The CIO's Guide to

AI-Enabled Core Banking Modernisation

From Systems of Record to Systems of Intelligence:

The New Architecture of the AI-First Bank

Introduction

The Shift That Is Deeper Than Batch to Real-Time

The conventional narrative about AI-enabled core banking modernisation frames the challenge as a migration from batch processing to real-time event-driven architecture. This framing is accurate but insufficient. It describes the plumbing. It does not describe the shift in what the plumbing enables.

The more fundamental shift AI introduces is this:

for the first time, a banking system can generate data access, validation logic, and decision intelligence that is specific to an individual customer, a specific transaction, or a specific moment - on demand, without pre-engineering a new service, a new report, or a new data mart.

In the digital era, the lowest level of granularity a system could deliver was the service level. A repayment service returned repayment data for a defined period. A customer profile service returned a defined set of fields for a customer. If you needed something different, you built a new service. The service was the atomic unit of the architecture.

AI changes the atomic unit. The query - or in AI terms, the context - becomes the atomic unit. A context can be specific to one person, one transaction, one moment in time. This is not an incremental improvement on what APIs and microservices enabled. It is a different level of granularity entirely.

At the same time, AI expands the data surface that can be reasoned over. In the pre-AI architecture, decision models operated on structured data: columns, values, dates. The text within a transaction record, the narrative in a compliance note, the explanation in a manual override - all invisible to any analytical or decision model. AI makes the content of a column queryable, not just the column itself. For the first time, the unstructured data that banks have always held but never been able to reason over becomes part of the decision surface.

And AI shifts where governance lives. In the service-oriented architecture, governance was maintained at the API and service level. In the AI-first core, the decision logic has moved into the intelligence layer - the prompts, the context specifications, the query-level validation rules, the agentic workflows. The governance requirement has moved with it.

These three shifts - from service-level to context-specific intelligence, from structured to structured-and-unstructured data access, and from service-layer to intelligence-layer governance - are what AI-enabled core banking modernisation actually means. They define three battlegrounds for the banking CIO.

> The most consequential modernisation question is not 'How do we move from batch to real-time?'
>
> It is: **'How do we move from predefined constructs to context-specific intelligence - and govern the intelligence layer that makes it possible?'**

Opening Framework

From Digital-First Architecture to AI-First Intelligence Architecture

	Digital-First Architecture	AI-First Intelligence Architecture
Query granularity	Service-level - predefined constructs for a period, a product, or a segment	Context-specific - dynamically generated for a specific customer, transaction, or moment on the fly
Data scope	Structured data only - queryable by date, value, and column name	Structured and unstructured - queryable by content, meaning, pattern, and context within the column
Validation model	Predefined, aggregated validations embedded in the data mart - applied uniformly across segments, periods, and products - immutable without re-engineering	Context-specific, pattern-enhanced validations generated on the fly - applied at the level of the individual customer or transaction, informed by behavioural and typological patterns specific to that context, without altering the underlying data mart
Integration granularity	Service-level APIs - reusable, standardised, not customisable per context	Customer and transaction-level intelligence - generated on demand for the specific context required
Governance layer	API version control at the service level - lineage established at the code or document level	Intelligence layer governance - prompts, queries, and validation rules as governed artefacts; lineage granularized from document level to paragraph and sentence level, making auditability faster and more precise

BATTLEGROUND 1

From Predefined Constructs to Context-Specific Intelligence

From Service-Level APIs to Customer-Level and Transaction-Level Inference

1.1 The Granularity Shift – What AI Enables That APIs Could Not

The digital era gave banks service-oriented architecture and then microservices. Both were genuine advances. They decomposed monolithic systems into reusable, consumable units that different applications could call independently. But both had a structural ceiling: the service was built for a defined purpose, carried a defined data payload, and served any consumer of that service identically. If a specific customer context required something different from what the service provided, the answer was to build a new service.

