WORKING PAPER

# Proliferation of Machine Intelligence

## Business into Hyper-Learning Machines

*From systems of record to systems of inference — and the assurance problem that follows.*

**Krishnakumar G**

**System Engineer @ Volvo.  Member @ IASEAI, IPA**

# About me

*Fifteen years, four industries*

01

### eCommerce

Supply Chain

02

### Healthcare

Digital Health & Smarter Care

03

### Telecom

Business & Operational System

04

### Automotive

Vehicle Software Supply Chain

**Background:** B.E. in Digital Communication & Technology · M.Sc. in Applied Neuroscience

# The gap is distributional, not technical

*Abstract — the condition the argument begins from*

Intelligent systems now span a continuum of capability and availability. Business models rooted in human expertise, labour and skill are being revalued against them.

The gap this opens is not a tooling deficit. It emerges faster than the institutions responsible for absorbing it can adapt — which makes it a socio-economic problem wearing technical clothing.

**Discretize**

Separate the indivisible units of the functional space from the variables of the continuum.

**Then select**

Technique follows decomposition — never the reverse. Dimensionality decides the tool.

**Then enable**

Process redesign needs organizational design; that needs a socio-economic settlement.

# Thesis argument in four movements

*Each movement is a precondition for the one after it*

I

## Discretization

A domain can only be governed once it has been carved into nameable functions with boundaries, inputs, outputs and ownership.

II

## Regime change

Records move from explicit memory to ontologies and representational memory; enumerated rules give way to generalized inference.

III

## Labour

The transcription function is displaced. The scarce competence shifts from producing representations to governing them.

IV

## Architecture

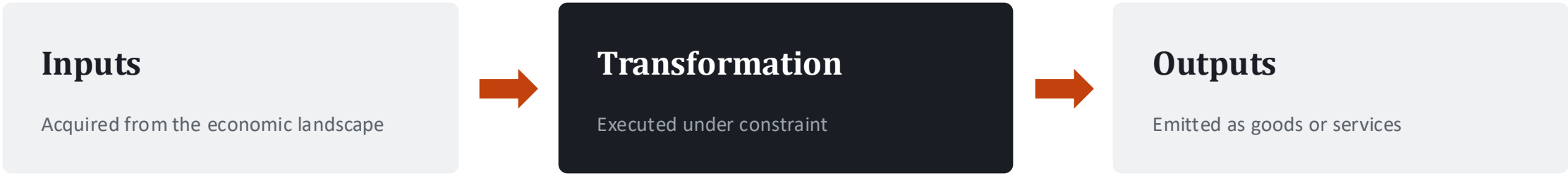
Enterprise architecture is superseded by enterprise systems engineering, on an epistemic rather than a framework basis.

# Discretization precedes institutionalization

*To discretize is to individuate — a regime can only govern what has first been carved out*

Until a continuous field of organizational activity is decomposed into discrete, nameable functions, the domain cannot be measured, delegated, audited or automated. Institutionalization is not imposed from outside; it is the consequence of having carved the domain into units that admit rules.

THE INVARIANT — every enterprise, irrespective of sector




**Profitability** is a property of transformation efficiency relative to the value the market assigns the output — not of the structure itself. That invariance is what makes cross-domain formalization possible: a retail supply chain, a clinical pathway and a vehicle software release train share one functional skeleton, differing only in the semantics of their artifacts and the tolerances of their constraints.

# Functional domain discretization as geometry

*The domain as a multi-dimensional metric space, where a distance function makes proximity measurable rather than metaphorical*

### Entry points

Locations at which inputs are organized and admitted.

*Structure what enters a functional space or subspace.*

### Intervention points

Locations at which multiple functional spaces or subspaces intersect.

*Enable emergent phenomena arising from the intersection.*

### Extraction points

Locations at which measurable output or payout is realized.

*Draw value from the functional space across space and time.*

## Hierarchy of leverage points

Following Meadows: intervention loci ranked by the magnitude of systemic change each can produce, from parameter adjustment at the weakest through to the paradigm from which the system's goals arise. Control is exercised through hierarchical abstraction.

## Adaptiveness with stability

Geometric signature and structural decomposability must evolve together: enough plasticity to reconfigure under environmental change, enough structural integrity to avoid dissolution under that reconfiguration. These are opposed requirements, and the architecture is defined by where it settles between them.

