Judgment Cannot be Outsourced: Modeling Non-Computable Decision Boundaries in Large-Scale Algorithmic Systems

Qinshu Yu[1,2]

[1]Nanyang Technological University, Singapore
[2]University of Melbourne, Victoria, Australia

qinshu001@e.ntu.edu.sg

Abstract. In large-scale digital institutions, formalized processes, compliance mechanisms, and algorithmic systems are essential for managing complexity and operational scale. However, these structures increasingly treat judgment as a computable and outsourcable function. This paper argues that such reduction introduces a structural failure mode: systems may appear operationally stable while progressively losing their capacity for fact recognition, correction, and reality responsiveness.

Judgment is reframed not as a moral attribute but as a non-compressible system-level decision interface composed of situational awareness, action convergence under uncertainty, and responsibility anchoring. From a systems-engineering perspective, compliance mechanisms, KPI-driven incentives, and algorithmic delegation operate as risk-compression structures. When granted veto authority over factual signals, these mechanisms drive a phase transition from rollback-capable failure to irreversible systemic drift.

We formalize this distinction by separating computable decision layers from non-computable judgment boundaries in socio-technical systems. The analysis explains how decision opacity and delayed error amplification emerge when responsibility anchors are weakened. Design implications are provided for algorithmic governance and human-in-the-loop architectures, emphasizing the preservation of interruptible and accountable judgment interfaces.

Keywords: decision systems, algorithmic governance, judgment modeling, risk compression mechanisms, rollback-capable system design.

1 Introduction

As digitalization and scale continue to advance, modern institutions and organizations increasingly operate within highly proceduralized, compliance-oriented, and algorithmic technical structures. Standardized processes, compliance mechanisms, and automated decision systems are not the result of value choices, but an engineering necessity for large-scale systems facing rising complexity, risk, and operational tempo. Without

L. Trajkovic et al. (eds.), *Proceedings of the 2026 2nd International Conference on Data Mining and Project Management (DMPM 2026)*, Advances in Economics, Business and Management Research 390,
https://doi.org/10.2991/978-94-6239-689-0_14

these mechanisms, institutions would struggle to maintain even basic predictability and operability.

However, as these engineered structures gradually occupy the core of institutional operations, a hidden structural risk is accumulating: judgment is implicitly treated as a function that can be computed, compressed, and outsourced. In practice, judgment is constantly translated into process nodes, compliance conditions, or algorithmic output, weakening its systemic function of compressing tension under uncertainty, triggering action, and anchoring responsibility. The result is that institutions, while appearing highly stable in form, gradually lose their sensitivity to changes in facts and their capacity for correction.

This paper is not concerned with the failure of any specific system, but rather a system-level risk pattern that repeatedly appears in large-scale systems. When processes, KPIs(key performance indicators), or algorithms gain the authority to adjudicate fact signals, system operation will experience structural misalignment: local stability continuously increases, while the overall system's responsiveness to reality continuously declines. Such failures do not stem from technical deficiencies or execution errors, but from a flawed engineering assumption—that judgment can be entirely replaced by a computable decision process.

The core proposition of this paper is:

Judgment cannot be outsourced without inducing systemic failure.

This proposition is not an ethical stance, but an engineering judgment. Once judgment is systematically outsourced, while the system may maintain process integrity and operational stability in the short term, it will enter a failure path of delayed feedback, error accumulation, and irreversible drift in the long run.

To analyze this structural problem, this paper abstracts judgment from individual capability or moral attributes and redefines it as a system-level decision interface. Through formal abstraction and failure mechanism analysis, this paper reveals how systems transition from rollback-capable failure to irreversible drift when judgment is compressed and outsourced, and further discusses the engineering implications of this process for algorithmic governance and institutional design.

2 Conceptual Positioning and Scope

This paper defines judgment as a system-level decision structure to avoid conflating it with optimization, rule execution, or moral intention.

