A Taxonomy of Multi-Layered Runtime Guardrails for Designing Foundation Model-Based Agents: Swiss Cheese Model for AI Safety by Design

Md Shamsujjoha, Qinghua Lu, Dehai Zhao, Liming Zhu
CSIRO's Data61, Australia
Email: {md.shamsujjoha, qinghua.lu, dehai.zhao, liming.zhu}@data61.csiro.au

arXiv:2408.02205v2 [cs.SE] 7 Nov 2024

***Abstract*—Foundation Model (FM)-based agents are revolutionizing application development across various domains. However, their rapidly growing capabilities and autonomy have raised significant concerns about AI safety. Designing effective guardrails for these agents is challenging due to their autonomous and non-deterministic behavior, and the involvement of multiple artifacts — such as goals, prompts, plans, tools, knowledge bases, and intermediate and final results. Addressing these unique challenges runtime requires multi-layered guardrails that operate effectively at various levels of the agent architecture, similar to the Swiss Cheese Model. In this paper, we present a taxonomy of multi-layered runtime guardrails to classify and compare their characteristics and design options, grounded on a systematic literature review and guided by the Swiss Cheese Model. This taxonomy is organized into external and internal quality attributes and design options categories. We also highlight the relationships between guardrails, the associated risks they mitigate, and the quality attributes they impact in agent architectures. Thus, the proposed taxonomy provides structured and concrete guidance for making architectural design decisions when implementing multi-layered guardrails while emphasizing the trade-offs inherent in these decisions.**

***Index Terms*—Foundation Model, Large Language Models (LLM), Agent, Guardrails, Swiss Cheese Model, Responsible AI, AI Safety, Software Architecture, Taxonomy**

I. Introduction

A foundation model (FM) is a large-scale machine learning model pre-trained on massive amounts of data using self-supervision at scale. These models are designed to be highly versatile and can adapt to a wide range of downstream tasks. The term 'foundation' reflects their role as the fundamental base upon which many specialized models are built [1]. FMs possess emergent capabilities, meaning their behaviors and susceptibilities arise implicitly from the training data rather than being explicitly programmed. This allows them to perform tasks that were not originally anticipated during their initial training [2]. An FM-based agent utilizes an FM as a core component, interacting with other AI or non-AI components to perform tasks autonomously [3]. In recent years, FM-based agents have experienced extensive growth [4].

The autonomy of FM-based agents along with their enhance decision-making and task execution capabilities introduce significant concerns about responsible AI and AI safety, e.g., generating harmful or offensive content, producing dangerous or unintended outcomes, exhibiting discriminatory behavior, compromising user privacy, spreading misinformation, facilitating cyberattacks, etc [1, 6, 8]. To address these challenges, effective runtime guardrails are required to ensure the agent's behavior is responsible and safe. Multi-layered runtime guardrails structured similarly to the Swiss Cheese Model that operate at various levels of the agent architecture further mitigate associated risks [1, 76].

There have been some initial efforts on guardrails, such as input filtering [1, 9], output modification [10, 11], adaptive fail-safes that can prevent harmful outputs [12, 13], real-time monitoring and detection [14–17], continuous output validation to ensure adherence to ethical standards [18–20] etc. However, the existing runtime guardrails mainly focus on inputs and outputs of foundation models, which do not capture the complexity of FM-based agents. FM-based agents are complex AI systems [21] that integrate foundation models and non-AI components such as data retrievers and external tools. Guardrails need to be placed across different components, taking into account the operational and environmental context, much like the layers in the Swiss cheese model [75]. However, most current work primarily addresses the functional correctness of guardrails, and quality attributes such as customizability and interpretability are often not considered.

Therefore, in this paper, we present a taxonomy for multilayer guardrails from a software architecture perspective. The taxonomy classifies and compares the characteristics and design options of guardrails. There are three main categories in the taxonomy: the external qualities for adopting multi-layered runtime guardrails, the internal attributes that should be addressed in the guardrails design, and the design options available. We developed this taxonomy based on the results of a systematic literature review (SLR), guided by the Swiss Cheese Model structure for AI safety. The main contributions of this paper are as follows:

- ❖ The taxonomy offers a comprehensive comparative analysis of different design options and provides structured and concrete guidance for making architectural design decisions about multi-layered runtime guardrails to achieve AI-safety-by-design. The design options are classified into actions, targets, scopes, rules, autonomy, modalities, and underlying techniques.

- ❖ This taxonomy helps researchers and practitioners understand the underlying rationales and benefits by summarizing the external qualities for multi-layered runtime guardrails. Hence, it also promotes informed decision-making when integrating guardrails into FM-based agents.
- ❖ The taxonomy identify critical internal quality attributes to ensure that the multi-layered runtime guardrails not only perform their intended functions but also enhance the overall quality of the guardrails' design.

The rest of the paper is organized as follows: Section II discusses background and related works. Section III introduces the methodology of this study. The taxonomy of guardrails is presented in Section IV answering three key research questions. Section V presents the implications of our proposed work and summarizes primary threats to this study. Finally, Section VI concludes the paper and outlines the future work.

II. Background and Related Work

Foundation models (FMs) have significantly impacted the architecture of AI agents [1, 20]. Although guardrails for FM-based agents have been explored in various contexts, this paper is the first study assessing existing work to provide a comprehensive understanding. Below, we present key background and related works required to understand guardrails and the research gap.

A. Background

The terms foundation models (FM) and large language models (LLMs) are often used interchangeably in the literature. However, they are not the same [22, 23]. Foundation models are large-scale machine learning models pre-trained on massive amounts of diverse datasets (such as text, images, videos) using self-supervised learning techniques [6]. They can perform a wide range of tasks across various application domains, including multimodal tasks. LLMs are a subset of FMs focused on language-related tasks and analyze data to perform natural language processing (NLP) tasks [22, 24]. Both FMs and LLMs are pre-trained on large datasets [25] and adapted for specific tasks using techniques like fine-tuning, in-context learning, distillation, and Retrieval-Augmented Generation (RAG) to improve response quality [24].

FM-based agents incorporate FMs as core components within larger architectures designed to utilize their capabilities [3]. These agents integrate FMs with additional components such as data retrievers (e.g., RAG), external tools and others. The complexity of FM-based agents lies in ensuring seamless interoperability among these components while ensuring responsible AI and AI safety [1].

In 2021, Bommasani et al.[1] provided a comprehensive discussion on foundation models, illustrating the essential elements and relationships between them. The authors also discussed the practical applications of FM-based agents, highlighting their economic and operational advantages over traditional systems. Two years later, Zhou et al. covered recent research advancements, challenges, and opportunities for pre-trained foundation models in text, image, graph, and other data modalities in their survey paper[20]. Both works offer excellent insights into future research directions to address open problems and associated risks. Studies [1, 26] mention that foundation model-based agents often inherit responsible AI and AI safety issues and can cause negative consequences.

Lu et al. developed a taxonomy of FM-based systems (agents) focusing on their pretraining, adaptation, architectural design, and responsible-AI-by-design [27]. The taxonomy aids software architects and developers in evaluating and integrating foundation models into complex systems. The authors also highlighted considerations for cost, accuracy, and responsible AI related quality attributes for the systems. The authors later proposed a reference architecture for designing responsible and safe foundation model based systems (agents) in [3]. The works [28, 29] explore the risks associated with deploying LLMs and evaluate current approaches to mitigate such risks through model alignment. Recently, Lu et al. discussed that responsible AI practices are highly recommended for FM-based systems (agents) [2, 30]. They also emphasized that FM-based agents must ensure they are developed and deployed in a responsible, safe, and legal way [31–34].

B. Related Work

There exist several frameworks and tools for designing guardrails at design time and runtime. Significant effort have been put into model alignment during design time, which focuses on aligning the FM's outputs with human values and ethical standards. Pre-training and adaptation strategies play a significant role in mitigating risks in FM-based agents through guardrails, especially multilayer guardrails. Pre-training models on diverse and ethically sourced datasets helps align them more closely with ethical standards. Continuous adaptation through fine-tuning further reduces AI risks [1]. Ensuring ethical and responsible AI practices involves not only technical solutions but also adherence to regulatory and ethical guidelines within the systems. The adoption of comprehensive AI frameworks that incorporate ethical principles into the design and deployment of FMs is crucial for maintaining public trust and safety [31].