Entry, intervention and extraction across the space–time domain

# The five loci of applied intelligence

*Intelligence attaches where a decision is made under uncertainty, on an observable state, against a feedback signal*

## 1 Artifact transformation

Change in the form or function of a knowledge or business artifact.

Governing question Semantic preservation under transformation

## 2 Process orchestration

Sequencing of workflows under contention, dependency and partial observability.

Governing question The gap between intended and actual flow

## 3 Value capture

Recording value transfer across the supply chain and across multiple channels.

Governing question Fidelity of the economic record to operational reality

## 4 Assurance and auditability

Retrospective verification of value flows and business transactions.

Governing question Whether the enterprise can give a defensible account of itself

## 5 Lifecycle traceability

Concept and design through manufacture, distribution and service.

Governing question Whether the causal chain survives time and handover of custody

Where intelligence attaches to the functional skeleton

# A diagnostic for any proposed application

*Three properties must be identifiable before intelligence is applied to a locus*

1

## Observable state

What can actually be seen of the world at the moment of decision.

2

## Dependent decision

The choice that turns on that state, made under uncertainty.

3

## Evaluation signal

The evidence by which the decision's quality is later established.

## Where the third is absent

What is being built is not an intelligent system but an unfalsifiable one. It will produce outputs that cannot be evaluated, and will therefore drift without detection. This is the principal failure mode in practice — and it is a design failure, not a modelling one.

# Two architectural regimes

*They differ not in purpose but in where knowledge resides and what computation acts on it*

| | Software 1.0 | Next generation |
|---|---|---|
| **Knowledge locus** | Explicit memory | Ontologies and representational memory |
| **Semantics** | Extrinsic, held by operators | Partly intrinsic to the representation |
| **Retrieval** | Exact, locational | Approximate, semantic |
| **Computation** | Enumerated rules | Generalized inference |
| **Competence boundary** | Sharp, at the case-set edge | Diffuse, at the distribution edge |
| **Failure mode** | Discontinuous and legible | Continuous and silent |
| **Primary artifact** | System of record | System of inference |
| **Assurance basis** | Reproducibility, rule provenance | Evaluation, monitoring, feedback |
| **Locus of inference** | External, in the operator | Internal, in the system |

*Purpose is unchanged throughout. What moves is the locus of knowledge, the form of computation over it, and — decisively — the way the system announces that it has stopped working.*

# The first regime: explicit memory and rules

*Behaviour specified directly by human-authored logic — Software 1.0 in Karpathy's sense*

### Determinism

Identical inputs and state yield an invariant output. The system is auditable and its failures reproduce.

### Transparent provenance

Every output traces to a specific authored rule, so responsibility for behaviour is assignable.

### Combinatorial brittleness

Competence degrades discontinuously at the case-set boundary. Maintenance cost grows superlinearly as rules interact.

The characteristic artifact is the system of record — an authoritative store whose function is custody of state rather than reasoning over it.

*Inference, in this architecture, is external to the system and resident in its operators.*

# The second regime: two forms of internalization

*Complementary, not competing — and the distinction decides the architecture*

## Ontologies

*Entities, relations and constraints declared formally.*

- Explicit, checkable, symbolically manipulable
- Supports constraint enforcement and formal derivation
- Commitments can be verified

## Representational memory

*The same content held as learned distributed representations.*

- Supports similarity, analogy and generalization
- Graceful on inputs never specified in advance
- Coverage bought at the cost of verifiability

**The design rule.** Retain ontological structure precisely where correctness must be demonstrable; admit learned representation where coverage matters more than proof. Systems that adopt only the second inherit its unverifiability across their entire operating surface.

# Seven operating conditions — a closed cycle

*Stated generally rather than enumerated case by case*

### Assessment
→ a description

### Evaluation
→ a verdict

### Recommendation
→ an option set

### Generation
→ a candidate

### Prediction
→ a distribution

### Action
→ a state transition

### Feedback
→ a learning signal

## The cycle closes at feedback — the load-bearing element

Assessment through action can each be implemented without it, and frequently are. The result is an apparatus that emits judge ments, recommendations and actions whose quality is never established. Such a system does not fail visibly — it drifts. Feedback instrumentation is therefore not an enhancement added after deployment; it is the condition under which deployment is defensible at all.

