Agent Runtime Infrastructure

Durable execution · Sandboxing · Task Queues · Temporal · Subagents

Senior SWE / AI Engineering — Deep Dive Recap

WHY THIS TOPIC

The plumbing beneath the agent harness

Named org function

Anthropic runs distinct Sandboxing and Product Sandboxing teams; OpenAI + Anthropic hiring skews toward agent infra.

Converged architecture

Claude Managed Agents & redesigned OpenAI Agents SDK both shipped the same harness/sandbox split, April 2026.

Industry consolidation

Temporal joined the Linux Foundation's Agentic AI Foundation as a Gold Member — durable execution is now core infra.

CORE DISTINCTION

Framework vs. runtime infrastructure

Framework

- Provides the slot: policy + interface.
- e.g. "when to checkpoint," "what key to save."
- LangGraph's checkpointer API is a framework decision.

Runtime Infra

- Fills the slot: what goes in it, what breaks.
- Postgres connection pool exhaustion under high-frequency checkpointing.
- Transferable across frameworks — a property of Postgres, not LangGraph.

Why agent work happens in discrete sessions

Two layers of statelessness

The Claude API itself (no server memory across calls) AND the harness process (an ordinary process — can crash, get OOM-killed, redeploy).

Checkpointer = the fix

Full state snapshot per superstep, persisted to external durable storage (Postgres/SQLite/Redis) — never in-memory (MemorySaver dies with the process).

Checkpointer is thread-scoped (single run). Store is cross-thread (shared across runs).

SANDBOXING — THE TIERS

Docker vs. gVisor vs. Firecracker

Docker / runc

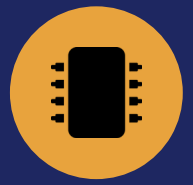
Shared kernel

Built for portability, not security. Isolation is a side effect — insufficient for untrusted, adversarial AI-generated code.

gVisor

Software isolation

Reimplements kernel logic in unprivileged userspace (Sentry). Drop-in runc replacement — runs anywhere containers run.



Firecracker

Hardware isolation

Guest runs its own real kernel under VT-x/AMD-V + KVM. Stronger isolation, needs bare-metal/KVM access.

Privilege demotion, not just reimplementation

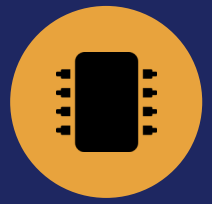
gVisor's Sentry is a full parallel reimplementation of kernel-equivalent logic — but it runs in unprivileged userspace and only issues narrow, controlled real syscalls to the host kernel on the sandboxed process's behalf.

Real kernel:
Full hardware privilege (ring 0). A bug there is catastrophic.

gVisor (Sentry):
Same logic, demoted to unprivileged userspace. A bug is contained — it still can't escalate.

FIRECRACKER — MECHANISM

The ring-0 illusion



A naively-booted guest kernel faces a bind: give it real ring 0 and it can overwrite host state, or give it ring 3 and it immediately faults on its own privileged instructions.

1

VT-x / AMD-V

CPU-level guest-mode privilege ring — the guest executes real instructions on real silicon.

2

Privileged op

Guest attempts something privileged (e.g. touch a page table).

3

KVM traps it

Hardware traps the operation, hands control to KVM (the host module).

KEY CORRECTION

Isolation ≠ zero access

Wrong model

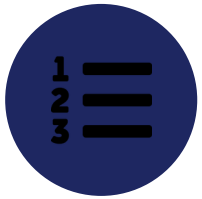
"Sandboxed" = the process can't reach the real system at all.

Correct model

Isolation = mediated access through one narrow, checked gate. Every crossing point is narrowed and checked — software check (gVisor) or hardware trap (Firecracker) — not eliminated.

TASK QUEUES

Durable storage for pending work



Direct call: harness calls a tool function in-process (blocking). If the OS kills the harness mid-call, the work vanishes — no code survives to run any except clause.

Queued call: the task is written to durable storage before execution, independent of any one process — same principle as the checkpointer, applied to individual task units instead of full conversation state.

Exception vs. crash: a tool call raising an exception (try/except-catchable) is not the same failure as the OS killing the entire harness process (nothing survives to catch anything).

