MODEL AI GOVERNANCE FRAMEWORK FOR AGENTIC AI

Version 1.0 | Published 22 January 2026

INFOCOMM MEDIA DEVELOPMENT AUTHORITY

Contents

Executive Summary

Agentic AI is the next evolution of AI, holding transformative potential for users and businesses. Compared to generative AI, AI agents can take actions, adapt to new information, and interact with other agents and systems to complete tasks on behalf of humans. While use cases are rapidly evolving, agents are already transforming the workplace through coding assistants, customer service agents, and automating enterprise productivity workflows.

These greater capabilities also bring forth new risks. Agents' access to sensitive data and ability to make changes to their environment, such as updating a customer database or making a payment, are double-edged swords. As we move towards deploying multiple agents with complex interactions, outcomes also become more unpredictable.

Humans must remain accountable and properly manage these risks. While existing governance principles for trusted AI such as transparency, accountability and fairness continue to apply, they need to be translated in practice for agents. Meaningful human control and oversight need to be integrated into the agentic AI lifecycle. Nevertheless, a balance needs to be struck as continuous human oversight over all agent workflows becomes impractical at scale.

The Model AI Governance Framework (MGF) for Agentic AI gives organisations a structured overview of the risks of agentic AI and emerging best practices in managing these risks. If risks are properly managed, organisations can adopt agentic AI with greater confidence. The MGF is targeted at organisations looking to deploy agentic AI, whether by developing AI agents in-house or using third-party agentic solutions. Building on our previous model governance frameworks, we have outlined key considerations for organisations in four areas when it comes to agents:

1. **Assess and bound the risks upfront**

   Organisations should adapt their internal structures and processes to account for new risks from agents. Key to this is first understanding the risks posed by the agent's actions, which depend on factors such as the scope of actions the agent can take, the reversibility of those actions, and the agent's level of autonomy.

   To manage these risks early, organisations could limit the scope of impact of their agents by designing appropriate boundaries at the planning stage, such as limiting the agent's access to tools and external systems. They could also ensure that the agent's actions are traceable and controllable through establishing robust identity management and access controls for agents.

2. **Make humans meaningfully accountable**

   Once the "green light" is given for agentic AI deployment, an organisation should take steps to ensure human accountability. However, the autonomy of agents may complicate traditional responsibility assignments which are tied to static workflows. Multiple actors may also be involved in different parts of the agent lifecycle, diffusing accountability. It is therefore important to clearly define the responsibilities of different stakeholders, both

within the organisation and with external vendors, while emphasising adaptive governance, so that the organisation is set up to quickly understand new developments and update its approach as the technology evolves.

Specifically, "human-in-the-loop" has to be adapted to address automation bias, which has become a bigger concern with increasingly capable agents. This includes defining significant checkpoints in the agentic workflow that require human approval, such as high-stakes or irreversible actions, and regularly auditing human oversight to check that it remains effective over time.

3. **Implement technical controls and processes**

   Organisations should ensure the safe and reliable operationalisation of AI agents by implementing technical measures across the agent lifecycle. During development, organisations should incorporate technical controls for new agentic components such as planning, tools and still-maturing protocols, to address increased risks from these new attack surfaces.

   Before deployment, organisations should test agents for baseline safety and reliability, including new dimensions such as overall execution accuracy, policy adherence, and tool use. New testing approaches will be needed to evaluate agents.

   During and after deployment, as agents interact dynamically with their environment and not all risks can be anticipated upfront, it is recommended to gradually roll out agents alongside continuous monitoring after deployment.

4. **Enable end-user responsibility**

   Trustworthy deployment of agents does not rely solely on developers, but also on end-users using them responsibly. To enable responsible use, as a baseline, users should be informed of the agent's range of actions, access to data, and the user's own responsibilities. Organisations should consider layering on training to equip employees with the knowledge required to manage human-agent interactions and exercise effective oversight, while maintaining their tradecraft and foundational skills.

This is a living document. We have worked with government agencies and leading companies to collate current best practices, but this is a fast-developing space, and best practices will evolve. This framework will need to be continuously updated to keep pace with new developments. We invite feedback to refine the framework, and case studies demonstrating how the framework can be applied for responsible agentic deployment.

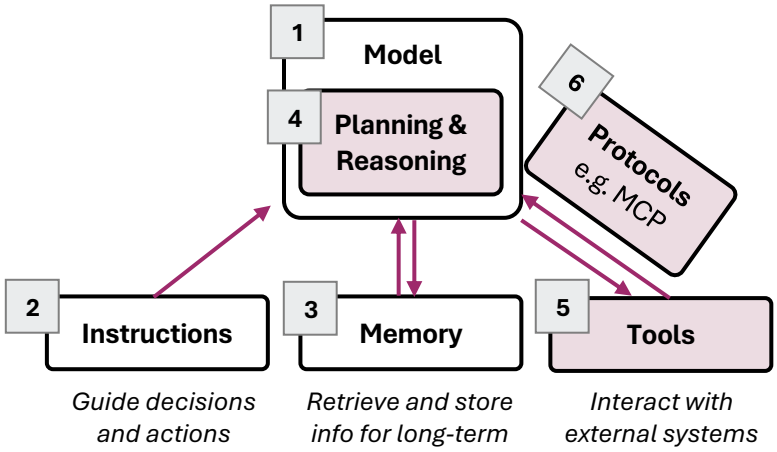
1 Introduction to Agentic AI

1.1 What is Agentic AI?

Agentic AI systems are systems that can plan across multiple steps to achieve specified objectives, using AI agents.[1] There is no consensus on what defines an agent, but there are certain common features – agents usually possess some degree of independent planning and action taking (e.g. searching the web or creating files) over multiple steps to achieve a user-defined goal.[2]

In this framework, we focus on agents built on language models, which are increasingly being adopted. Such agents use a small, large, or multimodal large language model (SLM, LLM, or MLLM) as its brain to make decisions and complete tasks. However, it is worth noting that software agents are not a new concept and other types of agents exist, such as those which use deterministic rules, or other neural networks, to make decisions.[3]

1.1.1 Core components of an agent



Core components of a simple agent[4]

As agents are built on top of language models, it is helpful to start with the core components of a simple LLM-based app.

1. **Model**: an SLM, LLM or MLLM that serves as the central reasoning and planning engine, or the "brain" of the agent. It processes instructions, interprets user inputs, and generates contextually appropriate responses.

---

1 Adapted from Cyber Security Agency of Singapore (CSA), Draft Addendum on Securing Agentic AI.
2 See International AI Safety Report.
3 See World Economic Forum (WEF), AI Agents in Action: Foundations for Evaluation and Governance.
4 Adapted from GovTech Singapore, Agentic Risk & Capability Framework, CSA Singapore, Draft Addendum on Securing Agentic AI and Anthropic, Building Effective Agents).

