High Level Design Document

This High Level Design (HLD) document outlines the architecture and core design of **GridInsightPro - Smart Energy Demand Forecaster**, a web application for energy providers to forecast demand using big data and AI. The platform enables secure data upload, real-time analytics, anomaly detection, and collaborative planning.

Project Name

GridInsightPro - Smart Energy Demand Forecaster

Purpose:
To provide energy providers with accurate, AI-driven demand forecasts, real-time analytics, anomaly detection, and collaborative planning tools for effective grid management.

---

1. System Architecture Overview

Architecture Summary:
GridInsightPro is a cloud-native, modular web application deployed on Google Cloud Platform (GCP). It consists of a React-based frontend, Python REST APIs, distributed data processing, AI/ML services, and secure storage.

Main System Modules

Module	Role
Frontend UI	User interface for data upload, visualization, and collaboration
Backend API	Handles business logic, user management, and orchestrates workflows
Data Processing	Distributed ingestion, validation, and transformation of datasets
AI/ML Engine	Generates demand forecasts and detects anomalies
Real-Time Analytics	Streams and updates live consumption metrics
Storage Layer	Stores raw/processed data, models, and audit logs
Auth/Security	Manages authentication, authorization, and audit logging

---

2. Component Interactions

Sequence Step	Interaction Description
1. User Login	Frontend authenticates via OAuth2 (GCP IAM)
2. Data Upload	User uploads data; API validates and stores in Cloud Storage
3. Data Processing	Backend triggers Spark jobs (Dataproc) for ingestion and transformation

4. Forecast/Anomaly Request	API invokes AI/ML engine for predictions and anomaly detection
5. Visualization	Frontend fetches processed data, forecasts, and anomalies via REST API
6. Collaboration	Users create/share scenarios and comments; API manages shared workspaces
7. Audit Logging	All actions logged for compliance and traceability

---

3. Data Flow Overview

Data Flow Step	Source	Destination	Description
Data Upload	User (UI)	Cloud Storage	Secure upload of CSV/Excel files
Data Processing	Cloud Storage	BigQuery	Spark jobs validate, clean, and store data
Forecast/Anomaly Generation	BigQuery	AI/ML Engine	Data fed to models for predictions/alerts
Results Delivery	AI/ML Engine	Frontend (UI)	Forecasts/anomalies visualized and exported
Collaboration Data	Frontend (UI)	Backend API	Scenario planning and comments stored
Audit Logs	All Components	Storage	User actions and data changes logged

---

4. Technology Stack

Layer	Technology/Tooling
Frontend	Next.js, React, D3.js/Chart.js
Backend API	Python (FastAPI/Flask), REST
Data Processing	Apache Spark (GCP Dataproc)
ML/AI	Python (scikit-learn, TensorFlow)
Storage	GCP BigQuery, Cloud Storage
Auth/Security	GCP IAM, OAuth2, TLS 1.2+
Deployment	GCP Cloud Run, GitHub Actions

---

5. Scalability, Reliability & Security

- **Scalability:**
  - Distributed Spark jobs and GCP managed services enable scaling to 100+ users and 1TB+ data.
  - Auto-scaling via Cloud Run and Dataproc.

- **Reliability:**
  - Target uptime ≥ 99.5% with managed GCP infrastructure.
  - Data upload/processing latency < 2 minutes for 10GB files.
- **Security:**
  - End-to-end encryption (TLS 1.2+), strict IAM roles, and audit logging.
  - Role-based access control (Admin, Analyst, Viewer).

End of High Level Design Document