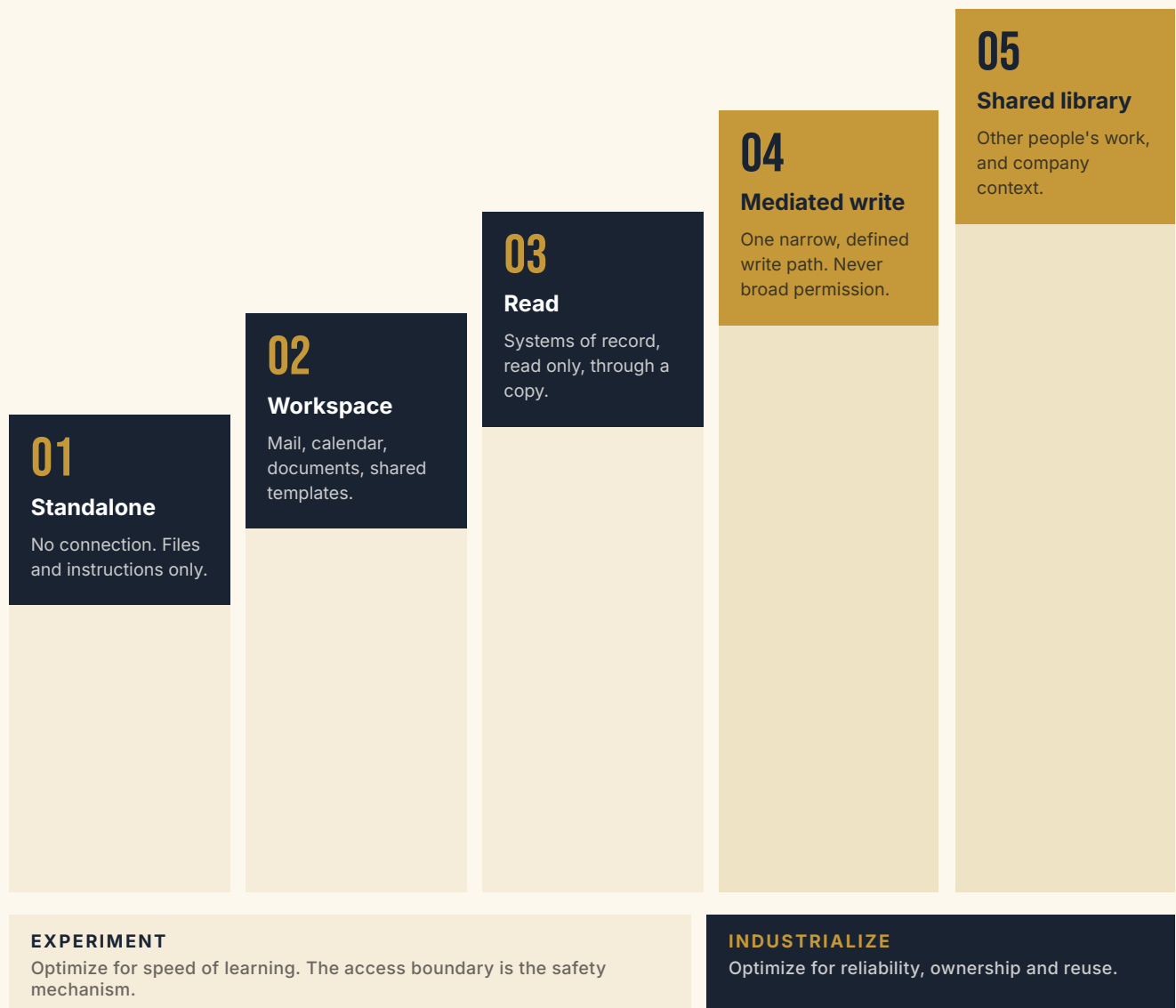


THE ACCESS LADDER

How AI agents move from a standalone assistant to your systems of record, one earned rung at a time.

A management team uses AI for six weeks, it works, and someone asks for the sales system next. **Opening a system of record is not one decision. It is five, and they arrive in order.** Each rung is earned by evidence produced on the rung below.



THE ARGUMENT

The question arrives before the *evidence* does.

A management team spends six weeks using AI. It works. Someone asks for the sales system next, and the question lands on a desk in IT: how far do we open this up?

The question is usually asked in the worst possible form. Opening a system of record to a language model sounds like a single decision, made once and early, by whoever is willing to sign it. Framed that way it has two losing answers. Say no and the capability stalls at personal productivity, which is where most organizations are stuck today. Say yes and you have granted broad permissions to a class of software nobody in the company has operated before.

It is not one decision. It is five, and they arrive in order.

Each rung is earned by evidence produced on the rung below. By the time the write question is real, three things are known that cannot be learned any other way: which uses people adopt, what the data can support, and who is accountable. Deciding early means deciding without any of them.

What you are optimizing for changes halfway up.

The first three rungs are an experiment. The goal is to find the handful of uses that people genuinely adopt, and the fastest way to kill that is to govern it like production. The access boundary is doing the safety work at this stage: no writes, enterprise accounts, nothing leaving the tenant. That is enough.

