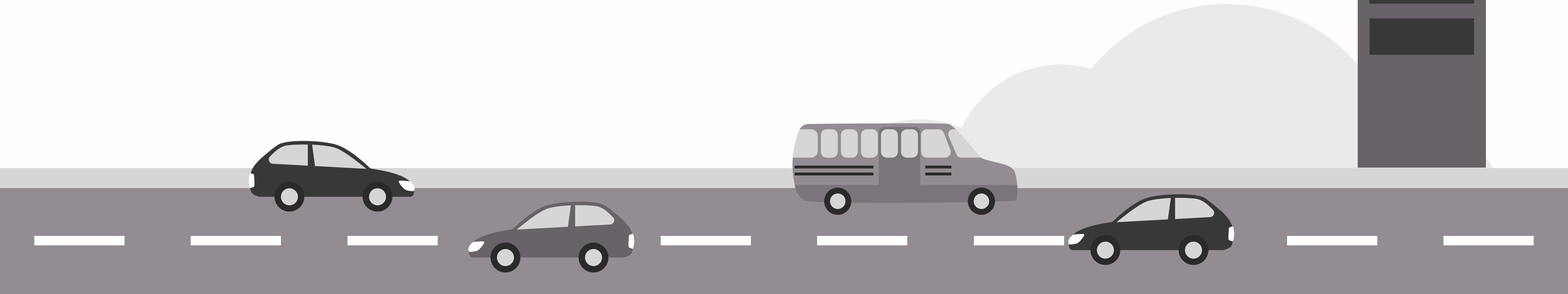
REAL-TIME DISTRIBUTED TRAFFIC MONITORING AND ALERT SYSTEM

Group 12: Aqsa, Rosalyn, Hassan

TABLE OF CONTENTS

01

INTRODUCTION



PROBLEM STATEMENT

Urban traffic congestion is a growing challenge, leading to longer commute times, increased pollution, and higher fuel consumption. Conventional centralized traffic monitoring systems are unable to effectively manage this complexity, as they suffer from:

High Latency

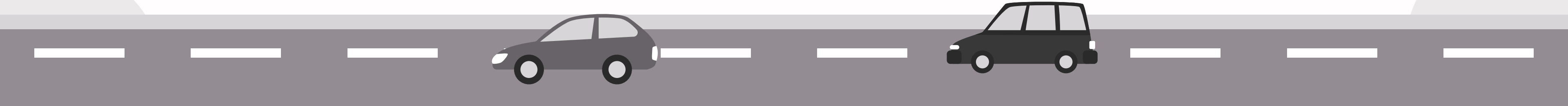
Delays in processing and reporting real-time traffic conditions.

Scalability Issues

Struggles to handle growing urban populations and expanding road networks.

Single Points of Failure

A centralized architecture makes these systems highly vulnerable to outages.



OUR SYSTEM

A real-time traffic monitoring system using distributed sensors and cameras utilizing decentralized data collection and processing. Our goal is to optimize scalability and fault tolerance while maintaining a cohesive data flow for a user interface that delivers alerts and congestion predictions.

