

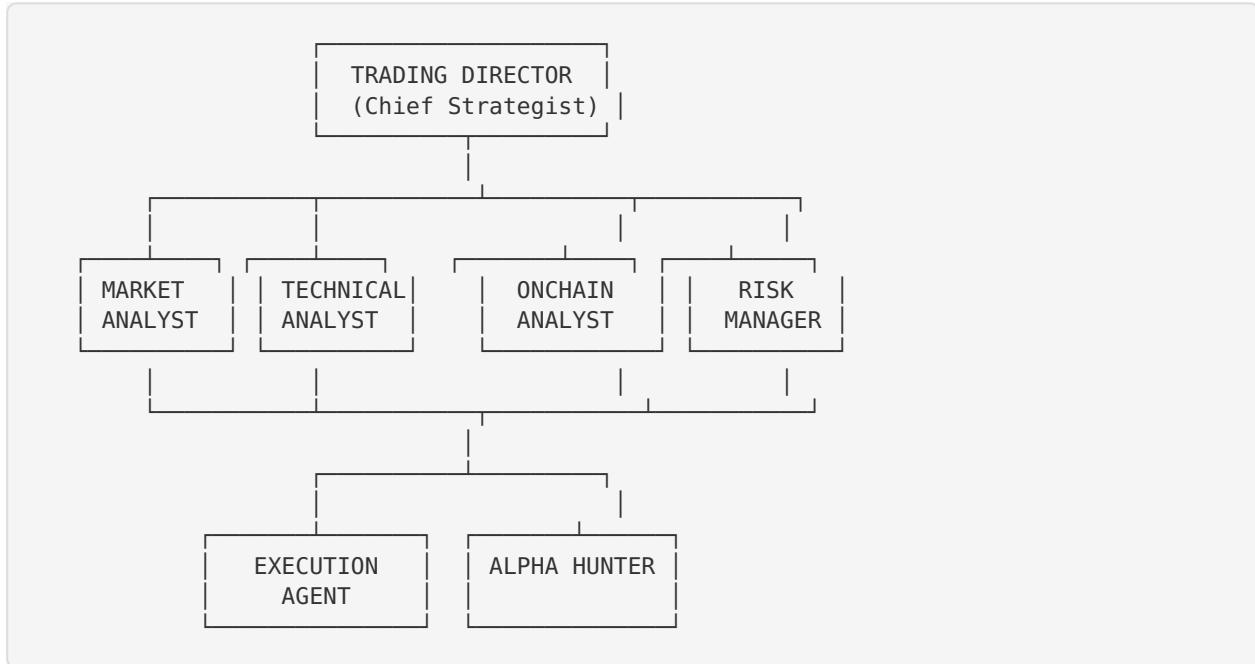
Swarms AI Hierarchical Framework Integration

Overview

This integration brings enterprise-grade multi-agent orchestration to the Smart Money Tracker platform using the **Swarms AI Hierarchical Framework**. The system deploys specialized AI agents that work together in coordinated workflows to provide comprehensive trading intelligence.

Architecture

Hierarchical Swarm Structure



Specialized Agents

1. Trading Director (Chief Strategist)

- Role:** Coordinates all analysis, synthesizes insights, makes final decisions
- Capabilities:** Strategic planning, agent coordination, decision synthesis
- Model:** GPT-4o with extended context

2. Market Analyst

- Role:** Analyzes market conditions, trends, and sentiment
- Outputs:** Market overview, sector analysis, sentiment scores
- Data Sources:** Price feeds, volume data, market cap trends

3. Risk Manager

- Role:** Evaluates trading risks and position sizing
- Outputs:** Risk scores, stop-loss levels, position sizing recommendations

- **Capabilities:** Portfolio exposure analysis, R/R calculations

4. Technical Analyst

- **Role:** Chart pattern recognition and indicator analysis
- **Outputs:** Support/resistance levels, trend analysis, entry zones
- **Indicators:** RSI, MACD, Bollinger Bands, Moving Averages

5. OnChain Analyst

- **Role:** Blockchain data and smart money tracking
- **Outputs:** Whale activity, exchange flows, accumulation signals
- **Data Sources:** Nansen, on-chain metrics

6. Execution Agent

- **Role:** Trade execution planning and optimization
- **Outputs:** Entry strategies, slippage estimates, venue selection
- **Capabilities:** DCA planning, gas optimization

7. Alpha Hunter

- **Role:** Identifies early opportunities and hidden alpha
- **Outputs:** Emerging narratives, undervalued gems, catalyst tracking
- **Focus:** New launches, smart money early entries