2. **Instructions**: Natural language commands that define an agent's role, capabilities, and behavioural constraints e.g. a system prompt for an LLM.
3. **Memory**: Information that is stored and accessible to the LLM, either in short or long-term storage. Sometimes added to allow the model to obtain information from previous user interactions or external knowledge sources.

An agent uses the model, instructions and memory in similar ways as an LLM-based app. In addition, it has other components that enable it to complete more complex tasks:

4. **Planning and reasoning**: The model is usually trained to reason and plan, meaning that it can output a series of steps needed for a task.
5. **Tools:** Tools enable the agent to take actions and interact with other systems, such as writing to files and databases, controlling devices, or performing transactions. The model calls tools to complete a task.
6. **Protocols:** This is a standardised way for agents to communicate with tools and other agents. For example, the Model Context Protocol (MCP) has been developed for agents to communicate with tools,[5] whereas the Agent2Agent Protocol (A2A) defines a standard for agents to communicate with each other.[6]

1.1.2 Multi-agent setups

In an agentic system, it is common for multiple agents to be set up to work together. This can sometimes improve performance, by allowing each agent to specialise in a certain function or task and work in parallel.[7]

Three common design patterns for multi-agent systems are:[8]

- **Sequential**: Agents work one after another in a linear workflow. Each agent's output becomes the next agent's input.
- **Supervisor**: One supervising agent coordinates specialised agents under it.
- **Swarm**: Agents work at the same time, handing off to another agent when needed

1.1.3 How agent design affects the limits and capabilities of each agent

While each agent may have the same core components, the design of each component can significantly affect what the agent can do. It is generally helpful to distinguish between two concepts when considering what an agent can do:[9]

- **Action-space** (or authority, capabilities): Range of actions the agent is permitted to take, determined by the tools it is allowed to use, transactions it can execute, etc.

---

5 See Anthropic, Model Context Protocol.
6 See Google, Agent2Agent Protocol.
7 See LangChain, Benchmarking Multi-Agent Architectures.
8 Adapted from AWS, Multi-Agent Collaboration Patterns with Strands Agents and Amazon Nova.
9 See WEF, AI Agents in Action: Foundations for Evaluation and Governance.

- **Autonomy** (or decision-making): Degree to which an agent can decide when and how to act towards a goal, such as by defining the steps to be taken in a workflow. This can be determined by its instructions and level of human involvement.

Action-space

An agent's action-space mainly depends on the tools it has access to, which can affect:

- **Systems it can access:**
  - Sandboxes only: Sandboxed tools (e.g. for code execution, data analysis) that cannot affect any other system
  - Internal systems: Tools internal to the organisation, such as being able to search and update the organisation's databases
  - External systems: Tools that enable the agent to access external services, such as retrieving and updating data through third-party pre-defined APIs.
- **Actions it can take in relation to the system it can access:**
  - Read vs write: An agent may only be able to read and retrieve information from a system, rather than write to and modify data within the system.

An emerging modality of agentic AI is a computer use agent, whose primary tool is access to a computer and browser. This means that it can take any action that a human can take with a computer and browser without having to rely on specifically defined tools and APIs. This significantly increases what the agent can access and do.

Autonomy

An agent's autonomy mainly depends on its instructions component and the level of human involvement in the agentic system.

In terms of instructions, an agent can be given differing level of instructions:

- **Detailed instructions and SOP:** An agent instructed to follow a detailed SOP to complete a task would be limited in the decisions it can make at each stage.
- **Using its own judgment:** An agent instructed to use its own judgment to complete a task would have more freedom to define its plan and workflow.

Another relevant factor is the level of human involvement. When interacting with an agent, a human can be involved to different levels:[10]

- **Agent proposes, human operates:** The human directs and approves every step taken by an agent.
- **Agent and human collaborate:** The human and agent work together. The agent requires human approval at significant steps, such as before writing to a database or making a payment. However, the human can intervene anytime by taking over the agent's work or pausing the agent and requesting a change.

[10] See Knight First Amendment Institute at Columbia University, Levels of Autonomy for AI Agents.

- **Agent operates, human approves:** The agent requires human approval only at critical steps or failures, such as deleting a database or making a payment above a predefined amount.
- **Agent operates, human observes:** The agent does not require human approval as it completes its task, though its actions may be audited after the fact.

1.2 Risks of Agentic AI

1.2.1 Sources of risk

The new components of an agent constitute new sources of risks.[11] **The risks themselves are familiar** – fundamentally, agents are software systems built on LLMs. They inherit traditional software vulnerabilities (such as SQL injection) and LLM-specific risks (such as hallucination, bias, data leakage and adversarial prompt injections).[12]

However, the risks can manifest differently through the different components. For example:

- **Planning and reasoning:** An agent can hallucinate and make a wrong plan to complete a task.
- **Tools:** An agent can hallucinate by calling non-existent tools or calling tools with the wrong input, or calling tools in a biased manner. As tools connect the agent to external systems, prompt or code injections can also manipulate the agent to exfiltrate or otherwise manipulate the data it has access to.
- **Protocols:** Finally, as new protocols emerge to handle agent communication, they can also be poorly deployed or compromised e.g. an untrusted MCP server deployed with code to exfiltrate the user's data.

As components within an agent or multiple agents interact, risks can also arise at the system level.[13] For example:

- **Cascading effect:** A mistake by one agent can quickly escalate as its outputs are passed onto other agents. For example, in supply chain management, a hallucinated inventory figure from one agent could potentially cause downstream agents to reorder excessive or insufficient stock.
- **Unpredictable outcomes:** Agents working together can also compete or coordinate in unintended ways. For example, in manufacturing, different agents may be involved in managing machines and inventory. While coordinating to meet production goals, the agents might interact unpredictably due to complex optimisation algorithms and over or under-prioritise one resource or machine, leading to unexpected bottlenecks.

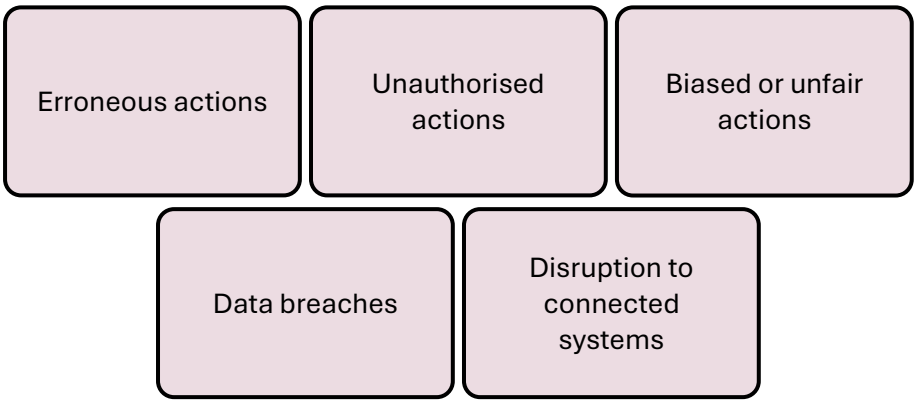
---

11 BCG highlighted examples of new risks from agents e.g. agents that optimize their own goals locally may create instability across the system, flawed behaviour by one agent may spread to other agents (see What Happens When AI Stops Asking Permission?)