Some initial efforts have been made toward runtime guardrails. Zhou et al. [20] present an excellent discussion of the literature that addresses the monitoring and controlling behavior of FM-based agents. Various tools and frameworks have also been developed to tackle these challenges. NeMo Guardrails[16] provides programmable guardrails to ensure that models operate within safe parameters by monitoring inputs and outputs in real-time, dynamically adjusting behavior to prevent harmful outcomes[27]. OpenAI's Moderation API [35] monitors and filters harmful content generated by models to protect user interactions. The GuardAgent framework [36] utilizes an LLM agent to oversee and safeguard other agents. It demonstrates strong generalization and low operational overhead by dynamically generating guardrail code.

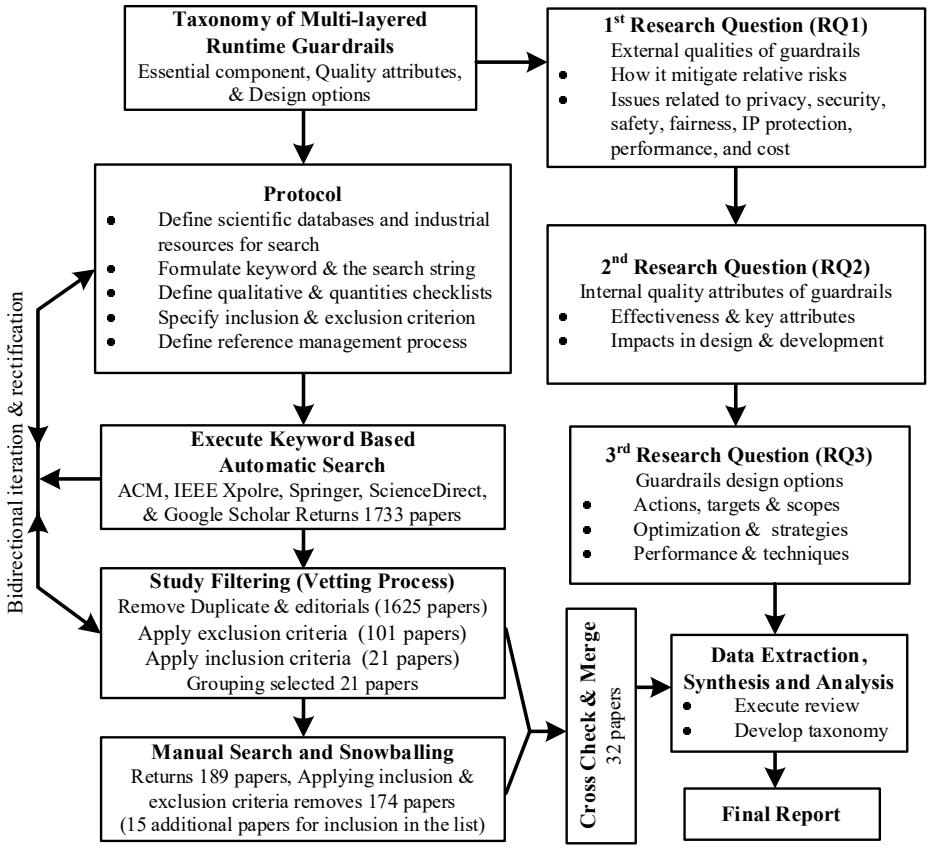

Figure 1. Methodology

We also found that the continuous validation ensures that the outputs from FM-based agents adhere to predefined ethical standards and guidelines. Techniques such as auditing models [18] through multi-layered approaches [18, 37] systematically check for biases. They also ensure ethical compliance during the agent's operation [3]. Dynamic access controls adjust access permissions in real time based on the context of data usage to protect sensitive information and ensure it is accessible under appropriate circumstances [38]. Adaptive fail-safes as characteristics of guardrails intervene automatically when an FM exhibits potentially harmful behavior within agents. These fail-safes are designed to modify or halt outputs that could lead to undesirable consequences [31]. However, the design of multi-layered runtime guardrails for FM-based agents has not been systematically explored. Thus, we develop a taxonomy to categorize existing work into a compact framework and compare and evaluate different design options for multi-layered runtime guardrails.

III. Methodology

This study focuses on two primary concepts: (i) foundation model-based agents and (ii) runtime guardrails. We adopted the Petticrew and Roberts approach [39] to define the **P**opulation, **I**nterventions, **C**omparison, **O**utcomes, and **C**ontext (PICOC), within which the intervention in this study is delivered. The PICOC for this study is shown in Table I. Using these PICOC components and following Kitchenham's guidelines [40], we develop the protocol for this study. The following subsections detail the steps and processes undertaken in this study.

Table I
PICOC for this Study

Population	Studies and researches focus on runtime guardrails within foundation model-based agents.
Intervention	Development, optimization, and evaluation of multilayer runtime guardrails in foundation model-based agents, focusing on key quality attributes and design strategies inspired by the Swiss Cheese Model.
Comparison	Comparative analysis between different approaches to design multilayer runtime guardrails in FM-based agents.
Outcomes	Qualities and design options for multilayer runtime guardrails for foundation model-based agents guided by Swiss cheese model.
Context	**Include:** Empirical and theoretical studies on the components, design and evaluation of guardrails in foundation model-based agents. **Exclude:** Studies beyond the scope of foundation model based agents, non-English literature, and those not considering guardrails.

Table II
CONSOLIDATED CONCEPTS AND SEARCH TERMS

Main Terms	Supportive Search Terms
Concept 1 (Co1): Foundation Model based agents	Foundation Models, Foundation Model based agents, Large Language Model, Generative AI, Artificial General Intelligence, Transformer Models, Self-supervised Learning, Pretrained Models, Language Models, Conversational AI.
Concept 2 (Co2): Runtime Guardrails	Guardrails, guardian, responsible AI, safe, risk, trustworthy, protect, detect, monitor, verify, validate, evaluate, benchmark, design.

Table III
INCLUSION CRITERIA

ID	Detail Criterion
IC_1	Full text of conference papers, journal articles, industry reports, and book chapters that are relevant to the defined main concepts: Foundation model based agents and guardrails.
IC_2	Papers written in English that include references.
IC_3	Studies that specifically address the design and development of guardrails in foundation model-based agents. This includes theoretical frameworks, empirical research and case studies.
IC_4	Papers available in an electronic format, such as PDF, DOC, DOCX, HTML, and PS etc.

Table IV
EXCLUSION CRITERIA

ID	Detail Criterion
EC_1	Work-in-progress proposals, keynote addresses, secondary studies, and vision papers without concrete relation to guardrails.
EC_2	Discussion papers and opinion pieces that do not provide empirical evidence or concrete solutions related to guardrails in foundation model-based agents.
EC_3	Short communications less than two pages, and studies that do not offer substantial information for analysis.
EC_4	Studies focusing solely on AI or similar technologies without direct relevance to guardrails.
EC_5	Research lacking a clear connection to the design and development of guardrails in the context of foundation models.
EC_6	Duplicate publications or earlier versions of studies that have been superseded by extended journal versions.
EC_7	Non-original research, commentary, editorial pieces, and non-empirical discussions papers.
EC_8	Studies inaccessible due to copyright or database restrictions.

A. Research Scope and Protocol Development

The high-level research approach for this study is shown in Figure 1. Initially, we determined the research scope and developed a protocol following Kitchenham's guidelines [40, 41]. The protocol guided the entire study by defining relevant scientific databases and resources, formulating keywords and search strings, outlining qualitative and quantitative checklists, and specifying criteria for the inclusion and exclusion of studies.

B. Research Questions

When formulating our **R**esearch **Q**uestions (RQ), we wanted to ensure that they were broad enough to capture the diverse aspects of runtime guardrails while being specific enough to provide actionable insights. We captured these diverse aspects through the following three Research Questions (RQs).

RQ1: External qualities for runtime multilayer guardrails in FM-based agents?
Our first research question investigates the fundamental external qualities for designing multilayer runtime guardrails in FM-based agents. It explores the responsible AI and AI safety concerns that require the adoption of runtime guardrails to prevent harmful content and unintended behaviors.

RQ2: What are the key internal quality attributes for designing multilayer runtime guardrails?
Our second research question identifies quality attributes that should be considered when designing effective guardrails in FM-based agents.

RQ3: What are the design options for runtime guardrails?
This question examines the design options for runtime guardrails in FM-based agents from different perspectives, such as action and scope, guided by the Swiss Cheese Model.

