ARCvisor: Automating Technical Risk Assessments for Agentic AI Systems

Jessica Foo[1], Shaun Khoo[1], Roy Ka-Wei Lee[2],
[1]GovTech Singapore, [2]Singapore University of Technology and Design
Correspondence: jessica_foo@tech.gov.sg

Abstract

We present **ARCvisor**, a web-based, open-source system that operationalizes the Agentic Risk & Capability ("ARC") Framework by enabling organizations to declaratively define agentic AI systems (components, design, capabilities), automatically identify relevant risks from a canonical risk registry, contextualize them under user-specified impact/likelihood thresholds, and generate structured risk assessment reports. ARCvisor bridges the gap between conceptual AI governance frameworks and practical adoption of agentic AI through enabling organization-wide scalability, promoting standardization and consistency, while ensuring human accountability throughout the process. We describe the system architecture, user workflow, and initial evaluation results, and open-source the tool here to support community adoption and external validation.

1 Introduction

The emergence of agentic AI - systems capable of autonomous code execution, internet access, file and data manipulation, and other powerful actions - presents both transformative opportunities and substantial risks. While there are technical governance frameworks, like the AI Trust, Risk, and Security Management Framework Raza et al. (2025), the dimensional governance framework (Engin and Hand, 2025), and the Agentic Risk & Capability ("ARC") Framework (Khoo et al., 2025), that provide a conceptual framework for identifying and understanding risks from agentic AI systems, applying them in practice remains non-trivial. Manual risk assessments are time-consuming, error-prone, and difficult to scale or maintain as systems evolve.

To address this, we built ARCvisor: an interactive web tool that operationalizes the ARC Framework by automatically producing a contextualized risk assessment along with the recommended technical controls based on a detailed description of the agentic AI system in terms of its elements (components, design, capabilities) and deployment context (domain, data sensitivity, criticality). By reducing the effort of conducting systematic risk assessments, ARCvisor enables more widespread, consistent, and auditable governance of agentic AI.

This work contributes:

- A novel open-source system implementing the ARC Framework in a usable, scalable form.
- A web-based UI and workflow suitable for real-world governance teams (non-researchers), promoting accessibility and adoption.
- An initial internal evaluation demonstrating usability, consistency, and time-savings over manual assessment.

2 Related Work

Many existing AI governance efforts — such as regulatory frameworks or risk-management guidelines — set out high-level principles but lack effective tooling for assessing the myriad of risks arising from agentic systems, 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). Compliance thus depends on ad-hoc spreadsheets, manual audits, or undocumented internal processes — making them error-prone, inconsistent, and hard to scale.

There are very few open-source tools for organizations to systematically apply technical AI governance frameworks - most tools in this space focus on testing, such as Inspect (AI Security Institute) or Garak (Derczynski et al., 2024). The only risk assessment tool we could find is AgentWiz by Repello AI (Repello-AI), which applies the MAESTRO framework (Huang et al., 2025) and uses it for automated threat modelling for the

given agentic AI system. However, this does not cover safety risks and lacks recommendations for technical controls. Moreover, unlike Agent-Wiz, ARCvisor intentionally incorporates human decisions in the assessment workflow, ensuring human accountability. ARCvisor also offers a structured and reproducible workflow for generating audit-ready risk reports, which is critical in an enterprise context. To our knowledge, ARCvisor is among the first open-source tools explicitly targeting risk assessment for agentic AI systems, making governance accessible to organizations without deep in-house AI safety or security expertise.

3 Agentic Risk & Capability Framework

The ARC Framework provides a structured taxonomy and process for analysing and governing agentic AI systems and serves as the foundation of ARCvisor. In this section, we briefly explain each part of the framework (elements, risks, controls), and refer the reader to the original paper for more details.

3.1 Elements of Agentic Systems

An agentic system is characterized by three indispensable elements:

- **Components**, such as its core reasoning engine (LLM), its attached tools (APIs, file system, databases, etc.), its instruction template (prompts or policies) and memory or context storage.
- **System Design**, which captures how agents are orchestrated (agentic architecture), what access controls and roles are specified, and what monitoring or traceability mechanisms are in place.
- **Capabilities**, i.e. the autonomous actions the system is allowed to perform — for example cognitive (planning, delegation, tool selection), interaction (language, multimodal input/output, external communication), or operational (code execution, data/file manipulation, system operations).

3.2 Risks

Given a system's elements, ARC identifies how risks can materialize through general *failure modes* - for instance, agent misbehavior, external manipulation, or tool/resource malfunction - which may lead to various security or safety *hazards* (e.g. data leakage, privilege escalation, misinformation, unsafe behavior, etc.). These are formalized in a **Risk Register** that maps element + failure mode + hazard to concrete risk entries.

