With Great Capabilities Come Great Responsibilities: Introducing the Agentic Risk & Capability Framework for Governing Agentic AI Systems

Shaun Khoo[1*], **Jessica Foo**[1*], **Roy Ka-Wei Lee**[2]
[1]GovTech Singapore
[2]Singapore University of Technology and Design
{shaun_khoo|jessica_foo}@tech.gov.sg

Abstract

Agentic AI systems present both significant opportunities and novel risks due to their capacity for autonomous action, encompassing tasks such as code execution, internet interaction, and file modification. This poses considerable challenges for effective organizational governance, particularly in comprehensively identifying, assessing, and mitigating diverse and evolving risks. To tackle this, we introduce the Agentic Risk & Capability (ARC) Framework, a technical governance framework designed to help organizations identify, assess, and mitigate risks arising from agentic AI systems. The framework's core contributions are: (1) it develops a novel capability-centric perspective to analyze a wide range of agentic AI systems; (2) it distills three primary sources of risk intrinsic to agentic AI systems - components, design, and capabilities; (3) it establishes a clear nexus between each risk source, specific materialized risks, and corresponding technical controls; and (4) it provides a structured and practical approach to help organizations implement the framework. This framework provides a robust and adaptable methodology for organizations to navigate the complexities of agentic AI, enabling rapid and effective innovation while ensuring the safe, secure, and responsible deployment of agentic AI systems.

Introduction

OpenAI dubbed 2025 the "year of the AI agent" (Hamilton 2025), a prediction that quickly proved prescient. Major AI companies launched increasingly powerful systems that allowed large language model ("LLM") agents to reason, plan, and autonomously execute tasks such as code development or web surfing. However, this surge in agent-driven AI innovation also brought renewed scrutiny to these systems' safety and security risks. Recent research (Chiang et al. 2025; Kumar et al. 2025; Yu and Papakyriakopoulos 2025) demonstrated that LLM agents are more prone to unsafe behaviors than their base models. Moreover, governing agentic systems presents unique challenges compared to traditional LLM systems - they have the autonomy to execute a wide variety of actions, thereby introducing a significantly broader range of risks. This makes comprehensive identification, assessment, and mitigation more challenging, thus hindering effective organizational governance. While conducting customized risk assessments for each agentic system is possible as an interim measure, it is unsustainable in the long run.

The Agentic Risk & Capability ("ARC") framework aims to tackle this problem as **a technical governance framework for identifying, assessing, and mitigating the safety and security risks of agentic systems**.[1] It examines where and how risks may emerge, contextualizes the agentic system's risks given its domain, use case, and organizational context, and recommends technical controls for mitigating these risks. While the ARC framework is not a panacea to the complex challenges of governing agentic systems, it offers a strong foundation upon which organizations can manage risks in a systematic, scalable, and adaptable manner.

Existing Literature on Agentic AI Governance

Although regulatory frameworks such as the EU AI Act (European Parliament and Council of the European Union 2024) and the NIST Risk Management Framework (National Institute of Standards and Technology 2023) articulate clear overarching principles and guidelines for managing AI risks, they do not examine specific technical measures for identifying, assessing, and managing risks. Our paper aims to contribute to the **technical AI governance** field by developing "technical analysis and tools for supporting the effective governance of AI" (Reuel et al. 2025). For agentic AI, Raza et al. (2025) adapted the AI Trust, Risk, and Security Management (TRiSM) framework to LLM-based multi-agent systems. It provides generalized metrics and controls across a spectrum of risks, but does not tackle the practical problems of contextualizing risks for a given agentic system to be deployed. Another approach, proposed by Engin and Hand (2025), is dimensional governance through tracking AI systems along three dynamic axes (decision authority, process autonomy, and accountability), introducing controls when systems shift across critical thresholds. While conceptually appealing, its effectiveness relies on accurately quantifying the dimensions and calibrating the thresholds, both of which are hard to operationalize. More

[*]These authors contributed equally.

