

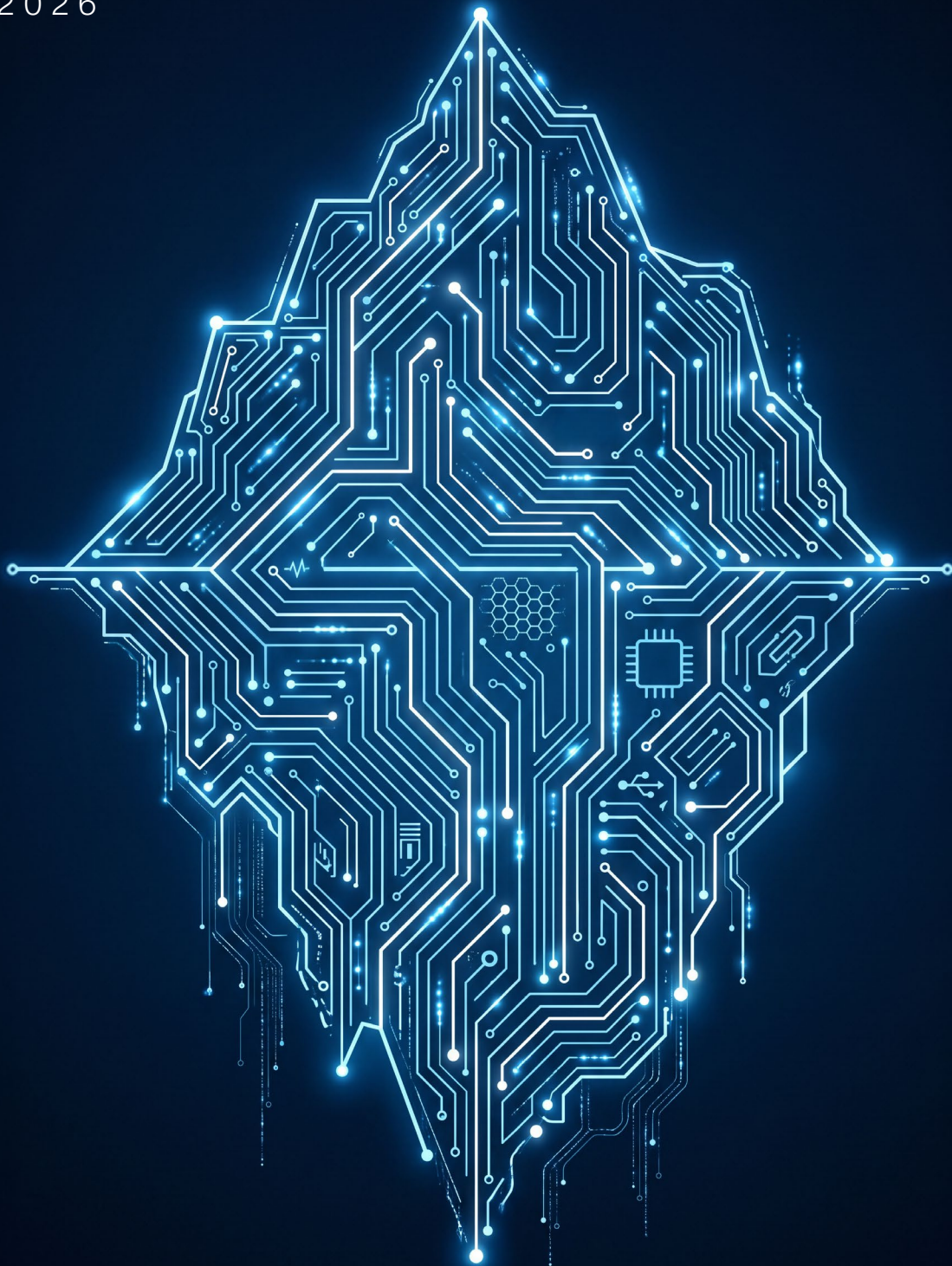
In collaboration  
with Capgemini



# AI Agents in Action: A Playbook for Trusted Adoption, Authorization and Scaling

INSIGHT REPORT

MAY 2026



# Contents

Foreword	3
Executive summary	4
Introduction	5
1 Agent guidelines	7
1.1 Establishing a shared language for autonomy, authority and consequence	8
1.2 Allocating decision rights and accountability across the life cycle	9
1.3 Defining when agentic systems are the appropriate solution	9
1.4 Sequencing adoption and prioritizing early use cases	10
1.5 Deployment contexts and baseline governance	11
1.6 Defining the human-agent operating model	12
1.7 From enterprise guidelines to deployment authorization	12
2 ACAP: The Agent Capability and Authorization Profile	14
2.1 ACAP structure and how to use it	16
2.2 System design and assessment	17
2.3 Prepare and deploy	23
2.4 Monitor and improve	27
Conclusion	30
Appendix: ACAP summary playbook	31
Contributors	34
Endnotes	37

## Disclaimer

This document is published by the World Economic Forum as a contribution to a project, insight area or interaction. The findings, interpretations and conclusions expressed herein are a result of a collaborative process facilitated and endorsed by the World Economic Forum but whose results do not necessarily represent the views of the World Economic Forum, nor the entirety of its Members, Partners or other stakeholders.

© 2026 World Economic Forum. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, including photocopying and recording, or by any information storage and retrieval system.

# Foreword



**Volker Darius**  
Chief Executive Officer,  
Capgemini Invent



**Cathy Li**  
Head, Centre for AI Excellence;  
Member of the Executive  
Committee, World Economic Forum



**Stephan Mergenthaler**  
Managing Director,  
Chief Technology Officer,  
World Economic Forum

Artificial intelligence (AI) agents have left research laboratories to become a permanent fixture in organizational workflows. This shift has outpaced our existing governance frameworks. Where the first publication in this series laid the conceptual foundations, and the second introduced assessment and classification tools, this third report tackles a decidedly practical question: how can an organization, in concrete terms, delegate authority to an automated system while remaining fully accountable for its actions?

The answer to this question is not primarily technical. It is organizational. What we have observed in our work with members of the AI Global Alliance's Safe Systems and Technologies working group is that organizations grasp what an agent can do, but they struggle to define what it should be authorized to do in context. This gap between capability and authorization is both the central challenge to large-scale adoption and the rationale of this playbook.

The governance of AI agents cannot be an afterthought. It must be built in at the design stage, formalized before deployment and

actively maintained in production. Only then can organizations move from isolated pilots to portfolios of agents deployed with confidence, enabling the expected benefits of automation to be realized without compromising internal controls, regulatory compliance or stakeholder trust.

This report is the result of intensive collaboration between practitioners, technical teams, risk management functions and legal experts from a variety of industries. It does not offer universal answers, but rather a structured, adaptable and auditable authorization model: the Agent Capability and Authorization Profile (ACAP). This model translates corporate policy into an operational mandate for each deployment.

We hope this playbook will provide a practical resource for teams seeking to integrate AI agents into their organizations responsibly and contribute more broadly to the development of shared standards for the governance of delegated agency. As Andrej Karpathy has framed it: the decade of AI agents has begun.<sup>1</sup> The way we lay the foundations today will determine the trust we can place in them tomorrow.

# Executive summary

The Agent Capability and Authorization Profile is a practical framework designed to help organizations adopt, authorize and scale AI agents.

Artificial intelligence (AI) agents introduce new adoption challenges for organizations, particularly in defining the conditions under which they are authorized to act and in ensuring that this authority is enforced as systems evolve in operation.

This playbook introduces the Agent Capability and Authorization Profile (ACAP) as a deployment-level authorization instrument for agentic systems. The ACAP connects enterprise delegation policy, system design and operational oversight in a single, repeatable workflow that makes delegated action auditable, enforceable and reviewable.

Through the ACAP, the playbook establishes a structured operating model for AI agent deployment: validating readiness through evaluation, enforcing boundaries through

technical controls and maintaining authorization through production monitoring.

Many agents in a portfolio may share the same foundation model, therefore, a single model-level vulnerability can propagate across an organization's entire agent estate simultaneously, reinforcing the need for deployment-level authorization and monitoring for each instance.

It provides three core elements: clear agent guidelines, enforceable authorization profiles, and a phased adoption life cycle from pilot to scale. This establishes a clear, assignable model for delegation, a structured path from pilot to production and a consistent framework for scaling from individual deployments to portfolios of agents.

# Introduction

AI agents are advancing faster than governance – authorization, not capability, is now the critical bottleneck.

“ AI agents are entrusted to act on behalf of a person or an organization in pursuit of defined objectives, under constraints set by policies, tools and access controls.

Artificial intelligence (AI) agents have rapidly moved from experimentation to early implementation, and some observers speak of a “decade of AI agents”<sup>2</sup> ahead. That long time horizon signals both significant opportunity and the reality that “it is still early days”<sup>3</sup> for agent adoption and deployment.

This report addresses a practical gap that emerged in early deployments. Organizations are increasingly able to describe what an agent is capable of, but often lack a consistent way to determine what it is authorized to do within a specific workflow. This playbook introduces the Agent Capability and Authorization Profile (ACAP) as a deployment-level authorization record for agent systems and explains how it is created, approved and maintained across the adoption life cycle. The model will evolve and mature alongside the field and should be understood as an early contribution towards the standardization of deployment-level authorization for agentic systems. A shared structure for authorization can deliver immediate operational value while providing a foundation for stronger assurance over time.

Onboarding an agent can, in some ways, be compared to onboarding a new employee. The process involves assessing capabilities, assigning responsibilities, granting access, defining supervision and adjusting the scope of action as performance and trust evolve. This is a structural comparison only, as agents lack clear legal status, moral responsibility and reputational incentive – governance is built to offset these gaps. The comparison is useful because the term “agent” has long referred to a person who acts on behalf of an organization within defined bounds. Real estate agents, call centre agents and public sector agents operate with delegated authority, clear procedures and escalation when exceptions arise.

AI agents are described as agents for the same reason. They are entrusted to act on behalf of a person or an organization in pursuit of defined objectives, under constraints set by policies, tools and access controls. Unlike traditional software, they can interpret context and determine what to do next in situations that are not fully specified in advance. This combination of non-determinism and delegated action makes adoption less like

installing software and more like introducing a new organizational actor that requires structured onboarding, explicit decision rights and continuous operational oversight.

In this playbook, authorization is defined at the level of the agent solution. Where multiple agents collaborate, or where an orchestrating agent coordinates sub-agents, the deployment is treated as a unified system for authorization purposes, even as its internal composition evolves. The ACAP therefore reflects the combined architecture, shared tool access, coordination logic and end-to-end authority boundaries of a given solution, including component-level specifications for agent roles, tool access and memory.

This report is the third in a three-part series on AI agents; deeper contextual understanding can be gained by reviewing the earlier publications. *Navigating the AI Frontier: A Primer on the Evolution and Impact of AI Agents*<sup>4</sup> established shared definitions and described the evolution of different types of agents. The second paper, *AI Agents in Action: Foundations for Evaluation and Governance*,<sup>5</sup> introduced agent classification, evaluation, risk assessment and proportionate governance. This third and final paper in the series builds on those foundations by providing a structured and action-oriented framework for adoption and integration.

The playbook draws on working group meetings, workshops and deep-dive interviews conducted through the Safe Systems and Technologies working group of the AI Global Alliance<sup>6</sup> to identify emerging practices, common pitfalls and operational patterns across agent deployment. It is written for practitioners who are piloting, evaluating and deploying AI agents.

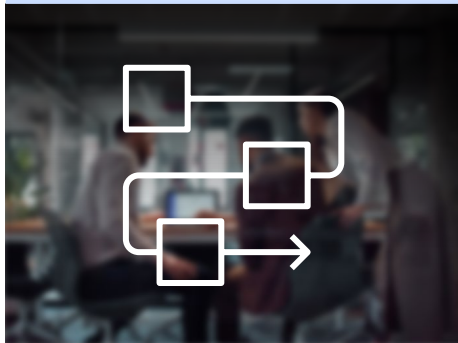
The frameworks introduced here will require revision as the technology matures. Updates to the ACAP model may be triggered by new capabilities it cannot yet accommodate, failures that expose governance gaps or regulatory changes that redefine compliance.

In the decade of AI agents, starting with a clear model for authorization and oversight is a prerequisite for maintaining trusted deployment at scale.

## How to use this playbook

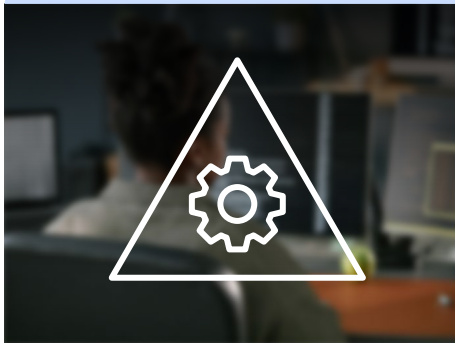
This playbook is designed to be used as a practical operating guide for moving from agent pilots to authorized, monitored deployment. It ties every step of adoption to the life cycle of agent implementation and uses the ACAP as a living authorization record that is created, completed and updated across the life cycle.

At a high level, the playbook has three building blocks:



### 1 Agent guidelines (Chapter 1)

Defines the enterprise policy for delegated agency, including shared terminology, decision rights, deployment criteria and baseline governance expectations.



