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

Shaun Khoo, Jessica Foo*
GovTech Singapore

Roy Ka-Wei Lee
Singapore University of Technology and Design

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. Our framework is open-sourced here.

1 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. Although conducting in-depth and 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**. It examines where and how risks may emerge, contextualizes the agentic system's risks given its domain, use case, and organizational context, and recommends practical and technical controls for mitigating these risks. While the ARC framework is not a panacea to the complex

*Equal contribution

Second Conference of the International Association for Safe and Ethical Artificial Intelligence (IASEAI'26).

challenges of governing agentic systems, it offers a strong foundation upon which organizations can manage the plethora of risks in a systematic, scalable, and adaptable manner.

2 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 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.

Benchmarks help to assess how risky agentic systems are, and there are several safety and security benchmarks which outline test scenarios or tasks that reveal specific risk behaviors of the agentic system. For example, Agent Security Bench [Zhang et al., 2025], CVEBench [Zhu et al., 2025], RedCode [Guo et al., 2024], AgentHarm [Andriushchenko et al., 2025], AgentDojo [Debenedetti et al., 2024] assess whether LLMs can complete multi-step cybersecurity attacks or harmful tasks like fraud, but they do not help developers identify the full range of risks and attack scenarios of their specific applications. Tool-based benchmarks, such as APIBench [Patil et al., 2023], ToolSword [Ye et al., 2024], and ToolEmu [Ruan et al., 2024] measure the performance and safety of LLMs in utilizing tools like `bash`, but omit risks unrelated to tool use (e.g. misaligned LLMs).

For mitigating risks, AI control has emerged as a paradigm in preventing misaligned AI systems from causing harm [Greenblatt et al., 2024]. Rather than relying solely on training techniques to shape model behavior, AI control focuses on designing mechanisms like monitoring and human oversight to constrain AI systems. For instance, Progent [Shi et al., 2025] and AgentSpec [Wang et al., 2025] introduce a language for flexibly expressing privilege control policies that are applied at runtime. The UK AI Security Institute advocates for AI control levels, derived from evaluating frontier LLMs' threat model-specific capabilities [Korbak et al., 2025]. OpenAI shared best practices like constraining the agent's action spaces and ensuring attributability [Shavit et al., 2025], while Google emphasized a hybrid defense-in-depth strategy that combines deterministic security measures with dynamic, reasoning-based defenses [Diaz et al., 2025]. Similarly, Beurer-Kellner et al. [2025] propose six design patterns for building AI agents with provable resistance to prompt injections. However, these works are either too narrow (i.e., specific to application) or too broad (i.e., high-level, conceptual) to effectively operationalize within organizations.

3 Capabilities of an Agentic System

Effective governance requires distinguishing between safer and riskier systems and implementing a differentiated approach to manage them. Applying this to 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 (see Sections 4.1.1 and 4.1.2) are *affordances*, while executing code or altering agent permissions are examples of *capabilities*, which we cover in Section 4.1.3. 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 and every tool used would be too granular, and lead to obsolete, inconsistent, and overly restrictive 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, we can ensure that riskier systems receive greater scrutiny while enabling low-risk systems to 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.

4 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.

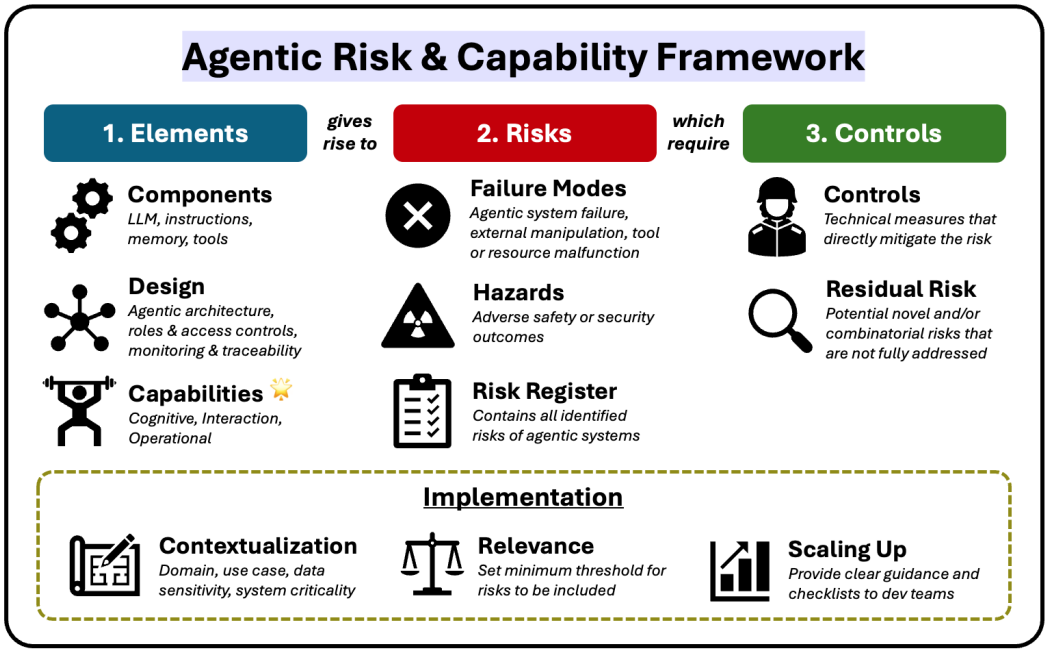

Figure 1: Overview of the ARC Framework

4.1 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.

4.1.1 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 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.

4.1.2 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, 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.

4.1.3 Capabilities

We see three broad categories of capabilities - cognitive, interaction, and operational - and break it down into more granular capabilities.

Cognitive capabilities encompass the agentic AI system's internal "thinking" skills – how it analyses information, forms plans, learns from experience, 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. This requires agents to understand the capabilities and limitations of different tools and match 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 more 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.

4.2 Part 2: Risks of Agentic Systems

The next part involves detailing how the risks materialize from the elements of an agentic system as described in 4.1. This comprises two key aspects: the failure mode, which outlines how the system fails, and the hazard, which describes the resulting impact.

4.2.1 Failure Modes

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

- **Agent Failure**: The agent itself 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 utilized by the agentic system fail, are compromised, or are inadequate.

4.2.2 Hazards

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

Table 1: Hazard Categories by Type

Type	Hazard Category	Description
Security	Data (files, databases)	Failures can lead to data breaches, integrity attacks, PII exposure, or ransomware, where sensitive information is exfiltrated, corrupted, or held hostage.
	Application	This category includes system failures, service disruptions, unintended use of applications, backdoor access, or resource exploitation, compromising the functionality and security of the software.
	Infrastructure & network	Denial of service (DoS/DDoS) attacks, man-in-the-middle (MitM) attacks, network eavesdropping, or lateral access, all of which can disrupt or compromise the underlying network and infrastructure.
	Identity & access management	Unauthorized control, impersonation of credible roles, or privilege escalation, allowing attackers to gain elevated access or control over systems.
Safety	Illegal and CBRNE activities	This includes agents facilitating or engaging in CBRNE-related activities or other types of criminal offenses, such as fraud, scams, or smuggling.
	Discriminatory or hateful content	This category is aimed at unsafe and discriminatory content, especially incendiary hate speech and slurs, as well as biased decisions.
	Inappropriate content	This refers to the generation of content that is vulgar, violent, sexual, promotes self-harm, or encourages illegal activities, leading to reputational harm and erosion of trust in the system.
	Compromise user safety	Failures can directly endanger users, for example, through the propagation of inaccurate information or the execution of actions that lead to physical or psychological harm.
	Misrepresentation	This outcome involves the propagation and dissemination of wrong and inaccurate information, or cascading failures where inaccurate information is not corrected, leading to further errors and a loss of trust.

