TÀI LIỆU ĐỊNH HƯỚNG KỸ THUẬT DỰ ÁN ITAPIA (v2.0)

1. Mục đích

Tài liệu này đóng vai trò là bản thiết kế kỹ thuật (technical blueprint) cho việc triển khai toàn bộ hệ thống ITAPIA. Nó định nghĩa rõ ràng vai trò, trách nhiệm, luồng dữ liệu và sự tương tác giữa các microservices, làm cơ sở cho việc phát triển và tích hợp các thành phần một cách nhất quán và hiệu quả. Phiên bản 2.0 phản ánh một sự tiến hóa trong kiến trúc, tập trung vào khả năng mở rộng, tính giải thích sâu sắc và các quy trình AI tiên tiến.

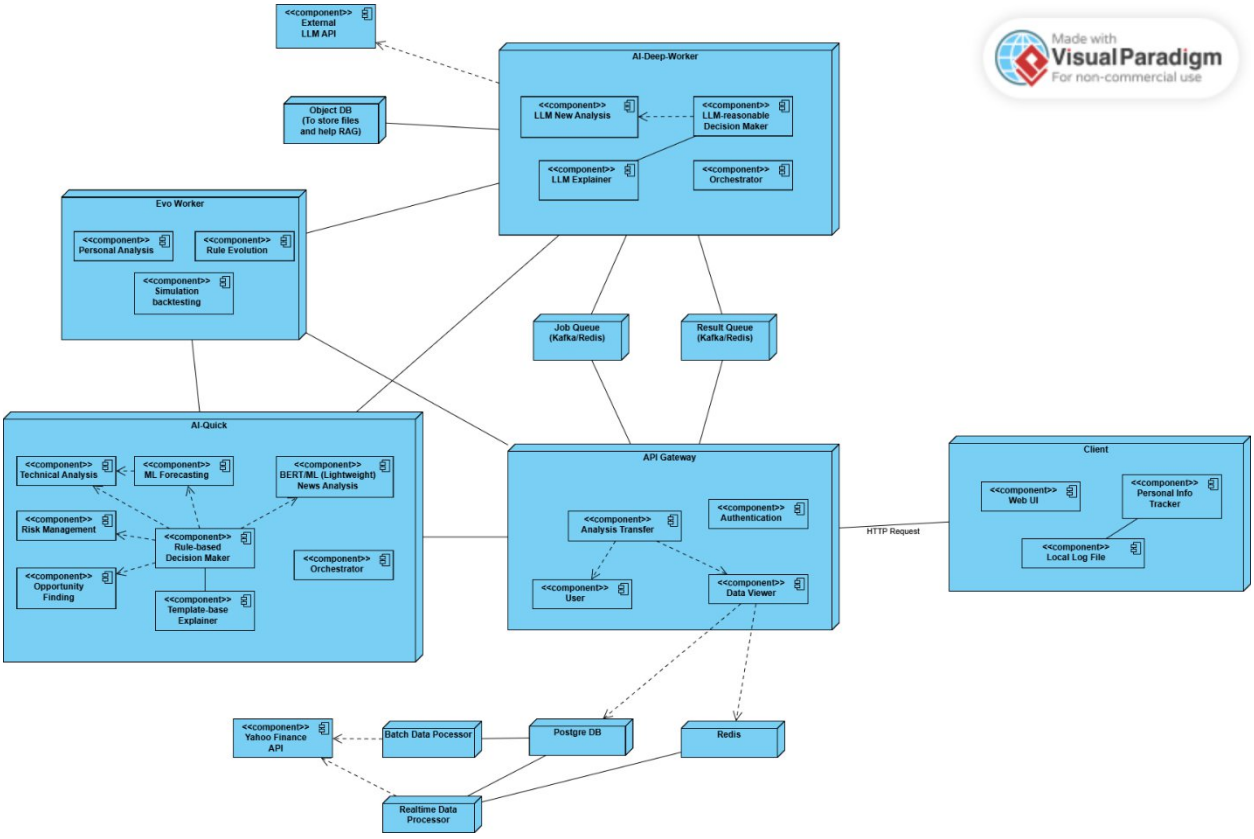
2. Nguyên tắc Kiến trúc Cốt lõi

Kiến trúc của ITAPIA v2.0 được xây dựng dựa trên các nguyên tắc nền tảng sau:

