Designing a Scalable Short-Form Video Platform Inspired by TikTok

Juan Sebastian Colorado Caro, Walter Alejandro Suarez Fonseca
Department of Computer Engineering
Universidad Distrital Francisco José de Caldas
Bogotá, Colombia
Email: jscoloradoc@udistrital.edu.co, wasuarezf@udistrital.edu.co

***Abstract*—TikTok and other short video platforms have transformed the way people consume digital material, resulting in large and ongoing user involvement that produces gigabytes of interaction data every day. However, there are significant technological problems in maintaining such large amounts of real-time data while guaranteeing quick query execution, insightful analytics, and tailored content delivery. For a short video platform, this project suggests a distributed architecture that is optimized for real-time insights, high availability, and continuous data ingestion. Cloud-native BI tools, NoSQL databases, machine learning-based recommendation engines, and Apache Kafka are all integrated into the system. The design offers multi-region access, scalable performance, and actionable intelligence for platform administrators and marketers, according to the results.**

***Index Terms*—data, analytics, databases, big data, performance, SQL, NoSQL**

I. Introduction

In recent years, short-form video platforms—such as TikTok, Instagram Reels, and YouTube Shorts—have transformed the way users interact with digital content. These platforms allow users to upload and consume videos that last less than a minute, with rapid scrolling and algorithmic recommendation systems that drive engagement. As a result, they generate enormous volumes of data every second: likes, comments, video uploads, playback metrics, and ad impressions, all of which must be ingested, processed, and analyzed in real time.

The back-end systems of these platforms have serious technical issues despite their widespread use. Scalable ingestion pipelines must, first and foremost, be able to handle thousands of concurrent events per second without experiencing any delays. Second, even when data grows into terabytes or petabytes, query execution needs to be quick. Third, to guarantee a seamless user experience globally, data from various geographic locations must be stored and retrieved with little latency. Lastly, to improve user engagement and facilitate strategic choices made by platform administrators, marketers, and content producers, platforms need to provide recommendation systems and business information modules.

Existing solutions often rely on fragmented architectures, where analytics, storage, and real-time processing are poorly integrated or difficult to scale. Moreover, many open-source platforms lack built-in compliance with privacy regulations or fail to deliver consistent performance across regions.

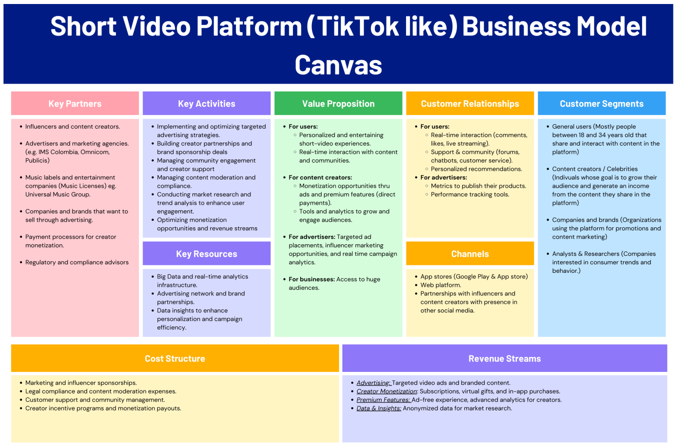
To address these issues, this project proposes the design of a short video platform backend built upon modern distributed systems and big data technologies. The architecture is designed to:

- Ingest and process high-frequency interaction data in real time.
- Offer fast and scalable queries for engagement analytics.
- Provide business intelligence dashboards to administrators and advertisers.
- Provide personalized recommendations to users.
- Ensure high availability and low-latency access across geographic regions.

II. Methods and Materials

This section presents the system design and architectural decisions made to support the functional and non-functional requirements of the platform. The solution is driven by the need to ensure low-latency data processing, continuous ingestion, multiregion availability, and actionable insights through analytics.