The last two rungs are industrialization. Something works well enough that other countries and functions want it, and the goal changes to reliability, ownership and reuse. Here the light-touch posture that made the experiment possible becomes the actual risk.

Most programs fail by applying one posture to both halves. Production governance on rung one means nothing ever ships. Experiment governance on rung four means an agent writes to a system of record with no owner and no log.

RUNG	WHAT THE AGENT REACHES	PHASE
1 Standalone	Nothing. Files and instructions only	EXPERIMENT
2 Workspace	Mail, calendar, documents, shared templates	EXPERIMENT
3 Read	Systems of record, read only, through a copy	EXPERIMENT
4 Mediated write	One narrow, defined write path	INDUSTRIALIZE
5 Shared library	Other people's work, and company context	INDUSTRIALIZE

UNDERNEATH ALL FIVE

A control layer of identity, traceability, ownership and a quality bar. It grows in step with autonomy, and the last page says when each part of it is needed.

Nothing to integrate, and everything to *learn*.

01 Standalone

REACHES NOTHING

WHY START HERE People cannot specify the tool they have never held. Ask a leadership team which agents they want and you get a list of what they already do, slightly faster. Give them two working ones and the requests change completely within a fortnight. This rung exists to change what the organization is capable of imagining, and it happens to be the rung where IT carries no new risk at all.

OPTIMIZING FOR Speed of learning. The number of people who form a daily habit.

RISKS One real one: people paste confidential material into consumer accounts. Enterprise accounts before anything else closes it. The other risk is treating this rung as the destination, running a successful pilot, and stopping. That is the common failure and it looks like success for about a quarter.

HOW IT WORKS Skills defined as instructions plus reference files, held in the user's own workspace. The person uploads what the work needs. No connection to any system.

LEAVE WHEN Two or three uses are in genuine daily use by people who are not enthusiasts, and the requests coming back are for the tool to see something it cannot currently see.

02 Workspace

MAIL · CALENDAR · DOCUMENTS

WHY HERE NEXT The first thing every one of those requests asks for is context the person already has: the thread, the calendar, last quarter's deck. This is the cheapest capability increase available, and it runs entirely inside collaboration tooling the organization already governs.

OPTIMIZING FOR Removing the copy and paste. The tool stops being a place you visit and becomes part of the working day.

RISKS Access is granted per person and inherits that person's permissions, so an over-permissioned employee becomes an over-permissioned agent. This is an existing access-governance problem that AI makes visible rather than a new one, and it is worth resolving on its own terms. Expect central IT policy to be the blocker rather than technology.

HOW IT WORKS Connectors to mail, calendar and the document store, scoped read only, under the individual's existing rights. Shared assets that many skills draw on, such as the presentation template and brand basics, are installed at tenant level rather than carried around by each person.

LEAVE WHEN People start asking questions the answer to which lives in the CRM or the ERP.

Where the argument stops being about *productivity*.

03 Read

SYSTEMS OF RECORD, READ ONLY,
THROUGH A COPY

WHY THIS RUNG CHANGES THINGS Everything up to here improves how an individual works. Reading the systems of record is where the conversation moves to the business, and it is the first rung where IT does substantial work. It is also, in practice, the rung that resolves the write debate, because it is where the organization finds out what its data is actually like.

OPTIMIZING FOR Learning two things at once: whether agents grounded in real data produce materially better judgment, and whether the data can support it. The second answer is frequently no at first, and finding that out on a copy costs nothing.

Three risks worth naming

Going to production. Direct access to a live system of record means a jump host, a security review and a queue, and it can take a month or more before anyone learns anything. A prepared extract into a separate store is faster, safer and better. Nothing at this rung needs live data.

Dumping the schema. A legacy CRM has hundreds of tables and thousands of relations. Handing over all of it produces an agent that retrieves confidently and wrongly. The work is choosing the twenty to fifty attributes that carry the meaning, and that work belongs to whoever owns the data, not to whoever owns the database.

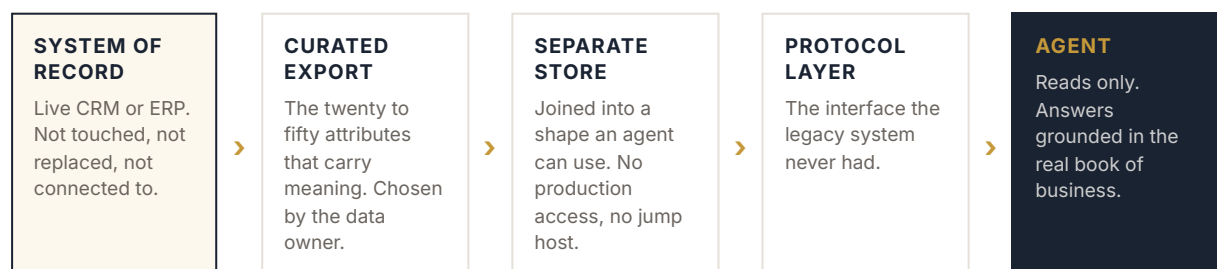
Silent staleness. An agent reading partial or out-of-date data will answer anyway. Coverage has to be stated, not assumed.