Available Workflows

1. Hierarchical Swarm (Default)

```
const result = await swarmsClient.executeHierarchicalSwarm(
  "Analyze BTC market conditions",
  { marketData: { ... } }
);
```

- Director-led comprehensive analysis
- All specialists contribute insights
- Final synthesized recommendation

2. Concurrent Analysis

```
const result = await swarmsClient.executeConcurrentAnalysis(
  "Quick market scan",
  ['MARKET_ANALYST', 'TECHNICAL_ANALYST', 'RISK_MANAGER']
);
```

- Parallel execution for speed
- Multiple perspectives simultaneously
- Best for quick multi-view analysis

3. Sequential Workflow

```
const result = await swarmsClient.executeSequentialWorkflow(
  "Full analysis pipeline for ETH"
);
```

- Step-by-step analysis
- Each agent builds on previous output
- Market → Technical → OnChain → Risk → Execution

4. Alpha Hunter

```
const result = await swarmsClient.huntAlpha({
  sector: 'AI',
  timeframe: '24h'
});
```

- Identify early opportunities
- Smart money tracking
- Emerging narrative detection

5. Smart Router

```
const result = await swarmsClient.routeToSpecialist(
  "What's the risk of going 5x long on SOL?"
);
```

- Automatic specialist selection
- Task-appropriate routing
- Optimal agent matching

API Endpoints

POST /api/agents/analyze

Main analysis endpoint for all workflows.

```
{
  "task": "Analyze BTC market conditions",
  "workflow": "hierarchical",
  "includeMarketData": true,
  "token": {
    "symbol": "BTC",
    "chain": "ethereum"
  }
}
```

POST /api/agents/quick

Quick action endpoints for common tasks.

```

// Market Sentiment
{ "action": "market_sentiment" }

// Analyze Token
{ "action": "analyze_token", "data": { "symbol": "ETH" } }

// Risk Assessment
{ "action": "assess_risk", "data": { "position": "BTC Long", "size": 10000, "leverage": 3 } }

// Hunt Alpha
{ "action": "find_alpha", "data": { "sector": "DeFi" } }

// Whale Analysis
{ "action": "whale_analysis", "data": { "chain": "ethereum" } }

// Execution Plan
{ "action": "execution_plan", "data": { "action": "buy", "token": "SOL", "amount": 25000 } }

```

GET /api/agents/analyze

Returns available workflows and agents.

UI Dashboard

Access

Navigate to `/agentic-trading` in the application.

Features

1. **Workflow Selection:** Choose from 5 different multi-agent workflows
2. **Quick Actions:** One-click common analysis tasks
3. **Single Agent Mode:** Direct access to specific specialists
4. **Context Options:** Include market data, specify tokens
5. **Analysis History:** Track recent analyses
6. **Real-time Results:** Formatted output with agent attribution

Configuration

Environment Variables

```
SWARMS_API_KEY=sk-xxxxx # Your Swarms API key
```

Client Initialization

```

import { swarmsClient } from '@lib/ai-agents/swarms-client';

// Client auto-initializes with env variable
const result = await swarmsClient.executeHierarchicalSwarm(task);

```

Files Created

File	Purpose
/lib/ai-agents/swarms-client.ts	Main Swarms client with all agents and workflows
/app/api/agents/analyze/route.ts	Primary analysis API endpoint
/app/api/agents/quick/route.ts	Quick action API endpoints
/app/agentic-trading/page.tsx	UI dashboard for agent interactions

Best Practices

Task Formulation

- Be specific about what you want analyzed
- Include relevant context (token, timeframe, chain)
- Specify desired output format

Workflow Selection

- **Hierarchical:** Comprehensive analysis, important decisions
- **Concurrent:** Quick insights, multiple perspectives
- **Sequential:** Step-by-step analysis, full pipeline
- **Alpha:** Finding opportunities, early entries
- **Router:** When unsure which specialist to use

Performance

- Enable market data for richer context
- Use single agent for simple queries
- Use concurrent for time-sensitive analysis

Response Format

```
interface SwarmResult {
  job_id: string;
  status: 'completed' | 'pending' | 'failed';
  output: any; // Analysis results
  number_of_agents: number;
  execution_time: number; // seconds
  billing_info?: {
    total_cost: number;
    token_usage: number;
  };
  agent_outputs?: AgentOutput[]; // Individual agent results
}
```

Integration with Existing Features

Market Data Integration

- Automatically includes real-time prices from price-service
- Supports Nansen smart money data injection

Whale Tracking

- OnChain Analyst leverages whale tracking data
- Exchange flow analysis included

Risk Management

- Portfolio-aware position sizing
 - Stop-loss recommendations
-

Built with Swarms AI • Enterprise Multi-Agent Orchestration