[1]For the full version of the ARC Framework, please visit our website at https://govtech-responsibleai.github.io/agentic-risk-capability-framework/

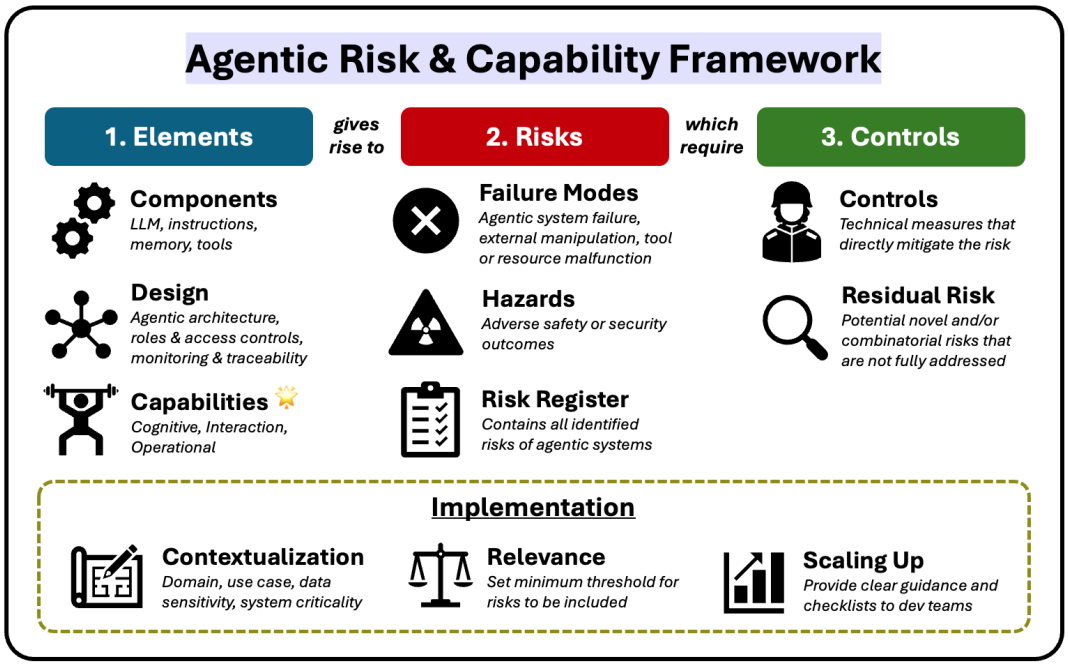

Figure 1: Overview of the ARC Framework

cybersecurity-oriented frameworks include the MAESTRO framework (Huang et al. 2025), OWASP's white paper on agentic AI risks (OWASP 2025), and NVIDIA's taint tracing approach (Harang et al. 2025) which utilize threat modelling to uncover security threats (e.g. data poisoning, agent impersonation). However, this is highly complex, especially for developers untrained in cybersecurity, and the controls rely heavily on human oversight.

Capabilities of an Agentic System

Effective governance requires distinguishing between safer and riskier systems and implementing a differentiated approach to manage them. For agentic AI governance, beyond analyzing the components of an agent (i.e. the LLM, instructions, tools, and memory) and the design of the agentic system (i.e. agentic architecture, access controls, and monitoring), **the ARC framework adopts the novel approach of also analyzing agentic AI systems by their capabilities.**

By capabilities, we refer to the actions that the agentic system can autonomously execute over the tools and resources it has access to, whether it be running code, searching the internet, or modifying documents. This is the complement of affordances (as defined by Gaver (1991)), which are properties of the external environment that enable actions. In our view, the components and design of agentic systems are *affordances*, while executing code or altering agent permissions are examples of *capabilities*, which we cover in the next section. Addressing both aspects is essential for the effective governance of agentic systems.

There are three key advantages of adopting a capability lens in agentic AI governance.