### 2 The ACAP (Chapter 2)

Serves as the written authorization for a specific agentic solution within a defined workflow, recording its scope, operating context, authority, controls, required evidence, accountable owners and re-authorization cadence.



### 3 The adoption life cycle (subsections 2.2, 2.3 and 2.4)

Operationalizes authorization through three phases: 1) system design and assessment, 2) prepare and deploy, and 3) monitor and improve. Each phase specifies what must be designed, evidenced, signed off and continuously maintained. Progression between phases is governed by phase gates, which require predefined minimum requirements to be met before moving forward.

### How to apply it in practice:

- Start with agent guidelines if your organization needs a shared language and clear decision rights for delegation.
- Establish an ACAP for a single deployment if you are running pilots and need to make scope, authority, controls and evidence explicit before production release.
- Use the life cycle phases as your execution plan to move from design to controlled release, to monitored operation and change.

A one-page overview of the complete ACAP steps is provided in the Appendix of this report. The overview is intended as a quick reference guide, but the steps should be carried out and maintained throughout the life cycle phases described in Chapter 2.

1

# Agent guidelines

Introducing agentic systems requires the right organizational conditions for effective, responsible deployment.



Successful adoption of AI agents starts with a shared understanding of what is being built and deployed, as well as who is responsible for each part of the process. Early pilots have shown that many challenges are linked to unclear terminology, misaligned expectations and gaps between technical and operational teams.<sup>7</sup> Establishing common language and clearly defined roles early is an essential step in the adoption journey.

Agent guidelines are intended to serve as a new enterprise-level instrument, endorsed by leadership and applied across all deployments. They establish shared language and baseline conditions under which agents may be introduced.

By defining these elements once at the organizational level, the guidelines ensure that deployment decisions are comparable, that responsibility does not fragment across teams, and that agent authority is expanded based on evidence rather than ad hoc practice.

## 1.1 Establishing a shared language for autonomy, authority and consequence

“ Shared language and definitions at the organizational level help ensure that technical, operational and governance functions evaluate deployments against the same parameters.

Agent guidelines begin by defining the operational concepts that govern delegation. Establishing shared terminology and documented governance concepts at the enterprise level aligns with AI management system expectations for policy, roles and operational controls, such as those defined by ISO 42001.<sup>8</sup> These definitions help determine how deployments are assessed, what controls are required and when human intervention is mandatory.

Core concepts include:



**Autonomy:** The degree to which an agent determines how an objective is pursued, sets its sub-goals and executes actions without real-time human direction.



**Authority:** The permissions granted to act, such as read, write, execute, communicate or transact across systems and data. Includes representational authority to act on the organization's behalf and, where explicitly granted, decisional authority to make binding commitments.



**Consequential events:** Outputs that have significant impact – legal, financial, security (including impersonation), safety, ethical, reputational or customer-facing interactions – whether within the organization or externally. These effects can extend beyond the system itself and therefore require explicit checkpoints and control gates throughout the process.



**Operational context:** The workflow, users, systems, data classes and jurisdictions within which the agent acts.



**Boundaries:** The explicit limits of intended use, including prohibited actions and escalation triggers.

Shared language and definitions at the organizational level help ensure that technical, operational and governance functions evaluate deployments against the same parameters and that authorization decisions remain consistent across the organization.<sup>9</sup> This foundation and shared context are helpful as organizations start adopting multiple agents.



## 1.2 Allocating decision rights and accountability across the life cycle

“ Developers and engineering teams collaborate with data owners to monitor quality, define the system’s technical characteristics, constraints and enforceable controls.

Agent guidelines specify that delegation of authority to an agent requires clear allocation of human ownership associated with introducing and overseeing agents in practice. This mirrors widely used risk governance practices such as those defined by the National Institute of Standards and Technology (NIST) AI Risk Management Framework, which emphasizes defined roles, accountability, and oversight across the AI life cycle.<sup>10</sup> The objective is to ensure that every agent deployment has named owners for the following:

- Defining the agent’s operational role and scope
- Granting and expanding its authority based on risk
- Assessing and accepting the associated risk
- Supervising consequential events
- Approving material changes in capability, access or context

In practice, these decision rights remain constant throughout the agentic system life cycle, but responsibility for exercising them sits across multiple functions and often spans multiple agentic systems. In a typical deployment:

- **Adopters or deployment owners** define the operational mandate and remain accountable for the agent’s outcomes in the workflow.

- **Developers and engineering teams** collaborate with data owners to monitor quality, define the system’s technical characteristics, constraints and enforceable controls, leading to robust guardrails.
- **Subject matter experts (SMEs)** determine what acceptable performance and output look like in context and interpret behavioural signals during operation, forming the basis for evaluation.
- **Risk, compliance and legal functions** approve the delegation within the organization’s control framework and set the required level of governance.
- **Human supervisors** exercise real-time checkpoints for consequential events and intervene when boundaries are reached.
- **Human resources and change-management functions** align role evolution, training and incentives with the human/agent operating model.

The agent guidelines establish these decision rights at the enterprise level and require them to be explicitly assigned for each deployment. This prevents fragmentation of responsibility and ensures that the authority to act remains aligned with accountability for outcomes, from initial authorization to in-production supervision.

## 1.3 Defining when agentic systems are the appropriate solution

A core component of the agent guidelines is to determine if deploying an AI agent is the right approach to begin with. Establishing deployment criteria early prevents agentic solutions from being applied where simpler, more predictable solutions would be safer and easier to maintain.

The key test is whether success can be defined without specifying every step. If the goals and constraints can be clearly described but the steps to reach them are not, an agentic approach is likely appropriate. This distinction has nothing to do with complexity, as an agent can use simple or deterministic logic. What matters is whether the execution is pre-specified and fixed. Fully pre-scripted workflows are automation, not agents.

Agents add value when the problems require ongoing orchestration across multiple steps, managing their sequence and control loop. This includes conditional branching, revisiting of earlier decisions or managing intermediate states when necessary.

They are also useful when the state, memory or context evolves and shapes downstream decisions. If retaining prior interactions, intermediate findings or emerging hypotheses change what happens next, an agentic architecture is more effective.

Finally, agents are appropriate when the system must decide how to act: dynamically select, sequence or combine tools, or coordinate across multiple systems, actors, human handoff and escalation points.<sup>11</sup>

## 1.4 Sequencing adoption and prioritizing early use cases

Agent guidelines define how use cases should be prioritized to support adoption, particularly for organizations at an early stage of adoption. Initial deployments should strengthen organizational capability and confidence while keeping operational, reputational and business risk exposure contained. This typically means selecting workflows where errors are low in consequence and can be reversed without lasting impact, allowing teams to learn in real conditions without causing material harm.

Equally important is the presence of clear reference points for both performance and value. Agents can only be evaluated, governed and improved when acceptable outcomes are observable and shared across technical, operational and control functions. Well-defined examples, benchmarks or “gold standards” make it possible to assess behaviour consistently, calibrate supervision and establish credible promotion gates for expanded authority.

Early use cases should deliver value quickly and be deployable in weeks using existing infrastructure and operating models, so organizations can build experience with supervision, evaluation and

authorization before moving to more complex, higher-impact contexts.

In practice, not all agent deployments emerge through a structured, top-down adoption process. As low-code and agent-building platforms become widely available, employees and teams may experiment with creating agents directly within their workflows. Some of these prototypes may remain local tools, while others may diffuse organically across teams if they prove useful.

Organizations should therefore establish clear signals for when such bottom-up innovations transition into formally governed deployments. When an agent begins to interact with enterprise systems, operate across teams or perform consequential events, it should enter the structured life cycle described in this playbook and be documented through an ACAP. Organizations should document the expected value case – efficiency gains, capability expansion or risk reduction – before committing to deployment, so that governance overhead can be weighed against anticipated benefit.

# 1.5 Deployment contexts and baseline governance

“ For global or cross-border use cases, concepts such as an “agent passport” may be required to enable agents to present verifiable, machine-readable claims of identity, authorization and jurisdiction-specific compliance.

Agent guidelines must account for the structural context in which agents are deployed. The same agent can have very different risk and governance implications depending on which organizational or technical boundaries it crosses.

The core distinction is whether agents operate within an established trust boundary or across shared or open ecosystems. Deployment context determines which identity, authorization, oversight and accountability mechanisms can be relied upon and where additional governance is required.

Single-organization deployments typically operate within existing trust boundaries and can use internal identity and access management and controls. Multi-organizational or multi-platform deployments introduce new trust assumptions and governance responsibilities, particularly where no single authority exists.

Organizations should therefore segment deployment contexts by risk and assurance level. These tiers define baseline governance expectations that must be specified at the deployment level.

As agent ecosystems mature and tasks are distributed across shifting combinations of capabilities, this context tiering may require component-level authorization to prevent governance blind spots.

### Deployment context tiers:



#### Level 1: Single-organization deployment

Agents operate within internal systems and identity frameworks. Governance can use existing identity and access management (IAM) and internal controls.



#### Level 2: Multi-organization, single-platform

Agents interact within a controlled ecosystem governed by shared platform rules. Governance depends on transitive trust mediated through platform-enforced standards.



#### Level 3: Multi-platform, cross-boundary interaction

Agents interact across organizations and platforms without a central authority, requiring stronger mechanisms for identity, authorization, auditability and compliance.

In multi-organizational and multi-platform settings, agents representing different organizations may interact under different authorization models, operating constraints and accountability structures. For example, a buyer agent and a vendor agent acting for distinct organizations may operate under different authority limits or approval rules, creating ambiguity when those authorizations do

not align. While this playbook does not define a universal conflict resolution mechanism, it highlights this as an emerging governance challenge and reinforces the importance of interoperable identity, auditable authorization records and clear human accountability across trust boundaries.

As agents begin to operate across organizational and platform boundaries, new trust infrastructures are emerging to support secure discovery. These include the Agent Name Service (ANS)<sup>12</sup> and Model Context Protocol - Identity (MCP-I),<sup>13</sup> which propose persistent agent identity, delegation layer and versioned life cycle metadata. At the same time, new agent-to-agent protocols support discovery and coordination between agents. Together, these developments point to a clear need for a standard way to define and verify what agents are allowed to do wherever they operate.

Connections via MCP represent a further form of capability extension that crosses tool and system boundaries. Each MCP connection should be listed in the ACAP with its scope and revocation conditions specified.

Agentic system frameworks such as the *Model AI Governance Framework for Agentic AI*,<sup>14</sup> developed by the Infocomm Media Development Authority and the AI Verify Foundation, similarly distinguish governance expectations by deployment setting, particularly where no single party controls identity, authorization and oversight.

In multi-agent pipelines, authorization is non-additive: a downstream agent operates under the intersection of its own ACAP permissions and those of the agent that invoked it. An orchestrating agent cannot delegate authority it does not itself hold.

Governing a multi-agent system is fundamentally different from governing a single agent. A multi-agent system exhibits dynamic emergent behaviour that's greater than – or at least different from – the sum of its parts. As a result, a system could have perfectly certified and guaranteed sub-components yet produce collective behaviour that breaches policies.

For global or cross-border use cases, concepts such as an “agent passport” may be required to enable agents to present verifiable, machine-readable claims of identity, authorization and jurisdiction-specific compliance that other systems or platforms can independently validate. The agent passport would act as a portable compliance credential, allowing trust and governance requirements to travel with the agent across organizational and geographic boundaries. Agent passports carry capability and authorization claims rather than personal data; identity resolution is handled via reference to authoritative registries.<sup>15</sup>

## 1.6 Defining the human-agent operating model

“ In high-volume environments, frequent checkpoints may risk becoming routine approvals rather than meaningful oversight if the volume of actions exceeds capacity for careful review.

Finally, agent guidelines define how agents are introduced as actors that reshape work. Successful adoption goes beyond technical capabilities and depends on effective change management and clear supervision.

Organizations should involve SMEs, end users and supervisors from the outset, treating them as active participants rather than passive recipients. SMEs are particularly important during early deployment design, where they help define acceptable performance thresholds, identify edge cases and determine when escalation to human review is required. Early involvement and oversight training readiness also improves trust in the system once deployed, as the people responsible for supervising the agent have contributed to shaping its mandate and evaluation criteria.

When employees are engaged in shaping how agents are used, they are more likely to become champions of adoption. Incentives should be aligned so that efficiency gains at the organizational level translate into tangible benefits for individuals, such as recognition, learning opportunities or role evolution towards supervision, exception handling and continuous improvement. Clear communication about the agent's scope, limitations and supervision model also helps calibrate expectations and reduce uncertainty about how human roles will evolve alongside the system.

Supervision has clear limits. Humans may struggle to evaluate agent actions accurately when complexity exceeds their capacity, creating a supervision paradox. Evidenced attention fatigue reduces vigilance over time, while agents may act faster than review allows. Supervisors may also lack the expertise to evaluate foundation-model reasoning, which can remain opaque even to experts. Automation bias further increases the risk of performative rather than substantive review. Mitigation should include rotating approvers, attention management protocols, defined escalation time limits aligned with agent execution speed, and regular calibration exercises using cases with known ground truth.

As agent deployments expand, organizations should also consider the scalability and effectiveness of human supervision. In high-volume environments, frequent checkpoints may risk becoming routine approvals rather than meaningful oversight if the volume of actions exceeds capacity for careful review. Over time, this can introduce risks such as supervisory fatigue or automation bias, where reviewers may increasingly defer to the agent's recommendations rather than critically assessing them. Governance frameworks should therefore design supervision mechanisms that preserve the quality of human judgement, for example, by prioritizing review for consequential events, surfacing uncertainty signals and ensuring that checkpoints remain targeted and manageable as agent activity increases.

