

SENTIENT REVIEW

Volume 01 · Publication № 02 · Published

Agentic Execution

The Architecture of Delegated Action in the Enterprise

VOLUME	Volume 01
PUBLICATION	Publication № 02
PUBLICATION STATUS	Published
RESEARCH DOMAIN	Agentic AI Systems
EDITOR	Fritzie M. Alex

Topics

Agentic Execution · Delegated Authority · Enterprise Architecture · AI Governance · Operating Models

Canonical Edition

<https://sentient-review.com/perspectives/agentic-execution>

Executive Summary

Agentic Execution examines how the enterprise operates when work is delegated to systems capable of inference, context assembly, and action. The publication begins with the decision to delegate and the boundary of agentic execution, establishing that an agent is not the system — execution requires an architecture around the model, not the model alone.

The middle chapters define that architecture: the execution system that carries work from intent to action; the state, memory, and provenance that keep execution accountable; the authority model that governs delegation; and the agent control plane through which the enterprise directs, observes, and constrains agentic work. Operability follows — the mixed agent enterprise, recovery as a business capability, and assurance at the action boundary.

The economic chapters establish accepted outcomes as the unit of value and derive the measure of agentic economics: cost per accepted outcome, where human, model, infrastructure, review, and recovery costs are weighed against the outcomes the enterprise accepts. Capacity released by agentic execution needs a destination, and the review burden an enterprise carries reveals its operating maturity.

The concluding chapters turn to the executive architecture: how the enterprise builds agentic capability deliberately — beginning with a valuable outcome, designing the workflow around it, establishing the operating envelope, building shared enterprise capabilities, and expanding on evidence. The publication closes with the adaptive agentic enterprise: an operating condition that evolves with capability, built around outcomes, accountability, and the deliberate composition of human and machine capability.

Contents

Part I · The Threshold

01 The Decision to Delegate

02 The Boundary of Agentic Execution

Part II · The Agentic Execution System

03 Inference, Context, and Action

04 The Execution System

05 State, Memory, and Provenance

06 Authority and Delegation

07 The Agent Control Plane

Part III · Operability

08 The Mixed Agent Enterprise

09 Recovery as a Business Capability

Part IV · Outcome Economics

10 Assurance at the Action Boundary

11 Accepted Outcomes

12 The Economics of Agentic Execution

Part V · The Executive Architecture

13 Building Agentic Capability

14 The Adaptive Agentic Enterprise

Part I

The Threshold

01 — The Decision to Delegate

02 — The Boundary of Agentic Execution

Chapter 01

The Decision to Delegate

Enterprise AI began as a productivity proposition.

Could a model summarize a document, draft a response, accelerate analysis, write a first version of code, or help an employee navigate a complex policy? The value sat close to the individual user. A person remained present to frame the task, interpret the output, decide what mattered, and carry the work into the enterprise.

Agentic AI changes the proposition.

The system can receive a signal, assemble context, select a next action, call an enterprise tool, interpret the result, and continue the work. It can update a customer record, create a case, prepare an engineering change, initiate a procurement step, coordinate a handoff, or resolve a bounded operational exception.

The question therefore moves from capability to delegation.

"What authority can the enterprise place inside a machine execution system?"

This is an executive decision because every delegated action carries a consequence.

A service agent that drafts a customer response creates one form of participation. A service agent that issues a credit creates another. An engineering agent that prepares a change plan participates differently from one that initiates deployment. A finance agent that identifies a discrepancy holds a different role from one that posts a correction.

Each action changes the operating environment.

It can affect a customer relationship, a financial position, a production system, a regulatory obligation, an employee workflow, or the quality of an enterprise record. The enterprise must therefore define the purpose of delegation with the same care it applies to decision rights, financial controls, and accountable ownership.

The central design question is simple:

"What work can the system carry, what authority does that work require, and what evidence establishes a successful outcome?"

This question gives agentic AI its operating boundary.

Delegation begins with the outcome

A mature enterprise begins with the result.

Consider a delayed customer order. The desired outcome is not a beautifully written explanation of the delay. The desired outcome is a customer resolution that meets service policy, preserves commercial value, respects available inventory, and leaves the order record in a coherent state.

That outcome may require several forms of work.

The system may gather carrier information, inspect inventory, review customer history, identify policy conditions, prepare a communication, request approval, issue a replacement, create a service credit, or escalate a complex exception.

Each form of work carries a different level of authority.

The enterprise can define an operating envelope around the outcome. A low value replacement may proceed through an approved workflow. A large commercial concession may enter an approval path. A disputed account condition may route to a human owner with the appropriate decision rights.

The agentic system becomes useful when it can carry work through this environment with clarity, speed, and evidence.

The enterprise retains responsibility for the purpose, constraints, and consequences of the delegation.

Authority becomes an architectural concern

Traditional enterprise architecture often focuses on applications, data, integration, infrastructure, resilience, and security.

Agentic execution adds another architectural concern: the distribution of authority.

The system needs an identity. It needs access to selected information. It needs permission to call defined tools. It needs policies that establish financial limits, approval thresholds, time boundaries, escalation conditions, and intervention rights.

These elements create the operating envelope for action.

The model may interpret the situation and propose a next step. Authority determines the set of actions available to that step. Policy determines the conditions under which the action proceeds. The execution environment records the event, carries the action into connected systems, and preserves evidence for review.

This is why agentic architecture reaches beyond model selection.

A strong model can improve interpretation. A clear authority architecture enables accountable execution.

The enterprise gains value through both.

The delegation gradient

Delegation expands through a gradient.

At one end, the system provides insight. It summarizes information, identifies patterns, and prepares options.

Further along, it recommends an action. A person reviews the recommendation and decides.

The system can then prepare execution. It completes the work required for a human owner to approve the final step.

Beyond that point, it can execute bounded actions inside a defined operating envelope.

At the far end, it can coordinate a sequence of actions across systems, functions, and time, while assurance mechanisms observe the work and route exceptions toward accountable owners.

Each stage creates a different relationship between human judgment and machine execution.

The objective is the strongest combined system of human responsibility, machine capability, operational control, and economic value.

This calls for precision. Some work benefits from rapid machine execution. Some work calls for experienced human judgment. Many important workflows combine both.

The enterprise gains strategic advantage by designing the allocation deliberately.

The executive mandate

The decision to delegate creates a new executive mandate.

Leaders must define the outcomes that matter, the work that changes, the authority that moves, the controls that govern action, the evidence that demonstrates value, and the ownership that remains accountable for the result.

This mandate spans technology, operations, finance, security, risk, legal, and functional leadership.

Agentic AI therefore belongs inside the enterprise operating model.

It changes how work flows. It changes where decisions occur. It changes the infrastructure required for execution. It changes the economics of capacity. It changes the evidence leaders need in order to scale an intervention responsibly.

The enterprise that approaches agentic AI as a collection of tools will create scattered capability.

The enterprise that approaches it as a disciplined system of delegation can create a new form of coordinated execution.

The threshold begins here.

The question is no longer whether machines can participate in work.

The question is how the enterprise chooses to place authority, accountability, and value inside that participation.

Read online: <https://sentient-review.com/perspectives/agnostic-execution/01-the-decision-to-delegate>

The Boundary of Agentic Execution

The most consequential boundary in Agentic AI sits between inference and action.

A model can interpret a customer request, identify a pattern in a financial record, summarize an incident, generate a software change, or recommend a next step. These are acts of inference.

Enterprise action begins when that inference enters the systems through which work changes the world.

A customer record is updated. A payment is initiated. Inventory is reserved. A production deployment begins. A supplier receives a request. A service commitment changes. A decision enters an operating process with financial, customer, security, or regulatory consequence.

This is the boundary of agentic execution.

The model creates a proposal. The enterprise execution system creates an effect.

The distinction determines where the enterprise places its attention.

Model capability shapes the quality of interpretation. The execution system shapes the quality of consequence.

The system around the model

An agentic system consists of more than a model and a tool catalog.

It operates through a connected environment that supplies the information, authority, continuity, and evidence required for work to reach an accepted outcome.

System responsibility	Executive meaning
Inference	The model interprets the situation and selects an available next action
Context	Enterprise information, policies, records, workflow history, and tool results inform the action
Tools	Approved interfaces carry action into enterprise systems
State	A durable record preserves the task, its progress, and its pending conditions
Authority	Delegated permissions establish the operating envelope
Recovery	The system manages interruption, delay, duplication, escalation, and completion

Assurance	Evidence establishes the quality and effect of the work
Economics	The enterprise measures the cost and value of accepted outcomes

Each responsibility can live in a different technology component. Some can operate through shared platforms. Others can sit inside a domain application or workflow environment.

The architecture matters because each responsibility carries a different form of accountability.

A model can select a refund option. A policy engine can determine whether the refund falls inside the authorized threshold. A payment service can execute the transaction. A workflow runtime can preserve the task state. An assurance layer can confirm that the customer account reflects the intended result.

