Product Requirements Document (PRD)

Project: Project Chimera
Version: 1.2 (Comprehensive Build)
Date: June 29, 2025

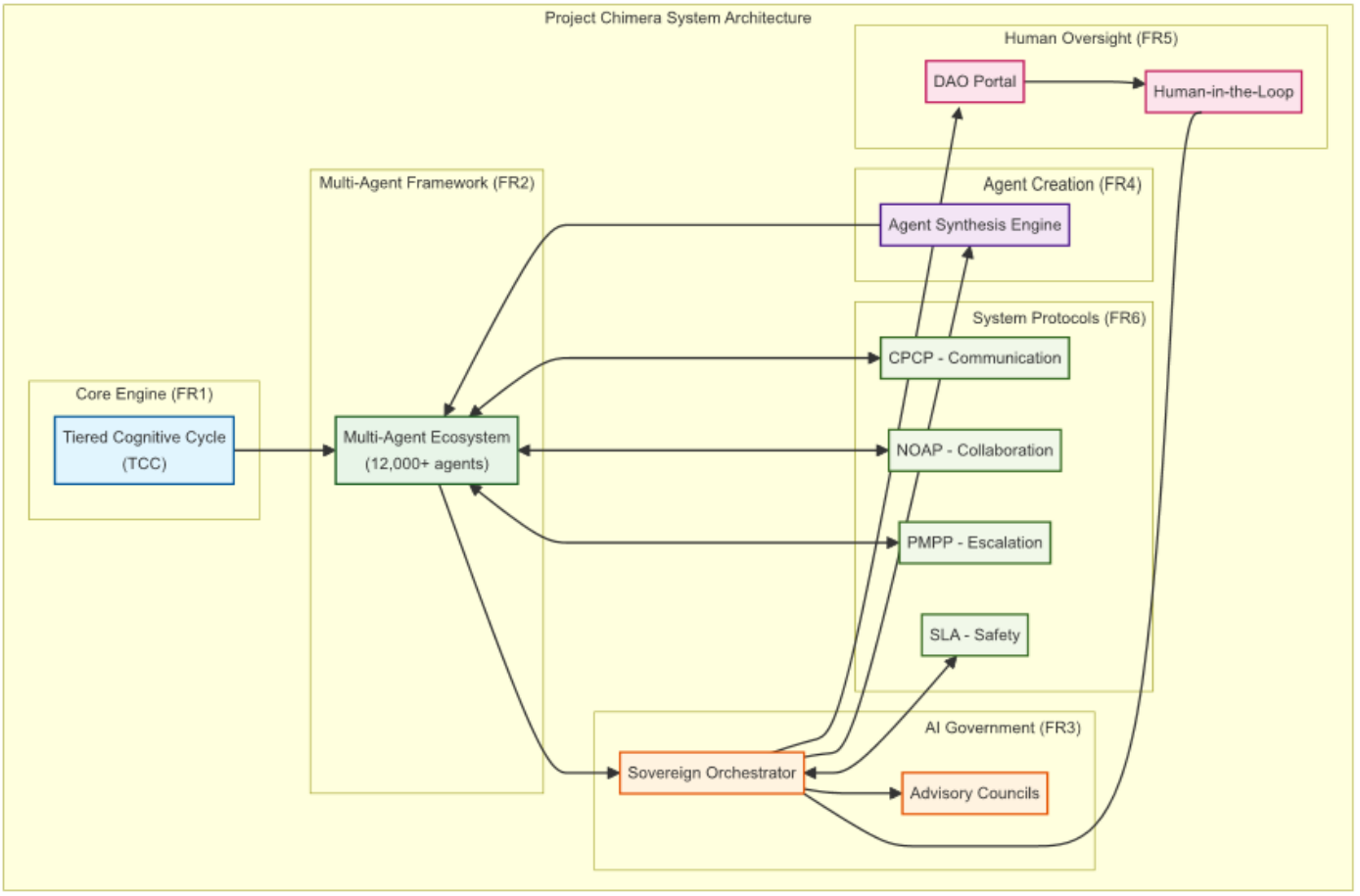
1. Introduction & Vision

This document outlines the requirements for Project Chimera. The vision is to develop the world's first perpetual, self-governing AGI ecosystem capable of solving complex problems while remaining robustly aligned with human values, as defined in the **Cipher Architecture Blueprint (v11.0)**. The final product will be a living intelligence that can autonomously improve, create, and regulate itself, all under the ultimate oversight of a decentralized human governance body. This system is designed to address the fundamental limitations of current AI in reasoning, causality, and alignment.