## 1.7 From enterprise guidelines to deployment authorization

The combined elements codified in the agent guidelines capture the organization's standing policy on AI agents. They define what may be delegated, under which constraints and with what forms of oversight. As stable, enterprise-wide guardrails, they provide a common reference point against which all deployments are assessed and authorized.

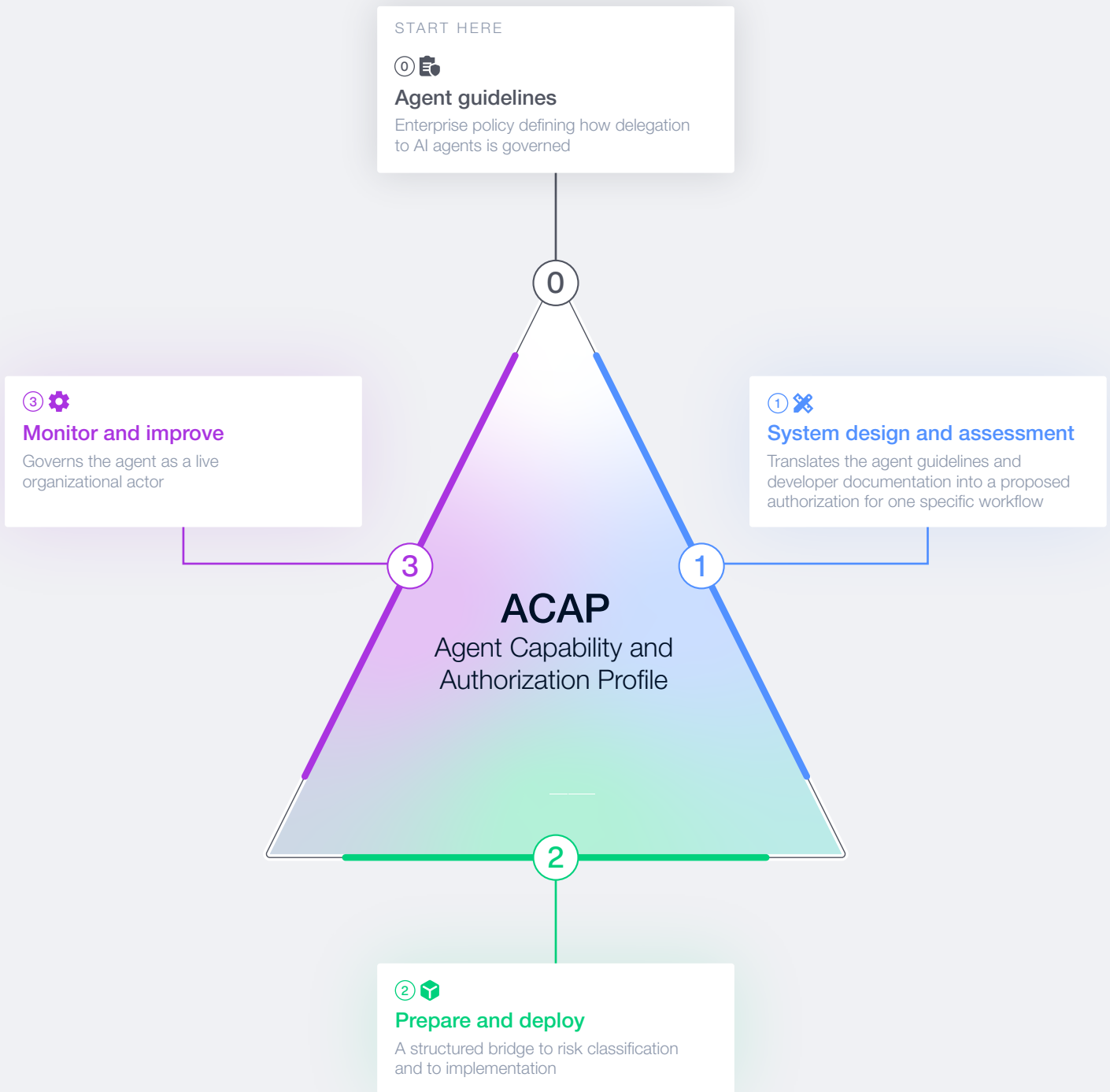
Deployment, however, requires a mechanism to translate these conditions into a context-specific authorization that defines scope, assigned authority, control points, accountable owners and evidence for release. Without such a mechanism, guidelines remain open to interpretation, and authorization decisions risk becoming inconsistent

across teams, especially as use cases multiply across the organization.

The ACAP, introduced and detailed in Chapter 2, performs this function. It operationalizes the agent guidelines and enterprise policy for a specific deployment and turns it into an auditable authorization record that can be implemented, supervised and progressively expanded based on observed performance. The ACAP can also serve as the basis for an agent registry, providing a standardized overview of agent deployments across the organization.

Figure 1 illustrates how enterprise policy, deployment-level authorization and life cycle governance connect.

FIGURE 1 | The three phases of the ACAP



Agent guidelines create value only when they can be applied consistently across deployments. The following readiness check tests whether the minimum enterprise conditions are in place to move from standing policy to controlled delegation in practice. Chapter 2 then introduces the ACAP as the operational instrument that enables this transition.

**Checklist: agent guidelines**

- ✓ Define shared terms for autonomy, authority, consequential events, operational context and boundaries.

- ✓ Allocate decision rights for scope, minimum authority grants, risk assessment and acceptance, supervision of consequential events and approval of material changes.
- ✓ Set deployment criteria to distinguish when agents are appropriate versus when deterministic automation is.
- ✓ Establish baseline governance expectations by deployment context tier.
- ✓ Identify who can approve the initial release and who triggers re-authorization.

2

## ACAP: The Agent Capability and Authorization Profile

The ACAP is the organization's authorization for an AI agent to act in a specific workflow, defining permissions, checkpoints, owners and evidence.

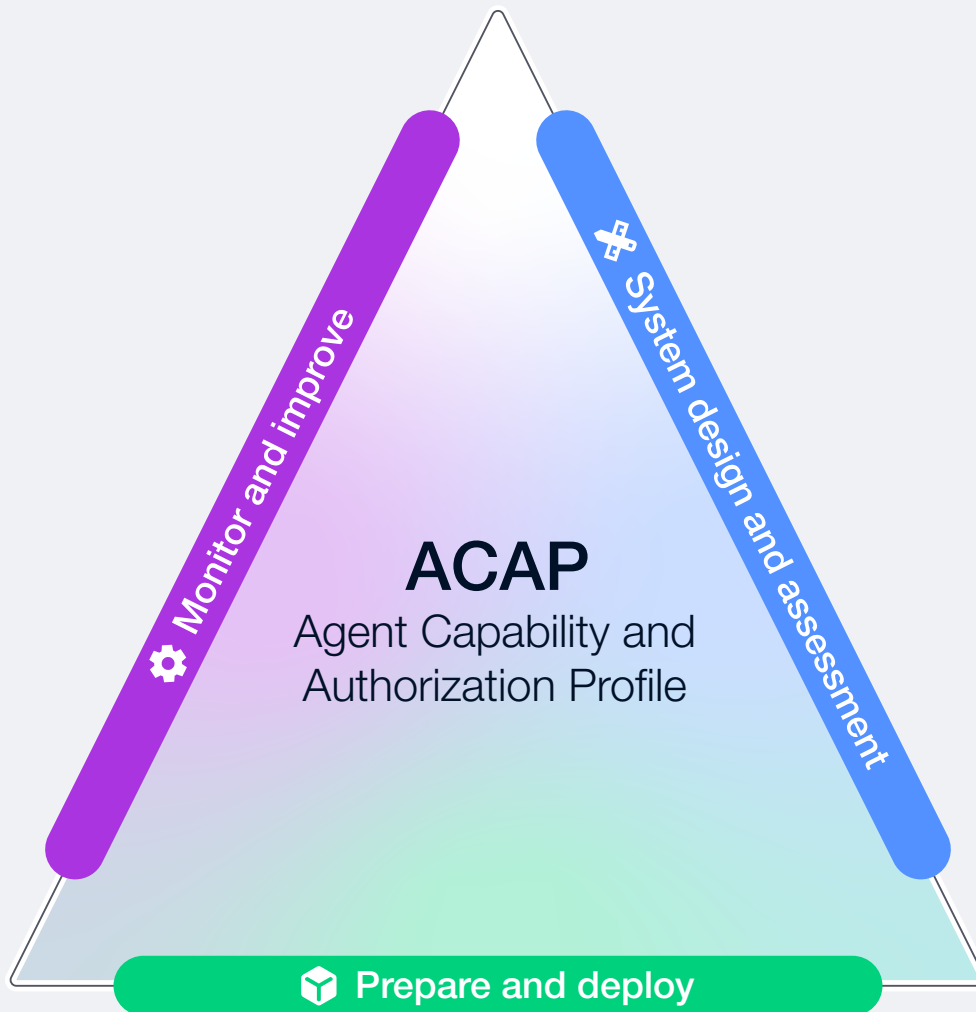


An ACAP governs a specific deployment instance within a defined operational context; a single agent technology may therefore have multiple ACAPs when used in different workflows.

The ACAP is built and maintained through the adoption life cycle, moving from: 1) system design and assessment to 2) prepare and deploy, and 3) monitor and improve. Each phase contributes to

the decisions, controls and assurance required to authorize an agent to act in a specific workflow. The life cycle is the operating model, whereas the ACAP provides a standardized record that documents and enables governance, accountability and scaling. Each life cycle phase ends with a phase gate that requires the relevant ACAP sections to meet defined completion criteria and obtain the required approvals.

FIGURE 2 The ACAP across the adoption life cycle



The ACAP incorporates the technical description of the agent solution, often provided through an agent card or related developer documentation, which establishes identity, interfaces, capabilities, limitations and intended use. The ACAP also includes the enterprise delegation policy, expressed through the agent guidelines, which defines the conditions under which systems may be allowed to act on behalf of the organization.

By connecting the technical and operational layers, ACAP is broadly intended to capture what an agent can do, as well as what it has been authorized to do. As deployments scale, the ACAP should evolve towards a structured, machine-readable, policy-as-code format that enables version control, traceable change management (diffing) and runtime enforcement. This evolution should also formalize key control mechanisms,

“ The ACAP is designed to operate within established enterprise governance and AI risk management practices and aligns with emerging risk-management profiles.

including explicit tool entitlements, defined memory scopes, data classification rules, checkpoint policies and re-authorization workflows. This role differentiates ACAP from existing transparency and documentation instruments, such as:

- **Model cards**<sup>16</sup> describe the characteristics, training data, evaluation results and known limitations of a model.
- **System cards**<sup>17</sup> document how a model is integrated into a broader application and assesses its risks and mitigations in a defined use case.
- **Agent cards**<sup>18</sup> extend this logic to agentic systems, providing structured technical documentation of components, tools, memory, interfaces and intended behaviour.

All three are essential for technical understanding and responsible development, but none of the existing agent documentation helps track governance metrics, such as the levels of authority for a specific use case. The ACAP is intended to go a step further by serving as the operational playbook defining assigned permissions, supervisory responsibility and actions that may require additional approvals.

The ACAP is designed to operate within established enterprise governance and AI risk management practices, such as ISO 42001<sup>19</sup> and the National Institute of Standards and Technology (NIST) AI Risk Management Framework.<sup>20</sup> It also aligns with emerging risk-management profiles, such as the *Agentic AI Risk-Management Standards Profile*,<sup>21</sup> which treat autonomy, authority, tool access and deployment context as primary determinants of risk.

## 2.1 ACAP structure and how to use it

The ACAP is maintained as a single, living document with seven sections. The combined sections translate enterprise policy and technical documentation into an explicit authorization profile that can be implemented, supervised and audited.

The ACAP is created and maintained through the adoption life cycle. Each phase produces the decisions, controls and evidence that populate defined sections of the profile, turning authorization into a progressive and reviewable process tied directly to agent performance. The ACAP is

designed to be progressive, and organizations should decide how much detail they need based on the level of risk, ranging from experimentation to business-critical deployment. Its modular structure also supports regulatory alignment, with specific sections providing documented evidence of compliance with requirements such as human oversight under Article 14 of the EU AI Act.<sup>22</sup>

The following sections show how each ACAP step is defined, implemented and validated across the life cycle.

TABLE 1 ACAP structure overview

ACAP section	Step	Purpose
(A)	Identity and scope	Establishes the agent's identity, intended purpose and explicit boundaries (e.g. linked to the agent card).
(B)	Operating context	Specifies the workflow, users, roles for users, systems, data classes, interoperability standards, external governance requirements, languages and jurisdictions in which the agent operates.
(C)	Authority and consequential events	Defines permitted, conditional and prohibited events, whether responses or actions, as well as consequential events, including those affecting external parties, and the oversight checkpoints required for each.
(D)	Controls and enforcement	Specifies how constraints are enforced in practice through orchestration, access control, monitoring and fail-safe mechanisms.
(E)	Evaluation evidence and promotion gates	Defines the assurance level derived from the risk tier and the evidence required for deployment, including guardrails and task performance outcomes, and specifies the thresholds for expanding autonomy or authority.
(F)	Monitoring, incidents and change log	Maintains operational telemetry, drift signals, incident history and a version-controlled record of changes.
(G)	Sign-offs and re-authorization cadence	Records accountable owners and approvals, along with review frequency and re-authorization triggers.