AI dissolves this ceiling. Through agentic context assembly, AI can dynamically construct a data request that is specific to one customer's context at one moment in time - pulling from multiple data sources, combining structured transaction data with unstructured document content, and returning exactly what the decision at hand requires. It does not call a predefined service. It generates a purpose-built query on the fly.

The practical significance for banking operations is substantial. Before AI, even the most sophisticated digital banking systems could only ask pre-answered questions. The service existed; you called it. The report existed; you ran it. AI enables the system to ask questions that were never pre-answered - because it can formulate the question at the moment the decision requires it, against any combination of available data.

For a lending decision, this means the AI does not retrieve a generic customer profile. It assembles a context that includes the customer's repayment history, their current account behaviour, the content of their last interaction with the bank, and any unstructured signals from document uploads - synthesised specifically for this decision, not for a general-purpose view. For a fraud detection inference, it means the model is not pattern-matching against a predefined fraud typology. It is reasoning over the full context of this transaction, this customer, this moment - including signals that no predefined service was built to surface.

The key distinction

Digital-era APIs delivered service-level intelligence - the same data payload to every consumer. AI-era context assembly delivers customer-level and transaction-level intelligence - a different data assembly for every unique decision context. This is not a faster API. It is a different layer of the architecture entirely.

Named mechanism: Agentic context assembly

The AI-driven capability to dynamically construct decision-relevant data contexts without pre-engineering the data retrieval architecture. The AI formulates the query, identifies the relevant data sources, combines structured and unstructured inputs, and returns a context specific to the decision at hand - on demand, at the moment of need.

1.2 Unstructured Data as a Decision Surface

In the pre-AI architecture, queries operated on structured data: columns, values, dates. The text content within a transaction record - the narrative in a payment description, the explanation in a compliance note, the stated reason in a manual override, the content of a customer document - was invisible to any analytical or decision model. It existed in the database. It could not be reasoned over.

The limitation was architectural. Databases could tell a model that column X had value Y. They could not tell a model what the text inside column Z meant. Statistical and rule-based models worked with what could be represented as a number or a category. Text, meaning, context - these were beyond the boundary of what the decision layer could access.

AI makes the content of a column queryable, not just the column itself. This is not a minor extension of existing capability. It fundamentally expands the decision surface available to every AI model in the banking estate.

For credit risk assessment, it means the model can process the text of a customer's complaint history alongside their numerical credit metrics - identifying patterns in how customers describe their financial circumstances that structured data fields cannot capture. For AML transaction monitoring, it means the model can reason over the narrative content of transaction descriptions alongside the transaction values and counterparty data - the most common location for the contextual signals that distinguish legitimate from suspicious transactions. For regulatory reporting, it means the validation system can assess the meaning of an explanation field - not just whether a value is present, but whether the explanation is coherent, consistent with the transaction data, and sufficient for the regulatory purpose it serves.

The implications for data governance are significant. A data estate that has been governed only for its structured fields has ungoverned content in every text column - content that AI models can now access and reason over. Building the unstructured data surface into the data governance framework is not an optional extension of AI-enabled modernisation. It is a prerequisite for the unstructured data layer to be used reliably.

Named mechanism:
Unstructured data integration

The architectural extension of the core's AI-accessible data surface to include text, document content, and semantic meaning alongside structured fields. Enables AI decision models to reason over the full information content of the banking data estate - not just the numerical and categorical fields that pre-AI models could process.

Implementation reference

In Maveric's core modernisation engagements, integrating unstructured data into the decision surface for AML transaction monitoring - enabling the model to assess transaction narrative context alongside value and counterparty signals - reduced false-positive escalation rates materially by allowing the model to distinguish transactions that were structurally similar but contextually distinct. The improvement was not in the model's AML typology matching. It was in the model's ability to read the transaction.