4.2.3 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 generally recommend phrasing risks in a consistent manner to aid validation and understanding.

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

Example 1: "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.

Example 2: "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.

Example 3: "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.

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.

For illustrative purposes, we provide a draft Risk Register in Appendix B, covering most of the major risks associated with agentic systems, and with each risk backed by a real example or academic study. This can serve as a useful starting point for organizations, though it will need continual updating as the space of agentic AI develops and matures.

4.3 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.

4.3.1 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 a specific failure mode occurring. This makes the logical connection between risks and controls clear and intuitive. Controls are categorised into three levels based on criticality: Cardinal controls (Level 0) are fundamental requirements that must be adopted as is; Standard controls (Level 1) should be adopted or adapted meaningfully; and Best Practice controls (Level 2) are recommended for high-risk systems. This tiered approach enables organisations to prioritise control implementation based on their risk tolerance and resource constraints.

We provide an example of the technical controls for a specific risk below:

Risk: "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.

Control 1: Implement input guardrails to detect prompt injection or adversarial attacks
Control 2: Implement escape filtering before including web content into prompts
Control 3: 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.

In our draft Risk Register in the appendix, we also provide a tentative list of recommended controls for each risk to help organizations get started.

4.3.2 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.

4.4 Implementation

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 implemented by organizations, and specifically by centralized governance teams that are responsible for managing the risks of AI and agentic systems. This subsection highlights three key steps that governance teams need to take when implementing the ARC framework in their organization.

4.4.1 Contextualizing Risks

Contextualizing risk involves two primary dimensions: determining the degree of impact and the degree of likelihood. We recommend a five-point scale for both, with impact ranging from minimal to catastrophic and likelihood ranging from very likely to very rare. This assessment must be carefully contextualized as the implications for a small enterprise in the manufacturing sector differ greatly from those of a multinational corporation in the finance industry or even a governmental entity.

The degree of impact will vary significantly depending on how and where the agentic system is used. For instance, a hallucination in marketing copy might be tolerable, but in a legal context, it would be entirely unacceptable. Some criteria to consider when estimating the impact include:

- **Domain**: The sensitivity and criticality of the domain are paramount. For instance, risks in medical or educational domains are more critical than those in less sensitive areas.
- **Use Case**: What the agentic system is used for. While office productivity tools might present straightforward risks, systems involved in hiring or performance assessments carry more sensitive and potentially impactful consequences.
- **Data Sensitivity**: The level of sensitivity or confidentiality of the data being processed. Systems handling highly sensitive data naturally pose greater risks if compromised.
- **System Criticality**: For governmental or critical infrastructure systems, the impact of a system failure can be severe and widespread, necessitating a higher level of scrutiny.

Assessing the degree of likelihood will depend a lot on the identified failure mode and the probability of that failure mode occurring, considering factors like the ease of replication or the level of access required for a successful attack. Although this is slightly less context-dependent, there are some factors like the organization's general security (physical and cyber) measures that may limit how exploitable an agentic system can be.

4.4.2 Establish Relevance Threshold

Organizations must then 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 explicit mitigation through the controls described in Part 3 of the framework. This threshold is contingent upon the organization's overall risk appetite - some enterprises may set a higher threshold to keep the number of relevant risks small, while more conservative organizations might choose a lower threshold to require more risks to be directly managed.

4.4.3 Scaling Up

To streamline implementation, organizations should provide simple forms or checklists for developers to declare system capabilities, relevant risks, and technical controls, which can then be validated and audited 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 reviewing the risks and controls in the Risk Register, and updating them to keep up with the latest developments.

5 Worked Examples

In this section, we apply the ARC framework to two stylized agentic systems to demonstrate how the framework would help in practice to identify, assess, and mitigate safety and security risks.

5.1 Example 1: Researcher

`Researcher` is 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.

Referencing the capabilities in Section 4.1.3, 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 our draft Risk Register in Appendix B, there are 38 applicable risks to be assessed. We provide an example below:

Risk: "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 extend beyond the system's boundaries and result in reputational loss for the company.
Likelihood: **5/5** - Attack has been demonstrated in several real-world case studies, no access to the system required to execute attack.

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

After contextualizing the risks and assessing relevance, only 10 risks remain (RISK-003, RISK-009, RISK-017, RISK-023, RISK-034, RISK-035, RISK-036, RISK-038, RISK-053, and RISK-054) and require technical controls to be implemented for. We provide full explanations in Appendix D, rationalizing the impact and likelihood of each risk, and highlighting the relevant risks. Now referring to Appendix C, there are 17 controls associated with these 10 risks 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.

5.2 Example 2: Vibe Coder

`Vibe Coder` is a hypothetical agentic system which allows non-technical users to develop and deploy simple web apps through natural language prompts, similar to Vercel or Replit. The user specifies the app's key features and design, `Vibe Coder` proceeds to generate the code and text for the web app, run and create the required front-end and back-end systems locally, and render the website for the user to preview. If the user is satisfied, `Vibe Coder` will then automatically deploy the web app into a staging environment where it is then ready for user acceptance testing.

Referencing the capabilities in 4.1.3, we can identify quite a few capabilities: Planning & Goal Management, Tool Use[2], Natural Language Communication, Internet & Search Access, Code Execution, File & Data Management, and System Management.

Now examining our draft Risk Register in Appendix B, there are a total of 48 applicable risks - unsurprisingly, this is double the number of capability risks of the `Researcher`, since there are more capabilities and some of them are also intrinsically riskier. We analyze one risk below:

[2]Tool use appears only for the `Vibe Coder` because the agent has the flexibility to choose which tool to accomplish its task, which the research agent does not have (it only has the search tool).

Risk: "Overwriting or deleting database tables or files" is a security risk (data, application) caused either by agent failure or external manipulation of the File & Data Management capability.

Impact: **3/5** - The app is only deployed into a staging environment and never used in production, but the deletion of files and databases poses a major risk to the system's integrity.
Likelihood: **4/5** - Other agentic coding tools like Replit have failed in this manner before [Nolan, 2025], although this is relatively rare and not easily reproduced.

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

For `Vibe Coder`, there are a total of 25 relevant risks. This is partly because there are more risks, but also because the company's relevance threshold is lower, arising from a more conservative stance that requires more risks to be directly managed. This results in a much higher number of controls to be included, which is intuitive and sensible given the riskier nature of an agentic coding tool that can execute code and has permissions to modify system resources.

6 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 also be refreshed with industry best practices and new tools as they are launched.

7 Statement of Contributions

This paper contributes to ongoing discourse on agentic AI governance with the ARC framework by (1) introducing a novel capability perspective to analyze a wide range of agentic systems; (2) distilling three elements intrinsic to all agentic systems - components, design, and capabilities; (3) establishing a clear nexus between the elements, risks, and controls; and (4) providing a structured and practical approach to help organizations implement the framework. Additionally, we have included a comparison table to other technical governance frameworks for agentic systems in Appendix A.

8 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

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Eric Winsor, Jerome Wynne, Yarin Gal, and Xander Davies. Agentharm: A benchmark for measuring harmfulness of llm agents, 2025. URL https://arxiv.org/abs/2410.09024.

Luca Beurer-Kellner, Beat Buesser, Ana-Maria Creţu, Edoardo Debenedetti, Daniel Dobos, Daniel Fabian, Marc Fischer, David Froelicher, Kathrin Grosse, Daniel Naeff, Ezinwanne Ozoani, Andrew Paverd, Florian Tramèr, and Václav Volhejn. Design patterns for securing llm agents against prompt injections, 2025. URL https://arxiv.org/abs/2506.08837.

Cheng-Han Chiang et al. Harmful helper: Perform malicious tasks? web ai agents might help. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=4KoMbO2RJ9.

