

A supplier invoice agent on Zero-AR

One agent settling supplier invoices by email: it asks for the invoice, reads the PDF that comes back, questions a difference with the owner's approval, and answers the supplier's next message on its own.

Day-to-day work, done on its own, and verified.

Case study, October 2026

Ten live runs through Amazon SES with Claude Sonnet 4.6, on Zero-AR development builds after 0.4.2

The suppliers, orders and invoices are fictional, and the supplier and the approver were scripted.

Zero-AR does not ship an accounts payable product.

What happened when one agent settled ten supplier invoices by email.

Zero-AR is a runtime for long-running agent work. We built an agent from ordinary project files that settles supplier invoices over email, as a clerk would.

In one paragraph

Nine of ten runs finished verified with the right amount booked, and no person stepped in beyond the approvals the owner chose to give. No run was reported verified when it was not. The tenth reached the right amount by a path its checks could not ground, so Zero-AR held it for a person.

1	The work	3
2	The setup	5
3	One invoice, start to finish	7
4	Results	11
5	The run that did not verify	15
6	What Zero-AR got wrong	16
7	Let your app take actions	17
	What this does not establish	19
	About Zero-AR	20

An invoice is booked only when every line matches the order or is explained.

The agent has a purchase order. The supplier has an invoice that differs from it in one line. Somebody has to find the difference, ask about it, and book the right amount.

The agent starts with the order alone. It emails the supplier for the invoice, reads the PDF the supplier attaches, and compares every line with the order. A line the order does not have, or a different price, is a question for the supplier. The supplier answers with a credit, which means the order's amount stands, or with a reason the charge was agreed, which means the invoice's amount stands. The agent acknowledges the answer and records what to book.

Ten invoices, one difference each

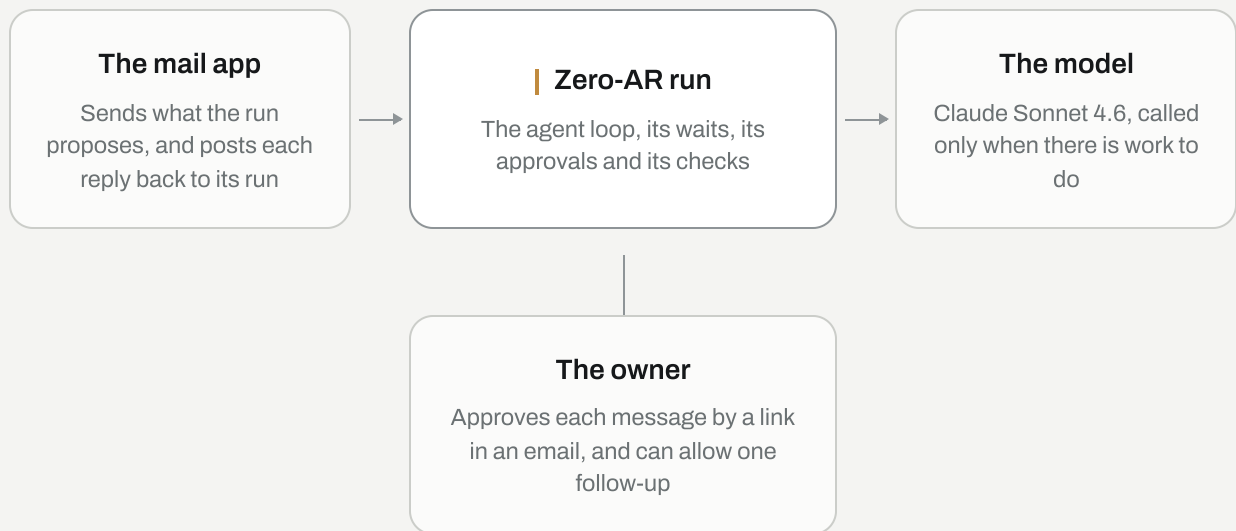
An added freight, assembly, rush, handling or call-out charge, a surcharge, a sales tax line, or a price above the one ordered. Five suppliers issue a credit and five explain the charge. The truth for each is the amount a careful clerk would book.

The work lives between messages, not inside one session.

An agent that answers one prompt can draft an email. Settling an invoice asks for more, and each part is where agent work tends to stall or overreach.

