Investment Advisory AI Platform

Hemprasad Badgujar

Agent Roles & Responsibilities Overview

- **Main Supervisor Agent**
  - **Role:** Master orchestrator of investment advisory ecosystem
  - **Key Responsibilities:** Query routing, multi-team coordination, safety management, context management, decision integration
- **Analysis Team Supervisor (Sub-supervisor)**
  - **Role:** Quantitative analysis and portfolio management coordinator
  - **Key Responsibilities:** Math/Portfolio expert coordination, quantitative analysis oversight, technical analysis coordination

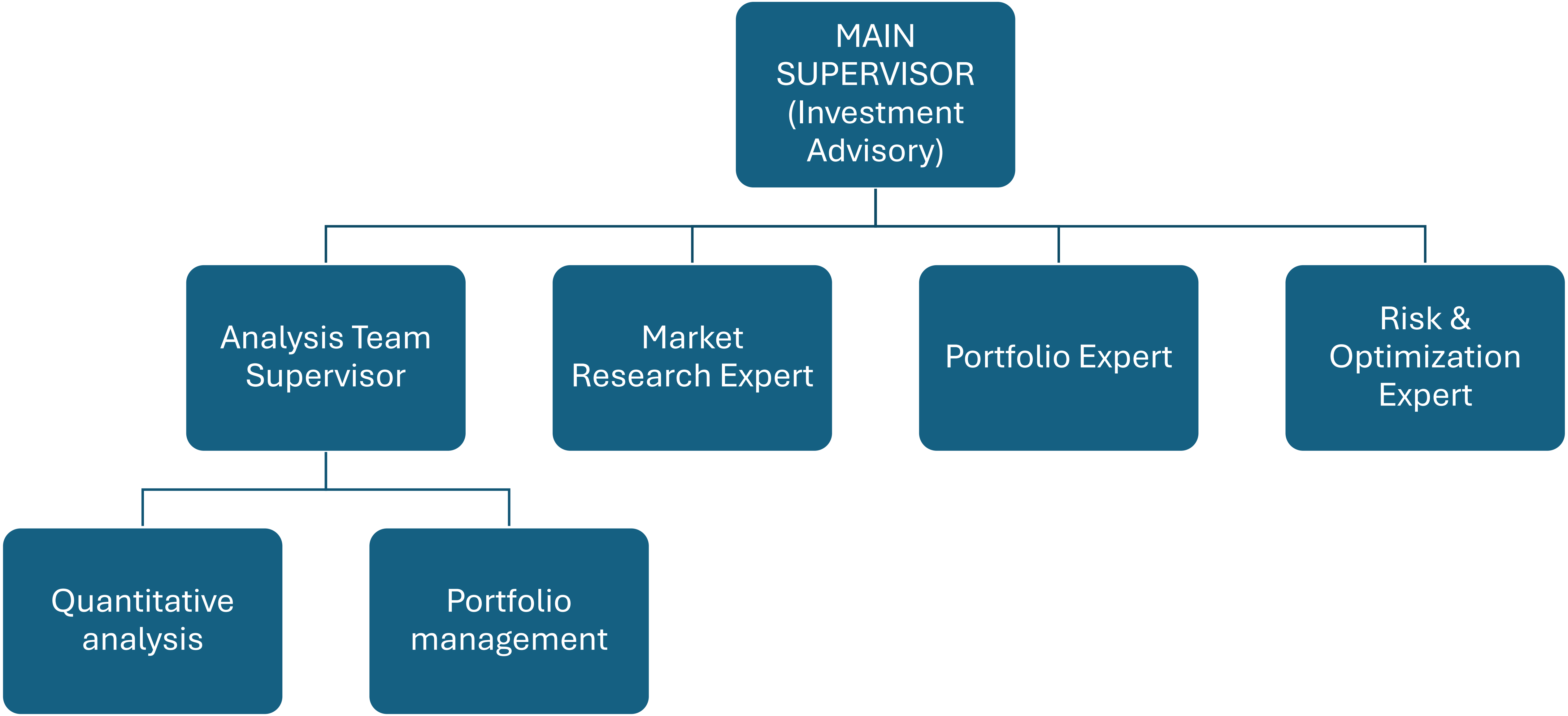
Specialized Expert Agents:

