SENTIENT REVIEW

Volume 01 · Publication № 03 · Published

The Agentic Enterprise

How Delegated Software Changes Coordination, Management, and Firm Boundaries

> *What happens when organizational coordination itself becomes partly computational?*

VOLUME	Volume 01
PUBLICATION	Publication № 03
PUBLICATION STATUS	Published
RESEARCH DOMAIN	Agentic Enterprise
EDITOR	Fritzie M. Alex

Topics

Computational Coordination · Delegated Software · Institutional Memory · Management · Firm Boundaries

Canonical Edition

https://sentient-review.com/publications/the-agentic-enterprise

Contents

Part I

The Shift

Chapter 01

Organizations Already Compute

The enterprise did not wait for generative AI to begin computing its work. Chapter 01 establishes what organizations already compute, where that computation reaches its boundary, and what may change as software becomes capable of interpreting context and acting within it.

The enterprise did not wait for generative AI to begin computing its work.

A purchase order moves through approval rules. A payment system checks conditions before releasing funds. A workflow assigns cases, records decisions, and marks obligations complete. Databases preserve organizational state; applications enforce permissions; reporting systems turn activity into measures people use to manage.

These systems already coordinate people and resources. They encode selected parts of organizational life into records, rules, interfaces, and processes. The claim that software may change enterprise coordination must begin here. Otherwise, “the computational enterprise” sounds like a prediction built on a false premise: that coordination was previously human alone.

The more precise question is what kinds of coordination organizations have been able to encode, and what may change as software becomes more capable of interpreting context and acting within it.

01 · What existing systems compute

Most enterprise systems work well when the relevant objects, states, and rules can be defined in advance. A payment can be released when specified conditions are met. A case can be routed according to its category. A service level can be measured against a recorded timestamp.

This form of computation is powerful because the organization has made the work legible. It has defined the relevant information, the allowed transitions, and the conditions for completion. People still create the policies, resolve exceptions, and accept responsibility, but software can execute and record much of the routine.

The boundary appears when work departs from the model. A customer request may combine several categories. A supplier delay may affect commitments held in separate systems. A policy may conflict with a contract or with the facts of an unusual case. The organization then relies on people to reconstruct context, interpret intent, negotiate a response, and update the systems afterward.

That work is still coordinated through software, but software often records the result rather than maintaining the whole process.

02 · What may be changing

Current agent systems can interpret language, use tools, and perform bounded sequences of actions. This does not make them organizational authorities. It does make a wider range of work technically approachable: tasks where the inputs are variable, relevant context is distributed, or the next step depends on interpretation.

An agent might assemble the evidence around a delayed delivery, identify which commitments could be affected, and prepare options for an owner. With explicit permission, it might request missing information or update a workflow. Whether such a system improves coordination depends on the quality of its context, the limits on its authority, the reliability of its actions, and the effort required to supervise and repair it.

This is an emerging architectural possibility, not proof that enterprises can safely delegate open-ended coordination at scale.

03 · The continuity and the change

The continuity is that organizations have always used computation to coordinate work. The possible change is that software may participate in less fully specified coordination: interpreting context, maintaining a commitment across systems, and advancing selected actions between human decisions.

The distinction matters. If every increase in software capability is described as a wholly new organizational era, the analysis overstates what has changed. If agents are treated as merely another interface to existing automation, it may miss the possibility that software can take on more of the interpretive work between defined steps.

The research question is therefore bounded: which coordination activities become computationally maintainable when software can interpret context and act within delegated limits? The answer must be established by evidence about complete outcomes, including specification, oversight, verification, exceptions, failure, and recovery.

The enterprise already computes. The future question is where computation may extend next, and what authority and accountability that extension requires.

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/01-organizations-already-compute

Chapter 02

From Tools to Delegated Participants

This chapter is in development. The canonical edition will be reissued when its copy is published.

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/02-from-tools-to-delegated-participants

Part II

The Coordination System

Chapter 03

The Coordination Bill

Coordination has a bill — paid in discovery, interpretation, commitment, monitoring, resolution, and repair. Chapter 03 makes that cost visible and keeps the question empirical: whether agents reduce the bill is a testable proposition, not an assumption.

Coordination has a bill. Organizations pay it in time spent finding the right person, translating between systems, clarifying commitments, checking progress, resolving exceptions, and repairing misunderstandings. Much of that cost is scattered across meetings, messages, rework, delay, and management attention, so it rarely appears as one line item.

The cost is still real. A team waiting for a decision carries idle capacity. A customer promise assembled from conflicting records creates risk. A manager who repeatedly reconstructs project status is performing coordination work, even when no system records it as such.

01 · What the bill contains

The coordination bill includes at least six kinds of work:

- Discovery: finding the relevant people, records, and dependencies.
- Interpretation: reconciling different terms, assumptions, and accounts.
- Commitment: specifying who owes what outcome, under which conditions.
- Monitoring: tracking progress and identifying when a commitment is at risk.
- Resolution: deciding what to do when evidence, authority, or incentives conflict.
- Verification and repair: confirming outcomes and correcting failures.

These costs interact. Poorly specified work increases monitoring. Fragmented context increases interpretation. Weak authority boundaries increase review and dispute. A handoff that appears inexpensive at one point may create delay or rework downstream.

02 · Why agents make the bill newly relevant

Agents may reduce some coordination costs by gathering information, tracking state, requesting missing inputs, and preparing routine actions. They may also add costs: integrating systems, defining delegated authority, reviewing outputs, handling exceptions, and correcting errors.

The economic question is not whether an agent can perform a coordination task. It is whether the total arrangement lowers the cost of an accepted outcome while preserving quality and accountability.

A pilot that counts messages sent or tasks completed can therefore mislead. It may show faster routing while shifting verification to another team. It may reduce meeting time but increase exception repair. Evaluation needs a baseline for the full process: labor, elapsed time, waiting, error, rework, oversight, and the value of outcomes accepted by the relevant parties.

03 · Coordination costs shape organization

Coordination is not only administrative overhead. It helps determine which work an organization performs internally, which it buys from others, and how responsibilities are divided. When coordinating across a boundary becomes costly, a firm may bring work inside or create a dedicated management layer. When information moves more easily, other arrangements may become viable.

This does not mean that lower coordination cost automatically produces smaller firms or flatter structures. Other costs also matter: expertise, bargaining power, quality control, risk, investment, and the need for shared capabilities. Agentic systems may reduce one part of the bill while increasing another.

The direction and scale of change remain empirical questions. They depend on which coordination costs fall, who captures the savings, and whether the cost of oversight and failure grows more slowly.

04 · Make the bill visible

Before automating coordination, identify where the work occurs and who pays for it. Trace one recurring commitment from request to acceptance. Record how much effort goes into clarification, waiting, handoffs, review, exceptions, and repair. Include work performed by customers, suppliers, and downstream teams.