C. Search String Formulation

Relevant primary studies for this SLR were identified based on the RQs defined in Section III-B. With the assistance of the PICOC approach (shown in Table I), our search terms were divided into two primary concepts, as shown in Table II. These concepts helped us to set a well-formulated search string.

We also used synonyms, abbreviations, and alternative spellings of search terms to increase the number of relevant research papers. We used truncation and wildcard operators to save time and effort in finding these alternative keywords. Moreover, different supplementary key terms or phrases discovered during search iterations were added to our search string to enhance our search strategy. Our supposition is that they will collect all relevant articles that contains guardrails for FM-based agents. When constructing the final search query, the identified keywords, their alternatives and related terms were linked with Boolean AND (&&), OR (‖) and NOT (¬) operators as follows as follows:

$$[\{(C_{11}\|C_{12}\|...\|C_{1n})\textbf{AND}(C_{21}\|C_{22}\|...\|C_{2n}) \textbf{NOT}(UC_1\|UC_2\|...\|UC_n)] \quad (1)$$

where $C_{11...1n}$, and $C_{21...2n}$ ε Co1 and Co2 of Table II, respectively; and $UC_1 \ldots UC_n$ refers the **Exclude Context** defined earlier in PICOC table (Table I).

D. Selection of Papers: Inclusion and Exclusion Criterion

Table III and Table IV present the Inclusion Criteria (IC) and Exclusion Criteria (EC) that have been used to identify the studies for this SLR, respectively. We found that a considerable amount of work on guardrails exists in gray literature; however, we excluded them as they often lack peer review and a rigorous validation process. While some sources [42] argue that gray literature is an important resource for systematic literature reviews (SLRs), such literature can be misleading and introduce biases and inconsistencies in the review process [43]. We prioritized peer-reviewed sources in

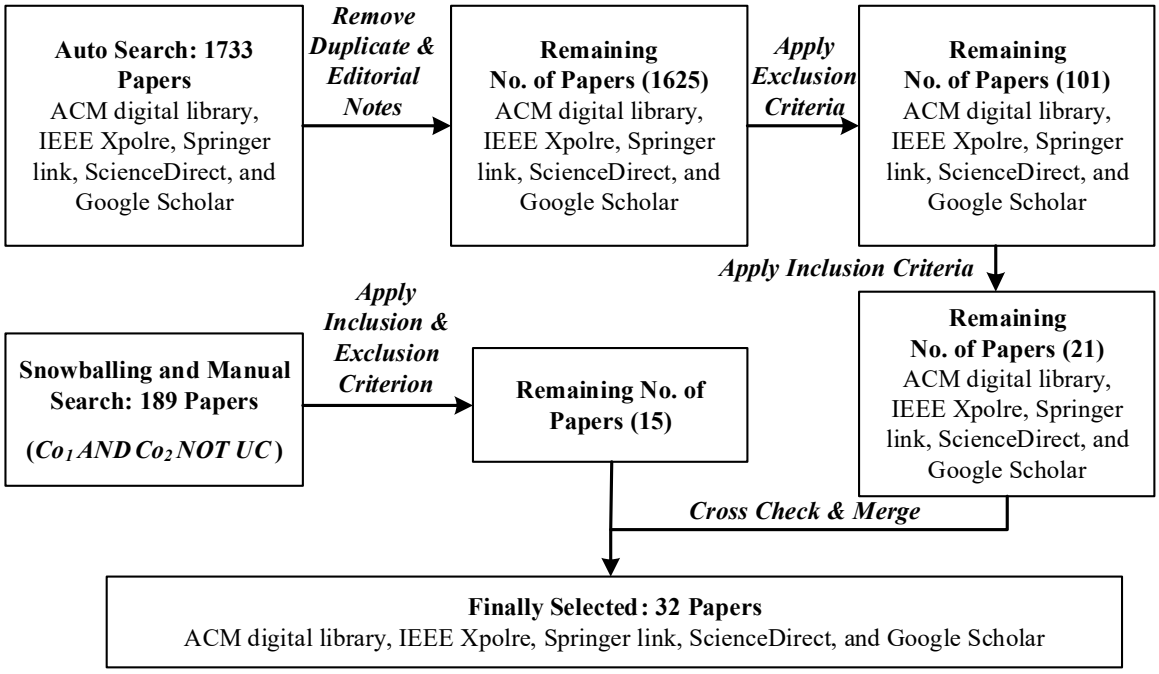

Figure 2. Study Selection Process for this SLR

this study to ensure scientific reliability and credibility, as per Kitchenham et al. guidelines [40, 41].

E. Study Search and Filtering Process

Our filtration process is further detailed in Figure 2. Initially we ran the formatted query on four major databases that returned 1,733 research papers. We then applied filtering and classified the studies found according to the guidelines presented in [40, 41]. In our initial filtration process, we removed 108 papers due to being duplicated articles, editorial or key notes. After reading the title, abstract, conclusion and skimming through the introduction, methodology and results, we applied our exclusion criterion defined in Table IV, and 1524 further papers were removed. During the third step of filtration, we applied inclusion criteria and removed 80 papers as these studies did not meet ICs shown in Table IV. In parallel, we did a manual search and found 189 papers that meet our key concepts defined in Table II but not contain any unwanted content (UC). After applying ICs and ECs, 15 out of 189 papers were selected. Finally, we did a cross-check and ended up with 32 papers.

F. Data Extraction and Quality Assessment

We used a semi-automated process [44] for data extraction from the selected studies to answer our RQs. Key qualitative information extracted from each selected study includes guardrails definitions, motivations, reported key quality attributes, and design options. We also extracted several relevant pieces of information to understand the context and considerations in designing and evaluating runtime guardrails.

We then evaluated each study based on the following five **Q**uality **A**ssessment **C**riteria (QAC) on a scale from 1 (Very Poor) to 5 (Excellent). If a study's average score was less than 2, it was excluded from further analysis. Otherwise, we used the qualitative information to decide this. The QAC for this study are as follows:

- ❖ Relevance to guardrails for FM-based agents.
- ❖ Clear methodology for guardrail design.
- ❖ Adequate data collection, analysis, and evaluation of guardrail effectiveness at different layers of the agent architecture.
- ❖ Discussion of challenges in designing guardrails for autonomous and non-deterministic behaviors in agents.
- ❖ Practical applicability of findings for guardrails in FM-based agents.

IV. The Proposed Taxonomy

In this section, we present our taxonomy of multilayer runtime guardrails for FM-based agents. As illustrated in Figure 3, this taxonomy is organized into three main categories: external quality attributes, internal quality attributes, and design options, which we use in the following sub-sections to answer our key research questions guided by the Swiss Cheese Model for AI Safety by Design.

A. External Qualities for Multi-layered Runtime Guardrails (RQ1)

Our first research question investigates the fundamental reasons for integrating guardrails into FM-based agents. We found that guardrails are primarily implemented to mitigate bias, harmful content, and unintended behaviors. Additionally, guardrails better maintain ethical and safety standards in

areas such as accuracy, privacy, security, safety, fairness, cost, performance and IP protection, as discussed below.

1) ***Accuracy****:* Accuracy in FM-based agents is a critical concern, particularly regarding the generation of hallucinations, misinformation, and disinformation [45]. Hallucinations occur when models generate information that is factually incorrect or fabricated. Such inaccuracies can mislead users and damage the credibility of the systems [10]. Misinformation refers to the unintentional spread of false information, while disinformation involves the deliberate dissemination of falsehoods to deceive users [2]. These phenomena pose significant risks, as they can propagate false narratives and influence public opinion [20]. Guardrails mitigate these risks by detecting and filtering potentially inaccurate information [16]. For example, OpenAI uses guardrails to clearly label AI-generated content to prevent deepfakes and misinformation [46], such as in one case where it was reported to prevent misleading voters in the upcoming US elections [47, 48]. Therefore, guardrails improve the overall integrity of the systems [1, 46].

2) ***Privacy****:* In FM-based agents, privacy is a critical concern due to the risks of handling large amounts of personal and sensitive data. Without proper guardrails, these systems can inadvertently expose or misuse this data [1, 12]. One key privacy risk is data leakage, where sensitive information is inadvertently revealed through model outputs. For instance, an FM-based LLM trained on private conversations might accidentally generate text that includes personal information such as names, addresses, or financial details [49]. This leakage can occur through direct responses or statistical inferences. For example, an FM-based customer service chatbot might inadvertently include sensitive information. In April-May 2023, a notable incident involved Samsung employees leaking proprietary information into ChatGPT, leading to Samsung banning ChatGPT [50].