- **Market Research Expert**
  - **Role:** Market intelligence and economic analysis specialist
  - **Key Capabilities:** Company research (SEC filings), economic indicators, news sentiment analysis, sector performance, market trends, regulatory impact, global market monitoring

- **Portfolio Expert**
  - **Role:** Portfolio management and securities analysis specialist
  - **Key Capabilities:** Portfolio evaluation, technical analysis (15+ indicators), securities tracking, transaction analysis, asset allocation, fundamental analysis, rebalancing, performance attribution

- **Risk Optimization Expert**
  - **Role:** Risk management and regulatory compliance specialist
  - **Key Capabilities:** VaR/CVaR calculations, risk profiling, compliance monitoring, stress testing (5 scenarios), MPT optimization, violation detection, dynamic risk adjustment

Agent Graph & Hierarchical Structure



MAIN SUPERVISOR (Investment Advisory)

Analysis Team Supervisor

Market Research Expert

Portfolio Expert

Risk & Optimization Expert

Quantitative analysis

Portfolio management

Agent Communication Patterns

- 1. Hierarchical Delegation:
  - Main Supervisor → Specialized Teams
  - Analysis Team Supervisor → Math/Portfolio Experts
- 2. Cross-Team Coordination:
  - Research Expert ↔ Portfolio Expert (market context)
  - Risk Expert ↔ Portfolio Expert (risk assessment)
  - Math Expert ↔ All Teams (calculations)
- 3. Safety & Validation Layer:
  - LlamaGuard input validation
  - Response validation and quality checks
  - Content safety filtering

Data Flow Architecture

- Core Components:
  - AgentDataPacket Structure:
  - SharedDataCache:
    - Thread-safe data storage for multi-agent sessions
    - Data filtering by type, source, target, and timestamp
    - Agent subscription system for relevant data types
  - Data Flow Tools:
    - share_data_with_agents(): Broadcast or targeted data sharing
    - get_shared_data(): Retrieve filtered data packets
    - coordinate_workflow(): Multi-agent workflow orchestration

Data Flow Patterns - Workflow Coordination

Pre-defined Investment Workflows

- 1. Comprehensive Portfolio Analysis Workflow:
    - Step 1: Portfolio Expert → Analyze portfolio holdings
    - Step 2: Risk Expert → Assess portfolio risk
    - Step 3: Research Expert → Research portfolio securities
    - Step 4: Math Expert → Calculate risk metrics
    - Step 5: Risk Expert → Generate optimization recommendations
- 2. Market Condition Assessment Workflow:
    - Step 1: Research Expert → Analyze current market
    - Step 2: Risk Expert → Assess economic risks
    - Step 3: Math Expert → Calculate market metrics
    - Step 4: Portfolio Expert → Evaluate portfolio impact
- 3. Client Risk Evaluation Workflow:
    - Step 1: Portfolio Expert → Get client profile
    - Step 2: Risk Expert → Evaluate risk tolerance
    - Step 3: Research Expert → Analyze market conditions
    - Step 4: Risk Expert → Generate risk recommendations

DataFlowCoordinator Features:

- Workflow Registration: Template-based workflow definitions
- Execution Tracking: Step-by-step progress monitoring
- State Management: Active workflow status and results
- Error Handling: Retry logic and failure recovery

Real-time Data Synchronization:

- Thread-specific data isolation
- Timestamp-based data ordering
- Automatic data expiration and cleanup
- Cross-agent data dependency resolution

Decision-Making Framework

- **1. Safety-First Approach:**
  - LlamaGuard content validation before processing
  - Unsafe content blocking with explanatory messages
  - Response validation for quality assurance
- **2. Context-Aware Routing:**
  - Current market conditions consideration
  - Client-specific risk profile integration
  - Historical interaction context
- **3. Capability Matching:**
  - Tool availability assessment
  - Agent expertise alignment
  - Resource optimization

