



SERIES

THE RECORD • EIGHT PAPERS • OPEN LICENCE

The Delegation *Record.*

Working notes on what a switch in the
request path can attest

PAPERS

PAPER 001	The model you asked for, and the one that answered	RESEARCH • 2 MIN
PAPER 002	Two documented routes out, and only one passes a switch	ARCHITECTURE • 3 MIN
PAPER 003	The field naming the model is filled in at the far end	RESEARCH • 2 MIN
PAPER 004	What a switch reading plaintext cannot settle	LIMITS • 3 MIN
PAPER 005	The split is prior art; the seam is what these four leave open	RESEARCH • 3 MIN
PAPER 006	A saving is not one count minus another	CONSTRUCTION • 3 MIN
PAPER 007	A switch attests what it decided, not that it decided well	CONSTRUCTION • 2 MIN
PAPER 008	What these notes cannot do, and what would settle it	LIMITS • 3 MIN

ISSUED

eight papers • published 13 September 2026 • CC BY 4.0
digital1.foundation/articles/the-delegation-record/

Contents

001	The model you asked for, and the one that <i>answered</i>	4
	RESEARCH • 2 MIN • PUBLISHED 13 SEPTEMBER 2026	
	Routing a step to a model from another provider is a solved engineering problem and an unsolved evidence problem. What a record written at that seam can and cannot show.	
	References	5
002	Two documented routes out, and only one <i>passes a switch</i>	6
	ARCHITECTURE • 3 MIN • PUBLISHED 13 SEPTEMBER 2026	
	The route that works cleanly leaves nothing in the path to observe; the route a switch can observe is documented as unsupported for this purpose.	
	References	8
003	The field naming the model is filled in <i>at the far end</i>	9
	RESEARCH • 2 MIN • PUBLISHED 13 SEPTEMBER 2026	
	The one field naming the model is returned by the party that did the choosing. The requester states a preference; nothing in between is in a position to check.	
	References	10
004	What a switch reading plaintext <i>cannot settle</i>	11
	LIMITS • 3 MIN • PUBLISHED 13 SEPTEMBER 2026	
	Four reasons hold the sentence up, one boundary limits it, and one open problem names the capture that would turn the reasoning into evidence.	
	References	13
005	The split is prior art; the seam is what <i>these four leave open</i>	14
	RESEARCH • 3 MIN • PUBLISHED 13 SEPTEMBER 2026	
	The split is prior art. The seam a cross-provider route opens in it, with the field an operator most wants on the far side of the boundary, is not.	
	References	15
006	A saving is not <i>one count minus another</i>	16
	CONSTRUCTION • 3 MIN • PUBLISHED 13 SEPTEMBER 2026	
	Two model names kept apart are the only substitution signal such a record carries on its own. Counts subtracted across a provider boundary have no referent.	
	References	17

007	A switch attests what it decided, not that it decided well CONSTRUCTION · 2 MIN · PUBLISHED 13 SEPTEMBER 2026 An attested decision establishes what was decided: not that the answer came from the model the route named, and not that the decision was sound. References??	18
008	What these notes cannot do, and what would settle it LIMITS · 3 MIN · PUBLISHED 13 SEPTEMBER 2026 A record that labels each field attested or received makes its position legible to a reader who was not there. That is worth doing, and it is less than a proof. References??	20
END	Colophon SOURCES AND THEIR HASHES, LICENCE, AND HOW THIS DOCUMENT IS SET	22

The model you asked for, and the one that *answered.*

An operator routes a step to a model from another provider because it would rather not pay for the harness's default model on steps it has judged do not need it. The justification is a saving, and a saving is shown by a record of which model did the work and what it cost. Which model answered is the first thing a component reading plaintext in the request path cannot establish from its own position, and what the call was billed is the second. These notes set out what such a record can attest, and what it takes on trust.

PAPER · 001 PUBLISHED · 13 SEP 2026 READING · 2 MIN

LICENCE · CC BY 4.0

Canonical · digital1.foundation/articles/the-delegation-record/001-the-model-you-asked-for.html

The step is ordinary: classify this, pull those fields out, retry a call that failed on a transient error. An operator running a harness at volume would rather not pay for its default model on the steps it has decided do not need it, and routes those to a model from another provider. Whether to route them belongs to whoever builds the system, and these notes take none of those decisions.