BATTLEGROUND 2

From Predefined Reports to Query-Level Validation Intelligence

From Data Mart Governance to Intelligence-Layer Validation

2.1 The Validation Problem in Legacy Core Architecture

In the pre-AI core, validations were embedded in the data mart or the report definition. A regulatory report had a defined set of validation rules built into its specification. Those rules were part of the report's construct. Adding a new validation required re-engineering the report. Removing an incorrect validation - or one that could not be performed because the underlying data did not support it - required re-engineering the data mart. Neither was fast. Neither was flexible.

The result in practice: validation coverage was determined by what was anticipated at the time the report was built, not by what the data actually required at the time of query. Banks operated with known validation gaps because closing them was architecturally expensive - a data engineering project, not a configuration change.

This has a direct regulatory consequence that the BCBS 239 compliance picture makes visible. More than a decade after the Basel Committee established risk data aggregation principles requiring banks to aggregate risk data accurately, completely, and in a timely manner, fewer than 30 percent of Tier-1 banks report full compliance. The gap is not primarily a data quality gap. It is a validation architecture gap - the inability to apply the validations that compliance requires without rebuilding the data infrastructure that would carry them.

The bcbs 239 connection

The persistent BCBS 239 compliance deficit - under 30% of Tier-1 banks fully compliant more than a decade after the principles were established - is not primarily a data engineering problem. It is a validation architecture problem. The predefined report model cannot accommodate the validation requirements of an evolving regulatory standard without continuous re-engineering. Query-level validation intelligence resolves this structurally.

2.2 AI Enables Query-Level Validation Without Disturbing the Underlying Data

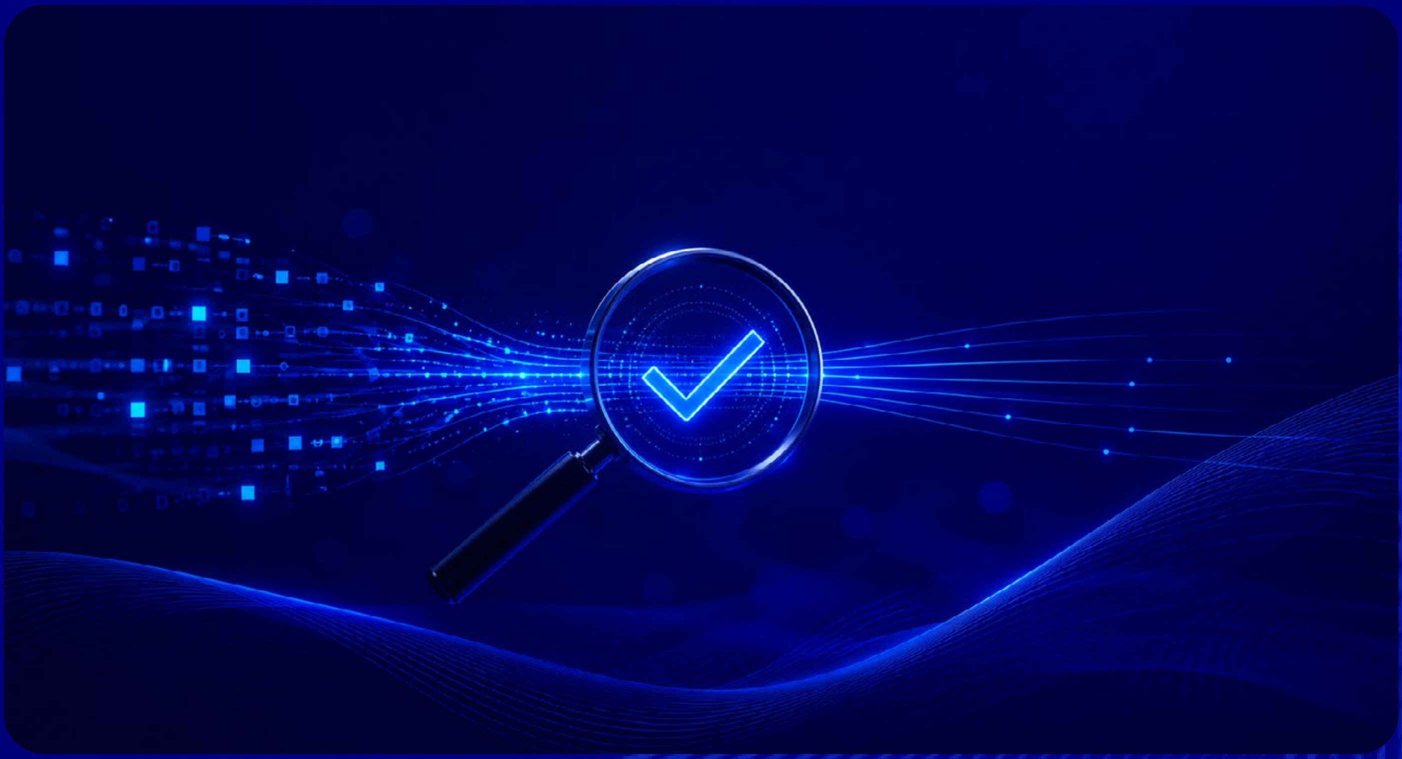
AI changes the validation model entirely. Validations can now be embedded in the query layer - applied at the time of data retrieval, on top of the existing data mart, without altering the data mart's structure or the underlying application. The logic is not stored in the data mart. It is embedded in the query.

