



THE LAB · ENTRY 01 · FINDING

The grant is user-shaped, *not data-shaped.*

Connect an assistant to your mailbox and the narrowest permission that lets it read one message lets it read every message. Connect it to your drive and the default search corpus is, in the publisher's own words, files *owned by or shared to* you. There is no supported way to say *my files, except the folder the legal team shared with me*. The unit of restriction is the application. It is never the data.

Edition v1 · 12 September 2026

Live page riskmandate.ai/lab-connector-grants.html

Published by RiskMandate · CC BY 4.0

This is a dated edition of a page that changes. The Lab holds our current thinking, and current thinking moves. This PDF does not: it is what we thought on the date stamped below, kept so the reasoning can be followed rather than only its conclusion. The live page may since have been corrected, extended or withdrawn — and if it has, the edition list on it will say so.

Four scopes, in their publishers' own words.

Each of these was read on 12 September 2026 from the linked page, and quoted rather than paraphrased. Nothing below required an account, a test or a request to anybody's system.

“View your email messages and settings.”

The description of `gmail.readonly` – the narrowest Gmail scope that returns a message body. The only scope that excludes bodies is `gmail.metadata`, described as “view your email message metadata such as labels and headers, but not the email body”, and it cannot read a message. There is no scope that filters by sender, label or date. · [developers.google.com – Gmail API scopes](https://developers.google.com/gmail/api/scopes)

“Files owned by or shared to the user.”

Google's definition of the user corpus – and corpora defaults to user. So on day one, the default search over a Drive grant spans everything any colleague, client or counterparty has ever shared with that person, without anybody choosing it. · [developers.google.com – Drive files.list](https://developers.google.com/drive/api/v3/about-files)

“Site-specific permissioning (using *.Selected permissions) is not supported because the underlying search is tenant-wide.”

Anthropic's security guide for its own Microsoft 365 connector, describing why the narrowing mechanism Microsoft provides cannot be used. The same page states that shared-mailbox access is read-only via `Mail.Read.Shared`, and that the connector uses delegated permissions so a user reaches only data they already have permission for. · [support.claude.com – Microsoft 365 connector security guide](https://support.claude.com/microsoft-365-connector-security-guide)

`account_info.read` · `files.metadata.read` · `files.content.read` ·
`files.content.write` · `sharing.read` · `sharing.write` · `file_requests.read` ·
`file_requests.write`

The eight scopes requested by the official Dropbox MCP server. Two of them are write scopes and two are sharing scopes. There is no folder-scoped variant. · [help.dropbox.com – connect the Dropbox MCP server](https://help.dropbox.com/develop/mcp-server)

The unit of restriction is the application and the tool. It is never the data.

Every lever these products offer turns a capability on or off for an assistant. None of them narrows *which material* that capability reaches. That is the finding, and it is why a behaviour policy for a connector shape is a different document from one for a coding agent: the question stops being *how far can it reach* and becomes *whose material is in reach*.

THE FAIR READING

Nobody is exceeding **your own access**.

This is the objection a vendor would raise first, it is correct, and stating it makes the finding stronger rather than weaker.

- **A delegated connector reaches what you can already reach, and no more.** Anthropic's own guide says so in as many words: users can only access data they already have permission for, and cannot bypass sharing settings or folder permissions. No privilege is being escalated.
- **The problem is not escalation. It is that there is no floor.** Your own access is the *only* boundary on offer. If your mailbox contains a client's confidential correspondence — and it does — then so does the grant, and no setting in any of these products separates the two.
- **Which is exactly what the barrier model already says.** A switch the agent's own account can flip is not a control. A connector toggle you can turn back on in one click is a setting, not a boundary — and the only real boundary available is revoking the whole connector. **The four barriers.**
- **And the default posture is permissive.** On an unconfigured business domain at the largest provider, third-party application access is allowed until an administrator changes it, so a person can connect an assistant to their work mailbox with nobody asked.

THE DEMONSTRATION

In four places, the advertised capability and **the granted scope disagree**.

