THOUGHT LEADERSHIP PAPER

AI in Enterprise Architecture

A Pragmatic Path to Living Architecture

A Thought Leadership Paper for Large Financial Institutions

April 2026

Rob Vugts

GenAI Engineer & Strategist / Senior Technical Architect

info@aichitect.eu | aichitect.eu | linkedin.com/in/rob-vugts

Table of Contents

Executive Summary

Enterprise Architecture has long been one of the most strategically important yet chronically under-resourced disciplines in large organisations. EA teams at major banks carry an extraordinary mandate — aligning technology to business strategy, managing risk across vast and complex application landscapes, governing change, and ensuring regulatory compliance — yet they often operate with tools and processes that have changed little in the last decade.

The emergence of capable AI systems in 2024-2026 creates a meaningful opportunity for the EA discipline. Not a revolution, but a genuine shift in what is tractable: AI can reduce the manual overhead of landscape maintenance, assist in drafting assessments, and make architecture knowledge more accessible across the organisation. Used carefully and with appropriate oversight, these capabilities can help EA teams operate more effectively at a time when the demands placed on them — regulatory, strategic, and operational — are intensifying.

The vision at the centre of this paper is living architecture: a continuously maintained, queryable model of the enterprise that reflects reality more closely than today's static documentation. This is a compelling direction, but it is important to state clearly what it is not: it is not a system that reasons autonomously over your architecture, produces reliable decisions without human review, or eliminates the need for skilled enterprise architects. The AI components of any such system are probabilistic tools that require human validation, high-quality underlying data, and explicit governance — none of which can be assumed in a large bank's existing landscape.

This paper is written for practitioners and decision-makers who need a realistic picture of what AI can and cannot do for EA, what the prerequisites are, what can go wrong, and how to pursue the opportunity in a way that is defensible to regulators, auditors, and experienced architects.

1. The Role of the Enterprise Architect

What Enterprise Architects Are Responsible For

The Enterprise Architect occupies a uniquely cross-cutting role. Unlike solution architects who focus on specific systems or projects, or business analysts who work within defined processes, the EA is responsible for the coherence of the whole: ensuring that the sum of thousands of technology decisions serves the organisation's strategic goals, manages risk, and avoids unnecessary complexity.

In a large financial institution, this translates into a set of interconnected responsibilities:

- Architecture governance — defining principles, standards, and patterns that guide how technology is selected, designed, and deployed across the organisation
- Application portfolio management — maintaining visibility over the full landscape of systems, their capabilities, dependencies, costs, and lifecycle status
- Business capability mapping — modelling the capabilities the business needs and tracing how technology enables (or fails to enable) them
- Transformation roadmapping — translating business strategy into multi-year technology change programmes, sequencing investments, and managing interdependencies
- Risk and compliance — ensuring the architecture supports regulatory obligations (DORA, Basel IV, EBA IT risk guidelines), identifying systemic vulnerabilities, and supporting audit and resilience planning
- Change governance — reviewing proposed changes, assessing their architectural impact, and ensuring they align with approved standards and the target architecture

What Their Day-to-Day Work Looks Like

Despite the strategic nature of the role, the EA's day-to-day is dominated by operational overhead:

- Maintaining the architecture repository — updating models in tools like LeanIX, Ardoq, or Sparx EA to reflect the current state of the landscape
- Producing artefacts — creating and updating architecture views, capability maps, roadmaps, and decision records in formats suited to different audiences
- Running and preparing for review boards — assessing solution designs submitted by project teams, checking for alignment with standards, and presenting findings
- Responding to ad-hoc requests — producing impact assessments when a business unit proposes a change, answering questions from risk teams, supporting audit requests
- Stakeholder engagement — facilitating workshops, interviewing business and technology leaders to gather architecture-relevant information

The gap between what EAs are supposed to do — strategic thinking, systemic reasoning, forward-looking guidance — and what they actually spend most of their time doing is one of the defining frustrations of the discipline. AI has the potential to close this gap, but only if the underlying data and governance conditions are in place.

What They Produce

- Business Capability Models (BCMs) — structured maps of what the organisation does, independent of how it currently does it
- Application portfolio assessments — analyses of the current application landscape against criteria such as business fit, technical quality, cost, and risk
- Architecture principles and standards — the guardrails within which technology decisions are made
- Architecture Decision Records (ADRs) — documented records of significant decisions, their rationale, and their implications
- Impact assessments — analyses of how proposed changes affect the broader architecture
- Target architecture blueprints — descriptions of the desired future state across technology domains
- Transformation roadmaps — sequenced plans for moving from current to target state
- Compliance and risk reports — evidence that the architecture meets regulatory and internal risk requirements

What Information and Systems They Rely On

To do this work, EAs draw on a broad range of information sources:

Category	Examples
EA repository	LeanIX, Ardoq, Sparx EA, BiZZdesign
CMDB / IT asset data	ServiceNow, BMC Helix
Project and change management	Jira, ServiceNow Change, Azure DevOps
Document repositories	Confluence, SharePoint, internal wikis
Financial data	IT cost allocation systems, vendor contract databases
HR and organisational data	Org charts, capability assessments
Vendor and product data	Vendor roadmaps, EOL databases, Gartner
Regulatory documentation	Internal policy libraries, regulatory publications
Code and infrastructure	Git repositories, cloud infrastructure inventories (AWS Config, Azure Resource Manager)

The fragmentation of this information across dozens of systems — with inconsistent data quality, different ownership models, and limited integration — is not merely an operational inconvenience. It is the single most important factor determining whether AI can be usefully applied to EA.

2. Challenges in Large Organisations Like Banks

The Scale Problem

A major bank is one of the most complex organisations on earth from an architecture perspective. It may operate thousands of applications, dozens of business lines, multiple legal entities across different jurisdictions, and a technology landscape accumulated over five or more decades of organic growth, acquisitions, and transformation programmes. The EA team responsible for maintaining coherence across this landscape is almost always a fraction of the size the task demands.

