

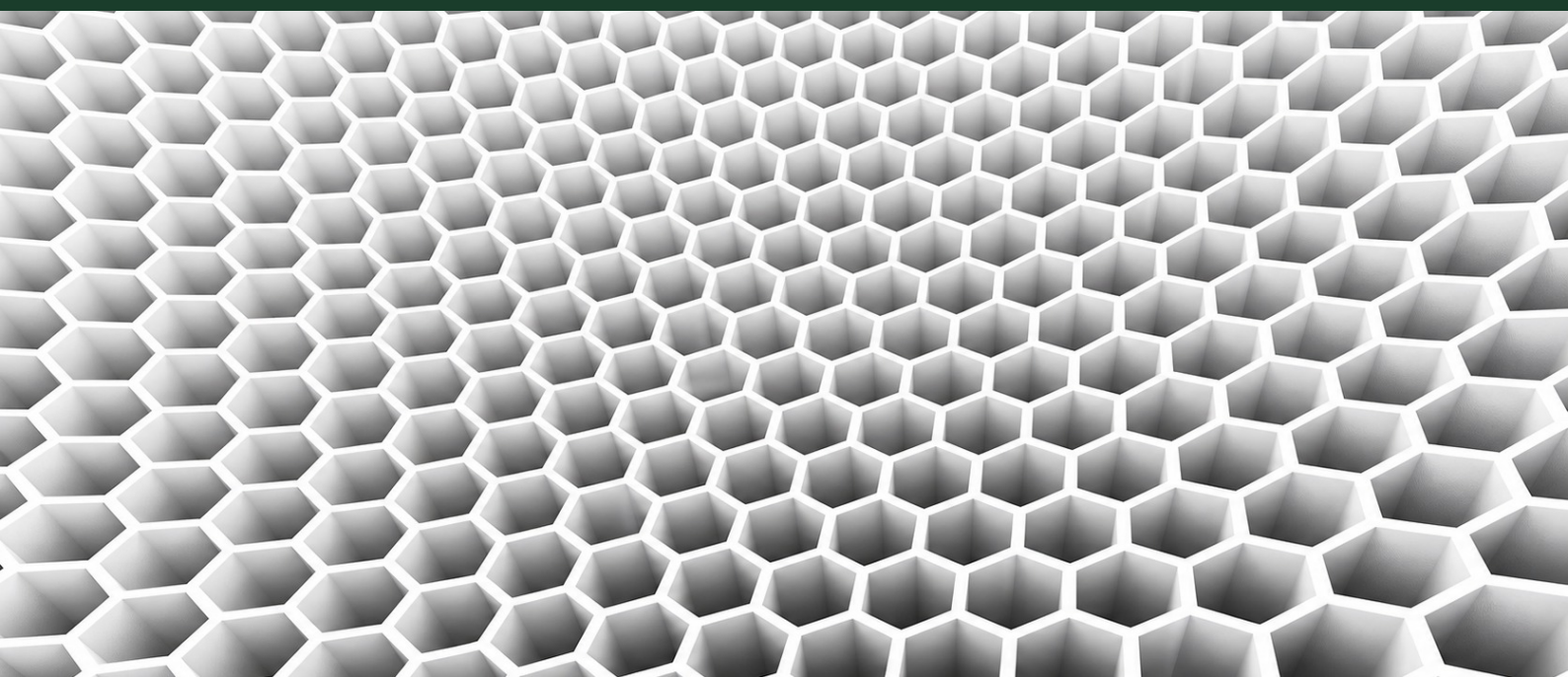
BEYOND RECONCILIATION

One Version of the Truth Across Platforms That Were Never Meant to Share It

A Practitioner's White Paper on Data Governance in Multi-Platform Banking
Environments