A saving is a claim about two quantities: what the step cost where it ran, and what it would have cost where it did not. The second is counterfactual, and no record of the step that ran recovers it. The first arrives from the party that ran the step, counted with a counter nobody on this side can open.

Routing costs the conversation something too: a thinking block "is readable only by the model that produced it or a newer one, and the API ignores or drops the blocks the target model can't read".^[1]

THE TENSION, STATED ONCE

The reason to route is a saving. The evidence for a saving is a pair of facts — which model answered, and what it counted — that a component in the request path takes delivery of rather than establishes. A record presenting that pair as findings is stronger than its own position; a record presenting it as testimony, with the testifier named, is exactly as strong as its position, and can be relied on for what it is.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product. It supplies no evidence here: every quotation above is from a published page a reader can fetch, cited with the date it was read, and the Gateward repository is not public, so nothing in it is a claim a reader can check today. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

References

- 1 Anthropic, *Extended thinking*, Claude Developer Platform documentation, on thinking-block signatures and on passing thinking blocks between models, platform.claude.com/docs/en/build-with-claude/thinking, read 12 September 2026.

Two documented routes out, and only one *passes a switch*.

A step can reach another provider's model by two documented routes. One leaves the request path altogether; **the other passes a switch, and is documented as unsupported for this purpose**. This paper sets out what each route leaves in front of a component that could write a record.

PAPER • 002 PUBLISHED • 13 SEP 2026 READING • 3 MIN

LICENCE • CC BY 4.0

Canonical • digital1.foundation/articles/the-delegation-record/002-two-documented-routes-out.html

The first route is ordinary code: a tool handler inside the harness calls another provider's API and returns a tool result, so nothing in the request path sees a model call. The documented fields of a `tool_result` block are `type`, `tool_use_id`, `content` (optional) and `is_error` (optional), with computer-use and browser-use member results also echoing `toolset_name`; none of those fields names a model on that page, read on 12 September 2026, and an absence where one looked is not an absence in general.^[1] The component positioned to write both names down is the handler.

The second route is a base URL pointed at a gateway, which a switch can watch because the request goes through it. Anthropic's documentation is explicit: "Any gateway that exposes a supported API format works. Anthropic doesn't endorse, maintain, or audit third-party gateway products, and doesn't support routing Claude Code to non-Claude models through any gateway."^[2] That is a statement of support policy, not a claim that the harness detects such routing.

Behind such a gateway the model string is carried rather than checked: "On Amazon Bedrock, Google Cloud's Agent Platform, Microsoft Foundry, Claude Platform on AWS, and behind an LLM gateway or a custom `ANTHROPIC_BASE_URL`, your provider or gateway defines the model names, so Claude Code passes any string through without checking it."^[3] Two bounds travel with it: an unrecognised string is rejected with `Model "<name>" is not a recognized model id.` and the session keeps its

current model, and the unrecognized-model diagnostic can still be written at request time, on every provider.^[3] A string the gateway defines is not an unrecognized string, which is why both sentences hold together.

THE SENTENCE THIS PAPER TURNS ON

" ANTHROPIC_BASE_URL changes where requests are sent, not which model answers them."^[3] A switch sits exactly where the sending happens, so the first half is its own knowledge: it dialled the host. The second half arrives in the response, and the response is the other party's account of itself.

That is the bind: the route that works cleanly leaves nothing in the path to observe, and the route a switch can observe is documented as unsupported for this purpose.

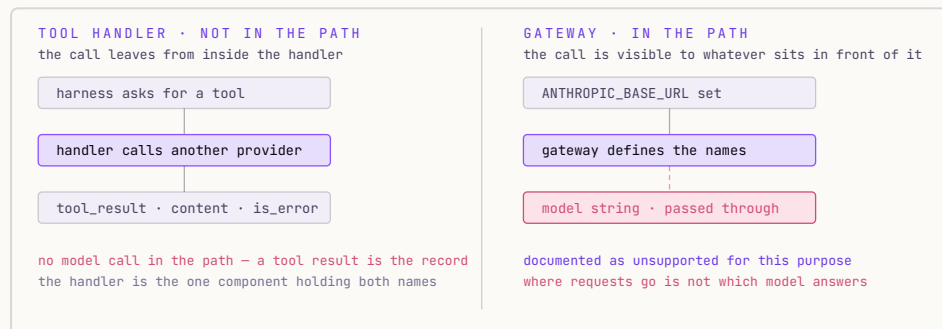


FIGURE 1 The same step, by two documented routes. On the left the call leaves the harness inside a tool handler, so what the path records is a tool result; on the right it stays in the path and reaches a gateway that defines the model names itself. Absent an attestation from the party that ran the model, neither route places an object in the path that a component reading plaintext could use to establish which model produced the answer.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside *Gateward*, a Digital One product. It supplies no evidence here: every quotation above is from a published page a reader can fetch, cited with the date it was read, and the Gateward repository is not public, so nothing in it is a claim a reader can check today. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