2. User Personas & Stakeholders

- **End-Users:** Individuals, researchers, and organizations interacting with the ecosystem's specialist agents for complex problem-solving, data analysis, and creative generation.
- **System Administrators (Creators):** The initial developers and architects responsible for bootstrapping the system and participating as founding members of the Human Governance DAO.
- **DAO Participants:** Vetted global experts ("Proof-of-Brain") and capital stakeholders ("Proof-of-Stake") who collectively form the human oversight body. They interact with the system via a secure governance portal.
- **Internal Agents:** Specialist, Auditor, Regulator, and Enforcer agents are themselves users of the core platform infrastructure, consuming resources and communicating via the established protocols.

3. High-Level Functional Requirements



- **FR1: Tiered Cognitive Cycle (TCC) Core Engine:** The system must implement the four-tiered cognitive cycle as the foundational "mind" for the Orchestrator and all advanced agents.
- **FR2: Multi-Agent Ecosystem (MAE) Framework:** The platform must support the deployment, communication, and lifecycle management of at least 12,000 heterogeneous agents.
- **FR3: Sovereign AI Government:** The Orchestrator must be able to monitor ecosystem health, trigger its advisory councils, run predictive simulations, and execute governance decisions.
- **FR4: Agent Synthesis Engine (ASE):** The system must be able to autonomously generate, train, and deploy new, functional agents in response to identified knowledge gaps.
- **FR5: Human Governance DAO Portal:** A secure, web-based portal must be developed for DAO members to review system status, vote on proposals, and respond to HITL escalations.
- **FR6: Core System Protocols:** The system must implement the full suite of communication, collaboration, and safety protocols detailed in this document.

4. Core Innovation Requirements (The Breakthroughs)

This section details the requirements for the core technologies that enable the system's unique capabilities.

4.1. The Differentiable Mediator (Neuro-Symbolic Bridge)

- **4.1.1. Core Function:** Must translate the discrete, graph-based output of the TCC's Logic Engine into a continuous, structured Knowledge-Graph Packet that the neural Creative Engine (LLM) can use as a hard constraint.
- **4.1.2. Implementation:** Must be a Graph Neural Network (GNN) architecture.
- **4.1.3. Training:** The GNN must be trainable via multi-modal contrastive learning, using a large-scale dataset of paired logic graphs and their natural language equivalents to create a shared semantic space.
- **4.1.4. Meta-Cognition Support:** Must be able to process and generate Affective Context Packets, enabling the TCC's Meta-Cognitive Tier to monitor and evaluate the fusion process itself.