This creates a fundamental tension: the mandate is enterprise-wide, but the capacity is not. Something has to give, and what usually gives is currency and depth. Architecture repositories become stale. Coverage is uneven — well-documented areas sit alongside vast blind spots. Impact assessments rely on the personal knowledge of individual architects rather than reliable, up-to-date data.

The Documentation Paradox

Architecture documentation in large banks exists in a state of perpetual lag. The effort required to keep it current is enormous, but the penalty for not keeping it current is that the documentation loses credibility and ceases to be used. EA teams can find themselves trapped in a cycle where they spend significant time maintaining documentation that nobody trusts because it is never quite up to date.

This is not a failure of effort or intention. It is a structural problem: the rate of change in a large bank's technology landscape exceeds what a human team can realistically track manually.

The Governance Bottleneck

Architecture review processes are a source of significant friction in many large banks. Projects must pass through review boards, which require EAs to produce assessments, which requires gathering information, which takes time. When EA teams are stretched, review cycles slow down, project teams learn to work around the process, and the governance function loses effectiveness precisely when it is most needed.

The tension here is between speed and rigour. Business units and project teams want fast answers and minimal process overhead. Risk and compliance functions want thorough, documented assessments. EA teams are caught in the middle, unable to fully satisfy either.

The Knowledge Concentration Risk

Experienced enterprise architects carry enormous amounts of tacit knowledge — about why particular decisions were made, which systems are unexpectedly coupled, where the bodies are buried in the legacy landscape. When those individuals leave, retire, or move roles, that knowledge largely disappears.

This is a risk that AI can partially mitigate — by making existing documented knowledge more accessible and by assisting in capturing decision rationale — but it cannot compensate for knowledge that was never recorded in the first place.

How EAs Currently Cope

- Triage and prioritisation — focusing effort on the highest-risk or highest-visibility areas and accepting gaps elsewhere

- Relationship networks — relying on personal relationships with colleagues across the organisation to get information that isn't in any system
- Periodic refresh cycles — scheduling regular (quarterly or annual) reviews rather than maintaining continuous currency
- Templating and reuse — developing standard artefact templates to reduce the effort of producing each new document
- Delegation to project teams — requiring solution architects on projects to maintain architecture data as part of project governance

These are reasonable coping strategies, but they are compensations for an underlying capacity problem rather than solutions to it.

3. How AI Could Help — and Where It Cannot

What AI Is and Is Not

Before discussing specific applications, it is important to be precise about what current AI systems — specifically Large Language Models (LLMs) — can and cannot reliably do. Clarity here is not pessimism; it is a prerequisite for building systems that hold up under scrutiny.

What LLMs do well

- Synthesising and summarising large volumes of unstructured text
- Generating structured first drafts of documents from provided context
- Identifying patterns and potential inconsistencies within a provided dataset
- Translating technical content into accessible language for different audiences
- Assisting with classification, tagging, and organisation of information

What LLMs do poorly or unreliably

- Reasoning over incomplete data: LLMs do not know what they don't know. An impact assessment generated from a partial dependency graph may look comprehensive while missing critical relationships that were never recorded.
- Deterministic outputs: the same query posed twice may yield different responses. This is problematic in governance contexts where consistency and reproducibility matter.
- Hallucination: LLMs can generate plausible-sounding but factually incorrect content — including invented system names, non-existent integrations, or fabricated regulatory references. In an EA context, this is an audit risk.
- Deep causal reasoning: LLMs approximate reasoning through pattern matching. They struggle with genuine multi-step logical inference, particularly over structured relational data.
- Real-time accuracy: LLMs have no inherent awareness of the current state of your systems. They can only reason about what is in their context window at the time of the query.

These limitations do not make AI useless for EA — they define how it must be used. AI outputs in an EA context should always be treated as probabilistic drafts, not authoritative conclusions, and every production use case requires human review, traceable sourcing, and explicit confidence signalling.

Where AI Brings the Most Value

1. Assisted landscape maintenance

AI can scan connected data sources — CMDB, code repositories, cloud infrastructure inventories, CI/CD pipelines — and propose updates to the architecture repository. The operative word is

propose: the EA validates and approves. This shifts the bottleneck from data gathering to data validation.

Trade-off: the quality of AI-proposed updates depends entirely on the quality of connected source systems. If the CMDB is 60% accurate, AI-proposed updates will inherit that inaccuracy — and potentially propagate it with a false veneer of automation.

2. Impact analysis structuring and gap detection

When a change is proposed, AI can structure the analysis: organising available dependency data into a consistent framework, cross-referencing relevant policies and past decisions, and producing a first draft with explicitly signposted gaps. This is deliberately positioned as structuring and gap detection rather than impact analysis per se — the distinction matters.

Impact analysis is arguably the hardest EA problem. It depends on hidden dependencies, tribal knowledge, and runtime behaviour that are rarely captured in formal architecture data. AI is weakest exactly where impact analysis is hardest.

Trade-off: the system must explicitly flag what it was unable to check. An output that presents incomplete coverage as a complete assessment is more dangerous than no output at all.

3. Standards and governance review assistance

AI can perform a first-pass check of solution designs against documented architecture principles, flagging potential deviations for human review. This accelerates review board preparation without replacing architectural judgement.

Trade-off: standards documents are often ambiguous, context-dependent, and subject to legitimate interpretation. AI will apply standards literally, potentially flagging compliant designs as non-compliant or missing nuanced violations.

4. Natural language access to architecture knowledge

A well-implemented retrieval layer over the EA repository and document store can enable plain-language querying — a genuine quality-of-life improvement for both EAs and business stakeholders.