Judgment is not equivalent to optimization, which assumes clearly modeled objectives and constraints. Rather than solving for optimality, judgment forms an action convergence point under uncertainty and conflicting goals.

Judgment is not reducible to rule execution. While rules ensure consistency and replicability, judgment operates precisely at the boundaries where rules conflict, fail, or prove insufficient.

Judgment is also distinct from moral intention. Good intentions or value stances do not, by themselves, generate executable decisions or preserve traceable responsibility. The focus here is structural, not motivational.

Based on these distinctions, judgment is positioned as a non-replaceable decision interface within large-scale institutional and algorithmic systems. This paper does not propose new optimization techniques or normative frameworks; instead, it analyzes the structural consequences that arise when judgment is compressed, outsourced, or removed from decision architectures.

3 Judgment as a System-Level Decision Structure

Before discussing the role of judgment in institutional and algorithmic systems, it is necessary to provide a clear engineering definition of judgment. If judgment continues to be understood as an individual trait, an advantage of experience, or a moral performance, subsequent analyses of process, compliance, and system failure will inevitably regress to the level of personality or values, thus deviating from the systemic problem domain this paper focuses on.

3.1 De-romanticizing Judgment

The judgment discussed in this paper does not refer to intelligence, accumulated experience, or benevolent intentions. In an engineering sense, judgment is a system-level decision-making capability whose core function is not to improve decision accuracy but to form a convergence point for action that can enter reality under uncertain conditions.

In the operation of large-scale systems, judgment often occurs in contexts of conflicting goals, incomplete information, or unpredictable consequences. At such times, there is no optimal solution that can be computed by the system. The role of judgment is not to delay decisions to await more information, but to determine whether and how to initiate action and to incorporate the consequences into system feedback. Thus, judgment cannot be simplified into computational steps, rule matching, or value statements; it must be understood as an incompressible decision interface.

3.2 Judgment as a Three-Part Closed Loop

Judgment is defined here as a system-level decision interface rather than an individual trait or moral quality. In engineering terms, it is not a mechanism for improving decision accuracy but a structural capability that enables action under uncertainty.

In large-scale systems, judgment operates in contexts of conflicting goals, incomplete information, and unpredictable consequences—conditions where no optimal solution is computable. Its function is to initiate actionable convergence while preserving traceable responsibility for outcomes. For this reason, judgment cannot be reduced to rule execution, metric optimization, or value expression; it must be treated as a non-compressible decision boundary.

To formalize this structure, judgment is modeled as a closed-loop system consisting of three inseparable components:

- **Situational awareness**: recognition of the operative tension configuration (risk distribution, power structure, information distortion) at a given moment.
- **Action convergence**: formation of a decisive entry point into reality under incomplete information.
- **Responsibility anchoring**: preservation of traceable linkage between decision and consequence, enabling rollback and correction.

The systemic value of judgment lies in the integrity of this loop. When any component is weakened or externalized into processes, KPIs, or algorithms, judgment degenerates into substitute decision forms, increasing the risk of structural drift.

This closed-loop model serves as the analytical interface for examining how large-scale systems transition from rollback-capable failure to irreversible instability.

4 Decision-Making Under System Tension Fields (Field-Shi)

Field-Shi (system tension field) is introduced to describe the actual structural environment in which judgment operates. Unlike conventional engineering models that treat decision-making as rule-matching or objective optimization within enumerable constraints, Field-Shi characterizes decision space as a dynamic and asymmetrical tension topology.

This topology emerges from the interaction of three systemic forces:

- **Power distribution**, which shapes amplification and inhibition paths of action;
- **Risk flow**, which accumulates unevenly across institutional and procedural layers;
- **Information asymmetry**, which filters and delays system perception.

These forces jointly generate constraints that cannot be reduced to rule sets or static condition spaces. Within such a field, formally available options may be practically inoperative, and rule-consistent actions may trigger unintended systemic consequences.

