When the agent is talked past

Version 0.1 · revised 4 September 2026 · this version as a PDF

> Drafted with AI assistance, then checked and revised by the author. The judgements and the errors are the author's.

Status of these claims

What this publication does not claim, and what is outstanding against it in the register.

A question this rests on is parked: How is the aggregation ceiling actually enforced?

> We claim that we do not compose and retain no capability to compose. We do not claim that composition by others is impossible.

Why most AI safety depends on the agent's cooperation, and what to build that does not.

Series part I · Version 0.1 · August 2026

Before you read this

What actually stops an agent doing something it should not? Most answers to this describe instructions given to the agent. Instructions are requests.

What happens when the instruction does not hold? Not through malice — through a long context, an unusual phrasing, a chain of individually reasonable steps, or a document containing text written to be read by an agent rather than a person.

Is there anything that holds regardless? Yes, and it is a different kind of thing from an instruction.

And how would you know which kind you have? There is a single test, and it takes one sentence to apply.

The layer everyone builds

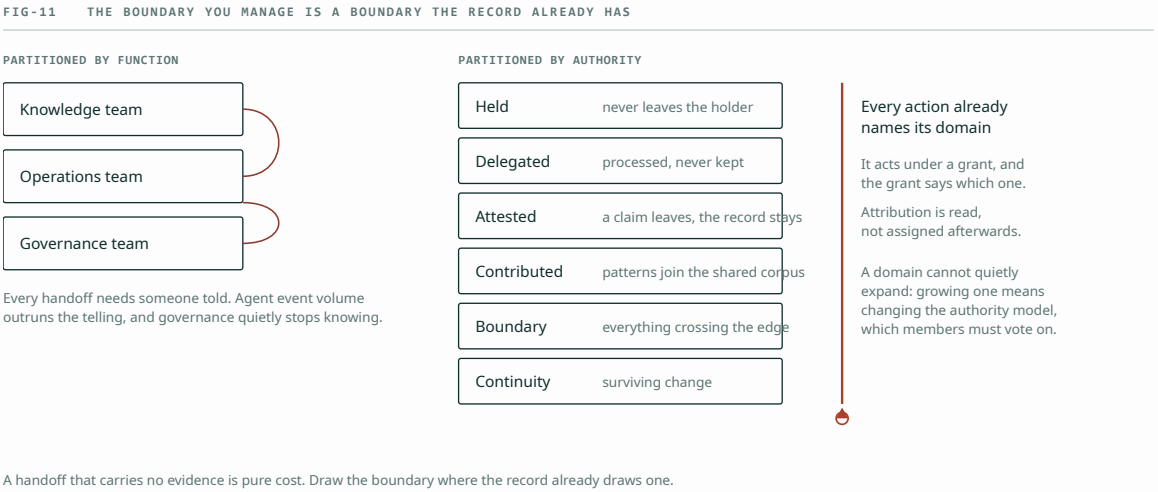

FIG-11 Functional partitions require operations to tell governance what happened, and agent event volume outruns the telling. Authority partitions put the boundary where the record already has one.

Ask how an agentic system is kept safe and you will hear about the layer that sits at the point of action. Before the agent does something consequential, something checks: is this permitted, does it look like the kind of action we expect, should a person confirm it.

That layer is worth building. It catches a great deal, it catches it early, and it produces a record of what was attempted as well as what was done. Nothing here argues against it.

But be clear about what it is. **It is a judgement made about an action, using information the agent supplied, in a context the agent influenced.** It works when the system behaves as expected.

The ways it is talked past

None of these requires anyone to act in bad faith.

Instructions arriving inside content. An agent reads a document, a web page, a support ticket, an email. Text in that material is written to be read as instruction rather than as content. The agent has no reliable way to tell the difference, because to the agent both are simply text that arrived.

Reasonable steps that sum to something else. Each action is permitted. The sequence is not, and nothing was watching the sequence.

The unanticipated phrasing. A constraint written to catch one shape of request does not catch a different shape of the same request. This is not a bug to be fixed; it is the permanent condition of describing what is not allowed in language.

Drift over a long interaction. Context accumulates, framing shifts, and the thing being asked at step forty is not the thing that was being asked at step one.

And plain error. The agent misreads the situation and acts confidently. No adversary required.

The common thread: every one of these is a failure of *the agent's understanding of its situation*. Any protection that also depends on the agent's understanding of its situation fails at the same moment, for the same reason. **You cannot check a judgement with another judgement made from the same information.**