A. Business Model Summary



Short Video Platform (TikTok like) Business Model Canvas

Key Partners

- Influencers and content creators.
- Advertisers and marketing agencies. (e.g. IMS Colombia, Omnicom, Publicis)
- Music labels and entertainment companies (Music Licenses) eg. Universal Music Group.
- Companies and brands that want to sell through advertising.
- Payment processors for creator monetization.
- Regulatory and compliance advisors

Key Activities

- Implementing and optimizing targeted advertising strategies.
- Building creator partnerships and brand sponsorship deals
- Managing community engagement and creator support
- Managing content moderation and compliance.
- Conducting market research and trend analysis to enhance user engagement.
- Optimizing monetization opportunities and revenue streams

Key Resources

- Big Data and real-time analytics infrastructure.
- Advertising network and brand partnerships.
- Data insights to enhance personalization and campaign efficiency.

Value Proposition

- **For users:**
  - Personalized and entertaining short-video experiences.
  - Real-time interaction with content and communities.
- **For content creators:**
  - Monetization opportunities thru ads and premium features (direct payments).
  - Tools and analytics to grow and engage audiences.
- **For advertisers:** Targeted ad placements, influencer marketing opportunities, and real time campaign analytics.
- **For businesses:** Access to huge audiences.

Customer Relationships

- **For users:**
  - Real-time interaction (comments, likes, live streaming).
  - Support & community (forums, chatbots, customer service).
  - Personalized recommendations.
- **For advertisers:**
  - Metrics to publish their products.
  - Performance tracking tools.

Channels

- App stores (Google Play & App store)
- Web platform.
- Partnerships with influencers and content creators with presence in other social media.

Customer Segments

- General users (Mostly people between 18 and 34 years old that share and interact with content in the platform)
- Content creators / Celebrities (Indivuals whose goal is to grow their audience and generate an income from the content they share in the platform)
- Companies and brands (Organizations using the platform for promotions and content marketing)
- Analysts & Researchers (Companies interested in consumer trends and behavior.)

Cost Structure

- Marketing and influencer sponsorships.
- Legal compliance and content moderation expenses.
- Customer support and community management.
- Creator incentive programs and monetization payouts.

Revenue Streams

- *Advertising:* Targeted video ads and branded content.
- *Creator Monetization*: Subscriptions, virtual gifts, and in-app purchases.
- *Premium Features:* Ad-free experience, advanced analytics for creators.
- *Data & Insights:* Anonymized data for market research.

Fig. 1. Business Model Canva

The proposed platform is a video-centric social media application with monetization features. The core business model is based on a multisided platform connecting four types of user: viewers, content creators, advertisers, and administrators.

Funded by *Your Funding Source* under grant No. 12345.

Viewers consume and interact with video content; creators upload and monetize videos; advertisers launch targeted campaigns; and administrators ensure moderation and platform health. Revenue streams include advertising fees, sponsored content, and user microtransactions (e.g., virtual gifts). The value proposition of the platform is built on real-time engagement, intelligent content discovery, and data-driven business optimization.

B. *Functional and Non-functional requirements*

The proposed system is designed to meet the complex needs of a data-intensive short video platform. The following functional and non-functional requirements were defined to ensure the platform supports high user engagement, operational scalability, and intelligent content delivery.

Functional requirements The system supports the following core functionalities:

1) User registration and authentication
   - Users can register using secure credentials and authenticate through OAuth2 protocols. This enables access to platform features in a secure and scalable manner.
2) Video upload and management
   - Content creators can upload video files with metadata (title, tags, location) and manage drafts. Metadata is stored in NoSQL databases to allow flexible querying and aggregation.
3) Content interaction
   - Viewers can like, comment on, and share videos, as well as, follow creators and receive notifications.
4) Search and discovery
   - Users can search for content using indexed filters (categories, hashtags, duration).
5) Reporting and moderation
   - Users can report inappropriate content. Admins access a moderation dashboard with real-time updates on flagged media, supported by ML-based auto-moderation components.
6) Monetization and Payments
   - Advertisers create campaigns with audience targeting, and receive real-time analytics (impressions, CTR, etc.). Payments to creators are processed through secure integrations with payment providers.
7) Business intelligence dashboards
   - Admins and creators access dynamic dashboards showing real-time content performance, user retention, and revenue trends.