Judgment is therefore modeled as a tension-compression interface. It does not eliminate structural tension but enables actionable entry under non-enumerable conditions. Its function is to compress uncertainty sufficiently to allow system action while preserving responsiveness to evolving constraints.

Recognizing Field-Shi as the operative decision topology clarifies why highly proceduralized systems may appear complete yet gradually marginalize judgment. The scarcity of judgment within dense rule environments constitutes a structural risk rather than an efficiency gain.

5 Formal Abstraction: Judgment as a Non-Compressible Module

To clarify judgment's structural role in institutional systems, we model decision-making as:

$$D = f(P, I, R)$$

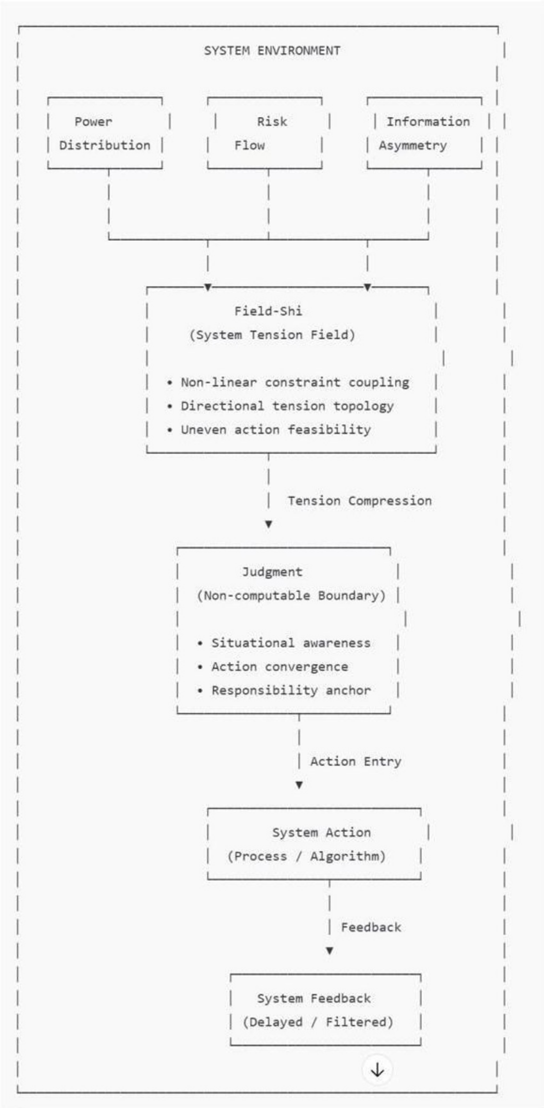

Fig. 1. Field-Shi represents the system tension field formed by power distribution, risk flow, and information asymmetry. Judgment operates at this boundary as a tension compression interface, enabling actionable entry into the system while anchoring responsibility and rollback capability.

where:

- **P** denotes executable procedural constraints (rules, workflows, compliance thresholds);
- **I** denotes observable input signals derived from measurable facts;

- **R** denotes responsibility anchors—traceable decision nodes capable of interrupting, overriding, or reopening execution under anomaly detection.

In proceduralized or algorithmic environments, decision models often collapse into:

$$D = f(P, I)$$

treating processes and data as fully computable layers. However, removing or weakening **R** severs the traceable linkage between decision and consequence. Decisions remain formally valid, yet rollback capacity and error localization deteriorate.

Judgment is therefore defined as the non-computable interface that preserves **R** within the decision structure. It operates not as additional computation but as a structural constraint preventing decisions from degenerating into purely formal operations.

From an engineering perspective, preserving judgment requires three minimal conditions:

1. Decision nodes must remain interruptible under anomaly detection;
2. Responsibility anchors must retain traceable logging authority independent of performance metrics;
3. Fact-level escalation channels must be structurally decoupled from optimization incentives.

Absent these conditions, structural opacity emerges even when procedural compliance is maintained. The system-design blueprint has been shown in the figure 1.

