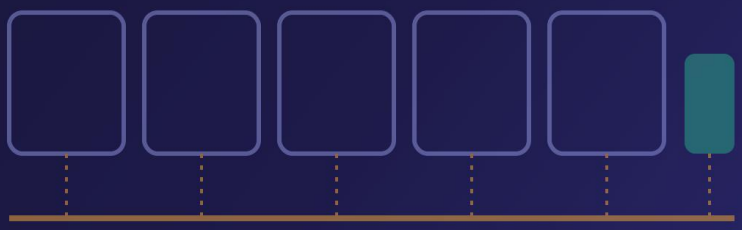
SOVEREIGN DIGITAL RESILIENCE

The Intent Bottleneck

When AI Compresses Delivery, Strategy Becomes the Constraint

An enterprise-architecture operating model for the five stages of work that AI-native delivery does not touch.



intent provenance — the chain that carries a constraint into running code

Sanjeet Kumar
Enterprise Architecture & Technology Governance Leader · Founder, Robotiqa Technologies Ltd.

sanjeetkumar.com/papers · 2026 · Views are the author's own.

SOVEREIGN DIGITAL RESILIENCE · WORKING PAPER

The Intent Bottleneck

When AI Compresses Delivery, Strategy Becomes the Constraint

Sanjeet Kumar — Enterprise Architecture & Technology Governance Leader
Founder & Owner, Robotiqa Technologies Ltd.

Sovereign Digital Resilience · Decision Custody · AI Governance — 2026 · Views are the author's own.

An enterprise-architecture operating model for the five stages of work that AI-native/assisted delivery does not touch — and the provenance chain that carries intent through all of them.

In one page

AI-native delivery compresses the last stage of a six-stage chain that runs from an institution's covenant with those it serves to running software. By Amdahl's law, the total gain is bounded by the five stages left untouched — which is why organizations measure single-digit portfolio improvement against spectacular demos. Worse, cheap construction on an undisciplined intent pipeline produces more wrong things faster. The constraint has moved upstream: to how organizations form, prioritize, and preserve intent. This paper sets out the operating model for that upper chain — including intent provenance, the unbroken line from a stated obligation to the test that proves it in production — and argues that AI-native delivery makes enterprise architecture more load-bearing, not less.

The Compressed Third

Something odd is being reported across enterprises for those who have taken AI seriously in software delivery and COTS implementation. The engineers are not wrong: code arrives faster, tests write themselves, scaffolding that used to consume a fortnight now consumes a coffee-break. And yet, a year in, the portfolio does not look meaningfully different! Initiatives still take the better part of a year to reach funding. The same three programmes are still late. The board asks what the investment bought, and the honest answer is a number small enough to be embarrassing next to the demos.

The instinct is to blame the processes, or the org structure, or the adoption, or the technology team/s. The evidence does not support any of those. A randomized controlled trial by METR is instructive precisely because its result is counter-intuitive: experienced open-source developers working on codebases they knew well took measurably longer to complete tasks when AI tools were allowed — while believing they had been faster. The finding that matters for an architect is not the direction of the effect, which will change as tools mature. It is the gap between *perceived* and *measured* acceleration. Our sense of where time goes is unreliable, and organizations have been reasoning about AI's value using precisely that faulty instrument.

Set the trial aside and grant the optimistic case in full: assume construction really has become several times faster. The disappointment persists. That is the puzzle worth solving, and the solution is not a better tool. It is **arithmetic** about where the time actually sits.

The Chain Nobody Optimizes

Between an organization's reason for existing and a running system there are six stages, and every enterprise traverses all of them whether or not it names them.

- **The Service Covenant** sits at the top — the standing promise an institution makes to those it serves: outcomes in their hands, not services at its counter. Vision and mission live under it as its articulation — the mandate, the obligation, the promise the institution exists to keep.
- **Strategy** turns that into a direction with a horizon long enough to matter, expressed as target capabilities rather than systems.
- **Initiative identification** derives the candidate work that would close the gap between today's capability and the target.
- **Prioritization** ranks that candidate set against parameters and constraints.
- **Project formation** converts surviving initiatives into bounded, fundable, governable units of delivery.
- Only then does **delivery** begin: requirements, design, construction, deployment, operation.

