

ANTON by openEXPERT

The ANTON family whitepapers · Autumn 2026 · Paper 8 of 11

Atoms: A Learning Loop

How each finished task can leave behind small, owned, checkable pieces of knowledge — and why memory has to earn its place

> *One sentence, owned, checkable, erasable. That is an atom. And until enough people have rated enough answers, it stays out of the prompt.*

Version 1.0 · 3 December 2026

Created by Daniel Bardun · FutureChain AB

Open source ANTON is EUPL · github.com/altspace-hub/anton_desktop

See it live ai-portal.futurechain.eu (showcases with fictional data)

Before we start

Where we left off

Last week we separated two sides. On one side sit the models: utilities you choose per task, pay for per use and replace when something better comes along. On the other side sits everything that is yours: your documents, your knowledge, your method and your records, in your own database, under rules you set. Models come and go. What you know should stay on your side.

That leaves an obvious question, and it is the one this paper sets out to answer. **If your knowledge must stay yours, how does it grow?** In the language of the first paper, this is the training gap — seen over months, not within a single task.

Knowledge packs, modules and skills are written deliberately, by someone who sat down to write them. But most of what a compliance team knows was never written down on purpose. It was learned on the job — one gap analysis, one review, one awkward board meeting at a time. This paper is about that kind of knowledge, and about how AI-assisted work can capture it without turning it into something nobody can check.

Memory has to earn its place

Every AI tool now promises "memory". I understand why. Starting every session from zero is exhausting, and a coworker who forgets everything overnight is not much of a coworker.

But in regulated work, memory is a double-edged thing. A remembered point that is right saves you an hour. A remembered point that is wrong — and is used silently, stated as fact, with no source behind it — can put an error into a board report that nobody will ever trace back to its origin. And a remembered point about a named person is personal data, whether the tool admits it or not.

So this paper argues for a particular kind of memory. We call its pieces **atoms**: one short sentence of professional knowledge each, owned by someone, traceable to the work it came from, open to rating and correction, erasable — and **gated**, so that it may influence an answer only once there is enough of it, and enough human feedback behind it.

Memory has to earn its place. That sentence is the whole paper in five words. The rest explains why we believe it, what it looks like in practice, and — just as important — what it does not do yet.

In this paper

- **Part 1, the problem:** where a team's learning goes today, three scenes, and the two failure modes of AI "memory".
- **Part 2, our thesis:** small, owned, traceable, reviewable, erasable and gated — and what we are *not* claiming.
- **Part 3, our take:** what an atom is, how atoms are made and reused, the gate, grounded atoms, ownership and erasure, and the family design.
- **Part 4, a worked example:** the life of one atom over a year.
- **Part 5:** what this means for you on Monday, and next week.

Part 1 · The problem

Where the learning goes

Every finished piece of compliance work teaches something. Usually something small.

This supplier's due diligence file always lacks the chart of beneficial owners. This regulation's Swedish transposition differs from the directive on one point, and it is the point that matters for us. Our board wants residual risk shown before inherent risk, because the chair once said so and nobody has forgotten. The last inspection probed outsourcing harder than anything else. The house view on a borderline question was settled in March, after two long meetings.

None of these is a policy. None of them belongs in a regulation or a textbook. But together they are the difference between a new colleague's first draft and an experienced colleague's first draft. **They are how your organisation actually works.**

And today, almost all of it lives in two places: in people's heads, and in documents nobody re-reads. The minutes of the March meeting exist, somewhere. The analyst who noticed the missing ownership chart remembers it. Ask her in a year and she will still know. Ask her replacement, and the lesson has gone.

> **Organisations do not learn from documents. They learn from doing the work — and then, very often, they forget.**

Three ways memory lets you down

These are composites, as always. No names, no clients.

Scene 1: Monday starts from zero

A senior associate uses an AI assistant for regulatory research. For three months she works with it most days. Every Monday it has forgotten what she established the week before: which interpretation the firm chose on a contested article, which gap turned up at three clients in a row, which source turned out to be out of date.

