PHASE 19 AND PHASE 20 EXECUTION SUMMARY

PROJECT CONTEXT This document summarizes the decisions, constraints, and execution model established in this chat for the MCP Spirit Enterprise LLC Tier 1 project. It is intended to carry forward operational continuity into subsequent chat windows.

CORE OBJECTIVE The project is pursuing full automation via SDK agents, with Phase 19 focused on hardening identity, provenance, atomic writes, idempotency, and auditability prior to introducing autonomous agents in Phase 20.

AGENT MODEL CLARIFICATION 1. Custom GPT Agents - Serve as interactive assistants only - Can read, analyze, generate scripts, validate outputs, and walk checklists - Require explicit manual approval for any read or write - Do NOT possess system identity - Do NOT perform autonomous execution - Are not considered system principals

1. SDK Agents (Phase 20)
2. Are first-class system principals
3. Have immutable AgentIDs
4. Have enforced write scopes
5. Can execute autonomously
6. Produce auditable artifacts
7. Identity precedes intelligence

Phase 19 explicitly DOES NOT require building SDK agents. It requires locking identity and execution boundaries first.

PHASE 19 AGENT REQUIREMENTS Phase 19 requires Agent Identity Materialization, not agent intelligence.

Required: - Canonical agent roster definition - Immutable AgentIDs - Defined write scopes - Workflow binding to AgentIDs - Blocking anonymous execution

Not Required in Phase 19: - SDK logic implementation - Autonomous tasking - Agent-to-agent coordination

---

CUSTOM GPT ROLE IN PHASE 19 Custom GPT agents are authorized to: - Assist in executing the Phase 19 checklist - Generate scripts, configs, schemas, and test vectors - Parse outputs and validate pass/fail - Identify missing data or failures

Custom GPT agents are NOT authorized to: - Perform final production writes - Own AgentIDs - Bypass idempotency or atomic-write enforcement

Human-in-the-loop approval is mandatory for all execution during Phase 19.

CODEX (C-O-D-E-X) USAGE CLARIFICATION Codex refers to OpenAI's coding model and CLI tooling, not media codecs.

Codex Characteristics: - Runs in a separate execution environment from custom GPT - Acts as a code generation and scripting assistant - Can prepare scripts and tooling that interact with NAS paths - Does not inherit identity or authority

Codex does NOT: - Act autonomously - Possess AgentIDs - Perform unapproved execution

---

CODEX CLI DEPLOYMENT DECISION Codex CLI on GSA-1000 (Windows 11 Pro) was selected as the preferred operational surface because: - It runs locally under the user's Windows identity - It can access mapped NAS drives - It supports script generation, testing, and validation - It aligns with Phase 19 manual approval constraints

Codex CLI Setup Summary: - Install via npm: npm install -g @openai/codex - Authenticate via codex auth login - Create a dedicated Phase 19 workspace - Initialize Git for auditability - Enforce operational rules via CODEX_RULES.md

Codex is used as a toolsmith, not an operator.

PHASE 19 EXECUTION MODEL Correct execution pattern: 1. Custom GPT manages checklist and reasoning 2. Codex generates scripts and tooling 3. Human reviews and approves 4. Execution occurs manually or via n8n under controlled identity 5. Results are validated and logged

Optional allowance: - A temporary PHASE19_BOOTSTRAP AgentID may be created for commissioning only - Heavily scoped - Disabled at Phase 19 close

---

PHASE 20 DEPENDENCIES Phase 20 assumes the following are complete: - Identity enforcement - Atomic writes - Idempotency - Continuity logging - Codec/schema enforcement (if present)

Phase 20 introduces: - SDK agent implementation - Persistent agent state - Autonomous execution - Agent-to-agent coordination

---

BOTTOM LINE Phase 19 establishes trust, structure, and auditability. Custom GPT and Codex are assistants. SDK agents are the only autonomous actors. Identity is locked before intelligence is introduced.

This document should be treated as authoritative Phase 19-20 context for subsequent project work.