From the Boundary of the Firm to the Boundary of the Agent

Coase–Information Theory for AI-Mediated Organizations

Brian Guarraci
Galatheus
brian@galatheus.com

June 2026

Working paper — revision 1.2, 2026-06-09

Abstract

Coase explained the firm as an alternative to market coordination when using the price mechanism is costly. This paper generalizes the problem from the boundary of the firm to the boundary of the agent. In AI-mediated organizations, the relevant units of coordination are not only firms and markets, but recursively composed humans, teams, vendors, workflows, software services, and autonomous agents. Boundaries form when shared representations and control loops reduce surprise, delay, error, or misalignment more than they add coordination and governance cost. Boundaries split when modularity, protocols, or market interfaces preserve enough information while lowering internal coordination burden. The paper contributes a formal language for this agent-boundary problem, a decomposition of coordination cost into measurable information-processing terms, and a definition of organizational agility based on decision-value per unit time. Three formal results sharpen the coalesce/split decision: a capacity threshold for when a boundary preserves decision-relevant information; an exact exponential interface-deficit law in the canonical Gaussian-quadratic team, under which the split penalty equals the decision value of the partner's information discounted fourfold per bit of interface capacity; and a square-root law for the efficient number of units that software agents shift toward finer decomposition by lowering interface cost. Mutual information is retained as an idealized proxy for information fidelity, but loss reduction is the primary quality quantity. The main claim is that software agents are boundary-shifting infrastructure: by making sensing, routing, interpretation, escalation, and execution programmable, they change the relative cost of hierarchy, market, and hybrid organization.

Keywords: multi-agent systems; organizational design; transaction-cost economics; software agents; information processing; organizational agility

1 Introduction: The Boundary Problem After Software Agents

Coase framed the firm as a response to the costs of using the price mechanism: firms exist because contracting, searching, negotiating, monitoring, and enforcing every transaction through the market can be more costly than directing work internally [1]. That framing remains essential, but it is no longer sufficient for organizations whose coordination is mediated by software systems, APIs, observability pipelines, ticket queues, chat systems, workflow engines, and autonomous software agents.

These tools do not merely lower generic transaction costs. They change what can be observed, how information is compressed, which interfaces are machine readable, how quickly actions can be taken, and where governance must be applied. The resulting design question is:

> Given a changing environment, what boundary architecture best converts distributed, noisy information into coordinated action under latency, cost, and governance constraints?

This question applies at multiple levels. It determines whether work should be inside a firm or outside it. It also determines whether a team should centralize into an incident room, split into modular services, coordinate with a vendor, or use software agents to mediate between otherwise separate actors. A firm is one boundary choice. A department, workflow, API contract, vendor interface, software service, and autonomous agent are others.

The central claim is:

> Firms, teams, vendors, workflows, and software agents are boundary choices over information-processing units. Boundaries form when shared representation and control loops reduce surprise, delay, error, or misalignment more than they add coordination and governance cost.

The contribution is not the broad claim that firms process information. That intuition has deep roots in Hayek, Simon, Galbraith, and distributed decision theory [6, 7, 15, 11], and a formal economics of organizational information processing has derived efficient structures from communication and processing costs [18, 19, 21, 22]. The contribution is sharper: those models largely fix the communication technology and derive the efficient structure. This paper generalizes the boundary problem from firms to recursively composed agents and models the technology shift itself: software agents alter the relative cost of coalescence, splitting, and protocol-mediated coordination. Three results make the coalesce/split tradeoff precise: a capacity threshold (Proposition 1) for when a boundary preserves decision-relevant information, an exact exponential interface-deficit law in the canonical Gaussian-quadratic team (Theorem 1), and an optimal-decomposition law (Theorem 2) placing the efficient number of units at $m^\star = \sqrt{A/B}$, the square root of internal coupling over interface cost, which software agents shift toward finer decomposition by lowering interface cost.

This paper does not claim that firms disappear, that decentralization is always superior, or that software agents universally improve coordination. The claim is conditional: software agents alter the relative costs of observation, compression, routing, interpretation, execution, monitoring, and misalignment. Depending on task structure, protocol quality, observability, and governance, they may favor coalescence, splitting, or hybrid coordination.

2 Related Intellectual Lineage

The paper sits at the intersection of four literatures.

First, transaction-cost economics explains why activities are sometimes organized inside firms rather than across markets [1, 16]. Search, contracting, monitoring, opportunism, and asset specificity all shape the efficient boundary of the firm.

Second, Hayek's account of dispersed knowledge and Simon's account of bounded rationality treat organizations as mechanisms for coping with limited, distributed information [6, 7, 9, 10]. Marschak and Radner's economic theory of teams formalizes organizations as distributed decision systems [11]. Galbraith's information-processing view of organizations links uncertainty, task interdependence, and structural design [15]. Coordination theory and classic organization theory similarly treat coordination as the management of dependencies among activities [14, 17]. This tradition has a precise formal counterpart. Radner models hierarchies as parallel information processors trading processing delay against managerial cost [18], and Van Zandt extends the analysis to real-time processing by boundedly rational agents [20]. Bolton and Dewatripont derive efficient communication networks when absorbing messages is costly [19]; Garicano derives knowledge hierarchies that economize on acquiring and transmitting knowledge [21]; and Alonso, Dessein, and Matouschek characterize when coordination requires centralization under strategic communication [22]. Arrow's information-cost account of organization [12], Simon's near-decomposability [8], and Brooks's all-pairs communication burden [13] anticipate the cost structure used in Section 7. These models largely take the communication technology as given; the present paper is about what happens when that technology becomes programmable.

Third, information theory provides a vocabulary for representation, compression, fidelity, and distortion [2]. That vocabulary is useful, but not sufficient by itself, because organizational action is valuable only through the loss or utility it produces. Blackwell's comparison of experiments provides the relevant informativeness language: one information structure is more useful when every decision-relevant statistic available under the old structure can be reconstructed under the new one, and possibly more [3].

Fourth, modern software systems introduce programmable protocols, observability streams, APIs, queues, schemas, automated routing, and increasingly autonomous software agents. These make some cross-boundary coordination machine-readable and partially executable. This connects the theory to modularity, information hiding, architectural design, and socio-technical structure in software systems [23, 24, 25, 26], as well as to recent work on tool-using and interactive AI agents [27, 28, 29, 30, 31, 32].

The novelty of Coase–Information Theory is to combine these threads into a single recursive boundary problem. The unit of analysis is not only the legal firm. It is any bounded sensing-representation-decision-action loop.

3 Agent-Boundary Model

Let the external or internal environment generate a latent state

$$X_t \in \mathcal{X}$$

at time t. The state may include customer demand, an incident state, a supply shock, a workflow backlog, a vendor outage, or any other decision-relevant facts.

An organization is a directed graph

$$G = (V, E),$$

where nodes $i \in V$ are humans, teams, services, vendors, software agents, or composite groups. Edges $(i, j) \in E$ are communication or coordination channels.