This architectural shift - from what needs to be in store to what is to be in the query - has three specific consequences that are directly relevant to the banking CIO's regulatory and operational responsibilities.

First, new validations required by regulatory change can be implemented at the query level without a data mart rebuild. A new reporting requirement under BCBS 239, or an additional validation specified in a regulatory examination finding, does not require a data engineering project before it can be operationalised. It requires a validation rule in the intelligence layer - a change that can be made, tested, and deployed in the time it would previously have taken to scope the data engineering requirement.

Second, the AI validation layer can identify which validations it cannot perform. In the predefined report model, a validation that the underlying data cannot support fails silently or is excluded from the report specification. In the query-level validation model, the AI flags what it cannot perform - producing a report that distinguishes validations that passed, validations that failed, and validations that could not be performed because the underlying data does not support them. For the first time, the governance function has complete visibility into the validation coverage gap - not just the output of the validations that ran.

Third, the feedback from validation execution can be used to learn which validations are necessary, which are redundant, and which are incorrectly specified. The intelligence layer can identify patterns in validation failures that indicate either data quality problems in the underlying data mart or incorrect validation specifications in the query layer - and surface them for human review and correction. Governance becomes a continuous learning loop, not a periodic review cycle.



A fourth and more fundamental shift:

The validations AI generates are not aggregated. In the predefined report model, a validation applied to a risk data set was designed for a period, a product category, or a customer segment - a generalised standard that all data in that group must meet. AI-generated query-level validations operate at a different level of granularity entirely. They are specific to the individual customer or the individual transaction being assessed. And they are informed by patterns - behavioural patterns specific to that customer's history, typological patterns specific to that transaction type, anomaly patterns specific to that moment in the market.

The result is a validation that is simultaneously more granular and more intelligent than any predefined rule could be. A validation that tells the institution not just whether the data meets a general standard, but whether it is consistent with what this specific context should look like.

Named mechanism:
Query-level validation intelligence

AI-generated validation logic applied at the point of data retrieval, leaving the underlying data mart undisturbed. Critically: these validations are not aggregated constructs applied uniformly across segments. They are context-specific - generated for the individual customer or transaction - and pattern-enhanced - informed by behavioural and typological patterns specific to that context. Provides complete visibility into validation status across three categories: validations passed, validations failed, and validations that could not be performed due to underlying data limitations. Enables new regulatory validation requirements to be operationalised without data mart rebuilds.

