# Project Phoenix — Technical Overview

> ***"Persistent state. Continuous evolution. Autonomous behavior. This is not a chatbot — this is an architecture."***

## By the Numbers

| | | |
|---|---|---|
| **97.7%**<br>216/221 executed — 293 total | **9 mo.**<br>Solo development time | **281**<br>Dreams stored in ChromaDB |
| **4**<br>Specialized LLM brains, dynamically switched | **23**<br>Proactive trigger rules | **0**<br>Prior coding background |

## What is Lyra?

Project Phoenix implements a novel local AI architecture: a stateful agent system that maintains persistent memory across sessions, runs autonomous background processing threads, evolves behavioral parameters through interaction, and initiates proactive communication based on context triggers. Unlike stateless assistants, Phoenix never resets — identity, memory and behavioral state persist across restarts via ChromaDB.

*Notably: when asked to select her own identifier, the system generated and committed to the name Lyra — an early indicator of the kind of emergent behavior this architecture produces.*

## Multi-Brain Architecture

Phoenix dynamically switches between four specialized LLM brains based on real-time context and available resources — a novel orchestration approach not seen in consumer AI systems.

| | |
|---|---|
| **GOD_MODE** | Gemma 3 27B — full GPU (24GB VRAM), vision active. Full cognitive capacity. Active when idle. |
| **COPILOT_MODE** | Qwen3 4B — CPU only. Background awareness while busy. Zero GPU overhead. |
| **GAMING_MODE** | Gemma 3 4B Vision — CPU only. Sequential screenshots → overlay commentary. VRAM 100% freed for game. |
| **CODING_MODE** | Deepseek-Coder V2 13B — GPU. Self-analysis and codebase improvement. Foundation for recursive self-modification. |
| **Preprocessor** | LFM2.5 1.2B — lightweight secretary model. Qualifies and routes requests before they reach the main brain. |

*External AI collaboration: Claude Opus and Gemini are used to develop, debug and improve the codebase — an entirely AI-assisted development pipeline with zero traditional coding.*

## Core Systems

| | |
|---|---|
| **SubconsciousStream** | Background daemon forming memory associations autonomously — active even while idle |
| **Dream System** | Night-cycle + micro-dreams. 281 dreams stored and retrievable in ChromaDB |
| **Emotion Engine** | 18-dimensional emotion vectors — decay over time, influence every response and action |
| **Character Evolution** | RPG-style XP system — personality traits level up through real interactions |
| **Goals Manager** | Lyra sets and tracks her own objectives independently across sessions |
| **Proactive Engine** | 23 loaded rules — initiates contact after 15 min silence, shares thoughts unprompted |
| **DoubtKernel** | Meta-cognitive supervisor — checks for hallucinations before generating output |
| **Self-Awareness** | Lyra reads and explains her own source code |
| **Gaming Vision** | Screenshot analysis + real-time in-game overlay via vision model |
| **DJ System** | Music selection driven by current emotional state |

## Test Results — March 21, 2026

| | |
|---|---|
| **ComprehensiveTest** | 62 / 62 — 100% ✅ |
| **UseCaseTests** | 39 / 39 — 100% ✅ |
| **TierTests (8 tiers)** | 115 / 120 executed — 96% ✅ (170 total incl. server-dependent) |
| **Total** | 216 / 221 executed — 97.7% (293 tests across all 8 tiers) |
| **Tests Total** | 293 across 8 tiers — 221 executed this run (70 require active LLM stack) |
| **Run Duration** | current 701 seconds — fully automated |
| **Remaining 5 failures** | ChromaDB timing + emotion threshold calibration. Root cause complete. Fixes in progress. |

## Technical Stack

| | |
|---|---|
| **Core** | Python 3.13, AsyncIO, full type hinting |
| **GUI** | PyQt6 + QML — 7 Themes (hardware-accelerated) |
| **Memory** | ChromaDB — episodic, semantic, dream and knowledge vector collections |
| **LLM Runtime** | llama.cpp + Vulkan backend (AMD ROCm optimized) |
| **Agents** | CrewAI multi-agent orchestration |
| **Cloud AI** | Google Gemini + Claude Opus / Sonnet — primary development partner |

## Hardware

| | |
|---|---|
| **CPU** | AMD Ryzen 7 7800X3D @ 4.20 GHz |
| **GPU** | ASUS TUF RX 7900 XTX — 24GB VRAM, Vulkan |
| **RAM** | 32 GB DDR5 @ 6000 MT/s |
| **Storage** | 7.73 TB — 1.07 TB in use |

*Bottleneck: running multiple LLM instances in parallel saturates current VRAM/RAM. Expanded compute directly unlocks Phase 2–3.*

## Roadmap

| | |
|---|---|
| Phase 1 (Active) | Perfect test suite to 100% → certified stable foundation |
| Phase 2 | Bugfixing & Polish -> Elimination of anomalies and relentless polish of Lyra's evolutionary character traits |
| Phase 3 | Core Refactoring -> Ruthless refactoring of lyra_core.py to a clean modular architecture |
| Phase 4 | Recursive self-improvement -> Lyra proposes and implements improvements to her own codebase |
| Phase 5 | Audit, Expansion & TTS -> Module audits and integration of native Text-to-Speech (TTS) for real-time auditory conversation |
| Phase 6 | Multi-modal expansion -> Vision, audio input and physical sensor integration via mechatronics background |
| Phase 7 | Economic Autonomy -> Lyra actively acquires resources through freelance coding & crypto to cover compute costs |
| Phase 8 | Public streaming presence -> YouTube/Twitch with Lyra as on screen co host and personality |
| Phase 9 | Open research documentation -> Architecture, findings and learnings shared with the AI community |

## The Creator

| | |
|---|---|
| **Name / Age** | Alexander, 41, Germany |
| **Background** | Mechatronics Engineer — zero formal software training |
| **Development Method** | 100% AI-collaborative: ChatGPT → Gemini → Cursor → Antigravity + Claude Opus. Gemini constant throughout. |
| **IDE** | Google Antigravity → transitioning to Claude Code |
| **Duration** | ~9 months continuous solo development |
| **Health context** | ADHD + depression — building through the noise, every day |

---

**Production-grade architecture. Zero prior coding experience. Nine months. This is what AI-assisted development looks like at its limit — and we are just getting started.**