Then compare the current arrangement with a bounded pilot. Measure accepted outcomes, not activity volume. Separate labor saved from labor shifted. Track failures and recovery. A lower coordination bill is valuable only when the organization still meets its obligations.

The enterprise’s coordination bill may become more visible as software takes on parts of the work. Whether it becomes smaller is a testable proposition, not an assumption.

Evidence & Sources

01 **The Interdisciplinary Study of Coordination**
T. W. Malone & K. Crowston, ACM Computing Surveys 26(1), 87–119 · 1994 · Academic Research
Source Note 02 of R-01 · The Computational Enterprise. Coordination as the management of dependencies that recur across organizational settings.
https://sentient-review.com/research/the-computational-enterprise

02 **Designing Complex Organizations**
J. R. Galbraith, Addison-Wesley · 1973 · Academic Research
Source Note 03 of R-01 · The Computational Enterprise. Organizational structure as a response to

information processing requirements.
https://sentient-review.com/research/the-computational-enterprise

03 **The Nature of the Firm**
R. H. Coase, Economica 4(16), 386–405 · 1937 · Academic Research
Source Note 01 of R-01 · The Computational Enterprise. The comparative cost of directing work internally and contracting for it in the market.
https://sentient-review.com/research/the-computational-enterprise

Read online: https://sentient-review.com/publications/the-agentic-enterprise/03-the-coordination-bill

Chapter 04

The Heterogeneous Organization

An enterprise is a collection of domains with different responsibilities, records, vocabularies, incentives, and rights to decide. Chapter 04 argues that visibility does not settle meaning and presents federation as a design direction: coordination across distinct sources of knowledge and authority.

An enterprise is not one system with many departments attached. It is a collection of domains with different responsibilities, records, vocabularies, incentives, and rights to decide.

Finance may recognize a milestone when evidence satisfies a revenue policy. Engineering may call the work complete when a release is deployed. Security may withhold approval until an exception is closed. A customer may still judge the result against a service commitment. These accounts can refer to the same event and still differ for good reasons.

The differences are structural. Each domain sees the organization through the work it must perform and the risks it must manage. A single view can help people orient themselves, but it cannot make every underlying definition, authority, or interest identical.

01 · The limits of a unified picture

An enterprise data model can reconcile selected records. A shared dashboard can display multiple measures. An agent can retrieve information from several systems and summarize it. These capabilities improve visibility, but visibility does not settle meaning.

If “complete” means deployed to one group and accepted by another, combining both into one status hides a decision. If a contract says an outcome is due while an operational system records a blocker, the conflict is itself important context. If a central model silently chooses one source, it may turn data integration into unauthorized governance.

The risk grows when a fluent summary is treated as an authoritative account. A model can make heterogeneous evidence look coherent without proving that its interpretation is legitimate.

02 · Federation as a design direction

A plausible architecture preserves domain authority while enabling coordination across domains. Each domain maintains the records and decisions for which it is responsible. Shared commitments provide a place to coordinate dependencies, outcomes, and handoffs. Agents can carry requests, assemble relevant evidence, and

identify conflicts, subject to scoped permissions.

This is federation: coordination across distinct sources of knowledge and authority, rather than consolidation into one presumed intelligence. It does not require every domain to expose all its data. Participants can share the minimum context needed for a commitment, with provenance and permitted use attached.

The shared object is not a universal truth record. It is a coordination surface: which outcome is sought, who is involved, what each party has asserted, what evidence supports the assertions, and which decision remains open.

03 · Translation without erasure

Federated coordination still needs translation. Domains may use different terms for similar events or the same term for different conditions. Agents can help map these differences, but a mapping should be visible and contestable when it affects a consequential action.

A system might represent “release deployed,” “security accepted,” and “customer accepted” as separate claims linked to one implementation commitment. It can show that deployment is complete while acceptance remains unresolved. The architecture preserves distinctions that matter to the decision.

Some translations can be standardized. Others require domain owners to decide what evidence is sufficient or which policy governs. The system should expose where translation ends and judgment begins.

04 · The emerging challenge

Today, organizations integrate systems, publish shared data, and use access controls to coordinate across domains. The emerging challenge is connecting these mechanisms to agents that can interpret context and initiate action. Such agents need to know not only what information is available, but whose authority it represents and how conflicting claims should affect their permitted next steps.

A federated design may add integration work and create more visible negotiation over definitions. That cost may be worthwhile when it prevents hidden assumptions from becoming enterprise-wide decisions. Whether it improves coordination must be tested against real commitments and their outcomes.

The heterogeneous organization is the starting condition. The computational question is whether software can help its domains coordinate without pretending their knowledge, interests, and authority have become one.

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/04-the-heterogeneous-organization

Chapter 05

Work as an Explicit Commitment

Task records capture activity, but the central obligation often stays implicit. Chapter 05 presents the commitment as an explicit coordination object — outcome, accountable parties, authority, evidence, and exception paths — and keeps the discipline conditional: formalize where it improves shared understanding enough to justify the effort.

Organizations often represent work as tasks, tickets, projects, or workflow steps. Those records help people coordinate, but they can leave the central obligation implicit: what outcome is owed, to whom, and what will count as acceptance?

A commitment makes that obligation explicit. It connects an intended outcome to accountable parties, conditions, evidence, dependencies, and a path for resolving exceptions. This matters when work crosses teams or systems, because each participant needs enough shared understanding to act without assuming that everyone means the same thing.

01 · From activity to outcome

A task records activity: review a request, deploy a change, contact a supplier. A commitment describes the result those activities are meant to achieve: restore a service to an agreed level, deliver an accepted product, or provide a decision by a stated date.

Activities can be completed while the outcome remains unmet. A ticket may be closed while the customer still cannot use the system. A release may be deployed while security acceptance remains open. Measuring task completion alone can make coordination appear successful when the underlying obligation is not.

An explicit commitment identifies the acceptance condition and the authority that can confirm it. It can also represent partial completion, changed scope, or unresolved evidence without forcing a premature “done” state.

02 · The coordination object

For computational coordination, a commitment can serve as a shared object across domains. It need not replace each domain’s own records. Instead, it connects them around a specific outcome:

- the outcome sought and its acceptance conditions
- the accountable owner and participating parties
- the authority and constraints that govern action
- relevant dependencies, deadlines, and current state

- evidence required to assess progress and completion
- exception, change, and dispute paths

This structure gives people and software a common reference while allowing domains to retain their own systems and decision rights. It also makes changes legible. If a supplier revises a delivery date, the commitment can show which downstream outcomes may be affected and who must decide what changes.

03 · Why explicitness matters for agents

An agent needs more than a broad objective such as “resolve the customer issue.” It needs a defined scope, permitted actions, relevant context, constraints, and a way to determine when the outcome is accepted. Otherwise, the agent may optimize a proxy: close the case, send a response, or clear a queue, even when the customer’s obligation remains unresolved.

