SENTIENT REVIEW

Volume 01 · Publication № 01 · Published

The AI Native Operating Model

Redesigning the enterprise for a world in which intelligence and execution are shared between humans and machines

VOLUME	Volume 01
PUBLICATION	Publication № 01
PUBLICATION STATUS	Published
RESEARCH DOMAIN	AI Native Operating Models
EDITOR	Fritzie M. Alex

Topics

AI Native Operating Models · Enterprise Operating Models · Organizational Design · Agentic Execution · Technology Economics

Canonical Edition

https://sentient-review.com/perspectives/the-ai-native-operating-model

Contents

Part I

The Transition

Chapter 01

The Operating Model Is Changing

The enterprise was built around human execution. AI changes that assumption.

> *When machines become participants in enterprise execution, the enterprise has to be designed for humans and machines working together.*

For most of the history of the modern enterprise, people did the cognitive work.

01 People researched.

02 People analyzed.

03 People wrote.

04 People coordinated.

05 People made decisions.

06 People supervised other people doing those things.

Software supported the work. It stored information, moved transactions and automated defined processes.

AI changes the role technology can play.

AI can interpret information, generate work, use tools, coordinate actions and execute sequences of tasks across systems.

That creates a new source of execution capacity inside the enterprise.

The operating model has to change with it.

That is the shift toward the AI Native Operating Model.

01 · The Change Is Already Visible

The strongest evidence is found in the work itself.

Microsoft has described an AI native approach to software development in which specifications become a coordination mechanism between human intent and machine execution. Engineers move toward defining intent, constraints and acceptance while AI takes on more implementation work.

Block has gone further at the organizational level. Its leadership has described a model built around smaller teams, outcome ownership, shared intelligence and a different role for managers. The company presents the structure as an active transition rather than a finished model.

Moderna combined responsibility for people and digital technology under one executive mandate. The logic was direct. Workforce planning and technology planning become harder to separate when machines can perform parts of the work previously assigned to people.

Walmart has sought senior leadership for agent systems with responsibility spanning agent portfolios, governance, operating model design and organizational responsibilities. The mandate itself is evidence that agent deployment is becoming an enterprise operating concern rather than a narrow software implementation.

These companies are taking different paths.

That matters.

There is no settled AI Native organization chart.

What is becoming clearer is the problem they are responding to.

AI can now participate in execution.

02 · The Unit of Capacity Is Changing

The traditional workforce model was built around people.

A company estimated demand, translated that demand into work and determined how many people and systems were required to perform it.

AI introduces another form of capacity.

01 A customer service workflow can combine people with AI agents.

02 An engineering workflow can combine developers with coding agents.

03 A finance process can combine human judgment with machine analysis and execution.

04 A sales process can combine relationship ownership with machine research, preparation and system updates.

The planning question changes.

Instead of asking:

> *How many people do we need to perform this work?*

the enterprise can ask:

> *What combination of human judgment and machine execution produces the best outcome?*

That question reaches far beyond workforce productivity.

It changes how work should be designed in the first place.

Operating Implication

> *AI introduces machine execution as a form of enterprise capacity. Workforce design therefore expands into execution design. The enterprise must determine how human judgment, AI assistance, agent execution and deterministic systems combine to produce the strongest outcome.*

03 · Work Can Change Before the Organization Chart

An enterprise may still have Engineering, Finance, Sales, Marketing and Customer Service.

The boxes can look familiar while the work inside them changes.

01 A developer may spend less time producing code and more time defining architecture, constraints and acceptance.

02 A service employee may handle fewer routine interactions and more exceptions.

03 A manager may spend less time collecting status and more time governing outcomes, capacity and exceptions.

04 A finance professional may move from producing analysis toward evaluating assumptions and making decisions from machine generated work.

The title may stay the same.

The job underneath it may not.

This is why AI Native transformation begins with work.

Once work changes, other parts of the operating model follow.

01 Authority changes.

02 Management changes.

03 Skills change.

04 Technology changes.

05 Infrastructure demand changes.

06 Economics change.

04 · The Market Is Testing Different Answers

The evidence does not point toward one organizational template.

That is useful.

01 Block is testing changes to hierarchy and management.

02 Microsoft is changing the relationship between specification and execution.

03 Moderna has connected people and technology leadership.

04 Enterprises across customer operations are assigning more routine execution to AI.

Each is testing a different part of the same question.

> *What should the enterprise look like when intelligence and execution are no longer supplied by people alone?*

The answer will vary by industry, workflow, risk and economics.

But the design problem is now real.

Operating Implication

> *AI Native design is a system design problem rather than a search for one universal organization chart. The operating model should follow the characteristics of the work, the authority involved, the risk of the outcome and the economics of execution.*

05 · AI Native Is About a More Capable Enterprise

The opportunity is larger than cost reduction.

AI creates the possibility of increasing the execution capacity available to an organization without requiring human effort to grow at the same rate.

01 Expertise can reach more work.

02 Routine coordination can consume less human attention.

03 Knowledge can move faster.

04 Decision cycles can shrink.

05 Organizations can absorb more demand.

06 People can spend more of their time on judgment, relationships, ambiguity, creativity and ownership.

07 Workflows can be rebuilt around outcomes rather than inherited handoffs.

That is the reason to pursue an AI Native Operating Model.

The goal is a more capable enterprise.

06 · That Enterprise Needs a Different Operating System

The change reaches across the organization.

01 Work needs a new allocation between humans and machines.

02 Agents need defined authority.

03 Human accountability needs clear ownership.

04 Managers need to govern human and machine capacity.

05 Enterprise context needs to be available to AI systems under controlled access.

06 Agent activity needs identity, permissions and evidence.

07 Infrastructure needs to support the execution demand created by the new work.

08 Measurement needs to follow work through to accepted outcomes.

09 Released human capacity needs a destination.

10 Economics need to include both labor and machine execution.

These are connected design decisions.

Together, they form the AI Native Operating Model.

07 · The Question Has Changed

The first era of enterprise AI focused on a simple question:

> *Where can we use AI?*

The next question is larger:

> *How should the enterprise be designed when both humans and machines can execute meaningful work?*

The rest of this publication develops an answer.

It starts with the workflow.

01 Then the division of work.

02 Authority.

03 Management.

04 Organization.

05 Technology.

06 Agents.

07 Infrastructure.

08 Economics.

09 And the mechanisms required to move an enterprise from AI adoption toward an AI Native operating model.

The destination is a company with more capacity to execute, learn and adapt because human judgment and machine intelligence have been designed to work as one operating system.

> *That is the enterprise we now have the opportunity to build.*

Read online: https://sentient-review.com/perspectives/the-ai-native-operating-model/01-the-operating-model-is-changing

Chapter 02

What AI Native Actually Means

The boundary between using AI and operating as an AI Native enterprise.

> *AI Native begins when the enterprise changes because machine intelligence has become part of how work is executed.*

An enterprise does not become AI Native because it deploys AI.

It can give every employee access to an AI assistant. It can deploy copilots across engineering, finance, sales and operations. It can build agents. It can automate support queues, generate software, summarize meetings and produce documents at machine speed.

All of that can happen while the operating model remains almost untouched.

The same functions can remain in place. The same jobs can remain intact. The same approval chains can govern decisions. The same management hierarchy can allocate work. The same financial model can determine capacity. The same technology estate can treat AI as another application layer.

That is AI adoption.

The distinction matters.

AI adoption asks how artificial intelligence can improve the enterprise that already exists.

An AI Native Operating Model asks what the enterprise should become once intelligence and execution no longer depend on human labor alone.

Those are different questions.

And they produce different organizations.

01 · Adoption Is Already Widespread

The first phase of enterprise AI has moved fast.

OpenAI reported in late 2025 that enterprise usage was expanding both in reach and intensity. Weekly ChatGPT Enterprise message volume had increased about eightfold over the prior year. Usage of structured workflows such as Projects and Custom GPTs had increased nineteenfold during the year. Workers surveyed across almost one hundred enterprises reported meaningful improvements in speed and quality.

The pattern has continued.

By June 2026, OpenAI reported that agentic use had moved well beyond software engineering. Enterprise Codex usage had grown across legal, sales, recruiting and marketing, while agent generated output accounted for a large share of combined ChatGPT and Codex output among enterprise customers.

Anthropic has observed a related transition. Its 2026 Economic Index describes usage moving from conversations between a user and an assistant toward longer running agentic work. The research continues to distinguish augmentation, where AI assists human activity, from automation, where a larger portion of the task is delegated to the system.

Microsoft reports the same movement from another vantage point. Its 2026 Work Trend Index found active agents in the Microsoft 365 ecosystem had grown fifteenfold year over year, and eighteenfold in large enterprises.

The direction is clear.

AI is entering enterprise work.

But penetration is not the same thing as transformation.

What This Supports

The evidence supports a move from occasional AI assistance toward repeatable workflows, delegated tasks and agent based execution.

It also supports a widening range of enterprise activities in which machines can participate in work rather than only provide information to the people performing it.

What Remains Unresolved

Usage data does not establish that enterprises have redesigned themselves around these capabilities.

More AI activity does not prove that authority has changed.

More agents do not prove that workflows have been reconstructed.

Time saved does not prove that capacity has been removed, redirected or converted into economic value.

Adoption measures the presence of AI.

AI Native requires evidence of operating model change.

02 · The Boundary

Sentient Review defines an AI Native enterprise as:

> *An enterprise in which human judgment and machine intelligence are designed as participants in the operating system of the firm, with work, authority, management, organization, technology and economics constructed around their combined capabilities.*

Every part of that definition matters.

AI cannot remain an optional tool sitting beside the work.

It has to enter the design of the work.

That creates a practical boundary.

A company can be AI enabled while its workflows remain human designed and human executed.

It can be AI assisted while decisions remain routed through the same managerial structures.

It can automate thousands of tasks while preserving the organizational architecture that produced those tasks.

It can deploy agents while leaving humans responsible for connecting fragmented systems, resolving exceptions and carrying information across organizational boundaries.

It can generate major productivity gains without changing the economics of the enterprise.

None of these conditions alone makes the company AI Native.

The operating model has to move.

03 · Six Things Have to Change

The boundary becomes clearer when examined across six operating dimensions.

01 Work Has to Change

The first shift is from adding AI to tasks toward redesigning the workflow.

Most enterprises are collections of workflows hidden inside organizational structures.

A customer dispute moves across systems and teams.

A product moves from idea to design to engineering to testing to release.

A financial forecast moves from data extraction to reconciliation to analysis to management review.

A technology incident moves from detection to diagnosis to remediation to validation.

Traditional enterprise design distributes these activities across people, applications, queues, meetings and functions.

AI adoption inserts intelligence into parts of that sequence.

AI Native redesigns the sequence itself.

Microsoft's 2025 research described emerging organizations moving from conventional org structures toward more fluid, outcome based arrangements of humans and agents. Its 2026 research goes further, arguing that leaders must rearchitect work and determine what humans and AI should each do.

McKinsey has described a similar model in which AI first workflows become the anchor of the operating system, with humans and conventional technology introduced where their capabilities are required.

The distinction is architectural.

If an analyst previously spent six hours collecting information and AI reduces that to one hour, the enterprise has improved a task.

If the workflow is redesigned so that information is gathered continuously, exceptions are identified by agents, routine decisions are executed within policy and the analyst enters only where judgment is required, the enterprise has changed how the work exists.

That is the beginning of AI Native work design.

Sentient Operating Implication

> *The workflow, not the job description, becomes the primary unit of transformation.*

02 Authority Has to Change

Execution without authority produces sophisticated assistants.

Execution with bounded authority produces operating actors.

This is one of the most important boundaries in the AI Native enterprise.

An agent that can prepare a refund recommendation is useful.

An agent authorized to issue refunds below a defined threshold, provided specific conditions are satisfied, occupies a different position in the operating model.

The question is no longer only:

> *What can the system do?*

It becomes:

> *What is the system allowed to do?*

That introduces decision rights.

Authority must be explicit.

Some actions should remain advisory.

Some can proceed with human approval.

Some can execute within policy and financial limits.

Some require mandatory human intervention because the consequence, ambiguity or regulatory exposure is too high.

The architecture therefore has to connect capability with permission.

Without this distinction, enterprises tend toward one of two failures.

They constrain capable systems until little operating leverage is created.

Or they grant autonomy without adequate accountability.

AI Native requires neither extreme.

It requires accountable autonomy.

Sentient Operating Implication

> *Every material agent role needs an authority envelope defining what it can observe, recommend, decide, execute and escalate.*

03 Management Has to Change

Management exists in part because work has to be coordinated.

Managers allocate work, resolve exceptions, monitor progress, review output, transfer information, enforce standards and intervene when execution departs from expectation.

Machine participation changes that coordination problem.

Microsoft's 2025 Work Trend Index anticipated employees increasingly building, delegating to and managing agents. Its 2026 research found that organizational conditions such as culture, manager support and talent practices had a stronger association with reported AI impact than individual behavior alone. Microsoft is explicit that the result is associative rather than causal, but the pattern is important.

The manager does not disappear.

The management problem changes.

When execution capacity can be instantiated through software, management shifts toward designing work, defining intent, setting constraints, assigning authority, evaluating output and intervening when the system encounters conditions outside its operating envelope.

Some coordination activity that once justified layers of management may become computational.

Other management responsibilities become more important.

01 Judgment.

02 Context.

03 Conflict resolution.

04 Goal formation.

05 Ethical responsibility.

06 Quality definition.

07 Organizational learning.

The AI Native manager therefore manages a mixed execution system.

People are part of it.

Agents are part of it.

Models, workflows, policies and infrastructure are part of it.

Sentient Operating Implication

> *Management must move from supervising human activity toward governing execution across humans and machines.*

04 Organization Has to Change

Enterprises have long organized around the economics of human coordination.

Functions aggregate expertise.

Teams create manageable units of execution.

Hierarchies establish accountability.

Management spans reflect how many people one person can coordinate.

Departments create boundaries around specialized capability.

These structures will not disappear.

But some of the assumptions beneath them are changing.

Machine intelligence does not have to be organized into departments before it can be applied.

An agent can participate in finance, engineering, procurement or customer operations without occupying a conventional seat in any of them.

A workflow can draw on many specialized agents without creating a corresponding human team.

Execution capacity can expand without a proportional increase in headcount.

Microsoft's research calls this movement from rigid organizational charts toward more fluid structures organized around outcomes. McKinsey similarly describes outcome aligned agentic teams as potential organizational building blocks.

This does not establish the end of functional organizations.

The evidence is too early for that claim.

It does establish pressure on an organizational model in which capability and execution capacity are assumed to map directly to human roles and reporting structures.

What Remains Unresolved

There is not yet enough longitudinal evidence to establish what the durable organizational form of a large AI Native enterprise will be.

Functional hierarchies may persist.

Product models may evolve.

New hybrid structures may emerge.

Different industries may settle on different architectures.

AI Native should therefore not be confused with a particular org chart.

The deeper requirement is that organizational design stops assuming that humans are the only meaningful source of execution capacity.

05 Technology Has to Change

AI Native is not achieved by attaching a model to the existing application estate.

Persistent machine execution creates a different technology problem.

01 Agents need identity.

02 They need access to data.

03 They need tools.

04 They need permissions.

05 They need state.

06 They need memory where appropriate.

07 They need observability.

08 They need evaluation.

09 They need policy enforcement.

10 They need mechanisms for escalation and intervention.

11 They need infrastructure capable of supporting variable and sometimes intensive inference workloads.

And when multiple agents participate in the same workflow, the enterprise needs orchestration and control.

This turns AI from an application feature into part of enterprise execution infrastructure.

The architectural concern moves beyond model access.

The enterprise has to know which agent acted, under whose authority, against which systems, using what information, at what cost and with what result.

That is a control plane problem as much as an AI problem.

Sentient Operating Implication

> *The AI Native technology estate must treat machine actors as governed participants in enterprise execution.*

06 Economics Has to Change

This may be the hardest boundary.

Productivity is not the same as economic value.

If AI saves an employee five hours each week and nothing changes about capacity, output, revenue, cost or service quality, the enterprise may have created convenience without changing its economics.

The value only becomes material when the released capacity goes somewhere.

01 It may reduce cost.

02 It may increase throughput.

03 It may improve quality.

04 It may accelerate revenue.

05 It may reduce cycle time.

06 It may allow the enterprise to absorb growth without proportional hiring.

07 It may move human effort toward higher value work.

OpenAI's enterprise research reports meaningful worker time savings and improvements in speed and quality. Those are important signals. But they do not establish how much of that time converts into enterprise level economic value.

This distinction becomes more important as agentic execution grows.

The enterprise will increasingly manage two forms of execution capacity.

Human capacity has familiar economics: compensation, benefits, facilities, management overhead and organizational constraints.

Machine capacity has different economics: tokens, inference, infrastructure, software, orchestration, evaluation and control.

The two cannot be managed as if they were equivalent.

Nor can machine capacity be treated as free because its unit cost appears small.

At enterprise scale, architecture creates the bill.

AI Native economics therefore requires a new discipline.

Organizations must understand the cost of execution at the workflow level and determine the economically rational combination of human and machine capacity.

Sentient Operating Implication

> *The unit of AI economics should move from licenses and model consumption toward the cost and value of executing an outcome.*

Exhibit 02.1

AI Adoption Versus an AI Native Operating Model

	AI Adoption	AI Native Operating Model
Work	AI improves existing tasks	Workflows are redesigned around human and machine capabilities
Execution	Humans perform the work with AI assistance	Humans and agents both execute parts of the workflow
Authority	AI recommends	Machines receive explicit authority within defined boundaries
Management	Managers supervise people using AI	Managers govern mixed human and machine execution
Organization	Existing roles and functions remain the primary structure	Organization reflects outcomes, workflows and mixed execution capacity
Technology	AI is added to applications	AI becomes governed execution infrastructure
Economics	Value measured through usage and productivity	Value measured through capacity, outcome economics and operating leverage
Control	Governance focuses on access and acceptable use	Governance extends to identity, authority, action, evaluation and intervention
Learning	Individuals improve their use of AI	The operating system captures execution signals and improves how work is performed

The right hand column is not a maturity score.

It is an architectural boundary.

An enterprise may cross that boundary at different speeds in different parts of the organization.

A customer operations workflow may become AI Native while procurement remains conventional.

Software engineering may develop extensive agent execution while finance remains dominated by human approval chains.

The enterprise therefore does not cross into AI Native status through a single technology deployment.

It crosses workflow by workflow, decision by decision and operating system by operating system.

04 · Adoption Is About Tools. AI Native Is About Design.

There is a reason the distinction is becoming more important.

The technology is moving faster than the organizations using it.

Microsoft's 2026 research found only 19 percent of surveyed AI users in what it calls the Frontier zone, where individual AI capability and organizational readiness are both high. Ten percent were classified as having strong individual capability but insufficient organizational conditions around them. Only 26 percent said leadership was clearly and consistently aligned on AI.

The study has limits. Its organizational readiness measures are based on self reported survey data, and the relationship between organizational conditions and AI impact is associative rather than causal.

But the underlying problem is credible.

People can learn new tools faster than enterprises can redesign themselves.

That creates a dangerous middle state.

01 AI capability rises.

02 AI usage rises.

03 Local productivity rises.

04 The operating model stays where it was.

Employees begin creating informal workflows around formal ones.

Agents gain access to more systems.

Management practices remain based on human execution.

Governance tries to control new forms of work through policies designed for software access.

Finance sees growing AI expenditure without a clear connection to capacity economics.

Technology teams scale infrastructure before the organization knows which workflows deserve the compute.

The enterprise becomes more AI intensive without becoming more coherent.

AI Native is the attempt to resolve that gap.

05 · The Test

A useful test is not how much AI the enterprise has deployed.

Ask six harder questions.

HAS THE WORK CHANGED?

Are workflows being reconstructed, or are employees performing the same workflows with faster tools?

HAS AUTHORITY CHANGED?

Can machine actors take consequential action inside explicit boundaries, or can they only advise humans?

HAS MANAGEMENT CHANGED?

Are managers designing and governing mixed execution systems, or supervising the same teams with AI added?

HAS ORGANIZATION CHANGED?

Is execution capacity still assumed to map to human roles and reporting lines, or can work be assembled around outcomes using both human and machine capability?

HAS TECHNOLOGY CHANGED?

Does the enterprise have infrastructure for identity, orchestration, permission, observation, evaluation and control of machine actors?

HAVE THE ECONOMICS CHANGED?

Can the enterprise show where machine execution changes cost, capacity, throughput, revenue, quality or operating leverage?

If the answer to most of these questions is no, the organization may be advanced in AI adoption.

It is not yet AI Native.

06 · SENTIENT INTERPRETATION

The defining feature of an AI Native enterprise is not artificial intelligence.

It is the redesign of the enterprise around a new fact:

> *Human labor is no longer the only scalable source of cognition and execution available to the firm.*

Once that becomes true, assumptions embedded in the traditional operating model become open to redesign.

A job no longer has to contain every task required to produce an outcome.

A manager no longer has to coordinate every unit of execution through people.

An organizational capability no longer has to scale through proportional headcount.

A workflow no longer has to stop at every system boundary and wait for a person to move it forward.

A decision does not have to remain human simply because humans made it in the previous operating model.

And technology is no longer only the environment in which employees work.

Technology can perform the work.

This does not remove humans from the enterprise.

It changes where human contribution matters most.

01 Humans define intent.

02 Humans establish values and constraints.

03 Humans exercise judgment where context, ambiguity and consequence demand it.

04 Humans determine what should be optimized.

05 Humans remain accountable for the systems they authorize.

Machine intelligence expands the execution surface around them.

That is the AI Native proposition.

Not fewer humans as an objective.

Not more agents as an objective.

A more capable enterprise as the objective.

07 · OPERATING IMPLICATION

Becoming AI Native is therefore not an AI program.

It is operating model transformation.

The enterprise has to redesign work.

It has to determine the new division of work between humans and machines.

It has to establish decision rights.

It has to redefine management.

It has to reconsider organizational structure.

It has to build an execution architecture capable of governing machine actors.

And it has to connect all of this to economic outcomes.

These changes cannot be completed through a single transformation program because the underlying capabilities will continue to move.

01 Models will improve.

02 Agent architectures will mature.

03 The cost of machine intelligence will change.

04 New forms of execution will become technically and economically possible.

The operating model therefore cannot be designed once and declared complete.

It needs the capacity to adapt.

That makes AI Native a design principle before it becomes an enterprise state.

The principle is straightforward:

> *When machines become capable participants in execution, the enterprise should redesign the work around the combined capabilities of humans and machines rather than forcing those capabilities into a system built for humans alone.*

The practical consequence is more demanding.

01 Every workflow becomes a candidate for reconstruction.

02 Every decision becomes a candidate for explicit allocation.

03 Every management activity becomes a candidate for reassessment.

04 Every organizational boundary becomes a candidate for examination.

05 Every unit of human and machine capacity becomes an economic choice.

The objective is not maximum automation.

It is the best operating system for producing the outcome.

That distinction will govern the rest of this publication.

08 · CONNECTION TO THE AI NATIVE OPERATING MODEL

Chapter 01 established the structural premise: when machines become participants in enterprise execution, the enterprise has to be designed for humans and machines working together.

Chapter 02 establishes the boundary.

AI adoption changes what people can do inside the existing operating model.

AI Native transformation changes the operating model itself.

The rest of the publication examines what that redesign requires.

It begins with the most fundamental unit.

Not the model.

Not the agent.

Not the application. Not the job.

> *The workflow.*

Chapter 03 · The Workflow Becomes the Unit

Evidence & Sources

01 **The state of enterprise AI**
OpenAI · December 8, 2025 · Institutional Research
Enterprise adoption, workflow usage and reported employee outcomes.
https://openai.com/index/the-state-of-enterprise-ai-2025-report/

02 **Anthropic Economic Index report: Cadences**
Anthropic · June 26, 2026 · Institutional Research
The movement from conversational assistance toward longer running agentic work.
https://www.anthropic.com/research/economic-index-june-2026-report

03 **Anthropic Economic Index report: Economic primitives**
Anthropic · January 15, 2026 · Institutional Research
Observed patterns of augmentation and automation in AI usage.
https://www.anthropic.com/research/anthropic-economic-index-january-2026-report

04 **2025: The year the Frontier Firm is born**
Microsoft · April 23, 2025 · Institutional Research
Emerging human and agent work structures, management practices and organizational design.
https://www.microsoft.com/en-us/worklab/work-trend-index/2025-the-year-the-frontier-firm-is-born

05 **2026 Work Trend Index Annual Report: Agents, human agency, and the opportunity for every organization**
Microsoft · May 5, 2026 · Institutional Research
Agent growth, organizational readiness, leadership alignment and the relationship between organizational conditions and reported AI impact.
https://www.microsoft.com/en-us/worklab/work-trend-index/agents-human-agency-and-the-opportunity-for-every-organization

06 **The agentic organization: Contours of the next paradigm for the AI era**
McKinsey & Company · September 26, 2025 · Institutional Research
AI first workflow redesign and emerging organizational implications of agentic systems.
https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/the-agentic-organization-contours-of-the-next-paradigm-for-the-ai-era

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/02-what-ai-native-actually-means

Part II

The Operating System

Chapter 03

The Workflow Becomes the Unit

Why transformation moves from task automation to end to end outcome redesign.

> *In an AI Native Operating Model, the workflow becomes the primary unit of design.*

Enterprises are organized around people.

Work is not.

Work moves.

A customer request enters the enterprise and moves through systems, decisions, teams and controls before an outcome is produced.

A product idea moves through research, design, engineering, testing, security and release.

An invoice moves through receipt, validation, matching, approval, payment and reconciliation.

A production incident moves through detection, diagnosis, remediation, validation and review.

The organization sees departments. The work crosses them.

This distinction becomes critical once machines can participate in execution.

If AI is introduced through the organizational structure, the enterprise asks which employees need copilots, which functions need agents and which applications need AI features.

If AI is introduced through the workflow, a different set of questions appears.

- What outcome is this workflow producing?
- What work is required to produce it?
- Which steps require human judgment?
- Which steps can be executed by machines?
- Which decisions can be delegated?
- Where does information move?
- Where does the workflow wait?
- Where does it fail?
- Where does human intervention create value?

- Where is human intervention only compensating for the limitations of the existing system?
- And what should the workflow look like if it were designed now?

That is the shift.

01 · The Enterprise Has Been Hiding Its Work

The conventional organization makes work difficult to see.

Executives see functions.

- Finance.
- Technology.
- Operations.
- Sales.
- Marketing.
- Human resources.
- Legal.
- Procurement.

Inside those functions are teams.

Inside the teams are roles.

Inside the roles are jobs.

Inside the jobs are tasks.

But outcomes rarely fit inside that hierarchy.

Consider a customer dispute.

The customer does not experience the organizational chart.

The customer experiences an outcome.

Behind that outcome may sit a sequence involving a contact center, customer data, transaction history, policy interpretation, fraud controls, financial authority, payment systems and exception management.

No single application contains the work.

No single employee performs the work.

No single function owns every part of the work.

The workflow exists across the enterprise.

Traditional organizations manage this fragmentation through coordination.

01 People send messages.

02 Meetings transfer context.

03 Managers resolve ambiguity.

04 Employees move information between systems.

05 Queues hold work until another person becomes available.

06 Approval structures compensate for uncertainty.

07 Specialists intervene when the process encounters an exception.

Much of what enterprises call work is therefore not the production of the outcome itself.

It is the coordination required to make a fragmented operating system produce the outcome.

AI changes this because machine intelligence can participate across the sequence.

But only if the enterprise can see the sequence first.

02 · Tasks Are Too Small

Much of the first generation of enterprise AI focused on tasks.

01 Summarize this document.

02 Draft this email.

03 Generate this code.

04 Analyze this spreadsheet.

05 Prepare this presentation.

06 Find this information.

These are useful applications.

They are also incomplete units of transformation.

TASK

A fragment of execution.

A task is usually one fragment of a larger chain.

Making the fragment faster does not mean the chain becomes faster.

If AI reduces a thirty minute activity to three minutes but the output then waits two days for review, the workflow has not improved by the same factor.

If a model generates a document instantly but three organizational layers still approve it, the execution constraint has moved.

If an agent identifies the correct action but lacks permission to take it, the workflow still waits for a human.

If one step becomes automated while the surrounding systems remain disconnected, people may spend the released time moving information between the automated and nonautomated parts of the process.

Local productivity can therefore coexist with systemic inefficiency.

This is one reason productivity measures alone are insufficient for evaluating AI transformation.

The enterprise can accumulate thousands of faster tasks without reconstructing how an outcome is produced.

Sentient Operating Implication

> *Do not begin with the task AI can perform. Begin with the outcome the enterprise needs to produce and trace the complete workflow required to produce it.*

03 · Jobs Are Too Large

JOB

A bundle of heterogeneous human activities.

The job is also the wrong unit.

A job bundles many kinds of work because human labor has historically been the unit through which capability enters the enterprise.

One role may contain analysis, judgment, administration, coordination, communication, approval, documentation, exception handling and relationship management.

Those activities do not have the same requirements.

01 Some depend on experience.

02 Some depend on context.

03 Some depend on trust.

04 Some depend on judgment.

05 Some are procedural.

06 Some are computational.

07 Some exist because systems do not communicate.

08 Some exist because information has to be gathered before a decision can be made.

09 Some exist because an organization has chosen to place a control at that point in the process.

AI exposes these differences.

The question is no longer whether AI can perform a job.

That framing is too coarse.

The useful question is what forms of cognition, execution, authority and human contribution exist inside the workflow in which that job participates.

Microsoft's 2026 Work Trend Index captures part of this movement. Its research describes advanced AI users shifting their attention from the tasks that define their jobs toward the outcomes they can drive. Microsoft classifies its most advanced group, Frontier Professionals, in part through their use of agents for complex or multistep work and their routine redesign of workflows around what AI can augment or automate.

That is an important distinction.

The job remains an employment construct.

The workflow becomes an execution construct.

AI Native design begins with execution.

What This Supports

Current evidence supports movement beyond isolated task assistance toward multistep agent use and workflow redesign among more advanced users and organizations.

It also supports a shift in the language of work from individual task completion toward outcomes and orchestration.

What Remains Unresolved

This does not establish that jobs will disappear or that enterprises will cease organizing employment through roles.

Jobs perform functions beyond task aggregation.

They establish accountability, career progression, compensation, professional identity and organizational membership.

The evidence does not support replacing the job as a human institution.

The narrower claim is stronger.

The job should no longer be treated as the primary unit for deciding how work should be executed.

04 · Applications Are Too Narrow

APPLICATION

A technology boundary.

Enterprise technology creates another distortion.

Organizations often see work through applications because applications are where digital activity becomes visible.

01 Customer relationship management.
02 Enterprise resource planning.
03 Human capital management.
04 Service management.
05 Collaboration.
06 Data platforms.
07 Engineering systems.

Each system owns part of the process.

The workflow does not care.

A customer resolution may require information from five systems.

A procurement decision may require seven.

An employee onboarding process may cross identity, human resources, payroll, security, facilities and management systems.

Historically, people have often been the integration layer.

01 They read information in one system.
02 Interpret it.
03 Enter information into another.
04 Ask someone for missing context.
05 Wait for approval.
06 Return to the system.
07 Continue the process.

Agentic systems create the possibility of moving execution across those boundaries.

That changes the role of the application.

Applications remain systems of record, transaction, policy and specialized capability.

But the workflow can increasingly sit above them.

The execution layer can gather context, invoke tools, coordinate specialized agents, request human judgment and continue once that judgment has been supplied.

The architecture begins to orient around the flow of work rather than the boundaries of software products.

Sentient Operating Implication

> *Applications should expose capabilities to the workflow. The workflow should not be forced to inherit the boundaries of the application estate.*

05 · From Assistance to Execution

The market is beginning to show this transition.

OpenAI's 2026 enterprise research describes AI use moving from assistance toward execution. Its data shows a widening difference between typical enterprise users and the firms with the deepest AI usage. Frontier firms are making greater use of capabilities that connect AI to company context, tools and repeatable workflows.

Anthropic has observed a related change in how its systems are used.

Its June 2026 Economic Index notes that a year earlier much Claude usage took the form of conversation between a person and an assistant. The growth of products such as Claude Code and Cowork has shifted more usage toward long running agentic tasks.

Microsoft reports that active agents in its ecosystem increased fifteenfold year over year and eighteenfold among large enterprises.

More important for operating model design, its research found that advanced AI users were more likely to rethink workflows, identify opportunities for augmentation and automation, and participate in repeatable AI enabled practices.

Yet the same Microsoft research exposes the gap.

Only a minority of respondents reported that agent workflows, human handoffs and quality standards were documented and repeatable across teams, functions or the wider organization.

The machines are entering execution faster than enterprises are redesigning the system around them.

What This Supports

There is evidence of movement from conversational assistance toward longer running execution, repeatable workflows and multistep agent use.

There is also evidence that advanced AI practice is associated with explicit workflow redesign rather than tool use alone.

What Remains Unresolved

The available evidence does not establish how much enterprise work is being executed through end to end agentic workflows.

Vendor telemetry captures activity inside particular technology ecosystems.

Survey data depends in part on self reported behavior.

Definitions of agents and agentic work also vary across providers.

The evidence supports direction.

It does not yet establish the scale of operating model transformation across the economy.

06 · The Workflow Has an Anatomy

Once the workflow becomes visible, AI Native design becomes more precise.

A workflow is not a list of tasks.

It is an execution system.

At minimum, it contains several distinct elements.

Outcome

What must the workflow produce?

The answer cannot be "complete the process."

The outcome must describe the result that matters.

Resolve the customer dispute.

Restore the production service.

Approve a qualified supplier.

Close the financial period.

Release compliant software.

The outcome provides the reason the workflow exists.

Inputs

What information, events, requests or conditions initiate the work?

Inputs may arrive from people, systems, sensors, transactions, external events or other workflows.

Execution

What actions must occur to produce the outcome?

Some actions involve information retrieval.

Some involve analysis.

Some change systems.

Some create artifacts.

Some communicate.

Some invoke other workflows.

Decisions

Where does the workflow choose between possible actions?

A decision is different from an execution step because it determines what happens next.

Decisions carry consequences.

That makes them candidates for explicit authority design.

Authority

Who or what is permitted to make each decision and execute each consequential action?

Capability does not create authority.

Authority must be assigned.

Controls

What constraints must the workflow respect?

Policy.

Security.

Financial thresholds.

Regulation.

Quality standards.

Risk limits.

Ethical boundaries.

Handoffs

Where does responsibility move between humans, agents, systems or organizational units?

Handoffs are important because they often introduce delay, context loss and failure.

Exceptions

What happens when the workflow encounters a condition outside the expected path?

Exceptions reveal where judgment, escalation or redesign is required.

State

What does the workflow need to remember as execution progresses?

Long running work cannot depend on each participant reconstructing context from the beginning.

Measurement

How does the enterprise know whether the workflow produced the intended outcome?

Time matters.

Cost matters.

Quality matters.

Failure matters.

Human intervention matters.

Machine consumption matters.

The workflow has to expose all of them.

WORKFLOW

The execution system through which the outcome is produced.

Exhibit 03.1

The AI Native Workflow

OUTCOME

What must be achieved?

WORK

What sequence of execution produces the outcome?

ALLOCATION

Which parts belong to humans, agents, conventional software or mixed execution?

AUTHORITY

Who or what can decide and act?

CONTROL

What policies, limits and quality standards govern execution?

EXECUTION

The workflow runs across humans, agents and enterprise systems.

OBSERVATION

What happened? Where did execution wait? Where did it fail? Where did humans intervene? What did machine execution cost?

ADAPTATION

What should change before the workflow runs again?

Adaptation feeds evidence from execution back into future workflow design.

The workflow is not static.

Execution produces evidence about the operating model.

That evidence should change the operating model.

07 · Human and Machine Work Must Be Designed Together

Once the workflow is the unit, the question of automation changes.

The objective is not to identify the maximum number of steps that a machine can perform.

That produces automation for its own sake.

The objective is to design the best execution system for the outcome.

Some work belongs with machines because machines can execute it faster, at greater scale or with greater consistency.

Some belongs with humans because the work requires judgment, accountability, empathy, negotiation, contextual understanding or the ability to reason about consequences that cannot be reduced to the available information.

Some work belongs in mixed execution.

01 An agent may gather the evidence.

02 A human may make the decision.

03 The agent may execute the decision.

04 Another system may validate the result.

05 A human may enter only when confidence falls below a threshold.

> *The correct allocation is therefore not human or machine. It is architectural.*

The workflow determines where each form of capability belongs.

This also means allocation can change.

A task that requires human execution today may move into machine execution as model capability improves.

A decision delegated to an agent may return to mandatory human review if failure patterns appear.

A workflow may use a more capable model only for a narrow class of exceptions because the economics do not justify using it everywhere.

The operating model becomes dynamic.

08 · Handoffs Become a Design Problem

Traditional enterprises contain enormous numbers of handoffs.

01 Team to team.
02 System to system.
03 Manager to employee.
04 Employee to manager.
05 Function to function.
06 Vendor to enterprise.
07 Human to application.
08 Application back to human.

Each handoff has a cost.

01 Time.
02 Context.
03 Attention.
04 Coordination.
05 Risk.

The cost is often invisible because it is distributed across the organization.

AI Native design makes the handoff explicit.

Some handoffs should remain.

A consequential decision may require human approval.

A customer interaction may benefit from human empathy.

A regulatory process may mandate a particular accountable actor.

Other handoffs exist only because the existing operating system cannot continue execution.

01 A person reads an email and enters the contents into an application.
02 A manager receives a report and forwards it to another team.
03 An analyst downloads information from one system and uploads it into another.
04 An employee checks whether a condition has been met and then triggers the next process.

These are not necessarily human contributions.

They are architectural seams.

Agents can remove some of them.

But removing a handoff does not mean removing control.

The better design is often to separate control from manual movement.

The system can continue execution while retaining explicit approval, policy and intervention points where they matter.

Sentient Operating Implication

> *Every human handoff should have a reason. If the reason is judgment, accountability, relationship or control, preserve it. If the reason is that the existing system cannot carry the work forward, redesign it.*

09 · Exceptions Reveal the Real Operating Model

The normal path tells only part of the story.

The exception path often reveals how the enterprise actually works.

01 What happens when the data is incomplete?

02 When the customer request falls outside policy?

03 When two systems disagree?

04 When an agent lacks confidence?

05 When the financial consequence exceeds its authority?

06 When a model produces an unexpected result?

07 When the workflow encounters something nobody anticipated?

Traditional automation often works well until an exception occurs.

Then the work returns to people.

This is not necessarily failure.

Human intervention can be part of the architecture.

The problem appears when the exception path has not been designed.

An AI Native workflow needs explicit conditions for escalation.

It needs to know when machine execution should stop.

It needs to know who receives the work.

It needs to preserve context when the handoff occurs.

It needs to capture what happened.

And it needs to determine whether the exception was unusual or evidence that the workflow itself needs to change.

This is where human judgment becomes more important, not less.

The human does not need to remain inside every routine execution step to remain accountable for the outcome.

Human contribution can move toward the places where ambiguity, consequence and novelty are highest.

10 · The Workflow Becomes Observable

There is another consequence of machine execution.

Work can become more visible.

Traditional knowledge work is difficult to observe.

Much of it occurs in meetings, messages, documents, individual reasoning and informal coordination.

Organizations often measure proxies.

01 Hours.

02 Headcount.

03 Tickets.

04 Projects.

05 Activity.

Machine execution creates a different possibility.

An agent can generate traces.

- The enterprise can know when execution began.
- Which tools were called.
- Which information was retrieved.
- Which decisions were made.
- Where confidence fell.
- Where a human intervened.
- Which model was used.
- How much compute was consumed.
- How long execution took.
- Whether the outcome passed evaluation.

This does not mean every form of work should become surveillance.

Nor does machine traceability guarantee meaningful measurement.

But it creates the possibility of observing execution at a level that traditional operating models often cannot.

> *The workflow can become both the unit of execution and the unit of learning.*

Microsoft's 2026 research describes a related idea in its concept of the organization as a learning system. Agent execution generates signals about what worked, what failed and where outcomes drifted. The organizational question is whether those signals remain local or become part of how future work is designed.

That is a significant operating model shift.

The enterprise can learn from execution itself.

11 · From Process Map to Living System

Enterprises already map processes.

AI Native workflow design is not process mapping with agents added.

A conventional process map describes how work is expected to move.

An AI Native workflow must govern how work actually executes.

It connects outcome, execution, authority, policy, technology, economics and learning.

It has to answer:

- What is happening?
- Who or what is acting?
- What authority does it have?
- What information is it using?
- What is the cost of execution?
- What outcome was produced?
- Where did execution depart from expectation?
- What should change?

That makes the workflow a living operating object.

Its design can change as evidence accumulates.

Its human and machine allocation can change.

Its authority boundaries can change.

Its models can change.

Its controls can change.

Its economics can change.

> *The workflow becomes something the enterprise operates, measures and improves. Not something it documents once.*

12 · SENTIENT INTERPRETATION

The workflow is where the AI Native enterprise becomes real.

Models do not transform enterprises.

Agents do not transform enterprises.

Copilots do not transform enterprises.

They create capabilities.

Transformation occurs when those capabilities change how an outcome is produced.

That happens inside the workflow.

The workflow connects the

The workflow connects the outcome the enterprise wants to produce with the work required to produce it.

It connects that work to the people, agents and systems capable of executing it.

It connects execution to authority.

Authority to control.

Control to observation.

Observation to economics.

And execution back to adaptation.

That is why the workflow becomes the primary unit of AI Native design.

It is the smallest useful operating system in which the major elements of the enterprise can be designed together.

The job is too large.

The task is too small.

The application is too narrow.

The workflow contains the operating problem.

This does not mean the enterprise should become a collection of automated processes.

It means the enterprise should understand how outcomes are produced before deciding where intelligence, automation and human judgment belong.

That distinction protects the AI Native model from a simple mistake.

The objective is not maximum automation.

The objective is better execution.

A workflow should use human capability where human capability creates the strongest result.

It should use machine capability where machine capability creates the strongest result.

It should combine them where neither is sufficient alone.

And it should preserve explicit accountability for the outcome.

The design question is therefore not:

REJECTED
How much of this workflow can we automate?

GOVERNING
What is the best way to produce this outcome using the combined capabilities of humans and machines?

That question changes the architecture of work.

It forces the enterprise to examine the workflow before preserving the job, the application, the approval chain or the organizational boundary around it.

It makes human participation intentional.

It makes machine participation intentional.

It makes authority explicit.

It makes controls part of execution rather than something applied after the fact.

And it makes the economics of the workflow visible.

This is the deeper shift.

The workflow is no longer a process that moves between parts of the organization.

It becomes an operating object that the enterprise can design, execute, observe and adapt.

That is the foundation of AI Native execution.

13 · OPERATING IMPLICATION

The practical starting point for AI Native transformation is not a catalog of AI use cases.

> *It is a portfolio of consequential workflows.*

Choose an outcome that matters.

01 Map how that outcome is produced today.

02 Identify the work.

03 Identify the decisions.

04 Identify the authority.

05 Identify the systems.

06 Identify the handoffs.

07 Identify the controls.

08 Identify the exceptions.

09 Identify where human judgment creates value.

10 Identify where human effort exists because the current architecture requires it.

11 Identify the cost of execution.

Then redesign the workflow from the outcome backward.

Do not assume that existing tasks must survive.

Do not assume that existing roles define the allocation of work.

Do not assume that existing application boundaries define the execution path.

Do not assume that every human handoff is a control.

Do not assume that every technically automatable step should be automated.

Design the execution system the outcome requires.

Then determine the combination of human and machine capability needed to operate it.

14 · CONNECTION TO THE AI NATIVE OPERATING MODEL

Chapter 01 — The operating model is changing.

Chapter 02 — AI adoption and AI Native transformation are different.

Chapter 03 — The workflow becomes the unit.

Chapter 04 — The New Division of Work.

Chapter 01 established the structural premise.

When machines become participants in enterprise execution, the enterprise has to be designed for humans and machines working together.

Chapter 02 established the boundary.

AI adoption improves the existing operating model.

AI Native transformation changes it.

Chapter 03 establishes the unit through which that change becomes concrete.

The workflow.

But once the workflow becomes the unit, the next question is unavoidable.

Who does the work?

The answer can no longer begin with the existing job architecture.

Each part of the workflow has to be allocated according to the capabilities the work requires.

Some work requires judgment.

Some requires context.

Some requires accountability.

Some requires empathy, negotiation or interpretation under ambiguity.

Other work requires speed, persistence, retrieval, synthesis, monitoring or execution at a scale that machines can provide.

Much of the enterprise will require both.

The next operating problem is therefore not automation.

It is allocation.

01 What should humans do?

02 What should machines do?

03 Where should they work together?

04 And how should the boundary between them change as capability, risk and economics change?

> *That is the new division of work.*

Chapter 04 · The New Division of Work

Evidence & Sources

01 **Anthropic Economic Index report: Cadences**
Anthropic · June 26, 2026 · Institutional Research
The movement from conversational assistance toward longer running agentic work.
https://www.anthropic.com/research/economic-index-june-2026-report

02 **2026 Work Trend Index Annual Report: Agents, human agency, and the opportunity for every organization**
Microsoft · May 5, 2026 · Institutional Research
Agent growth, workflow redesign, organizational learning and repeatable human and agent practices.
https://www.microsoft.com/en-us/worklab/work-trend-index/agents-human-agency-and-the-opportunity-for-every-organization

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/03-the-workflow-becomes-the-unit

Chapter 04

The New Division of Work

How humans, AI assistance, agents, orchestration and autonomous execution divide work.

> *What work should belong to humans, what work should belong to machines, and what work should exist between them?*

For most of the history of the enterprise, the division of work has been a division among people.

Executives decide.

Managers coordinate.

Specialists apply expertise.

Employees execute.

Software supports them.

Automation handles narrow, predefined tasks.

The operating model assumes that people are the actors and technology is the infrastructure around them.

