

Zero-AR

The runtime beneath long-horizon work.

Zero-AR (Zero-layer Autonomous Runtime) runs your agent's work from the first step to a finish your own checks confirm, across waits, failures, people and handoffs. You build the agent. Zero-AR runs the work.

The problem

Real work outlives one conversation. It waits days for a reply or an approval, survives restarts and model changes, and touches systems where a mistake costs money. Most agent stacks let the agent decide when it is done, lose the thread when a process stops, and leave nobody able to say what happened.

One job, start to finish

An agent asks a supplier for an invoice by email and reads the PDF that comes back. A line differs from the order, so it asks the owner to approve a question to the supplier. The owner approves and allows one follow-up. The supplier answers on a new thread, and the agent replies on its own. The work is done when the owner's checks pass: the figures it quotes appear on the invoice, the amount to book adds up, and every message went out. In our study it took 82 seconds, 24 of them spent waiting for people. While it waits, the run holds no worker and makes no model call.

What you get

- **Done means your checks passed.** The agent cannot finish its own work by saying so. Work that cannot finish says what is left and why.
- **Work that survives failure.** Every step is kept in the run's record, so a crash, a restart or a week-long wait resumes where it stopped. Several Zero-AR cells can share one database, and a run continues when the cell running it stops. A run can also be handed to another cell, which continues it with its history and receipts.
- **Budgets set before spending.** Each model call reserves its share of the run's budget before it runs, and tools run within the limits you declare. A run that reaches its limit stops and says why.
- **Actions under your authority.** An action that changes an outside system waits for an exact approval, or goes out under a grant you set: a follow-up allowed with an approval, or a cap on the amount an action may carry. It is sent once, with a receipt, and an unknown outcome is checked with your

system before anything is sent again. Your application performs the action behind a signed call, and one command checks your endpoint before you go live.

- **Approvers you already know.** People approve through your own identity provider, or your product vouches for them with its own key.
- **Sub-runs your agent can delegate to.** An agent can hand part of its objective to sub-runs. Each one works inside the tools and budget its parent holds. The parent waits at no cost and shows how many are finished and how many are still running.
- **Knowledge at real size, with no index to build.** Agents search up to 20,000 documents per source by the words people write, in any case, ranked, and continued across calls. Long PDFs are read in place. Your checks see the passages the agent read.
- **A record anyone can check.** Each run exports as a bundle that rebuilds the work without the server, and each deployment reports the capabilities it actually runs.

What we measured

Across 58 runs of everyday work with seven models (tax returns, freight audits, and supplier invoices by email):

- **Taken at their word, the agents finished 43 runs, and one in three of those was wrong.** Zero-AR's checks confirmed 26, and none of those was wrong.
- **22 runs finished with no one stepping in**, apart from approvals the owner chose to keep.
- **Every tax return finished** with Claude Sonnet 4.6 and with Claude Haiku 4.5, at about \$0.07 a return with Haiku.
- **Nine of ten supplier invoices settled by email** with no one stepping in beyond the approvals the owner kept.

The people on the other side were scripted, so every run met the same situation.

Where it fits

Zero-AR sits under your product. Run it on one machine, or as a hosted cell inside your cloud on PostgreSQL. Connect through the TypeScript SDK, a typed client, the HTTP API, the command line or MCP. Tested with Claude, OpenAI models and models that run on a laptop.

Zero-AR does not decide what correct means: you attach the checks for each job. It does not make every outside action reversible, and it is not an agent framework or a finished vertical application.

Work with us

We are looking for design partners with everyday work to hand to agents. Write to hello@finesample.com.