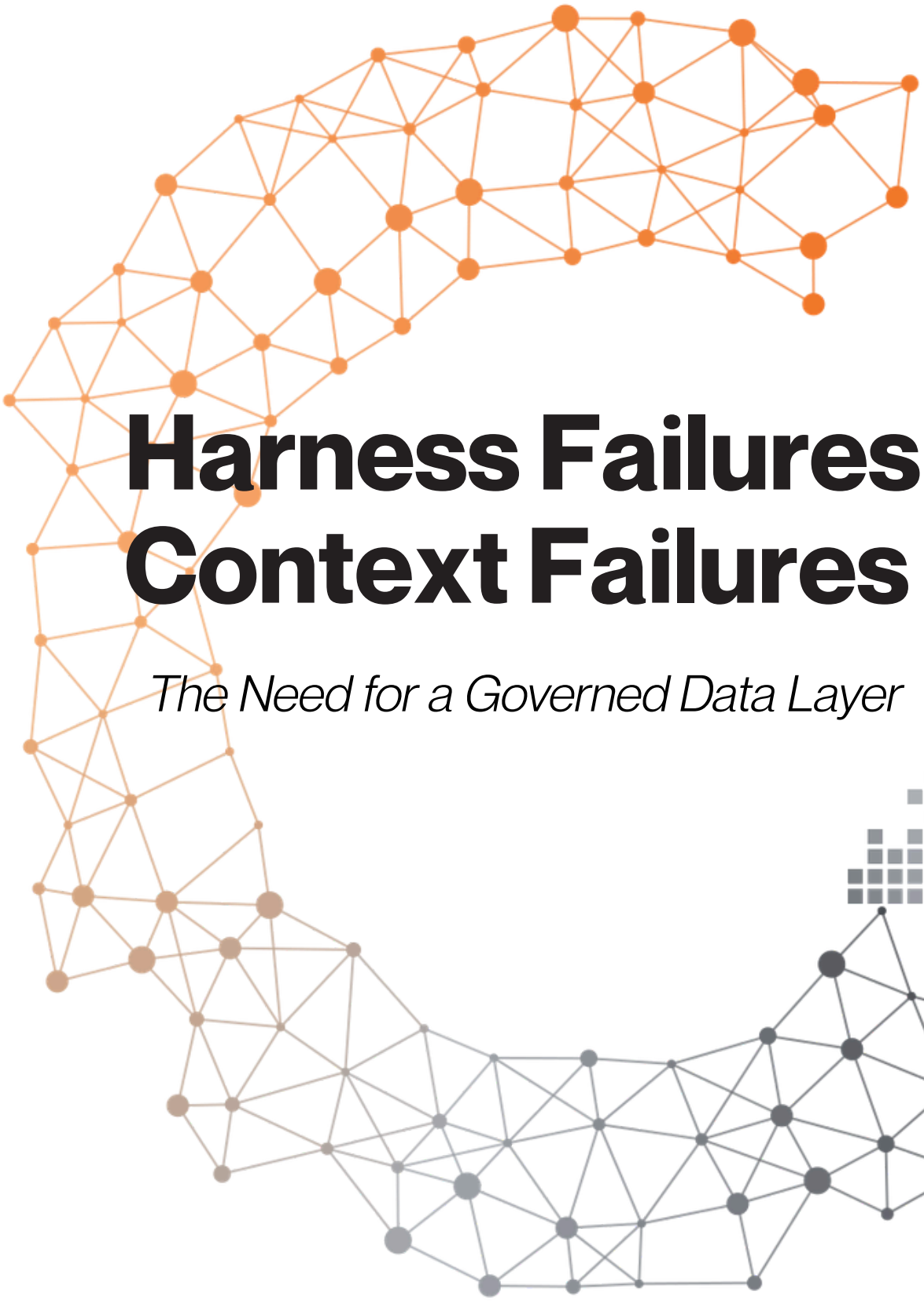
Modern Data 101 Community

November 2026



Harness Failures and Context Failures

The Need for a Governed Data Layer



© 2026 Modern Data 101 (by TMDC). All rights reserved.