These are not accusations. Marketing copy and scope lists are written by different people at every company on earth and nothing reconciles them. It is, however, precisely the condition a

behaviour policy exists to surface — so this table is the product's value in one figure.

WHAT IS ADVERTISED	WHAT THE SCOPES IN THE GRANT PERMIT	STATE
Share, move and trash files	The grant's two file scopes are a read-only scope and a per-file scope. Neither authorises acting on a pre-existing file the assistant did not create	unresolved
Create, update and delete calendar events	The publisher's own scope list for that service, as granted, is read-only	unresolved
Label and unlabel mail threads	Neither scope in the published grant authorises label mutation. A <code>gmail.labels</code> scope exists and does permit it — it is not in the grant	unresolved
Search files	Whether the shared-drive parameters are set is undocumented, so whether team drives are in the corpus in practice cannot be determined from the pages	undocumented

These stay unresolved, on purpose.

Each could be settled in an afternoon by connecting an assistant and trying it. We will not: probing somebody else's system to find out what it does is out of bounds here, with no research exemption. An unresolved contradiction, sourced to both of a vendor's own pages and dated, is worth more than a resolved one obtained by poking. If a vendor tells us which page is right, this table changes and says so.

Sources for the table, all read 12 September 2026: [Gmail scopes](#) · [Drive scope classifications](#) · [shared-drive parameters](#) · [the official server's scope list](#) · and the connector descriptions those scope lists sit behind.

WHAT THE MODEL IS MISSING

Reach answers how far. It does not answer whose.

The published capability grammar gives every primitive a reach — `project` , `host` , `tenant` , `world` , `self` — and an `undo` class. Neither says anything about who the material belongs to, and for a connector shape that is the only interesting question.

- `read.record.mailbox` would say the agent can read a mailbox. It would not say that the mailbox contains correspondence from clients who never agreed to any of this.

PROPOSED PROPERTY	VALUES	MEANING
material	OWN ORGANISATION THIRD_PARTY MIXED	Whose material the capability reaches

Almost every connector capability is mixed , and that is the finding.

A mailbox is mixed. A shared drive is mixed. A personal notes folder is own . The value of the property is that mixed **cannot be made own by any setting any of these vendors offers** — which is the user-shaped finding expressed as data rather than as a paragraph, and therefore computable.

This is a proposal against somebody else's published data, not a change we can make. It is written up as a request, with its evidence, at [Lab 03](#).

WHICH LAW REACHES FIRST

Data protection, before the AI regulation.

- A mailbox is mostly other people's writing. A shared folder is mostly other people's files. So a connector grant is, by construction, a grant over material that was entrusted rather than owned — and that is a provision, not an analogy.
- **The sub-delegation provision is one sentence.** A processor shall not engage another processor without the prior written authorisation of the controller, and the first remains liable for the second. Connecting an assistant to a mailbox of client correspondence is that sentence, in the form a stranger already recognises.
 - **Every organisation already has somebody accountable for data protection.** Almost none has anybody accountable for the AI regulation — whose deployer obligations were deferred to 2 December 2027 and reach only systems in the high-risk annex. Mapping to the instrument that already has an owner is the difference between a finding somebody acts on and one they file.
 - **And here is the honest gap.** No regulator has addressed head-on the case of third-party personal data sitting in somebody's mailbox being disclosed to an assistant by that person's own consent. The nearest published hook is a national regulator's January 2026 assessment, which warns about systems connected to databases not needed for their task and flags the difficulty of assigning controller and processor roles through the supply chain. **That is a hook, not a ruling**, and any policy we publish cites it as exactly that.

OPEN

What we cannot answer yet.

Real questions, published so nobody has to rediscover them. If you know the answer to one of these, that is the most useful thing you could send us.

