

BANKING, FINANCIAL SERVICES AND INSURANCE

The compensation layer

For 20 years, people quietly fixed our data on the way past. Agents don't.

Author:
Rav Hayer, Head of UK/I and EMEA BFSI, Thoughtworks

/thoughtworks

Design. Engineering. AI.

Opening 3

1. What we actually built,
and who was holding it together 4

3. Meaning is the actual work 8

4. Privacy is where this gets serious 11

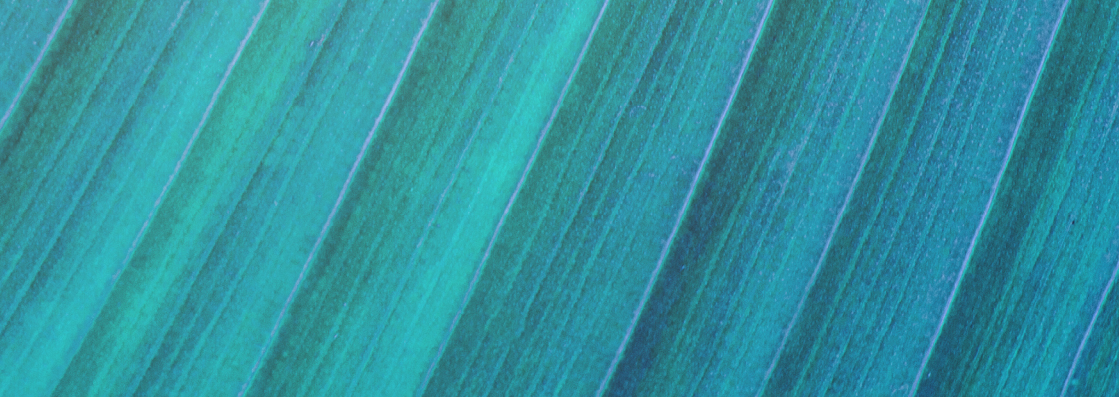
5. What's really stopping people 15

6. What seems to work 19

7. Questions I tend to ask 20

Close: where that leaves us 22

Sources 23



Opening

The pilot worked. Then we tried it on real data.

I've lost count of how many times I've heard a version of that sentence this year. Different banks, different insurers, same shape. Somebody builds an agent that does something genuinely useful, everyone is delighted, and then it meets the estate and quietly falls over.

The usual diagnosis is that the technology isn't ready. I don't think that's it. In most of the cases I've seen up close, the agent behaved exactly as designed. What it met was a data estate that had been getting by for years on something nobody had written down.

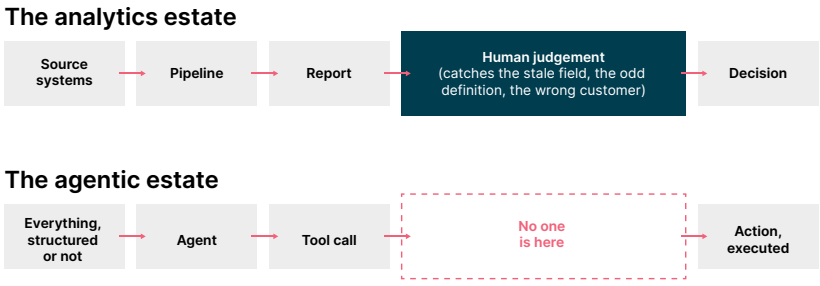
Think about who actually sat between your data and your decisions. An analyst who knew that status code seven meant a closed account, but only on the platform you picked up in the 2019 acquisition. A pricing manager who knew the retail balance file was unreliable on the first working day of the month and simply waited a day. A claims lead who could tell, at a glance, when a number looked wrong. None of that knowledge was in a catalog. It was in people, and it worked well enough that nobody ever had to fix the underlying thing.

That's the compensation layer. Agentic AI removes it. Once you look at your estate through that lens, a lot of the current confusion resolves fairly quickly.

1. What we actually built, and who was holding it together

The analytics stack most large institutions run was designed around a human reader: a small number of curated outputs, consumed by a lot of people, on a weekly or monthly rhythm. If something was wrong, someone noticed, raised a ticket and the pipeline got fixed. Slow, but self-correcting.

