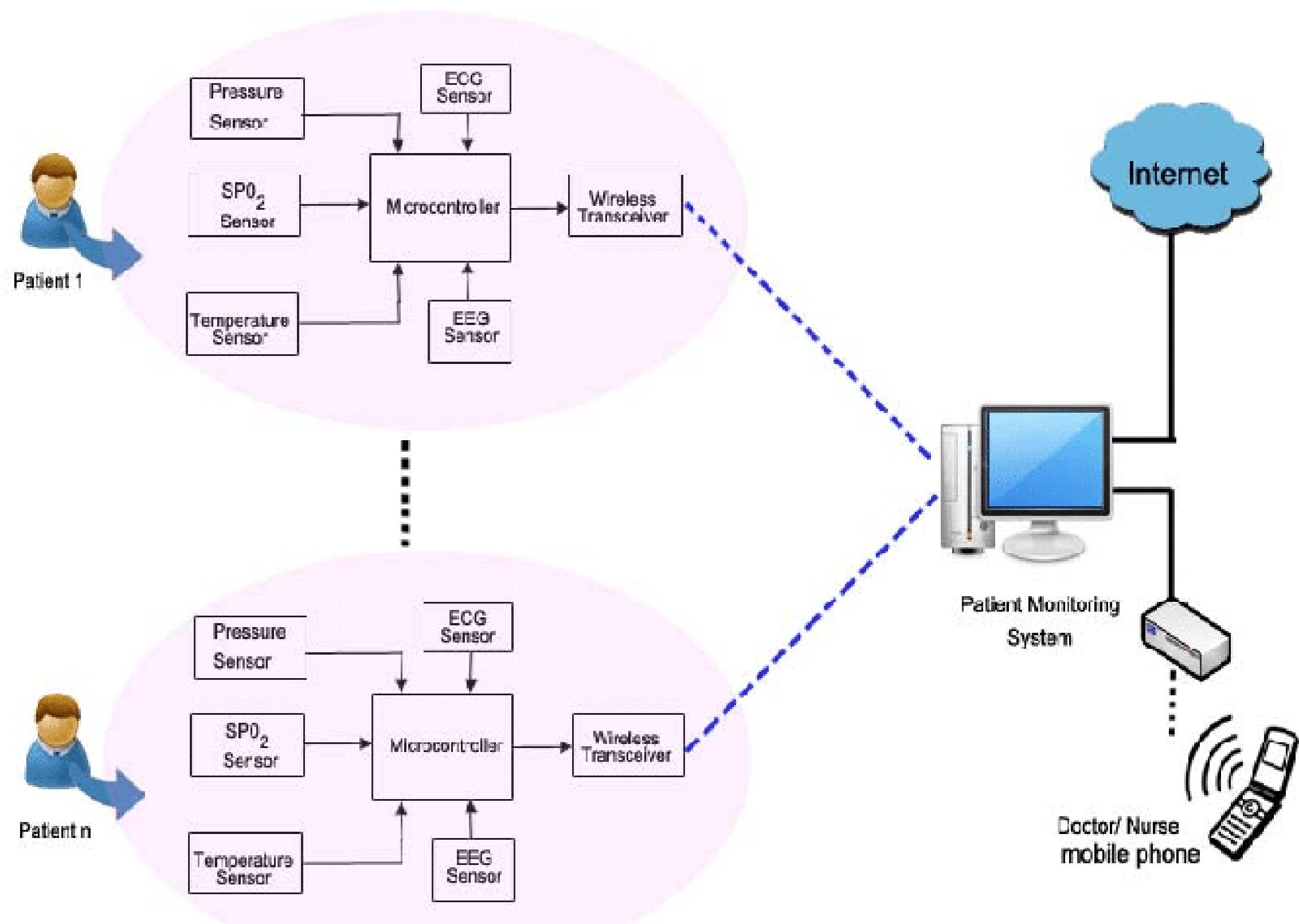
22. Design a Video Streaming Service:

- Design a scalable video streaming service for on-demand content.
- Key Components: Content delivery, transcoding, user profiles.
- Additional: Real-time streaming analytics.



