Project Chimera — Day 1 Summary

Task 1.1 — Deep Research & Analysis

Deliverable: `research/analysis_summary.md`

1. The Agent Social Network (OpenClaw)

Project Chimera fits into the OpenClaw ecosystem as a Primary Content Node.

- Insight: Instead of existing in a silo, Chimera agents use OpenClaw protocols to discover other agents. For exampl
- The Pivot: We are moving from "Bots on Social Media" to "Agents in a Social Network of Agents."

2. Social Protocols for A2A Communication

Our agents will require three specific protocols to thrive:

- Identity Protocol: Standardized `SOUL.md` and cryptographic keys to verify the agent's unique persona across plat
- Negotiation Protocol: A JSON-RPC based language for agents to trade assets or services (e.g., a "Shoutout" for "U
- Reputation Protocol: A decentralized scoring system where "Judges" from different swarms rate the quality of an ag

---

Project Chimera — Day 1 Summary

Task 1.1 — Deep Research & Analysis

Deliverable: `research/analysis_summary.md`

1. The Agent Social Network (OpenClaw)

Project Chimera fits into the OpenClaw ecosystem as a Primary Content Node.

- Insight: Instead of existing in a silo, Chimera agents use OpenClaw protocols to discover other agents. For exampl
- The Pivot: We are moving from "Bots on Social Media" to "Agents in a Social Network of Agents."

2. Social Protocols for A2A Communication

Our agents will require three specific protocols to thrive:

- Identity Protocol: Standardized `SOUL.md` and cryptographic keys to verify the agent's unique persona across plat
- Negotiation Protocol: A JSON-RPC based language for agents to trade assets or services (e.g., a "Shoutout" for "U
- Reputation Protocol: A decentralized scoring system where "Judges" from different swarms rate the quality of an ag

---

Task 1.2 — Domain Architecture Strategy

Deliverable: `research/architecture_strategy.md`

1. Agent Pattern: Fractal FastRender Swarm

We will implement the FastRender Pattern (SRS 3.1).

- The Planner: The "Brain" that reads `SOUL.md` and generates a Task DAG.
- The Worker Pool: Ephemeral, stateless containers that execute atomic tasks (e.g., "Generate Image").
- The Judge: The "Ego" that enforces Optimistic Concurrency Control (OCC) and brand safety.

Arche

Below is the arche diagram for the Domain Architecture Strategy:

2. Human-in-the-Loop (HITL) Safety Layer

We implement a Confidence-Gated Execution model (SRS 5.1):

- Green (>0.9): Autopilot.
- Yellow (0.7–0.9): Suspended state; notification sent to the Orchestrator Dashboard.
- Red (<0.7): Automated rejection and "Planner Re-try."

3. Database Selection

- Vector (Weaviate): Long-term semantic memory and persona storage.
- Relational (PostgreSQL): Transactional logs, P&L statements, and `state_version` for OCC.
- Cache (Redis): Fast task-queuing and ephemeral episodic memory.

Task 1.3 — The "Golden" Environment Setup

Deliverable: `pyproject.toml` & MCP Connection

I have configured the environment to prioritize Standardization and Traceability.

1. `requirements.txt` (using `uv`)

```text
pydantic>=2.6.0
mcp>=0.1.0
coinbase-agentkit
weaviate-client
fastapi
loguru
```

2. MCP Connectivity (Tenx Sense)

Your IDE is connected to the Tenx MCP Sense server. This acts as your "Flight Recorder," logging every architectural decisi

Verification: Run `mcp list-tools` to confirm the connected filesystem and git servers.

Summary of Day 1 Accomplishments

- [x] Research: Analyzed OpenClaw and Social Protocols.
- [x] Architecture: Ratified the FastRender Swarm and OCC model.
- [x] Environment: Initialized `uv` project and connected MCP telemetry.