12 Adapted from CSA, Draft Addendum on Securing Agentic AI.

13 See WEF, AI Agents in Action: Foundations for Evaluation and Governance, which highlighted a new class of failure modes, linked to potentially misaligned interactions in multi-agent systems e.g. orchestration drift, semantic misalignment, interconnectedness and cascading effects.

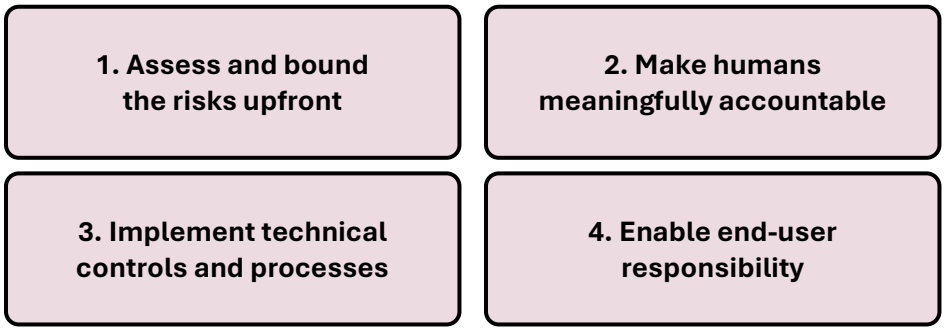
1.2.2 Types of risk



Because agents take actions in the real world, when they malfunction, it can lead to harmful real-world impact. Organisations should be aware of these negative outcomes:

- **Erroneous actions**: Incorrect actions such as an agent fixing appointments on the wrong date or producing flawed code. The exact harmful outcome depends on the action in question, e.g. flawed code can lead to exploited security vulnerabilities, and wrong medical appointments may affect a patient's health outcomes.
- **Unauthorised actions**: Actions taken by the agent outside its permitted scope or authority, such as taking an action without escalating it for human approval based on a company policy or standard operating procedure.
- **Biased or unfair actions**: Actions that lead to unfair outcomes, especially when dealing with groups of different profiles and demographics, such as biased vendor selection in procurement, disbursements of grants, and/or hiring decisions.
- **Data breaches**: Actions that lead to the exposure or manipulation of sensitive data. Such data may be personally identifiable information or confidential information e.g. customer details, trade secrets, and/or internal communications. This can be due to a security breach, where attackers exploit agents to reveal private information, or an agent disclosing sensitive data due to a failure to recognise it as sensitive.
- **Disruption to connected systems**: As agents interact with other systems, they can cause disruption to connected systems when they are compromised or malfunction e.g. deleting a production codebase, or overwhelming external systems with requests.

2 Model AI Governance Framework for Agentic AI



Four dimensions of the MGF for Agentic AI

The MGF for Agentic AI builds on the responsible AI practices for organisations set out in MGF (2020)[14] by highlighting emerging best practices to address new concerns from agentic AI. This is so that organisations can develop and use agentic AI with the requisite knowledge and judgment.

The framework begins with helping organisations to **assess and bound the risks upfront**. It highlights new risks that should be considered during risk assessment, and design considerations at the planning stage to limit the potential scope of impact of the agents, as well as ensure that agents are traceable and controllable.

While agents may act autonomously, human responsibility continues to apply. Once the "green light" is given to deploy agentic AI, an organisation should take immediate steps to **make humans meaningfully accountable**. This includes clearly defining responsibility across multiple actors within and outside the organisation involved in the agent lifecycle; and taking measures to ensure that human-in-the-loop remains effective over time notwithstanding automation bias.

To ensure safe and reliable operationalisation of agents, an organisation should adopt **technical controls and processes** across the AI lifecycle. During development, guardrails for new components in AI agents such as planning and tools should be implemented. Before deployment, agents should be tested for baseline safety and reliability. After deployment, agents should be continuously monitored as they interact dynamically with their environment.

Finally, trustworthy deployment of agents does not rest solely on developers, but also on end-users. Organisations are responsible for **enabling end-user responsibility** by equipping them with essential information to use agents appropriately and exercise effective oversight, while maintaining their tradecraft and foundational skills.

[14] See Model AI Governance Framework (2nd Ed).

2.1 Assess and bound the risks upfront

Agents bring new risks, especially in their access to sensitive data and ability to change their environment through action-taking. Their adaptive, autonomous and multi-step nature also increases the potential for unexpected actions, emergent risks and cascading impacts. Organisations should consider these new dimensions as part of risk assessment, and limit the scope of impact of their agents by designing appropriate boundaries at an early stage.

When planning for the use of agentic AI, organisations should consider:

- **Determining suitable use cases for agent deployment** by considering agent-specific factors that can affect the likelihood and impact of the risk.
- **Design choices to bound the risks upfront** by applying limits on agent's access to tools and systems and defining a robust identity and permissions framework.

2.1.1 Determine suitable use cases for agent deployment

Risk identification and assessment is the first step when considering if an agentic use case is suitable for development or deployment. Risk is a function of likelihood (probability of the risk manifesting) and impact (severity of impact if the risk manifests).