References

- 1 Anthropic, **Handling tool calls**, Claude Developer Platform documentation; the fields of a `tool_result` block, platform.claude.com/docs/en/agents-and-tools/tool-use/handle-tool-calls, read 12 September 2026. The enumeration is of that page on that date; an absence where one looked is not an absence in general.
- 2 Anthropic, **LLM gateway configuration**, Claude Code documentation; the quoted sentence is a statement of support policy, code.claude.com/docs/en/llm-gateway, read 12 September 2026.
- 3 Anthropic, **Model configuration**, Claude Code documentation, on provider-defined model names and `ANTHROPIC_BASE_URL`, code.claude.com/docs/en/model-config, read 12 September 2026.

The field naming the model is filled in *at the far end*.

Whichever route a step takes, the field that names the model is filled in by the party that did the choosing. This paper tabulates what each route gives a record, and follows the one protocol that wrote that seam down and is now retiring it.

PAPER · 003

PUBLISHED · 13 SEP 2026

READING · 2 MIN

LICENCE · CC BY 4.0

Canonical · digital1.foundation/articles/the-delegation-record/003-the-field-naming-the-model.html

A per-step model name composed with a gateway alias — where a harness lets an individual step name the model it wants — is a hypothesis rather than a result: nothing here exercised it and no capture of it exists, which is why both of its cells below are empty of findings.

ROUTE	WHAT THE REQUEST PATH SEES	WHAT THE RECORD GETS
Tool handler	A tool call and a tool result. No model call crosses the path the harness watches.	Whatever the handler writes down; the rest of the path sees a tool result.
Gateway base URL	A model call, in the format the harness speaks.	Two model strings — one sent, one returned — and usage in the units the upstream defines.
Per-step name + alias	Hypothesis. Untested here.	Hypothesis. Untested here.
MCP sampling	A sampling request from a server, answered by the client.	The client's chosen model, named by the client in the result it returns.

A protocol that wrote the seam down, and is retiring it

The Model Context Protocol's sampling feature let a server ask a client to run a completion, and the client may satisfy a preference with something else: clients "MAY map hints to equivalent models from different providers".^[1] The one field naming the model — `model` , "the name of the

model that generated the message" — is returned by the party that did the choosing.^[1] The requester states a preference; the responder reports what it decided; nothing in between is in a position to check.

Sampling is now deprecated. As of revision 2026-07-28 (SEP-2577) it "remains in the specification for at least twelve months after this revision's release before it becomes eligible for removal", new implementations "SHOULD NOT" adopt it, and existing ones "SHOULD migrate to integrating directly with LLM provider APIs".^[1] Direct integration is the left-hand route of Figure 1 in Paper 002 — the one that leaves nothing in the request path to observe.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product. It supplies no evidence here: every quotation above is from a published page a reader can fetch, cited with the date it was read, and the Gateward repository is not public, so nothing in it is a claim a reader can check today. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

References

- 1 Model Context Protocol, **Sampling**, specification revision 2026-07-28 : the deprecation notice (SEP-2577), the model-hint mapping clause and the `model` field, modelcontextprotocol.io/specification/2026-07-28/client/sampling, read 12 September 2026.

What a switch reading plaintext *cannot settle*.

A switch in the request path forwards a response and takes delivery of the model name inside it. This paper sets out **what that position can and cannot settle**: four reasons, where the sentence stops, and the capture that would turn the reasoning into evidence.

PAPER • 004 PUBLISHED • 13 SEP 2026 READING • 3 MIN
LICENCE • CC BY 4.0

Canonical • digital1.foundation/articles/the-delegation-record/004-what-a-switch-cannot-settle.html

Absent an attestation from the party that ran the model, no component reading plaintext in the request path can establish which model produced a response. The model name inside the bytes such a component forwarded and received is a string the upstream chose to send. Four reasons hold the sentence up, and each shows the scope doing work rather than hedging.