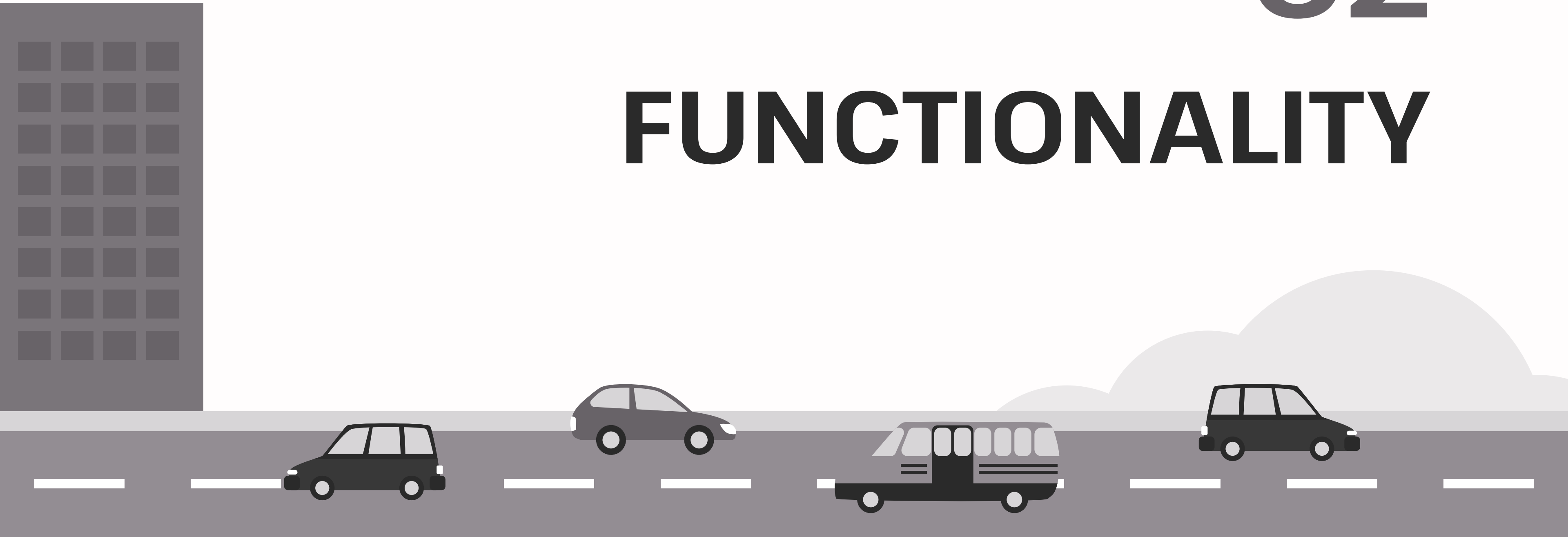
COMPARISON TO EXISTING SYSTEMS

(Google Maps and Waze)

Feature	Existing Systems	Our System
SYSTEM DESIGN	Centralized architecture, prone to single points of failure	Decentralized data collection minimizes single points of failure
DATA PROCESSING	Relies on centralized servers, leading to potential bottlenecks	Edge processing reduces central server load and latency
SCALABILITY	Limited scalability; performance degrades with increased data load	Scalable design easily accommodates additional nodes
RELIABILITY	High latency and less reliable during peak load or failures	Fault-tolerant design ensures continuous operation
LATENCY	Higher latency due to dependence on a centralized server	Lower latency through local processing at sensor nodes

02

FUNCTIONALITY



Modular Architecture

- Central server processes and aggregates data from distributed traffic nodes.
- Independent traffic nodes simulate real-world distributed data sources.

Real-Time Communication

- Efficient data flow between nodes, the server, and the dashboard and the web
- Low latency ensures real-time visualization of traffic conditions.

Scalability

- Nodes can be added or removed dynamically without affecting the system's overall operation.

Fault Tolerance

- Isolated components ensure that node or server failures do not collapse the system.

03

DEMONSTRATION

04

ANALYSIS

DISTRIBUTED SYSTEM FEATURES

Usability

The modular design of the system, with separate .jar files for the central server, nodes, and dashboard, ensures smooth operation, by displaying real-time traffic data (temperature, speed, vehicle count).

Performance

The system efficiently handles concurrent connections from multiple nodes, with minimal latency observed in transmitting and visualizing traffic data.

Security

The current implementation lacks encryption and node authentication, making it vulnerable to unauthorized access and requiring TLS for secure communication.

DISTRIBUTED SYSTEM FEATURES

✓

Scalability

The system supports multiple nodes successfully, but further optimizations like load balancing are necessary to handle larger-scale deployments.

✓

Reliability

The system can operate even if individual nodes fail, but it lacks advanced recovery mechanisms like data caching to enhance resilience during disruptions.

✓

Transparency

The distributed architecture's complexity is abstracted, providing users with a unified and intuitive dashboard interface.

FUTURE WORK

Real-Time WebSocket

- Updates for TrafficDashboardWeb:Replace file polling with WebSocket connections for instant data updates.

Kafka Integration

- In future versions, Kafka can be added to handle high traffic data efficiently.

Q&A