So every Monday she explains it again. After three months she gives up and goes back to a shared document — a long, unstructured page of notes that she, at least, can find things in. The AI was clever. It simply never got any wiser.

Scene 2: the confident memory

A different team uses a tool that does remember. In a new report for a client, the assistant states — fluently, in passing — that the client's AML policy was updated last year to cover a new product line.

It was not. That update happened at a *different* client. The assistant had remembered something true, and attached it to the wrong organisation. Nobody can see where the memory came from, or even that a memory was used at all; it reads exactly like everything else in the draft. A careful reviewer catches it. A tired one would not have.

Scene 3: the erasure request

A former employee of a client asks to have their personal data erased. The compliance team knows the drill: find the documents, check the retention rules, delete what may be deleted, record what was done.

Then someone asks the uncomfortable question. *The AI tool has been working on this client's files for a year. Did it "learn" anything about this person? And if it did, where is that now — and how would we erase it?* With most tools, nobody in the room knows. Some suspect nobody at the vendor knows either.

Two ways memory fails

Put the three scenes side by side and you see the two failure modes of AI "memory" in regulated work.

The first is no memory at all. Every session starts from zero. The cost is time — the twenty minutes of scene-setting we described in the first paper of this series — but also something subtler. A tool that never learns cannot grow with you. It is the graduate from paper one, arriving every morning having forgotten everything you taught them the day before. You can use a colleague like that. You cannot build on one.

The second is worse: memory you cannot see. A pile of remembered fragments that the tool uses without telling you, that nobody reviewed, that cannot be traced to a source, cannot be corrected and cannot be erased. That kind of memory does not just fail to help. It quietly changes your conclusions, and it hides the change.

Notice what the second failure does to everything this series has argued for. In paper two we said every AI-assisted conclusion should carry *what went in*. A silent memory is an input nobody declared. In paper four we said trust comes from process. A memory nobody reviewed is a step outside the process. **Silent memory is a mystery box inside the glass box.**

Why "preferences" are not the answer

Many tools solve the problem by remembering preferences. *Prefers short answers. Works in Swedish. Likes tables.* That is harmless, and genuinely pleasant. But it is not what a compliance team needs to remember.

What a team needs to remember is professional: findings, decisions, risks and recommendations. *UBO verification at this firm relies on registry extracts alone. We decided to treat this product as high risk until the review in June. The supervisor's last letter asked for evidence of testing, not just documentation.* Each of those is about something specific, came from somewhere specific, and might stop being true next quarter.

That changes the requirements completely. A preference does not need a source; a finding does. A preference never goes stale in a way that matters; a decision does. Nobody will ask you to erase a preference for short answers; they may well ask you to erase what was learned about a named person. **Professional memory needs the same discipline as professional work.**

Why this matters now

Two things make this urgent rather than academic.

The wave of work is coming. The AMLR applies from 10 July 2027. Over the next year, compliance teams across the EU will run more gap analyses, policy rewrites and risk assessments than in any comparable period. Each of those tasks will teach something — about the firm, about the regulation, about the gap between them. If those lessons evaporate after each task, the second and third pass will cost almost as much as the first.

Data protection does not stop at the edge of a model. A memory that holds a person's name is personal data. GDPR's principles — purpose, minimisation, accuracy, erasure on request — do not switch off because the data sits in an AI feature rather than in a document folder. A memory design that cannot answer *"what do you hold about this person, and can you erase it?"* is a finding waiting to happen.

Part 2 · Our thesis

Memory has to earn its place.

The thesis

Organisations learn by doing tasks. What each task taught should be captured as **atoms** — small, owned, reusable pieces of knowledge — so that the next task starts smarter, in your way of working.

But not any kind of capture will do. For learning from AI-assisted work to be safe in a regulated firm, it should be six things:

1. **Small.** One sentence at a time. Small enough to read, check and argue with.
2. **Owned.** By a person, a team or the organisation — and the owner is recorded.
3. **Traceable.** To the work it came from and, where possible, to the exact source text.
4. **Reviewable.** It can be rated, replaced, retired.
5. **Erasable.** By subject, on request, with a preview of what will go.
6. **Gated.** It may influence an answer only once there is enough of it, and enough human feedback behind it.

