The Shrinking AI Surface

Why well-designed processes minimize the role of AI

Vanni Torelli, Gluon Consulting Pte. Ltd.

April 2026 (1.0)

Contents

Executive summary

The prevailing narrative in enterprise technology is that artificial intelligence should be placed at the centre of every business process. Hundreds of billions of dollars are flowing into GPU clusters, foundation models, and AI transformation programmes built on the premise that organisations need ever-larger models to tame ever-growing chaos.

This paper argues the opposite. Drawing on historical precedent, first-principles analysis, and direct implementation experience in electrical utilities operations, we propose that the highest-return investment is often not in more sophisticated AI, but in better-designed processes that eliminate the need for AI in the first place.

The AI value is at the margins where the real world resists structure.

When processes are well-designed, inputs are captured cleanly at source, and exceptions are handled collaboratively and in a timely fashion, the surface area where AI is genuinely irreplaceable shrinks to a thin boundary layer. The intelligence required to manage a system drops as the system matures. In many cases, a deterministic rules engine outperforms a neural network, not because the neural network is inadequate, but because the problem no longer warrants it.

This is not an argument against AI. It is an argument for deploying AI where it earns its complexity budget, and for recognising that the most successful digital transformations are those that minimise, rather than maximise, the role of non-deterministic computation.

The Roman lesson

In 443 BC, the Roman Republic established the office of the censor. Over the following centuries, the *censores* became responsible for one of the ancient world's most ambitious data-collection exercises: the census of the Roman people.

The Romans did not build a central bureau and wait for information to arrive. They understood, with remarkable clarity, that data quality is a collection problem, not a processing problem. Trained officials, the *censores* and their staff, were dispatched to every corner of the Republic, and later the Empire, to capture information at source: household composition, property holdings, citizenship status, military eligibility. The methodology was standardised. The collectors were domain experts who understood both the legal framework and the local context.

The result was a dataset of extraordinary reliability by pre-modern standards, one that underpinned Roman taxation, military conscription, and public works for centuries. The intelligence was embedded in the collection process itself, not in some downstream reconciliation apparatus.

Contrast this with the implicit model behind most modern enterprise AI deployments: allow messy, unstructured, inconsistent data to accumulate, then apply increasingly powerful models to extract meaning after the fact. This is the computational equivalent of letting illiterate scribes record the census on whatever medium they find, then hiring a team of scholars to decipher the results.

The Roman approach embodies a principle that modern software architecture often forgets: the cheapest data to clean is data that was never dirty.

The complexity inversion

Artificial intelligence, particularly large language models and neural vision systems, earns its complexity budget by handling ambiguity, variability, and noise. These systems excel precisely where inputs are non-deterministic: handwritten documents with inconsistent layouts, natural language with contextual meaning, images with variable quality and framing.

Yet, this strength implies a corollary that receives almost no airtime in the current hype cycle.

If you systematically remove noise through institutional reform, standardisation, and structured data mandates, you collapse the problem space back into territory where deterministic systems are not just adequate but superior.

Rules engines, state machines, and simple validators are faster, cheaper, fully auditable, explainable. And, crucially, they do not hallucinate. When a process is deterministic and its inputs are structured, AI is not merely unnecessary: it introduces risk. A neural model that can handle edge cases brilliantly will also, occasionally, fabricate confidence where none is warranted. A rules engine, operating on clean structured data, will never do this.

We call this *the complexity inversion*: the more successful your digital transformation is, the less AI you need.

The table below is not an argument that rules engines are universally better. It is an argument that the choice between approaches should be driven by the nature of the input data, not by the assumption that AI is always the more advanced option.

Attribute	AI/Neural approach	Rules engine approach
Best suited for	Ambiguous, noisy, unstructured inputs	Deterministic, structured, well-defined inputs
Auditability	Low: probabilistic, black-box reasoning	Full: every decision traceable to a rule
Hallucination risk	Non-zero, increases with edge cases	Zero: fails explicitly or succeeds
Infrastructure cost	GPU clusters, model serving, monitoring	Standard compute; negligible marginal cost
Latency	Milliseconds to seconds per inference	Sub-millisecond
Maintenance	Model drift, retraining, prompt tuning	Rule updates, fully version-controlled

The forgotten middle

The contemporary debate tends to present a binary choice: either you use AI (neural, probabilistic, GPU-intensive) or you use traditional programming (if-then rules, hardcoded logic). This framing omits an entire class of technologies that occupied the centre of artificial intelligence research for decades before the deep learning revolution: formal knowledge representation and ontological reasoning.

