

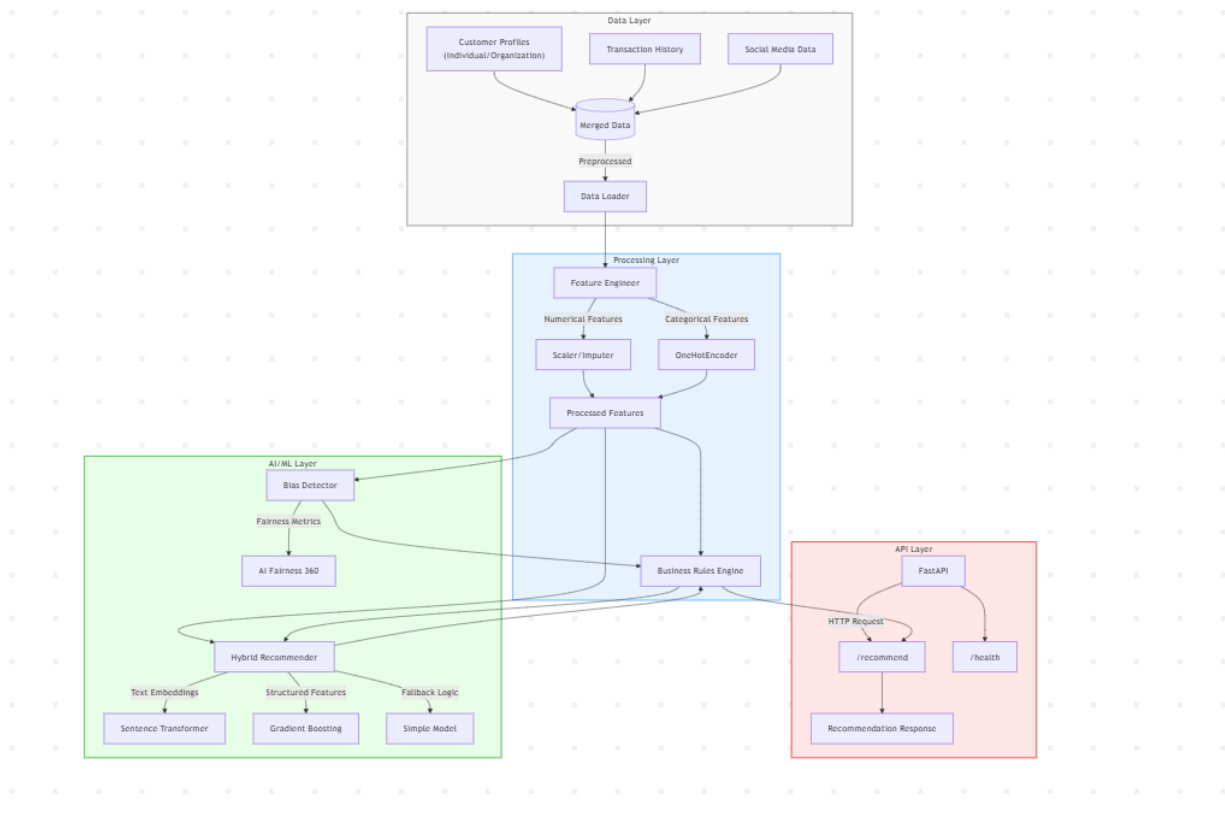


AI-Driven Hyper Personalization & Recommendations Architecture

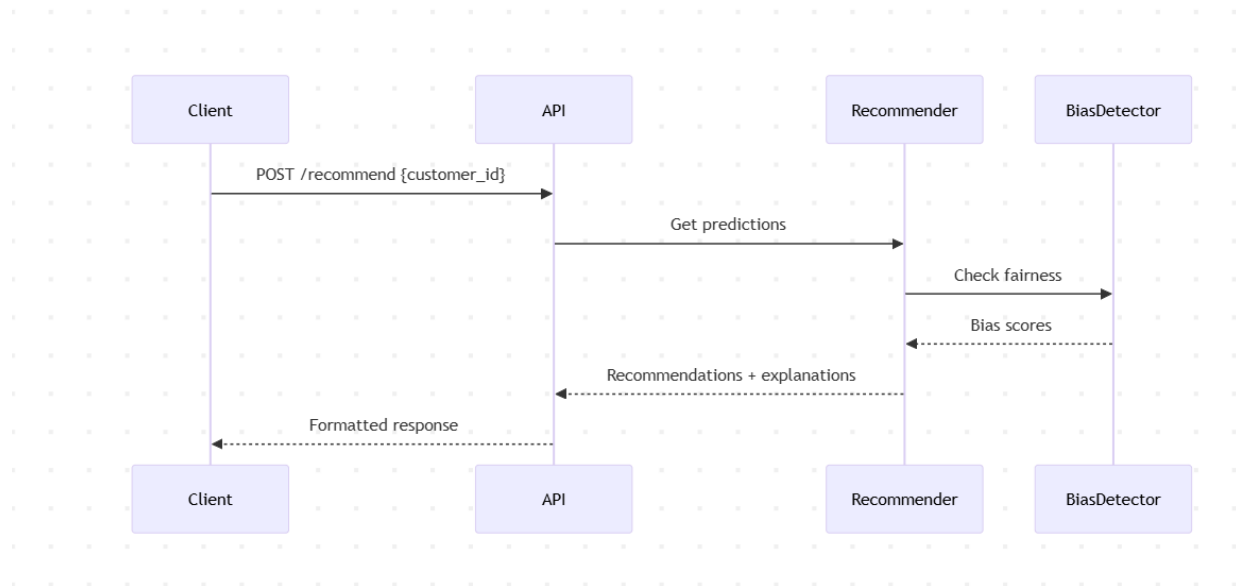
Overview

This solution provides a hyper-personalized, bias-aware recommendation system for financial products, combining transactional data, customer profiles, and social media signals to deliver tailored financial product suggestions.

Architecture:



Sequential Diagram:



Architectural Components

1. Data Layer

- **Data Loader (`data_loader.py`)**
 - Ingests multiple data sources (customer profiles, transactions, social media)
 - Performs comprehensive data cleaning and transformation
 - Handles missing values and type conversions
 - Merges datasets into a unified customer view
 - Includes sentiment analysis on social media content

2. Processing Layer

- **Feature Engineering (`feature_engineer.py`)**
 - Defines categorical and numerical feature pipelines
 - Implements standard preprocessing (imputation, scaling, encoding)
 - Creates consistent feature transformation for model training and inference
- **Business Rules Engine (`business_rules.py`)**
 - Applies domain-specific boosting rules
 - Adjusts recommendations based on customer attributes
 - Ensures business logic is incorporated into final recommendations

3. AI/ML Layer

- **Hybrid Recommender (`recommender.py`)**
 - Combines multiple ML techniques:

- Gradient Boosting for baseline predictions
 - Sentence Transformers for text embeddings
 - BERT for advanced text understanding
- Includes sophisticated fallback mechanisms
- Calculates financial risk profiles
- Handles class imbalance with SMOTE and balanced classifiers
- **Bias Detection (`bias_detector.py`)**
 - Uses AIF360 toolkit for fairness metrics
 - Monitors protected attributes (gender, age, location)
 - Calculates disparate impact and statistical parity
 - Provides individual-level bias checks

4. API Layer (`app.py`)

- **FastAPI Application**
 - RESTful endpoints for recommendations
 - Health check and monitoring