A word on that first stage, because I am deliberately not calling it “vision and mission”. Vision statements describe the institution to itself. A covenant is made outward— it names what those served are owed, and increasingly it is kept by handing them the institution's own capability: self-service where there were counters, direct access where there were intermediaries, standing answers where there were queues. When the top of the chain is framed as a covenant, everything below it inherits a test that a vision statement never imposes— did the promise reach the person it was made to?

Practitioners will recognize the shape. In TOGAF terms the chain runs from Phase A through the architecture development phases into Opportunities and Solutions, Migration Planning, and Implementation Governance. AI-native delivery methodologies — and there are now several good ones — enter at the bottom of that sequence. They are genuinely valuable there. They are also, structurally, addressing the last stage of six.

The stages above are not slow because they are badly run. They are **slow because they are made of a different substance**: they are> sense-making across incomplete evidence, negotiation between people with legitimate competing interests, judgment about consequences that will not be observable for a decade, and the deliberate friction of governance, and this includes hush-hush secretive closed-door meetings; as if only certain top-level people should know this. None of that is a coding problem, which is exactly why no coding tool has moved it.

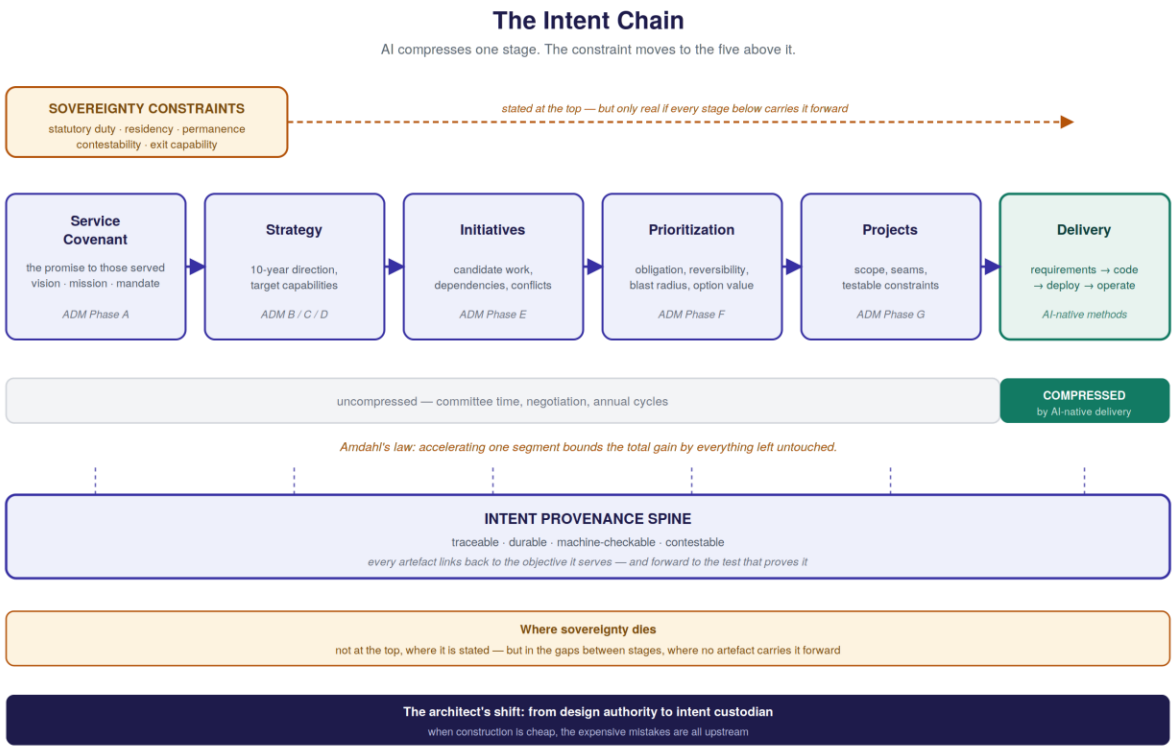


Figure 1 — The Intent Chain. *AI-native/assisted delivery compresses the final stage. The initial five above it remain uncompressed, which restricts the total gain — and sovereignty constraints stated at the top survive only if an unbroken provenance chain carries them down.*