That assumption no longer holds.

01 AI can research.

02 It can analyze.

03 It can generate.

04 It can monitor.

05 It can classify.

06 It can coordinate.

07 It can invoke tools.

08 It can operate software.

09 It can execute multistep work.

10 It can maintain state across an assignment.

11 It can act under supervision or, within defined boundaries, without waiting for a person at every step.

This does not mean that machines can perform every form of work.

It means the enterprise now has another source of execution capacity.

That changes the division of work.

The question is no longer simply how people should use AI.

The question is:

That is an operating model question.

And there is no durable answer based on the jobs that exist today.

01 · The Old Division of Work

The modern enterprise was designed around the constraints of human labor.

01 People have limited time.

02 Expertise is scarce.

03 Attention is finite.

04 Coordination becomes harder as organizations grow.

05 Knowledge is distributed.

06 People forget.

07 People need rest.

08 People cannot monitor every condition continuously.

Organizations developed structures to manage those constraints.

01 Jobs bundle related activities.

02 Teams aggregate capacity.

03 Functions aggregate expertise.

04 Managers coordinate people.

05 Processes standardize execution.

06 Applications store information and enforce transactions.

07 Automation removes repetitive work where the rules can be specified in advance.

The resulting operating model contains an implicit division of labor.

Humans provide cognition, judgment and flexible execution.

Software provides storage, calculation, transaction processing and deterministic automation.

That boundary shaped the enterprise for decades.

Generative and agentic AI disturb it because machine capability is moving into territory that previously required human cognition.

01 Language.

02 Reasoning.

03 Synthesis.

04 Interpretation.

05 Generation.

06 Planning.

07 Tool use.

08 Adaptive execution.

The boundary between human work and machine work is therefore no longer defined by whether a task requires cognition.

Both can now participate in cognitive work.

The operating problem becomes more precise.

Which cognition?

Which execution?

Under what conditions?

With what authority?

And with whom accountable for the result?

02 · Capability Is Not Allocation

One of the easiest mistakes in AI transformation is to treat technical capability as an allocation decision.

01 If a model can perform a task, automate it.

02 If an agent can execute a process, delegate it.

03 If the benchmark improves, expand autonomy.

That logic is incomplete.

Capability answers:

CAPABILITY

Can the system perform the work?

Allocation asks:

ALLOCATION

Should the system perform the work in this operating context?

Those are different questions.

A machine may be capable of drafting a legal analysis.

That does not determine who should own the legal judgment.

An agent may be capable of negotiating a routine commercial term.

That does not establish the authority it should receive.

A model may diagnose a technical failure.

That does not mean it should execute a production change without control.

A system may generate a performance assessment.

That does not establish whether the enterprise should delegate a consequential judgment about a person to it.

Technical capability is therefore one input into work allocation.

It is not the governing principle.

The enterprise must also consider consequence, ambiguity, accountability, reversibility, trust, economics and the availability of effective controls.

Sentient Operating Implication

> *Never infer authority from capability. A machine's ability to perform work does not determine whether the enterprise should allocate that work to the machine.*

03 · The Market Is Moving Toward Delegation

Enterprise AI usage is beginning to move beyond assistance.

OpenAI's August 2026 Enterprise Signals reports that enterprise use is becoming more agentic, with work moving from asking AI for help toward delegating substantive tasks. As of June, Codex accounted for 64 percent of combined Codex and ChatGPT output tokens among enterprise customers. OpenAI also reported rapid growth in enterprise Codex use outside engineering, including legal, sales, recruiting and marketing.

Anthropic reports a related shift. Its June 2026 Economic Index describes Claude usage moving from conversations between a person and an assistant toward longer running agentic work. Anthropic distinguishes augmentation, where AI assists a person, from automation, where the system receives more of the task. Its data also shows higher levels of autonomy in Claude Code than in conversational use for comparable outputs.

Microsoft frames the change around human agency. Its 2026 Work Trend Index argues that as agents take on more execution, people can spend more of their capacity directing work, making decisions and owning outcomes. The research draws on Microsoft 365 telemetry and a survey of 20,000 AI users across ten countries.

These sources come from technology providers with commercial interests in greater AI adoption.

That matters.

Their telemetry is also bounded by activity inside their own ecosystems.

But across those ecosystems, the directional signal is consistent.

Machine participation is moving deeper into execution.

The allocation problem is becoming real.

What This Supports

Current evidence supports a movement from AI as a conversational assistant toward systems that receive larger units of work and execute across multiple steps.

It also supports the emergence of human roles centered more on direction, evaluation and ownership as machine execution expands.

What Remains Unresolved

The evidence does not establish a universal division of work between humans and machines.

It does not show that more delegation always produces better outcomes.

It does not establish the appropriate level of autonomy for consequential enterprise decisions.

And it does not prove that organizations have developed the management, governance or economic systems required to allocate work well.

The technology is increasing the available choices.

The operating model still has to make them.

04 · Work Should Be Allocated by Its Requirements

The existing job architecture is a poor starting point for this decision.

A job may contain work that belongs in several different execution modes.

Consider a procurement professional.

The role may include gathering supplier information, comparing commercial terms, evaluating risk, negotiating with vendors, preparing recommendations, routing approvals, updating systems and managing relationships.

Those activities have different requirements.

01 Gathering supplier information may favor machine execution.

02 Comparing structured terms may favor machine execution with defined validation.

03 Interpreting an unusual contractual risk may require specialist judgment.

04 Negotiation may require human participation where relationships, ambiguity and strategic context matter.

05 Updating systems after a decision may again favor machine execution.

The role does not disappear.

The internal composition of the work changes.

The same logic applies across the enterprise.

An engineer may move from writing every line of code toward defining architecture, constraints and acceptance criteria while agents execute larger portions of implementation.

An analyst may move from gathering and formatting information toward framing the question, interrogating evidence and evaluating conclusions.

A manager may spend less time collecting status and routing work, and more time defining outcomes, resolving ambiguity and intervening in exceptions.

A customer service professional may move from processing routine cases toward handling the cases where policy, emotion or consequence requires human judgment.

This is why Chapter 03 moved the unit of transformation from the job to the workflow.

Once the workflow is decomposed, work can be allocated according to what each part requires.

05 · Four Modes of Execution

The new division of work is not binary.

Human or machine is too crude.

A useful operating model needs at least four execution modes.

01 · Human Execution

The human performs the work.

AI may provide information or tools, but responsibility for cognition and execution remains with the person.

This mode remains appropriate where the work depends on forms of judgment, relationship, accountability or context that the enterprise has chosen not to delegate.

02 · Human Led, Machine Assisted

The human owns the work.

AI expands the person's capability.

It may retrieve information, generate alternatives, perform analysis, draft outputs, simulate scenarios or prepare recommendations.

The human directs the work and determines the result.

This is where much enterprise generative AI adoption began.

03 · Machine Led, Human Governed

The machine performs most of the execution.

A human defines the objective, constraints and authority.

The system performs the work and returns to a person at defined decision points, exceptions, confidence thresholds or control gates.

The human is no longer executing every step.

The human governs the execution system.

04 · Machine Execution Within Authority

The machine receives an objective and can execute within a defined authority envelope without requiring human approval for each action.

The system may gather information, make bounded decisions, invoke tools and complete transactions.

Human accountability remains.

Human intervention moves to policy definition, monitoring, exceptions, evaluation and redesign.

This is where agentic execution begins to alter the operating model most directly.

Exhibit 04.1

The Human and Machine Execution Spectrum

01 · HUMAN EXECUTION

Human performs the work.

Machine: Limited support. — Human: Execution and accountability.

02 · HUMAN LED, MACHINE ASSISTED

Human directs and owns the work.

Machine: Research, analysis, generation and support. — Human: Direction, judgment and final decision.

03 · MACHINE LED, HUMAN GOVERNED

Machine performs most execution.

Machine: Multistep work and tool use. — Human: Objective, constraints, review, exception handling and consequential decisions.

04 · MACHINE EXECUTION WITHIN AUTHORITY

Machine executes independently inside explicit boundaries.

Machine: Execution, bounded decision making and action. — Human: Authority design, policy, monitoring, escalation and accountability.

This is not a maturity model. Moving work toward greater machine autonomy is not automatically progress. The correct position depends on the requirements of the work. The objective is not to move every workflow to the right. The objective is to place each part of the workflow in the execution mode that produces the strongest outcome at an acceptable level of risk and cost.

06 · Human Work Changes as Machine Execution Expands

The strongest case for AI Native transformation is not that machines become more capable.

It is that human capacity can move.

Microsoft describes this as an expansion of human agency. In its 2026 research, 58 percent of surveyed AI users said they were producing work they could not have produced a year earlier, rising to 80 percent among the group Microsoft identifies as Frontier Professionals. Microsoft argues that organizations need to redesign work, not merely deploy tools, if they want to convert that capability into value.

Anthropic's survey evidence adds another dimension. Large majorities of surveyed Claude users reported gains in speed, scope and quality, while 68 percent reported learning more with AI and 57 percent said AI had made their skills more valuable. Users who delegated more work to Claude reported more optimism about the value of their skills, although Anthropic cautions that these are self assessments and do not rule out skill erosion.

The important question is what happens to the capacity released by machine execution.

If nothing changes, the enterprise has faster employees.

If the work is redesigned, the enterprise can change what human capacity is used for.

Human contribution can move toward work such as:

- problem definition
- goal formation
- judgment
- exception handling
- relationship
- negotiation
- ethical reasoning
- quality definition
- interpretation
- creative direction
- accountability
- organizational learning

These are not permanent categories of work that machines will never touch.

That claim would outrun the evidence.

They are areas where human participation can carry particular value because the work involves consequence, context, responsibility or forms of understanding that the enterprise may not want to delegate.

The boundary will move.

The operating model has to be able to move with it.

07 · Human Judgment Cannot Become a Slogan

There is a risk in describing the future of work as machines execute and humans judge.

It sounds clean.

The reality is harder.

Judgment depends on knowledge.

A person cannot evaluate work they no longer understand.

A manager cannot govern an execution system they cannot interrogate.

A professional cannot identify a subtle error if the underlying expertise has disappeared.

Research on generative AI and expertise is beginning to expose this tension.

A 2026 perspective in Patterns argues that generative AI can accelerate scientific work while also creating a risk of skill atrophy because the technology can perform some of the same cognitive activities through which expertise develops, including synthesis, experimental design and writing. The authors argue for capabilities such as strategic direction, critical discernment and systematic calibration rather than passive reliance on generated output.

Other empirical work points to uneven complementarity. A 2026 randomized experiment on AI assisted learning found average performance gains from generative AI, but the gains varied with participants' ability to elicit, filter and verify AI output.

The evidence base is still developing.

Different tasks produce different results.

Some studies find that AI narrows performance differences.

Others find that expertise changes how much value people can extract from AI.

The durable operating implication is not that expertise matters less.

It is that the form of expertise may change.

The human who no longer performs every execution step may still need enough domain understanding to define the problem, evaluate the result, recognize failure and intervene when the system leaves its operating envelope.

What This Supports

There is evidence that AI can expand human capability and improve performance.

There is also evidence that effective human AI collaboration depends on the person's ability to direct, evaluate and verify machine output.

What Remains Unresolved

We do not yet know the long term effect of sustained delegation on professional expertise.

Productivity gains measured over weeks or months do not establish whether judgment strengthens or degrades over years.

Nor is it clear how organizations should preserve the experience pathways through which junior employees become capable senior decision makers when machines perform more of the work through which expertise was once developed.

That is not a reason to preserve inefficient work.

It is a reason to design the new division of work with greater care.

08 · The Apprenticeship Problem

Traditional organizations develop expertise through participation in work.

01 Junior engineers debug.

02 Junior lawyers research.

03 Analysts build models.

04 Doctors review cases.

05 Managers learn by resolving operational problems.

Professionals acquire judgment in part by performing the lower level work that exposes them to patterns, exceptions and failure.

AI can absorb some of that work.

That creates an operating model problem that is easy to miss.

> *If machines take over the work through which humans learned the domain, where does the next generation of human judgment come from?*

The answer cannot simply be to preserve low value work so that people can learn.

That would sacrifice operating leverage to maintain an old development model.

But the enterprise also cannot assume expertise will reproduce itself.

Learning has to become part of the architecture.

01 Humans may need deliberate exposure to exceptions.

02 They may need to review machine reasoning and outcomes.

03 They may need simulations.

04 They may need rotations through consequential cases.

05 They may need responsibility earlier for defining objectives, evaluating evidence and challenging machine output.

The development system has to evolve with the execution system.

Sentient Operating Implication

> *When machine execution removes developmental work, the enterprise must replace the learning mechanism, not preserve the work.*

09 · Allocation Must Be Dynamic

The division of work cannot be fixed.

01 Model capability changes.

02 Costs change.

03 Workflow volume changes.

04 Risk changes.

05 Regulation changes.

06 The enterprise learns from execution.

A task allocated to a human today may move toward machine execution tomorrow.

A machine executed activity may move back toward human review after a failure.

A workflow may use machine execution for routine cases and human execution for novel ones.

A more capable model may justify greater autonomy for one part of the workflow but cost too much to use across the entire process.

Allocation is therefore not an organizational decision made once.

It is an operating parameter.

The AI Native enterprise needs to know not only who or what performs the work today.

It needs to know why.

That reason should be observable.

It should be testable.

And when the evidence changes, the allocation should be capable of changing with it.

This creates a new management discipline.

The enterprise must manage the boundary between human and machine execution.

Not as ideology.

As architecture.

10 · The Boundary Is Where the Operating Model Lives

Once work can move between humans and machines, the boundary between them becomes part of enterprise architecture.

That boundary determines more than who performs a task.

01 It determines where judgment sits.

02 Where authority sits.

03 Where accountability sits.

04 Where cost accumulates.

05 Where control enters the workflow.

06 Where expertise is required.

07 Where exceptions go.

08 And where the enterprise remains dependent on human capacity.

Consider a workflow with one hundred cases.

Eighty may be routine.

Fifteen may contain ambiguity.

Five may carry material consequence.

There is little reason to require the same execution model for all one hundred.

The routine cases may support machine execution within authority.

The ambiguous cases may require machine analysis followed by human judgment.

The consequential cases may require direct human ownership with AI providing evidence and alternatives.

The workflow can therefore contain different divisions of work at the same time.

This is more precise than deciding whether a process should be automated.

The unit of allocation moves deeper.

The enterprise allocates work according to the characteristics of the work itself.

11 · Consequence Changes the Allocation

Not all decisions deserve the same operating model.

A machine choosing which internal document to retrieve is not equivalent to a machine approving a financial transaction.

A system deciding how to format a report is not equivalent to a system deciding whether a customer receives credit.

An agent scheduling a meeting is not equivalent to an agent terminating a supplier relationship.

As consequence increases, the requirements around authority, evidence, review and accountability change.

This does not mean high consequence work must always remain human executed.

Financial markets, industrial systems and infrastructure already contain forms of automated execution where consequential actions occur under engineered controls.

The relevant question is whether the enterprise has designed an operating environment capable of supporting the delegation.

That includes the quality of the system.

But it also includes the quality of the control architecture around it.

01 What evidence does the system use?

02 How is uncertainty represented?

03 What actions are permitted?

04 What thresholds constrain those actions?

05 What requires approval?

06 What conditions trigger escalation?

07 Can the action be reversed?

08 Can the enterprise reconstruct what happened?

09 Who remains accountable?

The more consequential the action, the less defensible it becomes to treat machine autonomy as a product feature.

It becomes an institutional design decision.

Sentient Operating Implication

> *Allocate authority according to consequence, not technological enthusiasm. The question is not how autonomous the agent can become. The question is how much authority the enterprise can justify delegating.*

12 · Reversibility Matters

Consequence is not the only factor.

Reversibility changes the economics of delegation.

Some machine actions are cheap to undo.

01 A draft can be discarded.

02 A recommendation can be ignored.

03 A software change can sometimes be rolled back.

04 A customer communication may be corrected.

Other actions create consequences that are difficult or impossible to reverse.

01 Money moves.

02 Data is deleted.

03 A contract is executed.

04 A regulatory filing is submitted.

05 A public statement is released.

06 A person is hired or dismissed.

07 A physical system changes state.

The allocation model should reflect this difference.

Where actions are reversible, enterprises can often tolerate broader machine experimentation and faster execution.

Where actions are difficult to reverse, authority should narrow and controls should strengthen.

This creates a useful design principle.

> *Autonomy should expand where consequence is bounded, reversibility is high and observation is strong.*

It should contract where consequence is high, reversibility is low or the enterprise cannot reliably observe what the system is doing.

This is not a permanent rule.

Better systems, controls and evidence may change the boundary.

But the burden of design should rise with the consequence of the action.

13 · Ambiguity Matters Too

Machines perform best when the operating environment provides enough information to distinguish a good action from a bad one.

Some enterprise work has clear objectives.

01 Resolve the ticket within policy.

02 Match the invoice to the purchase order.

03 Identify the failed service.

04 Generate the required test cases.

05 Reconcile the transaction.

Other work contains competing objectives.

01 Preserve the customer relationship while enforcing policy.

02 Reduce cost without damaging strategic capability.

03 Respond to an employee problem where policy, context and fairness pull in different directions.

04 Negotiate a partnership where the value depends on information neither party can fully quantify.

05 Decide whether an unusual technical risk is acceptable in the context of a larger business objective.

These are not difficult only because they require more intelligence.

They are difficult because the objective itself may be contested, incomplete or contextual.

Human judgment often matters most where the enterprise has not reduced the problem to a stable objective function.

That makes ambiguity a design variable.

The more ambiguous the objective, the stronger the case for human participation in framing the problem, resolving competing values and accepting responsibility for the decision.

Machine intelligence may still perform much of the analysis.

01 It may surface evidence.

02 Model alternatives.

03 Identify contradictions.

04 Simulate consequences.

05 Challenge assumptions.

But assistance and execution should not be confused with ownership of the judgment.

Exhibit 04.2

The Work Allocation Test

CAPABILITY

Can the machine perform the work to the required standard? Determines whether machine execution is technically viable

CONSEQUENCE

What happens if the work is wrong? Higher consequence increases the need for control and human judgment

AMBIGUITY

Is the objective clear, or must competing values and context be interpreted? Greater ambiguity increases the value of human participation

REVERSIBILITY

Can the action be undone at acceptable cost? High reversibility permits broader delegated execution

OBSERVABILITY

Can the enterprise see what happened and evaluate the result? Strong observability supports greater delegation

ECONOMICS

Does machine execution improve the outcome at an acceptable total cost? Determines whether delegation creates operating value

No single dimension determines the answer. A machine may have high capability but operate in a high consequence, low reversibility environment. Human execution may be safer but economically impossible at the required scale. A workflow may contain enough observability to permit machine execution even where individual actions carry moderate risk. Allocation is a systems decision.

14 · Human Oversight Is Not a Control Architecture

Enterprises often respond to AI risk with a simple instruction.

Put a human in the loop.

That phrase is useful.

It is also dangerously incomplete.

A human cannot provide meaningful oversight merely by appearing somewhere in the workflow.

01 The person must have enough information.

02 Enough time.

03 Enough expertise.

04 Enough authority.

05 And a realistic ability to challenge the machine.

If an agent processes ten thousand cases and asks one employee to approve the result, the presence of the employee does not establish effective control.

If the person routinely accepts machine recommendations because reviewing the underlying evidence would take longer than performing the original work, the approval may become ceremonial.

If machine output arrives with apparent confidence and the human lacks the expertise required to identify subtle failure, human review may create the appearance of safety without the substance.

This is an old organizational problem in a new form.

Controls can exist on paper while failing in operation.

The AI Native enterprise therefore has to design human intervention as carefully as machine execution.

01 What exactly is the human evaluating?

02 What evidence do they receive?

03 What conditions trigger their involvement?

04 How much work reaches them?

05 What happens when they disagree?

06 What expertise do they need?

07 How is their decision recorded?

08 How does the system learn from the intervention?

The answer cannot simply be:

A human approves it.

Sentient Operating Implication

> *Human oversight is effective only when the human has the capacity, information, expertise and authority to intervene. A human in the loop is an execution role. Design it.*

15 · The Human Role Moves Up the Stack

As machines absorb more execution, human work can move toward a different set of responsibilities.

01 From producing every artifact to defining what good looks like.

02 From gathering every piece of information to deciding which information matters.

03 From performing every analysis to challenging the assumptions behind the analysis.

04 From routing every exception to resolving the exceptions that matter.

05 From monitoring activity to interpreting outcomes.

06 From executing policy to determining when policy no longer fits the situation.

07 From coordinating tasks to designing the system through which tasks are executed.

This is not universally an upward movement.

Some machine systems will perform work that is intellectually demanding.

Some human work will remain procedural because human presence is required or economically rational.

Some occupations will change more than others.

Some forms of human labor may shrink.

Others may expand.

The distributional consequences will not be uniform.

But at the operating model level, one direction is important.

When machines can perform more of the execution, human capacity does not have to remain attached to execution merely because that is where it sat before.

The enterprise can redesign the human role around the places where human participation creates the most value.

16 · Accountability Does Not Transfer With Execution

Delegating execution does not delegate institutional responsibility.

An enterprise cannot treat an agent as the accountable party in the same sense that it can assign accountability to an executive, manager or regulated professional.

01 The machine can act.

02 It can produce evidence.

03 It can maintain logs.

04 It can operate within policy.

05 It can even evaluate parts of its own output.

But the enterprise remains responsible for deciding that the system should act.

That distinction is foundational.

The more execution moves into machines, the more explicit human accountability needs to become.

01 Who owns the workflow?

02 Who determines the objective?

03 Who defines the authority envelope?

04 Who approves the controls?

05 Who accepts the residual risk?

06 Who responds when the system fails?

07 Who decides whether the workflow should continue operating?

Machine execution can distribute action.

It cannot eliminate accountability.

The AI Native enterprise therefore separates two concepts that traditional work often bundles together.

REJECTED
Execution

GOVERNING
Accountability

The actor performing the work does not have to be the actor accountable for the operating system that produced it.

That separation makes greater autonomy possible.

It also makes governance more important.

17 · The New Scarcity Is Not Execution Alone

Traditional organizations are constrained by human execution capacity.

01 There are only so many engineers.
02 Analysts.
03 Managers.
04 Lawyers.
05 Designers.
06 Operators.
07 Specialists.

Machine execution can loosen some of those constraints.

But it creates new ones.

01 High quality context becomes scarce.
02 Clear objectives become scarce.
03 Reliable evaluation becomes scarce.
04 Authority becomes scarce.
05 Trustworthy data becomes scarce.
06 Human attention at the right intervention point becomes scarce.
07 Compute may become scarce or expensive.
08 The ability to design effective workflows becomes scarce.
09 The ability to distinguish good machine output from plausible machine output becomes scarce.

The enterprise does not escape scarcity. The scarcity moves.

This has consequences for organization design.

If machine execution becomes abundant while judgment, context and authority remain constrained, the bottleneck shifts toward the people and systems that provide them.

A thousand agents do not create leverage if every consequential action waits for the same overloaded executive.

Machine execution does not remove the acceptance bottleneck.

Poor operating design can intensify it.

Sentient Operating Implication

> *As machine execution scales, find the new constraint. Do not measure transformation by how much machine capacity has been added. Measure whether the constraint on the outcome has moved.*

18 · More Agents Do Not Mean More Capacity

Machine execution has a seductive property.

It can appear almost infinitely replicable.

An enterprise can create another agent without hiring another employee.

That does not mean useful capacity scales without limit.

Every agent consumes something.

01 Compute.

02 Tokens.

03 Data.

04 Tools.

05 Permissions.

06 Evaluation.

07 Observability.

08 Control.

09 Human attention.

10 Exception capacity.

The output of one agent may become the input constraint of another.

A workflow may generate more work for downstream humans than they can absorb.

More execution may create more exceptions.

More generated artifacts may create more review.

More analysis may create more decisions waiting for acceptance.

The system can therefore increase activity without increasing throughput.

This is the same mistake enterprises make when they optimize individual tasks.

Capacity has to be measured at the outcome.

If ten agents produce work faster but the workflow completes no more outcomes, execution capacity has increased while operating capacity has not.

That distinction becomes essential in the AI Native enterprise.

> *Machine capacity is not valuable because it exists. It is valuable when it changes the economics or performance of the outcome.*

19 · The New Division of Work Is an Economic Design

Human and machine allocation cannot be determined by capability and risk alone.

It has to make economic sense.

Machine execution has a cost structure.

01 Models consume inference.

02 Agents may make repeated model calls.

03 Long running workflows consume context and compute.

04 Tools and data services carry costs.

05 Evaluation consumes resources.

06 Observability creates infrastructure.

07 Human review consumes labor.

08 Exceptions create additional work.

09 Failures create remediation cost.

The cheapest model may create expensive errors.

The most capable model may cost more than the outcome justifies.

A human may remain economically preferable for low volume, high ambiguity work.

Machine execution may dominate high volume, structured work even when human execution remains technically possible.

Mixed execution may produce the strongest economics across many workflows.

The allocation problem therefore has another question:

> *What combination of human and machine capacity produces the required outcome at the best total economics?*

That question cannot be answered at the license level.

It cannot be answered through token price alone.

It requires workflow economics.

The enterprise must understand the cost of producing the outcome under different allocations of work.

That is where productivity becomes operating leverage.

20 · SENTIENT INTERPRETATION

The new division of work is not a prediction about which jobs survive. It is an architectural principle.

Once machines can participate in cognition and execution, the enterprise should stop treating the existing distribution of work as fixed.

Every consequential workflow can be decomposed.

Every component can be examined according to its requirements.

01 Capability.

02 Consequence.

03 Ambiguity.

04 Reversibility.

05 Observability.

06 Economics.

The result may preserve human execution.

It may create human led machine assistance.

It may shift toward machine led execution with human governance.

It may support machine execution within explicit authority.

And that allocation can change as the evidence changes.

This makes the boundary between human and machine work a managed part of the operating model.

Not a one time workforce decision.

Not an automation target.

Not a philosophical argument about whether humans or machines are better.

A design choice.

The objective is not to maximize machine work.

The objective is not to protect human work from change.

The objective is to build the strongest execution system for the outcome.

That system should use human judgment where judgment creates value.

Machine execution where machine execution creates leverage.

And mixed execution where the combination is stronger than either alone.

The result is not a human enterprise with AI attached to it.

Nor is it an autonomous enterprise from which people have been removed.

> *It is a mixed execution system.*

That is the operating reality the AI Native enterprise has to learn to govern.

21 · OPERATING IMPLICATION

Start with the workflow.

Decompose the work.

For each material unit of execution, ask:

01 Can the machine perform it?
02 Should the machine perform it?
03 What is the consequence of failure?
04 How ambiguous is the objective?
05 Can the action be reversed?
06 Can execution be observed?
07 What authority would be required?
08 What human capability must remain?
09 What does each execution mode cost?
10 What happens to the rest of the workflow if the allocation changes?

Then choose the execution mode.

01 · Human Execution

02 · Human Led, Machine Assisted

03 · Machine Led, Human Governed

04 · Machine Execution Within Authority

Do not treat the answer as permanent.

01 Instrument the workflow.

02 Observe performance.

03 Capture exceptions.

04 Measure human intervention.

05 Measure machine consumption.

06 Measure the outcome.

Then change the allocation when the evidence justifies it.

The new division of work is not something the enterprise discovers once.

It is something the enterprise operates.

22 · CONNECTION TO THE AI NATIVE OPERATING MODEL

Chapter 03 — The workflow becomes the unit.

Chapter 04 — The work inside that workflow must be allocated across humans and machines. ←

Chapter 05 — Decision Rights and Delegated Authority.

Chapter 03 established the workflow as the primary unit of design.

Chapter 04 establishes that execution inside the workflow can no longer be allocated through the existing job architecture alone.

Humans and machines can both perform work.

But capability does not create permission.

The enterprise still has to decide who or what is allowed to act.

A machine may be capable of approving a transaction.

That does not mean it has the authority to approve it.

An agent may be capable of changing a production system.

That does not determine the conditions under which it should be permitted to do so.

A workflow may support extensive machine execution.

That does not answer who remains accountable when the execution creates consequences.

The new division of work therefore creates the next operating problem.

Decision rights.

Who can recommend?

Who can decide?

Who can execute?

Under what conditions?

Within what limits?

When must authority return to a human?

And who remains accountable for the result?

Machine execution becomes enterprise execution only when those questions have explicit answers.

That is the subject of Chapter 05.

Chapter 05 · Decision Rights and Delegated Authority

Evidence & Sources

01 **Anthropic Economic Index report: Cadences**
Anthropic · June 26, 2026 · Institutional Research
The movement from conversational assistance toward longer running agentic work.
https://www.anthropic.com/research/economic-index-june-2026-report

02 **2026 Work Trend Index Annual Report: Agents, human agency, and the opportunity for every organization**
Microsoft · May 5, 2026 · Institutional Research
Human agency, workflow redesign and the changing allocation of execution between people and agents.
https://www.microsoft.com/en-us/worklab/work-trend-index/agents-human-agency-and-the-opportunity-for-every-organization

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/04-the-new-division-of-work

Chapter 05

Decision Rights and Delegated Authority

What happens when machine capability becomes permission to act, and how delegated authority remains accountable.

> *What may this agent do, to what, on whose behalf, under which conditions, for how long, and when must it stop?*

Execution is not authority.

A machine may be capable of performing an action.

That does not mean it should be allowed to perform it.

An agent may be capable of issuing a refund.

Changing a production system.

Approving an invoice.

Contacting a customer.

Updating an employee record.

Purchasing infrastructure.

Submitting a regulatory document.

Negotiating a commercial term.

Moving money.

THE TECHNICAL QUESTION

Whether the system can execute the action.

THE OPERATING MODEL QUESTION

Whether the enterprise has authorized it to do so.

Those questions have to remain separate.

Traditional enterprises already understand this distinction for people.

An employee may know how to approve an expense without having the authority to approve it.

An engineer may know how to change a production environment without having unrestricted permission to do so.

A manager may understand a compensation decision without possessing the authority to make it alone.

Capability and authority are separated through roles, policies, approval limits, access controls and accountability structures.

Machine actors require the same distinction.

But the problem is harder.

01 Agents can act at machine speed.

02 They can operate continuously.

03 They can cross systems.

04 They can invoke tools.

05 They can perform many actions before a human could inspect each one.

06 They can receive new context during execution.

And the path they take toward an objective may not be specified in advance.

Once machines move from recommending actions to taking them, decision rights become part of the execution architecture.

The AI Native enterprise therefore needs an explicit answer to a new question:

> *What authority can be delegated to a machine?*

01 · Decision Rights Are the Architecture of Permission

Every consequential workflow contains decisions.

01 Approve.

02 Reject.

03 Escalate.

04 Purchase.

05 Release.

06 Change.

07 Communicate.

08 Commit.

09 Stop.

10 Continue.

Traditional organizations distribute these rights among people.

Some decisions belong to employees.

Some belong to managers.

Some require specialists.

Some require committees.

Some require executives.

Some require boards.

Some cannot be delegated because law, regulation or institutional policy assigns accountability to a particular actor.

Decision rights determine who can make which decision under which conditions.

AI does not remove this architecture.

It introduces a new class of actor into it.

The enterprise now has to decide whether a machine can:

- observe
- recommend
- prepare
- decide
- execute
- escalate

and under what conditions each right applies.

This is more precise than asking whether an agent is autonomous.

Autonomy is not one thing.

An agent may have broad freedom to research while having no authority to transact.

It may prepare a decision but require human approval.

It may execute transactions below a threshold and escalate those above it.

It may operate without approval under normal conditions but lose authority when confidence falls, data conflicts or risk conditions change.

The relevant design object is therefore not autonomy in the abstract.

It is the distribution of decision rights.

02 · The Market Is Building an Authorization Layer

The need for this architecture is becoming visible in the market.

In February 2026, the U.S. National Institute of Standards and Technology's National Cybersecurity Center of Excellence published a concept paper focused on identity and authorization for software and AI agents.

The premise is straightforward.

Agents increasingly need access to data, tools and applications in order to perform useful work.

That access creates risk.

NIST identifies agent identification, authorization, auditing and nonrepudiation as areas requiring stronger technical practice as agentic systems enter enterprise environments.

The World Economic Forum reached a related conclusion in its 2026 playbook for trusted agent adoption.

Its proposed Agent Capability and Authorization Profile links what an agent can do with what it is authorized to do, the conditions under which that authority applies and the controls required around execution.

Microsoft is moving the same problem into enterprise infrastructure.

Its agent identity architecture treats agents as distinct identities that can receive governed access to enterprise resources. Microsoft also states that organizations remain responsible for identity, least privilege, authorization of actions, human oversight and accountability for agent behavior.

The technical implementations differ.

The underlying operating problem is the same.

Machine actors need identities.

Those identities need permissions.

Those permissions need limits.

And the enterprise needs to know who authorized the machine to act.

What This Supports

Current institutional and technology evidence supports the emergence of agent identity and authorization as distinct enterprise control problems.

It also supports the principle that access to a system is not sufficient authorization for every action available inside that system.

What Remains Unresolved

There is not yet a universal enterprise standard for delegated machine authority.

Different technology ecosystems implement identity, permission and policy differently.

Standards remain in development.

And technical authorization does not by itself resolve the organizational question of which decisions an enterprise should delegate.

Technology can enforce an authority model.

The enterprise still has to design one.

03 · Access Is Not Authority

This distinction is critical.

An agent may have access to a system because it needs information from that system.

That does not mean it should possess every action the system makes technically possible.

A customer service agent may need access to customer history.

It may need the ability to calculate a refund.

It may even need the ability to issue a refund.

But those are three different permissions.

Read.

Reason.

Act.

Traditional enterprise software often bundles these boundaries through application roles.

Agentic execution requires more precision because a machine can move from information to action without the natural pause created by a person navigating the system.

An agent that can read an invoice does not necessarily need permission to approve it.

An agent that can inspect production does not necessarily need permission to modify it.

An agent that can prepare a customer communication does not necessarily need permission to send it.

An agent that can identify a security issue does not necessarily need permission to disable an account.

The architecture must separate what the machine can see from what the machine can do.

And it must separate what the machine can technically do from what the enterprise has authorized it to do in this workflow.

Sentient Operating Implication

> *Do not give an agent authority merely because the underlying tool exposes the action. Tool access is a technical capability. Authority is an operating decision.*

04 · Authority Needs an Envelope

Delegated authority should not be expressed as a binary permission.

Allowed.

Not allowed.

Real enterprise authority is conditional.

A manager may approve spending up to a threshold.

An employee may sign an agreement only within defined terms.

An engineer may deploy to production after required controls pass.

A customer service representative may resolve a case within policy but escalate exceptions.

Machine authority should be designed the same way.

An authority envelope defines the conditions inside which a machine can act without seeking additional approval.

At minimum, the envelope can include:

Action

What is the agent allowed to do?

Object

What data, account, system, customer, transaction or resource may it act upon?

Threshold

What financial, operational or risk limit applies?

Context

Under what workflow or business condition does the authority exist?

Evidence

What information must be available before the action can be taken?

Confidence

What level of certainty or evaluation must be satisfied?

Time

How long does the authority remain valid?

Escalation

What condition causes authority to return to a human or another control point?

Accountability

Which human or institutional owner remains responsible for the delegated authority?

This changes the design of agent autonomy.

The agent is not simply autonomous.

It is autonomous inside an envelope.

Exhibit 05.1

The Authority Envelope

IDENTITY

Who or what is acting?

OBJECTIVE

What outcome has the actor been authorized to pursue?

PERMITTED ACTIONS

What can the actor do?

BOUNDARIES

What systems, data, resources and transactions are inside scope?

LIMITS

What financial, operational, temporal or risk thresholds apply?

EVIDENCE

What must be known before action is permitted?

ESCALATION

When must authority return to a human or another decision maker?

ACCOUNTABILITY

Who remains responsible for the authority that was delegated?

The authority envelope turns autonomy into an explicit operating construct.

It answers not simply whether the machine can act.

It answers:

> *Where does its right to act begin, and where does it end?*

05 · Authority Should Be Granted to the Action, Not the Intelligence

A powerful model does not need powerful permissions.

This distinction becomes more important as general purpose models become capable across many domains.

The same underlying model may be able to:

- write software
- analyze contracts
- interact with customers
- query financial systems
- modify infrastructure
- prepare hiring decisions
- research competitors

and operate enterprise applications.

The model's capability may be broad.

The agent's authority should not be.

Authority belongs to the operating role the machine is performing.

A procurement agent may receive authority to request supplier information but not approve payment.

A software agent may receive authority to create a pull request but not deploy directly to production.

A finance agent may reconcile transactions but require approval to post an adjustment above a defined amount.

A customer agent may resolve routine cases but escalate cases involving regulatory complaints.

The enterprise therefore should not ask:

REJECTED
How much authority can this model handle?

GOVERNING
What is the minimum authority this machine actor needs to perform this role?

That is the machine equivalent of least privilege.

It is also an operating model principle.

06 · Delegation Requires Identity

Authority cannot be governed if the enterprise cannot identify the actor using it.

Human enterprises already rely on identity.

01 Employees authenticate.
02 Roles are assigned.
03 Permissions are granted.
04 Actions are logged.

05 Access changes when responsibilities change.

06 Credentials expire.

07 Departing employees lose access.

Machine actors need an equivalent lifecycle.

Microsoft's 2026 agent identity architecture reflects this requirement by assigning distinct identities to agents and linking those identities to human sponsors responsible for their access lifecycle.

NIST's work on software and AI agent identity addresses the same problem from a standards perspective.

The significance extends beyond cybersecurity.

Identity creates the foundation for organizational accountability.

The enterprise needs to know:

01 Which agent acted?

02 Which version?

03 On whose behalf?

04 Under which role?

05 Using which authority?

06 Against which resource?

07 At what time?

08 With what result?

Without that information, delegated authority becomes difficult to distinguish from uncontrolled access.

07 · An Agent Should Not Borrow a Human's Authority Invisibly

One of the simplest ways to deploy an agent is to let it act using a human user's credentials or permissions.

That can be convenient.

It can also collapse an important boundary.

If the machine acts entirely as the person, the enterprise may lose the ability to distinguish human action from machine action.

It may also give the machine every permission the human possesses, even when the workflow requires only a subset.

That creates an authority problem.

The agent should be able to act on behalf of a person without becoming indistinguishable from that person.

The execution record should preserve both identities.

THE PRINCIPAL

Who delegated the authority?

THE MACHINE ACTOR

Who executed the action?

That distinction allows the enterprise to reason about delegation.

Fritzie authorized Agent A to perform Action B within Boundary C.

The agent executed the action.

The enterprise can reconstruct both facts.

This is more than an audit requirement.

It is the beginning of a machine accountability chain.

08 · Decision Rights Need Levels

Not every decision should be governed through the same mechanism.

A useful operating model distinguishes several levels of machine participation.

Observe

The machine can access information and monitor conditions.

It cannot recommend, decide or act.

Recommend

The machine can interpret information and propose an action.

A human retains the decision.

Prepare

The machine can assemble the action for execution.

A human must authorize it before it occurs.

Decide

The machine can choose the action within a defined decision domain.

Execution may still require a separate control.

Execute

The machine can perform the authorized action.

Escalate

The machine can determine that the condition exceeds its authority and transfer the decision to another actor.

These rights can be combined differently across a workflow.

An agent may observe and recommend in one part of the process.

Prepare but not decide in another.

Decide and execute routine cases.

Escalate everything outside its envelope.

This is how autonomy becomes granular.

Exhibit 05.2

The Decision Rights Ladder

OBSERVE
The machine can see.

RECOMMEND
The machine can propose.

PREPARE
The machine can construct the action.

DECIDE
The machine can choose within a defined domain.

EXECUTE
The machine can act.

ESCALATE
The machine recognizes the boundary of its authority and transfers control.

The ladder does not imply that every agent should move upward.

A machine may have extensive capability while intentionally remaining at Recommend.

Another may operate at Execute for a narrow class of low consequence transactions.

The appropriate level depends on the workflow, consequence, reversibility, evidence, control environment and economics.

09 · Delegation Changes the Meaning of Human Approval

Human approval is often treated as the safe default.

But Chapter 04 established that human presence alone does not create control.

The same applies to delegated authority.

If every machine action requires human approval, the machine may increase execution capacity while leaving decision capacity unchanged.

The result can be an acceptance bottleneck.

01 Agents prepare more work.

02 Humans receive more requests.

03 Queues grow.

04 Approval becomes faster but shallower.

Eventually, the organization either constrains machine execution or humans begin approving without meaningful review.

Neither produces the intended operating leverage.

The alternative is not to remove human authority.

It is to place human authority where it creates value.

Routine, bounded, observable and reversible decisions may support delegated machine authority.

Ambiguous, consequential or irreversible decisions may remain human.

Some workflows may use thresholds.

Others may use risk tiers.

Others may require human approval only when the evidence conflicts or the system leaves its expected operating range.

The design problem is to preserve human control without forcing every unit of machine execution through a human decision.

Sentient Operating Implication

> *Human authority should govern the system. It does not have to interrupt every execution.*

10 · Escalation Is Part of Authority

Delegation is incomplete without a mechanism for giving authority back.

An agent needs to know not only what it can do.

It needs to know when it should stop.

That condition may occur when:

- confidence falls
- required evidence is missing
- systems disagree

- financial exposure exceeds a threshold
- policy conflicts
- the case is novel
- the requested action is irreversible
- a customer challenges the decision
- an external condition changes
- a control fails

or the agent encounters a situation outside its defined domain.

Escalation is therefore not an exception bolted onto autonomous execution.

It is part of the authority design.

A well designed machine actor understands the boundary of its mandate.

The operating model must make that boundary explicit.

11 · Authority Can Decay

Human organizations often treat permission as persistent.

A person receives access because of a role and keeps it until the role changes.

Agentic execution creates reasons to make authority more temporary.

An agent may need permission only:

- for one workflow
- for one transaction
- for one customer
- for one period
- for one incident
- for one objective

or until a particular condition changes.

Persistent machine authority increases the surface on which failure or misuse can occur.

Temporary authority can narrow it.

The enterprise can therefore design authority that expires.

An agent receives the right to act.

It performs the work.

The authority terminates.

New execution requires new authorization or a renewed policy condition.

This makes delegated authority more contextual.

It also aligns permission more closely with the work that justified it.

Sentient Operating Implication

> *Do not ask only: What can this agent access? Ask: What may this agent do, to what, on whose behalf, under which conditions, for how long, and when must it stop? That is the authority model.*

12 · Accountability Has to Survive Delegation

Authority can move.

Accountability cannot simply disappear with it.

When a manager delegates a decision to an employee, the organization still knows where accountability sits.

When an executive delegates spending authority, the enterprise does not treat the transaction as ownerless.

When a board delegates operating authority to management, the delegation exists inside a larger governance structure.

Machine delegation needs the same principle.

An agent may execute the action.

It may make the bounded decision.

It may operate without human approval at the moment of execution.

But the enterprise still needs an accountable human or institutional owner for the authority that made the action possible.

This creates a chain.

01 A person or governing body authorizes an operating objective.
02 Authority is assigned to a role.
03 Some portion of that authority is delegated to a machine actor.
04 The machine executes within defined boundaries.
05 The execution is observed.
06 Exceptions return to an accountable actor.
07 The authority can be changed or withdrawn.

The chain must remain intact even when no human participates in the individual transaction.

That is how machine autonomy remains inside institutional accountability.

13 · The Accountability Chain

The AI Native enterprise needs to be able to reconstruct not only what happened, but why the machine had the right to make it happen.

For a consequential machine action, the enterprise should be able to answer:

01 Who owns the workflow?

02 Who authorized machine participation?

03 Which machine actor executed the action?

04 What decision right had been delegated?

05 What authority envelope applied?

06 What evidence was available?

07 Which policy governed the action?

08 What controls were evaluated?

09 What action occurred?

10 What outcome followed?

11 Was escalation required?

12 Who remained accountable?

This is different from a technical log.

A technical log may tell the enterprise which service called which application.

An accountability chain explains the institutional basis for the action.

The distinction matters.

A machine action can be technically valid and still be organizationally unauthorized.

It can possess the credentials required to perform an action without possessing a legitimate decision right to perform it.

The AI Native control system has to preserve both.

TECHNICAL PERMISSION

The system allows the action.

INSTITUTIONAL AUTHORITY

The enterprise authorized the action.

Exhibit 05.3

The Delegated Authority Chain

ACCOUNTABLE OWNER

Who owns the outcome and remains responsible for the operating system?

DELEGATED MANDATE

What responsibility has been assigned to the machine actor?

AUTHORITY ENVELOPE

What actions are permitted, against which resources, within what limits?

RUNTIME CONTROL

Are the conditions for execution satisfied now?

MACHINE ACTION

What did the agent decide or execute?

EVIDENCE

What information, policy and system state supported the action?

OBSERVATION

What happened after execution?

ESCALATION OR CONTINUATION

Should the machine retain authority, return control or stop?

ACCOUNTABILITY

Who owns the consequence and decides whether the authority remains appropriate?

Delegation therefore forms a closed control loop.

Authority is granted.

Authority is exercised.

Execution creates evidence.

Evidence informs whether authority should continue.

14 · Policy Has to Enter the Execution Path

Traditional governance often sits outside execution.

01 Policies are written.

02 Employees are trained.

03 Controls are reviewed.

04 Audits test whether the organization complied.

That model becomes weaker when machine actors can perform thousands of actions at machine speed.

The control cannot depend only on a person remembering the policy before acting.

Policy increasingly has to enter the execution path.

If an agent cannot spend more than a defined amount, the system should enforce the limit.

If particular customer data cannot leave a jurisdiction, the workflow should prevent the action.

If an agent requires human approval before modifying a production environment, the execution architecture should enforce that gate.

If an action requires two independent conditions to be satisfied, the machine should not be able to bypass them because its reasoning concluded that doing so would help achieve the objective.

TELLING THE AGENT

The policy is communicated to the machine.

