

Harness Components

Six Harness Components

- State and persistence — durable memory and workflow checkpoints across long-running runs.
- Security and governance — permissioning, sandboxing, credential scoping.
- Orchestration and tool routing — how planner/executor/verifier roles hand off work.
- Context assembly — the same responsibility described in the Context Engineering layer below.
- Observability — traceable artefacts across the full agent lifecycle (AgentOps).
- Recovery and self-healing — retries, re-planning, and graceful degradation when a step fails.

Harness vs. Framework

A framework (LlamaIndex, LangGraph) supplies reusable primitives. A harness is the specific, opinionated configuration of those primitives — plus the repo-local instructions, feedback loops, and verification gates — that a team assembles for one production system. Two teams using the same framework can have very different harnesses, and that difference is now measurable: Harness-Bench reports swings of over 20 points in task success when the harness changes but the underlying model pool does not.

Revision #1

Created 2026-08-24 11:59:10 UTC by Aisha M. Nur

Updated 2026-08-24 11:59:29 UTC by Aisha M. Nur