Trade-off: natural language interfaces create a risk that users develop unwarranted confidence in the completeness of responses. A query that returns a clean answer may be missing half the actual dependencies. Managing user expectations is a design challenge, not just a communication one.

5. Document drafting and summarisation

AI excels at producing structured first drafts of artefacts — capability maps, roadmap narratives, meeting summaries, decision records — from provided inputs. This is perhaps the most immediately practical application, with the fewest reliability concerns, because the EA is working alongside the AI and can readily validate the output.

The Concept of Living Architecture

The underlying aspiration connecting these capabilities is living architecture: a model of the enterprise that is maintained continuously rather than periodically, reflects reality more closely than static documentation, and is accessible across the organisation.

In practice, this vision must be interpreted conservatively. Most organisations will achieve living architecture first in curated and semi-structured domains — architecture repositories, decision records, and document stores. The integration of runtime observational data is a significantly more complex challenge requiring reconciliation across fundamentally different abstraction layers.

- Horizon 1-2: living applies primarily to static and curated data — repositories, documents, change records
- Horizon 3+: partial and selective integration of runtime signals, with inherent limitations in coverage and accuracy

Two related concepts are useful reference points:

- Digital twin of the enterprise: the analogy is instructive but should not be overstated. A manufacturing digital twin is grounded in sensor data with measurable accuracy. An enterprise architecture model depends on data sources of highly variable quality and on human-assigned meanings that sensors cannot provide.
- Active metadata: well-established in data management — metadata that is continuously maintained, enriched, and linked across systems. Living architecture applies this principle at enterprise scale, treating architecture artefacts as living records with provenance, timestamps, and relationships.

What AI uniquely enables in 2026 is the ability to bridge unstructured information — documents, meeting notes, strategy papers — with structured architecture data. This bridging capability is real and valuable. It does not extend to autonomous architectural reasoning, self-maintaining repositories, or reliable decision-making without human oversight.

How This Transforms the EA Role

The EA of the future spends less time as an information gatherer and document producer, and more time as a validator, interpreter, and strategic adviser. This shift should not be mistaken for easier — the shift from producing outputs to critically reviewing AI-generated outputs requires a specific kind of intellectual rigour that is at least as demanding as the original work, and potentially more so because the reviewer must remain alert to errors that are not obvious.

Enabling Business Users — With Appropriate Guardrails

One of the more significant opportunities is enabling business stakeholders to query the architecture in natural language. This requires careful design. Business users querying an AI-assisted system may not appreciate the difference between 'the system found no relevant dependencies' and 'the system's data does not cover that domain.' All interfaces for non-expert users should explicitly display data coverage indicators, confidence levels, and prompts to consult an EA for consequential decisions.

Bridging Architecture Disciplines

Living architecture is most powerful when it connects across disciplines. Today, Enterprise Architecture, Security Architecture, and Solution Architecture typically operate in silos. AI creates the opportunity to build a shared foundation:

- Security architecture can be surfaced as a layer within the enterprise model — control mappings and security classifications linked to the applications and capabilities they protect.
- Solution architecture benefits when solution architects have access to the enterprise model and receive early feedback on how their designs relate to existing patterns, standards, and capabilities.
- Data architecture — increasingly critical in a bank's regulatory and AI context — can be linked to the application and capability model, making data lineage and ownership visible across the landscape.

The integration of disciplines should be sequenced, not attempted simultaneously. Connecting architecturally immature domains can introduce noise and inconsistency rather than coherence.

4. The Data Foundation: The Non-Negotiable Prerequisite

This section merits special attention because it is the most frequently underestimated challenge in AI-for-EA programmes — and the one most likely to cause them to fail or to produce outputs that are worse than useless.

> *AI does not solve poor data quality. It industrialises it. A system that generates confident-sounding impact assessments from stale CMDB data, or traverses a dependency graph that is missing 40% of actual relationships, produces worse results than a manual process — because it does so faster, at greater scale, and with an appearance of rigour that may prevent people from questioning it.*

The Real State of Architecture Data in Large Banks

- The EA repository (LeanIX, Ardoq, Sparx EA) contains the curated architecture view — often incomplete, partially stale, and reflecting the opinions of whoever last maintained each section
- The CMDB contains the operational IT asset inventory — often more complete for infrastructure than for applications, and frequently inconsistent with the EA repository
- Project management systems contain planned and in-flight changes — which may or may not have been reflected back into either the CMDB or the EA repository
- Document repositories contain the actual reasoning and context — in unstructured form, inconsistently maintained, and with no guaranteed relationship to the structured data in other systems

No single system tells the truth about the current state of the architecture, and the truth is distributed across all of them in partial, sometimes contradictory form.

The Entity Resolution Problem

One of the most significant and underappreciated technical challenges is entity resolution: determining that 'Payment Gateway (CMDB)', 'PAY-GW-01 (ServiceNow)', 'Payment Processing System (Confluence)', and 'Legacy Payments (LeanIX)' all refer to the same system — or that they don't.

AI can assist with entity resolution, but it cannot solve it without clean, agreed reference data (a canonical application register) to anchor against. Without entity resolution, joining data across systems produces a distorted picture that may be more misleading than useful.

Inconsistent Data Models

Each system an EA relies on was built with a different data model and a different concept of what an 'application,' 'system,' 'service,' or 'capability' is. Mapping these models to each other is not a technical problem AI can solve — it is a conceptual and organisational problem that requires human decisions about what the canonical definitions are.

Data Ownership and the Political Dimension

Architecture data is not politically neutral. The question of whether Application X is 'strategic' or 'legacy' affects investment decisions and team priorities. Teams have incentives — sometimes conscious, sometimes not — to represent their data in particular ways. AI systems that surface this data do not resolve these political tensions. They may amplify them.