From frames to OWL

In 1974, Marvin Minsky published *A Framework for Representing Knowledge*, introducing frame-based knowledge representation. Frames organise knowledge into structured units: slots, fillers, defaults, constraints, and inheritance hierarchies. These units model a domain the way a human expert thinks about it. A frame for electrical inspection might have slots for property type, wiring standard, load capacity, and compliance status, with inheritance rules that propagate regulatory constraints from parent categories to specific instances.

This lineage evolved through KL-ONE in the 1980s, description logics in the 1990s, and culminated in the W3C's OWL (Web Ontology Language) standard for the Semantic Web. At each stage, the core proposition remained the same: if you can formalise a domain's concepts, relationships, and constraints into a machine-readable ontology, you can perform deterministic inference over it. This means classifying new instances, checking consistency, deriving implied facts, without any probabilistic computation at all.

The classifier, the automated reasoning engine at the heart of these systems, does not guess, it deduces. Given a well-formed ontology and a set of assertions, it will derive every valid conclusion and flag every inconsistency, with complete traceability. No hallucination, no confidence scores, no GPU required.

Why ontologies were eclipsed

If ontological reasoning is so powerful, why did the industry abandon it in favour of neural approaches? The answer is straightforward: building ontologies is hard. Formalising a domain requires deep expertise, iterative refinement, and a level of upfront intellectual investment that most organisations were unwilling to make. The promise of deep learning was that you could skip the formalisation step entirely: feed raw data into a sufficiently large model and let it learn the structure implicitly.

This worked spectacularly for domains with abundant unstructured data and tolerance for probabilistic outputs: image recognition, natural language generation, recommendation systems. It has, however, proven far less satisfactory in domains that demand auditability, regulatory compliance, and deterministic correctness. These are precisely the domains where most enterprise workflows operate.

The convergence: clean data meets formal reasoning

As we said before, well-designed processes produce clean, structured data at source. Once data is clean and the domain is well-defined, the precondition for ontological reasoning is met. When the data collection process itself enforces structure, the expensive upfront investment in formalisation, the very thing that made ontologies impractical when data was messy, becomes viable.

Ontological reasoning is not a replacement for AI. It is what AI was supposed to become, once the domain matures enough to support it.

Consider the progression in a domain like electrical utility service applications. In the early stages, when Electrical Practitioners submit handwritten notes and degraded photographs, you need neural approaches: OCR, vision models,

language understanding. The data is noisy, the domain encoding is implicit, and probabilistic inference is the only viable strategy.

As the platform matures, guided forms replace handwritten notes, structured data replaces free text, and the domain rules become explicit in the workflow itself. At that point, the reasoning required is no longer probabilistic, it is classificatory and deductive. An ontology-based reasoner can determine whether an inspection meets compliance requirements, whether a given wiring configuration falls within regulatory parameters, and whether an estimate is consistent with the catalogued components, all deterministically, all traceably.

The maturity spectrum

We can now articulate a maturity spectrum for intelligent systems in enterprise workflows.

Stage	Data character	Appropriate technology	Reasoning mode
1. Nascent	Unstructured, noisy, inconsistent	Neural AI (LLMs, vision models, OCR)	Probabilistic inference
2. Structured	Guided capture, mostly clean, known schema	Ontological reasoners, knowledge graphs	Deductive classification
3. Mature	Fully structured, standards-compliant	Rules engines, state machines, validators	Deterministic execution

Most enterprise domains exist somewhere along this spectrum, and different sub-processes within the same domain may sit at different stages. The critical insight is that the appropriate technology should match the maturity of the data, not the ambition of the technology vendor. Deploying a large language model where an ontological reasoner would suffice is not just wasteful — it introduces unnecessary risk. Deploying an ontological reasoner where the data is still noisy is equally misguided.

Neurosymbolic architectures: the emerging synthesis

The research community is converging on this insight from the opposite direction. The emerging field of *neurosymbolic AI* seeks to combine neural and symbolic approaches: using LLMs for the messy, unstructured boundaries while grounding their outputs in formal ontological structures that enforce consistency and enable deterministic reasoning.

Palantir's Ontology architecture, which surfaces deterministic logic assets as tools for LLM-based agents, is one commercial instantiation. Academic work on ontology-constrained neural reasoning demonstrates that formal semantic grounding can dramatically reduce hallucination and domain drift in enterprise AI agents.

Yonyou's Large Ontology Model

A striking recent example comes from Yonyou AI Lab, one of China's largest enterprise software providers. In April 2026, Yonyou released what it calls the Large Ontology Model (LOM), a system built on a *Construct-Align-Reason* architecture that autonomously builds structured business ontologies from raw enterprise data and then performs deterministic reasoning on top of them.