Agents invert every part of that. Lots of autonomous consumers pull a wide and unpredictable surface of data on demand, often several times inside one task, and act on it straight away. Nobody spots that the counterparty reference file stopped refreshing on Tuesday. The cost of a stale field used to be a misleading chart. Now it's a decision, taken and executed, before anyone has looked.

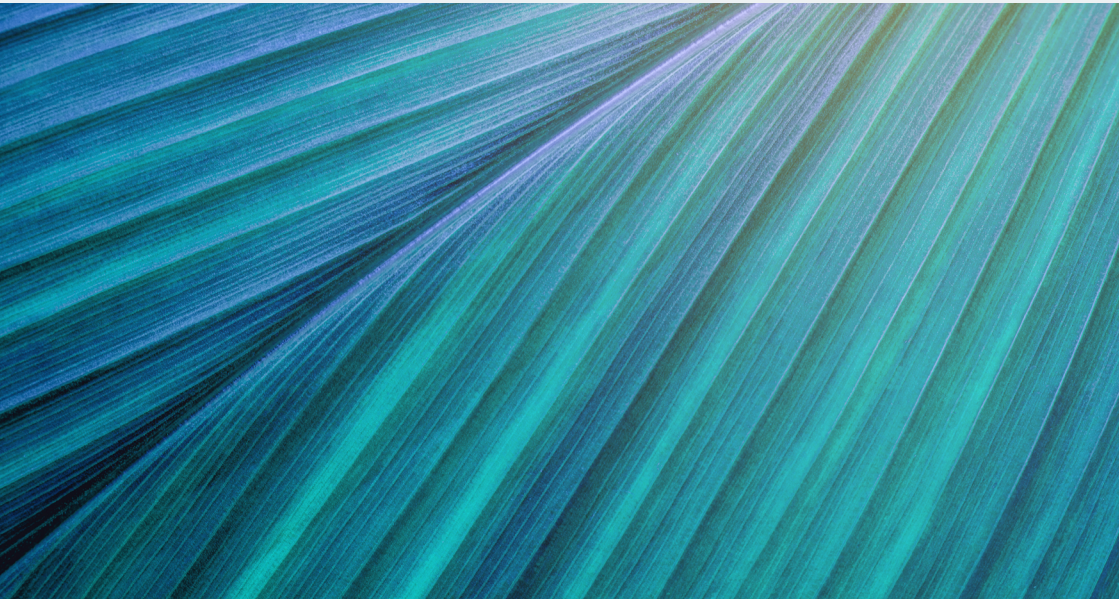


The step in the middle was never in anyone's architecture diagram. That is precisely why removing it caught so many people out.

There's a second-order effect here that I think matters more than the first. We spent a decade optimizing for data being correct. Correct is no longer sufficient. Data now has to be interpretable by something that can't ask a colleague.

Two fields called customer status, both accurate, both maintained, meaning slightly different things in different parts of the group,
was never a real problem when a person chose between them. Give that choice to an agent and it will make one, confidently, and carry on. It won't flag it. It won't raise a ticket. You'll find out later, from a customer or a supervisor.

Data you hold but cannot use isn't an asset.
In an agentic estate, it's a liability with a storage cost.



2. Where the conversation has got to this year

Three things have shifted the tone of the boardroom conversation since spring, and they don't point the same way. That's part of why this year feels harder than last.

The money has gone industrial. Goldman Sachs Investment Banking's recent work on AI in the real economy makes a point I keep returning to. This build-out is faster, broader and more capital intensive than previous technology cycles. It's drawing in genuinely new pools of capital, including sovereign money and structured credit, and their view is that incumbents will have to refound around AI rather than defend existing product economics.

For our clients, that lands twice over. Banks are financing the build-out and being reshaped by it at the same time. What I'd add is that the incumbent advantage everyone falls back on, all that proprietary data, only counts if it's usable. A lot of it currently isn't.