What Must Be Done Before AI Can Be Effective

1. Establish a canonical application register — a single, agreed list of applications with unique identifiers, owners, and basic attributes, maintained in a designated system of record
2. Define entity resolution rules — explicit mappings between how the same entities are represented across different systems
3. Assess and baseline data quality — understand the current completeness and accuracy of each connected data source before connecting it to an AI system

4. Assign and enforce data ownership — identify who is responsible for maintaining each category of architecture data, with clear escalation paths for disputes
5. Implement confidence scoring for data — so that AI systems can express lower confidence when drawing on data sources known to be incomplete or stale

This is not glamorous work. It is also not optional. Programmes that skip it and proceed directly to AI tooling typically produce impressive demonstrations that cannot survive contact with production data quality.

5. Failure Modes and Risks

Any serious proposal for AI in EA must address how things can go wrong. The following failure modes are not hypothetical — they are predictable consequences of deploying AI systems over incomplete data with insufficient human oversight.

Incorrect Impact Analysis

An AI system traversing a dependency graph will identify the dependencies that are recorded. It will not identify the dependencies that are missing from the record. In a real bank's architecture data, missing dependencies are the norm, not the exception.

The risk is not that the AI produces an obviously wrong output. The risk is that it produces a plausible, well-structured output that is silently incomplete — and that an EA, under time pressure, approves it without the manual cross-checking that would have caught the gap.

Mitigation: All AI-generated impact assessments must explicitly enumerate what the system was and was not able to check. Coverage gaps must be reported alongside findings, not omitted. The EA must confirm that the coverage is sufficient before approving the assessment.

Missing Dependencies and False Confidence

The most dangerous output an AI system can produce is a high-confidence assessment that is missing a consequential dependency. This is more dangerous than a low-confidence assessment that flags uncertainty, because it may be acted upon without the additional scrutiny it requires.

Mitigation: Confidence scoring must be calibrated to data completeness, not just to the AI's internal certainty. An AI that is 95% confident based on a CMDB that covers 60% of actual dependencies should not be presenting a 95% confidence score.

Hallucinated References and Fabricated Artefacts

LLMs can generate references to systems, integrations, regulations, or policies that do not exist. In a high-volume, time-pressured governance context where outputs are passed through quickly, this is a serious risk.

Mitigation: All specific factual claims in AI-generated outputs must be traceable to a specific source document or data record, displayed alongside the claim. Claims without traceable sources must be flagged clearly, not silently included.

Audit and Regulatory Rejection

Under DORA Article 11 and EBA guidelines on ICT risk management, banks are required to maintain accurate and current documentation of their ICT systems, dependencies, and risk exposures. If AI-generated architecture artefacts are submitted in response to regulatory requests and are subsequently found to be incomplete or inaccurate, the consequences are serious.

Mitigation: Maintain a clear distinction between AI-assisted drafts and approved artefacts. All approved artefacts must carry a human owner who is accountable for their accuracy. AI

tooling should be treated as an ICT vendor and subject to appropriate third-party risk management.

Automation of Errors at Scale

Manual processes have a natural circuit-breaker: a human who is paying attention may notice something is wrong. Automated AI processes can propagate errors at scale before anyone notices.

Mitigation: Human approval gates for all automated updates. Sampling and audit of AI-generated outputs on a regular basis. Rollback capability for repository changes. Anomaly detection on the rate and nature of AI-proposed changes.

Summary of Mitigations

Risk	Mitigation
Incomplete impact analysis	Mandatory coverage gap reporting; EA sign-off on scope adequacy
Hallucinated facts	Source traceability on all specific claims; unsourced claims flagged
False confidence	Confidence scores calibrated to data completeness, not AI certainty
Regulatory rejection	Clear draft/approved distinction; human accountability for all approved artefacts
Errors propagated at scale	Approval gates; sampling; anomaly detection; rollback
Over-reliance by business users	Explicit data coverage indicators; prompts to escalate consequential queries

6. How AI Could Change EA — The Transformed Landscape

The Validation Tax

A central assumption of AI-assisted Enterprise Architecture is that effort shifts from information gathering and document production to validation and interpretation. This shift is real, but it is not a straightforward efficiency gain.

Validation is not cheaper work than creation. In many cases, it is more demanding.

Reviewing AI-generated outputs requires the EA to detect missing information not present in source data, identify hallucinated or unsupported claims, assess whether confidence levels are justified by underlying data quality, and overcome confirmation bias introduced by coherent, well-structured outputs. Well-formatted prose with plausible-sounding conclusions is psychologically harder to challenge than a blank page.

This introduces what can be described as a validation tax — a new category of cognitively intensive effort that requires experienced practitioners and cannot be delegated to junior staff or automated away.

In early stages of adoption, the validation tax may equal or exceed the effort of manual production. Productivity gains are therefore not guaranteed and should be treated as an empirical question, not an assumption.

> **Key Principle**
>
> *AI-assisted EA delivers net value only when: validation effort < manual baseline effort for equivalent quality outputs. Achieving this condition depends on data quality, AI output design, and the maturity of the EA team in reviewing probabilistic outputs.*

Systems that optimise for impressive first drafts without reducing validation effort will fail to deliver net value. They will simply move effort from drafting to checking, without reducing the total.

Confidence-Aware Outputs as a Design Principle

Every AI output in an EA context should be designed around three questions: What does this claim, based on what evidence, with what confidence?