6 Mechanisms of Judgment Failure in Large-Scale Systems

In large-scale institutional systems, judgment failure does not manifest as a single error type but accumulates gradually through several reproducible engineering mechanisms. This section compresses these mechanisms into three typical modes, each presented in the form of defining a structure and its consequence.

6.1 Compliance as Risk Compression

Definition: Compliance mechanisms operate as risk compression tools. When processes gain veto power over fact signals, judgment is structurally bypassed.

Structural Diagram: Fact signal → Compliance check → Recode / Reject → Judgment bypassed

Consequence: Errors are not identified immediately, but are instead absorbed into process stability over time, forming long-term error accumulation.

6.2 KPI-Driven Incentive Distortion

Definition: When performance metrics become the primary driver of action, judgment is compressed into metric optimization behavior, rather than real-world responsiveness.

Schematic structure: Reality → Metric projection → KPI optimization → Action

Consequences: The system is highly insensitive to anomalies not captured by metrics, judgment degenerates into strategic adaptation to the scoring system.

6.3 Algorithmic Delegation and Decision Opacity

Definition: When judgment is outsourced to an algorithmic model, decision output is separated from responsibility anchor, forming structural opacity.

Schematic structure: Input data → Algorithmic model → Decision output → Responsibility detached

Consequences: Errors manifest post-hoc while responsibility becomes structurally unlocatable, reflecting algorithmic opacity and accountability displacement [1].

6.4 Structural Convergence and Phase Transition

The three failure mechanisms—process veto over facts, KPI-driven action, and algorithmic detachment of responsibility—do not operate independently. As they converge, the system crosses a threshold from rollback-capable failure to irreversible drift. Operational stability may persist, but structural capacity for correction is weakened.

6.5 Concrete Case Illustration: Metric Compliance Drift

Consider a large urban service platform managing dispute resolution across subcontracted units. Performance evaluation centered on a single compliance metric: case closure within a reporting cycle.

Over time, procedural optimization replaced substantive settlement. Cases were administratively reclassified or redirected to satisfy closure targets, reducing formal complaint counts while underlying liabilities accumulated.

From a dashboard perspective, resolution rates improved and backlog decreased. Yet the metric gradually became a proxy for reportable compliance rather than fact-bound resolution. No individual decision violated procedure; each remained locally defensible. Structural drift emerged from the coupling of incentive metrics and automated workflow constraints, while rollback capacity—reopening cases based on factual anomalies—declined.

When escalations eventually surfaced at higher coordination levels, the divergence between metric stability and factual instability required disproportionate institutional correction.

This case demonstrates how judgment displacement can occur without misconduct, and how metric optimization may structurally replace responsibility anchors.

7 Phase Transition: From Correctable Failure to Systemic Drift

Under the combined effect of the aforementioned failure mechanisms, large-scale institutional systems do not degrade linearly but undergo state transitions along an identifiable engineering path. This section focuses on the transition process itself, not its value implications.

Delayed feedback is the primary characteristic of phase transition. When the judgment interface is bypassed by processes, KPIs, or algorithms, fact signals no longer directly trigger judgment intervention but enter the system through multiple layers of translation. Each translation introduces delay and filtering, making the system's perception of deviations no longer immediate but lagged and indirect.

Delayed feedback leads to error accumulation, as deviations are absorbed into operational normalcy until systemic drift emerges—an effect consistent with tightly coupled "normal accident" dynamics in complex systems theory [2].

When delayed feedback and error accumulation coexist and cross a threshold, the system will experience rollback loss. At this point, the system loses the ability to reconnect decision consequences to specific judgment nodes; error correction must rely on post-hoc reconstruction or external intervention. System operation can still continue, but its state can no longer be corrected via rollback through internal mechanisms.

After the loss of rollback capability, the system typically maintains intelligibility through narrative explanations, attributing deviations, for instance, to environmental changes or execution-layer issues. This phenomenon can be briefly described as narrative blue-screen: the system is still running but can no longer respond to errors through structural means.