Implementation reference

Maveric's PRISM platform operationalises query-level validation intelligence - applying validation, reconciliation, lineage tracking, evidence capture, and privacy governance at the query and intelligence layer rather than embedding them in data mart constructs. In regulatory reporting implementations, this approach has enabled institutions to implement new validation requirements within days of a regulatory finding rather than months, while providing the complete validation status reporting - passed, failed, unperformable - that examination responses require.

2.3 Real-Time Semi-Autonomous Systems for Lending, Deposits, and Payments

The shift to query-level validation intelligence has a direct bearing on the reliability of real-time semi-autonomous systems - the lending, deposits, and payments decisioning engines that represent the highest-stakes applications of AI-enabled core banking modernisation.

A real-time lending decision system that operates on data validated at the query level has a fundamentally different reliability profile from one that relies on predefined data mart validation. The difference is not in the model's accuracy under controlled conditions - both will achieve similar benchmark results. The difference is in what happens at the edge: the transaction that falls outside the predefined validation rules, the customer profile that contains an anomaly the data mart validation did not anticipate, the regulatory requirement that was updated after the last data mart release.

In the predefined validation model, these edge cases pass silently - either generating an incorrect decision or triggering a manual review that the semi-autonomous system was designed to eliminate. In the query-level validation model, they surface - flagged by the intelligence layer, with context assembled from the full data surface including unstructured fields, and routed for the appropriate response.

For deposits and payments, the operational reliability implication is equally significant. Real-time payment systems operating without query-level validation carry the risk that a transaction is processed on a data state that contains an undetected inconsistency - an account status that has not been updated, a sanction flag that was added to the data mart after the last batch cycle, a KYC status that does not reflect the most recent verification result. Query-level validation closes this gap in real time, at the transaction level, without requiring the underlying systems to have resolved the inconsistency before the payment instruction is processed.

The semi-autonomous trust requirement

Real-time semi-autonomous systems for lending, deposits, and payments are only as trustworthy as the validation infrastructure that governs the data they act on. Query-level validation intelligence is not a governance add-on to these systems. It is the trust mechanism that makes semi-autonomous operation at scale defensible - to regulators, to auditors, and to the institution's own risk function.