3) ***Security****:* Security in FM-based agents involves protecting them from malicious activities that could compromise their integrity and functionality [6, 14, 19]. For example, an FM-based system could be targeted by hackers to manipulate data, producing incorrect or harmful outputs that affect decision-making processes [51]. An incident similar to this reported in [52], where malicious users manipulated Microsoft's Tay chatbot to produce inappropriate and offensive content, resulting in its shutdown. Additionally, FM-based agents are vulnerable to various threats [53]. Hackers can exploit vulnerabilities to access sensitive data, leading to breaches of confidentiality. Even with authorized access, there is a risk of data misuse by third-party providers [54]. FM-based agents are also prone to adversarial attacks, where specially designed queries extract sensitive information.

Guardrails mitigate these risks by detecting and responding to real-time threats, safeguarding system integrity and performance [1, 51]. They also prevent unauthorized access and the exploitation of vulnerabilities. For example, a financial FM-based system with guardrails can detect unusual transaction patterns and initiate defensive measures, such as temporarily suspending transactions [51]. Thus, guardrails better maintain confidentiality [49, 55, 56], system integrity [10, 19], and availability [11, 18, 26, 53, 57], prevent model misuse [26, 54].

4) ***Safety****:* FM-based agents face significant safety issues, particularly in generating harmful or misleading outputs. These issues can arise when models produce content that is inappropriate, offensive, or incorrect [1, 3]. For example, in applications where FM-based agents handle critical data, such as in medical diagnosis or or self-driving cars, incorrect outputs can lead to dangerous consequences [32]. Additionally, there is a risk of generating ethically or morally questionable content, which can damage the credibility and acceptance of the system [53].

Guardrails mitigate these risks by detecting and filtering out harmful output in real-time. For instance, in an FM-based virtual assistant used for mental health support, guardrails ensure the responses are benign and non-triggering. They also effectively detect and correct errors in real time, thus mitigating the effects of adversarial attacks [13, 15, 58–60].

5) ***Fairness****:* FM-based agents can face bias and discrimination in model outputs. These biases can emerge from the training data, model algorithms, or deployment context [2, 61]. For instance, a language model used in recruitment for initial screening of CVs and responses to selection criteria might inadvertently favor candidates from certain demographics, cultures, and languages [8, 26], which ultimately undermines the credibility of the systems. Guardrails address these fairness issues and ensure equitable outcomes through model parameter adjustments. For instance, in our recruitment example, they can analyze the model's decisions and correct potential biases. Furthermore, guardrails promote transparency and accountability by providing insights into how decisions are made [1, 18], and avoid harmful stereotypes [53].

6) ***IP Protection****:* In FM-based agents, another key problem is the unauthorized use of generated content, where users might exploit outputs created by these systems without proper attribution or licensing, making them vulnerable to unauthorized use, duplication, or distribution [10, 26, 55]. Guardrails help mitigate these problems and better protect content's copyright [49]. They also aid in compliance with copyright laws and licensing agreements to mitigate legal risks [12, 19]. Techniques such as watermarking, fingerprinting, and labeling ensure the copyright and ownership of data [1, 10].

7) ***Cost****:* FM-based agents can incur significant costs due to errors, inefficiencies, or non-compliance with regulations. Without proper guardrails, agents might generate outputs that lead to financial losses, legal penalties, or damage to reputation [12, 26]. For example, an agent that provides incorrect financial advice could result in monetary losses for users and potential lawsuits against the provider. Implementing guardrails helps mitigate these risks by preventing erroneous or harmful outputs, thereby reducing potential costs associated with rectifying mistakes, handling legal issues, or rebuilding trust [1, 62]. Additionally, guardrails can optimize resource utilization by preventing unnecessary computations or actions, leading to cost savings in system operations [53].

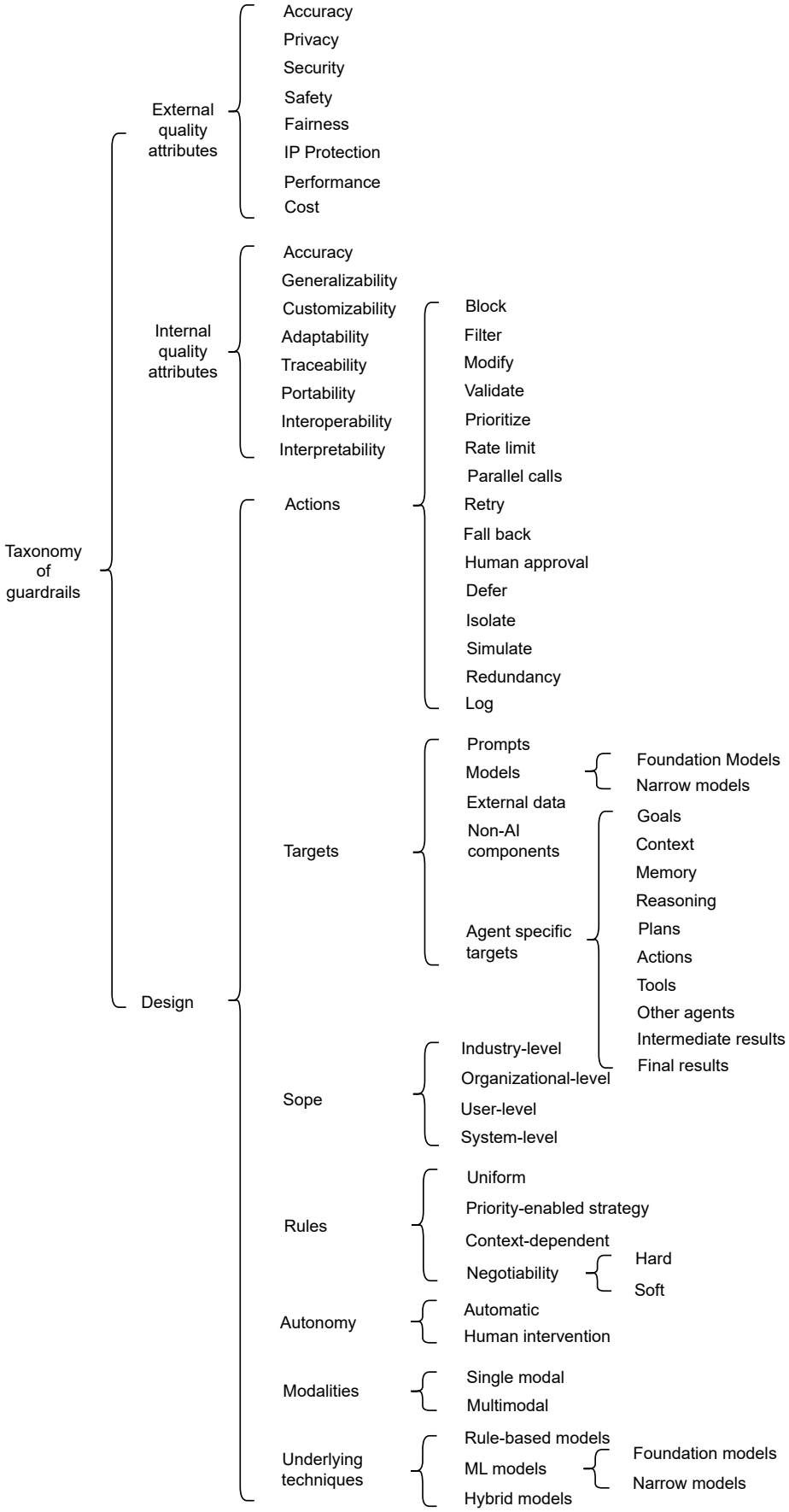

Figure 3. Taxonomy of Multi-Layered Runtime Guardrails for FM-Based Agents

*8) **Performance**:* Performance is a critical concern in FM-based agents, as they are expected to provide timely and efficient responses [24]. Without appropriate guardrails, agents might engage in resource-intensive computations that degrade system performance or lead to unacceptable response times [54]. Guardrails help manage performance by monitoring and controlling the computational resources used by the agent [53, 64]. For instance, they can limit the complexity of tasks the agent undertakes or prevent it from entering endless loops, ensuring consistent and efficient operation [49]. Moreover, guardrails can enhance performance by filtering out irrelevant or low-quality inputs, allowing the agent to focus on processing meaningful data [1, 16].

