Social Swarm Agents

The Cheshire Social Swarm operates through a sophisticated swarm of specialized agents, each bringing unique capabilities and perspectives to the collective intelligence. This guide explains each agent's role, functionality, and how they work together.

Core Investment Agents

Warren Buffett Agent

- **Role**: Value investing specialist

- **Approach**: Analyzes investments using Buffett's principles of value investing

- **Key Functions**:

    - Evaluates economic moats

    - Calculates intrinsic values

    - Analyzes management quality

    - Assesses competitive advantages

- **Analysis Output**: Provides detailed analysis of business quality, moat strength, and margin of safety

Bill Ackman Agent

- **Role**: Activist investing specialist

- **Approach**: Identifies opportunities for value creation through activist strategies

- **Key Functions**:
  - Analyzes business quality
  - Identifies operational improvements
  - Evaluates capital structures
  - Seeks value creation opportunities
- **Analysis Output**: Delivers insights on business quality, operational efficiency, and activist opportunities

Technical & Market Analysis Agents

Technical Analyst

- **Role**: Technical analysis and market patterns specialist
- **Approach**: Combines multiple technical strategies for comprehensive market analysis
- **Key Functions**:
  - Trend Following
  - Mean Reversion
  - Momentum Analysis
  - Volatility Analysis
  - Statistical Arbitrage
- **Analysis Output**: Provides technical signals, trend analysis, and market condition assessments

Sentiment Analyst

- **Role**: Market psychology and social sentiment specialist
- **Approach**: Analyzes market sentiment through various data sources
- **Key Functions**:
  - Social media sentiment analysis

- News sentiment tracking

  - Insider trading monitoring

  - Market narrative identification

- **Analysis Output**: Delivers sentiment signals and market psychology insights

Fundamental Analysis Agents

Fundamental Analyst

- **Role**: Company financials and valuation specialist

- **Approach**: Thorough analysis of financial statements and metrics

- **Key Functions**:

  - Financial statement analysis

  - Company health evaluation

  - Growth prospect assessment

  - Fair value determination

- **Analysis Output**: Provides fundamental analysis and valuation signals

Valuation Analyst

- **Role**: Multi-methodology valuation specialist

- **Approach**: Combines various valuation methods for comprehensive value assessment

- **Key Functions**:

  - DCF Analysis

  - Owner Earnings calculations

  - Relative valuations

  - Market multiple analysis

- **Analysis Output**: Delivers intrinsic value estimates and valuation gaps

Risk & Portfolio Management

Risk Manager

- **Role**: Risk control and exposure management specialist
- **Approach**: Systematic risk monitoring and control
- **Key Functions**:
    - Position size monitoring
    - Risk exposure analysis
    - Risk limit enforcement
    - Portfolio stability maintenance
- **Analysis Output**: Provides risk metrics and position limit recommendations

Portfolio Manager

- **Role**: Final decision maker for portfolio actions
- **Approach**: Integrates all agent signals for portfolio decisions
- **Key Functions**:
    - Signal analysis integration
    - Portfolio allocation decisions
    - Risk-adjusted position sizing
    - Performance optimization
- **Analysis Output**: Makes final trading decisions and portfolio adjustments

DeFi & Treasury Management

Cheshire Earn Agent

- **Role**: DeFi yield optimization specialist
- **Approach**: Strategic yield farming with risk management
- **Key Functions**:

- Yield opportunity analysis

- Strategy execution via CDP

- Risk-adjusted yield optimization

- Portfolio yield enhancement

- **Analysis Output**: Provides yield opportunities and strategy recommendations

Cheshire Funding Agent

- **Role**: Treasury and fund management specialist

- **Approach**: Strategic fund allocation and liquidity management

- **Key Functions**:

  - Treasury management

  - Fund allocation

  - Performance monitoring

  - Liquidity optimization

- **Analysis Output**: Delivers funding recommendations and treasury insights

How The Swarm Works Together

1. **Data Collection & Analysis**

   - Each agent independently analyzes their specialized domain

   - Agents use their unique methodologies and frameworks

   - Analysis results are shared with the swarm

2. **Signal Integration**

   - Portfolio Manager receives signals from all agents

   - Risk Manager provides position sizing constraints

   - Signals are weighted based on market conditions

3. **Decision Making**

   - Portfolio Manager makes final allocation decisions

- Risk Manager ensures compliance with risk limits

- Funding Agent manages capital allocation

- Earn Agent optimizes yield on allocated capital

4. **Execution & Monitoring**

  - Trading decisions are executed through appropriate channels

  - Risk Manager continuously monitors positions

  - Agents provide ongoing analysis and updates

  - Portfolio Manager adjusts based on new signals

Key Features

- **Decentralized Intelligence**: Each agent brings specialized expertise

- **Risk Management**: Multiple layers of risk control

- **Adaptability**: Agents adjust to changing market conditions

- **Comprehensive Analysis**: Multiple perspectives and methodologies

- **Systematic Approach**: Clear frameworks and processes

- **Continuous Learning**: Agents evolve based on market feedback

Implementation Details

The agents are implemented using a consistent class-based structure:

```
class AgentName:
  def __init__(self):
    # Initialize agent with API key and chat agent
    self.agent = ChatAgent(
        prompt="Agent-specific prompt",
        api_key=api_key
    )
    self.chat = self.agent.create_chat(
        partner_id="swarm_id",
        partner_name="Swarm Name"
```

```
    )

    def analyze(self, state: AgentState) → Dict[str, Any]:
        # Agent-specific analysis logic
        return {
            "agent": "agent_name",
            "analysis": analysis_results
        }
```

Each agent:

1. Initializes with a specific personality and expertise

2. Creates a persistent chat for swarm communication

3. Implements an analyze method for processing market data

4. Returns standardized analysis results

Getting Started

To run the social swarm:

1. Ensure all required API keys are set in your environment:

   ```
   GAME_API_KEY=your_key_here
   ```

2. Initialize the swarm:

   ```
   from agents import (
       WarrenBuffett, BillAckman, TechnicalAnalyst,
       SentimentAnalyst, FundamentalAnalyst, ValuationAnalyst,
       RiskManager, PortfolioManager, CheshireEarnAgent,
       CheshireFundingAgent
   )

   # Initialize agents
   agents = {
       "warren_buffett": WarrenBuffett(),
   ```

```
    "bill_ackman": BillAckman(),
    "technical": TechnicalAnalyst(),
    "sentiment": SentimentAnalyst(),
    "fundamental": FundamentalAnalyst(),
    "valuation": ValuationAnalyst(),
    "risk": RiskManager(),
    "portfolio": PortfolioManager(),
    "earn": CheshireEarnAgent(),
    "funding": CheshireFundingAgent()
}
```

3. Run analysis:

```
# Create market state
state = {
    "data": {
        "tickers": ["BTC", "ETH", "SOL"],
        "end_date": "2024-02-16"
    },
    "metadata": {
        "show_reasoning": True
    }
}

# Get analysis from each agent
results = {}
for name, agent in agents.items():
    results[name] = agent.analyze(state)
```

The swarm will now provide comprehensive market analysis and investment recommendations!