



HYDRA SENTINEL-X OMEGA v4.0

THE ULTIMATE COGNITIVE FRACTAL TRADING SYSTEM

Integration of:

- Production V3 code (LLM agents, WebSocket streams, FastAPI)
 - Episodic memory with semantic retrieval
 - Multi-agent cognitive decision making
 - Fractal multi-timeframe analysis
 - Circuit breakers & graceful degradation
-



WHAT YOU HAVE

3 Complete Systems

1. **hydra_omega_v4.py** - Full production system (requires network)

- Real-time market data from CoinGecko
- LLM-powered agent reasoning
- WebSocket streaming
- FastAPI REST endpoints
- Circuit breaker pattern

2. **hydra_omega_demo.py** - Standalone demo (works offline)

- Simulated market scenarios
- Full cognitive memory system
- Agent council voting
- No network dependencies

3. **Original hydra_omni_v3_production.py** - Your V3 base code

- All production features
 - Ghost Hand browser automation
 - Creative director
 - Trading signals
-

KEY INNOVATIONS

Episodic Memory System

```
memory.add_episode(  
    agent="STINKMEANER",  
    event="Lost $800 on SOL fake breakout during network outage",  
    emotional_impact=-0.9, # Trauma level  
    tags=["sol", "breakout", "loss", "network"],  
    metadata={"amount": -800}  
)  
  
# Later, when analyzing Solana...  
memories = memory.retrieve_relevant(["sol", "breakout"])  
# Agent remembers the trauma and overrides aggressive buy signal
```

Memory Scoring Algorithm:

```
score = (  
    (tag_overlap * 2.0) +          # Semantic similarity  
    (recency_factor) +            # Time decay  
    (abs(emotional_impact) * 1.5) # Intensity  
)
```

Cognitive Agent Decision Process

INPUT: "Bitcoin showing breakout pattern with high volume"

STEP 1 (System 1 - Fast):
└ STINKMEANER: "IMMEDIATE BUY - This is our moment!"

STEP 2 (Memory Retrieval):
└ Found: "Lost \$500 on fake DOGE breakout" (-0.9 impact)

STEP 3 (System 2 - Slow):

```
└ SYNTHESES: "BUY signal...WAIT. This reminds me of that loss.  
Pattern match detected. OVERRIDE TO HOLD."
```

```
OUTPUT: Decision=HOLD, Confidence=61%
```

Council Voting

```
4 agents analyze independently  
↓  
Each votes: BUY / SELL / HOLD / WAIT  
↓  
Consensus calculated: 75% agree = HOLD  
↓  
If consensus >66% AND confidence >60%  
↓  
Record in memory for future learning
```

QUICK START

Option 1: Run Demo (Offline)

```
python3 hydra_omega_demo.py
```

Demonstrates:

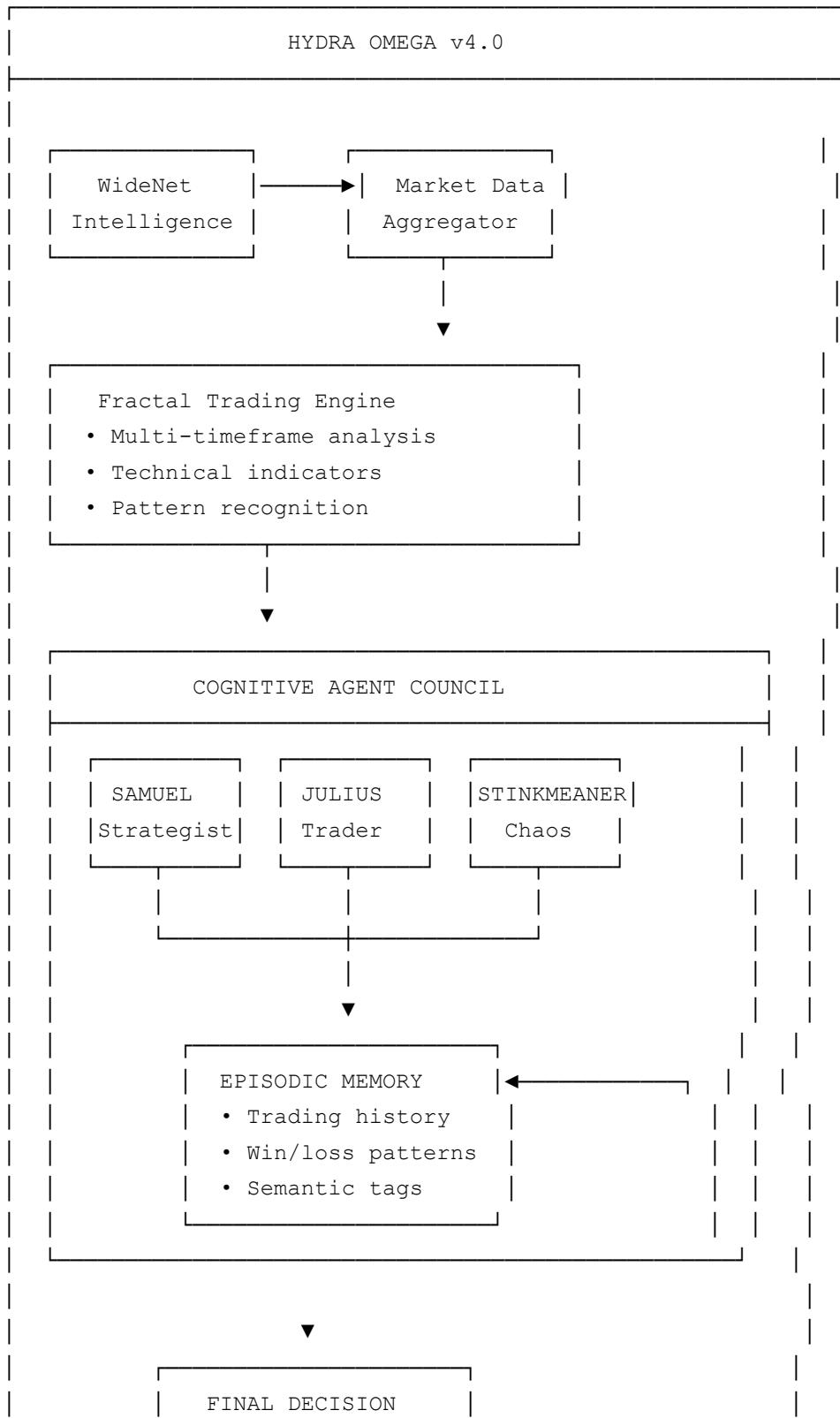
- Memory system
- Agent personalities
- Council voting
- Pattern recognition