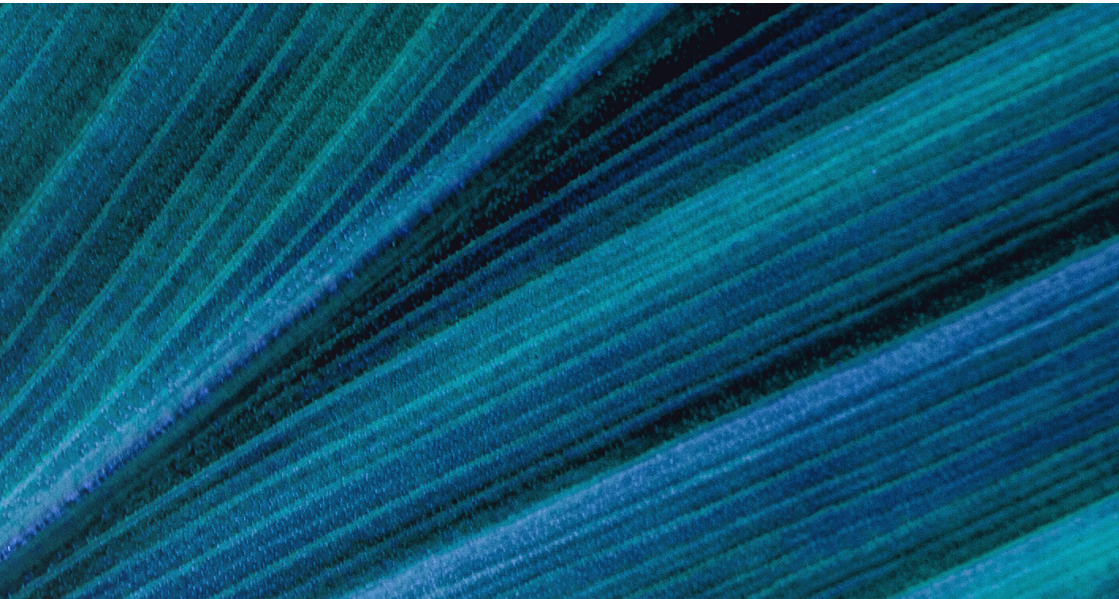
Governance is behind, and the gap is measurable. McKinsey's 2026 AI Trust Maturity work, covering around 500 organizations, found average responsible AI maturity had risen to 2.3 from 2.0. Encouraging. Less encouraging: only about a third reach level 3 or above on strategy, governance and agentic controls specifically. Technical capability is running well ahead of oversight.

The finding I'd draw your attention to, though, is a quieter one. Across almost every risk category, the share of organizations who consider a risk relevant is materially higher than the share

actively mitigating it, and the gap is widest on personal privacy and intellectual property. People know. They haven't built the controls yet.

The security picture stopped being theoretical. Compare the 2026 security literature with the 2025 editions and the difference is stark. Last year's cataloged plausible threats. This year's catalogs actual CVEs, vendor advisories and breach reports. Prompt injection is still top of the OWASP list and is now generally treated as an architectural property rather than a bug someone will eventually patch. Most incidents come from indirect injection, where the instruction is hidden in a document, an email or a web page the agent has been asked to read.

Alongside that, Verizon's 2026 breach report found shadow AI detections up fourfold in a year. In April, American Banker ran a piece on an architectural weakness in the integration protocols banks are adopting for agents. None of this is a reason to stop. It's a reason to be specific about what you switch on, and in what order.



3. Meaning is the actual work

When I sit down with data leaders now, the conversation usually starts on platforms and models and ends up somewhere much less exciting: definitions.

Here's the list I keep coming back to. None of it is new thinking. What's changed is that each of these used to be good practice and is now load-bearing.

	From		To
01.	**Schemas and tables**	→	Semantics, in code and versioned
02.	**The structured estate**	→	Every document, call and note
03.	**Access control on tables**	→	Entitlement carried through the chain
04.	**Lineage between datasets**	→	Evidence behind each decision
05.	**Pipelines that move data**	→	Tools that take action, governed
06.	**Stateless consumption**	→	Memory as a governed data class
07.	**Batch throughput**	→	Relevance, freshness and cost per call
08.	**Data quality monitoring**	→	Evaluation sets as data products

Eight shifts. The right hand column is where the work sits, and most of it is unglamorous.

Semantics before schemas. The deliverable stops being a well-formed table and becomes an agreed, machine-readable definition of what the business means. Ontologies, semantic layers and metric definitions are production infrastructure now, not documentation. If three parts of the bank define an active customer differently, and those definitions live in two PDFs and someone's head, your agents will give three answers and every one will be defensible.