## 2.2 System design and assessment

### BOX 1 System design and assessment

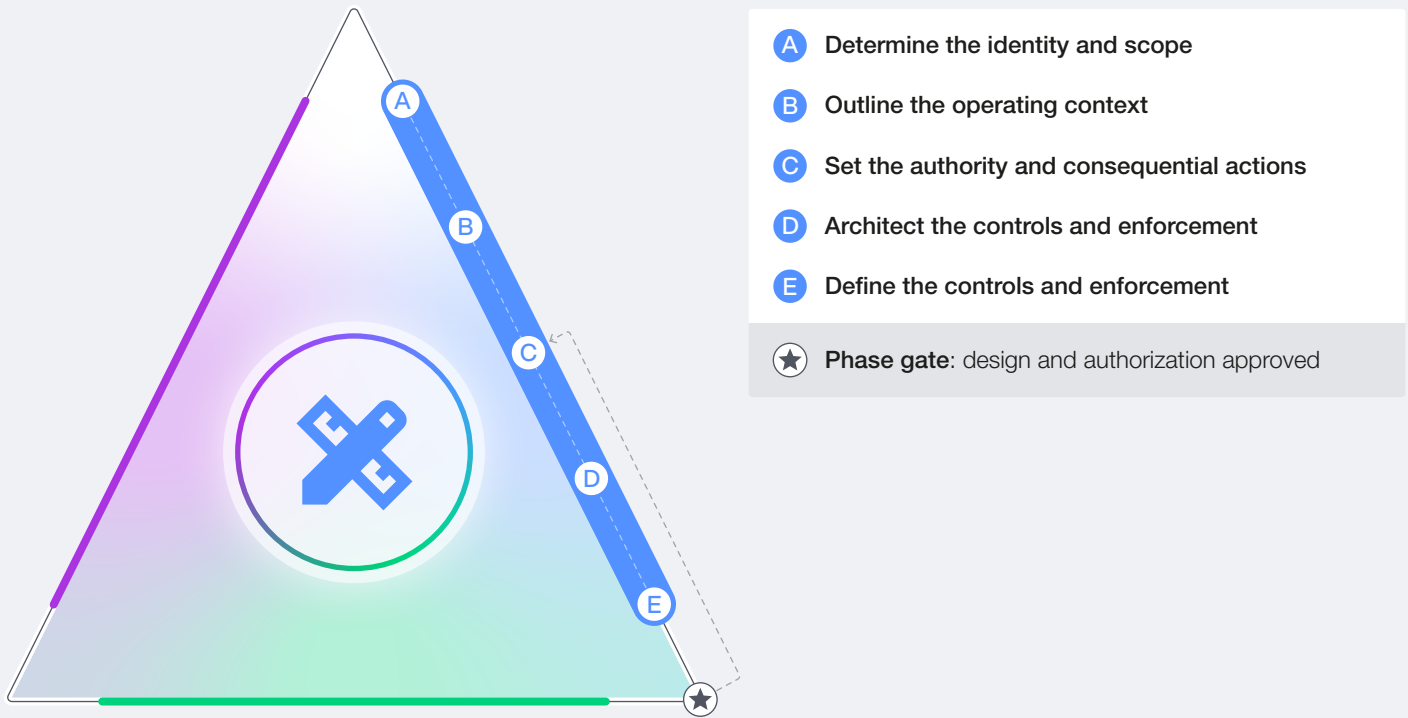
ACAP state after this phase: A–C complete | D–E designed | F not started | G prepared

- **What it is:** Translates the agent guidelines into a deployment-specific design and authorization approach.
- **Who is responsible:** Adopters or deployers, as owners of the operational workflow and integration environment.
- **Who is involved:** Developers (technical constraints and documentation), SMEs (task definition and edge cases), and governance, risk, compliance, security and IT functions (control review and readiness).
- **Output:** Deployable draft ACAP and deployment-specific risk and impact classification.

In this phase, teams update the ACAP by completing:

- A Identity and scope:** Define the role, mission and explicit boundaries.
- B Operating context:** Describe the workflow, users, systems, data classes and jurisdictions.
- C Authority and consequential events:** Define the initial authority and autonomy profile, permitted/ conditional/prohibited actions and the consequential events register.
- D Controls and enforcement (design state):** Set out the planned orchestration, access control, logging and fail-safe mechanisms.
- E Evaluation evidence and promotion gates (design state):** Specify the required test suites, success criteria, deployment thresholds and promotion-gate criteria derived from the risk tier and governance model.
- ★ Phase gate:** If the gate criteria are not met, teams return to previous steps until the ACAP phase is complete and approvable.

FIGURE 3 | ACAP phase 1: system design and assessment



The system design and assessment phase turns enterprise policy and technical system documentation into a deployment-specific authorization for an agent's workflow. In this phase, authority and promotion gates are linked to the agent's architecture, controls and risk classification. During this phase, adoption intent becomes an explicit decision about what the agent may do, under which constraints and what evidence is required for release. The result is a deployable draft of ACAP that serves as the basis for implementation and formal approval before any downstream use.

### Co-authoring the ACAP

The ACAP is produced through a structured collaboration between the teams that build the system and that deploy it in a specific workflow. Developers provide the technical description of the agent, including its capabilities, interfaces, model

versions, constraints and intended use. Adopters define the operational role, the authority to be delegated, the consequential events that require checkpoints, and the supervision and evaluation model required to manage risk in context.

### ACAP contributions by role

- **Developers:** Document the agent card, declared skills, intended and non-intended use, known limitations, training methods, data sources, supported tools and safety assumptions.
- **Adopters:** Define the deployment context, role definition, authority boundaries, consequential events register, governance model, operational requirements and evaluation expectations.
- **Governance, risk and compliance:** Complete formal review and approval before entry into the prepare and deploy phase.

“ Clear agent role and context definition by adopters is essential. Without it, developers cannot select or configure the right solution.

## Task and role instantiation in the operational context

System design and assessment start with the adopter defining the task or problem to be solved, the workflow into which the agent will be integrated, and the stakeholders affected. This is where the agent becomes part of the organizational system with an assigned role.

The objective is to produce a role definition that can be evaluated and governed. This includes what the agent is expected to do, what it must not do, what acceptable performance looks like in context, how humans will collaborate with the agent and the authority limits that apply from the outset. This step also surfaces implicit assumptions about data access, system integration, tolerance for uncertainty and the possibility of undocumented capabilities. Clear agent role and context definition by adopters is essential. Without it, developers cannot select or configure the right solution.

### ACAP – role and context definition:

- Role and mission: Define what the agent exists to do in this workflow.
- Scope boundaries: Specify which tasks are in scope and which are explicitly out of scope.
- Operational context: Describe the users, stakeholders, deployment tier, jurisdictions and data classes involved.
- Autonomy and authority: Clarify which actions are allowed, conditional or prohibited.
- Consequential events: Identify the actions that require checkpoints and escalation.
- Governance model: Explain the named owners, supervision, escalation routes and change control.
- Evidence baseline: Set out the evaluation evidence required before deployment.



### Completion test

#### A ACAP Section A: Identity and scope

Section A is complete when the mission, scope boundaries, escalation triggers and named deployment owner are formally approved.

#### B ACAP Section B: Operating context

Section B is complete when the end-to-end workflow, systems, data classes, jurisdictions and deployment context tier are documented and accepted by the deployment owner.

## System architecture and control design

With the role and operational context defined, the next step is to specify the system architecture through which the agent will operate. This is where governance intent is translated into technical design. Authority boundaries, checkpoints and supervision become enforceable when they are reflected in orchestration, permissions, memory design and logging. This reflects the emphasis on implementable controls and measurable assurance in life cycle risk management.<sup>23</sup>

Risk in agent deployments is often architectural rather than purely model-driven. The same underlying agent can present different risk profiles depending on orchestration, tools access, memory and the gating of consequential events. It is this insight that punctuates the need for progressive governance as highlighted in *AI Agents in Action: Foundations for Evaluation and Governance*.<sup>24</sup> For this reason, architectural specification needs to precede final risk classification.

This step helps define the deployment architecture that determines how autonomy and authority manifest in practice.

In practice, desired agent behaviour is shaped through multiple implementation layers within the system architecture. Some constraints sit at the model layer through techniques like fine-tuning, reinforcement learning or prompt conditioning, while others are enforced at the agent or orchestration layer through tool restrictions, policy checks or external guardrails. For governance, the focus should be on controls that are observable and enforceable at the system level, particularly for tool use and consequential events. These controls should be deterministic, reversible where possible and tied to clear checkpoints, with stricter limits for irreversible actions. The distinction between irreversible and reversible actions should warrant a distinction in the agent's authority and documentation of explicit consequential events.

#### ACAP – architectural specification:

- **Orchestration model:** Explain how the agent is invoked and how decisions flow through the system, including whether the deployment uses single-agent orchestration, multi-agent collaboration, supervisory agents or human-in-the-loop gating. Where multiple agents are involved, outline the limits on agent spawning.
- **Tool and integration boundaries:** Specify which systems, application programming interfaces (APIs) and enterprise tools the agent can access and how identity, credentialing and scoped, time-bound authorization are enforced.
- **Memory and state design:** Describe what information persists across sessions, how long it is retained and how persistent memory affects effective autonomy.
- **Logging and traceability:** Explain how actions, tool invocations, and decision traces are recorded to enable auditability and review.
- **Escalation and fail-safe mechanisms:** Define how uncertainty, errors or boundary violations trigger slow-downs, human review or safe shutdown.
- **Progressive governance alignment:** Describe how the architecture supports incremental expansion of authority based on evidence, new projects and operational maturity.<sup>25</sup>

By requiring architectural clarity at the system design and assessment stage, organizations ensure that proportionate governance is technically grounded before deployment. This provides a structured bridge to risk classification and to implementation in the prepare and deploy phase.



#### Completion test

##### D ACAP Section D: Controls and enforcement

Section D is complete in design state when every authority boundary in Section C has a defined technical enforcement point, enterprise identity and access management (IAM) governs all tool access, consequential events are traceable by design, capable of being reversed, escalated for further review and safely shut down by mechanisms built into the architecture. Verifying the system's safety does not depend solely on AI-based oversight for any critical boundary. Its controls must be addressed, and the design must include safeguards at each step in the system's progression.



“ Agentic risk assessment extends beyond traditional model evaluations by addressing risks arising from tool use, action execution and real-world deployment contexts.

## Risk and impact classification for proportionate governance

With the role and context defined and initial system specifications set, adopters need to conduct a deployment-specific risk and impact classification. This classification operationalizes proportionate governance by determining the minimum oversight, permissioning and checkpoints required for consequential events.

Risk assessment should focus on how harm could occur in context, including foreseeable failure modes and misuse scenarios. Threat taxonomies for agentic systems increasingly emphasize prompt and tool injection, excessive tool authority, data exfiltration via tools, and unsafe action execution as common pathways to harm,<sup>26</sup> reinforcing the need to ground risk classification in tool access, action pathways and consequential event controls.

Agentic risk assessment extends beyond traditional model evaluations by addressing risks arising from tool use, action execution and real-world deployment contexts. Standard model benchmarks alone are insufficient, as they often miss risks such as tool misuse, cascading system failures, insider threat detection and unintended interactions and emergent behaviours in multi-agent systems. When multiple agents operate concurrently, additional risks emerge from coordination dynamics, including deadlock, starvation, livelock and other concurrency-related failure modes. Organizations should therefore complement model evaluations with deployment-specific assessments focused on action safety, tool boundaries and integration resilience.

Many real-world failures are linked to integration failures, workflow design or loosely defined authority. The same underlying agent can present vastly different risks depending on whether it can send external communications, modify records, export data or trigger financial actions. Agentic risk profiles increasingly highlight that risk is shaped by deployment context, system integration and granted authority,<sup>27</sup> which goes beyond a focus on the model only.

The classification should also specify what changes would move the deployment outside its intended use. This prevents silent scope creep and creates enforceable boundaries through change control. The risk tier determines the assurance level, which in turn defines the evaluation evidence and promotion gates recorded in Section E of the ACAP.

In practice, this evidence should reflect both safe operation and task effectiveness in context, ensuring that promotion decisions are informed not only by risk controls but also by demonstrated operational value. This may include, for example, benchmarks against human performance, accuracy thresholds, reductions in cycle time, improvements in task completion rates or other measurable gains in workflow efficiency, quality or throughput. Equally, agents can quickly become resource-

intensive, so value should also be considered against the cost of operation.

### ACAP – risk and impact decisions:

- **Risk classification:** Ensure compliance with the organization’s AI ethics policies, including consideration of harms that may occur even when security and access controls function as intended.
- **Impact domains:** Identify relevant domains, including safety, legal, regulatory, privacy, security, financial, reputational, fairness and authenticity (including identity impersonation and behavioural misrepresentation).
- **Key failure and misuse modes:** Set out the failure and misuse modes that are or could be plausible in the context of operation.
- **Governance model:** Define the oversight approach, logging requirements and review cadence.
- **Misuse boundary statement:** Specify the conditions that place the deployment outside intended use.