HOW IT WORKS A curated export from the warehouse into a separate store, joined into a shape an agent can use, with a protocol layer over it. The legacy system is not touched and not replaced. It gains an interface. For most organizations this is a few days of work once the scoping question is answered, and the scoping question is the whole job.

Sequence the people as deliberately as the data. Bring in the system's data owner and process owner now, while the request is only to read. They are the people whose agreement rung four will require, and they will have spent a rung watching it work before anyone asks them for it.

LEAVE WHEN Grounded answers are demonstrably better than ungrounded ones, the data gaps are documented, and the system owner has seen enough to have an opinion about writing.

WHAT RUNG THREE ACTUALLY BUILDS



The scoping question, not the plumbing, is the whole job.

The usual fear is aimed at the *wrong thing*.

04 Mediated write

ONE NARROW WRITE PATH

WHY The objection to write access is that AI will degrade the data. It is a reasonable fear and it is pointed at broad write permission, which nobody should grant. It is not an argument against writing.

Look at what the free-text fields in a mature CRM actually contain. One person writes three words. Another writes everything that was said. Most write nothing, because the meeting is over and the next one starts in five minutes. The record is already inconsistent, and it is inconsistent because writing it is manual work that competes with selling.

A skill writing through a fixed structure writes the same fields, in the same shape, every time. Our experience across engagements is that quality goes up rather than down, and the mechanism is unglamorous: the agent does not get bored and does not run out of time. Treat that as a finding from the field, not a measured result, and design the first write so it proves itself either way.

OPTIMIZING FOR Getting knowledge out of individual inboxes and into shared systems. That is the actual prize. A meeting outcome sitting in one person's mail is invisible to the company. The same outcome written to the customer record is available to everyone who touches that account next.

RISKS Broad write scopes, obviously. Less obviously, writes that no human ever reads back, which is how a small systematic error becomes ten thousand records. And accountability going vague at exactly the moment it starts to matter.

How it works: three constraints, all of them ordinary engineering

Constrain the surface. A gateway sits between the agent and the system and exposes one operation, not the API. Create an activity. Nothing else. The narrowness is the control, and it is far stronger than a policy saying which fields to be careful with.

Constrain the shape. The agent writes a defined structure rather than free prose. A structured header inside the field the system already has costs no schema change, and the people reading the record ignore it while the machine consumes it. Choose fields the business actually wants and does not reliably get today, such as buying signals, objections raised, and who is involved in the decision.

Keep a person in the loop, at first. Every write is proposed and approved by the person responsible for it. When the approvals stop changing anything, that is the evidence for widening, and it is evidence rather than confidence.

LEAVE WHEN The write path has run under human approval long enough to have a boring error rate, and the system owner will say so in writing.

The moment it stops being *personal*.

05 Shared library

OTHER PEOPLE'S WORK, AND
COMPANY CONTEXT

WHY No new permission is granted here. Something built by one team turns out to be wanted in six countries, and without somewhere to put it you get one of two bad outcomes: everyone rebuilds it slightly differently, or a central team becomes a queue.

RISKS The first attempt is almost always a page of downloadable files with no version control behind it. It works for a month, then nobody knows which copy is current. The quieter risk: handing someone a skill is not the same as them using it, and a library measured on installs rather than use will look healthy while nothing changes.

HOW IT WORKS Two halves. A curated library people install from, filtered by role, with an owner, a version history and a published quality bar. And a company context layer every skill draws on: who we are, the competitive picture, how we write, the current template. Distribution carries the training with it, because the constraint beyond the library is confidence, not access.

OWNERSHIP MIGRATES A presentation skill starts as one person's shortcut. Once it is used in four countries it belongs to whoever owns the brand, because when the template changes the skill must change with it. Decide the destination early and move it deliberately.

THE CONTROL LAYER · TRACEABILITY SCALES WITH AUTONOMY, NOT WITH ACCESS

While the work is reactive, a person asks and the agent answers. That person is the accountable actor and existing controls cover it. When work runs without a person in the loop each time, three things become necessary: an identity per agent so actions are attributable, a log of what was done and on what basis, and a named human owner for every agent in production.

Policies are not the mechanism. Nobody reads the policy. Make the correct path the cheap one and build the rules into the tools people already use.

Governance fails in two directions. Too little is the one everyone anticipates. The other is a review body that meets and does not decide, and the early symptom is a long argument about vocabulary, usually whether a given tool counts as an agent. Give each decision a named owner and a date.

The first ninety days

WEEKS 1 TO 4

Enterprise accounts. One leadership group on rung one. No integration work.

WEEKS 4 TO 8

Workspace connectors for that group. Open the rung-three scoping conversation with the data owner. It is a business conversation about which attributes matter, not a technical one.

WEEKS 8 TO 12

A curated extract in a separate store, and agents rebuilt against it. Choose the single write case that would most improve the record and design its gateway. Do not build it yet.

By then the organization argues from evidence rather than opinion. The system owners are already in the room, and the write decision has become small and specific.