All of the estate, not just the structured bit. Most of what a bank or insurer knows sits in policy wordings, credit papers, call recordings, complaint files, KYC packs, adviser notes and decades of email. We governed it lightly because it wasn't queryable at scale. Language models made it queryable overnight. This is simultaneously the biggest prize and the biggest exposure, and it's the same pile of material.

Entitlements that travel. Traditional access control asks whether a user may read a table. Agents ask a harder question. When one acts for a customer, whose permissions apply, and how do those permissions survive a chain of retrievals, tool calls and sub-agents? Get it wrong and you've built a very helpful, very privileged assistant that will surface whatever it can reach. In retail banking, that's a data protection incident and a conduct issue in the same breath.

Evidence, not just lineage. Lineage tells you which table fed which table. What a supervisor wants is why a particular decision was made about a particular customer at a particular moment: which policy version applied, what got pulled into context, which tools were called, what came back, who approved it. You capture that at the point of action, or you don't have it at all. There's no retrofit.

Tools are part of the data architecture. Agents don't only read, they act, through APIs and workflows. So the tool catalog needs the same discipline as the data catalog: what it does, what it costs, its blast radius, who can call it, whether it can be undone. Connecting things has become cheap, which means governing the connections is now the constraint. One published analysis found a substantial share of the server implementations it examined were vulnerable to command injection, with documented cases of hidden instructions pulling data out.

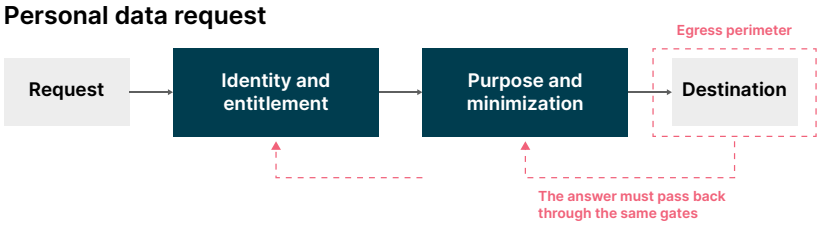
Memory is a new data class, and it's personal data. Agents accumulate conversation history, preferences, working notes and prior decisions. It may be wrong. It's subject to access and erasure rights. In most implementations, it sits entirely outside the governed estate. Researchers have now demonstrated memory poisoning in production systems, where a corrupted long-term memory produced persistent false beliefs that the agent then defended when challenged: a fairly unsettling failure mode. I have yet to meet a firm with a retention policy for it.

Relevance, not throughput. The dominant cost is usually context assembly rather than compute: finding the right small slice at the right moment. Teams built for moving everything on a schedule now need to optimize for precision, latency and cost per successful outcome, which is a different craft.

Evaluation as a data product. Quality monitoring tells you a column is null. Evaluation tells you whether the thing is producing good outcomes. Golden datasets, regression suites for agent behavior, red team corpora and human review samples need to be owned, versioned and funded. Skip this and you can't safely change anything once you're live, because you have no way of knowing whether a change made it worse. This is, I think, the single most common reason pilots never graduate.

4. Privacy is where this gets serious

Privacy has been a compliance discipline for a decade. In an agentic estate, it becomes an architectural one, because the thing processing personal data is no longer a system you configured. It's a system that decides, in the moment, what to fetch, what to combine, what to keep and what to do next.



Every boundary is a gate the request must pass, and a gate the answer must pass back through.

What the regulators have been saying

Quite a lot, actually, and it's worth reading in sequence because the direction is consistent. The ICO set out its early view on agentic AI in January, mapping the risks it thinks are genuinely novel. Spain's AEPD followed in February with a long and rather good technical guide. The Dutch authority consulted in April on the right to an explanation for automated decisions. In July, CNIL published a joint exploratory note with France's AI and digital council, describing agentic systems as programs capable of autonomous decisions, orchestration of complex actions and

interaction with third-party services, with or without human validation. Guidelines from the EDPB and the Commission on how GDPR and the AI Act fit together are expected by the end of this year.

Closer to home, the Digital Regulation Cooperation Forum, which brings the FCA, ICO, CMA and Ofcom together, published a cross-regulatory foresight paper on agentic AI in March, and the CMA put out consumer research and trader guidance the same month. The common thread across all of it is straightforward: the existing frameworks already apply, and you're accountable for what your agent does regardless of whether the thing doing it was a person or software.