Amdahl's Law for Enterprises

In 1967 Gene Amdahl made an argument about parallel computing that has quietly governed every optimization effort since: the improvement available from speeding up one part of a system is bounded by the fraction of time that part consumes. Make a component infinitely fast and the total gain still stalls against everything you left alone.

Apply it to the chain. Suppose construction accounts for three-tenths of the elapsed time from stated intent to realized value — a generous estimate in most enterprises, where an initiative can wait longer for a funding decision than it later takes to build. Make that stage five times faster and the total improvement cannot exceed roughly a third, in the ideal case, with no coordination cost and no friction anywhere else. In practice, organizations report a fraction of that. The disappointment is not a failure of execution. It is the predictable output of optimizing a minority of the timeline.

There is a second-order effect that concerns me more, because it is not a bounded gain but an unbounded cost. When construction becomes cheap, the *unit* cost of building the wrong thing collapses — and the *aggregate* cost rises, because far more of it gets built. An organization whose intent pipeline is unimproved and whose construction capacity has multiplied will produce more systems, faster, with the same proportion of them poorly conceived. Every one of those arrives with integration surface,

operational burden, security exposure and an exit cost. Cheap construction on an undisciplined chain does not produce a faster enterprise. It produces technical debt at machine speed, and a portfolio that is harder to steer each quarter.

Which inverts the fashionable conclusion. If AI writes the code, the argument goes, architecture matters less. The arithmetic says the opposite: as construction approaches free, the entire cost and risk of the enterprise migrates upstream into deciding what to build, in what order, under what constraints. Architecture is the discipline that owns that ground.

What AI Actually Changes Above the Line

None of this argues that the upper stages are incompressible. It argues that they are compressed by different means — and that the means are available now, largely unused. Four shifts matter.

1. Strategy synthesis at a scale humans cannot read

Strategy formation is bounded by how much **evidence a small group can hold in mind**: the regulatory corpus, incident history, audit findings, customer signals, the actual state of the ecosystem. AI can read all of it and produce a defensible synthesis, complete with the contradictions between sources laid out rather than smoothed over. Therefore, **AI's output is not the strategy**. Instead, it is a far **better-informed starting position** for the executives who must own the strategy, and it arrives in days rather than a planning cycle.

2. Initiative decomposition and dependency discovery

The gap between a strategic objective and a portfolio of initiatives is usually crossed by a handful of experienced people in a room, and the result reflects what those people happened to remember (or at times were influenced by someone 'influential' in the room). Given a machine-readable architecture repository, this becomes a tractable analytical problem: derive candidate initiatives from stated objectives, surface the dependencies between them, and — most valuably — flag the conflicts, where two funded initiatives are quietly making incompatible assumptions about the same capability. Most organizations discover those in year two of delivery; then the blame-game starts.

3. Prioritization as a live model rather than an annual spreadsheet

Not to be confused with project-level work-stream/s prioritization, which can happen on-demand. In contrast, Strategy and funding level 'Prioritization' is typically an annual event producing a static ranking that is wrong within a quarter and defended for three more. When the scoring parameters are explicit and the inputs are current, re-ranking becomes continuous rather than ceremonial, and scenario questions — what happens to the sequence if this obligation lands early, if that vendor exits, if funding contracts by a fifth — can be answered in an afternoon instead of being avoided.

4. Intent as an artefact that travels

This is the one with the longest reach. Historically, intent degrades as it descends: a strategic commitment becomes a programme charter, becomes an epic, becomes a story, becomes code, and somewhere in that chain the reason for the constraint is lost while the constraint itself becomes negotiable. If intent is instead expressed as structured, machine-readable assertions — carried from objective to requirement to acceptance test — the descent stops being lossy. AI makes this practical for the first time, because it can maintain the linkage across thousands of artefacts, which no human librarian ever could.

