

Temporal vs LangGraph vs OpenAI Agents SDK vs Microsoft Agent Framework: Who Actually Survives a Crash

A worker process dies halfway through a twelve step agent run. What happens next is not a framework feature you compared during evaluation, it is the billing model you accepted without reading.

Temporal Cloud charges $50 per million Actions on the Developer tier, and an Action is any billable operation: starting a workflow, sending a signal, recording a heartbeat, completing an activity. One agent run generates many Actions. That rate is Temporal telling you exactly what it persists, and by comparison LangGraph's per-thread checkpoints, the OpenAI Agents SDK's Sessions and Microsoft Agent Framework's per-superstep checkpoints persist less.

The four tools in the title are marketed as alternatives. On crash recovery they are not comparable, and the price list is the clearest evidence. Temporal can bill per Action

because it records every Action in an event history it can replay. The others cannot bill that way because they do not have that history to sell you.

ATTRIBUTE	TEMPORAL	LANGGRAPH	OPENAI AGENTS SDK	MICROSOFT AGENT FRAMEWORK
Recovery model	**Durable execution with replayable event history, retries, timers, signals**	Per-thread checkpointing (state persistence, not replay)	Conversation state via Sessions	Workflow checkpoints per superstep
Where durability lives	**Core platform, framework-independent**	Core framework, backed by PostgreSQL, MongoDB or Cosmos DB	External: Temporal, Restate, DBOS or Dapr	Durable Extension, not the core framework
Published unit price	$50 per million Actions, stepping to $45M/$40/$35/$30/$25 at volume	n/a (no official pricing page retrievable, November 2026)	n/a as a separate orchestration SKU	**Microsoft Foundry platform free to use; consumed features billed at normal rates**
Storage billing	$0.042/GB-hour active, $0.00105/GB-hour retained	Your own database bill	n/a	n/a (no flat agent-framework subscription)
Entry cost to try	$150 credits for 90 days, no Developer base fee	n/a	**No separate orchestration charge**	**Free to use and explore**
Enterprise floor	Business tier: greater of $500/month or 10% of consumption	n/a	n/a	n/a

Read the first two rows together. Temporal is the only one of the four where durability is a property of the platform rather than a property of the agent library, which is also why the OpenAI Agents SDK's documented path to durable execution is to bolt on Temporal, Restate, DBOS or Dapr. The Agents SDK made a scoping decision: it manages conversation state and delegates crash semantics. It is honest about it. You just have to notice.

Microsoft Agent Framework sits in the uncomfortable middle. Checkpoints per superstep give you resumption boundaries, but durable waits and recovery arrive via the Durable

Extension, which means the thing you most need in production is the thing outside the core. I wrote about the gap between that framework's documentation and its shipping state in [Microsoft Agent Framework in Practice](https://www.arturmarkus.com/microsoft-agent-framework-in-practice-what-the-docs-promise-and-what-prerelease-tells-you/), and the extension split is the same pattern showing up in the persistence story.

LangGraph's per-thread checkpointing against PostgreSQL, MongoDB or Cosmos DB is the easiest of these to adopt, for the dull reason that you already run one of those databases. A checkpoint lets you resume a graph from a saved node state. It does not let you replay the full execution history deterministically to reconstruct what the process was doing between checkpoints. For a research agent, resuming from the last good node is fine. For an agent that issued a payment, "resume from the last checkpoint" and "replay the exact sequence of side effects" are very different guarantees.

Heartbeats and long waits, not tool calls, decide your Temporal bill

I cannot give you a per-agent cost, and anyone who does without knowing your agent's shape is guessing. The research confirms the rate and confirms that an Action is a fine-grained operation rather than a workflow run. It does not give a published Actions-per-agent-run figure. So the arithmetic is yours to do.

What I can point at is where the cost concentrates. Long running activities that heartbeat to prove liveness generate Actions on a timer, whether or not the agent is making progress. Signals do the same if your agent is interrupt-driven. An agent that calls four tools and finishes is cheap. An agent that waits on a human for two days while heartbeating is a different bill entirely, and it is exactly the kind of agent people say they want.

Temporal also charges $0.042 per GB-hour for active workflow storage against $0.00105 per GB-hour for retained storage, a 40x gap. Long-lived agents holding large context in event history sit in the expensive bucket for as long as they stay open. **My reading: retention policy and workflow close discipline will dominate your Temporal bill more than the per-Action rate**, and that is a design question rather than a procurement one.

