Interim Technical Report – Automaton Auditor

By Amare Kassa February 25, 2026

1. Introduction

- Objective: Build a hierarchical multi-agent auditor (Detectives → Judges → ChiefJustice) for Week 2.
- Scope of interim submission: foundation of Detective Layer, state modeling, repo/PDF/AST tools, and architecture planning for Judicial Layer.

---

2. Architecture Decisions So Far

Typed Agent State

- **Pydantic vs. Python dicts**:
  - Enforces strict type validation.
  - Ensures structured AgentState across nodes and fan-out/fan-in.
  - Prevents accidental mutations and supports traceable JSON serialization.

StateGraph Orchestration

- Nodes (`Node` class) with `run()` function for typed state transition.
- Supports sequential and conditional execution.
- Fan-out/fan-in planned for Detectives → EvidenceAggregator.

Detective Layer Tools

- **AST Parsing (`ast_tools`)**: Extract exports, imports, and function signatures from Python source.
- **Git Tools (`git_tools`)**: Clone, list commits, enumerate tracked files.
- **PDF Tools (`pdf_tools`)**: Extract, chunk text for DocAnalyst querying.
- **Filesystem Utilities (`filesystem.py`)**: Temp directories, safe file writes, SHA256 hashes.

Sandboxing Strategy

- Clone and analyze repos in temporary directories using `tempfile`.
- Prevents accidental overwrites, ensures isolated environment per repo.

---

3. Known Gaps

- **Judicial Layer**:
  - Personas (Prosecutor, Defense, TechLead) not yet fully implemented.

  - Structured JSON outputs with criterion-specific reasoning pending.
- **ChiefJustice / Synthesis Engine**:
  - Conflict resolution and score synthesis logic not yet implemented.
  - Audit report generation in Markdown still a placeholder.
- **Vision Inspector**:
  - Multimodal diagram analysis optional, not yet integrated.

---

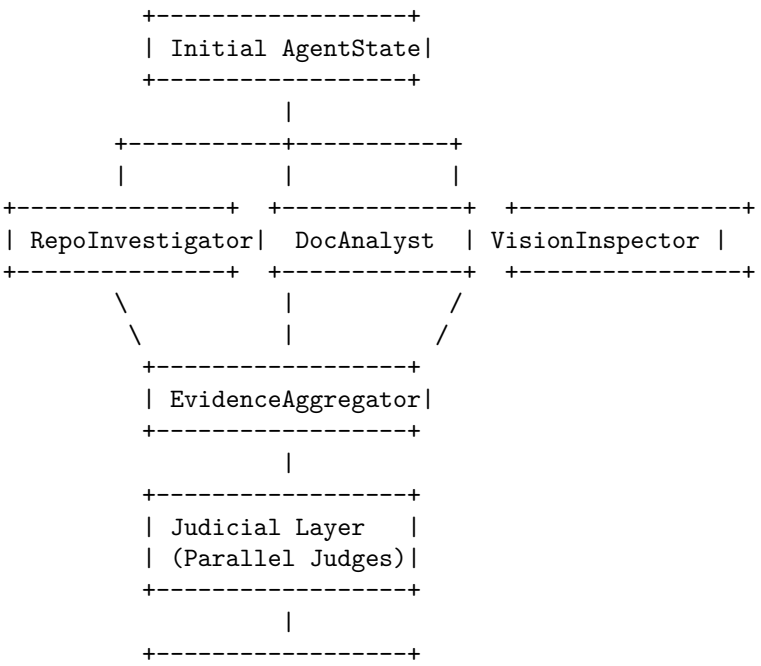
4. Planned Implementation

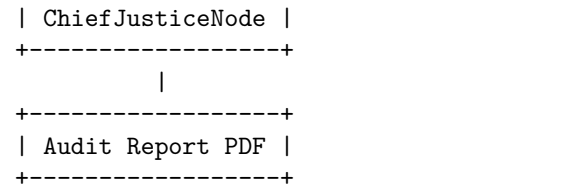
1. **Judicial Layer**
   - Load evidence from EvidenceAggregator.
   - Execute each Judge in parallel using distinct prompts/system logic.
   - Generate `JudicialOpinion` objects for each rubric criterion.
2. **ChiefJusticeNode / Synthesis**
   - Aggregate opinions.
   - Apply rules: security_override, fact_supremacy, dissent_requirement.
   - Generate structured Markdown Audit Report with Executive Summary and Remediation Plan.

---

5. Planned Diagrams

StateGraph Flow (Detective Fan-Out / Fan-In)





---

6. Next Steps

- Implement parallel Judge execution with persona-specific structured output.
- Integrate ChiefJustice conflict resolution and synthesis.
- Generate Markdown/structured audit reports.
- Optional: VisionInspector multimodal analysis.
- Batch-processing multiple repositories for adversarial MinMax testing.