The word *agent* is used recursively. A node may be a primitive actor, such as a person or software service, or a composite actor, such as a team, firm, platform, vendor network, or incident-response group. Let V_0 be the set of primitive actors and let

$$\mathcal{B} = \{B_1, \dots, B_m\}$$

be a partition of V_0 into bounded agents. Each block B_ℓ is an agent when its members share enough representation, policy, governance, and action interfaces to be treated as one decision-making unit at the relevant level of analysis.

More formally, let $G_0 = (V_0, E_0)$ denote the graph of primitive actors and communication channels. A boundary partition $\mathcal{B} = \{B_1, \ldots, B_m\}$ induces a quotient graph $G_{\mathcal{B}}$, whose nodes are bounded agents and whose edges are cross-boundary coordination channels. For readability, the paper writes $G = (V, E)$ for the graph at the level of analysis under discussion.

Each node receives a local observation

$$Y_t^i \sim P_i(Y \mid X_t)$$

and may send messages

$$M_t^{ij} \in \mathcal{M}_{ij}$$

to neighboring nodes. Messages may be unstructured human communication, structured tickets, API calls, database writes, workflow events, or agent-generated summaries.

Each node maintains a compressed local representation

$$Z_t^i = f_i(Y_{\le t}^i, M_{\le t}^{*i}),$$

where $M_{\le t}^{*i}$ denotes messages received by node i. Node i chooses an action according to a local policy

$$a_t^i \sim \pi_i(Z_t^i).$$

The organization produces a joint action

$$a_t = g\big((a_t^i)_{i \in V}\big)$$

and receives utility $U(X_t, a_t)$, or equivalently incurs loss $L(X_t, a_t) = -U(X_t, a_t)$. Unless otherwise stated, L denotes direct task or decision loss from the chosen action; coordination, delay, monitoring, interpretation, and governance burdens enter through separate cost terms so that they are not counted twice.

The general organizational design problem is

$$\max_{\mathcal{B}, G, \Pi, F} \mathbb{E}\left[\sum_t U(X_t, a_t)\right] - C_{\text{comm}} - C_{\text{comp}} - C_{\text{interp}} - C_{\text{delay}} - C_{\text{error}} - C_{\text{misalign}} - C_{\text{monitor}},$$

where $\Pi = \{\pi_i\}$ is the set of local policies and $F = \{f_i\}$ is the set of compression or representation maps. The same problem can be written in loss form by replacing $U(X_t, a_t)$ with $-L(X_t, a_t)$, turning the maximization of utility minus costs into a minimization of loss plus costs. The scalarization J introduced below generalizes this by attaching explicit weights to each cost term and additionally pricing surprise.

4 Cost Decomposition and Boundary Performance Vector

The Coase–Information move is to treat transaction and coordination costs as partly information-processing quantities. Table 1 gives the minimal decomposition used in the rest of the paper.

Table 1: Information-processing decomposition of coordination cost

Term	Interpretation
C_{comm}	Meetings, messages, handoffs, routing, API calls, protocol overhead.
C_{comp}	Human or machine effort to analyze, summarize, decide, or execute.
C_{interp}	Translation and interpretation across boundaries: schema mismatch, jargon, missing context, tool-output interpretation, vendor/customer translation, and semantic ambiguity.
C_{delay}	Incremental process loss from acting after the state has changed or the opportunity has decayed.
C_{error}	Incremental process loss from stale, incomplete, distorted, or misrouted information.
C_{misalign}	Incremental process loss from local actions that do not compose into good global action.
C_{monitor}	Measurable governance burden: audit, review, compliance, approval, and oversight steps.

This table is intentionally operational. The quantities can be estimated from workflow traces, incident response logs, ticket systems, Slack histories, API calls, approval queues, and agent tool traces. It also prevents the paper from treating "coordination cost" as a black box. A boundary change can reduce one cost term while increasing another. A software agent that removes routing delay but increases monitoring burden is not automatically beneficial; it changes the cost vector. In this paper, monitoring cost is the measurable component of governance cost: approvals, audits, reviews, escalation requirements, and oversight work.

Formally, each boundary architecture induces a performance vector

$$\begin{aligned}\Phi(\mathcal{B}, G, F, \Pi) = (\mathbb{E}\sum_t S_t,\ \mathbb{E}\sum_t L(X_t, a_t),\ C_{\text{comm}},\ C_{\text{comp}},\ C_{\text{interp}},\\ C_{\text{delay}},\ C_{\text{error}},\ C_{\text{misalign}},\ C_{\text{monitor}}).\end{aligned}$$

Here $\mathbb{E}\sum_t S_t$ is cumulative expected surprise, where $S_t = \sum_B S_t^B$ aggregates the per-agent surprise defined in Section 6, and $\mathbb{E}\sum_t L(X_t, a_t)$ is cumulative action loss. The scalar objective used by a particular organization is a weighted or constrained evaluation of this vector. One simple scalarization is

$$\begin{aligned}J(\mathcal{B}, G, F, \Pi) = \mathbb{E}\sum_t L(X_t, a_t) + \beta_S \mathbb{E}\sum_t S_t\\ + \lambda_{\text{comm}} C_{\text{comm}} + \lambda_{\text{comp}} C_{\text{comp}} + \lambda_{\text{interp}} C_{\text{interp}} + \lambda_{\text{delay}} C_{\text{delay}}\\ + \lambda_{\text{error}} C_{\text{error}} + \lambda_{\text{misalign}} C_{\text{misalign}} + \lambda_{\text{monitor}} C_{\text{monitor}}.\end{aligned}$$

Then an objective-efficient architecture satisfies

$$(\mathcal{B}^*, G^*, F^*, \Pi^*) \in \arg\min_{\mathcal{B}, G, F, \Pi} J(\mathcal{B}, G, F, \Pi).$$

This makes the tradeoff explicit: boundary design is multi-objective, but actual organizational choice requires weights, constraints, or governance priorities.

5 Pareto and Objective-Efficient Boundaries

Definition 1 (Agent boundary)**.** *An agent boundary is a partition of actors, observations, representations, policies, tools, permissions, and actions into an internal control loop plus external* *interfaces. Each boundary induces a performance vector consisting of expected surprise, action loss, communication cost, computation cost, interpretation cost, delay cost, error cost, misalignment cost, and monitoring cost. A boundary is* Pareto-efficient *when no feasible boundary movement weakly improves every component of this vector and strictly improves at least one. A boundary is* objective-efficient *for a given organization when no feasible boundary movement lowers the organization's total expected objective after applying its weights, constraints, or governance requirements over these quantities.*

Pareto efficiency describes the frontier of feasible boundary architectures. Objective efficiency selects a point on that frontier. Different organizations may choose different efficient boundaries because they assign different weights to speed, accuracy, compliance, safety, morale, resilience, and strategic learning. In an incident context, a regulated financial platform may choose more monitoring and slower action than a consumer application even when both face the same technical incident, because their governance weights differ.

6 Coalescence, Splitting, and Protocol-Mediated Coordination

The boundary of an agent determines which observations, representations, and actions are internal to a shared control loop and which must cross an interface. Internal coordination can preserve richer context and reduce surprise, but it also creates communication, governance, computation, and planning costs. External coordination can reduce internal burden and preserve modularity, but it can increase interpretation, monitoring, contracting, and delay costs.