Giovanni Leonardi

July 2026



Contents

1. Executive Summary	3
2. The Problem Is Not the Technology, and Treating It as Though It Were Is the First Mistake	3
3. Why Multi-Platform Environments Resist a Single Truth	4
4. Master Data Management as the Spine	4
5. The Entity Conflict Problem: A Framework for Resolving Conflicting Definitions	5
6. Data Lineage: Trust Requires Traceability	6
7. Data Quality as a Managed Service, Not a Project	7
8. The Governance Operating Model	7
8.1 The Governance RACI	8
8.2 A Recommended Cadence of Data Quality Reviews	9
9. Recommendation	9

One Version of the Truth Across Platforms That Were Never Meant to Share It

Programme · No. 258

July 2026

"A single version of the truth is not a database you build; it is an agreement you govern."

1. Executive Summary

Most banks that set out to build "a single version of the truth" begin by buying technology and end by discovering they bought the wrong thing. The obstacle to a governed, trustworthy view of the customer is rarely the absence of a platform capable of holding it. It is that customer data lives across four or five systems — each with its own schema, its own update cadence, and its own working definition of what a customer is — and no amount of integration technology resolves a disagreement about meaning. Integration moves data; it does not reconcile the human decisions encoded in that data.

This paper argues that data governance in a multi-platform environment is an operating-model problem before it is a technology problem. The systems were never designed to talk to each other because they were never designed together; they were bought at different times, for different purposes, by different parts of the business. Bolting a data platform across them without an operating model that assigns ownership, resolves conflicting definitions, and enforces quality simply relocates the fragmentation into a new and more expensive place.

The case developed here rests on four load-bearing capabilities — master data management as the spine, data lineage as the basis of trust, data quality as a continuously managed service rather than a project, and a governance operating model that gives all three a home. It offers a practical, repeatable framework for resolving the conflicting definitions of core entities — customer, account and product — that are the root cause of most reconciliation pain. And it lands on two concrete instruments an institution can adopt immediately: a governance RACI that names who does what, and a review cadence that keeps quality from silently decaying. The recommendation is unambiguous: stand up the operating model first, and let the technology serve it — not the reverse.

2. The Problem Is Not the Technology, and Treating It as Though It Were Is the First Mistake

The recurring pattern across banks running multiple core and satellite platforms is a persistent inability to answer apparently simple questions. How many customers do we

have? What is our total exposure to a single relationship that spans lending, deposits and investments? Which accounts belong to entities we are obliged to treat as connected for regulatory purposes? These are not hard questions because the data is missing. They are hard because the same real-world thing is represented differently in each system, and the differences are not errors to be cleaned but decisions that were made — reasonably — by each system's designers to serve each system's purpose.

A lending platform models a customer as a borrower with a credit assessment. A deposits system models the same person as an account holder with a balance. An investment platform models them as a beneficial owner within a possibly complex ownership structure. Each representation is correct for its own domain. The friction begins the moment the institution needs a view that spans them, because there is no neutral, authoritative definition of "customer" sitting above the platforms to arbitrate between the three. The data platform the bank builds to unify them inherits this ambiguity; it does not dissolve it. Technology that faithfully integrates three conflicting definitions produces one well-integrated contradiction.

The most expensive mistake in multi-platform data work is to treat a disagreement about meaning as a problem of plumbing. You can move the data perfectly and still be unable to trust the answer, because the systems never agreed on what the data meant in the first place.

3. Why Multi-Platform Environments Resist a Single Truth

It is worth being precise about why fragmentation is so stubborn, because the reasons dictate the remedy. Three forces are usually at work at once.

- Divergent purpose. Each platform was optimised for its own transactional job. The schema, the validation rules and the update frequency all reflect that job, and all of them are, locally, correct. There is no villain here — only the accumulated logic of systems built to do different things well.
- Temporal drift. Systems update on different clocks. A change of address captured in the channel system today, in the core overnight, and in the investment platform on the next reconciliation cycle means that at any given moment the three disagree, and all three are "right" as of their own last update.
- Definitional divergence. The deepest force. Each system encodes its own answer to questions the business never centrally resolved — what counts as an active customer, when an account is genuinely closed, whether a product is the thing sold or the thing booked. These are semantic decisions masquerading as data fields.