Non-Functional requirements To support the platform at scale, the system adheres to the following non-functional constraints:

1) Performance and Scalability
   - Support at least 10,000 daily active users with query latencies under 200 milliseconds during peak traffic.
   - Achieve horizontal scalability through microservices and database sharding.
2) High Availability and Fault Tolerance
   - Ensure 99.95% uptime via multi-region cloud deployments and automated failover.
3) Big Data Ingestion
   - Store raw data in a data lake and processed data in a data warehouse.
   - Constant ingestion pipeline via Kafka or Kinesis for streaming interaction and video metadata.
4) Security and Privacy
   - Enforce end-to-end encryption using TLS for data in transit and AES-256 for data at rest.
5) Maintainability and Extensibility
   - Use infrastructure-as-code for deployment; modular microservices architecture.
6) Multi-location Accessibility
   - Distribute storage and compute across multiple data centers for regional proximity.

C. *System Architecture*

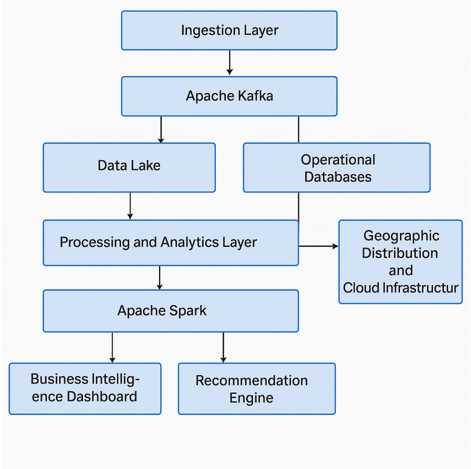

Ingestion Layer
Apache Kafka
Data Lake
Operational Databases
Processing and Analytics Layer
Geographic Distribution and Cloud Infrastructur
Apache Spark
Business Intelligence Dashboard
Recommendation Engine

Fig. 2. Initial architecture

To fulfill the high-level requirements of performance, scalability, availability, and analytical insight, the system architecture consists of four layers:

1) Ingestion Layer: Real-time ingestion is handled through Apache Kafka, a distributed messaging system optimized for high-throughput event streaming. Kafka enables decoupling between producers (frontend services) and consumers (processing engines), making it ideal for ingesting user interactions (likes, comments, uploads) at scale.
   a) Kafka provides fault-tolerant logs, scalable topic partitioning, and stream processing support through Kafka Streams and external connectors [1].
2) Storage layer: A hybrid approach is used to balance the need for structured transactional data and flexible schemaless analytics:

a) SQL databases are used for structured data such as users, authentication, profiles, and financial transactions. SQL databases ensure ACID compliance, strong consistency, and relational integrity, which is critical for user identity and monetary operations [2].

b) NoSQL databases are used for semi-structured and analytical data, such as interaction logs, video metadata, and aggregated engagement metrics. NoSQL databases provide horizontal scalability, high availability, and support for evolving schemas. This makes them suitable for storing large volumes of event data with unpredictable structure [3].

3) Processing and Analytics Layer: To extract real-time insights and transform raw data into useful analytics, the system employs Apache Spark Streaming as its processing engine.

a) Spark Streaming offers scalable in-memory computation, micro-batch processing, and rich APIs for SQL, ML, and graph processing. [4]

This layer generates: Trending metrics (views, likes, shares), Retention and engagement statistics, Aggregated insights for business intelligence dashboards.

4) Serving Layer

a) Business Intelligence (BI) Dashboards: Created using tools like Power BI, these tools simplify data visualization and exploration for non-technical stakeholders, reducing the need for manual queries. [5]

References

[1] “Apache Kafka vs. RabbitMQ: Comparing architectures, capabilities, and use cases”, Quix.io. [En línea]. Disponible en: https://quix.io/blog/apache-kafka-vs-rabbitmq-comparison. [Consultado: 15-may-2025].

[2] “PostgreSQL 17.5 documentation”, PostgreSQL Documentation, 08-may-2025. [En línea]. Disponible en: https://www.postgresql.org/docs/current/index.html. [Consultado: 15-may-2025].

[3] “¿Qué Es NoSQL? Descripción De Las Bases De Datos NoSQL”, MongoDB. [En línea]. Disponible en: https://www.mongodb.com/es/resources/basics/databases/nosql-explained. [Consultado: 15-may-2025].

[4] M. Zaharia *et al.*, “Apache Spark: A unified engine for big data processing”, *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[5] “Power BI documentation”, Microsoft.com. [En línea]. Disponible en: https://learn.microsoft.com/en-us/power-bi/. [Consultado: 15-may-2025].