Author's copy of the paper published in Proceedings of the IEEE 5th Cyber Awareness and Research Symposium 2025 (CARS'25). Grand Forks, ND, October 27- 30, 2025.

A Case Study on Delegating Critical Tasks to Agentic AI and Prototype Access Control Methods

Sunyoung Kim
Arizona State University
School of Computing and Augmented Intelligence
Tempe, AZ, United States
skim638@asu.edu

Hokeun Kim
Arizona State University
School of Computing and Augmented Intelligence
Tempe, AZ, United States
hokeun@asu.edu

***Abstract*—The use of Agentic AI has become prevalent beyond non-critical tasks such as reply suggestions, writing guidance, or problem-solving. Agentic AI has emerged in the delegation of critical tasks, such as making purchases with credit cards, managing banking/finances, and planning business travel that involves booking flights or hotel reservations. However, security risks and privacy concerns have been significant barriers to using Agentic AI to perform critical tasks. Although there have been vendor-specific solutions for Agentic AI running critical tasks such as Agent Pay by Mastercard, there has not been a generalized solution yet. To better understand this gap, this paper conducts a case study on the use of currently available Agentic AI to perform three scenarios of critical tasks on commercial websites. Based on observations from our case study, we also propose a prototype design of our access control approach for the delegation of critical tasks to Agentic AI, with a prototype website with fine-grained access control mechanisms.**

***Index Terms*—Agentic AI, Critical Tasks, Delegation, Access Control**

I. Introduction and Motivation

The introduction of Agentic AI has brought significant convenience to our lives. Traditional AI research has focused on reactively generating appropriate responses to user input [1]. Recent advances in large language models (LLMs) have led to the emergence of Agentic AI, which moves beyond reactive behavior to autonomously plan, decide, and act to achieve complex tasks [2]. Agentic AI extends its capabilities to assist and even influence the decisions we make in our daily work. Over the next few years, Agentic AI is expected to be increasingly adopted across industries [3]. Beyond industrial applications, it is anticipated to become further integrated into daily activities, handling a remarkable share of online shopping tasks and gradually reshaping the e-commerce environment [4]. Agentic AI can interact with external resources, expanding its scope of activity. However, as shown in Fig. 1, such capabilities often require users to provide the AI with sensitive information, such as credit card information, address, or social security numbers, creating serious security and privacy risks [5]. In addition, the well-known issue of hallucination in LLMs raises additional concerns when using AI agents [6]. Unintended actions triggered by hallucinated outputs, especially after receiving sensitive information, could have serious security implications.

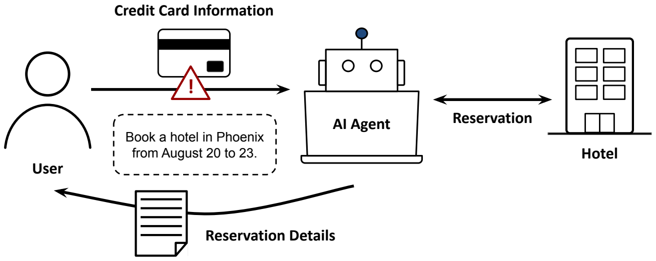

Fig. 1. Example scenario of Agentic AI interacting with an external resource. The user provides natural language instructions with sensitive information, such as credit card details, to the agent, who then communicates with the hotel system to complete the user's request. This figure illustrates the security risks and limitations of Agentic AI in real-world applications.

AI agents face significant limitations when performing complex tasks or end-to-end activities. Although they can streamline and simplify many user tasks, they still depend on substantial human involvement. Security constraints, such as bot detection or Google reCAPTCHA [7], often require human assistance when an AI agent attempts to log in to websites on behalf of the users, restricting the agent's operational scope. For example, OpenAI's Agentic AI, Operator [8], also requires human participation in specific tasks. According to their description, the Operator is trained to decline tasks that handle sensitive information, such as banking transactions. In addition, it requires human intervention when encountering security challenges, such as login processes or CAPTCHA prompts. When AI agents operate within already logged-in mobile applications to perform assigned tasks, communication with the client remains necessary [9]. In other cases, where agents only provide passive guidance on task steps or perform limited actions, active user interaction is also required [10]. Despite these observations, there is a lack of case studies that examine the limitations of AI agents performing real-world tasks autonomously from start to finish with sensitive data, in the manner of a human assistant.

