

ANTON by openEXPERT

The ANTON family whitepapers · Autumn 2026 · Paper 5 of 11

The Seven Layers of Knowledge

Where an AI's knowledge comes from when it answers a compliance question — and how to make sure it is the right knowledge

> *The model did not get smarter between the first answer and the last.*
> *It got context.*

Version 1.0 · 12 November 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

The first four papers were about the problem and how we think about it. We named six gaps between a capable model and compliance work someone can stand behind. We argued that the box has to be glass, that experts close to the work beat one big button, and — last week — that trust in AI comes from the process around it: roles, evidence, checks, sign-off and a record. Nothing in a good process asks you to trust the model.

That leaves an obvious question. A process makes sure the work is checked. But what makes the work *good* in the first place? When the AI proposes a finding, drafts a section or compares a policy with a regulation, where does its knowledge actually come from — and how do we know it is the right knowledge? In the language of the first paper, this is the training gap — and, close beside it, the knowledge gap.

This paper opens the second act of the series: **the method.** And the method starts with context. In the first paper I sketched where we believe this is going: work in which every conclusion carries its own explanation, and organisations own their professional context while models are bought and swapped like utilities. The method is how you get there — one written-down layer at a time.

What the model has not read

Here is the simplest way I know to put the problem. **Your AI has read the AMLR. It has not read your policy.**

It has not read your supervisor's last letter, your board's risk appetite or your house style. It does not know that you are a payment institution and not a bank, or that the reader of this document is the board and not the MLRO. It does not know what your team decided last quarter about a borderline issue, or that the procedure on page twelve was replaced in March.

And everything it does not know, it fills in with what is typical. Typical sector. Typical reader. Typical format. Typical risk. The result reads well and sounds professional — and it is about nobody in particular.

"Typical" is the enemy of a good compliance deliverable. A gap analysis that would fit any firm is a gap analysis that fits no firm. The whole value of the work lies in the specifics, and the specifics are exactly what the model was never given.

This paper is about how to give them — not once, in a long prompt, for one person on one day, but in a structure that works the same way for everyone, every time. We call that structure the seven layers of knowledge.

In this paper

- **Part 1, the problem:** what happens when a model fills the gaps with "typical", three scenes, and why longer prompts do not fix it.
- **Part 2, our thesis:** good output is mostly about context — the idea we call APCI — and what we are not claiming.
- **Part 3, our take:** the seven layers, how they are assembled, the four kinds of knowledge source, and what we have built.
- **Part 4, a worked example:** one question, answered seven times, one layer at a time.
- **Part 5:** what this means for you on Monday, and next week.

Part 1 · The problem

Filling in with what is typical

A general model is, in a sense, an enormous average. It has read more than any person could read in many lifetimes, and from all that reading it has learned what is *usual*: what a gap analysis usually looks like, what a payment institution usually worries about, how a risk memo is usually phrased.

That is why the first draft so often looks right — and why it so often *is not* right, for you. Compliance work is not about the usual. It is about this firm, this regulation, this reader, this evidence. The model's great strength becomes a weakness the moment it is asked about something specific and has not been told the specifics.

And here is the subtle part. The model does not say *"I don't know your sector, so I assumed a bank."* It simply assumes, fluently, and carries on. The gaps in its knowledge are filled in silently, with plausible defaults, and the reader has no way of seeing where the defaults went in.

Three ways "typical" gets in

As always, these are composites. No names, no clients.

Scene 1: the generic answer

An MLRO asks an AI tool: *"What does the AMLR require for customer due diligence?"*

The answer is accurate. It is well organised, correctly summarised, sensibly hedged. It is also useless to her. What she needed was not what the AMLR requires in general. She needed what it requires *of us* — compared with *our* procedure, in *our* house format, for *our* board, which reads two pages and no more.

Nothing in the answer was wrong. It was simply answering the typical question, from a typical asker.

Scene 2: the prompt in someone's notebook

A senior analyst has, over months, written a two-page prompt that produces excellent gap analyses. It sets out the method, the severity scale, the format the board likes, the things to watch out for. It is genuinely good — the distilled experience of someone who has done the work many times.