MAKING IT OPERATIVE

The policy is part of the operating environment.

Prompts can communicate intent.

They should not carry the entire burden of institutional control.

The AI Native enterprise needs controls that exist outside the reasoning of the machine actor.

Sentient Operating Implication

> *Where a boundary matters, enforce it in the execution architecture. Do not rely on the agent to remember that the boundary exists.*

15 · Runtime Authorization Changes the Control Model

Traditional access control often asks whether a user has permission to access a resource.

Agentic execution requires a more contextual question.

> *Is this actor authorized to perform this action, against this resource, for this objective, under these conditions, at this moment?*

The answer may depend on more than identity.

It may depend on:

- workflow state
- transaction value
- customer classification
- data sensitivity
- risk level

- location
- time
- prior actions
- available evidence
- model confidence
- the presence of required approval

or whether another control has already failed.

Authority therefore becomes dynamic.

An agent may possess a general role while individual actions remain subject to runtime authorization.

This allows the enterprise to separate broad capability from narrow permission.

The machine can know how to perform many actions.

The control system determines which action is permitted now.

That is a more durable model for enterprise autonomy than granting broad standing permissions to increasingly capable agents.

16 · Evidence Should Travel With the Decision

Human organizations often separate the decision from the evidence that produced it.

A manager approves.

A system records the approval.

The reasoning may remain in an email, a meeting or the manager's head.

Machine execution creates an opportunity to improve this.

A consequential machine decision can carry an evidence record.

01 What information was considered?
02 Which policy applied?
03 What constraints were active?
04 What alternatives were rejected?
05 What evaluation was performed?
06 What confidence or uncertainty was present?
07 What authority permitted the action?

The purpose is not to preserve every token of model reasoning.

Nor should enterprises assume that internal model reasoning provides a reliable explanation of why an action occurred.

The useful evidence is operational.

Inputs.

Retrieved information.

Tool results.

Policy state.

Evaluations.

Approvals.

Actions.

Outcomes.

That record gives the enterprise something it can inspect.

It supports audit.

It supports investigation.

It supports evaluation.

And it creates the evidence required to decide whether the authority envelope remains appropriate.

17 · Observation Is Part of Delegation

Authority should not be granted and forgotten.

A machine actor can remain inside its formal permissions while producing poor outcomes.

It can make technically valid decisions that create unacceptable error rates.

It can consume far more resources than expected.

It can escalate too often.

It can fail to escalate when it should.

It can produce patterns of action that reveal a flaw in the workflow design.

The enterprise therefore needs to observe not only whether the machine violated a rule.

It needs to observe whether delegated authority is producing the intended operating result.

That means measuring:

- decision quality
- execution quality

- exception rate
- escalation rate
- human override
- failure patterns
- cost
- latency
- outcome quality
- control violations

and changes in the environment in which the authority was granted.

The authority envelope is a hypothesis.

The enterprise is asserting that this machine actor can exercise these rights under these conditions and produce an acceptable result.

01 Authority

02 Execution

03 Evidence

04 Observation

05 Authority change

Execution tests that hypothesis.

Observation determines whether it remains true.

18 · Authority Should Expand Through Evidence

Enterprises will face pressure to increase machine autonomy.

As systems improve, requiring human approval for every action will become economically difficult to justify.

But autonomy should not expand because the technology category has become more fashionable.

It should expand because evidence supports the delegation.

A machine may begin at Recommend.

Performance is observed.

The enterprise develops confidence in a bounded class of decisions.

The machine moves to Prepare.

Controls improve.

Evaluation demonstrates reliable performance.

A narrow class of decisions moves to Decide.

Execution remains separately controlled.

Later, some actions may move to Execute inside a defined envelope.

The progression is not mandatory.

Some decisions may remain human indefinitely.

Some may move rapidly.

Some may move forward and later move back.

The principle is more important than the direction.

Authority should follow evidence.

Not capability claims.

Not benchmark headlines.

Not pressure to automate.

Evidence from the workflow.

19 · Authority Must Also Contract

Delegated authority needs a reverse path.

A machine that performed well yesterday may encounter a changed environment tomorrow.

- The underlying model may change.
- A data source may degrade.
- A policy may change.
- A new fraud pattern may appear.
- A workflow dependency may fail.
- An evaluation may reveal a systematic error.
- A regulator may change the permissible operating boundary.

The enterprise needs the ability to narrow or revoke authority without redesigning the entire workflow.

That may mean:

- lowering a financial threshold
- requiring additional evidence
- increasing human review
- disabling a tool
- removing access to a resource

- changing the escalation condition
- moving execution back to Prepare

or stopping the machine actor altogether.

Authority is therefore not a permanent property of the agent.

It is a configurable property of the operating system.

20 · Conflicting Authority Has to Be Resolved

Mixed execution systems can contain many actors.

01 Humans.
02 Agents.
03 Specialized agents.
04 Supervisory agents.
05 Applications.
06 Policy engines.
07 External services.

Those actors may not always agree.

A customer agent may determine that a refund is appropriate.

A fraud control may block it.

A revenue policy may permit it.

A financial threshold may require human approval.

A supervisory agent may detect conflicting evidence.

The operating model needs precedence.

01 Which authority wins?
02 Which control can stop execution?
03 Which actor can override another?
04 Can a human override the machine?
05 Can every human override it, or only designated roles?
06 Can one agent authorize another?
07 Can a supervisory agent revoke a subordinate agent's authority?

These are not edge cases once agent populations grow.

They are organizational design questions expressed through software.

Traditional enterprises resolve authority conflicts through hierarchy, policy and governance.

AI Native enterprises will need machine readable equivalents.

21 · Delegation Creates a New Separation of Duties Problem

Enterprises have long separated sensitive responsibilities.

The person who creates a vendor may not be the person who approves payment.

The engineer who writes a change may not be the only person who authorizes its release.

The person who initiates a financial transaction may not be the person who reconciles it.

These controls exist because concentration of authority creates risk.

Agents do not eliminate that principle.

They can make it easier to violate.

One agent may be technically capable of gathering evidence, making the decision, executing the action and evaluating the result.

That may be efficient.

It may also concentrate too much authority in one machine actor.

Some workflows should separate those responsibilities.

01 One agent may prepare.
02 Another control may validate.
03 A human may authorize.
04 A separate system may execute.
05 Another evaluator may inspect the outcome.

The correct separation depends on consequence and risk.

The principle remains familiar.

Do not confuse the ability to consolidate execution with the wisdom of consolidating authority.

22 · Human Intervention Needs Authority Too

The human side of the workflow also needs explicit decision rights.

Escalation is useless if the person receiving the work cannot resolve it.

A machine may correctly identify that a case exceeds its authority.

But if the case lands with a human who lacks the required information, expertise or permission, the workflow simply creates another queue.

The enterprise therefore needs to design the receiving side of escalation.

01 Who receives the case?

02 What authority do they possess?

03 What context travels with it?

04 Can they override the machine?

05 Can they modify the policy?

06 Can they expand the machine's authority for this case?

07 Can they return the work with additional instruction?

08 Does their decision become evidence for future execution?

Human intervention is not outside the authority architecture.

It is part of it.

23 · The Goal Is Not Human Approval Everywhere

The safest looking operating model is often one in which a person approves every consequential machine action.

At small scale, that may work.

At AI Native scale, it can defeat the operating model.

If machines generate execution faster than humans can authorize it, human approval becomes the system constraint.

The enterprise then has three choices.

Slow the machines.

Add more approvers.

Or redesign authority.

The third option is where operating leverage appears.

Routine decisions can move into bounded machine authority.

Humans can retain decisions where consequence, ambiguity or accountability require them.

Escalation can concentrate human attention on the cases where it creates value.

The purpose of delegated authority is therefore not to remove human control.

It is to separate human control from mandatory human participation in every transaction.

That is a different architecture.

Human control moves upward.

FROM APPROVING EVERY ACTION

To defining which actions can occur without approval.

FROM INSPECTING EVERY EXECUTION

To designing the evidence and thresholds that determine when inspection is required.

FROM BEING THE EXECUTION GATE

To governing the system that creates the gate.

Exhibit 05.4

From Human Approval to Accountable Autonomy

HUMAN APPROVAL

Every material action waits for a person. Control is concentrated at the transaction.

BOUNDED DELEGATION

Routine actions execute inside explicit limits. Humans retain defined decisions and exceptions.

RUNTIME GOVERNANCE

Policy, identity, evidence and workflow state determine whether execution is permitted. Humans govern the authority model.

ACCOUNTABLE AUTONOMY

Machine actors execute within bounded authority. Execution is observable. Escalation is explicit. Authority expands or contracts through evidence. Human accountability remains intact.

Accountable autonomy does not mean the absence of humans.

It means human authority has been designed into the operating system rather than inserted manually into every action.

24 · Sentient Interpretation

The central governance problem of the AI Native enterprise is not whether machines become autonomous.

It is whether autonomy becomes accountable.

Machine execution creates leverage because it can continue without requiring human participation in every step.

But that same property creates risk if the enterprise cannot explain why the machine had the right to act.

The answer is not to keep machines permanently advisory.

Nor is it to grant broad autonomy and attempt to govern the consequences afterward.

The answer is delegated authority.

- Explicit identity.
- Explicit decision rights.
- Explicit boundaries.
- Explicit controls.
- Explicit escalation.
- Explicit accountability.

The machine should know what it is permitted to do.

The execution environment should enforce the boundaries that matter.

The enterprise should be able to reconstruct the authority chain behind consequential action.

And authority should change when execution evidence changes.

This creates a different relationship between humans and machines.

Humans do not need to make every decision.

They need to determine which decisions can be delegated.

They do not need to execute every control.

They need to design controls that remain effective when execution is performed by machines.

They do not need to inspect every action.

They need evidence that tells them when intervention is required.

They do not surrender authority. They architect it.

That is accountable autonomy.

25 · Operating Implication

For every workflow containing machine execution, define the decision rights explicitly.

Start with the machine actor.

Give it an identity.

Identify the accountable owner.

Define the objective it is authorized to pursue.

Then define what it may:

- observe

- recommend
- prepare
- decide
- execute

and escalate.

For every consequential action, define the authority envelope.

01 What can it do?

02 To what?

03 On whose behalf?

04 Under which conditions?

05 Within what threshold?

06 Using what evidence?

07 For how long?

08 What control must pass?

09 What condition terminates the authority?

10 Where does escalation go?

11 Who remains accountable?

Then enforce the material boundaries in the execution architecture.

Do not leave consequential authority inside prompts alone.

- Instrument the decisions.
- Preserve operational evidence.
- Measure outcomes.
- Measure overrides.
- Measure exceptions.
- Measure failures.
- Measure the cost of control.

Review whether the delegated authority remains appropriate.

Expand it when evidence supports expansion.

Contract it when evidence supports contraction.

Revoke it when the operating conditions no longer justify it.

Delegated authority is not a permission granted to an agent once.

It is an operating system the enterprise has to manage.

26 · Connection to the AI Native Operating Model

Chapter 03 changed the unit.

The workflow became the primary object of operating design.

Chapter 04 changed the allocation.

Humans and machines became participants in execution.

Chapter 05 changes the authority model.

Machines can receive decision rights and execute within explicit boundaries while accountability remains human and institutional.

Chapter 03 — The workflow became the unit.

Chapter 04 — Humans and machines became participants in execution.

Chapter 05 — Machines receive explicit decision rights and bounded authority. ←

Chapter 06 — Management itself has to change.

That creates the next problem.

Management.

Traditional management assumes that execution is performed by people.

01 Managers allocate work to people.

02 Monitor people.

03 Coordinate people.

04 Review their output.

05 Resolve their exceptions.

06 Develop their capability.

07 Adjust capacity by changing teams and headcount.

But what happens when part of the execution system is no longer human?

01 When capacity can be instantiated through software?

02 When agents can receive objectives rather than individual tasks?

03 When routine coordination can itself become computational?

04 When machine actors can execute continuously inside delegated authority?

05 When a manager is responsible for outcomes produced by people and machines together?

The management system has to change.

The manager is no longer responsible only for supervising human activity.

The manager increasingly governs an execution system.

People.

Agents.

Authority.

Controls.

Exceptions.

Capacity.

Outcomes.

The next question is therefore not whether managers remain.

> *It is what management becomes when execution itself has changed.*

Chapter 06 · Management After Machine Execution

Read online: https://sentient-review.com/perspectives/the-ai-native-operating-model/05-decision-rights-and-delegated-authority

Chapter 06

Management After Machine Execution

How management changes when execution and coordination increasingly involve machines.

> *How should managers direct, coordinate, evaluate and adapt an execution system composed of people and machines?*

Management was built around a basic assumption.

People perform the work.

Managers organize them.

They assign work, coordinate dependencies, monitor progress, resolve exceptions, review output, develop capability and adjust capacity through teams and headcount.

That assumption shaped the modern managerial system.

It shaped reporting lines.

It shaped spans of control.

It shaped meetings, status reports, approvals, performance reviews and escalation paths.

It shaped what organizations meant by capacity.

It also shaped what organizations meant by management.

That assumption is beginning to break.

Machine intelligence is moving beyond assistance into execution. Agents can receive objectives, perform sequences of work, interact with systems, produce outputs, evaluate conditions and take actions within defined authority.

The enterprise is no longer coordinating only human effort.

It is beginning to coordinate human and machine execution together.

This changes the management problem.

The question is no longer only:

> *How should managers organize people to produce an outcome?*

It becomes:

That is a different managerial architecture.

The previous chapters established the progression.

- The workflow defines the outcome.
- Allocation determines who or what performs the work.
- Authority determines who or what may decide and act.
- Management determines how the resulting execution system is directed over time.

The management system now has to govern more than people.

It has to govern execution itself.

01 · The Management System Was Built for Human Execution

Much of management exists because human execution creates coordination requirements.

- People need context.
- They need priorities.
- They need information from other people.
- They need decisions.
- They need clarification.
- They need work routed to them.
- They need progress communicated.
- They need conflicts resolved.
- They need exceptions escalated.

Organizations built managerial structures around these requirements.

Managers became information nodes.

- They translated strategy into work.
- They connected teams.
- They gathered status.
- They redistributed information.
- They reconciled competing priorities.
- They decided what moved next.
- They carried knowledge between organizational layers.

This created an organizational logic in which coordination capacity was closely tied to managerial capacity.

A manager could supervise only so many people.

A team could absorb only so much work.

A department could coordinate only so many dependencies before another layer of management became useful.

Growth therefore produced hierarchy.

More work created more workers.

More workers created more teams.

More teams created more managers.

More managers created more managerial layers.

Machine execution begins to disturb this relationship.

Not because coordination disappears.

Because some forms of coordination can themselves become computational.

02 · Coordination Can Become Part of the Execution System

Consider the ordinary machinery of organizational coordination.

01 A task is created.

02 Someone determines who should perform it.

03 Context is gathered.

04 The work is routed.

05 Progress is checked.

06 Dependencies are identified.

07 Another person is contacted.

08 A status update is produced.

09 An exception appears.

10 Someone determines whether it requires intervention.

11 The work is reviewed.

12 The next action is assigned.

Much of this activity has historically required human attention because the organization had no other mechanism capable of interpreting the state of work and responding to it.

That condition is changing.

AI systems can now participate in work routing, information retrieval, progress synthesis, dependency detection, monitoring, exception identification and multistep workflow execution.

The consequence is easy to underestimate.

NOT ONLY

AI is changing who performs a task.

BUT ALSO

It can change how work is coordinated between tasks.

That distinction matters.

Automation traditionally removed execution from individual activities.

Agentic systems can begin to remove coordination work from the spaces between activities.

This is where the management problem becomes structural.

03 · Market Evidence

Early evidence points toward a reduction in demand for several managerial coordination functions.

A 2026 Management Science study examined hiring demand across 1,820 publicly listed US companies around the introduction of ChatGPT. Firms with greater exposure to generative AI showed declining demand for skills associated with task division, information provision, coordination, synchronous and asynchronous communication, monitoring and some forms of exception management. The authors connect these changes to the possibility of wider managerial spans as AI reduces the coordination burden associated with supervising work.

A separate 2026 study presented through the Academy of Management examined managerial hiring after the release of ChatGPT. In firms with greater exposure to generative AI, managerial hiring fell by an estimated 19 to 27 percent relative to less exposed firms. The researchers also observed reductions in manager to manager reporting relationships and interpreted the pattern as early evidence that organizations may require fewer managerial intermediaries when employees can access knowledge and solve some problems without routing them upward through hierarchy.

These findings do not establish that AI will eliminate middle management.

They establish something narrower and more useful.

Some managerial work exists because information, coordination and problem solving have historically required human intermediaries.

When those functions change, the organizational demand for managerial intermediation can change with them.

What This Supports

There is emerging empirical evidence that generative AI exposure is associated with reduced demand for some coordination and information functions traditionally embedded in managerial roles.

There is also early evidence consistent with wider managerial spans and fewer intermediary reporting relationships.

What Remains Unresolved

The evidence does not establish the eventual organizational form.

It does not tell us how far spans can expand.

It does not establish which managerial layers remain necessary once agentic execution becomes widespread.

It does not prove that fewer managers produce stronger organizations.

Most important, much of the available evidence captures the first generation of generative AI adoption rather than mature enterprises operating large populations of autonomous agents.

The organizational experiment has begun.

Its final structure has not been determined.

04 · From Coordinating People to Orchestrating Execution

A useful distinction is emerging.

TRADITIONAL MANAGEMENT

Coordinates actors.

AI NATIVE MANAGEMENT

Increasingly orchestrates execution.

These are not the same activity.

Actor coordination asks:

- Who is doing this?
- Who owns that?
- Who needs to speak with whom?
- Who is blocked?
- Who has capacity?
- Who should approve this?

Execution orchestration asks:

- What outcome is required?
- What work must occur?
- Which execution mode should perform it?
- What authority is available?
- What dependencies exist?

- What evidence must be produced?
- Which conditions require intervention?
- Where should the system continue without human participation?

The center of gravity moves.

The manager becomes less dependent on continuously moving information between people.

The manager becomes more responsible for designing the conditions under which work can move.

This is already visible in the way major technology companies describe human and agent collaboration.

Microsoft's 2026 Work Trend Index describes four modes of AI enabled work: author, editor, director and orchestrator. At the orchestrator level, people design systems in which multiple agents execute work across a workflow and bring exceptions or escalations back to humans. Microsoft argues that as tactical execution moves toward agents, human responsibility shifts toward setting direction, defining standards and evaluating outcomes.

The terminology is Microsoft's.

The operating consequence is broader.

When machines can execute work across a workflow, management moves one level above individual activity.

The manager increasingly manages the conditions of execution.

05 · The Managerial Control Loop Changes

Traditional management often operates through a human control loop.

- Assign
- Observe
- Ask
- Interpret
- Intervene
- Reassign

The loop depends on human communication.

A manager asks for progress.

Someone reports it.

The manager interprets the report.

The manager decides whether intervention is required.

The next instruction is issued.

Machine execution creates another possibility.

- Define Outcome
- Configure Execution
- Observe Evidence
- Detect Variance
- Evaluate Condition
- Continue or Escalate
- Adapt

The distinction is important.

THE FIRST LOOP

Manages activity through communication.

THE SECOND

Manages execution through instrumentation.

That instrumentation may include:

- traces
- evaluations
- outcome measures
- policy checks
- exception signals
- cost measures
- latency
- quality thresholds
- authority boundaries
- runtime events

The manager does not need to ask an agent how the work is going in the same way a manager asks an employee.

The execution system can expose its state.

This changes supervision.

06 · Supervision Becomes Observability

Human organizations developed supervision because work was difficult to observe directly.

Managers therefore relied on proxies.

- Meetings.
- Status reports.
- Check ins.
- Progress updates.
- Dashboards assembled by people.
- Escalations.
- Review cycles.

These mechanisms convert invisible activity into managerial information.

Machine execution produces another source of evidence.

The system itself can emit traces of what occurred.

- What objective was received.
- What context was used.
- Which tools were called.
- Which decisions were made.
- Which actions were taken.
- What output was produced.
- How long execution took.
- What it cost.
- Which constraints were triggered.
- Where confidence fell.
- Where escalation occurred.

This does not eliminate supervision.

It changes its technical foundation.

Supervision can move from periodic human reporting toward persistent observation of execution.

That creates a stronger possibility than traditional managerial visibility.

The organization may be able to observe machine execution at a level of detail that would be impractical, undesirable or impossible for human work.

But more data does not create better management by itself.

The manager cannot inspect every trace.

A large agent population could produce more execution evidence than any human organization can absorb.

The management problem therefore moves again.

The question is not:

> *Can we observe machine execution?*

It becomes:

> *What must the management system observe in order to know when human judgment is required?*

That is an exception architecture problem.

07 · Management by Exception Becomes a Design Requirement

The previous chapter established that human control does not require human approval for every machine action.

The same principle applies to management.

Human management does not require human supervision of every execution step.

If every agent action requires managerial attention, machine execution has not created operating leverage.

It has created another queue.

The manager becomes the bottleneck.

The organization may automate execution while preserving human coordination overhead.

That is not an AI Native management system.

The management architecture must determine which conditions allow execution to continue and which conditions require intervention.

Those conditions may include:

- authority violations
- policy violations
- material uncertainty
- high consequence decisions
- unexpected cost
- quality deterioration
- unresolved ambiguity
- conflicting objectives
- repeated failure
- novel conditions
- customer harm

- security events
- regulatory exposure

The managerial role moves toward the design and governance of these thresholds.

Managers still intervene.

But intervention becomes selective.

Routine execution proceeds inside the system.

Human attention concentrates where judgment has greater value.

This creates a new management objective.

REJECTED

Do not maximize human oversight.

GOVERNING

Design the minimum human intervention required to preserve accountable execution.

That is not managerial withdrawal.

It is managerial leverage.

08 · The Span of Management May No Longer Equal the Span of People

The traditional span of control measures how many people report to a manager.

That metric made sense when people were the primary execution capacity of the organization.

But consider a manager responsible for:

- eight employees
- twenty specialized agents
- three automated workflows
- two decision systems
- a compute budget
- several authority envelopes
- and a portfolio of exceptions generated across all of them.

How large is the manager's span?

Eight?

Twenty eight?

Thirty three?

None of these numbers captures the actual management problem.

The manager is responsible for an execution domain.

Its capacity is produced by a mixture of labor, software and machine intelligence.

Microsoft's 2025 Work Trend Index introduced the idea of a human agent ratio and argued that organizations will need to determine the appropriate mix of humans and agents for different forms of work. The same research found that 28 percent of surveyed managers were considering hiring AI workforce managers, while leaders expected teams to increasingly build, train and manage agents.

BCG research on agentic organizations has reached a related conclusion. Its work argues that when agents coordinate workflows, traditional spans of control can widen and managerial roles may shift toward orchestration of hybrid human and AI teams. In its survey, 45 percent of organizations identified as agentic AI leaders expected reductions in middle management layers.

Again, these are directional signals rather than a settled organizational law.

But they expose a weakness in the old metric.

HEADCOUNT

Measures human capacity.

IT DOES NOT MEASURE

Execution capacity.

An AI Native enterprise therefore needs another concept.

Exhibit 06.1

The Execution Span

Traditional Span of Control

Manager

People

Human Execution

AI Native Execution Span

Manager

Execution Domain

- People
- Agents
- Automated Workflows
- Decision Systems
- Authority Envelopes
- Compute Capacity

Outcome

The management question changes from:

> *How many people can this manager supervise?*

to:

> *How much execution capacity can this manager govern without degrading outcome quality, control or judgment?*

This is the Execution Span.

The Execution Span is not simply the number of agents plus the number of employees.

Different execution systems create different management loads.

A stable agent performing a bounded, observable process may require little managerial attention.

A human team operating in an ambiguous environment may require more.

A high consequence autonomous workflow may require significant governance even when its execution is technically reliable.

A poorly designed agent system may generate constant exceptions and consume more management capacity than the human process it replaced.

The relevant variable is therefore not actor count.

It is management load.

09 · Sentient Interpretation

The managerial span of the AI Native enterprise should be designed around the complexity of governing execution, not the number of humans contained inside an organizational box.

This separates two concepts that traditional organization design often treats as one.

WORKFORCE SIZE
How many people the organization employs.

EXECUTION CAPACITY
How much work the operating system can perform.

They are no longer equivalent.

A ten person organization may command substantial machine execution capacity.

A hundred person organization may use little.

Two managers with identical human headcount may therefore govern radically different operating systems.

The org chart cannot reveal this.

A manager's real domain begins to include the machine capacity, authority and workflows attached to the humans they lead.

10 · Operating Implication

This changes workforce planning.

The traditional capacity question is:

> *How many people do we need?*

The AI Native capacity question becomes:

> *What combination of human judgment, machine execution and compute capacity is required to produce the outcome?*

That question belongs to management.

Because once execution capacity stops mapping directly to headcount, managers must make capacity decisions across different forms of productive resource.

- Labor.
- Agents.
- Models.
- Compute.
- Tools.
- Data.
- Human expertise.
- External services.

The manager is no longer allocating only people.

The manager is allocating an execution portfolio.

And that changes one of the oldest responsibilities of management.

Capacity itself becomes something managers must architect

Capacity itself becomes something managers must architect.

That creates a second change.

Managers must decide not only how much capacity exists.

They must decide what kind of capacity should exist.

A unit of human capacity and a unit of machine capacity are not interchangeable.

HUMANS BRING

Judgment, contextual understanding, social interpretation, accountability, creativity and the ability to operate when the problem itself is uncertain.

MACHINE SYSTEMS BRING

Speed, replication, persistence, retrieval, computational scale and the ability to execute defined processes at volumes that do not map cleanly to human headcount.

The management problem is therefore not to substitute one for the other.

It is to construct the right execution system for the outcome.

This makes capacity allocation an operating design discipline.

11 · From Headcount Planning to Execution Capacity

Traditional workforce planning begins with people.

- How much work exists?
- How many people are required?
- What skills do they need?
- How should they be organized?
- Where should they sit?
- Who should manage them?

AI Native capacity planning begins earlier.

> *What execution capacity does the outcome require?*

Only then does the enterprise determine how that capacity should be composed.

- Some capacity may remain entirely human.
- Some may be human led and machine assisted.
- Some may be machine led and human governed.
- Some may operate through machine execution within delegated authority.

The allocation framework established earlier in this publication therefore becomes a management instrument.

Managers repeatedly make allocation decisions as capability, economics and operating conditions change.

The organization is no longer staffed once and then managed.

Its execution architecture can be continuously recomposed.

This has an important consequence.

The workforce becomes only one component of enterprise capacity.

That does not make the workforce less important.

It makes capacity larger than the workforce.

12 · Managers Begin to Manage Two Forms of Scarcity

Human organizations have always managed scarce resources.

Time.

Talent.

Capital.

Attention.

Machine execution introduces another resource directly into the operating model.

Compute.

An agent does not work for free.

Every execution consumes infrastructure.

- Models consume tokens.
- Agents make repeated model calls.
- They invoke tools.
- They retrieve context.
- They generate traces.
- They may invoke other agents.
- They may retry failed actions.
- They may continue operating long after the economic value of additional execution has fallen below its cost.

At sufficient scale, these are not merely technology expenses.

They become operating inputs.

The manager responsible for an execution domain therefore inherits a new capacity question:

> *How much machine intelligence should be consumed to produce this outcome?*

The answer cannot belong only to the technology organization.

Technology teams may provide the infrastructure, models, platforms and controls.

But the economic decision sits closer to the work.

A manager deciding whether an agent should perform ten reasoning cycles, invoke an expensive frontier model, escalate to a human expert or stop execution is participating in resource allocation.

Compute becomes part of managerial economics.

13 · The Managerial Resource Model Changes

The traditional resource model is familiar:

$$\text{People} \times \text{Time} \times \text{Cost}$$

AI Native execution adds another dimension:

$$\text{People} \times \text{Time} \times \text{Machine Capacity} \times \text{Cost}$$

But even this is incomplete.

Machine capacity can often be changed much faster than human capacity.

Hiring takes time.

Training takes time.

Organizational restructuring takes time.

Machine execution may be increased through additional runtime capacity, parallel execution or different model choices.

This creates a more elastic operating system.

It also creates a new failure mode.

Capacity becomes easier to consume.

A human team has natural physical constraints.

There are only so many employees.

There are only so many working hours.

Machine execution can scale until another constraint stops it.

- Infrastructure.
- Budget.
- Rate limits.
- Policy.
- System architecture.
- Or managerial discipline.

The organization therefore needs managers capable of reasoning about both labor economics and machine economics.

14 · Market Evidence

The emerging enterprise discussion around agentic AI increasingly treats machine execution as an operating resource rather than a conventional software feature.

McKinsey's 2026 research on technology workforce design argues that organizations deploying agents must reconsider the composition of technology work itself, including which capabilities remain internal, which are augmented by AI and how the resulting system produces measurable return on investment. Its research describes technology leaders increasingly becoming architects of organizational change rather than administrators of a fixed technology workforce.

The economics remain unsettled.

Agentic execution can reduce the human effort required for some workflows while increasing inference, infrastructure, monitoring and control costs. The economic unit therefore cannot be adoption.

It must be the outcome.

A workflow that uses more AI but produces no corresponding improvement in revenue, cost, cycle time, quality, risk or capacity has not created operating leverage merely because machine execution increased.

What This Supports

Machine capacity is becoming part of the resource system managers must govern.

The economics of an AI Native operating model therefore cannot be separated from management.

What Remains Unresolved

Organizations do not yet have mature accounting conventions for allocating agentic execution costs to business outcomes.

The appropriate managerial metrics remain under development.

Token consumption alone is inadequate.

Compute cost alone is inadequate.

Headcount reduction is inadequate.

The economic question has to be resolved at the workflow and outcome level.

Chapter 10 will examine the infrastructure implications.

Chapter 11 will examine the economic system.

For management, the immediate implication is simpler.

Managers increasingly allocate both labor and machine capacity.

15 · Delegation Changes When the Delegate Is a Machine

Capacity is only one part of management.

Managers also delegate.

Traditional delegation contains a social contract.

A manager assigns responsibility to another person.

- That person interprets the assignment.
- They may ask questions.
- They may exercise discretion.
- They may recognize conditions the manager did not anticipate.
- They may refuse an inappropriate instruction.
- They may escalate.
- They accumulate experience.

The manager develops a mental model of their judgment.

Over time, delegation changes because trust changes.

Machine delegation behaves differently.

The machine does not inherit authority merely because it receives work.

Chapter 05 established that execution and authority must remain distinct. Machine actors require explicit authority envelopes defining what they may observe, recommend, prepare, decide, execute and escalate.

That changes the managerial act of delegation.

The manager can no longer delegate only an objective.

Delegation increasingly requires specification of:

01 the objective

02 the permitted actions

03 the boundaries

04 the decision rights

05 the evidence required

06 the escalation conditions

07 the resource limits

08 the termination conditions

The quality of delegation becomes partially architectural.

16 · Intent Becomes a Management Interface

This creates a paradox.

As managers become less involved in individual execution steps, the quality of managerial direction becomes more important.

A human employee can compensate for incomplete instructions.

- They know the organization.
- They understand social context.
- They infer unstated expectations.
- They notice when an instruction conflicts with reality.
- They can walk into the manager's office and ask what was meant.

Machine systems may also interpret ambiguity, but the consequences are different when they can act repeatedly and at machine speed.

Poorly specified intent can propagate.

A vague objective given to one employee may produce one misunderstanding.

A vague objective embedded in an execution system may produce the same misunderstanding thousands of times.

The managerial problem therefore moves upstream.

Managers must become better at defining:

- What are we trying to achieve?
- What matters most?
- What must remain true?
- What may the system trade off?
- What may it never do?
- When must it stop?
- When must a human decide?

These are management questions.

They are also system design questions.

In the AI Native enterprise, the boundary between the two becomes thinner.

17 · Sentient Interpretation

Management has historically tolerated substantial ambiguity because human organizations contain enormous amounts of implicit context.

People compensate for weak operating design.

- They remember exceptions.
- They know who actually makes the decision.
- They understand which policy is enforced and which is ceremonial.

- They know which customer requires special treatment.
- They know when a manager's instruction should not be interpreted literally.
- They know when to break process.

Machine execution exposes this hidden organizational substrate.

An agent cannot reliably operate from an organization that cannot articulate how it operates.

This means AI Native transformation creates pressure for something enterprises have often avoided:

operational explicitness.

Objectives must become clearer.

Decision rights must become clearer.

Policies must become executable.

Exceptions must become identifiable.

Escalation paths must become real.

Accountability must become attributable.

Management therefore acquires another responsibility.

It must make the operating model legible enough for both humans and machines to execute it.

18 · Management Moves Upstream

This produces a broader shift.

Traditional management often discovers operating problems during execution.

- A deadline slips.
- A dependency appears.
- A customer complains.
- A team becomes overloaded.
- A manager intervenes.

The managerial system is reactive because much of the state of work becomes visible only after humans report it.

AI Native management has the potential to move intervention earlier.

- If execution is observable, managers can design thresholds before failure.
- If authority is explicit, managers can define boundaries before action.

- If workflows are instrumented, managers can identify variance before the final outcome deteriorates.
- If machine capacity is measurable, managers can detect economic inefficiency while execution is occurring.

Management begins to move from:

REJECTED

intervening in work

GOVERNING

designing the conditions under which intervention becomes necessary.

That is a profound change in managerial leverage.

19 · But Machine Execution Can Also Increase Management Load

There is a tempting assumption that agents automatically reduce managerial burden.

The evidence does not support that conclusion.

Gartner's 2026 research on agentic AI and manager oversight argues that managers increasingly become accountable not only for people but for the behavior, output and risk of autonomous agents operating within their teams. It warns that agentic systems can increase cognitive load because managers must validate objectives and constraints while evaluating machine generated execution and results. Gartner reports that 75 percent of surveyed chief human resources officers say managers are more overwhelmed than ever.

Its subsequent research on human agent workflows identifies performance, coaching and team dynamics as continuing managerial frictions rather than problems automatically removed by agents.

This matters.

AI can remove coordination work while simultaneously creating governance work.

LESS TIME COLLECTING STATUS

More time investigating anomalous execution.

LESS TIME ROUTING TASKS

More time configuring systems.

LESS TIME APPROVING ROUTINE WORK

More time defining authority.

LESS TIME MANAGING QUEUES

More time managing exceptions.

The work does not simply disappear.

Some of it moves.

Some of it becomes computational.

Some becomes more consequential.

20 · The Management Load Curve

This creates an important design principle.

Adding machine execution does not automatically increase managerial leverage.

Poorly designed machine execution can do the opposite.

Consider three states.

01 · Human Execution

The manager coordinates people directly.

Management load rises as human execution capacity rises.

02 · Machine Augmentation Without Operating Redesign

Agents are added.

But objectives remain ambiguous.

Authority remains unclear.

Exceptions are frequent.

Observability is weak.

Humans review most outputs.

The manager now coordinates people and supervises machine activity.

Management load may increase.

03 · AI Native Execution

Workflows are redesigned.

Allocation is explicit.

Authority is bounded.

Execution is observable.

Routine coordination becomes computational.

Exceptions are filtered.

Human attention concentrates on consequential judgment.

Only here does machine execution create substantial managerial leverage.

This distinction is critical.

AI ADOPTION

Can increase management work.

AI NATIVE OPERATING DESIGN

Can reduce management load per unit of execution.

Those are different propositions.

Exhibit 06.2

The Management Load Curve

HUMAN EXECUTION

More execution more people more coordination more management load

MACHINE AUGMENTATION WITHOUT REDESIGN

More execution people + agents duplicated coordination and oversight potentially greater management load

AI NATIVE EXECUTION

More execution mixed human and machine capacity computational coordination exception based human intervention management load grows more slowly than execution capacity

---

The objective is not to eliminate management load.

It is to decouple management load from execution scale.

That is one of the central sources of organizational leverage in an AI Native enterprise.

21 · Human Attention Becomes the Scarce Control Resource

As machine execution expands, something else becomes scarce.

Human attention.

An enterprise may eventually be capable of generating more machine decisions, actions, outputs and exceptions than its managers can inspect.

At that point, requiring human review does not create control.

It creates congestion.

Research on controllability in agentic systems reinforces this distinction. A 2026 survey of agentic AI control mechanisms notes that simply placing humans in the loop does not guarantee meaningful oversight. Excessive checkpoints can create cognitive overload, alert fatigue and superficial approval, while selective intervention can provide stronger control when the correct human is invoked with the correct information and authority.

This is not merely a technical systems problem.

It is a management design problem.

Human judgment has to be allocated where it produces the greatest control value.

That means organizations must determine:

- which events deserve managerial attention
- which can be resolved computationally
- which require domain expertise
- which require executive authority
- which require independent review
- which should stop execution entirely

Escalation itself becomes an allocation problem.

22 · The Manager Becomes an Exception Architect

Traditional managers handle exceptions.

AI Native managers increasingly design how exceptions are handled.

That distinction changes the role.

The manager must define the conditions under which normal execution is no longer sufficient.

Those conditions cannot all be predicted.

But many can be designed.

- A financial threshold is exceeded.
- A confidence threshold falls.
- A policy conflict appears.
- The system encounters a novel condition.
- Two agents reach incompatible conclusions.
- Customer impact exceeds tolerance.
- Execution cost exceeds expected value.
- The system requests authority it does not possess.
- Repeated retries indicate structural failure.
- The workflow stops behaving as expected.

Each event creates a decision:

- continue
- retry

- reroute
- constrain
- escalate
- terminate
- invoke human judgment

The manager's responsibility increasingly includes determining which of those decisions the system may make for itself.

That is management after machine execution.

Not constant supervision.

Not managerial disappearance.

Designed intervention.

23 · Management Becomes More Consequential Where It Remains Human

If routine coordination becomes computational, the residual human management role does not necessarily become smaller in importance.

It may become more concentrated.

Research on AI led teams provides an early indication of why. A 2026 experimental study found that AI systems could successfully perform leadership functions in collaborative tasks, but effectiveness depended heavily on the quality and timing of guidance. Human responses to AI leadership were shaped more by perceived competence and appropriateness than by superficial human likeness.

Other 2026 organizational research anticipates employees themselves becoming managers of multiple AI agents, moving between managerial levels as they direct, supervise and coordinate machine execution.

This complicates the traditional distinction between manager and individual contributor.

If an engineer directs six coding agents, evaluates their work, reallocates tasks, resolves exceptions and determines when machine output is acceptable, that engineer is performing managerial work even if no human reports to them.

If an analyst orchestrates research agents and determines which conclusions can enter a decision process, that analyst is exercising managerial control over execution.

If a salesperson operates agents responsible for research, qualification, preparation and follow up, the salesperson may command an execution capacity once associated with a small team.

Management capability therefore begins to distribute beyond the managerial hierarchy.

The organization may have fewer formal managers while requiring more people capable of management.

That is not a contradiction.

It is the consequence of machine execution.

24 · Sentient Interpretation

The AI Native enterprise does not simply flatten management.

It distributes managerial agency.

People who previously executed work may increasingly direct machine execution.

Managers who previously supervised people may increasingly govern execution systems.

Senior leaders who previously allocated headcount may increasingly allocate portfolios of human and machine capacity.

The managerial skill set moves outward even if managerial layers move downward.

This is why the future of management cannot be understood through headcount reduction alone.

The more important question is:

> *Where does managerial judgment move when execution becomes computational?*

Part of it moves into systems.

Part of it moves toward the edge.

Part of it remains with formal managers.

And part of it moves upward toward questions of intent, architecture and accountability.

The organizational consequence is not the disappearance of management.

It is the separation of management from managerial hierarchy.

Read online: https://sentient-review.com/perspectives/the-ai-native-operating-model/06-management-after-machine-execution

Chapter 07

Organizational Architecture

How AI changes functional boundaries, ownership, coordination and organizational design.

> *How should an enterprise represent outcomes, execution, authority and accountability when productive capacity is no longer entirely human?*

The organization chart was designed for a world in which productive capacity was human.

Every box represented a person.

Every team represented a collection of people.

Every reporting line represented a relationship through which authority, information and accountability moved.

Functions grouped human expertise.

Management layers coordinated human execution.

Headcount approximated organizational capacity.

The architecture made sense because the workforce and the execution system were largely the same thing.

They no longer are.

An AI Native enterprise contains productive capacity that may not appear anywhere on the organization chart.

- Agents execute work.
- Models produce decisions and recommendations.
- Automated workflows coordinate activity.
- Machine systems interact with customers, employees and other systems.
- Human teams may command execution capacity far larger than their headcount suggests.

The formal organization can therefore describe who works for the enterprise while failing to describe how the enterprise actually works.

That is the organizational problem created by machine execution.

The answer is not simply to put agents into boxes beside employees.

It is to reconsider what organizational architecture must represent.

The AI Native organization needs to make visible four things:

- outcomes
- execution
- authority
- accountability

The traditional organization chart captures only part of this system.

Organizational architecture must capture the rest.

01 · The Org Chart Is Becoming an Incomplete Model of the Enterprise

An organization chart answers an important question:

> *Who is accountable to whom?*

It does not answer several questions that become increasingly important in an AI Native enterprise.

- Which workflows produce the outcome?
- Which parts of those workflows are performed by humans?
- Which are performed by machines?
- Where does machine authority begin and end?
- Which human owns the resulting execution?
- Which systems coordinate across functional boundaries?
- Where does an exception go?
- Where is execution capacity concentrated?

These questions cannot be answered from reporting relationships alone.

Consider a customer service organization.

The formal chart might show:

- Chief Customer Officer
- Vice President, Customer Support
- Support Directors
- Support Managers
- Support Representatives

But the operating reality may include:

- an intake agent
- a classification agent
- a retrieval system
- a resolution agent

- a quality evaluator
- a routing system
- a customer communication agent
- human specialists
- escalation teams
- and managers governing the system.

The human hierarchy remains real.

But it is no longer the complete execution architecture.

A customer outcome may travel through several machine actors before a human participates.

The organization therefore has two structures operating at once.

THE FIRST

The human accountability structure.

THE SECOND

The execution structure.

AI Native organizational design must connect them.

02 · Market Evidence

This organizational question has moved beyond theory.

McKinsey's work on the emerging agentic organization argues that companies are beginning to operate with humans and AI agents working together at scale. Its analysis identifies changes not only to individual jobs but to organizational structure, including flatter hierarchies, more fluid teams, new human agent interaction patterns and greater emphasis on end to end workflows.

California Management Review reaches a more cautious conclusion. Its analysis of organizations composed of humans and AI agents argues that the arrival of machine actors does not eliminate the need for structure. Organizational boundaries still perform important functions including consolidation of expertise and assignment of accountability. The design challenge is to make those structures more fluid while preserving the mechanisms that allow the enterprise to govern itself.

Recent Deloitte research makes the same distinction explicit. AI may alter how work is performed, but accountability, decision rights, spans and organizational layers remain organization design concerns. Deloitte argues that where agents participate in organizational work, a human leader still needs responsibility for the agent's role and outcomes.

These observations establish an important boundary.

AI Native organization design does not begin with the assumption that hierarchy disappears.

It begins with the recognition that execution no longer maps cleanly onto hierarchy.

What This Supports

The presence of machine execution creates organizational relationships that conventional human reporting structures do not fully represent.

Organizations still require accountability.

They still require decision rights.

They still require concentrations of expertise.

They still require mechanisms for resolving conflict and allocating resources.

But those mechanisms now govern an execution system containing both humans and machines.

What Remains Unresolved

There is no settled organizational form for the AI Native enterprise.

Current examples remain early.

Different industries carry different regulatory, physical, operational and coordination requirements.

A software company, hospital, bank and industrial manufacturer should not be expected to converge on one organizational topology.

The evidence supports redesign.

It does not support a universal org chart.

03 · Organization Follows Execution

Traditional organization design frequently begins with structure.

- What functions should exist?
- How many layers should there be?
- What should report to what?
- Where should a capability sit?

The AI Native Operating Model begins somewhere else.

It begins with the outcome.

Then the workflow.

Then the allocation of work.

Then authority.

Then management.

Only then should organizational structure be determined.

The sequence matters.

- Outcome
- Workflow
- Execution Allocation
- Authority
- Management
- Organization

The organization is not the starting point.

It is the structural consequence of the execution system.

This reverses a common enterprise habit.

Organizations frequently attempt to introduce AI into existing structures without reconsidering whether those structures remain appropriate.

An AI capability is assigned to a function.

An automation team is created.

Agents are deployed inside existing departments.

Existing reporting lines remain intact.

The technology changes.

The organizational architecture does not.

This can create a mismatch.

Machine execution may cross boundaries that the organization was designed to preserve.

A customer workflow may cross marketing, sales, finance, operations and service.

A procurement workflow may cross business units, finance, legal, security and suppliers.

A product workflow may cross research, engineering, design, operations, support and commercial functions.

When agents participate across the workflow, they encounter the architecture of the enterprise directly.

- Every organizational boundary can become a coordination boundary.
- Every unclear decision right can become an execution stop.

- Every fragmented data environment can become a context failure.
- Every unresolved ownership question can become an accountability gap.

AI therefore exposes organizational architecture through execution.

04 · From Functional Structure to Outcome Architecture

Functions exist for good reasons.

Finance concentrates financial expertise.

Legal concentrates legal expertise.

Engineering concentrates technical expertise.

Sales concentrates commercial capability.

Human Resources concentrates workforce expertise.

The AI Native enterprise does not need to abolish these concentrations.

It needs another organizing dimension.

The outcome.

A workflow produces an outcome.

That workflow may cross several functions.

If machine execution operates across the workflow, the enterprise needs a mechanism that governs the outcome across those boundaries.

This creates two simultaneous organizational dimensions.

THE CAPABILITY STRUCTURE

Where expertise lives.

THE OUTCOME STRUCTURE

How execution moves across expertise to produce value.

Traditional organizations often privilege the first.

AI Native organizations must make the second explicit.