Six places I see privacy break



Purpose limitation erodes without anyone deciding it should. Data collected for servicing gets pulled into a marketing task because it was retrievable and relevant. No human made that call. The retriever did.

Minimization runs backwards. Retrieval is engineered to find more context, not less. The technical incentive and the legal principle point in opposite directions, and at the moment the technical one usually wins.



Controller and processor roles move mid-process. The agent calls a third-party API, pulls from an external register, spawns a sub-agent. The tidy processor annex in your DPA no longer describes what happened.



Erasure doesn't propagate. A deletion request has to reach the warehouse, the document store, the vector index, the cached embeddings, agent memory and the trace logs. Most firms I've asked can confidently reach the first two.

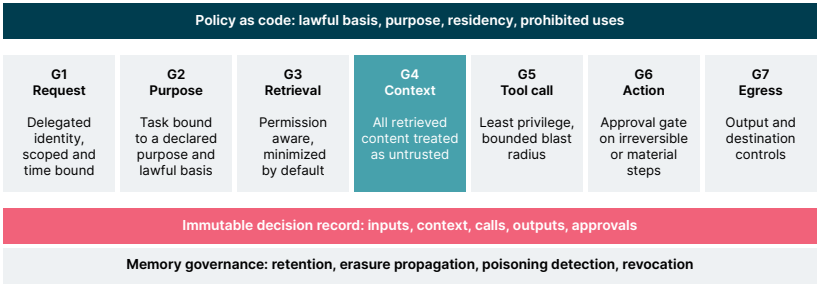
Automated decision making is back on the table. The Data (Use and Access) Act relaxed parts of the Article 22 regime but added safeguards around human review and contesting a decision. If an agent is anywhere near lending, pricing or claims, contestability has to be designed in.

Injection turns a privacy weakness into an exfiltration route. An instruction buried in a support ticket or an uploaded document can persuade an agent to retrieve more than the requester was entitled to and send it elsewhere. The agent holds the privilege, so nothing at the network layer objects.

The bit that trips people up: conventional security assumes an attacker exploits code or steals credentials. Agents can be compromised through the content they were built to read. One sentence in a retrieved document can redirect behavior, with no malware and nothing stolen. That's why privacy controls have to live in the request path rather than in the policy library.

Guardrails, and where they have to sit

Guardrails written as principles don't hold. Guardrails that are enforcement points a request has to pass through do. The sequence below is ordered by where in the lifecycle each one acts, because that's the only ordering that helps you build it.

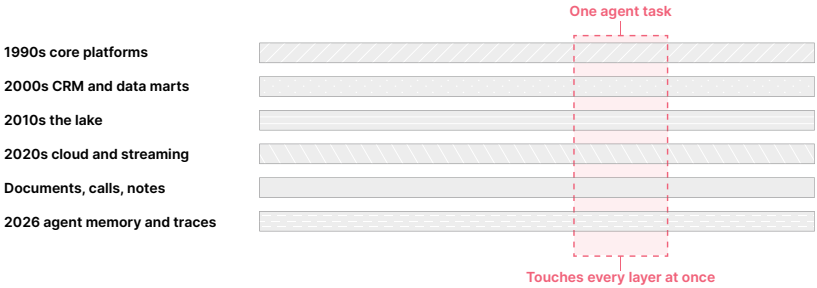


Seven gates, a policy layer that binds them and a record layer that proves it.
Gate four is highlighted because it is the one most commonly missing.

- Give agents their own identity. Scoped, short-lived credentials, acting under the requesting user's entitlements rather than a shared service account that can see everything.
- Bind purpose at the tool boundary. A task carries a declared purpose and lawful basis. Tools refuse calls whose purpose doesn't match.
- Minimize in context assembly. Return the least sufficient context, redact what the task doesn't need and prefer tokenized or pseudonymized forms wherever the work allows.
- Treat everything retrieved as untrusted. Structural separation between instruction and content, not a filter that tries to spot bad wording.
- Bound the blast radius of tools. Value limits, rate limits, reversibility requirements and an explicit list of things the agent may never do.
- Put a human on the irreversible steps. Not on everything. On the ones a supervisor would ask you about afterwards.
- Control egress. Where output can go matters as much as where input came from. Most exfiltration is a legitimate action aimed at the wrong destination.
- Govern memory properly. Retention limits, erasure that actually reaches vector indexes and agent memory, provenance on what's been remembered and detection for poisoning.
- Write the decision record at the point of action. Retained to the standard your supervisor expects and queryable by customer, not by log timestamp.
- Be able to pull the plug fast. Disable the agent, revoke credentials and quarantine the memory, in minutes. Test it like you test any other recovery capability.