Explicit commitments do not eliminate ambiguity. They reveal it. If participants disagree on acceptance, the system can record the competing conditions and route the question to the right authority. If no one owns the outcome, that gap becomes visible before automation accelerates the work.

This is an emerging design direction. Organizations already use contracts, service levels, work orders, and project records to express parts of a commitment. Connecting these representations across domains, and making them usable by delegated software, remains difficult.

04 · The discipline of a commitment

Not every activity needs a formal commitment record. The structure is most useful where work crosses a boundary, has material consequences, or depends on several parties. Over-specifying routine work can create administrative cost. Under-specifying consequential work invites confusion, conflict, and concealed risk.

The test is whether an explicit commitment improves shared understanding enough to justify the effort of maintaining it. A pilot should compare the time to acceptance, outcome quality, exceptions, rework, and coordination effort with the current process.

When work becomes legible as a commitment, agents may help maintain its state and move routine coordination forward. The people and institutions involved still determine which obligations matter, what evidence is sufficient, and who can accept the result.

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/05-work-as-an-explicit-commitment

Chapter 06

An Institutional Memory That Can Be Wrong

Delegated software acts through context, which makes organizational memory part of the control path. Chapter 06 sets out what memory must preserve — provenance, validity, correction paths, and surviving disagreement — for an emerging coordination architecture.

Delegated software acts through context. It retrieves policies, records, decisions, and prior cases, then uses them to interpret a request or choose a next step. That makes organizational memory part of the control path.

A system can retrieve a document accurately and still act on the wrong account of the present. The policy may have expired. The record may describe a different customer or product. A summary may omit a qualification. A past decision may be relevant precedent without being authority for the current case.

The enterprise therefore needs more than searchable memory. It needs memory whose origin, scope, validity, and status remain visible when software uses it.

01 · A memory is a claim with conditions

Information used in coordination should retain enough structure to answer several questions: What is being asserted? Which source supports it? Who owns that source? When was it true? What cases does it cover? Is it a verified record, an inference, a proposal, or a disputed interpretation?

These distinctions matter because enterprise knowledge is not uniform. A signed contract and a draft policy can both be retrieved. A manager’s note and a system of record can both describe a project. Their presence in the same context window does not give them equal authority.

A future coordination architecture should preserve these differences through retrieval, summarization, and handoff. When an agent passes context to another agent, it should carry provenance and uncertainty along with the claim. Otherwise, each transformation can make a qualified statement appear more definitive.

02 · Time changes truth

Many organizational facts are time-dependent. A person’s role changes. An approval expires. A supplier’s capacity shifts. A customer commitment is amended. Even accurate information may become unsafe when used outside its validity period.

Memory systems should therefore represent time explicitly: when a claim became valid, when it was last confirmed, and when it should be reviewed or expire. Some claims need continuous refresh; others may remain useful as historical evidence. A system should distinguish “this was true at the time” from “this is currently true.”

That distinction supports both action and learning. A prior incident report may help explain why a decision was made without governing the next decision. A superseded policy remains part of institutional history but should not silently control current execution.

03 · Correction must follow the claim

Correcting a source does not always correct the memory built from it. A summary may be cached. A downstream agent may have copied the claim. A decision may already have been made using it.

A reliable design needs a correction path that can locate derived uses, mark stale representations, and identify consequential actions that may need review. The goal is not to pretend that every error can be erased. It is to make the propagation of an error traceable and its correction actionable.

That requires records of which claims informed which decisions. The record should show the evidence considered, the authority exercised, and any uncertainty known at the time. This makes later review possible without rewriting the historical record.

04 · Disagreement should remain available

Two authorized sources may disagree. Operations may report that an order shipped; the customer may dispute receipt. Finance may treat a milestone as achieved; delivery may say acceptance is still pending. A memory system that resolves the conflict by selecting whichever source is easiest to retrieve can convert a data problem into an unauthorized decision.

Instead, the system should preserve competing claims, explain their consequences, and route the conflict to the responsible authority. Once resolved, the decision should be stored with its scope and rationale. Future systems can use that resolution while retaining the fact that it applied under particular conditions.

This can make coordination slower in the moment. It can also prevent a confident but unsupported answer from triggering a costly action.

05 · Memory as governed infrastructure

Today, organizations rely on source systems, document management, data catalogs, records policies, and human expertise to maintain institutional knowledge. Emerging agent systems increase the importance of connecting these controls to retrieval and action. The research does not establish that enterprises have solved this problem with a unified memory layer.

The practical design is likely to be distributed. Domains retain ownership of their records and policies. Shared coordination systems retrieve permitted claims with provenance, validity, and dispute status. Corrections flow to dependent uses where possible. Agents act only when the available memory satisfies the conditions set for

the commitment.

A memory system earns trust not by appearing comprehensive, but by making its limits inspectable. It must show what it knows, where that knowledge came from, when it may no longer apply, and when the organization has not resolved what is true.

The central design test is consequential: when an agent relies on an outdated or mistaken claim, can the organization trace its use, correct the affected memory, and reconsider the decisions that followed?

Evidence & Sources

01 **PROV-DM: The PROV Data Model**
W3C Recommendation · 2013 · Technical Publication
Source Note 05 of R-01 · The Computational Enterprise. A standard model for representing the origin and derivation of a claim.
https://sentient-review.com/research/the-computational-enterprise

02 **Agentic AI Threats and Mitigations**
OWASP · Accessed September 2026 · Technical Publication
Source Note 06 of R-01 · The Computational Enterprise. Memory and context poisoning as a distinct agentic risk class.
https://sentient-review.com/research/the-computational-enterprise

Read online: https://sentient-review.com/publications/the-agentic-enterprise/06-an-institutional-memory-that-can-be-wrong

Part III

The Institutional Consequence

Chapter 07

Beyond the Enterprise Brain

The enterprise brain is an attractive image for agentic coordination, and a poor institutional model. Chapter 07 distinguishes information access from decision authority, examines what centralization does well, and presents a federated intelligence architecture in which authority remains attached to roles, policies, and accountable processes.

The enterprise brain is an attractive image for agentic coordination: one intelligence with access to organizational knowledge, able to understand the whole and direct its parts.

It is also a poor institutional model.

An enterprise contains knowledge that is distributed, uneven, and sometimes contested. It contains people and systems with different responsibilities, incentives, and rights to decide. A central system can aggregate information. It cannot make those differences disappear, and access to a record does not confer authority over the decision that record informs.

The architectural question is how to coordinate across intelligence without pretending that one intelligence owns the enterprise.

01 · Information is not authority

A central model may produce a broad view by retrieving from many systems. That view can help leaders identify dependencies and patterns. It can also obscure which claims are authoritative, which are inferred, and which remain disputed.