The motivation is telling. Yonyou's own experience deploying large language models in enterprise scenarios revealed a persistent pattern: probabilistic token prediction, no matter how many parameters were added, produced unstable reasoning, inconsistent outputs, and a fundamental disconnect from actual business logic. Their response was not to build a bigger LLM. It was to build a system that constructs a formal ontological representation of the business domain, what they call a "business logic universe", and then reasons within that structure deterministically.

The results are instructive. On their own benchmark of 19 graph reasoning tasks (Zhang & Zhu, arXiv:2602.00029v1), their 4-billion-parameter ontology model achieved 89.47% overall accuracy, outperforming DeepSeek-V3.2 and other

mainstream large language models with orders of magnitude more parameters. The performance advantage was most pronounced on complex algorithmic reasoning: PageRank, minimum spanning tree, topological sorting, where general-purpose LLMs scored near zero. The advantage comes not from scale but from structure: once the domain is formalised into a coherent ontology, reasoning becomes a matter of traversal and deduction rather than probabilistic generation.

This validates our central thesis from an unexpected direction. A major enterprise software company, after years of deploying neural AI in production, concluded independently that the path to reliable enterprise reasoning runs through ontological formalisation, not through larger models. The precondition, as always, is clean, well-structured data, which is why Yonyou's system begins with an ontology construction phase that converts fragmented business information into standardised, machine-interpretable form.

However, this synthesis only works if the ontology exists. Building the ontology requires the same disciplined process design and structured data capture that this paper has been advocating throughout. The circle closes: good process design enables ontological formalisation, which enables deterministic reasoning, which reduces the need for probabilistic AI.

The AI surface shrinks not because AI has failed, but because the domain has matured past the point where probabilistic computation is the right tool.

Modular ontologies: engineering for bounded access

A common objection to ontology-based approaches in enterprise settings is the apparent requirement to model an entire organisation's operations before any value can be delivered. In practice, no external consultancy or platform vendor has access to the entirety of a client's business processes. After years of deep engagement, a technology partner may understand perhaps a quarter of a large enterprise's operational landscape. The rest remains behind organisational, political, and contractual boundaries.

This is not an obstacle. It is a design constraint that modular ontologies address directly.

Bounded domains with formal interfaces

A modular ontology is a self-contained formal model of a bounded domain, with explicitly specified interfaces to adjacent domains that may not yet be modelled. Each module is independently valid: it defines its own entities, relationships, constraints, and inference rules, and it can support deterministic reasoning within its scope without requiring knowledge of the broader enterprise.

The interfaces are the critical architectural element. Where a bounded domain produces or consumes data from an adjacent domain, the interface formally specifies the boundary entities, their expected types, and the contractual assertions that must hold. When the adjacent domain is eventually modelled in a future engagement, by a different team, or as part of a platform integration, the interface is already defined. Integration becomes a matter of connecting two formally specified boundaries rather than reverse-engineering implicit dependencies.

This mirrors well-established patterns in software architecture. Bounded contexts in domain-driven design, interface contracts in microservices, API specifications in distributed systems. They all operate on the same principle: model what you control rigorously and specify your boundaries precisely. Modular ontologies apply this principle to knowledge representation.

Leveraging vendor platform documentation

For enterprises built on established platforms - Oracle CC&B for utility billing, SAP IS-U for energy distribution, Salesforce for customer management - a significant shortcut exists. These platforms ship with rigorously documented data models: entity relationship diagrams, data dictionaries with field-level descriptions, documented business object hierarchies with inheritance, and specified state transition rules. The platform's schema is, in effect, the *de facto* ontology of the client's operations within that domain.

This documentation transforms ontology construction from a discovery problem into a translation problem. The entities are named and described. The relationships are explicit, including cardinality. The constraints are documented. Converting a well-documented vendor data model into a formal ontological representation is a structured engineering exercise, not a research project.

The commercial implication is significant. A reference ontology for a major industry platform, built once from vendor documentation, becomes a reusable asset across every client running that platform. The core entity model is common; only the client-specific configurations require adaptation. This inverts the typical engagement dynamic: instead of asking the client to explain their business processes, the consultancy arrives with a formal understanding of their platform's domain model and works outward from there.

Incremental coverage

Modular ontologies grow organically. A first engagement produces a formal model of one workflow domain. A second engagement extends coverage to an adjacent module. Each module deepens the reference ontology, and every subsequent client operating in the same industry benefits from the accumulated model.