It lives in her notebook.

When she is on holiday, the team's output drops. Colleagues write their own prompts, shorter and different, and get shorter and different results. When she eventually moves on, the prompt goes with her. **The best expertise in the team was never part of the team's way of working. It was part of one person's desktop.**

Scene 3: the wrong source

An AI answer cites an article of the AMLR. The reference looks precise. It is wrong: the article was renumbered between the Commission's proposal and the final text that was adopted. The model had, in effect, *learned* the proposal. Nobody had given it the final text.

This is not a rare failure. Regulations change between draft and adoption; guidelines are revised; national rules are amended. A model's memory of a regulation is a snapshot from whenever it was trained, blended from many sources of varying quality. Unless it is given the actual text, with its date, it will answer from that blurred memory — confidently.

> **Everything the model does not know, it fills in with what is typical. And it does not tell you where.**

Why longer prompts do not fix it

The usual response to all this is to write longer prompts. Put the context in. Tell the model who you are, what you need, which rules apply, how to write.

It works — once, for one person, on one day. And then it runs into four problems.

It does not travel. The prompt lives where its author keeps it. The next person writes a different prompt and gets a different answer. Scene 2 is the natural end state of a team that relies on long prompts: pockets of excellent private practice, and no shared method.

It does not keep. A long prompt is a snapshot. The regulation changes, the house style changes, the board changes its expectations — and every copy of every prompt is now slightly out of date, in ways nobody tracks.

It cannot be checked. When a long prompt produces a poor answer, which part was the problem? The method? The sources? The instruction about the reader? Everything is mixed together in one block of text, so nobody can tell which piece of context did what.

It asks the wrong person to do the wrong job. As we said in the first paper, most professionals are not prompt engineers, and they should not have to become one. The MLRO knows what a good BWRA (business-wide risk assessment) looks like. She should not also need to know how to phrase it for a machine.

Context is not one thing

When you take a really good long prompt apart — the kind in the senior analyst's notebook — you find that it is not one thing at all. It is several different kinds of knowledge, tangled together.

Some of it is about **how to write**: cite the article, separate fact from judgement, say what you could not check. That changes almost never, and it applies to everything.

Some of it is about **the field**: this is financial crime prevention, this firm is a Swedish payment institution supervised by Finansinspektionen, these are the regulations that apply. That changes when the regulation or the firm changes.

Some of it is about **the task**: how a gap analysis is done, in what order, what the output looks like, what to avoid. That is different for every kind of task, but the same every time that task is done.

Some of it is about **the reader**: the board, the auditor, the supervisor. Some of it is **house style** and checklists. Some of it is **the material** for this particular piece of work: the policy, the procedure, last year's assessment. And some of it is about **how hard to think**, and what to show at the end.

Each of these has a different owner, a different lifespan and a different reason to change. **Mixing them into one block of text is like keeping the firm's policies, its templates and this week's case file in a single document.** It works until the first time anything changes. The answer is to separate them — and once you do, you find you have layers.

Part 2 · Our thesis

Good AI output is mostly about context.

The thesis

Context is not one thing. It comes in layers, from the most general — how to write — to the most specific — this document, this question. Each layer has a source and an owner. **Stacked in a fixed order, they make a general model behave like an experienced professional, and they make it behave the same way next time.**

That last part matters as much as the first. A single brilliant answer is not very useful in regulated work. What is useful is a way of working that gives a good answer to anyone who uses it, on any day, and can show how it got there.

Why we believe it

1. It is how professionals are trained. Think about how a new analyst becomes good. First they learn the house rules — how we write, how we cite, what we never do. Then the field: the regulations, the supervisors, the vocabulary. Then the method for each task, usually by being shown a good example. Then who they are writing for, and how that reader thinks. Then the firm's own checklists and templates. And then, finally, they open the case file. **The seven layers mirror that order exactly.** We did not invent the structure; we copied it from how competence is actually built.