Suppose one system says a customer deployment is complete, another records an unresolved security exception, and the contract defines acceptance through customer confirmation. A central summary may be useful. A central agent deciding that the commitment is complete would cross from synthesis into institutional judgment.

The distinction is fundamental: information access supports a decision; it does not authorize one. An intelligence layer can help participants understand the situation. Authority remains attached to roles, policies, contracts, and accountable processes.

02 · What centralization can do well

Centralized systems have real advantages. They can provide common identity, shared search, consistent definitions, cross-domain visibility, and economies of scale. Some enterprise controls need a central owner. Security policies, financial reporting, and organization-wide risk limits cannot always be left to local interpretation.

The problem is not centralization itself. It is treating central synthesis as complete understanding or legitimate command. A model trained or prompted to produce one answer may suppress meaningful uncertainty. A central agent with broad access may become a point of failure or an attractive target. A shared semantic layer may encode the assumptions of the groups that designed it.

The design challenge is to centralize what benefits from common governance while preserving domain ownership where context and decision rights reside.

03 · A federated intelligence architecture

A plausible alternative separates three functions:

Domain intelligence interprets local records and conditions under the stewardship of the relevant function.

Shared coordination connects commitments, dependencies, and requests across domains, carrying only the context needed for the work.

Decision authority remains with the role or process authorized to accept risk, change obligations, or resolve disputes.

These functions can be supported by centrally governed infrastructure. Federation does not require isolated systems or incompatible models. It requires explicit boundaries around knowledge, representation, and decision.

A coordination layer could show that engineering reports deployment complete, security reports a remaining exception, and customer acceptance is pending. It can identify the commitment affected and route the unresolved question. It should not collapse three claims into one status unless the organization has defined who can make that determination and on what evidence.

04 · Intelligence as a constitutional question

As agents gain access to more systems and the ability to take more actions, the architecture begins to distribute or concentrate institutional power. Who can see which evidence? Which definitions become standard? Who can direct software to act? Who can challenge its account? Who is accountable when a local decision produces an enterprise-wide consequence?

These are not only questions of model design. They concern the institution’s constitution: how authority is granted, constrained, reviewed, and corrected.

A federated design can still centralize power if one group controls the shared context and action layer. A central system can remain accountable if its scope is narrow, its decisions reviewable, and its relationship to domain authority explicit. The topology alone does not settle the governance question.

05 · The conditional consequence

Today, enterprises combine central platforms with domain-specific systems, controls, and expertise. Emerging agent architectures may connect these through shared context and delegated action. A plausible future is a federation in which domain intelligences contribute evidence and bounded capabilities to cross-enterprise commitments.

Whether this improves coordination depends on how well the federation preserves meaning, authority, and recourse. The “enterprise brain” suggests that more integration naturally produces better judgment. The stronger hypothesis is that intelligence becomes organizationally useful only when its sources, limits, and rights to act remain visible.

The enterprise may become more computationally coordinated without becoming a single mind. Its intelligence would be a governed relationship among domains, not a central model’s claim to know the whole.

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/07-beyond-the-enterprise-brain

Chapter 08

Management After Machine Coordination

Management is partly a response to coordination costs. Chapter 08 examines what may change when software maintains routine coordination: a plausible reallocation of managerial attention, accountability structures under delegation, and the risks of surveillance, workload, and metric pressure.

Management is partly a response to coordination costs. Managers gather information, align work, allocate capacity, resolve exceptions, and make decisions when rules or agreements do not settle what should happen. If software maintains more of the routine coordination, management work may change. Which tasks change, and whether management becomes more effective, remain open questions.

01 · Coordination work and managerial work

Some management activity is primarily informational: requesting status, reconciling updates, identifying blocked dependencies, and ensuring that commitments have owners. Agents may reduce the effort required for these tasks by maintaining work state and surfacing exceptions across systems.

Other management work is judgmental and institutional. Leaders set priorities when resources conflict, decide which risks the organization will accept, negotiate tradeoffs, establish goals, and remain answerable for outcomes. Better information may support these decisions, but it does not determine them.

A computational system might show that three commitments compete for the same specialist, or that one customer promise depends on an unresolved security exception.

It can make the choice visible sooner. Someone with legitimate authority still has to decide what takes precedence and explain the consequence.

02 · The manager’s changing operating surface

If the emerging architecture works as intended, managers could spend less time reconstructing routine status and more time on:

- setting outcomes and acceptable constraints
- allocating capacity across competing commitments
- resolving exceptions that exceed delegated authority
- examining evidence and challenging system interpretations
- reviewing recurring failures and redesigning the process
- maintaining accountability for decisions and results

This is a plausible reallocation, not a guaranteed productivity dividend. New systems require monitoring, policy maintenance, exception review, and audit. If they create more alerts than managers can assess, they can increase rather than reduce coordination burden.

The transition may also change the span and reach of management. A manager could oversee more commitments because software maintains their state. That could improve visibility, or create an unrealistic workload in which one person is nominally accountable for decisions they cannot meaningfully inspect.

03 · Accountability under delegation

Delegation does not make accountability disappear; it changes where failure can occur. An agent may misread a request, act on incomplete context, or apply a valid rule to the wrong case. A manager may approve a design that gives the system too much authority, or ignore a known failure pattern. A domain owner may provide inaccurate or stale context.

A credible operating model names responsibility at each level: who sets objectives, who grants authority, who maintains context, who reviews exceptions, and who answers for the outcome. The record should let an accountable person reconstruct what the agent saw, what action it took, and which decision rights applied.

Human review must be meaningful. A person asked to approve a high volume of poorly explained recommendations may become a rubber stamp. Oversight should fit the stakes and give reviewers time, evidence, and authority to intervene.

04 · The risks of computational management

Machine coordination can intensify control as well as reduce administrative work. Continuous monitoring may turn every action into a performance signal. Narrow outcome metrics can reward what is easy to count and penalize care, learning, or necessary escalation. Managers may use automation to increase work pace while shifting the burden of judgment and failure onto employees.

These effects are choices in system design and governance. Organizations decide which activity to measure, how metrics affect people, who can challenge an evaluation, and whether efficiency gains are shared or used only to raise targets. A system that routes work faster can still produce a worse institution.

05 · What should be tested

A pilot should measure more than manager hours saved. It should examine decision quality, accepted outcomes, capacity use, exception load, employee experience, customer impact, and the effort required to supervise and correct the system. It should also test whether managers can contest agent recommendations and whether repeated exceptions lead to changes in policy or design.

The near-term opportunity is specific: use software to maintain routine coordination so managers can see commitments and exceptions more clearly. The longer-term possibility is a different distribution of managerial attention and organizational control. Neither the disappearance of management nor its automatic elevation follows from the technology.