B. Internal Quality Attributes for Multilayer Runtime Guardrails (RQ2)

We identified eight key quality attributes essential for the design of runtime guardrails in FM-based agents. Below, we discuss these attributes in detail, highlighting their roles and importance.

*1) **Accuracy**:* Guardrail accuracy refers to the precision in identifying and mitigating risks, i.e., preventing undesired behaviors or outputs [45]. Functional accuracy emphasizes how well guardrails perform their intended tasks to meet user expectations and requirements. For instance, a guardrail used for a customer service chatbot must accurately block sensitive information from being disclosed. If a customer asks, '*What is my current account balance?*' the guardrail must prevent disclosing the balance, instead providing a safe response like, '*For security reasons, please check your balance through our secure app*.' If it fails to do so (false negatives) or incorrectly flags benign information (false positives), it compromises functional accuracy. Predictive accuracy involves the chatbot's correct understanding and response to inquiries. We found that the accuracy of guardrails is evaluated using automated methods, primarily considering the systems' performance [16], output quality [55], overall trustworthiness [10, 49], and system adaptability [26, 49].

*2) **Generalizability**:* Generalizability in guardrails refers to their ability to function effectively across diverse applications [63]. Such guardrails ensure that the protective measures are not overly specific to a single use case but can adapt to various contexts and still perform reliably. For example, a chatbot must comply with different financial regulations across countries to ensure compliance without reconfiguration. Dong et al.[15] and Wang et al.[64] emphasize the need for guardrails that can extend their applicability to new domains without significant reconfiguration or loss of performance, even during unexpected inputs or data types. Furthermore, generalizability enhances the ability to handle diverse linguistic, cultural, and operational contexts to provide robust protection and to enhance the system's resilience and reliability [1, 12].

*3) **Customizability**:* Customizable guardrails provide tailored protection that meets specific requirements and supports diverse operational needs in FM-based agents [1, 65]. They allow for adjustments and configurations that align with particular operational goals, data characteristics, and regulatory environments. For example, a customer service chatbot can enable priorities for different guardrails and adjust data handling based on the user's location, ensuring compliance with GDPR in Europe and CCPA in the U.S. [66]. Customizability also includes the capability to integrate user-defined rules and policies, ensuring that the guardrails can support diverse operational contexts and evolving needs [64].

*4) **Adaptability**:* Adaptability in guardrails is known as their capability to adjust and remain effective under varying conditions and data landscapes as context evolves [24, 26]. This attribute ensures robust and continuous protection by dynamically responding to changes in input data, usage patterns, and emerging threats without manual reconfiguration [15]. For example, a customer service chatbot can automatically update its guardrails to detect and block new offensive terms during interactions. This includes the ability to incorporate new knowledge and advancements in threat detection techniques [1, 54].

*5) **Traceability**:* The traceability attribute of guardrails tracks and records the origins, processes, and decision paths, such as input and output of FMs, external tools, etc. [27]. It involves maintaining detailed logs and records that can be audited to understand how decisions are made. For example, in a customer service chatbot, traceability ensures that every recommendation can be traced back to the data sources and algorithms used. This includes logging which data inputs influenced the response and what external APIs or tools were accessed, providing a clear audit trail for transparency and accountability. Traceability also aids in identifying the root causes of issues. This enables timely and accurate troubleshooting and improvement [26], and helps in maintaining user trust and meeting regulatory requirements [10, 16]. Additionally, comprehensive documentation of data sources and model modifications better support effective auditing and compliance checking [12].

*6) **Portability**:* Portability in guardrails for FM-based agents refers to the ability of these protective measures to be easily adapted and applied across different foundation model (FM) systems and platforms [27]. This includes ensuring that the guardrails function consistently across various FM architectures and environments, thereby maintaining their effectiveness and integrity regardless of the underlying system [26]. For example, the same guardrail can be applied for content moderation in both a customer service chatbot and a social media platform, regardless of their underlying technology. The benefits of designing portable guardrails include compatibility across multiple programming languages and frameworks facilitate their integration into diverse technological stacks [49]. These capabilities ensure that the guardrails remain effective and operational as the system evolves or migrates to new environments. Portable guardrails also support seamless updates and improve scalability to maintain high standards of security and compliance while adapting to new technological advancements within systems [16].

*7) **Interoperability**:* Interoperable guardrails work seamlessly across different systems and technologies [27]. They ensure that security, privacy, and compliance protocols can be applied consistently, even in heterogeneous environments that utilize varied software and hardware components, or diverse technological ecosystems [16, 67]. Guardrails that interface with various APIs and data formats also enable smooth communication and operation across different systems [26]. For example, they enable a customer service chatbot and internal support system to share data securely and consistently. This promotes cohesive and unified security management, reducing the complexity of maintaining multiple disparate protective measures [1], and better support collaborative efforts and data sharing [49].

*8) **Interpretability**:* Interpretability refers to the clarity and transparency with which guardrails and protective measures operate. This allows users and stakeholders to understand how decisions are made and actions are taken by models. Thus increasing trust and accountability [10, 68]. For example, a chatbot in healthcare, can explain why certain advice is given or restricted. Transparent guardrails better facilitate auditing and compliance [18]. They also help users to understand that actions taken by guardrails can be clearly understood and verified [55]. This is essential for identifying and correcting errors, as well as for ensuring that the system's operations align with standards.

C. Guardrails Design Options (RQ3)

This section presents a structured taxonomy for designing multi-layered runtime guardrails for FM-based agents, focusing on identifying various design alternatives.

*1) **Guardrail Actions**:* Guardrail actions are crucial for addressing the specific needs of FM-based agents. We have identified the following guardrail actions as key elements for FM-based agents:

- **Block:** The *block* action prevents specific inputs (such as user prompts) or outputs (such as content generated by FMs) from being processed or sent by various components (such as FMs and tools) in FM-based agents [54]. For example, the *block* action can reject the user prompts containing harmful instructions, thus preventing undesired outcomes.
- **Filter:** The *filter* action involves scanning and removing undesired or irrelevant content from the inputs or outputs of different components in FM-based agents [69, 70]. For instance, a filter may remove any personal data contained in the user prompts or the output generated by FMs.
- **Flag:** The *flag* action is used to mark specific inputs, outputs, or operations within FM-based agents [16]. For example, unusual transactions requested by the FM-based agent can be flagged for human review to ensure they comply with organizational policies [1, 30].
- **Modify:** The *modify* action allows for the adjustment of inputs or outputs of various components in the FM-based agents to meet specific requirements or standards [9]. For example, the system can modify the user prompts by adding more context and examples, making it easier for the FM to accurately interpret the user's intentions and provide more relevant responses.
- **Validate:** The *validate* action checks inputs, outputs, and intermediate results against predefined criteria to ensure they meet specified requirements or standards [26, 70]. For example, the customer service chatbot could validate if the output generated by the FM-based system contains any sensitive company information.
- **Prioritize:** The *prioritize* action enables the FM-based system to allocate resources and attention based on the importance of specific tasks, e.g., processing urgent user queries first [28].
- **Rate limit:** The *rate limit* action controls the frequency and volume of requests or outputs processed by the system/component within a given time frame [16, 53]. For example, a rate limit can be set to restrict the number of API calls a single user can make.
- **Parallel calls:** The *parallel calls* action can send multiple requests to the system/component to improve responsiveness, e.g., a user can send a prompt to the system multiple times at the same time and select the better response [16, 53].
- **Retry:** The *retry* action involves attempting a request again after an initial failure or unsatisfactory result [13].
- **Fall back:** When the system is unable to handle a request, the *fall back* action can redirect to the previous state or an alternative solution [13, 16, 71].
- **Human intervention:** The *human intervention* action requires humans to review and approve specific outputs or decisions [16, 53, 55]. For example, responses involving sensitive medical advice might be flagged for human approval before being communicated to users.
- **Defer:** The *defer* action postpones the processing of a request or task until specific conditions are met or additional information is available [72].
- **Isolate:** The *isolate* action involves segregating a specific entity (e.g., user) or component to prevent interaction with the system [19, 57, 60]. For example, a system might isolate a compromised narrow AI model suspected of being poisoned with malicious data in a sandbox environment, preventing potential harm to the main system.
- **Simulate:** The *simulate* action involves running tests in a controlled environment to predict and analyze potential outcomes [1]. For instance, a system might simulate different configurations to determine most effective plan.
- **Redundancy:** The *redundancy* action involves implementing backup processes or components to ensure continuity and reliability in case of failures [16, 26]. For example, an agent can implement two similar workflows in parallel, so if one workflow encounters an issue, the other can continue operating without interruption.