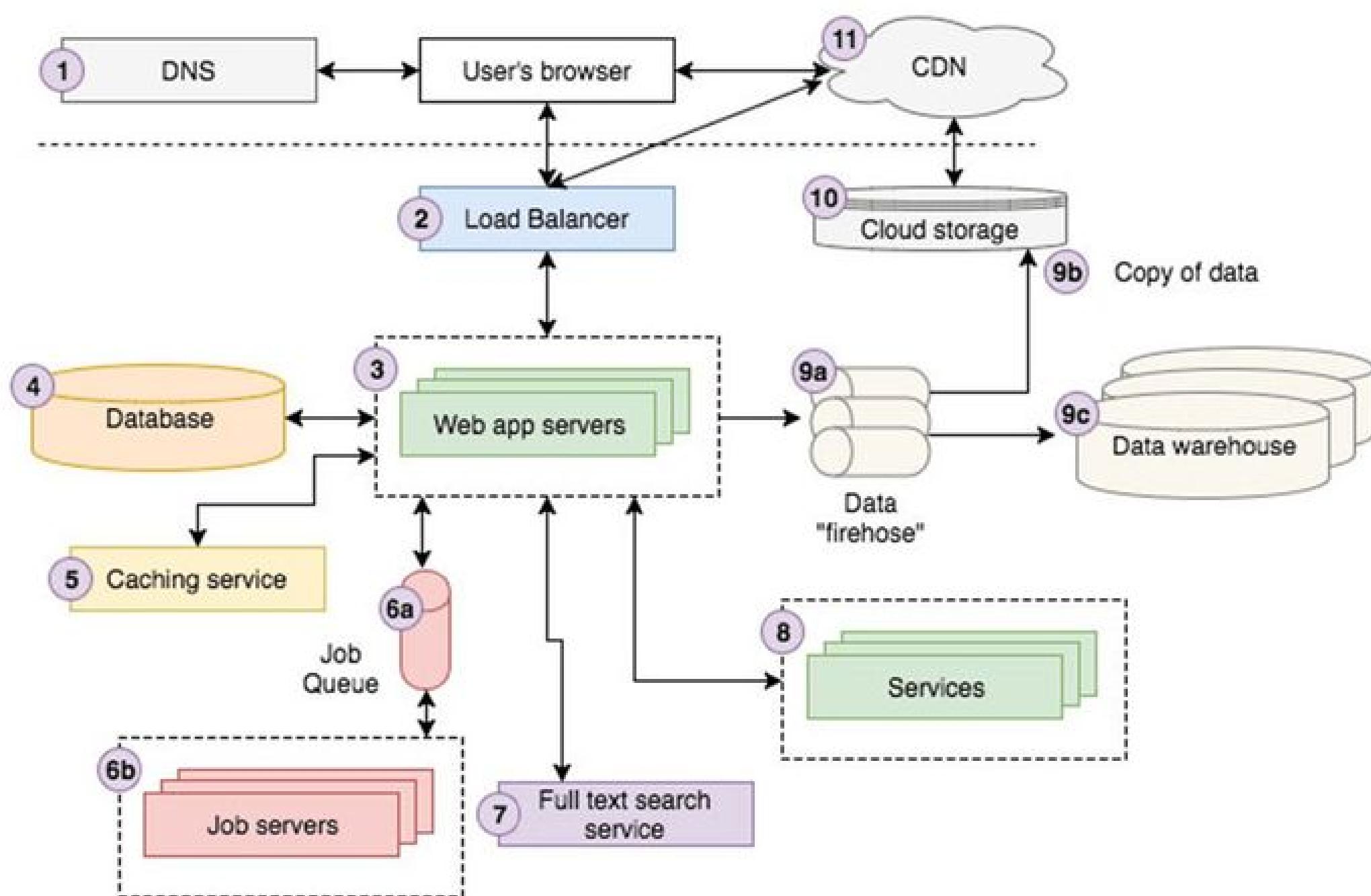
23. Design a Health Monitoring System:

- Design a scalable video streaming service for on-demand content.
- Key Components: Content delivery, transcoding, user profiles.
- Additional: Real-time streaming analytics.