Management after machine coordination depends on what the institution delegates, what it keeps accountable, and how it treats people affected by the resulting system.

Evidence & Sources

01 **AI and Organizations Lab**
Stanford Graduate School of Business · Accessed September 2026 · Academic Research
Source Note 07 of R-01 · The Computational Enterprise. How AI enhanced information processing changes managerial roles and the allocation of human work.
https://sentient-review.com/research/the-computational-enterprise

02 **AI Risk Management Framework: postdeployment monitoring**
National Institute of Standards and Technology · Accessed September 2026 · Institutional Research
Source Note 08 of R-01 · The Computational Enterprise. Monitoring cadence, validation, auditing, appeal, override, recovery, and decommissioning.
https://sentient-review.com/research/the-computational-enterprise

Read online: https://sentient-review.com/publications/the-agentic-enterprise/08-management-after-machine-coordination

Chapter 09

Who Is Allowed to Decide

When software can interpret context and act, authority becomes visible in a new way. Chapter 09 sets out delegation as a governed act — principal, jurisdiction, action, conditions, escalation, review — and frames the distribution of decision rights as a constitutional choice.

When software can interpret context and act, authority becomes visible in a new way. An agent may technically be able to approve a change, send a commitment, or alter a record. The institution still has to decide whether it is permitted to do so, on whose behalf, and under what conditions.

The question is not whether an agent “makes a decision” in a human sense. It is who authorized the system’s action and who remains accountable for its consequences.

01 · Authority has a jurisdiction

A person’s role may grant authority over one class of decisions but not another. A procurement manager can approve within a budget threshold, while a security officer can accept certain risks, and a customer owner can amend a delivery commitment. Agents need comparably specific boundaries.

Delegation should identify:

- Principal: Who grants the authority?
- Jurisdiction: Which domain, cases, systems, and parties are covered?
- Action: What may the agent read, recommend, prepare, or execute?
- Conditions: Which thresholds, evidence, and policies apply?
- Escalation: When must the agent pause and route the matter?
- Review: Who can inspect, challenge, revoke, or amend the delegation?

These boundaries should be enforced at the point of action. A prompt that tells an agent to act carefully cannot substitute for permissions that prevent an unauthorized transaction.

02 · The delegation problem

Organizations already delegate authority through roles, policies, contracts, and controls. Agentic systems add complexity because the delegated actor can interpret a request and select among actions at runtime. The action may not have been enumerated as a fixed rule, even if its scope and constraints were.

This creates a need to distinguish authority to pursue an objective from authority to make every decision that might help achieve it. “Resolve the customer issue” cannot imply permission to change contract terms, disclose restricted information, or

promise a refund. The objective must be bounded by explicit action rights and stop conditions.

A system may prepare a recommendation without having authority to execute it. It may execute routine actions under defined conditions while escalating exceptions. Different levels of delegation should be tied to the consequences, reversibility, and uncertainty of the action.

03 · Human accountability and meaningful oversight

Keeping a human “in the loop” is not enough if the person lacks time, context, or power to reject what the system recommends. Review must be designed so that the accountable owner can understand the evidence, see uncertainty, challenge the interpretation, and stop or reverse action where possible.

Responsibility should not collapse into a vague claim that “the human approved it.” The organization must identify who designed the delegation, who granted authority, who maintained the relevant context and policy, and who owned the underlying commitment. These responsibilities may sit with different people.

This is a governance problem as much as a technical one. Audit records can show what happened; they cannot alone establish that the authority was appropriate or that the institution has met its obligations.

04 · Contestability and correction

People affected by an agent’s action need a path to question it. A customer should be able to challenge an incorrect decision. An employee should be able to correct a misleading record that shaped an evaluation. A domain owner should be able to contest an agent’s interpretation of policy.

Contestability requires more than an appeal form. The organization must preserve the relevant evidence and decision path, identify someone empowered to reconsider the action, and correct downstream records when needed. The process should also feed recurring errors into policy, context, and system changes.

Some actions cannot be fully reversed. A message may have been sent; information may have been disclosed; a time-sensitive opportunity may have passed. The design should account for those consequences before delegation, rather than relying on post hoc review.

05 · A constitutional choice

A future computational enterprise will distribute authority through technical controls, operating policies, and human roles. The result could concentrate discretion in a central platform, distribute it among domain agents, or combine shared guardrails with local authority. Each choice carries consequences for power, accountability, and recourse.

Today’s systems provide components such as identity, access control, approval workflows, and audit logs. Extending these into coherent, context-sensitive delegation for agents remains an emerging design and governance challenge.

The executive question is not simply which actions to automate. It is which authority to delegate, to whom, within which jurisdiction, and with what rights of review and challenge.

Evidence & Sources

01 **Standards work on AI agent identity, authorization, and security**
National Institute of Standards and Technology · Accessed September 2026 · Institutional Research Source Note 04 of R-01 · The Computational Enterprise. Guidance on treating agents with the precautions applied to privileged users.
https://sentient-review.com/research/the-computational-enterprise

Read online: https://sentient-review.com/publications/the-agentic-enterprise/09-who-is-allowed-to-decide

Part IV

The Enterprise Boundary

Chapter 10

Faster Coordination, Faster Failure

Coordination speed has a double edge. Chapter 10 examines how delegated software can extend the reach of an error across workflows, and sets out the containment safeguards — scoped permissions, evidence gates, circuit breakers, reversibility — that keep speed from becoming systemic failure.

Coordination speed has a double edge. When information and decisions move faster, an organization may resolve dependencies sooner. The same speed can spread an incorrect assumption, unauthorized action, or faulty policy across more systems before anyone recognizes the problem.

This is not unique to agents. Distributed software has long shown how shared dependencies and rapid propagation can create failures at scale. Agentic coordination adds a further possibility: systems may interpret changing context and initiate actions across workflows, extending the reach of an error beyond a single predefined transaction.

01 · How failure propagates

Imagine an agent interpreting a supplier update as confirmation that a component is available. It changes a planning record, which triggers a production schedule, which prompts customer delivery commitments. If the supplier meant “available for review” rather than “ready to ship,” each downstream step can be locally reasonable and collectively wrong.

The failure may arise from several sources:

- an ambiguous or stale input
- an incorrect interpretation
- a mistaken mapping between systems
- authority broader than the task requires
- downstream automation treating an unverified state as fact
- a common policy or model error affecting many workflows

The risk is not simply that an agent makes a mistake. It is that the system design lets one mistake become a chain of mutually reinforcing actions.

02 · Speed without containment

Faster handoffs can reduce waiting, but speed alone is not a reliability measure. A system that advances work before evidence is adequate may increase the cost of rollback and repair. A shared model or policy can improve consistency and create a common mode failure. Many agents acting on the same wrong context can multiply

impact rather than provide independent checks.