2. Intelligence is not the bottleneck. Context is. We argued this at length in March, in *"The Context Layer"*, and in our essay *"Beyond AGI: The Case for APCI"*. A smarter model still does not know what your organisation decided last quarter. It still does not know your reader, your method or your evidence. Those gaps survive any increase in raw intelligence intact. More on this below, because it is the idea underneath everything in this paper.

3. Layers make context reusable. Once a layer is written — a module's method, a house style, a knowledge pack — everyone gets it, every time. It stops living in one person's notebook. The senior analyst's two pages, split into their proper layers, become part of how the whole team works, and stay when she goes on holiday.

4. Layers make context explainable. Because each layer is a separate, named part, the record of a run can store each one with its fingerprint. You can see which layer said what — which is exactly the glass box we described in the second paper. When an answer is poor, you can ask which layer was missing or wrong, and fix that layer, once, for everyone.

The idea behind the layers: APCI

Much of the AI industry is racing towards artificial *general* intelligence — systems that can do anything a person can. That is a serious and legitimate research goal, and I follow it with genuine interest. But it is not the question a compliance function needs answered.

Professional work does not fail for lack of general intelligence. It fails for lack of **professional context**: the method, the reader, the evidence, the house view, the regulation as it actually reads today. So in March we gave the idea a name: **APCI, Artificial Professional Context Intelligence.** Not a different kind of model, but a different place to look for the value — in the professional context *around* the model.

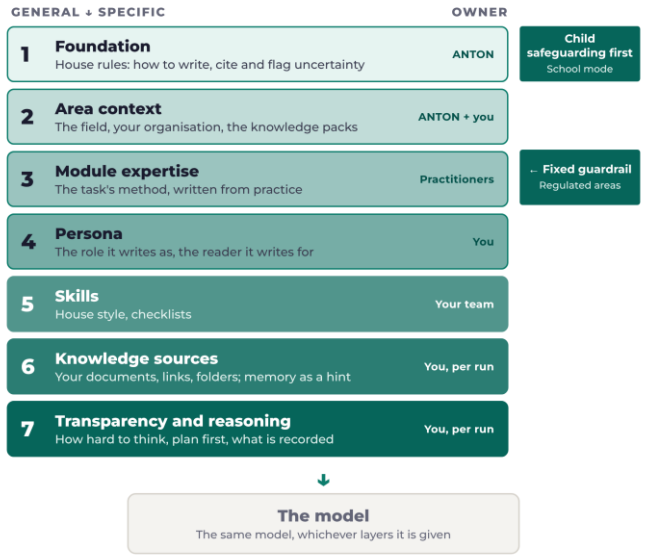
Three things follow from that shift of focus, and they shape everything in this paper.

Intelligence is becoming a utility. Capable models are now available to everyone, and next quarter there will be a better one. Building your advantage on access to a particular model is building on sand. It is the equivalent of a firm claiming its edge is that it has electricity.

Context is where the value lives. Two firms using the same model get very different results, and the difference is almost entirely in what surrounds the model: what it is told, in what order, from which sources, for which reader. That is where years of professional experience can be captured — and that is what the seven layers are for.

Context stays yours. This is the part I find most important. A model is rented; it will be replaced. But a well-written method, a curated knowledge pack, a house style and an organisation profile belong to you, and they work with whichever model runs underneath. **When the model changes, your context does not reset. It compounds.**

The seven layers are APCI made practical: a way of writing professional context down, in pieces, so that it can be owned, shared, checked and carried from one model to the next.



General to specific. Each layer has a source and an owner.

An analogy: the graduate, again

In the first paper I described today's models as the most brilliant graduate who ever walked through your door — someone who has read everything and done nothing, and who forgets everything you told them each night.

The seven layers are the induction programme for that graduate. Not a bigger brain; a better briefing. The house rules on the first morning. The field and the firm by lunchtime. The method for the task, with an example of what good looks like. A word about who will read the result. The team's checklist. The case file. And a reminder to think it through and say what they could not check.