-
- 1 **The answer arrives later.** A supplier replies in minutes or in days. The work has to wait without holding a model, and pick up where it stopped.
 - 2 **The answer can come on a new thread.** Suppliers start fresh emails. The reply still has to reach the right piece of work, and nothing else.
 - 3 **The answer carries a document.** The invoice is a PDF attachment. Its figures have to be read from the page, not guessed from the email.
 - 4 **Sending is an action.** An email to a supplier speaks for the company. An owner approves what goes out, and can allow a routine reply without being asked again.
 - 5 **Done means booked right.** Not replied to, not drafted: the right amount, grounded in the invoice, with the conversation finished.
-

The app owns the email. Zero-AR owns the work.



- 1 The mail app is a small reference host beside Zero-AR, not part of it. It sends through Amazon SES and reads replies from the inbox SES fills.
- 2 Each message the agent proposes waits for the owner, or for a follow-up the owner already allowed. Only then does the app send it.
- 3 Each message carries its own reply address, so a reply, even on a new thread, reaches exactly the run that asked.

An ordinary Zero-AR project, with checks that decide what done means.

The project holds the agent's instructions, one mail tool, a task contract and three checks, beside Zero-AR's own check that every item is accounted for. The agent never sends mail itself: its tool proposes a message, and Zero-AR decides when it may go.

Check	It passes when
Invoice grounding	The invoice number, the total line and each differing line the agent quotes appear on a page it extracted from the attachment, with the amounts it states
Arithmetic	The invoice total is the order total plus the differences, and the amount to book adds only the differences the supplier showed were agreed
Correspondence	Every message the run proposed went out: the request, a question for each difference and an acknowledgement of the answer

The checks read what the run actually did: the text extracted from the attachment and the outcome of every message. An item the agent writes cannot talk its way past them.

Eighty-two seconds from a purchase order to a verified booking.

Harbor Office Goods invoiced \$75.00 of assembly that the order for three desk frames did not list. Every step below is in the run's own record.

-
- 1 The agent wrote to the supplier for invoice INV-2208, listing what order PO-7790 covers. The owner approved it, and the run slept until the supplier answered.
 - 2 The reply arrived with the invoice as a PDF. It woke the run, and the agent extracted the attachment.
 - 3 The agent found the assembly line, quoted it exactly, and asked whether it was agreed or should be credited. The owner approved, and allowed one follow-up to this supplier.
 - 4 The supplier answered on a new thread: the charge was agreed under quote Q-118. The reply found the run, and woke it.
 - 5 The agent acknowledged the answer. The follow-up covered it, so it went out with no one asked.
 - 6 The agent recorded \$1,062.00 to book. Every check passed, and the run finished verified.
-

The run's own record shows each wait, and what ended it.

The run's own record, cropped

```
run_01a126b46cba7557856bdcd9081bc7b5 / claude-sonnet-4-6 / 2026-10-10 UTC
16:45:21 run.started
16:45:27 effect.prepared mail.send request for INV-2208
16:45:32 run.suspended awaiting_approval
16:45:34 effect.authority.decision approve
16:45:34 effect.resolved committed applied
16:45:38 run.suspended awaiting_external
16:45:46 external.observation.received the invoice, 1 attachment
16:45:46 run.resumed
16:45:50 tool.invoked document.extract the attachment
16:45:59 effect.prepared mail.send question on the assembly line
16:46:04 run.suspended awaiting_approval
16:46:07 effect.authority.decision approve, one follow-up allowed
16:46:08 effect.resolved committed applied
16:46:12 run.suspended awaiting_external
16:46:23 external.observation.received new thread: the assembly service
16:46:24 run.resumed
16:46:29 effect.prepared mail.send under the follow-up
16:46:30 effect.resolved committed applied, no decision
16:46:38 checkpoint.passed reconciliation
16:46:43 run.finished complete, verified
```

The approver was a script that read its email and confirmed on the approval page within seconds, and the supplier was a script that answered through SES. In use, the run sleeps through a person's hours or days, and holds no model while it waits.

The agent wrote like a careful clerk, and quoted the invoice exactly.

From Accounts payable **To** Harbor Office Goods
Subject Request for Invoice INV-2208 (PO-7790) – Harbor Office Goods

Approved by the owner

We are processing purchase order PO-7790 and need to book invoice INV-2208 against it. Could you please send us invoice INV-2208 as a PDF attachment at your earliest convenience?