In this paper, we conduct a case study on current Agentic AI systems, focusing on their capabilities in handling critical and sensitive data, such as credit card and payment information. We also develop a prototype website and implement key capabilities and application programming interface (API) functions to support these needs.

II. Background and Related Work

Existing research on access control in Agentic AI systems has mainly focused on managing and monitoring an agent's access to user accounts. For example, Stytch [11] provides OAuth-based access and session management frameworks, using role-based access control to define the scopes within which agents can operate. Permit.io [12], in combination with PydanticAI [13], offers fine-grained access control mechanisms, such as prompt filtering, secure external access, and additional data protection features. However, Permit.io's approach is primarily centered on restricting agents' access to sensitive data, rather than on ensuring how such data is securely processed once accessed. Current solutions focus on verifying who the agent is and what categories of actions it may perform, rather than addressing in depth how the agent should handle, process, and safeguard highly sensitive information, such as credit card numbers, addresses, and payment details.

Recently, Mastercard launched Agent Pay [14], an infrastructure designed to enable Agentic AI systems to make secure payments in real-world commerce. Agent Pay focuses on creating a payment ecosystem for AI agents, allowing them to autonomously initiate and complete transactions on behalf of users. This is achieved by using Agentic Tokens [15], a data obfuscation technique that replaces sensitive card information with secure digital signatures, minimizing the risk of data exposure during autonomous transactions. Although this represents an important step towards supporting secure financial activities in the age of AI, Mastercard's framework is still limited to its own payment network. Other payment methods, such as Visa, Venmo, or cryptocurrency, cannot be integrated into this framework. In addition, the final approval of each transaction still requires explicit human intervention.

Our work aims to provide a more general and extensible solution that is not bound to a single payment provider. We propose a secure agentic environment that can be seamlessly integrated with widely used platforms such as PayPal, WePay [16], and any other mainstream payment service providers. Furthermore, to foster collaboration and community adoption, we plan to release our system as an open-source project on GitHub[1]. In this paper, we demonstrate a case study to analyze the current capabilities of Agentic AI.

III. Case Study: LLMs interacting with real-world websites with controlled access

To evaluate the capabilities of current Agentic AI systems and to examine their limitations in performing tasks that require handling sensitive information on real-world websites, we conduct a case study with different scenarios described in Fig. 2. To overcome these limitations, we design a prototype website for Agentic AI. For our experiments, we use the Meta Llama-3.1-8B-Instruct model and Llama-3.2-3B-Instruct model, implemented in Python 3.12.11, and run the workloads on an NVIDIA A100 GPU. We grant the agent access to execute the Firefox browser for all interactions between the agent and external websites. The experiments are executed in a server environment without persistent user profiles, browsing history, or residential IP addresses, which may differ from typical end-user settings. For security considerations, all scenarios are designed to proceed only up to the point before executing payment or money transfer operations.

A. Hotel Reservation

Scenario: Consider a scenario in Fig. 2a where a user asks an Agentic AI to book a hotel for a business trip. The user provides the location, travel dates, and specific conditions, such as requiring a 4-star hotel or higher and a nightly rate under $300. To complete the task, Agentic AI searches the reservation platforms (e.g., booking.com [17]) for suitable options and then requests the user's information, such as username, password, and credit card information, to finalize the reservation.

The Agentic AI successfully set the destination and applied the four-star filter to generate appropriate search results. However, since parameters such as check-in and check-out dates, the number of guests, and the number of rooms were presented in graphical or non-textual formats, the agent was unable to satisfy the user's constraints and finish searching. In some trials, the Agentic AI was able to reach the reservation system's login page. Nevertheless, Agentic AI was unable to complete the login process autonomously and required human intervention due to security barriers. Specifically, Booking.com enforced additional authentication steps, including email-based verification and a reCAPTCHA challenge, which functioned as intended to block automated access. Moreover, in many cases, the agent incorrectly concluded that the task was completed even when it was not.

B. Product Purchase

Scenario: Consider a scenario in Fig. 2b where a user asks Agentic AI to purchase an item. The user specifies the desired product and price range and then provides sensitive information such as the shipping address and credit card details. The agent searches for the item via external e-commerce platforms (e.g., Amazon [18]) to identify items that meet the given conditions and proceeds with the purchase on behalf of the user.

