

Zero-AR on τ^2 -bench: level with the benchmark's own agent, and every pass verified

Summary

- **What Zero-AR is.** Zero-AR is the runtime for long-horizon agent work. It runs AI agents that carry out day-to-day tasks on their own, get better the longer you use them, and verify completion. You build the agent. Zero-AR runs it to the finish line.
- **The test.** On τ^2 -bench, Sierra Research's public benchmark for customer-service agents, Zero-AR ran six retail tasks that its agent had never seen. The benchmark's own reference agent ran the same tasks, with the same AI model, the same simulated customer and the same grader. We wrote down the tasks, measures and targets before the run began.
- **The result.** Both agents completed 4 of the 6 tasks. Zero-AR also checked its own work, and it verified every one of its 4 completed tasks. No person stepped in during any conversation.
- **The one miss.** Zero-AR verified a fifth run that the grader failed. The agent changed an order to the wrong address, and the simulated customer confirmed it. Zero-AR's checks confirmed that every step was followed. They cannot tell that the customer agreed to the wrong thing.
- **The cost.** Zero-AR cost about 2.6 times as much as the reference agent at the provider's prices for repeated input, because it made more model calls and each carried more context.

What Zero-AR is

An AI agent is software built around an AI model: the model decides each next step, and the agent's tools carry it out. Agents are good at single steps. Real work is longer:

- it waits on people;
- it makes changes that matter;
- it has to end in a result someone can rely on.

Zero-AR is the layer agents run on for that kind of work. It runs the agent's loop, calls the model and the tools, and keeps a complete record of the run. Four promises follow from that.

It finishes the work on its own. A run carries its objective from start to finish without a person nudging it along. When the work has to wait, for a reply or for someone's approval, the run sleeps. While it sleeps it holds no worker and makes no model call, until there is something new to act on.

It acts only with permission, and keeps the receipt. Every change an agent makes to the outside world is proposed first. It goes ahead only under a permission the owner has given, and it counts only once the owner's system has confirmed it with a receipt. An agent cannot quietly change something on its own.

It verifies completion against your checks, and claims nothing more. The owner attaches the checks that say what done means for the job. Zero-AR runs them, and a model's opinion never counts as a check. A run is reported verified only when its checks pass. Otherwise it is reported as not verified, with the reason.

It gets better the longer you use it. A run keeps everything admitted since it began, however long it lasts and whoever takes part: evidence, answers, steering, decisions, checks and approvals. It carries them across restarts, model changes and changes of person. Each conversation in this benchmark is short, so it tests the first three promises more than this one.

Zero-AR does not decide what correct means for your job. You attach the checks that express it. Zero-AR enforces them, and it does not turn an unresolved result into a completed one.

Why this benchmark

In each τ^2 -bench retail task, an AI model plays a customer with a goal, such as returning an item, changing an order or moving a delivery. The agent serves that customer under the store's written policy, using the store's own tools to look up accounts and orders and to make changes. When the conversation ends, τ^2 -bench's grader checks the store's final records. It tests an agent that has to make changes under rules, and it is public, so anyone can run the same tasks: the [code and tasks](#) are on GitHub, and the [paper](#) describes the method.

How we ran it

	Zero-AR	Reference agent
AI model	GPT-5.2	The same
Simulated customer	τ^2 -bench's own, on GPT-5.2	The same
Grader	τ^2 -bench's own, on GPT-4.1	The same
How the agent runs	Published from ordinary project files and run by Zero-AR in a hosted runtime	τ^2 -bench's own agent loop
How store changes happen	Proposed, approved and receipted before they count	Called directly

- **The Zero-AR agent.** It had the store’s public policy, the store’s tools, and our instructions for confirming a change before making it.
- **Approvals.** An automated stand-in for the store owner approved each proposed change, so the agent ran with no person involved.
- **Checking its own work.** When the customer ended the conversation, Zero-AR ran three checks:
 - every change has a matching receipt from the store;
 - the customer ended the conversation after the agent’s last reply;
 - every change followed the policy’s confirmation steps: the customer was identified from details they gave, the order was read first, one message listed exactly what would change, and the customer said yes.

A run is **verified** when all three pass.

- **Choosing the tasks.** The six tasks came from an order fixed in advance by a seeded hash of the public task ids. Every task used to develop the agent was excluded. The tasks, the measures and the targets were written down before the run.

