GovSight: July 24 Design Status

This document summarizes the current design and implementation status of the GovSight project as of July 24, 2025. It includes the strategic design goals, current implementation audit, memory architecture evaluation, retrieval pipeline status, and a feature-by-feature completion report. This document is intended to serve as a checkpoint and re-priming artifact for future development sessions.

# 1. North Star Vision & Strategic Goals

GovSight is a persistent, domain-tuned AI teammate for lobbying. It remembers clients, monitors legislation, and generates actionable deliverables such as board briefs, legislative updates, and funding eligibility summaries. The system is designed to operate with the intelligence and recall of a research team, allowing one person to scale their impact.

Core Goals:

* • Lobbying Intelligence Brain
* • Push-Button Client-Ready Report Generation

# 2. Architecture Overview

GovSight is composed of three retrieval layers and a persistent memory system:

* • Local Storage (SQLite/File Archive): for structured data and version tracking
* • Pinecone Vector Search: for semantic recall
* • Web Retrieval Layer: for real-time fact discovery when memory layers return nothing

# 3. Codebase Audit Summary

Modules reviewed:

* • talk.py — Legacy procedural engine still active
* • chat\_cli.py — Shim CLI to phase out talk.py
* • govsight.memory.\* — Modular memory system (R1)
* • memory\_manager.py — Legacy GPT-based fact extraction and session tracking
* • migrate\_memory\_db.py — Handles legacy-to-R1 schema migration
* • serp\_client.py & web\_reasoner.py — Partial or missing web fallback logic
* • smoke\_memory\_test.py — Present but unverified coverage

# 4. Memory System Refactor Status

The new modular memory API under govsight.memory is cleanly designed and mostly implemented. It includes:

* • Memory class for sessions, facts, files
* • Dataclasses for structured record handling
* • Schema bootstrap and version control logic

Migration blocker: missing column `subject\_slug` in legacy DB is handled in migrate\_memory\_db.py but row migration is incomplete.

# 5. Feature Implementation Audit (Design vs. Reality)

See the attached table for a detailed feature audit across all major modules:

|  |  |
| --- | --- |
| Feature | Status |
| Persistent memory API (sessions, messages, facts) | ✅ Implemented in govsight.memory.memory |
| File logging and source tracking | ✅ Implemented via register\_file in memory.py |
| Slug generation for entity indexing | ✅ Present in memory.py |
| Fact extraction using GPT | ✅ Implemented in memory\_manager.py, also referenced in talk.py |
| Conversation buffer and context merging | ⚠️ Mentioned in talk.py logic, but not modularized |
| Watchlist creation and tracking | ✅ Present in memory\_manager.py |
| Semantic retrieval with Pinecone | ✅ Active in talk.py |
| Web fallback using SerpAPI | ❌ Not present in talk.py or serp\_client.py (may be partially moved) |
| Structured SQLite schema with migration | ✅ Present, pending full legacy fact migration |
| Modular CLI entrypoint | ✅ Implemented in chat\_cli.py |
| Legacy talk.py still running engine | ✅ True, via shim |
| Inline prompts and rules | ⚠️ Still embedded in legacy memory\_manager.py |
| Web UI for future use | 🚧 Placeholder only |
| Cascade logic (DB → Pinecone → Web) | ⚠️ Pinecone works; DB logic indirect; Web missing |
| Tests / validation scripts | ⚠️ Present (smoke\_memory\_test.py), coverage unknown |
| Cost/logging dashboard | ❌ Not implemented |
| Inline citation formatting | ❌ Not implemented |
| Long-doc summarization | ❌ Not implemented |
| Trust/recency score tuning | ❌ Not implemented |