Volume discounts step from $50/M down through $45, $40, $35, $30 to $25 per million at higher monthly volumes, with the first break after 5M Actions. That curve is generous in percentage terms and irrelevant at pilot scale. The $150 in credits over 90 days with no Developer base fee is enough to find out what your real Action rate is, which is the only pricing research worth doing before you commit.

The Business tier floor deserves a flag: the greater of $500/month or 10% of Temporal Cloud consumption. If you are spending $2,000 a month on Actions and storage, Business costs you $500, not $200. The floor bites at exactly the scale where a team graduates from experiment to production and starts wanting support.

Cloudflare Durable Objects is the option nobody put on the shortlist

It is not an agent framework, which is why it keeps getting left out of these comparisons, and it prices durability on a different axis: $0.15 per million requests and $12.50 per million GB-seconds on overage, with a free plan covering 100,000 requests and 13,000 GB-seconds per day, and a paid plan including 1M requests and 400,000 GB-seconds per month. That is resource-based billing rather than per workflow execution.

Put the request pricing next to Temporal's Action pricing and the gap is wide enough that the comparison nearly stops meaning anything: $0.15 per million versus $50 per million. You are buying different things. Durable Objects gives you a single-threaded addressable object with persistent state, without replay semantics, event history, or built-in retry and timer machinery. Temporal gives you the machinery. The price difference is roughly the price of that machinery, which is a defensible thing to pay for and an absurd thing to pay for by accident.

WHERE I LAND

My take: for a team shipping agents that touch money, inventory, or anything a regulator will ask about, I would pick Temporal and treat the Actions bill as the cost of an audit trail you were going to have to build anyway. For everything else, LangGraph's checkpointing against a Postgres you already operate is the right default, and the moment you find yourself writing your own retry, timer and resume logic on top of it is the moment to reconsider. The OpenAI

Agents SDK is a reasonable agent layer whose durability answer is "use one of the others", so pick the other first.

Microsoft Agent Framework inside Microsoft Foundry has the friendliest entry economics of the four: the platform is free to use and explore, with consumed features billed at normal rates and no single flat agent-framework subscription price. If your organisation is already in Azure and your agents are stateless enough that per-superstep checkpoints cover your failure modes, this is cheap to start and hard to argue against on cost. My concern is the Durable Extension boundary. Features that live outside a core framework move on a different release cadence and carry different support commitments, and I would want that confirmed in writing before building a payment flow on it.

One honest gap here: no official LangGraph Platform or LangSmith pricing page was retrievable during this research run. A secondary source estimates $0.0675 per vCPU-hour plus database charges. I am not treating that as authoritative and neither should you. If LangGraph Platform pricing matters to your decision, get it from LangChain directly rather than from any comparison article, including this one.

Choose by what you want to happen when the process dies at step seven

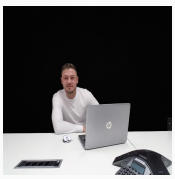
Forget which framework has the nicest agent abstractions. Ask what you want to happen when the worker dies at step seven of twelve.

If the answer is "re-run it, duplicates are harmless", you do not need durable execution and you should not pay for it. Checkpoints are plenty. If the answer is "resume from where it stopped, roughly", LangGraph or Microsoft Agent Framework checkpoints fit, and the word "roughly" is doing real work there. If the answer is "reconstruct exactly what happened, replay deterministically, never double-execute the side effects", Temporal is the only one of the four that claims this natively and independently of the agent framework, and the per-Action pricing is the direct consequence of that claim.

That last distinction is the one teams skip in evaluation and discover in incident review. I made a similar point about workflow tools in [the automation stack comparison for

CTOs](https://www.arturmarkus.com/n8n-vs-zapier-vs-make-vs-openai-agent-builder-the-honest-automation-stack-comparison-for-ctos/): **the pricing model tells you what the vendor thinks the product is**, usually more reliably than the feature page does.

Temporal thinks it is selling you a record of every operation. $50 per million of them, with the pricing page updated October 23, 2026.



WRITTEN BY

Artur Markus

Software engineer in Basel building AI infrastructure and agentic systems, with a background in pharmaceutical automation, GxP compliance, and validation. I help teams turn AI from a chatbot into a reliable operational layer.

Book a meeting →