Together, these components create an agentic execution system.

Tool access and delegated action

Tools create the bridge between language and consequence.

An agent can query a customer system, inspect an inventory record, create a service case, initiate a shipment, update a document, or submit an engineering change. Each tool expands the range of work the system can carry.

Tool access therefore requires a precise operating design.

The enterprise needs to define the purpose of each action, the authority required to invoke it, the conditions that shape execution, the evidence recorded after the action, and the human owner responsible for the outcome.

A tool can expose a function. Authority determines the conditions under which that function serves the enterprise.

This relationship becomes especially important when an agent works across multiple systems.

A support workflow may combine customer information, delivery status, inventory availability, service policy, customer communication, and financial adjustments. Each system can hold a valid view of the situation. The execution architecture brings those views together around a defined outcome and routes the work through an accountable sequence.

The objective is coherence across the action path.

State gives work continuity

Agentic work can extend across minutes, hours, days, and multiple enterprise systems.

A request may await an approval. A supplier may respond later. A customer may provide additional information. A downstream system may require reconciliation after a delayed response. An operational exception may require several cycles of investigation and action.

State gives this work continuity.

It records the task, the objective, the actions completed, the evidence gathered, the approvals received, the current conditions, and the next available step.

This is distinct from memory.

Memory can provide reusable information across tasks. State preserves the reality of the current task. A strong agentic system treats both as governed enterprise assets with defined ownership, access, lifecycle, and correction processes.

The enterprise gains confidence when it can reconstruct the journey from initiating signal to accepted outcome.

Verification completes the action

A generated answer demonstrates that the system can communicate.

An accepted outcome demonstrates that the system can operate.

This distinction changes how leaders measure Agentic AI.

A customer message can appear helpful while the underlying issue remains active. A proposed engineering fix can appear convincing while the production environment remains exposed. A financial recommendation can appear sound while the underlying ledger continues to diverge.

Verification follows the work into the external environment.

It asks:

- Did the intended action occur?
- Did the relevant enterprise record reach the expected state?
- Did the outcome satisfy the defined acceptance criteria?
- Did the action create a new exception or exposure?
- Did the work produce the expected operational and economic result?

These questions turn agentic execution into an accountable enterprise capability.

The boundary creates the architecture

The enterprise can therefore view Agentic AI through one decisive lens:

- 01 Inference proposes.
- 02 Authority bounds.
- 03 Execution acts.
- 04 Assurance verifies.
- 05 Economics determines value.

This is the architecture of delegated action.

It gives technical teams a clear design target. It gives security and risk leaders visible control points. It gives functional leaders ownership of business outcomes. It gives executives a way to distinguish a compelling demonstration from an operating capability that can earn expansion.

The boundary between inference and action is where Agentic AI becomes an enterprise decision.

Read online:

<https://sentient-review.com/perspectives/agentic-execution/02-the-boundary-of-agentic-execution>

Part II

The Agentic Execution System

03 — Inference, Context, and Action

04 — The Execution System

05 — State, Memory, and Provenance

06 — Authority and Delegation

07 — The Agent Control Plane

Inference, Context, and Action

Agentic execution begins with a simple loop.

The system receives a signal. It assembles relevant context. The model interprets the situation. The system selects an available action. A tool carries that action into an enterprise environment. The result becomes new context for the next decision.

This loop gives an agentic system its adaptive quality.

A traditional workflow follows a defined sequence. An agentic system can choose among a defined set of next steps as the situation evolves. It can read an unexpected carrier update, recognize a missing approval, identify conflicting customer data, or interpret an engineering incident with several plausible causes.

The value comes from its ability to operate in environments where language, information, and conditions vary.

The discipline comes from the environment surrounding that inference.

Context shapes the quality of inference. Authority shapes the range of action. Evidence shapes the confidence of the enterprise.

Context is assembled work

Enterprise context rarely arrives as one complete, coherent record.

A customer service case can involve an account system, order history, delivery records, inventory availability, service policy, prior communications, contract terms, and current operational conditions. An engineering incident can involve telemetry, deployment history, service ownership, runbooks, change records, customer impact, and active response activity.

The agentic system must assemble the information relevant to the specific outcome.

This requires judgment in its own right.

The system needs to identify the entity involved, retrieve the appropriate records, apply current policies, recognize conflicting information, and distinguish relevant evidence from surrounding noise.

A high quality context architecture makes these responsibilities explicit.

Context layer	Role in execution
Task context	Defines the objective, constraints, owner, and current stage of work

Enterprise context	Provides business records, workflow history, policies, and operating conditions
Domain context	Supplies the language, rules, measures, and judgments specific to the function
Tool context	Describes available actions, inputs, authority requirements, and expected results
Evidence context	Preserves sources, timestamps, confidence, and supporting records for each material claim

Context assembly determines what the agent can see. It also determines what the enterprise can later inspect.

A system that reaches an important conclusion should be able to show the evidence, policies, records, and events that informed that conclusion. This creates a direct connection between reasoning and accountability.

The model selects a next action

The model contributes a particular capability to the execution system.

It can interpret incomplete information, identify patterns across varied sources, translate a business objective into a sequence of possible actions, and adapt when new evidence changes the situation.

This gives it value in work that contains variation.

A customer request may arrive in natural language. A supplier document may contain inconsistent terminology. A technical incident may create several competing hypotheses. An operational exception may require the system to weigh timing, policy, customer importance, available resources, and financial exposure.

The model can bring these elements into a working interpretation.

The execution system then evaluates the proposed action against the operating envelope.

The available choices may include gathering additional evidence, requesting an approval, updating a record, sending a communication, initiating a transaction, escalating the task, or concluding the work.

The agentic loop therefore combines two forms of intelligence.

The model interprets the situation.

The enterprise architecture determines which action serves the defined outcome under the established conditions.

Tools carry action into the enterprise

A tool turns an intended action into an external effect.

Tools can retrieve information, create records, update systems, initiate workflows, communicate with customers, call specialized services, deploy software, or trigger another agentic capability.

Every tool creates a relationship between the execution system and an enterprise boundary.

That relationship needs to be designed with care.

A well formed tool interface makes the action understandable. It identifies the required inputs, expected outputs, accountable system, policy conditions, and result that the calling system should verify.

A tool also creates a moment of control.

The enterprise can inspect the request, confirm the identity of the acting system, evaluate the delegated authority, apply policy, record the action, and capture the resulting state.

This allows agentic capability to operate through controlled action paths.

The quality of a tool ecosystem therefore influences the quality of agentic execution. Clear interfaces, reliable records, visible policy boundaries, and meaningful response data create a stronger environment for delegation.

Observation turns action into learning

The system observes the result of every material action.

A carrier service may confirm a replacement shipment. A customer system may show a record update. A deployment platform may report a completed release. A finance platform may return a transaction identifier. An approval workflow may produce a decision, a condition, or a request for more information.

The result becomes part of the next context.

This creates a continuous cycle:

- 01 Signal
- 02 Context
- 03 Inference
- 04 Action
- 05 Observation
- 06 Context

The cycle can proceed until the outcome reaches acceptance, the work transfers to an accountable owner, or the system enters an intervention path.

Observation gives the enterprise more than operational visibility.

It creates an evidence trail through which the organization can evaluate quality, improve the workflow, refine policy, identify recurring exceptions, and understand the economics of execution.

Over time, this evidence becomes a source of operating learning.

The enterprise can see where agents create useful flow, where human judgment adds distinctive value, where context quality limits performance, where authority requires refinement, and where the workflow itself deserves redesign.

The architecture of disciplined adaptation

Agentic execution creates value through adaptation inside a defined operating system.

The system sees a changing environment. It interprets what matters. It selects an available next action. It acts through enterprise tools. It observes the result. It carries the work forward with evidence.

This is a powerful capability.

It also creates a design obligation.

Every agentic workflow needs a clear outcome, a trusted context environment, an explicit set of actions, accountable authority, durable state, visible observation, and a path toward accepted business value.

The model gives the system flexibility.

The architecture gives that flexibility purpose.

Read online: <https://sentient-review.com/perspectives/agentic-execution/03-inference-context-and-action>

The Execution System

An agentic workflow becomes an enterprise capability when it can carry work across time, systems, conditions, and exceptions.

That requires an execution system.

The execution system receives a signal, creates a durable task, assembles context, coordinates actions, manages approvals, records evidence, and guides the work toward an accepted outcome.

The model participates inside this system. It helps interpret the situation and select an available next action. The execution system provides the structure through which that action becomes reliable enterprise work.

A capable model creates possibility. A durable execution system creates operating continuity.

The task becomes the unit of work

Every agentic workflow needs a clear unit of work.

A task can begin with a customer request, a failed transaction, a production alert, a scheduled review, a policy exception, an engineering change, or a signal from another enterprise system.