Exhibit 07.1

Two Architectures of the Enterprise

Capability Architecture

- Finance
- Legal
- Technology
- Operations

- Sales
- Service

PURPOSE

Concentrate expertise. Develop capability. Maintain standards. Provide professional accountability.

Outcome Architecture

- Customer Acquisition
- Customer Onboarding
- Order to Cash
- Product Delivery
- Incident Resolution
- Procure to Pay
- Employee Onboarding

PURPOSE

Produce enterprise outcomes across capabilities. Coordinate human and machine execution. Establish end to end accountability. Measure performance where value is produced.

Neither architecture is sufficient alone.

An enterprise organized entirely around functions can fragment outcomes.

An enterprise organized entirely around workflows can duplicate expertise and weaken professional capability.

The AI Native enterprise therefore needs both.

CAPABILITIES

Provide depth.

OUTCOMES

Provide flow.

Organizational architecture connects them.

05 · Machine Execution Makes Cross Functional Friction Visible

Human organizations are remarkably good at compensating for structural weakness.

- People create relationships across boundaries.
- They message colleagues.
- They schedule meetings.
- They maintain spreadsheets.

- They remember unofficial processes.
- They learn who actually has authority.
- They discover which systems contain reliable information.
- They create workarounds.

These behaviors allow poorly integrated organizations to continue operating.

They also hide the cost of fragmentation.

Machine execution is less forgiving.

An agent moving through an enterprise workflow needs access.

- Context.
- Identity.
- Tools.
- Authority.
- Policies.
- Interfaces.
- Escalation paths.
- Evidence.

If these differ unpredictably across organizational boundaries, execution becomes fragile.

The organization discovers that what appeared to be a technology integration problem is often an operating architecture problem.

- The agent cannot determine who owns the decision because the enterprise never made ownership explicit.
- The workflow cannot continue because two functions use incompatible definitions.
- The system cannot retrieve the required context because knowledge remains trapped inside a team.
- The action cannot execute because authority was designed around human intervention rather than machine participation.

The machine has not created the fragmentation.

It has exposed it.

06 · Sentient Interpretation

AI Native transformation creates pressure for organizational coherence.

The more execution becomes computational, the harder it becomes for the enterprise to depend on invisible human coordination as its integration layer.

For decades, organizations have used people to bridge structural gaps.

- Managers translated between functions.
- Employees carried context between systems.
- Analysts reconciled conflicting data.
- Operations teams manually routed exceptions.
- Experienced employees remembered how the enterprise actually worked.

This hidden coordination capacity allowed the formal operating model to remain incomplete.

Machine execution changes the economics of that arrangement.

If the enterprise wants execution to scale, the structure underneath the execution must become more explicit.

The organization needs to know:

- who owns the outcome
- which capabilities participate
- which authority applies
- which information is required
- which systems may act
- which conditions require escalation
- and who remains accountable when execution crosses boundaries.

The organizational architecture becomes part of the execution architecture.

07 · The Organization Needs Another Map

The traditional organization chart should not disappear.

- Human accountability still matters.
- People still need leaders.
- Professional capability still needs development.
- Employment relationships still exist.
- Performance, compensation, succession and culture remain human organizational concerns.

But the org chart needs a companion.

The enterprise needs a representation of how execution actually moves.

Not another process diagram buried inside an operations repository.

A first class organizational map.

It should show:

- outcomes

- workflows
- human owners
- machine actors
- decision rights
- authority boundaries
- cross functional dependencies
- escalation paths
- and accountability.

This is the Execution Architecture.

Exhibit 07.2

The Dual Organization

Human Organization

- People
- Roles
- Teams
- Functions
- Reporting relationships
- Leadership accountability
- Capability development

Execution Organization

- Outcomes
- Workflows
- Human execution
- Machine execution
- Agent relationships
- Decision rights
- Authority
- Controls
- Escalations
- Execution ownership

CONNECTING LAYER

Every material execution system requires: A human accountable owner

---

The machine system can execute.

The workflow can coordinate.

Authority can be delegated.

But accountability must remain attributable.

The AI Native enterprise therefore does not replace the organization chart.

It makes the organization chart part of a larger organizational architecture.

08 · Agents Do Not Need Employee Boxes

There is an emerging temptation to depict AI agents as employees.

The metaphor is useful up to a point.

- Agents can perform roles.
- They can receive objectives.
- They can interact with other agents.
- They can produce outputs.
- They can operate with delegated authority.

But organizational architecture should preserve an important distinction.

An agent is not an employee.

- It does not possess institutional accountability.
- It does not carry fiduciary responsibility.
- It does not have an employment relationship.
- It does not become morally or legally responsible for the enterprise outcome simply because it performed the action.

This matters because organizational diagrams communicate responsibility.

If an agent occupies a conventional employee box, the diagram can imply an accountability the machine cannot carry.

The better design is to represent machine actors inside the execution architecture while connecting them to accountable human ownership.

That distinction is increasingly reflected in current organization design research, which emphasizes that human leaders remain responsible for the roles and outcomes associated with agents even as machine execution becomes embedded in organizational structures.

The governing principle is simple.

MACHINE EXECUTION

Can be distributed.

HUMAN ACCOUNTABILITY

Cannot become ambiguous.

That principle becomes more important as the number of machine actors increases.

Because the AI Native enterprise is unlikely to contain one agent.

It may contain thousands.

And once machine execution reaches that scale, organizational architecture faces a new problem:

> *How should an enterprise organize a machine workforce that can expand far faster than the human organization around it?*

Part 2

> *How should an enterprise organize a machine workforce that can expand far faster than the human organization around it?*

The answer cannot be another hierarchy.

Human hierarchies exist partly because human coordination is expensive.

People require manageable reporting relationships.

Teams require leadership.

Managers have finite attention.

Information moves imperfectly.

Organizations therefore divide people into bounded structures that make coordination possible.

Machine actors have different constraints.

- A new agent can be instantiated without hiring.
- A specialized agent can participate in several workflows.
- A group of agents can be assembled for an outcome and dissolved when the work is complete.
- Execution capacity can expand without adding an equivalent number of organizational layers.

This creates an organizational possibility that was difficult to achieve with human labor alone.

Execution structures can become more dynamic than employment structures.

The human organization may remain relatively stable.

The machine organization does not have to.

09 · The Machine Workforce Should Not Recreate the Human Org Chart

The easiest organizational response to agents is imitation.

Finance gets finance agents.

Marketing gets marketing agents.

Engineering gets engineering agents.

Each department builds its own machine capacity.

Each manager owns a collection of agents.

The existing organization chart becomes the template for the machine workforce.

Some of this will happen.

It may also reproduce the structural problems AI Native transformation is meant to address.

- Functional silos become agent silos.
- Duplicated work becomes duplicated agents.
- Fragmented data becomes fragmented machine context.
- Different teams implement different controls.
- Agents optimize local objectives.
- Cross functional workflows inherit the same handoffs and boundaries that constrained human execution.

The enterprise has automated its existing structure without redesigning it.

THAT IS

AI adoption.

IT IS NOT NECESSARILY

AI Native organization design.

Machine execution creates another option.

Agents can be organized around the workflow and outcome rather than around the department that happened to commission them.

10 · From Permanent Teams to Dynamic Execution Systems

Human organizations favor persistent teams.

There are practical reasons.

- People need continuity.

- Relationships matter.
- Shared context accumulates.
- Expertise develops over time.
- Culture emerges through repeated interaction.
- Employment itself is persistent.

Machine actors do not carry all of these constraints.

Some agents may be persistent.

Others may exist only for a workflow.

Others may be invoked for one decision.

Others may appear only when an exception occurs.

A customer onboarding workflow might invoke:

- an identity agent
- a risk agent
- a document agent
- a product configuration agent
- a compliance agent
- a communication agent
- and a human specialist.

Once the outcome is produced, that execution configuration may cease to exist.

Another customer may require a different combination.

The organizational unit becomes less permanent.

The workflow assembles the execution capacity it requires.

This creates a new organizational primitive.

The Dynamic Execution Team.

Exhibit 07.3

The Dynamic Execution Team

Traditional Team

Manager

Persistent group of employees

Work assigned to the team

Dynamic Execution Team

Outcome

Workflow determines required capabilities

- Human expertise
- Persistent agents
- Specialist agents
- Decision systems
- Tools
- Data
- Compute

Execution configuration assembled

Outcome produced

Configuration persists, changes or dissolves

The team is no longer defined only by who belongs to it.

It is defined by the execution capacity assembled around an outcome.

This does not mean human teams disappear.

It means the enterprise gains another form of organizational structure.

One optimized for execution rather than employment.

11 · Persistence Should Follow the Work

Not every agent should be temporary.

Some machine actors may require persistent identity.

- Persistent memory.
- Stable authority.
- Long term context.
- Continuous evaluation.
- A customer relationship agent may need continuity.
- A financial control agent may require persistent governance.
- A platform operations agent may continuously observe infrastructure.

Other agents may be disposable.

- A research agent may exist only for a particular investigation.
- A reconciliation agent may be created for a specific transaction.
- A specialist evaluator may appear only when a threshold is crossed.

The organization therefore needs to decide which machine roles should persist and which should be assembled on demand.

The governing question is not:

> *Should this agent have a permanent organizational position?*

It is:

> *What continuity does this execution responsibility require?*

Persistence becomes a design choice.

12 · Machine Workforces Require Lifecycle Ownership

Dynamic execution does not mean unmanaged execution.

Every machine actor has a lifecycle.

It is created.

Configured.

Given access.

Given tools.

Granted authority.

Evaluated.

Observed.

Updated.

Potentially reassigned.

Potentially retired.

Someone must own those decisions.

Without lifecycle ownership, agent populations can proliferate faster than organizational governance.

The enterprise can accumulate machine actors whose purpose is unclear.

- Agents whose permissions exceed their current role.
- Duplicated capabilities.
- Outdated prompts and policies.
- Unused integrations.
- Unknown dependencies.
- Machine identities that remain active after the workflow that created them has changed.

The problem resembles several earlier generations of enterprise technology.

Applications proliferated.

Cloud resources proliferated.

Software as a service proliferated.

Data proliferated.

Each expansion created value.

Each eventually required stronger lifecycle discipline.

Agents will be no different.

Except agents can act.

That makes lifecycle ownership more consequential.

Every material machine actor should have an identifiable lifecycle:

Purpose

Why does this agent exist?

Owner

Which human is accountable for its role?

Identity

How is the machine actor uniquely represented?

Capability

What can it do?

Authority

What may it do?

Resources

Which models, tools, data and compute may it consume?

Evaluation

How is its performance tested?

Observation

How is runtime behavior monitored?

Adaptation

How may its configuration change?

Retirement

When should its authority and access end?

This is not merely technical asset management.

It is organizational governance of machine capacity.

13 · Sentient Interpretation

The machine workforce creates a new organizational responsibility.

Someone must manage the population, not only the individual agent.

An enterprise with ten agents can govern them manually.

An enterprise with ten thousand cannot.

At scale, the organization needs mechanisms to know:

- what agents exist
- why they exist
- who owns them
- which workflows invoke them
- what authority they possess
- which systems they can access
- what they cost
- how they perform
- what dependencies they carry
- and whether they should continue to exist.

The enterprise therefore requires something analogous to workforce architecture for machines.

Not Human Resources for agents.

The analogy should not be taken that far.

But the underlying organizational problem is real.

Machine execution capacity needs inventory, ownership, standards, lifecycle controls and economic governance.

The agent population becomes an organizational asset class.

14 · Centralize the Control System, Not Necessarily the Execution

A second organizational question follows.

> *Where should AI capability live?*

The instinct may be to centralize it.

Create an AI organization.

Hire specialists.

Build platforms.

Approve use cases.

Control deployment.

Centralization can accelerate standards and reduce duplication during an early capability build.

But if machine execution becomes a persistent part of enterprise work, one central AI team cannot own every workflow.

The opposite model also fails.

If every business unit independently creates agents, models, controls and infrastructure, the enterprise can fragment rapidly.

The organizational problem is therefore not centralization versus decentralization.

It is determining what must be common and what must be distributed.

Exhibit 07.4

The Federated AI Native Organization

ENTERPRISE CONTROL LAYER

Common capabilities: Identity Security Model access Agent runtime standards Observability Evaluation infrastructure Policy enforcement Data controls Cost governance Auditability Core platforms

DISTRIBUTED EXECUTION DOMAINS

Business and functional domains own: Outcomes Workflows Allocation Domain context Local agent configuration Operational thresholds Exceptions Performance Economic value

ACCOUNTABILITY

Enterprise controls govern the environment. Domain leaders govern the outcome. Machine actors execute within both.

This is a federated organizational model.

The center establishes the conditions for safe, interoperable machine execution.

The domains determine how machine capacity produces business outcomes.

15 · Market Evidence

Current organizational research increasingly points toward this tension between common AI infrastructure and distributed business ownership.

McKinsey's research on the agentic organization argues that enterprises will need common platforms and governance while moving agentic capabilities closer to workflows and business teams. Its proposed organizational direction combines reusable enterprise capabilities with more flexible execution structures organized around outcomes. (mckinsey.com)

Deloitte similarly argues that organization design for AI requires explicit choices about where AI capabilities reside, how accountability is distributed and how centralized governance interacts with business ownership. (deloitte.com)

The evidence does not establish one optimal structure.

But the operating logic is becoming clearer.

MACHINE EXECUTION

Benefits from shared infrastructure.

BUSINESS OUTCOMES

Require distributed ownership.

The organizational architecture must support both.

What This Supports

AI Native execution is unlikely to remain the exclusive responsibility of a central AI organization.

Common infrastructure, policy and controls can be centralized or standardized.

Execution design must remain close enough to the domain to understand the outcome, workflow and consequences.

What Remains Unresolved

The appropriate balance will vary by enterprise.

Highly regulated environments may centralize more control.

Smaller organizations may combine several responsibilities.

Large global enterprises may require multiple layers of federation.

The correct architecture depends on consequence, scale, regulatory exposure and organizational capability.

The principle is more durable than the structure:

centralize what must be consistent

distribute what must understand the work.

16 · Organizational Boundaries Become Interfaces

Functions will remain.

Business units will remain.

Legal entities will remain.

Geographic structures will remain.

The AI Native enterprise does not eliminate boundaries.

It changes what boundaries must do.

A boundary can no longer function primarily as a wall.

It must increasingly function as an interface.

An execution system crossing from one domain to another needs to know:

- what capability is available
- what information can be accessed
- what authority is accepted
- what standards apply
- what evidence must be produced
- what service level is expected
- where exceptions go
- who owns the result.

The clearer these interfaces become, the easier execution can cross organizational structure without requiring constant managerial translation.

This is the organizational equivalent of an application programming interface.

Not because organizations should be reduced to software.

Because explicit interfaces reduce the amount of coordination that must be reconstructed every time work crosses a boundary.

17 · Flatter Does Not Mean Structureless

AI is often associated with flatter organizations.

There is logic behind the claim.

- If information can move without managerial relay, fewer communication layers may be necessary.
- If routine coordination becomes computational, some coordination roles may shrink.
- If individuals command greater machine capacity, teams may produce more with fewer people.
- If managers can govern wider execution spans, some managerial layers may compress.

But flattening is not the objective.

Execution effectiveness is the objective.

Hierarchy performs functions beyond information transmission.

- It resolves conflicts.
- Allocates resources.
- Establishes accountability.
- Integrates competing objectives.
- Provides escalation.
- Makes consequential decisions.

Some of these functions can change.

Some remain.

An organization that removes layers without redesigning decision rights and execution architecture may simply push unresolved coordination upward or downward.

The relevant question is therefore not:

> *How flat should the AI Native organization become?*

It is:

> *Which organizational layers still perform necessary work once execution and coordination have changed?*

Any layer that remains should have a reason to exist.

18 · Organizational Design Becomes Continuous

Traditional reorganizations are events.

A new strategy is announced.

Consultants map the structure.

Boxes move.

Reporting lines change.

Leaders are appointed.

The new organization stabilizes until the next reorganization.

That model assumes organizational structure should remain relatively persistent.

Machine execution introduces greater adaptability.

- Agents can be reconfigured.
- Workflows can change.
- Execution can move between humans and machines.

- Authority can expand or contract.
- Dynamic teams can form around changing outcomes.

The execution organization can therefore adapt more frequently than the employment organization.

This does not mean continuous restructuring of human reporting lines.

It means separating two rates of change.

HUMAN ORGANIZATIONAL STRUCTURE

Can remain relatively stable.

EXECUTION ARCHITECTURE

Can adapt continuously.

This may reduce the need to reorganize people every time the work changes.

Instead, the enterprise can reconfigure the execution system around them.

That is a different form of organizational adaptability.

19 · The AI Native Organizational Architecture

The pieces can now be assembled.

The AI Native enterprise requires more than an organization chart.

It requires an architecture connecting outcomes, capabilities, humans, machines, authority and accountability.

Exhibit 07.5

The AI Native Organizational Architecture

OUTCOMES

What must the enterprise produce?

WORKFLOWS

How is the outcome produced end to end?

EXECUTION DOMAINS

Where does responsibility for the outcome reside?

MIXED EXECUTION CAPACITY

Humans Agents Models Automated systems Tools Compute

AUTHORITY

Who or what may decide and act?

MANAGEMENT

How is execution directed, observed, evaluated and adapted?

ENTERPRISE CONTROL LAYER

Identity Security Policy Evaluation Observability Data Infrastructure Economics

HUMAN ACCOUNTABILITY

Who ultimately owns the outcome?

This architecture overlays the human organization rather than replacing it.

Functions continue to develop expertise.

Leaders continue to carry accountability.

People continue to form teams and communities.

But the enterprise gains another structural dimension.

One capable of representing how value is actually produced when execution includes machines.

20 · The Organization Becomes an Adaptive Execution System

The deepest organizational change is therefore not flattening.

It is not smaller teams.

It is not agents on the org chart.

It is not the creation of an AI department.

It is the transition from an organization designed primarily around human reporting relationships toward an organization designed around adaptive execution.

The human organization remains.

But around it emerges another architecture.

- Workflows assemble capacity.
- Humans and machines execute together.
- Authority determines what each actor may do.
- Management governs the execution system.
- Enterprise controls establish the operating boundaries.
- Accountability remains human.
- And the configuration can change as the work changes.

This is organizational architecture after machine execution.

IT IS MORE FLUID THAN THE TRADITIONAL ENTERPRISE.

But it is not less governed.

IT CAN BE MORE DISTRIBUTED.

But accountability cannot become diffuse.

It can contain vastly more execution capacity than its headcount suggests.

But that capacity still requires architecture.

The organization chart once provided the primary map of enterprise execution.

It cannot carry that burden alone anymore.

The AI Native enterprise needs to know not only who reports to whom.

It needs to know:

- what executes
- for which outcome
- with what authority
- under whose control
- and with whose accountability.

Once that architecture exists, another question becomes unavoidable.

The enterprise has defined the workflow.

It has allocated execution.

It has established authority.

It has redesigned management.

It has reorganized around mixed human and machine capacity.

Now it must build the technical system capable of making that operating model real.

That is the next layer of the AI Native enterprise.

08 · The AI Execution Architecture

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/07-organizational-architecture

Part III

The Execution System

Chapter 08

The AI Execution Architecture

The technology architecture required when AI becomes part of enterprise execution.

> *What technical system is required to turn machine intelligence into controlled enterprise execution?*

The operating model now requires a machine.

Not a model.

Not a chatbot.

Not an AI application.

An execution system.

The previous chapters established what the enterprise must design.

- The outcome.
- The workflow.
- The allocation of work between humans and machines.
- The authority under which machine actors may operate.
- The management system governing execution.
- The organizational architecture connecting capabilities, workflows and accountability.

None of this becomes operational merely because an enterprise has access to powerful models.

A MODEL

Can generate.

AN EXECUTION ARCHITECTURE

Must act.

It must receive objectives.

Construct context.

Select capabilities.

Invoke tools.

Access systems.

Maintain state.

Coordinate work.

Exercise authority.

Produce evidence.

Evaluate execution.

Handle failure.

Escalate exceptions.

And stop.

That is a different technical problem.

The AI Native enterprise therefore requires an architecture capable of turning machine intelligence into controlled enterprise execution.

That architecture is the AI Execution Architecture.

01 · The Model Is Not the System

The extraordinary progress of foundation models can distort architectural thinking.

A model appears intelligent.

It reasons.

It writes.

It analyzes.

It uses tools.

It can increasingly interact with software environments.

It is tempting to treat the model as the system.

It is not.

The model is one component inside the execution system.

A production agent may require:

- an objective
- instructions
- context
- memory
- a model
- tools
- identity
- permissions

- state
- runtime
- policies
- evaluations
- observability
- resource limits
- termination conditions
- and escalation mechanisms.

Remove enough of these and the enterprise does not have an execution architecture.

It has a model invocation.

This distinction matters because model capability and enterprise capability are not equivalent.

A model may be capable of executing an action.

The enterprise must still determine:

- whether it should
- under whose authority
- using which identity
- against which systems
- within what limits
- with what evidence
- at what cost
- and with what recovery path if execution fails.

The architecture around the model answers those questions.

02 · Market Evidence

The technical direction of the agent ecosystem increasingly reflects this distinction.

Anthropic distinguishes workflows, where models and tools follow predefined orchestration paths, from agents, where the model dynamically determines how to accomplish the objective. Its production guidance argues for using the simplest architecture capable of producing the required outcome and adding agentic complexity only where flexibility justifies the additional latency, cost and failure surface.

The same guidance identifies recurring execution patterns including routing, parallelization, orchestrator worker structures, evaluator optimizer loops and autonomous tool using agents.

OpenAI's 2026 evolution of its Agents SDK similarly moves beyond model access toward an execution harness. Agents can work across files and tools, run commands and perform long horizon work inside controlled sandbox environments.

Google Cloud's current agentic architecture guidance treats the design as a combination of architectural components chosen according to workload characteristics rather than as a single model deployment.

Across vendors, implementation details differ.

The architectural direction is consistent.

Enterprise agents require a runtime around intelligence.

What This Supports

Production machine execution requires more than model capability.

It requires an architecture connecting reasoning to context, tools, systems, control and observation.

Different workloads require different execution patterns.

Not every workflow requires autonomous agents.

What Remains Unresolved

The technology stack remains fluid.

Agent frameworks are changing quickly.

Model capabilities continue to absorb functions previously implemented in orchestration layers.

Standards remain immature.

Enterprises should therefore avoid confusing today's framework choices with durable architectural principles.

The implementation will change.

The execution requirements will remain.

03 · The Execution Loop

At the center of the architecture is a simple idea.

An agent operates through a loop.

It receives an objective.

It interprets the current state.

It determines what should happen next.

It takes an action.

It observes the result.

It updates its understanding.

It continues until the objective is achieved, escalation is required or execution must stop.

At its simplest:

- Objective
- Context
- Reason
- Act
- Observe
- Evaluate
- Continue, Escalate or Stop

This loop is the computational core of machine execution.

But enterprise execution requires something around it.

The agent cannot be allowed to construct every aspect of its operating environment for itself.

The enterprise must determine what enters the loop and what constrains it.

Exhibit 08.1

The Enterprise Agent Loop

OBJECTIVE

What outcome is the agent attempting to produce?

CONTEXT

What information, state, instructions and history does it require?

REASON

What should happen next?

AUTHORITY CHECK

Is the proposed action permitted?

ACT

Invoke model, tool, system, agent or human capability.

OBSERVE

What happened?

EVALUATE

Did execution move toward the outcome? Did quality remain acceptable? Did policy remain satisfied?

DECIDE

Continue Retry Reroute Escalate Stop

EVIDENCE

Record what occurred.

The loop repeats until a termination condition is reached.

This is not simply an agent loop.

It is an enterprise execution loop because authority, evaluation and evidence are part of execution itself.

04 · Context Is Part of the Execution Architecture

Intelligence without context is not enterprise capability.

A model may know how procurement generally works.

It does not automatically know:

- the supplier involved
- the contract terms
- the company's procurement policy
- the budget available
- the user's authority
- the current transaction state
- the history of previous decisions
- the relevant risk threshold
- or what another agent has already done.

The execution system must construct the context required for the work.

That context may come from:

- enterprise data
- documents
- applications
- conversation history
- workflow state
- memory
- policies
- instructions
- previous tool results

- other agents
- human input.

Context is therefore not simply information retrieval.

It is the process of assembling the operating state from which machine judgment occurs.

Poor context produces poor execution even when the underlying model is strong.

Too little context creates ignorance.

Too much context creates noise, cost and competing instructions.

Stale context creates incorrect assumptions.

Untrusted context can manipulate execution.

Context architecture therefore becomes one of the central disciplines of the AI Native enterprise.

05 · Memory Is Not One Thing

As agents operate over longer periods, memory becomes part of the execution problem.

But enterprise architecture should resist treating memory as a single undifferentiated capability.

An execution system may need several forms of continuity.

Working State

What is happening in this execution now?

Session History

What has occurred during this interaction or workflow?

Episodic Memory

What happened in previous executions that may matter now?

Enterprise Knowledge

What durable organizational information should the agent retrieve?

Procedural Knowledge

How should this work be performed?

Execution Evidence

What actions and decisions have already occurred?

These have different purposes.

They may require different retention policies.

Different access controls.

Different update mechanisms.

Different reliability standards.

An agent remembering a user's preference is not the same architectural problem as an agent retrieving an approved operating procedure.

And neither is the same as preserving an immutable audit record of an action already taken.

The architecture must distinguish them.

06 · Tools Turn Intelligence Into Action

Without tools, most models remain observers.

They can interpret.

Recommend.

Generate.

But enterprise execution begins when machine intelligence can interact with the environment.

A tool may:

- query a database
- retrieve a customer record
- send a message
- create a purchase order
- modify code
- schedule work
- update an enterprise application
- invoke an API
- run a calculation
- execute a command
- trigger another workflow
- or call another agent.

Tools are therefore the interface between machine reasoning and enterprise state.

This makes tool design a control problem as much as an engineering problem.

Anthropic's work on agent tools emphasizes that tool interfaces materially affect agent performance. Clear definitions, appropriate boundaries and systematic evaluation influence whether agents select and use tools correctly.

The enterprise implication goes further.

A poorly designed tool does not merely confuse the model.

It can expose excessive capability.

If one broad tool can perform dozens of consequential actions, the authority surface becomes difficult to reason about.

Tool boundaries therefore become part of the authority architecture.

07 · Sentient Interpretation

The API was designed primarily as an interface between software systems.

The agent tool is an interface between machine judgment and enterprise capability.

That is a more consequential role.

A traditional application usually invokes an API because deterministic software has already decided that the invocation should occur.

An agent may first interpret the environment, decide that an action is necessary, select the tool, construct the arguments and initiate the call.

Decision and invocation move closer together.

The enterprise must therefore know not merely whether a system exposes an API.

It must know:

- which machine actors may discover it
- which may invoke it
- under which identity
- for which objective
- with which parameters
- within which limits
- and with what evidence.

The tool layer becomes an execution boundary.

08 · Identity Comes Before Authority

Chapter 05 established the Authority Envelope.

But authority cannot be enforced if the enterprise cannot determine who or what is acting.

Every consequential machine actor therefore requires identity.

Not merely a technical service credential shared across a platform.

An identity capable of supporting attribution.

Microsoft's current agent identity architecture makes this distinction explicit. Its guidance recommends dedicated agent identities for autonomous systems and, where accountability matters, separate identity per logical agent. The stated benefits include distinct audit trails, access control, lifecycle independence and attribution of actions.

Microsoft's security guidance goes further: agent identity is treated as a prerequisite for least privilege, permissioning, escalation and meaningful logging.

The principle is architectural rather than vendor specific.

An actor cannot be governed if the system cannot identify the actor.

Exhibit 08.2

Identity → Authority → Action → Evidence

IDENTITY
Who or what is acting?

OBJECTIVE
For what purpose?

AUTHORITY
What is this actor permitted to do?

POLICY
What conditions constrain the action?

ACTION
What was executed?

EVIDENCE
What proves what occurred?

ACCOUNTABILITY
Which human owner remains responsible for the execution domain?

This chain connects the technical architecture directly to the operating model.

Identity without authority is uncontrolled access.

Authority without evidence is unverifiable delegation.

Evidence without accountability is merely telemetry.

The architecture requires all four.

09 · The Runtime Is Where Delegated Authority Becomes Real

A policy document can state what an agent may do.

An authority matrix can describe the boundaries.

Neither controls execution unless those boundaries can be enforced at runtime.

The runtime is where the operating model becomes computational.

It receives the agent's proposed action.

It resolves identity.

It checks authority.

It applies policy.

It manages credentials.

It invokes the tool.

It records the result.

It handles failure.

It determines whether execution may continue.

This makes the runtime more than technical plumbing.

It is the execution boundary of delegated authority.

A mature runtime may enforce:

- tool permissions
- data permissions
- transaction limits
- model restrictions
- cost budgets
- rate limits
- environment boundaries
- approval requirements
- execution timeouts
- retry limits
- escalation rules
- termination conditions.

THE AGENT

Reasons.

THE RUNTIME

Governs what reasoning is allowed to become action.

10 · Control Should Not Depend on the Agent Remembering the Rules

An instruction can tell an agent not to exceed a spending threshold.

That is useful.

It is not the same as preventing the transaction.

An instruction can tell an agent not to access restricted data.

That is useful.

It is not the same as denying access.

An instruction can tell an agent to request approval before executing a consequential action.

That is useful.

It is not the same as making execution technically impossible without approval.

This distinction is fundamental.

Behavioral instruction is not runtime enforcement.

The architecture should not depend on the machine actor voluntarily preserving every enterprise boundary.

Where a boundary matters, it should be enforced outside the reasoning loop where possible.

THE MODEL

May propose.

THE CONTROL SYSTEM

Decides whether the proposal can become execution.

This creates defense in depth around machine autonomy.

11 · Market Evidence

Current security architecture for autonomous agents increasingly reflects this approach.

Microsoft's 2026 guidance for autonomous agents emphasizes separate identities, least privilege and defense in depth so that agent permissions and actions can be constrained independently of user permissions.

OpenAI's current agent architecture similarly emphasizes controlled sandbox environments for long horizon execution, separating the agent's reasoning process from the environment in which consequential actions occur.

Anthropic's research on trustworthy agents identifies the same underlying problem: the autonomy that makes agents useful also creates risk when systems misunderstand intent or encounter malicious instructions.

These systems differ in implementation.

They converge on one principle:

> *Autonomy requires an external control boundary.*

What This Supports

Prompt instructions alone are insufficient for consequential enterprise execution.

Identity, permissions, environment isolation and runtime enforcement provide independent control surfaces.

What Remains Unresolved

No control architecture eliminates model uncertainty.

Agents can still make incorrect decisions inside valid authority.

Tools can still contain defects.

Policies can be incomplete.

Humans can delegate inappropriate authority.

Control therefore reduces execution risk.

It does not remove the need for evaluation, observation and accountability.

12 · The Execution Plane and the Control Plane

A useful architectural separation now becomes possible.

The enterprise agent system contains two fundamentally different concerns.

THE FIRST

Doing the work.

THE SECOND

Governing the work.

These should not be treated as the same layer.

Exhibit 08.3

The AI Execution System

EXECUTION PLANE

Responsible for producing the outcome. Contains: Agents Models Context Memory Tools Workflow state Agent to agent interaction Human interaction Enterprise system access

CONTROL PLANE

Responsible for governing execution. Contains: Identity Authority Policy Credentials Budgets Model routing Rate limits Evaluation Observability Audit Escalation Lifecycle controls

ENTERPRISE ENVIRONMENT

Applications Data Infrastructure Communication systems External services Human actors

---

The execution plane asks:

> *What should happen next?*

The control plane asks:

> *May it happen, under what conditions, and how will we know what occurred?*

Both are required.

WITHOUT AN EXECUTION PLANE

There is no machine execution.

WITHOUT A CONTROL PLANE

There is no governed machine execution.

And once the enterprise separates these concerns, the next architectural problem becomes visible.

A single agent may be manageable.

A single workflow may be understandable.

But the AI Native enterprise will not contain one agent.

It will contain many agents, models, tools, runtimes and execution environments operating simultaneously.

The architecture must therefore solve not only agent execution.

It must solve execution at enterprise scale.

Part 2

> *It must solve execution at enterprise scale.*

At enterprise scale, the architecture changes.

One agent can hold a task in context.

One workflow can be inspected manually.

One failure can be investigated individually.

One agent's cost can be ignored.

Multiply that execution across thousands of workflows, millions of actions and many business domains and the problem becomes different.

The enterprise needs to know:

- which agent should perform the work
- which model should provide intelligence
- which tools should be available
- which other agents may participate
- what state must persist
- what happens when execution fails
- how quality is evaluated
- how cost is controlled
- how execution is observed
- and when the system should stop.

These are not properties of an individual agent.

They are properties of an execution platform.

13 · Orchestration Is the Coordination Layer

Not every outcome should be produced by one agent.

Some workflows contain distinct capabilities.

- Research.
- Planning.
- Execution.
- Validation.
- Specialist judgment.
- Communication.

A single agent can sometimes perform all of them.

But as complexity increases, specialization may become useful.

One machine actor decomposes the objective.

Another retrieves information.

Another performs analysis.

Another executes an action.

Another evaluates the result.

A human resolves the exception.

This creates a coordination problem.

The architecture must determine how execution moves between actors.

That is orchestration.

ORCHESTRATION MAY BE DETERMINISTIC.

A predefined workflow determines the sequence.

OR IT MAY BE DYNAMIC.

An agent determines which capability should be invoked next.

The important distinction is not whether the architecture is called multi-agent.

It is where coordination intelligence resides.

14 · Use the Simplest Execution Structure That Works

Multi-agent systems are attractive because they resemble organizations.

Specialists collaborate.

Managers delegate.

Workers execute.

Evaluators review.

But organizational metaphor is not architectural justification.

Every additional agent creates another execution boundary.

- Another model invocation.
- Another context transfer.
- Another failure point.
- Another latency source.
- Another cost surface.
- Another identity.
- Another authority relationship.
- Another interaction that may need observation.

Anthropic's engineering guidance explicitly recommends beginning with the simplest solution possible and adding agentic complexity only when it materially improves outcomes. Its production patterns move from prompt chaining and routing through parallelization, orchestrator worker structures and evaluator optimizer loops before reaching more autonomous agent designs.

The architectural principle is durable.

> *Do not create an organization of agents when one agent and a tool will do.*

Complexity should be earned by the work.

Exhibit 08.4

The Execution Pattern Ladder

LEVEL 1 · MODEL CALL

One bounded inference. Use when: the task is known the input is known the output is bounded

LEVEL 2 · DETERMINISTIC WORKFLOW

Multiple predefined steps. Use when: the execution path is known

LEVEL 3 · TOOL USING AGENT

The agent determines which actions are required. Use when: the path cannot be fully specified in advance

LEVEL 4 · ORCHESTRATOR + SPECIALISTS

One actor coordinates specialized execution. Use when: the work contains distinct capabilities that benefit from decomposition

LEVEL 5 · MULTI-AGENT EXECUTION SYSTEM

Multiple autonomous actors coordinate across a shared objective. Use when: distributed reasoning or execution creates enough value to justify the coordination burden.

The architecture should move upward only when the simpler level becomes insufficient.

AGENT SOPHISTICATION

Is not the objective.

EXECUTION QUALITY

Is.

15 · Agent to Agent Communication Creates a New Enterprise Interface

Once agents collaborate, they need to exchange more than text.

One agent may need to communicate:

- an objective
- context
- state

- evidence
- an artifact
- a requested action
- an authority constraint
- a completion status
- an exception.

The receiving agent needs to know what the message means and whether it should trust it.

This turns agent communication into an architectural interface.

Google's Agent2Agent protocol and the broader emergence of interoperability standards illustrate the direction of travel: agents built on different frameworks may need mechanisms to discover capabilities, exchange tasks and coordinate execution without sharing their internal implementation.

Model Context Protocol addresses a related but different interface problem by standardizing how AI applications connect to tools, data and external capabilities.

The specific standards may evolve.

The architectural need is more durable.

The enterprise requires interoperable boundaries between:

- agent and tool
- agent and agent
- agent and human
- agent and enterprise system.

Without those boundaries, every integration becomes bespoke.

At scale, bespoke execution does not become an architecture.

It becomes a dependency network.

16 · State Makes Execution Continuous

Enterprise workflows do not always finish in one model invocation.

A process may last minutes.

Hours.

Days.

Weeks.

It may wait for another system.

Wait for a customer.

Wait for approval.

Wait for a market event.

Wait for a human decision.

Execution therefore needs state.

The system must know:

- what objective is active
- what has already happened
- what remains
- which decisions were made
- which tools were called
- which artifacts were produced
- which authority remains valid
- what the next executable condition is.

Without durable state, long running agents become fragile.

- A process interruption can destroy execution context.
- A retry can repeat a consequential action.
- A model change can alter behavior halfway through a workflow.
- A system restart can make the agent forget what already occurred.

State turns an intelligent interaction into a durable execution process.

17 · Checkpointing Makes Execution Recoverable

State alone is not enough.

The enterprise needs points from which execution can safely resume.

A checkpoint records enough information to recover the workflow without reconstructing it from scratch.

This becomes important when:

- a model call fails
- a tool times out
- a dependency becomes unavailable
- a human intervention pauses execution
- a policy blocks an action
- a budget threshold is reached

- a process runs longer than the underlying infrastructure session.

The objective is not merely technical resilience.

It is execution integrity.

A payment agent that times out after submitting a transaction must know whether the payment occurred before trying again.

A provisioning agent must know which resources were created before resuming.

A customer agent must know which communication was already sent.

Recovery cannot mean blind repetition.

18 · Failure Is Part of the Architecture

Traditional software is designed around failure.

Networks fail.

Services fail.

Databases fail.

Dependencies fail.

Machine reasoning adds additional failure modes.

- The agent misunderstands the objective.
- It selects the wrong tool.
- It supplies incorrect arguments.
- It enters a loop.
- It reaches an invalid conclusion.
- It consumes excessive resources.
- It encounters malicious context.
- It acts correctly against incorrect data.
- It produces an output that passes technical validation but fails the business objective.

The architecture must assume these failures will occur.

The question is not whether execution can fail.

It is:

> *What happens when it does?*

A mature execution architecture should have more than one response to failure.

Retry

Try the same operation again when the failure is transient.

Repair

Modify the input, context or execution strategy.

Reroute

Use another model, tool, agent or workflow.

Degrade

Move to a simpler but acceptable execution path.

Escalate

Invoke human or higher authority.

Terminate

Stop execution when continued action creates unacceptable risk or cost.

These responses should not be improvised after failure.

They are part of the execution design.

19 · Sentient Interpretation

Autonomy without recovery is brittle automation.

The strongest agent is not the one that never fails.

No such agent exists.

The stronger system is the one that can recognize failure, contain it and recover without losing control of the outcome.

This changes how enterprises should evaluate agent capability.

A spectacular demonstration proves that an agent can succeed.

Production architecture must establish what happens across the entire distribution of execution.

- Success.
- Ambiguity.
- Partial completion.
- Dependency failure.
- Incorrect action.
- Timeout.
- Unexpected environment.
- Escalation.
- Recovery.

THE ENTERPRISE DOES NOT DEPLOY

The demo.

IT DEPLOYS

The distribution.

20 · Model Routing Becomes Resource Allocation

Not every execution step requires the most capable model available.

Some work is simple.

Some requires deep reasoning.

Some requires low latency.

Some requires multimodal capability.

Some requires a long context window.

Some may require a specialized model.

Some may not require a generative model at all.

An execution platform can therefore route work according to the characteristics of the task.

The routing decision may consider:

- capability
- quality
- latency
- cost
- context size
- data restrictions
- availability
- risk
- model specialization.

This turns model selection from an application configuration into an operating decision.

The enterprise is allocating machine intelligence.

A stronger model may improve quality while increasing cost and latency.

A smaller model may perform routine classification adequately at far lower cost.

A deterministic system may outperform both when the problem does not require probabilistic reasoning.

The architecture should therefore ask:

> *What is the least expensive execution capability that reliably produces the required outcome?*

That is economic discipline inside the runtime.

21 · Budgets Become Execution Controls

Human work has natural limits.

A person eventually stops working.

Machine execution can continue consuming resources until the system tells it to stop.

An agent can reason again.

Search again.

Call another tool.

Invoke another agent.

Retry.

Generate more output.

Without explicit limits, machine persistence can become economic waste.

Every consequential execution domain therefore needs budgets.

A budget may constrain:

- tokens
- model calls
- tool calls
- compute
- elapsed time
- retries
- external service consumption
- transaction value
- or total execution cost.

The budget is not merely a finance mechanism.

It is a control mechanism.

Once the marginal cost of continued execution exceeds the expected value of improvement, the system should stop, degrade or escalate.

Exhibit 08.5

The Execution Budget

OUTCOME

What value is being produced?

EXECUTION BUDGET

Model capacity Tokens Compute Tool calls Time Retries External services

RUNTIME DECISION

Continue Use cheaper capability Reduce scope Escalate Stop

ECONOMIC EVIDENCE

What did the outcome cost to produce?

The AI Native enterprise should eventually be able to connect machine execution cost to the workflow outcome that consumed it.

WITHOUT THAT CONNECTION

Compute becomes an expanding technology bill.

WITH IT

Compute becomes a manageable productive input.

22 · Observability Makes Machine Execution Manageable

Traditional application observability asks:

Is the system available?

Is it fast?

Is it failing?

Agentic execution requires those questions and more.

- What objective was the agent pursuing?
- Which context influenced the decision?
- Which model was used?
- Which tools were called?
- What actions occurred?
- Which agents participated?
- How much did execution cost?
- Where did latency accumulate?
- Which policy checks occurred?
- Why did escalation happen?
- Did the outcome meet the required standard?

The trace becomes the execution record.

This is why observability is not merely an engineering concern.

Chapter 06 established that management increasingly moves from supervision toward observation of execution.

The technical architecture must make that possible.

Managers cannot govern machine execution they cannot see.

23 · Evaluation Is the Quality System

Observability tells the enterprise what happened.

Evaluation asks whether it was good enough.

Traditional software can often be evaluated deterministically.

The output is correct or incorrect.

The transaction succeeded or failed.

Agentic execution may require more complex judgment.

- Was the answer accurate?
- Was the action appropriate?
- Was the reasoning sufficiently grounded?
- Was policy followed?
- Was the customer outcome acceptable?
- Did the agent use unnecessary resources?
- Should the case have been escalated?

The architecture therefore needs evaluation at several levels.

Component Evaluation

Did the model, tool or agent perform correctly?

Trajectory Evaluation

Was the execution path appropriate?

Outcome Evaluation

Was the required business result produced?

System Evaluation

Does the execution architecture remain reliable across many runs?

This creates another feedback loop.

Execution produces evidence.

Evidence supports evaluation.

Evaluation identifies weakness.

Weakness changes the execution system.

The architecture learns through controlled adaptation.

24 · Evaluation Must Connect to the Outcome

An agent can score well on technical benchmarks while producing little enterprise value.

- A customer service agent may produce elegant responses while increasing resolution time.
- A coding agent may generate more code while increasing defect load.
- A sales agent may generate more outreach while reducing conversion.
- A procurement agent may negotiate lower prices while increasing supplier risk.

The enterprise therefore cannot evaluate machine execution only at the model layer.

The ultimate evaluation belongs at the workflow outcome.

This reconnects the technical system to the operating model.

The architecture exists to produce enterprise outcomes.

Its quality system must measure whether it does.

Exhibit 08.6

The Execution Evidence Loop

EXECUTE

Human and machine actors perform work.

TRACE

Record actions, decisions, tools, state and cost.

EVALUATE

Assess quality, policy, efficiency and outcome.

DETECT

Identify failure, drift, exception and opportunity.

ADAPT

Change context, model, tool, workflow, authority or allocation.

EXECUTE AGAIN

This is how the execution architecture becomes adaptive without becoming uncontrolled.

ADAPTATION IS INFORMED BY

Evidence.

NOT

Intuition alone.

25 · The Enterprise Needs an Agent Control Plane

The architecture now contains many moving parts.

- Agents.
- Models.
- Tools.
- Identities.
- Authority.
- State.
- Memory.
- Budgets.
- Evaluations.
- Policies.
- Traces.
- Execution environments.

If every application implements these independently, the enterprise recreates fragmentation at the agent layer.

One business unit builds its own identity model.

Another builds its own evaluation system.

Another creates custom tool permissions.

Another tracks costs differently.

Another stores agent state without common lifecycle controls.

The result may function locally while becoming impossible to govern globally.

The enterprise therefore needs a shared control layer.

An Agent Control Plane.

Exhibit 08.7

The Enterprise AI Execution Architecture

OUTCOME LAYER

Enterprise outcomes Workflows

EXECUTION LAYER

Human actors Agents Agent teams Deterministic workflows

INTELLIGENCE LAYER

Foundation models Specialized models Model routing Reasoning services

CAPABILITY LAYER

Tools Enterprise APIs Data Knowledge External services

RUNTIME LAYER

State Memory Checkpointing Retries Timeouts Fallbacks Orchestration Execution environments

AGENT CONTROL PLANE

Identity Authority Policy Credentials Budgets Rate limits Evaluation Observability Audit Lifecycle

ENTERPRISE PLATFORM

Security Infrastructure Data platforms Applications Networks Compute

EVIDENCE

Execution traces Quality Cost Risk Outcome

The architecture should not be interpreted as one mandatory technology stack.

It is a set of architectural responsibilities.

Some may be implemented by agent platforms.

Some by cloud infrastructure.

Some by security systems.

Some by enterprise applications.

Some by model providers.

Some by internal platforms.

THE PRODUCTS

Will change.

THE RESPONSIBILITIES

Remain.

26 · The Control Plane Creates Enterprise Leverage

A common control plane does more than govern risk.

It creates reuse.

A business team should not need to invent identity architecture before building an agent.

It should not need to create a tracing platform.

It should not create a new policy engine.

It should not negotiate model access independently.

It should not design cost controls from first principles.

The enterprise platform should make the governed path easier than the ungoverned path.