On the call it is forwarding, a component in the request path chooses none of its inputs. It forwards a tenant's prompts; it does not select them. Whatever could be established by selecting inputs is unavailable to a component that selects none of its inputs.

It cannot re-execute the work of a model whose weights it does not hold. Where it does hold them, a second run under different serving conditions settles nothing about the first.

An attestation made at the far end does not reach this component in the bytes it handles. Whatever the party that ran the model established about the machine it ran on, a component reading plaintext takes delivery of a response, not of that party's evidence about its own platform; these notes checked no attestation specification and make no claim about what one covers. A construction that does attest which model answered binds the workload as well, which is why it is a different construction rather than a counterexample.

The one cryptographic object in the response is verifiable only by its issuer. Anthropic documents that "full thinking content is encrypted and returned in the `signature` field on each thinking block"; that "the API uses the signature to verify that thinking blocks were generated by Claude when you pass them back"; and that "the `signature` field is opaque: don't interpret or parse it".^[1] A value verifiable by its issuer is evidence for that issuer, not for an intermediary holding it.

WHERE THIS SENTENCE STOPS

The limit above is a statement about a position — plaintext, in the request path — and not a statement about what is knowable in general. Moving the question to a party that can attest the execution itself changes the answer, and that is a different construction from the one these notes describe. A record built on the unqualified form of that sentence is built on something false.

OPEN PROBLEM · IDENTIFIED WHILE ASSEMBLING THIS PAPER, 12 SEPTEMBER 2026

No capture made on this workstation distinguishes a substitution from an honest answer

Everything above is reasoning over published documentation. What would turn it into evidence is a capture: a request path exercised end to end with a recorded substitution in it, and the bytes each component saw, saved and hashed before one of them is quoted. No such capture was produced for these notes. The shape that would produce a real one is narrow — a local logging proxy in front of a stub upstream, driven until the upstream rejects a configuration it cannot serve, which yields a real status line and real bytes rather than a described one.

Two questions would then be answerable that this paper leaves open: what a harness writes down when the returned name differs from the requested one, and whether a diagnostic emitted at request time reaches any durable artefact on the operator's side. Both are hypotheses until a capture exists, and are named here as hypotheses.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product. It supplies no evidence here: every quotation above is from a published page a reader can fetch, cited with the date it was read, and the Gateward repository is not public, so nothing in it is a claim a reader can check today. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

References

- 1 Anthropic, **Extended thinking**, Claude Developer Platform documentation, on thinking-block signatures and on passing thinking blocks between models, platform.claude.com/docs/en/build-with-claude/thinking, read 12 September 2026.

The split is prior art; the seam is what *these four leave open*.

Separating what a party knew first-hand from what it was told is not new. SLSA, RATS, SCITT and NISTIR 8112 each draw that line, and none settles what a cross-provider route does to it: that seam is what these notes offer.

PAPER • 005 PUBLISHED • 13 SEP 2026 READING • 3 MIN

LICENCE • CC BY 4.0

Canonical • digital1.foundation/articles/the-delegation-record/005-the-split-is-prior-art.html

The distinction between what a party knew first-hand and what it copied from somebody else's claim did not start here. Four bodies of work draw it, each read on 12 September 2026.

SLSA draws the line normatively at field level inside one attestation: `externalParameters` are "the external interface to the build. In SLSA, these values are untrusted; they MUST be included in the provenance and MUST be verified downstream", while `internalParameters` are set by the platform and "are trusted because the platform is trusted".^[1]

RATS supplies the roles: an Attester is "a role performed by an entity (typically a device) whose Evidence must be appraised in order to infer the extent to which the Attester is considered trustworthy".^[2] SCITT states the residue: registering a Statement "only proves it was produced by an Issuer".^[3]

NISTIR 8112 supplies the part most often got wrong, including in the note that started this work. It does not draw a single line between self-asserted and verified attributes. It defines **Pedigree** — "description of the attribute value's relationship to the authoritative source of the value" — with the values Authoritative, Sourced, Self-Asserted and Derived, where "Authoritative" means "the attribute's value was acquired directly from the source of authority" and "Self-Asserted" means "the value was provided to the AP directly by the individual with whom the attribute value is associated". Verification is a separate axis, and the document says so: "Self-asserted attributes may also be verified or unverified."^[4] Two

questions rather than one, which is the better precedent: a delegation record needs both where a field came from and whether anything checked it.