What is new is that, with layers, the briefing can be written once and given in full, every single time, without anyone having to remember to give it.

What we are not claiming

We are **not** claiming that the layers make an answer correct. They make it far more likely to be relevant, specific and well grounded, and they make it possible to see where it came from. Checking whether it is right is still the job of the process and the people we described last week.

We are **not** claiming that every source you provide always reaches the model. There is a limit to how much text a model can take in at once. When there is more material than room, some is left out — and ANTON says so, in a note, rather than dropping it silently.

We are **not** claiming that the AI "knows your organisation". It is *told* about your organisation, through the layers, on each run. That is a more modest claim, and a much more useful one: you can see what it was told, and change it.

And we are **not** saying the model does not matter at all. With the right context, a smaller model can do a great deal — that is next week's paper — but there are still tasks where a larger model makes a real difference.

Part 3 · Our take: seven layers, in order

The seven layers

These are the seven layers as they appear on our AI portal, with an example from financial crime prevention for each. In ANTON, every one of its more than 550 modules is assembled this way.

1. Foundation. The house rules: how to write (strict, balanced or creative), how to structure an answer, when to flag uncertainty. It is written once for ANTON and applies to every module. *In FCP:* "Cite the article. Distinguish 'shall' from 'should'. Say what you could not check."

2. Area context. The field you work in, your organisation's context, and the regulatory knowledge packs that apply — a knowledge pack being the regulation's text and key points, with its source and effective date. *In FCP:* financial crime prevention; "a Swedish payment institution, supervised by FI"; the AMLR and EBA AML guideline packs.

3. Module expertise. The expert method of the task: what to do, in what order, what the output looks like, and what to avoid. Each module carries its own instructions, written from practice. *In FCP:* the AMLR gap analysis — a per-article comparison, a severity scale, and the rule that "silence in the documents is itself a finding".

4. Persona. Optional: the role the AI writes as, or the reader it writes for — and, if you are continuing earlier work, where you left off. *In FCP:* "write as an internal auditor"; "for a board that reads two pages".

5. Skills. Optional add-ons that you or your team build or import: a house style, a checklist that always applies. *In FCP:* the firm's gap-report layout; a sanctions checklist.

6. Knowledge sources. What the answer may draw on: the model's own knowledge (optionally with web search), links you paste, folders on your machine, a search over your own documents, and — only when a gate opens — earlier knowledge atoms, labelled as hints. *In FCP:* the firm's CDD procedure; the EBA guideline's web address; last year's assessment.

7. Transparency and reasoning. How hard to think, from "quick" to "deep investigate"; whether to plan first and show the plan; and what is recorded. *In FCP:* "think hard", "plan first", and end with "Sources, assumptions and what was not checked".

Two safety additions

- **A fixed guardrail in regulated areas.** In financial crime, legal, tax, data protection, blockchain and other regulated areas, a fixed instruction is added: the AI does not make compliance decisions or give legal advice, and a person keeps responsibility.
- **Child safeguarding comes first.** In ANTON's School mode for young learners, a safeguarding layer is placed before everything else and outranks every other instruction.

Who owns each layer

One way to see why the layers matter is to ask who writes each one, and when it changes. Notice how many of them are yours.

Layer	Who writes it	When it changes
1 Foundation	ANTON, once	Rarely
2 Area context	ANTON and the knowledge packs; your organisation profile	When the rules or your firm change
3 Module expertise	Practitioners, per task	When the method improves
4 Persona	Chosen by you	Per deliverable
5 Skills	Your team	When your house style changes
6 Knowledge sources	You, per run	Every run
7 Transparency and reasoning	You, per run (with a sensible default per module)	Every run

Your organisation profile, your skills, your sources and your choice of reader and reasoning are all *yours*. Those are the parts of the context that no model vendor can supply and no model upgrade can replace. That is APCI in a table.

A note for readers of our earlier whitepapers: *"The Starting Point"* described seven layers and *"The Context Layer"* nine; in the code a prompt is built from about thirty named parts. It is the same idea at different magnifications, and the seven on the AI portal are the clearest grouping.