# Enterprise architecture → enterprise systems engineering

*A change in the epistemic basis of design, not a change of framework*

### Framework authority

Codified taxonomies and process methodologies resting on professional consensus.

### Quantified constraint

Traceable requirements, calculated margins, enumerated failure modes, verification before fielding.

## Defensible core

The practices that make aerospace engineering trustworthy — explicit requirements, quantified margins, systematic hazard analysis, verification before deployment, disciplined configuration management — are not specific to airframes, and are largely absent from enterprise practice. Their adoption would be a genuine advance.

## Vulnerable extension

The appeal to physical law as such. Aircraft rest on conservation laws that hold invariantly. Enterprise systems have no comparable body: their constituents respond strategically to the rules imposed on them, their environment is non-stationary, and their regularities are statistical and historically contingent.

# The enterprise system stack

*Five layers in dependency order — each resting on those below it*

| | |
|---|---|
| **Intent-driven interfaces** | Multi-modal human and application boundary; expresses desired outcomes rather than procedures |
| **Autonomous multi-agent core execution platform** | Decomposes intent into coordinated action; just-in-time component generation |
| **Self-synthesizing logic** | Constructs and revises operational logic rather than executing authored logic |
| **Continuous semantic ingestion** | Maintains the ontological model in correspondence with the organization’s epistemic reality |
| **Guardrails and trust architecture** | Constrains all layers above; enforces alignment, containment and validation |

# Four principles, each altered by self-synthesis

*The foundational concerns apply without modification — but acquire a distinctive character here*

| | | |
|---|---|---|
| 1 | **Reliability** | A self-synthesizing system's behaviour is not fixed at design time. Reliability becomes a property of the generative process rather than of any particular generated artifact. |
| 2 | **Maintainability** | Generated logic has no author to consult. Maintainability depends on the traceability of generation back to the semantic model that produced it. |
| 3 | **Safety** | Requires explicit hazard analysis of the action space the agentic platform is permitted — not reliance on the absence of observed incidents. |
| 4 | **Human factors** | The human role must be designed, not residual. Operators need the information, authority and interaction latency to intervene — routinely defeated by systems that act faster than supervisors can comprehend. |

# Guardrails and trust architecture

*Preventing epistemic drift — the progressive divergence of the internal model from the reality it represents*

## Misalignment

Divergence between the system’s optimization target and the organization’s actual objectives.

## Runtime assurance

Continuous verification that behaviour stays within sanctioned bounds during operation, not only at design time.

## Containment

Bounded blast radius, so a failure or compromise cannot propagate beyond a defined perimeter.

## Validation

Demonstrable correspondence between system outputs and ground truth.

# Layers of protection: from wires to AI

*Electrical safety as precedent: each physical safeguard has an AI counterpart*

### Insulation

Containment: access controls, permission limits, guardrails

### Circuit breakers

Monitoring, rate limits, rollback, shutdown, incident reporting

### Grounding

Human oversight and appeal routes for high-stakes decisions

### Building codes

Regulation and standards: EU AI Act, NIST AI RMF, ISO/IEC 42001

### Certification labs

Independent testing, red-teaming and third-party audits

### Licensed electricians

Clear liability and professional norms for builders

**Public literacy closes the loop:** people learning what AI can and can't be trusted to do.

# Where the analogy breaks down

*Five properties of AI with no counterpart in electrical safety*



**Unknown physics**

Electricity obeys settled laws; AI behavior is emergent. Interpretability and alignment research is our Ohm's law.



**Agency**

Electricity is passive; AI agents take actions, persuade and adapt.



**Deliberate misuse**

Insulators stop accidents, not intent: fraud, disinformation, cyberattacks, weapons.



**Pace**

Capabilities shift in months, so standards must be adaptive, not fixed rulebooks.