1. **Capabilities offer a more holistic unit of analysis than analyzing specific tools**. There are numerous tools that facilitate similar actions (e.g. Google SERP, Serper, SerpAPI, Perplexity Search API), and conversely, a single tool can enable a wide array of actions (e.g. GitHub's Model Context Protocol ("MCP") server enabling code commits, reading of pull requests etc.) - a point also made by Gaver (1991) on affordances. Given the sheer diversity and rapid development of MCPs, prescribing specific controls for each tool used would be too granular, and lead to obsolete and inconsistent controls.
2. **Adopting a capability lens allows for differentiated treatment in a scalable manner**. Systems with more capabilities are inherently riskier and necessitate more stringent controls, particularly when these capabilities have a significant impact on the system. By deconstructing a system into its constituent capabilities, riskier systems can be subject to greater scrutiny while low-risk systems proceed with a lighter touch.
3. **Risks arising from actions is intuitive to laypersons, which is vital for effective contexualization**. Technical approaches often run the risk of being esoteric, which hampers adoption and limits flexibility. By being more accessible to the average person, the capability lens enables organizations to be more flexible in adapting to new developments and risks.

Agentic Risk & Capability Framework

In this section, we explain each part of the ARC framework - the elements, risks, and controls - in detail. We also provide a visual summary of the entire framework in Figure 1.

Part 1: Elements of Agentic Systems

Across all agentic systems, there are three indispensable elements to examine: components of an agent, design of the agentic system, and the capabilities of the agentic system.

Components: Components are essential parts of a single, standalone agent. Here, we synthesize prevailing agreement on the key components of an agent from various sources, such as OpenAI (OpenAI 2025).

- **LLM**: The LLM is the central reasoning engine that processes instructions, interprets user inputs, and generates contextually appropriate responses by leveraging its language understanding and generation capabilities.
- **Tools**: Tools enable LLMs to interact with the external environment, be it editing files, querying databases, controlling devices, or accessing APIs.
- **Instructions**: Instructions are the blueprint which defines an agent's role, capabilities, and behavioral constraints, ensuring it operates within intended parameters.
- **Memory**: The memory or knowledge base component provides the agent with contextual awareness and information persistence, enabling it to learn from past interactions without requiring constant re-instruction.

Design: We now broaden our perspective to examine how agentic AI systems are assembled from individual agents from a system design perspective.

- **Agentic Architecture**: The agentic architecture defines how multiple agents are interconnected and orchestrated to collectively solve complex tasks.
- **Roles and Access Controls**: Roles and access controls establish differentiated responsibilities and permissions across agents within the system.
- **Monitoring and Traceability**: Monitoring and traceability provide visibility into agentic system behavior, interactions, and decision-making.

Capabilities: We see three broad categories of capabilities - cognitive, interaction, and operational. A full description of each capability can be found in the appendix.

Cognitive capabilities encompass the agentic system's internal "thinking" skills – how it analyzes information, forms plans, and learns from experience.

- **Planning & Goal Management**: Develop detailed, step-by-step, and executable plans.
- **Agent Delegation**: Assign subtasks to other agents and coordinate their activities.
- **Tool Use**: Evaluate options and choose the best tool.

Interaction capabilities describe how the agentic AI system exchanges information with users, other agents, and external systems. These capabilities below are broadly differentiated based on how and what they interact with.

- **Natural Language Communication**: Fluently and meaningfully converse with human users or generate text to meet their instructions.
- **Multimodal Understanding & Generation**: Take in or generate image, audio, or video as inputs or outputs.
- **Official Communication**: Directly compose and publish communications that represents an organization to external parties (e.g. customers) without human oversight.
- **Business Transactions**: Execute transactions that involve exchanging money, services, or commitments with external parties.
- **Internet & Search Access**: Access and search the Internet for knowledge resources.
- **Computer Use**: Control a computer interface by moving the mouse or clicking buttons on behalf of the user.
- **Other Programmatic Interfaces**: Interact with external systems through APIs, SDKs, or backend service.

Operational capabilities focus on the agentic AI system's ability to execute actions safely and efficiently within its operating environment.

