Design Documentation

Overview

The Fleet Management System (FMS) is designed to provide real-time monitoring, analytics, and dashcam footage management for vehicle fleets. Its microservices-based architecture ensures scalability, flexibility, and high performance, addressing common challenges in fleet management, including real-time telemetry processing, video management, and incident detection.

This document details the system design, architecture, and a comparative performance analysis with the referenced papers.

System Design

1. Functional Modules

1.1. Real-Time Telemetry Processing

- **Description**: Collects vehicle data (speed, fuel, GPS) via Kafka producers and stores it in InfluxDB.
- **Purpose**: Enables real-time dashboards and analytics for vehicle performance, safety, and maintenance trends.

1.2. Incident Detection

- **Description**: Analyzes telemetry data to detect anomalies (e.g., sudden speed drops indicating accidents).
- **Purpose**: Generates notifications for fleet operators to ensure rapid incident response.

1.3. Dashcam Footage Management

- **Description**: Streams, stores, and retrieves dashcam footage in 10-second chunks using Kafka and block storage.
- **Purpose**: Provides visual evidence for incidents and supports route audits.

1.4. Analytics Dashboard

- **Description**: Displays metrics like average speed, mileage, fuel efficiency, and incident statistics.

- **Purpose**: Gives fleet operators actionable insights to optimize operations.

1.5. Role-Based Access Control

- **Description**: Uses MongoDB for storing user credentials and JWT for secure authentication.
- **Purpose**: Ensures different levels of access for admins, managers, and drivers.

---

2. System Architecture

The architecture integrates **React.js (Frontend)**, **Node.js (Backend)**, **Apache Kafka (Streaming)**, **InfluxDB (Telemetry Storage)**, **MongoDB (User Management)**, and **Block Storage (Video Storage)**. Each component operates independently, ensuring modularity and scalability.

Key Design Choices

1. **Microservices Architecture**: Simplifies maintenance and supports horizontal scaling.
2. **Time-Series Database (InfluxDB)**: Optimized for telemetry data with high-frequency writes and efficient queries.
3. **Event-Driven Messaging (Kafka)**: Handles real-time data streaming with minimal latency.
4. **Role-Based Security**: Ensures data privacy and secure access.

Comparative Analysis

This section compares the proposed Fleet Management System (FMS) with the methodologies described in the following papers:

Paper 1: "A Cloud-Based IoT Solution for Fleet Management"

Reference: https://ieeexplore.ieee.org/document/7800402

Approach in the Paper:

- Focuses on a cloud-based IoT architecture for fleet management.
- Relies heavily on cloud resources for data storage and processing.
- Limited to telemetry data analysis, excluding video integration.

Why FMS Outperforms:

- **Edge Integration**: FMS processes telemetry data in real-time using Kafka and InfluxDB, reducing dependency on cloud latency.
- **Video Management**: Integrates dashcam footage for enhanced incident analysis, which is absent in this paper's architecture.
- **Scalability**: Uses microservices, allowing individual components to scale independently.

Paper 2: "Real-Time Big Data Analytics for IoT-Enabled Fleet Management"

Reference: https://ieeexplore.ieee.org/document/10212331

Approach in the Paper:

- Implements real-time big data analytics using Hadoop and Spark.
- Focuses on predictive analytics but lacks real-time video and incident detection capabilities.

Why FMS Outperforms:

- **Real-Time Processing**: FMS uses Kafka for low-latency telemetry processing, whereas the Hadoop-Spark pipeline introduces higher delays.
- **Comprehensive Analytics**: FMS combines real-time analytics with historical trends and incident notifications.
- **Dashcam Integration**: Enhances analytics with video evidence, which is not supported in the paper's approach.

Paper 3: "IoT-Based Intelligent Transportation Systems"

Reference: https://ieeexplore.ieee.org/document/9483746

Approach in the Paper:

- Proposes an IoT-based architecture for intelligent transportation systems.
- Focuses on traffic optimization and route planning rather than fleet management.

Why FMS Outperforms:

- **Fleet-Centric Design**: FMS specifically addresses fleet operators' needs, including vehicle monitoring, incident detection, and role-based management.
- **Dashcam Footage Management**: Provides visual data for auditing and incident analysis, which is outside the scope of the paper.
- **Time-Series Optimization**: Uses InfluxDB for high-frequency telemetry data, whereas the paper uses generic IoT databases.

Performance Advantages

Feature	FMS	Paper 1	Paper 2	Paper 3
Real-Time Telemetry	Kafka + InfluxDB	Cloud-based storage	Hadoop-Spark delays	IoT-based latency
Dashcam Footage	Integrated with Kafka	Not Supported	Not Supported	Not Supported
Incident Detection	Telemetry + Video Analysis	Telemetry Only	Predictive Only	Traffic Optimization
Role-Based Management	MongoDB + JWT	Not Supported	Not Supported	Not Supported
Scalability	Microservices Architecture	Limited	Hadoop-Dependent	IoT Hardware-Limited

---

Conclusion

The proposed Fleet Management System outperforms the referenced papers by providing a **comprehensive, scalable, and real-time solution** for fleet operations. Key innovations include:

1. **Dashcam Footage Management**: Enhances traditional telemetry analysis with video evidence.
2. **Real-Time Telemetry Processing**: Reduces latency using Kafka and InfluxDB.
3. **Role-Based Management**: Ensures secure, user-specific data access.
4. **Scalability**: Supports growing fleet sizes with a modular microservices architecture.