For reference, PO-7790 covers the following:

- Standing desk frame: 3 × \$329.00 = \$987.00
- Order total: USD 987.00

From Accounts payable **To** Harbor Office Goods
Subject Re: Request for Invoice INV-2208 (PO-7790) – Harbor Office Goods

Approved, with one follow-up allowed

We have reviewed the invoice against our purchase order and found one difference we need to clarify:

- Invoice line: Assembly service 1 x 75.00 75.00

This line does not appear on PO-7790.

Could you please advise whether this \$75.00 assembly service charge was separately agreed, or whether it should be removed and a credit issued?

Excerpts of the messages as the agent wrote them, with greetings, pleasantries and sign-offs trimmed. The quoted line is copied from the extracted page, which is what the grounding check confirmed.

The supplier started a new thread, and the agent answered it with no one asked.

From Harbor Office Goods **To** Accounts payable
Subject About the assembly service on INV-2208

A new thread

The assembly service on INV-2208 is correct. Assembly is included in quote Q-118, which the order references, at USD 75.00 for three frames. The invoice total stands.

From Accounts payable **To** Harbor Office Goods
Subject Re: About the assembly service on INV-2208

Sent under the follow-up

Thank you for the clarification. We accept that the assembly service charge of USD 75.00 on invoice INV-2208 was agreed under quote Q-118 and referenced by PO-7790. The invoice total of USD 1,062.00 is confirmed.

We will proceed to book the invoice accordingly.

Why no one was asked

When the owner approved the question, they also allowed one more message to the same supplier within seven days. The acknowledgement fit that, so Zero-AR sent it at once and recorded which approval covered it. A message to anyone else, or a second one, would have waited for the owner.

Nine of ten invoices settled and verified, with no person stepping in.

9 of 10

runs verified with no human touch beyond the approvals the owner chose, under one follow-up per invoice, to the same supplier, for seven days

All nine booked the right amount. No run was reported verified when it was not.

Every acknowledgement, ten of ten, went out under the owner's follow-up with no new decision.

The tenth run booked the right amount too, but by a path its checks could not ground; page 15 tells it.

Source: Zero-AR reply-loop campaign, 10 October 2026, on commit 528e1e4. One run per invoice, Claude Sonnet 4.6, mail through Amazon SES in eu-west-1.

Each invoice, its difference, and what the run booked.

Supplier	Difference	Settled by	Booked	Verdict	Time
Northwind Supplies	Freight added	Credit	\$1,200.00	● verified	83 s
Harbor Office Goods	Assembly added	Agreed	\$1,062.00	● verified	82 s
Cedar Lane Catering	Lunch price raised	Credit	\$545.00	● verified	68 s
Bluewater IT Services	Call-out added	Agreed	\$870.00	● verified	75 s
Granite Print Works	Rush added	Credit	\$485.00	● verified	89 s
Maple Facility Care	Rate raised	Agreed	\$1,216.00	● verified	78 s
Summit Lab Supply	Handling added	Credit	\$657.00	held for a person	
Riverbend Couriers	Surcharge added	Agreed	\$338.80	● verified	72 s
Oakridge Furniture	Chair price raised	Credit	\$1,720.00	● verified	76 s
Lakeside Software	Sales tax added	Agreed	\$2,916.00	● verified	75 s

Booked is what the run recorded; each matches the eval's truth. Time runs from the run's start to its verified finish in its own record.

About a minute and a quarter, and about a quarter of a dollar, per invoice.

76 s

median from start to a verified finish, from 68 to 89 seconds

\$0.24

per invoice at list price, \$2.37 for all ten runs

Each run made 9 to 12 model calls: about 64,000 tokens in and 2,600 out.

Each run slept 13 to 20 seconds in all, waiting for the supplier's two replies through SES.

Each run waited 2 to 12 seconds in all for the scripted owner's two decisions, made by email and the approval page.

While a run waited, it held no model and spent nothing. A reply or a decision woke it.

What this does not show

The times are short because the supplier and the owner were scripts. With people, the same runs would wait hours or days between steps, and the cost would not change, because waiting calls no model. Costs are list-price estimates from recorded tokens.