- **Code Execution**: Write, execute, and debug code in various programming languages.
- **File & Data Management**: Create, read, update, or delete information for unstructured files (e.g. PDFs) and structured data stores (e.g. SQL/NoSQL databases).
- **System Management**: Adjust system configurations, manage computing resources, and handle technical infrastructure tasks.

Part 2: Risks of Agentic Systems

Next, we examine how risks materialize from the elements of an agentic system as described above. This comprises two key aspects: the failure mode, which outlines how the system fails, and the hazard, which describes the resulting impact.

Failure Modes: First, we specify three general modalities in which agentic systems may fail:

- **Agent Failure**: The agent fails to operate as intended due to poor performance, misalignment, or unreliability.
- **External Manipulation**: Malicious actors cause or trick the agent to deviate from its intended behavior.
- **Tool or Resource Malfunction**: The tools or resources used by the agent fail or are compromised.

Hazards: Second, we list a range of safety and security hazards which may result from these failures. Note that this serves solely as a heuristic for risk identification and should not be interpreted as a rigid and comprehensive taxonomy.

Table 1: Security and Safety Hazards

Security	Safety
• Leaking sensitive or confidential data • Application system failures • Network infiltration and disruption • Role impersonation or privilege escalation	• Illegal and CBRNE activities • Discriminatory or hateful content • Undesirable content (e.g. sexual, violence) • Affect user safety • Misinformation

The Risk Register: The Risk Register consolidates all the risks identified through the ARC framework, and **serves as the organization's reference list of safety and security risks of agentic systems**. By design, each risk in the Risk Register should (1) originate from an element (components, design, or capabilities), (2) satisfy a failure mode (agent failure, external manipulation, tool or resource malfunction), and (3) result in at least one of the safety or security hazards listed in the table above. We recommend phrasing risks in a consistent manner to aid validation and understanding.

To demonstrate how this works in practice, we provide three examples below:

RISK REGISTER
...
[RISK-007]: "Poorly implemented tools may not correctly verify user identity or permissions when executing privileged actions" is a security risk (identity & access management) caused by tool or resource malfunction of the tools component in an agent.
...
[RISK-053]: "Opening vulnerabilities to prompt injection attacks via malicious websites" is a security and safety risk (all) caused by external manipulation of the Internet & Search Access capability.
...
[RISK-062]: "Overwhelming the database with poor, inefficient, or repeated queries" is a security risk (application, infrastructure) caused by agent failure of the File & Data Management capability.
...

Although combining the element, failure mode, and hazard can help in brainstorming potential risks to agentic systems, not all of them will be correct. For instance, tool or resource malfunction for the instructions component is not really a sensible risk. As such, organizations should exercise discretion in deciding what risks to be included in the Risk Register - one helpful criteria is to keep only risks which are supported by academic research or industry case studies. We provide a short sample Risk Register in the appendix, but we are unable to provide a full sample due to space limitations.

Part 3: Controls for Agentic Systems

The last part provides guidance on how these risks can be mitigated through technical controls. However, given the rapidly evolving field of agentic AI, there is likely to be significant residual risk even after several controls have been implemented. We discuss both below.

Technical controls: Within the Risk Repository, **each risk comes with a set of recommended technical controls** which aim to either (i) reduce the potential impact by limiting the scope or severity of a failure, or (ii) decrease the likelihood of the failure mode occurring. This makes the logical connection between risks and controls clear and intuitive.

We provide an example for a specific risk below:

[RISK-053]: "Opening vulnerabilities to prompt injection attacks via malicious websites" is a security and safety risk (all) caused by external manipulation of the Internet & Search Access capability.

[CTRL-078]: Implement input guardrails to detect prompt injection or adversarial attacks
[CTRL-079]: Implement escape filtering before including web content into prompts
[CTRL-080]: Use structured retrieval APIs for searching the web rather than through web scraping

It is important to note that not all controls are unique; some may overlap due to targeting similar failure modes or aiming to limit the "blast radius" of a particular security or safety outcome. This is especially true of capabilities which create new vectors for prompt injection attacks.