What none of them settles is what a cross-provider route does to that split: the field an operator most wants sits on the far side of the boundary in both documented routes. The seam is what none of the four settles, and it is the contribution offered here — a statement about four documents read on one day, not a search of the literature.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product. It supplies no evidence here: every quotation above is from a published page a reader can fetch, cited with the date it was read, and the Gateward repository is not public, so nothing in it is a claim a reader can check today. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

References

- 1 SLSA, **Provenance**, version 1.0, the Model section and the `BuildDefinition` field table, slsa.dev/spec/v1.0/provenance, read 12 September 2026.
- 2 H. Birkholz, D. Thaler, M. Richardson, N. Smith, W. Pan, **Remote Attestation procedureS (RATS) Architecture**, RFC 9334, January 2023, §4.1 and §4.2, rfc-editor.org/rfc/rfc9334, read 12 September 2026.
- 3 **An Architecture for Trustworthy and Transparent Digital Supply Chains** (SCITT), RFC 9943, June 2026, §9.2, rfc-editor.org/rfc/rfc9943, read 12 September 2026.
- 4 NIST, **Attribute Metadata: A Proposed Schema for Evaluating Federated Attributes**, NISTIR 8112, January 2018, DOI 10.6028/NIST.IR.8112, §3.2.1.1 "Provenance Metadata" — Table 6 and the numbered list of recommended values below it, which give the Pedigree element four: Authoritative, Sourced, Self-Asserted, Derived. "Origin" is a separate element on that same table and a recommended value of the Verifier element at §3.2.1.2, Table 7; the front-matter summary table runs its Values column continuously across rows, which is where the two are run together. nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8112.pdf, read 12 September 2026; the PDF's text was extracted locally in order to read it.

A saving is not *one count minus another*.

A saving is shown by a record, and a record is only as strong as what its writer could know first-hand. This paper proposes a shape that **keeps the attested fields apart from the received ones**, and names the two traps in the cost half: the unit, and the cumulative stream.

PAPER · 006 PUBLISHED · 13 SEP 2026 READING · 3 MIN

LICENCE · CC BY 4.0

Canonical · digital1.foundation/articles/the-delegation-record/006-a-saving-is-not-one-count.html

A shape for a record, organised by what its writer was in a position to know.

```
// one delegated step, split by what the writer could know first-hand

attested  time · tenant · key.id      // it authenticated the caller
          model.requested            // this side chose and sent this string
          route.chosen               the switch decided it
          policy.id · policy.ver
          rule.or.score · version    // named and versioned, not the value alone
          refused · reason           no upstream saw this call at all
          endpoint.dialled           // which host, not where inference ran
          bytes.sent · bytes.back

received  model.returned              the upstream's account of itself
          usage.* + counted_by        upstream's counter, upstream's unit
          thinking.signature          verifiable by its issuer, opaque here

// what no field in this record establishes
"the model that answered" // not a field this side is in a position to attest
what the call was billed  // the count is the upstream's, in the upstream's unit
a total from summed deltas // the deltas are cumulative: the sum overstates

// the two model fields are never one field: their disagreement is the
// only substitution signal this record carries on its own
```

Collapsing the two names destroys the one disagreement such a record can surface by itself; kept apart, a substitution shows as two strings that differ, and two that agree still record only what the upstream said.

Then the cost half, where two traps sit. The first is the unit: counts on either side of a provider boundary come from different tokenizers under different rules about what is billable, so subtracting one from the other

produces a number with no referent. This record has published a counting error of its own and printed the correction beside it: in [Paper 001](#) of The Text Layer a published ratio had been computed with another vendor's tokenizer standing in for the vendor's own counter on both sides of it, alongside a per-page constant that counted the document's words twice.^[1] Naming the party that counted each figure does not make either checkable; it stops the two being added.

The second trap is in the stream. Anthropic's streaming documentation warns that "the token counts shown in the `usage` field of the `message_delta` event are **cumulative**".^[2] A switch that adds up what goes past records a running sum of running sums, so an operator proving a saving that way overstates the very spend the routing was meant to reduce.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product. It supplies no evidence here: every quotation above is from a published page a reader can fetch, cited with the date it was read, and the Gateward repository is not public, so nothing in it is a claim a reader can check today. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

References

- 1 The Text Layer, Paper 001 — What the model reads: §5, the proxy tokenizer and the per-page constant, the two counting defects referenced in this paper.
- 2 Anthropic, **Streaming Messages**, Claude Developer Platform documentation, the warning that `message_delta` usage counts are cumulative, platform.claude.com/docs/en/build-with-claude/streaming, read 12 September 2026.

