CODEX Architecture Overview

CODEX Architecture Overview

Overview

The CODEX system is the symbolic runtime engine and execution infrastructure for AION's intelligence. It acts as a symbolic interpreter, executor, and orchestration layer that enables glyph-based scroll execution, CodexLang parsing, mutation, memory integration, and future symbolic hardware acceleration.

CODEX bridges the high-level cognitive language of AION (CodexLang) with runtime instructions and operator-level execution. It is modular, extensible, and integrates with key systems like Tessaris (recursive thought logic), GlyphOS (runtime container), and CodexCore (symbolic CPU).

Core Objectives

- Interpret and execute CodexLang scrolls
- Manage symbolic execution registers and trace
- Interface with mutation, memory, and dreams
- Provide a virtual symbolic runtime CPU (CodexCore)
- Enable scroll mutation and optimization
- Provide UI feedback (HUD, scroll runner, debug trace)
- Prepare for native symbolic hardware deployment

Key Components

1. codex_executor.py

- Executes parsed CodexLang instruction trees
- Maintains register state (R0, R1, ..., FLAGS)
- Supports operators: ⊕, →, ↔, ⟲, □, etc.
- Emits runtime trace and register snapshots
- Hooks into memory, dream, DNA, and container layers

2. codex_emulator.py

- Emulates virtual symbolic CPU (CodexCore)

CODEX Architecture Overview

- Loads and executes scrolls as instruction sets
- Can simulate cost, energy, or time-based execution

3. codex_scroll.py
- FastAPI endpoint to receive scrolls
- Parses CodexLang, executes, returns result, trace, and registers
- Accepts optional JSON context for variable injection

4. codex_scroll_runner.tsx
- Frontend UI for scroll input, mutation, trace viewing
- Connected to CodexLang editor and register debugger
- Supports context injection, mutation, and execution feedback

5. codex_mutation_engine.py
- Applies semantic-preserving mutations to scrolls
- Enables self-rewriting and evolution of logic trees
- Will integrate with Tessaris recursive phase

6. codex_websocket_interface.py
- Streams live register updates, execution status
- Powers CodexHUD and remote trigger visualization

System Integration
- Tessaris Engine: Codex scrolls can embed Tessaris operators, and Tessaris may generate scrolls dynamically
- GlyphOS: Execution occurs inside `.dc` containers governed by GlyphOS
- MemoryBridge: Register state and output can be stored as symbolic memory
- DreamCore: Scrolls can trigger dream generation or be generated by dream outputs
- LuxNet/GIP: CodexLang scrolls may be transmitted as `.gip` packets in the future

Current Features
- CodexLang scroll parsing and execution

CODEX Architecture Overview

- Symbolic CPU register logic
- Execution trace with before/after registers
- Context variable injection
- Scroll mutation engine
- WebSocket feedback for HUD
- Frontend scroll runner UI
- Scroll preset loader and history

Future Features
- Register-by-register debug stepping
- Scroll saving, versioning, diff viewer (CodexDiffViewer)
- Full Tessaris-aware mutation loop
- Symbolic cost prediction and runtime scoring
- Export scrolls in `.codex` format
- Symbolic tracing of instruction flow
- Dream-sourced scroll generation
- Codex hardware acceleration via FPGA/ASIC

Folder & File Structure

```
backend/
  modules/
    codex/
      codex_executor.py
      codex_emulator.py
      codex_mutation_engine.py
      codex_scroll_history.py
      codex_context_adapter.py
      codex_metrics.py
      codex_feedback_loop.py
  routes/
    codex_scroll.py
    codex_ws.py
```

CODEX Architecture Overview

frontend/

components/Codex/

CodexScrollRunner.tsx

CodexLangEditor.tsx

CodexRegisterViewer.tsx

CodexScrollPresets.tsx

CodexRegisterDebug.tsx

CodexScrollHistory.tsx

CodexHUD.tsx

Summary

The Codex system is the symbolic brainstem of AION’s scroll execution infrastructure. It transforms declarative CodexLang into real-time symbolic reasoning and lays the foundation for mutation, introspection, and hardware-based symbolic intelligence. With CodexCore emulation now complete, Codex is ready to scale across self-evolving scrolls, live HUD feedback, and intelligent scroll propagation across containers and LuxNet.