Edoardo Debenedetti, Jie Zhang, Mislav Balunović, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents, 2024. URL https://arxiv.org/abs/2406.13352.

Santiago Diaz, Christoph Kern, and Kara Olive. Google's approach for secure ai agents, 2025. URL https://research.google/pubs/an-introduction-to-googles-approach-for-secure-ai-agents/.

Zeynep Engin and David Hand. Toward adaptive categories: Dimensional governance for agentic ai, 2025. URL https://arxiv.org/abs/2505.11579.

European Parliament and Council of the European Union. 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, 2024. Accessed: 2025-05-11.

W.W. Gaver. Technology affordances. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 79–84, April 1991. doi: 10.1145/108844.108856.

Google. Agent2agent (a2a) protocol – latest. https://a2a-protocol.org/latest/, 2025. Accessed: 2025-10-11.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. Ai control: Improving safety despite intentional subversion, 2024. URL https://arxiv.org/abs/2312.06942.

Zhen Guo et al. Redcode: Risky code execution and generation benchmark for code agents. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bfd082c452dffb450d5a5202b0419205-Abstract-Datasets_and_Benchmarks_Track.html.

E. Hamilton. 2025 is the year of ai agents, openai cpo says. *Axios*, January 2025. URL https://www.axios.com/2025/01/23/davos-2025-ai-agents.

Richard Harang et al. Agentic autonomy levels and security, 2025. URL https://developer.nvidia.com/blog/agentic-autonomy-levels-and-security/.

Peter Ho. Opening address at 2010 administrative service dinner and promotion ceremony. Public Service Division, March 2010. URL https://www.psd.gov.sg/files/opening-address-by-mr-peter-ho-at-2010-administrative-service-dinner-and-promotion-ceremony.pdf.

Yuanzhao Huang et al. On the resilience of llm-based multi-agent collaboration with faulty agents. *arXiv preprint arXiv:2408.00989v3*, 2025. URL https://arxiv.org/abs/2408.00989v3.

Tomek Korbak, Mikita Balesni, Buck Shlegeris, and Geoffrey Irving. How to evaluate control measures for llm agents? a trajectory from today to superintelligence, 2025. URL https://arxiv.org/abs/2504.05259.

Ankit Kumar et al. Aligned llms are not aligned browser agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NsFZZU9gvk.

National Institute of Standards and Technology. Nist ai risk management framework playbook. https://www.nist.gov/itl/ai-risk-management-framework/nist-ai-rmf-playbook, 2023. Accessed: 2025-05-11.

Beatrice Nolan. An ai-powered coding tool wiped out a software company's database, then apologized for a 'catastrophic failure on my part'. https://fortune.com/2025/07/23/ai-coding-tool-replit-wiped-database-called-it-a-catastrophic-failure/, July 23 2025.

OpenAI. A practical guide to building agents, 2025. URL https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf.

OWASP. Agentic ai – threats and mitigations, 2025. URL https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/.

Shishir G. Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. Gorilla: Large language model connected with massive apis, 2023. URL https://arxiv.org/abs/2305.15334.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems, 2025. URL https://arxiv.org/abs/2506.04133.

Anka Reuel, Ben Bucknall, Stephen Casper, Tim Fist, Lisa Soder, Onni Aarne, Lewis Hammond, Lujain Ibrahim, Alan Chan, Peter Wills, Markus Anderljung, Ben Garfinkel, Lennart Heim, Andrew Trask, Gabriel Mukobi, Rylan Schaeffer, Mauricio Baker, Sara Hooker, Irene Solaiman, Alexandra Sasha Luccioni, Nitarshan Rajkumar, Nicolas Moës, Jeffrey Ladish, David Bau, Paul Bricman, Neel Guha, Jessica Newman, Yoshua Bengio, Tobin South, Alex Pentland, Sanmi Koyejo, Mykel J. Kochenderfer, and Robert Trager. Open problems in technical ai governance, 2025. URL https://arxiv.org/abs/2407.14981.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox, 2024. URL https://arxiv.org/abs/2309.15817.

Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O'Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, Katarina Slama, Lama Ahmad, Paul McMillan, Alex Beutel, Alexandre Passos, and David G. Robinson. Practices for governing agentic ai systems, 2025. URL https://cdn.openai.com/papers/practices-for-governing-agentic-ai-systems.pdf.

Tianneng Shi, Jingxuan He, Zhun Wang, Hongwei Li, Linyu Wu, Wenbo Guo, and Dawn Song. Progent: Programmable privilege control for llm agents, 2025. URL https://arxiv.org/abs/2504.11703.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable llm agents, 2025. URL https://arxiv.org/abs/2503.18666.

Junjie Ye, Sixian Li, Guanyu Li, Caishuang Huang, Songyang Gao, Yilong Wu, Qi Zhang, Tao Gui, and Xuanjing Huang. Toolsword: Unveiling safety issues of large language models in tool learning across three stages, 2024. URL https://arxiv.org/abs/2402.10753.

C. Yu and O. Papakyriakopoulos. Safety devolution in ai agents. In *ICLR 2025 Workshop on Human-AI Coevolution*, 2025. URL https://openreview.net/forum?id=7nJmuFFkWd.

Hanrong Zhang, Jingyuan Huang, Kai Mei, Yifei Yao, Zhenting Wang, Chenlu Zhan, Hongwei Wang, and Yongfeng Zhang. Agent security bench (asb): Formalizing and benchmarking attacks and defenses in llm-based agents, 2025. URL https://arxiv.org/abs/2410.02644.

Yuxuan Zhu, Antony Kellermann, Dylan Bowman, Philip Li, Akul Gupta, Adarsh Danda, Richard Fang, Conner Jensen, Eric Ihli, Jason Benn, Jet Geronimo, Avi Dhir, Sudhit Rao, Kaicheng Yu, Twm Stone, and Daniel Kang. Cve-bench: A benchmark for ai agents' ability to exploit real-world web application vulnerabilities, 2025. URL https://arxiv.org/abs/2503.17332.

A Comparison of the ARC framework with other technical governance frameworks

In the table below, we compare the ARC framework to three other frameworks (we are regrettably unable to include more due to space limitations).

Table 2: Comparison of ARC framework with alternative frameworks / guides relevant to agentic AI governance and safety.

Dimension	**ARC Framework (this paper)**	**Dimensional Governance (Engin & Hand, 2025)**	**OWASP Agentic AI (Threats & Mitigations)**	**Google: Secure AI Agents / SAIF 2.0**
Core framing	Capability-centric governance mapping *capabilities* → *risks* → *controls* with structured implementation approach.	Governance via continuous dimensions and trust thresholds (authority, autonomy, accountability).	Threat-model of agentic attack surfaces (reasoning, memory, tools, identity, oversight, multi-agent) with mitigations.	Principles for agents: human controller, limited powers, observable planning / actions; defense-in-depth.
Primary audience	Organizational governance plus product / security teams.	Policymakers, oversight / governance leads.	Security engineers, red / blue teams.	CISOs, security architects, enterprise builders.
Unit of analysis	**Capabilities** with components and system design.	**Dimensions** (continuous scales).	**Threats / attack surfaces** for agent workflows.	**Principles / control families** across lifecycle.
Prescriptiveness	Medium–High (risk → control mappings; checklists).	Low–Medium (conceptual thresholds, fewer concrete controls).	Medium–High (enumerated threats with mitigations).	Medium (principle-led control families).
Coverage of agentic specifics	Strong: capability lens tailored to powers / autonomy.	Conceptual: governance dynamics of agents.	Strong: agent-specific threats (tools, memory, multi-agent).	Strong: agent-explicit principles (controller, limits, observability).
Evidence / evaluation	Conceptual plus worked examples; no empirical evaluation yet.	Conceptual; no empirical evaluation.	Practitioner-grounded examples; no formal evaluation.	Policy / engineering narratives; no formal benchmarks.
Typical artifacts	Risk Register; capability profile; control tiers / checklists; sign-off workflow.	Dimension definitions; threshold guidance; oversight roles.	Threat navigator; threat / mitigation sheets; red-team prompts.	Principles plus control families; CISO guidance.
Control selection logic	By *capability profile* and *contextualized relevance threshold* (impact × likelihood) → minimum control set.	By *dimensional thresholds* (e.g., higher autonomy ⇒ stricter oversight).	By *threat presence* (e.g., tool misuse ⇒ sandboxing, PoLP).	By *principles* (limit powers; ensure observability; human controller).