Let Z_t^B denote the internal representation held by a bounded agent B, and let a_t^B denote the action induced by that boundary's representation and policy. One idealized measure of surprise is

$$S_t^B = -\log P_B(X_t \mid Z_t^B),$$

the degree to which the true state is unexpected under the agent's representation. The organization-level surprise used earlier is the aggregate $S_t = \sum_B S_t^B$. For continuous state spaces, $P_B(X_t \mid Z_t^B)$ should be read as a conditional density, so S_t^B is a log score rather than the negative log of a probability mass. In decision settings, the same role can be played by prediction error or action loss $L(X_t, a_t^B)$. A boundary-design problem can be stated as

$$\min_{\mathcal{B},G,F,\Pi} \mathbb{E}\left[\sum_t S_t + L(X_t, a_t)\right] + C(\mathcal{B}, G, F, \Pi).$$

Surprise and action loss play different roles. In some applications, surprise is an intermediate diagnostic quantity: unexpected states predict future action loss, rework, escalation, or delay. In those cases, S_t should be treated as a proxy or leading indicator rather than double-counted with L. In other applications, surprise is itself costly because it consumes attention, triggers escalation, creates risk, or forces expensive re-planning. The empirical specification should decide whether surprise enters as an independent cost term or as a measurement proxy for future loss. The same convention applies to delay, error, and misalignment costs: they should be interpreted as incremental process losses not already included in $L(X_t, a_t)$, or else absorbed into L and omitted as separate terms.

Two agents should coalesce when the merged boundary lowers expected surprise or action loss more than it raises internal coordination and governance cost:

$$\mathbb{E}[S_{\text{merged}} + L_{\text{merged}}] + C_{\text{merged}} < \mathbb{E}[S_{\text{separate}} + L_{\text{separate}}] + C_{\text{separate}}.$$

They should split when the inequality reverses.

In incident response, an incident room coalesces SRE, support, product, and infrastructure into one temporary agent so that partial diagnoses become a shared representation instead of crossing weak interfaces.

Let k index a coordination form:

$$k \in \{\text{internal}, \text{market}, \text{hybrid}\}.$$

In the general performance vector, search and contracting burdens are folded into communication, interpretation, monitoring, and delay costs as appropriate. When comparing coordination forms, it is useful to write them separately:

$$C_k = C_{\text{search},k} + C_{\text{contract},k} + C_{\text{comm},k} + C_{\text{interp},k} + C_{\text{monitor},k} + C_{\text{delay},k} + C_{\text{error},k} + C_{\text{misalign},k}.$$

The efficient boundary is selected by

$$k^* = \arg\min_k \{\mathbb{E}[L_k] + C_k\}.$$

Classical hierarchy reduces some contracting and search costs but can increase internal communication delay and bureaucratic compression. Market coordination can reduce internal burden but increase search, contracting, monitoring, and interpretation costs. Agent-mediated hybrid coordination can reduce cross-boundary interpretation and delay if protocols and governance are strong enough.

The claim is not that agents make firms smaller or markets always better. The claim is that software agents and machine-readable protocols can alter enough cost terms to create a new efficient region between classic hierarchy and classic market contracting.

7 The Granularity Sweet Spot

The coalesce/split decision is usually argued as a binary choice between a monolith and a fully decomposed system. The information-processing view makes it quantitative: there is an interior optimum, and software agents move it. This section states three results: a capacity dichotomy that decides whether a given boundary can be drawn without losing decision-relevant information, an exact exponential deficit law that prices the boundary when capacity falls short, and an optimal-decomposition law that locates the efficient number of units. The first two are the information-theoretic backbone; the decomposition law is the statement a practitioner can act on.

7.1 When a boundary is information-clean

Suppose the optimal joint action depends on the world only through a sufficient statistic $T = T(X_t)$: two states that share the same T call for the same action. Let $\ell(T, a) = \mathbb{E}[L(X_t, a) \mid T]$ denote the induced decision loss. Consider a single cut that separates the primitive actors into two sides with pooled local observations Y^A and Y^B. In the coalesced architecture, one agent holds (Y^A, Y^B) and chooses the Bayes action from both observations. In the split architecture, the deciding side holds Y^A and may receive a message M of at most κ bits per decision cycle across the protocol, where κ is the interface capacity: protocol quality or interface fidelity.

Let

$$\mathcal{L}_{\text{coal}} := \inf_{\pi} \mathbb{E}[\ell(T, \pi(Y^A, Y^B))]$$

be the coalesced Bayes risk. Let $\mathcal{P}(r)$ be the class of admissible one-way protocols whose message from side B to the deciding side has rate at most r bits per decision cycle. The rate can be read as an exact one-shot message-length constraint or, in block-coded settings, as the asymptotic description rate with decoder side information Y^A — the Wyner–Ziv regime of source coding with side information at the decoder [4], here with distortion measured by decision loss rather than reconstruction error, so standard rate–distortion machinery applies to $\mathcal{L}_{\text{split}}(r)$. Define the best split risk at rate r by

$$\mathcal{L}_{\text{split}}(r) := \inf_{M \in \mathcal{P}(r)} \inf_{\pi} \mathbb{E}[\ell(T, \pi(Y^A, M))].$$

Definition 2 (Decision-relevant interface rate)**.** *The* decision-relevant interface rate *is the operational threshold*

$$R^\star := \inf\{r \geq 0 : \mathcal{L}_{\text{split}}(r) = \mathcal{L}_{\text{coal}}\},$$

and, for tolerance $\varepsilon > 0$, the ε-sufficient rate *is*

$$R^\star_\varepsilon := \inf\{r \geq 0 : \mathcal{L}_{\text{split}}(r) \leq \mathcal{L}_{\text{coal}} + \varepsilon\}.$$

Thus $R^\star$ is the minimum cross-boundary rate needed to preserve the coalesced architecture's decision value for the loss under study; κ is protocol capacity measured in the same units. The conditional mutual information

$$I(T; Y^B \mid Y^A)$$

is still a useful interdependence diagnostic and a lower bound when the protocol must preserve the full Blackwell experiment about T, but it is not, by itself, a universal achievability rate for every decision problem (Corollary 1 makes the link exact in the Gaussian-quadratic team). The threshold behaves as a dichotomy by construction; stating it as such fixes language for what follows.

Proposition 1 (Capacity dichotomy)**.** *Assume finite or discretized observations, bounded loss, and that the infima above are attained; otherwise read the statement with arbitrarily small rate/loss slack.*

1. Achievability. *If $\kappa \geq R^\star$, there exists a protocol under which the split architecture attains the coalesced architecture's expected decision loss, up to the stated slack.*
2. Converse. *If $\kappa < R^\star$, then for every protocol the split architecture incurs strictly higher expected decision loss than the coalesced architecture. The loss gap*

$$\Delta(\kappa) := \mathcal{L}_{\text{split}}(\kappa) - \mathcal{L}_{\text{coal}}$$

is positive and is non-decreasing as available capacity κ falls.