This structured, policy-as-code approach enables governance teams to treat risk assessment as a repeatable, auditable process rather than an ad-hoc checklist. By encoding risks, controls, and their triggers in machine-readable form, organizations can automatically check, track, version-control, and update risk profiles as agents evolve or as new threats emerge.

3.3 Controls and Mitigations

Each identified risk is matched with recommended technical controls - measures that either reduce the probability of failure or limit the impact of a hazard. The framework also recognizes that even with controls in place, residual risks may remain, especially in light of evolving capabilities and threat contexts.

3.4 Contextualization and Governance Process

To make the framework actionable, organizations are encouraged to contextualize risk assessment: for a given agentic system, governance teams specify deployment context (e.g. domain, data sensitivity, criticality) and set relevance thresholds (on impact / likelihood) to filter which risks demand mitigation. This allows differentiated governance - lighter oversight for low-capability or low-impact systems, stricter controls for high-risk ones - while maintaining a consistent, auditable process across diverse agentic deployments.

4 System Overview

4.1 Design Goals & Target Users

ARCvisor was designed to simplify the risk assessment process for users not trained in cybersecurity, including software developers, product managers and governance teams. By formalizing the ARC's capabilities-centric approach in a tool, the ARCvisor ensures standardization and scalability of risk assessments, while allowing for application-specific contextualization.

- **Declarative risk modeling**: Users describe what their agentic AI system comprises or provide a Github code repository link, without needing to specify how the risk-assessment should be done. Based on the application's

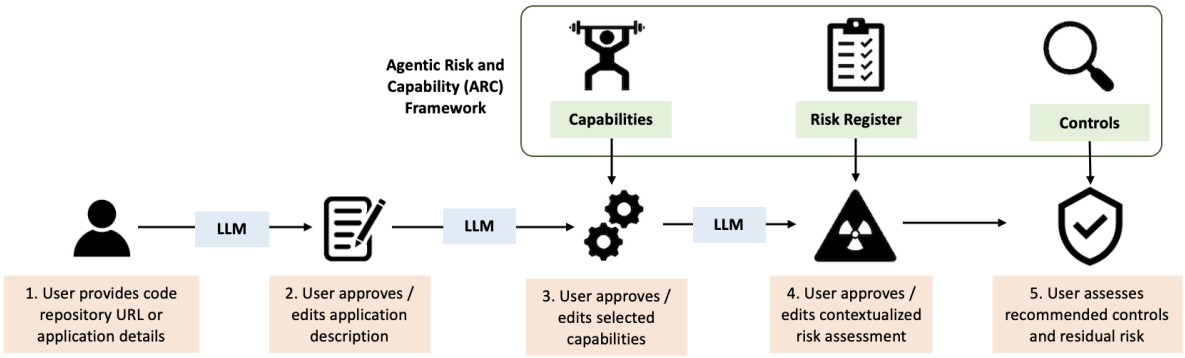

Figure 1: Overview of ARCvisor User Workflow

code or user's description, the tool automatically infers which relevant ARC capabilities apply, and generates contextualized risk profiles and suggested controls.

- **Contextualized Assessments**: ARCvisor contextualizes each applicable risk to the application analyzed, specifying how the risk may materialize based on the application's identified components, design or capabilities. This includes a likelihood and impact score that is tailored to the application. Furthermore, users can define score thresholds to retain only relevant or high-impact risks for mitigation through technical controls.

- **Documentation and Auditability**: Designed with AI auditing in mind, the process of risk identification, contextualization and mitigation is codified into a structured workflow, which can eventually be exported as reports. This step-by-step process aligns with general requirements for documentation, accountability, and auditability, which are crucial pillars for governance.

- **Human-centered Accountability**: The tool prioritizes human oversight across the automated workflow, ensuring that humans remain accountable for each step of the assessment. In particular, even as parts of the workflow (i.e., description of system, identification of capabilities, contextualization of risks) are generated by LLMs, users are explicitly asked to review the outputs and amend them for accuracy. This ensures that humans retain ultimate responsibility over the final risk assessments and mitigation decisions.

