

# High-Level Design (HLD)

## System Overview

The system is designed to predict the liquidity level (High, Medium, Low) of a cryptocurrency using a supervised machine learning model. It follows a structured ETL and ML pipeline with modular Python scripts.

## Architecture Components

### 1. Data Sources

- Two CSV datasets containing CoinGecko crypto market data (March 16 & 17)

### 2. Data Ingestion & Preparation

- Combined datasets into one
- Cleaned and normalized columns
- Exported cleaned data as `merged_cleaned_crypto_data.csv`

### 3. Feature Engineering

- Calculated liquidity ratio, price change ratio, and other indicators
- Categorized liquidity into 3 levels using `qcut`
- Output: `engineered_crypto_data.csv`

### 4. Model Training

- Model: Random Forest Classifier
- Scaled inputs using `StandardScaler`
- Saved model (`trained_model.pkl`) and scaler (`scaler.pkl`)

### 5. Streamlit Application

- Frontend app to input market values
- Loads trained model & scaler
- Predicts liquidity class

## Technologies Used

- **Python** (pandas, scikit-learn, joblib, streamlit)
- **Streamlit** for app interface
- **GitHub** for version control & deployment

## Key Design Principles

- **Modular Scripts:** Each pipeline step is a separate Python file
- **Reusability:** Data transformations and model are reusable for other dates
- **Transparency:** Intermediate CSVs are saved for inspection/debugging