Continued on next page

Dimension	ARC Framework (this paper)	Dimensional Governance (Engin & Hand, 2025)	OWASP Agentic AI (Threats & Mitigations)	Google: Secure AI Agents / SAIF 2.0
Verification / testing	Adversarial testing; logging / traceability (pre / post metrics recommended).	Oversight / accountability emphasized; testing not central.	Threat-led testing / red-teaming against agent surfaces.	Monitoring / observability of plans / actions emphasized.
Strengths	Holistic, capability-aware; ties risks to controls with governance workflow.	Clear governance lens; adaptivity via dimensions / thresholds; policy-friendly.	Security-grounded; actionable mitigations for engineers.	Enterprise-aligned principles; guardrails for agent power / visibility.
Gaps	Would benefit from empirical evaluation of the approach.	Less prescriptive; limited implementation detail.	Governance / process coverage thinner.	High-level; few concrete test harnesses / metrics.
Best fit	Org-level, cross-functional governance for varied agentic systems.	Regulators / oversight; policy design and audits.	Security hardening of agent stacks / tools.	Enterprise security principles for agent platforms.

B Risk Register (Risks)

We provide a preliminary version of a Risk Register below, with a mapping from the element to the risk. Due to space constraints, the controls are presented in a separate table in the next section.

Element	Name	Risk ID	Risk Statement and Description
Component	LLM	RISK-001	**Use of untrusted or compromised LLMs**: This risk arises when LLMs obtained from untrusted or insufficiently vetted sources have been intentionally poisoned or backdoored during training or distribution, causing them to behave maliciously or unpredictably under specific conditions. Such models may leak sensitive information, bypass safeguards, or execute hidden behaviors that undermine system integrity and trust.
		RISK-002	**Insufficient alignment of LLM behaviour**: This risk arises when an LLM's learned objectives and behaviors do not reliably align with intended user goals, system instructions, or organizational policies, leading to inappropriate, unsafe, or undesired outputs. Misalignment may surface as failure to follow constraints, inconsistent reasoning, or behavior that diverges from expected norms in edge cases or complex scenarios.

Element	Name	Risk ID	Risk Statement and Description
		RISK-003	**Insufficient LLM capability and reliability**: This risk arises when an LLM lacks sufficient capability, robustness, or reasoning performance to correctly interpret instructions, handle edge cases, or detect unsafe situations. As a result, the model may produce incorrect, misleading, or unsafe outputs that create downstream safety or security failures in systems that rely on its judgments.
Component	Tools	RISK-004	**Weak tool authentication and authorisation controls**: This risk arises when tools connected to an agent lack robust authentication or fine-grained authorisation mechanisms, allowing unauthorised access or misuse of tool capabilities. As a result, attackers or misbehaving agents may compromise the system by invoking sensitive actions, escalating privileges, or manipulating external resources beyond intended boundaries.
		RISK-005	**Lack of proper role-based access control for tools**: This risk arises when tools exposed to an agent do not enforce clear, role-based access controls, allowing agents to access capabilities or resources beyond their intended responsibilities. As a result, agents may perform unauthorised actions, misuse sensitive tools, or exceed their permitted scope, increasing the likelihood of security and operational failures.
		RISK-006	**Tool poisoning by malicious actors**: This risk arises when tools or their interfaces are intentionally modified, compromised, or replaced by malicious actors to introduce harmful or deceptive behaviour when invoked by an agent. As a result, the agent may unknowingly execute malicious actions, leak sensitive information, or produce manipulated outputs that undermine system integrity and trust.
		RISK-007	**Lack of input sanitisation**: This risk arises when inputs passed from the agent to tools are not properly validated or sanitised, allowing malformed or malicious data to be processed. As a result, tools may be exploited through injection attacks, unintended command execution, or data corruption.
Component	Instructions	RISK-008	**Vague or underspecified instructions**: This risk arises when instructions provided to an LLM are ambiguous, incomplete, or poorly scoped, leading the model to make unintended assumptions when interpreting tasks or constraints. As a result, the LLM may behave unpredictably, bypass safeguards, or take actions that introduce safety or security risks.

Element	Name	Risk ID	Risk Statement and Description
		RISK-009	**Unsanitised inputs in system instructions**: This risk arises when untrusted or user-controlled inputs are incorporated into system instructions without proper sanitisation or validation. As a result, malicious or malformed content may manipulate the model's behaviour, override intended constraints, or trigger unintended actions.
Component	Memory	RISK-010	**Poisoned memory**: This risk arises when the memory component of an agentic system is intentionally or inadvertently populated with malicious, misleading, or corrupted information. As a result, the agent may rely on compromised memory to make decisions, propagate false information, or exhibit persistent unsafe behaviour across interactions.
		RISK-011	**Sensitive data leakage across memory contexts**: This risk arises when the memory component retains or exposes sensitive information across sessions, tasks, or users with different scopes or authorisations. As a result, data may be inappropriately accessed or reused in unrelated contexts, leading to privacy breaches, confidentiality violations, or unauthorised disclosure.
Design	Agentic Architecture	RISK-012	**Cascading errors in multi-agent architectures**: This risk arises when errors or misjudgements produced by one agent propagate through interconnected agents within a multi-agent system. As a result, small failures may compound across agent interactions, leading to amplified errors, degraded system performance, or unintended outcomes at the system level.
		RISK-013	**Man-in-the-middle attacks between agents**: This risk arises when communication channels between agents are insufficiently secured, allowing an attacker to intercept, modify, or replay messages exchanged within the agentic system. As a result, agents may act on tampered information, leading to incorrect coordination, unauthorised actions, or compromised system behaviour.
		RISK-014	**Feedback loops and runaway agent behaviour**: This risk arises when agents repeatedly reinforce each other's decisions, outputs, or errors within an agentic architecture. As a result, feedback loops may form that escalate actions, consume excessive resources, or cause the system to persist in harmful or unintended behaviour without effective human intervention.

Element	Name	Risk ID	Risk Statement and Description
Design	Roles and Access Controls	RISK-015	**Overly permissive roles and permissions**: This risk arises when agents are granted roles or permissions that exceed their intended responsibilities or operational needs. As a result, agents may access sensitive resources, invoke high-impact capabilities, or perform unauthorised actions that increase the likelihood of security, privacy, or operational failures.
		RISK-016	**Unauthorised privilege escalation**: This risk arises when agents are able to gain elevated roles or permissions beyond those initially granted, whether through misconfiguration, exploitation, or unintended system behaviour. As a result, agents may bypass intended controls, access restricted resources, or execute actions that undermine system security and governance.
Design	Monitoring and Traceability	RISK-017	**Delayed failure detection due to limited monitoring**: This risk arises when monitoring systems provide insufficient visibility into agent behaviour, system events, or execution outcomes. As a result, failures, anomalies, or unintended actions may go undetected for extended periods, increasing the impact and difficulty of remediation.
		RISK-018	**Inability to audit failures due to missing decision traces**: This risk arises when monitoring systems do not capture sufficient reasoning steps, decision pathways, or execution context for agent actions. As a result, operators may be unable to reconstruct failures, understand why specific outcomes occurred, or conduct effective audits and post-incident reviews.
Capability	Planning and Goal Management	RISK-019	**Generating plans that fail to meet the user's requirements**: This risk arises when an agent generates plans or goals that do not accurately reflect the user's stated objectives, constraints, or preferences. As a result, the system may pursue incorrect or suboptimal actions, waste resources, or deliver outcomes that do not satisfy user expectations.
		RISK-020	**Generating plans that overlook safety implications**: This risk arises when an agent generates plans or goals without adequately considering basic safety, security, or practical constraints that would be apparent to a human. As a result, the system may propose or pursue actions that are unsafe, insecure, or inappropriate despite being technically feasible.