Over time, a consultancy practising this discipline accumulates a library of interconnectable domain modules: not a monolithic enterprise model, but a composable set of formally specified building blocks. The value compounds with each engagement, and the barrier to entry for competitors who lack this accumulated knowledge rises correspondingly.

AGI and the moving target

Much of the current discourse around Artificial General Intelligence implicitly benchmarks against a frozen snapshot of today's work landscape. Forecasts ask: when will AI be able to do what a human accountant, lawyer, or software engineer does today?

This narrative contains a hidden assumption: that the nature of work is static. It is not.

Consider financial accounting. Today, a significant portion of an accountant's cognitive effort goes into reconciling data across inconsistent formats, interpreting ambiguous entries, and producing structured reports from unstructured inputs. This is precisely the kind of task that AI handles well.

If regulatory convergence mandates machine-readable financial data, a trend already underway with XBRL and the EU's European Single Electronic Format, then the task itself changes. The "hard" part evaporates, not because AI solved it, but because the institutional context eliminated it.

The same pattern applies across domains. When governments mandate structured digital submission of forms, when industries adopt common data schemas, when interoperability standards mature, the cognitive burden that justified AI in the first place diminishes. The target moves while the arrow is in flight.

This has a counterintuitive implication for AGI timelines: a significant portion of what gets cited as "AGI-required tasks" are really symptoms of institutional and data infrastructure immaturity. We do not need a superintelligence to reconcile financial statements across companies: we need a common schema and an API. The intelligence required drops as the ecosystem matures.

A case study

Gluon's MobileAP platform, built for Aboitiz Power's distribution utility franchises, provides a concrete illustration of these principles in action.

MobileAP manages the end-to-end workflow for new electrical service applications: from customer data capture through inspection and pre-requisites installation, initial deposit payment, service agreement signing, meter installation, and electrification.

Four categories of actors – Customer, the accredited Electrical Practitioner (EP), and Aboitiz Power's own desk-based and field-based staff – participate in a collaborative workflow through mobile and desktop applications.

The design principle

Rather than allowing unstructured submissions and applying AI downstream, MobileAP embeds domain expertise into the collection instrument. A step-by-step guide, based on highly dynamic forms, guides the Customer through capturing the exact type and amount of data required to proceed with their electrical service application.

Any data asset that is too technical for the Customer to be familiar with, is captured by the EP, usually an electrical professional or a service company accredited with Aboitiz Power. Using their dedicated mobile app, the EP, too, is guided trough structured data capture: standardised inspection forms, validated photo submissions, itemised estimates based on a standard taxonomy.

The same "capture and act" approach is extended at every step that involves one actor interacting with another: Aboitiz Power's electrical inspectors and installers also use dedicated mobile apps to capture their slices of data domain, in a way that leverages their domain expertise while minimizing inconsistencies and errors.

The EP, inspector or installers are, in effect, the modern equivalent of the Roman censor: domain experts collecting structured data at the point of origin, aided by as much high-tech tools are necessary.

Real-time exception handling

In most enterprise systems, exceptions are discovered asynchronously: someone downstream identifies a problem, files a ticket, the issue returns upstream, and by then the context has evaporated. Reconstructing what went wrong often requires intelligent systems simply to reassemble the relevant information.

MobileAP collapses this loop. All actors share a workflow state. Exceptions surface immediately, with full context, and resolution happens while every participant still knows what they are looking at. Asynchronous steps with immediate notification and priority queues ensure that time-sensitive validations are processed before the window of physical presence closes, or as early as possible after it does.

Where AI adds genuine value

The core workflow is a deterministic state machine: application submitted, inspection scheduled, estimate produced, work approved, completion verified. AI does not power this engine, and it should not. The state machine is faster, cheaper, fully auditable, and incapable of hallucination.

AI enters the architecture only at the boundaries where the real world injects irreducible noise:

- handwritten inspection notes from senior EPs who resist digital forms
- degraded document scans from older properties with non-standard records
- ambiguous photographic evidence requiring visual interpretation
- natural language interfaces for customer-facing interactions.

These are legitimate use cases, and they will persist for years, particularly in emerging markets where institutional infrastructure is still maturing. Yet, they are

edge cases, not the main flow. The platform is designed so that the AI surface area is as small as possible and may shrink further as the ecosystem standardises.

The prevention principle

There is a saying in medicine: prevention is better than cure. The same principle applies to data architecture.