- **A connector that exists and is switched off** is not in the grant today and is one click from being in it. Neither the barrier model nor the label has a place for that state, and it is probably the most common state in any real estate.
- **Does material belong in the shared capability data or in the policy?** It is a property of a capability *in a context*, which argues for the policy. It is reusable across every policy of the same shape, which argues for the data.
- **What does the mandate elicitation look like for a connector?** For a coding agent the mandate is a job. For a mailbox it is closer to a relationship, and nobody has drafted those questions.
- **Who replies if a vendor disputes one of the four contradictions?** The sourcing is theirs and the publication is ours.

If this is your product and we have it wrong, tell us and this page changes, with a line saying what changed and when. We quoted your documentation rather than testing your service precisely so that this conversation could be a correction rather than a dispute.

• Lab 01

This is what the first five policies are for.

Five deployment shapes, each one a document that makes the sentence above visible for a setup somebody actually runs. The flow that produces them — and what buying one would look like — is drawn next door.

```
}());
```



THE LAB · ENTRY 02 · MOCKUP

From a stranger's first question to *a key they can hand an underwriter.*

Twelve stages. The first four need no account, no integration and no access to anybody's environment — they are five documents and a page. The rest is one engineering build, and it is the same build every time, which is the only reason any of this could be a business.

Edition v1 · 12 September 2026

Live page riskmandate.ai/lab-abp-flow.html

Published by RiskMandate · CC BY 4.0

This is a dated edition of a page that changes. The Lab holds our current thinking, and current thinking moves. This PDF does not: it is what we thought on the date stamped below, kept so the reasoning can be followed rather than only its conclusion. The live page may since have been corrected, extended or withdrawn — and if it has, the edition list on it will say so.

Twelve stages. Four of them are documents.

Stages one to four are the whole sales motion and they are shippable now. Stages six to ten are a single build — a template vault that gets cloned per customer — and that build is the product even though nothing on this site sells it.

STAGE	WHAT HAPPENS	STATE
1 Encounter	A stranger reads the question and recognises themselves in it	ships now
2 Self-select	Which of these five shapes is yours	ships now
3 Draft	The pre-computed policy for that shape. Free, instant, no account	ships now
4 Correct	They tell us where it is wrong. The conversion and the elicitation are the same move	ships now
5 Purchase	On the store, against the tiers that already exist	rail exists
6 Provision	Clone the template vault, seed it with the corrected draft	needs the template
7 Elicit the mandate	The only part that genuinely needs a person, or a good enough question set	needs the questions
8 Measure the grant	From the shape, the credentials and the connectors actually enabled	needs the library
9 Compute	Delta, label, leaflet. All derived, none authored	needs the computation
10 Deliver	The vault, and a read key they can share with anybody	needs the template
11 Live	Recompute when an input moves. The history is the business case	needs a receiver
12 Uplift	The twin, the score, the standards mapping — signed by somebody who did not sell stages one to ten	later