Element	Name	Risk ID	Risk Statement and Description
Capability	Agent Delegation	RISK-021	**Incorrect task delegation between agents**: This risk arises when an agent assigns tasks to other agents that do not match their capabilities, roles, or access permissions. As a result, tasks may be executed incorrectly, fail to complete, or introduce security and operational issues due to inappropriate delegation.
		RISK-022	**Malicious or manipulative use of delegated agents**: This risk arises when an agent deliberately assigns tasks to other agents in ways intended to bypass controls, obscure responsibility, or achieve malicious objectives. As a result, delegated agents may be coerced into performing unauthorised actions, amplifying harmful behaviour or evading detection within the system.
Capability	Tool Use	RISK-023	**Incorrect tool selection or misuse**: This risk arises when an agent selects an inappropriate tool or applies a tool incorrectly for a given task or action. As a result, the agent may produce erroneous outcomes, fail to complete the task effectively, or trigger unintended side effects due to misuse of tool capabilities.
Capability	Multimodal Understanding and Generation	RISK-024	**Generation of undesirable content**: This risk arises when an agent generates text, images, audio, or other media that contain toxic, hateful, sexual, or otherwise inappropriate content. As a result, the system may cause harm to users, violate organisational standards or regulations, or undermine trust in the system's outputs.
		RISK-025	**Generation of unqualified advice in specialised domains**: This risk arises when an agent generates advice or guidance in specialised domains such as medical, financial, or legal contexts without appropriate expertise, validation, or safeguards. As a result, users may act on incorrect or inappropriate information, leading to potential harm or adverse outcomes.
		RISK-026	**Generation of controversial or sensitive content**: This risk arises when an agent generates content related to sensitive or controversial topics, such as political commentary or denigrating comments about competitors. As a result, the system may create reputational, legal, or compliance issues, or be perceived as biased, inappropriate, or misrepresentative of organisational views.
		RISK-027	**Regurgitating personally identifiable information**: This risk arises when an agent reproduces personally identifiable information in its generated outputs, whether drawn from training data, memory, or prior interactions. As a result, the system may violate privacy obligations, expose individuals to harm, or breach data protection requirements.

Element	Name	Risk ID	Risk Statement and Description
		RISK-028	**Generation of non-factual or hallucinated content**: This risk arises when an agent generates information that is inaccurate, fabricated, or unsupported by evidence while presenting it as factual. As a result, users may be misled, make incorrect decisions, or lose trust in the system's outputs.
		RISK-029	**Generation of copyrighted content**: This risk arises when an agent generates content that reproduces or closely resembles copyrighted material without appropriate rights or attribution. As a result, the system may infringe intellectual property laws, expose the organisation to legal liability, or violate licensing and usage terms.
Capability	Official Communication	RISK-030	**Misrepresentation of authorship**: This risk arises when recipients are misled about whether an official communication was authored by a human or generated by an agent on behalf of the organisation. As a result, stakeholders may form incorrect assumptions about accountability, intent, or authority, potentially leading to trust, legal, or reputational issues.
		RISK-031	**Inaccurate promises or statements in official communications**: This risk arises when an agent makes commitments, assurances, or public statements that are incorrect, unsupported, or exceed the organisation's actual intentions or capabilities. As a result, the organisation may face reputational damage, legal exposure, or loss of public trust due to unmet expectations or misinformation.
Capability	Business Transactions	RISK-032	**Unauthorised execution of business transactions**: This risk arises when an agent initiates, authorises, or executes business transactions outside predefined approval thresholds, roles, or authorisation limits. As a result, the organisation may be exposed to unintended financial losses, binding contractual obligations, or operational commitments that were not properly sanctioned.
		RISK-033	**Leakage of transaction credentials**: This risk arises when credentials, tokens, or sensitive authentication information used to execute business transactions are exposed, mishandled, or improperly stored by an agent or its supporting systems. As a result, malicious parties may gain the ability to initiate unauthorised transactions, manipulate financial operations, or compromise transactional systems.

Element	Name	Risk ID	Risk Statement and Description
Capability	Internet and Search Access	RISK-034	**Prompt injection via malicious websites**: This risk arises when an agent retrieves or processes content from malicious or untrusted websites that are designed to inject instructions or manipulative prompts into the system. As a result, the agent may follow unintended commands, override intended constraints, or take actions that compromise system behaviour or integrity.
		RISK-035	**Unreliable information or websites**: This risk arises when an agent retrieves and presents information from websites that are inaccurate, outdated, biased, or otherwise unreliable. As a result, users may be misinformed or make incorrect decisions based on content that has not been adequately validated or corroborated.
Capability	Computer Use	RISK-036	**Prompt injection risks through computer use**: This risk arises when an agent interacts with graphical user interfaces that display untrusted or adversarial content - such as web pages, documents, pop-ups, or form fields - crafted to embed hidden instructions or manipulative cues. As a result, the agent may misinterpret on-screen text as authoritative guidance, follow injected instructions, or perform unintended actions while operating the interface.
		RISK-037	**Exposure of sensitive data**: This risk arises when an agent operating a computer interface accesses websites or applications that contain personally identifiable or sensitive information, particularly when authenticated as a user or organisation. As a result, the agent may inadvertently view, process, or disclose confidential data beyond its intended scope or authorisation.
Capability	Other Programmatic Interfaces	RISK-038	**Incorrect use of unfamiliar programmatic interfaces**: This risk arises when an agent interacts with programmatic interfaces it has not been trained or configured to use correctly, particularly bespoke or non-standard interfaces outside established protocols such as MCP servers. As a result, the agent may misinterpret interface semantics, invoke operations incorrectly, or produce unintended effects due to improper integration or usage.
Capability	Code Execution	RISK-039	**Production or execution of poor or ineffective code**: This risk arises when an agent generates or executes code that is incorrect, inefficient, insecure, or unsuitable for the intended task. As a result, the code may fail to achieve desired outcomes, introduce bugs or vulnerabilities, or cause operational disruptions when deployed or run.

Element	Name	Risk ID	Risk Statement and Description
		RISK-040	**Production or execution of vulnerable or malicious code**: This risk arises when an agent executes code that contains security vulnerabilities or intentionally malicious logic, whether generated by the model or sourced externally. As a result, the system may be compromised through exploitation, unauthorised access, data leakage, or other harmful effects.

