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: Autonomous audit graph with detectives, judges, and Chief Justice for code + PDF inputs.

- Latest self-audit outcome: 12/20 (recent run). Prior runs with same code: 14/20. Criterion scores: forensic_accuracy_code 4/5; forensic_accuracy_docs 2/5; judicial_nuance 4/5; langgraph_architecture 4/5.

- Key takeaways: Security cleared via sandboxed git; parallel graph implemented; docs score low when PDF claims are not tied to explicit code citations; Vision disabled by default.

- Actionable next steps: Add explicit failure edges; tighten PDF-to-code citations; decide on Vision (enable with evidence or drop claim).

2. Architecture Deep Dive and Design Rationale

- Parallel fan-out/fan-in implemented in src/core/graph.py: initialize -> repo_investigator, doc_analyst (optional vision_inspector); aggregate_evidence -> prosecutor, defense, tech_lead; judges -> handle_error -> chief_justice -> finalize.

- Dialectical synthesis: Prosecutor, Defense, Tech Lead run in parallel; Chief Justice (src/agents/justice/chief_justice.py) applies weighted synthesis with variance caps.

- Grounding and output validation: StructuredOpinion + _coerce_structured_response (src/agents/judges/base_judge.py) enforce JSON schema and prune unverified citations; force_json_mode for LM Studio to avoid tool-call errors.

- Personas: Distinct prompts in src/agents/judges/prosecutor.py ("Trust No One"), defense.py ("Reward Effort"), tech_lead.py ("Does it actually work").

- Security: RepositorySandbox.clone_repository in src/tools/git_tools.py; no raw os.system.

- Rationale: Pydantic models in src/core/state.py for validation; deterministic synthesis rules for repeatability; VisionInspector remains disabled (not production-ready).

Graph Flow (fan-out + fan-in)

```mermaid

flowchart TD

Start([START]) --> Init[Initialize]

Init -->|fan-out| Repo[RepoInvestigator\n(sandboxed git)]

Init -->|fan-out| Doc[DocAnalyst\n(PDF)]

Init -->|optional| Vision[VisionInspector\n(disabled)]

Repo --> Agg[Aggregate Evidence]
```

Doc --> Agg

Vision --> Agg

Agg -->|fan-out| Pros[Prosecutor\n(Trust No One)]

Agg -->|fan-out| Def[Defense\n(Reward Effort)]

Agg -->|fan-out| Tech[Tech Lead\n(Does it actually work)]

Pros --> Err[Handle Error\n(fan-in)]

Def --> Err

Tech --> Err

Err --> CJ[Chief Justice\n(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 (some runs 3/5). Evidence: sandboxed git (src/tools/git_tools.py); Pydantic models (src/core/state.py). Prosecutor noted missing explicit recovery edges.

- forensic_accuracy_docs: 2/5. Judges: Pros=1, Def=4, Tech=3 (capped). Gap: PDF must explicitly cite src/core/graph.py for parallelism and src/agents/judges/base_judge.py for structured enforcement.

- judicial_nuance: 4/5 (some runs 3/5). Evidence: distinct prompts in src/agents/judges/*.py; StructuredOpinion enforcement in base_judge.py. Prosecutor wants stronger proof in PDF context.

- langgraph_architecture: 4/5. Evidence: parallel fan-out/fan-in in src/core/graph.py; missing explicit recovery edges for evidence-missing/node-failure keeps it below 5.

Dialectical Tension

- Docs variance (1-4): Prosecutor challenges uncited PDF claims; Defense cites theory; Tech Lead mixed.

- Nuance variance (2-4): Prosecutor wants stronger proof of persona separation and JSON enforcement.

4. MinMax Feedback Loop Reflection

- Peer findings: hallucinated file references in earlier PDFs; sandbox proof weakly cited.

- Response actions: added sandboxed_git_clone evidence; persona differentiation evidence; force_json_mode for LM Studio.

- Our audit of peer: caught missing imports/file references; reinforced evidence-first judging.
```

- Bidirectional insight: explicit file citations plus StructuredOpinion grounding improve reliability more than prompt tweaks.

5. Remediation Plan (prioritized)

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

- High: Update PDF to cite code explicitly: src/core/graph.py (fan-out/fan-in), src/tools/git_tools.py (sandbox), src/agents/judges/base_judge.py (StructuredOpinion), judge prompt files.

- Med: Persist persona-differentiation evidence into judge context each run (RepoInvestigator output).

- Med: Decide VisionInspector: enable with OCR + evidence, or remove the claim.

- Low: Add a small log of judge JSON outputs for auditability.