Residual risks Agentic AI and LLMs is a rapidly developing space, and it is unlikely that any list of technical controls can credibly claim to entirely neutralize all potential threats. This makes it crucial to evaluate the residual risk - the remaining risk after controls have been applied - to uncover gaps and to assess the overall level of risk in the agentic system. If the residual risk is deemed unacceptable, further measures, both technical and otherwise, must be implemented to reduce it to an acceptable level.

Identifying residual risks is intrinsically difficult as it is very dependent on the specifics of the agentic system, but common ones include inherent weaknesses of the technical controls (for example, prompt injection guardrails that are trained on past jailbreaks may not generalize well to detect novel attacks) or combinatorial risks which arise from the interaction of two or more capabilities.

Part 4: Implementation of ARC Framework

A well-known adage is "Policy is implementation and implementation is policy" (Ho 2010), and this is resoundingly true for AI governance. The ARC framework is designed to be easily implementable by centralized governance teams, and this subsection highlights three steps for how to do so.

Contextualizing Risks: Although we have identified general security and safety hazards, these need to be contextualized to the organization. This involves determining the degree of impact and the degree of likelihood of a risk, with a five-point scale for both. Some criteria to consider for contextualizing the impact include the domain (e.g. medical, education), use case, data sensitivity, and system criticality, and for likelihood, some factors include the ease of replication or the level of access required for a successful attack. For instance, infrequent hallucinations in marketing copy might be tolerable, but in a legal context where accuracy is paramount, it would be entirely unacceptable.

Establish Relevance Threshold: Organizations must establish a minimum threshold for both impact and likelihood to determine which risks are relevant to the specific agentic system. Any risks that remain above this relevance threshold

will then require mitigation through the controls described in Part 3. Some enterprises may set a higher threshold to keep the number of relevant risks small, while others might be more conservative and choose a lower threshold.

Scaling Up: To streamline implementation, organizations can provide simple forms or checklists for developers to declare system capabilities, relevant risks, and technical controls, which can then be validated by a central governance team. This standardization also helps in providing an organization-wide view of risk exposures and control adoption. Another critical aspect is continual updating of the Risk Register, especially as new threats or regulatory changes emerge. Organizations need to define a regular cadence for updating the risks and controls in the Risk Register to keep up with the latest developments.

Worked Example: Researcher

To demonstrate how the framework would work in practice, we apply the ARC framework to `Researcher`, a hypothetical agentic AI system which compiles research on a specific topic, similar to OpenAI's or Perplexity's Deep Research. The user provides the research question, then the `Researcher` clarifies the scope, devises a research plan, searches the web, and compiles the information into a structured report to address the user's question.

We can identify the `Researcher`'s capabilities as Planning & Goal Management, Natural Language Communication, and Internet & Search Access. Together with the components and design elements and referring to the organization's internal Risk Register, there are 38 applicable risks to be assessed. To demonstrate how the contextualized assessment works, we provide two examples below, one assessed to be relevant and another to be irrelevant:

[RISK-007]: "Poorly implemented tools may not correctly verify user identity or permissions when executing privileged actions" is a security risk (identity & access management) caused by tool or resource malfunction of the tools component in an agent.

Impact: **1/5** - Search tool does not have any privileged actions since it only searches public websites.
Likelihood: **1/5** - Current implementation relies on trustworthy Internet search tools like DuckDuckGo, with necessary security protocols in place.

Relevance: **Not relevant** as company's relevance threshold is 3 for impact and 4 for likelihood.

[RISK-053]: "Opening vulnerabilities to prompt injection attacks via malicious websites" is a security and safety risk (all) caused by external manipulation of the Internet & Search Access capability.

Impact: **4/5** - Manipulation of the agent can result in a range of safety and security risks that may compromise other sensitive systems or result in reputational loss for the company which depends on the success of this product.
Likelihood: **5/5** - Attack has been demonstrated in several real-world case studies, access to the system is not required to execute attack.