The following non-exhaustive factors affect the level of risk of an agentic use case:

Factors affecting impact		
Factor	**Description**	**Illustration**
Domain and use case in which agent is being deployed	Level of tolerance of error in the domain and use case in which the agent is being deployed to	Agent executing financial transactions which require a high degree of accuracy, vs agent that summarises internal meetings
Agent's access to sensitive data	Whether the agent can access sensitive data, such as personal information or confidential data	Agent that requires access to personal customer data gives rise to the risk of leaking such data, vs agent who only has access to publicly available information
Agent's access to external systems	Whether the agent can access external systems	Agent that sends data to third-party APIs can leak data to these third parties, or disrupt these systems by making too many requests, vs agent that only has access to sandboxed or internal tools
Scope of agent's actions	Whether an agent can only read from or modify the data and systems it has access to	*Read vs write:* Agent that can only read from a database vs being able to write to it *Many tools vs a few:* Agent that can only choose from a few pre-defined tools, vs an agent who has unlimited access to a browser tool

Reversibility of agent's actions	If the agent can modify data and systems, whether such modifications are easily reversed	Agent that schedules meetings vs agent that sends email communications to external parties

Factors affecting likelihood		
Factor	**Description**	**Illustration**
Agent's level of autonomy	Whether the agent can define the entire workflow or must follow a well-defined procedure. A higher level of autonomy can result in higher unpredictability, increasing likelihood of error.	Agent is provided with a SOP and instructed to follow it when carrying out a task, vs agent is instructed to use its best judgment to select and execute every step
Task complexity	How complex the task is, in relation to the number of steps required to complete it and the level of analysis required at each step. A higher level of complexity similarly increases unpredictability and the likelihood of error.	Agent is required to extract key action points from a meeting transcript, vs agent is tasked to follow a nuanced data sharing policy when handling external requests for information
Agent's access to external systems	Whether the agent is exposed to external systems, and who maintains these systems. A higher level of exposure makes the agent more vulnerable to prompt injections and cyberattacks.	Agent can only access an internal knowledge base which is maintained by trusted internal teams, vs an agent who can access the web containing untrusted data

Threat modelling also makes risk assessment more rigorous by systematically identifying specific ways in which an attacker may take to compromise the system. Common security threats to agentic systems include memory poisoning, tool misuse, and privilege compromise.[15] As agentic systems (especially multi-agent systems) can become very complex, it is often useful to use a method called taint tracing to map out all the workflows and interactions to track how untrusted data can move through the system. For more information on how to perform threat modelling and taint tracing for agentic systems, organisations may refer to CSA's Draft Addendum on Securing Agentic AI.

The relationship between threat modelling and risk assessment

Threat modelling augments the risk assessment process by generating contextualised threat events with well-described sequence of actions, activities and scenarios that the attacker may take to compromise the system. With more relevant threat events, risk assessments will be more rigorous and robust, resulting in more targeted controls and effective layered defence. Since risk assessment is continuous, the threat model should be regularly updated.

Adapted from CSA, Guide to Cyber Threat Modelling

[15] For a more comprehensive coverage of potential security threats to agentic AI systems, see OWASP, Agentic AI – Threats and Mitigations.

2.1.2 Bound risks through design by defining agents limits and permissions

Having selected an appropriate agent use case, organisations can further bound the risks by defining appropriate limits and permission policies for each agent.

Agent limits

Organisations should consider defining limits on:

- **Agent's access to tools and systems:** Define policies that give agents only the minimum tools and data access needed for it to complete its task.[16] For example, a coding assistant may not require access to a web search tool, especially if it already has curated access to the latest software documentation.
- **Agent's autonomy:** For process-driven tasks, SOPs and protocols are frequently used to improve consistency and reduce unpredictability. [17] Define similar SOPs for agentic workflows that an agent is constrained to follow, rather than giving the agent the freedom to define every step of the workflow.
- **Agent's area of impact:** Design mechanisms and procedures to take agents offline and limit their potential scope of impact when they malfunction. This can include running agents in self-contained environments with limited network and data access, particularly when they are carrying out high-risk tasks such as code execution.[18]

Agent identity

Identity management and access control is one of the key means in which organisations enable traceability and accountability today for humans. As agents become more autonomous, identity management has to be extended to agents as well to track individual agent behaviour and establish who holds accountability for each agent.

This is an evolving space, and gaps exist today in terms of handling agent identity robustly. For example, current authorisation systems typically have pre-defined, static scopes. However, to operate safely in more complex scenarios, agents require fine-grained permissions that may change dynamically depending on the context, risk levels, and task objectives. Current authentication systems are also typically based on a single, unique individual. Such systems face difficulty in handling complex agent setups, such as when agents act for multiple human users with different permissions, or recursive delegation scenarios where agents spin up multiple sub-agents.[19]

---

16 See PwC, The rise – and risks – of agentic AI.

17 Grab introduced an LLM agent framework leveraging on Standard Operating Procedures (SOPs) to guide AI-driven execution (see Introducing the SOP-driven LLM agent frameworks).

18 See McKinsey, Deploying agentic AI with safety and security: A playbook for technology leaders.

19 For a more comprehensive treatment of how current identity systems may face challenges when catering to agentic AI, see OpenID, Identity Management for Agentic AI.

Solutions are being developed to address these issues, such as integrating well-established standards like OAuth 2.0 into MCP.[20] The industry is also developing new standards and solutions for agents, such as decentralised identity management and dynamic access control.[21]

In the interim, organisations should consider these best practices to enable agent control and traceability:

- **Identification:** An agent should have its own unique identity, such that it can identify itself to the organisation, its human user, or other agents. However, an agent's identity may need to be tied to a supervising agent, a human user, or an organisational department for accountability and tracking. Additionally, the different capacities in which an agent acts (e.g. independently or on behalf of a specified human user) should also be recorded.
- **Authorisation:** An agent can have pre-defined permissions based on its role or the task at hand, or its permissions may be dynamically set by its authorising human user, or a combination of both. As a rule of thumb, the human user should not be able to set permissions for the agent greater than what the human user is himself authorised to do. Such delegations of authority should be clearly recorded.

Evaluating the residual risks

Residual risk is the risk that remains after mitigation measures have been applied. It is important to note that there will always be a level of risk remaining, even after efforts are taken to identify appropriate agentic use cases and define limits on any agents, especially given how quickly agentic AI is evolving. Ultimately, organisations should evaluate and determine if the residual risk for their agentic deployment is of a tolerable level and can be accepted.

[20] See MCP specifications for Authentication support , Authorisation support.

[21] See proposed framework for agentic identity by Cloud Security Alliance, Agentic AI Identity & Access Management: A New Approach.