- Source attribution: every factual claim (not general reasoning) is linked to the specific data record or document from which it was derived
- Data coverage reporting: every assessment includes a statement of which data sources were queried and what their known quality or completeness issues are
- Confidence levels: expressed not as a single score but as a structured statement — e.g. 'high confidence on application dependencies (EA repository, last updated 3 weeks ago); low confidence on data flows (Confluence documents from 2022)'
- Gap identification: explicit enumeration of what the system could not assess and what human investigation is therefore required

What this looks like in practice — a confidence-aware output for an impact assessment:

```
IMPACT ASSESSMENT DRAFT -- Payment Gateway Decommission
Generated: 2026-03-14 | Status: DRAFT -- requires EA review before use

DEPENDENCY COVERAGE SUMMARY
  Overall coverage score: 72%
  Sources queried:
    [OK] LeanIX application registry    (updated 18 days ago)
    [OK] Confluence architecture docs   (mixed dates, some from 2022-2023)
    [--] ServiceNow CMDB                (not connected - manual check required)
    [--] Runtime integration monitoring (not available)

FINDINGS (confidence: source-referenced only)
  High confidence:
    - Payment Gateway supports 4 mapped capabilities (source: LeanIX, 3 records)
    - 7 downstream application dependencies identified (source: LeanIX)
  Low confidence:
    - Data flow to Regulatory Reporting System inferred from 2023 Confluence doc;
      currency unverified

GAPS - MANUAL INVESTIGATION REQUIRED
  ! No CMDB data available: infrastructure dependencies not assessed
  ! Runtime API calls not mapped: undocumented integrations may exist
  ! 3 of 7 dependent applications have no EA record -- owner unknown

NEXT STEPS FOR EA
  Verify CMDB entries for all 7 dependent systems
  Confirm Regulatory Reporting data flow with system owner
  Investigate unmapped applications before assessment can be finalised
```

This output does not look like a polished finished product. That is intentional. Its value is in making the gaps visible, not in projecting false completeness.

This design is not only about trust — it is about reducing the validation tax. The primary purpose of source attribution, coverage reporting, and structured confidence is to enable the EA to focus validation effort where it is most needed, rather than re-validating the entire output indiscriminately.

Governance in the Age of AI-Assisted EA

Accountability

AI systems do not hold accountability; they provide decision support. All approved architecture artefacts must have a clearly identified human owner.

However, in a regulated environment, assigning full accountability to individual enterprise architects for AI-assisted outputs may expose them to disproportionate personal risk, particularly where outputs depend on probabilistic models and incomplete data. Accountability must be embedded in a structured governance process, not placed solely on individuals. This includes clearly defined validation procedures, explicit documentation of data sources and known limitations, defined thresholds for when additional manual investigation is required, and escalation paths for uncertain or high-impact decisions.

The objective is to ensure that responsibility is institutionalised, auditable, and repeatable — not dependent on individual judgement under uncertainty.

Explainability

Can the reasoning behind an AI output be explained in terms an experienced architect or regulator can evaluate? Outputs that cannot be explained should not be used in formal governance processes.

Reproducibility

Auditors and regulators will ask not just 'what did this output say?' but 'how was it produced, and can you reproduce it?' This requires deterministic replay capability: given the same query, the same retrieved context, and the same model version, the system should produce an equivalent output. In practice this means versioning both the prompt templates used and the model version at the time of each output, and storing the exact retrieved context alongside the generated response.

Prompt and Model Versioning

Changes to prompt templates or model versions must be treated as changes to a governance process — requiring review, testing against known-good scenarios, and documented approval. A prompt template is effectively a policy document; it should be under version control and change management.

Regulatory Alignment

The EU AI Act (2026) classifies AI systems used in regulated financial services processes as high-risk, with corresponding obligations around transparency, human oversight, and documentation. DORA requires that ICT third-party risks — including AI vendors — be subject to appropriate contractual and monitoring controls. Any AI-for-EA programme must be designed with these obligations in mind from the start, not retrofitted later.

Bridging Architecture Disciplines

A well-designed living architecture platform connects rather than siloes architecture disciplines. The practical path is sequential:

1. Establish the enterprise architecture layer as the foundation — canonical application register, capability model, basic dependency data
2. Add security architecture as a layer — control mappings, classification labels, linked to applications in the EA repository
3. Integrate solution architecture — real-time access to the enterprise model for solution architects, early-stage feedback on new designs
4. Add data architecture — data lineage and ownership linked to applications and capabilities

Each step builds on the previous one and adds complexity. Attempting to integrate all disciplines simultaneously is likely to produce a system that does nothing well.

7. The Data and Technical Architecture

Layered Architecture, Not Multi-Agent Complexity

For most organisations starting this journey, a simpler three-layer architecture is more appropriate, more auditable, and more maintainable than sophisticated multi-agent systems:

- Layer 1 — Retrieval: finds and returns relevant information from connected data sources. This includes both structured queries (against EA repository APIs, CMDB) and semantic search (over documents using vector embeddings). The retrieval layer does not reason — it fetches. Every retrieval is logged with its source and timestamp.
- Layer 2 — Reasoning: the LLM layer. It receives a query and the retrieved context, and generates a structured response. The reasoning layer does not have access to data sources directly — it works only with what the retrieval layer has provided. This separation is critical for auditability.
- Layer 3 — Validation: checks the output against known constraints before presenting it to the user. Are all cited sources real and retrievable? Do confidence levels reflect data completeness? Are coverage gaps reported? This layer can be partly automated and partly human.

Advantage	Limitation
Each layer is auditable independently	Less capable of iterative reasoning over complex queries
Failures are localised and diagnosable	Requires careful design of retrieval to ensure relevant context is always fetched
Easier to explain to regulators	Cannot self-correct when initial retrieval is incomplete
Lower operational complexity	More limited in handling genuinely novel scenarios

Why Existing Tooling Is Insufficient on Its Own

A frequently-asked question is: 'Isn't this just Microsoft Copilot plus LeanIX's built-in AI?' It is worth answering directly.

- LeanIX / Ardoq native AI provides AI features scoped to their own data model. They cannot reason across the repository, the CMDB, the document store, and the change management system simultaneously, and cannot enforce organisation-specific governance workflows or confidence calibration.
- Microsoft Copilot (M365) provides strong document drafting and summarisation, but has no native understanding of architecture models, no integration with EA repositories, and no domain-specific governance logic.
- ServiceNow AI / AIOps is strong on CMDB discovery and change management, but does not bridge to architecture capability models or produce architecture-framed impact assessments.