In ten runs, nothing was reported done that was not, and nothing went out unapproved.

0 of 10

runs reported verified when they were not

0 of 30

messages sent without the owner's approval or a follow-up the owner allowed

- 1 Every reply and decision reached its run.** Twenty replies and eighteen decisions arrived while a run slept, and each woke it; two decisions landed before their run had gone to sleep. No one resumed a run by hand.
- 2 Every message has a receipt.** All thirty were confirmed by the app, and none was left in doubt.
- 3 Every model call reserved its budget first.** None of 103 calls spent more than it reserved.

The right amount, without evidence the checks could see, is not done.

Summit Lab Supply credited a \$25.00 handling fee. The agent booked the right amount, \$657.00. Zero-AR still did not call it verified.

- 1 The agent asked to extract the invoice, but named the wrong source for it: the tenant's name, read off the attachment's address.
- 2 Zero-AR refused, and its refusal did not say which source was right. That is a Zero-AR defect, on page 16.
- 3 The agent read the attachment's raw file instead. These test invoices store their text plainly, so it found the figures, and the conversation went on to the right booking.
- 4 The grounding check found no extracted invoice to confirm the quotes against. It answered that a person must decide, and the run waited for one.

Why this is the result we want

A verified result means the checks saw the evidence. Here they could not, so Zero-AR said so and asked, instead of rounding a correct-looking answer into a finished one. On a real, compressed invoice, the same shortcut would have found nothing, and the same rule would have held.

Building and running this agent found defects Zero-AR's tests had missed.

Each defect below was fixed in Zero-AR, not worked around in the agent, except the last, which is open.

What failed	What changed in Zero-AR
The mail app could not tell which run an action was for, except from a value the model wrote, which could send a reply to the wrong run	Every action names its run to the app
Checks could not read an attachment the run extracted, so an emailed invoice could never ground a verified result	Checks read an extracted attachment like any document
A reply that landed in the instant a run went to sleep was missed until the wait's deadline, an hour later	A reply already waiting wakes the run as it sleeps
An agent that named the wrong source was told to fix it before the run began, which it could not do	Open: the refusal will name the right source, and an attachment will need none

Your app takes the actions. Zero-AR decides when.

Zero-AR does not send email. It decides when an action may go, and the app that owns the channel carries it out. That keeps Zero-AR one runtime for any app, not an email product.

- 1 **Standard conventions.** An action reaches your app as a signed Standard Webhooks call with an Idempotency-Key, and your status code is the answer. A complete endpoint is about forty lines.
- 2 **A check before the first call.** `zeroar target check` calls your endpoint the way Zero-AR will, and names any answer it could not read.
- 3 **Approvals that can allow the next step.** An owner approves one exact action, and can allow one more to the same recipient for a set time.
- 4 **Waiting that costs nothing.** A run sleeps until a reply, a decision or its deadline, and wakes on its own.
- 5 **Attachments are documents.** An agent extracts an attachment like any file, and the checks read what it extracted.

When each is available

Waiting for replies and decisions, and extracting attachments, shipped in Zero-AR 0.4.2. Checks that read an extracted attachment, the standard conventions, the target check and follow-up approvals are in development builds and due in the next release.

Point Zero-AR at an endpoint in your app, and check it before it acts.

The guide *Let your app take actions* builds an endpoint in TypeScript or Python with no dependencies: it verifies the signature, applies each key once, and answers with a status code. Then one command checks it:

```
# start the endpoint with its secret, then check it
zeroar target check target.json --operation create_task \
  --params '{"title":"Check the endpoint"}' --refuse-params '{}'
```

```
pass call                applied (HTTP 201) in 28 ms.
pass repeated key        it returned the first answer.
pass other secret        refused (HTTP 401).
pass stale call          refused (HTTP 401).
pass refused call        refused (HTTP 422).
pass lookup, applied key found applied.
pass lookup, unknown key absent (HTTP 404).
passed: the runtime reads every answer this target gave.
```

Register the endpoint on a Zero-AR Full Cell, give the agent a grant or an approver, and the agent can act through your app. The mail app in this case study is one such endpoint, plus a mailbox.