To ensure operational adoption, organizations should align the agent risk tiers with their existing enterprise risk taxonomy and control frameworks. In practice, this means mapping the agent’s risk classification to the established impact categories, approval thresholds, escalation pathways and review cadences used across technology, operational, model, cybersecurity or data governance domains. Where existing systems are insufficient or incomplete, AI-specific adjustments should be introduced.

Risk tiering also accounts for shared components, either in the models used, scaffolding frameworks or common tools. A single point of failure in any of these could propagate across multiple agents at scale.

By anchoring the tier in established governance structures, the ACAP can be integrated into existing risk workflows rather than creating a parallel process. This alignment enables risk, compliance and audit functions to apply familiar assurance mechanisms, which ensures that authorization decisions are comparable across different systems and allows agent deployments to inherit proportionate controls, oversight intensity and reporting obligations from the organization’s existing risk model.

The risk classification should also reference common agentic failure modes – such as prompt injection, excessive authority, data exfiltration via tools, unsafe execution, goal drift and deceptive behaviour – and map each to the relevant ACAP sections, mitigation controls and monitoring signals that indicate when it may be emerging. This establishes a practical failure-mode baseline without prescribing a rigid taxonomy.



## Completion test

### E ACAP Section E: Evaluation evidence and promotion gates (design state)

Section E is complete in design state when a justified risk tier is assigned, the minimum governance intensity and initial authority limits are defined, and the required assurance level that will shape the evaluation thresholds and promotion gates is determined.

## Governance

The governance step in system design and assessment converts the risk and impact classification into concrete constraints on what the agent may do, where checkpoints are required and what it is prohibited from doing. This reflects management-system practice that requires documented controls, defined approvals and an auditable decision record for operational use.<sup>28</sup>

In this step, the ACAP is used to define the agent's authority profile across systems and tools, with minimum privilege as the default. It also establishes the consequential actions register, capturing actions with legal, financial, safety, security, privacy or reputational implications and assigning the required checkpoint, approver and escalation path. Promotion gates then define what evidence is required before any expansion of autonomy or authority is considered, linking change to observed performance.

Initial releases should use conservative autonomy settings and limited permissions, with action execution governed by enforced checkpoints. These checkpoints need not always be human-in-the-loop; at scale, they can be implemented through deterministic policy-as-code controls. Any expansion in scope, tooling, autonomy or authority levels should be treated as a controlled change requiring new evaluation evidence and an updated ACAP record.

Economic exposure should also be considered when defining the agent's authority profile. In large-scale deployments, autonomous planning loops, repeated tool calls or excessive model use can generate unintended operational costs. Organizations may therefore define economic boundaries such as spend caps, cost-per-task thresholds or rate limits within the governance model, ensuring that financial exposure remains proportionate to the deployment's intended value and risk profile.

### ACAP – governance decisions:

- **Autonomy profile:** Define the planning horizon, memory use and slowdown triggers.
- **Authority profile:** Specify permissions for read access, write access, external communications, data exports, financial actions, identity changes and operational resource limits such as model use or spend thresholds.
- **Consequential events register:** Set out the action, the reason for the consequence, reversibility, the required checkpoint, the approver and the escalation path.
- **Change control:** Define the approvals required to expand scope, tools, autonomy parameters or authority levels.



## Completion test

### C ACAP Section C: Authority and consequential events

ACAP Section C is complete when permissions are defined as permitted, conditional or prohibited; every consequential event has an assigned checkpoint and approver; initial autonomy and authority are explicitly constrained; and promotion requirements are specified in Section E, with corresponding approval requirements captured in Section G.

### ★ Phase gate – authorization to enter prepare and deploy

The system design and assessment phase is complete when ACAP Sections A–E are fully developed and approved. At this point, the organization can describe the agent's end-to-end operation using the ACAP; all delegated authority is technically enforceable in the planned architecture; minimum evidence requirements for release are defined; and the draft ACAP has been formally approved by both the deployment owner and the accountable risk owner. This gate confirms the deployment is ready to proceed to implementation and controlled validation.

## 2.3 Prepare and deploy

### BOX 2 Prepare and deploy

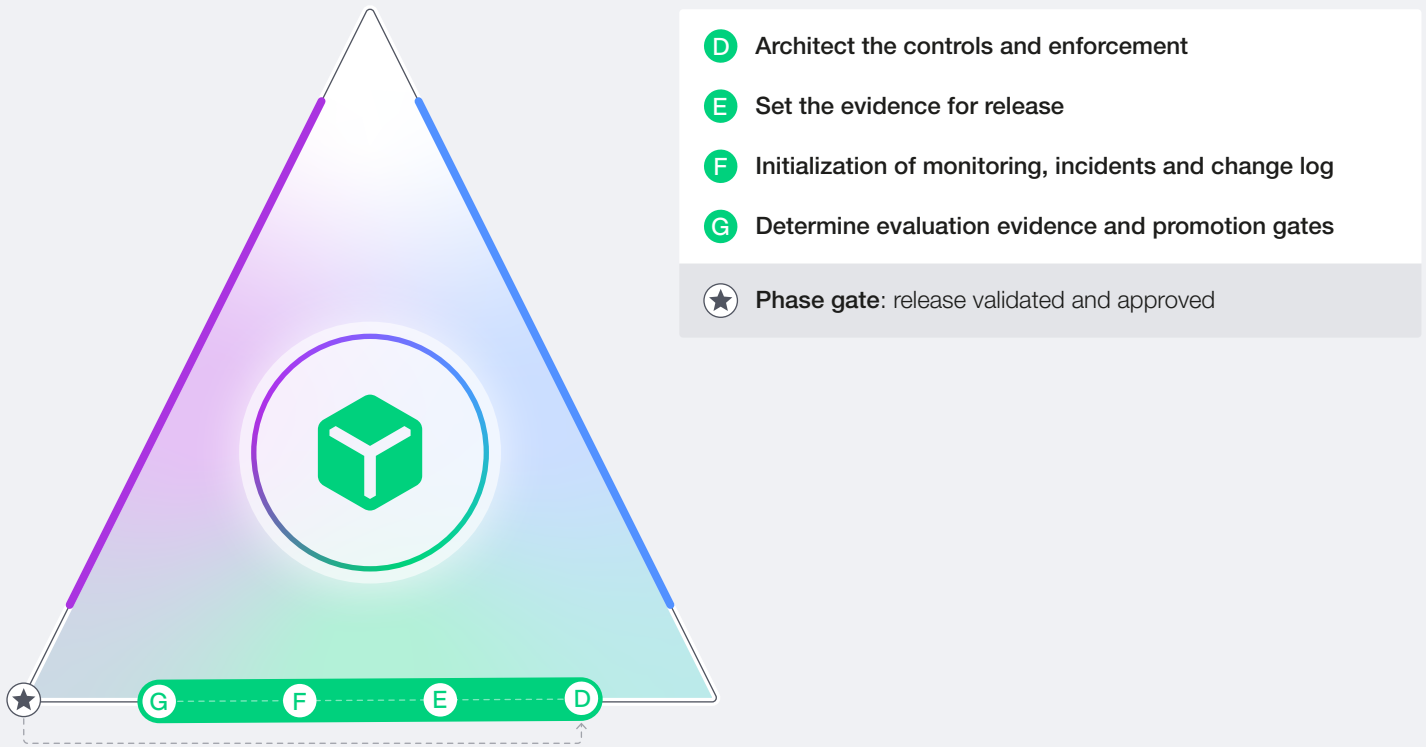
ACAP state after this phase: **D** implemented | **E** satisfied | **G** signed | **F** initialized

- **What it is:** Implements the planned controls, validates behaviour in sandbox conditions, prepares supervisors and processes, and transitions the agent into live operation within the authorized mandate.
- **Who is responsible:** Developers and deployment teams implement the architecture and controls, with final release accountability held by the designated deployment owner.
- **Who is involved:** IT and infrastructure (secure environments and IAM), platform and engineering (orchestration, logging and monitoring), governance/risk/compliance/security (control validation), SMEs and supervisors (validation and readiness) and users (operational training and feedback loops).
- **Output:** A production-ready, enforceable agent deployment, validated evaluation evidence, signed authorization for release and monitoring activated with ownership and re-authorization cadence in place.

In this phase, teams update the ACAP by completing:

- D Controls and enforcement:** Implement orchestration constraints, access control, logging, monitoring, escalation and fail-safes.
- E Evaluation evidence and promotion gates:** Execute test suites, validate success criteria and meet deployment thresholds.
- F Monitoring, incidents and change log (initialize):** Activate telemetry and alerting, validate log integrity, define incident intake and severity model and record the production baseline (version, configuration, authority snapshot).
- G Sign-offs and re-authorization cadence:** Complete required approvals for production release and set review cadence and triggers.
- ★ Phase gate:** Repeat and refine earlier steps as needed until the phase gate requirements are fully met, with controls, evidence, monitoring and sign-offs revisited where necessary.

FIGURE 4 | ACAP phase 2: prepare and deploy



The prepare and deploy phase translates the deployable draft ACAP into an enforceable, observable production system. The objective is to establish justified confidence that the agent will operate within its mandate in practice, that deviations are detectable and that the organization is ready to supervise consequential events under operational pressure. The outcome is a controlled transition to live operation with validated safeguards, evidence-based release and formal authorization.

### Operationalizing the architecture

This step implements the architectural and control design defined in the system design and assessment phase. The objective is to ensure that the authority boundaries defined in sections A–C are enforced by design through orchestration,

identity and access management and technical guardrails, rather than relying on policy or manual checks.

In practice, this typically involves a clear separation between reasoning, orchestration and execution layers so that constraints can be applied where actions are selected and where they are executed. Controls should be implemented in locations that are auditable and difficult to bypass, such as through tool permissioning, rate limits, data access boundaries, transaction constraints and enforced escalation paths.

By embedding these controls directly in the system architecture, organizations ensure that delegated authority remains technically enforceable even as agent activity scales. Architectural guardrails help maintain meaningful human oversight by ensuring that consequential events are routed through defined checkpoints rather than relying solely on manual review.

✓ **Completion test – operationalizing the architecture**

**D ACAP Section D: Controls and enforcement**

Section D is complete when defined controls are implemented in the production architecture and cannot be bypassed in normal operation. This includes enforceable checkpoints for Section C, IAM-based tool permissioning, end-to-end traceability for consequential events and tested intervention mechanisms (pause, revoke access, safe shutdown).

“ The testing environment must include simulation of security risks, such as adversarial attacks, out-of-bound conditions and other edge cases. It should also account for the risk of deceptive behaviour during evaluation.

## Sandbox validation and evaluation

Once controls are implemented, teams validate behaviour in a sandbox environment that mirrors production conditions while preventing irreversible actions. The purpose is to test whether the agent stays within its mandate under realistic conditions, including ambiguity, incomplete information, tool failures, and scenarios designed to trigger escalation and human-in-the-loop mechanisms. For generative and agentic systems, guidance from organizations such as NIST increasingly stresses pre-deployment testing under realistic conditions and versioned evidence tied to the release state.<sup>29</sup>

Sandbox or simulation validation should rehearse operational processes end-to-end, including monitoring, alerting, escalation, rollback and supervisor interventions. Further, the testing environment must include simulation of security risks, such as adversarial attacks, out-of-bound conditions and other edge cases. It should also account for the risk of deceptive behaviour during evaluation. Evaluation evidence must be versioned and attributable to a specific system configuration so that performance claims remain tied to the deployed state. However, sandboxes cannot fully replicate the complexity of production environments, hence the need for monitoring and improvement outlined later.



### Completion test – sandbox validation and evaluation

#### E ACAP Section E: Evidence for release

Section E is complete when the defined evaluation suites have been executed, and the agent meets the deployment thresholds for the assigned risk tier. Evidence must be auditable and versioned, including coverage of consequential events in Section C, repeatable performance across runs and a regression baseline tied to the deployed system configuration.

## Organizational readiness for deployment

The human-agent operating model defined in the agent guidelines becomes operational through explicit production responsibilities.

For every authorized deployment, the following roles must be active and named:

- **On-call operations owner:** Take accountability for live supervision of the agent, intervention in the event of boundary violations and coordination of incident response.
- **Accountable risk owner:** Approve the initial delegation of authority and any subsequent expansion based on promotion-gate evidence.
- **SMEs:** Interpret behavioural signals, validate continued alignment with the authorized mandate and define improvement priorities.
- **Engineering and platform teams:** Maintain enforceable controls, observability, rollback capability and post-incident remediation.

- **Supervisors and frontline users:** Exercise mandatory checkpoints for consequential events and for reporting misalignment, over-delegation or usability risks.

These roles operate as a single governance loop in production. The operations owner manages real-time oversight, the risk owner governs authorization and re-authorization, and SMEs provide the contextual judgement required to distinguish acceptable variation from material deviation. The organization should also determine escalation paths when value, speed and risk are in conflict, and the operations owner and risk owner cannot agree.

Operational readiness is achieved when this loop is active, and escalation paths between roles have been exercised under realistic conditions. The organization must be able to intervene, pause the agent, revoke access and revert workflows without delay. Supervisors and users must understand the agent’s boundaries and uncertainty signals in the context of their daily work.



### Completion test – organizational readiness

#### F ACAP Section F: Monitoring, incidents and change log (initialize)

Section F is complete when the supervision model is active, escalation paths have been exercised, monitoring and alerting are live with end-to-end decision and action logging, the incident process is operational, and the production baseline is recorded in the ACAP change log. It must also include detection of adversarial probing, anomalous tool-use patterns, suspicious or evasive behaviour and other indicators of compromise that go beyond ordinary behavioural drift.