Batch atomicity & async results

Partial-batch enqueue failure

LLM emits multiple tool calls; harness crashes after enqueueing only some. Fix: atomic multi-call batch writes — same shape as the checkpointer's full-snapshot-per-superstep.

Async result delivery

A queued call is inherently async. Results land in a durable result store the harness polls or is pushed to — trading the durability problem for a sync/async problem.

Harness-crash resume

A fresh harness instance checks the durable result store for pending tasks rather than needing perfect uptime — same resume shape as checkpointer restarts.

IDEMPOTENCY

The tool-layer fix for at-least-once delivery

Queues guarantee at-least-once delivery, not exactly-once. A worker retry after crash could fire send_welcome_email() twice.

Idempotency keys

belong at the tool/action layer, not the queue layer.

Queue's job: guarantee the call happens at least once.

Tool's job: guarantee safety-under-repeats (idem + potent, Peirce 1870).

TEMPORAL

Durable execution via replay, not checkpoints

Workflow

Deterministic orchestration code. Replayable from stored event history.

Activity

Side-effecting work (LLM calls, tool calls). Auto-retried with backoff.

The mechanism: execute_activity() reports to the Temporal server on the way out (real run), checks event history on the way in (replay) — no manual resume bookkeeping. Workers do all real work; the server holds durable history only.

Why isolate subagents from each other

Conversation state (message history, tool permissions) is cheap — just an object in memory, no new process needed. Sandbox isolation solves a different problem:

Resource contention

Parallel subagents writing to a shared filesystem collide — one overwrites another's changes. Fix: git worktrees or separate filesystem sandboxes.

Blast-radius containment

A compromised or buggy subagent (arbitrary LLM-generated code) shouldn't spread damage to the parent or siblings — same principle as gVisor/Firecracker, one level up.

SUBAGENT LIFECYCLE

Tracking, timeouts, and escalation

Per-subagent record: start time · status label (running / succeeded / failed / timed-out) · process/sandbox handle · output slot

A naive retry on timeout just reproduces the same failure. Escalate differently:

- **Decompose further (narrower sub-tasks)**
- **Escalate to a stronger model**
- **Escalate to a human**
- **Fail gracefully (surface the error)**

CONCURRENCY LIMITS

Depth caps termination; width caps rate

Depth limit

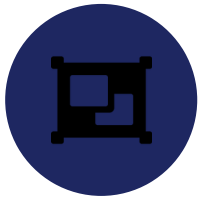
Nested subagent-spawns-subagent chains grow exponentially (3 → 9 → 27…). 2026 production guidance caps nesting at 5 levels — a structural, non-negotiable ceiling that guarantees the recursion terminates.

Width limit

Concurrent siblings contend for API rate limits, cost, and local compute. A queue/semaphore caps how many run at once — controls rate of resource use, not whether the total lineage ever ends.

RESULT AGGREGATION

Reducers, again — one level up



When parallel subagents finish at different times and write results to shared state, the parent needs the same thing you already know from LangGraph: a declared merge policy per key.

Overwrite (default)

Last write wins — other subagents' results are silently lost. Wrong choice here.

List-append reducer

Each result appended to a list (like operator.add). Nothing lost, order-independent.

Synthesis pass

A separate LLM call reads all N results and produces one combined, deduplicated answer.

INDUSTRY SIGNALS

This is now named, funded infrastructure

- Anthropic + OpenAI independently shipped the same harness/sandbox split within a 7-day window, April 2026
- Claude Managed Agents in production at Notion, Rakuten, Sentry — $0.08/session-hour + tokens
- Temporal joined the Linux Foundation's Agentic AI Foundation as a Gold Member
- Anthropic runs distinct Infra Sandboxing (Research) and Staff+ Product Sandboxing roles
- OpenAI Agents SDK: subagent orchestration routes work to isolated sandboxes natively

Agent Runtime Infrastructure — Complete

Interview one-liner:

"Frameworks give you the slot; runtime infra is knowing what goes in it and what breaks under load — durable storage, mediated isolation, and idempotent retries, all the way down."