- **Kiến trúc Microservices & Phân cấp:** Hệ thống được chia thành các service độc lập, chuyên biệt, giao tiếp với nhau qua API. Các service được phân thành các tầng rõ ràng (API Gateway, Business Services, Core AI Services) để quản lý sự phức tạp và tăng khả năng mở rộng.
- **Kiến trúc Hai cấp độ (Tiered Architecture):** Để cân bằng giữa tốc độ, chi phí và chiều sâu phân tích, hệ thống cung cấp hai quy trình xử lý riêng biệt:
    - **Quick Check (CPU-based, Synchronous):** Cung cấp phân tích nhanh, dựa trên các mô hình nhẹ và quy tắc, trả về kết quả gần như tức thời.
    - **Deep Dive (GPU-based, Asynchronous):** Cung cấp phân tích sâu, toàn diện, sử dụng các mô hình LLM và các thuật toán phức tạp, xử lý bất đồng bộ thông qua hàng đợi.
- **Ensemble Learning cấp Kiến trúc:** Hệ thống không dựa vào một "siêu AI" duy nhất. Thay vào đó, quyết định cuối cùng là sự tổng hợp thông minh từ nhiều module chuyên gia (Technical, News, Forecasting), coi Decision Maker là một "meta-model".
- **Minh bạch và Truy vết được (Explainability & Traceability):** Mọi kết quả phân tích và quyết định đều phải đi kèm với **"bằng chứng" (evidence)** có cấu trúc. Điều này cho phép hệ thống tự giải thích lý do đằng sau các khuyến nghị và giúp việc gỡ lỗi trở nên dễ dàng, xây dựng lòng tin cho người dùng.
- **"UTC Everywhere" & Giao tiếp Chuẩn hóa:** Toàn bộ hệ thống sử dụng múi giờ UTC để lưu trữ và xử lý thời gian. Giao tiếp giữa các service sử dụng các định dạng chuẩn (JSON qua REST API) và được định nghĩa rõ ràng bằng các schema (Pydantic).

3. Chi tiết các Thành phần và Luồng Dữ liệu

Sơ đồ triển khai tổng thể:



3.1. Data Processing Module

- **Vai trò:** Là nền tảng cung cấp "nhiên liệu" cho toàn bộ hệ thống, chịu trách nhiệm cho vòng đời dữ liệu từ thu thập, làm sạch đến lưu trữ.
- **Quy trình Batch:** Các script chạy định kỳ để thu thập và cập nhật dữ liệu lịch sử (giá hàng ngày, tin tức) vào **PostgreSQL**. Logic được điều khiển bởi dữ liệu, tự động xác định các cổ phiếu đang hoạt động và khoảng thời gian cần lấy.
- **Quy trình Real-time:** Một service chạy liên tục, tự động lọc các cổ phiếu có thị trường đang mở cửa (dựa trên metadata trong DB), thu thập giá mới nhất và ghi vào **Redis Streams**.
- **Quản lý DB:** Sử dụng các lớp Manager (DAO) để trừu tượng hóa việc tương tác với CSDL, triển khai cơ chế cache metadata để tăng hiệu năng.

3.2. API Gateway (api-gateway)

- **Vai trò:** Là "bộ mặt" và cổng vào duy nhất của hệ thống, chạy trên hạ tầng CPU.
- **Trách nhiệm:**
    - Xác thực và phân quyền người dùng (Authentication & Authorization).

    - Tiếp nhận request từ Client.
    - **Điều phối (Orchestrate):** Gọi đến các dịch vụ nội bộ (Data Service để lấy dữ liệu từ DB/Redis, AI Service để yêu cầu phân tích) và tổng hợp kết quả.
    - Chuyển tiếp các yêu cầu "Deep Dive" bằng cách đẩy "job" vào hàng đợi.
- **Tương tác:** Giao tiếp với Client qua REST API công khai, và giao tiếp với các service backend qua API nội bộ.

3.3. AI Service Quick (ai-service-quick)

- **Vai trò:** "Nhà máy AI" hiệu năng cao, chạy trên hạ tầng CPU, cung cấp các kết quả phân tích và dự báo nhanh.
- **Kiến trúc nội bộ:**
    - **Orchestrators:** Sử dụng kiến trúc điều phối phân cấp ("CEO" và "Trưởng phòng") để quản lý luồng công việc một cách sạch sẽ.
    - **Technical Analysis Module:** Bao gồm FeatureEngine và AnalysisEngine (cho cả daily và intraday) để cung cấp các phân tích về xu hướng, S/R, và mẫu hình kèm theo "bằng chứng".
    - **ML Forecasting Module:** Sử dụng các mô hình cây (LightGBM, XGBoost) đã được huấn luyện sẵn để đưa ra dự báo về phân phối giá và xác suất giao dịch thành công (Triple-Barrier), đi kèm giải thích từ SHAP.
    - **Lightweight News Analysis:** Sử dụng các mô hình nhỏ (BERT-tiny, VADER) để phân tích sentiment tin tức.
    - **Advisor Module (Quick):**
        - Rule-based Decision Maker: Áp dụng các quy tắc (có thể từ Evo Agent) để đưa ra quyết định sơ bộ.
        - Template-based Explainer: Sử dụng các "bằng chứng" từ các module khác để tạo ra các câu giải thích theo mẫu.
- **Giao tiếp:** Cung cấp một API nội bộ để API Gateway gọi đến.

