Automaton Auditor Architecture Report

This Automaton Auditor deeply relies on the principles of Dialectical Synthesis to evaluate repositories. By spawning three parallel judge personas, we ensure state-of-the-art Metacognition that is not superficial but baked directly into the graph edges.

Graph execution uses a Fan-In / Fan-Out model. First, Detectives fan-out to gather isolated forensic evidence. Then, State Synchronization takes place at the EvidenceAggregator node, before fanning out again to the Judges.

File organization details: We isolated the AST logic in src/tools/repo_tools.py. We implemented parallel Judges in src/nodes/judges.py. State definition resides in src/state.py, ensuring Pydantic and TypedDict safety. Finally, deterministic rules execute within src/nodes/justice.py. The entire graph is orchestrated in src/graph.py.

Below is the architectural state diagram showing START -> Detectives -> Aggregator -> Judges -> Justice -> END.