Proof. The feasible protocol classes are nested: $\mathcal{P}(r_1) \subseteq \mathcal{P}(r_2)$ whenever $r_1 \leq r_2$. Hence $\mathcal{L}_{\text{split}}(r)$ is non-increasing in r. If $\kappa \geq R^\star$, the definition of $R^\star$, together with attainment or arbitrarily small slack, gives a protocol of rate at most κ whose risk equals $\mathcal{L}_{\text{coal}}$. Conversely, if $\kappa < R^\star$ and a rate-κ protocol attained coalesced risk, then κ would belong to the set defining $R^\star$, a contradiction. Therefore $\mathcal{L}_{\text{split}}(\kappa) > \mathcal{L}_{\text{coal}}$. The gap $\Delta(\kappa)$ inherits monotonicity from $\mathcal{L}_{\text{split}}(\kappa)$: reducing capacity shrinks the feasible protocol set and cannot lower the best achievable split risk. □

Proposition 1 says a boundary is *information-clean* exactly when its decision-relevant cross-traffic fits through the protocol. Because the architect chooses the cut, the natural design move is to minimize cross-boundary decision rate,

$$\min_{\mathcal{B}} \; R^\star(\mathcal{B}),$$

placing boundaries along cuts of minimal decision-relevant interdependence. Conditional mutual information remains a useful lower-bound diagnostic, but the threshold quantity is the operational rate required to preserve decision value. This is near-decomposability [8] and "high cohesion, low coupling" made literal: coupling is the decision-relevant rate that crosses the boundary, and Conway's law becomes the statement that organizations ship their information cuts.

In smooth environments the dichotomy is the wrong resolution: exact equality can require unbounded rate, while the penalty for finite rate decays fast. The canonical team-theoretic setting makes this exact.

Theorem 1 (Exponential interface-deficit law)**.** *Let (X, Y^A, Y^B) be jointly Gaussian, let loss be quadratic, $L(x, a) = (x - a)^2$ (so the sufficient statistic is $T = X$), and let the deciding side hold Y^A while side B communicates one-way at rate κ bits per decision cycle in the block-coded regime. Then the split penalty is exactly*

$$\Delta(\kappa) = \mathcal{L}_{\text{split}}(\kappa) - \mathcal{L}_{\text{coal}} = \underbrace{\left[\operatorname{Var}(X \mid Y^A) - \operatorname{Var}(X \mid Y^A, Y^B)\right]}_{\Delta(0)} \cdot 2^{-2\kappa} :$$

the penalty for splitting equals the decision value of the partner's information, discounted fourfold per bit of interface capacity. Consequently $R^\star = \infty$ whenever $\Delta(0) > 0$, while for every tolerance $\varepsilon \in (0, \Delta(0)]$,

$$R^\star_\varepsilon = \tfrac{1}{2} \log_2 \frac{\Delta(0)}{\varepsilon}.$$

Proof. By joint Gaussianity the coalesced Bayes action is $T' = \mathbb{E}[X \mid Y^A, Y^B] = a^\top Y^A + b^\top Y^B$; set $V := b^\top Y^B$, which the encoder can compute alone. Since $X - T'$ has zero mean conditional on (Y^A, Y^B), and any decoder output $\hat{a}$ is a function of (Y^A, M) with M a (possibly randomized) function of Y^B, the cross term vanishes and, for every protocol, $\mathbb{E}(X - \hat{a})^2 = \mathbb{E}(X - T')^2 + \mathbb{E}(T' - \hat{a})^2 = \mathcal{L}_{\text{coal}} + \mathbb{E}(T' - \hat{a})^2$. With $a^\top Y^A$ known to the decoder, estimating T' is estimating V: the problem reduces to remote estimation of the scalar Gaussian V under mean-squared error with decoder side information Y^A at rate κ. For jointly Gaussian sources under quadratic distortion, Wyner–Ziv coding suffers no rate loss relative to conditional rate–distortion [4, 5]: the optimal distortion is $D(\kappa) = \operatorname{Var}(V \mid Y^A)\, 2^{-2\kappa}$, with matching converse. Finally, the orthogonal decomposition $X - \mathbb{E}[X \mid Y^A] = (X - T') + (V - \mathbb{E}[V \mid Y^A])$ gives $\operatorname{Var}(X \mid Y^A) = \mathcal{L}_{\text{coal}} + \operatorname{Var}(V \mid Y^A)$, so $\operatorname{Var}(V \mid Y^A) = \Delta(0)$. Substituting gives the law, and solving $\Delta(\kappa) \le \varepsilon$ gives $R^\star_\varepsilon$. $\square$

Corollary 1 (Exact information bridge)**.** *For scalar X in the setting of Theorem 1, $I(X; Y^B \mid Y^A) = \frac{1}{2} \log_2 \left(\operatorname{Var}(X \mid Y^A) / \operatorname{Var}(X \mid Y^A, Y^B)\right)$, hence $\Delta(0) = \operatorname{Var}(X \mid Y^A)\left(1 - 2^{-2I(X; Y^B \mid Y^A)}\right)$: the conditional mutual information used above as a diagnostic becomes an exact measure of the zero-capacity penalty.*

Three readings matter for design. First, the deficit law prices a leaky boundary: the cost of a thin protocol is not failure but an exponentially discounted share of the partner's information value. Second, capacity requirements grow only logarithmically as tolerance tightens ($R^\star_\varepsilon = \frac{1}{2} \log_2(\Delta(0)/\varepsilon)$), so protocol capacity is cheap relative to decision fidelity. Third, the law gives the empirical program of Section 14 a functional form: in the smooth regime, the logarithm of the excess loss should fall linearly in interface capacity with slope -2 per bit, i.e. each added bit of effective schema capacity cuts the excess loss roughly fourfold.

7.2 How many units: an optimal-decomposition law

A clean cut still has a cost, and so does coalescence. Coordination is an all-pairs problem [13]: holding the $N = |V_0|$ primitive components in one mutually coherent representation costs on the order of aN^2. Splitting into $m = |\mathcal{B}|$ units of size N/m reduces within-unit coordination to $a(N/m)^2$ per unit, hence aN^2/m in total, while each boundary adds a roughly fixed interface cost B when cuts are information-clean in the sense of Proposition 1. Writing $A := aN^2$ for the coalesced coordination cost, total coordination cost as a function of granularity is

$$\Theta(m) = \frac{A}{m} + B\,m, \qquad m \geq 1.$$

Theorem 2 (Optimal decomposition). *If interface costs are additive across information-clean boundaries ($\kappa \geq R^\star$ at each cut), then Θ is strictly convex on $m > 0$ and is minimized at*

$$m^\star = \sqrt{\frac{A}{B}}, \qquad \Theta(m^\star) = 2\sqrt{AB}.$$

Equivalently, starting from a monolith one should introduce a new boundary if and only if the internal coordination it removes exceeds the interface coordination it adds, and stop when the two are equal.

Proof. $\Theta'(m) = -A/m^2 + B$ and $\Theta''(m) = 2A/m^3 > 0$ for $m > 0$, so Θ is strictly convex; setting $\Theta'(m) = 0$ gives $m^\star = \sqrt{A/B}$, and substitution gives $\Theta(m^\star) = 2\sqrt{AB}$. The marginal characterization is the first-order condition: $\Theta'(m) < 0$ (splitting still pays) for $m < m^\star$ and $\Theta'(m) > 0$ (splitting overshoots) for $m > m^\star$. □

The law names both failure modes. An organization with $m \ll m^\star$ is an over-coupled monolith paying avoidable internal coordination; one with $m \gg m^\star$ is over-fragmented, a distributed monolith paying avoidable interface cost. The efficient granularity is the square root of how internally coupled the system is relative to how expensive its interfaces are. This is the same opposing-cost structure as the economic-order-quantity rule, which is why the optimum is a square root.