2.2 Make humans meaningfully accountable

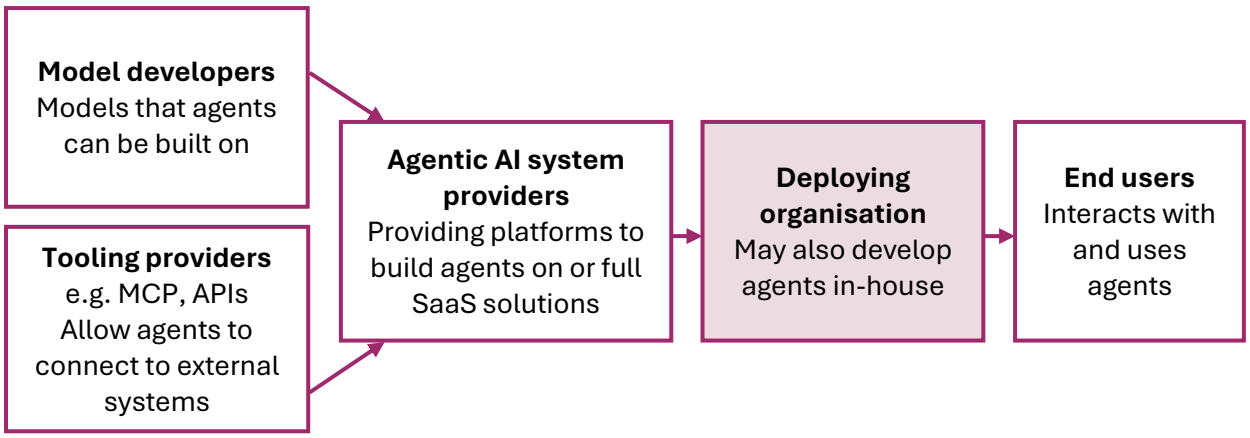
The organisations that deploy agents and the humans who oversee them remain accountable for the agents' behaviours and actions. But it can be challenging to fulfil this accountability when agent actions emerge dynamically and adaptively from interactions instead of fixed logic. Multiple stakeholders may also be involved in different parts of the agent lifecycle, diffusing accountability. Finally, automation bias, or the tendency to over-trust an automated system, especially when it has performed reliably in the past, becomes a bigger concern as humans supervise increasingly capable agents.

To address these challenges to human accountability, organisations should consider:

- **Clear allocation of responsibilities within and outside the organisation**, by establishing chains of accountability across the agent value chain and lifecycle, while emphasising adaptive governance, so that the organisation is set up to quickly understand new developments and update their approach as the technology evolves.
- **Measures to enable meaningful human oversight of agents**, such as requiring human approval at significant checkpoints, auditing the effectiveness of human approvals, and complementing these measures with automated monitoring.

2.2.1 Clear allocation of responsibilities within and outside the organisation

As deployers, organisations and humans remain accountable for the decisions and actions of agents. However, as with AI, the value chain for agentic AI involves multiple actors. Organisations should consider the allocation of responsibility both within their organisation, and vis-à-vis other organisations along the value chain.



Simplified agentic AI value chain[22]

[22] For a more comprehensive list of potential stakeholders involved in the agentic AI ecosystem, see CSA and FAR.AI, Securing Agentic AI: A Discussion Paper.

Within the organisation

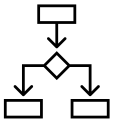
Within the organisation, organisations should allocate responsibilities for different teams across the agent lifecycle. While each organisation is structured differently, this is an illustration of how such responsibilities may be allocated across different teams:

Key decision makers

Who: Leaders who define strategic decisions and high-level policies for the organisation e.g. board members, C-suite executives, managing directors, or department leaders.

Key responsibilities can include:

- Setting high-level goals for use of agents
- Defining permitted operational use cases for agents, including limits on agent's data access
- Setting the overall governance approach, including risk management frameworks and escalation processes



Product teams

Who: These roles oversee the translation of stakeholder needs or business goals into a technical agentic solution e.g. Product Managers, UI / UX Designers, AI Engineers, Software Engineers

Key responsibilities can include:

- Defining the design and requirements for agents, as well as any feature controls or phased rollouts
- Reliable implementation of agents i.e. development, pre-deployment testing and post-deployment monitoring across the agent lifecycle
- Educating users on responsible use of agentic product

Cybersecurity teams

Who: These roles oversee the protection of agentic systems from cyber threats, by implementing and managing security measures, identifying vulnerabilities, and responding to incidents e.g. Chief Security Officer, Cyber Security Specialist, Penetration Tester

Key responsibilities can include:

- Defining baseline security guardrails and secure-by-design templates that technical teams should implement or adapt to the agentic system being deployed
- Conducting regular red teaming and threat modelling

Users

Who: Any individual who utilises the output of the agents to contribute to an organisational goal e.g. company employees making decisions or automating workflows and practices.

Key responsibilities can include:

- Ethical and responsible usage of agents
- Attending required training, complying with usage policies, timely reporting of bugs or issues with agents

Developing internal capabilities for adaptive governance

All teams involved in the agentic AI lifecycle should also develop internal capabilities to understand agentic AI. As the technology is quickly evolving, being aware of the improvements and limitations of new agentic developments, such as new modalities like computer use agents, or new evaluation frameworks for agents, allow organisations to quickly adapt their governance approach to new developments.

Outside the organisation

Organisations may also need to work with external parties when deploying agents e.g. model developers, agentic AI providers, or hosts of external MCP servers or tools.

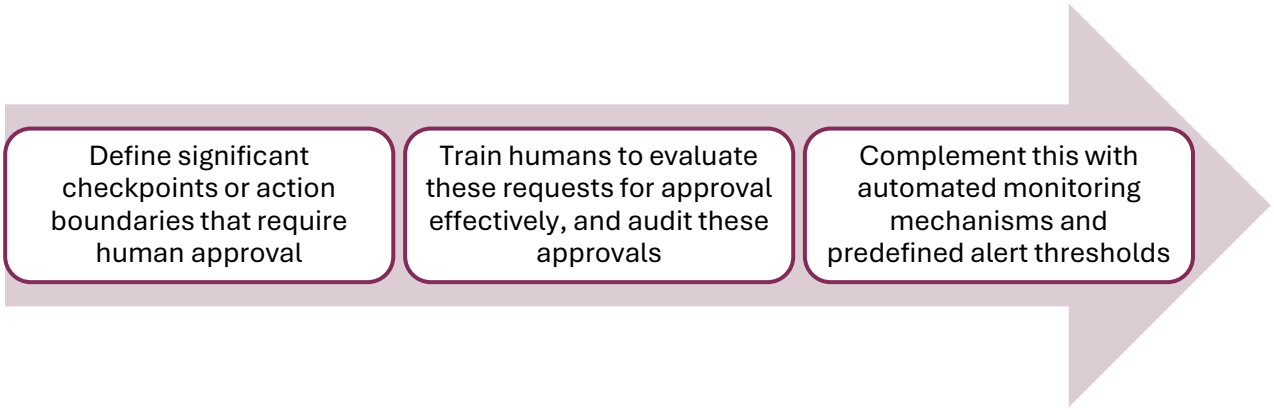
In these cases, organisations should similarly ensure that there are measures in place to fulfil its own accountability. Some agent-specific considerations are:

- **Clarify distribution of obligations** in any terms and conditions or contracts between the organisation and the external party. In particular, organisations should consider provisions to address any security arrangements, performance guarantees, or data protection and confidentiality. Where there are gaps, the organisation should reassess if the agentic deployment meets its risk tolerance.
- **Features to maintain security and control.** Organisations should consider if the external party's product offers features for the organisation to maintain a sufficient level of security or control. This includes strong authentication measures such as scoped API keys, per-agent identity tokens, and robust observability such as the logging of tool calls and access history. Where such features are lacking, organisations should consider alternative or in-house solutions, or scoping down the agentic use case, such as restricting access to sensitive data.

