Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2025TMID36498
Project Name	TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning
Maximum Marks	4 Marks

TrafficTelligence - Technology Stack

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date: 31 January 2025

Technical Architecture

The system is designed using a microservices-based architecture, enabling scalable and modular deployment. TrafficTelligence collects and processes real-time and historical traffic data, applies ML models for prediction, and serves data via a responsive web/mobile frontend. The architecture integrates with external APIs for weather and event data to enhance prediction accuracy.

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

Guidelines:

Include all the processes (As an application logic / Technology Block)
Provide infrastructural demarcation (Local / Cloud)
Indicate external interfaces (third party API's etc.)
Indicate Data Storage components / services

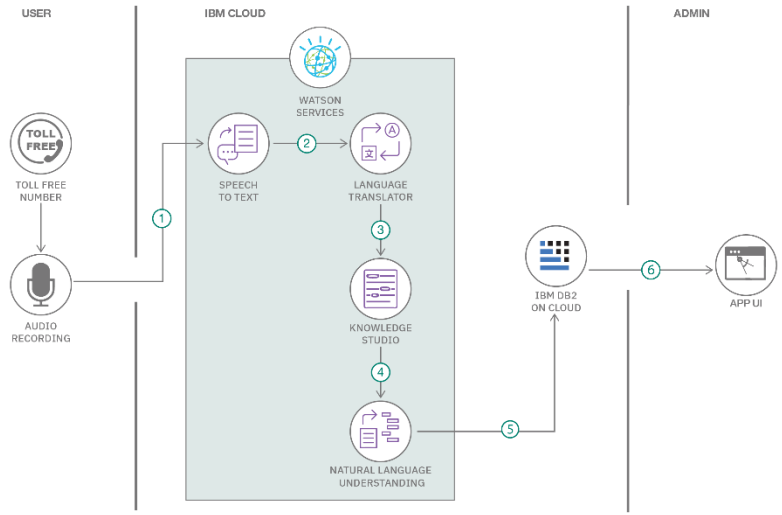

Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web and mobile interface for traffic insights, route planning	React JS, HTML, CSS
2	Application Logic-1	Real-time traffic data ingestion and processing	Python, Flask / FastAPI
3	Application Logic-2	Predictive modeling using machine learning	Scikit-learn, TensorFlow / PyTorch
4	Application Logic-3	Integration with city traffic control systems	REST APIs, WebSockets

5	Database	Store traffic records, event logs, weather data, predictions	PostgreSQL, MongoDB
6	Cloud Database	Cloud-based scalable traffic and historical data storage	AWS RDS, Google BigQuery
7	File Storage	Store models, logs, and sensor data files	AWS S3, Google Cloud Storage
8	External API-1	Weather data for traffic prediction	OpenWeatherMap API
9	External API-2	Event data (city event calendar integration)	Eventful API / Google Calendar API
10	Machine Learning Model	Traffic volume prediction and anomaly detection	Custom Regression + Time Series Models
11	Infrastructure	Deployment and orchestration	Docker, Kubernetes, AWS EC2 / GCP

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.

2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Use of open-source ML libraries and front-end frameworks	React, Flask, Scikit-learn, PyTorch
2	Security Implementations	Data encryption, secure APIs, role-based access control	HTTPS, JWT, OAuth 2.0, IAM

3	Scalable Architecture	Microservices with containerized deployments and scalable backend	Kubernetes, Docker
4	Availability	Cloud-based with load balancers and failover	AWS ELB, Multi-zone deployment
5	Performance	Caching, optimized queries, model inference acceleration, CDN for static assets	Redis, CDN, TensorRT / ONNX

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d