Of the three, definitional divergence is the one that technology cannot touch and the one this paper spends most of its weight on, because a governed single truth is impossible until the institution decides, once and deliberately, what its core entities mean.

4. Master Data Management as the Spine

Master data management is the discipline that gives a fragmented estate a spine. Done properly, it establishes a single authoritative record for each core entity — a golden record for the customer, the account, the product — to which every platform's local representation is mapped, and against which every downstream question is answered. The golden record is not a replacement for the source systems; it is the reconciled, governed layer that sits above them and mediates between them.

The critical design decision in MDM is the style of mastering, and it must be made consciously rather than defaulted into. A registry style keeps mastering light: the golden record is a cross-reference that links the source identifiers and resolves conflicts on read, leaving the source systems untouched. A consolidation style physically assembles golden records in a central hub for downstream consumption but does not write back. A co-existence or centralised style pushes governed values back into the source systems so that the estate converges over time. There is no universally correct answer. A registry approach is fastest to stand up and least invasive, and is often the right first move for a smaller institution; a co-existence approach delivers deeper convergence but demands a maturity of governance and a tolerance for writing back into operational systems that many banks do not initially have.

Whatever the style, three things are non-negotiable. There must be a single, agreed identifier strategy that lets the institution assert that two records in two systems are, or are not, the same entity — the matching and survivorship logic that decides which source wins for which attribute. There must be an explicit statement of which system is authoritative for which attribute, because authority is almost never whole-entity: the core may be authoritative for legal name while the channel system is authoritative for contact preferences. And there must be a governed process for changing any of this, because an identifier strategy or a survivorship rule that can be altered informally is not a control.

5. The Entity Conflict Problem: A Framework for Resolving Conflicting Definitions

Everything above depends on the institution being able to answer, authoritatively, what a customer, an account and a product are. This is the hardest and most neglected work in the whole endeavour, and it is where most programmes quietly fail — they build the machinery of MDM on top of definitions that were never actually agreed, and the machinery faithfully perpetuates the disagreement. The following framework is a repeatable method for resolving a contested entity definition. It is deliberately domain-driven and business-led; it is not a data-modelling exercise that can be delegated to a technical team.

1. Convene the parties who own the meaning, not the systems. For each contested entity, identify the business functions whose work depends on its definition — for the customer, that is typically relationship management, risk, compliance, and finance. The definition is theirs to settle; the data team facilitates and records, it does not decide.
2. Surface the implicit definitions already in force. Extract, from each platform, the working definition its behaviour actually encodes — when it counts a customer as

active, when it treats an account as closed, how it identifies a product. This is archaeology, not design: the goal is to make explicit the decisions the systems already embody so they can be compared.

3. Locate the specific points of conflict. Most of the definitions will agree. Isolate the handful of attributes and lifecycle rules where they genuinely diverge — the dormancy threshold, the treatment of a joint versus sole relationship, whether a lapsed product remains part of the customer's holdings. Naming the conflict precisely is most of the work.
4. Adjudicate to a canonical definition, purpose by purpose. Here the framework departs from the common error of seeking one definition to rule them all. The right move is often to accept that a single entity legitimately has more than one governed definition for different purposes — a regulatory definition of connected customers, a commercial definition of an active relationship — and to make each of them explicit, named, and owned, rather than pretending a single number can serve every question. What must never be tolerated is an ungoverned multiplicity, where the number differs because nobody decided it should.
5. Encode the canonical definition as a rule the platform enforces. A definition that lives in a policy document and not in the survivorship logic, the quality rules and the golden-record construction is decoration. The output of adjudication must be machine-enforceable rules, versioned and owned.
6. Assign an owner and a change process. Every canonical definition gets a named business owner accountable for it and a governed route to change it. Definitions are not settled once; they drift as products and regulation evolve, and an unowned definition will silently rot.