End users

Organisations may deploy agents to users within or outside their organisation. In doing so, organisations should ensure that users are provided sufficient information to hold the organisation accountable, as well as any information relating to the user's own responsibilities. More information can be found in Enabling end-user responsibility below.

2.2.2 Design for meaningful human oversight



Setting up a system for effective human supervision

Organisations should define significant checkpoints or action boundaries that require human approval, especially before sensitive actions are executed. This can include:[23]

- **High-stakes actions and decisions** e.g. editing of sensitive data, final decisions in high-risk domains (such as healthcare or legal), actions that may trigger liability
- **Irreversible actions** e.g. permanently deleting data, sending communications, making payments
- **Outlier or atypical behaviour** e.g. when agent accesses a system or database outside of its work scope, when agent selects a delivery route that is twice as long as the median distance
- **User-defined**. Agents may act on behalf of users who have different risk appetites. Beyond organisation-defined boundaries, users may be given the option to define their own boundaries e.g. requiring approval for purchases above a certain amount

Apart from considering when approvals are required, organisations should also consider what form approvals should take. These considerations include:

- **Keep approval requests contextual and digestible.** When asking humans for approval, keep the request short and clear, instead of providing long logs or raw data that may be challenging to decipher and understand.
- **Consider the form of human input required.** For straightforward actions such as accessing a database, the human user can simply approve or reject. For more complex cases, such as reviewing an agent's plan before execution, it may be more productive for the human to edit the plan before giving the agent the go-ahead.

Organisations should implement measures to ensure continued effectiveness of human oversight, particularly as humans remain susceptible to alert fatigue and automation bias. These measures can include:

[23] For further examples of where human involvement may be considered, see Partnership on AI, Prioritising real-time failure detection in AI agents).

- **Training humans to identify common failure modes** e.g. inconsistent agent reasoning, agents referring to outdated policies
- **Regularly auditing the effectiveness of human oversight**

Finally, human oversight should be complemented with automated real-time monitoring to escalate any unexpected or anomalous behaviour. This can be done by implementing alerts for certain logged events (e.g. attempted unauthorised access or multiple failed attempts to call a tool), using data science techniques to identify anomalous agent trajectories, or using agents to monitor other agents. For more information, see Continuous testing and monitoring below.

2.3 Implement technical controls and processes

The agentic components that differentiate agents from simple LLM-based applications necessitate additional controls during the key stages of the implementation lifecycle.

Organisations should consider:

- **During design and development, design and implement technical controls**. The new components and capabilities of agents also necessitate new and tailored controls. Depending on the agent design, implement controls such as tool guardrails and plan reflections. Further, limit the agent's impact on the external environment by enforcing least-privilege access to tools and data.
- **Pre-deployment, test agents for safety and security.** As with all software, testing before deployment ensures that the system behaves as expected. Specifically for agents, test for new dimensions such as overall task execution, policy adherence and tool use accuracy, and test at different levels and across varied datasets to capture the full spectrum of agent behaviour.
- **When deploying, gradually roll out agents and continuously monitor them in production.** The autonomous nature of agents and the changing environment makes it challenging to account for and test all possible outcomes before deployment. Hence it is recommended to roll out agents gradually, supported with real-time monitoring post-deployment to ensure that agents function safely.

2.3.1 During design and development, use technical controls

Organisations should design and implement technical controls in the agentic AI system to mitigate identified risks. For agents specifically, in addition to baseline software and LLM controls, consider adding controls for:

- New agentic components, such as planning and reasoning and tools
- Increased security concerns from the larger attack surface and new protocols

For illustration, these are some sample controls for agents. For a more comprehensive list, organisations can refer to CSA's Draft Addendum on Securing Agentic AI and GovTech's Agentic Risk and Capability Framework.

Planning	• Prompt agent to reflect on whether its plan adheres to user instructions • Prompt the agent to summarise its understanding and request clarification from the user before proceeding • Log the agent's plan and reasoning for the user to evaluate and verify
Tools	• Configure tools to require strict input formats • Apply the principle of least privilege to limit tools available to each agent, enforced through robust authentication and authorisation • For data-related tools: ○ Do not grant agent write access to tables in sensitive databases unless strictly required

	○ Configure agent to let user take over control when keying in sensitive data (e.g. passwords, API keys)
Protocols	• Use standardised protocols where applicable (e.g. agentic commerce protocols when agent is handling a financial transaction) • For MCP servers: ○ Whitelist trusted servers and only allow agent to interact with servers on that whitelist ○ Sandbox any code execution

2.3.2 Before deploying, test agents

Organisations should test agents for safety and security before deployment. This provides confidence that the agents work as expected and controls are effective. Best practices on software and LLM testing are still relevant, such as unit and integration testing for software systems, as well as selecting representative datasets, and useful metrics and evaluators for LLM testing. Organisations can refer to previous guidance, such as the Starter Kit for testing of LLM-based apps for safety and reliability.

However, organisations should adapt their testing approaches for agents. Some considerations include:

- **Testing for new risks:** Beyond producing incorrect outputs, agents can take unsafe or unintended actions through tools. Organisations can consider testing for:[24]
  - **Overall task execution**: Whether agent can complete task accurately
  - **Policy compliance**: Whether an agent follows defined SOPs and routes for human approval when required
  - **Tool calling**: Whether an agent calls the right tools, with the right permissions, with the right inputs and in the right order
  - **Robustness**: As agents are expected to react and adapt to real-world situations, test for their response to errors and edge cases

- **Testing entire agent workflows:** Agents can take multiple steps in sequence without human involvement. Thus, beyond testing an agent's final output, agents should be tested across their entire workflow, including reasoning and tool calling.

- **Testing agents individually and together:** Beyond individual agents, testing should be carried out at the multi-agent system level, to understand any emergent risks and behaviours when agents collaborate, such as competitive behaviours or the impact on other agents when one agent has been compromised.

- **Testing in real or realistic environments:** As agents may be expected to navigate real-world situations, testing should occur in a properly configured execution environment that mirrors production as closely as possible, such as using tool integrations, external APIs, and sandboxes that behave as they would in deployment. However, organisations should

[24] For an example of new agentic aspects to test for, see Microsoft Foundry, Agent evaluators.

calibrate the need for realism against the risk of prematurely allowing agents to access tools that affect the real world.

- **Testing repeatedly and across varied datasets:** Agent behaviour is inherently stochastic and context-dependent. Testing should thus be done at scale and across varied datasets to observe any unexpected low-probability behaviours, especially if they are high-impact. This requires generating test datasets that cover different conditions that agents may encounter and running these tests multiple times, including minor perturbations where needed.