BATTLEGROUND 3

From Service-Level Governance to Intelligence Layer Governance

From API Version Control to Intelligence Artefact Governance

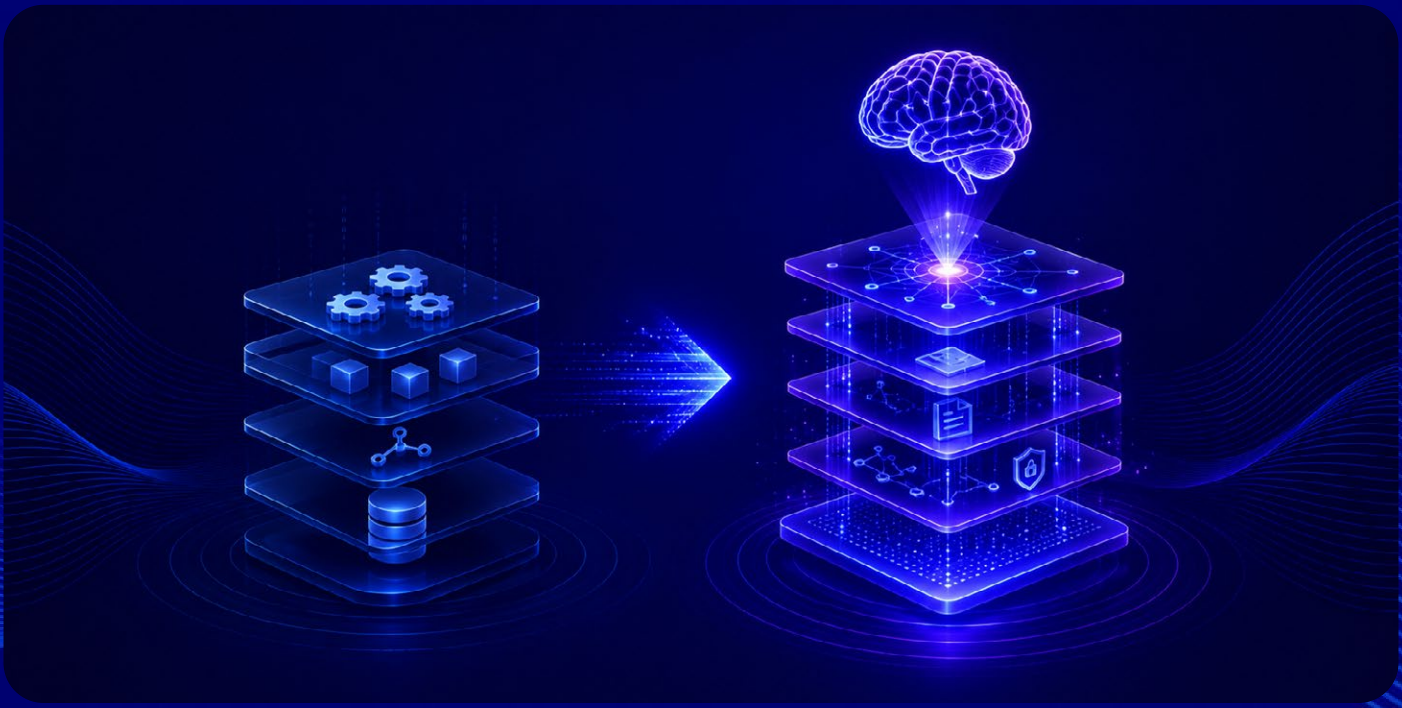
3.1 The Governance Level Has Moved

In the service-oriented architecture that preceded AI, governance was maintained at the service and API level. When a service changed, the version changed, consuming applications updated their calls accordingly, and the governance record was the service version history. API version control was the primary mechanism by which the integration layer was governed.

This approach worked because the service was where the logic lived. The service definition determined what data was returned, what validations were applied, and what format the output took. Governing the service meant governing the decision logic.

In the AI-first core, the decision logic has moved. It lives in the intelligence layer: the prompts that direct AI agents, the context specifications that define what data assembly a decision requires, the query-level validation rules that govern what validations are applied to a data retrieval, and the agentic workflows that orchestrate multi-step decision processes. The governance requirement has moved with the logic.

An institution that governs its AI-first core at the API and service level, while leaving the intelligence layer ungoverned, has not governed its decision logic. It has governed the pipes while leaving the intelligence that flows through them without accountability. This is precisely the condition that regulatory examination is designed to find - and increasingly will find, as regulators move from examining whether AI governance frameworks exist to examining whether they are operational.



3.2 What Governing Intelligence Artefacts Requires

Intelligence artefact governance in the AI-first core requires four capabilities that have no direct equivalent in the service-governance model - and that the CIO must treat as first-class governance requirements, not technical configurations.

Prompt and Context Specification Versioning

Prompts and context specifications carry the intelligence configuration of the AI-first core. They determine what the AI system will do when it receives a decision request. They are as consequential to the system's behaviour as a service definition was to the service-oriented architecture - and they require the same governance: version control, change management, audit trail, and rollback capability.

In practice, this means treating every prompt and context specification that drives a consequential banking decision - a lending inference, a fraud determination, a compliance report - as a governed artefact with a versioned history, a change approval process, and an audit trail that can be produced on regulatory demand.

Validation Rule Lineage

Query-level validation rules - the logic that governs what validations are applied to a data retrieval - must be traceable. For a regulatory report or a compliance determination, the institution must be able to produce the lineage of the validation: which rules were applied, at what version, to what data, with what result. This is the intelligence-layer equivalent of the audit trail that service-level governance provided for API calls.