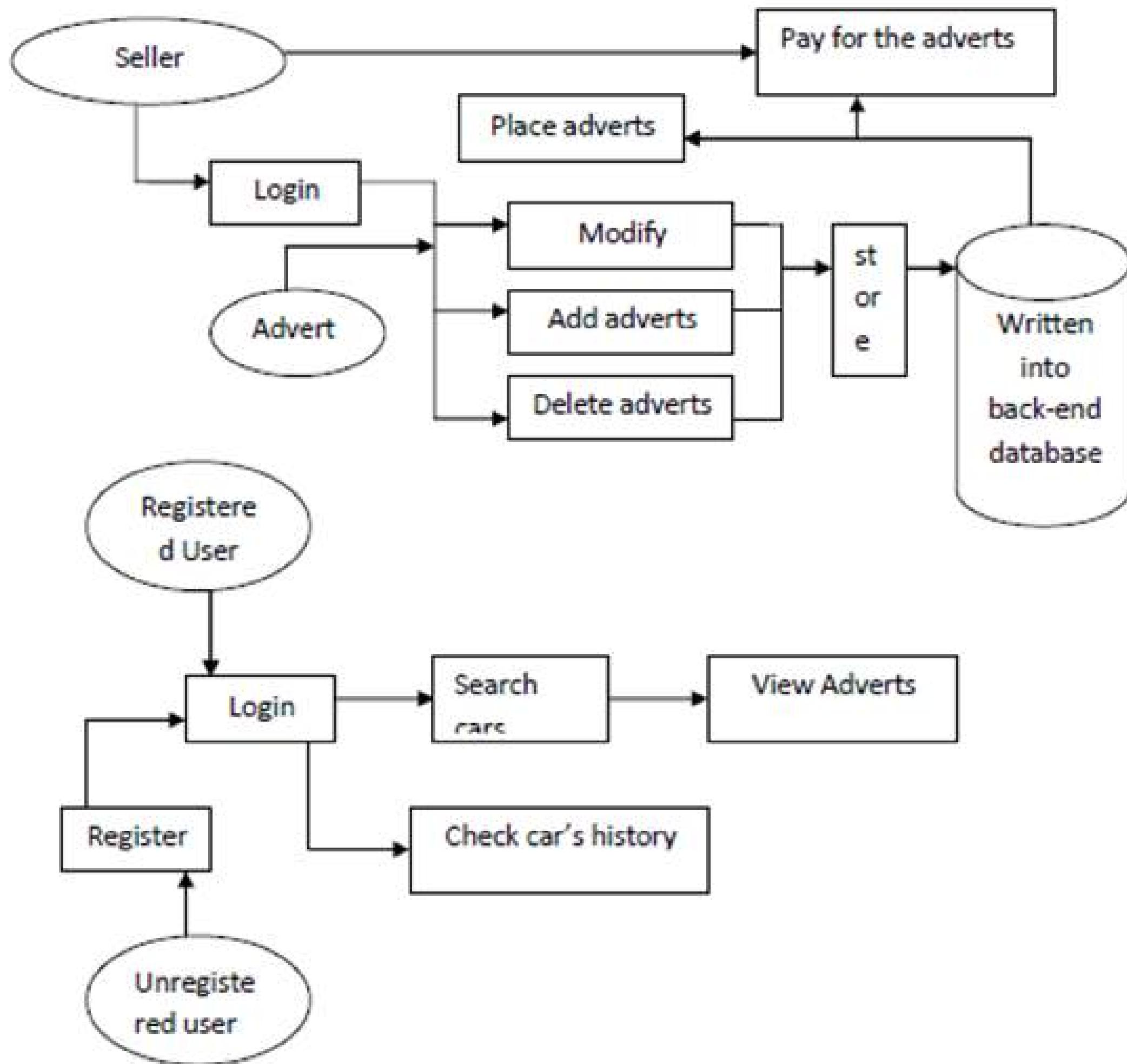
24. Design a Education Platform:

- Design an online education platform for courses and assessments.
- Key Components: Course content delivery, user profiles, grading system.
- Additional: Gamification elements for student engagement.



25. Design a Auction System:

- Design an online auction system for buying and selling goods.
- Key Components: Bidding engine, auction management, payment processing.
- Additional: Anti-sniping measures to prevent last-minute bidding..





AlgoTutor

WHY ALGOTUTOR



100% Placement Assistance



1-1 personal mentorship
from Industry experts



200+ Successful Alumni



147(Avg.)% Salary Hike



100% Success Rate



23 LPA (Avg.) CTC



Learn from scratch



Career Services

For Admission Enquiry



+91-7260058093



info@algotutor.io