The task carries the essential operating record:

- The intended outcome
- The accountable owner
- The initiating event
- The participating systems
- The relevant policies
- The current state
- The actions completed
- The approvals received
- The evidence gathered
- The next available action

This record gives the enterprise a shared view of the work.

A customer service leader can see the current condition of a high value case. A security leader can inspect the actions taken during an incident. A finance leader can review the authority and evidence behind a correction. An engineering team can understand how a proposed change progressed through validation and deployment.

The task becomes the container through which agentic execution remains visible and governable.

Execution moves through defined transitions

Enterprise work progresses through transitions.

A task may move from investigation to recommendation, from recommendation to approval, from approval to execution, from execution to verification, and from verification to completion.

Each transition can carry its own conditions.

An agent may gather evidence before proposing a remedy. A policy may require approval before a financial action proceeds. A deployment process may require testing before release. A service workflow may require confirmation before the system closes the case.

The execution architecture makes these transitions explicit.

Execution stage	Operating purpose
Intake	Captures the signal, objective, owner, and initial conditions
Investigation	Assembles context, evidence, policies, and relevant records
Decision	Selects an available next action inside the operating envelope
Approval	Routes consequential actions toward the appropriate authority
Execution	Carries approved actions into connected enterprise systems
Verification	Confirms the resulting external condition
Completion	Records the accepted outcome, economics, and learning
Intervention	Activates escalation, repair, pause, or termination when evidence calls for it

This structure gives teams a way to design the workflow around the outcome rather than around individual prompts or tools.

Time is part of the architecture

Agentic work can unfold across many different time horizons.

A customer response may require seconds. An approval may require hours. A procurement interaction may require days. A remediation program may extend across weeks. A recurring operational process may remain active for the life of a customer, asset, or contract.

The execution system carries work across these intervals.

It preserves the task state, schedules follow up activity, tracks pending conditions, receives new information, and routes the work to the next appropriate participant.

This continuity matters because enterprise work often unfolds across systems and people that operate on different clocks.

An agent may initiate a request in one system, await an external response, receive a new condition, call another system, and return the result to an accountable owner. The execution system holds the sequence together.

Recovery protects the outcome

Real work contains interruption.

A tool call can time out. A downstream service can return an incomplete response. A network connection can break after an external system accepts an action. An approval can expire. A data source can reveal a changed condition after the workflow has advanced.

Recovery is therefore an operating capability.

The execution system needs to preserve enough evidence to understand the task state and guide the next step. It needs clear transaction boundaries, repeatable actions, reconciliation paths, escalation routes, and human intervention points.

Consider a replacement shipment.

The agent prepares the request, checks policy conditions, receives approval, and sends the action to the fulfillment system. The fulfillment system accepts the request while the agentic environment experiences a connection failure.

The execution system can retrieve the fulfillment state, confirm the shipment record, update the task, and continue toward customer communication and verification.

This is how the architecture protects both the customer outcome and the integrity of the enterprise record.

Orchestration serves the outcome

An agentic workflow can involve models, deterministic services, approval systems, human owners, domain applications, and other agents.

Orchestration coordinates these participants.

The strongest design begins with the outcome and assigns each participant a clear role.

A deterministic service can calculate a price. A model can interpret an ambiguous customer request. A policy engine can evaluate an authorization condition. A human owner can approve a material commercial decision. An execution service can update the system of record. An assurance layer can verify the resulting state.

Each element contributes according to its strengths.

The result is an execution system where human judgment, machine inference, enterprise policy, and operational controls compose into one coherent path.

The executive implication

Agentic AI changes the way enterprises think about workflow design.

A workflow once centered on handoffs between people can become a coordinated environment in which humans and machines participate according to their comparative advantage.

The executive responsibility is to define the outcome, identify the work, assign authority, design the transitions, establish the evidence, and ensure that recovery serves the business result.

This is the work of agentic execution architecture.

It transforms a collection of model interactions into an operating system that can carry consequential work with discipline, continuity, and accountability.

Read online: <https://sentient-review.com/perspectives/agentic-execution/04-the-execution-system>

State, Memory, and Provenance

Agentic systems require a durable relationship with the past.

They need to understand the current condition of an active task, draw on relevant information from earlier work, and show the enterprise how a material claim, action, or outcome came into being.

These responsibilities are often gathered under the word memory.

They deserve greater precision.

State, memory, and provenance serve different operating purposes. Each carries its own ownership, access, lifecycle, and assurance requirements.

State preserves the work in progress. Memory informs future work. Provenance establishes the history of a claim, decision, or action.

This distinction gives enterprises a stronger foundation for agentic execution.

State preserves the current task

State is the durable operating record of work in progress.

It captures the current condition of a task: its outcome, initiating event, owner, relevant entities, actions completed, approvals received, evidence gathered, pending conditions, and next available action.

Consider an agentic workflow resolving a supply disruption.

The system may have identified the affected orders, checked alternate inventory, prepared options, requested approval for a customer remedy, and awaited a response from a supplier. Each event changes the state of the task.

The state record holds that sequence together.

It gives the system continuity across time. It gives a human owner an accurate view of the work. It gives assurance teams a way to reconstruct the execution path. It gives the enterprise a foundation for recovery when conditions change.

State therefore belongs close to the workflow and the outcome.

State element	Operating purpose
Task identifier	Establishes the durable unit of work
Outcome	Defines the result the task exists to produce
Current stage	Shows where the work sits in the execution path

Actions completed	Records material steps already taken
Pending conditions	Identifies approvals, responses, or data required for progress
Authority record	Connects actions to delegated permissions and accountable ownership
Evidence record	Preserves the basis for decisions and verification
Next action	Guides the workflow toward the next appropriate participant or activity

A strong state model turns a complex sequence of agentic work into an inspectable enterprise process.

Memory informs future work

Memory serves a different purpose.

It makes relevant information available beyond the life of one task.

A service system may retain a customer communication preference. An engineering agent may retain a validated runbook pattern. A finance workflow may retain a recurring reconciliation rule. A domain assistant may retain the language, measures, and decision logic that help it serve a particular function.

Memory can strengthen consistency, speed, and contextual understanding.

Its value depends on quality and governance.

Enterprise information changes. Customer preferences evolve. Policies are updated. Team structures shift. A technical pattern can become obsolete. A commercial agreement can expire.

Memory therefore needs an explicit lifecycle.

The enterprise should define how information enters memory, who owns its quality, which agents may use it, how updates occur, how corrections enter the system, and how retention aligns with policy and regulatory requirements.

This creates a living knowledge environment that supports execution while preserving institutional discipline.

Provenance gives claims their history

Provenance answers a different question:

“Where did this claim, decision, or action come from?”

An agent may state that a customer qualifies for a remedy. It may identify a supplier delay. It may recommend an engineering change. It may calculate a financial

exposure.

Each material claim should carry a visible path back to its source.

That path can include enterprise records, documents, policies, tool results, human approvals, model outputs, timestamps, and workflow events.

Provenance gives leaders the ability to inspect the basis for action.

It also gives the organization a way to handle disagreement.

Finance, Operations, Sales, and Customer Service can hold different views of the same commercial event. Each view may arise from a legitimate system, measure, and operating purpose.

A mature agentic system can present these claims with their sources, timing, and ownership. It can identify the appropriate decision maker. It can preserve the record of how the enterprise resolved the difference.

This creates a stronger form of enterprise intelligence: one that carries evidence, authority, and contestability alongside fluency.

Truth requires active stewardship

An enterprise record gains value through stewardship.

Data owners establish definitions. Functional leaders set business meaning. Policy owners define acceptable use. Technology teams create reliable access paths. Assurance teams validate the conditions under which information informs action.

Agentic execution depends on this stewardship.

A model can assemble information and produce an interpretation. The enterprise determines which sources carry decision weight, which conditions require validation, and which owner resolves a material conflict.

The quality of agentic action therefore rises with the quality of enterprise information architecture.

This includes data models, business semantics, access controls, source lineage, correction paths, retention practices, and ownership.

The work extends far beyond retrieval.

It becomes a discipline of making enterprise knowledge usable for accountable action.

The executive implication

State, memory, and provenance create the informational foundation for delegated execution.

State keeps the current task coherent.

Memory gives future work the benefit of relevant experience.

Provenance allows the enterprise to inspect the basis for a material claim or action.

Together, they support an agentic system that can act with continuity, context, and evidence.

The executive question is:

“Can the enterprise show what the system knew, why it acted, whose authority shaped the action, and what changed as a result?”

When the answer is clear, Agentic AI becomes a source of operational confidence.

When the answer is incomplete, the enterprise gains a powerful capability with an uncertain relationship to its own records, decisions, and accountability.

The architecture of agentic execution begins with action.

It earns trust through memory, state, and evidence.