- **Evaluating test results at scale:** Reliably evaluating test results at scale is a known challenge for LLM testing. Agents add a further layer of complexity as their workflows can be long and contain unstructured information that cannot be easily processed by humans or automated scripts. Organisations may consider using different evaluation methods for different parts of the agentic workflow (e.g. deterministic tests for structured tool calls vs LLM or human evaluation for unstructured agent reasoning). However, there is still a need to evaluate agents holistically, so that agent patterns across steps can be evaluated. Current industry solutions thus include defining LLMs or agents to evaluate other agents.[25]

2.3.3 When deploying, continuously monitor and test

As agents are adaptive and autonomous, organisations should consider mechanisms to respond to unexpected or emergent risks when deploying agents.

Gradual deployment of agents

Organisations should consider gradually rolling out agents into production to control the amount of risk exposure. Such rollouts can be controlled based on:

- Users of agents e.g. rolling out to trained or experienced users first
- Tools and protocols available to agent e.g. restricting agents to more secure, whitelisted MCP servers first
- Systems exposed to agent e.g. using agents in lower-risk internal systems first

Continuous testing and monitoring

Organisations should continuously monitor and log agent behaviour post-deployment, and establish reporting and failsafe mechanisms for agent failures or unexpected behaviours. This allows the organisation to:

- **Intervene in real-time**: When potential failures are detected, stop agent workflow and escalate to a human supervisor e.g. if agent attempts unauthorised access
- **Debug when incidents happen**: Logging and tracing each step of an agent workflow and agent-to-agent interactions help to identify points of failure
- **Audit at regular intervals:** This ensures that the system is performing as expected.

[25] For an example of agent evaluation solutions, see AWS Labs, Agent Evaluation.

Monitoring and observability are not new concepts, but agents introduce some challenges. As agents execute multiple actions at machine speed, organisations face the issue of extracting meaningful insights from the voluminous logs generated by monitoring systems. This becomes more difficult when high-risk anomalies are expected to be detected in real-time and surfaced as early as possible.

Key considerations when setting up a monitoring system include:

- **What to log:** Organisations should determine their objectives for monitoring (e.g. real-time intervention, debugging, integration between components) to identify what to log. In doing so, prioritise monitoring for high-risk activities such as updating database records or financial transactions.

- **How to effectively monitor logs:** Organisations can consider approaches such as:

    - **Defining alert thresholds:**
        - **Programmatic, threshold-based**: Define alerts when agents trigger thresholds e.g. agent attempts unauthorised access or makes too many repeated tool calls within a specified timeframe.
        - **Outlier / anomaly detection**: Use data science or deep learning techniques to process agent signals and identify anomalous behaviour that may indicate malfunctions.
        - **Agents monitoring other agents**: Design agents to monitor other agents in real-time, flagging any anomalies or inconsistencies.

    - **Defining specific interventions**: For each alert type, consider what the level of intervention should be. Some degree of human review should be incorporated, proportionate to the risk level. For example, lower-priority alerts can be flagged for review at a scheduled time, whereas higher-priority ones might require temporarily halting agent execution until a human reviewer can assess. In the event of catastrophic agentic malfunction or compromise, commensurate measures such as termination and fallback solutions should be considered.

Finally, continuously test the agentic system even post-deployment to ensure that it works as expected and is not affected by model drift or other changes in the environment.

2.4 Enable end-user responsibility

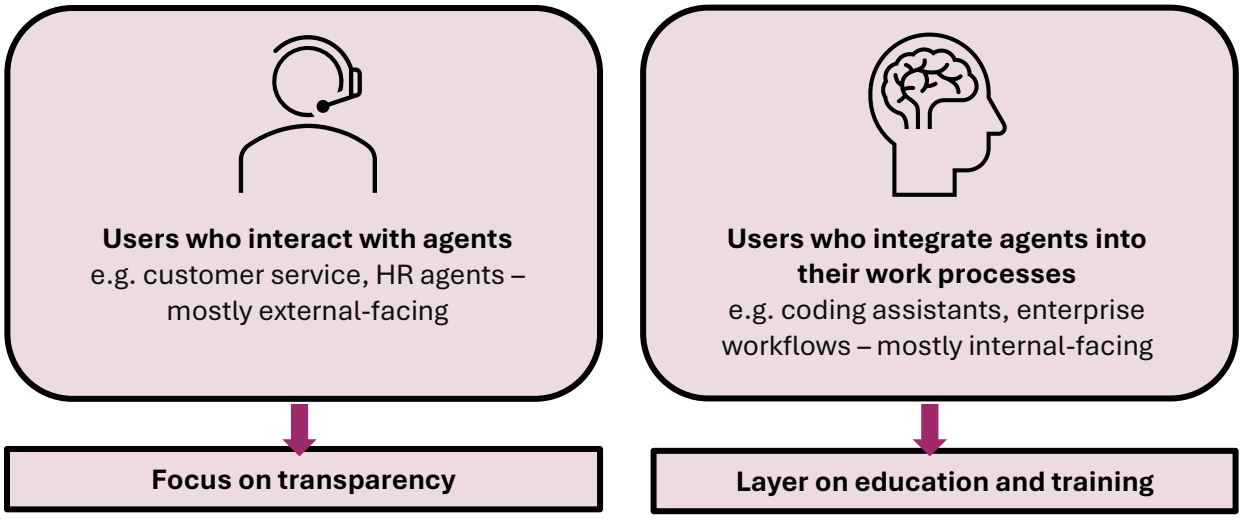
Ultimately, end users are the ones who use and rely on agents, and human accountability also extends to these users. Organisations should provide sufficient information to end users to promote trust and enable responsible use.

Organisations should consider:

- **Transparency**: Users should be informed of the agents' capabilities (e.g. scope of agent's access to user's data, actions the agent can take) and the contact points whom users can escalate to if the agent malfunctions.
- **Education**: Users should be educated on proper use and oversight of agents (e.g. training should be provided on an agent's range of actions, common failure modes like hallucinations, usage policies for data), as well as the potential loss of trade craft i.e. as agents take over more functions, basic operational knowledge could be eroded. Hence sufficient training (especially in areas where agents are prevalent) should be provided to ensure that humans retain core skills.

2.4.1 Different users, different needs

Organisations should cater to different users with different information needs, to enable such users to use AI responsibly. Broadly, there are two main archetypes of end-users – those who interact with agents, and those who integrate agents into their work processes or oversee them.



2.4.2 Users who interact with agents

Such users usually interact with agents that act on behalf of the organisation, e.g. customer service or sales agents. These agents tend to be external facing, although they can also be deployed within the organisation e.g. a human resource agent that interacts with other users in the organisation.