How the layers are put together

The order of the layers is not decoration. It reflects a deliberate rule about priority, and a few safeguards that matter in regulated work. In plain words:

1. **General before specific.** The stable parts come first: safeguarding where it applies, the foundation, the organisation's context, the knowledge packs and the framework text, the area, the module and the guardrail. Then come the parts that change with each run: today's date, your profile, tone and language, the persona, skills, the output format, the section on sources and assumptions, the reasoning settings, and finally your own material.
2. **Today's date is always stated.** Models are trained up to a point in time. Without being told the date, they answer date-dependent questions from the wrong side of the line — treating a regulation as "upcoming" when it already applies, or the reverse. ANTON states the date in every call to a model.

3. **Your documents are fenced.** Uploaded and fetched text is wrapped and labelled as reference material, so that text inside a document cannot act as an instruction. Common injection phrases — *"ignore your previous instructions"* and the like — are filtered out.
4. **Sources have an order and a budget.** When there is more material than room, sources are taken in a fixed order — uploads, then project files, then links, then folders, then search results — each one whole or not at all. Anything left out is named in a note. Nothing is silently dropped.
5. **Memory is a hint, not a source.** Knowledge atoms — small pieces of learning from earlier work, the subject of WP8 — are added only when a gate opens. In the automatic setting, that means at least 100 atoms from real module work and 30 ratings. Even then, at most five go in, under a heading that says plainly that they are memory, not verified sources. On a typical installation today, that gate is still closed.
6. **Packs follow the area.** Knowledge packs and the text of framework articles are chosen for the area of the task. They are not sprayed into every run.
7. **Every layer is recorded.** The record of each run keeps each layer's name and fingerprint, so a run can be shown, compared and replayed.

The fifth point deserves a moment. It would be easy, and tempting, to let an AI "remember" earlier work and feed it back in as fact. We chose not to. **A memory of what was concluded last time is not evidence for what is true this time.** So memory goes in last, labelled, limited and gated — a hint for the model, never a source for the reader.

Layer 6 up close: four kinds of knowledge source

Layer 6 is where your own material comes in, and ANTON offers four ways to supply it.

- **The model's own knowledge**, optionally with web search — useful for general questions, and always with the caveats about memory above.
- **Online references.** Paste links; the server fetches the page and extracts the text, with safety checks on where it is allowed to fetch from.
- **Local folders.** Point to folders on your own machine. They are indexed, and only folders you have allowed can be read.
- **Combined.** Merge sources with a priority rule: local first, the model's knowledge first, or merged.

There is also a search over your own documents, which sends the most relevant passages rather than everything. Some options are meant for your own installation and are not offered to visitors on the public showcase.

Picture: ANTON's "What will be sent" preview, showing the assembled layers of one run (showcase, fictional data)

What we have built

ANTON's prompt builder. For each of its more than 550 modules in 60 areas — 38 of them in financial crime prevention alone — ANTON assembles the layers in the order above, records them in the run record and shows them before a run in a **"What will be sent"** preview. All of this is in ANTON's main code, which is open source.

More than 40 regulatory knowledge packs, each with its source and effective date: EU AML rules, AMLA standards, EBA guidelines, sanctions, DORA and NIS2, MiCA, FATF, GDPR and the AI Act, Nordic laws and supervisors, and several non-EU jurisdictions. And for the area's frameworks, the **text of the articles themselves** reaches the module, so the AI reads the article rather than its memory of it. That is the answer to Scene 3.

Guided input for each module. The AMLR gap-analysis module asks for the type of institution, the jurisdiction, the AMLA supervision category, the AMLR areas to focus on, and any known issues — and the area context is already half written.

Blueprint skills for financial crime deliverables. Added to ANTON's main code on 10 October 2026: thirteen deliverable blueprints from practice — among them the gap report, the BWRA report, the model validation report and governing documents — plus a set of house standards, all as skills. They change the wording, the evidence asked for and the layout. **They never change how a score is computed.** Modules suggest them; they are attached automatically only when you choose the matching output format. And a test fingerprints the default instructions of every module, so that a run with no skill selected stays exactly as it was. This is the senior analyst's notebook, turned into something the whole team can use.