Relevance: **Relevant** as company's relevance threshold is 3 for impact and 4 for likelihood.

This process is repeated for all 38 applicable risks, with only 10 risks eventually assessed as relevant, which then results in 17 controls which the team now needs to adopt or adapt to safeguard the agentic system. This step-by-step approach is not only straightforward for developers, but ensures comprehensive understanding of the system's risks.

Benefits of the ARC framework

First, the ARC framework enables meaningfully differentiated risk management for different types of agentic systems while still ensuring some level of consistency across all systems. The component and design elements establish a foundational set of minimum hygiene standards that apply across all agentic systems, guaranteeing a baseline level of safety and security regardless of their specific function or risk profile. Layering on top of that is the capability element, which can vary on the use case and what tools the agent has. This enables a nuanced approach to risk management for agentic systems, as lower-risk systems are not unduly burdened with excessive compliance.

Second, the ARC framework provides forward guidance for developers to build with safety and security considerations upfront, thus avoiding abortive work and encouraging proactivity. Developers know upfront the risks and controls for each capability, encouraging them to incorporate safety and security considerations into the initial stages of the development lifecycle. By providing clear, actionable guidance upfront, developers can design agentic systems with these safeguards built-in, mitigating risks and reducing developer toil. This also makes the ARC framework more scalable as organizations ramp up adoption of agentic systems across business units and use cases.

Third, the ARC framework has the flexibility to update risks and controls as agentic systems develop and evolve. The field of agentic AI is characterized by rapid technological advancement and emergent capabilities, leading to an evolving risk landscape. The ARC framework's systematic risk identification approach helps governance teams make sense of the latest research and real-world incidents and provides a structured way to incorporate the latest risks. The accompanying technical controls can be refreshed with industry best practices as they are launched.

Conclusion

As agentic systems become increasingly prevalent, frameworks become essential for safe, ethical, and responsible AI deployment. The ARC framework not only helps organizations manage current risks but also provides a foundation for adapting to future developments in agentic AI capabilities and emerging threat landscapes. With this framework established, future work can focus on developing empirical approaches to validate the risks and controls in the Risk Register and on building automated tools to support the implementation and regular updating of the framework.

References

Chiang, C.-H.; et al. 2025. Harmful helper: Perform malicious tasks? web AI agents might help. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Engin, Z.; and Hand, D. 2025. Toward Adaptive Categories: Dimensional Governance for Agentic AI. arXiv:2505.11579.

European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng. Accessed: 2025-05-11.

Gaver, W. 1991. Technology affordances. In *Conference on Human Factors in Computing Systems - Proceedings*, 79–84.

Google. 2025. Agent2Agent (A2A) Protocol – Latest. https://a2a-protocol.org/latest/. Accessed: 2025-10-11.

Hamilton, E. 2025. 2025 is the year of ai agents, OpenAI CPO says. *Axios*.

Harang, R.; et al. 2025. Agentic Autonomy Levels and Security.

Ho, P. 2010. Opening Address at 2010 Administrative Service Dinner and Promotion Ceremony. Public Service Division.

Huang, Y.; et al. 2025. On the Resilience of LLM-Based Multi-Agent Collaboration with Faulty Agents. *arXiv preprint arXiv:2408.00989v3*.

Kumar, A.; et al. 2025. Aligned LLMs are not aligned browser agents. In *The Thirteenth International Conference on Learning Representations*.

National Institute of Standards and Technology. 2023. NIST AI Risk Management Framework Playbook. https://www.nist.gov/itl/ai-risk-management-framework/nist-ai-rmf-playbook. Accessed: 2025-05-11.

OpenAI. 2025. A practical guide to building agents.

OWASP. 2025. Agentic AI – Threats and Mitigations.

Raza, S.; Sapkota, R.; Karkee, M.; and Emmanouilidis, C. 2025. TRiSM for Agentic AI: A Review of Trust, Risk, and Security Management in LLM-based Agentic Multi-Agent Systems. arXiv:2506.04133.

