



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 The live page may since have been corrected, extended or withdrawn — and if it has, the edition list on it will say so.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.

```
}());
```