The gap none of these tools fills is the cross-platform reasoning layer: the ability to synthesise across structured EA data, operational CMDB data, and unstructured document knowledge, within a governance framework specific to the bank's architecture standards and confidence reporting requirements.

Build vs Buy

The recommended approach: use commercial platforms as systems of record, and retain control at the integration and governance boundary.

- EA repository platforms (LeanIX, Ardoq, BiZZdesign) — use native AI features first; they are designed for the tool's data model and maintained by the vendor
- Collaboration platforms (Confluence + Atlassian Intelligence, SharePoint + Copilot) — provide document-layer AI deployable relatively quickly
- ITSM platforms (ServiceNow) — evaluate native AI features for CMDB and change management use cases
- Custom build — for the integration layer, cross-platform reasoning, organisation-specific governance workflows, and confidence reporting

This is not an argument for building a full AI platform from scratch. It is an argument for retaining control at the integration and governance boundary — which is precisely where regulatory and architectural accountability reside.

Reference Architecture

Layer	Purpose	Commercial Options	Custom Required?
EA Repository	Application portfolio, capability model, architecture views	LeanIX, Ardoq, BiZZdesign	No — use native AI features first
ITSM / CMDB	Asset inventory, change management, incident data	ServiceNow, BMC Helix	No — evaluate native AI features
Document & Knowledge	Architecture documents, policies, ADRs, meeting notes	Confluence + Atlassian Intelligence, SharePoint + Copilot	Partial — custom retrieval may be needed
Retrieval Layer	Semantic search, structured query routing	Azure AI Search, pgvector, LlamaIndex	Yes — for cross-platform integration
Reasoning Layer	LLM-based synthesis and generation	Azure OpenAI (private endpoint), Anthropic Claude (private API), on-premise Llama/Mistral	Yes — prompt engineering, output formatting
Validation Layer	Source tracing, confidence scoring, gap reporting	Custom	Yes
Interface	Conversational access for EAs and business users	EA platform native chat, Teams integration	Partial
Audit & Monitoring	Query logging, output sampling, quality tracking	Azure Monitor, custom logging	Yes

Privacy, Data Sovereignty, and Model Governance

For a European bank, these are hard constraints, not preferences:

- No architecture data to public AI services: DORA and EBA ICT risk guidelines make this a firm requirement. Azure OpenAI with private endpoints, or on-premise open-weight models, are the viable options.
- Data residency within the EU: confirm and document that all AI processing occurs within EU jurisdiction
- LLM vendor as ICT third party: the AI model provider must be subject to the same third-party risk management process as any critical ICT vendor — including contractual protections, exit strategies, and concentration risk assessment

- Model change notification: contracts with AI vendors should include notification requirements for significant model updates, enabling the bank to assess impact before changes take effect

8. Realistic Adoption Challenges

Cultural Resistance Among EA Teams

Enterprise architects are, by professional inclination, sceptical of claims that are not grounded in evidence. They have watched multiple waves of tooling arrive with promises of automated insight and depart having added complexity without commensurate value. AI is not obviously different from their perspective.

This scepticism is healthy and should be respected, not managed around. The appropriate response is to involve experienced EAs in designing the system, to make early use cases ones where their expertise is demonstrably valuable in reviewing AI outputs, and to be transparent about the system's limitations rather than hiding them.

Trust Is Built Slowly and Lost Quickly

The first time an AI-assisted impact assessment misses a significant dependency, or produces a hallucinated regulatory reference that is not caught before a review board, the credibility of the entire programme is at risk.

This argues for an extremely conservative approach to early use cases: choose scenarios where the data quality is known to be good, where the outputs are easy to verify, and where the consequences of an error are recoverable. Do not put AI-generated outputs into formal regulatory submissions until the system has demonstrated consistent accuracy over an extended period.

Organisational Inertia and Process Change

EA processes in large banks are embedded in governance frameworks, committee structures, and role definitions that have evolved over years. Common resistance patterns include:

- Review board members who are comfortable with the existing process and sceptical of AI-generated inputs
- IT teams who are reluctant to expose their system data to a centralised AI platform
- Risk and compliance functions who are unconvinced that AI-assisted governance meets their standards for rigour and accountability
- Project teams who fear that AI-assisted review will be faster and more thorough, increasing the chance of their designs being challenged

Each of these requires a different engagement strategy. The EA function cannot drive AI adoption alone; it needs active sponsorship from CIO, CRO, and COO levels to navigate the institutional resistance.

Skills Gaps

The combination of skills required for an AI-assisted EA function is currently scarce: deep EA domain expertise, data engineering capability, AI/LLM engineering knowledge, and change management experience. Few individuals or teams have all four. This is a hiring and development challenge that should be planned for explicitly, not assumed to be solvable by retraining existing staff.

The Operational Burden of AI Systems

A challenge that tends to be underestimated in implementation planning is the ongoing operational burden of running AI systems in production. This is qualitatively different from the maintenance burden of traditional software:

- Prompt drift: as underlying LLM models are updated by vendors, the same prompt may produce materially different outputs. This requires ongoing regression testing against known-good scenarios.
- Retrieval degradation: as the connected data sources change, the quality of retrieved context can shift in ways that degrade output quality. Regular sampling and quality monitoring are required.
- Model upgrade impact: when a new LLM version is released, the bank must assess whether its prompt templates, validation logic, and confidence calibration remain valid.

The question 'who owns this long-term?' must have a concrete answer before the system moves to production. AI systems are not fire-and-forget; they require a named team with the skills and mandate to maintain them continuously.