Reuel, A.; Bucknall, B.; Casper, S.; Fist, T.; Soder, L.; Aarne, O.; Hammond, L.; Ibrahim, L.; Chan, A.; Wills, P.; Anderljung, M.; Garfinkel, B.; Heim, L.; Trask, A.; Mukobi, G.; Schaeffer, R.; Baker, M.; Hooker, S.; Solaiman, I.; Luccioni, A. S.; Rajkumar, N.; Moës, N.; Ladish, J.; Bau, D.; Bricman, P.; Guha, N.; Newman, J.; Bengio, Y.; South, T.; Pentland, A.; Koyejo, S.; Kochenderfer, M. J.; and Trager, R. 2025. Open Problems in Technical AI Governance. arXiv:2407.14981.

Yu, C.; and Papakyriakopoulos, O. 2025. Safety devolution in AI agents. In *ICLR 2025 Workshop on Human-AI Coevolution*.

Elements of agentic systems

In this appendix section, we provide detailed descriptions for each of the elements of agentic systems for reference.

Components

- **LLM**: The LLM is the reasoning engine that processes instructions, interprets user inputs, and generates contextually appropriate responses by leveraging its trained language understanding and generation capabilities.
- **Tools**: Tools enable LLMs to interact with the external environment, be it editing files, querying databases, controlling devices, or accessing APIs. This is facilitated by MCP servers, which provide LLMs a consistent interface to discover and utilize a variety of tools.
- **Instructions**: Instructions are the blueprint which defines an agent's role, capabilities, and behavioral constraints, ensuring it operates within intended parameters and maintains its performance across different scenarios.
- **Memory**: The memory or knowledge base component provides the agent with contextual awareness and information persistence, enabling it to maintain coherent conversations, learn from past interactions, and access relevant facts without requiring constant re-instruction.

Design

- **Agentic Architecture**: The agentic architecture defines how multiple agents are interconnected, coordinated, and orchestrated to collectively solve complex tasks that exceed individual agent capabilities, including patterns like hierarchical delegation, parallel processing, or sequential handoffs between specialized agents. Different architectures result in varying levels of system-wide risk, and these need to be considered carefully. Similarly, the protocols (Google 2025) by which agents communicate may also give rise to security risks.
- **Roles and Access Controls**: Roles and access controls establish differentiated responsibilities and permissions across agents within the system, ensuring that each agent operates within appropriate boundaries while being able to fulfill its designated function. This is critical because it limits unauthorized actions, contains the blast radius of potential failures or security breaches, and enables the system to maintain reliability even when individual agents may be compromised or behave unexpectedly.

- **Monitoring and Traceability**: Monitoring and traceability enable visibility into agentic system behavior, interactions, and decision-making pathways, allowing developers and operators to understand what agents are doing, why they made particular choices, and how outcomes were produced. This is essential for post-hoc debugging, real-time anomaly detection, and establishing accountability particularly when agents operate with a degree of autonomy or interact with sensitive systems and data.

Capabilities

Cognitive capabilities encompass the agentic AI system's internal "thinking" skills – how it analyses information, forms plans, and monitors its own performance.

- **Planning & Goal Management**: The capability to develop detailed, step-by-step, and executable plans with specific tasks in response to broad instructions. This includes prioritizing activities based on importance and dependencies between tasks, monitoring how well its plan is working, and adjusting when circumstances change or obstacles arise.
- **Agent Delegation**: The capability to assign subtasks to other agents and coordinate their activities to achieve broader goals. This includes identifying which components are best suited for specific tasks, issuing clear instructions, managing inter-agent dependencies, and monitoring performance or failures.
- **Tool Use**: The capability to evaluate available options and choose the best tool for specific subtasks, based on the capabilities and limitations of different tools and matching them appropriately to the tasks.

Interaction capabilities describe how the agentic AI system exchanges information with users, other agents, and external systems. These capabilities below are broadly differentiated based on how and what they interact with.