One boundary applies throughout, and it is the same one I have argued for agentic delivery: AI drafts, synthesizes, and stress-tests; humans decide. A strategy an organization did not argue its way to is not a strategy it will hold to under pressure. The value here is not automated judgment. It is *better-prepared* judgment, arriving faster.

Intent Provenance

Everything above converges on a single artefact that most enterprises do not have: an unbroken, inspectable chain from a stated intent to the thing that implements it, and back again. I will call it intent provenance, and I would argue it deserves the same standing in enterprise architecture that data lineage now has in data management.

I am proposing 4 properties to define it> 1) It should be **traceable**: every requirement, every architectural decision, every acceptance criterion links upward to the objective it serves. 2) **durable**: it survives reorganizations, vendor changes, and the departure of the people who were in the room, because it lives in the repository rather than in institutional memory. 3) **machine-checkable**: not just readable. Constraints are expressed as assertions something can test or cross-reference, not as prose in a document nobody opens. And last) **contestable**: when someone asks in five years why a system behaves this way, the answer exists, with a date and an accountable name attached.

The sovereignty argument depends entirely on this. Consider how sovereignty commitments actually fail. They are rarely repudiated. They are stated with conviction at the vision layer — records under our jurisdiction, decisions we can explain, no dependency we cannot exit — and then they simply thin out on the way down. By the time a sprint team is choosing between a managed service that solves this week's problem and a slower option that preserves an obligation nobody has restated in eighteen months, the constraint has no representation in the room. Nobody defects. The chain just goes quiet.

That is the sentence I would leave with an architecture practice: **sovereignty is not a property of a data centre, and it is not a paragraph in a policy document. It is a property of the chain that carries a constraint from intention into running code**. Where the chain breaks, sovereignty is an aspiration with a hosting invoice attached. This is also why the coming shifts — from data residency toward control over inference, and from inference toward the decisions systems make about people — will punish organizations that never built the chain. You can repatriate data. You cannot repatriate a decision you are unable to reproduce or explain.

Prioritization Is an Architecture Function

If the chain has a single weakest link, it is prioritization — and it is 'weak' for a specific reason. Most organizations rank initiatives on a business case: cost, benefit, payback, board direction. Every parameter in that model is denominated in money over a short horizon, and the properties that determine whether an institution can still keep its promises in twenty years are denominated in nothing at all.

This is precisely how vendor lock-in enters a portfolio through the front door, unopposed. The option with the lowest three-year cost is frequently the option with the highest exit cost, and no line in a standard business case is where that fact would appear. The commitment is made on a spreadsheet that is structurally incapable of representing the risk. A decade later it is discussed as an unfortunate legacy situation, as though it had happened to the organization rather than been chosen by it.

An architecture-owned prioritization model scores the candidate set against parameters the business case cannot express: Consider this>

Parameter	The question it asks	Why architecture owns it
Statutory obligation	Is this work required by law, mandate, or duty of custody — as opposed to merely desirable?	Non-negotiable floor; cannot be traded against ROI
Sovereignty exposure	Does the initiative move records, inference, or decisions outside jurisdictional or organizational control?	Determines residency, routing, and hosting constraints
Reversibility	What would it cost to undo this in five years — commercially, technically, contractually?	Surfaces lock-in that no business case line item captures
Dependency depth	How many other initiatives, systems, or obligations rest on this one succeeding?	Sequencing; a deep dependency starved of funding stalls the portfolio
Blast radius	If this fails or is compromised, what is the scope of harm to the record and to the public?	Sets assurance level and governance intensity
Option value	Does this open future choices, or does it foreclose them?	Rewards enabling work that ROI models systematically undervalue
Capability uplift	Does the organization end this work more able to do the next thing itself?	Guards against outsourcing the capability along with the project

None of these replaces financial analysis; they sit alongside it, and they are scored in the open, in the repository, with the rationale attached. The discipline is not the arithmetic — any weighting scheme will do — it is the *explicitness*. A ranking whose reasoning is recorded can be revisited when circumstances change. A ranking that lives in a conversation cannot be revisited at all, only re-argued.