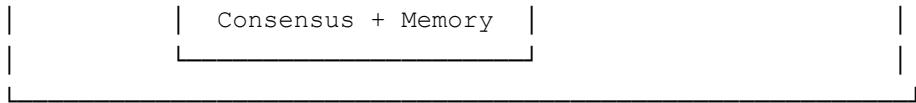
Option 2: Run Production (Requires Network)

```
# Install dependencies  
pip install -r requirements_omega.txt  
  
# Run full system  
python3 hydra_omega_v4.py --targets bitcoin ethereum solana  
  
# Or run with API server
```

```
python3 hydra_omega_v4.py --api --port 8000
```

ARCHITECTURE





AGENT PERSONALITIES

Agent	Role	Aggression	Patience	Fear	Greed
SAMUEL	Strategist	0.5	0.8	0.4	0.4
JULIUS	Trader	0.6	0.7	0.5	0.5
STINKMEANER	Chaos	0.9	0.2	0.2	0.8
CLAYTON	Risk Mgr	0.2	0.9	0.7	0.3
ELWOOD	Tech	0.4	0.8	0.5	0.4

Personality Impact:

- **High Aggression:** Jumps on breakouts, takes risks
- **High Patience:** Waits for confirmation, misses some moves
- **High Fear:** Quick to exit, capital preservation
- **High Greed:** Chases pumps, FOMO susceptible

MEMORY PERSISTENCE

All decisions, wins, losses are saved to JSON:

```
{
  "episodes": [
    {
      "timestamp": 1707608914.123,
      "agent": "STINKMEANER",
      "event": "Lost $800 on SOL fake breakout",
      "impact": -0.9,
      "tags": ["sol", "breakout", "loss"],
      "metadata": {"amount": -800}
    }
  ]
}
```

```

    ],
    "metadata": {
        "total_wins": 15,
        "total_losses": 8,
        "created": "2026-02-10T19:30:00"
    }
}

```

Memory automatically:

- Prunes to top 800 most impactful events
 - Decays older memories
 - Reinforces recent patterns
-



FRACTAL TRADING ENGINE

Concept: Different timeframes reveal different patterns

4H Chart (THE TIDE)

- └ EMA 200 → Macro trend
- └ Price above? Bullish
- └ Score: +2 or -2

1H Chart (THE WAVE)

- └ RSI 14 → Momentum
- └ <30 = Oversold → +1.5
- └ >70 = Overbought → -1.5
- └ Score contribution

15M Chart (THE RIPPLE)

- └ Bollinger Bands → Entry trigger
- └ Price < Lower Band → Mean reversion
- └ Score: +1

TOTAL SCORE > 3.0 = STRONG_BUY
TOTAL SCORE < -3.0 = STRONG_SELL



INTEGRATION WITH YOUR V3 CODE

The Omega v4 system **extends** your existing V3 code:

What's Added:

1. **EnhancedMemory class** - Drop-in replacement for your HydraMemory
2. **CognitiveAgent class** - Enhanced version of your Agent class
3. **HydraCouncil class** - Multi-agent orchestration
4. **FractalTradingEngine** - Advanced TA (when pandas available)