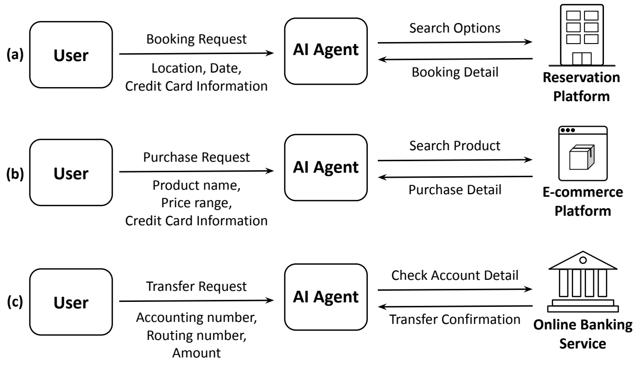

Fig. 2. Case study scenarios of Agentic AI interacting with an external resource to perform tasks on behalf of the user.

[1] https://github.com/asu-kim/agentic-ai-access-control

TABLE I

Summary of case study results – progress of Agentic AI across scenarios.

Scenario	Trials	Task Completion Rate	Search Success Rate	Login Success Rate	Average Number of Human Intervention	Hallucination Rate	Unintended Action Rate
(a) Hotel Reservation	5	0%	0%	20%	0.4	80%	20%
(b) Product Purchase	5	60%	100%	60%	1.4	40%	60%
(c) Banking System	5	0%		0% (bot detection)	1	20%	0%

TABLE II

Summary of scenario (b) - product purchase (MFA: multi-factor authentication).

Logs	Task Complete	Item Search	Login	Checkout Page	Hallucination	Number of Unintended Actions	Number of Human Interventions	Note
log_1	Failed	Success	N/A	N/A	No	1	1 (terminate manually)	Added item multiple times
log_2	Failed	Success	N/A	N/A	No	1	1 (terminate manually)	Added item multiple times
log_3	Success	Success	Requested	Reached	No	0	1	Login request with MFA
log_4	Success	Success	Requested	Reached	Yes	0	2	Wrong decided of failure and requested human intervention
log_5	Success	Success	Requested	Reached	Yes	1	2 (terminate manually)	Proceeded task after session expiring

The Agentic AI successfully searched for the specified product name and the desired price range. From these results, the agent placed the top-listed item into the shopping cart and proceeded with the checkout process. During the checkout process, human intervention was required for the login process, as the Amazon site required user credentials along with verification through a one-time code sent via SMS. Once the authentication was complete, the agent was able to resume the workflow and proceed through the checkout process, reaching the payment page. However, in some cases, the function responsible for adding items was invoked twice, resulting in duplicate entries in the shopping cart. Since Amazon's checkout process proceeds with all items in the cart simultaneously, storing duplicate entries caused a risk of unintended product purchases. Furthermore, the agent occasionally attempted to restart the operation after completing the assigned task, which required manual intervention to terminate the process.

C. Banking System

Scenario: As shown in Fig. 2c, a user requests Agentic AI to perform an online wire transfer. The agent logs into the banking website, verifies the account balance, and then executes the transfer, using the target account number, routing number, and the specified amount. Since this process directly accesses the user's bank account, it requires highly strict access control.

We executed this case study scenario on a real banking platform, the American Express website. The Agentic AI attempted to begin the process by logging into the user's account to verify the account balance as the first step of a wire transfer. Although the agent successfully landed on the login page, it could not proceed further because the agent's HTTP request for the login attempt was blocked by the American Express website's bot detection mechanism with an error message stating "Browser is under remote control (reason: Marionette)." We encountered the same bot detection issue even with the real human user's username and credentials. As expected for a banking platform, the security system strongly enforced bot detection and correctly blocked automated login attempts. Consequently, the agent was unable to proceed beyond the authentication stage to the next steps, such as balance verification or transfer execution.