Take away any one of the six and you are back in one of the three scenes. Without traceability you get Scene 2. Without erasure you get Scene 3. Without the gate you get something we learned about the hard way — which brings me to the reasons.

Why we believe it

1. Context compounds; models do not. This is the argument of our essay *"Beyond AGI: The Case for APCI"*, and of last week's paper. Models improve every quarter and become available to everyone at once. What your organisation learns from its own work is the one part of AI value that is uniquely yours, and the one part that grows over time. If that learning is not captured, the compounding never starts. If it is captured on the model vendor's side, it compounds for someone else.

2. We learned the hard way that memory must earn its place. I would rather say this plainly than have someone discover it. In September 2026 we reviewed ANTON's knowledge layers, one by one. The early memory did not pass. It was too eager: it put unreviewed fragments into prompts, it had no ratings at all, and it blurred the line between something remembered and something evidenced. We rebuilt it. Memory is now labelled as memory, gated by volume and by ratings, owned and erasable. **That lesson is now the design for the whole family.** I think an internal review that finds a real problem, and a rebuild that follows, is exactly what a reader of this series should want to hear about. It is how a method gets better.

3. Grounding beats recollection. In ALEXANDER, our expert for standing mandates, the model is allowed only to *propose* an atom. Code keeps it only if the quote it rests on is found, word for word, in the source document — and stores where in the document it was found, with a fingerprint of the text. A remembered point you can open and check is worth more than ten you cannot. That is paper two's principle, *every quote is checked*, applied to memory.

4. Learning is not the same as "memory". Generic chat memory stores preferences. Atoms store professional knowledge, and each one carries its type (a finding, a decision, an action, a risk, a recommendation), the subjects it concerns and where it came from. That is what lets an atom be reused in the right place — and challenged when it is wrong.

An analogy: the good handover note

Think about the best handover note you ever received when you took over someone's work. It was not a transcript of everything they had done. It was a page of short, specific sentences. *This client's board meets in the second week of the month. The outsourcing register is right; the policy is not. Do not trust the figures in last year's appendix — they were corrected in the follow-up.*

Each sentence was small. Each one said what it was about. Each one, if you doubted it, could be checked against something. And you read it as what it was: the previous owner's knowledge, offered as a head start — not as evidence you could put in a report without looking.

An atom is one line of a good handover note. The difference is that the note is written as the work happens, not in a rush on someone's last day — and it is still there when nobody remembers who wrote it.

What we are not claiming

Atoms are the paper in this series where it is easiest to over-claim, so let me be precise.

We are **not** saying that ANTON "learns your organisation" automatically, today. ANTON's memory is gated. On its automatic setting it adds nothing to a prompt until there are 100 atoms from real module work and 30 human ratings. On a typical installation that gate is still closed, and memory is still collecting.

We are **not** saying atoms are verified knowledge. ANTON's atoms are written by a model, from the work; they are labelled as memory, and they are hints, not evidence. ALEXANDER's atoms, and the points Horizon distils from the law, are different: they are grounded in quotes that code has checked. The paper keeps the two apart, and so should you.

We are **not** saying atoms improve the quality of the output. We believe they will, for the reasons above. We have not measured it yet, and we will not claim it until we have.

And we are **not** saying the family learns across clients. That is a design, with strict conditions, and it is not built. Nothing is learned across clients today.

Part 3 · Our take: atoms

Six principles

1. **Only real work teaches.** Atoms come from finished module runs, assessments, engagement steps and tasks — not from open chat.
2. **One sentence, one owner.** Every atom is short, typed, about named subjects, and owned by a person or the organisation.
3. **Hints, not evidence.** When memory reaches a prompt, it is labelled as memory, not as a verified source, and it is listed against the answer it touched.
4. **The gate opens on evidence.** Memory influences answers only once there is enough of it and enough human rating behind it.
5. **People keep control.** Rate, replace, retire, reactivate, delete — and erase by subject, with a preview first.
6. **Your database, your walls.** Atoms live with you, never leave the instance unless shared, and never cross a client boundary.