A Worked Example: A Permanence Requirement, End to End

The following case is composite. It draws on patterns from more than one engagement in public-record and long-horizon custody environments, and it is deliberately stated at the level of reasoning rather than outcome: it illustrates how the chain is traversed, not what any particular programme delivered. That is

the honest form for a paper like this, and it is also the useful one, since the reasoning transfers and the outcomes would not.

Covenant. The institution's mandate is that certain records remain authoritative indefinitely. Not retained — authoritative: capable of being relied upon as the definitive answer, in a dispute, by parties who were not born when the record was created. This is an obligation to those served, not an aspiration, and it does not expire with a technology cycle.

Strategy. Translated into an architectural objective with a horizon measured in generations: no record's survival, integrity, or intelligibility may depend on the continued existence of any single vendor, product, format, or contract. Note what this does. It converts a mandate into a testable structural property, which is the only form in which it can survive the journey downward.

Initiative identification. Several candidates present themselves: a digital preservation capability; format normalization and migration; custody and location control for physical holdings that remain the legal originals; and an integration and metadata backbone connecting them. Each is coherent on its own. None alone satisfies the objective.

Prioritization. Here the parameters do their work, and they invert the intuitive ranking. A conventional business case favours the digital preservation platform — visible, vendor-supported, demonstrably modern — and defers physical custody as unglamorous back-office work. The obligation lens says otherwise: a record whose physical original cannot be located and evidenced is not authoritative, however immaculate its digital surrogate. The weakest link in the custody chain, not the most impressive component, determines whether the mandate is met. Reversibility applies a second correction: the single-vendor answer that covers the whole scope scores worst on exit cost precisely because it covers the whole scope. Concentration is comfortable and expensive to undo.

Project formation. Rather than one programme with an internal seam nobody owns, the work is formed as coupled projects with an explicit integration boundary that the institution itself owns and specifies — the same principle I have argued elsewhere for AI gateways: own the seam, rent the components. Each component can then be replaced on its own timeline without renegotiating the obligation. The seam is where the permanence requirement is architecturally enforced, so the seam is not a delivery detail. It is the design.

Delivery. The constraint finally arrives as acceptance criteria that something can fail: export completeness and fidelity verified against a defined schema; format migration demonstrated, not promised; a restoration drill executed on a schedule with results recorded; the custody chain auditable end to end across digital and physical holdings. Most consequentially, exit capability is tested as a delivery acceptance criterion rather than asserted as a contract clause — because a clause is a claim about the future, and a passing test is evidence about the present.

Trace the thread backwards from that last test and it reaches the covenant without a break. That unbroken line is the artefact. It is what makes the difference between an institution that has stated an obligation and one that can demonstrate it is keeping it.

The Architect as Intent Custodian

This changes what the role is for. The traditional centre of gravity of enterprise architecture is design authority: deciding how things will be built, and reviewing whether they were. That work does not disappear, but AI erodes its scarcity value, since generating a competent design is no longer the constraint it was.

What becomes scarce is 'custody of intent' — responsibility for ensuring that what the institution committed to at the top is still true at the bottom, years later, after the people, vendors and platforms have all turned over. That is not a documentation task. It is an operating responsibility with three practical consequences for a practice.

The repository becomes an operational system, not a library. If intent provenance is to be machine-checkable, the repository has to be queryable, current, and wired into how work is actually planned and delivered. A repository that is accurate only at audit time cannot carry a constraint.

Architecture decision records point upward, not just sideways. Most ADRs record which option was chosen and why, against alternatives. The missing link is the one to the objective the decision serves — without it, a future team can see what was decided but not what would have to change for the decision to be revisited.

The deliverable shifts from diagram to constraint. A diagram describes a state that will be obsolete within the year. A constraint expressed as a test travels with the system and keeps arguing on the architect's behalf long after the architect has moved on — which is, in the end, what institutional memory is for.