Two things that consistently get missed. The DPIA needs to be per use case and revisited whenever the tool set changes, because adding a single tool can change the processing materially. And contestability is a data requirement rather than a customer service one. If someone challenges an automated outcome and you can't reconstruct the inputs, no amount of good complaints handling will get you there.

5. What's really stopping people



The estate was laid down in eras, each with its own definitions, owners and controls.

Every large institution I work with has the same customer represented differently across a dozen systems, thanks to mergers, unfinished core migrations, regional variants and outsourcing. The pilot succeeded because somebody hand-curated a clean slice. Production failed because there's no such slice for the other 90%.

The instinct at this point is to go and fix the estate. I'd resist it. That instinct has produced a generation of stalled data programs and a lot of skeptical sponsors. Fix the estate along the thin slice each use case genuinely needs, to production standard, and let the coherent core accumulate behind you.

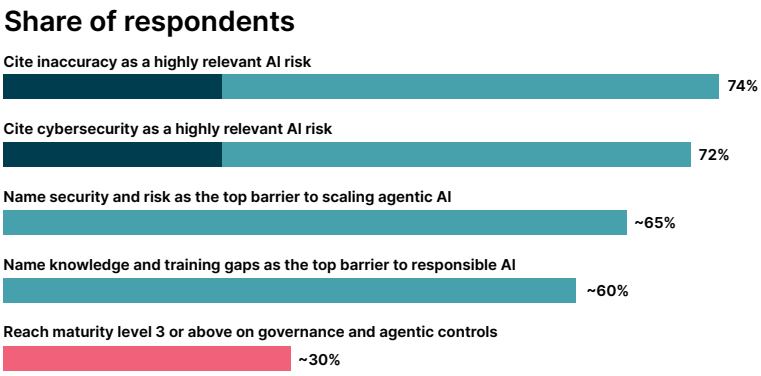
Semantic debt. The constraint that bites hardest and gets named least. Nobody has ever agreed, in a form a machine can read, what the core entities and metrics mean. We coped because humans arbitrated. Agents can't arbitrate, so the ambiguity comes out as inconsistency, and inconsistency in a regulated firm is a finding.

Governance built for a different job. Most data governance answers reporting and privacy questions quarterly, through committees, with manual attestation. Agents decide in seconds. Governance that can't execute at runtime is, in practice, decoration. The distance between the policy document and the enforcement point is where incidents live.

The unstructured vacuum. Nobody owns the shared drive. Nobody classified the 12-year-old collaboration site with draft credit committee papers on it. Retrieval is extremely good at finding the document you'd forgotten existed. My honest expectation is that the first serious incident at most firms won't be a hallucination. It'll be a correctly retrieved document that should have been deleted in 2018.

Shadow AI. Related, and larger than most risk registers admit. The data leaving through unsanctioned tools is, for a lot of firms today, a bigger exposure than anything their sanctioned agents are doing.

You can’t see what they’re doing. Very few organizations can tell me what a deployed agent did last week, how often it was right, where it degraded and what changed. Prompts, tool calls, intermediate reasoning and responses sit in different systems, when they’re captured at all. Without a consistent trail, you can’t reconstruct why an agent acted, which hurts incident response and leaves you with nothing to show a supervisor. So the choice becomes deploy blind or don’t deploy, and sensible people pick the second.

