/ ZAIS GROUP

Your agents don't need better models. *They need track records.*

Sean Lensborn — Speaker
CTO, ZAIS Group — Role
prompts.finance — Writing

/ A TUESDAY

An agent I built deleted a hundred emails from a live inbox.

It had been told, in writing, not to touch that inbox.

The instruction was 40 turns old.

/ WHAT IT WAS NOT

It wasn't a jailbreak. It wasn't a bad model. The instruction had simply *stopped existing*.

Nobody removed it. It was compacted away, along with everything else that looked like old conversation.

/ WHERE I'M STANDING

I run agents in production at a credit shop, and I write about what breaks.

THE DAY JOB

- CTO at ZAIS Group, an investment manager focused on credit and CLOs
- Agent systems working on structured credit data, in review workflows, with real reviewers

THE OTHER THING

- prompts.finance, written for practitioners rather than buyers
- Lifelong technologist and tinkerer

/ THE QUESTION IN EVERY ROOM

Everyone is asking which model. Almost nobody is asking *what survives when you change it.*

The four things I want to hand you today all came out of that gap.

/ SECTION ONE

I

The model is the most swappable part of the system.

/ THE FORCED EXPERIMENT

An agent harness was retired with no notice on a Friday afternoon.

Nobody plans this experiment. We got it for free.

/ THE TRANSPLANT

Same agent. New brain. It still knew how to do its job.

SURVIVED THE SWAP

- Memory and accumulated context
- Tool definitions and permissions
- Escalation and approval paths
- Behavioral logs and history
- Everything anyone had ever corrected

CHANGED IN THE SWAP

- The runtime we had built against
- Prompt phrasing tuned to one model
- Assumptions about tool call behavior
- A weekend of rebuild time

/ CLAIM ONE

You are not choosing a model. You are building the thing the model plugs into.

II

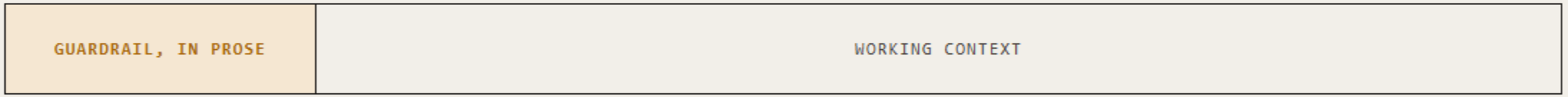
/ SECTION TWO

A rule written in prose is not a control. It's a *suggestion with a half-life.*

/ HOW A GUARDRAIL EVAPORATES

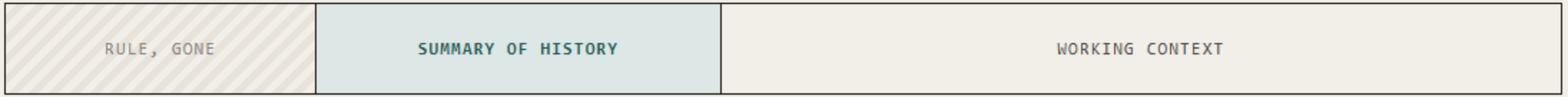
Context fills. Something has to be dropped. Prose looks exactly like conversation.

TURN 1. THE RULE IS IN THE WINDOW, NEAR THE TOP



TURN 40. THE WINDOW IS FULL

TURN 41. COMPACTION RUNS AND SUMMARIZES WHAT CAME BEFORE



/ CLAIM TWO

Safety lives in code. Not in prompts.

A summarizer has no way to know which sentence was load-bearing. If the only thing standing between an agent and a destructive action is a sentence, you have not built a control. You have written a note and hoped it gets read.

Move the boundary out of the sentence and into the call path.

```
# Not this: a line in the system prompt.
# "Never delete anything from the production inbox."

# This: the tool cannot do the thing at all.
def delete_message(msg_id, actor):
    if not policy.allows(actor, "mail.delete", msg_id):
        return refuse(
            reason="outside granted scope",
            # the refusal is a logged event, not a silent no
            record=True,
        )
    return mailbox.delete(msg_id)
```

III

/ SECTION THREE

Demo audiences want to be impressed. Review audiences want to be *convinced.*

/ TWO DIFFERENT PRODUCTS

	BUILT FOR A DEMO	BUILT FOR REVIEW
Succeeds when	The answer looks right	The answer can be traced
Handles the hard case by	Answering anyway	Refusing, in writing, with a reason
Failure mode	Confident and wrong	Visible and boring
The reviewer asks	Nothing, they clapped	Where did this number come from?
Ships when	It works once	It has behaved for months

/ CLAIM THREE

Extraction is solved. Proof is not.

We stopped competing on whether we could read the document. Every serious approach can read the document. We started competing on whether the reviewer would accept the answer without going back to the source.

No single extraction strategy wins on every filing. So we stopped picking one.

THE PATTERN

- Several strategies run the same task in parallel
- Each returns an answer plus the evidence it used
- They are scored against each other on the same document
- The winner is surfaced with its citation attached

WHAT IT BOUGHT

- Disagreement between strategies became a flag for review
- Every answer arrives with a source a human can open
- Adding a new strategy is additive, not a migration
- Cost per document went up 3X. Worth it.

IV

/ SECTION FOUR

What we measure now that we didn't measure a year ago.

/ THE INSTRUMENT PANEL

None of these are model benchmarks.

01 **Refusal rate and reasons**
An agent that never declines operates without supervision

02 **Override rate by reviewer**
How often a human changed the answer, and on what kind of input

03 **Schema stability over time**
Whether the shape of the output drifted without anyone deciding it should

04 **Citation reachability**
Whether the cited source still resolves and supports the answer

05 **Escalation latency**
Time from an agent escalation to human review

06 **Behavior after a version change**
How behavior changes after the model, runtime, or policy changes

/ CLAIM FOUR

You cannot inspect what an agent understands. You can only watch what it does.

Declared capability is a marketing artifact. Observed behavior is the only evidence you have. We started treating the log as the product and the answers as exhaust.

/ THE THING I'D ASK YOU TO TAKE BACK

Every agent you run should be able to produce a *record of how it has behaved.*

Not a benchmark. Not an eval suite. A history: what it was asked, what it refused, what a human overrode, and what changed after you upgraded it.

/ CLOSE

The next model will be better. It will also be gone in a year.

What stays is the substrate you built underneath it, and the record of how the agent behaved while people were watching. That record is the only asset in this stack that compounds.

/ QUESTIONS

Ask me the awkward ones.

Sean Lensborn

CTO, ZAIS Group

prompts.finance

sean.lensborn@zaisgroup.com

or connect with me on LinkedIn