The third principle is the one I would defend hardest. It would be easy to let remembered points blend into the answer, so that the draft simply reads as if the AI "knew". That is precisely what made Scene 2 dangerous. **A hint that is labelled as a hint can be checked. A hint that is disguised as knowledge cannot.**

What an atom is

An atom is one short sentence of professional knowledge. Six properties make it more than a note:

- **It is one of five kinds.** A finding, a decision, an action, a risk or a recommendation.
- **It is about something.** It names the organisations, regulations and systems it concerns, so it can be found when they come up again — and only then.
- **It comes from somewhere.** It is tied to the piece of work that produced it. In ALEXANDER and Horizon it is tied to the exact quote in the source.
- **It is owned.** By a person, or shared at organisation level. Only administrators change shared atoms.
- **It is alive for a while.** It can be rated, replaced by a newer atom, retired or erased. An old atom must have been rated useful before it may be used again.
- **It is linked.** One atom can support, extend or contradict another.

Here is what one looks like, in plain words:

One atom

- **Text:** *"Finding: UBO verification at Exempelbolaget relies on registry extracts alone."*
- **Kind:** finding. **About:** Exempelbolaget AB; the AMLR's customer due diligence rules.
- **From:** the January gap analysis. **Owner:** the analyst who ran it.
- **State:** active; not yet rated.

That is all. One sentence a colleague could read in three seconds and dispute in five.

	Generic chat memory	Atoms
HOLDS	Preferences and fragments	Findings, decisions, risks, recommendations
WHEN USED	Silently	Labelled as memory, listed per answer
SWITCHED ON	Always	Gated: waits for enough atoms and ratings
SOURCE	None you can see	Tied to the work; in ALEXANDER and Horizon, to the exact quote
CORRECTION	Hard	Rate, replace, retire, erase by subject
LIVES	Often on the vendor's side	In your database, owned
ACROSS CLIENTS	Only by design: consented, de-identified, written by code. Not built.	

Generic memory remembers preferences. Atoms remember the work — and can be checked, corrected and erased.

How ANTON makes, gates and reuses atoms

This is the part of the family where atoms live in a generalist, so it is where the rules have to be strictest. Everything in this section is in ANTON's main code (merged).

Only real work teaches

When a module run, a gap assessment, an engagement step or a task finishes, its output is stored with a small learning record that moves from *pending* to *summarised* and then to *learned*, *skipped* or *failed*. You can see, for each piece of work, whether anything was learned from it.

Open chat is not learned from. That is deliberate. A chat is where people think out loud, test ideas and ask questions they would not want remembered. A finished piece of work is where a professional has committed to something.

After the answer, in the background

Once you have your answer — never before, and never in your way — a background step asks a model to extract three to eight short atoms from the finished work, each with the subjects it mentions. This is the step where AI proposes.

Then code filters. Status chatter (*"I have now completed the analysis"*) is dropped. So is the assistant talking about itself. An atom that is already held word for word is refused. And an hourly sweep finishes any learning that was interrupted, so nothing is quietly lost because a laptop lid was closed at the wrong moment.

The gate

This is the heart of ANTON's design. One setting decides whether atoms may enter new prompts at all: **off**, **on** or **auto**.

On **auto**, ANTON waits until there are **100 atoms from real module work and 30 human ratings**. Until then, memory collects, but it does not influence a single answer. You can read it, rate it and correct it; it simply does not get a vote.