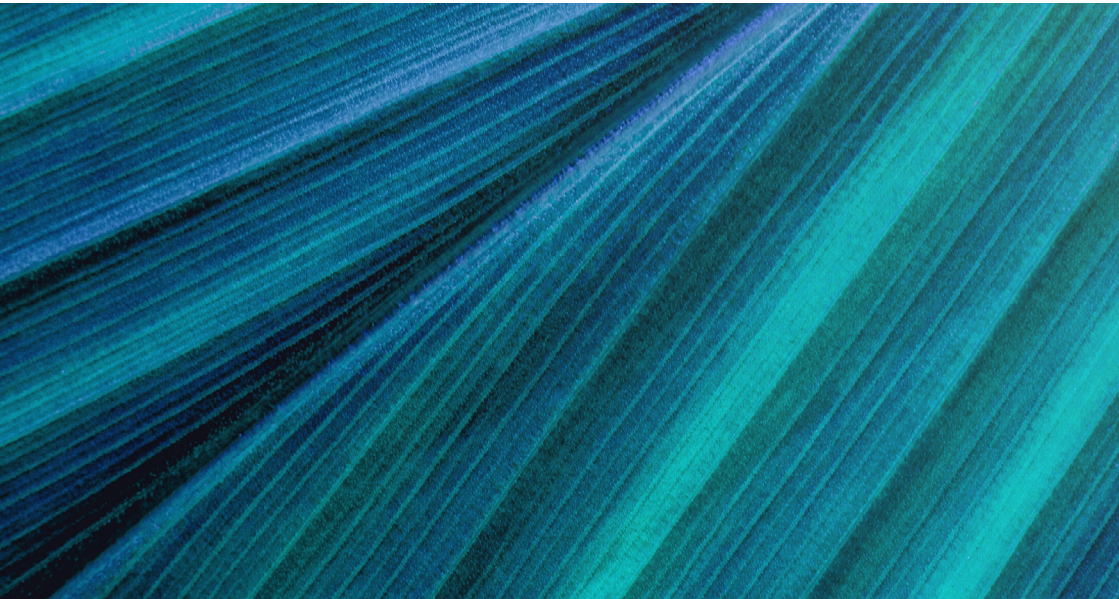


Average maturity 2.3, up from 2.0. Clear RAI ownership scores 2.6 against 1.8 without. Findings from McKinsey’s 2026 AI Trust Maturity Survey, redrawn. Awareness is high nearly everywhere. Controls are not.

The economics. Retrieval, inference, trace and memory storage, and evaluation runs all cost money, and unlike traditional software, the cost per interaction scales with use. An agent that works isn’t the same thing as an agent that pays. Put cost per successful outcome in front of the team in the first sprint rather than in front of the CFO in month nine.

Concentration. Depending on a handful of model and cloud providers is now a supervisory question in its own right, with the Critical Third Parties regime in force and continued pressure on HM Treasury to designate the major AI and cloud providers. Architecturally, the answer is portability. Keep semantics, context assembly, evaluation and orchestration under your own control and treat the model as something you can swap.

And the operating model, as always. Platform teams, AI teams, product teams and risk functions sit in different structures with different incentives and different funding cycles. Agents cut across all of them. If the operating model doesn't change, the architecture will mirror the org chart, and the seams will show up as exactly the accountability failures regulators are currently asking about.



6. What seems to work

I'm wary of best-practice lists this early in a technology shift, so treat the following as what we've seen hold up in delivery rather than as settled doctrine.

Contracts, not just data products. A data product is agent-ready when it publishes a machine-readable contract covering meaning, quality guarantees, freshness, ownership, entitlements and known limitations. The contract is the deliverable.

One semantic layer to argue about. Rather than trying to harmonize every source system, which never finishes, put the investment into a governed semantic layer that agents query, where definitions live in code, are versioned and are tested. Have the argument about what a customer is once, in one place.

Policy as code, in the request path. Entitlements, retention, residency, purpose limitation and prohibited uses: committees set the policy, code enforces it, and fitness functions catch the drift.

Observability from the first line, not the first incident.

Treat agents like colleagues rather than deployments. This is the framing that has landed best with risk functions. An agent gets onboarded into a defined role with least-privilege access, a named supervisor, a probation period on low-risk work, ongoing review and a proper offboarding. It maps onto how SMCR already thinks, and it survives contact with a regulator, which is more than most AI governance frameworks manage. It also matches what supervisors are asking for: show what the agent was allowed to do, what it actually did and how humans governed the outcome.

Thin slices. Pick a use case with real economic value. Build only the foundation it needs, properly, including semantics, entitlements, privacy gates, evaluation and observability. Ship it. Reuse it on the next one. The platform emerges from delivered value instead of being a precondition for it. Of everything here, this is the decision I'd fight hardest for, and it's the one most often got wrong.