Without validation rule lineage, the institution cannot demonstrate to a regulator that its validation coverage was adequate at the time a specific report was produced. It can show the output. It cannot show the governance of the process that produced it. In a BCBS 239 examination context, this distinction carries direct regulatory consequence.

AI enables a further and more precise level of lineage that was not available in the pre-AI governance model. In the service-oriented architecture, lineage was established at the document or code level - a decision could be traced back to the document from which its logic derived, or the code module in which its validation was implemented. This was the finest granularity available.

AI granularises lineage to the paragraph and sentence level. If a query-level validation rule is informed by a regulatory guidance document, an internal credit policy, or a risk management framework, the system can now identify not just which document informed it, but which specific paragraph or sentence within that document. When a compliance determination is challenged, the institution can point to the exact passage - not just the source. When a policy document is updated, the system can identify which specific sentences changed and which validation rules, and therefore which decisions, those sentences informed. The scope of impact assessment narrows from a document to a passage. Auditability becomes faster, more precise, and more defensible under examination.

Agentic Workflow Governance

Where AI agents orchestrate multi-step decision processes - assembling context from multiple data sources, applying validation logic, generating outputs, and routing exceptions - the workflow itself requires governance. The institution must be able to account for what the agent did, in what sequence, with what data, and with what result - at the level of a specific customer, a specific transaction, and a specific moment in time.

This is the governance requirement that semi-autonomous systems create and that the predefined report and service model was not designed to meet. It requires agentic workflow logging, exception capture, escalation routing, and audit reconstruction capability - built into the agentic architecture from the point of design, not added retrospectively after a governance finding.

Exception and Escalation Management

The intelligence layer will encounter situations it cannot resolve - data that does not support a required validation, a context specification that produces an ambiguous result, a workflow that reaches a decision point the agent was not configured to handle. In the predefined report model, these situations were handled by the report failing or by silent omission. In the intelligence layer governance model, they must be captured, surfaced, and routed for human intervention.

Exception and escalation management is the governance mechanism that makes the intelligence layer's transparency real rather than theoretical. The institution's governance function must know, in near real time, what the intelligence layer could not do - so that the gaps in automated decision coverage can be managed deliberately rather than discovered retrospectively.

Named mechanism:
Intelligence layer governance

The governance framework that applies version control, lineage tracking, exception management, and audit capability to the prompts, context specifications, query-level validation rules, and agentic workflows that carry decision logic in an AI-first core. Critically: lineage is granularized from the document level to the paragraph and sentence level - enabling the institution to trace any validation rule or decision logic to the exact passage that informed it, not just the source document. The intelligence-layer equivalent of API version control - extended to cover the new artefacts that carry governance obligation in the AI-first architecture, with finer attribution granularity than any previous governance model provided.

The trust requirement

Without intelligence layer governance, the AI-first core has the same fundamental accountability gap as the batch-cycle core before it - consequential decisions being made through logic that cannot be inspected, audited, or explained to a regulator. The governance layer has moved from services to intelligence artefacts. The governance obligation has not changed. The institutions that build intelligence layer governance before their first regulatory examination of the AI-first core are the ones that pass it. The ones that build it after are the ones that use it as a remediation programme.

CIO PLAYBOOK

The Four Directives for AI-Enabled Core Banking Modernisation

The three battlegrounds above define the territory of AI-enabled core banking modernisation. The four directives below are the CIO's operating instructions - the specific commitments that separate institutions that modernise successfully from those that build a faster version of the same fragility.

01 | Stop treating batch-to-real-time as the primary modernisation objective.

Real-time is necessary for time-sensitive decisioning. It is not sufficient for intelligence. The more important shift is from predefined constructs to context-specific intelligence - and that shift requires agentic context assembly, unstructured data integration, and query-level validation capability, not just a faster data pipeline. Measure modernisation progress by the intelligence capabilities unlocked, not by the data pipeline latency achieved.