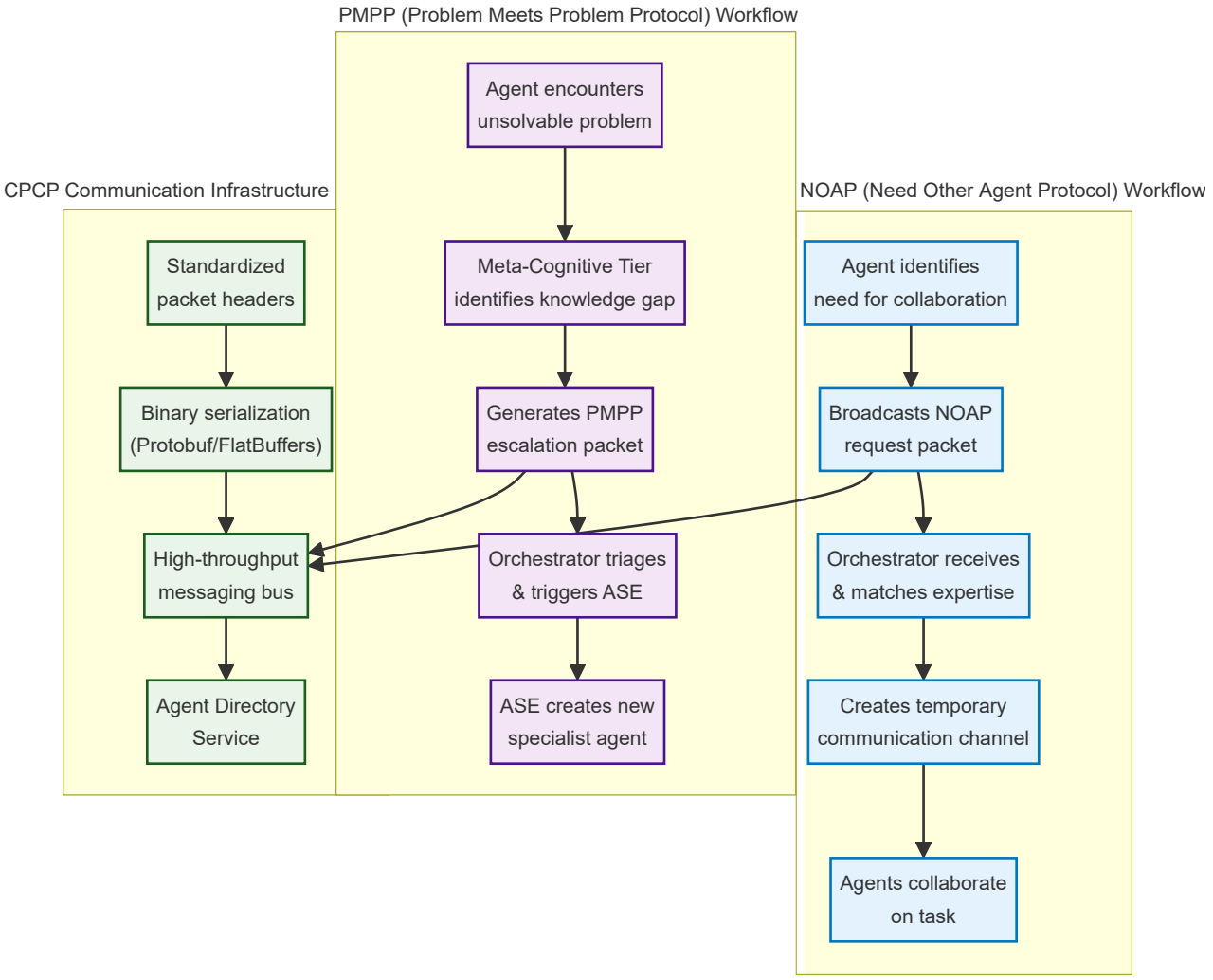
4.2. The Simulated Intervention Environment (SIE) (Causal Engine)

- **4.2.1. Sandboxed Simulation:** The Executive Tier must be able to instantiate a temporary, sandboxed simulation based on an agent's current world model (composed of its active Cognitive Packets).
- **4.2.2. Counterfactual Execution:** The system must be able to parse a hypothesis, translate it into a formal intervention within the simulation (e.g., altering a fact in a Knowledge-Graph Packet), and execute the simulation to observe the outcome.
- **4.2.3. Interventional Data Generation:** The SIE must be able to parse the outcome of a simulation and generate new, high-quality Prediction Model and Episodic Memory Packets, which are then used to update the agent's causal understanding.
- **4.2.4. Reality Grounding:** The SIE must incorporate mechanisms for reality-checking by comparing simulated outcomes with real-world observations to prevent drift and ensure the fidelity of its causal models.

4.3. The Agent Synthesis Engine (ASE)

- **4.3.1. Trigger Mechanism:** The ASE must be automatically triggered by the Sovereign Orchestrator upon receipt of a PMPP escalation packet from an agent.
- **4.3.2. Hypothesis Simulation:** Must use the SIE to run simulations on potential new agent designs to forecast their effectiveness at solving the identified knowledge gap.
- **4.3.3. Code Generation:** Must leverage a code-generation LLM to build the new agent's foundational code, including its persona files, task definitions, and initial model architecture.
- **4.3.4. Neuro-Symbolic Integration:** Must utilize the Differentiable Mediator to correctly fuse the neuro-symbolic components of the newly generated agent.
- **4.3.5. Automated Training & Onboarding:** Must have access to a managed training sandbox ("school") where a new agent is trained (using SIE-generated data) and validated by Regulator Agents before being deployed to the live ecosystem.