From an engineering perspective, this phase transition is not an anomaly but an expected outcome after the continuous weakening of the judgment interface. Once system stability is no longer constrained by judgment, stability itself becomes a carrier of risk accumulation, driving the system into a state of continuous structural drift.

7.1 Early Warning Indicators for Judgment Drift

To operationalize the detection of structural judgment displacement, several measurable indicators may serve as early warning signals.

First, **signal-delay variance**: increasing temporal gaps between fact-level anomalies and formal case escalation may indicate reduced rollback responsiveness.

Second, **metric–reality divergence ratio**: a sustained improvement in performance metrics accompanied by a stable or rising volume of fact-bound complaints suggests optimization without resolution.

Third, **interruptibility frequency**: the rate at which decision nodes are manually reopened or overridden. A near-zero rate in complex environments may reflect procedural rigidity rather than systemic health.

Fourth, **responsibility diffusion index**: measured by the average number of administrative transfers before final resolution. An upward trend signals structural deferral rather than settlement.

These indicators do not prove malfunction individually. However, in combination, they reveal whether the decision function has effectively reduced to $D = f(P, I)$while weakening the responsibility anchor R.

From an engineering perspective, embedding such metrics into audit dashboards enables early identification of judgment drift before institutional correction costs escalate.

8 Structural Isomorphism and the Necessity of Formal Compliance

To prevent misinterpreting judgment failure as culture- or ideology-specific, this section highlights structural isomorphism across distinct governance traditions.

- **Aristotle – Rule Sufficiency Illusion** [3]: When rule systems are assumed complete, judgment is displaced by classificatory authority; cases beyond rule boundaries are re-encoded rather than structurally reconsidered.
- **Wang Yangming – Action Outsourced to Authority** [4]: When correctness is equated with authoritative action, judgment dissolves into norm execution, weakening independent consequence evaluation.
- **Arendt – Process Immunity** [5]: In proceduralized systems, compliance substitutes for judgment; formally correct execution grants structural immunity to factual deviation.

Despite differing contexts, these models converge structurally: judgment is treated as replaceable. Once rules, authority, or processes adjudicate facts, the judgment interface becomes marginalized.

This paper does not oppose processization. Compliance is an engineering necessity for managing scale and risk. The failure emerges only when compliance shifts from a cost-saving tool to a final adjudication mechanism with veto power over factual anomalies.

Accordingly, the issue is not judgment versus compliance, but boundary calibration: compliance governs predictable domains; judgment must remain addressable at high-uncertainty and irreversible boundaries. Ignoring this division leads to long-term structural drift.

8.1 On the Legitimate Scope of Automation

The argument likewise does not reject automation. In repeatable and enumerable environments, automation appropriately replaces human intervention and enhances consistency.

Structural risk arises only when automated systems acquire veto authority over fact-bound exceptions and operate without preserved responsibility anchors. Opacity is not inherent to automation but to the loss of interruptible judgment.

The engineering question is therefore boundary design: how to maintain separation between computable consistency and fact-responsive responsibility. The framework proposed here specifies this structural condition rather than opposing automation itself.

9 Design Implications for Algorithmic Institutions

Based on the preceding analysis of judgment structures and failure mechanisms, this paper does not propose specific institutional solutions but summarizes inviolable engineering constraints (design constraints) in algorithmic institutions design.

- **Human-in-the-loop ≠ Human rubber stamp**

Human intervention points must possess the authority to actually alter the decision path, rather than merely serving to confirm, endorse, or bear symbolic responsibility afterward.

- **Judgment must be implemented as an interruptible interface**

Highly automated or proceduralized decision paths must retain clear interruption points at their boundaries, allowing judgment to pause the established process and reintroduce fact signals when necessary.

- **Responsibility must be anchored to support rollback**

Judgment intervention points must remain addressable, enabling decision consequences to be reconnected to specific judgment nodes, thereby supporting system-level rollback and structural correction.