Table Of Contents

1. Executive Summary
2. Two Ways an Agent Fails
3. What Harness Engineering Misses
4. Anatomy of a Context Failure
5. Why Harness Controls Outpace Data Readiness
6. The Governed Data Layer: Reference, Measurement, Limits
7. Why Governance Belongs Beneath the Harness
8. Building the Layer Without Rebuilding the Harness
9. Conclusion
10. References

© 2026 Modern Data 101 (by TMDC). All rights reserved.

1

Executive Summary

From The Modern Data 101 Community

This research brief addresses the critical distinction between harness failures (uncontrolled execution) and context failures (confidently acting on stale or ungoverned data). Based on benchmark data across 540+ enterprise practitioners, this paper outlines why traditional prompt and loop guardrails fall short, and introduces the strategic necessity of placing governance beneath the harness within a certified, governed data layer.

An enterprise agent can fail in two ways that look nothing alike. A harness failure is a failure of control: the loop never stops, a tool call breaks silently, a permission exists only as text in a prompt. A context failure is a failure of knowledge: the control structure works exactly as designed, and the agent acts, correctly and with authority, on information that was stale, conflicting, unapproved or never intended for use. The first kind is loud and gets fixed. The second kind is quiet, passes every runtime check, and is how agents end up confidently wrong.

Both failure families are already common. 80% of organisations report risky agent behaviour, including improper data exposure, and 26% of senior AI executives whose firms use agents cannot detect unauthorised agents operating internally.

Data is the binding constraint. 80% of banks now name data readiness and access as the top barrier to agent deployment, up from 63% a quarter earlier, while 73% of leaders express confidence in the governance they have.

- A harness decides how an agent acts. The data decides what it believes. Harness engineering cannot supply the second, because a harness has no independent way to know what is true.
- Context failures have two sources. Assembly failures, the context rot family, are fixable inside the harness. Supply failures are fixable only upstream, in the data layer.
- A governed data layer gives the agent's control loop three things it cannot generate for itself: a reference for what is correct, a measure of how trustworthy each input is, and enforced limits on what the agent may touch.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

2

Two Ways an Agent Fails

The vocabulary of agent engineering has grown quickly. Prompt, context, loop and harness engineering now describe four layers around the model, and four-lever diagnostics for locating a fault have become routine. The diagnostic habit is right. The grouping is too coarse for enterprise work, because it files two very different failures under one roof. A broken retry and a stale revenue figure both appear as "the agent got it wrong", yet they have different causes, different detectability and different owners.

	Harness failure	Context failure
What fails	Control of behaviour: the loop, the tools, the bounds	Knowledge: what the agent was given to reason over
Typical cause	A missing stop condition, a silent tool error, a rule that exists as text instead of as an enforced limit	Stale, conflicting, unapproved or restricted data reaching the model
What the logs show	Errors, retries, cost spikes, runaway loops	Nothing: calls validated, permissions held, action logged
Usually found by	The team operating the agent, quickly	A customer, an auditor or a regulator, late
Where the fix lives	The harness	Partly the harness, mostly upstream

Image: Harness failures break control and show up loudly. Context failures break knowledge and pass every runtime check | Source: Modern Data 101.

Detectability is what matters most. Harness failures announce themselves. Context failures are silent by construction, since the harness validated the call, checked the permission and logged the action. The silence has a measurable cost. In our 2026 survey, 73% discover after the fact that the data behind a decision was wrong or outdated, and 82% say their analytics tools are mostly not well connected to governance systems. Neither figure describes a model that malfunctioned. Both describe systems that worked and were wrong. In terms of another report, a context failure stretches the first clock, detection, because nothing fires.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

3

What Harness Engineering Misses

Harness engineering borrows from cybernetics, and the borrowing is worth taking further. A harness steers an agent in two ways. Guides act before the agent does: instructions, conventions, reference documents. Sensors act afterwards: tests, linters and review agents that observe what the agent did and feed corrections back. Every control loop needs a third element that this vocabulary leaves implicit, the reference signal a sensor compares an output against.

In software the reference is cheap and exact. A test suite passes or it fails, so a coding agent can be wrong and the harness will know. That property is why harness engineering took hold in coding first. Enterprise agents rarely have one. When an agent reports revenue, summarises an account or approves a credit, nothing in the runtime can say whether the figure is the governed metric, whether the account belongs to the right customer, or whether the approval used current policy. The harness can verify form: the call returned, the schema matched, the permission held. It cannot verify truth, because truth is a property of the data, and the runtime reaches the data only through the same retrieval it is trying to check.