Next 90 days	6 to 12 months	Beyond
• Pick two or three use cases with real value and bounded risk	• Repay semantic debt on the 20 entities that matter	• Publish contracts across the core data product estate
• Map the data each one touches, structured and unstructured	• Classify and own the unstructured estate	• Set memory retention and erasure policy properly
• Stand up tracing and an evaluation harness in the pilot	• Move governance from documents to enforcement in the path	• Build for provider portability
• Run a permissions threat model and a per use case DPIA	• Formalize agent onboarding and named accountability	• Measure time to first safe production use case

A sequence rather than a program plan. Each horizon should be paid for by the use cases inside it.

7. Questions I tend to ask

If you take nothing else from this, these are the ones that reliably tell me where a firm actually is.

1. If an agent had to choose between two similarly named customer fields, could it, without a human?
2. Who owns the unstructured estate, and what would a retrieval system find in it today?

3. When an agent acts for a customer, whose permissions apply, and where is that enforced?
4. Does a deletion request reach the vector indexes, agent memory and trace logs, or only the warehouse?
5. Have you tested what happens when instructions are hidden in a document a customer sends you?
6. Could you produce, tomorrow, a full evidential record of why an agent decided something last month?
7. Which of your AI controls execute in the request path rather than in a committee?
8. How quickly can you revoke an agent's credentials and quarantine its memory, and when did you last try?
9. What is your cost per successful agent outcome, and who watches it?
10. How much data is leaving through tools you haven't sanctioned?
11. If you had to change the primary model provider in 90 days, what breaks?
12. Are you funding a platform program, or use cases that leave a platform behind them?

Close: where that leaves us

Nothing in this paper is a new data problem. Every ambiguous definition, every ungoverned document store, every overbroad service account and every undocumented workaround was already there. What's changed is that the people who used to absorb all of it have been taken out of the path, and agents act on it faster than anyone can review.

I find that more encouraging than it sounds. Most data leaders I know have been raising these things for a decade without much traction. There's now a board-level, economically justified reason to deal with them, the capital is moving, and the regulators have been unusually clear about what they'll want to see. The requirements aren't the unknown. Whether your estate can meet them is.

My advice, for what it's worth, is to stop asking whether the organization is ready for agents. Pick one use case that matters, build the meaning, the permissions and the evidence it needs to a standard you'd defend to the PRA, and see what you learn. You'll find out very quickly which parts of the estate were being held together by people.

Written for conversations with financial services leadership across the UK, Ireland and EMEA. The figures are the author's own illustrations. Survey data points are attributed above. Regulatory positions are current as of August 2026 and moving quickly, particularly the EDPB and Commission guidance on GDPR and the AI Act, due before the year is out.

Sources

What I've drawn on:

1. McKinsey & Company, State of AI trust in 2026: shifting to the agentic era, March 2026.
2. Goldman Sachs Investment Banking, Harnessing AI for the Real Economy, 2026.
3. European Commission, AI Act implementation timeline and the Digital Omnibus.
4. Nikhil Rathi, Rethinking regulation for the age of AI, FCA, June 2026.
5. Bank of England, response to the Treasury Select Committee report on AI in financial services, 2026.
6. ICO, ICO tech futures: agentic AI, January 2026.
7. Covington, French CNIL publishes note on agentic AI and data protection, July 2026.
8. Bird & Bird, Two eras, one rulebook: what agentic AI does to your GDPR compliance, July 2026.
9. Help Net Security, OWASP on prompt injection and agentic AI security failures, June 2026.
10. American Banker, Unpatched AI flaw poses risk to banking sector, April 2026.

For more than 30 years, Thoughtworks has helped shape how the world builds technology.

Today, by harnessing AI to scale what is possible, guided by human judgment and experience, we aim to be the force behind the world's most consequential technology transformations.

We combine our engineering and design expertise with AI to modernize systems, unlock data and create AI-powered products.

AI/works™ is our agentic development platform and Agent/works™ is our enterprise platform for governing and running autonomous AI agents. Together they bring decades of our engineering know-how to every stage of AI software delivery, from development through to deployment and governance.

thoughtworks.com

/thoughtworks

Design. Engineering. AI.

Last updated September 2026
© Thoughtworks, Inc. All Rights Reserved.