

Idea :

1) database :**MongoDB**:

-**User Schema**: Store user credentials and profile information.

-**Video Schema**: Store video metadata, such as title, description, URL.

Why: Flexible schema design, horizontal scalability, and ease of integration with Java.

2) **user service** :

Implement registration and login using Spring Security with basic authentication.

3) **Media Service**:

-Implement endpoints for uploading and retrieving videos.

-Upload: Endpoint to upload videos and store metadata in MongoDB.

-Retrieve: Endpoint to fetch video metadata and serve video files.

3) **Kafka**: Set up Kafka topics for video uploads and user actions.

-Producer: Publish events like video upload to Kafka topics.

-Consumer: Consume events for processing tasks.

Why Kafka: High throughput, fault-tolerance, and real-time event streaming capabilities.

4) **RabbitMQ**: Use RabbitMQ for real-time notifications and task queuing.

-Publish: Send notifications to RabbitMQ queues.

-Consume: Process notifications and deliver them to users.

Why RabbitMQ: Robust messaging broker with excellent support for real-time messaging and task queues.

5) **Redis**:

-Use Redis to cache frequently accessed data, such as video metadata and user sessions.

Why Redis: In-memory data store with high performance and ease of use for caching.

6)frontend : react

Application :