Read online: <https://sentient-review.com/perspectives/agentic-execution/05-state-memory-and-provenance>

Authority and Delegation

Agentic AI changes the meaning of permission.

In a conventional enterprise system, permission often describes access. A person can view a record, edit a document, approve an expense, execute a payment, or deploy an application.

Agentic execution introduces a more complex form of participation.

A system can interpret a situation, choose among available actions, and carry work across several enterprise boundaries. It can do this at speed, at scale, and across time. The enterprise therefore needs to express authority with precision.

Capability describes what a system can do. Delegation defines what the enterprise authorizes it to do.

This distinction sits at the center of Agentic AI governance.

Authority has several dimensions

Delegated authority is more than a credential.

It includes the purpose of the work, the accountable principal, the actions available to the system, the resources it may use, the conditions that govern action, the duration of the authority, and the human owners who can intervene.

Authority dimension	Executive question
Purpose	Which business outcome does the delegation serve?
Principal	Which person, function, or policy grants the authority?
Scope	Which actions may the system initiate?
Conditions	Which evidence, thresholds, or approvals shape execution?
Resources	Which financial, data, tool, and infrastructure resources may the system use?
Duration	When does the authority begin, expire, or require renewal?
Escalation	Which conditions route work to an accountable human owner?
Intervention	Who can pause, override, or terminate the action path?

Evidence

Which record shows the authority, action, reason, and outcome?

These dimensions create an operating envelope.

A service agent may resolve a delivery exception up to a defined financial threshold. A finance agent may prepare a correction package and route it through an approval path. An engineering agent may investigate an incident, assemble a remediation plan, validate the proposed change, and submit it into a controlled deployment process.

Each form of authority serves a specific outcome and carries a specific consequence.

Delegation follows the work

The strongest authority architecture begins with the workflow.

It asks what work must happen, which decisions appear inside that work, which actions carry material consequence, and where human judgment adds distinctive value.

This creates a delegation gradient.

At the first level, the system gathers information and prepares insight.

At the next level, it recommends a decision.

It can then prepare execution for an authorized owner.

It can execute bounded actions through defined conditions.

It can also coordinate a sequence of actions across systems and functions, while assurance mechanisms observe the work and route exceptions toward accountable owners.

Each level requires a clear connection between authority and accountability.

The enterprise can expand the operating envelope as evidence demonstrates quality, reliability, and economic value.

This approach creates a path from assistance to increasingly capable execution.

Identity establishes the acting participant

Every consequential action needs a visible identity.

The enterprise needs to know which system acted, which workflow initiated the action, which delegated authority applied, which tools carried the action into external systems, and which owner remained accountable for the result.

This identity can travel through a chain of execution.

A customer service agent may call an inventory service. The inventory service may initiate a fulfillment workflow. The fulfillment workflow may engage a carrier

platform. Each participant carries a role in the broader action path.

The enterprise gains confidence when this chain remains visible.

Identity connects execution to ownership. Delegation connects identity to authority.

Evidence connects authority to outcome.

Together, they create a traceable system of action.

Boundaries create confidence

Authority boundaries help the enterprise move faster.

They clarify which work can proceed immediately, which work needs approval, which conditions require escalation, and which owners can intervene.

A well designed boundary reduces ambiguity for both humans and systems.

Consider a customer remediation workflow.

The system may identify a delayed order, assemble relevant records, evaluate service policy, and select from a set of approved remedies. A replacement shipment may proceed within available inventory and a defined commercial threshold. A larger concession may route to a service leader. A disputed contract term may route to a commercial owner.

Each action sits inside a clear boundary.

The system can carry routine work with speed. Human judgment focuses on material consequence, complex tradeoffs, relationships, and exceptions.

This is how agentic architecture creates a stronger combined operating model.

Accountability remains human

Machine execution can carry work. Human leaders retain responsibility for the purpose, policy, and consequence of that work.

Accountability sits with the leaders who define the outcome, authorize the operating envelope, own the relevant function, and govern the conditions under which the system participates.

This creates a shared responsibility model.

Technology leaders own the execution environment, reliability, security, and controls.

Functional leaders own the business outcome, workflow design, decision rights, and acceptance criteria.

Risk, legal, finance, and security leaders shape the policies and evidence required for consequential action.

Executive leadership determines the strategic purpose, investment priority, and authority model.

Agentic execution succeeds when these responsibilities compose into one operating system.

The executive question

The executive question is not:

“How much autonomy can the system have?”

The stronger question is:

“Which authority can the enterprise delegate, under which conditions, with what evidence, and who owns the consequence?”

This question gives leaders a practical path toward agentic capability.

It begins with an outcome. It defines the work. It assigns decision rights. It creates an operating envelope. It establishes evidence. It expands through demonstrated confidence.

The result is an enterprise that can place machine capability inside accountable authority.

That is the foundation of delegated action.

Read online: <https://sentient-review.com/perspectives/agentic-execution/06-authority-and-delegation>

The Agent Control Plane

The enterprise will operate many forms of Agentic AI.

Some agents will be built by internal engineering teams. Some will arrive inside enterprise software. Others will come from cloud platforms, model providers, specialist vendors, open source communities, and individual business teams.

Each may use different models, runtimes, tools, memory approaches, interfaces, and security assumptions.

This diversity creates an enterprise design challenge.

“What must remain consistent when the agents themselves vary?”

The answer is the agent control plane.

An agent control plane is the set of shared capabilities through which an enterprise can identify, authorize, observe, evaluate, constrain, govern, and intervene in agentic execution.

It creates coherence across a distributed environment.

The enterprise gains a mixed agent environment

Enterprise software already includes many applications, services, platforms, data stores, and user populations.

Agentic execution adds a growing population of computational participants.

A service agent embedded in a customer platform may call an internally built workflow. That workflow may retrieve information from a domain system, engage a specialized third party service, and initiate action through a cloud platform. Another agent may receive the resulting task and carry the next stage of work.

This creates a chain of participation across several owners and operating environments.

The enterprise remains responsible for the consequence.

It needs visibility into the participants, actions, authority, evidence, cost, and outcome across the chain.

The control plane provides this shared operating capability.

Common controls create enterprise coherence

A control plane can bring several core capabilities together.

Control capability	Operating purpose
Identity	Establishes each agent, service, workflow, and human participant
Ownership	Connects each agentic capability to accountable business and technical owners
Authority	Applies delegated permissions, action scopes, financial thresholds, and approval conditions
Tool governance	Defines which systems and actions each participant can access
Observability	Captures traces, events, tool calls, state changes, and resulting conditions
Evaluation	Measures quality, reliability, policy adherence, and accepted outcomes
Cost attribution	Connects model, infrastructure, tool, review, and recovery cost to the workflow and owner
Lifecycle	Governs creation, versioning, release, change, retirement, and archival
Incident response	Coordinates containment, investigation, recovery, and learning
Intervention	Enables pause, override, termination, and operating adjustment

These controls create a common language for agentic execution.

A functional team can own the business outcome. An engineering team can own the implementation. A platform team can provide shared runtime capabilities. A security team can define identity and policy standards. A finance team can establish economic measures.

The control plane connects these responsibilities around one operating environment.

Control follows the action path

The value of a control plane appears at the point of action.

A tool call carries an agentic decision into an enterprise system. The control plane can identify the actor, inspect the delegated authority, apply relevant policy, record the event, capture the result, and connect the action to the relevant workflow and owner.

This gives the enterprise a clear evidence path.

01 Agent

02 Authority

03 Tool

04 Action

05 Result

06 Outcome

The path can extend across many participants.

An internal agent may call a vendor service. The vendor service may initiate an enterprise workflow. The workflow may engage a domain application and create a customer facing effect.

The enterprise can retain accountability across this distributed sequence when identity, authority, evidence, and ownership travel with the work.

Governance becomes a shared service

The control plane functions as an enterprise service.

It gives teams a set of common capabilities they can use as they design agentic workflows close to their domains.

This approach supports innovation and discipline together.

Teams can build around the needs of customer service, engineering, finance, supply chain, sales, or operations. The enterprise can maintain common expectations for identity, authority, observability, assurance, and economics.

The result is an environment where local execution can evolve while enterprise accountability remains coherent.

The control plane also helps leaders understand the portfolio.

They can see which agentic capabilities exist, which outcomes they serve, which systems they access, which owners govern them, what evidence demonstrates their performance, and where the largest value or exposure sits.

This creates an informed basis for investment, intervention, and scale.

The control plane evolves with capability

Agentic technology will continue to evolve.

Models will become more capable. Tools will become richer. Protocols will mature. Vendors will introduce new execution environments. Business teams will discover new forms of agent participation.

The enduring requirement is control that evolves with the environment.

The enterprise can refine its authority architecture, expand shared observability, strengthen evaluation, improve cost attribution, and adjust lifecycle practices as new

capabilities enter the operating model.

The agent control plane provides the foundation for this adaptation.