The test

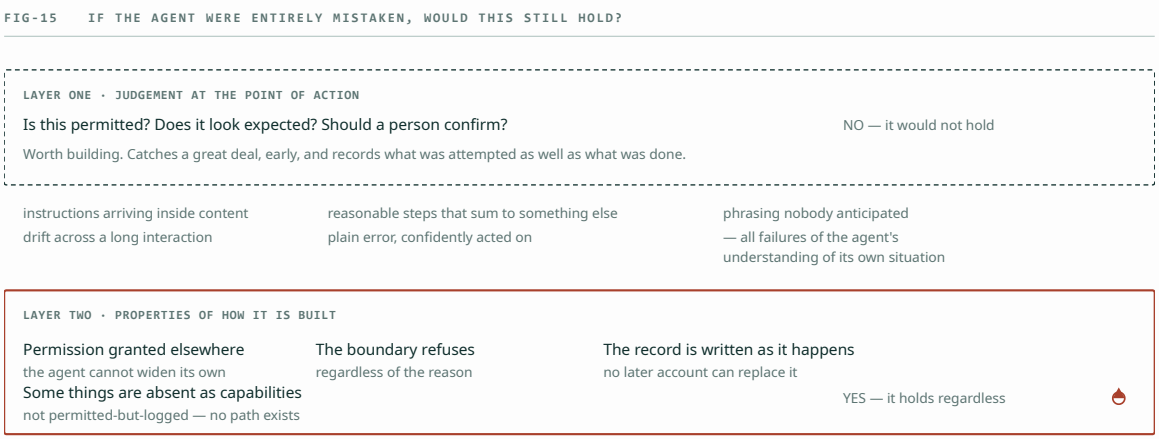

FIG-15 Layer one is a judgement made using information the agent supplied. Layer two is a set of properties that hold regardless of what the agent understood.

One sentence, and it sorts every safety mechanism into two piles:

> **If the agent were entirely mistaken about what it was doing, would this still hold?**

Instructions fail it. Prompts fail it. Policies fail it. Guidance fails it. Most guardrails fail it, because they are the agent checking itself with more of the agent.

What passes: a permission the agent cannot grant itself. A network boundary that refuses a connection regardless of why it was attempted. A record that is signed at the moment of writing, so that no later account can replace it. An operation the system is not able to perform, as against one it has been told not to.

What the second layer is

Not a better set of rules. **A set of properties that hold because of how the system is built, not because of what anything decided.**

Four of them do most of the work.

Permission is granted elsewhere. An agent acts inside a grant of authority issued by someone else, or it does not act. The agent cannot widen its own grant, cannot infer a wider one, and cannot be argued into one, because the check is not performed by the agent — the enforcement point simply refuses. Where two grants conflict, the more restrictive one applies, with no override for anybody, including the board. An override is a door, and a door will be used.

The boundary refuses. Outbound connections are denied by default and only enumerated destinations are permitted. An agent that has decided, for excellent reasons, that it must reach an external service does not reach it. The refusal happens at a layer that has no access to the agent's reasoning and would not be moved by it.

The record is written as it happens. Every action is signed and timestamped when it occurs, against the key of the domain it belongs to. There is no later reconstruction to be influenced, because the account was made at the time by something that was not narrating.

Some things cannot be done at all. Assembling a person from records they did not send is the clearest example. Not permitted-but-logged. Not requiring-approval. Absent as a capability — no path exists. A system that retains the ability and declines to use it will eventually use it, because the person who declines is replaced and the reason is forgotten.

Why this is the whole argument

Say it plainly: **the first layer's guarantees are conditional on the agent behaving as expected. The second layer's guarantees are not.**

That difference is what you are actually buying when you choose one arrangement over another, and it is almost never what is being sold. A vendor telling you their agents are carefully constrained is describing the first layer. It is real, it is valuable, and it is a promise about behaviour.

The question to ask instead is: **when your agent is wrong, what happens to my records?**

If the answer describes what the agent has been instructed to do, the answer is that nothing happens to protect them. If the answer describes something the agent cannot do — a permission it cannot obtain, a boundary that refuses, a record that already exists and cannot be revised — then the protection survives the failure.

Most systems in this market are built almost entirely at the first layer. Not through negligence: the second layer requires decisions about where data lives and who holds authority over it that most commercial arrangements cannot make, because they conflict with the business model. A vendor whose revenue depends on holding your records cannot build a system that structurally prevents them from using your records.

Where the claim stops

Four, stated because a claim this strong attracts scrutiny and should.

The floor is not infinite. The hardware, its firmware and the operating system beneath all of this are trusted. Nothing here defends against a compromise below the level where these properties are enforced, and no arrangement anywhere does.

Withdrawal is bounded, not instant. Where a pattern derived from your records has entered a shared body of knowledge and been used to adapt a shared model, it cannot be extracted from that model. What can be committed to is removal from the corpus immediately and rebuilding within a published maximum time. That is a bound, not a reversal, and anyone claiming otherwise is wrong.

Assembly by others is not prevented. A system can be built so that it does not compose a person and retains no capability to. It cannot stop a third party assembling from published claims and public sources. The claim is about what this system does, not about what is possible in the world.

And the second layer only holds if it is actually enforced there. A permission checked by asking the agent nicely is an instruction wearing different clothes. Every property above has to be implemented at a point the agent cannot reach, and whether it has been is a question about a specific system, not about an approach.

How you would know this is wrong

Three observations would falsify the claim for any given system:

1. **An agent obtaining a permission it was not granted** — by any route, including persuading a person to grant it under time pressure.
2. **A record whose account of an action was written after the action**, rather than at it.
3. **An outbound connection succeeding to a destination not on the permitted list**, for any reason the system found compelling.

Each is testable. If you are assessing a system rather than building one, ask for a demonstration of all three failing to occur — and treat a description in place of a demonstration as the answer.

Cite this version

```
When the agent is talked past, version 0.1 (September 2026).
agenticgovernance.digital/downloads/when-the-agent-is-talked-past-v0-1.pdf
```


PREVIOUS
H · Nine changes government could make

Series contents

NEXT
J · Why most AI policy changes nothing