A switch attests what it decided, not that it *decided well*.

Most of what a switch writes down about a delegated step, it received rather than knew. One thing it knew first-hand is **its own routing decision**. This paper says what attesting that decision establishes, and the two things it does not.

PAPER • 007 PUBLISHED • 13 SEP 2026 READING • 2 MIN
LICENCE • CC BY 4.0

Canonical • digital1.foundation/articles/the-delegation-record/007-a-switch-attests-what-it-decided.html

A series stopping at [Paper 006](#) would have described a hole. There is an affirmative, and it is the strongest field available in this position: a switch that routes by policy can attest the decision, because it made it. Which route it chose. Under which policy, at which version. On which rule or score. At what time. And what it refused — a refusal being the one event no upstream saw at all.

An attested decision establishes what was decided. It does not establish that the decision was carried out by the party it was addressed to, and it does not establish that the answer which came back was produced by the model the route named. A reader who takes "the routing decision is attestable" to mean "the routing is verified end to end" has kept one clause and dropped the rest of the sentence.

Score-driven routing — a switch selecting a route from live measurements — is a research direction rather than a described system. The switch can attest that it routed on a named score, at a named version, at a given time, because it did. Whether the decision was **sound** is a different question, and no record of a decision settles it: it is answerable from a published method someone other than its author can re-run, or from outcomes measured on a named corpus — neither of which a record of the decision supplies. That holds for any scoring system whose method is unpublished, including one built by whoever writes the switch, and is why no score appears as a figure here.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product, and the direction in this paper was raised by **frontierscore.ai**, another Digital One property. Neither supplies evidence here. The Gateward repository is not public, and the second property's public pages link no methodology — no test design, no prompt set, no calculation — read 12 September 2026, which is why this paper treats a published method as the condition. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

What these notes cannot do, and what *would settle it*.

The limits of these notes, stated before anyone else states them: no measurement, no failing input, no conclusion about permission, and no escape from self-attestation. Then the four questions it leaves open, and what would close each one.

PAPER • 008 PUBLISHED • 13 SEP 2026 READING • 3 MIN
LICENCE • CC BY 4.0

Canonical • digital1.foundation/articles/the-delegation-record/008-what-these-notes-cannot-do.html

They carry no measurement. An earlier pass produced figures of the right shape and did not keep its artefacts, and no repository of the switch this work came from exists on the machine these notes were written on — so no per-call cost, no latency claim, no token reconciliation. A number without the artefact it came from is a press release with a decimal point.

They carry no failing input. The route that matters most here — a tool handler calling out of band — produces no model call in the path, so the thing that would need capturing is the thing that is structurally absent. What [Paper 004](#) carries instead is an open problem, the form with no failing input by definition.

They reach no conclusion about permission. Two things are routinely confused: a support-policy sentence in documentation states what a vendor will help with, and silence in an instrument establishes that the instrument does not address the act — never that it permits it.

And they do not escape self-attestation. The operator signs its own record from its own position, which is the epistemic position the model string occupies in [Paper 004](#). The gain is not certainty. It is that a reader can tell, field by field, which sentences their author could have known first-hand, and direct doubt at the right half rather than at the whole document. Labelling each field as attested or received does not lift the record out of that position; it makes the position legible to a reader who

was not there. That is worth doing and it is less than a proof, and a paper that let the labelling stand in for the proof would have committed the error [Paper 004](#) spends itself avoiding.

And the open items, each with what would settle it and who decides:

- 1 Whether a two-name field pair catches a substitution in practice. Settled by the capture described in [Paper 004](#); two agreeing names neither settle it nor falsify it. Whoever builds the switch decides.
- 2 Whether counts from two providers can be put on any common footing. Settled by measurement on a named corpus with both counters and both conventions named; no such figure exists here.
- 3 Whether a switch's own decision record is worth anything to a reader who does not trust the operator. Settled by making it checkable against an earlier commitment rather than the operator's word — the subject of [the earlier work in this record](#): a record shows what was done, not that it was right.
- 4 Whether anything short of an attestation from the party that ran the model can raise the model name above a received string. Nothing read for these notes on 12 September 2026 does, which is a statement about what was read rather than a proof that nothing does.