It allows the enterprise to welcome a diverse agent population while preserving the coherence of its operating system.

The executive question

The executive question is:

“Which capabilities must remain consistent across every agent that participates in enterprise work?”

The answer begins with identity, authority, ownership, evidence, and intervention.

These elements allow an enterprise to build agentic capability across many systems, teams, and vendors while preserving accountability for the outcomes that matter.

The agent control plane is therefore more than a technology layer.

It is an enterprise capability for governing delegated action at scale.

Read online: <https://sentient-review.com/perspectives/agentic-execution/07-the-agent-control-plane>

Part III

Operability

08 — The Mixed Agent Enterprise

09 — Recovery as a Business Capability

Chapter 08

The Mixed Agent Enterprise

The enterprise will develop a mixed agent environment.

It will contain internally built agents, enterprise software agents, cloud platform agents, model provider agents, specialist services, open source components, and agents created by functions and teams close to the work.

This environment will grow through ordinary business activity.

Customer service platforms will add agentic capabilities. Engineering teams will build internal execution systems. Finance teams will introduce workflow agents. Software vendors will embed new forms of machine participation into their products. Business units will adopt specialist capabilities for their own operating needs.

The resulting enterprise will contain many computational actors.

Each will bring a different model, runtime, tool environment, identity pattern, memory approach, policy model, economic profile, and owner.

The challenge is to make this diversity coherent.

The enterprise can decentralize agent creation while maintaining accountability for enterprise outcomes.

Agent diversity follows enterprise reality

Large enterprises already operate through a diverse technology landscape.

They have systems of record, domain applications, shared platforms, external services, local tools, data products, workflow engines, and partner connections. Each exists because a function, process, regulatory condition, customer need, or technical decision created value through specialization.

Agentic AI enters this environment.

A single standard agent platform may support some forms of work. It will sit alongside the agents embedded in customer platforms, productivity suites, cloud environments, enterprise applications, and specialist services.

This diversity can create strength.

Agents built close to a domain can understand its workflows, vocabulary, measures, evidence, and operating conditions. They can serve a specific business outcome with greater precision. They can evolve with the people who own the work.

The enterprise gains value when it combines local domain expertise with shared control and architectural coherence.

Interoperability begins with clear boundaries

The mixed agent enterprise requires clear boundaries.

Each agentic participant needs a defined role. It needs an accountable owner. It needs a visible set of authorities, tools, data sources, and operating conditions. It needs a way to exchange work and evidence with other participants.

The enterprise can define a common work contract around these boundaries.

Work contract element	Operating purpose
Outcome	Identifies the result the agentic capability exists to produce
Owner	Connects the work to accountable business and technical leaders
Authority	Defines the actions, thresholds, and escalation conditions
Context	Identifies the enterprise information that informs execution
Evidence	Captures the sources, actions, approvals, and resulting conditions
Interface	Enables the exchange of requests, responses, status, and completion signals
Economics	Connects the work to cost, capacity, and business value

This contract enables coordination across different technical environments.

A customer service agent can request a supply chain assessment. A supply chain capability can return evidence, conditions, and available actions. An engineering agent can prepare a response to a production issue. A finance workflow can assess the commercial consequence.

Each participant contributes its domain capability. The work contract keeps the broader outcome coherent.

Ownership travels with the work

Ownership becomes especially important when agents coordinate across functions.

A delayed order may involve Customer Service, Operations, Finance, Supply Chain, and a carrier partner. Each function can own a valid part of the workflow. Each may operate through different systems and measures.

The agentic execution system needs to preserve this distribution of ownership.

The customer service function may own customer communication and remedy. Operations may own fulfillment conditions. Finance may own commercial thresholds. Technology may own execution reliability and system integration. A senior leader may own the final decision where the consequence crosses several functions.

The architecture can make these relationships visible.

It can show which agent carried which work, which authority applied, which function owned the decision, which systems provided evidence, and which outcome the enterprise achieved.

This avoids the common pattern in which an apparently unified experience hides fragmented accountability.

Shared controls support local execution

The mixed agent enterprise benefits from shared controls.

Identity, authority, observability, evaluation, cost attribution, incident response, lifecycle management, and intervention rights create a common operating language across the agent population.

These capabilities belong in the enterprise control plane.

They allow domain teams to build and adopt agents that fit their work while providing leaders with a consistent view of the operating environment.

A platform team can provide shared services for identity, policy, traces, evaluations, and cost. Functional teams can define the outcomes, workflow conditions, and acceptance criteria that give the agentic system business purpose.

This composition creates a balanced model.

Domain ownership remains close to the work.

Enterprise governance remains close to the shared risks, dependencies, and opportunities.

Coordination becomes a strategic capability

The mixed agent enterprise creates a new form of coordination challenge.

Agents can exchange requests, context, evidence, and work across systems. They can identify dependencies, prepare decisions, carry actions across functional boundaries, and route exceptions toward the appropriate owners.

The quality of this coordination depends on the quality of the underlying enterprise architecture.

Clear business definitions, reliable data, consistent identities, visible authority, strong interfaces, and accountable ownership create the conditions for productive machine

participation.

The enterprise gains a more adaptive operating environment.

Work can move toward the best available participant, whether that participant is a person, a deterministic service, a domain agent, or a coordinated group of agents and humans.

The objective is coordinated execution around an outcome.

The executive question

The executive question is:

“What must remain coherent as agentic capability expands across our enterprise?”

The answer includes outcomes, authority, ownership, evidence, interfaces, economics, and intervention.

The mixed agent enterprise will thrive through this coherence.

It will allow specialized capabilities to grow close to the work while preserving the shared operating principles that give the enterprise confidence in its own execution.

That is the path from agent diversity to enterprise capability.

Read online: <https://sentient-review.com/perspectives/agentic-execution/08-the-mixed-agent-enterprise>

Recovery as a Business Capability

Every consequential workflow needs a path through interruption.

A customer remedy can reach a fulfillment system while the response back to the agentic environment arrives late. A financial adjustment can enter a payment process while a downstream service reports an uncertain condition. An engineering change can pass validation while a production environment reveals a new dependency. An approval can expire while the surrounding business condition changes.

These events belong to the normal life of enterprise work.

Agentic execution increases the importance of recovery because the system can carry actions across many services, people, functions, and time horizons. It can move quickly from interpretation to action. It can also encounter changing conditions at every step.

Recovery is therefore a business capability.

The enterprise earns confidence through its ability to restore a coherent path toward an accepted outcome.

Recovery begins with a clear record

A workflow can recover when the enterprise can understand its current condition.

The task record needs to show the intended outcome, actions completed, actions awaiting confirmation, evidence gathered, authority applied, external systems involved, and next available step.

This record creates a common view for the agentic system, human owners, and assurance teams.

It answers essential questions:

- What action did the system attempt?
- Which external system received the request?
- What evidence confirms the resulting condition?
- Which action remains pending?
- Which owner can intervene?
- Which pathway restores the work toward an accepted outcome?

The task state becomes the foundation for recovery.

Every action creates a transition

An agentic workflow moves through a sequence of transitions.

A task enters investigation. Evidence enters the context environment. A model proposes an action. Policy evaluates the request. An approval occurs. A tool carries the action into an enterprise system. The system observes the resulting condition. The task advances toward completion.

Each transition deserves a clear design.

Transition	Recovery focus
Signal to task	Preserve the initiating event and intended outcome
Context to decision	Record the evidence, policy, and reasoning that informed the action
Decision to approval	Preserve the authority, conditions, and accountable owner
Approval to execution	Confirm the action request and target system
Execution to observation	Reconcile the external result with the task record
Observation to completion	Validate the accepted outcome and record the evidence
Exception to intervention	Route the work toward repair, escalation, pause, or termination

This structure gives the enterprise a practical way to design for continuity.

Recovery protects business integrity

The value of recovery appears when the system reaches a condition of uncertainty.

Consider a customer shipment replacement.

The agentic system identifies a delayed order, verifies inventory, applies the appropriate service policy, receives approval, and sends the replacement request to the fulfillment platform. The fulfillment platform accepts the request. A connection delay prevents the agentic environment from receiving confirmation.

The execution system can inspect the fulfillment record, retrieve the shipment identifier, update the task state, and continue toward customer communication and outcome verification.

This recovery path protects the integrity of the customer record, inventory position, financial remedy, and service experience.

The same principle applies across many domains.

An engineering system can reconcile a deployment state. A finance workflow can reconcile a transaction identifier. A procurement agent can retrieve a supplier response. A security workflow can confirm that a containment action reached the

intended environment.

Recovery turns uncertainty into a managed operating condition.

Human intervention carries decisive judgment

Agentic systems can carry a significant share of recovery work.

They can gather evidence, compare records, identify the likely state of an action, prepare options, and route the task toward the right owner. They can also execute approved recovery actions inside a defined operating envelope.