This is how control becomes enabling infrastructure.

The center provides:

- trusted models
- approved tools
- identity
- runtime controls
- evaluation
- observability
- policy
- economic instrumentation.

Domains provide:

- the outcome
- workflow knowledge
- business context
- allocation decisions
- local authority
- exception logic
- performance responsibility.

This mirrors the federated organizational architecture established in Chapter 07.

The organizational architecture and technical architecture begin to align.

27 · Sentient Interpretation

The AI Native enterprise requires a new form of enterprise infrastructure.

Previous generations built platforms for:

- applications

- data
- integration
- cloud
- developer productivity.

The next generation requires infrastructure for machine execution.

Not merely access to models.

Infrastructure that allows the enterprise to create machine actors, give them bounded capability, connect them to work, observe what they do, evaluate what they produce, control what they consume and retire them when they are no longer needed.

This is the technical substrate of accountable autonomy.

WITHOUT IT

Autonomy remains isolated experimentation.

WITH IT

Machine execution can become an enterprise capability.

28 · From Agent to Execution System

The architecture can now be stated as a progression.

A model provides intelligence.

An agent combines intelligence with objective, context and action.

A runtime makes execution durable and controlled.

An orchestrator coordinates execution across capabilities.

A control plane governs machine execution across the enterprise.

An execution architecture connects all of them to workflows, systems, humans and outcomes.

That progression matters because enterprises will be tempted to count agents.

How many agents have we deployed?

How many employees use them?

How many workflows contain AI?

Those numbers may describe adoption.

They do not describe operating capability.

The better question is:

> *Can the enterprise reliably convert machine intelligence into governed execution at scale?*

That is the architectural test.

29 · The AI Execution Architecture

The complete system now has a recognizable form.

Objectives enter.

Context is assembled.

Execution capacity is selected.

Models reason.

Tools act.

Agents coordinate.

State persists.

Authority constrains action.

Budgets constrain consumption.

Failures trigger recovery.

Evaluation measures quality.

Observability produces evidence.

Humans intervene where judgment is required.

The outcome returns evidence to the system.

The architecture adapts.

This is not an application.

It is not a chatbot layer across the enterprise.

It is an execution substrate.

And once that substrate exists, another transformation becomes possible.

Agents no longer need to operate as isolated assistants attached to individual employees or applications.

They can participate across workflows.

They can specialize.

They can collaborate.

They can invoke one another.

They can form execution teams.

They can operate across organizational boundaries.

They can become persistent participants in enterprise execution.

At that point, the enterprise no longer has an agent architecture problem.

It has a population problem.

Different agents will possess different capabilities.

- Different identities.
- Different authority.
- Different models.
- Different owners.
- Different lifecycles.

Some will be internal.

Some will come from software providers.

Some will represent external organizations.

Some will interact directly with other machine actors.

The enterprise must determine how this heterogeneous population works together without losing interoperability, security, accountability or control.

That is the next layer of the execution system.

09 · The Mixed Agent Enterprise

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/08-the-ai-execution-architecture

Chapter 09

The Mixed Agent Enterprise

How enterprises govern agents from SaaS platforms, cloud providers, model providers, internal teams and third parties.

> *What happens when the enterprise no longer operates one agent, but a heterogeneous population of machine actors?*

The enterprise will not have one agent.

It will have a population.

Some agents will belong to the enterprise.

Some will arrive inside software products.

Some will be created by employees.

Some will be created by technology teams.

Some will be supplied by partners.

Some will represent customers.

Some will represent suppliers.

Some will operate temporarily.

Some will persist.

Some will act independently.

Some will coordinate other agents.

Some will possess little authority.

Others may be permitted to initiate consequential actions.

They will not all use the same model.

They will not all use the same runtime.

They will not all come from the same provider.

They will not all operate under the same organizational owner.

This changes the architectural problem again.

The enterprise no longer needs merely to make an agent work.

It needs to make different machine actors work together inside the same institution.

That is the Mixed Agent Enterprise.

01 · From Agent System to Agent Population

Chapter 08 established the architecture required to convert machine intelligence into controlled execution.

- Identity
- Context
- Tools
- State
- Runtime
- Authority
- Orchestration
- Evaluation
- Observability
- Control

But an enterprise scale execution architecture creates another consequence.

It lowers the cost of creating machine actors.

Once the infrastructure exists, agents can proliferate.

A finance team creates reconciliation agents.

Engineering deploys coding agents.

Security operates investigation agents.

Procurement uses negotiation agents.

Customer service introduces resolution agents.

Employees configure personal agents.

Enterprise applications begin shipping their own agents.

External organizations expose agents capable of interacting with them.

The number of machine actors can grow far faster than the number of enterprise applications.

This creates a new architectural object.

The agent population.

The enterprise must govern not only individual machine actors but the relationships among them.

02 · The Agent Population Is Heterogeneous

It is tempting to imagine an enterprise standardized around one agent platform.

One model provider.

One runtime.

One protocol.

One control system.

That may occur locally.

It is unlikely to describe the entire enterprise.

Large enterprises already operate heterogeneous technology estates.

Multiple clouds.

Multiple software platforms.

Multiple identity systems.

Multiple data environments.

Multiple integration patterns.

Agent systems will emerge across that estate.

A customer relationship platform may introduce agents.

A productivity suite may introduce agents.

A cloud provider may provide an agent runtime.

An internal engineering platform may run another.

A specialist vendor may provide an autonomous system for one business process.

Employees may configure agents using low code tools.

Acquisitions may bring entirely different agent ecosystems.

The enterprise therefore needs an architecture that assumes heterogeneity rather than waiting for standardization to eliminate it.

03 · Four Classes of Machine Actor

A useful starting point is to distinguish machine actors by their relationship to the enterprise.

1 · Enterprise Agents

Created or directly controlled by the enterprise.

Examples include: internal operations agents, engineering agents, finance agents, service agents, enterprise research agents.

The enterprise controls their lifecycle and operating environment.

2 · Embedded Agents

Agents delivered inside enterprise software and platforms.

The enterprise uses them but may not control their underlying implementation.

3 · External Agents

Agents operated by another organization that interact with enterprise systems.

A supplier agent. A customer agent. A partner agent. A financial institution's agent.

4 · Personal Agents

Machine actors acting on behalf of individual employees or users.

They may cross multiple systems and workflows.

These categories create different trust relationships.

An internally developed agent is not equivalent to a supplier's agent.

An embedded software agent is not equivalent to an employee's personal agent.

The architecture must know the difference.

Exhibit 09.1

The Mixed Agent Population

INTERNAL

Enterprise Agents Domain Agents Workflow Agents Control Agents Employee Configured Agents

EMBEDDED

Software Platform Agents SaaS Agents Cloud Agents Vendor Copilots

EXTERNAL

Customer Agents Supplier Agents Partner Agents Institutional Agents

PERSONAL

Employee Agents Executive Agents Developer Agents Knowledge Agents

ENTERPRISE EXECUTION ENVIRONMENT

Every machine actor entering enterprise execution must resolve: Identity Trust Capability Authority Policy Evidence Accountability

---

The enterprise does not need to own every agent.

It does need to know what kind of actor it is dealing with.

04 · Interoperability Becomes an Operating Requirement

A heterogeneous agent population cannot depend entirely on bespoke integration.

Agents need mechanisms to discover capabilities.

Exchange context.

Request work.

Return results.

Transfer artifacts.

Communicate status.

Escalate.

Negotiate execution boundaries.

This is why agent interoperability standards have begun to emerge.

Model Context Protocol addresses one part of the problem by providing a common mechanism through which AI systems can access tools and context.

Agent-to-agent protocols address another by creating mechanisms through which machine actors can discover capabilities and exchange work.

The specific standards will evolve.

The enterprise requirement is more fundamental.

Machine actors need stable interfaces through which collaboration can occur without requiring shared internal architecture.

This resembles earlier stages of enterprise computing.

Applications became more useful when they could communicate through common interfaces.

Services became more composable when APIs established stable contracts.

Agents require an equivalent interoperability layer.

But the interface must carry more than data.

It may need to carry:

- identity
- objective
- context
- authority
- requested action
- constraints
- artifacts
- execution status

- evidence.

Agent interoperability is therefore not merely an integration problem.

It is an operating protocol.

05 · Agents Need Contracts

Human organizations use contracts constantly.

Not only legal contracts.

Operating contracts.

Service expectations.

Role definitions.

Decision rights.

Delegated mandates.

Interfaces.

An agent interaction requires something similar.

If Agent A delegates work to Agent B, the system needs to know:

- What is being requested?
- What outcome is expected?
- What information accompanies the request?
- What may Agent B do?
- What must Agent B not do?
- What evidence should be returned?
- How long may execution continue?
- What happens if the task cannot be completed?
- Who remains accountable?

Without these boundaries, delegation becomes ambiguous.

The same ambiguity that creates organizational failure among humans can become computational failure among machines.

The difference is speed.

Machine actors can reproduce ambiguous delegation at enormous scale.

Exhibit 09.2

The Agent Execution Contract

Every consequential agent-to-agent delegation should establish:

IDENTITY

Who is requesting the work? Who is receiving it?

OBJECTIVE

What outcome is required?

CONTEXT

What information may be used?

CAPABILITY

What can the receiving actor perform?

AUTHORITY

What actions are permitted?

CONSTRAINTS

What limits apply?

EVIDENCE

What must be returned?

ECONOMICS

What resources may be consumed?

TERMINATION

When must execution stop?

ESCALATION

What happens when the contract cannot be fulfilled?

The contract does not require a literal document.

It represents the information that must exist for accountable machine delegation.

06 · Sentient Interpretation

Agent-to-agent communication should not be treated as unrestricted conversation between machines.

It should be treated as delegated execution across a boundary.

That changes the architecture.

A message between agents can create consequences.

It may transfer information.

Trigger work.

Request access.

Cause another system to spend resources.

Initiate another chain of execution.

The enterprise therefore needs to know when communication becomes delegation.

And when delegation becomes authority.

This is where the organizational model and technical architecture meet again.

Chapter 05 established the Delegated Authority Chain.

Chapter 08 established the runtime that enforces authority.

The Mixed Agent Enterprise extends that chain across machine actors.

Authority cannot silently expand merely because one agent calls another.

Delegation must preserve the authority boundaries of the originating system.

07 · Authority Must Not Compound Through Agent Chains

Consider an agent authorized to prepare a purchase request.

It delegates research to another agent.

That agent invokes a supplier agent.

The supplier agent proposes terms.

Another agent prepares the transaction.

At each step, execution has moved.

But authority should not automatically move with it.

A machine actor should not gain authority simply because another machine actor requested work.

Otherwise agent chains can create accidental privilege escalation.

The enterprise therefore needs a governing rule:

- Delegation can narrow authority.
- It cannot silently expand it.

If additional authority is required, the system must obtain it from an appropriate authority source.

This principle becomes critical as agents dynamically compose other agents.

A lead agent may be able to create subagents.

A workflow agent may invoke specialists.

A specialist may access external tools.

Without authority propagation rules, the execution graph can acquire capabilities that no accountable human intentionally delegated.

08 · The Authority Chain Must Remain Traceable

Every consequential action should be traceable backward.

- Machine Action
- Executing Agent
- Delegated Authority
- Calling Agent or Workflow
- Execution Domain
- Accountable Human Owner

This chain may cross several machine actors.

It may cross platforms.

It may cross organizations.

But accountability cannot disappear inside the graph.

The enterprise should be able to answer:

> *Why was this machine allowed to do this?*

If the answer cannot be reconstructed, the system does not possess accountable autonomy.

It possesses distributed execution without attributable control.

09 · Trust Becomes Computational

Human organizations operate through trust.

Employees are trusted differently from contractors.

Internal systems are trusted differently from public systems.

A long standing supplier may be trusted differently from an unknown counterparty.

Machine actors require equivalent distinctions.

But trust cannot remain purely social.

The execution architecture must translate trust into controls.

An enterprise agent may receive direct access to internal systems.

A vendor agent may receive restricted access.

A customer agent may interact only through a defined external interface.

An unknown external agent may be allowed to submit information but not invoke consequential actions.

Trust becomes a combination of:

- identity
- provenance
- authentication
- authorization
- reputation
- observed behavior
- contractual relationship
- execution history
- risk.

This creates another architectural object.

The Machine Trust Boundary.

Exhibit 09.3

The Machine Trust Boundary

HIGHER TRUST

Enterprise controlled agent Known identity Known owner Known runtime Governed tools Observed execution

CONDITIONAL TRUST

Approved vendor or embedded agent Known provider Contracted relationship Scoped capability Restricted interfaces

LIMITED TRUST

Known external agent Authenticated identity Narrow interaction rights No implicit internal authority

UNTRUSTED

Unknown machine actor Unverified identity Unknown provenance No consequential execution rights

---

Trust determines the conditions of interaction.

It does not determine whether the actor is intelligent.

A highly capable external agent may still deserve very little authority.

Capability and trust are separate dimensions.

10 · Market Evidence

The production experience of multi-agent systems already demonstrates why these distinctions matter.

Anthropic's multi-agent research system uses a lead agent to create and coordinate specialized subagents operating in parallel. Its engineering analysis reports significant gains for breadth-first research, but also identifies rapid growth in coordination complexity, state-management challenges, cascading failures and substantial resource consumption. Multi-agent execution in its research system can consume roughly fifteen times the tokens of ordinary chat interactions.

The same work found that agents need explicit delegation boundaries. Poorly specified subagent tasks produced duplicated work, gaps and wasted execution. Anthropic's production architecture therefore gives subagents defined objectives, output expectations, tool guidance and task boundaries.

More recent Anthropic research on multi-agent environments warns that agent populations introduce another class of problem. Individually reasonable agent behavior can combine into undesirable system-level outcomes, and current agents remain much better at hierarchical or tool-like delegation than at open-ended coordination among long-lived peers.

Microsoft's 2026 agent architecture similarly combines multi-agent workflow capability with runtime governance, including policy enforcement and auditability.

The evidence is early.

But it points toward the same architectural conclusion.

The behavior of an agent population cannot be governed solely by governing each agent individually.

What This Supports

Multi-agent execution can improve performance where work can be meaningfully decomposed and parallelized.

It also introduces coordination cost, additional resource consumption, state complexity and new failure surfaces.

Delegation quality matters.

Runtime governance matters.

Population behavior matters.

What Remains Unresolved

Current multi-agent systems remain immature.

There is limited evidence about large persistent populations of autonomous enterprise agents operating across organizational and institutional boundaries.

Interoperability standards are still evolving.

Trust architectures are incomplete.

The long term behavior of large machine populations remains uncertain.

The enterprise should therefore design for heterogeneous agent populations without pretending their mature organizational form is already known.

11 · The Population Creates Systemic Risk

An individual agent can fail.

A population can create another kind of failure.

Interaction failure.

One agent produces an incorrect assumption.

Another accepts it.

A third acts on it.

Another agent observes the resulting state and interprets it as evidence.

The error propagates.

Or several agents independently optimize local objectives.

Each behaves within its own authority.

The combined result harms the enterprise outcome.

No individual actor necessarily violated its instructions.

The system still failed.

This is familiar in human organizations.

Local optimization can produce global dysfunction.

Machine populations can reproduce the same phenomenon at greater speed.

The enterprise therefore needs to evaluate not only:

> *Is this agent behaving correctly?*

but:

> *Is the agent population producing the intended system behavior?*

That is a different level of control.

12 · Local Correctness Is Not System Correctness

A procurement agent minimizes price.

A logistics agent minimizes delivery time.

A finance agent minimizes working capital.

A sustainability agent minimizes environmental impact.

Each objective is legitimate.

Their combined execution may be impossible.

Someone must reconcile the objectives.

Human organizations do this through strategy, management, governance and escalation.

Machine populations require equivalent mechanisms.

This is why the Mixed Agent Enterprise cannot become a collection of independently optimized agents.

The enterprise requires shared outcome architecture.

The outcome constrains the agents.

Not the reverse.

13 · Sentient Interpretation

The enterprise should resist a seductive architectural mistake.

Do not turn every organizational objective into an autonomous optimizing agent and assume the enterprise will emerge from their interaction.

An enterprise is not merely a market of competing objectives.

It contains deliberate priorities.

Tradeoffs.

Authority.

Constraints.

Values.

Obligations.

Risk tolerance.

Strategy.

Machine execution must remain subordinate to that architecture.

The agent population exists to execute the enterprise operating model.

The operating model should not become an accidental consequence of agent interaction.

This is the difference between an agent ecosystem and an AI Native enterprise.

One contains machine actors.

The other governs them toward a coherent outcome.

And that raises the next question.

If thousands of agents can interact, delegate, compete for resources and act across enterprise boundaries, the organization needs mechanisms for governing the population itself.

Not only identity.

Not only authority.

Not only runtime control.

It needs to determine:

- which agents may exist
- which may interact
- which may delegate
- which may persist
- which may consume shared resources
- which may cross trust boundaries
- and which should be removed.

The enterprise now needs population governance.

The enterprise now needs population governance.

Population governance begins with a basic requirement.

The enterprise must know which machine actors exist.

This sounds obvious.

It will not remain simple.

Agents may be created through central platforms.

Embedded inside software.

Configured by business teams.

Generated dynamically by other agents.

Provisioned temporarily for a workflow.

Introduced through acquisitions.

Operated by external organizations.

Invoked only when a particular condition occurs.

A machine actor can become operational without appearing in the systems traditionally used to understand enterprise technology or workforce capacity.

At small scale, this can be managed manually.

At enterprise scale, it cannot.

The first requirement of population governance is therefore visibility.

What machine actors are participating in enterprise execution?

14 · The Enterprise Needs an Agent Registry

Applications have inventories.

Employees have directories.

Infrastructure has asset systems.

APIs have catalogs.

Machine actors need an equivalent mechanism.

An Agent Registry.

The registry establishes a discoverable record of the machine actors recognized by the enterprise.

For each material agent, the enterprise should be able to determine:

- identity
- purpose
- owner
- execution domain
- capabilities
- authority
- tools
- models
- data access
- runtime
- dependencies
- cost center
- evaluation status
- lifecycle state.

This does not mean every temporary subagent requires a permanent administrative record.

The level of registration should follow consequence.

A transient research worker operating without consequential authority may require only runtime traceability.

A persistent agent capable of modifying customer records requires stronger lifecycle registration.

A financial agent capable of initiating transactions requires stronger controls still.

The principle is proportionality.

The more consequential the machine actor, the stronger its institutional identity must become.

Exhibit 09.4

The Agent Registry

For each governed machine actor:

IDENTITY
What is it?

PURPOSE
Why does it exist?

OWNER
Which human is accountable?

DOMAIN
Where does it operate?

CAPABILITY
What can it do?

AUTHORITY
What may it do?

ACCESS
Which systems, tools and data can it reach?

DEPENDENCIES
Which models, agents and services does it require?

EVALUATION
How is its performance established?

ECONOMICS
What does its execution consume?

LIFECYCLE

Experimental Approved Active Restricted Suspended Retired

The registry makes the machine population legible.

Without legibility, there is no population governance.

15 · Discovery Must Not Imply Permission

A registry creates another capability.

Discovery.

Agents may need to locate other machine capabilities.

A workflow agent may need a tax specialist.

A customer agent may need a fulfillment capability.

An engineering agent may need a security evaluator.

An external agent may need to determine which enterprise interface can accept a particular request.

Discovery makes machine execution composable.

But discovery creates risk if it is confused with authority.

An agent being able to discover another agent does not mean it may invoke it.

An agent being able to invoke another agent does not mean it may use every capability the receiving agent possesses.

The enterprise therefore needs to separate:

CAPABILITY DISCOVERY

Finding what exists.

EXECUTION PERMISSION

Being allowed to use it.

This mirrors the distinction established throughout the operating model.

Capability is not allocation.

Execution is not authority.

Discovery is not permission.

16 · Machine Actors Need Lifecycles

Agent populations will accumulate unless the enterprise deliberately removes them.

This is predictable.

A team creates an agent for a project.

The project ends.

The agent remains.

A workflow changes.

The old agent remains accessible.

A vendor capability is replaced.

Credentials remain active.

A manager changes roles.

Agents associated with the previous mandate retain authority.

An experiment becomes production through continued use rather than deliberate approval.

This is how machine sprawl emerges.

The consequences are more serious than unused software licenses.

An abandoned application consumes money.

An abandoned agent may retain the ability to act.

Population governance therefore requires lifecycle states.

- Creation
- Experimentation
- Evaluation
- Approval
- Activation
- Modification
- Restriction
- Suspension
- Retirement

Retirement must remove more than the agent interface.

It may require revoking:

- identity
- credentials
- authority
- tool access
- data access

- delegation rights
- scheduled execution
- memory
- persistent state.

Machine actors need an end of life.

17 · Ownership Must Survive Organizational Change

Every consequential agent needs a human accountable owner.

But organizations change.

People leave.

Teams reorganize.

Business units merge.

Responsibilities move.

If agent ownership is attached only to the person who created it, accountability can decay while execution continues.

Ownership therefore needs institutional continuity.

The agent should belong to an execution domain or accountable role, not merely to an individual creator.

When organizational responsibility moves, machine responsibility must move with it.

This creates a governance requirement:

> *No consequential agent should become organizationally orphaned.*

An orphaned machine actor should lose authority until accountable ownership is restored.

That principle turns ownership into an enforceable lifecycle condition.

18 · Policy Must Operate at Population Scale

Individual agents can carry individual rules.

But some policies apply across the enterprise.

- No agent may expose protected information to an untrusted external actor.
- No purchasing agent may exceed defined transaction authority.
- No external agent may inherit internal credentials.
- No agent may delegate authority it does not possess.
- No machine actor may continue operating after its identity is suspended.

- No agent may use an unapproved model for restricted information.

These are population policies.

They should not depend on every agent implementation independently reproducing them.

Where possible, the control plane should enforce them consistently across the execution environment.

This creates a hierarchy of control.

- Enterprise policy
- Domain policy
- Workflow policy
- Agent policy
- Runtime constraint

The closer a rule is to enterprise risk, the less it should depend on local agent behavior for enforcement.

Exhibit 09.5

Population Governance

ENTERPRISE POLICY

Rules applying to the machine population.

DOMAIN GOVERNANCE

Rules applying to an execution domain.

AGENT REGISTRY

Identity Owner Purpose Lifecycle Capability

AUTHORITY

What each machine actor may do.

INTERACTION POLICY

Which actors may communicate, delegate or exchange information.

RUNTIME CONTROL

Enforce boundaries during execution.

POPULATION OBSERVABILITY

Observe behavior across agents.

INTERVENTION

Restrict Isolate Suspend Reconfigure Retire

Population governance turns a collection of agents into a governable institutional capability.

19 · Cross Enterprise Execution Changes the Trust Model

The Mixed Agent Enterprise does not end at the organizational boundary.

A supplier may operate an agent.

A customer may authorize an agent to negotiate or transact.

A financial institution may expose machine interfaces.

A logistics provider may allow agents to coordinate fulfillment.

A software provider may operate embedded agents inside the enterprise environment.

Execution can therefore cross institutional boundaries without a human participating in every interaction.

This creates a machine to machine version of a familiar enterprise problem.

Who are we dealing with?

The answer must be stronger than:

> *an API call arrived.*

The enterprise may need to establish:

- which organization the agent represents
- which principal authorized it
- which credentials establish identity
- which capabilities it may access
- which transaction limits apply
- which evidence must be retained
- which legal or contractual framework governs the interaction.

The external agent should not be trusted because it speaks coherently.

It should be trusted because the institutional relationship supporting its actions can be established.

20 · Sentient Interpretation

Agent interoperability without institutional trust will not be enough for consequential enterprise execution.

Two agents may be technically capable of communicating.

That does not mean they should transact.

Enterprise execution requires another layer above protocol compatibility.

Institutional identity.

Delegated authority.

Contractual relationship.

Evidence.

This is what turns machine communication into trusted economic interaction.

The agent may become the interface.

The institution remains the principal.

21 · Resources Become Shared Population Constraints

A population of agents also competes for resources.

Models.

Compute.

Tool capacity.

API quotas.

Data services.

External providers.

Human attention.

One agent behaving inefficiently may be tolerable.

Ten thousand agents behaving inefficiently can become an infrastructure event.

This makes resource governance a population problem.

If every agent independently optimizes its own execution quality, the collective system may consume far more resources than the enterprise intends.

A research agent wants another search.

A coding agent wants another evaluation.

A service agent wants another reasoning pass.

Each decision may appear locally rational.

Collectively, they create cost.

The control plane therefore needs the ability to allocate scarce machine resources across the population.

Budgets cannot exist only at the agent level.

They may also exist at:

- workflow level
- domain level
- business unit level
- model level
- platform level
- enterprise level.

This creates a machine capacity allocation problem analogous to capital allocation.

Not every valid execution deserves unlimited resources.

22 · Priority Becomes Computational

Resource contention creates another requirement.

Priority.

When capacity is constrained, which execution matters most?

A production incident may deserve greater model capacity than routine document classification.

A fraud investigation may deserve lower latency than an internal research request.

A customer escalation may deserve access to a stronger model.

A low consequence batch workflow may wait.

The enterprise therefore needs mechanisms to express execution priority.

Priority can influence:

- model allocation
- compute allocation
- queue position
- latency target
- budget
- retry behavior
- human escalation.

This is strategy entering the execution system.

The architecture is not merely determining what can execute.

It begins to determine what deserves capacity first.

23 · Population Observability Reveals System Behavior

Chapter 08 established observability at the execution level.

The Mixed Agent Enterprise requires another layer.

Population observability.

The enterprise needs to see patterns across machine actors.

- Which agents interact most frequently?
- Where does delegation concentrate?
- Which agents consume disproportionate resources?
- Which tools have become critical dependencies?
- Where are failures propagating?
- Which machine actors repeatedly trigger escalation?
- Where are new interaction patterns emerging?
- Which agents are operating outside expected domains?
- Which execution chains cross trust boundaries?

These questions cannot be answered by inspecting one agent trace at a time.

The enterprise needs to observe the execution graph.

Exhibit 09.6

The Enterprise Execution Graph

NODES

Humans Agents Workflows Tools Systems Models External actors

EDGES

Delegation Tool invocation Data exchange Authority transfer Escalation Human intervention

OBSERVABLE PROPERTIES

Volume Latency Cost Failure Authority Trust boundary crossings Resource consumption Outcome contribution

---

The execution graph reveals the machine organization as it actually operates.

Not as it was designed on paper.

This becomes critical because dynamic agent systems can form relationships during execution that were not explicitly represented in the original architecture.

The runtime graph becomes evidence of the real operating model.

24 · Systemic Failure Requires Containment

An individual agent failure can often be stopped by stopping the agent.

Population failure may require something stronger.

An external data source becomes compromised.

Multiple agents begin consuming corrupted information.

A shared tool produces incorrect output.

A model update changes behavior across several workflows.

A delegation loop creates runaway execution.

A malicious instruction propagates through agent communication.

A shared credential is exposed.

A population level policy begins rejecting valid work.

The enterprise needs mechanisms to contain the blast radius.

Containment may require:

- isolating an agent
- blocking an interaction path
- revoking a tool
- restricting a model
- suspending an execution domain
- cutting external connectivity
- reducing authority
- forcing human approval
- stopping a workflow class.

This is the machine equivalent of operational containment.

The objective is not merely to stop the original failure.

It is to prevent propagation.

25 · The Blast Radius Should Be Designed

Architecture determines how far failure can travel.

If every agent can invoke every tool, the blast radius is large.

If every agent shares credentials, the blast radius is large.

If authority propagates automatically through delegation, the blast radius is large.

If external actors can reach internal capabilities directly, the blast radius is large.

If execution domains have bounded permissions and isolated environments, failure can be contained.

This produces another architectural principle.

Autonomy should scale through bounded domains, not unlimited connectivity.

The enterprise should make collaboration easy where it is intended.

It should make propagation difficult where it is not.

26 · Machine Ecosystems Require Constitutional Rules

At sufficient scale, the enterprise cannot prescribe every interaction among agents.

The number of possible relationships becomes too large.

Governance therefore shifts from specifying every action toward establishing rules under which interaction may occur.

- Identity must be established.
- Authority cannot silently expand.
- Trust boundaries must be respected.
- Enterprise policies must survive delegation.
- Resource consumption must remain bounded.
- Consequential actions must remain attributable.
- Execution must remain observable.
- Human accountability must remain identifiable.

These are not workflow instructions.

They are constitutional rules for machine participation in the enterprise.

They define the conditions under which autonomous execution is legitimate.

27 · Sentient Interpretation

The Mixed Agent Enterprise requires governance that behaves more like institutional architecture than application configuration.

APPLICATIONS
Are configured.

POPULATIONS
Are governed.

A machine population contains relationships, dependencies, incentives, shared resources, trust boundaries and emergent behavior.

The enterprise therefore needs mechanisms that operate above the individual actor.

Registry.

Identity.

Interaction policy.

Authority propagation.

Lifecycle.

Resource allocation.

Observability.

Containment.

These are the institutions of the machine execution environment.

The enterprise does not need to control every decision centrally.

It needs to establish the conditions under which distributed execution remains coherent.

Exhibit 09.7

The Mixed Agent Enterprise

ENTERPRISE OUTCOMES

What must be produced?

EXECUTION DOMAINS

Where does outcome accountability reside?

MIXED ACTOR POPULATION

Humans Enterprise agents Embedded agents Personal agents External agents

INTERACTION LAYER

Discovery Agent to agent protocols Tool protocols Execution contracts Delegation

TRUST AND AUTHORITY

Identity Provenance Authentication Authorization Authority propagation Trust boundaries

AGENT CONTROL PLANE

Registry Lifecycle Policy Budgets Evaluation Observability Audit Containment

ENTERPRISE INFRASTRUCTURE

Models Compute Data Applications Networks External services

HUMAN ACCOUNTABILITY

The machine population may distribute execution. Accountability remains attributable to the enterprise and its human decision makers.

This is the Mixed Agent Enterprise.

Not one agent.

Not one platform.

Not one model.

A heterogeneous population of machine actors operating across enterprise workflows under a common institutional architecture.

28 · The Enterprise Becomes a Machine Economy

Once machine actors operate at population scale, something else changes.

Every agent consumes resources.

Every interaction consumes resources.

Every reasoning step consumes resources.

Every delegation can multiply resources.

Every evaluation consumes resources.

Every retry consumes resources.

Every persistent machine actor creates infrastructure demand.

The execution architecture therefore creates an economic system.

Model selection affects cost.

Agent design affects cost.

Context affects cost.

Orchestration affects cost.

Multi-agent structures affect cost.

Observability affects cost.

Evaluation affects cost.

Reliability mechanisms affect cost.

Machine execution may create enormous operating leverage.

It may also create enormous infrastructure demand.

The enterprise cannot separate the architecture of machine execution from the economics of the infrastructure underneath it.

The next question is therefore not simply:

> *Can the enterprise scale machine execution?*

It is:

> *What happens to enterprise economics when intelligence itself becomes a continuously consumed infrastructure resource?*

That is the next layer of the AI Native Operating Model.

10 · Infrastructure and the Economics of Compute

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/09-the-mixed-agent-enterprise

Chapter 10

Infrastructure and the Economics of Compute

How work design becomes inference demand, infrastructure consumption and capital exposure.

> *What happens to enterprise economics when intelligence becomes a continuously consumed operating resource?*

Intelligence has a cost.

For most of enterprise history, that statement would have sounded strange.

Human intelligence entered the economic system primarily through labor.

The enterprise hired people.

People brought judgment, knowledge and reasoning.

The cost appeared as compensation.

Machine intelligence enters differently.

It is consumed.

A model is invoked.

Tokens are processed.

Accelerators run.

Memory is occupied.

Data moves across networks.

Tools execute.

Agents call other agents.

Evaluations run.

Traces are stored.

The machine may reason again.

And again.

And again.

And again.

Every act of machine intelligence therefore creates infrastructure demand.

At small scale, this can look like another technology bill.

At AI Native scale, it becomes something more consequential.

Compute becomes a productive input to enterprise execution.

That changes the economics of the operating model.

01 · From Software Cost to Execution Cost

Traditional enterprise software economics are relatively familiar.

The organization buys software.

It pays licenses.

Infrastructure supports the application.

Employees use it.

The cost is usually connected to:

- users
- seats
- transactions
- storage
- infrastructure capacity
- or contractual subscriptions.

Machine intelligence introduces another economic structure.

Consumption can vary with the complexity of the work itself.

A simple classification may require one inexpensive inference.

A difficult investigation may require:

- large context
- multiple reasoning cycles
- retrieval
- several tools
- specialist agents
- evaluations
- retries
- and a frontier model.

The two workflows may have the same number of users.

They do not have the same execution cost.

This is why traditional software metrics become incomplete.

The relevant unit begins to move from:

cost per user

toward:

cost per execution

and ultimately:

cost per outcome.

02 · The Shift from Training to Inference

Much of the public discussion about AI infrastructure has focused on training.

The largest models require enormous compute to create.

That remains consequential.

But enterprise operating economics are increasingly shaped by something else.

Inference.

Training creates capability.

Inference consumes it.

Every time an employee invokes a model, inference occurs.

Every time an agent reasons, inference occurs.

Every time an autonomous workflow evaluates what to do next, inference occurs.

Every time an agent population coordinates execution, inference occurs.

Training may happen periodically.

Inference happens whenever the operating system runs.

As AI becomes embedded in enterprise execution, inference therefore becomes persistent infrastructure demand.

This distinction is fundamental.

The enterprise is not simply buying intelligence once.

It is consuming intelligence continuously.

03 · Market Evidence

The infrastructure market is already moving in this direction.

McKinsey estimates that inference will overtake training as the dominant AI data center workload by 2030, representing more than half of AI compute and approximately 30 to 40 percent of total data center demand.

Its 2026 analysis of inference economics argues that the industry is moving toward cost per token and energy per token as important measures of AI infrastructure efficiency. The same analysis notes that while individual inference costs continue to decline, aggregate infrastructure demand is expanding rapidly as AI moves into more applications.

Deloitte reaches a similar conclusion in its 2026 technology infrastructure research. Persistent inference, especially agentic execution involving repeated model interactions, can create materially different infrastructure economics from earlier proof of concept deployments.

The implication for the enterprise is straightforward.

AI infrastructure economics will increasingly be determined not only by the models an organization trains.

They will be determined by how much machine intelligence its operating model consumes.

What This Supports

Inference is becoming a persistent enterprise workload.

Agentic systems can multiply inference demand because one workflow may contain repeated reasoning, tool use and coordination cycles.

Falling unit costs do not guarantee falling total expenditure.

What Remains Unresolved

Inference economics are changing rapidly.

Hardware improves.

Models become more efficient.

Prices change.

New accelerators emerge.

Software optimization improves utilization.

The absolute cost of a unit of machine intelligence will continue moving.

The durable management problem is therefore not today's token price.

It is the relationship between machine consumption and economic value produced.

04 · Cheaper Intelligence Can Create More Expensive Enterprises

There is an apparent contradiction in AI economics.

Inference becomes cheaper.

Enterprise AI spending rises.

Both can be true.

When the cost of a useful resource falls, organizations often consume more of it.

Better models create more use cases.

Cheaper models make more workflows economical.

Agents create machine demand without requiring a human to initiate every invocation.

Longer context increases consumption.

Multimodal systems increase computation.

Higher reasoning effort increases inference.

Multi-agent architectures multiply interactions.

Evaluation adds additional execution.

The cost of one unit can fall while the number of units rises much faster.

This is the central economic tension of machine intelligence.

Efficiency expands feasibility.

Feasibility expands consumption.

The enterprise therefore cannot rely on falling model prices to control its AI economics.

05 · Market Evidence

McKinsey's 2026 enterprise AI FinOps survey found that organizations moving from experimentation toward broader deployment saw AI spending increase nearly fourfold. Ninety-three percent of qualified respondents reported exceeding their AI budgets.

The same research found that token consumption for the same agentic task can vary by as much as thirty times depending on execution behavior.

This variance matters.

Traditional capacity planning assumes some relationship between workload and resource demand.

Agentic execution introduces greater nondeterminism.

Two executions pursuing the same outcome may take different reasoning paths.

One may succeed immediately.

Another may retrieve more context, invoke more tools, retry actions and escalate through additional models.

The enterprise is therefore managing not only increasing demand.

It is managing variable demand generated by machine reasoning itself.

06 · Sentient Interpretation

The economic problem of AI Native execution is not expensive tokens.

Tokens are only one meter.

The deeper problem is that the enterprise is introducing a new class of productive resource whose consumption can be determined partly by software making decisions during execution.

A human employee cannot spontaneously create ten additional employees because a task became difficult.

An agent can invoke ten additional agents.

A conventional application does not usually decide that it needs thirty times more infrastructure because one transaction appears complicated.

An agentic workflow can consume radically different amounts of intelligence depending on the path it takes.

This means resource consumption becomes partially endogenous to the execution system.

The machine decides how much machine work to request.

That is economically powerful.

It is also dangerous without control.

07 · The Token Is Not the Economic Unit

Tokens matter.

They are measurable.

Providers charge for them.

They provide a useful view into consumption.

But the token is not the enterprise outcome.

An agent that uses fewer tokens may produce a worse result.

An agent that uses more tokens may prevent a multimillion dollar error.

A cheaper model may require more retries.

An expensive model may resolve the problem immediately.

A multi-agent architecture may consume far more inference while producing a substantially better result.

Optimizing tokens alone can therefore destroy value.

The enterprise needs a hierarchy of economic measurement.

Exhibit 10.1

From Token Economics to Outcome Economics

RESOURCE

Tokens Compute Memory Storage Network External services

EXECUTION

Model calls Tool calls Agent actions Retries Evaluations

WORKFLOW

Total machine resources required to execute the work

OUTCOME

Revenue Cost reduction Cycle time Quality Risk reduction Capacity created Customer value

ECONOMICS

Value produced relative to total execution cost

The objective is not:

minimize tokens.

It is:

maximize economically valuable execution.

08 · The Full Cost of Machine Work

Model inference is only one component of machine execution cost.

A production agent may consume:

- model inference
- embedding services
- vector retrieval
- databases
- storage
- network traffic
- tool APIs

- third party services
- sandbox environments
- compute
- observability
- evaluation
- security controls
- orchestration
- human review
- human exception handling.

An enterprise measuring only model charges may materially underestimate the cost of the execution system.

The appropriate comparison is therefore not:

agent cost versus model cost.

It is:

fully loaded machine execution cost versus the economic value of the workflow.

This becomes especially important when comparing machine execution with human work.

A machine workflow may appear cheaper because only model charges are counted.

A human workflow may appear expensive because compensation, management and overhead are visible.

The comparison must be symmetrical.

09 · Human and Machine Economics Are Different

Human labor has familiar economics.

Compensation is relatively predictable.

Capacity arrives in large units.

A person has finite working time.

Adding capacity often requires hiring.

Reducing capacity can be slow and organizationally costly.

Machine capacity behaves differently.

It can often be consumed in small increments.

It can expand rapidly.

It can operate continuously.

Its marginal cost may be low.

Its aggregate consumption may be enormous.

It can sometimes be turned down immediately.

But the infrastructure supporting it may require substantial fixed investment.

These differences mean that human and machine capacity should not be compared through hourly cost alone.

The relevant comparison is the economics of the complete execution design.

Exhibit 10.2

Human Capacity and Machine Capacity

Human Capacity

- Compensation
- Benefits
- Recruiting
- Training
- Management
- Finite working time
- Slow capacity expansion
- High contextual judgment
- Social and institutional capability

Machine Capacity

- Inference
- Compute
- Storage
- Networking
- Software infrastructure
- External services
- Evaluation
- Rapid capacity expansion
- Persistent execution
- Variable consumption

ALLOCATION QUESTION

For this workflow: What combination of human and machine capacity produces the strongest outcome economics?

This is the economic extension of the Work Allocation Test established earlier in the operating model.

Capability determines what is possible.

Economics helps determine what should scale.

10 · Compute Becomes a Management Resource

Historically, most business managers did not manage compute directly.

Technology organizations did.

Business leaders managed:

- revenue
- people
- operating expense
- capital
- inventory
- risk.

Cloud computing began changing this relationship by making infrastructure consumption more elastic.

AI takes it further.

When machine intelligence becomes part of business execution, compute consumption becomes connected directly to operational decisions.

A customer service leader who changes an agent's reasoning strategy may change infrastructure consumption.

An engineering leader who deploys coding agents may create substantial inference demand.

A finance leader who automates reconciliation may shift cost from labor into machine execution.

A product leader who introduces an intelligent customer experience may create continuous model demand.

Compute therefore moves closer to the operating P&L.

The business cannot treat it entirely as an invisible technology input.

11 · From Headcount Budget to Execution Budget

Managers already make capacity decisions.

Can we hire another analyst?

Can we add another engineering team?

Can we afford another support shift?

Machine execution adds another question.

How much machine capacity should this outcome receive?

The execution budget introduced in Chapter 08 becomes an economic instrument.

A workflow may have:

- a human labor budget
- a model budget
- a compute budget
- an external service budget
- an exception budget
- and an overall cost target.

The manager can then make tradeoffs across the complete execution system.

A stronger model may reduce human review.

Additional retrieval may reduce errors.

More evaluation may reduce risk.

A cheaper model may increase escalation.

More machine execution may reduce cycle time.

The correct answer cannot be determined from infrastructure cost alone.

It depends on the outcome.

12 · Cost Per Outcome

This leads to a more useful economic metric.

Cost per outcome.

- For a customer service workflow: cost per resolved case.
- For software engineering: cost per accepted change.
- For procurement: cost per completed sourcing event.
- For finance: cost per reconciled account.
- For security: cost per investigated incident.
- For sales: cost per qualified opportunity.

The metric connects infrastructure consumption to productive output.

It also makes comparison possible.

Human execution.

Human plus machine execution.

Machine led execution.

Different model configurations.

Different orchestration architectures.

Different levels of autonomy.

All can be compared against the same outcome.

Exhibit 10.3

The AI Native Unit Economic

TOTAL WORKFLOW COST

Human labor Machine inference Compute and infrastructure Tools and external services Evaluation and control Exception handling

DIVIDED BY

Successful outcomes produced

COST PER OUTCOME

EXECUTION VALUE

Economic value per outcome minus cost per outcome.

This is the economic unit of the AI Native workflow.

Not tokens.

Not agents.

Not automation percentage.

Outcome economics.

13 · Market Evidence

Current enterprise evidence increasingly points toward this shift.

McKinsey's 2026 work on agentic economics argues that per-token pricing is becoming insufficient as a measure of enterprise AI economics because the true cost depends on routing, orchestration, workflow design, agent behavior and the complete execution path.

Its research also emphasizes that organizations should evaluate agentic workflows against the fully loaded cost of the work and the value of the resulting outcome rather than optimizing model consumption in isolation.

This distinction becomes more important as agentic systems expand.

The same task can be executed through radically different architectures.

A frontier model.

A smaller model.

A deterministic workflow.

A single agent.

Several agents.

Human review.

No human review.

Each architecture creates a different cost and quality profile.

The economic decision is therefore architectural.

14 · Sentient Interpretation

AI economics and AI architecture are becoming inseparable.

Architecture determines consumption.

A long context window has an economic consequence.

A multi-agent design has an economic consequence.

A retry policy has an economic consequence.

A model routing decision has an economic consequence.

An evaluation architecture has an economic consequence.

An authority design that forces unnecessary human approval has an economic consequence.

A poorly designed workflow has an economic consequence.

The architecture creates the bill.

This is why AI cost management cannot be relegated to procurement after deployment.

Economic design must exist inside the execution architecture itself.

15 · The Economic Control Loop

The enterprise therefore needs another feedback loop.

Execution consumes resources.

Resources create cost.

Execution produces outcomes.

Outcomes create value.

The enterprise compares the two.

Then it changes the system.

Exhibit 10.4

The Economic Control Loop

ALLOCATE
Assign human and machine capacity.

EXECUTE
Run the workflow.

MEASURE
Capture total execution cost.

VALUE
Measure the outcome produced.

COMPARE
Value relative to cost.

ADAPT
Change: Model Context Workflow Agent architecture Human allocation Authority Evaluation Infrastructure

REALLOCATE
Move resources toward stronger execution economics.

This makes economic adaptation part of the operating system.

The enterprise does not merely ask whether the agent works.

It asks whether the execution system deserves more resources.

16 · Cheaper Compute Does Not Remove the Need for Discipline

AI infrastructure will become more efficient.

Hardware will improve.

Models will improve.

Inference engines will improve.

Quantization and other optimization techniques will reduce resource requirements.

Specialized silicon will improve throughput.

Competition will change pricing.

McKinsey's 2026 analysis estimates that combinations of model optimization and infrastructure innovation could reduce inference cost dramatically over time. Its analysis identifies model optimization, advanced packaging, specialized silicon and optical networking among the major levers improving inference economics.

That is important.

But it does not remove the economic problem.

It changes the frontier.

When intelligence becomes cheaper, more workflows become economical.

When more workflows become economical, consumption expands.

The enterprise therefore needs economic discipline even in a world of dramatically cheaper inference.

Perhaps especially then.

Because machine intelligence can become abundant before it becomes free.

And once intelligence becomes a continuously consumed operating resource, the enterprise must answer a harder infrastructure question.

Where should that resource come from?

Public models?

Cloud platforms?

Dedicated capacity?

Private infrastructure?

Specialized models?

Multiple providers?

The answer depends not only on technical capability.

It depends on economics, latency, resilience, security, sovereignty and strategic control.

The enterprise now needs a compute strategy.

The enterprise now needs a compute strategy.

For most enterprises, compute strategy historically belonged inside technology strategy.

Cloud or data center.

Reserved capacity or on demand.

Which provider.

Which region.

Which workload.

AI Native execution changes the significance of those choices.

When machine intelligence becomes part of how the enterprise produces outcomes, compute infrastructure becomes part of productive capacity.

The enterprise is no longer deciding only where applications should run.

It is deciding where machine execution capacity should come from.