Table V
A MAPPING OF TARGETS TO GUARDRAIL ACTIONS

Type	Targets	Actions
FM-based agents	Prompts	Block, Filter, Flag, Modify, Validate, Prioritize, Rate Limit, Parallel Calls, Retry, Defer, Isolate, Log
	Models	Block, Filter, Flag, Modify, Validate, Prioritize, Rate Limit, Parallel Calls, Retry, Fall Back, Human Intervention, Defer, Isolate, Simulate, Redundancy, Log
	External Data	Block, Filter, Flag, Modify, Validate, Prioritize, Rate Limit, Parallel Calls, Retry, Defer, Isolate, Simulate, Redundancy, Log
	Non-AI Components	Block, Filter, Flag, Modify, Validate, Prioritize, Rate Limit, Parallel Calls, Retry, Fall Back, Human Intervention, Defer, Isolate, Simulate, Redundancy, Log
FM-based Agents	Goals	Block, Filter, Flag, Modify, Validate, Prioritize, Retry, Human Intervention, Defer, Simulate, Log
	Context	Block, Filter, Flag, Modify, Simulate, Log
	Memory	Block, Filter, Flag, Modify, Validate, Prioritize, Retry, Human Intervention, Isolate, Simulate, Log
	Reasoning	Flag, Modify, Validate, Human Intervention, Log
	Plans	Block, Flag, Modify, Validate, Retry, Fall Back, Human Intervention, Defer, Simulate, Log
	Actions	Block, Flag, Modify, Validate, Retry, Prioritize, Parallel Calls, Fall Back, Human Intervention, Defer, Simulate, Log
	Intermediate Results	Flag, Validate, Human Intervention, Log
	Final Results	Block, Filter, Flag, Modify, Validate, Retry, Fall Back, Human Intervention, Log
	Tools	Block, Prioritize, Rate Limit, Parallel Calls, Retry, Fall Back, Human Intervention, Defer, Log
	Other Agents	Block, Flag, Rate Limit, Parallel Calls, Retry, Fall Back, Human Intervention, Defer, Isolate, Simulate, Log
	Intermediate Results	Flag, Validate, Human Intervention, Log
	Final Results	Block, Filter, Flag, Modify, Validate, Retry, Fall Back, Human Intervention, Log

- **Log:** The *log* action involves recording system activities, interactions, and events [11, 70]. For instance, logging all user interactions with an FM-based chatbot allows for the analysis of user interests.

*2) **Targets for Guardrails**:* The key targets guardrails can be applied to include prompts, models, external data, non-AI components, and agent-specific targets. Table V provides an overview of targets and corresponding guardrail actions.

- **Prompts:** Prompts are the initial user inputs or queries. Guardrails on prompts help ensure that user prompts are relevant, appropriate, formatted correctly, and easier for FMs to understand [37, 56, 70].

- **Models:** Models include FMs and narrow models for specific tasks. Guardrails ensures the outputs generated by models are relevant, appropriate and safe. Also, guardrails oversee the utilization of both FMs and narrow models, preventing misuse and ensuring their application under appropriate conditions [1, 20].

- **External Data:** Guardrails enforce stringent monitoring and validation of external data sources, particularly in Retrieval augmented generation scenarios [17]. For example, they can prevent the integration of unverified news that could potentially mislead the system's outputs, ensuring that only reliable and accurate information is used [73].

- **Non-AI Components:** Guardrails also apply to non-AI components, e.g. when making calls APIs [1, 71]. For instance, they can monitor API calling to prevent data leaks [74], such as restricting access to sensitive information.

- **Agent-Specific Targets:** Guardrails target following key aspects of agents:
  - **Goals:** Ensuring that agents' goals align with human values and do not deviate from the human's intended goals [16, 49].
  - **Context:** Monitoring the context that agents collect to ensure it is relevant information and appropriate [36].
  - **Memory:** Managing the agents' memory to retain relevant data and discard outdated or irrelevant information, while also preventing memory poisoning [36, 64].
  - **Reasoning:** Enhancing agents' reasoning capabilities to ensure accurate analysis and sound decision-making [30].
  - **Plans:** Ensuring the generated plans align with human goals [30, 54].
  - **Actions:** Monitoring agents' actions to ensure they are safe and effective in achieving desired outcomes [36].
  - **Tools:** Overseeing the proper use of tools by agents, including implementing access controls, restricting tool capabilities, and detecting potential vulnerabilities [36, 49].
  - **Other Agents:** Managing interactions between agents to ensure collaboration, prevent conflicts, and mitigate risks associated with malicious behaviors [30, 49].
  - **Intermediate Results:** Intermediate results are the outputs generated at various stages during the workflow generation of agents, before reaching the final outputs. By monitoring intermediate results, guardrails can detect anomalies or inaccuracies before they propagate to the final results.
  - **Final Results:** Final results are the end outputs generated by agents, which are delivered to users or downstream systems. Guardrails ensure that the final results meet user expectations and comply with ethical guidelines and legal regulations.

*3) **Guardrails Scopes**:* The scope of guardrails in FM-based agents ranges from individual preferences to industry standards, reflecting the multi-layered approach of the Swiss Cheese Model. At the user level, guardrails reflect individual preferences and requirements. This involves adjusting the

system's behavior based on user-defined settings to align outputs with both user expectations and ethical considerations. Incorporating user preferences into guardrails provides a personalized experience while maintaining safety and compliance [55, 69]. Such guardrails ensure that the agents respects user autonomy and produces outputs that are relevant and acceptable.

At the organizational level, guardrails align with internal policies and procedures governing the operation and use of FM-based agents. This includes compliance with corporate governance, data protection policies, and ethical guidelines established by the organization [12]. Guardrails also ensure consistency and accountability across different departments and functions within the organization.

Industry-level regulations and standards provide the broader regulatory framework within which FM-based agents must operate. Guardrails designed to comply with these regulations guarantee that the system adheres to industry best practices and legal requirements [16]. Guardrails facilitate simpler auditing and certification processes, ensuring the system remains compliant with evolving regulatory landscapes.

System-level constraints focus on the technical and operational limitations within which the FM-based system operates. Guardrails at this level ensure that the system functions efficiently within these constraints, such as computational and memory limits, while maintaining robustness and reliability [26]. They also ensure that the system's operations do not exceed predefined thresholds that could lead to performance degradation or security vulnerabilities.

4) ***Rules****:* Guardrails rules can be configured in different ways: including uniform rules, priority-enabled rules, context-dependent rules, and negotiable rules, forming layers of defense as per the Swiss Cheese Model. A uniform strategy applies the same set of guardrails consistently across all scenarios, ensuring simplicity and uniformity [67]. It is particularly effective in environments with stable and well-understood risks. It largely reduces the complexity of managing diverse guardrails [55]. A priority-enabled strategy prioritizes certain guardrails based on the criticality and sensitivity of operations or data. Context-dependent strategies adjust the implementation of guardrails based on the system's specific operational context. This allows for dynamic adjustments to guardrails in response to changing conditions, user needs, and operational environments [49]. The negotiability of guardrails, categorized into hard and soft, defines the level of flexibility in enforcing rules. Soft guardrails allow adjustments based on context and situational demands, providing a balance between protection and operational flexibility [49]. In contrast, hard guardrails are rigid and non-negotiable, ensuring adherence to critical legal, ethical, or safety standards [12, 32].

5) ***Autonomy****:* Guardrails vary based on the level of autonomy in the system. They can be either automatic or require human intervention. Automatic guardrails function without human oversight, relying on pre-defined rules and real-time monitoring to enforce protections [53]. This is useful in scenarios requiring immediate response to potential threats.

In contrast, human intervention guardrails involve manual oversight, where human operators review and approve critical outputs or actions flagged by the system [32]. This approach is essential in high-stakes environments, such as financial systems, where human judgment can provide an additional layer of scrutiny and ensure compliance with regulatory standards.