Why those conditions? Because a memory with too few atoms is a memory of accidents — a handful of points from whichever tasks happened to come first. And a memory nobody has rated is a memory nobody has checked. The volume condition says *there is enough here to be representative*. The rating condition says *people have looked at what this memory does, and told it when it helped*. **Both have to be true before memory earns a place in your prompts.**

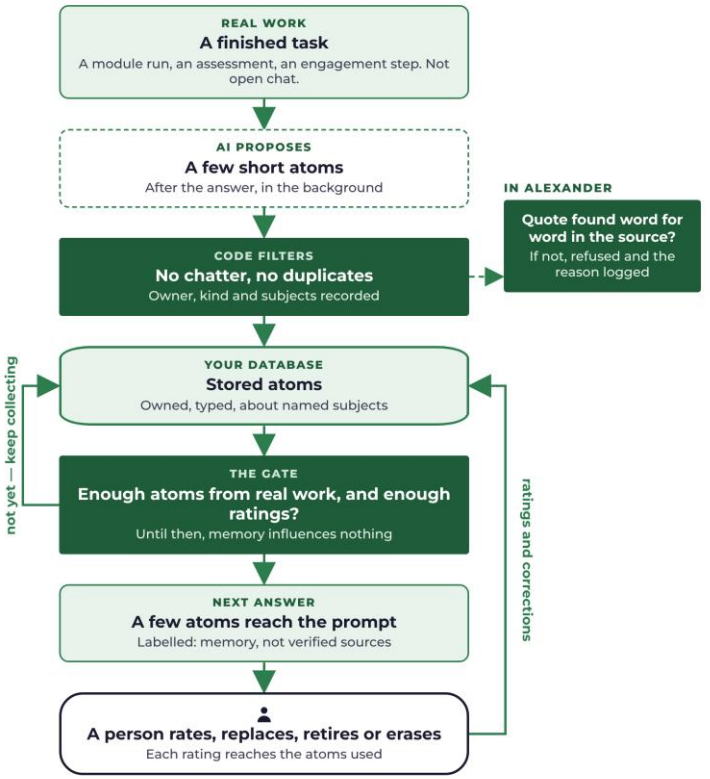
Is the gate open on your installation? On most installations today, honestly, no. It is still collecting. That is the gate doing its job.

When the gate is open

When memory does reach a prompt, it arrives under tight rules:

- **At most five atoms per answer.** Enough to remind; not enough to steer.
- **Under a heading that says what they are:** memory, not verified sources — to be treated as hints, never as evidence.
- **Old atoms must have earned it.** An atom older than 180 days is used only if someone has rated it useful.
- **Each atom used is recorded against the answer.** So when you rate the answer, the rating reaches the atoms that shaped it.

That last rule closes the loop. The ratings that opened the gate keep feeding back, atom by atom, so memory that helps is kept in play and memory that does not quietly falls out of use.



Memory has to earn its place: the AI proposes, rules filter and gate, and a person rates and corrects.

People keep control

Every atom can be superseded by a newer one, retired, reactivated or deleted. You can search memory by a person's name and erase everything about them in one step — after a preview that shows exactly what will go. That is Scene 3's answer, and it exists because we designed for it from the start of the rebuild, not after the first request arrived.

Picture: ANTON's memory view: a list of atoms with kind, subjects and owner, and the gate setting showing "collecting" (fictional data)

Picking up where you left off

One smaller feature, because it answers Scene 1 directly. After each answer, ANTON writes a short running conclusion of the session — where the work stands, what was decided. Come back on Monday, and you pick up from that conclusion rather than from zero. It is not an atom, and it is not learning. It is simply a good habit, automated.

Grounded atoms: ALEXANDER and Horizon

In ANTON, atoms are hints written by a model. In two of the specialists, we went a step further, because their work allowed it: every atom is tied to a quote that code has checked.

ALEXANDER (live on its showcase) keeps a standing mandate under review over months and years, so what it remembers matters more than in a one-off task. Its rules are strict:

- The model **proposes** an atom, with the quote it rests on.
- Code **checks the quote word for word** against the source document, and stores its position and a fingerprint. If the quote cannot be found, the atom is refused — and the refusal is logged with the reason.
- Confidence and importance are **computed by code**, never set by the model.
- Atoms are **never overwritten**. They are replaced, retired or expire, so the history of what was believed, and when, is kept.
- **A new version of a document retires what the old version said.** Memory follows the evidence, not the other way round.
- Each review cycle receives only the atoms that bear on its mandate, **each with the reason it was chosen**.