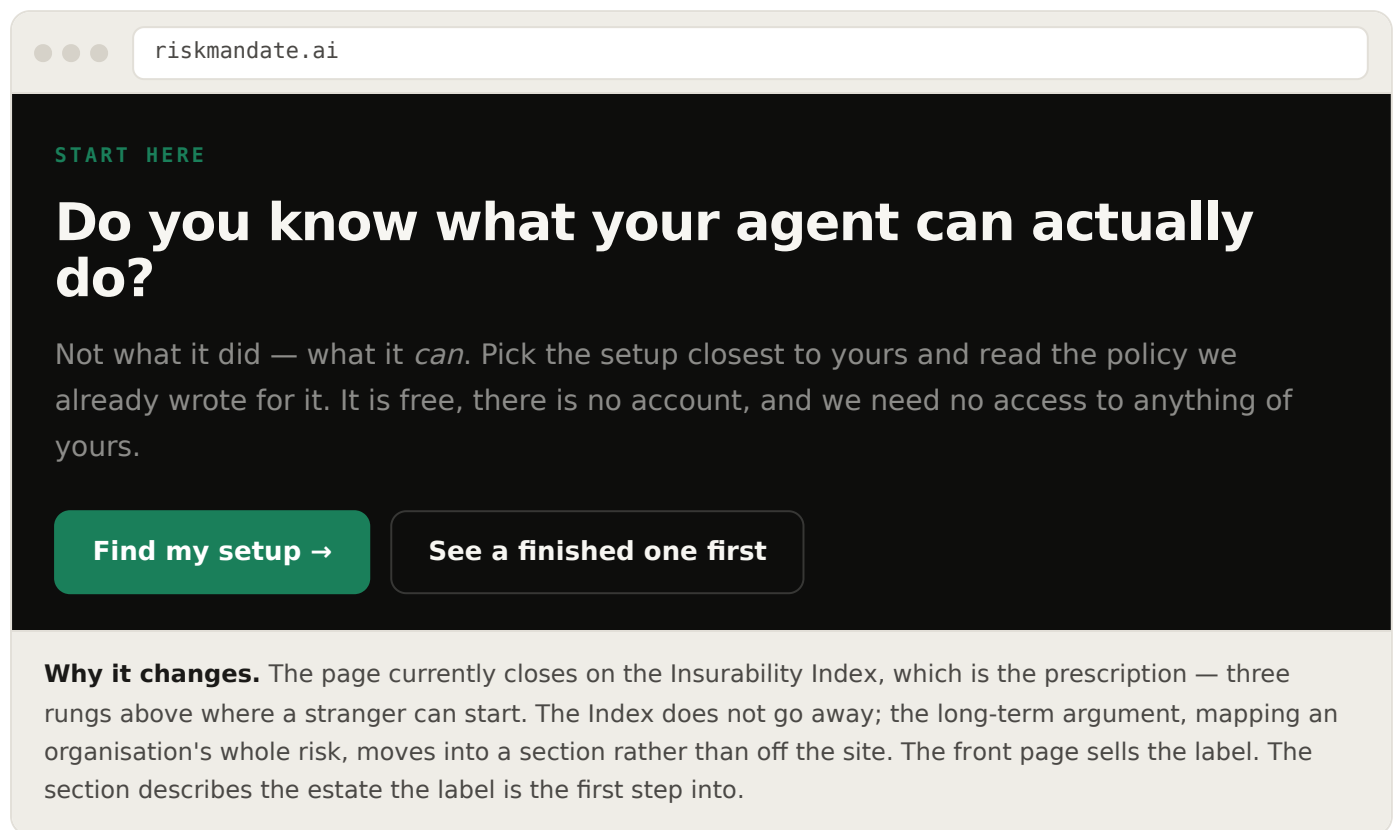
The first policy will be made by hand. If the second one costs the same, there is no business.

So the thing being built is not a document and not even a vault — it is the **template** plus the **shape library**. The marginal cost of a policy is the cost of its shape, and shapes are reused. The target is explicit: the second policy of a shape already in the library should take minutes, not days. If it does not, the library is not working and nothing should be priced yet.

MOCKUP 1 · STAGE 1

Encounter. The question, one example, the link.

A panel on this site's front page, in the place currently held by the Index. The sequence is: do you know what your agents can do → here is one → buy one. The question and the examples belong to the free library; the offer belongs here.



MOCKUP 2 · STAGE 2

Self-select. Five shapes, and one of them is yours.

Five, not fifty. Build the coding agent first because it is already in the published data and proves the method with no new research. Then the mailbox and the personal drive, which a stranger

recognises. Then the work mailbox and the corporate file estate, which are the ones somebody buys.

STEP 1 OF 2

Which of these is closest to yours?

Close is close enough. The draft is about a deployment *shape*, not about your estate — you will correct it in a minute, and the correcting is the useful part.

An assistant connected to my personal mailbox

Gmail, Outlook, or similar, connected to a chat assistant

► draft ready · 2 min read

An assistant connected to my work mailbox

The same grant, a different consequence — shared mail is in scope

draft ready · 3 min read

An assistant connected to my personal cloud drive

Drive, Dropbox, or similar

draft ready · 2 min read

An assistant connected to our corporate file estate

SharePoint, a tenant-wide search, an enterprise connector

draft ready · 4 min read

A coding agent on a developer machine

With confirmation prompts on, or off — we drafted both

draft ready · 2 min read

None of these

Tell us what you run and we will say whether we can draft it

a conversation

The honest bit. “None of these” is a real option and it goes to a person, because a shape we have not built is days of work and pretending otherwise would waste everybody's time. The five that are ready say so; nothing here implies a sixth exists.