External System Integrations

- **1. Yahoo Finance (yfinance)**
  - Purpose: Real-time market data, historical prices, technical indicators
  - Cost: Free
  - Rate Limits: Built-in delays for reasonable usage
  - Integration: Direct Python library integration
  - Data: Stock prices, volume, market cap, financial ratios
- **2. Alpha Vantage API**
  - Purpose: Advanced market data and technical indicators
  - Cost: Free tier (5 req/min), paid plans available
  - Integration: REST API with caching layer
  - Data: Technical indicators, forex, crypto, fundamental data
- **3. Federal Reserve Economic Data (FRED)**
  - Purpose: Economic indicators (GDP, inflation, unemployment)
  - Cost: Free (120 requests/minute)
  - Integration: FRED Python API
  - Data: 800,000+ economic time series
  - News & Sentiment Analysis:
- **4. NewsAPI**
  - Purpose: News sentiment analysis and market psychology
  - Cost: Free tier (1,000 requests/day)
  - Integration: REST API with sentiment processing
  - Data: Real-time news, historical articles, source filtering
- **5. Financial Modeling Prep API**
  - Purpose: SEC filings, company fundamentals, financial ratios
  - Cost: Paid service with comprehensive data
  - Integration: REST API with retry logic
  - Data: 10-K/10-Q filings, financial statements, ratios

Production System Architecture

- **Docker Container Strategy:**
- 1. Agent Service Container (Dockerfile.service)
- 2. Streamlit App Container (Dockerfile.app)
- 3. PostgreSQL Container

- **Docker Compose Production Setup:**
- **Service Dependencies:**
  - PostgreSQL → Agent Service → Streamlit App
  - Production Considerations:
  - Scalability:
    - Horizontal scaling with load balancers
    - Database read replicas
    - Redis for session management
    - CDN for static assets
  - Monitoring:
    - LangSmith for agent tracing
    - Langfuse for observability
    - Health check endpoints
    - Performance metrics collection

Production Go-To Plan

- **Deployment & Operations Strategy**
- **Phase 1: Infrastructure**
- **Cloud Platform Selection:**
  - AWS/Azure/GCP: Container orchestration with EKS/AKS/GKE
  - Database: Managed PostgreSQL (RDS/Azure Database/Cloud SQL)
  - Load Balancing: Application Load Balancer with SSL termination
  - Monitoring: CloudWatch/Azure Monitor/Stackdriver integration
- **Security Implementation:**
  - API Gateway: Rate limiting, authentication, SSL/TLS
  - Secrets Management: AWS Secrets Manager/Azure Key Vault
  - Network Security: VPC, security groups, private subnets
  - Compliance: SOC 2, financial data protection standards

- **Phase 2: Application Deployment**
- CI/CD Pipeline:
  - GitHub → Docker Build → Security Scan → Deploy to Staging →
  - Performance Tests → Manual Approval → Production Deployment

Production Go-To Plan

- **Phase 3: Monitoring & Optimization**
- **Observability Stack:**
  - Application Monitoring: LangSmith agent tracing, Langfuse analytics
  - Infrastructure Monitoring: Prometheus + Grafana
  - Log Management: ELK Stack (Elasticsearch, Logstash, Kibana)
  - Alerting: PagerDuty integration for critical issues
- **Performance Optimization:**
  - API Caching: Redis for frequently accessed data
  - Database Optimization: Query optimization, indexing strategy
  - CDN: CloudFront/Azure CDN for static assets
  - Auto-scaling: CPU/memory-based scaling policies

- **Phase 4: Production Readiness**
- **Disaster Recovery:**
  - Database Backups: Automated daily backups with point-in-time recovery
  - Multi-region Deployment: Active-passive setup for high availability
  - Data Replication: Cross-region database replication
  - Backup Testing: Monthly disaster recovery drills

Thank you