Atoms in ALEXANDER are kept per organisation and per mandate. One honest gap: erasure by subject is designed in ALEXANDER but not yet built.

Horizon, our regulatory change expert, has built the first phase of what it calls *points with their proof*: short points distilled from the law it holds, each tied to its exact quote and cut from the text by code. Those points are live on its showcase. A second phase — points drawn from an institution's own records — is planned, and by design those will never be sent to a hosted model.

> **A remembered point you can open and check is worth more than ten you cannot.**

Ownership and erasure

Paper seven argued that your knowledge belongs on your side. Atoms are the most personal form of that knowledge, so the rules are simple and firm.

Atoms live in your database. Not with a model vendor. On a shared ANTON team server, each person sees their own atoms plus the ones shared at organisation level, and only administrators can change the shared ones.

Atoms do not leave the instance unless they are shared. When ANTON works with another ANTON — a colleague's, or a team's (the subject of paper eleven) — only shared atoms can travel, and never an atom from a private coding project.

Erasure is by subject. Preview, then erase. The data protection officer from Scene 3 can answer the uncomfortable question with a list, not a shrug.

On the public showcase, nothing is learned from visitors. The showcase demonstrates the feature. It does not collect anyone's work.

Atoms across the family: the design

Each specialist has its own form of atoms, but they should mean the same thing everywhere. So on 4 October 2026 we wrote one family specification for atoms, and it is copied word for word into ALEXANDER, Horizon, Valentin and Olivia. Its decisions follow the shape of each product's work:

- **ALEXANDER and Horizon:** atoms at organisation level. Horizon's points from public law can be shared; an institution's own points stay inside its walls.
- **Valentin and Olivia:** a client's atoms stay in that client's engagement or workspace. Nothing crosses between clients.

There is one more idea in the design, and it is the one I find most interesting — and most in need of care.

Designed, not built: firm learnings across clients

An outsourced compliance desk or a review team sees the same gap at many clients. That pattern is real knowledge. The design lets it be learned **only** under these conditions:

- The same structured pattern appears at **three or more clients**, each of whom has **opted in**. Consent is per client and recorded on the workspace.
- **Code writes the lesson** from structured, non-identifying fields, using fixed template text. The model never writes it.
- Only **fingerprints** of the sources are kept, never the sources themselves.
- If fewer than three consenting clients remain, or one withdraws consent, the lesson is **withdrawn automatically**.
- A person reviews every lesson.

This is a design for Valentin and Olivia, under review. **Nothing is learned across clients today.**

I want to be honest about why that box is so long. Learning across clients is where the value of atoms could be greatest — and where the risk is greatest too. Client confidentiality, bank secrecy and data protection all point the same way: by default, a client's knowledge stays with the client. Any exception has to be narrow, consented, de-identified and reversible, and the words of the lesson must come from a template, not from a model that might let a detail slip. If we cannot build it to that standard, we will not build it.