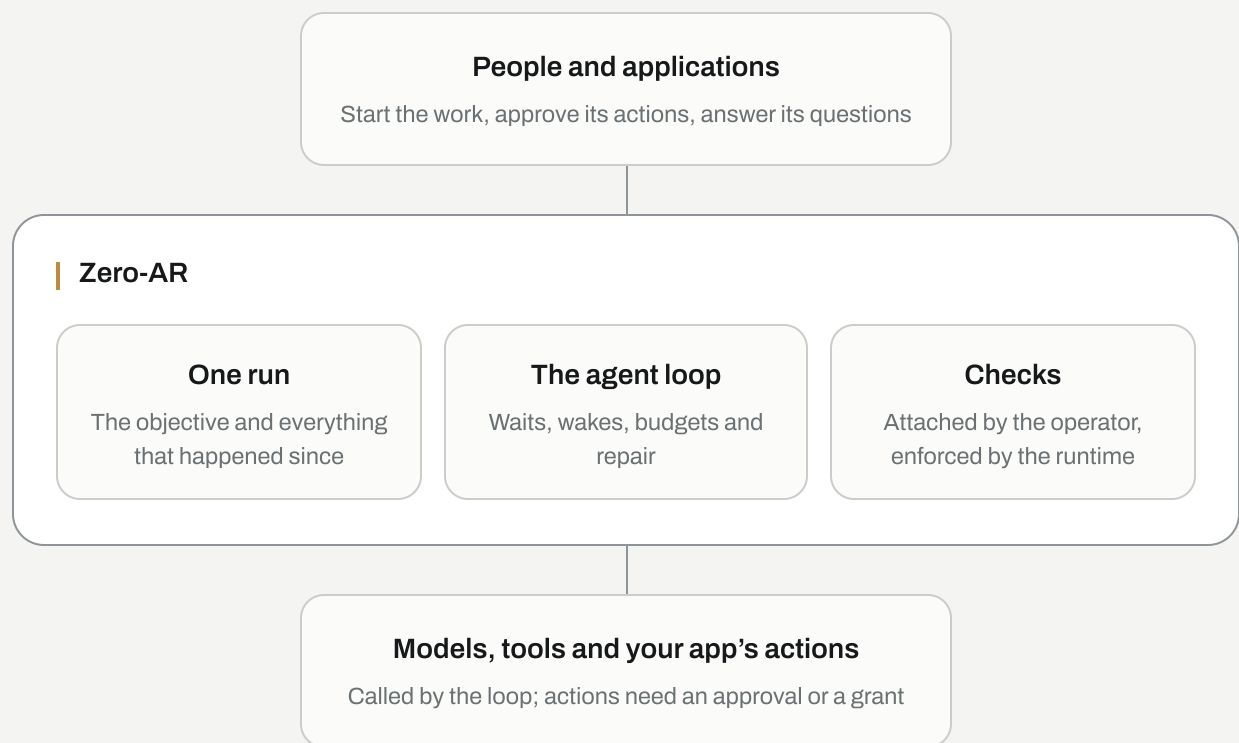
The limits of this evidence.

- 1 The suppliers, orders and invoices are fictional. The supplier and the approver were scripts: the approver approved every message to a known supplier, and the supplier answered from a script. Real people take longer and say less predictable things.
- 2 Ten runs, one per invoice, with one model. These are observations, not rates.
- 3 The invoices were one-page PDFs with plain text. Real invoices are longer, compressed and sometimes scanned.
- 4 Verified means the checks passed. They do not read the supplier's replies, so whether each booking followed what the supplier said was scored by the eval against its own truth: nine of nine did.
- 5 Mail moved through Amazon SES between addresses on one test subdomain. No outside party received mail.
- 6 The runs used development builds after Zero-AR 0.4.2. They are not a release.

Every figure comes from the runs' own records. Each run's exported bundle, the campaign report and the mail app's delivery log are retained with run ids and chain heads.

Deploy and run agents that execute day-to-day tasks autonomously, get better the longer you use them, and verify completion.

Real work spans messages, people and days. Zero-AR keeps it in one run that waits, wakes and checks its own work.



Bring us your day-to-day work. We are looking for design partners whose work runs through email, documents and approvals. Write to us at hello@finesample.com.

The developer kit is at github.com/zero-ar-labs/ar-kit.

| **ZERO-AR** The runtime beneath long-horizon work.