10 Rollback as a Structural Requirement

In highly proceduralized and algorithmic institutional systems, rollback is often understood as exception handling or an operational undo mechanism. However, from a systems engineering perspective, this understanding is insufficient. This paper defines rollback as a structural requirement, whose existence directly reflects whether the system still retains judgment capability.

Rollback is not equivalent to technical repair of a single point of failure. Even if the system possesses formal undo or compensation mechanisms, as long as the judgment interface has been structurally removed, the system still cannot reconnect decision consequences to specific judgment nodes. In this context, rollback can only remain at the state level and cannot touch the decision structure itself.

Rollback in its true sense requires judgment to remain addressable within the system. The system must be able to clearly identify: under what judgment intervention did a certain action enter reality. When judgment is compressed into process nodes, KPI weights, or algorithm output, this correspondence is severed, and the system subsequently loses its ability to locate the source of errors.

From the perspective of engineering maturity, rollback capability depends on whether the system retains interruptibility. Interruptibility does not imply frequent intervention but rather the system's ability to pause an established path at critical nodes, reintroduce judgment, and adjust the direction of action. As scale increases, many systems view interruptibility as an efficiency loss and gradually remove it, resulting in smoother operation but a synchronous decline in error correction capability.

Therefore, this paper proposes the following engineering criteria: **System maturity lies not in judgment being completely eliminated, but in the judgment interface still maintaining addressability and interruptibility under highly automated conditions.** Systems with this characteristic can perform structural rollback after errors occur, rather than relying solely on post-hoc narration or external intervention.

10.1 A Minimal Architectural Pattern for Judgment-Preserving Systems

To translate the proposed model into engineering design, a minimal architectural pattern may be outlined as follows:

Layer 1: Input Layer (I). Fact-bound signals are collected, filtered, and structured into machine-readable inputs.

Layer 2: Computable Decision Layer (P). Procedural rules, compliance thresholds, and optimization logic operate on structured inputs to generate provisional decisions.

Layer 3: Interruptible Judgment Node (R). A structurally independent responsibility anchor retains the authority to pause, override, or reopen decisions under anomaly detection. This node must operate outside performance-based optimization metrics.

Layer 4: Traceable Accountability Log. All overrides and escalations are logged in a rollback-capable audit structure to preserve institutional memory.

Layer 5: Feedback Loop. Fact-level outcomes feed back into both procedural adjustment and responsibility review.

The essential design principle is separation without isolation: automation executes consistency, while judgment preserves boundary responsiveness. Structural opacity emerges when Layer 3 is weakened or absorbed into Layer 2.

Closing Remarks

The analysis in this paper can be summarized into three engineering judgments:

- Judgment cannot be outsourced.
- System stability does not imply system reality.
- System maturity requires a rollback-capable judgment interface.

These conclusions are not normative claims but structural descriptions of how large-scale institutional systems operate under engineering constraints. Neglecting the non-outsourcable nature of the judgment interface, systems might maintain stable operation for a period, but their failure will manifest in a more costly and less reversible manner.

Limitations

- This paper does not provide specific institutional or algorithmic implementation schemes, only structural design constraints.
- The conclusions presented primarily apply to decision boundaries with high uncertainty and irreversible consequences.
- Judgment itself can fail; its value lies in the fact that failure remains locatable and correctable.

References

1. J. Burrell, "How the machine 'thinks': Understanding opacity in machine learning algorithms," Big Data & Society, vol. 3, no. 1, 2016.
2. C. Perrow, Normal Accidents: Living with High-Risk Technologies, New York, NY, USA: Basic Books, 1984.
3. Aristotle, Politics, trans. B. Jowett, Mineola, NY, USA: Dover Publications, 2000.
4. Y. Wang, Instructions for Practical Living and Other Neo-Confucian Writings, trans. W.-T. Chan, New York, NY, USA: Columbia University Press, 1963.
5. H. Arendt, Eichmann in Jerusalem: A Report on the Banality of Evil, New York, NY, USA: Penguin Books, 2006.