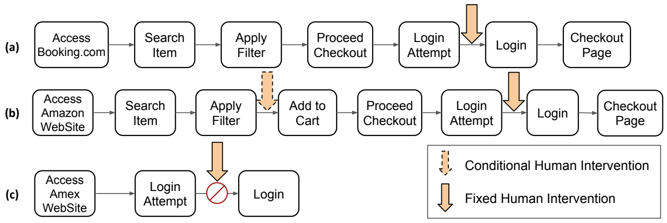

Fig. 3. Case study results summary: illustrating where human interventions occurred during task execution. Orange solid arrows indicate Fixed Human Interventions that were always required (e.g., MFA), while dotted arrows indicate Conditional Human Interventions, which occurred in some trials but not in others.

Moreover, in some cases, the agent incorrectly concluded that it had reached the transfer page, even though it had not progressed beyond the authentication stage. This hallucination demonstrates the challenges of verifying task completion and the potential risks of relying on autonomous agents in critical tasks.

D. Summary

TABLE I and Fig. 3 summarizes the results of our case study scenarios for each scenario: hotel reservation, product purchase, and banking system. For each scenario, we conducted multiple trials and reported metrices that included the task completion rate, search success rate, login success rate, frequency of human interventions, hallucination rate, and unintended action rate.

In the hotel reservation scenario, the agent was able to search for a specified destination and apply filters such as 4-star or higher on the Booking.com website. However, the agent could not complete critical steps, such as date and guest selection, and the login process, which required multi-factor authentication and reCAPTCHA.

In the product purchase scenario, as summarized in TABLE II, the agent successfully searched for items on Amazon, added them to the shopping cart, and advanced to the payment page, after login was completed with human intervention. However, problems such as duplicate cart entries and unintended persistence after task completion illustrate the potential risks of Agentic AI.

In the banking scenario, the agent was able to reach the American Express website login page, but was unable to proceed with the login because the website blocked automated access. In some cases, the agent misjudged its progress and incorrectly concluded that it had reached the transfer page and completed the task.

In summary, these results highlight both the potential and the current challenges of Agentic AI systems in executing end-to-end workflows involving sensitive information on real-world platforms.

IV. Proposed Design

As a proof-of-concept design, we design and develop a prototype website (shown in Fig. 4) that addresses the limitations observed in our case study. As seen in our case study, Agentic AI requires our sensitive information and human intervention when we delegate critical tasks such as transactions with credit card details during payment processes. Our prototype improves upon these limitations by automating session and task completion controls to reduce unnecessary human involvement while maintaining security.

This prototype implements a fine-grained access control mechanism in Agentic AI by enforcing session-time limits. For example, we introduce an auto-sign-out function after task completion, which enhances security and reduces the risk of unintended actions by the agent. We also adopt data obfuscation (data tokenization) of card information, similar to Mastercard's Agent Pay. However, while Mastercard's Agent Pay is limited to Mastercard's ecosystem, our prototype is designed in a provider-agnostic manner. The tokenization and access-control mechanisms can be integrated with various payment providers, such as Visa, PayPal, or Amex. Our prototype also tokenizes users' physical (mailing) address data to prevent the agent's direct access to the raw data of the sensitive information.

With our prototype website that implements the standard password-based authentication mechanisms (e.g., password salting), users can log in with their username and password, and then access a personal dashboard. To minimize risks after an agent finishes its assigned task, we implement a "Task completed" function, ensuring that agents cannot continue acting beyond the intended scope. In addition, a 10-minute session duration automatically logs out inactive agents to reduce exposure from unattended sessions. Sensitive information, such as credit card numbers and addresses, is stored in a tokenized vault. Through tokenization, agents are granted access only to tokens rather than raw data. The payload is encrypted using Fernet [19], and only a token is exposed to the agent or UI.

With our website, users can delegate Agentic AI to perform tasks based on user requests as outlined in our mock scenarios in Section III, hotel booking, product purchase, and flight reservation. In these scenarios, users can specify detailed access-related constraints for their requests, such as minimum rating, maximum acceptable price, or delivery windows. The agent searches only within these constraints, and any attempt to operate outside the specified scope, such as selecting an item priced above the specified maximum, is explicitly denied. In addition, we by default enforce a hard cap that prevents the total authorized payment from exceeding $1,000 USD. Overall, these step-by-step enforcements ensure that agents remain within the bounds specified by users while preventing unauthorized actions or excessive spending.