 - Request/response validation with Pydantic
 - Comprehensive logging

Key Design Patterns

- 1. Modular Pipeline Architecture**
 - a. Clear separation between data loading, feature engineering, and modeling
 - b. Each component can be developed and tested independently
- 2. Hybrid Recommendation Approach**
 - a. Combines traditional ML with transformer-based embeddings
 - b. Business rules provide final adjustment layer
- 3. Progressive Enhancement**
 - a. Sophisticated models when data is sufficient
 - b. Graceful degradation to simpler models when needed
 - c. Intelligent fallback recommendations
- 4. Bias-Aware Design**
 - a. Built-in fairness monitoring at both dataset and individual levels
 - b. Protected attribute tracking throughout pipeline
- 5. Explainability**
 - a. Explanations generated for recommendations
 - b. Transparent risk scoring

Data Flow

1. Initialization

- a. Load and merge all data sources
- b. Train recommendation models
- c. Set up bias detection baselines

2. Recommendation Process

```
Customer Request → Feature Transformation →  
Model Prediction → Business Rules Adjustment →  
Bias Check → Explanation Generation →  
Final Recommendation
```

3. Monitoring

- a. Continuous bias auditing
- b. Model performance tracking
- c. Data quality checks

Technical Stack

- **Core Framework:** FastAPI (Python)
- **ML Frameworks:** Scikit-learn, PyTorch, Transformers
- **NLP:** Sentence Transformers, NLTK
- **Bias Detection:** AIF360
- **Data Processing:** Pandas, NumPy
- **Deployment:** Uvicorn ASGI server

Quality Attributes

1. **Fairness:** Built-in bias detection at multiple levels
2. **Resilience:** Multiple fallback mechanisms
3. **Explainability:** Clear explanations for recommendations
4. **Performance:** Efficient feature engineering and model serving
5. **Maintainability:** Modular design with clear interfaces

This architecture provides a robust foundation for delivering personalized financial recommendations while maintaining ethical AI practices and business alignment.