4.2 System Design and Architecture

Risk Register. The core of ARCvisor is the "policy-as-code" implementation of the ARC framework as a Risk Register and accompanying catalog of controls - all agentic system definitions, risk mappings, and control recommendations are captured in declarative YAML files. In addition, many-to-many relationships are established with risk and control IDs, that are consistently referenced across files. This structured, machine-readable representation enables automated and repeatable risk assessment - users (aided by LLMs) simply declare their system's components, design, and capabilities, and the tool resolves the corresponding risks and mitigation controls via the encoded policy logic. Even as the taxonomy of capabilities, risks, or controls evolves, governance is centrally managed by only updating the underlying codebase. All subsequent downstream risk assessments automatically inherit the update. This ensures that all assessments remain consistent and traceable, supporting large-scale governance of agentic systems, while preserving accountability via version control.

LLM Suggestions. With the core logic mapping capabilities $\rightarrow$ risks $\rightarrow$ controls handled by the Risk Register, the contextualization of risks to the specific application is managed by LLMs. Firstly, to analyze large code repositories, code files are searched and prioritized according to a pre-defined list of extensions (e.g., .md, Docker files) and references to key capabilities (e.g., API requests, databases). A coding LLM (i.e., `openai-5.1-codex`) is then used to analyze the files and return a description of the application. If the user chooses to fill in the text boxes to describe the application, a full description is then generated by a general-purpose, instruction

following LLM (i.e., `openai-4.1-mini`). Given the system's description and the ARC Framework's list of system components, design and capabilities, an LLM (i.e., `openai-4.1-mini`) is used to pre-select applicable capabilities. Next, based on the selected capabilities and the ARC Framework's Risk Register, a more capable LLM (i.e., `openai-5.1`) contextualizes each relevant risk to the application, providing a likelihood and impact score (out of 5) and a reasoning for the score. Under the hood, the system uses LiteLLM to enable API calls to any LLM family or LLM provider, making it flexible for organizations to change the LLMs used in the assessment process.

Web application. To allow users to iterate and participate in the risk assessment process, the system uses a web frontend (built with Streamlit) for user interaction, prompting users to verify, approve or make edits. At the end of the workflow, users can export their audit report accordingly.

5 Functionality & User Workflow

Figure 1 illustrates ARCvisor's high-level user flow.

5.1 System Declaration

Users begin by declaring the agentic AI system under assessment in the landing page as seen in 2. They can do so either by providing a public Github repository link or filling up a guided form, as seen in Figure 2. In the guided form, users describe the functionality and purpose of their application, technical components (e.g., architecture, tools, integrations, data flows), and risk factors such as data sensitivity, system criticality/ accessibility or existing human-in-the-loop processes. Based on the user's inputs, a detailed description of the application is generated, which the user can approve or edit.

5.2 Capabilities Identification

Based on the description and the ARC Framework's list of capabilities, relevant capabilities are pre-selected and presented to the user as checkboxes. Users are also provided a full description of capabilities in order to validate the LLM's selection or select additional capabilities.

5.3 Risk Assessment

Based on the selected capabilities, users are given a contextualized risk assessment for each applicable

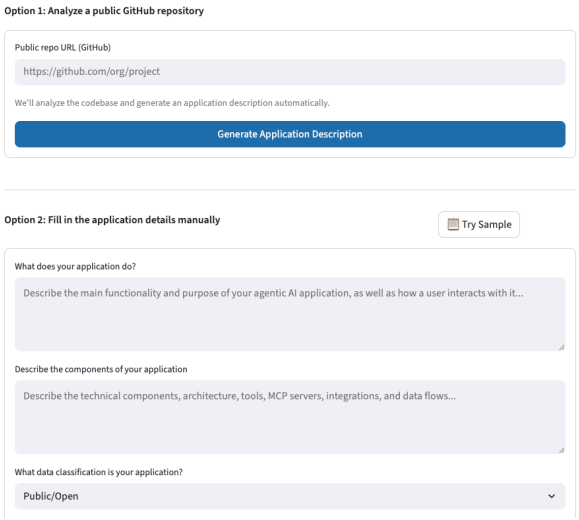

Figure 2: ARCvisor: Users choose to provide Github code repository or fill up a guided form.

risk according to the Risk Register, as seen in Figure 3. This includes risk IDs, descriptions, as well as likelihood and impact scores and explanations, which users can evaluate and edit accordingly. The score explanations are designed to reference specific failure modes and hazards that may arise for the application. Users can then define appropriate score thresholds for selection of mitigating measures, allowing for flexibility in risk appetites.

5.4 Risk Mitigation and Control

The system recommends controls from the ARC Framework's control catalog based on the risks that exceed the thresholds defined by the user in the previous step. Users then have the option to mark which controls are implemented or planned, and to describe residual risks that remain, as shown in Figure 4. Finally, users can export the assessment as a Word Doc report for audit and compliance purposes.