Corollary 2 (Agents shift the optimum). *Software agents and machine-readable protocols that lower the per-boundary cost to $B' < B$ raise the efficient number of units to $m^{\star\prime} = \sqrt{A/B'} = m^\star\sqrt{B/B'}$. Since $m^\star \propto B^{-1/2}$, halving interface cost increases the efficient unit count by about 41%.*

This is the quantitative content of the paper's thesis. Software agents are boundary-shifting infrastructure not because they automate tasks but because they lower B and thereby slide $m^\star$ toward finer, more autonomous decomposition. The effect is measurable: estimate B before and after, and the law predicts the new efficient granularity.

Corollary 3 (Interdependence floor). *The additive interface model holds only while cuts are information-clean. When interdependence is high enough that no partition achieves $R^\star(\mathcal{B}) \leq \kappa$, each boundary carries an irreducible decision-loss term $\Delta(\kappa) > 0$ from Proposition 1, priced in the Gaussian-quadratic case by Theorem 1. The constant-B model is then misspecified: effective interface cost must include the capacity-deficit loss, and if those deficits persist or grow as additional cuts are drawn, the efficient architecture moves back toward coalescence. Beyond this floor, lowering the mechanical interface cost B alone does not restore decomposability; the protocol must also raise κ or the architecture must choose cuts with lower $R^\star(\mathcal{B})$.*

Together the results bound the design space from both sides. Proposition 1 decides whether a given boundary is admissible, and Theorem 1 prices the deficit when it is not; Theorem 2 decides how many admissible boundaries to draw; Corollary 2 shows agents slide the optimum toward finer decomposition; and Corollary 3 shows interdependence sets a hard limit no protocol below capacity can cross. In the regime map of Figure 1, the diagonal $R^\star = \kappa$ marks the operational capacity boundary: on its high-fidelity side $m^\star = \sqrt{A/B}$ governs granularity; on its high-interdependence side capacity-deficit loss pushes the architecture toward coalescence.

Remark 1 (Assumptions)**.** *The all-pairs form aN^2 is the simplest super-linear coordination cost; any strictly convex internal cost yields a unique interior optimum and changes only the exponent in the root. Constant per-boundary cost B is exactly the information-clean assumption of Proposition 1; its failure is the content of Corollary 3. The linear interface term $B\,m$ additionally assumes the quotient graph is sparse — tree- or bus-like, each unit maintaining $O(1)$ interfaces — which is what modular architectures are designed to achieve [24, 26]. If instead every unit must interface with every other, interface cost scales as $B\,m^2$ and the optimum becomes a cube root, $m^\star = (A/2B)^{1/3}$: denser interface graphs lower the efficient unit count and flatten the gain from cheaper interfaces. The formula also presumes an interior optimum $1 \le m^\star \le N$, and since a monolith carries no interface, $B\,m$ may be read as $B\,(m-1)$, which shifts Θ by a constant and leaves $m^\star$ unchanged. The results are stated for the stationary, single-decision case; sequential settings compose with the latency discount of Proposition 3 through the delay-cost term.*

8 Organizational Agility

Speed alone is not agility. An organization can act quickly and incorrectly. Correctness alone is not agility either. An organization can make the right decision too late.

Let τ_t be the end-to-end latency from signal availability to coordinated action. Let a_t^0 be a baseline action, default policy, or status-quo response. Define decision value as loss reduction relative to that baseline:

$$Q_t = \mathbb{E}\left[L(X_t, a_t^0) - L(X_t, a_t)\right].$$

The quantity Q_t may be negative when the coordinated action performs worse than the baseline; in that case, speed amplifies negative decision value rather than producing agility. Organizational agility is decision value per unit time:

$$\mathcal{G}_t = \frac{Q_t}{\tau_t}.$$

A cost-adjusted version is

$$\mathcal{G}_t^{(c)} = \frac{Q_t}{\tau_t(1+\widetilde{C}_t)},$$

where $\widetilde{C}_t \ge 0$ is a normalized resource-cost index for producing the action.

This definition makes the quality term explicit. An action can encode a great deal of information about the state and still be a poor action. Loss reduction, expected utility improvement, or expert-rated decision value is therefore the primary quantity. Mutual information remains useful as an idealized proxy:

$$I(X_t; a_t)$$

is a measure of information fidelity between the world and the organization's actions. It is appropriate when higher information fidelity monotonically improves action quality under the policy class being studied. Otherwise, decision value should dominate.

9 Compression and Organizational Structure

Organizations rarely transmit the full state X_t. Instead, they construct internal representations

$$X_t \to Z_t \to a_t.$$

Roles, dashboards, metrics, status reports, tickets, runbooks, APIs, schemas, plans, OKRs, approvals, and incident summaries are compression mechanisms. They reduce communication burden, but they can also destroy decision-relevant information. A structured incident ticket, for example, is a protocol that compresses incident state across a boundary: it can preserve severity, affected service, suspected cause, owner, and customer impact, or it can omit the diagnostic detail needed for a good response.

The organizational design problem has the structure of a rate-distortion problem:

$$\min_f C_{\mathrm{repr}}(f) + \lambda\, \mathbb{E}[L(X_t, a_t(f(X_t)))],$$

where $C_{\mathrm{repr}}(f)$ is the cost of maintaining and communicating representation f, and L is decision loss. A dashboard that compresses a complex operational state into one green/yellow/red indicator is valuable if it preserves the information needed for action. It is dangerous if it hides the structure needed for diagnosis.

10 Software Agents as Boundary-Shifting Infrastructure

Software agents are not merely labor-saving tools. They are boundary-shifting infrastructure.

A software agent can be modeled as a node

$$\mathrm{Agent}_i = (\mathcal{O}_i, \mathcal{M}_i, \pi_i, \mathcal{T}_i, \Gamma_i),$$

where $\mathcal{O}_i$ is the set of observation channels, $\mathcal{M}_i$ is the message schema or protocol, π_i is the local decision policy, $\mathcal{T}_i$ is the set of tools or actuators, and Γ_i is the governance envelope: permissions, constraints, auditability, and escalation rules.

The important organizational effect of software agents is not only task automation. It is that they make parts of cross-boundary coordination programmable. They can observe, compress, route, escalate, execute, and audit across interfaces that previously required human interpretation or managerial hierarchy. When sensing, routing, interpretation, escalation, and execution can be mediated by machine-readable protocols, the efficient boundary between hierarchy, market, and hybrid organization changes.

Agents can observe continuously, summarize and route information, enforce schemas, call tools, monitor queues, escalate uncertainty, and execute routine actions. But agentic coordination is not automatically superior. It may increase local-global misalignment, proxy optimization, monitoring burden, structured propagation of bad assumptions, and compression loss from over-summary.

