MCP Phase 19–20 Context and Decisions Summary

Purpose

This document summarizes the key conclusions, decisions, and guidance from the current chat regarding ChatGPT project architecture, NAS file organization, and RAG/DB strategy. It is intended as a handoff reference for continuation of Phase 19 (Hardening) and Phase 20 (NAS Database Integration) work within the **MCP for Spear Enterprise LLC – Tier 1** project.

---

1. ChatGPT Project Architecture (Context)

- ChatGPT **Projects are flat containers**. There is **no native support** for:
- Subfolders
- Nested projects

- Parent/child project hierarchies

- A "Portfolio Project" is **not a special feature**. It is simply:

- A newly created project
- Run with stricter governance rules (index, decisions, summaries only)

- Optionally configured with **project-only memory** (must be set at creation time or via sharing)

- Because nesting is unavailable and no public OpenAI roadmap exists for nested projects, **external structure must be enforced outside ChatGPT**.

Conclusion: ChatGPT Projects are UI workspaces only. The real hierarchy and memory discipline must live on the NAS and in the Phase 20 database.

---

2. Portfolio Pattern (Operating Model)

The adopted model is a **Portfolio Pattern**, which simulates nesting via discipline:

- One top-level **Portfolio / Boardroom project**

- Index, decision log, risk register only

- Separate **Program Hub projects** (Heber Campus, MCP, OSY, etc.)

- Separate **Workstream projects** for active iteration

Critical rule: - No raw thinking in the Portfolio project - Only promoted, decision-grade outputs move upward

This pattern remains valid even if ChatGPT later adds nesting.

---

3. Tier 1 NAS Structure (Authoritative Source)

Your Tier 1 NAS structure defines `/SE_T1/` as the **single immutable root of truth**.

Key points from governance documents: - MCP is the **infrastructure backbone**, not a content project - ATB-approved artifacts are mirrored into `/SE_T1/GOVERNANCE/` - Lifecycle separation is expected (WORKING / FINAL / ARCHIVE)

Implication: - File organization and truth ownership are enforced at the NAS level - ChatGPT projects reference this structure, not replace it

---

4. Phase 19 Hardening – Required Controls

Phase 19 must lock governance before Phase 20 ingestion.

Required by end of Phase 19:

1. **Lifecycle Lanes (Non-Negotiable)**
2. `SOURCE` – raw imports, vendor PDFs, dumps
3. `WORKING` – drafts, in-flight analysis
4. `FINAL` – decision-grade, authoritative

5. `ARCHIVE` – superseded finals

6. **Retrieval Priority Rule (for Agents / RAG)**

7. FINAL first
8. WORKING only if FINAL has no hits

9. SOURCE as evidence only, not narrative truth

10. **Naming & Revision Enforcement (on promotion only)**

11. ASCII only
12. Explicit REV-A / REV-B

13. Do not rename legacy files during import

14. **Continuity Registry Discipline**

15. All ingest, promotion, and supersession events append to registry/log

---

5. Phase 20 – NAS Database Strategy

Purpose of the DB

The Phase 20 DB is the **real hierarchy** and **anti-fragmentation layer**: - Stable IDs - Version control - Relationship mapping - Deterministic retrieval for agents

Folders are for humans. The DB is for machines.

DB Placement (per Tier 1 governance)

Place DB infrastructure under MCP:

- `/SE_T1/INFRASTRUCTURE_CORE/MCP/DB/`
- `index/` (SQLite initially)
- `records/` (JSON metadata per artifact)
- `objects/` (content-addressed storage by hash, if used)

- `ingest/` (atomic drop zone)

- Continuity and audit logs under:

- `/SE_T1/INFRASTRUCTURE_CORE/CONTINUITY_REGISTRY/`

Content remains in its owning branches (SDC_COMMS, ENERGY_DIVISION, etc.).

---

6. File Migration & Placement Strategy (500 Files)

Key Decision

Do not reorganize first. Mirror first.

Recommended Steps:

1. Mirror-copy existing `SDC-COMMS` content into:
2. `/SE_T1/SPECIAL_PROJECTS/SDC_COMMS/SOURCE/RAW_IMPORT/<YYYYMMDD>/...`

3. Generate a manifest:

4. Full path
5. File type

6. Modified time
7. Size

8. SHA-256 hash

9. Deduplicate by hash

10. One canonical candidate per duplicate group

11. Classify by lifecycle (SOURCE / WORKING / FINAL)

12. Stand up DB records pointing to existing paths (no renames)

13. Promote only top-value authoritative docs (30–50) into `FINAL/`

---

7. Canonical Artifact Model (Anti-RAG Fragmentation)

Each artifact must have a **stable ID**, independent of filename or folder.

Minimum DB fields: - `artifact_id` - `title` - `owning_branch` - `status` (source | working | final | archived | governance) - `revision` - `sha256` - `canonical_path` - `source_path` - `supersedes` - `tags`

Hard rule: - Only one current FINAL per `artifact_id`

This rule alone eliminates most RAG hallucination and contradiction issues.

---

8. Domain-Based Organization (Not Project-Based)

For PUBLISHED/FINAL content, organize by **domain**, not by project:

- GOVERNANCE
- ARCHITECTURE
- SECURITY
- OPERATIONS
- ENGINEERING (POWER, H2, DATA-CENTER)
- FINANCE
- LEGAL

Projects are expressed via DB tags, not folder depth.

---

9. Bottom-Line Guidance for Next Chat

- Phase 19 is about **locking rules**, not moving files
- Phase 20 DB is the **real hierarchy and memory system**
- RAG quality depends on lifecycle separation, not chunking tricks
- Only FINAL + GOVERNANCE should be default retrieval truth
- The Tier 1 NAS structure already supports this—do not fight it

---

Open Items for Next Chat

- Finalize Phase 19 hardening checklist (explicit gates)
- Define Phase 20 DB v0 schema (SQLite)
- Decide initial promotion set (top 30–50 artifacts)
- Define agent retrieval policy in MCP (scope + ranking)

End of Summary