Guides have the same limit

A system prompt can tell an agent to use only approved sources. It cannot define which sources are approved. A verification step can check an answer against retrieved context, but if that context came from the same ungoverned source as the answer, the check confirms consistency with the input and says nothing about correctness. A feedback loop without an independent reference still converges. It converges on whatever it was last given.

This is the central claim of the paper. The sensors of an enterprise agent are data-layer assets: certified definitions, resolved entities, freshness commitments and classification labels. An organisation that has invested heavily in harness observability and lightly in these assets has built a precise instrument with nothing to measure against. It will know, with great accuracy, that its agents behaved as designed.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

4

Anatomy of a Context Failure

Context failures enter the control loop at five points, and the point of entry decides what could have caught them.

Loop stage	How the failure enters	What it looks like	What would have caught it
Perceive	The agent retrieves from a source nobody approved, because it ranked best on similarity	A fluent answer and no errors	Source authority: which assets are certified for this task
Interpret	Two valid definitions, or two records for one entity, reach the model and it reconciles them by guessing	Three successful tool calls and one invented answer	Meaning: one versioned definition and one resolved identity
Decide	A constraint that lives in policy or in someone's head never reaches the context	A plausible action that breaks a rule	Policy attached to the data instead of the prompt
Act	An authorised call returns records the acting identity should not see, or should not use for this purpose	A clean permission log and exposed data	Entitlement: classification and purpose-based access at record level
Remember	A fact is summarised into memory, outlives the record it described, and shapes later steps	Slow drift with no single failing step	Currency: freshness rules and change signals that invalidate memory

Image: Context failures enter the agent loop at five stages, each with its own symptom and the data-layer control that would have caught it | Source: Modern Data 101.

Context rot belongs in a different category. The family of degradations that includes lost-in-the-middle, attention dilution, distractor amplification and repetition bias consists of assembly failures. They occur when a harness handles good context badly, and they appear even when every source is perfect. Their fixes live in the harness: position, eviction, deduplication and compaction. Supply failures are the mirror image. The harness handles context perfectly and the context is wrong.

	Governed supply	Ungoverned supply
Strong assembly	Reliable and accurate. The agent sees the right facts in a shape it can use.	Reliably wrong. Clean logs, tidy context windows and confident answers built on stale or unapproved inputs.
Weak assembly	Accurate inputs and degraded reasoning. Context rot erodes good data as sessions grow, which is a harness problem with a harness fix.	Unreliable and wrong. Noisy, inconsistent and hard to diagnose, because both layers are failing at once.

Image: Strong harness assembly cannot compensate for ungoverned supply: the agent stays reliable and wrong | Source: Modern Data 101.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

Governed supply reduces rot

Compaction, eviction and reranking are partly compensations for abundance without metadata. When every retrieved item carries certification, freshness and entity identity, the harness can rank by eligibility before relevance, deduplicate by entity instead of by text, and evict by staleness instead of by position in the window. The data layer does not replace context engineering. It gives context engineering better inputs.

A swap test to tell the families apart

Replace the context for a failing task with a hand-verified one and rerun it. If the failure disappears, it was a context failure. If it persists, swap in a minimal reference harness. If the failure disappears now, it was a harness failure. If it survives both swaps, look at the model. The test costs an afternoon and prevents weeks of prompt tuning against a data problem.

5

Why Harness Controls Outpace Data Readiness

Governance confidence is rising faster than data readiness, and that gap explains why context failures are the growth category. 73% of leaders now express confidence in the governance they have in place, up from 57% a quarter earlier. In the same survey, 80% of banks name data readiness and access as the top challenge to deploying agents, up from 63%. Across industries, the share citing data quality as a barrier rose from 37% to 65% within a year, and 73% of leaders rank data privacy and security as the leading AI risk.

The pattern has a simple explanation. Harness-side governance is tractable and visible, so it moves first. 70% of organisations now run AI monitoring dashboards, 74% include cost reviews in approval processes, and 43% set token budgets. All three controls sit at the harness. Data-layer governance crosses team boundaries and takes longer, so it lags. In our own survey, almost 70% of practitioners say their data is not clean or trustworthy enough for AI, and 53% have no easy way to trace its lineage.