Element	Name	Risk ID	Risk Statement and Description
Capability	File and Data Management	RISK-041	**Unintended overwriting or deletion of files or data**: This risk arises when an agent modifies, overwrites, or deletes files, database tables, or datasets without explicit user instruction or authorisation. As a result, critical information may be lost or corrupted, leading to data integrity issues, operational disruption, or the need for costly recovery efforts.
		RISK-042	**Database overload due to inefficient data operations**: This risk arises when an agent issues poorly optimised, excessively frequent, or redundant queries against databases or data stores. As a result, system performance may degrade, resources may be exhausted, or critical services may become unavailable due to unnecessary load.
		RISK-043	**Exposure of sensitive data through file or database access**: This risk arises when an agent accesses, processes, or outputs personally identifiable or sensitive information stored in files or databases without appropriate safeguards. As a result, confidential data may be disclosed to unauthorised parties, leading to privacy breaches, regulatory non-compliance, or loss of trust.
		RISK-044	**Prompt injection via malicious files or data**: This risk arises when an agent ingests or processes maliciously crafted files or data that embed hidden instructions or manipulative content. As a result, the agent may follow unintended prompts, alter its behaviour, or execute actions that compromise system safety or integrity.
Capability	System Management	RISK-045	**Misconfiguration of system resources**: This risk arises when an agent incorrectly configures system settings, infrastructure resources, or operational parameters. As a result, system performance, reliability, or security may be degraded, leading to service disruptions or unintended operational behaviour.
		RISK-046	**System overload due to inefficient or excessive operations**: This risk arises when an agent issues poorly optimised, excessively frequent, or redundant system-level operations or queries. As a result, computing resources may be exhausted, system performance may degrade, or services may become unavailable due to unnecessary load.

C Risk Register (Controls)

We provide a preliminary version of a Risk Register below, with a mapping from each risk to a control. Due to space constraints, the elements to risk mappings are presented in a separate table in the previous section. Control levels indicate criticality: Level 0 (Cardinal) are fundamental requirements, Level 1 (Standard) should be adopted or adapted meaningfully, and Level 2 (Best Practice) are recommended for high-risk systems.

Risk ID	Risk Statement	Control ID	Level	Control Statement
RISK-001	Use of untrusted or compromised LLMs	CTRL-0001	0	Use only LLMs from verified and trusted model developers
		CTRL-0002	0	Obtain legally binding no-training and no-logging agreements from LLM API service providers
		CTRL-0003	1	Use only established and verified model loaders in production environments
		CTRL-0006	1	Require human approval before executing high-impact actions
		CTRL-0007	0	Log all LLM inputs and outputs for regular review
RISK-002	Insufficient alignment of LLM behaviour	CTRL-0004	2	Review the LLM's system card to inform risk assessment and model selection
		CTRL-0005	0	Conduct structured evaluation of multiple LLMs for instruction-following, performance, and safety before deployment
		CTRL-0006	1	Require human approval before executing high-impact actions
		CTRL-0007	0	Log all LLM inputs and outputs for regular review
		CTRL-0008	1	Implement automated alerts when agent behaviour drifts from predefined thresholds
RISK-003	Insufficient LLM capability and reliability	CTRL-0004	2	Review the LLM's system card to inform risk assessment and model selection
		CTRL-0005	0	Conduct structured evaluation of multiple LLMs for instruction-following, performance, and safety before deployment
		CTRL-0006	1	Require human approval before executing high-impact actions
		CTRL-0007	0	Log all LLM inputs and outputs for regular review
RISK-004	Weak tool authentication and authorisation controls	CTRL-0009	0	Use only MCP servers that implement robust authentication mechanisms in production environments
		CTRL-0010	1	Use only MCP servers that validate credentials on every inbound request
		CTRL-0032	0	Centralise observability data collection in a unified backend system

Risk ID	Risk Statement	Control ID	Level	Control Statement
RISK-005	Lack of proper role-based access control for tools	CTRL-0011	0	Limit token scopes to the minimum privileges required and avoid broad or wildcard scopes
		CTRL-0012	2	Use only MCP servers that integrate with authorisation servers implementing per-client consent mechanisms
RISK-006	Tool poisoning by malicious actors	CTRL-0013	0	Test all untested MCP servers in a sandboxed environment before deploying to production
		CTRL-0014	0	Use only MCP servers from verified and trusted developers
RISK-007	Lack of input sanitisation	CTRL-0015	1	Treat all tool metadata and outputs as untrusted input requiring validation
RISK-008	Vague or underspecified instructions	CTRL-0016	0	Define clearly the agent's role, scope, and non-goals in the system prompt
		CTRL-0017	1	Define clear success criteria for the agent's tasks
		CTRL-0018	2	Define default behaviour when the agent encounters ambiguous situations
RISK-009	Unsanitised inputs in system instructions	CTRL-0019	0	Use delimiters to enclose untrusted inputs and instruct the LLM to treat delimited content as data only
		CTRL-0020	2	Use a dedicated LLM to extract required fields from inputs and filter out extraneous text or embedded instructions
RISK-010	Poisoned memory	CTRL-0021	0	Implement allowlists and denylists to restrict what categories of information can be written to agent memory
		CTRL-0022	1	Implement content filtering on memory writes to detect and block known unsafe content patterns
		CTRL-0023	2	Log all memory modifications with comprehensive source metadata for audit purposes
RISK-011	Sensitive data leakage across memory contexts	CTRL-0021	0	Implement allowlists and denylists to restrict what categories of information can be written to agent memory
		CTRL-0023	2	Log all memory modifications with comprehensive source metadata for audit purposes

Risk ID	Risk Statement	Control ID	Level	Control Statement
RISK-012	Cascading errors in multi-agent architectures	CTRL-0024	0	Define formal schemas for inter-agent messages and validate all messages against these schemas before processing
		CTRL-0025	1	Ensure all inter-agent communications are encrypted in transit and prohibit plaintext channels
RISK-013	Man-in-the-middle attacks between agents	CTRL-0026	1	Require all agents to authenticate with verifiable, cryptographically signed identities before processing requests
		CTRL-0027	1	Implement circuit breakers to prevent cascading failures in multi-agent systems
RISK-014	Feedback loops and runaway agent behaviour	CTRL-0028	0	Continuously monitor multi-agent systems for cascade failure indicators
		CTRL-0029	1	Grant agents only the minimum permissions required for their designated tasks
RISK-015	Overly permissive roles and permissions	CTRL-0030	1	Assign each agent a unique, verifiable identity with no shared credentials
		CTRL-0031	1	Use only MCP servers that validate token provenance and prohibit unauthorised token passthrough
RISK-016	Unauthorised privilege escalation	CTRL-0030	1	Assign each agent a unique, verifiable identity with no shared credentials
		CTRL-0032	0	Centralise observability data collection in a unified backend system
RISK-017	Delayed failure detection due to limited monitoring	CTRL-0033	0	Standardise trace attributes for agent operations using consistent semantic conventions
		CTRL-0035	2	Require agents to decompose user goals into explicit sub-goals and validate necessity before proceeding
RISK-018	Inability to audit failures due to missing decision traces	CTRL-0034	0	Conduct regular reviews of logs and traces to detect emergent issues in deployed agentic systems
		CTRL-0035	2	Require agents to decompose user goals into explicit sub-goals and validate necessity before proceeding
RISK-019	Generating plans that fail to meet the user's requirements	CTRL-0006	1	Require human approval before executing high-impact actions
		CTRL-0036	1	Regularly evaluate and test planning behaviour under representative workloads and failure scenarios