“Stop asking which system holds the right definition of the customer. Ask which definitions the business has actually agreed to be accountable for — and govern those.”

The discipline this framework enforces is that meaning is decided before it is engineered. An institution that runs this process for its handful of core entities has done the single most valuable thing available to it, because every downstream capability — MDM, quality, lineage, reporting — becomes tractable once the entities beneath them are governed.

6. Data Lineage: Trust Requires Traceability

A governed single truth that cannot show its work will not be trusted, and a number that is not trusted might as well not exist. Data lineage is the capability that lets the institution answer, for any figure it publishes, where it came from, what was done to it, and on what authority. In a multi-platform environment this is not a nicety; it is the difference between a data platform people rely on and one they quietly work around by going back to the source systems and rebuilding the number in a spreadsheet.

Lineage earns its cost in three specific moments. When a figure is challenged — by a regulator, an auditor, or a senior executive — lineage turns a multi-week investigation into a traceable answer. When a source system changes — a schema alteration, a new product, a migration — lineage reveals what downstream depends on it, so that change is managed rather than discovered through breakage. And when data quality fails, lineage locates the point of failure rather than leaving teams to argue about whose number is wrong. The practical requirement is that lineage be captured at the level of meaningful transformation, not merely at the level of table-to-table movement: it must record not just that data flowed from A to B, but what business rule was applied in between, because that rule is usually where the trust is won or lost.

7. Data Quality as a Managed Service, Not a Project

Data quality is where good intentions go to die, because it is almost always funded as a project — a one-off cleanse tied to a programme milestone — when it is by nature a continuous service. Data decays. Every day the business opens accounts, changes addresses, launches products and closes relationships, and every one of those events is an opportunity for the golden record to drift from reality. An institution that cleanses its data once and declares victory has merely reset a clock that immediately begins running down again.

The reframe that works is to treat quality as a managed service with defined dimensions, owned rules, and continuous measurement. The standard dimensions are enough to organise the work.

- Completeness — are the attributes the business relies on actually populated?
- Accuracy — do the values correspond to the real-world truth they claim to represent?
- Consistency — do the platforms agree where they are supposed to agree?
- Timeliness — is the data current enough for the decisions that depend on it?
- Uniqueness — is each real-world entity represented once, not several times?
- Validity — do the values conform to the rules and formats the business has defined?

Each dimension needs measurable rules tied to a specific business consequence, a threshold that distinguishes acceptable from unacceptable, and a named owner who is accountable when the threshold is breached. Measurement must be continuous and visible, because a quality metric that is computed once a year is a post-mortem, not a control. The institutions that get this right run quality the way they run any other operational service: with monitoring, with alerting, with an owner, and with a standing forum that reviews the numbers and acts on them — which is the subject of the operating model to which this paper now turns.

8. The Governance Operating Model

Capabilities without an operating model are orphans. MDM, lineage and quality each require someone accountable, someone doing the work, and a forum where disputes are

settled and standards are set. The operating model is the connective tissue that turns a set of tools into a governed institution. Three roles carry most of the weight.

- The Business Data Owner is a senior figure in the function that depends most on an entity or domain — accountable for its definitions, its quality thresholds, and the business consequences of getting it wrong. Ownership is a business accountability, never a technical one; the most common failure in banking data governance is to place ownership in IT, where it has responsibility without authority.
- The Data Steward is the operational custodian who does the day-to-day work — monitoring quality, applying and maintaining rules, investigating breaches, and stewarding the golden record. Stewards sit close to the business but work hand-in-glove with the platform teams.
- The Platform / System Owner is accountable for the source or target system itself — its integrity, its changes, and its faithful implementation of the rules the owners and stewards define.