Trust follows the same skew. In PwC's agent survey, 38% of executives trust agents with data analysis, nearly double the 20% who trust them with financial transactions. Trust is highest where a context failure is hardest to see. A wrong analysis looks like an analysis, while a wrong payment looks like a wrong payment.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

What the pattern predicts

If governance investment keeps concentrating at the harness, the next generation of agent incidents will share a signature: every runtime control passed, every log was clean, and the answer was wrong. Organisations will respond by tightening the harness further, which cannot help, because the harness was never the failing layer. The investment that changes the outcome is the one that is currently hardest to fund, because it belongs to no single team.

6

The Governed Data Layer: Reference, Measurement, Limits

A governed data layer is the set of capabilities that turn enterprise data into context an agent can be held to: certification, versioned definitions, entity resolution, classification, freshness and lineage, exposed through the interfaces agents already use. It does not replace the harness. It supplies the three signals a harness cannot generate for itself.

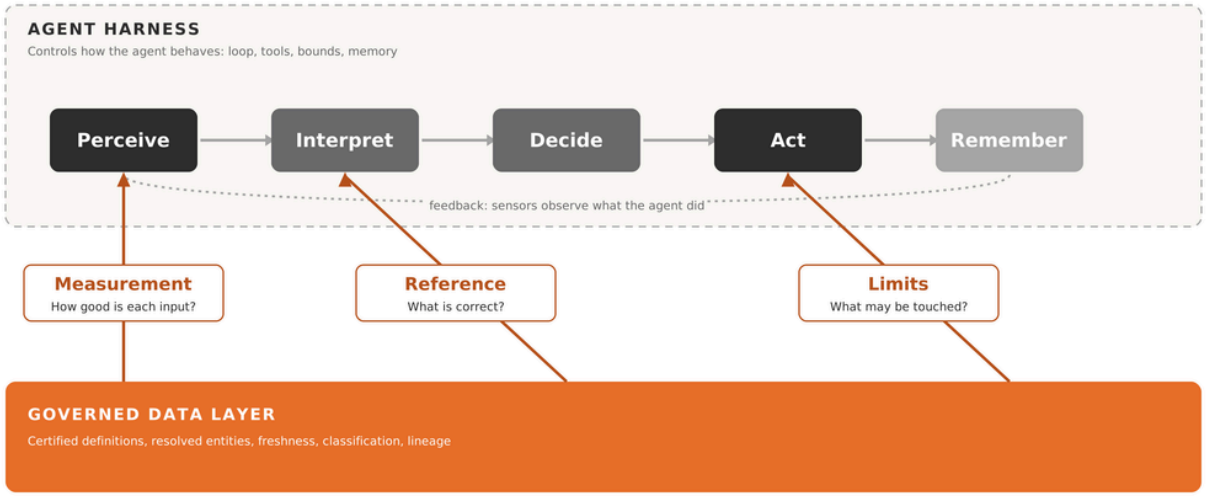


Image: A governed data layer gives the agent loop three signals: measurement, reference and limits | Source: Modern Data 101.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

The reference answers what is correct. Certified metrics, versioned business definitions and resolved entities give the harness something to compare an output against, the enterprise equivalent of a test suite. Without it, verification is circular.
The measurement answers how good each input is. Freshness, quality scores, source authority and lineage travel with the data, so the harness can rank by eligibility before relevance and can tell a current fact from a remembered one.

The limits answer what may be touched and what the data permits. Classification and purpose-based entitlement decide which records the acting identity may use, and a permitted action attached to a fact tells the agent what it may do next. A rule enforced at the data cannot be left out of a prompt.

Each field is cheap to attach once governed data products exist and expensive to reconstruct per agent. The practical test of a data layer is simple: can an agent ask for "revenue" and receive either the certified metric with its freshness and lineage, or an explicit statement that none exists? Anything less leaves the harness to guess.