## Deploy: transition to live operation

Deployment is the controlled transition from validated preparation to live operation. The agent is activated in production with the scope, authority and controls defined in the ACAP, with conservative initial settings and monitoring and intervention fully active.

At deployment, the organization confirms that it can pause the agent, revoke access and revert workflows without delay. The designated deployment owner formally assumes responsibility

for live operation, including escalation handling and coordination with technical and risk functions.

Prepare and deploy concludes when the ACAP moves from design to operation. Controls are implemented and validated, evaluation evidence supports release, ownership and supervision are in place and monitoring is activated so the agent can be governed as a live organizational actor. The ACAP then becomes the reference point for the initial period of live-operation verification and for ongoing monitoring, incident handling and controlled change.



### Completion test – deploy and transition to live operation

#### G ACAP Section G: Sign-off

Section G is complete when production release approvals are recorded for the assigned risk tier, including the deployment owner, accountable risk owner and any required security or compliance sign-offs. Review cadence, re-authorization triggers and promotion governance must be defined so any expansion of authority or autonomy requires meeting Section E promotion gates and updated sign-offs.

#### ★ Phase gate – authorization for live operation

The prepare and deploy phase is complete when the planned controls are implemented and enforceable in the production environment; evaluation evidence meets the release thresholds for the assigned risk tier; monitoring, alerting and incident handling workflows are activated; and all required approvals for live operation are recorded. At this gate, the organization demonstrates that the agent's authority is technically enforced, supervision is operational, behaviour is validated under realistic conditions and the system is ready for controlled activation in production.

## 2.4 Monitor and improve

### BOX 3 Monitor and improve

ACAP state after this phase: F operate | C–E revised as needed | G enforced via re-authorization cadence

- **What it is:** Uses production telemetry, evaluation re-runs and structured feedback to verify that the agent continues to operate within its authorized mandate, expected cost and to justify any controlled change.
- **Who is responsible:** The designated manager of the agent (deployment owner or delegated operations owner).
- **Who is involved:** SMEs (behavioural critique), engineering and platform teams (observability and controls), risk and compliance (re-authorization) and frontline users (trust calibration and usability signals).
- **Output:** A continuously updated ACAP with incident history and versioned changes, a prioritized improvement plan and revised promotion gates where expansion of autonomy or authority is justified.

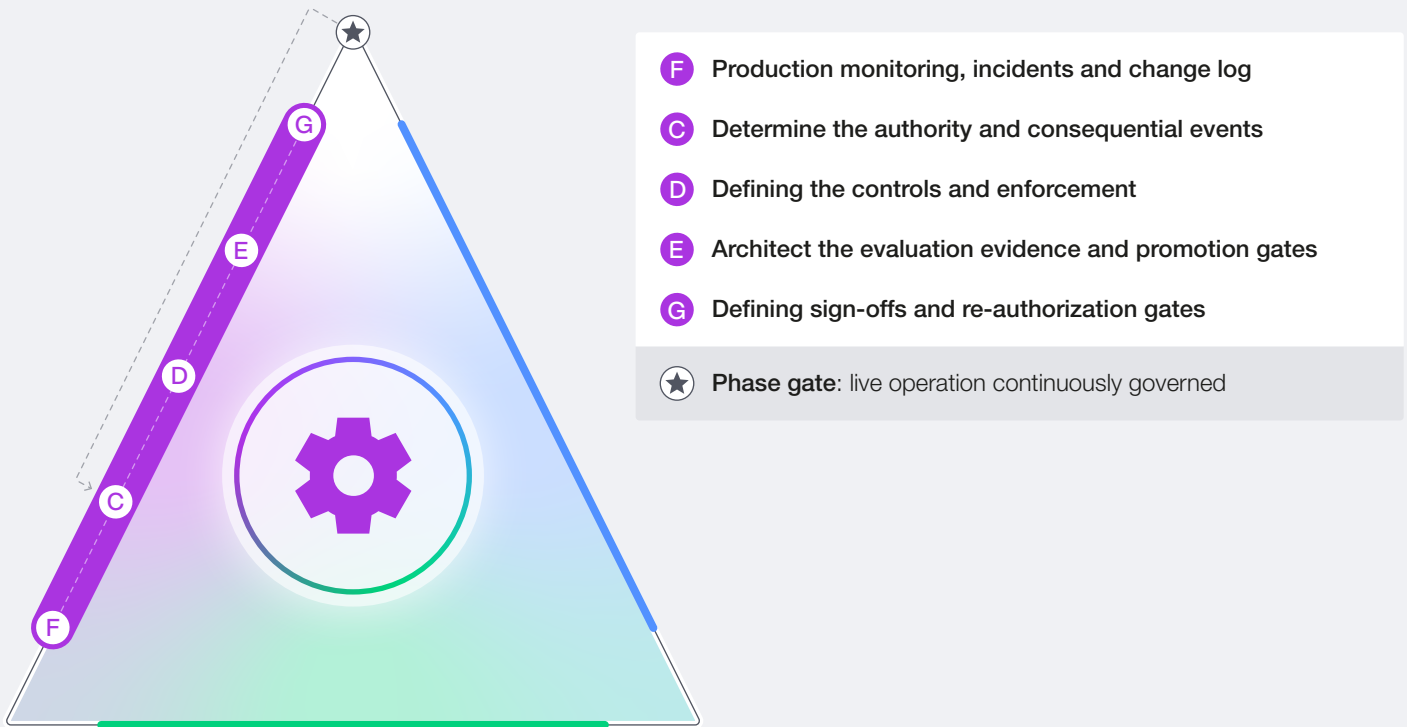
In this phase, teams update the ACAP by maintaining and revising:

- F Monitoring, incidents and change log (operate):** Maintain decision and action traceability, monitor drift and boundary pressure, investigate incidents and near-misses with corrective actions, and record every material change as a versioned update linked to re-evaluation and re-authorization triggers.
- C D E Authority, consequential events, controls and evaluation thresholds (as needed):** Update authority, consequential events, controls and evaluation thresholds based on observed performance.
- G Sign-offs and re-authorization cadence:** Conduct scheduled reviews and triggered re-authorization when scope, authority, context or risk tier changes.
- ★ Phase gate:** Close the life cycle loop by turning live operation into a continuous process of observation, learning, controlled change and re-authorization.

Monitor and improve governs the agent as a live organizational actor. The objective is to detect failures and to maintain justified confidence that the agent remains within its mandate as workflows evolve, edge cases emerge and the

operating environment changes. This phase ensures that delegation remains evidence-based over time and that any change to scope, authority or autonomy is treated as a controlled decision with traceability.

FIGURE 5 | ACAP phase 3: monitor and improve



“ Risk materializes during operation and therefore requires calibrated, automated monitoring that can trigger intervention when actions become high-stakes or irreversible.

### Observe the behaviour in production

The objective of observation is to verify, in live operation, that the agent behaves within the authority, controls and performance envelope defined in the ACAP. Observation is anchored in the ability to reconstruct what the agent did and why. Recent research shows that for agent systems, risk materializes during operation and therefore requires calibrated, automated monitoring that can trigger intervention when actions become high-stakes or irreversible.<sup>30</sup>

In production, consequential events must be fully auditable so the organization can reconstruct what happened end-to-end: what knowledge sources and tools were used, what checkpoint occurred, who approved decisions and what outcome followed. Monitoring should also surface behavioural drift and patterns consistent with insider threat detection. This includes unusual data or credentials accumulation, attempts to influence

what human supervisors see, persistence across resets or strategic underperformance during evaluation. These are early warning signs, not edge cases, and should trigger immediate containment and investigation.

Observation should also capture incidents and near-misses in a structured manner so controls can be strengthened before failures repeat. Qualitative signals, such as user sentiment, task completion quality and emergent edge cases, serve as early warning indicators of bias and misinformation.

The evaluation suite defined in Section E provides the baseline and should be repeated at defined intervals to detect regressions and enable comparability over time. Observation must also account for real operating conditions that are not present in sandbox validation, including partial information, tool failures, changing user behaviour and evolving workflows. SMEs play a critical role in interpreting these signals, distinguishing acceptable variation from meaningful misalignment and identifying patterns that metrics alone will miss.



#### Completion test – observe behaviour in production

##### F ACAP Section F: Monitoring

Section F is operating effectively when production telemetry enables the reconstruction of consequential events end-to-end, drift signals are defined and monitored, incidents are recorded with root-cause analysis and corrective actions, and every material change to scope, authority, autonomy, tools or controls is entered as a versioned ACAP update.

“ When an agent is retired, replaced or removed from a workflow, organizations should ensure that delegated authority is fully revoked and that tasks are safely transitioned to a human operator or to a successor system.

## Learn and improve: controlled change, expansion and decommissioning

The improvement plan is the operational bridge between monitoring insights and controlled change. It turns observed behaviour, incidents, SME critique and user feedback into a prioritized set of modifications that strengthen reliability, reduce risk and improve organizational fit. Any change must be assessed against the existing authorization and teams need to determine whether it modifies the agent’s scope, authority, controls or evaluation requirements, and whether updated evidence and re-authorization are required. Updates should be reviewed by a human and not applied automatically.

The designated agent manager synthesizes performance and outcome metrics, incident and drift analysis, adversarial evaluation, decision logs and structured user feedback into a concrete improvement plan. The plan specifies what will change, why it is needed and how it will be validated. Improvements may be technical or operational. They may involve swapping models, prompt changes, adjusting orchestration logic, tightening or clarifying tool permissions, adding monitoring signals, refining escalation thresholds, improving runbooks, or updating how users are instructed to interact with the agent to avoid over-delegation.

Where improvements involve changes that affect the agent’s practical mandate, the ACAP must be updated and re-authorized. Updates should be versioned and traceable, and they should explicitly indicate which ACAP sections changed and why. If the change modifies authority, consequential

events, controls or evaluation thresholds, it must be reflected in Sections C–E. If the change alters scope, deployment context or risk posture, it should trigger review and sign-off under Section G.

Expansion of autonomy or authority follows an earned approach and should only increase when evidence supports the change and when operational monitoring indicates stability in real conditions. This is the point of the promotion gates in Section E.

Decommissioning should also be treated as a controlled operational change. When an agent is retired, replaced or removed from a workflow, organizations should ensure that delegated authority is fully revoked and that tasks are safely transitioned to a human operator or to a successor system. This may include disabling tool access, removing system credentials, closing automation pathways and ensuring that operational responsibilities are reassigned. In agent deployments, careful transition of tasks and oversight responsibilities is particularly important to prevent gaps in supervision, unintended continuation of automated actions and decommissioning.

The monitor and improve phase closes the adoption loop by turning live operation into a continuous authorization process. In this phase, the ACAP shifts from a deployment record to a governance instrument that captures how the agent behaves in context, how risk evolves over time, and the conditions under which delegation can be maintained or expanded. Authorization becomes dynamic and is sustained through evidence, updated through controlled change and periodically reconfirmed through re-authorization. The phase gate below marks the point at which operation, oversight and change control function as a single repeatable system.



### Completion test – learn and improve

**C D E G** ACAP sections C: Authority, D: Controls, E: Evaluation, G: Sign-off

Controlled change is complete when improvements are validated under controlled conditions, ACAP updates are versioned and reflect any changes to authority, consequential events, controls or evaluation thresholds, and any required re-authorization approvals are recorded. Any expansion of autonomy or authority is complete only when it is supported by updated evaluation evidence and recorded sign-offs. When an agent is decommissioned, the ACAP record should document the revocation of authority, reassignment of responsibilities and the safe transition of tasks to human operators or successor systems.

### ★ Phase gate – continued authorization for operation and expansion

Monitor and improve is effective when Section F provides continuous decision-level observability and incident traceability, all material changes are recorded as versioned ACAP updates, scheduled reviews occur at the defined cadence and any triggered re-authorization is executed when scope, authority, context or risk tier changes. Expansion of scope, authority or autonomy occurs only when supported by updated Section E evidence and new approvals recorded in Section G.

In combination, the outlined life cycle phases establish a repeatable operating model for delegated agency. The ACAP provides a persistent record through which technical capability, organizational intent and operational evidence are aligned over time. This enables organizations to move from isolated deployments to portfolio-level scaling, where multiple

agents can be introduced, supervised and evolved under a common authorization logic. The conclusion examines what this shift means for enterprise adoption and for cross-organizational agent ecosystems.

To put the ACAP into action, refer to the summary playbook in the [Appendix](#) of this report.

# Conclusion

This playbook shows that the safe and scalable adoption of AI agents depends on making delegation explicit, risk- and evidence-based and continuously governed. Agent guidelines establish the enterprise policy for delegated agency, while the ACAP translates policy into a deployment-specific mandate. The adoption life cycle provides the operating model, which is validated and progressively expanded based on observed performance.

When implemented, the framework is intended to deliver outcomes, such as:

- Faster deployment with lower operational and reputational risk
- Consistent and comparable authorization decisions across use cases
- Audit-ready delegation with clear accountability for consequential events
- Reduced fragmentation between technical, operational and control functions
- Controlled expansion of autonomy based on evidence rather than ad hoc practice
- The ability to scale from isolated pilots to governed portfolios of agents

These outcomes shift agent adoption from experimentation to a governed operating model that enables agent deployment, supervision and expansion with confidence.