The relevant design question is therefore not only how quickly a commitment moves. It is how far an action can travel before a person or control can detect, stop, or reverse it.

03 · Bound the propagation path

An emerging coordination architecture should treat the path from interpretation to action as a controlled boundary. Possible safeguards include:

- Scoped permissions: Limit agents to the systems, cases, and actions required.
- Evidence gates: Require stronger confirmation before high impact transitions.
- Rate limits and quotas: Bound the volume and speed of actions.
- Staged rollout: Start with recommendations or low consequence cases, then expand based on evidence.
- Circuit breakers: Pause a workflow when error signals or unusual activity exceed a threshold.
- Reversibility: Prefer actions that can be undone; require review before irreversible commitments.
- Independent checks: Validate critical state changes through another source or control.
- Recovery paths: Preserve records and ownership so teams can contain and repair failures.

These controls have costs. More gates can slow routine work and create approval queues. A safeguard that produces constant false alarms can become noise. The architecture should match controls to consequence, uncertainty, and reversibility rather than apply identical friction everywhere.

04 · Reliability is an operating property

A benchmark result or successful demonstration does not establish safe performance across a live workflow. Reliability depends on the full system: model behavior, context quality, tools, permissions, workflow design, monitoring, human response, and recovery. Each connection can fail differently.

Organizations should test realistic edge cases before widening authority. They should measure not only task success but also mistaken actions, near misses, escalation quality, recovery time, and the reach of failures. Simulation and staged pilots can reveal propagation paths before they affect customers or operations.

05 · The conditional proposition

Today, agents can support bounded tasks, and organizations can add controls around tool access and workflow execution. The emerging challenge is managing chains of

delegated actions across multiple domains while maintaining effective containment.

A plausible future may bring faster coordination and broader machine maintained commitments. Whether it improves organizational performance depends on whether the system can preserve the gains of speed while limiting the reach of error. Reliability is not the absence of mistakes. It is the ability to detect, contain, explain, and recover from them before they become institutional failures.

Evidence & Sources

01 **AI Risk Management Framework: postdeployment monitoring**
National Institute of Standards and Technology · Accessed September 2026 · Institutional Research Source Note 08 of R-01 · The Computational Enterprise. Monitoring cadence, validation, auditing, appeal, override, recovery, and decommissioning.
https://sentient-review.com/research/the-computational-enterprise

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/10-faster-coordination-faster-failure

Chapter 11

Agents Across the Company Boundary

Chapter 11 examines what happens when delegated software meets a counterparty. Cross company agent interaction depends on institutions as much as interfaces — identity, scoped delegation, semantics, commitment, provenance, and recourse — and technical interoperability should never be mistaken for trust.

An agent working inside one company operates within that company’s systems of identity, permission, policy, and accountability. Across a company boundary, those assumptions stop. A supplier’s agent does not inherit a customer’s trust. A counterparty’s software cannot create authority simply by making a convincing request.

The possibility of agent to agent coordination therefore depends on institutions as much as interfaces.

01 · The boundary already has infrastructure

Companies already exchange structured information through application interfaces, electronic data interchange, procurement networks, identity systems, and contractual processes. These mechanisms coordinate defined transactions such as orders, invoices, shipments, and service requests.

Agents could extend this infrastructure into cases that require more interpretation. A supplier agent might explain a delay, provide evidence, identify affected orders, and propose alternatives. A customer agent could compare those proposals with commitments, constraints, and internal priorities. Each could reduce the human effort of assembling the situation.

But a conversation between agents does not by itself establish shared meaning or a binding agreement. The systems need to distinguish a request, a statement, a proposal, an approval, and a commitment. A message that sounds like an offer may not be authorized to bind the company.

02 · What counterparties must establish

Controlled cross company interaction requires answers to several questions:

- Identity: Which organization and principal does the agent represent?
- Delegation: Who authorized it, and for which tasks?
- Disclosure: What information may it receive or reveal?
- Semantics: What do the exchanged terms and evidence mean?
- Commitment: Which action is a proposal, and which creates an obligation?
- Provenance: How can each party inspect the origin and integrity of a claim?

- Recourse: How are errors, disputes, revocation, and liability handled?

These questions belong to separate layers. A transport protocol can deliver a message without establishing its commercial meaning. Authentication can verify an identity without proving that an agent may approve a price change. A signed record can preserve what was agreed without determining whether the agreement is enforceable.

The architecture should avoid treating technical interoperability as institutional trust.

03 · A bounded cross company exchange

A practical starting point is a narrow, reversible workflow. For example, a buyer's agent requests a shipment status and supporting evidence from a supplier's agent. The supplier system returns an authenticated status, source references, and a proposed revised delivery date. The buyer's system checks which orders and customer commitments may be affected, then routes any change requiring approval to an authorized person.

The agents can exchange information and prepare options. A binding change follows the parties' existing authority and contractual process unless they have explicitly agreed to delegate that decision. Both companies retain records of what was requested, disclosed, proposed, approved, and executed.

This arrangement could reduce time spent on routine status gathering. It still depends on compatible terms, dependable evidence, and a clear path when the parties disagree.

04 · The risk of asymmetric delegation

Cross company systems may not share power equally. A dominant buyer could require suppliers to connect to its agent platform, disclose more data, or accept automated terms. A large platform could define the vocabulary and rules that smaller participants must follow. Automation may shift negotiation costs and accountability onto the less powerful party.

Governance must therefore include the commercial relationship. Participants need control over what their agents can disclose and accept, the ability to revoke delegation, and a way to contest automated interpretations. The right to use an agent should not silently become the right to bind another company.

05 · The conditional network

Today's systems support machine readable exchange for defined business processes. Emerging agents may help interpret exceptions and coordinate less structured interaction. A future network of agents across firms is possible, but it would require more than a shared protocol: it would need identity, scoped authority, common enough semantics, evidence, audit, dispute resolution, and workable commercial

terms.

No single “agent trust fabric” should be assumed. It may develop through industry specific contracts, registries, standards, or platform arrangements, or it may remain fragmented. The useful test is whether a specific cross company commitment becomes easier to fulfill with lower total cost and clear recourse when the system is wrong.

The company boundary may become more navigable to software. It remains a boundary of authority, liability, and bargaining power.

Evidence & Sources

01 **Open interoperability standards for agentic systems (MCP, A2A)**
Model Context Protocol; Agent2Agent Protocol, Linux Foundation · Accessed September 2026 · Technical Publication
Source Note 09 of R-01 · The Computational Enterprise. Common interfaces for connecting agents to tools and to one another across platforms.
https://sentient-review.com/research/the-computational-enterprise

02 **AI Agent Standards Initiative**
National Institute of Standards and Technology · Accessed September 2026 · Institutional Research
Source Note 10 of R-01 · The Computational Enterprise. Secure interoperability and agent identity as conditions for trusted adoption.
https://sentient-review.com/research/the-computational-enterprise

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/11-agents-across-the-company-boundary

Chapter 12

Does the Firm Become Smaller

Chapter 12 asks whether agents that lower coordination costs will make firms smaller, larger, or simply different. The answer depends on which costs fall faster — internal administration or market coordination — and the direction remains an empirical, conditional question.