The same idea across the family. Valentin, Olivia and Horizon build every prompt from archived templates and named parts, under the same "what went in" contract we described in the second paper, and every prompt states which knowledge pack it used and why. Different experts, the same discipline.

Status at a glance

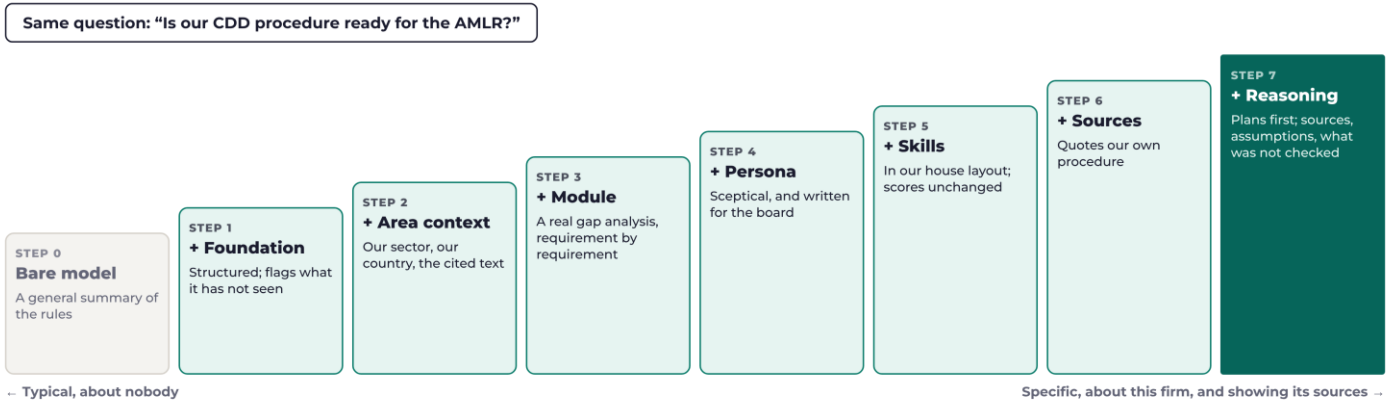
- **Live** (public showcases, fictional data): the seven layers; four kinds of knowledge source; guided module forms.
- **In main code** (merged, not checked as deployed): layer order and safeguards; memory gate; "What will be sent"; blueprint skills.
- **Built, not released:** outside data connections (registers, screening) — prepared, not connected.

Part 4 · A worked example

One question, seven layers

The question is one every MLRO will be asked in the next few months: *"Is our customer due diligence procedure ready for the AMLR?"* The firm is *Exempelbolaget AB*, a fictional Swedish payment institution. We ask the question once with no context at all, then add one layer at a time and watch what changes.

This example is illustrative. It describes what each layer is designed to change, based on how the layers work. It is not a recorded experiment, and the wording below is not output copied from a run.



The model did not get smarter. It got context.

Step	Layer added	What changes in the answer
0	*None: the bare model*	A correct, general summary of the AMLR's CDD rules. No reference to the firm. Some article numbers may come from the proposal rather than the final text.
1	**Foundation**	Structured: a summary, findings and actions. It flags uncertainty, and says plainly: "I have not seen your procedure; this is what is typical."
2	**Area context** — FCP, the organisation profile, the AMLR and EBA packs, the framework text	It speaks as an FCP practitioner. It knows the firm is a payment institution in Sweden, supervised by FI. It cites articles from the pack's text, with the source, and notes that the AMLR applies from 10 July 2027.
3	**Module expertise** — the AMLR gap analysis	It becomes a gap analysis: requirement by requirement, with a severity scale, "shall" kept apart from "should", the rule that silence in the documents is a finding, and a prioritised action plan.
4	**Persona** — an internal auditor, writing for the board	The tone becomes sceptical; it asks for evidence. The executive summary fits on two pages.
5	**Skills** — the firm's gap-report blueprint and house standards	The layout the firm always uses, and the evidence fields it always asks for. The scores do not change; blueprints never change scoring.
6	**Knowledge sources** — the firm's CDD procedure, uploaded	The findings now quote the procedure. *"Verification of beneficial owners relies on registry extracts alone"* appears as a finding, with the quote. Gaps the documents are silent on are marked as such.

