Project Chimera — SRS Report (Canonical)

Executive Summary

Project Chimera (2026 Edition) defines an autonomous, swarm-based Influencer Network that combines the Model Context Protocol (MCP), a FastRender Swarm, and Agentic Commerce. This report summarizes the SRS, analyzes the Trillion Dollar AI Code Stack, and explains how OpenClaw protocols influenced our openclaw.json integration strategy.

•

Analysis: The "Trillion Dollar AI Code Stack"

•

Core idea: The "Trillion Dollar AI Code Stack" positions AI-native infrastructure (model access, data mesh, tooling, governance) as the backbone for next-gen software. Key layers:

• Model Abstraction Layer: standardized model access (MCP analog).

• Context & Memory Layer: vector DBs, RAG patterns.

• Tooling Layer: MCP Servers wrapping external APIs.

•

Governance Layer: policy enforcement, traceability, audit logs.

•

Relevance to Chimera: Project Chimera maps directly onto this stack:

• MCP provides the Model Abstraction + Tooling layers.

• Weaviate + Redis + Postgres supply the Context & Memory and Transactional layers.

•

CI/spec-driven development implements the Governance layer (specs/, schema checks, .cursor/rules).

•

Strategic implications: Prioritize interface contracts (JSON Schemas, MCP Tool specs) over internal implementations. This reduces drift when swapping model or tool providers and enables safe automation at scale.

•

OpenClaw Influence & openclaw.json integration

•

OpenClaw (Agent Social Network) introduces social protocols for agent interoperability: presence, capability discovery, and capability negotiation.

• How it influenced openclaw.json (design decisions implemented):

• Capability Advertisement: agents publish a small, stable descriptor listing channels, public_tools, and rate_limits so other agents can discover and interoperate.

• Presence & Availability: status fields and heartbeat metadata to avoid ghost participants.

•

Security: signed capability manifests to prevent impersonation; minimal public surface to avoid leaking secrets.

•

Practical integration: openclaw.json acts as a lightweight MCP Server descriptor: it is parsed by the Orchestrator to register agent endpoints and map MCP Tool calls to OpenClaw channels (e.g., social.post → openclaw://agent/<id>/post).

•

Key Recommendations

•

Enforce Spec-Driven Development: Keep specs/ authoritative; fail CI on spec drift.

• Interface-first engineering: Implement JSON Schema for Task and Tool payloads early.

• Governance: Add AGENTS.md for operational roles and .cursor/rules for agent behavior.

•

Agentic Commerce caution: Prototype with test wallets and strict budget controls (CFO Judge pattern) before enabling live transfers.

•

Deliverables in this repo (current)

•

specs/ — functional + technical specs (partial).

• skills/ — skill scaffolds and adapters (db adapter, etc.).

• tests/ — TDD scaffolding and integration tests (Postgres integration implemented).

•

docs/Project_Chimera_SRS_Report.md — this canonical report (Markdown).

•

Submission & PDF Export Instructions

To produce a final PDF suitable for upload to Google Drive, run one of the following locally (recommended: pandoc):

• Using pandoc:

pandoc docs/Project_Chimera_SRS_Report.md -o docs/Project_Chimera_SRS_Report.pdf --pdf-engine=xelatex

• Using Python (headless, no LaTeX):

python -m pip install markdown weasyprint

python - <<'PY'

import markdown, weasyprint

html = markdown.markdown(open('docs/Project_Chimera_SRS_Report.md').read())

weasyprint.HTML(string=html).write_pdf('docs/Project_Chimera_SRS_Report.pdf')

PY

Upload the resulting docs/Project_Chimera_SRS_Report.pdf to Google Drive and set Sharing to "Anyone with the link can view." Submit the Drive "file link" (not the folder link).

•

Notes on Submission Statement (what to include in the Drive doc metadata)

•

Title: "Project Chimera — SRS Report (FDE Submission)"

• Description: include the repo URL, CI status badge, and a short bullet list of included artifacts (specs/, tests/, Dockerfile, Makefile, .cursor/rules).

• Accessibility: Confirm sharing set to "Anyone with the link can view" and include the link in your submission form.

If you want, I can convert this Markdown to PDF inside the container and create docs/Project_Chimera_SRS_Report.pdf for you — shall I proceed to generate the PDF in-repo now? (This will run the pandoc or weasyprint command inside the repo container.)