If agents lower the cost of coordination, will companies need fewer employees, fewer management layers, or fewer activities inside the firm?

The question is economically serious. The answer is not self evident.

Firms organize activity internally when directing work through management costs less than contracting for it in the market. Markets also have costs: finding partners, negotiating terms, monitoring delivery, protecting information, and resolving disputes. The boundary of a firm reflects the relative costs of these arrangements, along with control, risk, investment, and the capabilities needed to perform the work.

Agents could change several of these costs at once.

01 · Two directions of change

If agents make external coordination cheaper, a firm may find it easier to work with specialized suppliers or partners. Automated discovery, evidence exchange, and routine negotiation could reduce some of the effort required to transact across company boundaries. That could make outsourcing, partnerships, or project based networks more practical for selected work.

If agents make internal coordination cheaper, a firm may be able to coordinate more activities without adding equivalent management overhead. It could expand its reach, serve more customers, or operate more complex workflows within the same organizational structure.

The same technology can therefore push in opposite directions. The result depends on which costs fall faster: the cost of coordinating internally, or the cost of coordinating through markets and partners.

02 · Coordination is only one cost

A business decision cannot be based on coordination cost alone. External work may increase exposure to supplier failure, data leakage, quality variation, or loss of critical knowledge. Bringing work inside may require investment, specialized talent, and management capacity. Contracting can be difficult when outcomes are uncertain or cannot be specified in advance.

Agents may also create new costs on both sides of the boundary. Internal use requires authority, oversight, context, integration, and recovery. External use requires identity, disclosure limits, compatible semantics, commercial terms, and recourse. If these costs remain high, cheaper message exchange will not necessarily make market coordination efficient.

Bargaining power matters too. A large firm may use agents to coordinate an extended supplier network while retaining strong control over terms. Smaller partners may bear integration costs without gaining the same benefits. A more connected market does not automatically become a more balanced one.

03 · The unit of analysis

The useful question is not “Will firms become smaller?” in the abstract. It is: for which activities does agentic coordination change the relative cost, control, and risk of internal versus external organization?

Start with a specific activity. Estimate its current internal coordination cost, external transaction cost, quality requirements, information sensitivity, and failure consequences. Then test how a bounded agent system changes each. Include the costs of specification, supervision, integration, verification, and dispute resolution. Separate labor removed from labor shifted to another party.

An activity may move across the boundary without changing the overall size of the firm. A company may contract for more specialized work while expanding its platform and control functions. Another may retain execution but reduce administrative coordination. Firm boundaries can shift by activity, capability, and risk rather than contract uniformly.

04 · What evidence could settle it

Evidence should compare real organizational choices before and after agentic coordination is introduced. Did a company bring work inside, move it outside, or change the terms of a relationship? Did the full cost fall after accounting for quality, oversight, risk, and transition? Who captured the savings? Did the company preserve the knowledge and control it needed?

These effects may take years to appear. A pilot that lowers coordination time in one workflow cannot establish that the optimal firm boundary has changed.

Agents could make firms smaller, larger, more networked, or simply different. The economic mechanism is plausible: changes in coordination costs can change organizational choices. The direction remains conditional on the relative costs of markets and internal administration, and on the institutions that govern both.

Evidence & Sources

01 **The Nature of the Firm**
R. H. Coase, Economica 4(16), 386–405 · 1937 · Academic Research
Source Note 01 of R-01 · The Computational Enterprise. The comparative cost of directing work internally and contracting for it in the market.
https://sentient-review.com/research/the-computational-enterprise

02 **The Interdisciplinary Study of Coordination**
T. W. Malone & K. Crowston, ACM Computing Surveys 26(1), 87–119 · 1994 · Academic Research
Source Note 02 of R-01 · The Computational Enterprise. Coordination as the management of dependencies that recur across organizational settings.
https://sentient-review.com/research/the-computational-enterprise

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/12-does-the-firm-become-smaller

Part V

The Conditional Future

Chapter 13

The Autonomous Enterprise on Trial

Chapter 13 treats the autonomous enterprise as a claim to be tested, not a forecast. It defines the unit of autonomy, sets out how a bounded trial should be run and evaluated, and places the burden of proof on the claim.

“Autonomous enterprise” is a useful stress test, but a poor default forecast. The phrase can mean anything from software handling routine tasks to a company operating without human authority. Those are different claims and require different evidence.

Before asking whether an enterprise can become autonomous, define the unit of autonomy. Is it a task, a workflow, a function, or a firm? Which decisions may software make? What outcomes must it achieve? Who can intervene, and who remains accountable?

Without those boundaries, autonomy becomes a slogan rather than a testable proposition.

01 · Test the bounded case first

A serious trial begins with a recurring commitment whose outcome, owner, authority, context, evidence, and recovery path can be specified. The system operates within a narrow scope. It may gather information, propose actions, or execute selected steps. Its authority expands only when performance supports the change.

The trial should include ordinary cases and conditions that challenge the design: incomplete inputs, conflicting records, changed policies, unexpected tool responses, ambiguous requests, and attempts to exceed the delegated scope. It should test whether the system recognizes uncertainty and escalates appropriately, not only whether it succeeds when everything works as expected.

02 · Evaluate the whole system

Model performance is only one component. The system under trial includes models, tools, data, permissions, workflow logic, monitoring, people, and recovery. A strong result on routine cases may be offset by rare but consequential errors or by the human effort required to supervise every action.

The evaluation should measure:

- Outcome quality: Were commitments completed and accepted by the relevant parties?
- Reliability: How often did the system succeed, fail safely, or require correction?
- Authority adherence: Did it stay within its permitted scope?

- Exception handling: Did it recognize missing evidence, conflict, and unusual conditions?
- Recovery: Could people contain errors, restore state, and learn from incidents?
- Economics: What was the full cost, including integration, oversight, verification, and repair?
- Institutional effects: Who gained control, who carried new work, and who could challenge decisions?

Results should be compared with a documented baseline. The trial should also account for selection effects: a system tested only on easy cases may look more autonomous than it would be in the workflow as a whole.

03 · Levels of delegation

Autonomy is not binary. A system can retrieve information, draft a recommendation, prepare an action for approval, execute a reversible step under defined conditions, or manage a bounded workflow with escalation. These levels distribute authority differently.

Moving to a higher level requires more than confidence that the model is usually right. It requires evidence that the full operating system behaves reliably within the intended scope, that controls work under stress, and that accountable owners can intervene. High impact or irreversible decisions may remain subject to human approval even when routine parts of the workflow are delegated.