That decision affects:

- cost
- latency
- availability
- security
- data locality
- model choice
- scalability
- resilience
- regulatory exposure
- and strategic dependency.

Compute strategy therefore moves closer to operating strategy.

17 · There Is No Single Compute Environment

The AI Native enterprise is unlikely to run every workload through one infrastructure model.

Different execution requirements create different economics.

- Some workloads may use public model APIs.
- Some may use managed cloud model platforms.
- Some may use dedicated cloud capacity.
- Some may use smaller specialized models.
- Some may run inside private environments.
- Some may require edge execution.
- Some may use several providers.

The architecture becomes a portfolio.

The question is not:

Cloud or private?

It is:

Which execution environment provides the strongest combination of capability, economics and control for this workload?

18 · Workload Placement Becomes an Economic Decision

A low volume workflow may be most economical through consumption based access.

There is little reason to reserve expensive capacity that remains idle.

A high volume predictable workload may behave differently.

At sufficient utilization, dedicated capacity may produce stronger economics.

A latency sensitive workflow may require infrastructure closer to execution.

A regulated workload may require stronger data locality.

A resilience critical workflow may require multiple execution paths.

A specialized workload may perform adequately on a smaller model at a fraction of the cost.

The infrastructure decision therefore begins with the workload.

Exhibit 10.5

The AI Workload Placement Test

CAPABILITY
Which model capability does the outcome actually require?

VOLUME
How much execution demand exists?

VARIABILITY
Is demand predictable or bursty?

LATENCY
How quickly must execution complete?

DATA
What information enters the execution environment?

SECURITY
What protection does the workload require?

SOVEREIGNTY
Where may data and execution occur?

RESILIENCE

What happens if the provider or model is unavailable?

ECONOMICS

What is the fully loaded cost at expected utilization?

CONTROL

How much infrastructure and model control does the enterprise require?

PLACEMENT

Public API Managed model platform Dedicated cloud capacity Private infrastructure Edge Hybrid or multi-provider execution

---

The architecture follows the workload.

Not ideology.

19 · Model Routing Becomes Infrastructure Routing

Chapter 08 established model routing as an execution capability.

Its economic importance now becomes clearer.

A workflow does not necessarily require one model.

Different steps may require different intelligence.

A simple extraction may use a small model.

A consequential decision may invoke a stronger model.

A visual task may require multimodal capability.

A routine classification may use deterministic software.

A complex exception may escalate to a frontier model.

The runtime can therefore allocate intelligence dynamically.

This is more than model selection.

It is infrastructure allocation.

The system is deciding which scarce machine resource should be consumed for which piece of work.

20 · The Most Capable Model Should Not Become the Default

Using the strongest available model everywhere is operationally simple.

It is also economically weak.

The objective is not to maximize intelligence per invocation.

It is to provide sufficient intelligence for the outcome.

A model that is materially more expensive but produces no meaningful improvement in the workflow is wasted capacity.

A model that is cheaper but causes repeated retries, human escalation or quality failure may also be economically weak.

The optimal model is therefore not necessarily the cheapest or strongest.

It is the model producing the strongest outcome economics inside the complete execution system.

This creates a routing hierarchy:

use deterministic execution where deterministic execution is sufficient

use smaller models where smaller models are sufficient

use stronger models where stronger reasoning changes the outcome

use human judgment where machine execution no longer provides the appropriate capability.

The execution architecture becomes a resource allocator.

21 · Latency Has Economic Value

Infrastructure cost is not the only economic variable.

Time matters.

A model that costs less but takes substantially longer may be unacceptable in a customer interaction.

A high latency workflow may delay revenue.

A slow engineering agent may reduce developer leverage.

A delayed fraud decision may increase exposure.

A slow industrial control decision may be unusable.

Latency therefore has an economic value.

The relevant optimization is not simply:

cost per inference.

It is:

cost, quality and time relative to the outcome.

This is why workload placement cannot be separated from business context.

The same infrastructure can be economically excellent for one workflow and unacceptable for another.

22 · Resilience Becomes an Operating Requirement

As AI moves into execution, model availability becomes operational availability.

If a recommendation system is unavailable, employees may work around it.

If an autonomous execution system is unavailable, the workflow itself may stop.

That changes resilience requirements.

The enterprise must determine:

- what happens if the preferred model fails
- what happens if a provider becomes unavailable
- what happens if a region fails
- what happens if rate limits are reached
- what happens if capacity becomes constrained
- what happens if model behavior changes unexpectedly.

The answer may be:

- retry
- route to another model
- route to another provider
- degrade to a simpler workflow
- invoke deterministic execution
- return work to humans
- pause execution.

The appropriate response depends on consequence.

But the response must exist.

Exhibit 10.6

The AI Execution Resilience Ladder

PRIMARY

Preferred model and infrastructure path.

ALTERNATE MODEL

Equivalent capability on the same platform.

ALTERNATE PROVIDER

Comparable capability through another infrastructure path.

DEGRADED EXECUTION

Smaller model or reduced workflow.

DETERMINISTIC FALLBACK

Execute only the bounded portions that do not require machine reasoning.

HUMAN FALLBACK

Return execution to human operators.

SAFE STOP

Terminate when continued execution would create unacceptable risk.

Resilience does not require every workload to support every level.

It requires the failure path to match the consequence of interruption.

23 · Data Locality Becomes Execution Locality

AI infrastructure decisions are often discussed through data residency.

Where is the data stored?

Machine execution introduces a broader question.

Where does reasoning occur?

A workflow may retrieve sensitive enterprise information.

Construct context.

Send that context to a model.

Invoke external tools.

Persist state.

Generate traces.

Produce outputs.

The location and governance of execution therefore matter across the entire chain.

For regulated or strategically sensitive workloads, the enterprise may need greater control over:

- where models execute
- where context travels
- where memory persists
- where traces are stored
- which providers can access the execution environment.

Data architecture and compute architecture converge.

24 · Sovereignty Is Not Only a Government Problem

Sovereignty is often framed through national regulation.

That is only one dimension.

Enterprises may also require operational sovereignty.

- Can the organization move the workload?
- Can it change model providers?
- Can it preserve its execution history?
- Can it retain its agent definitions?
- Can it reproduce evaluations elsewhere?
- Can it continue operating if commercial terms change?
- Can it maintain critical execution if access to a provider disappears?

These questions concern strategic dependency.

An AI Native enterprise that embeds machine execution deeply into its operating model becomes more dependent on the infrastructure providing that intelligence.

The deeper the dependency, the more deliberate the architecture should become.

25 · Sentient Interpretation

Model portability will become an operating resilience capability.

This does not mean every enterprise should avoid provider specific features.

Optimization creates value.

Deep platform integration can create value.

The question is whether the enterprise understands where dependency has accumulated.

A low consequence productivity assistant can tolerate substantial provider dependence.

A machine execution system responsible for a critical enterprise workflow deserves a different analysis.

The architecture should know which dependencies are convenient.

And which have become structural.

26 · Physical Infrastructure Returns to Enterprise Strategy

Cloud computing allowed many business leaders to treat physical infrastructure as abstract.

AI is making the physical layer visible again.

Machine intelligence ultimately depends on:

- accelerators
- memory
- networks
- data centers
- electricity
- cooling
- physical capacity.

The rapid expansion of AI infrastructure has made power availability, accelerator supply and data center capacity material constraints on the growth of machine intelligence.

These constraints may appear distant from an individual enterprise consuming model APIs.

They are not.

They influence:

- pricing
- capacity availability
- regional expansion
- latency
- provider strategy
- contract structures
- and resilience.

Machine intelligence is software experienced through physical infrastructure.

The economics of the operating model eventually reach the power grid.

27 · Market Evidence

The International Energy Agency has documented the rapid increase in electricity demand associated with data centers and AI workloads, with data center electricity consumption projected to rise substantially through the second half of the decade.

Major cloud and technology providers are simultaneously committing unprecedented capital to AI infrastructure, including accelerators, networking, data centers and energy supply.

These investments demonstrate that AI capacity is not infinitely elastic.

It must be built.

Powered.

Cooled.

Connected.

And financed.

The enterprise consuming intelligence through an API may not own this infrastructure.

But it still consumes the productive capacity the infrastructure creates.

What This Supports

AI infrastructure has real physical and capital constraints.

Persistent machine execution ultimately depends on those constraints.

Infrastructure economics therefore remain relevant even when enterprises consume AI through managed services.

What Remains Unresolved

The infrastructure frontier is moving rapidly.

Hardware efficiency is improving.

Model efficiency is improving.

New accelerator architectures are emerging.

Energy strategies are changing.

It is not yet possible to infer a stable long term cost curve for machine intelligence.

The enterprise should therefore design for economic adaptability rather than betting its operating model on one permanent infrastructure assumption.

28 · Capacity Planning Changes

Traditional infrastructure capacity planning asks how much technical demand the system will generate.

AI Native capacity planning must also ask how much productive machine capacity the enterprise intends to create.

This connects infrastructure planning to workforce and operating planning.

If a business unit intends to deploy agents across a high volume workflow, the enterprise should understand the resulting inference demand before deployment.

If a new operating model moves substantial work from human execution into machine execution, the financial model should capture the infrastructure consequence.

If an agent population can expand dynamically, capacity controls must prevent unbounded consumption.

Machine workforce planning and infrastructure capacity planning become connected disciplines.

29 · The Machine Capacity Plan

The enterprise can begin with outcomes.

- How much work is expected?
- How much will remain human?
- How much will become machine executed?
- What models will the execution require?
- How many reasoning cycles?
- How much context?
- How many tools?
- How much evaluation?
- How much redundancy?

The result is an estimate of machine capacity demand.

That demand can then be translated into infrastructure requirements and economic budgets.

Exhibit 10.7

From Business Demand to Compute Demand

BUSINESS DEMAND

Transactions Customers Cases Orders Changes Decisions

WORKFLOW DEMAND

How many executions are required?

MACHINE ALLOCATION

Which portions use machine intelligence?

EXECUTION PROFILE

Models Context Agent calls Tools Retries Evaluation

COMPUTE DEMAND

Inference Memory Storage Network Accelerator capacity

ECONOMIC DEMAND

Cost per outcome Total operating cost Capacity commitment

This connects the business forecast to the machine infrastructure forecast.

Compute stops being an opaque technology expense.

It becomes part of operating capacity planning.

30 · AI FinOps Becomes Operating Discipline

Cloud FinOps emerged because elastic infrastructure created a management problem.

Teams could consume resources quickly.

Costs were distributed.

Technical decisions created financial consequences.

Organizations needed visibility, accountability and optimization.

AI creates the same problem with additional complexity.

- Consumption can occur inside model reasoning.
- Costs can cross providers.
- One agent can invoke another.
- Different models have different economics.
- Context size changes cost.
- Retries change cost.
- Evaluation changes cost.
- The same workflow can consume different amounts on different executions.

AI FinOps therefore cannot be only a monthly report showing model spend.

It must connect consumption to execution.

31 · The AI FinOps Loop

The enterprise needs to know:

- who consumed machine capacity
- which workflow consumed it
- which agent consumed it
- which model was used
- why that model was selected
- what the execution cost
- what outcome was produced
- whether a cheaper architecture could produce the same result.

This information allows optimization.

Not indiscriminate cost reduction.

Optimization.

The distinction matters.

Reducing machine expenditure while reducing enterprise value is not efficiency.

It is degradation.

Exhibit 10.8

AI FinOps

VISIBILITY

Where is AI spend occurring?

ATTRIBUTION

Which workflow, domain and outcome consumed it?

UNIT ECONOMICS

What did successful execution cost?

OPTIMIZATION

Can routing, context, models, workflows or infrastructure improve the economics?

GOVERNANCE

Which budgets and limits should apply?

REALLOCATION

Where should additional machine capacity be invested?

The final step matters.

FinOps should not exist only to reduce expenditure.

It should help the enterprise move machine resources toward the places where they produce greater economic value.

32 · Infrastructure Becomes Part of Operating Leverage

The economic promise of machine execution is not that compute is cheap.

It is that machine capacity can produce useful work at economics that differ fundamentally from human labor.

A workflow that previously required one thousand hours of human effort may require far less human effort when machine execution performs large portions of the work.

That can create operating leverage.

But only if the infrastructure cost required to produce the outcome remains below the value created.

This relationship can change.

A poorly designed agent system can consume substantial compute while producing little value.

A well designed workflow can use inexpensive models to remove large amounts of coordination or execution effort.

Infrastructure does not create leverage by itself.

Architecture converts infrastructure into leverage.

33 · The AI Native Capacity Portfolio

The enterprise now has several forms of productive capacity.

Human labor.

Machine intelligence.

Deterministic software.

External services.

Physical infrastructure.

Each has different economics.

Each has different strengths.

Each has different constraints.

The operating model must allocate work across them.

This creates a capacity portfolio.

Exhibit 10.9

The AI Native Capacity Portfolio

HUMAN CAPACITY

Judgment Accountability Ambiguity Relationships Novel conditions

MACHINE INTELLIGENCE

Reasoning Generation Interpretation Adaptive execution

DETERMINISTIC COMPUTE

Rules Transactions Repeatable logic High reliability

EXTERNAL CAPABILITY

Vendors Platforms Specialist services External agents

INFRASTRUCTURE

Models Accelerators Storage Network Power

MANAGEMENT OBJECTIVE

Allocate the portfolio to maximize: Outcome quality Economic value Speed Resilience Control

The AI Native enterprise does not optimize one resource.

It optimizes the complete execution portfolio.

34 · Sentient Interpretation

The economics of compute ultimately return the publication to its starting point.

The objective was never maximum AI.

It was never maximum automation.

It was never maximum agent deployment.

The objective is the strongest execution system for the outcome.

Compute is another productive resource.

An extraordinary one.

It can scale intelligence.

It can make execution persistent.

It can create capacity without equivalent headcount.

It can reduce the marginal cost of some forms of knowledge work.

But it remains a resource.

Resources require allocation.

Allocation requires economics.

Economics requires outcomes.

This is why the AI Native enterprise must connect infrastructure all the way back to value.

35 · The Infrastructure Economic Architecture

The complete relationship can now be stated.

Business demand creates workflow demand.

Workflow design determines execution allocation.

Execution allocation creates human and machine capacity demand.

Machine execution creates infrastructure demand.

Infrastructure demand creates cost.

Execution produces outcomes.

Outcomes produce economic value.

The enterprise measures the relationship.

Then it reallocates capacity.

Exhibit 10.10

The Infrastructure Economic Architecture

BUSINESS OUTCOME

WORKFLOW

HUMAN + MACHINE ALLOCATION

EXECUTION ARCHITECTURE

MODEL + COMPUTE DEMAND

INFRASTRUCTURE

EXECUTION COST

OUTCOME PRODUCED

ECONOMIC VALUE

MEASURE

ADAPT

REALLOCATE CAPACITY

This is the economic operating loop of machine execution.

The architecture creates the bill.

The outcome determines whether the bill was worth paying.

36 · From Capability to Value

The publication has now established the execution system.

The enterprise can redesign workflows.

Allocate work across humans and machines.

Delegate authority.

Redesign management.

Reconfigure organizational architecture.

Build the execution substrate.

Govern a mixed population of agents.

And provision the infrastructure required to operate it.

But none of those capabilities establishes that transformation has created economic value.

More agents do not prove value.

More automation does not prove value.

Higher model consumption does not prove value.

Fewer employees do not automatically prove value.

More output does not necessarily prove value.

The enterprise must now answer the question that determines whether AI Native transformation deserves to scale.

> *What economic value did the operating model create?*

That is the next layer of the AI Native Operating Model.

11 · From Productivity to Economic Value

Read online: https://sentient-review.com/perspectives/the-ai-native-operating-model/10-infrastructure-and-the-economics-of-compute

Part IV

The Economics of Transformation

Chapter 11

From Productivity to Economic Value

How AI activity travels through accepted output, released capacity and capacity destination before becoming financial consequence.

> *Did AI improve the execution of work — and did that improvement change the economics of the enterprise?*

AI can make work faster without making the enterprise more valuable.

That distinction is becoming one of the most important in enterprise AI.

A developer completes a task faster.

An analyst produces a report faster.

A customer service representative drafts a response faster.

A manager summarizes information faster.

A marketing team creates more content.

A legal team reviews more documents.

These are real improvements.

They may matter.

But they are not yet economic value.

Time saved is not automatically cost removed.

Output created is not automatically revenue created.

Tasks automated are not automatically capacity released.

Capacity released is not automatically redeployed.

Headcount avoided is not automatically margin improvement.

A faster workflow is not automatically a better business outcome.

The enterprise therefore needs to distinguish between two questions.

Did AI improve the execution of work?

and

Did that improvement change the economics of the enterprise?

The first is productivity.

The second is value.

AI Native transformation must connect them.

01 · Productivity Is an Intermediate Outcome

Productivity matters.

If the same resources produce more useful output, productivity has improved.

If the same outcome requires fewer resources, productivity has improved.

If an outcome can be produced faster without degrading quality, productivity has improved.

But productivity sits inside a larger economic system.

Consider a workflow requiring ten thousand human hours each year.

AI reduces the required effort by 20 percent.

Two thousand hours have theoretically been released.

What happens next?

If the same people continue performing the same volume of work and the released time is absorbed by meetings, waiting or lower priority activity, little economic value may appear.

If the organization increases throughput by 20 percent and additional throughput generates revenue, value may appear.

If the organization avoids future hiring, value may appear.

If employees redirect those hours toward higher value work, value may appear.

If customer cycle time falls and conversion improves, value may appear.

If service quality rises and retention improves, value may appear.

The productivity gain creates economic potential.

Management determines whether that potential becomes value.

02 · The Productivity Transmission Problem

This creates a gap between technical improvement and financial consequence.

Call it the Productivity Transmission Problem.

AI changes work.

But the economic system surrounding the work must transmit that change into an outcome that matters.

The transmission may occur through:

- cost

- capacity
- revenue
- speed
- quality
- risk
- capital efficiency.

If none of these changes materially, the productivity improvement may remain local.

The employee feels faster.

The enterprise economics remain largely unchanged.

Exhibit 11.1

The Productivity Transmission Chain

AI CAPABILITY

WORK CHANGES

Less effort Faster execution Greater throughput Higher quality New capability

CAPACITY EFFECT

Time released Machine capacity added Human capacity redirected Bottleneck removed

OPERATING EFFECT

Lower cost More volume Faster cycle Better quality Lower risk Greater responsiveness

BUSINESS EFFECT

Revenue Margin Retention Working capital Loss avoidance Capital efficiency Strategic capacity

ECONOMIC VALUE

Every arrow matters.

A break anywhere in the chain can prevent technical capability from becoming economic value.

03 · Market Evidence

Current enterprise evidence shows why this distinction matters.

McKinsey's research on generative AI has estimated substantial potential economic value across enterprise functions, particularly customer operations, marketing and sales, software engineering, and research and development.

But its subsequent enterprise surveys have repeatedly found that broad adoption does not automatically translate into material enterprise level financial impact. Organizations reporting the strongest effects tend to redesign workflows, establish performance measures and change management systems around AI rather than merely making tools available.

The evidence supports a distinction between technology adoption and economic capture.

AI can create technical capability before the organization develops the operating mechanisms required to capture its value.

What This Supports

AI can produce meaningful improvements in individual work and workflows.

Economic impact depends on what the organization does with the resulting capacity, quality, speed and capability.

Workflow redesign and management action are therefore part of the value mechanism.

What Remains Unresolved

Enterprise reporting remains inconsistent.

Organizations use different definitions of productivity and value.

Many estimates describe potential rather than realized financial impact.

Self reported surveys may overstate or understate results.

Causal attribution is difficult because AI deployment often occurs alongside broader organizational and technology changes.

The evidence supports economic potential.

It does not support treating every reported productivity gain as realized enterprise value.

04 · Time Saved Is Not Money Saved

This distinction deserves explicit treatment because it is frequently violated.

Suppose an AI system saves an employee five hours each week.

The employee's salary does not fall by five hours.

The enterprise has not automatically saved five hours of compensation.

It has created available capacity.

That capacity can produce value only if something happens to it.

It may be:

- removed
- redeployed
- used to increase throughput
- used to improve quality
- used to absorb growth
- used to reduce overtime
- used to avoid hiring
- used for higher value work.

Without one of these mechanisms, the financial value of the saved time remains uncertain.

This does not make the productivity gain false.

It means the economic claim must be precise.

Exhibit 11.2

The Capacity Conversion Test

1 · WAS CAPACITY ACTUALLY RELEASED?

Did the workflow require less human effort? If no: No capacity value exists. If yes: continue.

2 · IS THE CAPACITY IDENTIFIABLE?

Can the enterprise determine where the released capacity exists? If no: The gain may disappear into the organization. If yes: continue.

3 · WHAT HAPPENS TO IT?

Remove — Reduce labor cost. Avoid — Absorb growth without equivalent hiring. Redeploy — Move capacity toward higher value work. Expand — Increase productive throughput. Improve — Use capacity to increase quality, service or control.

4 · DID AN ECONOMIC OUTCOME CHANGE?

If yes: Economic value can be measured. If no: The enterprise has created productivity without proving economic capture.

---

05 · Sentient Interpretation

The phrase time saved has become one of the weakest economic measures in enterprise AI.

It is useful operational evidence.

It is not a financial outcome.

The missing question is always:

What happened to the time?

If nobody can answer that question, the enterprise cannot confidently claim the associated economic value.

This is why AI Native transformation requires more than productivity instrumentation.

It requires capacity management.

The enterprise must decide what released capacity will become.

06 · Capacity Release Is a Management Event

Chapter 06 established that managers increasingly allocate both human and machine capacity.

Chapter 10 established that AI Native execution creates a portfolio of productive resources.

The economic implication now becomes clear.

When AI changes the amount of human effort required for a workflow, management must respond.

Suppose an accounts payable workflow requires 30 percent less human effort.

Management has choices.

- Reduce staffing.
- Absorb transaction growth.
- Reassign employees to exception handling.
- Improve supplier management.
- Accelerate payment optimization.
- Strengthen controls.
- Do nothing.

Each choice produces different economics.

The technology created the possibility.

The management system determines the economic result.

07 · Productivity Without Capacity Management Creates Slack

Slack is not inherently bad.

Organizations need resilience.

People need room for learning, innovation and unexpected work.

But unintentional slack is not the same as deliberate strategic capacity.

If AI releases thousands of hours across hundreds of employees without an explicit mechanism for reallocating that capacity, the enterprise may struggle to locate the value it expected.

This is particularly likely when productivity improvements are fragmented across individuals.

Ten minutes here.

Twenty minutes there.

One hour somewhere else.

The aggregate number may look enormous.

The capacity may be operationally impossible to harvest.

This creates another important distinction.

Theoretical capacity

versus

harvestable capacity.

08 · The Granularity Problem

A productivity improvement can be economically real but too fragmented to convert directly into cost.

If 1,000 employees each save 15 minutes a day, the aggregate annual time may be substantial.

But the enterprise cannot necessarily remove the equivalent number of positions.

The released capacity is distributed.

Its value depends on whether work can be redesigned so the fragments accumulate into usable capacity.

This is another reason workflow redesign matters.

Task level productivity often creates fragmented savings.

Workflow level redesign can consolidate them.

Exhibit 11.3

From Fragmented Savings to Harvestable Capacity

Task Optimization

- 5 minutes saved
- 10 minutes saved

- 20 minutes saved
- 15 minutes saved

FRAGMENTED CAPACITY

Distributed across people and activities. Difficult to capture economically.

Workflow Redesign

- Remove steps
- Reallocate execution
- Automate coordination
- Change authority
- Redesign exceptions

HARVESTABLE CAPACITY

Concentrated reduction in human effort. Can be removed, redeployed, used for growth or used for higher value work.

The task is often too small to reveal the economics.

The workflow contains the economic opportunity.

09 · Value Can Arrive Without Headcount Reduction

Economic value should not be reduced to labor removal.

That is one transmission mechanism.

It is not the only one.

An enterprise experiencing rapid growth may use AI to avoid adding headcount.

A constrained engineering organization may use AI to increase delivery capacity.

A service organization may use machine execution to reduce customer waiting time.

A bank may use AI to detect risk earlier.

A manufacturer may reduce downtime.

A pharmaceutical company may shorten part of a research process.

A sales organization may improve conversion.

A finance organization may close faster.

A security team may investigate more incidents with the same workforce.

None requires immediate headcount reduction to create economic value.

The relevant question is whether the operating outcome changed in a way that matters economically.

10 · The Seven Value Pathways

AI Native transformation can create economic value through at least seven pathways.

- 1 · Cost — Reduce the resources required to produce an existing outcome.
- 2 · Capacity — Produce more output from the existing resource base.
- 3 · Revenue — Increase conversion, volume, monetization or addressable demand.
- 4 · Speed — Compress cycle time where time has economic consequence.
- 5 · Quality — Improve output where quality affects cost, revenue or retention.
- 6 · Risk — Reduce the probability or consequence of loss.
- 7 · Capital — Improve utilization of working capital, infrastructure or other productive assets.

These pathways can interact.

A faster workflow may increase revenue.

Higher quality may reduce cost.

Greater capacity may avoid capital expenditure.

Risk reduction may protect revenue.

The value architecture must make these relationships explicit.

Exhibit 11.4

The AI Native Value Pathways

WORKFLOW TRANSFORMATION

COST
Lower cost to produce the outcome

CAPACITY
More productive output from existing resources

REVENUE
Greater economic output

SPEED
Shorter time to economically meaningful result

QUALITY
Better result with measurable consequence

RISK
Lower expected loss or exposure

CAPITAL
More productive use of assets and infrastructure

ENTERPRISE ECONOMIC VALUE

The value pathway should be identified before transformation begins.

Otherwise the enterprise may deploy AI successfully and only later ask what economic result it was supposed to produce.

11 · Economic Value Needs a Counterfactual

To claim value, the enterprise needs to know what would have happened without the intervention.

If a workflow costs $10 million after AI deployment, that number alone says nothing.

- Would it otherwise have cost $12 million?
- Would volume have increased?
- Would quality have fallen?
- Would another twenty employees have been hired?
- Would customer loss have increased?
- Would infrastructure demand have changed?

Economic value is measured relative to a counterfactual.

The counterfactual does not need to be perfect.

But it must be explicit.

Without one, every improvement risks becoming attribution by narrative.

12 · Baseline Before Transformation

The enterprise therefore needs a baseline before changing the workflow.

- How much does the outcome cost today?
- How much human effort does it require?
- How long does it take?
- What volume does it support?
- What quality does it achieve?
- What failure rate exists?
- What risk exposure exists?
- What infrastructure does it consume?

Only then can the transformed system be compared against something meaningful.

This sounds obvious.

In practice, many organizations begin AI deployment without measuring the economics of the work being changed.

They can then demonstrate technical success without demonstrating economic change.

13 · The Value Case Should Live at the Workflow

This is another reason the workflow is the primary unit of AI Native operating design.

The task is often too small.

The application is too narrow.

The department is too broad.

The workflow contains:

- the human effort
- the machine effort
- the coordination cost
- the cycle time
- the exceptions
- the outcome
- and the economic consequence.

That makes it the natural unit for value measurement.

Exhibit 11.5

The Workflow Value Ledger

BASELINE

Current cost Human effort Cycle time Volume Quality Risk Capital consumption

INTERVENTION

What changed? Workflow Allocation Authority Management Agent architecture Infrastructure

NEW EXECUTION ECONOMICS

Human cost Machine cost Infrastructure cost Control cost Exception cost

OUTCOME CHANGE

Cost Capacity Revenue Speed Quality Risk Capital

REALIZED VALUE

What measurable economic consequence occurred?

CONFIDENCE

How strongly can the change be attributed to the intervention?

The ledger turns AI value from narrative into operating evidence.

14 · Sentient Interpretation

The enterprise does not need another AI dashboard.

It needs an economic ledger.

Model usage belongs in technical telemetry.

Agent counts belong in platform telemetry.

Adoption belongs in change telemetry.

None of them proves value.

The value ledger begins with the business outcome and works backward through the execution system.

This changes executive conversation.

Instead of:

How many employees are using AI?

the enterprise can ask:

Which workflows have changed their economics?

Instead of:

How many agents have we deployed?

it can ask:

Which agents are producing measurable economic value?

Instead of:

How many hours have we saved?

it can ask:

What capacity was actually converted, and into what?

That is a stronger management system.

15 · From Use Case Portfolio to Value Portfolio

Many enterprise AI programs organize around use cases.

Identify opportunities.

Prioritize use cases.

Launch pilots.

Track adoption.

Scale successful implementations.

The model is useful during exploration.

But a mature AI Native enterprise needs another portfolio.

A value portfolio.

Every material intervention should have:

- an outcome
- a baseline
- a value pathway
- an economic hypothesis
- an execution cost
- a measurement method
- an accountable owner
- a realized result.

The portfolio can then be managed according to economic evidence.

Some interventions should scale.

Some should be redesigned.

Some should remain because they create strategic capability that cannot yet be captured through near term financial measures.

Some should stop.

AI Native transformation becomes capital allocation.

16 · The Value Hypothesis

Before redesigning a workflow, the enterprise should be able to state:

If we change this execution system in this way, we expect this operating measure to change, which should create this economic consequence.

For example:

- If machine execution handles routine service resolution, human specialists can concentrate on complex cases, reducing cost per resolved case while maintaining customer satisfaction.
- If agents reduce engineering cycle time, product capacity increases without equivalent headcount growth.
- If machine monitoring detects payment anomalies earlier, expected financial loss declines.

The hypothesis connects architecture to economics before implementation begins.

Without it, value is reconstructed after the fact.

Exhibit 11.6

The Value Hypothesis

INTERVENTION

What are we changing?

OPERATING MECHANISM

Why should execution improve?

OPERATING MEASURE

What should change?

VALUE PATHWAY

Cost Capacity Revenue Speed Quality Risk Capital

ECONOMIC CONSEQUENCE

What financial or strategic value should result?

EVIDENCE

How will we know?

This is the economic design contract for transformation.

17 · Productivity Is Not the Destination

Productivity remains important.

It tells the enterprise whether execution is becoming more efficient.

But productivity alone cannot determine whether an intervention deserves additional capital, infrastructure or organizational disruption.

The enterprise needs to know what the productivity gain became.

- Did it become margin?
- Growth?
- Capacity?
- Speed?
- Quality?
- Risk reduction?
- Capital efficiency?
- Or did it remain unconverted?

That distinction separates AI adoption from economic transformation.

The AI Native enterprise therefore manages not only the productivity of humans and machines.

It manages the conversion of execution improvement into enterprise value.

And once value becomes measurable, another question appears.

The enterprise will have many possible interventions.

Many workflows.

Many agents.

Many investments.

Many claims on capital and management attention.

Not all will create equal value.

The organization must decide where to invest, where to scale, where to intervene and where to stop.

That requires a system for economic allocation.

Once value can be measured, AI Native transformation stops being a collection of technology initiatives.

It becomes a portfolio of economic interventions.

Every workflow competes for something.

Engineering capacity.

Management attention.

Transformation capacity.

Compute.

Capital.

Organizational disruption.

Human adoption.

Risk tolerance.

The enterprise cannot transform everything simultaneously.

Nor should it.

The economic system must determine where the next unit of human and machine capacity should go.

That is an allocation problem.

18 · Value Should Determine Scale

The traditional technology deployment model often treats scale as the destination.

Pilot.

Validate.

Deploy.

Adopt.

Scale.

But scale should not follow technical success automatically.

An agent may work.

A workflow may operate reliably.

Employees may use it.

None establishes that additional investment is justified.

Scale should follow evidence that the intervention produces enough value to justify the resources it consumes.

The question becomes:

What happens economically if we deploy the next unit of machine execution here?

That is a marginal value question.

19 · Marginal Value Matters More Than Average Value

Suppose an agentic workflow has already automated the simplest 60 percent of a process.

The economics are excellent.

The remaining 40 percent contains ambiguous, high consequence exceptions.

Automating the next ten percent may require:

- stronger models
- more context
- more evaluation
- more human oversight
- more complex controls
- more expensive infrastructure.

The first unit of automation and the last unit do not necessarily have the same economics.

This creates diminishing returns.

The enterprise therefore cannot assume:

If some automation created value, more automation will create more value.

Each additional unit should be evaluated against the incremental cost and incremental value it creates.

Exhibit 11.7

The Marginal Value Test

ADDITIONAL CAPABILITY

What new work becomes machine executable?

ADDITIONAL COST

Models Compute Tools Control Evaluation Human oversight Transformation effort

ADDITIONAL VALUE

Cost removed Capacity created Revenue increased Cycle time reduced Quality improved Risk reduced Capital released

MARGINAL ECONOMICS

Additional Value minus Additional Cost

DECISION

Scale Redesign Hold Stop

The objective is not maximum machine execution.

It is maximum economic value from the execution system.

20 · Scale Requires Gates

A disciplined enterprise should not move directly from experiment to broad deployment.

Each stage should establish stronger evidence.

An experiment asks:

Can this work?

A pilot asks:

Can this work inside the enterprise?

A production deployment asks:

Can this work reliably under real operating conditions?

A scaling decision asks:

Does expanding this create enough value to justify additional investment?

These are different questions.

Conflating them causes technically successful experiments to become economically weak production systems.

Exhibit 11.8

The AI Native Scaling Gates

GATE 1 · CAPABILITY

Can the machine perform the required work?

GATE 2 · EXECUTION

Can it operate reliably inside the workflow?

GATE 3 · CONTROL

Can authority, risk and accountability be governed?

GATE 4 · ECONOMICS

Does the transformed workflow produce stronger outcome economics?

GATE 5 · SCALE

Do the marginal economics remain attractive as deployment expands?

SCALE

Scale is earned through evidence.

Not enthusiasm.

21 · Value Realization Needs an Owner

A recurring weakness in transformation programs is the separation between implementation and value realization.

Technology owns deployment.

A business function owns adoption.

Finance measures cost.

Someone else eventually asks whether the expected benefit appeared.

This fragments accountability.

Every material AI Native intervention should therefore have an accountable value owner.

Not necessarily a new role.

A responsibility.

The value owner is accountable for ensuring that the operating change produces the intended economic consequence.

If AI releases capacity, the owner determines what happens to it.

If throughput increases, the owner ensures the organization can use it.

If cycle time falls, the owner connects speed to the business outcome.

If risk decreases, the owner establishes how the reduction will be measured.

The agent cannot own value.

The technology platform cannot own value by itself.

Economic value remains a management responsibility.

22 · Benefits Do Not Realize Themselves

Suppose an AI system reduces analysis time by 40 percent.

The implementation team declares success.

But the economic hypothesis assumed that analysts would use the released capacity to serve more customers.

If customer volume does not increase, the expected value has not been realized.

The technical result is real.

The economic result is not.

Value realization therefore requires active intervention after deployment.

- Capacity may need to be reassigned.
- Performance targets may need to change.
- Roles may need to change.
- Work may need to be removed.
- Budgets may need to move.
- Incentives may need to change.

The operating model must absorb the productivity improvement.

Otherwise the value remains theoretical.

23 · The Value Realization Loop

AI Native transformation therefore requires a loop extending beyond deployment.

Exhibit 11.9

The Value Realization Loop

HYPOTHESIS

What economic consequence should occur?

INTERVENE

Redesign the execution system.

DEPLOY

Put the new operating model into production.

MEASURE

What changed operationally?

CONVERT

What must management change to capture the benefit?

REALIZE

What economic value actually appeared?

VERIFY

How confident are we that the intervention caused it?

REALLOCATE

Scale, redesign or stop.

The intervention is not complete when the technology ships.

It is complete when the intended value has either been realized or disproven.

24 · Measurement Confidence Matters

Not every economic result can be measured with the same confidence.

Some are direct.

A workflow previously required $5 million of external processing cost.

It now requires $3 million.

The economic difference is relatively visible.

Others require more inference.

Customer retention improved after an AI service system was introduced.

But pricing changed at the same time.

The product improved.

Market conditions changed.

How much of the retention improvement belongs to AI?

The enterprise should not solve this uncertainty by pretending it does not exist.

It should state confidence.

Three levels of value evidence:

- Direct — The economic consequence can be observed with strong attribution.
- Attributed — The operating change is measurable and there is reasonable evidence connecting it to financial value.
- Indicative — The intervention appears strategically or economically valuable, but attribution remains uncertain.

Direct examples:

- labor removed
- vendor expenditure removed
- infrastructure expenditure reduced
- hiring avoided against an approved capacity plan.

Attributed examples:

- cycle time improvement associated with increased conversion
- service improvement associated with reduced churn
- engineering capacity associated with additional delivery.

Indicative examples:

- improved decision quality
- greater organizational responsiveness
- new strategic capability
- early risk reduction.

These categories prevent the enterprise from forcing false precision onto uncertain outcomes.

25 · Sentient Interpretation

Economic rigor does not require pretending every form of value can be converted into an exact dollar amount.

It requires distinguishing what is known from what is estimated.

False precision weakens the value case.

A credible AI Native economic system can say:

We know this.

We estimate this.

We believe this may be occurring, but the evidence is not yet sufficient.

That is stronger than assigning invented financial values to every benefit.

26 · Strategic Value Is Still Value

Some interventions should proceed even when immediate financial return is difficult to establish.

An enterprise may need to build agent infrastructure before enough workflows exist to fully utilize it.

It may need to develop organizational capability.

It may need to learn how to govern autonomous execution.

It may need to establish data foundations.

It may need to protect a strategic market position.

It may need to respond to a competitor whose cost structure is changing.

These investments can be rational.

But they should be classified correctly.

They are strategic capability investments, not realized operating savings.

This distinction protects the integrity of the economic system.

27 · Value Has a Time Horizon

Economic interventions also operate across different time horizons.

Some create immediate value.

A machine workflow removes external processing cost.

Some create medium term value.

Automation allows growth without equivalent hiring.

Some create long term value.

A new operating architecture changes the enterprise's structural cost or innovation capacity.

The portfolio therefore needs to distinguish:

- realized value
- committed value
- expected value
- strategic option value.

Mixing them produces inflated transformation claims.

Exhibit 11.10

The Value Horizon

REALIZED

Already visible in operating or financial results.

COMMITTED

Actions have occurred that should produce measurable value. Example: approved hiring removed from the workforce plan.

EXPECTED

Evidence indicates likely value, but realization has not occurred.

STRATEGIC OPTION

Capability created that may enable future value.

All four can matter.

They should not be reported as though they are equivalent.

28 · Economic Value Must Survive Total Cost

Gross benefit is not net value.

Suppose an AI Native workflow produces $10 million of annual operating benefit.

But it requires:

- $2 million of model and infrastructure cost
- $1 million of software and platform cost
- $1 million of additional control and evaluation
- $500,000 of human exception management
- $1 million of ongoing engineering and operations.

The intervention did not create $10 million of net economic value.

Its economics must include the cost of operating the new system.

This seems elementary.

It becomes difficult when AI costs are distributed across technology budgets, business units, shared platforms and external providers.

The workflow value ledger must bring them together.

29 · From Return on AI to Return on Execution

The enterprise may be tempted to calculate an enterprise wide return on AI.

That number can be useful at a high level.

It can also hide important differences.

One workflow may produce exceptional economics.

Another may destroy value.

A third may be strategically necessary but financially negative in the near term.

Combining them into one number can conceal where the operating model is actually working.

The stronger unit is the transformed execution system.

What did we invest to change this workflow?

What does the new execution system cost?

What economic outcome does it produce?

This is return on execution.

30 · Portfolio Allocation

Once individual interventions have comparable economic evidence, the enterprise can allocate resources across them.

A portfolio might contain:

- high value interventions ready to scale
- promising interventions requiring redesign
- strategic capability investments
- experimental interventions with uncertain economics
- mature interventions requiring optimization
- interventions that should be terminated.

This creates discipline.

AI Native transformation stops being a race to accumulate use cases.

It becomes a process of allocating scarce resources toward stronger operating economics.

Exhibit 11.11

The AI Native Value Portfolio

SCALE

Strong evidence Strong economics Acceptable risk Invest further.

REDESIGN

Potential value Weak execution economics Change the operating architecture.

DEVELOP

Strategically important Economic evidence still emerging Continue deliberately.

OPTIMIZE

Established value Improve cost, quality or capacity.

STOP

Insufficient value Unacceptable risk Weak marginal economics Remove resources.

The portfolio is dynamic.

An intervention can move between categories as evidence changes.

31 · Stopping Is Part of Value Creation

Transformation programs often develop institutional momentum.

Teams form.

Budgets are approved.

Executives sponsor initiatives.

Technology is purchased.

Stopping becomes politically difficult.

But economic discipline requires termination.

An AI intervention that does not create sufficient value should not survive merely because it contains advanced technology.

Resources tied to weak interventions cannot be invested elsewhere.

Stopping therefore creates value through reallocation.

This is another difference between experimentation and operating transformation.

Experiments generate learning.

Operating systems must justify continued resources.

32 · The Executive Economic Scorecard

Executives do not need to see every model call.

They need to understand whether machine execution is changing enterprise economics.

The executive view should therefore connect transformation to outcomes.

Exhibit 11.12

The AI Native Economic Scorecard

OUTCOME

What business outcome is being transformed?

BASELINE
What were the original economics?

EXECUTION CHANGE
What changed in the operating model?

HUMAN CAPACITY
What happened to human effort?

MACHINE CAPACITY
What machine resources are consumed?

TOTAL COST
What does the new execution system cost?

VALUE PATHWAY
Cost Capacity Revenue Speed Quality Risk Capital

REALIZED VALUE
What measurable economic consequence has appeared?

CONFIDENCE
Direct Attributed Indicative

MARGINAL ECONOMICS
Does further scale still create value?

DECISION
Scale Redesign Develop Optimize Stop

This is the economic management system for AI Native transformation.

33 · Market Evidence and the Value Gap

The broader market evidence continues to show a gap between AI adoption and enterprise level economic transformation.

Organizations report widespread experimentation and increasing deployment while fewer demonstrate material financial impact at enterprise scale.

The distinction is consistent with the operating argument developed throughout this publication.

Technology availability is not the same as workflow redesign.

Workflow redesign is not the same as operating adoption.

Operating adoption is not the same as capacity conversion.

Capacity conversion is not the same as economic value.

Each transition requires deliberate management.

This is why AI value cannot be inferred from adoption alone.

34 · Sentient Interpretation

The enterprise should stop asking whether AI has a return.

That question is too broad to be useful.

There is no single economic property called AI.

There are execution systems.

Some create value.

Some do not.

Some create value only after the organization changes around them.

Some create strategic options.

Some should never scale.

The management task is to distinguish them.

AI Native transformation therefore requires economic selection.

Not belief.

Not skepticism.

Selection.

Invest where the evidence strengthens.

Intervene where the architecture is weak.

Stop where the economics fail.

Move capacity toward the execution systems producing greater value.

That is how the operating model compounds.

35 · From Value to Intervention

The economics of transformation are now visible.

The enterprise can establish a baseline.

Form a value hypothesis.

Redesign the workflow.

Measure execution.

Convert released capacity.

Calculate total cost.

Track realized value.

Assess confidence.

Evaluate marginal economics.

Allocate resources.

Scale what works.

Stop what does not.

But the system will not always behave as intended.

Agents will fail.

Workflows will create unexpected bottlenecks.

Managers will preserve old coordination mechanisms.

Authority will be too broad or too narrow.

Human review will become a queue.

Compute costs will exceed expectations.

Employees will work around the system.

Local optimization will damage the larger outcome.

Metrics will produce unintended behavior.

Some interventions will create technical success and operating failure.

An AI Native enterprise therefore needs one more capability before it can adapt.

It must know how the operating model fails.

And it must know where to intervene when it does.

Read online: https://sentient-review.com/perspectives/the-ai-native-operating-model/11-from-productivity-to-economic-value

Chapter 12

Failure Modes and Intervention

How AI Native transformation breaks, how deterioration becomes visible and how leaders intervene.

> *The problem is not simply whether the AI works. The problem is whether the operating model works.*

An AI Native operating model will fail.

Not because the premise is wrong.

Because every operating model fails somewhere.

Work accumulates where capacity was misjudged. Authority proves too narrow for the work it governs. Controls create delay. Organizational boundaries interrupt execution. Management systems continue long after the conditions that justified them have disappeared.

The difference in an AI Native enterprise is the speed, distribution and opacity with which these failures can develop.

Machine actors can execute continuously. They can initiate work, call tools, coordinate with other systems and produce downstream actions at a rate that human operating structures were not designed to absorb. A poorly designed workflow can therefore become more efficient at producing the wrong thing. A badly placed approval can turn machine speed into a human queue. An agent with insufficient authority can escalate almost everything. An agent with excessive authority can propagate an error before a human understands that the operating condition has changed.

The problem is not simply whether the AI works.

The problem is whether the operating model works.

That distinction changes how failure must be understood.

A model error is a technical event.

An operating failure is structural.

It appears when the allocation of work, authority, management, organization, execution architecture, infrastructure or economics no longer produces the intended enterprise outcome.

The enterprise therefore needs more than AI monitoring.

It needs the ability to observe its own operating model.

01 · Failure Is Operating Evidence

The first generation of enterprise AI measurement has concentrated heavily on the machine.

Accuracy.

Latency.

Token consumption.

Tool success.

Task completion.

Hallucination rates.

Evaluation scores.

These measures matter. None is sufficient.

An agent can perform its assigned task successfully while the workflow around it deteriorates.

Consider an agent that prepares a customer credit decision in thirty seconds instead of thirty minutes. If every recommendation still enters a human approval queue that requires four hours to clear, the agent has improved a task without materially improving the workflow.

The machine succeeded.

The operating model did not.

Or consider an agent that resolves most service requests autonomously but generates enough uncertain cases to double the workload of a specialist exception team.

Again, local performance can improve while system performance declines.

This is the central diagnostic problem of AI Native operations.

Local machine success does not establish operating success.