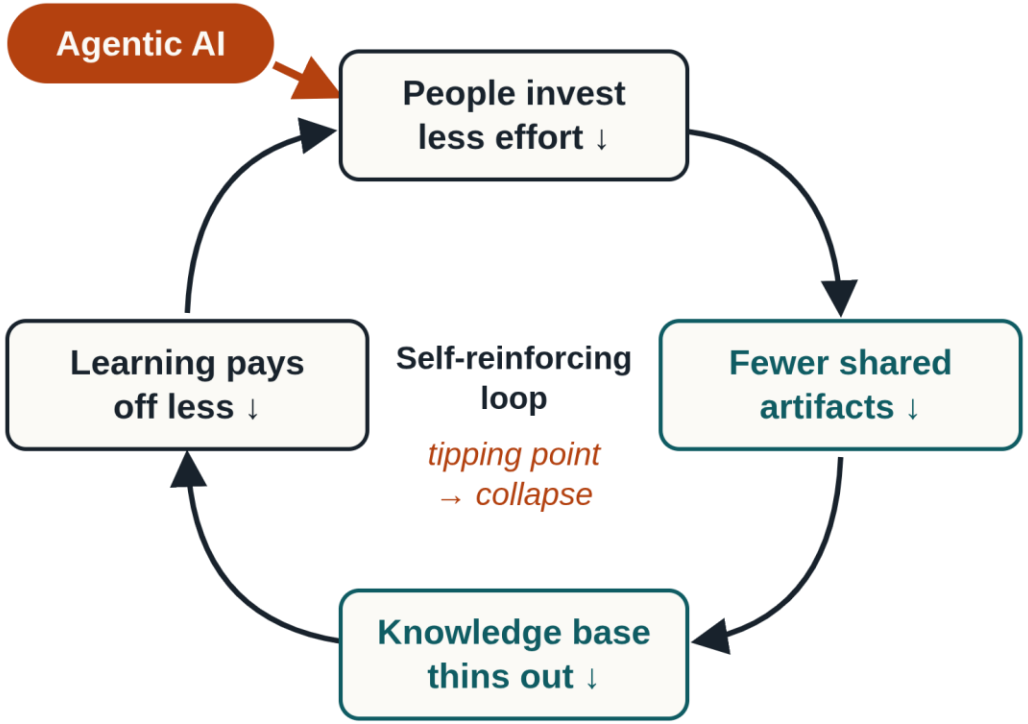


**Global reach**

Models copy worldwide at near-zero cost; safeguards need international coordination.

KNOWLEDGE COLLAPSE · AN ARCHITECTURE VIEW

# When agents supply the answers, the enterprise stops learning




## WHERE IT HITS

| | |
|---|---|
| **Business** | Tacit know-how leaves with people |
| **Information** | Decision records go stale |
| **Application** | Agents run on a frozen corpus |

## ARCHITECT FOR IT

**Write back every decision**

**Keep human effort floors**

**Track knowledge health**

**Principle:** every agentic workflow should give knowledge back, not just consume it.

Source: Acemoglu, Kong & Ozdaglar, "AI, Human Cognition and Knowledge Collapse," MIT Economics, Feb 2026

# Why cross-domain borrowing is functionally necessary

*Software engineering's native vocabulary has no terms for complex adaptive behaviour*

| | |
|---|---|
| **Economics** | Incentive structure, equilibrium and its absence, externality, mechanism design |
| **Psychology** | Bounded rationality, systematic bias, attention as a scarce resource |
| **Physics** | Scaling relations, phase transition, conservation, dimensional reasoning |
| **Control theory** | Feedback, stability margins, observability, requisite variety |
| **Ecology** | Selection pressure, robustness–efficiency trade-offs, cascading failure |

## The discipline it requires

*A borrowed concept earns its place only when:*

- the structural conditions that made it valid in its origin domain demonstrably obtain in the target;
- it yields a prediction that would not have been made without it;
- that prediction is falsifiable against the target system.

*Where only the vocabulary transfers, the borrowing is doing rhetorical work — and in an engineering context that is a liability, not a sophistication.*

CONCLUSION

# Capability is gained. The visibility of failure is lost.

### The finding

Each transition enables the next, and each makes assurance harder. The first regime failed brittlely and legibly. This one fails by drift, with no event to mark it.

### The reordering

The scarce skill is no longer producing representations but governing them: what is represented, under what constraints, and by what evidence the result is held to account.

### The open question

Whether concentrating capability into fewer practitioners expands human capacity or displaces it. Both produce the same signature. They diverge only at failure.

# These systems will carry economic activity as infrastructure carries load.

Infrastructure is made safe not by the sophistication of its components but by the discipline of its design: constraints stated before capability is deployed, failure modes enumerated before they are met, behaviour answerable to someone equipped to judge it.

***Safe by design, or not safe at all.***

Krishnakumar G ·