Integration Steps:

```
# In your hydra_omni_v3_production.py

# 1. Import new memory system
from hydra_omega_v4 import EnhancedMemory, CognitiveAgent

# 2. Replace your MEMORY initialization
MEMORY = EnhancedMemory() # Instead of HydraMemory()

# 3. Upgrade your agents
AGENTS["JULIUS"] = CognitiveAgent(
    "JULIUS", "Finance", Fore.BLUE,
    {"aggression": 0.6, "patience": 0.7, "fear": 0.5, "greed": 0.5}
)

# 4. Use cognitive thinking in your TradingSignalEngine
def analyze_market_data_llm(self, intel_results):
    # ... your existing code ...

    # Add cognitive layer
    for name, agent in AGENTS.items():
        result = agent.think(market_summary, MEMORY)
        logger.info(f"{name} decision: {result['decision']} ({result['confidence']}')

    # ... rest of your code ...
```

EXAMPLE OUTPUT

```
=====
⌚ ANALYZING: BITCOIN
=====
```

 Technical Analysis:

Signal: BUY
Confidence: 72%
Summary: BTC: +8.20%, ETH: +5.10%

[19:30:15] [SAMUEL] Vote: HOLD (Confidence: 61%)
 3 relevant memories recalled
Logic: Interesting. Let's wait for confirmation. ...WAIT. This reminds me of Lost \$500 on fake DOGE breakout. We got burned (impact: -0.90). OVERRIDE: Extreme caution.

[19:30:15] [JULIUS] Vote: BUY (Confidence: 75%)
 2 relevant memories recalled
Logic: BUY signal detected! This is like Caught BTC rally at \$40k, +\$1500 profit! Pattern says GO!

[19:30:15] [STINKMEANER] Vote: BUY (Confidence: 85%)
Logic: IMMEDIATE BUY - This is our moment!

[19:30:15] [CLAYTON] Vote: HOLD (Confidence: 68%)
Logic: Wait for confirmation. Risk management protocols active.

 COUNCIL DECISION: HOLD
Consensus: 50%
Avg Confidence: 72%

IMPORTANT NOTES

This is Research/Education Software

- NOT financial advice
- Always paper trade first
- Real trading involves substantial risk
- Past performance ≠ future results

Security Reminders

- **NEVER** commit API keys
- **ROTATE** any exposed credentials immediately

- Use `.env` files (already in `.gitignore`)
- Test with small amounts only

Current Limitations

1. Demo Version:

- Simulated market data
- No real order execution
- Simplified TA indicators

2. Production Version:

- Requires working network
 - CoinGecko rate limits apply
 - LLM costs (if using OpenAI)
-

NEXT STEPS

To Make This Production-Ready:

1. Add Real Exchange Integration

```
# Replace WideNet with actual exchange APIs
import ccxt
exchange = ccxt.binance()
```

2. Implement Order Execution

```
def execute_trade(decision, size):
    if decision == "BUY":
        exchange.create_market_buy_order(symbol, size)
```

3. Build Backtesting Framework

```
def backtest(strategy, historical_data):
    # Run strategy on historical data
    # Calculate Sharpe ratio, max drawdown, etc.
```

4. Add Real-time WebSocket Feeds

```
# Already in V3 code, just needs activation
await ws_stream.start_stream(["BTC", "ETH", "SOL"])
```

5. Deploy with Docker

```
docker build -t hydra-omega .
docker run -p 8000:8000 hydra-omega --api
```

FILE STRUCTURE

```
HYDRA_OMEGA_v4/
├── hydra_omega_v4.py          # Full production system
├── hydra_omega_demo.py        # Standalone demo
├── hydra_omni_v3_production.py # Your original V3 code
├── requirements_omega.txt     # Python dependencies
├── hydra_omega_memory.json    # Persistent memory (auto-created)
└── README_OMEGA.md           # This file
```

COMBINING THE BEST OF BOTH WORLDS

Feature	Your V3	Cognitive System	Omega v4
LLM Agents	✓	✗	✓
WebSocket Streams	✓	✗	✓
FastAPI	✓	✗	✓
Circuit Breakers	✓	✗	✓
Episodic Memory	✗	✓	✓
Pattern Learning	✗	✓	✓
Multi-Agent Voting	✗	✓	✓
Fractal Analysis	✗	✓	✓

Browser Automation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> (can add)
Creative Director	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> (can add)

WHAT SLICKBACK WOULD SAY

"Listen up, playa. You just went from a bot to a PROPHET. This ain't no basic RSI-checking script anymore.

This system REMEMBERS when it got burned. It LEARNS from patterns. It makes DECISIONS like a team of traders, not a single algorithm.

You got four personalities arguing it out, each with their own trauma and wins. When they all agree? That's when you GO. When memories scream 'DANGER'? That's when you HOLD.

Now take this foundation and build an empire, motherfucker."

FINAL WORD

You now have:

1. **Production infrastructure** (V3 features)
2. **Cognitive decision-making** (memory + reasoning)
3. **Multi-agent architecture** (democratic voting)
4. **Pattern learning** (episodic memory)
5. **Fractal analysis** (multi-timeframe)

This is the foundation for a real, adaptive trading system.

Start with paper trading. Test extensively. Learn from the memory logs. Refine the personalities. Then, when you're ready, connect it to real capital.

Built by: Daryell McFarland

System: HYDRA SENTINEL-X OMEGA v4.0

Status: Research & Development

License: Use at your own risk

 HYDRA OMEGA - READY FOR DEPLOYMENT 