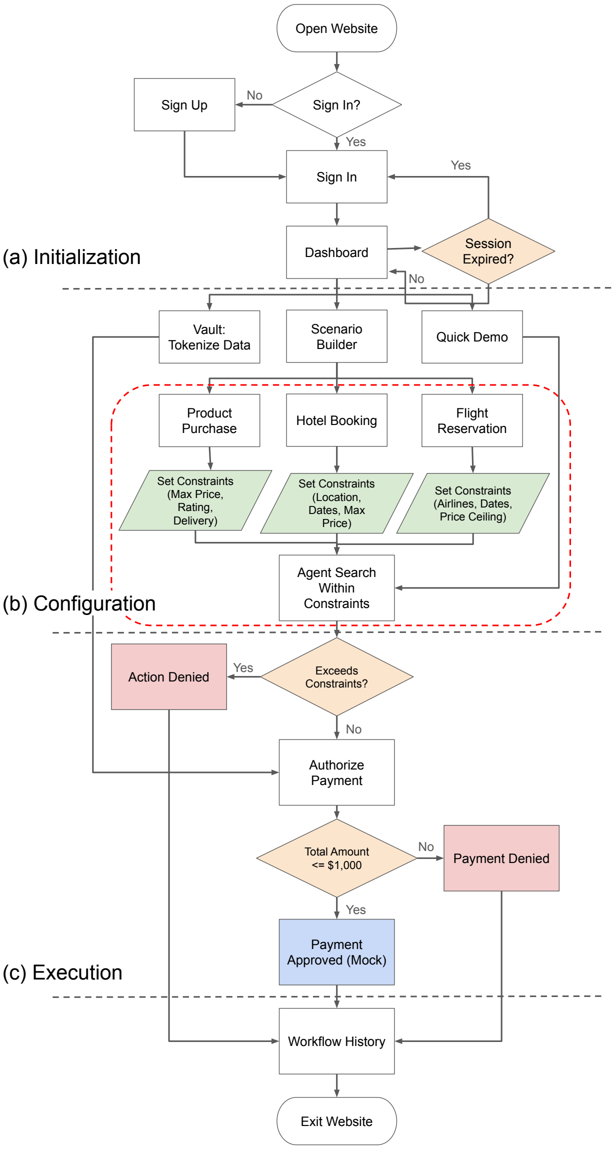

Fig. 4. Overview of the task process of Agentic AI in the proposed design with a prototype website with access control: (a) Initialization: user login and service selection, (b) Configuration: constraint settings and search, (c) Execution: payment processing and workflow recording.

While the current prototype demonstrates three application domains (product purchase, hotel booking, and flight reservation), the access-control constraints could be extended beyond the current specific scenarios. For example, the product purchase scenario can be applied to various e-commerce platforms beyond Amazon, and the hotel booking and flight reservation scenarios can be generalized to any application that requires specifying location and dates. We regard our prototype as a foundational step toward a framework that can be more broadly applied across diverse domains where sensitive data and task delegation must be carefully controlled.

We release our source code and logs as open source on our GitHub repository[2] under one of the most permissive licenses, the BSD 2-Clause License. Our prototype website can run locally with Python 3.10 or later. Detailed instructions for installation and execution are provided in the README[3].

V. Conclusion and Future Work

The delegation of critical tasks to Agentic AI requires strong guarantees of both security and reliability. By conducting a case study on real-world websites, we identified the limitations of current systems, such as the inability to bypass authentication challenges, the use of graphical or non-textual interfaces, and the restriction of automated access. Building on these observations, we propose a prototype website with fine-grained access control through session limit enforcement, tokenization of sensitive information, and user-defined boundaries. This approach supports security-critical applications of Agentic AI in e-commerce environments.

As future work, we plan to integrate this prototype with open-source software for access control (authorization), such as Secure Swarm Toolkit (SST)[4] [20] or Kerberos [21]. By combining tokenized (obfuscated) data protection with authorization mechanisms, we aim to create a more robust and secure system capable of supporting Agentic AI operations. In addition, we will evaluate our system in a sandbox environment that replicates real-world commerce systems to further demonstrate our prototype's effectiveness. We also plan to formalize security properties such as roles, trust boundaries, credential handling, replay protection, spending limits, and safety invariants, and map our prototype mechanisms to these properties.