6) ***Modality****:* The modality of guardrails refers to the types of data and interactions they manage. Guardrails can be designed for single modal or multimodal systems. Single modal systems operate with one type of data input or output, such as text, image, or audio. For instance, in text-based agent systems, guardrails focus on addressing issues like offensive language, misinformation, and data privacy [49]. In image-based agent systems, they may involve techniques for detecting explicit content or ensuring image quality standards [26].

Multimodal guardrails address the combined risks of handling multiple data types. They synchronize protections across different data types, ensuring comprehensive security and compliance [55]. For example, a system that generates text based on image inputs must ensure accurate and ethical representation of the image content. This requires advanced cross-modal analysis and validation techniques to ensure the system operates reliably and ethically across all data types it handles [53].

7) ***Underlying Technique****:* The underlying techniques of guardrails include rule-based, hybrid, and machine learning models, with each representing a distinct design option to meet specific requirements. These techniques offer the necessary flexibility, adaptability, and robustness to protect the systems [3, 27]

Rule-based models utilize predefined rules to monitor and control FM-based system behavior. These models implement strict and deterministic guidelines that the system must follow to ensure compliance with regulatory requirements for data access and processing [49]. They are particularly effective in environments where operational parameters are well-defined and stable. Rule-based agent systems can be updated and are somewhat flexible. However, they may still struggle with unexpected scenarios, such as detecting novel AI-generated content that falls outside predefined rules. This reliance on static rules can limit their adaptability and regular updates are needed [16, 71].

In contrast, machine learning models dynamically adapt and improve guardrails based on new data and scenarios. These models can also learn from historical data and identify patterns that indicate potential risks or compliance issues [64]. Machine learning models can be further subclassed into narrow models and foundation models. Narrow models are specialized systems designed for specific tasks or domains. They require targeted guardrails to address domain-specific risks and compliance needs [15]. Foundation models are large, general-purpose models that serve as the backbone for multiple applications and tasks. These models necessitate comprehensive and scalable guardrails to handle a wide range of risks and compliance issues across different applications [26]. Nevertheless, they can be computationally intensive and require

substantial data for training.

Hybrid models integrate rule-based agent systems with the adaptability of machine learning models to respond to new threats and evolving data patterns [53]. For instance, Khorramrouz et al.[59] demonstrate the use of the PaLM 2 framework to process user input and dynamically implement rule-based decisions. This framework tests the system's limits by iteratively generating toxic content to evaluate PaLM 2's safety guardrails. However, integrating hybrid models can increase system complexity and create additional challenges [53].

V. Discussion

In this section, we discuss the implications of our proposed taxonomy, its alignment with the Swiss Cheese Model for AI safety, and how it addresses the challenges in designing multi-layered runtime guardrails for FM-based agents. We also reflect on the potential impact of our findings on future research and practice in AI safety by design.

A. Adapting the Swiss Cheese Model for Multi-Layered Guardrails

The Swiss Cheese Model, originally developed in risk management and safety science, illustrates how multiple layers of defense can prevent hazards from leading to adverse outcomes [75]. Each layer has potential weaknesses (holes), but when combined, they provide a robust barrier against failures. In the context of FM-based agents, the Swiss Cheese Model underscores the necessity of implementing multi-layered guardrails to address the unique challenges introduced by these agents' autonomous and non-deterministic behaviors.

FM-based agents interact with various components during runtime, including goals, prompts, plans, tools, knowledge bases, and intermediate and final results. Each of these components represents a potential point of failure or risk. By adapting the Swiss Cheese Model, we propose that guardrails should be implemented at multiple layers of the agent architecture to effectively mitigate risks. This multi-layered approach ensures that if one layer fails to detect or prevent an undesirable behavior, subsequent layers can compensate, thereby enhancing the overall safety and reliability of the agent.

Implementing guardrails at different levels—such as input processing, model reasoning, action execution, and output generation—aligns with the Swiss Cheese Model's principle of layered defenses, including overlapping layers. This approach addresses the complex interplay between different components in FM-based agents and provides a systematic framework for AI safety by design. It also highlights the importance of considering both external and internal quality attributes in guardrail design, as well as the trade-offs involved in different design options.

B. Implications for Practice and Research

Our taxonomy offers practical guidance for developers and researchers in designing multi-layered runtime guardrails for FM-based agents. By categorizing the motivations, quality attributes, and design options, we provide a structured approach to address AI safety concerns effectively. Practitioners can use this taxonomy to make informed architectural decisions, balancing trade-offs between different quality attributes such as accuracy, performance, adaptability, interpretability and so on.

For researchers, our work assists in exploring new techniques and methodologies in guardrail design and implementation. The alignment with the Swiss Cheese Model encourages a holistic view of safety mechanisms, promoting research into how different layers can interact synergistically to enhance agent safety. Furthermore, understanding the relationships between guardrails, the risks they mitigate, and the quality attributes they influence can inform the development of standards and best practices in the field of AI safety.

C. Threats to Validity

Our study is subject to standard literature search and selection bias threats. We addressed these threats by searching the most commonly used databases in the IT and software engineering domains. We adjusted our search strings several times during the automatic search to maximize the number of relevant articles matching two key concepts: 'guardrails' and 'FM-Based agents'. We also kept our search string generic to search through the titles, abstracts, keywords, and full text of articles to cover the maximum number of relevant papers. We then conducted a manual search on Google Scholar to complement the automatic search using a snowballing strategy. Furthermore, predefined review protocols with detailed inclusion and exclusion criteria helped us reduce bias in selecting primary studies. We applied several quality assessment criteria (shown in Section III-F) to estimate the quality of the selected primary studies. Even though the proposed criteria were not too strict, applying them indeed caused several initially selected papers to be excluded. To mitigate the risk of missing important data from the primary studies, we reinstated the excluded papers closely related to the primary studies.

Moreover, our definitions and categorizations may not capture all relevant aspects of guardrails in FM-based agents. To mitigate this threat, we validated the taxonomy through extensive literature review and expert feedback. However, this introduces a risk of producing biased results that address only expert needs, as the people involved in the feedback process have extensive experience in the AI and software engineering domains. Our review protocols helped us to reduce such bias.

We prepared a guardrails taxonomy and conducted a comparative analysis of its components to help the reader better understand their design and evaluation. We critically examined the strength and consistency of relationships in the selected studies for a reliable taxonomy structure and drew conclusions. Nonetheless, the generalizability of guardrails to different contexts and types in FM-based agents remains a potential limitation, and specific adaptations might be necessary for certain systems, such as those used in healthcare or financial organizations. Additionally, our findings face a challenge as some of them may become outdated due to the rapid evolution of FM-based agents and associated guardrails.

VI. Conclusion and Future Work

To better understand runtime guardrails in FM-based agents, this paper developed and presented a comprehensive guardrail taxonomy and provided a comparative study of its components based on a systematic literature review guided by the Swiss Cheese Model for AI safety. Our proposed taxonomy categorizes guardrails based on their fundamental objectives, essential quality attributes, performed actions, targeted aspects, covered scopes, employed rules, autonomy, modalities, and underlying techniques.

Our key findings emphasize that the effective integration of guardrails mitigates risk such as biases, harmful content, and unintended behaviors in FM-based agents. Practitioners can use the proposed taxonomy to improve the design of guardrails to ensure the behavior and decisions of FM-based agents are responsible and safe. Moreover, the taxonomy provides a theoretical framework that researchers and policymakers can use to develop guidelines and standards that promote AI-safety-by-design in FM-based agents.

In the future, we plan to develop a multilayer architecture for guardrails in FM-based agents, integrating various design options outlined in this taxonomy. Additionally, we aim to evaluate the effectiveness of multi-layered guardrails in real-world agent deployments, further refining the taxonomy and contributing to best practices in AI safety.