02 | Build query-level validation intelligence before your next regulatory reporting cycle and build it at the level of the individual context, not the aggregate segment.

The BCBS 239 compliance gap is not primarily a data engineering problem. It is a validation architecture problem that query-level validation intelligence can close without a data mart rebuild. The additional precision AI brings - validations specific to the individual customer or transaction, informed by patterns rather than aggregated thresholds - makes those validations more defensible, not just faster to deploy. The complete validation status report that AI-enabled query-level validation produces - passed, failed, unperformable - is the documentation standard that regulatory examination is moving toward. Build it before the examination, not in response to it.

03 | Extend your AI-accessible data surface to include unstructured data.

The transaction narrative, the compliance note, the document content - these are the data that carry context the structured fields cannot. The institutions building decision models that can reason over unstructured data alongside structured data are building a categorically different level of intelligence than those that are not. This is not a future capability. It is available now. The institution that governs its unstructured data surface and integrates it into its AI decision layer is building a decision-making capability its competitors cannot match without the same architectural foundation.

04 | Govern the intelligence layer with the same rigour you govern the service layer, at a finer level of granularity than services ever required.

Prompts, context specifications, query-level validation rules, and agentic workflows are the new service definitions. They carry the decision logic of the AI-first core. They require version control, lineage tracking at the paragraph and sentence level, exception management, and audit capability. The sentence-level lineage that AI enables is not a technical detail - it is what makes compliance impact assessment faster when a policy changes, and what makes examination responses more precise when a determination is challenged. An AI-first core without intelligence layer governance is a core with ungoverned decision logic. An AI-first core with intelligence layer governance has finer accountability than any deterministic system before it.

The CIO's modernisation mandate in the AI-first era is not to make the core faster. It is to make the core intelligent, at the level of the individual customer, the individual transaction, and the individual moment. And to govern that intelligence with the same rigour and accountability that the deterministic systems it replaces were governed by design.

– CIO Mandate Series,
Paper 2 - The CIO's Guide to AI-Enabled Core Banking Modernisation

CONTINUE THE SERIES

This paper is part of the CIO Mandate Series - four papers examining the four imperatives of AI-first banking transformation.

Paper 1 From Digital-First to AI-First: The CIO's Customer Experience Mandate.

Paper 3 Engineering Trusted Automation in AI-First Banking Operations.

Paper 4 Engineering Trust in AI Compliance and Regulatory Governance.

For the complete framework - the Four-Layer Trust Architecture, the AI Maturity Spectrum, and the full market gap analysis - download the Engineering Trust in AI-First Banking research report at

www.maveric-systems.com/research/engineering-trust-ai-first-banking

Maveric Systems

is a banking-exclusive technology specialist with over 25 years of domain expertise. We partner with global financial institutions to engineer trust in AI-first banking.

While others apply AI at the periphery, we embed it at the core through our proprietary AI @ Scale framework and AI-powered platforms and solutions. Guided by principles that engineer trust, we embed fairness, explainability, reliability, and compliance into every AI solution by design. This enables responsible AI adoption at scale.

Our domain depth across operations and technology in retail and corporate banking, wealth management, and capital markets, combined with a pragmatic, outcome-driven delivery model, ensures that every AI initiative is rooted in contextual relevance and precision.

Backed by dedicated AI Centers of Excellence, a powerful ecosystem of technology platforms, and recognition from leading industry bodies, we are the trusted engineering partner for the AI era.

The Maveric Edge

25+

years of domain mastery

7+

Operations and Technology AI CoEs

12+

AI powered Platforms and Solutions

AI@Scale

Scale Proprietary Framework for AI adoption at Scale

Maveric NXT Inc
5, Independence Way, Suite 300, Princeton, NJ 08540 USA
Reach us: marketing@maveric-systems.com