3.4. AI Service Deep Worker (ai-service-deep-worker)

- **Vai trò:** Worker bất đồng bộ, chạy trên hạ tầng GPU, chịu trách nhiệm cho các tác vụ AI/LLM tính toán nặng và đòi hỏi khả năng suy luận sâu.
- **Luồng hoạt động:** Lắng nghe các "job" từ **Job Queue** (Kafka/Redis), xử lý, và đẩy kết quả vào **Result Queue**. Mỗi job sẽ chứa kết quả từ một quy trình "Quick Check" làm ngữ cảnh đầu vào.

- **Kiến trúc nội bộ:**
    - **LLM News Analysis:**
        - **Nhiệm vụ:** Vượt xa việc chỉ phân tích sentiment. Sử dụng LLM để thực hiện các tác vụ phức tạp trên nội dung tin tức.
        - **Chức năng:**
            - **Tóm tắt Thông minh (Intelligent Summarization):** Tổng hợp nhiều tin tức liên quan thành một bản tóm tắt mạch lạc.
            - **Phân tích Tác động (Impact Assessment):** Đánh giá mức độ ảnh hưởng tiềm tàng của một tin tức đến giá cổ phiếu (ví dụ: Thấp, Trung bình, Cao).
            - **Trích xuất Thông tin (Information Extraction):** Rút ra các thực thể, con số tài chính, và mối quan hệ quan trọng từ văn bản.
    - **LLM-based Decision Maker & Explainer:**
        - **Nhiệm vụ:** Tổng hợp thông minh và tạo lập luận.
        - **Logic:** Sử dụng các mô hình LLM đã được fine-tune (thông qua **Knowledge Distillation** và **RLAIF**) để nhận các tín hiệu cấp cao từ Quick Check **và** kết quả phân tích sâu từ LLM News Analysis. Từ đó, nó xây dựng một lập luận toàn diện, cân nhắc các yếu tố xung đột và đưa ra quyết định cuối cùng.
    - **RAG Integration:** Tích hợp với một **Knowledge Base** (lưu trên Object DB/Vector DB) để truy xuất thông tin tài chính đã được xác thực, giảm thiểu "ảo giác" và tăng độ tin cậy của cả Decision Maker và Explain Agent.
    - **(Tương lai) MCP Server (Model Context Protocol):** Có thể tích hợp khả năng gọi đến một service ngoài để lấy dữ liệu real-time (tin tức, chỉ số vĩ mô) nhằm làm giàu thêm ngữ cảnh cho LLM.

3.5. Evolution Module (evo-worker)

- **Vai trò:** Worker chạy offline, chịu trách nhiệm "tiến hóa" và tối ưu hóa các chiến lược giao dịch.
- **Kiến trúc:**
    - **Co-evolution:** Sử dụng phương pháp Đồng tiến hóa, với một quần thể dùng **Genetic Programming (GP)** để tối ưu hóa cấu trúc quy tắc và một quần thể khác dùng các thuật toán như **CMA-ES/DE** để tối ưu hóa các ngưỡng hằng số trong các quy tắc đó.

- **Simulation Module:** Evo Agent gọi đến Simulation Module (backtester) hàng ngàn lần để đánh giá Fitness của mỗi chiến lược ứng viên.
- **Adaptive Fitness Function:** Hàm fitness là một hàm tổng hợp có trọng số, có thể được điều chỉnh dựa trên hồ sơ rủi ro của người dùng để tạo ra các chiến lược được cá nhân hóa.

Bản cập nhật v2.0 này không chỉ thêm các tính năng mới mà còn định hình một kiến trúc hệ thống rõ ràng, mạch lạc và chuyên nghiệp. Nó đặt nền móng vững chắc cho việc phát triển ITAPIA thành một sản phẩm AI toàn diện, minh bạch và thực sự có giá trị.