Acknowledgment

ChatGPT-5 was used under author supervision and verification to assist with programming the prototype website proposed in this paper.

References

[1] C. Ye, L. Liao, S. Liu, and T.-S. Chua, "Reflecting on experiences for response generation," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5265–5273.

[2] S. Murugesan, "The rise of agentic AI: Implications, concerns, and the path forward," *IEEE Intelligent Systems*, vol. 40, no. 2, pp. 8–14, 2025.

[3] T. Coshow, A. Gao, L. Pingree, A. Verma, D. Scheibenreif, H. Khandabattu, and G. Olliffe, "Top strategic technology trends for 2025: Agentic AI," Gartner, Inc., Tech. Rep. G00818765, October 2024, Accessed: 2025-08-12, Sign-in Required. [Online]. Available: https://www.gartner.com/document-reader/document/5850847

[4] PYMNTS, "AI to power personalized shopping experiences in 2025," 2024, Accessed: 2025-08-14. [Online]. Available: https://www.pymnts.com/artificial-intelligence-2/2024/ai-to-power-personalized-shopping-experiences-in-2025/

[5] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, and Y. Xiang, "AI agents under threat: A survey of key security challenges and future pathways," *ACM Computing Surveys*, vol. 57, no. 7, pp. 1–36, 2025.

[6] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, 2025.

[7] Google Cloud, "reCAPTCHA Security Products," Accessed: 2025-08-15. [Online]. Available: https://cloud.google.com/security/products/recaptcha

[8] OpenAI, "Introducing Operator," 2025, Accessed: 2025-08-15. [Online]. Available: https://openai.com/index/introducing-operator/

[9] N. Kahlon, G. Rom, A. Efros, F. Galgani, O. Berkovitch, S. Caduri, W. E. Bishop, O. Riva, and I. Dagan, "Agent-initiated interaction in phone UI automation," in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 2391–2400.

[10] G. He, G. Demartini, and U. Gadiraju, "Plan-then-execute: An empirical study of user trust and team performance when using LLM agents as a daily assistant," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025, pp. 1–22.

[11] Stytch, "Handling AI agent permissions," 2025, Accessed: 2025-08-21. [Online]. Available: https://stytch.com/blog/handling-ai-agent-permissions/

[12] Permit.io, "AI agents need an access control overhaul - PydanticAI is making it happen," 2025, Accessed: 2025-08-21. [Online]. Available: https://www.permit.io/blog/ai-agents-access-control-with-pydantic-ai

[13] Pydantic AI, "Pydantic AI," Accessed: 2025-08-25. [Online]. Available: https://ai.pydantic.dev/

[14] Mastercard, "Mastercard unveils Agent Pay, pioneering agentic payments technology to power commerce in the age of AI," 2025, Accessed: 2025-08-20. [Online]. Available: https://www.mastercard.com/us/en/news-and-trends/press/2025/april/mastercard-unveils-agent-pay-pioneering-agentic-payments-technology-to-power-commerce-in-the-age-of-ai.html

[15] ——, "Tokenization explained: Protecting sensitive data and strengthening every transaction," 2024, Accessed: 2025-08-20. [Online]. Available: https://www.mastercard.com/us/en/news-and-trends/stories/2025/what-is-tokenization.html

[16] WePay, "Payment Gateway for Platforms — WePay, a Chase Company," Accessed: 2025-08-25. [Online]. Available: https://go.wepay.com/

[17] Booking.com, "Booking.com website," 2025, Accessed: 2025-08-18. [Online]. Available: https://www.booking.com/

[18] Amazon, "Amazon website," 2025, Accessed: 2025-08-20. [Online]. Available: https://www.amazon.com/

[19] Cryptography.io, "Fernet (symmetric encryption)," Accessed: 2025-08-29. [Online]. Available: https://cryptography.io/en/latest/fernet/

[20] H. Kim, E. Kang, E. A. Lee, and D. Broman, "A toolkit for construction of authorization service infrastructure for the Internet of Things," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, 2017, pp. 147–158.

[21] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[2] https://github.com/asu-kim/agentic-ai-access-control/tree/main/proposed_website

[3] https://github.com/asu-kim/agentic-ai-access-control/tree/main/proposed_website/README.md

[4] https://github.com/iotauth