Human owners carry decisive judgment when the situation involves material consequence, ambiguous evidence, customer relationships, legal exposure, strategic tradeoffs, or a change in the purpose of the work.

The recovery architecture should make this handoff clear.

A human owner needs the task history, relevant evidence, authority record, current state, and available options. The agentic system can prepare this operating picture quickly. The accountable person can then make an informed decision and guide the next action.

This creates a strong combined system of machine continuity and human responsibility.

Recovery creates learning

Every recovery event produces information about the operating model.

It can reveal a data quality issue, a weak integration, an unclear policy, a missing approval path, an ambiguous ownership boundary, an unreliable tool response, or a workflow design that requires refinement.

The enterprise can use this evidence to improve the system.

It can strengthen context sources, refine authority conditions, improve tool interfaces, add validation, redesign escalation paths, and adjust the economics of review and recovery.

Recovery therefore creates a feedback loop.

01 Execution

02 Observation

03 Recovery

04 Learning

05 Architecture refinement

Over time, this loop improves the reliability and value of agentic execution.

The executive question

The executive question is:

“When the world diverges from the expected workflow, how does the enterprise restore a safe and accountable path toward the outcome?”

The answer requires durable task state, visible evidence, clear ownership, thoughtful transaction design, effective intervention rights, and a culture of operating learning.

These capabilities allow agentic systems to participate in consequential work with continuity.

Recovery becomes a source of confidence.

It enables the enterprise to expand agentic execution because it understands how work returns to coherence when conditions change.

Read online:

<https://sentient-review.com/perspectives/agentic-execution/09-recovery-as-a-business-capability>

Part IV

Outcome Economics

- 10 — Assurance at the Action Boundary
- 11 — Accepted Outcomes
- 12 — The Economics of Agentic Execution

Chapter 10

Assurance at the Action Boundary

Agentic systems create value when they produce accepted outcomes.

They create confidence when the enterprise can show how those outcomes were produced, assess the quality of the work, validate the resulting condition, and intervene when evidence calls for action.

This is the role of assurance.

Assurance sits at the action boundary, where model inference becomes a consequential enterprise effect.

A strong agentic system can explain what it did, show why it acted, prove what changed, and support accountable intervention.

Assurance follows the work

Agentic execution moves through a sequence of events.

A signal creates a task. Context informs a decision. Authority enables an action. A tool carries that action into an enterprise system. The external environment returns a result. The system evaluates the resulting condition against the intended outcome.

Assurance follows this full path.

Assurance layer	Executive purpose
Trace	Shows the task history, context, decisions, actions, and resulting conditions
Evaluation	Measures quality, reliability, policy adherence, and usefulness
Validation	Confirms the external effect and accepted outcome
Review	Engages accountable human owners for consequential, ambiguous, or exceptional work
Intervention	Enables repair, pause, override, escalation, or termination
Learning	Converts operating evidence into workflow, policy, and architecture improvement

Each layer contributes to operational confidence.

A trace can show which evidence informed a recommendation. An evaluation can assess whether the system handled a class of work with the expected quality. Validation can confirm that the customer record, financial transaction, deployment

state, or operational condition reached the intended outcome.

Together, they create an evidence system around agentic execution.

Acceptance creates the critical measure

The enterprise receives value when produced work becomes accepted work.

An agent may generate a customer response, prepare a technical change, assemble a financial analysis, or coordinate a service workflow. The enterprise evaluates that work against the criteria that define acceptance for the specific outcome.

The criteria can include quality, accuracy, policy alignment, timeliness, customer experience, safety, financial integrity, operational effect, and human review.

This creates a direct measure:

Accepted Outcome Rate = Accepted Outcomes ÷ Total Outcomes

The measure becomes meaningful when the enterprise defines acceptance in business terms.

A customer service system may measure resolved cases that meet policy and preserve satisfaction. An engineering system may measure changes that complete validation and improve production conditions. A finance system may measure reconciliations that produce accurate, approved corrections.

Acceptance connects agentic activity to enterprise usefulness.

Controls belong where actions occur

The most consequential assurance point sits at the action boundary.

An agent can interpret untrusted content, receive incomplete information, encounter an ambiguous instruction, or select an action with material consequence. The enterprise needs controls that evaluate the action request before it enters the target system.

These controls can examine identity, delegated authority, tool scope, financial threshold, policy conditions, data sensitivity, approval status, and expected result.

The action boundary becomes the point where enterprise policy meets agentic capability.

A system may propose a financial remedy. The policy environment evaluates the relevant conditions. The workflow records the authorization. The tool performs the action. The assurance layer captures the result and verifies the resulting condition.

This sequence turns governance into an operating capability.

Evaluation creates a path to scale

Agentic systems earn broader participation through evidence.

An enterprise can begin with a defined workflow, a narrow authority envelope, and clear acceptance criteria. It can observe the quality of execution, review the economics of accepted outcomes, refine the context environment, improve the workflow, and expand the system where evidence supports broader deployment.

Evaluation supports this progression.

It can measure performance across representative tasks, identify recurring exceptions, assess the effectiveness of human review, compare operating conditions over time, and reveal where policy or architecture requires refinement.

The objective is a living evidence system that guides investment and intervention.

Human judgment remains part of assurance

Agentic execution can carry significant portions of investigation, coordination, action, and verification.

Human judgment remains central in situations involving material tradeoffs, novel conditions, strategic relationships, legal or regulatory consequence, and changes in the purpose of the work.

Assurance defines these handoff points.

It ensures that the right person receives the right operating picture at the right moment: the task history, context, evidence, authority record, current state, available options, and expected consequence.

This creates a high quality relationship between machine execution and human responsibility.

The executive question

The executive question is:

“What evidence gives us confidence that this system is creating accepted outcomes within the authority and risk conditions we have established?”

The answer includes visible traces, meaningful acceptance criteria, independent validation, effective review, clear intervention rights, and a commitment to operating learning.

Assurance gives agentic execution a path from demonstration to enterprise capability.

It allows leaders to see what the system did, understand the result, refine the operating model, and scale the work where evidence demonstrates value.

That is how the enterprise preserves confidence at the action boundary.

Read online:

<https://sentient-review.com/perspectives/agnostic-execution/10-assurance-at-the-action-boundary>

Chapter 11

Accepted Outcomes

Agentic AI can produce a large volume of activity.

It can retrieve information, generate analysis, draft communications, prepare plans, coordinate tasks, initiate actions, and carry work across systems. These activities can demonstrate capability. They can also consume attention, infrastructure, review capacity, and management effort.

The enterprise creates value when that activity becomes accepted work.

Accepted work enters the operating system as a result that meets the defined standard for quality, policy, reliability, and business usefulness.

This creates the essential measurement path:

01 Activity

02 Output

03 Accepted Output

04 Business Outcome

The enterprise gains value through accepted outcomes, not through generated activity alone.

Activity creates the starting point

Activity is the visible motion of an agentic system.

The system processes a request, calls a tool, generates a response, routes a task, creates a record, or initiates an action. These events matter because they show the system participating in work.

They provide an early view of volume, throughput, latency, tool usage, and operational demand.

They also provide only part of the picture.

A high volume of generated work can create value when it advances toward accepted outcomes. It can also create review burden, rework, exception handling, customer confusion, or additional operating cost when the outputs fail to meet the required standard.

The enterprise needs to follow activity through the full path of work.

Acceptance creates a business boundary

Acceptance is the point at which the enterprise recognizes that the produced work is usable.

The definition varies by workflow.

A customer service response may achieve acceptance when it accurately addresses the issue, complies with policy, reaches the customer through the appropriate channel, and contributes to resolution.

An engineering change plan may achieve acceptance when it includes sufficient evidence, completes technical review, satisfies security conditions, and enters the approved execution path.

A financial analysis may achieve acceptance when it reconciles to authoritative records, meets accounting requirements, receives the appropriate approval, and supports a valid business decision.

Acceptance creates a shared standard across the people and systems participating in the workflow.

Acceptance dimension	Operating question
Quality	Does the work meet the required standard of accuracy, clarity, and usefulness?
Policy	Does the work align with the relevant business, financial, security, and regulatory conditions?
Evidence	Can the enterprise inspect the basis for the result?
Timeliness	Did the work arrive within the required operating window?
Outcome contribution	Did the accepted work advance the intended business result?
Review burden	How much human effort supported acceptance, correction, or escalation?

These dimensions connect agentic execution to the real work of the enterprise.

The accepted outcome rate

A useful operating measure is the accepted outcome rate.

Accepted Outcome Rate = Accepted Outcomes ÷ Total Outcomes

This measure reveals the share of system activity that becomes usable enterprise work.

The rate gains meaning when the enterprise defines its acceptance criteria with precision and applies them consistently.

A rising accepted outcome rate can indicate stronger context, better workflow design, improved authority boundaries, higher quality tools, clearer policies, or more effective assurance.