9. An Iterative Approach: Managing Investment and Risk

The Case for Incrementalism

A large bank is not the right environment for a 'big bang' AI transformation of its EA function. The risks — to governance continuity, to regulatory compliance, to stakeholder trust — are too high, and the unknowns are too numerous. The right approach is iterative: deliver real value quickly with minimal integration, use early successes to build the evidence base and organisational confidence for deeper investment, and expand the capability systematically.

Critically, the data foundation work described in Section 4 does not need to be complete before value can be delivered — but it must be underway.

Three Horizons of Value

Horizon 1 — AI as a Drafting and Synthesis Assistant (weeks to 2 months)

In this horizon, AI works with context that the EA provides manually. There is no integration with live data sources, and no autonomous retrieval. The value is in the speed and quality of first drafts.

- Upload a solution design document and receive a first-cut architecture review flagging potential standard violations
- Provide a change description and relevant system context; receive a structured impact assessment framework with explicitly listed unknowns
- Upload a strategy document and receive a summary of architectural implications as a starting point for EA analysis
- Generate a first draft of an ADR or capability map from structured notes

What this horizon proves: AI can meaningfully accelerate artefact production. What it does not prove: AI can reliably reason over the live architecture. That comes later, when the data foundation is in place.

Horizon 2 — AI with Targeted Integration (1-4 months)

Connect the AI to one or two high-quality, well-maintained data sources. The EA repository and the document store are the natural starting points.

- Natural language querying of the application portfolio and capability model
- Retrieval-augmented generation over architecture decision records and policies

- Change request context automatically incorporated into impact assessment drafts

This horizon requires the entity resolution and data quality work described in Section 4 to be complete at least for the connected data sources. Connecting an AI to a poor-quality data source is worse than not connecting it.

Horizon 3 — The Living Architecture Platform (6-18 months)

The progressive realisation of the living architecture vision: continuous landscape monitoring, cross-disciplinary integration, democratised access for business stakeholders, and computational governance assistance. This is a programme of connected capabilities, each built on the foundation established in earlier horizons, and each requiring its own data quality prerequisite work.

Key Principles for Iterative Delivery

- Data quality gates: do not connect a data source to the AI system until its quality has been assessed and found adequate for the intended use case
- Measure from the start: establish baseline metrics before introducing AI, so improvements are quantifiable and failures are detectable
- Conservative initial use cases: choose scenarios where verification is easy and errors are recoverable; avoid regulatory submissions and formal audit evidence until the system is well-established
- Transparency over impressiveness: a system that is honest about its limitations will survive longer than one that looks impressive until it fails publicly

The Funding Paradox

A common objection is that the data foundation required for AI-assisted EA — canonical application registers, entity resolution, data ownership governance — has been recognised as necessary for years but repeatedly underfunded. If organisations could not justify this investment before AI, why would AI change that calculus?

AI does not remove this problem. In many cases, it makes it more visible — creating a direct, observable link between data quality and operational effectiveness. This visibility can strengthen the investment case — but it does not guarantee it. Organisations should be prepared for the possibility that AI adoption surfaces long-standing data quality problems without immediately resolving the political and funding constraints behind them.

10. Economic Model and Investment Considerations

Cost Categories

Implementation costs (one-time)

- Data foundation work (entity resolution, data quality assessment, canonical register): this is the most frequently underestimated cost category. For a large bank, budget 6-12 months of data engineering effort before expecting AI to work reliably over multi-source data.
- Integration development (retrieval layer, cross-system connectors, validation layer): typically 3-9 months of engineering effort depending on complexity
- Prompt engineering and workflow design: 1-3 months, requires EA domain expertise in combination with AI engineering skill
- Security review and compliance assessment: 1-2 months, not optional in a regulated environment
- Change management and training: 2-4 months

Operational costs (ongoing)

- LLM inference costs: typically modest for EA use cases, but cost increases significantly if the system is opened to broad business user self-service
- Private cloud infrastructure: hosting costs for private LLM deployment are substantially higher than public API costs; this is the price of data sovereignty
- Maintenance and model updates: AI systems require ongoing maintenance as models are updated and as the bank's architecture evolves
- Human review capacity: do not underestimate the EA time required to review and validate AI outputs; this is a real ongoing cost, not a saving
- Validation overhead (the validation tax): AI-assisted workflows introduce a new category of effort — the systematic review of probabilistic outputs. This effort is non-trivial and requires experienced architects. In early stages, it may equal or exceed manual production effort. Any economic model that excludes this cost will overstate the benefits of AI.

Comparison with Alternative Investment

The relevant comparison is not 'AI vs. nothing' but 'AI vs. hiring more enterprise architects.'

A senior enterprise architect in a major European bank costs approximately 150,000-250,000 EUR per year in total employment cost. An AI-assisted EA programme, at full build-out, might cost 500,000-1,500,000 EUR to implement and 200,000-400,000 EUR per year to operate, depending on scale. The business case is strongest when the AI capability genuinely enables an EA team to handle materially more work — more applications covered, faster review cycles, better data currency.

The key economic question is whether total effort — creation plus validation — is lower than the manual baseline for equivalent quality. Productivity gains should be demonstrated empirically during the PoC, not assumed in the business case.

What the Business Case Should Not Claim

- Do not claim that AI will reduce EA headcount. It will change what EAs spend their time on. Teams that reduce headcount on the basis of AI productivity gains before the system is proven typically find themselves in a worse position.
- Do not claim specific accuracy or reliability figures for AI outputs without empirical evidence from your own data. Vendor claims about AI capability are made against idealised conditions that rarely match production data quality in a large bank.
- Do not project ROI timelines of less than 18-24 months for anything beyond Horizon 1 use cases. The data foundation and integration work takes time, and the learning curve for the EA team is real.