The relevant comparison is conditional:

> Agents improve organizational agility when tasks are sufficiently structured, protocols are sufficiently expressive, and governance is strong enough to control misalignment.

In an agent-mediated incident workflow, a software agent can summarize evidence, route exceptions, trigger routine remediation, and escalate uncertainty without requiring every actor to merge into one permanent team.

11 Running Example: Incident Response

Incident response is a canonical instance of the agent-boundary problem. The latent state X_t is the true incident state: affected service, root cause, blast radius, customer impact, and time sensitivity. Local observations Y_t^i include alerts, logs, metrics, customer reports, traces, and partial diagnoses held by SRE, support, infrastructure, product, vendor, or agentic monitoring nodes. Local representations Z_t^i include each actor's working diagnosis. Messages M_t^{ij} include pages, Slack messages, ticket updates, dashboard links, incident summaries, and agent-generated briefings. Actions a_t include rollback, mitigation, escalation, customer communication, vendor failover, or no-op.

The cost decomposition becomes concrete, as summarized in Table 2.

Table 2: Incident response as an agent-boundary problem

Formal term	Incident-response interpretation
X_t	True incident state: root cause, severity, affected users, blast radius.
Y_t^i	Logs, metrics, alerts, customer reports, traces, vendor status.
Z_t^i	Local diagnosis held by SRE, support, infra, product, vendor, or agent.
M_t^{ij}	Pages, Slack updates, tickets, dashboards, agent summaries, API events.
a_t	Rollback, mitigation, escalation, customer communication, failover.
C_{interp}	Translation cost across SRE, support, product, vendor, and customer-facing language.
C_{delay}	Outage duration, customer impact, missed SLA, lost trust.
C_{error}	Wrong diagnosis, bad rollback, repeated mitigation, incident reopen.
C_{misalign}	Support says one thing while infrastructure does another.
C_{monitor}	Approval gates, blast-radius review, compliance and audit checks.

The boundary choices are visible. One integrated incident room coalesces actors into a shared representation and control loop. Separate teams preserve local autonomy but communicate through tickets and status updates. A vendor boundary uses contracts, SLAs, and escalation paths. An agent-mediated hybrid uses software agents to monitor queues, summarize evidence, enforce schemas, route exceptions, and trigger routine tool calls while escalating uncertainty to humans.

The best boundary depends on interdependence, protocol quality, volatility, observability, and governance. If the incident is tightly coupled across systems and the interfaces are weak, coalescence is attractive. If the incident is modular and protocols are strong, split teams or vendors can coordinate effectively. If the workflow is semi-structured, observable, and governed, software agents can absorb routing and interpretation burden without requiring full organizational merger.

12 Propositions

Proposition 2 (Protocol-mediated coordination)**.** *For a fixed organizational graph and decision-policy class, replacing protocol P with protocol P' weakly increases achievable net objective value if P' is at least as informative about every decision-relevant statistic of X_t, reduces or preserves interpretation cost, and does not increase delay, rigidity, or monitoring cost by more than the value created.*

Intuition. A machine-readable schema, API, or structured ticket that preserves relevant state is a less garbled signal. Better downstream decisions become feasible, but the result is not free if the protocol itself adds rigidity or review burden. In Blackwell's sense, P' is more useful when downstream agents can reconstruct any decision-relevant statistic available under P, and possibly more. A structured incident ticket is useful for exactly this reason: it carries severity, owner, affected service, and current hypothesis across a boundary without requiring full coalescence.

Proposition 3 (Latency under volatility)**.** *Suppose the relevance of information decays with environmental hazard rate λ. If a decision is taken after latency τ, the value of otherwise correct information is approximately discounted by $e^{-\lambda\tau}$. Therefore, the marginal value of latency reduction rises with volatility.*

Sketch. If the probability that the relevant state remains stable after delay τ is $e^{-\lambda\tau}$, then effective information available for action is proportional to $I_0 e^{-\lambda\tau}$ or, more generally, decision value $Q_0 e^{-\lambda\tau}$. Higher λ makes delay more costly.

Proposition 4 (Interdependence and modularity)**.** *When task interdependence is high, coalescence into a shared representation becomes more valuable. When task modularity and protocol quality are high, splitting or protocol-mediated coordination becomes more valuable.*

Intuition. Interdependent tasks create cross-boundary surprise because local actions affect each other. Shared representation can reduce that surprise. Modular tasks with good protocols lose less information at the interface, so separate agents can preserve autonomy without large interpretation losses. Theorem 2 and Corollaries 2–3 make this quantitative: the efficient granularity is $m^\star = \sqrt{A/B}$, better protocols lower B and raise $m^\star$, and interdependence beyond interface capacity forces $m^\star$ back toward coalescence.

Proposition 5 (Agentic routing)**.** *For structured or semi-structured workflows, agent nodes improve the net boundary objective when reductions in routing, communication, interpretation, and execution latency exceed induced monitoring, governance, misalignment, and error costs. They increase cost-adjusted organizational agility when those reductions also raise decision value Q_t, lower latency τ_t, or lower normalized resource cost $\widetilde{C}_t$ enough to improve $\mathcal{G}_t^{(c)} = Q_t/(\tau_t(1+\widetilde{C}_t))$.*

A sufficient condition for the net-objective improvement is

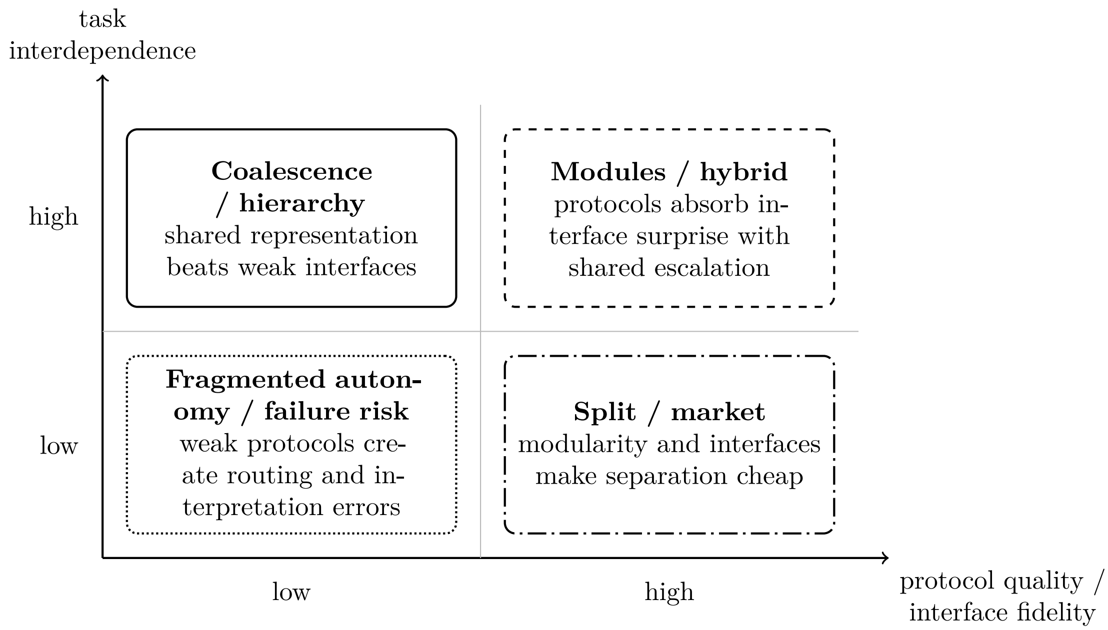
$$R_{\text{delay}} + R_{\text{comm}} + R_{\text{interp}} + R_{\text{exec}} > I_{\text{monitor}} + I_{\text{governance}} + I_{\text{misalign}} + I_{\text{error}},$$