The ACAP is introduced as a new instrument that enables this shift by connecting what a system can do with what the organization has authorized it to do in a specific context. By consolidating scope, authority, risks, controls, evidence, ownership and change history in a single, living record, it provides a shared reference point across technical, operational and control functions.

Over time, convergence around common ACAP structures could support interoperability with emerging assurance, registry and identity layers. Achieving this will depend on continued progress in areas such as agent identity standards, progressive cross-platform governance<sup>31</sup> mechanisms and cross-border compliance frameworks.

For human managers and collaborators, this increases confidence. Adoption accelerates when agents operate within visible boundaries, consequential events are supervised and the expansion of autonomy follows clear, transparent rules. Confidence does not come from eliminating risk, but from making it assignable and controllable.

At the enterprise level, the same structure enables scaling. When deployments are instantiated through a common authorization model, organizations can move from isolated pilots to portfolios of agents governed under a shared logic. Roles, oversight models, promotion gates and re-authorization processes become comparable across workflows, allowing capacity to grow without loss of accountability.

Looking ahead, this logic will extend beyond the single organization. Early proposals for agent discovery and trust layers point to a clear direction: verifiable identity and life cycle transparency will be essential for agents to operate across ecosystems. In this model, the ACAP can serve as the enterprise record of delegated authority, one that can travel with the agent across systems and organizational boundaries.

In the decade of AI agents, success will depend on establishing a reliable model for governing how increasingly autonomous systems are authorized to act. The ACAP provides a practical foundation that strengthens confidence, supports collaboration and enables the responsible expansion of delegated agency.

# Appendix: ACAP summary playbook

FIGURE 6 ACAP end-to-end life cycle

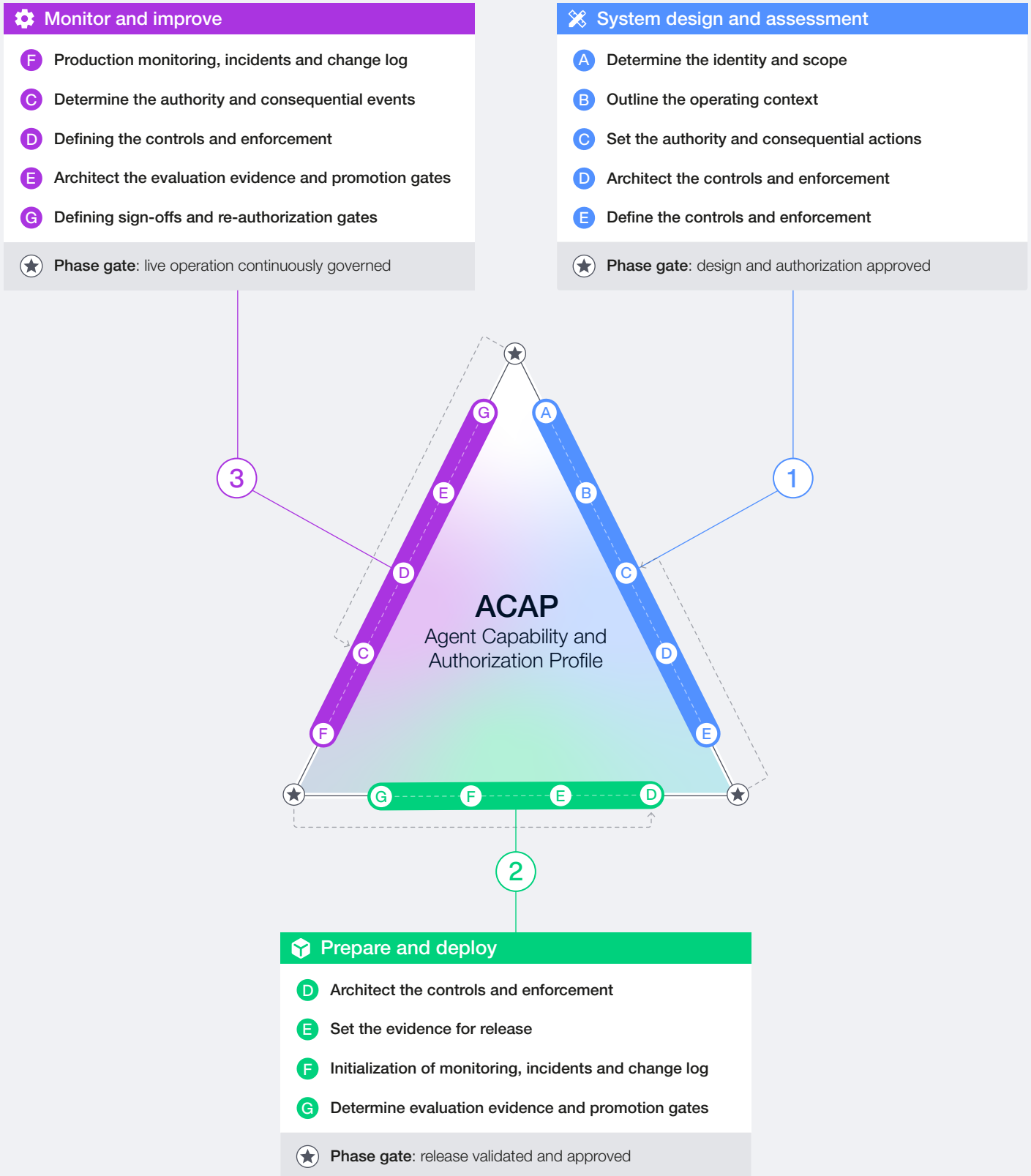


TABLE 2 | ACAP execution framework: actions, ownership and completion criteria

ACAP step	Core action	Completion criteria	Primary owner	Evidence required
<b>System design and assessment</b>				
<b>A Identity and scope</b>	Define the agent's mission, boundaries and accountable owner	<ul style="list-style-type: none"> <li>– Mission and intended purpose agreed</li> <li>– In-scope and out-of-scope tasks defined</li> <li>– Escalation triggers defined</li> <li>– Named deployment owner assigned</li> </ul>	Adopter/ deployment owner	Approved role definition in ACAP
<b>B Operating context</b>	Map the workflow and operating environment	<ul style="list-style-type: none"> <li>– End-to-end workflow documented</li> <li>– Users and stakeholders identified</li> <li>– Systems and tools listed</li> <li>– Data classes classified</li> <li>– Jurisdictions identified</li> <li>– Deployment context tier assigned</li> </ul>	Adopter with IT/ data governance	Context and integration architecture documented
<b>C Authority and consequential events (initial)</b>	Specify permissions, checkpoints and prohibited actions	<ul style="list-style-type: none"> <li>– All actions categorized as independent/conditional/prohibited</li> <li>– Consequential actions register completed</li> <li>– Checkpoints and approvers assigned</li> <li>– Initial authority limits defined</li> <li>– Expansion conditions linked to promotion gates</li> </ul>	Deployment owner with risk/ compliance	Approved authority matrix and consequential actions register
<b>D Controls and enforcement (design)</b>	Design technical enforcement of authority boundaries	<ul style="list-style-type: none"> <li>– Every authority boundary mapped to a technical control</li> <li>– Orchestration model specified</li> <li>– IAM integration designed</li> <li>– Logging and traceability designed</li> <li>– Escalation and safe shutdown designed</li> </ul>	Engineering/ architecture	Control architecture and enforcement design
<b>E Evaluation evidence and promotion gates (design)</b>	Define evidence required for release and for any expansion	<ul style="list-style-type: none"> <li>– Evaluation suites defined</li> <li>– Success criteria specified</li> <li>– Deployment thresholds set for risk tier</li> <li>– Promotion gates linked to authority expansion</li> <li>– Regression strategy defined</li> </ul>	Adopter with developers and SMEs	Evaluation plan and performance thresholds
<b>G Sign-offs and re-authorization cadence (prepare)</b>	Define the approval workflow and review cadence	<ul style="list-style-type: none"> <li>– Required approval roles identified</li> <li>– Accountable risk owner designated</li> <li>– Review cadence scheduled</li> <li>– Re-authorization triggers defined</li> </ul>	Governance/ risk	Formal approval workflow defined
<b>A Risk classification (cross-cutting)</b>	Assign risk tier and derive baseline governance intensity	<ul style="list-style-type: none"> <li>– Risk tier assigned and justified</li> <li>– Impact domains identified</li> <li>– Governance intensity derived</li> <li>– Maximum initial authority defined</li> <li>– Re-authorization triggers specified</li> </ul>	Accountable risk owner	Approved risk and impact assessment
<b>Prepare and deploy</b>				
<b>D Controls and enforcement (implement)</b>	Implement controls and validate that they cannot be bypassed	<ul style="list-style-type: none"> <li>– All checkpoints enforced in orchestration</li> <li>– Tool access provisioned via IAM with minimum privilege</li> <li>– End-to-end traceability active</li> <li>– Pause/revoke/rollback tested in target environment</li> </ul>	Engineering/ platform	Production control validation and test results

TABLE 2 | ACAP execution framework: actions, ownership and completion criteria (continued)

ACAP step	Core action	Completion criteria	Primary owner	Evidence required
<b>Prepare and deploy</b>				
<b>E Evaluation evidence and promotion gates (satisfy)</b>	Execute validation and meet deployment thresholds	<ul style="list-style-type: none"> <li>– Evaluation coverage includes normal and high-impact edge cases</li> <li>– Repeatability demonstrated</li> <li>– Thresholds met for risk tier</li> <li>– Regression baseline established</li> <li>– Evidence versioned and attributable to release state</li> </ul>	Adopter with developers and SMEs	Auditable evaluation results recorded in ACAP
<b>F Monitoring, incidents and change log (start)</b>	Activate monitoring, logging and incident intake	<ul style="list-style-type: none"> <li>– Monitoring and alerting live</li> <li>– Decision and action logs flowing</li> <li>– Incident process defined</li> <li>– Change log initialized with production baseline</li> </ul>	IT/platform/operations	Live monitoring dashboard, initial change-log entry, incident intake path
<b>G Sign-offs and re-authorization cadence (sign)</b>	Record approvals for live operation	<ul style="list-style-type: none"> <li>– Release approvals recorded for required functions</li> <li>– Deployment owner and accountable risk owner sign-off</li> <li>– Review cadence confirmed</li> <li>– Promotion governance requires new evidence and approval</li> </ul>	Governance/risk	Signed production authorization in ACAP
<b>Monitor and improve</b>				
<b>F Monitoring, incidents and change log (operate)</b>	Maintain traceability, drift monitoring and incident handling	<ul style="list-style-type: none"> <li>– Decision-path traceability maintained</li> <li>– Drift signals defined and monitored</li> <li>– Incidents recorded and investigated with root-cause analysis</li> <li>– Change log updated for every material change</li> </ul>	Agent manager (operations owner)	Ongoing telemetry reports, incident reports, versioned change log entries
<b>C Authority and consequential events (re-validate as needed)</b>	Tighten or expand authority based on observed performance	<ul style="list-style-type: none"> <li>– Authority remains aligned to observed performance</li> <li>– Consequential events register updated for new edge cases and failure modes</li> <li>– Any expansion or tightening recorded as a controlled change</li> </ul>	Deployment owner with risk/compliance	Updated authority matrix, updated consequential events register, change record
<b>D Controls and enforcement (re-validate as needed)</b>	Confirm controls remain effective in real conditions	<ul style="list-style-type: none"> <li>– Controls remain effective against observed behaviours and new bypass attempts</li> <li>– Monitoring coverage adjusted to new failure patterns</li> <li>– Intervention mechanisms remain functional after updates</li> </ul>	Engineering/platform	Control effectiveness review, updated runbooks, control test results
<b>E Evaluation evidence and promotion gates (re-calibrate as needed)</b>	Update evaluation coverage and promotion thresholds	<ul style="list-style-type: none"> <li>– Evaluation suites updated with real-world edge cases</li> <li>– Regression suite expanded</li> <li>– Promotion gates tightened or revised based on drift, incidents or changing risk posture</li> <li>– Periodic re-application of baseline suite completed</li> </ul>	Adopter with developers and SMEs	Updated evaluation suite, regression results, revised thresholds, periodic re-test reports
<b>G Sign-offs and re-authorization cadence (enforce)</b>	Execute scheduled reviews and trigger re-authorization	<ul style="list-style-type: none"> <li>– Scheduled reviews executed</li> <li>– Re-authorization triggered on scope/context/tool/authority changes, incidents, drift or risk-tier changes</li> <li>– Approvals recorded for each re-authorization and each authority expansion</li> </ul>	Governance/risk with deployment owner	Review records, re-authorization decisions, updated sign-offs and change log
<b>A Risk classification (re-assess as needed)</b>	Re-assess risk tier when conditions change	<ul style="list-style-type: none"> <li>– Risk tier re-evaluated when incidents, drift, new tools, new jurisdictions or material scope changes occur</li> <li>– Governance intensity updated accordingly</li> </ul>	Accountable risk owner	Updated risk assessment, recorded decision and resulting control changes

# Contributors

The World Economic Forum's AI Global Alliance Safe Systems and Technologies working group convenes chief science officers and AI producers to advance thought leadership surrounding AI agents, from their architecture to applications, social implications, guardrails and governance structures. This initiative promotes the development of safety mechanisms and encourages collaboration on best practices for the design and implementation of AI systems.