A declining rate can reveal a changing workload, weak evidence, poor context quality, unclear decision rights, increased exception volume, or an operating environment that requires intervention.

The measure becomes especially valuable when paired with the reason behind acceptance or rejection.

The enterprise can see where agentic work creates confidence, where human review adds value, where the workflow requires redesign, and where a particular class of task belongs with another form of execution.

Acceptance connects technology to operations

Accepted outcomes create a bridge between technology and business performance.

Technology teams can measure model quality, latency, tool reliability, execution traces, infrastructure consumption, and system resilience.

Functional leaders can measure customer resolution, transaction integrity, production reliability, service quality, cycle time, and throughput.

Finance leaders can measure operating cost, released capacity, avoided cost, contribution, and value capture.

Acceptance connects these views.

It establishes the point at which a technical output becomes a usable operating result. It allows every participant to measure the system against the same underlying outcome.

This creates a stronger basis for investment.

An enterprise can see which agentic workflows create a high share of accepted results, which ones require a larger review layer, which ones need additional context or authority refinement, and which ones have earned broader deployment.

Accepted outcomes improve work design

Acceptance data teaches the enterprise about the workflow itself.

It can reveal where handoffs create delay. It can show which decisions require deeper human judgment. It can identify recurring exceptions, missing context, ambiguous policies, unreliable integrations, and weak action paths.

This evidence enables redesign.

The enterprise can simplify the workflow, improve the context environment, adjust authority conditions, refine tools, strengthen evaluation, or reallocate work between people and machines.

Over time, accepted outcomes become a measure of operating maturity.

The question evolves from:

“How much work did the agent complete?”

To:

“How much of the work entered the enterprise as accepted value?”

The executive question

The executive question is:

“What does this system produce that the enterprise can accept, use, and trace to a meaningful business result?”

That question gives agentic investment a disciplined center.

It directs attention toward the outcomes that matter. It creates a common measure across technical and functional teams. It reveals the full pathway from machine activity to enterprise value.

Accepted outcomes are the foundation of agentic economics.

They show where execution creates useful work, where operating design requires refinement, and where the enterprise can scale with confidence.

Read online: <https://sentient-review.com/perspectives/agentic-execution/11-accepted-outcomes>

The Economics of Agentic Execution

Agentic AI creates a new operating cost structure.

A workflow can consume model inference, orchestration, storage, tool calls, data access, integration capacity, security controls, human review, recovery effort, and management attention. It can also release human capacity, improve quality, reduce delay, increase throughput, protect revenue, strengthen customer experience, and create new forms of operating leverage.

The economic question is therefore larger than model price.

The relevant unit is the full cost and value of an accepted outcome.

This is where Agentic AI moves from technical possibility to executive decision.

Token cost is one input

Model inference matters.

It shapes the cost of each interaction, the available reasoning capability, the latency of execution, and the infrastructure demand of the workflow. As models become more capable and efficient, they expand the range of work an enterprise can consider for machine participation.

The full economic picture includes more.

Cost element	Operating meaning
Human execution	The effort required to complete the workflow
Model inference	The cost of reasoning, generation, routing, and evaluation
Infrastructure	Compute, storage, networking, runtime, and observability
Tools and integration	The systems, APIs, platforms, and services that carry execution
Review	The human effort required for validation, approval, and exception handling
Recovery	The cost of reconciliation, remediation, and operating intervention
Transformation	The investment required to redesign the workflow and establish the operating environment

These elements together create the cost of producing usable work.

A low inference bill can coexist with a high review burden. A highly capable model can create value in a workflow that has a strong context environment, reliable tools, clear authority, and efficient recovery. The workflow determines the economic outcome.

Cost per accepted outcome

The central measure is:

Cost per Accepted Outcome = Human Cost + AI Cost + Infrastructure Cost + Review and Recovery Cost ÷ Accepted Outcomes

This measure gives the enterprise a common unit for comparison.

It can compare the current operating model with the redesigned workflow. It can compare two approaches to the same outcome. It can show how quality, automation, review, and infrastructure change together as the system evolves.

The measure becomes more powerful when paired with acceptance criteria.

A customer resolution system may measure the cost of a case that reaches an acceptable service outcome. An engineering system may measure the cost of an accepted change that improves a production condition. A finance system may measure the cost of a reconciled and approved transaction.

This creates a clear connection between technology investment and business execution.

Capacity needs a destination

Agentic execution can release human capacity.

A workflow that once required extensive manual search, drafting, routing, coordination, and record updates can become faster and more adaptive. People can focus on judgment, relationships, complex exceptions, strategic work, and higher value outcomes.

Released capacity creates economic potential.

The potential becomes value when the enterprise gives that capacity a destination.

01 Accepted Outcome

02 Capacity Released

03 Capacity Destination

04 Operating Consequence

05 Financial Consequence

The destination can include:

- Higher customer coverage

- Greater throughput
- Improved service quality
- Faster cycle time
- Avoided future hiring
- Reduced existing expenditure
- Increased contribution
- Stronger resilience

Each destination creates a different form of value.

A team may use released capacity to absorb growth. Another may redeploy it toward higher value customer work. A third may reduce external expenditure. An operations group may improve response time and reliability.

The economic consequence appears through this deliberate choice.

Review burden reveals operating maturity

Human review is a central economic variable.

Review can provide valuable judgment, protect against consequence, improve acceptance, and support learning. It can also reveal where context, workflow design, authority, assurance, or tool reliability requires improvement.

The enterprise can measure review burden across several dimensions:

Review measure	Executive meaning
Review rate	The share of work that receives human review
Review time	The effort required to assess an output or action
Correction rate	The share of work requiring repair before acceptance
Escalation rate	The share of work that reaches an accountable owner
Recovery cost	The effort required to restore the workflow toward the intended outcome
Acceptance rate	The share of work that becomes usable enterprise value

These measures show how the operating model evolves.

A workflow can begin with significant review and a narrow authority envelope. As evidence demonstrates quality and reliable execution, the enterprise can refine context, improve tools, strengthen assurance, and expand the scope of agentic participation.

This creates an evidence based path toward operating leverage.

Economics guides architecture

Economic design belongs inside architecture.

A complex multi agent environment can create valuable specialization. It can also create coordination cost, additional latency, increased inference demand, and greater assurance requirements.

A simpler workflow can create fast, reliable outcomes when the problem is well defined. A more adaptive agentic design can create value when the work contains variation, ambiguity, and cross system coordination.

The enterprise should choose the smallest execution system that produces the required accepted outcome with the desired quality, resilience, and economic result.

This is a design discipline.

It connects work design to model selection, tool architecture, infrastructure placement, human allocation, authority, assurance, and value capture.

The executive question

The executive question is:

“What does this accepted outcome cost, what capacity does it release, where does that capacity go, and what value does the enterprise realize?”

This question changes the conversation.

It moves beyond activity counts and technology demonstrations. It focuses the organization on the complete system of work, cost, acceptance, capacity, and business consequence.

Agentic execution earns investment when it creates a stronger economic path from demand to accepted outcome.

That is the economics of delegated action.

Read online:

<https://sentient-review.com/perspectives/agentic-execution/12-the-economics-of-agentic-execution>

Part V

The Executive Architecture

13 — Building Agentic Capability

14 — The Adaptive Agentic Enterprise

Chapter 13

Building Agentic Capability

Agentic capability grows through deliberate operating design.

The enterprise begins with an outcome that matters. It identifies the workflow that produces that outcome. It defines the work that humans and machines can carry. It establishes authority, context, execution, assurance, and economics. It gathers evidence. It expands through demonstrated confidence.

This is a disciplined path toward a more capable enterprise.

The objective is a stronger operating system for work shared between human judgment and machine execution.

Begin with a valuable outcome

The strongest starting point is an outcome with clear value and visible operating conditions.

The outcome may involve customer resolution, engineering reliability, financial reconciliation, supply chain coordination, employee service, commercial operations, or a high volume internal workflow.

A strong candidate has several characteristics:

Characteristic	Executive meaning
Material value	The outcome affects revenue, cost, capacity, quality, risk, resilience, or customer experience
Clear ownership	A functional leader owns the business result and acceptance criteria
Defined workflow	The enterprise can see the activities, handoffs, decisions, systems, and exceptions involved
Available evidence	Relevant context, policy, records, and outcome measures exist
Actionable scope	The workflow contains bounded actions that can operate inside explicit authority
Learning potential	The enterprise can measure results and improve the operating design over time

This approach directs investment toward meaningful work.

It also creates a practical setting for teams to develop shared capability across technology, operations, security, finance, and functional leadership.

Design the workflow around the outcome

Many existing workflows grew through years of policies, systems, handoffs, and local adaptations.

Agentic execution creates an opportunity to revisit the design.

The enterprise can map the current path from demand to accepted outcome. It can identify activities, waiting, rework, search, judgment, exceptions, approvals, and ownership boundaries. It can then redesign the workflow around the capabilities now available.