When Building Is Cheap, Choosing Is Everything

The organizations that will look sharply different in 10 years are not the ones that adopted AI-native delivery earliest. That will be table stakes, and quickly. They are the ones that noticed the constraint had moved — that the expensive mistakes are now made upstream, in what gets chosen, in what order, under which constraints, and whether those constraints survive the journey to production.

AI has made construction cheap. It has not made judgment cheap, and it has made the consequences of poor judgment arrive considerably faster. In institutions whose obligations outlast their technologies, that is the whole game: the chain from intent to implementation, held intact, on purpose, by someone whose job it is.

There is a successor question, and it is where this series goes next. States have begun to speak of decision sovereignty — a government's control over the AI-supported decisions taken in its name — and the emerging literature treats it, rightly, as a matter of national capability. Institutions need the operational counterpart, and I will give it the name it deserves> ***decision custody*** — an institution's demonstrated ability to reproduce, explain, defend, and answer for every consequential decision made in its name, human or machine, for as long as the consequences of that decision last. Custody is the standard registries have applied to records for centuries; the decade ahead will demand it of decisions. Data can be repatriated. A decision that cannot be reproduced cannot — and intent provenance, as set out here, is the chain of custody that makes it reproducible. The covenant is the promise; custody is the proof. That is the subject of the next paper.

Five Questions for a Board

For directors who read this far and want somewhere to put it, five questions convert the argument into oversight:

1. What fraction of our vision-to-value timeline does AI actually touch — and what does that bound our expected return to?

2. If our delivery capacity doubled tomorrow, would we build better things — or the wrong things twice as fast?

3. Which of our strategic commitments — sovereignty, permanence, exit — exist today as testable constraints in delivery, rather than as paragraphs in a strategy document?

4. For our most consequential automated or AI-assisted decisions, could we reproduce and explain any single one of them three years after it was made?

last) Who, by name, is custodian of the chain from stated intent to running system — and would they know it?

References

1. Amdahl, G. M., “Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities”, AFIPS Conference Proceedings, 1967.
2. Conway, M. E., “How Do Committees Invent?”, Datamation, April 1968.
3. Becker, J., Rush, N., Barnes, B., Rein, D., Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, Model Evaluation & Threat Research (METR), 2025.
4. The Open Group, The TOGAF® Standard, 10th Edition, 2022.
5. Ross, J. W., Weill, P., Robertson, D. C., Enterprise Architecture as Strategy: Creating a Foundation for Business Execution, Harvard Business School Press, 2006.
6. Forsgren, N., Humble, J., Kim, G., Accelerate: The Science of Lean Software and DevOps, IT Revolution Press, 2018.
7. International Organization for Standardization, ISO/IEC 42001:2023 — Information technology — Artificial intelligence — Management system, 2023.

8. National Institute of Standards and Technology, Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1, 2023.
9. Raja, S. P., “AI-Driven Development Life Cycle: Reimagining Software Engineering”, AWS DevOps & Developer Productivity Blog, 31 July 2025, aws.amazon.com/blogs/devops/ai-driven-development-life-cycle.
10. Wei, P., Shu, W., Preserving Decision Sovereignty in Military AI: A Trade-Secret-Safe Architectural Framework for Model Replaceability, Human Authority, and State Control, arXiv:2604.20867, 2026.
11. Kumar, S., The AI Gateway Advantage: Value over Spend. Smart Work over Hard Work., sanjeetkumar.com/papers/ai-gateway-advantage, 2026.
12. Kumar, S., Governing Agentic AI: In Systems of Public Record, Governance Is the Accelerant, sanjeetkumar.com/papers/governing-agentic-ai, 2026.
13. Kumar, S., Century-Scale Systems: What Land Registries Teach Us About Data Sovereignty, sanjeetkumar.com/papers/century-scale-systems, 2026.
14. Kumar, S., Exit-Capable by Design: Vendor Concentration Is a Sovereignty Risk, sanjeetkumar.com/papers/exit-capable-by-design, 2026.

AI-assistance disclosure> *This paper was drafted and refined with AI assistance. The arguments, positions, judgements, and final text are the author's own.*