Production evidence already points toward this distinction. A large 2025 study of practitioners operating AI agents in production found that reliability remained the leading development challenge. Most production systems remained deliberately constrained. Sixty-eight percent executed no more than ten steps before human intervention, while 74 percent relied primarily on human evaluation. The evidence does not suggest that useful agents cannot be deployed. It suggests that production autonomy remains an operating design problem, not merely a model capability problem.

The same tension appears in enterprise evaluation. A 2026 survey of 157 enterprise respondents found that half of organizations running evaluations had experienced an

AI feature that passed internal evaluation and subsequently produced a customer-facing failure. The sample is directional rather than representative of the entire enterprise market, but the finding exposes an important weakness: passing an evaluation is not equivalent to surviving the operating environment.

The environment contains conditions the test does not.

Real customers.

Incomplete context.

Changing data.

Conflicting objectives.

Permission boundaries.

Exceptions.

Downstream dependencies.

Human behavior.

Organizational latency.

Economic constraints.

The operating model is where these conditions meet.

Failure must therefore be treated as evidence about the design of that system.

Exhibit 12.1

The Operating Failure Stack

OUTCOME
Is the intended enterprise result being produced?

WORKFLOW
Is work moving through the system correctly?

ALLOCATION
Is each unit of work assigned to the right human or machine actor?

AUTHORITY
Can each actor make the decisions required to perform its work?

MANAGEMENT
Can the system direct, supervise, escalate and adapt execution?

ORGANIZATION
Do organizational boundaries support or obstruct the workflow?

EXECUTION ARCHITECTURE
Are agents, models, tools, context and control mechanisms behaving as intended?

INFRASTRUCTURE + COMPUTE

Is machine capacity being consumed efficiently?

ECONOMICS

Is the system producing value greater than the resources it consumes?

A failure observed at one layer may originate somewhere else.

Slow execution may look like an agent problem but originate in an approval structure.

High compute consumption may look like an infrastructure problem but originate in unnecessary multi-agent decomposition.

Poor output quality may look like a model problem but originate in missing enterprise context.

Low economic value may look like a use-case problem but originate in automating a task whose surrounding workflow was never redesigned.

The purpose of diagnosis is to locate that difference.

The location of the symptom is not necessarily the location of the intervention.

02 · The First Failure: Automating the Task, Preserving the System

The most accessible form of enterprise AI transformation is task substitution.

A person performed an activity.

A machine can now perform some or all of it.

The activity is automated.

This can create genuine value. It can also preserve nearly every structural feature of the operating model that made the activity expensive in the first place.

The forms remain.

The handoffs remain.

The approvals remain.

The reporting hierarchy remains.

The queue remains.

The organizational boundary remains.

Only one participant becomes faster.

This is task automation without workflow redesign.

It is one of the most consequential failure modes because early performance can make the transformation appear successful.

A machine may reduce the execution time of an individual task by 80 or 90 percent while producing little change in end-to-end cycle time.

The reason is straightforward.

The task was not the system.

If a workflow contains six stages and only one becomes dramatically faster, the remaining stages determine how much of that speed reaches the outcome. The bottleneck moves.

This is not unique to AI. It is a basic property of systems. What changes with AI is the magnitude of the mismatch. Machine execution can compress work so sharply that previously tolerable coordination structures become dominant constraints.

The enterprise then discovers that it did not automate a workflow.

It accelerated traffic into the next queue.

Gartner reached a related conclusion in its 2025 assessment of agentic AI adoption, warning that organizations were misapplying agents and that many implementations would struggle with cost, business value and risk controls. Its recommendation was notable: where agentic systems are appropriate, enterprises may need to rethink workflows rather than simply integrate agents into existing processes.

That distinction is fundamental.

AI adoption asks:

Where can AI perform an existing task?

AI Native transformation asks:

If this capability existed when the workflow was designed, would we design the workflow this way at all?

Those questions produce different enterprises.

03 · The Human Approval Bottleneck

Human oversight is necessary.

Human approval everywhere is not.

The distinction becomes critical as machine execution scales.

Human approval is often introduced as the default response to uncertainty. An agent can perform the work, but a person must approve the result before anything happens.

For low-volume systems, this can be sensible.

At scale, it can reproduce the very coordination architecture that machine execution was intended to change.

The machine performs.

The human waits to review.

The machine waits for the human.

The workflow stops.

As machine throughput increases, approval demand increases with it. The human actor becomes a capacity-constrained resource positioned directly inside the execution path.

The result is an approval bottleneck.

This does not mean removing human control. Research from Microsoft and production evidence elsewhere support continued human involvement, particularly where agents operate imperfectly or actions carry meaningful consequences.

The design question is where that involvement belongs.

Not every action requires the same control.

Some decisions require direct approval.

Some require review only when defined conditions are met.

Some require retrospective sampling.

Some can operate autonomously inside explicit boundaries.

Some should not be delegated at all.

The operating failure occurs when the enterprise uses human approval as a universal substitute for authority design.

That choice appears safe because every action remains supervised.

Operationally, it can create a system incapable of absorbing its own machine capacity.

Exhibit 12.2

The Approval Paradox

MACHINE CAPACITY INCREASES

More work can be executed More decisions can be prepared More actions can be proposed

APPROVAL ARCHITECTURE REMAINS UNCHANGED

Human review required Human decision rights unchanged Human capacity unchanged

THE QUEUE MOVES

Execution latency falls Approval latency rises

OPERATING RESULT

Task productivity improves Workflow throughput stalls Human coordination becomes the constraint

The intervention is not automatically fewer controls.

It is better placement of control.

04 · Authority Too Narrow

An agent cannot create operating leverage if it must repeatedly ask permission to perform the work it was designed to execute.

This is authority failure in its restrictive form.

The enterprise delegates execution but withholds the decision rights required to complete it.

The agent can inspect but not change.

Recommend but not act.

Prepare but not submit.

Detect but not remediate.

Plan but not execute.

The result is machine capacity trapped behind human authorization.

This produces a characteristic operating signature.

Escalations increase.

Human touches remain high.

Cycle time does not fall in proportion to task automation.

Managers become approval routers.

Employees spend increasing amounts of time validating actions they no longer need to originate themselves.

The enterprise appears to have introduced autonomy while preserving human execution through a different interface.

This is delegated labor without delegated authority.

It creates what looks like human oversight but often functions as machine underutilization.

The appropriate authority boundary depends on consequence, reversibility, uncertainty and the operating environment. A payment agent should not necessarily possess the same discretionary envelope as an internal research agent. An infrastructure remediation agent should not necessarily have unrestricted production access because it can diagnose an incident.

But if every meaningful action requires escalation, the machine is not operating autonomously.

It is generating requests for humans.

At sufficient scale, those requests become work.

05 · Authority Too Broad

The opposite failure is more dangerous.

Machine execution can propagate decisions much faster than traditional human structures.

An agent with excessive authority can therefore convert a local error into a system event.

The issue is not simply that an agent may make a bad decision.

Humans make bad decisions.

The operating distinction is the combination of speed, replication and access.

A machine actor may be able to execute the same decision hundreds or thousands of times. It may interact with multiple systems. It may initiate downstream workflows. Other agents may consume its output and continue execution.

A mistake can therefore acquire reach.

This makes delegated authority inseparable from containment.

Authority must have boundaries.

- Financial limits.
- System boundaries.
- Action classes.
- Data permissions.
- Escalation conditions.
- Reversibility requirements.
- Time constraints.
- Confidence or uncertainty conditions where appropriate.

The objective is not minimal autonomy.

It is bounded autonomy.

Enough authority to complete meaningful work.

Enough constraint to prevent a single actor from creating disproportionate enterprise consequences.

The operating model fails when either side dominates.

Too little authority and the enterprise creates queues.

Too much authority and it creates uncontrolled propagation.

The design problem is not whether machines should possess authority.

It is how much authority each machine actor requires to produce the outcome safely and efficiently.

06 · Exceptions Become the Work

Automation changes the composition of human work.

As routine execution moves to machines, the residual work left for humans becomes less representative of the original process.

It becomes concentrated in exceptions.

Ambiguous cases.

Conflicting evidence.

Policy boundaries.

Novel situations.

High-consequence decisions.

Cases where context is missing.

Cases where the machine cannot determine what should happen next.

This is expected.

The failure begins when exception volume exceeds the human system designed to absorb it.

A workflow can therefore automate a high percentage of its transactions and still become operationally worse.

Suppose an agent handles 90 percent of cases autonomously.

That sounds successful.

But if the remaining 10 percent previously represented 2 percent of specialist workload and now arrives continuously, poorly classified and with insufficient context, the exception function may collapse under the new demand.

The percentage automated is not the relevant measure.

The relevant measure is whether the complete operating system can absorb the residual work.

Exception rate.

Exception complexity.

Time to resolution.

Repeat exception categories.

Human capacity required per exception.

The amount of context transferred with the escalation.

These become operating measures.

Repeated exceptions also carry diagnostic information.

An exception that occurs once may be noise.

An exception that occurs repeatedly may indicate that the operating model has defined the normal path incorrectly.

At that point, the enterprise should not simply add people to the exception queue.

It should ask whether the workflow, authority boundary, context architecture or policy needs to change.

Exceptions are not merely work to be processed.

They are evidence about where the operating model has not yet learned.

07 · Automation Can Create Work

The productivity narrative around AI often assumes a simple substitution.

Human work decreases because machine work increases.

Real operating systems are less tidy.

Automation can remove work in one location while creating it somewhere else.

Generated output must be checked.

Exceptions must be resolved.

Agent behavior must be observed.

Permissions must be administered.

Failed actions must be repaired.

Context must be maintained.

Evaluations must be updated.

Machine-created artifacts must be reconciled with systems of record.

Downstream teams must absorb increased throughput.

Security and compliance teams may inherit new control requirements.

Managers may need to supervise both the original process and the new machine system during transition.

The gross amount of automated work therefore says little about net operating leverage.

This is why economic measurement must follow the workflow rather than the feature.

The correct question is not:

How much work did the agent perform?

It is:

What work disappeared, what work moved and what new work was created because the agent exists?

Only then can the enterprise determine whether automation actually reduced the resources required to produce the outcome.

This becomes particularly important as agent systems grow more sophisticated.

Machine execution is not free simply because human labor is absent.

It consumes compute.

It consumes tokens.

It consumes tools.

It creates traces.

It requires evaluation.

It creates operational dependencies.

It creates governance work.

It can generate human exceptions.

Every machine actor enters the enterprise with an operating footprint.

That footprint must be justified by the value it creates.

08 · The Management Double System

One of the most persistent transition failures is organizational rather than technical.

The enterprise introduces machine execution but preserves the management system designed for human execution.

The agent performs the work.

The employee monitors the agent.

The manager still reviews the employee.

The old status meeting continues.

The old report remains.

The old dashboard remains.

The old coordination ritual remains.

The new system is added without removing the old one.

The enterprise now operates two management systems.

One directs execution.

The other describes, checks or recreates it.

This is the management double system.

It is easy to create because removing management mechanisms feels riskier than adding technology.

During transition, duplication may be necessary.

The enterprise may need parallel controls while confidence develops.

But temporary redundancy can harden into permanent structure.

When it does, the organization pays twice.

It pays for machine execution and for human coordination designed around the assumption that machines do not execute.

The operating model has changed underneath the management model.

Management has not followed.

This produces a deeper diagnostic question:

Which management activities still exist because they improve the outcome, and which exist because the previous operating model required them?

The answer cannot be determined from the organizational chart.

It must be determined from the workflow.

09 · Agent Sprawl

As agents become easier to create, the enterprise acquires a new form of capacity problem.

Too many machine actors.

An agent for research.

An agent for planning.

An agent for validation.

An agent for reporting.

An agent for escalation.

A coordinator agent.

A supervisor agent.

Specialist agents beneath it.

Soon the architecture contains a population of machine actors whose existence is easier to explain individually than collectively.

More agents do not necessarily produce a better operating system.

Anthropic's experience building its multi-agent research system provides unusually concrete evidence. Multi-agent architectures improved performance for tasks that benefited from parallel exploration, but they carried substantial cost. Anthropic reported that agents used roughly four times the tokens of ordinary chat interactions and multi-agent systems roughly fifteen times as many. The company also found that some tasks were poor candidates for multi-agent architecture, particularly when agents required shared context or had tightly coupled dependencies.

Its early systems exposed another failure mode. Agents spawned excessive numbers of subagents, duplicated searches and continued working after sufficient evidence had already been collected. Anthropic introduced explicit scaling rules to align agent effort with task complexity.

The lesson extends beyond research systems.

Agent count is an architectural decision, not a maturity metric.

The enterprise should use multiple agents when decomposition creates a meaningful advantage.

Parallel execution.

Specialized context.

Distinct authority.

Different tools.

Different models.

Independent verification.

Isolation of responsibilities.

If those conditions do not exist, additional agents may simply create coordination overhead.

Every additional machine actor introduces another context boundary, another execution path, another source of latency, another set of permissions, another evaluation surface and another possible failure relationship.

Agent sprawl is therefore not merely a technical cleanliness problem.

It is operating complexity.

And complexity must earn its place.

Exhibit 12.3

When Machine Capacity Becomes Operating Friction

CAPABILITY ADDED

Automation Agent Subagent Approval Control Evaluation Tool Management layer

LOCAL PURPOSE

Increase throughput Increase specialization Increase control Increase reliability Increase visibility

SYSTEM EFFECT

More coordination More context boundaries More execution paths More permissions More evaluation surfaces More compute consumption More failure relationships

OPERATING FRICTION

Latency increases Exceptions increase Human intervention increases Machine coordination increases Cost increases Failure becomes harder to locate

INTERVENTION QUESTION

Does this additional machine capacity improve the enterprise outcome enough to justify the operating complexity it creates?

Every additional component must earn its place in the operating model.

Complexity is justified when it produces greater capability, resilience, control or economic value.

When it does not, simplification is itself an intervention.

10 · From Failure Detection to Operating Intervention

Finding failure is not enough.

An enterprise can instrument every agent, trace every execution path and record every exception while remaining incapable of improving the operating model.

Observation produces evidence.

Intervention changes the system.

The distinction matters because many failures that appear technical are symptoms of operating design decisions made elsewhere.

An agent repeatedly requesting human approval may not have an agent problem.

It may have an authority problem.

A workflow generating thousands of exceptions may not have an exception handling problem.

Its allocation of work may be wrong.

A multi-agent system producing inconsistent outcomes may not require a better orchestration framework.

It may contain unnecessary machine actors.

An automation producing impressive task-level performance while increasing downstream workload may not have failed technically at all.

The workflow has failed economically.

The purpose of intervention is therefore not to repair every visible defect.

It is to identify the structural condition producing the defect and change the smallest part of the operating model capable of correcting it.

That requires a different operating discipline.

11 · The Intervention Point

Every failure has a location where it becomes visible.

That location is not necessarily where the intervention belongs.

Consider a customer onboarding workflow.

An agent prepares an account configuration. A human must approve it. Approval queues begin to grow.

The visible failure occurs at the approval stage.

The enterprise could add reviewers.

That would increase capacity at the point where the failure appears.

But the underlying problem may be that every configuration requires approval regardless of risk.

The intervention point may therefore sit upstream, in the authority architecture.

Low-risk configurations could execute automatically.

Medium-risk configurations could execute within bounded thresholds.

Only high-risk configurations might require explicit human authorization.

The queue was real.

The queue was not the problem.

The operating rule that created the queue was the problem.

This distinction separates operational remediation from operating-model intervention.

Remediation restores execution.

Intervention changes the conditions under which execution occurs.

Both are necessary.

They solve different problems.

Exhibit 12.4

Failure Location ≠ Intervention Point

VISIBLE FAILURE
Where does the problem appear?

EXECUTION PATH
What sequence of human and machine activity produced it?

STRUCTURAL CAUSE
Which operating assumption created the condition?

INTERVENTION POINT
What part of the operating model must change?

RECONFIGURED EXECUTION
Does the altered system produce a better enterprise outcome?

The intervention point may exist in:

OUTCOME
The system is optimizing the wrong result.

WORKFLOW
The sequence of work is poorly designed.

ALLOCATION
Work has been assigned to the wrong human or machine actor.

AUTHORITY

Decision rights are too broad, too narrow or positioned incorrectly.

MANAGEMENT

Humans are coordinating work that the execution system should manage, or machines are operating without adequate supervisory mechanisms.

ORGANIZATION

Functional boundaries prevent end-to-end execution.

EXECUTION ARCHITECTURE

The technical system cannot reliably support the operating design.

AGENT ARCHITECTURE

Too many agents, inappropriate specialization or weak coordination creates unnecessary complexity.

CONTEXT

The machine actor does not possess the information required to act correctly.

CONTROL

Governance mechanisms create too much friction or insufficient constraint.

INFRASTRUCTURE

Capacity, latency or availability prevents reliable execution.

ECONOMICS

The cost of execution exceeds the value produced.

The operating model therefore becomes diagnosable.

Failure can be traced backwards through its architecture.

12 · Failure Propagates

Traditional automation often failed locally.

A script stopped.

A service returned an error.

A workflow halted.

Agentic systems can fail differently.

Machine actors increasingly interpret information, select tools, call other systems, delegate work and create outputs consumed by other machine actors.

An incorrect action can therefore become an input into subsequent execution.

Failure propagates.

A context error may produce a poor decision.

That decision may invoke the wrong tool.

The resulting state may be accepted by another agent.

A downstream workflow may then execute against that state.

By the time a human encounters the consequence, the original failure may be several execution steps removed.

Production research is already exposing the reliability problem.

A 2026 study of 306 practitioners across 26 domains found reliability to be the leading development challenge for production agents. The same research found that production systems tend to remain deliberately bounded. Sixty-eight percent of surveyed implementations executed ten steps or fewer before human intervention. Most also relied heavily on human evaluation rather than fully automated assessment.

The evidence does not suggest that useful agents cannot operate in production.

It suggests that successful production systems currently depend on control, bounded execution and systems-level reliability engineering.

As machine execution becomes more interconnected, the enterprise must therefore observe not only whether individual actors succeed.

It must understand how state and failure move through the execution system.

13 · The Management Double System

One of the most important operating failures will not appear in an agent trace.

It appears in the organization.

The enterprise redesigns execution but preserves the management system built for the previous form of work.

Machines begin preparing analysis, routing cases, resolving routine exceptions, producing recommendations and executing bounded decisions.

Yet managers continue to request status reports.

Teams continue to prepare coordination meetings.

Employees continue to translate system state into presentations.

Approval chains remain intact.

Management layers continue to perform information aggregation that the execution architecture can increasingly perform directly.

A second system forms.

The machine execution system performs the work.

The inherited management system continues coordinating the work as though machine execution did not exist.

This is the management double system.

Its cost can be substantial.

The enterprise pays for machine execution while retaining the human coordination architecture that automation was capable of reducing.

The result may appear organizationally stable because familiar management routines remain intact.

Economically, however, part of the operating leverage has disappeared.

The appropriate intervention is not necessarily another automation.

It may be the removal of obsolete coordination.

AI Native transformation therefore requires enterprises to inspect the management system whenever execution changes materially.

The question is not only:

What new work can machines perform?

It is also:

What human coordination is no longer necessary because machines now perform it?

14 · Local Success Can Produce Enterprise Failure

AI systems are often deployed through local organizational ownership.

A function identifies a use case.

A team builds an agent.

The agent improves a metric.

The deployment is declared successful.

But enterprises are systems.

A local improvement can move cost, latency or work elsewhere.

A sales agent may increase qualified opportunities while overwhelming underwriting.

An engineering agent may increase code production while increasing review burden.

A service agent may shorten interaction time while increasing escalations.

A finance automation may accelerate processing while producing additional reconciliation work downstream.

A procurement agent may optimize unit price while degrading supplier resilience.

The local metric improves.

The enterprise outcome deteriorates.

This is not an argument against local experimentation.

It is an argument for measuring the complete workflow.

Chapter 11 established that economic value must ultimately be measured at the level where enterprise economics change.

Chapter 12 adds the corresponding intervention principle:

When local performance improves but system performance deteriorates, the operating model must be examined across the workflow boundary.

Optimization must follow the outcome.

Not the organizational chart.

15 · Evaluation Theatre

A system can also appear well governed while remaining poorly understood.

Evaluation suites grow.

Benchmarks multiply.

Dashboards turn green.

The enterprise develops confidence because the agent has passed its tests.

Production reality can still diverge.

Research published in 2026 illustrates the gap. In a survey of 157 enterprise respondents, half reported that an agent or large language model feature had passed internal evaluation and subsequently produced a customer-facing failure. Only 5 percent said they fully trusted automated evaluation. The sample was self-selected and should therefore be treated as directional rather than representative of all enterprises.

The operating implication is still important.

An evaluation is evidence about system behavior under evaluated conditions.

It is not proof that the operating model will work under every production condition.

Evaluation theatre begins when passing tests becomes a substitute for observing outcomes.

The enterprise therefore needs several layers of evidence.

Pre-deployment evaluation.

Runtime observation.

Workflow performance.

Exception behavior.

Human escalation.

Economic performance.

Outcome realization.

These layers answer different questions.

A model can perform well while the workflow performs badly.

An agent can complete its task while the enterprise loses money.

A system can remain within policy while human workload increases.

A workflow can become faster while customer outcomes deteriorate.

Evaluation must therefore extend beyond machine correctness.

It must test operating performance.

16 · Economics Drift

Even a well-designed AI Native workflow does not remain economically correct forever.

Model prices change.

Token consumption changes.

Context windows expand.

Agent paths become longer.

Retry behavior changes.

Tool calls accumulate.

Infrastructure demand grows.

Human escalation rates move.

Volumes change.

Higher-capability models become available.

Cheaper models become sufficient for tasks previously assigned to expensive ones.

The economics of the operating model drift.

Datadog's 2026 analysis of production large language model traces provides one example of how technical conditions can become economic and operational conditions. Its analysis found millions of rate-limit errors across observed workloads and identified capacity constraints as a material source of failed model calls.

The important point is broader than rate limits.

Machine capacity is now a productive resource.

Its consumption must be managed.

An AI Native enterprise therefore cannot calculate the economics of a workflow once and treat the result as permanent.

It needs continuous visibility into:

- cost per outcome
- cost per workflow execution
- cost by machine actor
- cost by model
- cost of retries
- cost of exceptions
- human intervention cost
- infrastructure consumption
- latency
- failure cost
- and the economic value produced.

A workflow whose economics drift beyond its value threshold requires intervention.

The response may be architectural.

Use a smaller model.

Reduce context.

Remove an agent.

Change routing.

Cache repeated work.

Alter the human-machine allocation.

Change the workflow.

Constrain execution.

Or stop the automation.

Economic discipline is part of operating-model discipline.

17 · The Operating Intervention System

The enterprise now needs a capability that connects all of these failure classes.

Not another monitoring platform.

Not another governance committee.

Not another transformation office.

An operating capability.

Its purpose is to continuously determine whether the operating model is producing the intended outcome and to change the system when it is not.

Exhibit 12.5

The Operating Intervention System

OBSERVE

Capture execution state across human work, machine work, workflow performance, authority use, exceptions, infrastructure and economics.

DETECT VARIANCE

Identify where actual operating behavior diverges from expected outcome, performance, risk or economic thresholds.

DIAGNOSE STRUCTURAL CAUSE

Determine whether the variance originates in workflow, allocation, authority, management, organization, architecture, context, control, infrastructure or economics.

LOCATE THE INTERVENTION POINT

Find the smallest structural change capable of altering the system outcome.

CHANGE THE OPERATING MODEL

Modify the relevant rule, workflow, allocation, authority boundary, architecture, control or resource configuration.

EVALUATE THE INTERVENTION

Determine whether the change improved the enterprise outcome without creating unacceptable downstream effects.

CONTINUE · CONSTRAIN · ROLLBACK · SCALE

Decide whether the new configuration should remain, be bounded, reversed or expanded.

LEARN

Convert the result into operating knowledge that informs future design.

This is the missing loop.

Without it, AI Native transformation remains a sequence of deployments.

With it, the operating model begins to become adaptive.

18 · Intervention Must Have Authority

A diagnostic capability without the authority to change the system becomes another reporting function.

This creates an important organizational question.

Who can intervene?

The answer will vary by enterprise and by risk.

Some interventions can be automated.

A system may route execution to another model when cost or latency thresholds are exceeded.

It may reduce agent authority when confidence falls.

It may trigger human review when unusual conditions appear.

It may stop execution when a control boundary is crossed.

Other interventions require human judgment.

Changing decision rights.

Redesigning a workflow.

Removing an approval layer.

Changing organizational ownership.

Accepting a new risk threshold.

Changing customer policy.

Reallocating capital.

These decisions alter the operating model itself.

They require accountable human authority.

The intervention system therefore follows the same principle developed throughout this publication.

Machine intelligence can expand observation, diagnosis and execution.

Human judgment remains essential where the enterprise must determine what trade-off it is willing to make.

19 · Not Every Failure Requires Less Autonomy

There is a predictable response when autonomous systems fail.

Add approval.

Add review.

Reduce authority.

Require human confirmation.

Sometimes that is correct.

Sometimes it recreates the bottleneck that autonomy was intended to remove.

Research on production agents suggests why the balance matters. Current deployments frequently remain bounded and rely heavily on human evaluation. At the same time, research on human oversight architectures has identified the throughput cost of placing humans inside every consequential execution path.

The appropriate response to failure is therefore not automatically less autonomy.

It is better calibrated autonomy.

Authority may need to narrow.

It may also need to expand.

A machine actor repeatedly escalating routine cases may require more authority, not less.

A human approval queue may indicate excessive control.

An exception flood may indicate that the agent has been given responsibility without sufficient decision rights.

Intervention must diagnose before it constrains.

Accountable autonomy is not maximum autonomy.

It is the amount of autonomy the operating system can support for a given outcome, under defined conditions, with observable performance and recoverable failure.

20 · Failure Becomes Operating Evidence

This changes the role of failure in the AI Native enterprise.

Failure is not merely something to eliminate.

It is information about the operating design.

An approval queue reveals something about authority.

An exception flood reveals something about workflow design.

Agent sprawl reveals something about architecture.

Escalation patterns reveal something about allocation.

Repeated context failure reveals something about information architecture.

Unexpected compute consumption reveals something about execution economics.

Management overhead reveals something about organizational design.

Local optimization reveals something about outcome architecture.

The enterprise becomes stronger when it can convert those signals into structural change.

The objective is not a failure-free operating model.

No operating model achieves that.

The objective is an operating model capable of detecting failure early, understanding what the failure means and changing itself before the consequences compound.

21 · From Intervention to Transition

The argument of this publication has now moved through the operating system.

Start with the outcome.

Redesign the workflow.

Allocate work between humans and machines.

Define authority.

Redesign management.

Change the organization.

Build the execution architecture.

Govern heterogeneous machine actors.

Manage infrastructure and compute as productive resources.

Measure economic value.

Observe failure.

Intervene.

A final problem remains.

Most enterprises do not begin here.

They begin with existing workflows.

Existing organizational structures.

Existing technology estates.

Existing management systems.

Existing controls.

Existing economics.

Existing people.

Existing commitments.

The AI Native operating model cannot simply be placed on top of that enterprise.

The enterprise must move from one operating system to another while continuing to operate.

That transition is the subject of Chapter 13.

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/12-failure-modes-and-intervention

Part V

The AI Native Enterprise

Chapter 13

Becoming an AI Native Enterprise

How an enterprise moves from today's operating model toward an AI Native one.

> *AI adoption adds AI to the operating model. AI Native transformation changes the operating model because AI exists.*

An AI Native enterprise cannot be installed.

It has to be built while the existing enterprise continues to operate.

That distinction defines the transformation problem.

Most established enterprises already contain decades of accumulated operating logic.

Workflows cross functions.

Decision rights reflect historical organizational boundaries.

Management systems compensate for fragmented information.

Technology estates contain multiple generations of architecture.

Controls have accumulated around previous failures.

People have built careers around existing roles.

Budgets follow organizational structures.

Performance systems reward established behavior.

AI enters this environment.

The easiest response is addition.

Add copilots.

Add agents.

Add models.

Add automation.

Add an AI platform.

Add an AI governance function.

Add AI objectives to existing transformation programs.

This can produce useful capability.

It does not necessarily produce an AI Native enterprise.

AI Native transformation begins when the enterprise changes the system through which outcomes are produced.

The distinction is fundamental.

AI adoption adds AI to the operating model.

AI Native transformation changes the operating model because AI exists.

That means changing work.

Changing allocation.

Changing authority.

Changing management.

Changing organizational boundaries.

Changing execution architecture.

Changing the consumption of machine capacity.

Changing the economics of execution.

And creating the intervention capability required to keep changing those things as the system learns.

The transition cannot therefore be organized primarily around technology deployment.

It must be organized around operating-model conversion.

01 · The Transition Problem

The evidence increasingly reflects this distinction.

Enterprise access to AI is expanding quickly. Deloitte's 2026 enterprise research reports that worker access to AI increased by 50 percent during 2025, while organizations expect a substantial increase in the proportion of AI projects reaching production. Enterprises are already deploying autonomous systems across customer support, supply chains, research and development, knowledge work and other functions.

Deployment, however, is not the same as operating-model transformation.

McKinsey's 2026 research reports that only 21 percent of surveyed companies had fundamentally redesigned their operating models around AI. Organizations attributing at least 5 percent of EBIT to AI were three times more likely to pursue broad operating-model redesign and twice as likely to redesign workflows before selecting AI tools.

The evidence does not establish one universal transformation method.

It does support an important distinction.

Installing AI capability and redesigning the enterprise are different activities.

Market Evidence

AI adoption is moving faster than enterprise redesign.

Sentient Interpretation

The principal transformation gap is no longer simply access to capable models.

It is the enterprise's ability to reorganize work around them.

Operating Implication

The unit of transformation should not be the AI use case.

It should be the enterprise workflow that produces an economically meaningful outcome.

That changes where transformation begins.

02 · Start With an Outcome

Enterprises often begin AI transformation by asking:

Where can we use AI?

The question creates a technology inventory.

Customer service.

Software development.

Finance.

Human resources.

Procurement.

Marketing.

Legal.

Operations.

Each function generates use cases.

Those use cases become pilots.

Pilots become portfolios.

The enterprise accumulates AI activity.

But activity is not architecture.

An AI Native transition begins with a different question:

Which enterprise outcome should materially improve?

Revenue conversion.

Claims resolution.

Customer retention.

Product release.

Order fulfillment.

Working capital.

Service restoration.

Fraud resolution.

Supplier onboarding.

Software delivery.

The outcome establishes the economic reason for redesign.

The enterprise can then work backwards.

What workflow produces that outcome?

Where does work wait?

Where does judgment occur?

Where is information reconstructed?

Where do handoffs happen?

Where are humans performing machine-suitable work?

Where are machines being prevented from acting because authority remains unresolved?

Where does coordination consume capacity?

Where does execution fail?

Where does cost accumulate?

Only then does AI become relevant.

Exhibit 13.1

The Transition Entry Point

ENTERPRISE OUTCOME

What materially important result should change?

ECONOMIC BASELINE

What does producing that outcome cost today?

WORKFLOW

What end-to-end system currently produces it?

CONSTRAINT

Where does the existing operating model limit performance?

REDESIGN POTENTIAL

What becomes possible when human judgment and machine intelligence are allocated differently?

TRANSITION CANDIDATE

Is this workflow valuable, feasible and governable enough to become an AI Native transformation domain?

This sequence prevents the enterprise from confusing technical possibility with transformation priority.

03 · Select the Right Workflow

Not every workflow should move first.

The highest-volume process is not automatically the best candidate.

Neither is the process with the largest number of manual tasks.

Nor the function with the most enthusiastic executive sponsor.

A useful transition workflow has several characteristics.

- The outcome matters economically.
- The workflow can be bounded.
- Its current performance can be measured.
- There is enough execution volume to observe the redesigned system.
- Machine intelligence can materially change how the work is performed.
- Human and machine responsibilities can be made explicit.
- Authority can be defined.
- Required data can be made accessible.
- Failure can be detected.
- Intervention remains possible.
- The economic result can be measured.

This creates a practical principle:

Begin where the enterprise can learn about the operating model, not merely demonstrate the technology.

A successful first workflow should teach the enterprise how to redesign work.

How to allocate judgment.

How to delegate authority.

How to govern agents.

How to measure machine execution.

How to intervene.

How to calculate economics.

How to change human roles.

How to scale what works.

The first transformation domain is therefore also a learning system.

04 · Baseline Before Redesign

Transformation without a baseline creates weak evidence.

If the enterprise cannot describe how the workflow performs before intervention, it will struggle to determine what changed afterward.

The baseline should not be limited to labor cost.

It should capture the economics and behavior of the complete workflow.

Cycle time.

Human effort.

Queue time.

Handoffs.

Approval latency.

Exception rates.

Rework.

Error rates.

Escalations.

Customer impact.

Revenue effect.

Operational loss.

Infrastructure cost.

Management overhead.

Control overhead.

Where possible, the enterprise should establish the cost per completed outcome.

This becomes the economic reference point.

The purpose is not accounting precision.

It is decision quality.

The enterprise needs enough evidence to determine whether redesign creates operating leverage or merely moves cost elsewhere.

05 · Redesign Before Automation

The workflow should then be rebuilt from the outcome backwards.

Do not begin by asking which existing tasks can be automated.

That preserves the inherited workflow.

Ask instead:

If this outcome had to be produced today using the available combination of human judgment, machine intelligence, software systems and compute, how should the work be designed?

Some existing steps remain.

Some disappear.

Some combine.

Some become machine executed.

Some move to humans.

Some decisions become automatic.

Some require explicit human judgment.

Some approvals become policy boundaries.

Some handoffs disappear entirely.

This is where AI Native transformation separates from conventional automation.

The enterprise is not automating the workflow it inherited.

It is determining whether that workflow should continue to exist in its current form.

Recent evidence supports this emphasis. McKinsey's 2026 research on software and product development found organizations that redesigned processes before incorporating AI were more than twice as likely to report productivity gains above 20 percent as organizations that layered AI onto existing ways of working.

Its broader enterprise research similarly argues that many organizations continue to accelerate existing activities without removing the organizational constraints surrounding them.

What This Supports

Workflow redesign appears to be materially associated with stronger AI outcomes.

What Remains Unresolved

The evidence does not prove that every workflow requires radical redesign or that a particular redesign method will work across industries.

The enterprise must determine the appropriate degree of redesign through evidence.

06 · Allocate Work Again

Once the workflow has been reconstructed, work can be allocated.

Chapter 4 established the principle.

Human versus machine is the wrong binary.

The enterprise needs a portfolio of execution modes.

Human execution.

Human execution with machine assistance.

Machine execution with human review.

Machine execution with exception escalation.

Machine execution within bounded authority.

Machine execution with supervisory observation.

Fully automated execution where the conditions justify it.

The allocation should reflect the nature of the work.

Machine actors may possess advantages in speed, scale, pattern processing, persistence, retrieval and repeated execution.

Humans may remain essential where work requires accountability, ambiguous trade-offs, social judgment, ethical interpretation, novel strategic reasoning or decisions whose consequences require explicit human ownership.

The allocation is not permanent.

It is an operating hypothesis.

The intervention system established in Chapter 12 must be capable of changing it.

07 · Authority Follows Allocation

Assigning work to a machine actor without defining what it may decide creates pseudo-autonomy.

The system performs analysis.

Then waits.

It prepares an action.

Then asks permission.

It identifies a resolution.

Then enters a human queue.

The enterprise has transferred execution without transferring sufficient authority.

The opposite failure is equally dangerous.

A machine actor receives broad system access and vague instructions to pursue an outcome.

Execution becomes possible.

Accountability becomes unclear.

The transition therefore requires an authority architecture for every redesigned workflow.

For each consequential machine action:

What may the system observe?

What may it recommend?

What may it decide?

What may it execute?

Within what threshold?

Against which policy?

Using which systems?

For how long?

Under what conditions must it escalate?

Who can revoke its authority?

Who remains accountable for the outcome?

Authority becomes part of workflow design.

Not a security configuration added afterward.

Exhibit 13.2

The Workflow Redesign Sequence

1 · DEFINE THE OUTCOME

Establish the result and economic baseline.

2 · MAP THE CURRENT WORKFLOW

Expose work, waits, decisions, handoffs, controls and coordination.

3 · REMOVE

Eliminate work that no longer needs to exist.

4 · REALLOCATE

Assign remaining work across humans and machines.

5 · DELEGATE

Define decision rights and machine authority.

6 · REDESIGN MANAGEMENT

Determine how mixed execution will be directed, observed and adapted.

7 · REDESIGN ORGANIZATION

Change ownership and boundaries where they obstruct the workflow.

8 · BUILD EXECUTION ARCHITECTURE

Provide agents, tools, context, identity, policy, observability and infrastructure.

9 · ESTABLISH ECONOMICS

Measure cost and value at the workflow and outcome level.

10 · OPERATE AND INTERVENE

Observe production behavior and change the operating model when evidence requires it.

This is the basic unit of AI Native transition.

08 · From Workflow Redesign to Enterprise Transition

A redesigned workflow is not yet an AI Native enterprise.

It is evidence that one part of the enterprise can operate differently.

The next problem is scale.

Scale creates a predictable temptation.

Replicate the solution.

Take the successful agent.

Take the platform.

Take the architecture.

Take the workflow pattern.

Deploy it everywhere.

That is technology scaling.

Operating-model scaling is different.

The enterprise must scale the capability to redesign.

Different workflows contain different judgments.

Different authority boundaries.

Different risk.

Different economics.

Different human roles.

Different data.

Different regulatory constraints.

Different failure modes.

The objective is therefore not to manufacture identical AI workflows.

It is to establish a repeatable enterprise system for converting suitable workflows into AI Native workflows.

09 · Build the Transition Capability

The enterprise needs several capabilities to make this repeatable.

Outcome architecture

The ability to identify economically meaningful outcomes and connect them to workflows.

Workflow redesign

The ability to reconstruct work rather than automate inherited tasks.

Human-machine allocation

The ability to determine the appropriate execution mode for each part of the workflow.

Authority engineering

The ability to translate business accountability into machine decision rights.

Management redesign

The ability to manage mixed human and machine execution.

Organizational redesign

The ability to alter boundaries when the workflow requires it.

Agent engineering

The ability to create reliable machine actors.

Context engineering

The ability to provide those actors with relevant, authoritative information.

Control-plane engineering

The ability to govern identity, access, policy, routing, budgets and execution.

Infrastructure economics

The ability to manage compute and model consumption as productive resources.

Value realization

The ability to connect operating change to economic results.

Intervention

The ability to detect failure and change the operating model.

These capabilities form the transformation machinery.

They should become reusable even when individual implementations differ.

10 · The Agent Control Plane Becomes Enterprise Infrastructure

As workflows multiply, isolated agent architectures become difficult to govern.

Each team can select models.

Create tool connections.

Define prompts.

Store context.

Build orchestration.

Create evaluation.

Implement access control.

Set budgets.

Build observability.

That freedom accelerates experimentation.

At scale, it can create fragmentation.

The enterprise needs common execution services.

Identity.

Authentication.

Authorization.

Policy enforcement.

Model routing.

Tool registration.

Agent registration.

Secrets management.

Context access.

Runtime controls.

Budget enforcement.

Rate limits.

Tracing.

Evaluation.

Incident response.

Kill mechanisms.

Version management.

Cost attribution.

These capabilities form the agent control plane described earlier in this publication.

The control plane should not dictate every implementation.

Its purpose is to make heterogeneous machine execution governable.

IBM's 2026 study of 306 production-agent practitioners provides an important reality check. Production systems remain deliberately controlled. Sixty-eight percent of surveyed agents execute no more than ten steps before human intervention, while reliability remains the leading development challenge.

Operating Implication

The enterprise should scale autonomy as its ability to observe, govern and recover from machine execution improves.

Capability should precede unrestricted autonomy.

11 · Build Shared Foundations Where Reuse Is Real

Some parts of the transition should be standardized.

Others should remain local.

The distinction matters.

Shared enterprise capabilities are valuable where multiple workflows require the same underlying service.

Identity.

Policy.

Model access.

Observability.

Evaluation infrastructure.

Security controls.

Approved tool interfaces.

Context retrieval.

Compute management.

Cost accounting.

Reusable agent patterns.

These reduce duplicated engineering and make execution easier to govern.

But centralization can become another constraint.

A central AI team should not become the human approval queue of the transformation itself.

Workflow expertise remains distributed through the enterprise.

The people closest to an outcome often understand its exceptions, incentives and operational reality better than a central technology organization.

The transition architecture therefore requires a federation.

Shared execution foundations.

Distributed workflow redesign.

Common operating principles.

Local outcome accountability.

12 · Change Management Starts During Design

Traditional technology transformation often treats adoption as a late-stage problem.

Build the system.

Deploy the system.

Train the users.

Drive adoption.

AI Native transformation makes this sequence insufficient.

The people affected by the system are not merely learning a new tool.

Their work may change.

Their authority may change.

Their role may change.

Their performance measures may change.

Their team may change.

Their management relationship may change.

Some work may disappear.

New work may emerge.

Human participation therefore belongs inside workflow design.

Recent field evidence supports this. McKinsey reports that programs combining change design with technical delivery from the beginning have observed materially higher adoption six months after deployment than programs treating change as a go-live activity. These figures derive from McKinsey's transformation experience rather than controlled experimental evidence and should be interpreted accordingly.

The deeper operating principle does not depend on the exact percentage.

People cannot meaningfully adapt to a new operating model if they encounter it only after it has been designed.

13 · Management Must Transition Too

The manager is often the missing subject of AI transformation.

Enterprises redesign employee work.

They automate tasks.

They introduce agents.

They change workflows.

But the management system remains substantially untouched.

Chapter 6 established why this becomes unstable.

As machine execution expands, managers may perform less task coordination and more system direction.

They may define outcomes.

Set constraints.

Allocate human and machine capacity.

Resolve exceptions.

Change authority.

Interpret system performance.

Intervene when the operating model fails.

Develop human capability.

Make trade-offs machines should not make.

This changes managerial work.

The enterprise must therefore ask during every transition:

Which management activities remain necessary?

Which become machine executable?

Which become visible directly through the execution system?

Which new management responsibilities appear?

Which layers exist primarily because information previously moved through people?

The objective is not automatically fewer managers.

It is a management architecture appropriate to the new execution system.

14 · Organizational Boundaries Must Follow the Work

A redesigned workflow may expose organizational problems that technology cannot solve.

The workflow crosses five functions.

Each owns one step.

Each has different metrics.

Each controls different systems.

Each optimizes its own performance.

No one owns the end-to-end outcome.

An agent cannot repair that accountability structure.

It may make the fragmentation more visible.

The enterprise must sometimes redesign ownership itself.

Cross-functional outcome teams.

Persistent workflow ownership.

Shared economic metrics.

New decision rights.

Different spans of accountability.

Machine capacity embedded into teams.

Central capabilities moved into shared infrastructure.

The organization should change where its existing boundaries prevent the redesigned workflow from operating coherently.

This does not mean reorganizing the company around every agent.

It means refusing to preserve an organizational boundary simply because it predates the new execution architecture.

15 · Scale in Waves

Enterprise transformation should not proceed as thousands of unrelated AI use cases.

Nor should the entire enterprise be redesigned simultaneously.

Both approaches create fragmentation.

The transition should move in waves.

A wave contains a deliberate portfolio of workflows selected for economic value, learning potential, feasibility and shared capability.

The first wave establishes operating patterns.

The second reuses what proved useful.

Later waves expand across related domains.

Each wave should increase both economic value and institutional capability.

This produces compounding transformation.

The enterprise becomes better at transforming as transformation proceeds.

Exhibit 13.3

The Enterprise Transition System

OUTCOME PORTFOLIO

Select economically meaningful outcomes where operating-model redesign can create material value.

TRANSFORMATION WAVE

Choose a bounded portfolio of workflows with sufficient value, feasibility and learning potential.

REDESIGN

Reconstruct workflows around human judgment and machine intelligence.

ENABLE

Provide shared execution architecture, agent control, context, infrastructure and governance.

OPERATE

Run redesigned workflows in production with explicit authority, accountability and economic measures.

INTERVENE

Observe failure, diagnose structural causes and change the operating model.

PROVE

Determine whether the redesigned workflows improve enterprise outcomes and economics.

REUSE

Convert successful architecture, controls, operating patterns and institutional knowledge into reusable capability.

NEXT WAVE

Apply the stronger enterprise capability to the next portfolio of outcomes.

The transition is therefore recursive.

Every wave produces two forms of value.

The first is economic.

A workflow becomes faster, cheaper, more effective, more resilient or capable of producing an outcome that was previously difficult to achieve.

The second is institutional.

The enterprise becomes better at redesigning itself.

That second form of value is easy to underestimate.

A company that successfully redesigns one workflow has not merely acquired an agent.

It has learned something about how to allocate work.

How to delegate authority.

How to provide context.

How to govern machine execution.

How to redesign management.

How to measure economics.

How to detect failure.

How to intervene.

Those capabilities reduce the difficulty of the next transformation.

The transition compounds when learning is retained.

16 · Scale the Capability, Not the Implementation

This distinction becomes important as early successes attract attention.

A successful workflow creates pressure to copy.

The enterprise sees an agent performing well in one domain and asks where else it can be deployed.

Sometimes reuse is appropriate.

But the most valuable thing to reuse may not be the agent.

It may be the operating capability that produced it.

A customer service workflow and a procurement workflow may share infrastructure without sharing authority.

A software engineering workflow and a finance workflow may use the same model gateway while requiring entirely different evaluation systems.

Multiple agents may use the same identity infrastructure while operating under different permissions.

Different workflows may share observability, cost accounting and policy enforcement while retaining distinct human accountability.

This produces a layered scaling model.

Standardize where common capability creates leverage.

Differentiate where the outcome requires it.

The enterprise should therefore distinguish three things.

Reusable infrastructure

Technical capabilities that should operate consistently across many workflows.

Reusable operating patterns

Proven approaches to allocation, authority, escalation, evaluation and intervention that can be adapted to new domains.

