Week 2 Final Report (Alignment - Take 7)

Date: 2026-03-01

Repository: https://github.com/GrimVad3r/automaton-auditor

Primary Model: mistralai/ministral-3-14b-reasoning via LM Studio
(OPENAI_BASE_URL=http://127.0.0.1:1234/v1, JSON mode)

1. Executive Summary

- Scope & Purpose: Autonomous audit graph (detectives + judges + Chief Justice) for code and PDF inputs.
- Outcome: Latest self-audit scored 12/20 (current run). Prior runs with same code yielded 14/20; variance driven by docs criterion.
- Key Takeaways: Security cleared via sandboxed git; parallel graph implemented; docs criterion penalized for perceived uncited claims; Vision disabled.
- Next Steps: Tighten PDF evidence citations; add recovery edges; decide on Vision enablement.

2. Architecture Deep Dive & Design Rationale

- Parallel fan-out/fan-in: src/core/graph.py (initialize -> repo_investigator, doc_analyst; aggregate_evidence -> prosecutor, defense, tech_lead; handle_error -> chief_justice -> finalize).
- Chief Justice synthesis: src/agents/justice/chief_justice.py with weighted synthesis and variance caps.
- Grounded outputs: StructuredOpinion + _coerce_structured_response in src/agents/judges/base_judge.py; force_json_mode for LM Studio.
- Personas: distinct prompts in src/agents/judges/prosecutor.py ("Trust No One"), defense.py ("Reward Effort"), tech_lead.py ("Does it actually work").
- Security: sandboxed git in src/tools/git_tools.py using RepositorySandbox.clone_repository; no raw os.system.
- Design rationale: Pydantic models in src/core/state.py for validation; deterministic synthesis rules for repeatability; VisionInspector remains disabled (not implemented for production).

Graph Flow

```
flowchart TD
    Start([START]) --> Init[Initialize]
    Init -->|fan-out| Repo[RepoInvestigator
(sandboxed git)]
    Init -->|fan-out| Doc[DocAnalyst
(PDF)]
    Init -->|optional| Vision[VisionInspector
(disabled)]
    Repo --> Agg[Aggregate Evidence]
    Doc --> Agg
    Vision --> Agg
    Agg -->|fan-out| Pros[Prosecutor
(Trust No One)]
```

```
    Agg -->|fan-out| Def[Defense
(Reward Effort)]
    Agg -->|fan-out| Tech[Tech Lead
(Does it actually work)]
    Pros --> Err[Handle Error
(fan-in)]
    Def --> Err
    Tech --> Err
    Err --> CJ[Chief Justice
(weighted synthesis + caps)]
    CJ --> Finalize[Finalize + Report]
    Finalize --> End([END])
    subgraph Enforcement
      Pros -.-> SO1[StructuredOpinion JSON]
      Def  -.-> SO2[StructuredOpinion JSON]
      Tech -.-> SO3[StructuredOpinion JSON]
    end
```

3. Self-Audit Criterion Breakdown

- forensic_accuracy_code: 4/5 (prior runs) or 3-4/5 (current). Evidence: src/tools/git_tools.py sandbox; src/core/state.py Pydantic models.
- forensic_accuracy_docs: 2/5. Gap: PDF must explicitly cite src/core/graph.py and src/agents/judges/base_judge.py to prove parallelism and structured enforcement.
- judicial_nuance: 4/5 (some runs 3/5). Distinct prompts in judge files; StructuredOpinion enforcement in base_judge.py.
- langgraph_architecture: 4/5. Parallel branches proven; missing explicit recovery edges keeps it below 5.

Dialectical Tension

- Docs variance (1-4): Prosecutor flags uncited claims; add code citations to PDF.
- Nuance variance (2-4): Prosecutor wants stronger proof of persona separation and JSON enforcement; ensure PDF cites judge prompt files and base_judge.py.

4. MinMax Feedback Loop Reflection

- Peer findings: hallucinated file references in prior PDFs; sandbox proof insufficiently cited.
- Actions taken: added sandboxed_git_clone evidence; persona differentiation evidence; force_json_mode for LM Studio.
- Peer audit: caught missing imports/file references, reinforcing evidence-first judging.
- Insight: explicit file citations + StructuredOpinion grounding reduce hallucination penalties more than prompt tweaks.

5. Remediation Plan

- [High] Add conditional edges for evidence-missing/node-failure in src/core/graph.py.

- [High] Update this PDF to include explicit code citations for parallelism (src/core/graph.py), sandbox (src/tools/git_tools.py), StructuredOpinion (src/agents/judges/base_judge.py), personas (src/agents/judges/*.py).
- [Med] Persist persona-differentiation evidence into judge context each run (RepoInvestigator output).
- [Med] Decide VisionInspector: enable with OCR + evidence, or drop the claim entirely.
- [Low] Add a small log of judge JSON outputs for auditability.