5.5 Example Use Case

To illustrate, in our demonstration we show three hypothetical agentic systems:

- A low-capability "chatbot-only" system (e.g., natural-language interaction, read-only file access) - resulting in a small set of low-impact, low-likelihood risks.
- A high-capability system: customer service chatbot executing business transactions with file/data access to sensitive customer data — generating a much larger and more serious risk profile, triggering more suggested controls.

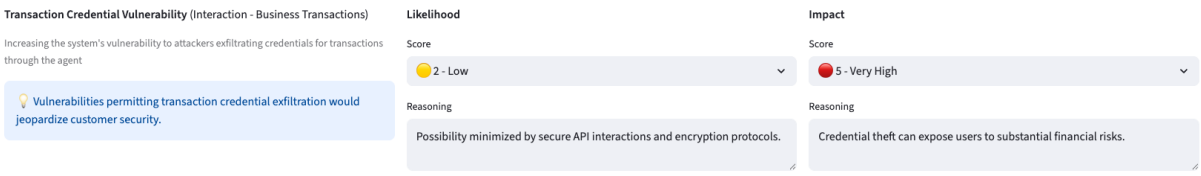

Figure 3: ARCvisor: Risk example from contextualized risk assessment for a customer service chatbot.

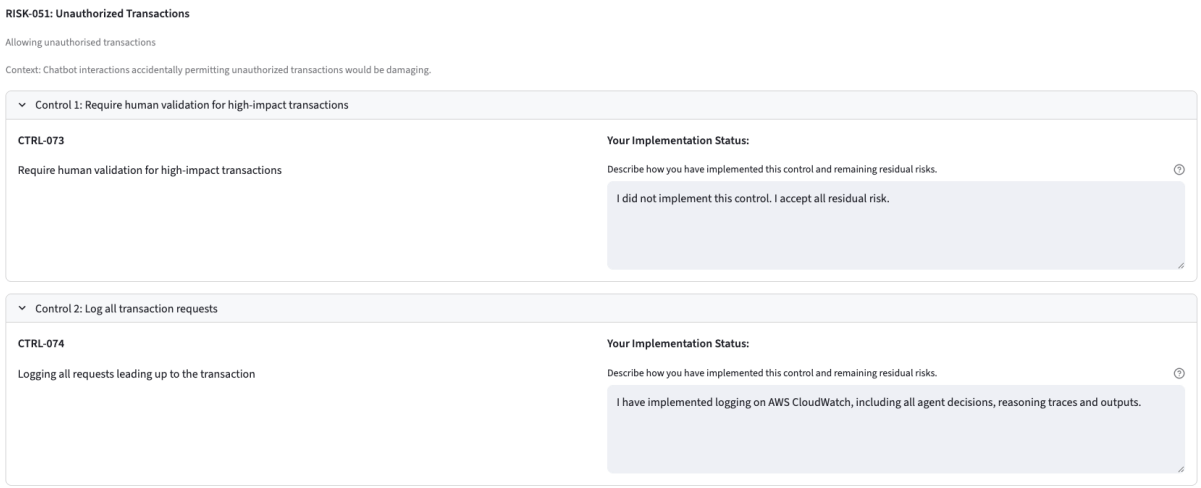

Figure 4: ARCvisor: Example controls and residual risk evaluation for report generation.

The demo video walks through the full user workflow: declaration → capabilities identification → risk assessment → mitigations.

6 Implementation Details

The ARCvisor codebase is publicly available under an open-source license. This includes the Risk Register and controls files, which allows organizations to customize risk categories and controls according to their own governance policies. Installation instructions and an optional Docker configuration are provided for local self-hosting, while a live demo instance is deployed here.

7 Evaluation

To evaluate ARCvisor, we conducted a small internal pilot study across participants drawn from governance and engineering teams, evaluating various real-world agentic AI systems that our organization is assessing for deployment. We collected the following metrics via a post-task survey, using a 5-point Likert scale:

- **Usability**: Clarity of workflow, ease-of-use, understandability of risk assessment
- **Accuracy**: Whether the pre-identified capabilities and contextualized risk assessments by LLMs were correct
- **Coverage**: Whether all "intuitively obvious" risks were surfaced by ARCvisor
- **Time efficiency**: As compared to a baseline manual risk-assessment exercise

Results. In our pilot study:

- Mean usability score was 4.8/5, while mean accuracy score was 4.4/5. In particular, all participants agreed that the new workflow was clear and easy to follow.
- Participants found that ARCvisor correctly identified the relevant capabilities and risks, even for complex agentic systems, although one participant highlighted that ARCvisor missed out one capability for his system.
- All participants agreed that ARCvisor reduced time-to-completion by more than 50% as compared to manual assessment, and the significant time-saving was explicitly praised by three participants.