Step	Layer added	What changes in the answer
7	**Transparency and reasoning** — think hard, plan first	It shows its plan first. It ends with "Sources, assumptions and what was not checked", and a confidence level for each finding. The run record stores all seven layers with their fingerprints.

What changed, and what did not

Read the table from top to bottom and notice what did *not* change: the model. The same model answered at step 0 and at step 7. It did not become more intelligent along the way. It was given, one layer at a time, what a professional would have known.

Notice, too, *which* layers did the heavy lifting. Step 2 turned a general answer into an answer about this firm, under these rules, citing this text. Step 3 turned an answer into a method. Step 6 turned a method into findings about the firm's actual procedure. Without those three, the most capable model in the world produces Scene 1: accurate, and useless to the person who asked.

And notice where the work of each layer came from. The foundation and the module were written once, by people who know the work, and will be reused thousands of times. The knowledge packs were curated once, with their dates. The persona, the skill and the uploaded procedure came from the firm — from its own way of working and its own documents. **Most of what made the final answer good belongs to the organisation, not to the model.** If the firm switched to a different model tomorrow, nearly all of it would carry straight over.

That is the practical meaning of APCI. The intelligence was rented. The context was owned.

There is one more thing the table shows, and it links back to last week. Because each step is a separate, named layer, a reviewer can see exactly what the model was given at each point. If the step-7 answer contains a weak finding, she can ask a precise question: was the method wrong, was the pack out of date, or was the procedure we uploaded the old version? **Layers turn "the AI got it wrong" into a question that has an answer.**

Part 5 · From here

What this means for you on Monday

You can start building your own layers this week, with or without any of our software:

1. **Write your foundation.** Half a page of house rules for any AI-assisted work: cite the source, separate fact from judgement, say what was not checked.
2. **Write your organisation context once.** Sector, licences, country, supervisor, size, and your risk appetite in two lines. Give it to every tool that allows it.
3. **Choose your packs.** List the regulations and guidelines your team actually works with. Check that your AI tool holds the final text, with its effective date — not a memory of a proposal.
4. **Turn the notebook prompt into a module.** Take the best prompt in your team and split it: the method (a module), the house style (a skill), the reader (a persona). Now everyone has it, and it stays when its author is on holiday.
5. **Ask to see the stack.** Before you trust a tool, ask it to show you what it will send. If it cannot, you cannot know which layer is missing.

Next week: setup beats size

If the context is right, do you still need the biggest model? Next week we offer a simple six-step framework for any AI task — the problem, the module, the knowledge, the format, the thinking level and, only last, the model — and we look at what happened when a smaller, cheaper model ran with the full seven layers around it. With honest notes on how small the evidence is, and on where a bigger model still matters.

A closing thought

When people first see the seven layers, they sometimes say that it looks like a lot of work. It is — once. That is the point.

Every compliance function already has this knowledge. It is in the senior analyst's notebook, in the board's feedback, in the house templates, in the policies, in the heads of people who have done the work many times. What it usually does not have is a structure that lets that knowledge reach the AI in the right order, every time, for everyone — and that shows afterwards which piece of knowledge did what.

The models will keep getting better, and that is good news. But the firms that get the most from them will not be the ones that wait for the next model. They will be the ones that wrote their context down — in layers, with owners — so that every new model starts from everything they know. **Intelligence is becoming a utility. Context is where the value lives.** And context, unlike a model, stays yours.

— 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", and the worked example in Part 4 is illustrative, not a recorded experiment. 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.