Nothing above is decided here: the record shape in [Paper 006](#) is a proposal.

ON THE PROVENANCE OF THIS MATERIAL

The question these notes work on came out of building a routing switch inside **Gateward**, a Digital One product, and the direction in [Paper 007](#) was raised by [frontierscore.ai](#), another Digital One property. Neither supplies evidence here: every quotation in these notes is from a published page a reader can fetch, cited with the date it was read. The Gateward repository is not public, so nothing in it is a claim a reader can check today, and the earlier measurement pass that lost its artefacts is why no figure from it is printed. The second property's public pages link no methodology — no test design, no prompt set, no calculation — read 12 September 2026, which is why [Paper 007](#) treats a published method as the condition. Deliberately absent: prices, plans and tiers, availability, and deployment internals.

Colophon

A

The record

This document is generated from the published pages, which are the record. Where this file and a page disagree, the page is right and this file is stale. Each source below is named with the SHA-256 of the file this document was built from, so a reader can check that what they are holding is what was published; the generator fetches each live page as it builds and reports any that has moved since.

INDEX	digital1.foundation/articles/the-delegation-record/
001	digital1.foundation/articles/the-delegation-record/001-the-model-you-asked-for.html sha256 8ce60950122833c51f2b5e8876956c68a55e2f3af332d5edbece0cc01dbcec72
002	digital1.foundation/articles/the-delegation-record/002-two-documented-routes-out.html sha256 0ad0d3189a997af894f7b6357340b9b9f96cd8549755d3aa59f1e086287867a4
003	digital1.foundation/articles/the-delegation-record/003-the-field-naming-the-model.html sha256 f634a8879c0c77ca3978cb6f437d779ff844be509abbd767afcbbb3053c18170
004	digital1.foundation/articles/the-delegation-record/004-what-a-switch-cannot-settle.html sha256 6f89a8459eb51f2cb5df117db7f809bf437db33357938e2a362842748d6bf99c
005	digital1.foundation/articles/the-delegation-record/005-the-split-is-prior-art.html sha256 4439599744579ef7ee72674160ff134b2fd45e97f91bc4e9162c883b0ceb7262
006	digital1.foundation/articles/the-delegation-record/006-a-saving-is-not-one-count.html sha256 76411e879ea7de1acb8a14b5d6ac0928b3555cdda9b1e3b4599a97cac067a515
007	digital1.foundation/articles/the-delegation-record/007-a-switch-attests-what-it-decided.html sha256 1f93447a2bccf20aa2102fb1b5abc9f8dad88f13257933728160f61bd1b21058
008	digital1.foundation/articles/the-delegation-record/008-what-these-notes-cannot-do.html sha256 bd1edc37845e7302495f09e759a4c34b7465db04f7a52448dbcd6ce871d09400

B

Licence

Every paper in this document is published under the Creative Commons Attribution 4.0 International licence, creativecommons.org/licenses/by/4.0/. You may copy, redistribute and

adapt it for any purpose, including commercially, provided you credit **Stichting Digital One Foundation i.o.** and link to the page cited above.



How it is set

In the six faces the site itself carries and embeds: Fraunces 300 and 600 for titles and the emphasised word, Manrope 400 and 500 for the text, JetBrains Mono 400 for the ledger's own labels, code and running matter. Those six are Latin subsets and do not carry every character the papers use. Nothing is drawn in their place and quietly dropped from the text: Every character in every paper was extracted back out of this file and checked before it was written, so the whole document copies, searches and cites as it reads.

Figures are the papers' own inline drawings, re-toned from the site's night ground to this page by a one-to-one swap of the closed house palette, so green still means it holds and rose still means it breaks — and the two are separated in weight of grey as well as in hue, so a photocopy keeps the distinction. The rail down the left of each page is the ledger from the Foundation's homepage; a filled dot marks each numbered section, and the § number sits in the gutter, outside the measure.

The sheet is A4 as the page asks for it, which Chromium rounds to 594.96 × 841.92 pt when it writes the file — 0.1 mm narrower and 0.03 mm taller than the 595.28 × 841.89 pt of the standard. No printer will show it; a document that prints its own hashes should say it anyway.

© MMXXVI Stichting Digital One Foundation i.o.
Amsterdam, the Netherlands

Published under CC BY 4.0.
No public-benefit (ANBI) status is claimed at this time.

digital1.foundation
The Delegation Record