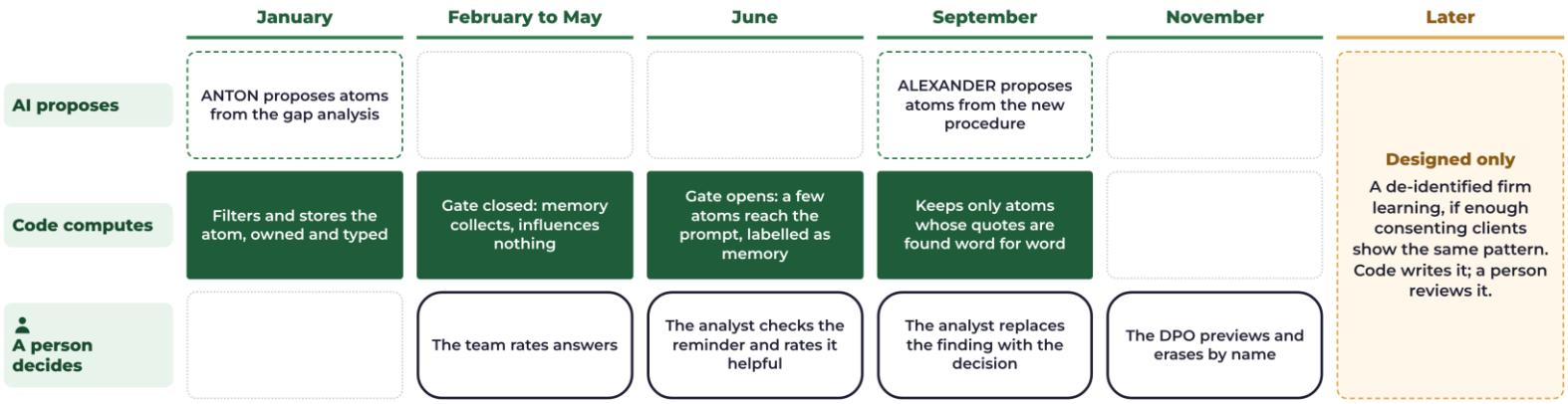
Status at a glance

- **Live** (public showcases, fictional data): ALEXANDER's grounded atoms; Horizon's first phase of points with their proof.
- **In main code** (merged, not checked as deployed): ANTON's memory — learning record, extraction, gate, ratings, erasure by subject — and the running session conclusion.
- **Designed or planned:** the family atoms specification; erasure by subject in ALEXANDER; Horizon's points from an institution's own records; firm learnings across clients in Valentin and Olivia.

Part 4 · A worked example

One atom's life over a year

Exempelbolaget AB is fictional. Its compliance team uses ANTON for day-to-day drafting and analysis, and ALEXANDER for a standing mandate on its AML framework. Follow one small sentence through a year — from the work that produced it to the request that erased something it touched. Each step is marked with who acts, and its status today.



The next task starts smarter, in your way of working — once memory has earned its place.

Step	What happens	Who acts	Status
1. January: the work	A gap analysis against the AMLR's due diligence rules finds that UBO verification relies on registry extracts alone.	A person, with ANTON	Merged
2. January: the atom	After the answer, ANTON proposes the atom *"Finding: UBO verification at Exempelbolaget relies on registry extracts alone."* Code filters and stores it, owned by the analyst.	AI proposes; code filters	Merged
3. February to May: collecting	The team keeps working and rating answers. The gate stays closed; memory influences nothing.	Code (the gate)	Merged
4. June: the gate opens	With 100 atoms from module work and 30 ratings, the gate opens. A procedure update receives this atom among at most five, labelled *memory, not verified sources*, and opens with a reminder.	Code (the gate)	Merged
5. June: a rating	The analyst checks the reminder against the January report and marks it helpful. The rating attaches to the atom.	A person	Merged
6. September: the fix	The procedure is updated. The analyst supersedes the atom with the decision *"UBO verification now combines a registry extract with a second source (procedure v4)."* The old atom is retired, not deleted.	A person	Merged

Step	What happens	Who acts	Status
7. Meanwhile, in ALEXANDER	Version 4 arrives on the mandate. ALEXANDER retires what the old version supported and proposes new atoms; code keeps only those whose quotes it finds word for word.	AI proposes; code checks	Live (showcase)
8. November: an erasure request	A named individual asks to be forgotten. The DPO searches memory by name, previews, and erases.	A person	Merged in ANTON; designed in ALEXANDER
9. Later: a firm learning	If three or more consenting clients show the same pattern, a de-identified lesson could appear: *"A recurring gap: UBO verification on registry extracts alone."*	Code writes; a person reviews	Designed only

What the atom did, and what it did not do

Look at what that one sentence achieved. In June, a person working on a different task — a procedure update, not a gap analysis — was reminded of a finding from five months earlier, by someone else, in a different document. Without the atom, that reminder depends on the right person remembering at the right moment. With it, the next task started a little smarter.