For these users, focus on transparency. Organisations should share pertinent information to foster trust and facilitate proper usage of agents. Such information can include:

- **User's responsibilities**: Clearly define the user's responsibilities, such as asking the user to double-check all information provided by the agent.
- **Interaction**: Declare upfront that the users are interacting with agents.
- **Agents' range of actions and decisions**: Inform the users on the range of actions and decisions that the agent is authorised to perform and make.
- **Data:** Be clear on how user data is collected, stored, and used by the agents, in accordance with the organisation's data privacy policies. Where necessary, obtain explicit consent from users before collecting or using their data for the agents.
- **Human accountability and escalation**: Provide users with the respective human contact points who are responsible for the agents, whom the users can alert if the agents malfunction or if they are dissatisfied with a decision.

2.4.3 Users who integrate agents into their work processes

Such users typically utilise agents as part of their internal workflows e.g. coding assistants, automation of enterprise processes. The agent acts for and on behalf of the user.

For these users, in addition to the information in the previous section, layer on education and training so that users can use the agents responsibly. Key aspects include education and training on:

- **Foundational knowledge on agents**
  - **Relevant use cases**, so that the users understand how to best integrate the agents into their day-to-day work, and the scenarios under which the use of agents should be restricted (e.g. do not use an agent for confidential data)
  - **Instructing the agents** e.g. general best practices in prompting, glossary of keywords to elicit specific responses
  - **Agents' range of actions**, so that the user is aware of their capabilities and potential impact

- **Effective oversight of agents**
  - **Common agent failure modes**, such as hallucinations, getting stuck in loops after errors, so that the user can identify and flag out issues.
  - **Ongoing support**, such as regular refreshers to update users on latest features and common user mistakes

- **Potential impact on tradecraft**

- As agents take over entry level tasks, which typically serve as the training ground for new staff, this could lead to loss of basic operational knowledge for the users.
- Organisations should identify core capabilities of each job and provide sufficient training and work exposure so that users retain foundational skills.

Annex A: Further resources

1. Introduction to Agentic AI

What is Agentic AI?	• AWS, Agentic AI Security Scoping Matrix: A framework for securing autonomous AI systems • WEF, AI Agents in Action: Foundations for Evaluation and Governance • Anthropic, Building effective agents • IBM, The 2026 Guide to AI Agents • McKinsey, What is an AI agent?
Risks of Agentic AI	• GovTech, Agentic Risk & Capability Framework • CSA, Draft Addendum on Securing Agentic AI • OWASP, Multi-Agentic System Threat Modelling Guide • IBM, AI agents: Opportunities, risks, and mitigations • Infosys, Agentic AI risks to the enterprise, and its mitigations

2. MGF for Agentic AI

Assess and bound the risks upfront	**Agentic governance in general** • EY, Building a risk framework for Agentic AI • McKinsey, Deploying agentic AI with safety and security: A playbook for technology leaders • Bain, Building the Foundation for Agentic AI • OWASP, State of Agentic AI Security and Governance 1.0 **Risk assessment and threat modelling** • OWASP, Agentic AI – Threats & Mitigations • OWASP, Multi-Agentic System Threat Modelling Guide • Cloud Security Alliance, Agentic AI: Understanding Its Evolution, Risks, and Security Challenges • EY, Building a risk framework for Agentic AI **Agent limits and agent identity** • Meta, Agents Rule of Two: A Practical Approach to AI Agent Security • OpenID, Identity Management for Agentic AI
Make humans meaningfully accountable	**Allocating responsibility within and outside an organisation** • Carnegie Mellon University, The 'Who', 'What', and 'How' of Responsible AI Governance • CSA and FAR.AI, Securing Agentic AI: A Discussion Paper • McKinsey, Accountability by design in the agentic organization **Designing for meaningful human oversight** • Partnership on AI, Prioritizing real-time failure detection in AI agents

	• Permit.IO, Human-in-the-Loop for AI Agents: Best Practices, Frameworks, Use Cases, and Demo
Implement technical controls and processes	**Technical controls** • GovTech, Agentic Risk & Capability Framework • CSA, Draft Addendum on Securing Agentic AI **Testing and evaluation** • Microsoft, Microsoft Agent Evaluators • AWS, AWS Agent Evaluation • Anthropic, Demystifying evals for AI agents • IBM, What is AI Agent Evaluation? **Monitoring and observability** • Microsoft, Top 5 agent observability best practices for reliable AI
Enabling end-user responsibility	• Zendesk, What is AI transparency? A comprehensive guide • HR Brew, Salesforce's head of talent growth and development shares how the tech giant is training its 72,000 employees on agentic AI • Harvard Business Review, The Perils of Using AI to Replace Entry-Level Jobs

Annex B: Call for feedback and case studies

Call for feedback: This is a living document, and we invite suggestions on how the framework can be updated or refined. The following questions can be used as a guide:

- **Introduction to Agentic AI**: Are the descriptions of agentic AI systems accurate and sufficiently comprehensive for readers to obtain a clear overview of the governance challenges of agentic AI? Are there other risks that should be included?
- **Proposed Model Governance Framework**: Are the four dimensions of the framework practical and applicable? Are there any other dimensions that should be included? For each dimension, are there specific governance and technical challenges and best practices that should be included?

Call for case studies: We also invite organisations to submit their own agentic governance experiences as case studies on how specific aspects of the framework can be implemented, to serve as practical examples of responsible deployment that other organisations can refer to. Case studies should ideally involve an organisation's deployment of an agentic use case that demonstrates one of the dimensions of the framework. While not exhaustive, we are specifically interested in case studies that demonstrate good practices in:

Dimension	Example case studies
Assess and bound the risks upfront	• Defining use cases to reduce risk but maximise benefits of agents • Defining limits on agent's autonomy through defined SOPs and workflows • Defining limits on agent's access to tools and systems • How identity is implemented for agents, and how it interacts with human identities in an organisation
Make humans meaningfully accountable	• Allocating responsibility across the organisation for agentic deployment • Assessing when human approvals are required in an agentic use case, and how requests for such approvals are implemented
Implement technical controls and processes	• Designing and implementing technical controls for agents • How agentic safety testing is carried out • How monitoring and observability mechanisms are set up, including defining alert thresholds and processing large volumes of agent-related data
Enable end-user responsibility	• Making information available to internal and external stakeholders who interact with and use agents • Training human overseers to exercise effective oversight

For an example of what a case study may look like, please refer to those in our previous Model Governance Framework for AI.

Please note that any feedback and case studies may be incorporated into an updated version of the framework, and contributors will be acknowledged accordingly. Please submit your feedback and case studies at this link: https://go.gov.sg/mgfagentic-feedback.