

# Component Relationships and Dependencies Diagram

---

```

graph TB
    subgraph "Entry Points"
        Main[main.py
CLI Entry Point]
        App[ui/app.py
Streamlit Web UI]
        Setup[setup.py
Environment Setup]
    end

    subgraph "Core Orchestration"
        Graph[graphs/investment_graph.py
LangGraph Definition]
        Exec[graphs/execution.py
Workflow Execution]
    end

    subgraph "Agent System"
        News[agents/news_agent.py
News Agent]
        Data[agents/data_agent.py
Data Agent]
        Analyst[agents/analyst_agent.py
Analyst Agent]
        Expert[agents/financial_expert_agent.py
Financial Expert]
    end

    subgraph "Tools & Utilities"
        Search[tools/search_tool.py
Search Tool]
        Config[config/settings.py
Configuration]
        Logging[config/logging_config.py
Logging]
    end

    subgraph "External Dependencies"
        GROQ[GROQ API
LLM Service]
        Yahoo[Yahoo Finance
Financial Data]
        DDG[DuckDuckGo
News Search]
    end

    subgraph "Output & Storage"
        Analysis[outputs/analysis/

```

```
Analysis Reports]
    Recs[outputs/recommendations/
Recommendations]
    Logs[outputs/logs/
System Logs]
end

%% Entry Point Dependencies
Main --> Graph
Main --> Exec
App --> Graph
App --> Exec
Setup --> Config
Setup --> Analysis
Setup --> Recs
Setup --> Logs

%% Core Orchestration Dependencies
Graph --> News
Graph --> Data
Graph --> Analyst
Graph --> Expert
Exec --> Graph
Exec --> Analysis
Exec --> Recs
Exec --> Logs

%% Agent Dependencies
News --> Search
News --> GROQ
Data --> Yahoo
Data --> GROQ
Analyst --> GROQ
Expert --> GROQ

%% Tool Dependencies
Search --> DDG
News --> Config
Data --> Config
Analyst --> Config
Expert --> Config

%% Logging Dependencies
Main --> Logging
App --> Logging
Setup --> Logging
News --> Logging
Data --> Logging
Analyst --> Logging
Expert --> Logging

%% Configuration Dependencies
News --> Config
Data --> Config
```

```
Analyst --> Config
Expert --> Config
Logging --> Config

style Main fill:#e3f2fd
style App fill:#e3f2fd
style Setup fill:#e3f2fd
style Graph fill:#f3e5f5
style Exec fill:#f3e5f5
style News fill:#fff3e0
style Data fill:#fff3e0
style Analyst fill:#fff3e0
style Expert fill:#fff3e0
style Config fill:#e8f5e8
style Logging fill:#e8f5e8
```

## Component Dependencies

### Entry Points

- **main.py**: Primary CLI interface, depends on orchestration and execution modules
- **ui/app.py**: Web interface, depends on orchestration and execution modules
- **setup.py**: Environment setup, depends on configuration and output directories

### Core Orchestration

- **investment\_graph.py**: Defines LangGraph workflow and agent interactions
- **execution.py**: Executes workflow and manages output generation

### Agent System

- **news\_agent.py**: News gathering and sentiment analysis
- **data\_agent.py**: Financial data retrieval and processing
- **analyst\_agent.py**: Information synthesis and analysis
- **financial\_expert\_agent.py**: Investment recommendations

### Tools & Utilities

- **search\_tool.py**: DuckDuckGo search integration
- **settings.py**: Centralized configuration management
- **logging\_config.py**: Logging setup and management

### External Dependencies

- **GROQ API**: Large Language Model service
- **Yahoo Finance**: Financial data provider
- **DuckDuckGo**: News search service

### Output & Storage

- **analysis/**: Generated analysis reports
- **recommendations/**: Investment recommendations
- **logs/**: System execution logs

## Key Relationships

- **Configuration Centralization**: All agents depend on centralized settings
- **Logging Integration**: All components use centralized logging system
- **LLM Integration**: Multiple agents use GROQ for AI processing
- **Data Flow**: Clear dependency chain from input to output generation