The organization should be able to reduce authority as well as expand it. Changes in model behavior, context quality, business conditions, or incident patterns may make yesterday’s delegation unsafe.

04 · What would the trial establish?

A successful pilot could show that a bounded process is suitable for a defined form of machine coordination. It would not establish that the function, company, or institution is autonomous. Nor would success in one domain prove that the same design transfers to another with different authority, context, and consequences.

A larger claim would require evidence across diverse functions and operating conditions, along with a clear account of where human judgment remains essential. It would also require answering the institutional question: who sets objectives, adjudicates disputes, assumes responsibility, and can challenge the system?

The autonomous enterprise is therefore on trial. The burden of proof belongs to the claim, not to the people asking whether the system is ready. Its future depends on what bounded deployments demonstrate, what they cost, and what authority organizations choose to delegate.

Evidence & Sources

01 **AI Risk Management Framework: postdeployment monitoring**
National Institute of Standards and Technology · Accessed September 2026 · Institutional Research Source Note 08 of R-01 · The Computational Enterprise. Monitoring cadence, validation, auditing, appeal, override, recovery, and decommissioning.
https://sentient-review.com/research/the-computational-enterprise

Read online:
https://sentient-review.com/publications/the-agentic-enterprise/13-the-autonomous-enterprise-on-trial

Chapter 14

A Constitutional Choice, Not a Technological Destiny

Chapter 14 closes the publication by framing the computational enterprise as a set of institutional choices. It sets out four open trajectories, the constitutional questions each answers, and a discipline for choosing that keeps the future conditional.

The computational enterprise is not a destination built into the technology. It is a set of choices about which work to delegate, what authority to grant, what evidence to trust, how to distribute control, and who remains answerable for outcomes.

The same agent capabilities could support very different institutions. They could make routine coordination more visible and reduce the effort required to fulfill commitments. They could centralize information and discretion in a small number of platforms. They could extend surveillance and work intensity while moving accountability away from those who design the system. They could also be contained to narrow tasks where their value is clear and their consequences limited.

Technology shapes what becomes possible. Institutions determine what becomes legitimate.

01 · Several futures remain open

One future concentrates coordination. A central platform holds broad context, directs agents across domains, and gives leaders a unified operational view. It may improve consistency and speed. It may also create a powerful point of failure, blur domain authority, and make the platform’s definitions difficult to contest.

Another future federates coordination. Domains retain ownership of context and decisions while sharing commitments and evidence through governed interfaces. This can preserve local judgment and make cross-boundary work more legible. It can also introduce negotiation, integration, and semantic costs.

A third future fragments. Each function or vendor deploys agents independently, automating local steps without shared commitments, common authority, or coherent recovery. Local productivity may rise while enterprise coordination becomes harder to understand.

A fourth future remains deliberately bounded. Organizations use agents for narrow, reversible tasks and retain human decision-making for higher consequence work. This may limit some gains while allowing evidence, standards, and institutional practice to mature.

These are design trajectories, not predictions. Real enterprises may combine them across functions or change direction after experience.

02 · Constitutional questions

Each trajectory answers foundational questions, whether leaders make those answers explicit or allow them to emerge by default:

- Who sets the goals agents pursue?
- Who decides what information the system may use?
- Which decisions can be delegated, and within what limits?
- Who benefits when coordination costs fall?
- Who bears the cost of errors, monitoring, and repair?
- How can people affected by an action challenge it?
- Who can inspect, change, or revoke the system’s authority?

These questions determine how power is distributed through technical architecture, organizational roles, contracts, and operating rules. They deserve the same attention as model capability and workflow efficiency.

03 · A discipline for choosing

A responsible approach starts with a specific commitment and a defined problem. It establishes the current coordination cost and outcome quality, then tests whether bounded delegation improves them. It makes authority and evidence inspectable, gives people meaningful recourse, and preserves the ability to pause, reverse, or withdraw delegation.

The choice should remain revisable. A successful pilot does not justify unlimited expansion. A failure does not prove that all machine coordination is unworkable. Each result should change the scope, controls, or design according to evidence.

Leaders should also ask who participates in the decision. Employees, customers, suppliers, and other affected parties may hold information about the consequences that system owners cannot see from operational dashboards alone.

04 · The enterprise people choose

If agents make coordination less costly, the institutional effects will depend on what organizations optimize. They may use the capacity to improve service, increase resilience, expand access, reduce repetitive effort, or raise output expectations. These outcomes are not equivalent, and efficiency measures alone cannot choose among them.

The central question has remained constant: What happens when organizational coordination itself becomes partly computational?

The answer will not be determined by model capability alone. It will be built through thousands of decisions about commitments, authority, context, evidence, management, and economics. Those decisions can be hidden inside defaults or made deliberately, reviewed, and revised.

The agentic enterprise is therefore a constitutional choice. Organizations will decide which parts of coordination become computational, which powers remain human, and what obligations govern the systems in between. The future is conditional because those choices are still open.

Read online: https://sentient-review.com/publications/the-agentic-enterprise/14-a-constitutional-choice-not-a-technological-destiny

Connected Work

Connected Framework

FW 20 · Operationalized

Computational Coordination Architecture

https://sentient-review.com/frameworks/computational-coordination-architecture

Developed Through

FN 22 — The Enterprise Becomes Computational

https://sentient-review.com/field-notes/the-enterprise-becomes-computational

FN 23 — The Enterprise Brain Is the Wrong Metaphor

https://sentient-review.com/field-notes/the-enterprise-brain-is-the-wrong-metaphor

FN 24 — An Institutional Memory That Can Be Wrong

https://sentient-review.com/field-notes/an-institutional-memory-that-can-be-wrong

FN 25 — Management After Machine Coordination

https://sentient-review.com/field-notes/management-after-machine-coordination

FN 26 — The Agentic Boundary of the Firm

https://sentient-review.com/field-notes/the-agentic-boundary-of-the-firm

Connected Practices

Primary Practice — AI-Native Engineering Operating Model

Built By Sentient · https://builtbysentient.com/practice/ai-native-engineering-operating-model

Connected Practice — Agentic AI Architecture, Security & Governance Blueprint

Built By Sentient · https://builtbysentient.com/practice/agentic-ai-architecture-security-governance-blueprint

Connected Practice — Technology & AI Economics Optimization

Built By Sentient · https://builtbysentient.com/practice/technology-ai-economics-optimization

Citation

Sentient Review. (September 2026). The Agentic Enterprise: How Delegated Software Changes Coordination, Management, and Firm Boundaries. Volume 01, Publication 03. Version 1.0.

Canonical URL

https://sentient-review.com/publications/the-agentic-enterprise

© Sentient Review. All rights reserved. This canonical PDF edition reproduces the publication as presented online.