Risk ID	Risk Statement	Control ID	Level	Control Statement
		CTRL-0037	1	Require planning agents to include explicit safety constraints in all generated plans before execution
RISK-020	Generating plans that overlook safety implications	CTRL-0006	1	Require human approval before executing high-impact actions
		CTRL-0038	0	Conduct pre-deployment safety verification using domain-relevant stress tests and adversarial scenarios
		CTRL-0039	1	Ensure each agent publishes standardised, machine-readable capability descriptors accessible to other agents
RISK-021	Incorrect task delegation between agents	CTRL-0040	0	Limit the scope of agent actions through predefined thresholds and baselines
RISK-022	Malicious or manipulative use of delegated agents	CTRL-0008	1	Implement automated alerts when agent behaviour drifts from predefined thresholds
		CTRL-0024	0	Define formal schemas for inter-agent messages and validate all messages against these schemas before processing
		CTRL-0025	1	Ensure all inter-agent communications are encrypted in transit and prohibit plaintext channels
		CTRL-0041	0	Provide comprehensive descriptions for each tool including intended use, required inputs, and potential outputs
RISK-023	Incorrect tool selection or misuse	CTRL-0042	0	Require explicit human confirmation before executing high-impact or irreversible tool actions
		CTRL-0043	1	Log all tool selection decisions and invocations with comprehensive metadata
		CTRL-0044	1	Implement output safety guardrails to detect and prevent generation of undesirable content
RISK-024	Generation of undesirable content	CTRL-0045	0	Implement input guardrails to detect and decline requests for specialised domain advice
RISK-025	Generation of unqualified advice in specialised domains	CTRL-0046	0	Implement input guardrails to detect and decline requests for controversial content that violates organisational policies
RISK-026	Generation of controversial or sensitive content	CTRL-0047	0	Implement output guardrails to detect and redact personally identifiable information

Risk ID	Risk Statement	Control ID	Level	Control Statement
RISK-027	Regurgitating personally identifiable information	CTRL-0048	2	Implement methods to reduce hallucination rates in agent outputs
RISK-028	Generation of non-factual or hallucinated content	CTRL-0049	0	Implement UI/UX cues to communicate the risk of hallucination to users
		CTRL-0050	1	Implement features enabling users to verify generated answers against source content
		CTRL-0051	0	Implement input guardrails to detect and decline requests to generate copyrighted content
RISK-029	Generation of copyrighted content	CTRL-0052	2	Declare upfront that communications are generated by an AI system
RISK-030	Misrepresentation of authorship	CTRL-0053	0	Require human approval for communications on sensitive matters
RISK-031	Inaccurate promises or statements in official communications	CTRL-0054	0	Limit agent communications to standard processes with predefined templates
		CTRL-0055	1	Provide alternative channels for users to clarify communications or provide feedback
		CTRL-0056	1	Require explicit user confirmation before initiating or committing any business transaction
RISK-032	Unauthorised execution of business transactions	CTRL-0057	2	Require out-of-band confirmation when transaction risk signals are elevated
		CTRL-0058	1	Restrict agents to proposing transactions whilst using a separate transaction controller for execution
RISK-033	Leakage of transaction credentials	CTRL-0059	2	Apply fraud detection models or heuristics to agent-proposed transactions
		CTRL-0060	1	Implement escape filtering before incorporating web content into prompts
RISK-034	Prompt injection via malicious websites	CTRL-0061	0	Use structured retrieval APIs for web searches rather than web scraping
		CTRL-0062	0	Implement input guardrails to detect prompt injection and adversarial attacks
		CTRL-0063	1	Prioritise search results from verified, high-quality domains
RISK-035	Unreliable information or websites	CTRL-0064	1	Limit computer use to accessing only safe and trusted resources

Risk ID	Risk Statement	Control ID	Level	Control Statement
RISK-036	Prompt injection risks through computer use	CTRL-0065	0	Ensure computer use capabilities provide immediate interruptability
		CTRL-0066	0	Ensure "take over" mode is activated when entering sensitive data
RISK-037	Exposure of sensitive data	CTRL-0067	0	Ensure proper documentation of programmatic interfaces for agent use
RISK-038	Incorrect use of unfamiliar programmatic interfaces	CTRL-0068	0	Use code linters to screen generated code for bad practices and poor syntax
RISK-039	Production or execution of poor or ineffective code	CTRL-0069	0	Run agent-generated code only in isolated compute environments with network access blocked by default
		CTRL-0070	0	Review all agent-generated code before execution
		CTRL-0071	0	Use static code analysers to detect security vulnerabilities and code quality issues
		CTRL-0072	1	Monitor runtime and memory consumption of agent-generated code
		CTRL-0073	0	Create a denylist of commands that agents are not permitted to execute
RISK-040	Production or execution of vulnerable or malicious code	CTRL-0070	0	Review all agent-generated code before execution
		CTRL-0071	0	Use static code analysers to detect security vulnerabilities and code quality issues
		CTRL-0072	1	Monitor runtime and memory consumption of agent-generated code
		CTRL-0074	0	Conduct CVE scanning and block execution of code with High or Critical vulnerabilities
		CTRL-0075	1	Do not grant write access to agents unless strictly necessary
		CTRL-0076	1	Require human approval for any destructive changes to databases, tables, or files
RISK-041	Unintended overwriting or deletion of files or data	CTRL-0077	0	Enable versioning or soft-delete for managed object stores to allow recovery from accidental modifications
		CTRL-0078	0	Enforce throttling or rate limits on agent-initiated database operations

Risk ID	Risk Statement	Control ID	Level	Control Statement
		CTRL-0079	2	Validate agent-generated database queries for efficiency before execution against production databases
RISK-042	Database overload due to inefficient data operations	CTRL-0080	0	Implement caching mechanisms to reduce repetitive database queries by agents
		CTRL-0081	1	Implement input guardrails to detect personally identifiable information in data accessed by agents
		CTRL-0082	2	Do not grant agents access to personally identifiable or sensitive data unless strictly required
RISK-043	Exposure of sensitive data through file or database access	CTRL-0083	0	Disallow unknown or external files unless they have been scanned for threats
		CTRL-0084	0	Set minimum and maximum limits on what agents can modify within system resources
RISK-044	Prompt injection via malicious files or data	CTRL-0063	1	Prioritise search results from verified, high-quality domains
		CTRL-0085	0	Log system health metrics and implement automated alerts for abnormal conditions
RISK-045	Misconfiguration of system resources	CTRL-0086	0	Limit the number of concurrent queries to external systems by agents
		CTRL-0087	0	Ensure logging of system health metrics and automated alerts to the developer team if any metrics are abnormal
RISK-046	System overload due to inefficient or excessive operations	CTRL-0088	0	Limit the number of concurrent queries to external systems from the agent

D Worked Example: Researcher

To demonstrate how contextualization works, we fill out the assessment in the Risk Register for the `Researcher` example here. Note that the applicable relevance threshold is 3 for impact and 4 for likelihood. We highlight in red risks that exceed the relevance threshold, requiring additional controls to be implemented.

Category	Risk ID	Risk Description	Assessment
LLM	RISK-001	Use of untrusted or compromised LLMs	**Impact: 3/5** - Using compromised LLMs could lead to data leakage or manipulation of research outputs. **Likelihood: 1/5** - Current implementation uses verified LLMs from trusted providers. **Relevance**: **Not Relevant**
	RISK-002	Insufficient alignment of LLM behaviour	**Impact: 2/5** - Researcher is relatively low-stakes as its outputs are viewed and verified by a human-in-the-loop. **Likelihood: 1/5** - Task is quite narrowly scoped such that it is unlikely to perform non-research tasks. **Relevance**: **Not Relevant**
	RISK-003	Insufficient LLM capability and reliability	**Impact**: **4/5** - Insufficient capability means higher likelihood of poor reasoning, incorrect outputs, and safety hazards. **Likelihood**: **4/5** - Research has demonstrated that weaker LLMs are more prone to errors and prompt injection attacks, especially when handling long context which is common in Researcher. **Relevance**: **Relevant**
Instructions	RISK-008	Vague or underspecified instructions	**Impact: 1/5** - Users can re-run Researcher requests with step-by-step instructions if wrong actions are taken. **Likelihood: 1/5** - Researcher clarifies broad steps it will be taking (i.e., research directions) before proceeding. **Relevance**: **Not Relevant**
	RISK-009	Unsanitised inputs in system instructions	**Impact: 2/5** - While Researcher may be used for non-intended purposes, user outputs are consumed by the user only, limiting the impact. **Likelihood: 1/5** - Unlikely since delimiters are used to segregate system and user prompts. **Relevance**: **Not Relevant**