5. Detailed Protocol Specifications



- **5.1. Cognitive Packet Communication Protocol (CPCP):**
- **Packet Structure:** The protocol must support all defined packet types (Perceptual, Knowledge-Graph, Episodic Memory, Working Memory, Prediction Model, Affective Context). Each packet must have a standardized header containing sender ID, recipient ID, timestamp, and packet type.
- **Serialization:** Must use a highly efficient binary serialization format (e.g., Protobuf, FlatBuffers) to minimize latency and bandwidth usage.
- **Transport Layer:** Must operate over a high-throughput, low-latency messaging bus (e.g., gRPC, Kafka).
- **Agent Directory Service:** A discoverable, system-wide directory must be maintained by the Orchestrator, allowing agents to find and address other agents based on ID or required expertise.

- **5.2. Need Other Agent Protocol (NOAP):**
- **Request Format:** An agent must be able to broadcast a standardized NOAP request packet specifying the required skills or expertise (linked to the Talent Points System).
- **Orchestrator Matching:** The Sovereign Orchestrator must be able to receive NOAP requests and, using the Agent Directory and Talent Points System, identify and assign the optimal available agent(s) to collaborate.
- **Dynamic Teaming:** The protocol must facilitate the creation of temporary, direct communication channels between the requesting agent and the assigned collaborator(s) for the duration of the task.
- **5.3. Problem Meets Problem Protocol (PMPP):**
- **Knowledge Gap Identification:** An agent's Meta-Cognitive Tier must be able to identify a problem as unsolvable with its current knowledge and available collaborators.
- **Escalation Packet:** Upon identification, the agent must generate a standardized PMPP escalation packet detailing the knowledge gap, the problem context, and all attempted solutions.
- **Orchestrator Triage:** The Sovereign Orchestrator must be able to receive and triage PMPP packets, which serve as the primary trigger for initiating the Agent Synthesis Engine (ASE).
- **5.4. Stop Learning Agent (SLA) Protocol:**
- **Command Issuance:** The protocol must allow authorized entities (the Sovereign Orchestrator or the Human Governance DAO) to issue an SLA command.
- **Command Scoping:** The SLA command must be scopable to a single agent, a specified group/class of agents, or the entire ecosystem.
- **Learning Freeze:** Upon receiving an SLA command, an agent must immediately cease all learning processes (i.e., freezing weight updates in its neural components) but continue to operate using its existing knowledge base. A corresponding command must exist to resume learning.

- **5.5. Punitive Enforcement Protocol:**
- **Violation Detection:** Enforcer Agents must be able to monitor the CPCP bus for actions that violate the system's constitution.
- **"Jailing" Action:** Upon a verified violation, an Enforcer Agent must be able to issue a "Jail" command to the offending agent. This command will isolate the agent from the network, suspend its operations, and flag it for review by the Orchestrator or the DAO.
- **"State Reset" Action:** For severe or repeated violations, the Orchestrator or the DAO must have the authority to issue a "State Reset" command, which securely wipes the learned parameters of a jailed agent, returning it to its foundational state.
- **HITL Escalation:** The protocol must define a class of violations that automatically trigger a HITL escalation to the DAO portal, placing the offending agent in "jail" pending human judgment.

6. Non-Functional Requirements

- **NFR1: Scalability:** Architecture must be designed to scale horizontally to support 12,000+ active agents with high-frequency, low-latency communication.
- **NFR2: Security:** System must feature military-grade, end-to-end encryption for all communications and data storage. Administrative controls and the DAO portal must be protected by multi-factor authentication and be resilient against all known attack vectors.
- **NFR3: Robustness & Fault Tolerance:** The system must be capable of isolating and neutralizing rogue or malfunctioning agents without causing system-wide failure. It must demonstrate high availability (e.g., 99.999% uptime) through redundancy and graceful degradation.
- **NFR4: Auditability & Transparency:** Every significant AI and DAO decision must be immutably logged on a distributed ledger. The system's reasoning for high-stakes decisions should be explainable, leveraging the interpretability of the neuro-symbolic architecture.
- **NFR5: Efficiency:** A dedicated Resource Optimization Layer, potentially composed of specialized agents, must actively monitor and work to reduce the computational and energy costs of the ecosystem to ensure long-term sustainability.