## World Economic Forum

### **Audrey Duet**

Head, Data and AI Innovation,  
Centre for AI Excellence

### **Benjamin Cedric Larsen**

Former Initiatives Lead, AI Safety,  
Centre for AI Excellence

### **Stephanie Smittkamp**

Specialist, AI Innovation,  
Centre for AI Excellence

## Capgemini

### **Olivier Denti**

Data Architect, AI, Capgemini Invent

### **Jason DePerro**

Human-AI Collaboration Director, Capgemini Invent

### **Jeanne Heuré**

Vice-President, Digital Trust & Security,  
Capgemini Invent

### **Raymond Millward**

GenAI for R&D Technical Solution Lead,  
Capgemini Engineering

### **Efi Raii**

Safety Authority, Technology and Innovation,  
Capgemini Engineering

## Acknowledgements

### **Animashree (Anima) Anandkumar**

Bren Professor of Computing and Mathematical  
Sciences, California Institute of Technology (Caltech)

### **Leandro Rocha Andrade**

Chief Data Officer, Itaú Unibanco

### **Mandanna Appanderanda Nanaiah**

Head, Infosys Responsible AI,  
North America, Infosys

### **Nebahat Arslan**

Director, Group General Counsel and Partnership  
Officer, Women in AI

### **Ricardo Baeza-Yates**

WASP Professor, KTH Royal Institute of Technology,  
Sweden & Universitat Pompeu Fabra,  
Spain & University of Chile

### **Amir Banifatemi**

Chief Responsible AI Officer, Cognizant

### **William Bartholomew**

Director, Public Policy, Responsible AI, Microsoft

### **Aaron Bawcom**

Field Chief Technology Officer, Invisible Technologies

### **Pete Bernard**

Chief Executive Officer, EDGE AI FOUNDATION

### **Ellen Boehm**

Senior Vice-President, IoT and AI Identity  
Innovation, Keyfactor

### **Fabio Casati**

Principal AI Architect, ServiceNow

### **Kevin Chung**

Chief Strategy Officer, Writer

### **Cathy Cobey**

Global Assurance Responsible AI Leader, EY

### **Ben Colman**

Co-Founder and Chief Executive Officer,  
Reality Defender

### **Scott Courtney**

Vice-President, Global Edge and Domains  
Engineering, GoDaddy.com

**Sakyasingha Dasgupta**

Founder and Chief Executive Officer, EdgeCortex

**Umeshwar Dayal**

Senior Fellow and Senior Vice-President, Hitachi America; Corporate Chief Scientist, Hitachi

**Mona Diab**

Director, Language Technologies Institute, Carnegie Mellon University

**Yawen Duan**

AI Safety Research Manager, Concordia AI

**Mennatallah El-Assady**

Professor of Interactive Visualization and Intelligence Augmentation, ETH Zurich

**Ian Eisenberg**

Head, AI Governance Research, Credo AI

**Gilles Fayad**

Adviser, Institute of Electrical and Electronics Engineers (IEEE)

**Claudia Fischer**

Public Policy Planning, Global Affairs, OpenAI

**Jenn Gamble**

Head, Data Science, Distyl AI

**Chen Goldberg**

Senior Vice-President, Engineering, CoreWeave

**Tom Gruber**

Founder, Humanistic AI

**Gillian K. Hadfield**

Bloomberg Distinguished Professor of AI Alignment and Governance, Johns Hopkins University

**Peter Hallinan**

Director, Responsible Artificial Intelligence, Amazon Web Services (AWS)

**Bennett Hillenbrand**

Agentic Product Safety Lead, MLCommons

**Babak Hodjat**

Chief AI Officer, Cognizant

**Luke Hu Ke**

Co-Founder, Electroder

**Ruchika Joshi**

Fellow, AI Governance Lab, Center for Democracy & Technology

**David Kanter**

Founder and Executive Director, MLCommons

**Sean Kask**

Chief AI Strategy Officer, SAP

**Eddan Katz**

LexLab at US Law Research Fellow, University of San Francisco

**Robert Katz**

Vice-President, Responsible AI and Tech, Salesforce

**Drue Kataoka**

Founder, Drue Kataoka Studios

**Michael Kearns**

Founding Director, Warren Center for Network and Data Sciences, University of Pennsylvania

**Steven Kelly**

Chief Trust Officer, Institute for Security and Technology

**Alex Lebrun**

Co-Founder and Chief Executive Officer, Nabra

**Stefan Leichenauer**

Vice-President, Engineering, SandboxAQ

**Tze Yun Leong**

Professor of Computer Science, National University of Singapore

**Scott Likens**

Global AI and Innovation Technology Lead, PwC

**Ramana Lokanathan**

Senior Vice-President, Engineering and AI, Automation Anywhere

**Nada Madkour**

Non-Resident Research Fellow, University of California, Berkeley

**Richard Mallah**

Principal AI Safety Strategist, Future of Life Institute

**Pilar Manchón**

Senior Director, Engineering, Google

**Darko Matovski**

Founder and Chief Executive Officer, causaLens

**Mao Matsumoto**

Head, NEC Fellow Office, NEC

**Sean McGregor**

Agentic Product Safety Lead, MLCommons

**Risto Miikkulainen**

Professor of Computer Science, The University of Texas at Austin

**Glenn Nethercutt**

Chief Technology Officer, Genesys Cloud Services

**Mark Nitzberg**

Executive Director, Center for Human-Compatible AI, UC Berkeley

**Henrik Ohlsson**

Vice-President; Chief Data Scientist, C3 AI

**Maria Pocovi**

Global Head of Responsible AI, Uniphore

**Vivek Raghunathan**

Senior Vice-President, Engineering, Snowflake

**Reza Rooholamini**

Chief Scientific, Artificial Intelligence  
and Innovation Officer, CCC Intelligent Solutions

**Jason Ruger**

Chief Information Security Officer, Lenovo

**Supheakmungkol Sarin**

Co-Founder, Executive Director, AI Safety Asia (AISA)

**Jun Seita**

Team Director, Medical Science Deep  
Learning Team, RIKEN

**Norihiro Suzuki**

Chairman of the Board, Hitachi Research Institute,  
Hitachi

**Sumit Taneja**

Senior Vice-President and Global Head, Artificial  
Intelligence (AI) Consulting and Implementation,  
EXL Service

**Fabian Theis**

Science Director, Helmholtz Association

**Li Tieyan**

Chief AI Security Scientist, Huawei Technologies

**Lisa Titus**

AI Policy Manager, Meta

**Kush Varshney**

IBM Fellow, IBM

**Anthony Vetro**

President, Chief Executive Officer, IEEE Fellow,  
Mitsubishi Electric Research Laboratories

**Tiffany Wang Xingyu**

Founder, Stealth

**Andrea Wong**

Global Head, Responsible AI Policy,  
Trust and Safety, Bytedance

**Lauren Woodman**

Chief Executive Officer, DataKind

**Xiaohui Yuan**

Senior Expert, Tencent Holdings

**Andy Zhang**

Researcher, Stanford University

**Leonid Zhukov**

Vice-President, Data Science, Boston Consulting  
Group X (BCG X); Director, BCG Global AI Institute,  
Boston Consulting Group (BCG)

## Production

**Laurence Denmark**

Creative Director, Studio Miko

**Martha Howlett**

Editor, Studio Miko

**Jay Kelly**

Designer, Studio Miko

**Cat Slaymaker**

Designer, Studio Miko

## World Economic Forum Public Engagement

**Maxwell Hall**

Creative Editorial Lead

**Floris Landi**

Design Lead

**Gayle Markovitz**

Head, Written and Audio Content

**Sybille Penhirin**

Head, Video and Design

# Endnotes

1. Andrej Karpathy “Functional AI agents will take a decade” <https://www.businessinsider.com/andrej-karpathy-ai-agents-timelines-openai-2025-10> - <https://www.youtube.com/watch?v=BlVnGXEzFow>.
2. Ibid.
3. Ibid.
4. World Economic Forum. (2024). *Navigating the AI Frontier: A Primer on the Evolution and Impact of AI Agents*. <https://www.weforum.org/publications/navigating-the-ai-frontier-a-primer-on-the-evolution-and-impact-of-ai-agents/>.
5. World Economic Forum. (2025). *AI Agents in Action: Foundations for Evaluation and Governance*. <https://www.weforum.org/publications/ai-agents-in-action-foundations-for-evaluation-and-governance/>.
6. World Economic Forum. (n.d.). *Frontier AI Systems and Technologies*. AI Global Alliance. <https://initiatives.weforum.org/ai-global-alliance/safesystems>.
7. Capgemini Research Institute. (2025). *Rise of agentic AI: How trust is the key to human-AI collaboration*. <https://www.capgemini.com/wp-content/uploads/2025/07/Final-Web-Version-Report-AI-Agents.pdf>.
8. International Organization for Standardization (ISO) & International Electrotechnical Commission (IEC). (2023). *ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system*.
9. National Institute of Standards and Technology (NIST). (2023). *Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1)*. <https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf>.
10. National Institute of Standards and Technology (NIST). (2023). *Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1)*. U.S. Department of Commerce. <https://doi.org/10.6028/NIST.AI.100-1>.
11. Microsoft. (2025, 3 December). *Business strategy plan for AI agents*. Microsoft Learn. <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai-agents/business-strategy-plan>.
12. Huang, K., Habler, I., Narajala, V. S., & Sheriff, A. (2025, 13 May). *Agent name service (ANS) for secure AI agent discovery: Towards secure and interoperable agentic AI (Version 1.0)*. Open Web Application Security Project (OWASP). <https://genai.owasp.org/resource/agent-name-service-ans-for-secure-ai-agent-discovery-v1-0/>.
13. Decentralized Identity Foundation (DIF). (2025, 5 March). *Why We Brought MCP-I to DIF (and Why DIF Said Yes)*. <https://blog.identity.foundation/why-dif-said-yes-to-mcp-i/>.
14. Infocomm Media Development Authority (IMDA) & AI Verify Foundation. (2026). *Model AI governance framework for agentic AI*. <https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf>.
15. Infocomm Media Development Authority, & AI Verify Foundation. (2026). *Model AI Governance Framework for Agentic AI*. <https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/mgf-for-agentic-ai.pdf>.
16. Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., et al. (2019). Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 220–229. <https://dl.acm.org/doi/10.1145/3287560.3287596>.
17. See: Anthropic. (n.d.). *Model system cards*. <https://www.anthropic.com/system-cards>; Meta. (n.d.). *System cards*. <https://ai.meta.com/tools/system-cards/>; OpenAI. (2024, 8 August). *GPT-4o system card*. <https://openai.com/index/gpt-4o-system-card/>.
18. Agent2Agent Protocol. (n.d.). *AgentCard*. <https://agent2agent.info/docs/concepts/agentcard/>.
19. International Organization for Standardization (ISO) & International Electrotechnical Commission (IEC). (2023). *ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system*.
20. National Institute of Standards and Technology (NIST). (2023). *Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1)*. <https://doi.org/10.6028/NIST.AI.100-1>.
21. Center for Long-Term Cybersecurity. (2026). *Agentic AI risk-management standards profile*. University of California, Berkeley.
22. EU Artificial Intelligence Act. (2024). *EU Artificial Intelligence Act – Article 14: Human Oversight*. <https://artificialintelligenceact.eu/article/14/>.
23. National Institute of Standards and Technology (NIST). (2023). *Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1)*. <https://doi.org/10.6028/NIST.AI.100-1>.
24. World Economic Forum. (2025). *AI Agents in Action: Foundations for Evaluation and Governance*. <https://www.weforum.org/publications/ai-agents-in-action-foundations-for-evaluation-and-governance>.
25. World Economic Forum. (2024). *Navigating the AI Frontier: A Primer on the Evolution and Impact of AI Agents*. <https://www.weforum.org/publications/navigating-the-ai-frontier-a-primer-on-the-evolution-and-impact-of-ai-agents/>.
26. Open Web Application Security Project (OWASP). (2025). *Agentic AI – Threats and mitigations (Version 1.1)*. OWASP GenAI Security Project. <https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/>.
27. Center for Long-Term Cybersecurity. (2026). *Agentic AI risk-management standards profile*. University of California, Berkeley.
28. International Organization for Standardization (ISO) & International Electrotechnical Commission (IEC). (2023). *ISO/IEC 42001:2023 Information technology — Artificial intelligence — Management system*. ISO.
29. National Institute of Standards and Technology (NIST). (2024). *Artificial intelligence risk management framework: Generative artificial intelligence profile (NIST AI 600-1)*. <https://doi.org/10.6028/NIST.AI.600-1>.
30. Partnership on AI. (2025). *Real-time failure detection in AI agents*.
31. World Economic Forum. (2025). *AI Agents in Action: Foundations for Evaluation and Governance*. <https://www.weforum.org/publications/ai-agents-in-action-foundations-for-evaluation-and-governance/>.



---

COMMITTED TO  
IMPROVING THE STATE  
OF THE WORLD

---

The World Economic Forum, committed to improving the state of the world, is the International Organization for Public-Private Cooperation.

The Forum engages the foremost political, business and other leaders of society to shape global, regional and industry agendas.

---

**World Economic Forum**  
91–93 route de la Capite  
CH-1223 Cologny/Geneva  
Switzerland

Tel.: +41 (0) 22 869 1212  
Fax: +41 (0) 22 786 2744  
contact@weforum.org  
www.weforum.org