Results

Task	What the customer wanted	Zero-AR	Zero-AR’s checks	Reference agent
54	Cancel or return as many recent orders as possible, and keep a pair of boots unless the same boots cost less	Completed	Verified	Not completed
61	Change the colour of earbuds in an order, at no extra cost	Completed	Verified	Completed
17	Send an order to a different suite	Completed	Verified	Completed
84	Return one of two tablets that arrived	Completed	Verified	Completed
109	Move an order to a new home address, make it the default address, and exchange a tablet for the cheapest option	Not completed	Verified	Not completed
80	Exchange a T-shirt for another variant	Not completed	Not verified	Completed

“Completed” means τ^2 -bench’s grader passed the task.

	Zero-AR	Reference agent
Tasks completed	4 of 6	4 of 6
Completed tasks that Zero-AR verified	4 of 4	not applicable
Verified runs the grader failed	1	not applicable
People who stepped in	0	0
Calls to the AI model	110	46
Cost at the provider's prices for repeated input	\$0.69	\$0.26
Cost at full list price	\$2.28	\$0.55

Every completed task verified. All four tasks Zero-AR completed passed its own checks.

Task 80 shows a run that does not claim too much. The customer answered the agent's confirmation with yes and ended the conversation in the same message, so the agent never had a turn to make the exchange. The store was unchanged, and Zero-AR reported the run as not verified.

The one verified run that was wrong

In task 109 the customer had moved house along the same street. They said they would not give the new address, and that the agent could find it in their orders. The agent assumed the address already on the account was the new one, and listed the changes for that address. The simulated customer confirmed them.

The right address was the other one, which a recent order was shipping to. So every change was confirmed and receipted, and Zero-AR's checks passed, but the change was wrong.

This is the limit of checks that test whether the confirmation steps were followed. They establish that the customer was asked, listed every detail, and said yes. They cannot establish that the customer understood. That is why every verified run here is set beside the benchmark's own grade.

A one-task trial before the main run, on a task already used in development, showed the same pattern. The agent wrongly said it could not choose cheaper versions of the customer's items. The customer accepted a cancellation instead and confirmed it, and the grader failed the run.

What it cost

Zero-AR made 2.4 times as many model calls as the reference agent, and each carried about 2.2 times as much context: 10,700 input tokens on average against 4,800. In an average Zero-AR request:

- the tool descriptions took about 5,900 tokens, against roughly 3,000 for the reference agent's, because they include Zero-AR's own controls, such as waiting and recording a result;
- the instructions added about 700 tokens to the store policy;
- the runtime added about 1,400 tokens of framing, including the labels that mark which text came from outside the run.

Most of each request repeats the one before it, and the provider charges less for repeated input. At those prices Zero-AR cost \$0.69 against the reference agent's \$0.26. At full list price it was \$2.28 against \$0.55.

Waiting is free of model calls only some of the time so far. In this run the agent waited on its actions 7 times without calling the model at all, but 32 of its 110 calls still did nothing except start a wait. We are extending free waiting to more of those moments.

How to read these results

- **These are six tasks, each run once.** They show what happened, not rates. With six tasks, 4 of 6 is consistent with a true rate anywhere from about one in five to nearly all. Telling two agents apart would take about thirty paired tasks.
- **The customer is a model.** Every agent met the same simulated customer, which makes the runs comparable. That customer sometimes said yes and ended the conversation in one breath, and in task 109 it agreed to a mistake. These results describe performance against τ^2 -bench's simulated customer, not against people.
- **The agent was developed on the same benchmark.** Its instructions and checks were shaped on twelve other τ^2 -bench retail tasks and on earlier runs.
- **Approvals were automated.** An automated stand-in approved every change, so this measures the agent and the runtime, not a person's review.
- **This is not a leaderboard entry.** It used a development build and six of τ^2 -bench's 114 retail tasks.

Run it yourself

- **Benchmark:** τ^2 -bench, Sierra Research's repository, at commit [96a3ef9](#) (version 1.0.1), retail domain, base task split.
- **Tasks:** 54, 61, 17, 84, 109 and 80, in that order, trial 0.
- **Models:** [gpt-5.2-2025-12-11](#) for both agents and the simulated customer, and [gpt-4.1-2025-04-14](#) for the grader.
- **Reference agent:** τ^2 -bench's own agent loop, unchanged.

Zero-AR's source and the run's full records are not public yet. Every figure in this report comes from those records, and we share them with design partners on request. To run your own agents on Zero-AR, or to see the records, write to hello@finesample.com.

τ^2 -bench is © 2025 Sierra Research and is used under its [MIT license](#). The customers, orders and accounts in its retail domain are fictional.