Field the response carries	What the harness gets	Failure it prevents
Definition and version	Which meaning applies to this term	Conflicting definitions guessed into one answer
Certification status	Whether the source is approved for this task	Uncertified sources outranking governed ones
Entity identifier	One resolved identity across systems	Three valid records for one customer
Freshness and validity window	How current the fact is and when to refresh	Stale facts surviving in memory
Classification and entitlement result	Whether this identity may use this record for this purpose	Authorised calls that expose restricted data
Permitted action	What the fact allows the agent to do next	Plausible actions that break policy
Lineage pointer	Where the value came from, for audit	Execution traces that stop at the tool boundary

Image: Seven fields a governed data response should carry so the harness never has to guess | Source: Modern Data 101.

7

Why Governance Belongs Beneath the Harness

Safe AI is easiest to reason about as a race between how fast harm can spread and how fast an organisation can respond. We use the term trust latency for the organisation's side of that race: the time between an AI system beginning to behave badly and the organisation stopping it. It splits into three clocks. Detect measures how long before anyone knows. Explain measures how long before the cause is understood well enough to act on. Contain measures how long before the behaviour is stopped or reversed.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

Governance can live in the harness or in the data. Three pressures push it into the data.

Portability

Policies written into a harness leave with it. Change the model, adopt a framework or buy a vendor agent, and the policy set has to be rebuilt and re-audited. Policies attached to data assets stay put, because the asset is what every agent eventually touches.

Scale

Enterprises will run fleets. 62% of organisations are building, deploying or developing agents, and 25% are developing multi-agent systems, up from 6%. Nearly three in four plan to deploy agentic AI within two years, yet only 21% report mature governance for it. Per-harness governance multiplies the work with every new agent and governs the same record differently depending on who asks.

Accountability is the third pressure. An audit asks what the agent acted on. A harness log answers what it did. Only lineage joined to the execution trace answers the first question, and most organisations have not built that join: 49% of organisations using agents have not updated their governance framework for agentic risk, and 64% of surveyed large organisations have already lost more than $1 million to AI-related risks. Guardrails left to individual teams will be applied unevenly, which is the condition under which losses accumulate.

The seam nobody owns

Harness failures have an owner: the team that runs the agent. Source-data failures have an owner: the team that runs the source. The failure that matters here occurs at the join, where data becomes context, and the join belongs to nobody. The agent team reports that every call succeeded. The data team reports that the source was correct. Both are right, and the customer record is still wrong. Naming one accountable owner for what enters an agent's context window, with authority over the certification and entitlement rules behind it, closes the seam. In most enterprises that role does not yet exist.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

8

Building the Layer Without Rebuilding the Harness

The data layer attaches to an existing harness at one point: retrieval. Whatever the agent uses to fetch data, whether a vector search, a SQL tool or a protocol server, should resolve through a governed path that applies eligibility, entitlement and freshness before ranking, and returns the fields in Section 6 with every result.

Three design choices keep the integration cheap. First, expose governed data products through the interface agents already speak, so that any harness receives identical answers, which is what makes model and framework changes survivable. Second, let traces carry lineage identifiers, so an execution log can be joined to the data it acted on without a forensic project. Third, point the harness's verification step at the reference, meaning certified definitions and entity records, instead of at the context it just retrieved.

None of this requires replacing the runtime. It requires treating the data layer as part of the agent's control system, with an owner and a service level, instead of a source the agent happens to read. The change in posture matters more than the tooling. Teams that adopt it stop asking whether the agent behaved and start asking whether the agent was entitled to know what it knew.

Start narrow
Begin where a wrong answer costs most. Take the highest-stakes decision an agent already supports, run the swap test on its last failure, and attach the context contract to the one data product behind it. A single governed path that removes one recurring silent failure is worth more than a catalogue of ungoverned assets, because it shows the organisation what a context failure looks like when it stops happening.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

9

Conclusion

Harness engineering made agents reliable. It cannot make them right. The difference between the two is a data problem, and it will define the next phase of enterprise agents. Organisations that give their agents a governed reference, a measure of input quality and enforced limits will find that harness failures become routine engineering and context failures become rare. Organisations that do not will keep discovering, after the fact, that every log was clean. We expect the measure of a mature agent programme to shift within two years from how reliably its agents behave to what its agents were entitled to know when they acted.

© 2026 Modern Data 101 (by TMDC). All rights reserved.