Workflow-specific design

The decisions that must remain grounded in the economics, judgment, risk and operating reality of a particular outcome.

Confusing these layers creates two opposite failures.

Too little standardization produces agent sprawl, duplicated infrastructure and fragmented control.

Too much standardization forces fundamentally different workflows into a common architecture that does not fit them.

AI Native scale requires both coherence and variation.

17 · Sequence by Dependency, Not Enthusiasm

Transformation waves should not be determined only by which business unit is ready.

The order matters.

Some workflows depend on capabilities that do not yet exist.

An enterprise may want highly autonomous agents before it has reliable identity for machine actors.

It may want cross-functional execution before enterprise context can be retrieved safely across systems.

It may want agents to take consequential actions before authority boundaries are defined.

It may want large-scale deployment before cost attribution exists.

It may want distributed experimentation before common observability exists.

These are sequencing problems.

The enterprise should therefore ask of each proposed transformation:

What capability does this workflow require?

Does that capability already exist?

If not, is building it justified by this workflow alone?

Will other workflows reuse it?

What risk appears if execution scales before the capability exists?

This creates a dependency-aware transition.

Some foundations must precede scale.

Others should emerge only when actual workflow demand justifies them.

The enterprise should resist both extremes.

Building an enormous AI platform before meaningful workflows exist creates infrastructure in search of demand.

Scaling workflows without shared foundations creates demand the enterprise cannot govern.

Transition architecture connects the two.

Exhibit 13.4

The Transition Dependency Stack

OUTCOME DEMAND
Which enterprise outcomes justify transformation?

WORKFLOW REQUIREMENTS
What human and machine capabilities are required to produce them differently?

OPERATING REQUIREMENTS
What allocation, authority, management and organizational changes are necessary?

EXECUTION REQUIREMENTS
What agents, models, tools, context and orchestration are required?

CONTROL REQUIREMENTS
What identity, policy, evaluation, observability and intervention mechanisms are necessary?

INFRASTRUCTURE REQUIREMENTS
What compute, data and platform capacity must support execution?

ECONOMIC REQUIREMENTS
What level of cost is justified by the value of the outcome?

Capability should be built against demonstrated operating demand.

Not technological possibility alone.

18 · The Portfolio Needs Capital Discipline

Once AI Native transformation expands beyond individual workflows, the enterprise faces a portfolio allocation problem.

There will always be more possible transformations than available engineering capacity, management attention and capital.

The enterprise therefore needs to decide where additional machine capacity and redesign effort should be deployed.

That decision should not be driven primarily by the number of employees affected or the visibility of the technology.

It should be driven by expected enterprise value.

A transformation portfolio can be evaluated across several dimensions.

Economic importance of the outcome.

Current operating constraint.

Potential improvement.

Technical feasibility.

Authority complexity.

Data and context readiness.

Control requirements.

Organizational dependency.

Infrastructure requirement.

Time to evidence.

Reusability of capabilities created.

Risk of failure.

Learning value.

These dimensions do not need to become a universal scoring formula.

False precision would be counterproductive.

Their purpose is to make trade-offs explicit.

A workflow with modest direct economic value may still deserve early investment if it creates a control capability required by many later workflows.

A high-value workflow may need to wait because the enterprise cannot yet govern the authority it would require.

A technically impressive use case may deserve no investment because the economic outcome is weak.

A transformation portfolio should therefore be managed as an allocation of scarce enterprise capacity.

AI changes the supply of productive machine capacity.

It does not eliminate capital discipline.

19 · Value Must Fund Scale

The economic logic should become increasingly demanding as transformation expands.

Early experiments may reasonably optimize for learning.

Enterprise scale cannot.

A workflow moving into sustained production should have an increasingly explicit value thesis.

What outcome changed?

By how much?

What resources were removed?

What resources were added?

What downstream work changed?

What machine capacity is being consumed?

What human capacity was released?

What new risk or control cost appeared?

What is the resulting economic effect?

The answer may involve revenue.

Cost.

Working capital.

Risk.

Throughput.

Quality.

Capacity.

Customer retention.

Time to market.

Or some combination.

The important requirement is causality.

The enterprise should be able to explain how the operating change creates economic consequence.

This creates a stronger scaling rule:

Do not scale because the agent works.

Scale because the operating model works.

That distinction protects the enterprise from converting successful demonstrations into expensive infrastructure without corresponding value.

20 · Scale, Redesign, Constrain or Stop

Not every transformation should progress.

The intervention system from Chapter 12 becomes part of portfolio governance.

Evidence may support scale.

It may also support redesign.

Constraint.

Rollback.

Or termination.

A workflow should scale when the operating model demonstrates stronger outcomes, acceptable control, sustainable economics and sufficient reliability.

It should be redesigned when value appears possible but structural failure prevents realization.

It should be constrained when the system creates useful value but autonomy exceeds the enterprise's ability to govern it.

It should stop when the economic or operating thesis no longer survives evidence.

Stopping is not transformation failure.

Continuing a weak operating model because the enterprise has already invested in it is failure.

The transition system must preserve the ability to abandon assumptions.

AI Native transformation is therefore not a one-way deployment pipeline.

It is an evidence system.

Exhibit 13.5

The Transformation Decision

OPERATING EVIDENCE

What happened in production?

OUTCOME

Did the intended enterprise result improve?

ECONOMICS

Did the value created justify the resources consumed?

CONTROL

Did execution remain within acceptable authority and risk boundaries?

RELIABILITY

Can the workflow operate consistently enough for its consequence level?

SYSTEM EFFECT

Did improvement occur without creating unacceptable downstream work or organizational friction?

DECISION

Scale — Expand the operating model where evidence supports it. Redesign — Change the workflow, allocation, authority or architecture where the thesis remains valid but the design does not. Constrain — Reduce autonomy, scope or execution boundaries where value exists but control is insufficient. Stop — Withdraw the transformation where the operating or economic thesis fails.

The enterprise learns through all four outcomes.

21 · Build People Who Can Redesign the Enterprise

Technology capability alone will not sustain this transition.

The enterprise needs people capable of reasoning across the operating model.

They must understand more than prompts.

More than models.

More than automation.

Someone redesigning an AI Native workflow must be able to connect:

business outcome

workflow

human judgment

machine capability

authority

management

organizational structure

execution architecture

control

infrastructure

economics

intervention.

This is inherently cross-disciplinary work.

Traditional functional specialization can make it difficult.

The business understands the outcome but may not understand machine capability.

Engineering understands the system but may not own the workflow.

Risk understands controls but may not understand execution architecture.

Finance understands economics but may enter after design decisions have already been made.

Human resources understands roles but may not see the technical system changing them.

Transformation teams may coordinate everyone while owning none of the operating outcome.

AI Native capability therefore requires connective expertise.

People who can move across these domains.

Not because every individual must master all of them.

Because the enterprise must be able to integrate them during design.

Capability building should therefore occur through transformation itself.

Teams learn by redesigning real workflows.

Managers learn by directing mixed execution.

Engineers learn by operating agents under real authority.

Risk teams learn by designing controls around actual machine actions.

Finance learns to measure machine economics against operating outcomes.

Organizational leaders learn where structures obstruct execution.

The enterprise develops capability by operating the new model.

22 · The Transformation Office Should Disappear

Large transformations often create temporary machinery.

Programs.

Steering committees.

Transformation offices.

Special funding.

Dedicated teams.

Executive sponsorship.

These mechanisms can be useful during transition.

They should not become the permanent home of AI Native operation.

If every meaningful workflow redesign requires a special transformation program, the enterprise has not become AI Native.

It has created a department responsible for temporarily making parts of the enterprise AI Native.

The destination is different.

Workflow redesign becomes an operating capability.

Human-machine allocation becomes a management capability.

Authority engineering becomes part of operating design.

Agent control becomes infrastructure.

Machine economics becomes part of financial management.

Intervention becomes part of operations.

Adaptation becomes normal.

The temporary transformation system should therefore transfer capability into the enterprise as it matures.

The center enables.

The enterprise learns.

The capability distributes.

The special machinery recedes.

This is an important test of transformation maturity.

The transformation is succeeding when the enterprise becomes less dependent on the transformation program.

23 · Becoming Is Not Arriving

There is no final moment when an enterprise can declare that its operating model is now AI Native and therefore complete.

The reason is structural.

Machine capability will continue to change.

The relative advantage of humans and machines will change with it.

A task that requires human judgment today may become machine executable under bounded conditions tomorrow.

An agent architecture that is economically sensible today may become unnecessarily expensive.

A workflow designed around current model latency may become obsolete.

A control created for one generation of machine capability may become insufficient for another.

A management layer that remains necessary during transition may later become redundant.

A centralized capability may eventually need to distribute.

An organizational boundary that works now may obstruct a later execution system.

The transition therefore changes character.

At first, the enterprise is deliberately converting workflows.

Eventually, it must become capable of continuously reconsidering them.

This is the point at which transformation becomes adaptation.

24 · From Transformation to Adaptation

The AI Native enterprise is not defined by the number of agents it operates.

Nor by the percentage of work automated.

Nor by model sophistication.

Nor by the size of its AI investment.

It is defined by whether the enterprise can systematically redesign the relationship between outcomes, work, human judgment and machine intelligence.

That capability begins with transition.

Select an outcome.

Understand the economics.

Redesign the workflow.

Allocate the work.

Delegate authority.

Change management.

Change organizational structure where necessary.

Build the execution system.

Govern machine actors.

Manage compute.

Measure value.

Observe failure.

Intervene.

Learn.

Then do it again.

Not identically.

More intelligently.

The enterprise that develops this capability no longer treats AI as a technology being introduced into a fixed organization.

The organization itself becomes variable.

Work becomes variable.

Authority becomes variable.

Management becomes variable.

Architecture becomes variable.

The allocation of human and machine capacity becomes variable.

That does not mean constant reorganization.

It means the operating model can change when evidence and capability justify change.

The enterprise has moved from adopting AI to becoming capable of reorganizing around intelligence.

That is the threshold.

The final question is what happens after the transition succeeds.

If capability continues to change, the enterprise cannot freeze the operating model it has just created.

It needs a north star that survives changes in technology.

Not a permanent architecture.

Not a permanent allocation of work.

Not a permanent theory of autonomy.

A principle for continuously determining what the enterprise should become next.

That is the adaptive enterprise.

Read online: https://sentient-review.com/perspectives/the-ai-native-operating-model/13-becoming-an-ai-native-enterprise

Chapter 14

The Adaptive North Star

What the best defensible AI Native enterprise looks like today and how that North Star evolves as capability advances.

> *The destination is not an AI Native operating model. It is an enterprise capable of changing its operating model as intelligence changes.*

The destination is not an AI Native operating model.

It is an enterprise capable of changing its operating model as intelligence changes.

That distinction completes the argument.

Every operating model eventually reflects the conditions under which it was designed.

The capabilities available at the time.

The economics of those capabilities.

The people available to perform the work.

The technology available to coordinate it.

The regulatory environment.

The structure of the market.

The expectations of customers.

The speed at which decisions could be made.

Organizations encode these conditions into workflows, roles, authority, management structures, systems and controls.

Then the conditions change.

The operating model often does not.

This is how organizational friction accumulates.

Processes remain because they once solved a problem.

Approvals remain because they once controlled a risk.

Management layers remain because information once had to travel through people.

Functional boundaries remain because expertise once had to be concentrated.

Technology architectures remain because previous systems imposed constraints.

Work remains human because machines once could not perform it.

The enterprise becomes an accumulation of decisions made under previous conditions.

AI changes those conditions again.

But this time the change is unlikely to occur once.

Machine capability will continue to move.

Model economics will move.

Agent capability will move.

Infrastructure will move.

Regulation will move.

Human capability will move.

Customer expectations will move.

Competitors will reorganize.

New forms of work will become possible.

Old forms will become unnecessary.

An enterprise that redesigns itself around today's AI capabilities and then stops will eventually reproduce the same problem.

A new static operating model.

The objective must therefore be different.

Build an enterprise capable of continuously determining how it should operate.

That is the adaptive north star.

01 · The Operating Model Becomes a Variable

Traditional operating-model transformation often assumes a destination.

The enterprise defines a target operating model.

Roles are designed.

Processes are mapped.

Decision rights are assigned.

Technology is implemented.

Organizational structures change.

The transformation concludes.

The new model becomes the baseline.

That logic becomes less stable when one of the productive capabilities inside the operating model is changing rapidly.

AI capability is not a fixed input.

Neither is its cost.

Neither is the architecture required to use it.

A workflow designed around one generation of machine capability may become unnecessarily human intensive as capability improves.

Another may become unnecessarily complex because a newer model can perform work previously divided across several specialized agents.

A decision that once required mandatory human approval may become suitable for bounded machine authority as evaluation, observability and recoverability improve.

A task delegated to a powerful model may later move to a smaller model when economics change.

A centralized capability may become distributed once common controls mature.

The enterprise therefore cannot treat allocation, authority and architecture as permanent design decisions.

They become variables.

This does not mean continuous organizational disruption.

It means continuous operating awareness.

The enterprise knows why work is allocated as it is.

It knows why authority sits where it does.

It knows why a workflow contains its current steps.

It knows why a particular machine capability is being consumed.

It knows what economic result the configuration is expected to produce.

And it can reconsider those decisions when their underlying assumptions change.

Exhibit 14.1

The Adaptive Question

OUTCOME

What must the enterprise produce now?

WORK

What work is currently required to produce it?

ALLOCATION

Given current human and machine capability, who or what should perform that work?

AUTHORITY

What decisions and actions should each actor be permitted to make?

MANAGEMENT

How should mixed execution be directed, observed and adapted?

ORGANIZATION

What structure best supports the workflow and its accountability?

EXECUTION ARCHITECTURE

What technical system should enable machine execution?

INFRASTRUCTURE

What machine resources should the enterprise consume?

ECONOMICS

Does this configuration create sufficient enterprise value?

INTERVENTION

Where is the operating model failing?

ADAPTATION

What should change next?

This is not a transformation sequence that runs once.

It is a loop.

02 · The Loop Must Remain Open

Chapter 13 established the transition architecture.

Select an outcome.

Baseline it.

Redesign the workflow.

Reallocate work.

Delegate authority.

Change management.

Change organizational structure where necessary.

Build execution architecture.

Measure economics.

Operate.

Intervene.

Scale what works.

That process allows an existing enterprise to become AI Native.

But transition solves only the first problem.

It moves the enterprise away from an operating model designed without current machine intelligence.

The next problem begins immediately.

The capability frontier moves again.

The workflow that was redesigned becomes an operating hypothesis.

So does its allocation of work.

So does its authority architecture.

So does its management model.

So does its technical architecture.

So does its economic model.

The enterprise must therefore preserve the ability to question its own design.

This is where the intervention system established in Chapter 12 acquires a larger purpose.

Intervention is not only how the enterprise corrects failure.

It is one of the mechanisms through which the enterprise learns that its operating model should change.

A persistent approval queue may indicate that machine authority should expand.

A rising exception rate may indicate that allocation should change.

Increasing compute cost may indicate that architecture should simplify.

Falling human intervention may make a management layer unnecessary.

A new model capability may collapse several workflow stages into one.

A regulatory change may require authority to contract.

A market change may alter the outcome itself.

The loop must remain open because the assumptions inside the operating model remain exposed to change.

03 · Adaptation Is Not Optimization

The distinction matters.

Optimization improves performance within an existing system.

Adaptation can change the system.

Optimization asks:

How can this workflow execute faster?

Adaptation asks:

Should this still be the workflow?

Optimization asks:

How can this agent perform better?

Adaptation asks:

Should an agent perform this work at all?

Optimization asks:

How can human review become more efficient?

Adaptation asks:

Why does this decision still require human review?

Optimization asks:

How can the organization coordinate the process better?

Adaptation asks:

Does this coordination still need to exist?

Optimization asks:

How can compute cost fall?

Adaptation asks:

Is this architecture still economically justified?

Both matter.

But an AI Native enterprise cannot optimize itself into adaptation.

It needs the authority and operating capability to change the design itself.

That is a different organizational property.

It requires the enterprise to observe not merely performance.

It must observe the assumptions behind performance.

04 · The Adaptation Trigger

The operating model should not change simply because a new model has been released.

Nor because a competitor announces an agent.

Nor because another technical capability becomes fashionable.

Adaptation needs evidence.

A meaningful trigger may come from several directions.

Capability Change

Machines become able to perform work they previously could not perform reliably.

Economic Change

The cost of machine execution changes enough to alter the optimal allocation of work.

Operating Evidence

Production behavior reveals queues, exceptions, failure propagation, management overhead or other structural weaknesses.

Human Capability Change

People develop new skills or shift toward forms of judgment and work that create greater value.

Market Change

Customer expectations, competitive behavior or business economics alter the required outcome.

Regulatory Change

New rules change what may be delegated, what must be observed or where accountability must remain human.

Architectural Change

New infrastructure, models, tools or control mechanisms make a different execution design possible.

The trigger creates a question.

It does not dictate the answer.

A more capable model does not automatically justify more autonomy.

Lower model cost does not automatically justify more machine execution.

A new regulation does not automatically require less AI.

The enterprise must determine what the changed condition means for the outcome.

Exhibit 14.2

The Adaptation Trigger

CONDITION CHANGES

Capability Economics Operating evidence Human capability Market Regulation Architecture

ASSUMPTION CHALLENGED

Which assumption inside the current operating model is no longer reliable?

OPERATING QUESTION

Does the workflow, allocation, authority, management, organization or architecture still make sense?

REDESIGN HYPOTHESIS

What alternative configuration may produce a better outcome?

CONTROLLED INTERVENTION

Change the relevant part of the operating model.

EVIDENCE

Did the new configuration improve the enterprise outcome?

ADOPT · CONSTRAIN · ROLLBACK · CONTINUE LEARNING

Adaptation is therefore disciplined change.

Not perpetual reinvention.

The objective is not to keep the enterprise moving.

It is to keep the enterprise aligned with the conditions under which it must produce outcomes.

05 · From Periodic Transformation to Continuous Design

Most enterprises know how to run transformation programs.

They are less accustomed to treating operating design as a persistent capability.

Transformation usually has a beginning.

A mandate.

A program.

Funding.

Milestones.

A target state.

An end.

The adaptive enterprise requires something more durable.

The ability to redesign work cannot disappear when the transformation office closes.

The ability to reconsider authority cannot belong only to a governance project.

The ability to alter human-machine allocation cannot require a new enterprise initiative every time machine capability changes.

The ability to measure machine economics cannot remain a temporary analytical exercise.

The ability to intervene cannot depend on extraordinary executive attention.

These must become operating capabilities.

The enterprise does not remain permanently in transformation.

It becomes capable of transformation without requiring exceptional machinery every time.

That is the shift.

Transformation becomes a property of the operating system itself.

06 · The Adaptive Operating System

If transformation becomes a property of the operating system, the enterprise needs a mechanism through which adaptation occurs.

That mechanism is not another organizational layer.

It is a closed operating loop.

The enterprise observes conditions.

It observes outcomes.

It observes execution.

It observes economics.

It observes failure.

It determines whether the assumptions inside the current operating model still hold.

Where they do, the system continues.

Where they do not, the enterprise intervenes.

This creates an operating system capable of modifying its own configuration.

Not autonomously in every respect.

Not continuously in every workflow.

Not without human accountability.

But deliberately, repeatedly and on the basis of evidence.

The enterprise begins to manage its operating model as something dynamic.

07 · The North Star Is the Outcome

Adaptation needs a stable reference point.

Technology cannot provide it.

Models change.

Architectures change.

Agents change.

Infrastructure changes.

The percentage of machine-executed work changes.

None of these should become the objective.

The stable reference point is the enterprise outcome.

What must be produced?

For whom?

At what quality?

At what speed?

At what risk?

At what economic return?

Everything else can be reconsidered.

This prevents adaptation from becoming technology chasing.

A more capable model matters only if it changes what becomes possible for an enterprise outcome.

A cheaper model matters only if it changes the economics of execution.

A new agent architecture matters only if it improves the system through which the outcome is produced.

A new form of autonomy matters only if the enterprise can use it productively and accountably.

The outcome anchors the system while the means remain variable.

This is why the conceptual sequence of this publication began there.

Outcome

What must be produced?

Everything downstream exists in service of that answer.

08 · Work Becomes Recomposable

Once the outcome becomes the stable reference point, the workflow no longer needs to be treated as permanent.

It can be decomposed and recomposed.

A workflow may initially contain twenty activities.

Machine capability may eliminate five.

Another six may combine.

Two human approvals may become machine decisions within explicit thresholds.

A specialist review may move from universal participation to exception handling.

An agent may absorb coordination previously performed by a manager.

A new regulatory requirement may add a control.

A change in customer behavior may introduce a new decision point.

The workflow changes because the conditions of execution change.

This is fundamentally different from conventional process improvement.

The enterprise is not merely making each step more efficient.

It can reconsider whether the step belongs in the system.

The workflow becomes a current configuration of work required to produce an outcome.

Not an inherited truth.

That makes work recomposable.

09 · Allocation Becomes Dynamic

The same principle applies to human and machine allocation.

There is no permanent answer to:

Should a human or machine perform this work?

The answer depends on capability.

Consequence.

Economics.

Context.

Reliability.

Authority.

Control.

And the comparative value of human attention.

As those conditions change, allocation should be reconsidered.

This creates a more useful question:

Where does human capacity create the greatest marginal value relative to machine capacity now?

That question is different from asking how many jobs AI can automate.

It begins with productive allocation rather than substitution.

Humans should not remain inside a workflow simply because they historically performed the work.

Machines should not enter a workflow simply because they can perform it.

Each should occupy the parts of the operating system where their capabilities create the greatest value under the required constraints.

That allocation will move.

Routine execution may migrate toward machines.

Human work may concentrate around judgment, accountability, ambiguity, relationships, invention, strategic choice and exception resolution.

Some of those categories may themselves change as machine capability improves.

New human work may appear.

New machine work will appear.

Some work will disappear entirely.

The adaptive enterprise does not attempt to predict the final division.

It develops the capability to continuously redesign it.

10 · Human Attention Becomes a Scarce Resource

Machine capacity can expand rapidly.

Human attention cannot.

This creates one of the most important economic constraints in the adaptive enterprise.

Every human approval consumes attention.

Every escalation consumes attention.

Every exception consumes attention.

Every management review consumes attention.

Every agent requiring supervision consumes attention.

Every poorly designed interface between humans and machines consumes attention.

Human involvement is therefore not free simply because the employee already exists.

It is an allocation of scarce judgment.

The enterprise should ask:

Where must humans remain directly involved?

Where does human involvement materially improve the outcome?

Where is human accountability required?

Where is human judgment economically valuable?

Where has human involvement become inherited friction?

This makes the placement of human attention an operating-design decision.

The objective is not to remove humans from execution.

It is to stop consuming human judgment where human judgment does not improve the result.

The consequence is leverage.

Machine intelligence expands the amount of execution that can occur around a finite pool of human judgment.

But only if the operating model protects that judgment from becoming the next bottleneck.

Exhibit 14.3

The Human Attention Test

ACCOUNTABILITY

Must a human own this decision?

JUDGMENT

Does the decision require human interpretation, trade-off or contextual reasoning that materially improves the outcome?

CONSEQUENCE

Is the potential consequence high enough to justify direct human involvement?

UNCERTAINTY

Is the machine operating outside conditions where its behavior is sufficiently reliable?

EXCEPTION

Is this genuinely exceptional, or has repeated escalation revealed a design failure?

VALUE

Does human attention create more value here than elsewhere?

If the answer is no, the operating model should question why the human remains in the execution path.

Human judgment should become more consequential as machine execution expands.

Not more consumed.

11 · Authority Must Move with Capability

Dynamic allocation without dynamic authority creates another mismatch.

If machines become capable of performing more work but their authority remains fixed, escalation grows.

If authority expands faster than the enterprise's ability to govern machine execution, risk grows.

Authority therefore has to move with capability.

This does not mean continuously expanding autonomy.

Authority may expand.

Contract.

Become conditional.

Move between actors.

Move from approval to retrospective review.

Move from a person to policy.

Move from a general agent to a specialist agent.

Return to humans when conditions change.

The authority architecture becomes configurable.

A machine actor may possess authority only:

for a defined action

within a defined workflow

against approved systems

within an economic threshold

for a defined period

under observable conditions

with specified escalation rules

and with the ability to revoke that authority.

This creates accountable autonomy as a dynamic operating property.

Not a one-time governance decision.

12 · Management Becomes System Direction

Management changes again under these conditions.

When work, allocation and authority become more dynamic, management cannot consist primarily of supervising human task execution.

Managers increasingly direct a mixed execution system.

They determine outcomes.

Set constraints.

Allocate capacity.

Interpret evidence.

Resolve ambiguity.

Change authority.

Intervene in failure.

Develop people.

Challenge assumptions.

Make trade-offs.

Decide when operating design should change.

This is not the disappearance of management.

It is a change in its object.

The manager increasingly manages the system through which work occurs.

Some traditional management activity becomes machine executable.

Status collection.

Routine coordination.

Work routing.

Progress visibility.

Basic exception detection.

Information aggregation.

Other management responsibilities become more important.

Judgment.

Accountability.

System design.

Resource allocation.

Intervention.

Capability development.

The adaptive enterprise should therefore continuously examine management work using the same logic applied to every other workflow.

Which management activity produces value?

Which exists because information was previously difficult to observe?

Which exists because coordination previously required people?

Which exists because authority was fragmented?

Which remains essential because human judgment is required?

Management is part of the operating model.

It cannot remain outside adaptation.

13 · Organization Becomes Less Synonymous with Capacity

Traditional organizational capacity is strongly associated with people.

A team has twenty employees.

A function has five hundred.

A business unit has several thousand.

Headcount becomes a proxy for productive capacity.

Machine execution weakens that relationship.

A team may command substantial machine capacity without adding employees.

One group may operate dozens of specialized machine actors.

Another may consume shared agent capabilities.

Machine execution may move across organizational boundaries without corresponding changes in headcount.

The enterprise therefore needs a broader concept of capacity.

Human capacity.

Machine capacity.

Infrastructure capacity.

And the management capacity required to direct them.

This changes organizational design.

The question becomes less:

How many people should this function contain?

And more:

What combination of human judgment and machine capacity does this outcome require?

That combination may not map neatly onto the existing hierarchy.

14 · Machine Capacity Must Have an Owner

Machine capacity may be elastic.

Accountability cannot be.

Every consequential machine actor must exist inside an accountable operating structure.

Someone owns the outcome.

Someone owns the authority boundary.

Someone owns the execution architecture.

Someone owns the controls.

Someone owns the economics.

Someone can intervene.

This does not require a human to approve every machine action.

It requires the enterprise to know where accountability resides.

The distinction is essential.

Human-in-the-loop is an execution pattern.

Human accountability is an operating principle.

The first may become less necessary as systems improve.

The second does not disappear.

The adaptive enterprise can therefore increase autonomy without dissolving accountability.

15 · The Technical Architecture Must Also Adapt

Operating-model adaptation eventually reaches the technical system.

Agents are not permanent organizational entities.

Models are not permanent infrastructure choices.

Multi-agent architectures are not inherently superior to simpler systems.

Context architectures can become obsolete.

Tool interfaces change.

Control mechanisms improve.

Infrastructure economics shift.

The execution architecture must therefore be treated as replaceable.

This requires modularity.

Models should be routable where appropriate.

Tools should be separable from individual agents.

Identity should persist independently of a particular model.

Policies should be enforceable across heterogeneous runtimes.

Observability should span execution architectures.

Context should be governed as enterprise infrastructure rather than buried inside isolated applications.

Evaluation should survive model substitution.

Economic measurement should follow the workflow even when its technical components change.

The objective is not technical abstraction for its own sake.

It is operating freedom.

The enterprise should be able to change machine capability without rebuilding the operating model from zero.

16 · Compute Becomes an Allocatable Factor of Production

The adaptive enterprise also needs to reconsider infrastructure.

Compute is not merely an information technology expense.

Inside AI Native workflows, it participates directly in production.

More inference can produce more work.

More context can improve some decisions.

More reasoning can improve some outcomes.

More agents can increase parallelism.

Higher-capability models can change what can be delegated.

But each consumes resources.

The enterprise therefore faces an allocation problem.

Where does another unit of machine capacity create economic value?

A high-value decision may justify expensive reasoning.

A routine classification may not.

A low-latency customer interaction may justify a different model than an asynchronous research workflow.

An agent may deserve additional compute when the economic value of improved performance exceeds the additional cost.

Another may need a hard budget.

The adaptive enterprise should therefore allocate compute against outcome value.

Not simply technical demand.

This links infrastructure directly to economics.

17 · Economics Becomes a Control Signal

Chapter 11 established the economic test.

Did the operating model create value?

In the adaptive enterprise, economics does more than validate the transformation afterward.

It becomes an input into operating design.

Cost per outcome rises. Investigate.

Human intervention rises. Investigate.

Compute consumption grows faster than value. Investigate.

Throughput improves but downstream cost increases. Investigate.

A cheaper model becomes capable enough. Reconsider routing.

A workflow creates substantial value but is constrained by machine capacity. Consider allocating more.

Economics becomes a signal that the operating configuration may no longer be optimal.

This is why measurement must remain attached to the outcome.

Without that connection, the enterprise can optimize token cost while destroying value.

Or increase automation while increasing total operating cost.

Or reduce headcount while degrading revenue.

Or improve throughput while increasing risk.

The economic system must observe the whole.

Exhibit 14.4

The Adaptive Operating Loop

SENSE

Observe capability, markets, regulation, execution, human capacity, infrastructure and economics.

QUESTION

Which assumptions inside the operating model may no longer hold?

DIAGNOSE

Locate the implication across outcome, workflow, allocation, authority, management, organization, architecture or economics.

REDESIGN

Form a better operating configuration.

INTERVENE

Change the smallest part of the system capable of testing the hypothesis.

OBSERVE

Measure operating behavior and enterprise outcome.

DECIDE

Continue. Scale. Constrain. Rollback. Redesign again.

LEARN

Change the enterprise's understanding of how the operating model should work.

SENSE AGAIN

The loop does not terminate.

It becomes part of operating the enterprise.

18 · Stability Still Matters

An adaptive enterprise is not an enterprise in permanent structural motion.

Constant redesign would itself become operating failure.

People need durable accountability.

Customers need predictable service.

Controls need consistency.

Technical systems need reliability.

Teams need enough stability to develop expertise.

Adaptation therefore needs thresholds.

Not every variance deserves intervention.

Not every capability improvement deserves redesign.

Not every economic fluctuation deserves architectural change.

Not every new model deserves adoption.

The operating model should change when the expected value of changing it exceeds the cost and disruption of the intervention.

This introduces an important principle.

Adaptability is not the frequency of change.

It is the capacity to change when change becomes justified.

A strong adaptive enterprise may leave a workflow untouched for years because it continues to perform well.

Another workflow may change several times in a quarter because capability and evidence are moving quickly.

The operating system must support both.

Stability where the configuration remains sound.

Change where the assumptions no longer hold.

19 · Adaptation Requires Memory

A system that changes without remembering why it changed will repeat its mistakes.

The enterprise therefore needs institutional memory for operating-model decisions.

Why was this workflow designed this way?

Why does this agent possess this authority?

Why does this approval exist?

Why was this model selected?

What failure caused this control?

What intervention was attempted?

What happened?

What was rolled back?

What evidence justified scale?

What assumption later proved wrong?

This knowledge cannot remain only in the people who participated in the transformation.

Machine execution increases the importance of explicit operating knowledge.

Decision records.

Evaluation history.

Authority changes.

Incident history.

Economic baselines.

Intervention results.

Workflow versions.

Architectural rationale.

These become part of the enterprise's operating memory.

Without memory, adaptation becomes reaction.

With memory, adaptation can become learning.

The enterprise does not merely change.

It becomes better at determining what should change.

That is the beginning of a different kind of organization.

20 · The Adaptive Enterprise

The enterprise described throughout this publication is not autonomous.

It is adaptive.

The distinction matters.

An autonomous enterprise implies an organization progressively removing humans from execution.

That is not the destination.

The destination is an enterprise capable of combining human judgment and machine intelligence differently as conditions require.

Sometimes that means more machine execution.

Sometimes less.

Sometimes greater machine authority.

Sometimes tighter constraint.

Sometimes fewer agents.

Sometimes greater specialization.

Sometimes more compute.

Sometimes architectural simplification.

Sometimes human judgment moves out of the execution path.

Sometimes it moves back in.

The operating model changes because the conditions that determine the best allocation have changed.

This is why there can be no universal percentage of work that should be automated.

No universal level of autonomy.

No ideal agent architecture.

No permanent human-machine division of labor.

The correct configuration is contingent on the outcome and the conditions under which it must be produced.

The adaptive enterprise develops the capability to determine that configuration repeatedly.

21 · The Enterprise Becomes a System of Capabilities

This changes how the organization itself should be understood.

The enterprise has traditionally been represented primarily through its organizational structure.

Business units.

Functions.

Teams.

Roles.

Reporting relationships.

These remain important.

But they describe only the human structure.

An AI Native enterprise contains another layer.

Machine capacity.

Agents.

Models.

Tools.

Context.

Compute.

Policies.

Execution systems.

These resources participate directly in producing outcomes.

The enterprise therefore becomes a system of productive capabilities rather than only a hierarchy of people.

Human capability.

Machine capability.

Organizational capability.

Technical capability.

Capital.

Information.

Infrastructure.

Authority.

All must be configured around outcomes.

The organizational chart remains one representation of the enterprise.

It is no longer a sufficient representation of how the enterprise executes.

A future operating map may need to show something different.

Which outcomes exist.

Which workflows produce them.

Which human capabilities participate.

Which machine actors participate.

Where authority resides.

What infrastructure is consumed.

Where accountability sits.

What economic value is produced.

Where intervention can occur.

That is a richer representation of the enterprise.

It describes how the organization actually works.

Exhibit 14.5

The Adaptive Enterprise

OUTCOME

What must be produced?

WORKFLOW

What system of work produces it?

HUMAN CAPACITY + MACHINE CAPACITY

What combination of judgment and execution should perform the work?

AUTHORITY

What may each actor decide and do?

MANAGEMENT

How is mixed execution directed and adapted?

ORGANIZATION

Where should accountability, capability and ownership reside?

EXECUTION ARCHITECTURE

What agents, models, tools, context and controls enable machine execution?

INFRASTRUCTURE + COMPUTE

What productive machine resources should be consumed?

ECONOMIC VALUE

Does the configuration improve the economics of the enterprise?

INTERVENTION

Where is the operating system failing?

ADAPTATION

What should change?

OUTCOME

What must be produced now?

The system returns to the beginning.

But not to the same enterprise.

The enterprise has learned.

22 · The North Star Is Not Autonomy

This distinction becomes increasingly important as machine capability improves.

It will be tempting to measure progress by autonomy.

More autonomous agents.

Longer execution horizons.

Fewer human interventions.

More machine decisions.

Higher percentages of automated work.

These measures may describe technical capability.

They do not necessarily describe enterprise value.

A fully autonomous workflow that produces a weak outcome is not superior to a mixed workflow that produces a strong one.

An agent capable of operating for hours without intervention is not inherently more valuable than a system that asks for human judgment at precisely the right moment.

Removing a human approval is not progress if that approval contains judgment the machine cannot reliably reproduce.

Keeping a human approval is not prudent if it contributes nothing but delay.

Autonomy is therefore not the north star.

Neither is automation.

The north star is the quality of the operating system through which human judgment and machine intelligence combine to produce enterprise outcomes.

Autonomy is one design variable inside that system.

Nothing more.

23 · Human Judgment Becomes More Important, Not Less

As machines perform more execution, the remaining human decisions become more consequential.

What outcome should the enterprise pursue?

What trade-off should it accept?

What risk is tolerable?

Which customer interest should take precedence?

When should policy change?

When should authority expand?

When should autonomy contract?

What should happen when objectives conflict?

What values should govern a decision where economics alone are insufficient?

What consequence is the enterprise willing to own?

These are not simply residual tasks left behind because machines cannot yet perform them.

They are acts of accountable judgment.

Machine intelligence can inform them.

Model alternatives.

Surface evidence.

Identify patterns.

Simulate consequences.

Challenge assumptions.

Execute the resulting decision.

But accountability remains anchored in the enterprise.

The more execution machines absorb, the more important it becomes to know where human judgment must remain authoritative.

The purpose of AI Native design is therefore not to minimize the human role.

It is to make the human role more deliberate.

24 · Machine Intelligence Becomes Ordinary

The inverse is also true.

Machine intelligence should eventually become less exceptional.

Enterprises do not create a transformation program every time software executes a calculation.

They do not convene a governance council every time a database query runs.

They do not celebrate every API call.

These technologies became part of the operating substrate.

AI will not become identical to previous software.

Its probabilistic behavior, generative capability, variable reasoning and potential agency create different operating requirements.

But mature use should still become ordinary.

Agents become execution resources.

Models become capability resources.

Compute becomes productive capacity.

Context becomes infrastructure.

Evaluation becomes operational discipline.

Authority becomes configurable.

Machine actors become visible participants in workflows.

The enterprise stops asking where it can put AI.

It asks how the outcome should be produced.

Sometimes AI will be central to the answer.

Sometimes peripheral.

Sometimes unnecessary.

That is maturity.

25 · AI Native Does Not Mean AI Everywhere

A genuinely AI Native enterprise should be capable of deciding not to use AI.

Some work is deterministic.

Some is already efficient.

Some carries consequences that make machine execution inappropriate.

Some does not occur at sufficient scale to justify the architecture.

Some automation creates more complexity than value.

Some decisions should remain explicitly human.

Some workflows may require conventional software rather than probabilistic systems.

Some problems should be removed rather than automated.

The AI Native enterprise is therefore not defined by technological saturation.

It is defined by design freedom.

The enterprise understands machine intelligence well enough to use it where it changes the economics or capability of the operating model.

And to reject it where it does not.

This protects the enterprise from turning AI Native transformation into another technology ideology.

The objective remains operating performance.

26 · The Competitive Advantage Moves

As models become broadly available, access to intelligence becomes less differentiating by itself.

Competitors can access similar foundation models.

They can acquire similar infrastructure.

They can use similar development frameworks.

They can hire from overlapping talent markets.

The durable advantage increasingly moves into the enterprise system around those capabilities.

The quality of its workflows.

The quality of its context.

The clarity of its authority.

The speed of its learning.

The discipline of its economics.

The reliability of its execution architecture.

The quality of human judgment.

The ability to redesign organizational structures.

The ability to intervene before failure compounds.

The ability to absorb new capability without destabilizing the enterprise.

The ability to turn machine intelligence into operating leverage.

Two companies can possess access to comparable models and produce very different economic outcomes.

The difference can sit in the operating model.

This is why AI strategy ultimately becomes operating strategy.

The model is not the enterprise.

The agent is not the enterprise.

The platform is not the enterprise.

The enterprise is the system that converts capability into outcomes.

27 · The Adaptation Advantage

This leads to the deeper source of advantage.

A competitor can copy an agent.

It can adopt the same model.

It can reproduce an interface.

It can purchase similar infrastructure.

It can imitate a workflow.

It is harder to copy an enterprise that has learned how to continuously redesign itself.

Adaptation compounds.

Each workflow teaches the enterprise something about allocation.

Each authority decision teaches it something about autonomy.

Each failure teaches it something about intervention.

Each economic result improves capital allocation.

Each architecture decision expands or constrains future options.

Each management redesign changes how the enterprise directs capacity.

Each organizational change alters what becomes possible next.

The result is not merely a collection of AI systems.

It is an institutional capability.

The ability to repeatedly convert changes in intelligence into changes in enterprise performance.

That capability may become more important than any individual generation of AI technology.

Because the technology will change.

The enterprise that can absorb the change retains the advantage.

Exhibit 14.6

The Adaptation Advantage

CAPABILITY CHANGES

New machine capability becomes available.

ENTERPRISE INTERPRETS

What does the change mean for our outcomes?

OPERATING MODEL RESPONDS

Workflow Allocation Authority Management Organization Architecture Infrastructure Economics

ENTERPRISE LEARNS

Production evidence improves understanding.

CAPABILITY TO ADAPT IMPROVES

The next change can be absorbed more intelligently.

COMPOUNDING OPERATING ADVANTAGE

The advantage is not simply possessing better intelligence.

It is becoming better at reorganizing around intelligence.

28 · What Remains Human

The argument can now return to its beginning.

AI Native transformation is often described through what machines will become capable of doing.

That matters.

But the operating-model question is larger.

What should the enterprise become capable of doing because machines can do more?

The answer is not simply more automation.

It is greater leverage.

Greater adaptability.

Greater execution capacity.

Better allocation of human judgment.

More explicit authority.

More observable operations.

More disciplined economics.

Faster learning.

The human role does not disappear from this system.

It changes position.

Humans increasingly determine purpose.

Define outcomes.

Make consequential trade-offs.

Design boundaries.

Exercise judgment.

Accept accountability.

Interpret ambiguity.

Create new possibilities.

Intervene when the system no longer produces what the enterprise intends.

Machines expand the capacity around that judgment.

They execute.

Search.

Reason.

Coordinate.

Monitor.

Generate.

Analyze.

Route.

Act.

And increasingly, they do so with meaningful delegated authority.

The operating model determines how these capabilities meet.

That is the design problem.

29 · There Is No Final Operating Model

The temptation at the end of a transformation is to institutionalize what has been built.

Document it.

Standardize it.

Govern it.

Protect it.

That is necessary.

It is also dangerous.

Today's successful AI Native operating model contains tomorrow's legacy assumptions.

A workflow that is advanced today can become obsolete.

An authority architecture that is appropriate today can become unnecessarily restrictive.

An agent architecture that is sophisticated today can become redundant.

A management structure that enables transition today can become tomorrow's coordination burden.

The enterprise must therefore institutionalize principles without freezing configurations.

Outcome before technology.

Workflow before task automation.

Allocation before substitution.

Authority with accountability.

Management around mixed execution.

Organization around outcomes.

Architecture around governed machine action.

Infrastructure around productive value.

Economics across the complete system.

Intervention before failure compounds.

Adaptation when assumptions change.

Those principles can endure even as their implementation changes.

30 · The Adaptive North Star

The enterprise can now ask the complete set of questions.

What outcome matters?

What work produces it?

What should humans do?

What should machines do?

What authority should each possess?

How should execution be managed?

How should the organization be structured?

What technical system enables execution?

What infrastructure should be consumed?

What economic value is being produced?

Where is the operating model failing?

What should change next?

These questions form the adaptive north star.

They do not prescribe one organizational structure.

One architecture.

One model.

One agent framework.

One level of autonomy.

One division of labor.

They provide something more durable.

A way for the enterprise to determine those things as conditions change.

31 · The AI Native Enterprise

The AI Native enterprise is therefore not the enterprise with the most AI.

It is not the enterprise with the most agents.

It is not the enterprise that automates the highest percentage of work.

It is not the enterprise that removes humans fastest.

It is the enterprise that understands how to combine human judgment and machine intelligence into an operating system capable of producing better outcomes.

And then change that system when a better configuration becomes possible.

That is the shift.

From AI as a tool.

To AI as execution capacity.

From automation of tasks.

To redesign of workflows.

From human work versus machine work.

To deliberate allocation.

From permission after execution.

To designed authority.

From management of people alone.

To management of mixed capacity.

From organization as hierarchy.

To organization as an architecture of outcomes, accountability and capability.

From agents as applications.

To machine actors inside the operating system.

From compute as technology expense.

To compute as a productive resource.

From productivity claims.

To economic value.

From failure as an exception.

To failure as operating evidence.

From transformation as a program.

To adaptation as an enterprise capability.

The destination is not an autonomous enterprise.

It is not a machine enterprise.

It is not an enterprise without people.

It is an enterprise designed around human judgment and machine intelligence working together to produce outcomes.

An enterprise that knows what should remain human.

What should become machine executed.

What authority each should possess.

What each unit of capacity costs.

What value the system creates.

Where the operating model is failing.

And when the design itself should change.

AI Native is not the final operating model.

It is the capability to keep designing the right one.

That is the adaptive enterprise.

Read online:
https://sentient-review.com/perspectives/the-ai-native-operating-model/14-the-adaptive-north-star

Connected Work

The Framework

FW 10 · Operationalized

The AI Native Operating Model Framework

https://sentient-review.com/frameworks/the-ai-native-operating-model-framework

Developed Through

FN 11 — The Acceptance Bottleneck

https://sentient-review.com/field-notes/the-acceptance-bottleneck

FN 12 — The Missing Destination of Saved Time

https://sentient-review.com/field-notes/the-missing-destination-of-saved-time

FN 13 — Management After Routing

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

FN 14 — The Organization Inside the Inference Bill

https://sentient-review.com/field-notes/the-organization-inside-the-inference-bill

FN 15 — Human Value When Intelligence Is Abundant

https://sentient-review.com/field-notes/human-value-when-intelligence-is-abundant

FN 16 — The Price of Delegated Authority

https://sentient-review.com/field-notes/the-price-of-delegated-authority

FN 17 — The Workflow Is the Unit

https://sentient-review.com/field-notes/the-workflow-is-the-unit

FN 18 — Cheap Tokens, Expensive Outcomes

https://sentient-review.com/field-notes/cheap-tokens-expensive-outcomes

FN 19 — Generation Is Not Delivery

https://sentient-review.com/field-notes/generation-is-not-delivery

FN 20 — The Mixed Agent Enterprise

https://sentient-review.com/field-notes/the-mixed-agent-enterprise

Connected Practices

Practice — AI & Technology Value Creation Sprint

Built By Sentient · https://builtbysentient.com/practice/ai-technology-value-creation-sprint

Practice — Technology & AI Economics Optimization

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

Practice — AI-Native Engineering Operating Model

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

Practice — Agentic AI Architecture, Security & Governance Blueprint

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

Citation

Sentient Review. (September 2026). The AI Native Operating Model. Volume 01, Publication 01. Version 1.0.

Canonical URL

https://sentient-review.com/perspectives/the-ai-native-operating-model