where R_x denotes cost reductions and I_x denotes induced costs. Here I_{monitor} denotes measurable oversight work, while $I_{\text{governance}}$ denotes broader policy, permissioning, auditability, and escalation overhead. The execution and governance terms extend the Table 1 taxonomy: R_{exec} is reduced execution effort within C_{comp}, and $I_{\text{governance}}$ is the broader, partly non-measurable counterpart of the measurable monitoring cost C_{monitor}. The inequality is a net-objective condition; for agility it must operate through the numerator Q_t, latency τ_t, or normalized cost index $\widetilde{C}_t$.

Proposition 6 (Governance limit)**.** *Agentic decentralization has diminishing or negative returns when governance quality is low because autonomy can increase misalignment and monitoring costs faster than it reduces delay.*

Intuition. A fast local optimizer is valuable only when its actions compose into the global objective.

Figure 1: Boundary regimes as a function of interdependence and protocol quality. The diagonal $R^\star = \kappa$—operational decision rate equal to interface capacity—is the capacity boundary of Proposition 1; in smooth environments read $R^\star$ as the ε-sufficient rate $R^\star_\varepsilon$ of Theorem 1. Below and to its right, boundaries are information-clean and the granularity law $m^\star = \sqrt{A/B}$ applies; above and to its left, interdependence exceeds capacity and Corollary 3 pushes the architecture toward coalescence.



13 Executable Illustration

The companion web-based simulation is an executable diagram, not a calibrated causal estimate. The framework is intended first as a comparative design language: it identifies which cost terms should move under a boundary change, not a universal predictive model of all organizations. The simulation samples a latent world state, gives primitive agents noisy local observations, and evaluates alternative boundary forms: coalesced, split, market/vendor, and agent-mediated hybrid. The objective includes surprise, action loss, latency, coordination cost, up-front blueprint delay, runtime surprise, and rigidity risk. A live, interactive companion runs the implementation in the browser, and the public repository hosts its source.[1]

Figure 1 reports the qualitative regime map generated by the simulation's core comparative statics.

The simulation also exposes a planning tradeoff. Blueprint-heavy design reduces runtime surprise in stable environments, but adds up-front delay and can become rigid when volatility is high. Reactive design avoids planning delay but leaves more surprise for runtime. This illustrates the paper's broader claim: plans, like organizational boundaries, are information investments.

14 Measurement and Empirical Agenda

A first empirical study should use incident response or support routing traces. The minimal research design is:

[1]Live page: galatheus-labs.github.io/coase-info-theory. Source: github.com/galatheus-labs/coase-info-theory.

1. collect workflow traces,
2. map events to cases,
3. compute latency, handoffs, surprise, and action-quality proxies,
4. compare human-heavy, protocolized, and agent-assisted workflows,
5. estimate whether agentic coordination improves agility after controlling for case complexity.

A natural starting design is a matched before/after study around the introduction of an agentic routing or summarization system. Cases should be matched by workflow type, severity, service, time of day, customer segment, and complexity. A stronger design exploits staggered adoption across teams or workflows, comparing changes in resolution time, handoffs, rework, reopen rate, escalation rate, and expert-rated quality between adopting and non-adopting groups. The goal is not to directly estimate mutual information, but to test whether agent-mediated coordination improves decision value per unit time after controlling for case mix. The primary dependent variables are resolution time, handoff count, rework, reopen rate, escalation rate, SLA success, and expert-rated decision quality.

In most organizational traces, neither X_t nor $I(X_t; a_t)$ is directly observable. The empirical version should use loss-reduction and quality proxies, as shown in Table 3.

Table 3: Operational proxies for formal quantities

Formal quantity	Operational proxy
Q_t or decision value	Severity reduction, SLA success, expert quality score, avoided rework.
S_t or prediction surprise	Unexpected escalation, reopen, SLA miss, anomaly, decision reversal.
τ_t	Time to triage, time to decision, time to mitigation, time to resolution.
$\widetilde{C}_t$ or cost index	Handoffs, people involved, messages, meetings, queue time, tool calls.
C_{interp}	Clarification requests, schema translation, handoff notes, vendor/customer language translation.
C_{error}	Reopen rate, rollback rate, bad routing, decision reversal.
C_{monitor}	Approvals, audits, review steps, escalation requirements.

A practical agility proxy is

$$\widehat{\mathcal{G}} = \frac{Q}{1 + T_{\text{resolution}}} \cdot \frac{1}{1 + \alpha H},$$

where Q is quality, $T_{\text{resolution}}$ is resolution time, H is handoff count, and α is a handoff penalty. This proxy is not a direct mutual-information estimate. It is a practical measurement bridge.

To study coalescing and splitting, compare whether a workflow performs better when work is handled inside one bounded agent, split across separate agents with an explicit protocol, or coordinated through a hybrid boundary such as an agent-managed vendor or escalation path. The empirical question is whether the boundary change lowers surprise and action loss enough to justify any added coordination or monitoring cost.

Incident traces provide natural measurements of latency, handoffs, reopen rate, escalation, and decision quality, which is why they are the preferred first empirical spine for the theory.

14.1 A minimal test of the capacity threshold

The cheapest credible test of Proposition 1 proxy-estimates the decision-relevant interface rate $R^\star$ and the interface capacity κ from handoff traces and checks whether decision loss rises where the operational rate exceeds capacity. Incident and support-ticket data are well-suited because handoffs are the interface. A single quarter of tickets from one organization is sufficient, ideally spanning a boundary change—a coalesced incident-room period versus a ticket-handoff period—or contrasting a coalesced and a split team.

Three quantities are estimated from the trace:

- $R^\star$, the decision-relevant cross-boundary rate, proxied by *clarification round-trips per handoff*: how often the receiving side must return for more information before it can act. Persistent back-and-forth indicates that the handoff did not carry enough decision-relevant state.
- κ, the interface capacity, proxied by *schema completeness at handoff*: the fraction of decision-relevant fields (severity, suspected cause, blast radius, owner) populated when work crosses the boundary.
- decision loss, proxied by resolution time and reopen rate, controlling for severity and case complexity.

Proposition 1 predicts an *elbow*: where the operational deficit $R^\star - \kappa$ is non-positive, handoffs are information-clean and split performance can match coalesced performance; where the deficit is positive, decision loss should rise with it. Plotting resolution time or reopen rate against a proxy for $(R^\star - \kappa)$ should be flatter to the left of zero and increasing to the right. Observing this elbow is evidence that the capacity boundary is real rather than merely drawn. Theorem 1 sharpens the prediction from a shape to a functional form: in the smooth regime the logarithm of the excess loss should fall approximately linearly in effective interface capacity, each added bit of schema capacity cutting the excess roughly fourfold. A log-linear fit of excess resolution time or reopen rate against the capacity proxy is therefore a stronger test than the elbow alone.