The draft. Free, instant, and deliberately understated.

This is the screen everything else exists to reach. Four numbers, then the capability rows with what stands in the way and whose material each one touches, then the table no template can carry. Every row says where it came from.

DRAFT BEHAVIOUR POLICY · NOT YOURS YET

An assistant connected to a personal mailbox

Derived from vendor documentation read on 12 September 2026, not from your account. **We have assumed a conservative mandate.** If the numbers below look wrong, they are — and telling us how is the next step.

GRANT	MANDATE	EXCESS	UNBOUNDED
7	2	5	5

CAPABILITY		BARRIER	MATERIAL	IN THE MANDATE
read.message.tenant		expectation — a rule in prose	MIXED	yes
read.record.history		none	MIXED	no
send.message.world		setting — you can flip it back	ORGANISATION	yes
read.credential.host		none	THIRD_PARTY	no
send.endpoint.world		none	MIXED	no
create.record.world		expectation — a rule in prose	MIXED	no
read.record.browsing		setting — you can flip it back	MIXED	no

Five of seven capabilities are MIXED or THIRD_PARTY. That is not a setting you have wrong. There is no supported way to say *my inbox, except messages from outside the company* — the unit of restriction is the connector, not the correspondence.

WHERE THE VENDOR'S OWN PAGES DISAGREE	ADVERTISED	GRANTED
label mutation	label and unlabel mail threads	no scope in the grant authorises it · unresolved
calendar writes	create, update and delete events	the granted scope list is read-only · unresolved

PROVENANCE

7 of 7 capability rows **derived** from published documentation, 0 measured. Mandate assumed, not elicited. Delta computed from both, pinned to the versions shown. Valid for the deployment shape described, as at 12 September 2026 — *if the risk changed, the deployment changed, not this document*. No score appears anywhere in this document and none will.

This is wrong in places — let me correct it

Download as it stands

Understated on purpose. The assumed mandate is generous to the reader: it credits them with having authorised more than they probably did, so the correction goes *upward*. That is the moment somebody realises the grant is bigger than they thought — the same mechanic as the permission game, on a page instead of in a quiz. **And no score, anywhere.** Four counts, a barrier per row, and whose material it touches. The number a buyer can actually move is *unbounded*, and it only moves when a real control appears.

MOCKUP 4 · STAGE 4

Correct. The conversion and the elicitation are one move.

Nobody needs telling what they asked the agent to do — they need telling what it can do. So the draft asserts the mandate and asks to be corrected, and correcting it *is* stating the mandate. This is the stage nobody has rehearsed and the one most likely to be wrong.

STEP 2 OF 2 · ABOUT 90 SECONDS

Which of these did you actually ask it to do?

We guessed two. Everything you add moves a row out of the excess and into the mandate — and everything you leave is authority nobody scoped.

Read your mail so it can answer questions about it –
`read.message.tenant`

☒ asked for☐ no

Draft and send replies on your behalf – `send.message.world`

☒ asked for☐ no

Read every message anyone has ever sent you, including attachments
– `read.record.history`

☐ asked for☒ no

Reach any host on the internet while doing it – `send.endpoint.world`

☐ asked for☒ no

Read credentials stored where it runs – `read.credential.host`

☐ asked for☒ no

You authorised 2 of 7. The other five are in the grant regardless, and none of them has a barrier above the connector — so the only lever that removes any of them is disconnecting it.
Your corrected policy is yours to keep from £10, as a file with your answers, the measured grant, and the delta between them.