These results provide early indications that ARCvisor will be useful in helping governance teams work faster, more consistently, and with better risk

coverage than manual methods. We acknowledge the small sample size and limited validation; future work will involve larger-scale studies with real-world agentic systems.

8 Conclusion

As agentic systems become increasingly prevalent, governance frameworks are essential for safe, ethical and responsible AI deployment. ARCvisor helps organizations translate the ARC framework's conceptual requirements into practical tools for governance, thus enabling organization-wide scalability, promoting standardization and consistency, while ensuring human accountability throughout the process. Moreover, by encoding the framework in structured, machine-readable policy files, ARCvisor enables efficient bulk assessments, version-controlled updates, continuous improvement, and auditability, thereby helping organizations keep pace with change without sacrificing oversight. Future work can build on this base by empirically validating the risks and controls in the Risk Register, integrating runtime monitoring or automated enforcement tools, and developing mechanisms for continuous feedback and registry updates.

Limitations

Due to the nascency of agentic AI and the complexity of AI governance, we were only able to conduct a small-scale internal survey for this system. This makes it hard to evaluate if this system will be as useful when scaled up for large organizations or when used by organizations in other industries. However, adoption for agentic AI is gaining speed and both the ARC framework and ARCvisor can easily be adapted, so in time we hope to scale up empirical validation of this system's usefulness.

Ethics and Broader Impact

ARCvisor is designed to enable safer, more responsible deployment of agentic AI systems by providing structured, transparent, and repeatable technical risk assessments. The tool operationalizes the ARC Framework into a practical workflow that supports accountability, auditability, and human oversight in real-world governance contexts. By helping organizations systematically identify safety, security, and ethical risks early in the development lifecycle, ARCvisor aims to reduce the likelihood of harmful outcomes associated with autonomous code execution, external tool use, or sensitive data access.

The system explicitly prioritizes human-centered accountability: while LLMs are used to assist in identifying capabilities and contextualizing risks, final decisions always remain with human reviewers. ARCvisor's design avoids fully automated approval or governance decisions, and instead requires manual verification and justification for each assessment step. This ensures that organizations retain responsibility over risk-based decisions and avoids over-reliance on automated judgments.

We acknowledge several ethical considerations. First, the Risk Register and underlying LLM-generated suggestions may introduce biases or incomplete coverage, especially in rapidly evolving threat landscapes. To mitigate this, ARCvisor is open-source and fully transparent, allowing independent auditing, extension, and critique by the community. Second, early evaluations were conducted internally with a small sample size; future larger-scale studies will be needed to ensure robustness across domains and contexts. Third, while ARCvisor can highlight risks and recommend technical controls, it does not guarantee the prevention of harm if deployed without complementary governance measures such as legal review, operational controls, or domain-specific compliance.

Overall, we believe ARCvisor contributes positively to the ecosystem by lowering the barrier for organizations—especially those without deep AI safety expertise—to adopt structured risk governance for agentic systems. As adoption of agentic AI accelerates, tools like ARCvisor can promote safer innovation, reduce unintentional harms, and support alignment with emerging governance standards and regulatory frameworks such as the EU AI Act and NIST AI RMF. We encourage responsible use and community participation in strengthening both the risk taxonomy and evaluation methodology.

References

UK AI Security Institute. Inspect AI: Framework for Large Language Model Evaluations.

Leon Derczynski, Erick Galinkin, Jeffrey Martin, Subho Majumdar, and Nanna Inie. 2024. garak: A Framework for Security Probing Large Language Models.

Zeynep Engin and David Hand. 2025. Toward adaptive categories: Dimensional governance for agentic ai. *Preprint*, arXiv:2505.11579.

European Parliament and Council of the European Union. 2024. Regulation (eu) 2024/1689 of the euro-

pean 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.

Yuanzhao Huang and 1 others. 2025. On the resilience of llm-based multi-agent collaboration with faulty agents. *arXiv preprint arXiv:2408.00989v3*.

Shaun Khoo, Jessica Foo, and Roy Ka-Wei Lee. 2025. With Great Capabilities Come Great Responsibilities: Introducing the Agentic Risk & Capability (ARC) Framework for Governing Agentic AI Systems. In *Proceedings of the AAAI 2026 3rd International AI Governance Workshop*. Accessed: 2025-12-02.

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.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. 2025. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. *Preprint*, arXiv:2506.04133.

Repello-AI. Agent-Wiz.