# **Implementation Plan**

**🔄 Phase 0: Vision, Strategy & Stakeholder Alignment**

**0.1 Define Vision & Goals**

* **Objective: Reduce commute time, optimize routes, enhance urban mobility.**
* **KPIs: Avg. commute time saved, ML model accuracy, latency, stakeholder satisfaction.**

**0.2 Stakeholder Discovery**

* **Personas: Commuters, traffic controllers, policy makers, logistics.**
* **Conduct JTBD analysis.**
* **Define feedback loops.**

**🔄 Phase 1: Project Planning & Architecture Design**

**1.1 Requirements Engineering**

* **Functional: Real-time/batch ingestion, forecasting, dashboards, APIs.**
* **Non-functional: Security, scalability, uptime, latency.**

**1.2 Architecture Design (L0 → L2)**

* **Conceptual (L0): APIs → Kafka → Beam/Spark → BigQuery → Vertex AI → Power BI/Streamlit**
* **Logical (L1): Data zones, event flows, ML endpoints.**
* **Physical (L2): GCP services, Kubernetes, Dockerized pipelines.**

**1.3 Toolchain Planning**

* **Data stack: Kafka, Airflow, Spark, dbt, BigQuery, Vertex AI, GitHub Actions.**
* **DevOps: Docker, Terraform, CI/CD workflows.**

**🔄 Phase 2: Data Ingestion & Handling**

**2.1 Programming & Data Handling Foundations**

* **Python: API calls, data parsing, exception handling.**
* **File types: CSV, JSON, XML**
* **Libraries: Pandas, NumPy, Regex for parsing**
* **SQL: Joins, subqueries, window functions, performance tuning (EXPLAIN, indexing).**

**2.2 Ingestion Layer**

* **Batch: Airflow DAGs for APIs**
* **Streaming: Kafka + Pub/Sub for live traffic/events**
* **Format validation: JSON Schema, Pydantic models**

**2.3 NoSQL & Relational Data Capture**

* **Document logs: MongoDB**
* **Wide-column stats: Cassandra**
* **Caching: Redis for quick route lookups**

**🔄 Phase 3: Data Storage & Warehousing**

**3.1 Lakehouse Design**

* **Layers: Raw → Cleaned → Curated → Feature Store**
* **Formats: Parquet, JSON, partitioned by time/route**

**3.2 Data Warehousing**

* **Platform: BigQuery (GCP), Snowflake (optional)**
* **Modeling: Star schema (traffic\_fact, weather\_dim, route\_dim, time\_dim)**
* **OLAP features: Partitioning, clustering, denormalization**
* **SCD handling: Type 2 via dbt**

**🔄 Phase 4: ETL / ELT Pipelines**

**4.1 Transformation Pipelines**

* **Batch: dbt with SQL models**
* **Stream: Apache Beam / Spark streaming**

**4.2 Workflow Orchestration**

* **Apache Airflow for DAG scheduling**
* **Metadata lineage tracking: OpenLineage integration**

**4.3 Data Quality & Governance**

* **Validation: Great Expectations for checks (nulls, types, ranges)**
* **Metadata Cataloging: Amundsen or DataHub**
* **Security: Encryption, IAM roles, PII masking**

**🔄 Phase 5: ML System Engineering**

**5.1 Feature Engineering**

* **Lag features, rolling stats, time windows**
* **Geospatial encoding (geohash), weather-event signals**

**5.2 Model Training & Management**

* **Algorithms: XGBoost, Prophet, LSTM**
* **Tools: Vertex AI, MLflow (tracking, experiments)**
* **Metrics: MAE, RMSE, latency**

**5.3 Model Deployment & Monitoring**

* **Serving: Vertex AI Endpoint / FastAPI**
* **Drift detection, canary releases**
* **Auto-retraining jobs with Airflow triggers**

**🔄 Phase 6: Visualization & BI**

**6.1 BI Dashboards (Macro)**

* **Tool: Power BI**
* **Metrics: Congestion trends, peak hours, policy simulations**
* **KPIs: Time saved, emission impact, hotspot maps**