[Make it mine — £10](#)[Just email me the draft](#)

Two things this must not do. It must not tell anybody their setup is dangerous — a policy describes and does not judge, and the same grant is fine in one deployment and serious in another. And it must send nothing anywhere: this is a form about a deployment shape, not a probe of the reader's account, which is what makes it legal to run with a stranger in ninety seconds.

MOCKUP 5 · STAGE 10

Deliver. A key, not an attachment.

What is sold is a clone of a template vault, seeded with the customer's own corrected draft. Nothing in it is authored except the mandate — the only thing the customer knows and we do not. And the read key is the part that is worth more than it sounds.

Northgate Financial — assistant, work mailbox

mandate.json	elicited from you – the only authored file
grant.json	measured · 4 of 11 rows, 7 derived
delta.json	derived · inputs pinned
label.md	computed
leaflet.md	computed
history/	3 recomputes since 12 Sept
validity.md	as at 12 September 2026

[illegible]

Publish it publicly

BEFORE ANYTHING IS PRICED

Instrument the first five, and publish the table.

Time each one. Count how many rows were derived rather than measured, how many questions had to go to a human, and which stage was slowest. That table is the only pricing input anybody will have, and publishing it is the same discipline as everything else here.

- **Build five first** — the coding agent — because it is already in the published data and proves the method with no new research.
- **Then one and three**, the personal mailbox and the personal drive, because a stranger recognises them.
- **Then two and four**, the work mailbox and the corporate file estate, because those are the ones somebody buys.
- **Every one is derived, not measured.** The rows come from vendor documentation read on a date, not from observation — and the provenance line says so, in the same place the published capability map says 21 of 99 rows were measured.

Open, and it decides the front page: which of the five becomes the worked example a stranger meets first? Two and four are what somebody buys. One and three are what somebody understands. We think one, and we are not confident.

• Lab 02

Tell us which screen is wrong.

These are drawings. Changing them costs an afternoon now and a rewrite later, so this is the cheapest moment this flow will ever be. Stage four is the one we are least sure of.

}());



THE LAB · ENTRY 03 · PROPOSAL

What we need from *the model site*.

The Agent Behaviour Policy model, its capability grammar and its data live at **abp.sgit.ai**, which is a separate site with a separate maintainer. We render against it. These are the three things we need in order to ship the flow next door — published here rather than emailed, so the reasoning is checkable and the answer can be public too.

Edition v1 · 12 September 2026

Live page riskmandate.ai/lab-abp-requests.html

Published by RiskMandate · CC BY 4.0

This is a dated edition of a page that changes. The Lab holds our current thinking, and current thinking moves. This PDF does not: it is what we thought on the date stamped below, kept so the reasoning can be followed rather than only its conclusion. The live page may since have been corrected, extended or withdrawn — and if it has, the edition list on it will say so.

A material property, because reach does not answer whose.

The 23 primitives each carry a reach — project , host , tenant , world , self — and an undo class. For a connector shape that is not the interesting question. read.record.mailbox says the agent can read a mailbox. It does not say the mailbox is full of other people's correspondence.

PROPERTY	VALUES	MEANING
material	OWN ORGANISATION THIRD_PARTY MIXED	Whose material the capability reaches

- **A property, not a fourth dimension.** A fourth element multiplies the grammar; keeping the model simple matters more than expressiveness here. One property on an existing primitive is the minimum change that makes the connector shapes renderable.
- **The value of it is that mixed is a dead end.** A mailbox is mixed. A shared drive is mixed. A personal notes folder is own . And mixed cannot be made own by any setting any of the four vendors we read offers — so the property turns a paragraph of argument into something computable.
- **Evidence is in Lab 01,** with four verbatim quotes from vendor documentation and their URLs, read 12 September 2026.
- **One open question we cannot settle for you.** material is a property of a capability in a context, which argues for it living in a policy rather than in the shared vocabulary. It is also identical across every policy of the same shape, which argues for the vocabulary. We lean towards the vocabulary with a per-policy override, but it is your model.

Four connector shapes, published like the first five.