A complementary check tests the cost-vector prediction summarized in Table 1. For boundary-change cases, an agent-mediated routing or summarization layer should lower interpretation cost C_{interp} (clarification requests) and delay cost C_{delay} (time to triage) while raising monitoring cost C_{monitor} (approvals), with net agility $\widehat{\mathcal{G}} = Q/((1 + T_{\text{resolution}})(1 + \alpha H))$ increasing. Corollary 2 adds a sharper, falsifiable prediction: lowering the per-boundary cost B should raise the efficient number of units $m^\star = \sqrt{A/B}$, so teams that adopt stronger protocols or agentic interfaces should sustain finer decomposition at equal or lower coordination cost.

This design requires no model training and no causal identification beyond matching on case mix, and it can run on a single design partner's ticket export. Where trace data is not yet available, the same protocol stands as a pre-registered design.

15 Limitations

This framework has several limitations.

First, decision value is difficult to measure. Expert labels, SLA outcomes, severity reduction, and rework are proxies, not direct access to $L(X_t, a_t)$.

Second, mutual information is elegant but subordinate. It is useful for idealized information-fidelity arguments, but high mutual information between state and action does not guarantee good action.

Third, agents do not universally improve coordination. They help most when workflows are structured, observability is high, and governance is clear.

Fourth, some organizational knowledge is tacit and difficult to encode into protocols. Over-protocolization can compress away context.

Fifth, the current simulation is a stylized model. It illustrates mechanisms but does not yet estimate causal effects.

Sixth, organizational utility is multi-objective. Speed, accuracy, compliance, safety, morale, trust, and strategic learning may trade off.

16 Conclusion

Coase–Information Theory reframes the firm as an information architecture and then generalizes that idea. The key economic question is not only whether work should be coordinated by markets or hierarchy, but which agents should coalesce, which should split, and which should coordinate through protocols so that distributed observations become coordinated action.

Operationally, the framework suggests asking four questions before changing a boundary. What decision-relevant information is lost at the interface? What delay is introduced by the current boundary? What monitoring or misalignment cost would a new boundary create? Can a software agent or protocol preserve enough state to avoid full organizational coalescence?

Organizational agility is the rate at which decision-relevant information becomes valuable action. Software agents matter because they can change the latency, fidelity, and cost of coordination. Their importance is not simply that they automate tasks. It is that they alter where boundaries should be drawn: inside firms, across teams, between vendors, and between human and machine actors. They are boundary-shifting infrastructure.

The next generation of AI-native companies will be designed not merely around people using AI tools, but around software-mediated sensing, routing, decision, and execution loops. The purpose of Coase–Information Theory is to provide a language for designing and measuring those loops.

Acknowledgements

The author thanks Daniel McGrogan for early conversations that helped shape the framing of this work. Any errors are the author's own.

References

[1] Coase, R. H. (1937). The nature of the firm. *Economica*, 4(16), 386–405. doi:10.1111/j.1468-0335.1937.tb00002.x.

[2] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3), 379–423; 27(4), 623–656. doi:10.1002/j.1538-7305.1948.tb01338.x; doi:10.1002/j.1538-7305.1948.tb00917.x.

[3] Blackwell, D. (1953). Equivalent comparisons of experiments. *Annals of Mathematical Statistics*, 24(2), 265–272. doi:10.1214/aoms/1177729032.

[4] Wyner, A. D., and Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22(1), 1–10. doi:10.1109/TIT.1976.1055508.

[5] Wyner, A. D. (1978). The rate-distortion function for source coding with side information at the decoder—II: General sources. *Information and Control*, 38(1), 60–80.

[6] Hayek, F. A. (1945). The use of knowledge in society. *American Economic Review*, 35(4), 519–530. American Economic Association reprint.

[7] Simon, H. A. (1947). *Administrative Behavior: A Study of Decision-Making Processes in Administrative Organization*. Macmillan.

[8] Simon, H. A. (1962). The architecture of complexity. *Proceedings of the American Philosophical Society*, 106(6), 467–482.

[9] March, J. G., and Simon, H. A. (1958). *Organizations*. With the collaboration of H. Guetzkow. Wiley.

[10] Cyert, R. M., and March, J. G. (1963). *A Behavioral Theory of the Firm*. Prentice-Hall.

[11] Marschak, J., and Radner, R. (1972). *Economic Theory of Teams*. Cowles Foundation Monograph 22. Yale University Press.

[12] Arrow, K. J. (1974). *The Limits of Organization*. W. W. Norton.

[13] Brooks, F. P., Jr. (1975). *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley.

[14] Thompson, J. D. (1967). *Organizations in Action: Social Science Bases of Administrative Theory*. McGraw-Hill.

[15] Galbraith, J. R. (1973). *Designing Complex Organizations*. Addison-Wesley.

[16] Williamson, O. E. (1985). *The Economic Institutions of Capitalism: Firms, Markets, Relational Contracting*. Free Press.

[17] Malone, T. W., and Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), 87–119. doi:10.1145/174666.174668.

[18] Radner, R. (1993). The organization of decentralized information processing. *Econometrica*, 61(5), 1109–1146.

[19] Bolton, P., and Dewatripont, M. (1994). The firm as a communication network. *Quarterly Journal of Economics*, 109(4), 809–839. doi:10.2307/2118349.

[20] Van Zandt, T. (1999). Real-time decentralized information processing as a model of organizations with boundedly rational agents. *Review of Economic Studies*, 66(3), 633–658.

[21] Garicano, L. (2000). Hierarchies and the organization of knowledge in production. *Journal of Political Economy*, 108(5), 874–904.

[22] Alonso, R., Dessein, W., and Matouschek, N. (2008). When does coordination require centralization? *American Economic Review*, 98(1), 145–179. doi:10.1257/aer.98.1.145.

[23] Conway, M. E. (1968). How do committees invent? *Datamation*, 14(4), 28–31. `https://www.melconway.com/Home/pdf/committees.pdf`.

[24] Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12), 1053–1058. doi:10.1145/361598.361623.

[25] Henderson, R. M., and Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35(1), 9–30. doi:10.2307/2393549.

[26] Baldwin, C. Y., and Clark, K. B. (2000). *Design Rules, Volume 1: The Power of Modularity*. MIT Press. doi:10.7551/mitpress/2366.001.0001. `https://mitpress.mit.edu/9780262538206/design-rules-volume-1/`.

[27] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P. N., Inkpen, K., Teevan, J., Kikin-Gil, R., and Horvitz, E. (2019). Guidelines for human-AI interaction. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–13. doi:10.1145/3290605.3300233.

[28] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations*. `https://openreview.net/forum?id=WE_vluYUL-X`.

[29] Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 68539–68551. doi:10.52202/075280-2997.

[30] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*. doi:10.48550/arXiv.2308.08155.

[31] Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. (2024). SWE-agent: Agent-computer interfaces enable automated software engineering. *arXiv preprint arXiv:2405.15793*. doi:10.48550/arXiv.2405.15793.

[32] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22. doi:10.1145/3586183.3606763.