The prevailing enterprise AI model is curative. It assumes dirty data as a given and applies increasingly powerful models to clean it after the fact. This approach has three structural weaknesses:

1. Cost escalation. Every additional layer of AI inference adds infrastructure cost, operational complexity, and failure modes. GPU clusters, model serving pipelines, monitoring for drift and hallucination. These are not free.
2. Context loss. The further downstream a data quality issue is detected, the more expensive it is to resolve. If an EP has left the Customer's premises before a photo quality issue is flagged, a return visit may be required, multiplying costs.
3. Accountability gaps. When an AI model mediates between raw input and structured output, the chain of accountability becomes opaque. Who is responsible when the model misinterprets a handwritten entry? The collector, the model operator, or the downstream consumer of the data?

The preventive model invests intelligence at the point of capture. Guided forms, real-time validation, immediate feedback loops, and domain-expert collectors ensure that data arrives clean. The downstream system can then be simple, deterministic, and reliable.

This does not eliminate the need for AI. It constrains AI to where it is genuinely irreplaceable: the boundary between structured process and unstructured reality. A well-designed form eliminates more busywork than a billion-parameter model.

The device constraint

A theoretically elegant argument for edge AI, running lightweight models on the mobile device at the point of data capture, must contend with practical reality. In emerging markets like the Philippines, the average field worker is not carrying a flagship smartphone with a neural processing unit capable of running even a small language model. Mid-range Android devices with limited RAM, storage, and intermittent connectivity are the norm.

This constraint reinforces the architectural point: prevention is best achieved through process design, not through on-device compute. Structured forms, guided workflows, and server-side validation with sub-second response times and push notifications achieve the same outcome, catching problems before the field worker leaves the site, without requiring anything from the device beyond a camera, GPS, and a data connection.

Server-side validation has additional advantages: it is centrally updatable, monitorable, and can improve without touching the client application. An on-device model is frozen at whatever version shipped with the last APK update and constrained by hardware it cannot outgrow.

Some (honest) drawbacks

Intellectual honesty requires acknowledging the limits of this thesis.

Institutional reform is slow	Mandating structured data formats across an entire industry or regulatory jurisdiction takes years or decades. In the interim, AI remains the pragmatic solution for dealing with messy inputs at scale. The curative model is not wrong: it is expedient.
Some domains are irreducibly noisy	Medical imaging, natural language understanding, creative tasks, and scientific discovery involve genuine ambiguity that no amount of process design can eliminate. For these domains, AI is not at the margins, it is the core.
Process design requires upfront investment	Building collaborative workflows, guided forms, and real-time validation infrastructure is expensive. Many organisations find it cheaper in the short term to bolt an AI layer onto existing broken processes than to redesign those processes from scratch.
The argument applies unevenly across markets	In mature markets with established digital infrastructure, the transition to structured data may happen faster. In emerging markets, inconsistent institutional frameworks and legacy paper-based processes will sustain the need for AI-powered data extraction for years to come.

These are real limitations. The thesis of this paper is not that AI is unnecessary, but that its optimal role is narrower than the current industry narrative suggests, and that the highest-return investment is often in the process itself.

A decision framework

For technology leaders evaluating where to allocate resources, we propose a simple heuristic.

If your primary problem is...	Then invest in...
Messy, unstructured inputs from sources you do not control	AI: OCR, NLP, vision models for extraction and classification
Messy inputs from sources you do control (employees, partners, field workers)	Process design: guided forms, validation at source, real-time exception handling
Reconciling data across inconsistent external systems	Standards adoption and integration architecture, with AI as an interim bridge
Genuine ambiguity requiring judgment (medical, legal, creative)	AI as core capability, with human oversight for high-stakes decisions
Deterministic processes currently handled manually	Rules engines and state machines: AI is overkill
Well-defined domain with structured data, requiring auditable inference (compliance, classification, eligibility)	Modular ontologies with deterministic reasoning; no GPU required

The key question is not “can AI do this?” but “is AI the right tool for this?” In many cases, the answer is that a well-designed process, a structured form, or a deterministic engine will do the job more reliably, at lower cost, with full auditability.

Conclusions

The Roman *censores* understood something that the modern AI industry has largely forgotten: the most reliable data system is one where intelligence is embedded in the collection process, not bolted on after the fact.

As organisations invest in digital transformation, the temptation is to place AI at the centre of every workflow, handling every input, mediating every decision. This approach is expensive, opaque, and fragile. It conflates capability with necessity.

The alternative is to design processes that produce clean data by construction, handle exceptions collaboratively in real time, and reserve AI for the genuinely irreducible boundary between structured systems and unstructured reality. This results in platforms that are cheaper to operate, easier to audit, more robust to failure, and, paradoxically, better positioned to integrate AI where it genuinely matters.

The goal is not to build the most intelligent system. The goal is to build the system that needs the least intelligence.

The shrinking AI surface is not a failure of ambition. It is a sign of architectural maturity.