**6.2 Micro Insight App (User)**

* **Tool: Streamlit**
* **Features: Route planner, ETA predictions, time savings**
* **Frontend tech: Leaflet.js for maps**

**🔄 Phase 7: DevOps, CI/CD, & Infra Automation**

**7.1 Version Control & Deployment**

* **GitHub, Git, branching strategies**
* **GitHub Actions: Auto testing (Pytest, dbt, GE), ML pipeline deployment**

**7.2 Infrastructure as Code**

* **Terraform to manage GCS, Pub/Sub, IAM, BigQuery**
* **Docker + Kubernetes (for model/API containers)**

**7.3 Environment Management**

* **Dev, staging, prod isolation**
* **Configuration files & secrets management**

**🔄 Phase 8: Monitoring, Observability & Cost**

**8.1 Monitoring Suite**

* **Prometheus + Grafana: System metrics**
* **Stackdriver: Logs and error traces**
* **OpenLineage: Pipeline dependency graphs**

**8.2 Alerts & Recovery**

* **Slack/email alerts**
* **Retry strategy, failover plans**

**8.3 Cost Optimization**

* **Monitor BQ/Vertex AI cost**
* **Auto-suspend dev jobs, lifecycle rules in GCS**

**🏐 Final PoC Deliverables**

* **Data ingestion pipeline (Kafka, Airflow)**
* **ETL workflows (dbt, Beam)**
* **Lakehouse and data warehouse schemas**
* **Forecasting ML pipeline**
* **Streamlit user dashboard + Power BI policy dashboard**
* **CI/CD pipelines, Terraform infrastructure**
* **Observability dashboard**
* **Documentation: Runbooks, pipeline specs, onboarding guides**

**Final Deliverables Summary**

|  |  |
| --- | --- |
| **Deliverable** | **Description** |
| Ingestion Pipeline | Kafka + Pub/Sub + Airflow batch + streaming pipelines |
| Data Models | dbt-based curated marts with SCD, fact/dim layers |
| ML System | Forecasting pipeline with training, serving, drift alerts |
| Dashboards | Power BI (macro), Streamlit (micro insights) |
| CI/CD System | GitHub Actions, Terraform, model deploy automation |
| Monitoring Suite | Prometheus, Grafana, Stackdriver, OpenLineage |
| Documentation | System diagrams, runbooks, onboarding, data catalog |

**Gantt Chart Overview (12–14 Weeks)**

|  |  |
| --- | --- |
| Week | Milestone |
| 1–2 | Design & architecture finalization, Terraform scaffold |
| 3–4 | API ingestion + raw zone ingestion validation |
| 5–6 | ETL + dbt staging + Great Expectations setup |
| 7–8 | DWH modeling + ML feature store + data marts |
| 9–10 | Model training, endpoint deployment, dashboards (Power BI + Streamlit) |
| 11 | CI/CD setup, testing framework integration, staging deployment |
| 12–13 | Monitoring, alerts, logging, drift detection setup |
| 14 | E2E test run + stakeholder walkthrough/demo |

**📘 Notion / Jira Epic Breakdown**

* **EPIC 1: Infra Setup**
  + Terraform + cloud provisioning
* **EPIC 2: Ingestion Pipelines**
  + Kafka, Pub/Sub, DAGs, retry handling
* **EPIC 3: Data Modeling & Warehouse**
  + dbt, SCDs, marts, snapshot design
* **EPIC 4: Machine Learning**
  + Feature engineering, model lifecycle, deployment
* **EPIC 5: Dashboards & UX**
  + Streamlit app + Power BI workspace
* **EPIC 6: CI/CD + Testing**
  + Lint, test, deploy workflows
* **EPIC 7: Monitoring + Observability**
  + Logs, metrics, alerting, lineage, drift reports
* **EPIC 8: Documentation + Handoff**
  + Internal wiki, runbooks, architecture decks, onboarding guides