10

References

[1] Are your AI guardrails keeping pace with your agents? McKinsey & Company, Sep 2026. https://www.mckinsey.com/featured-insights/themes/are-your-ai-guardrails-keeping-pace-with-your-agents

[2] Deploying agentic AI with safety and security: A playbook for technology leaders. McKinsey & Company, 2025 (reproduced by Informa Connect). https://informaconnect.com/deploying-agentic-ai-with-safety-and-security-a-playbook-for-technology-leaders/

[3] EY survey finds that autonomous AI implementation outpaces oversight. EY US AI Risk and Governance Survey, 202 senior AI executives, Sep 2026. https://www.ey.com/en_us/newsroom/2026/09/ey-survey-finds-that-autonomous-ai-implementation-outpaces-oversight-yielding-an-ai-governance-gap

[4] AI governance has entered its next phase: closing the confidence gap. EY, Sep 2026. https://www.ey.com/en_us/insights/assurance/ai-governance-has-entered-its-next-phase-closing-the-confidence-gap

[5] How can responsible AI bridge the gap between investment and impact? EY Responsible AI Pulse, 975 C-suite leaders, 2025. https://www.ey.com/en_gl/insights/ai/how-can-responsible-ai-bridge-the-gap-between-investment-and-impact

[6] AI's value story sharpens: KPMG Quarterly AI Pulse, Q3 2026. KPMG, 314 US leaders, Sep 2026. https://kpmg.com/us/en/media/news/q3-ai-pulse-2026.html

[7] AI at scale: KPMG Quarterly AI Pulse, Q4 2025. KPMG, Jan 2026. https://kpmg.com/us/en/media/news/q4-ai-pulse.html

[8] AI agents are scaling faster than their guardrails. Deloitte Insights, 3,235 leaders, 2026. https://www.deloitte.com/us/en/insights/topics/emerging-technologies/ai-agents-scaling-faster.html

[9] The State of AI in the Enterprise. Deloitte AI Institute, Jan 2026. https://www.deloitte.com/cz-sk/en/services/consulting/research/the-state-of-ai-in-the-enterprise.html

[10] PwC's AI Agent Survey. PwC US, 300 senior executives, May 2025. https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-agent-survey.html

[11] The Modern Data Report 2026: The Data Activation Gap (web). Modern Data 101, 540+ practitioners, 2026. https://www.moderndata101.com/modern-data-report-2026

© 2026 Modern Data 101 (by TMDC). All rights reserved.

[12] The Modern Data Report 2026 (full report PDF, p. 35). Modern Data 101, 2026. https://cdn.prod.website-files.com/6538b3836b3dce952f05ff81/69835768b916bc0c44fcd5ec_The%20Modern%20Data%20Report%202026.pdf

[13] The Agent Harness Data Layer: How to Govern What AI Agents Actually Know. Informatica (a Salesforce company). https://www.informatica.com/resources/articles/agent-harness-data-layer.html

The Agent Harness Data Layer: How to Govern What AI Agents Actually Know. Informatica (a Salesforce company). https://www.informatica.com/resources/articles/agent-harness-data-layer.html

[14] Agent Harness Explained: What Turns Context Into Trusted Answers. WisdomAI, Oct 2026. https://www.wisdom.ai/resources/guides/agent-harness/what-is-a-agent-harness

[15] Prompt, Context, Loop, Harness: Four Layers, Four Different Failures. What was that again?, Oct 2026. https://whatwasthatagain.com/prompt-context-loop-harness-four-layers-four-different-failures/

[16] Harness Engineering: The 4 Levers Behind Almost Every Agent Failure. Damian Galarza, Jun 2026. https://www.damiangalarza.com/posts/2026-06-08-harness-engineering-four-levers/

