

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

Date	24 JUNE 2025
Team ID	LTVIP2025TMID33800
Project Name	TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning
Maximum Marks	4 Marks

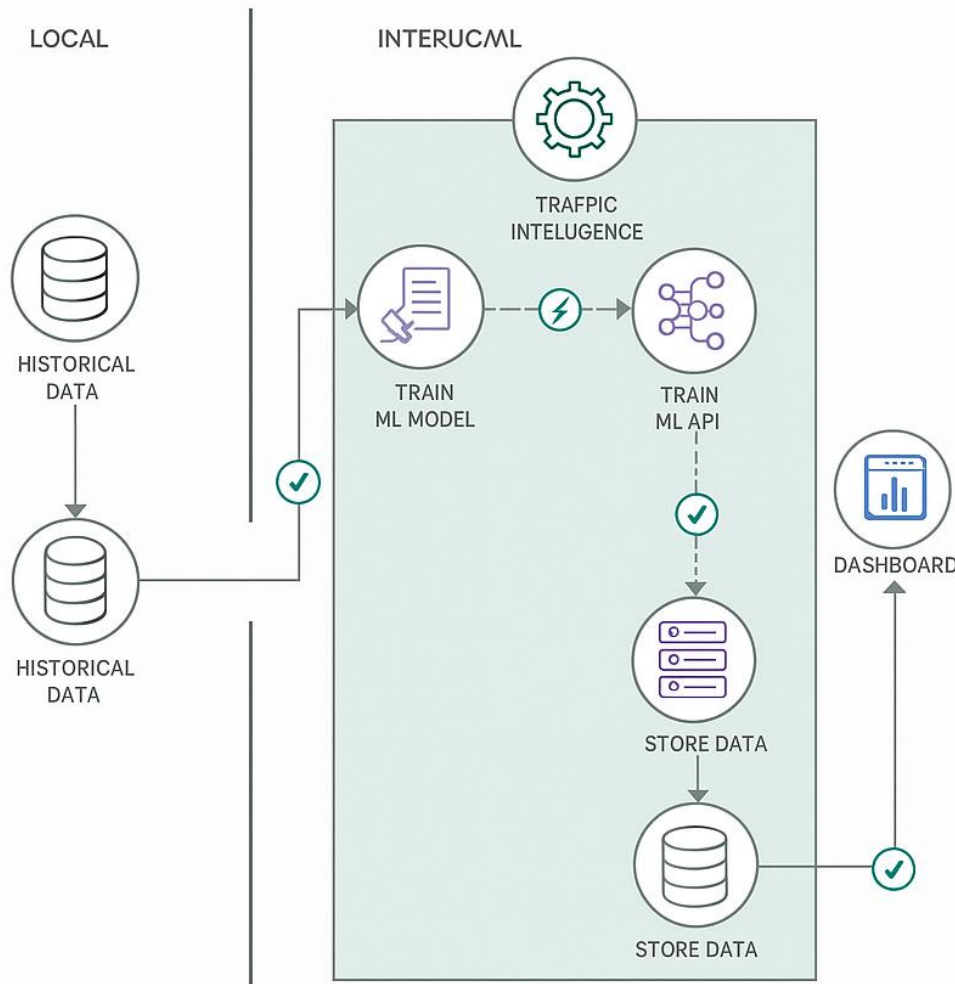
**Technical Architecture:**

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/>

# Example Technical Architecture



## 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  
If applicable

**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	Web-based dashboard	. HTML, CSS, JavaScript, React.js
2.	Application Logic-1	Data ingestion & cleaning logic	Python (Pandas, NumPy, Requests)
3.	Application Logic-2	ML model training & prediction	Python (scikit-learn, TensorFlow)
4.	Application Logic-3	API service for real-time access	Flask / FastAPI
5.	Database	Store structured traffic data	MongoDB / PostgreSQL
6.	Cloud Database	Scalable cloud database for analytics	AWS RDS / Google Cloud Firestore
7.	File Storage	Store raw files and model artifacts	AWS S3 / Google Cloud Storage
8.	External API-1	Get real-time weather data	OpenWeather API
9.	External API-2	Get city event data	Eventbrite / Ticketmaster API
10.	Machine Learning Model	Predict traffic volumes using time-series regression	LSTM / XGBoost models
11.	Infrastructure (Server / Cloud)	Cloud deployment using container orchestration	Docker, Kubernetes, AWS / GCP / Azure

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Use of open-source ML and web frameworks	scikit-learn, Flask, React, MongoDB
2.	Security Implementations	API authentication, data encryption, role-based access	HTTPS, JWT, OAuth2, IAM
3.	Scalable Architecture	Microservice-based with auto-scaling support	Kubernetes, Docker
4.	Availability	Load-balanced deployment, multi-zone cloud setup	AWS ELB, GCP Load Balance
5.	Performance	Caching and optimized model inference; CDN for frontend	Redis, Cloud CDN, Flask async

#### 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>