Above these sits a Data Governance Council — the standing cross-functional forum that adjudicates definitional conflicts that cross domains, sets institution-wide standards, and arbitrates the disputes that individual owners cannot resolve between themselves. It is the body that runs the entity-definition framework described earlier and holds the ring when two functions want incompatible things.

8.1 The Governance RACI

The following RACI makes the model concrete across the activities that actually recur in a multi-platform environment. R marks who does the work, A the single point of accountability, C those consulted, I those informed.

Activity	Business Data Owner	Data Steward	Platform Owner	Governance Council
Define / approve canonical entity definition	A	R	C	C
Maintain matching & survivorship rules	A	R	C	I
Set data quality thresholds	A	R	C	I
Monitor & report quality against thresholds	I	R	C	I
Investigate & remediate a quality breach	A	R	R	I
Approve a change to a source system schema	C	C	A	I
Maintain lineage for a data flow	I	R	R	I
Adjudicate a cross-domain definition conflict	C	C	C	A
Approve exceptions to governance standards	C	I	I	A

The single most important discipline the RACI enforces is that every activity has exactly one accountable party. A governance model in which accountability is shared is a governance model in which, when something breaks, it belongs to no one. Note that accountability for entity definitions and quality thresholds sits with the business owner, while accountability for system change sits with the platform owner, and accountability for cross-domain conflict sits with the council — authority is placed where the knowledge and the consequence actually live.

8.2 A Recommended Cadence of Data Quality Reviews

Governance that does not meet on a rhythm decays into governance that meets only in a crisis. The following cadence is a defensible default for a mid-sized institution and should be tuned to risk, not adopted mechanically.

- Continuous / automated — quality rules run against the golden record on the data's natural update cycle, with automated alerting when a threshold is breached. This is monitoring, not a meeting, and it is the foundation everything else rests on.
- Weekly (Data Stewards) — stewards review the exceptions and breaches surfaced by monitoring, triage them, and remediate or escalate. This is the operational heartbeat of the model.
- Monthly (Business Data Owners) — owners review quality trends for their domain against thresholds, sign off remediation, and confirm that definitions still fit the business as products and regulation move.
- Quarterly (Data Governance Council) — the council reviews institution-wide quality posture, adjudicates cross-domain issues that have accumulated, approves standard and definition changes, and reports the state of the data estate to executive and, where relevant, board risk committees.
- Annually — a full review of the operating model itself: are the definitions still right, are the thresholds still calibrated, is ownership still correctly placed, and has the estate changed enough to warrant rethinking the MDM approach?

The principle behind the cadence is that issues should be caught at the lowest level and the shortest interval that can resolve them, and escalated only when they genuinely cross a boundary. A model that pushes every quality question to a quarterly council will drown it; a model that never escalates will let systemic issues fester in the weekly steward review. The layered cadence is what keeps each forum solving the problems it is actually equipped to solve.

9. Recommendation

The institutions that achieve a governed single version of the truth across fragmented platforms are not the ones that bought the best data platform. They are the ones that understood the problem was never fundamentally technical, and sequenced their work accordingly. The recommendation of this paper is therefore specific and ordered.

Stand up the operating model first — the owners, the stewards, the council — because without accountable humans, every downstream capability is orphaned. Run the entity-definition framework for your core entities before building the machinery that

depends on those definitions, accepting where necessary that an entity has more than one governed definition for different purposes, but never tolerating an ungoverned one. Choose an MDM style that matches your governance maturity rather than your ambition, starting lighter than you think you should. Treat quality as a continuously managed service with owned rules and a review cadence, not as a one-off cleanse. Capture lineage at the level of business transformation, so that trust can be evidenced rather than asserted. And only then let the technology serve the operating model you have built.

A single version of the truth is not a database you build; it is an agreement you govern. The platforms will always have been designed apart. What unifies them is not integration technology but the deliberate, owned, continuously maintained decision about what the institution's data means — and that decision is the one piece of the problem no vendor can sell you.