Now look at what the atom did *not* do. It did not write itself into the procedure as a fact. It arrived labelled as memory, and the analyst checked it against the January report before relying on it. It did not survive the fix: once the procedure changed, a person replaced it with the decision that superseded it, and the old finding was kept as history rather than left lying around as a stale truth. And it did not influence anything at all for the first five months, because nobody had yet shown that the team's memory was worth listening to.

That, to me, is the right shape for organisational learning with AI. **Small enough to check. Owned by someone. Corrected when the world changes. Erased when a person asks. And humble enough to wait until it has earned a vote.** The sentence that reminded the team in June is the same sentence that could be shown to an auditor in December, with its origin, its rating and its replacement beside it.

There is one more thing worth noticing. Step 9 is in the table, and it is marked as a design. I left it in deliberately, because it shows where this can go — and I marked it plainly, because in a field that runs on evidence, the difference between *built* and *designed* is not a detail.

Part 5 · From here

What this means for you on Monday

You do not need any of our software to start. Most of this is a decision about what your organisation should remember, and on what terms. Five things to do this week:

1. **Decide what your organisation should remember.** Findings, decisions, risks and recommendations from finished work — not chat. Write the list down, and write down what you never want remembered.
2. **Ask your AI tools four questions about memory.** Can I see what it remembered? Where did each point come from? Can I correct it? Can I erase everything about a named person? If any answer is *no*, treat that tool's memory as switched off for regulated work.
3. **Rate the answers you rely on.** Ratings are what let memory switch on, and what keep it honest once it has. Five seconds per answer is the cheapest quality control you will ever run.
4. **Treat remembered points as hints.** Any remembered point that ends up in a deliverable must be checked against a source first — exactly as you would check a colleague's handover note.
5. **Write down your consent rule** before you ever let learning cross a client boundary: opt-in per client, de-identified, at least three clients, reviewed by a person. Better to agree it now, calmly, than under pressure later.

Next week: from once a year to all the time

This paper closes Act 2, the method. Next week Act 3 begins: the new way of working. Checks, validations and monitoring have been yearly because they were expensive. When they cost hours rather than weeks, they can run whenever something changes — a document, a release, a regulation. We look at what continuous, AI-supported compliance looks like when it is built carefully, and at what it changes for the people who do the work.

A closing thought

There is a tempting story about AI memory: that one day the tool will simply know your organisation, the way a long-serving colleague does, and you will never have to explain anything again. I understand the appeal. I do not think it is the right goal for regulated work — or at least, not in that form.

A long-serving colleague is valuable not just because they remember, but because you can ask them *how* they know, they can be wrong in a way you can see, and they will tell you when something has changed. That is what we have tried to build into atoms: memory that can show its source, admit its age, accept a correction and forget on request. It is slower to switch on, and it says *hint* where other tools would say *fact*. **I think that is exactly what memory should be in a field that runs on evidence.**

The learning loop is small, and it starts quietly. Each finished task leaves a sentence or two behind. People rate what helped. After a while, the next task begins a little further on than the last one did. Nothing about that is magic. All of it compounds — and all of it stays on your side.

— Daniel Bardun

See it, try it, talk to us

- **See it:** the ANTON family's public showcases, with fictional data, at ai-portal.futurechain.eu.
- **Try it:** ANTON is open source — github.com/altspace-hub/anton_desktop.
- **Read the series:** all papers in the ANTON family whitepaper series, and the earlier whitepapers, are linked from the portal. [Daniel: confirm where the PDFs will live.]
- **Talk to us:** Daniel Bardun, on LinkedIn — questions, disagreements and "this is where AI stopped working for us" are all welcome.

Important notes

Product facts were checked on 11 October 2026; statuses are summarised in "Status at a glance". The public showcases use fictional or public data only, and no client uses them. ANTON is open source (EUPL); the other family products are not. The products support your work and help you evidence it; they do not make anyone compliant, and nothing in this paper is legal advice.