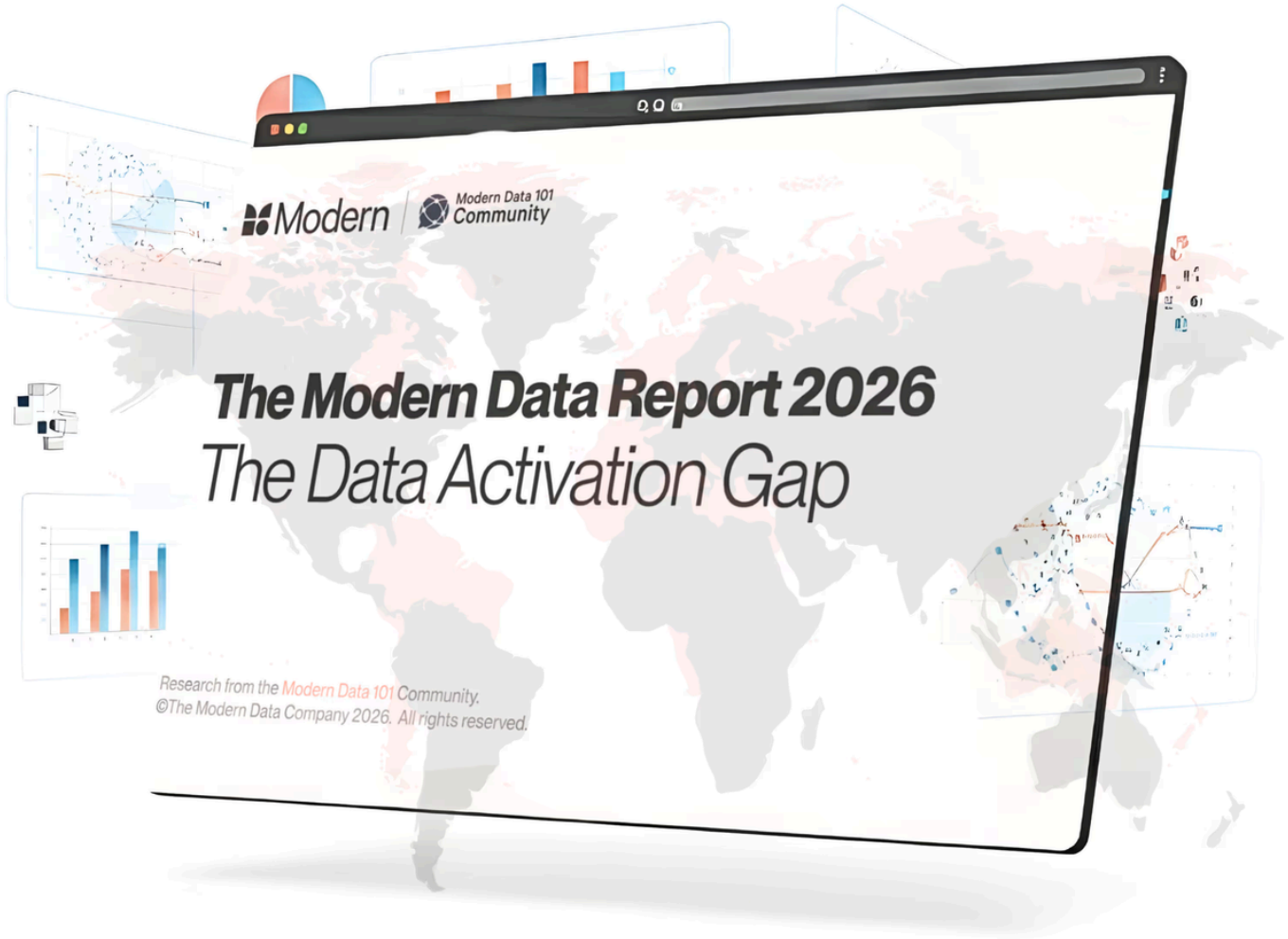
[17] Types of Context Rot in Harness Engineering. Fikayo Adepoju, DEV Community, Sep 2026. https://dev.to/coderonfleek/types-of-context-rot-in-harness-engineering-4fpc

[18] Harness engineering for coding agent users. Birgitta Böckeler, martinfowler.com, Apr 2026. https://martinfowler.com/articles/harness-engineering.html

[19] The Anatomy of an Agent Harness. LangChain. https://www.langchain.com/blog/the-anatomy-of-an-agent-harness

[20] Vol. 05 The State of AI Observability. Modern Data 101, 2026. https://www.moderndata101.com

© 2026 Modern Data 101 (by TMDC). All rights reserved.



540+ Respondents, 64 Countries, 29 Industries

Access the report now

© 2026 Modern Data 101 (by TMDC). All rights reserved.

Modern Data 101 Community



Subscribe

moderndata101.com