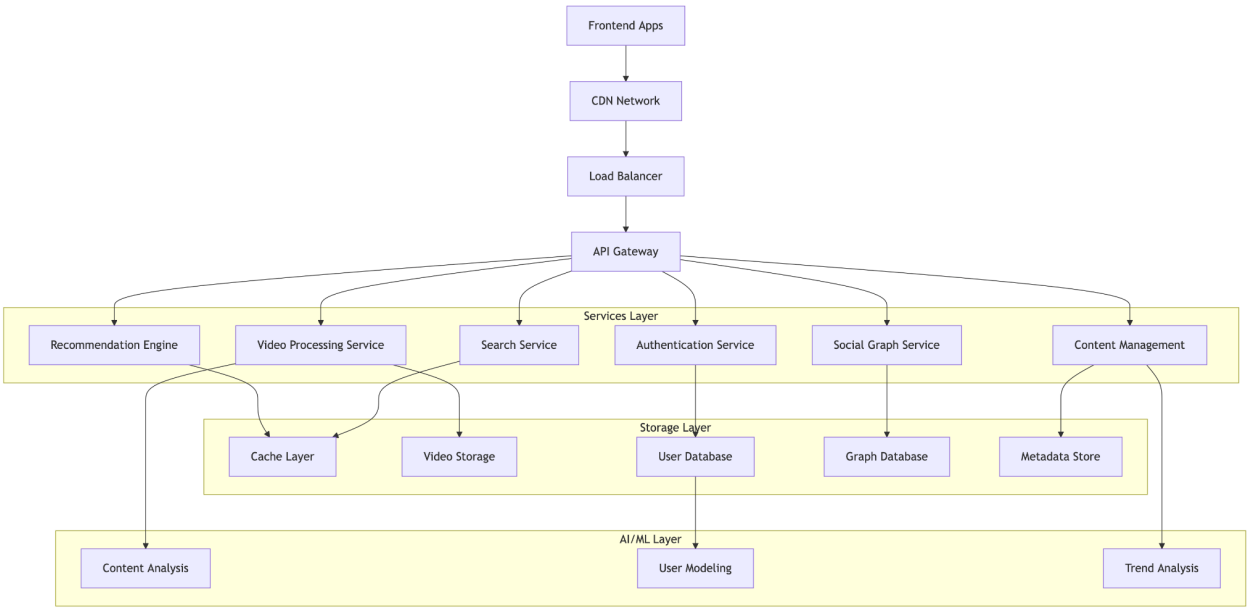
Lab 2, Microservices at TikTok

Yiwen (Echo) Wang (4V, A01347203)

1. Architecture Overview

TikTok employs a microservices architecture to divide its application into independent services, each responsible for a specific functionality. Below is a simplified representation of TikTok's architecture:



The request flow starts from Frontend Apps, going through a CDN network for content delivery, then a Load Balancer for traffic distribution, and an API Gateway that routes requests to various services.

The system has 3 main layers:

Services Layer

Handles all core business logic, including user authentication, video processing, content recommendation, search functionality, social relationships, and content management.

Storage Layer

Responsible for all data storage and access, including video files, user information, social graph data, metadata, and caching for faster data retrieval.

AI/ML Layer

Provides intelligent features through content analysis, user behaviour modelling, and trend analysis, supporting the recommendation system with data insights.

2. Example of Microservices

- **Recommendation Engine**: Provides personalized video suggestions based on user behavior and preferences.
- **Video Processing Service**: Handles video uploads, transcoding, thumbnail generation, and optimization for playback.
- **Search Service**: Enables users to search for videos, users, and hashtags efficiently.
- **Authentication Service**: Manages user login, registration, and secure access control.
- **Social Graph Service**: Manages relationships and interactions between users, such as follows, likes, and comments.
- **Content Management**: Oversees the organization, moderation, and publishing of user-generated content.

3. Pros and Cons

Pros

- **Scalability**: Each service can scale independently to meet specific demands, improving resource utilization.
- **Resilience**: Failures in one service do not affect the entire system, enhancing fault tolerance.
- **Technology Diversity**: Different technologies can be used for different services, optimizing performance and development.
- **Deployment Flexibility**: Services can be developed, tested, and deployed independently, enabling faster iteration.

Cons

- **System Complexity**: Managing multiple services increases the complexity of the system.
- **Communication Latency**: Network communication between services can introduce latency.
- **Data Consistency**: Maintaining consistency across distributed services is challenging.
- **Deployment and Maintenance Costs**: Requires robust deployment and monitoring tools, increasing operational expenses.

References

1. How TikTok Works: Decoding System Design & Architecture with Recommendation System
2. Exploring TikTok's Technology Stack – The Tech Behind Series
3. 抖音的互联网架构分析:高可用系统的设计与思考