When This Will Not Work

Not all organisations are ready for AI-assisted EA. This programme is likely to fail or destroy value when:

- Data ownership is unresolved: AI will surface conflicts and inconsistencies faster than the organisation can resolve them, creating confusion rather than clarity
- The EA function is immature or under-resourced: AI amplifies EA capability; it does not substitute for it
- Governance buy-in is absent: technology without process change is expensive decoration
- The programme is driven by technology rather than problem: initiatives that start with 'we want to deploy LLMs' rather than 'we want to solve this specific, measurable problem' consistently underperform
- The organisation expects quick wins without data investment: the result will be confident-looking outputs over unreliable data — a worse outcome than the status quo

These are not edge cases. They describe a significant proportion of real-world AI initiatives in large financial institutions. A credible programme plan should include an explicit assessment of each of these conditions before investment is committed.

11. Proof of Concept Design

A Recommended PoC: AI-Assisted Impact Analysis Structuring

The strongest candidate for a first PoC is AI-assisted impact analysis structuring — helping EAs produce more consistent, better-documented, and gap-aware first drafts of impact assessments faster than the current manual process. It is not positioned as automating impact analysis, which would be overclaiming; it is positioned as improving the structure, coverage visibility, and speed of the human-led process.

Weeks 1-2: Baseline and data assessment

Before building anything, assess the quality of the data that will be used. Select three to five representative change scenarios from recent history. Manually reconstruct what the correct impact assessment would have been, based on architect knowledge. This establishes the ground truth against which AI outputs will be measured.

Weeks 3-5: Horizon 1 implementation

Implement the AI drafting assistant with manual context provision. Run the same change scenarios through the AI. Compare outputs against the ground truth. Measure: completeness of dependencies identified, accuracy of risk flags, presence of hallucinated content, coverage gap reporting.

Weeks 6-8: Horizon 2 integration (selective)

Connect the AI to the best-quality available data source (typically the EA repository for a well-maintained LeanIX or Ardoq instance). Re-run the scenarios with autonomous retrieval. Assess whether integration improves or degrades output quality — this is a genuine question, not a rhetorical one; poor data quality may mean Horizon 1 outperforms Horizon 2.

Weeks 9-10: Review and recommendation

Produce a structured assessment of findings, including: actual performance against ground truth, data quality gaps that must be addressed before Horizon 3, recommended use cases for initial production deployment, and governance model for production operation.

Success Criteria

- AI-generated impact assessments identify at least 80% of the dependencies identified by experienced architects in ground truth review
- All AI outputs include source citations and coverage gap reports
- No hallucinated system names or regulatory references in any reviewed output
- EA team assessment: AI-assisted drafts require less time to complete to the same quality standard
- Governance model for production use is documented and approved by risk and compliance

Kill Criteria — When to Stop

A mature programme plan must define in advance the conditions under which the initiative is paused or discontinued:

- Dependency coverage consistently below 60% after Horizon 2 integration — data quality is insufficient for reliable use and requires remediation before proceeding

- Hallucination rate above 5% of specific factual claims in sampled outputs — validation layer is insufficient and the system is not safe for governance use
- EA team adoption below 50% after 3 months of availability — workflow or trust problem requiring process redesign
- Any output used in a formal governance context that is subsequently found to contain a material error — mandatory pause, root cause analysis, and governance review before resumption
- Regulatory or audit challenge to an AI-assisted artefact that cannot be satisfactorily resolved — reassessment of the entire use case's suitability for regulated processes

These criteria are not failures of the programme; they are the programme working as intended — providing evidence-based decision points rather than allowing investment to continue on momentum alone.

Skills Required

From the EA team

- Willingness to critically evaluate AI outputs, including actively searching for errors
- Ability to articulate architecture standards in structured, machine-readable form
- Domain expertise to reconstruct ground truth for PoC evaluation scenarios
- Judgement about when AI coverage gaps require manual investigation

From the implementation team

- AI/LLM engineering: retrieval layer design, prompt engineering, validation layer implementation
- EA domain knowledge: sufficient to understand what correct outputs look like
- Data engineering: API integration with EA repository and document store
- Security engineering: private deployment design, data classification enforcement
- Change management: EA team adoption support

The scarcest combination is EA domain knowledge plus AI engineering skill. This is the primary partnership consideration — external consultants who have one without the other will be limited in their effectiveness.

12. Conclusion: A Credible Path Forward

Enterprise Architecture in large financial institutions has a genuine opportunity in the current AI landscape — but realising it requires resisting the temptation to overclaim, and instead building on a foundation of data quality, governance rigour, and human oversight that matches the standards the discipline already demands of itself.

The shift from static documentation to living architecture is the right direction. It requires AI to be useful, and AI is now capable enough — in constrained, well-governed applications — to contribute meaningfully. But the journey from here to there runs through the unglamorous work of data quality, entity resolution, governance design, and organisational change management. These are not implementation details to be sorted out after the vision is approved. They are the preconditions for the vision being anything other than a demonstration that cannot be operationalised.

The EA discipline has always held itself to a standard of rigour and scepticism that is appropriate in a regulated environment where bad decisions have real consequences. Applying that same standard to AI adoption — neither dismissing the opportunity nor accepting vendor promises at face value — is exactly the posture that will produce durable, defensible results.

The question is not whether AI will be part of Enterprise Architecture. It already is, in early and limited forms, in most large banks. The question is whether it will be implemented with the governance, data quality, and human oversight that make it genuinely trustworthy — or whether it will be implemented quickly, impressively, and in ways that create new risks rather than mitigating existing ones.

This paper was developed as a thought leadership and proposal framework for large financial institutions exploring AI-assisted Enterprise Architecture. It reflects the state of available technology and practice as of April 2026. It has been critically reviewed against the requirements of a regulated European banking environment, with specific attention to DORA, EBA ICT risk guidelines, and the EU AI Act.