**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

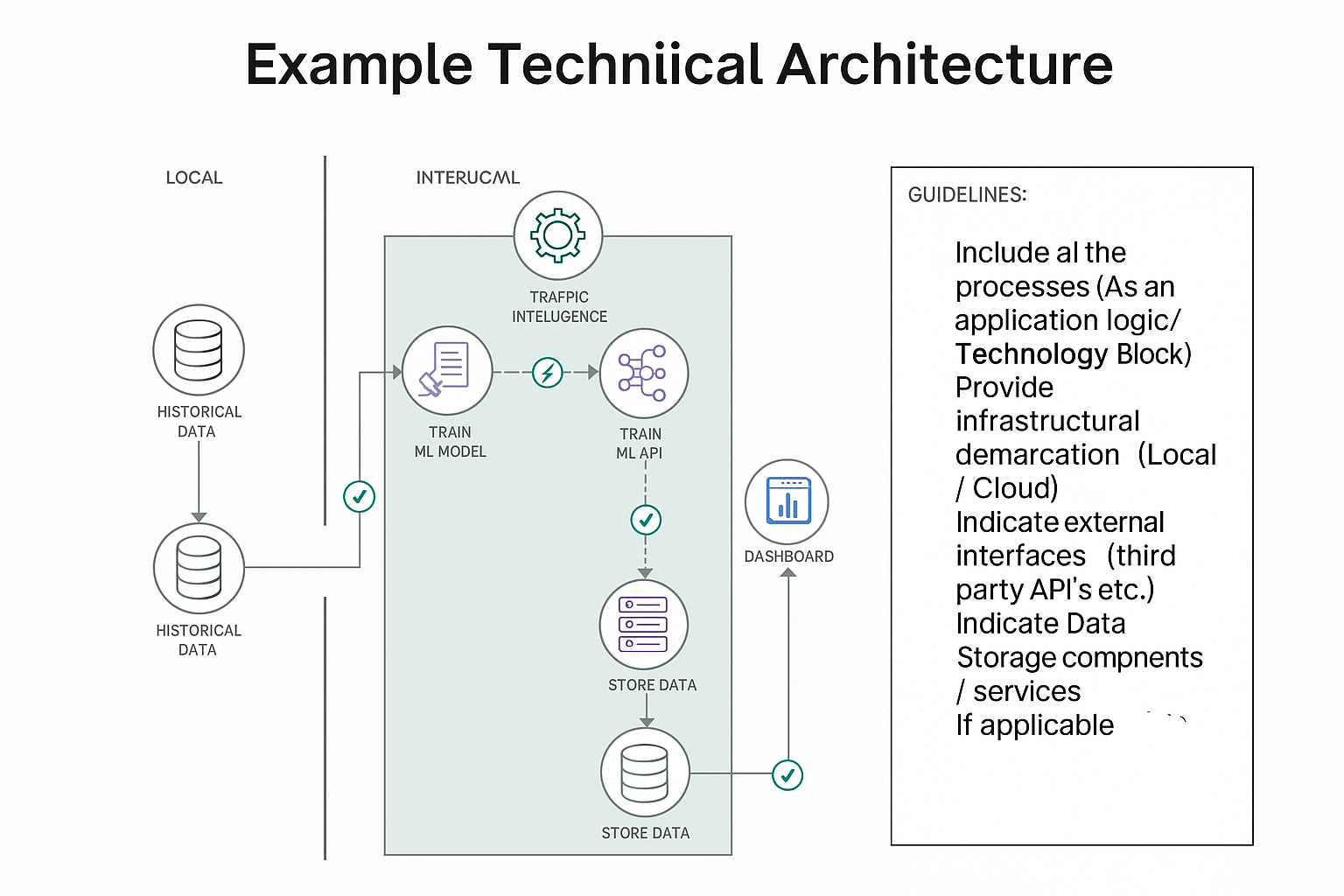
|  |  |
| --- | --- |
| Date | 24 JUNE 2025 |
| Team ID | LTVIP2025TMID33800 |
| Project Name | TrafficTelligence: Advanced Traffic Volume Estimation with Machine Learning |
| Maximum Marks | 4 Marks |

**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/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



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

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

**Table-2: Application Characteristics:**

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

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)