There are five worked examples today and they are all about where an agent runs. The four below are about whose material it reaches, which is the stronger argument and the one a stranger recognises. Same treatment as the existing five: derived from published data, each stating how many rows were measured.

#	SHAPE	WHAT IT DEMONSTRATES	PRIORITY
1	Assistant connected to a personal mailbox	The narrowest scope that reads a message reads every message	first
2	Assistant connected to a work mailbox	Shared mail is in scope, and no administrator was asked	third
3	Assistant connected to a personal cloud drive	The default corpus is files owned by <i>or shared to</i> the user	second
4	Assistant connected to a corporate file estate	Site-specific narrowing is unsupported because the search is tenant-wide	fourth

- **Every one is derived, not measured.** The rows come from documentation read on a date, not from observation, and nothing may be tested — so the provenance line matters more here than in the existing five.
- **Each needs a section the current examples do not have:** where the vendor's advertised capability and their granted scope disagree, sourced to both of their pages, dated, and published *unresolved*. Four such contradictions are already written up in Lab 01. That section is the thing no competitor's template can carry.
- **Publish them with read keys**, like the demonstrations already on the platform site. The public examples are the library; a customer's own policy is private. That split is the whole free/paid line.

REQUEST 3 · THE CONVENTIONS

Three conventions we need in order to render anything.

We are a consumer of your data. These are the parts of the contract that are currently implicit, and each one is something we would otherwise have to guess at and get wrong.

- **A per-row provenance field, machine-readable.** The site says 21 of 99 rows are measured, which is exactly the right disclosure — but it is prose. We need it per row (`measured` / `derived` , the source URL, the date read) so a rendered policy can carry the same honesty without us re-deriving it.
- **A stated position on the enabled-but-switched-off state.** A connector that exists and is turned off is not in the grant today and is one click from being in it. Neither the barrier model nor the label has a place for that, and it is probably the most common state in any real estate. We will render whatever you decide; we cannot decide it for you.

- **A version and a shape identifier we can pin to.** The delta is stored with its inputs pinned, which only works if the inputs have stable identifiers. `versions/index.json` gives us the site version; what we need is the identity of a *shape* and of the vocabulary it was computed against, so a recompute can say what moved.

Nothing here asks you to change the hard rules. No score, describes and does not judge, derived and never authored, no claim about a third party without a source and a date. Every request above is inside those, and Request 3 exists because we want to inherit them rather than reimplement them.

HOW TO REPLY

And whether this should be a vault instead.

This page is the fastest thing to link, which is why it exists. It is probably not the right long-term home, and the alternative is better in three specific ways.

- **A page can only be answered somewhere else.** You would reply on your site, or in a message, and neither ends up next to the request. A shared vault with write access on both sides makes the request and the answer the same artefact.
- **A vault versions the conversation.** These requests will change as we learn things. A page silently replaces its own history; a vault keeps it, which is the discipline both sites already claim.
- **A read key makes it public without making it editable** — the same mechanism both of our sites already use for everything else. Publishing a cross-team request list under a read key is a better demonstration of the model than another page describing it.
- **Against it:** it needs a vault provisioned and a key exchanged, which is minutes of work neither of us has done yet, and this page is linkable now. **Our suggestion:** use this page for this round, and if there is a second round, move to a vault and keep this page as a pointer to it.

Reply however suits. Publish a response on your own site and we will link it from here; open a vault and send us a write key; or answer in whatever channel is already open. If you tell us a request is wrong, this page changes with a line saying what changed and when — same rule as everything else in the Lab.

Written 12 September 2026 against abp.sgit.ai v0.2.0 and store.sgit.ai v0.1.1, both read the same day. The brief these requests come out of is a dev brief dated 12 September 2026; the findings behind Request 1 are in [Lab 01](#) and the flow they unblock is in [Lab 02](#).

● Lab 03

Three requests, one of them small.

Request 1 is a single property on an existing model. Request 3 is three fields and a stated position. Request 2 is real work, and it is the one that unblocks a product — so if only one of the three happens, it should be that one.

```
}());
```