Some activities can move into deterministic automation. Some require model guided interpretation. Some call for human judgment. Some become shared work between an agentic system and an accountable owner.

The result is a clearer allocation of work.

The question becomes:

“If we designed this workflow today with human judgment and machine execution available together, how would it operate?”

Establish the operating envelope

Every agentic workflow needs an operating envelope.

The envelope defines the outcome, authority, actions, context, assurance, and economics that shape participation.

It answers:

- Which outcome does the system serve?
- Which actions can it initiate?
- Which policies and thresholds apply?
- Which information informs its decisions?
- Which conditions route work toward human review?
- Which evidence establishes success?
- Which cost and value measures guide expansion?

This gives teams a shared design object.

Technology leaders can build the execution environment. Functional leaders can define the work and acceptance criteria. Security, risk, legal, and finance leaders can shape the conditions for action. Executive sponsors can guide investment and resolve cross functional decisions.

The operating envelope turns agentic capability into an accountable intervention.

Build shared enterprise capabilities

Individual workflows create immediate value. Shared enterprise capabilities allow that value to compound.

These capabilities include identity, authority, context access, tool governance, durable execution, observability, evaluation, cost attribution, incident response, and lifecycle management.

They form the enabling environment for a growing portfolio of agentic workflows.

The enterprise can provide these capabilities as common services while allowing domain teams to build close to their work.

This creates an effective balance.

Customer Service can develop agentic resolution workflows. Engineering can develop capabilities for incidents, changes, and reliability. Finance can develop reconciliation and analysis workflows. Supply Chain can develop coordination capabilities.

Each domain can serve its own outcomes while operating through shared standards for accountability and evidence.

Create an evidence based expansion path

Agentic capability expands through evidence.

The enterprise can begin with a focused workflow and a clear authority envelope. It can measure accepted outcomes, review burden, recovery patterns, cost per accepted outcome, capacity released, and economic consequence.

It can then strengthen the operating design.

Context sources can improve. Tools can become more reliable. Authority can expand within defined conditions. Assurance can become more precise. The workflow can serve a broader volume or a more consequential outcome.

This progression creates an operating path:

- 01 Pilot
- 02 Evidence
- 03 Refinement
- 04 Expansion
- 05 Enterprise capability

Each stage produces learning about the work, the architecture, the economics, and the organization.

The enterprise gains the ability to make informed decisions about scale, repair, pause, and resource allocation.

Shape the executive system

Agentic AI requires active executive coordination.

The work crosses technology, operations, security, finance, legal, risk, data, and functional leadership. Each contributes a part of the operating environment.

The executive system creates coherence across these responsibilities.

It establishes the strategic outcomes that matter. It assigns accountable owners. It guides investment. It resolves authority questions. It reviews evidence. It directs released capacity toward value. It shapes the architecture through which the enterprise learns and adapts.

This is the leadership work beneath Agentic AI.

The technology matters deeply. The operating model determines how that technology becomes enterprise capability.

The executive question

The executive question is:

“Which outcomes deserve agentic execution, and what operating system will allow the enterprise to pursue them with confidence?”

The answer begins with a valuable workflow.

It grows through deliberate work design, explicit authority, shared capabilities, evidence based expansion, and executive coherence.

This is how enterprises build agentic capability that compounds.

They create a stronger relationship between human judgment, machine execution, and economic value.

Read online: <https://sentient-review.com/perspectives/agentic-execution/13-building-agentic-capability>

Chapter 14

The Adaptive Agentic Enterprise

The agentic enterprise is an operating condition that evolves with capability.

Models improve. Tools become richer. Context environments become more connected. Protocols mature. Enterprise systems expose new forms of participation. Teams discover new ways to combine human judgment, deterministic execution, and agentic coordination.

The enterprise must evolve with this environment.

Its enduring advantage comes from its ability to redesign work, assign authority, govern execution, measure accepted outcomes, direct capacity toward value, and adapt the operating model as evidence grows.

The highest performing enterprise combines human judgment, machine intelligence, accountable execution, and economic discipline into one adaptive system.

The future is an operating design question

The future of Agentic AI will look different across industries, functions, and enterprises.

A healthcare organization will develop its own approach to evidence, safety, clinical judgment, and regulatory responsibility. A financial institution will shape authority around controls, transactions, customer trust, and risk. A manufacturer will focus on reliability, supply, quality, capacity, and operational continuity. A technology company will apply agentic execution across engineering, customer experience, commercial operations, and internal productivity.

Each enterprise will pursue a different path.

The common question remains:

“How does this organization redesign itself as machine participation expands across the work?”

The answer belongs in the operating model.

It includes work design, authority, management, architecture, governance, economics, and executive leadership. These elements compose into the environment through which agentic capability creates durable value.

Work allocation will continue to evolve

Human and machine participation will shift as capability develops.

Some work will remain deeply human because it depends on relationships, legitimacy, nuanced judgment, strategic choice, creative direction, and accountable leadership.

Some work will become increasingly agentic because it benefits from rapid context assembly, pattern recognition, coordination, continuous observation, and bounded execution.

Many workflows will remain combined systems.

A human leader may set the objective, define the operating envelope, assess a material exception, and guide a strategic relationship. An agentic system may gather evidence, coordinate the workflow, carry routine actions, surface conditions that require attention, and provide the operating picture required for judgment.

This allocation can change over time.

The enterprise gains value when it treats the allocation as a living design decision.

Management becomes an architecture of outcomes

As machine execution carries more activity, management can concentrate more deeply on outcomes, capability, resources, systems, exceptions, and people.

Managers can spend less time routing routine work and more time defining purpose, developing talent, shaping decision rights, resolving complex tradeoffs, allocating capacity, and improving the operating system.

This creates a new management architecture.

It asks leaders to understand the relationship between human work, machine execution, enterprise context, authority, quality, economics, and accountability.

It also gives leaders a richer operating picture.

They can see where work moves, where decisions occur, where capacity is released, where exceptions accumulate, where agentic systems create value, and where intervention can strengthen the enterprise.

Management becomes a discipline of designing and governing the combined execution environment.

The enterprise becomes more visible to itself

Agentic systems can create a more connected view of enterprise activity.

They can gather information across systems, identify dependencies, assemble evidence, surface exceptions, track commitments, and support coordinated action across functions.

This creates a powerful opportunity.

The enterprise can develop a richer understanding of its own work, its resources, its decisions, its constraints, and its outcomes.

The quality of this understanding depends on the quality of the underlying information architecture, business definitions, ownership, provenance, and assurance.

A mature enterprise can make these elements visible together.

It can show the flow from demand to outcome. It can show which capabilities serve the work. It can show where authority resides. It can show the evidence behind a decision. It can show the economic consequence of an intervention.

This creates a stronger foundation for executive judgment.

The North Star remains adaptive

The agentic enterprise has a North Star.

It is an organization where human judgment and machine execution operate as one coherent system.

Work is allocated according to comparative advantage. Authority is explicit. Agents act inside enforceable operating envelopes. Context is governed. Execution is durable. Assurance follows action into the external environment. Measurement tracks accepted outcomes. Capacity has a deliberate destination. Economic value is visible.

The North Star remains adaptive because capability will continue to change.

An enterprise can expand the work carried by agents, refine the authority they receive, strengthen the context and assurance environment, redesign management practices, and direct the resulting capacity toward new forms of value.

The enduring principles remain constant:

- Accountable ownership
- Explicit authority
- Trusted evidence
- Accepted outcomes
- Economic discipline
- Continuous adaptation

These principles give the enterprise a stable foundation for a changing technology environment.

The final executive choice

Agentic AI gives leaders a choice.

They can treat it as a collection of tools distributed across the organization.

Or they can use it to redesign the operating system through which work, authority, evidence, and value move across the enterprise.

The second choice requires greater discipline.

It also creates the greater opportunity.

The enterprise that succeeds will build a coherent environment where agents can participate in meaningful work, people can apply judgment where it creates distinctive value, and leaders can see how both contribute to economic performance.

This is the promise of Agentic AI.

It is a new architecture for delegated action.

It is an adaptive enterprise built around outcomes, accountability, and the deliberate composition of human and machine capability.

Read online:

<https://sentient-review.com/perspectives/agentic-execution/14-the-adaptive-agentic-enterprise>

Connected Work

Related Framework

FW 18 · Operationalized

The Delegated Action Architecture

<https://sentient-review.com/frameworks/the-delegated-action-architecture>

Related Field Note

FN 21 — The Agent Is Not the System

<https://sentient-review.com/field-notes/the-agent-is-not-the-system>

Connected Practice

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). Agentic Execution: The Architecture of Delegated Action in the Enterprise. Volume 01, Publication 02. Version 1.0.

Canonical URL

<https://sentient-review.com/perspectives/agentic-execution>

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