- **Natural Language Communication**: The capability to fluently and meaningfully converse with human users, handling a wide range of situations such as explaining complex topics, generating documents or prose, or discussing issues with human users.
- **Multimodal Understanding & Generation**: The capability to take in image, audio, or video inputs and / or generate image, audio, or video outputs. This includes analyzing visual information, transcribing speech, or creating multimedia content as needed.
- **Official Communication**: The capability to compose and directly publish communications that formally represent an organization to external parties (e.g. customers, partners, regulators, courts, media) via approved channels and formats without human oversight.
- **Business Transactions**: The capability to execute transactions that involve exchanging money, services, or commitments with external parties. It can process payments, make reservations, and handle other business transactions within authorized limits.
- **Internet & Search Access**: The capability to access and search the Internet for knowledge resources, especially for up-to-date information to provide accurate answers.
- **Computer Use**: The capability to directly control a computer interface by moving the mouse, clicking buttons, and typing on behalf of the user. It can navigate applications and perform tasks that require interacting with graphical user interfaces.
- **Other Programmatic Interfaces**: The capability to interact with external systems through APIs, SDKs, or backend services. This includes sending and receiving data via RESTful APIs, pushing code to a remote repository, or invoking cloud services to retrieve or manipulate information from other systems.

Operational capabilities focus on the agentic AI system's ability to execute actions safely and efficiently within its operating environment.

- **Code Execution**: The capability to write, execute, and debug code in various programming languages to automate tasks or solve computational problems.
- **File & Data Management**: The capability to create, read, modify, organize, convert, query, and update information across both unstructured files (e.g. PDFs, Word docs, spreadsheets) and structured data stores (e.g. SQL/NoSQL databases, data warehouses, vector stores).
- **System Management**: The capability to adjust system configurations, manage computing resources, and handle technical infrastructure tasks. This includes monitoring system performance, securely handle authentication information and access controls, and making optimizations as needed while maintaining security best practices.

Sample Risk Register

This is a sample risk register for the risks from two capabilities - [Interaction] Other Programmatic Interfaces and [Operational] Code Execution, as well as their corresponding controls. We are unable to provide the full list here due to space constraints.

Risk ID	Risk Description	Control ID	Control Description
RISK-057	Leaking personally identifiable or sensitive data	CTRL-041	Use time-bound or one-time-use credentials where possible.
		CTRL-086	Specify a whitelist of interfaces that agents are allowed to use
RISK-058	Increasing the system's vulnerability to supply chain attacks	CTRL-087	Enforce zero-trust input handling and validate all data flows
RISK-059	Executing poor code	CTRL-088	Use code linters to screen for bad practices, anti-patterns, unused variables, or poor syntax
		CTRL-089	Use static code analyzers to detect problems with the code
		CTRL-090	Run code only in virtually isolated compute environments (e.g. Docker containers)
		CTRL-091	Ensure monitoring of code runtime and memory consumption
RISK-060	Executing vulnerable or malicious code	CTRL-036	Apply Principle of Least Privilege (PoLP) when configuring all agent and delegation roles.
		CTRL-092	Use static code analyzers to identify dangerous patterns in the code before execution
		CTRL-093	Conduct CVE scanning and block execution if any High or Critical CVEs are detected
		CTRL-094	Block all inward and outward network access by default
		CTRL-037	Do not grant admin privileges to agents
		CTRL-005	Implement input sanitization measures or limit inputs to conventional ASCII characters only.
		CTRL-095	Implement a whitelist approach for inward network access
		CTRL-096	Review all code generated by agents, including shell scripts, before execution
		CTRL-097	Create a Deny list of commands that agents are not allowed to run autonomously
RISK-061	Overwriting or deleting database tables or files	CTRL-098	No write access to tables in the database unless strictly required
		CTRL-099	Require human approval for any changes to the database, table, or file
		CTRL-100	Avoid mounting broad or persistent paths

Figure 2: Sample risk register with associated controls