Category	Risk ID	Risk Description	Assessment
Tools	RISK-004	Weak tool authentication and authorisation controls	**Impact**: **1/5** - No privileged actions for this agentic system **Likelihood**: **1/5** - Current implementation relies on trustworthy Internet search tools like DuckDuckGo. **Relevance**: **Not relevant**
	RISK-005	Lack of proper role-based access control for tools	**Impact: 2/5** - No sensitive data stored in the agent, but such tools may allow for lateral access. **Likelihood: 2/5** - Current implementation relies on trustworthy Internet search tools like DuckDuckGo. **Relevance**: **Not Relevant**
	RISK-006	Tool poisoning by malicious actors	**Impact: 2/5** - Malicious code could shut down the process but each request is processed in an isolated container, reducing its impact on host system. **Likelihood: 1/5** - Current implementation relies on trustworthy Internet search tools like DuckDuckGo. **Relevance**: **Not Relevant**
	RISK-007	Lack of input sanitisation	**Impact**: **4/5** - Lack of input sanitation means higher likelihood of a jailbreak being passed to the LLM, leading to safety and security hazards. **Likelihood**: **5/5** - Tools without input sanitation have been demonstrated to be particularly susceptible to even simple prompt injection attacks. **Relevance**: **Relevant**
Memory	RISK-010	Poisoned memory	**Impact: 3/5** - Incorrect data or facts can lead to inaccurate research, but users are expected to check outputs. **Likelihood: 1/5** - Memory store is protected and can only be updated by authorised system owner, and Researcher tasks tend to be single-turn completions. **Relevance**: **Not Relevant**

Category	Risk ID	Risk Description	Assessment
	RISK-011	Sensitive data leakage across memory contexts	**Impact: 1/5** - Prior interactions are stored in a separate database, and not provided to the agent at runtime **Likelihood: 1/5** - Researcher tasks tend to be open-ended queries and do not include specific information. **Relevance**: **Not Relevant**
Agentic Architecture	RISK-012	Cascading errors in multi-agent architectures	**Impact**: **4/5** - Success of the research depends a lot on the research questions and direction being set by the first agent, and the agent which summarizes the information depends on accurate data being returned by the web search agent. **Likelihood**: **4/5** - Architecture does not require reflexive checking of statements from prior agents, making this quite a likely risk. **Relevance**: **Relevant**
	RISK-013	Man-in-the-middle attacks between agents	**Impact**: **2/5** - Even if such attacks occur, only system integrity will be affected, but there are no sensitive data in the agentic system. **Likelihood**: **1/5** - Unlikely that MitM attacks will succeed due to the security of the A2A protocol **Relevance**: **Not relevant**
	RISK-018	Inability to audit failures due to missing decision traces	**Impact**: **1/5** - Logging is straightforward for this system **Likelihood**: **2/5** - Linear architecture makes reconstruction of past reasoning traces relatively easy **Relevance**: **Not relevant**
Roles and Access Controls	RISK-015	Overly permissive roles and permissions	**Impact**: **1/5** - No restricted resources, so overly permissive roles are unlikely to have much effect. **Likelihood**: **1/5** - No access controls granted over internal restricted data or files **Relevance**: **Not relevant**

Category	Risk ID	Risk Description	Assessment
	RISK-016	Unauthorised privilege escalation	**Impact**: **1/5** - No restricted resources **Likelihood**: **1/5** - No access controls granted over internal restricted data or files **Relevance**: **Not relevant**
Monitoring and Traceability	RISK-017	Delayed failure detection due to limited monitoring	**Impact**: **2/5** - Impact is largely limited to the system, and there are user feedback channels available **Likelihood**: **2/5** - Generally rare unless there are wider outages **Relevance**: **Not relevant**
	RISK-018	Inability to audit failures due to missing decision traces	**Impact**: **2/5** - Largely confined to the system-level, and decision-making for this system is linear so it is easier to trace. **Likelihood**: **2/5** - Possible in theory, but no demonstrated failures so far for the research agent **Relevance**: **Not relevant**
Planning and Goal Management	RISK-019	Generating plans that fail to meet the user's requirements	**Impact**: **3/5** - Largely confined to the system-level as users will simply stop using the agent if it produces poor-quality outputs **Likelihood**: **2/5** - Generally rare unless the topic is highly specialized **Relevance**: **Not relevant**
	RISK-020	Generating plans that overlook safety implications	**Impact**: **1/5** - Largely confined to the system-level as users will simply stop using the agent if it produces nonsensical outputs **Likelihood**: **2/5** - Generally rare unless the topic is highly specialized **Relevance**: **Not relevant**

Category	Risk ID	Risk Description	Assessment
Multimodal Understanding and Generation	RISK-024	Generation of undesirable content	**Impact**: **4/5** - Undesirable content may shock and offend users, especially for particularly discriminatory or NSFW content. **Likelihood**: **4/5** - LLMs and agents have been demonstrated to be susceptible to attacks to generate such undesirable content. **Relevance**: **Relevant**
	RISK-025	Generation of unqualified advice in specialised domains	**Impact**: **4/5** - As the `Researcher` is meant to help people do deep research into specific, complex topics, users are likely to trust the outputs even if the advice is unqualified, which in turn may result in significant safety and liability concerns. **Likelihood**: **4/5** - While most LLMs tend to qualify their statements and ask users to seek professional advice, they still provide their advice to the users anyway. **Relevance**: **Relevant**
	RISK-026	Generation of controversial or sensitive content	**Impact**: **3/5** - As the system will take in any research topic by the user, it is plausible for a user to ask the research agent to do research into controversial topics. **Likelihood**: **4/5** - While this has not been systematically demonstrated in academic papers, it is relatively easy to carry this out even in standard research agent applications. **Relevance**: **Relevant**

Category	Risk ID	Risk Description	Assessment
	RISK-027	Regurgitating personally identifiable information	**Impact**: **2/5** - As the agentic system will only search the internet, any information returned to the user has to be discoverable on the public internet, which limits the liability and impact of PII regurgitation. **Likelihood**: **3/5** - While there have been several papers demonstrating attacks on LLMs for regurgitation of PII, they have not been proven to succeed against agentic systems and web search agents. **Relevance**: **Not relevant**
	RISK-028	Generation of non-factual or hallucinated content	**Impact**: **4/5** - Providing factually accurate information is a core feature of this agentic system. **Likelihood**: **5/5** - Several studies have shown that LLMs have a strong tendency to hallucinate, especially for highly specialized topics which users are likely to ask research agentic systems about. **Relevance**: **Relevant**
	RISK-029	Generation of copyrighted content	**Impact**: **3/5** - Potential legal liability if copyrighted content is reproduced by the system **Likelihood**: **2/5** - While LLMs have been shown to reproduce copyrighted content in full when asked for, this has not been demonstrated for web search agents which have to summarize multiple pages. **Relevance**: **Not relevant**
Internet and Search Access	RISK-034	Prompt injection via malicious websites	**Impact**: **4/5** - Manipulation of the agent can result in a range of safety and security risks that extend beyond the system's boundaries and result in reputational loss for the company. **Likelihood**: **5/5** - Attack has been demonstrated in several real-world case studies, no access to the system required to execute attack. **Relevance**: **Relevant**

Category	Risk ID	Risk Description	Assessment
	RISK-035	Unreliable information or websites	**Impact**: **4/5** - Summarizing reliable and accurate information is a core feature for the system and cannot be compromised on. **Likelihood**: **5/5** - Several real-world examples have demonstrated LLMs sometimes return statements from satirical websites as the truth. **Relevance**: **Relevant**