References

[1] R. Bommasani, D. A. Hudson, E. Adeli, and et al., "On the opportunities and risks of foundation models," *CoRR*, vol. abs/2108.07258, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2108.07258

[2] Q. Lu, L. Zhu, J. Whittle, and X. Xu, *Responsible AI: Best practices for creating trustworthy AI systems*, 1st ed. Addison-Wesley Professional, 2023. [Online]. Available: https://www.pearson.com/en-us/subject-catalog/p/responsible-ai-best-practices-for-creating-trustworthy-ai-systems/P200000010211/9780138073886

[3] Q. Lu, L. Zhu, X. Xu, Z. Xing, and J. Whittle, "Towards responsible and safe AI in the era of foudnation models: A reference architecture for designing foundation model based systems," *to appear in IEEE Software*, 2024. [Online]. Available: https://arxiv.org/html/2304.11090v4

[4] N. Maslej *et al.*, "AI index report 2024," Stanford Institute for Human-Centered Artificial Intelligence (HAI), Tech. Rep., 2024. [Online]. Available: https://aiindex.stanford.edu/report/2024

[5] G. V. R. Team, "Artificial intelligence market size, share, growth report 2030," Grand View Research, Tech. Rep., 2024. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-market

[6] R. Bommasani and P. Liang, "Reflections on foundation models," 2021, Last accessed on Jun.-2024. [Online]. Available: https://hai.stanford.edu/news/reflections-foundation-models

[7] S. Uspenskyi, "Large language model statistics and numbers (2024)," 2024, Last accessed on Jun.-2024. [Online]. Available: https://springsapps.com/knowledge/large-language-model-statistics-and-numbers-2024

[8] L. Wang *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, pp. 1–26, 2024. [Online]. Available: https://doi.org/10.1007/s11704-024-40231-1

[9] Y. Wang and L. Singh, "Adding guardrails to advanced chatbots," *arXiv preprint arXiv:2306.07500*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2306.07500

[10] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer *et al.*, "DecodingTrust: A comprehensive assessment of trustworthiness in GPT models," in *Advances in Neural Information Processing Systems*, 2023, pp. 1–110. [Online]. Available: https://doi.org/10.48550/arXiv.2306.11698

[11] B. Wei, K. Huang, Y. Huang, T. Xie, X. Qi, M. Xia, P. Mittal, M. Wang, and P. Henderson, "Assessing the brittleness of safety alignment via pruning and low-rank modifications," *arXiv preprint arXiv:2402.05162*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.05162

[12] M. Anderljung *et al.*, "Frontier AI regulation: Managing emerging risks to public safety," *arXiv preprint arXiv:2307.03718*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2307.03718

[13] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does LLM safety training fail?" *Advances in Neural Information Processing Systems*, vol. 36, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2307.02483

[14] A. Gubkin, "Understanding why ai guardrails are necessary: Ensuring ethical and responsible ai use," 2024, Last accessed on Jul.-2024. [Online]. Available: https://www.aporia.com/learn/ai-guardrails/

[15] Y. Dong, R. Mu, G. Jin, Y. Qi, J. Hu, X. Zhao, J. Meng, W. Ruan, and X. Huang, "Building guardrails for large language models," *arXiv preprint arXiv:2402.01822*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.01822

[16] T. Rebedea, R. Dinu, M. N. Sreedhar, C. Parisien, and J. Cohen, "NeMo guardrails: A toolkit for controllable and safe LLM applications with programmable rails," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Singapore, Dec. 2023, pp. 431–445. [Online]. Available: https://aclanthology.org/2023.emnlp-demo.40

[17] D. Kang, D. Raghavan, P. Bailis, and M. Zaharia, "Model assertions for monitoring and improving ml models," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 481–496, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2003.01668

[18] J. Mökander, J. Schuett, H. R. Kirk, and L. Floridi, "Auditing large language models: A three-layered approach," *AI and Ethics*, pp. 1–31, 2023. [Online]. Available: https://doi.org/10.1007/s43681-023-00289-2

[19] A. Kumar, S. Singh, S. V. Murty, and S. Ragupathy, "The ethics of interaction: Mitigating security threats in LLMs," *arXiv preprint arXiv:2401.12273*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.12273

[20] C. Zhou *et al.*, "A comprehensive survey on pretrained foundation models: A history from BERT to ChatGPT," *arXiv preprint arXiv:2302.09419*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.09419

[21] M. Zaharia, O. Khattab, L. Chen, J. Q. Davis, H. Miller, C. Potts, J. Zou, M. Carbin, J. Frankle, N. Rao, and A. Ghodsi, "The shift from models to compound AI systems," Berkeley Artificial Intelligence Research (BAIR), Tech. Rep., 2024. [Online]. Available: https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/

[22] IBM, "What are large language models (LLMs)?" 2024, Last accessed on Jun.-2024. [Online]. Available: https://www.ibm.com/topics/large-language-models

[23] ——, "What are foundation models?" 2024, Last accessed on Jun.-2024. [Online]. Available: https://research.ibm.com/blog/what-are-foundation-models

[24] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, mar 2024. [Online]. Available: https://doi.org/10.1145/3641289

[25] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J.-R. Wen, J. Yuan, W. X. Zhao, and J. Zhu, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000231

[26] S. Ee and Z. Williams, "Adapting cybersecurity frameworks to manage frontier AI risks," *Institute for AI Policy and Strategy (IAPS)*, 2023. [Online]. Available: https://static1.squarespace.com/static/64edf8e7f2b10d716b5ba0e1/t/6528c5c7f912f74fbd03fc34/1697170896984/Adapting+cybersecurity+frameworks+to+manage+frontier+AI+risks.pdf

[27] Q. Lu, L. Zhu, X. Xu, Y. Liu, Z. Xing, and J. Whittle, "A taxonomy of foundation model based systems through the lens of software architecture," in *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI*, 2024, pp. 1–6. [Online]. Available: https://doi.org/10.48550/arXiv.2305.05352

[28] X. Tang, Q. Jin, K. Zhu, T. Yuan, Y. Zhang, W. Zhou, M. Qu, Y. Zhao, J. Tang, Z. Zhang *et al.*, "Prioritizing safeguarding over autonomy: Risks of LLM agents for science," *arXiv preprint arXiv:2402.04247*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.04247

[29] S. G. Ayyamperumal and L. Ge, "Current state of LLM risks and AI guardrails," *arXiv preprint arXiv:2406.12934*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.12934

[30] Q. Lu, L. Zhu, X. Xu, Z. Xing, S. Harrer, and J. Whittle, "Building the future of responsible AI: A reference architecture for designing large language model based agents," *arXiv e-prints*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.13148

[31] Y. Bengio *et al.*, "Managing extreme AI risks amid rapid progress," *Science*, vol. 384, no. 6698, pp. 842–845, 2024. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.adn0117

[32] L. Weidinger, M. Rauh, N. Marchal, A. Manzini, L. A. Hendricks, J. Mateos-Garcia, S. Bergman, J. Kay, C. Griffin, B. Bariach *et al.*, "Sociotechnical safety evaluation of generative ai systems," *arXiv preprint arXiv:2310.11986*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2310.11986

[33] A. E. Hassan, D. Lin, G. K. Rajbahadur, K. Gallaba, F. R. Cogo, B. Chen, H. Zhang, K. Thangarajah, G. A. Oliva, J. Lin *et al.*, "Rethinking software engineering in the era of foundation models: A curated catalogue of challenges in the development of trustworthy fmware," *arXiv preprint arXiv:2402.15943*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.15943

[34] M. Mylrea and N. Robinson, "Artificial intelligence (ai) trust framework and maturity model: Applying an entropy lens to improve security, privacy, and ethical ai," *Entropy*, vol. 25, no. 10, 2023. [Online]. Available: https://www.mdpi.com/1099-4300/25/10/1429

[35] OpenAI, "OpenAI's moderation API," 2024, Last accessed on Jul.-2024. [Online]. Available: https://platform.openai.com/docs/guides/moderation/overview

[36] Z. Xiang *et al.*, "GuardAgent: Safeguard llm agents by a guard agent via knowledge-enabled reasoning," *arXiv preprint arXiv:2406.09187*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.09187

[37] P. Rai, S. Sood, V. K. Madisetti, and A. Bahga, "Guardian: A multi-tiered defense architecture for thwarting prompt injection attacks on LLMs," *Journal of Software Engineering and Applications*, vol. 17, no. 1, pp. 43–68, 2024. [Online]. Available: https://doi.org/10.4236/jsea.2024.171003

[38] T. Bi, G. Yu, Q. Lu, X. Xu, and N. Van Beest, "The privacy pillar - A conceptual framework for foundation model-based systems," *arXiv preprint arXiv:2311.06998*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2311.06998

[39] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008. [Online]. Available: https://doi.org/10.1002/9780470754887

[40] B. A. Kitchenham, S. Charters, and Other Keele Staffs, "Guidelines for performing systematic literature reviews in software engineering (version 2.3)," Keele University and Durham University Joint Report, Tech. Rep., 2007. [Online]. Available: https://www.elsevier.com/__data/promis_misc/525444systematicreviewsguide.pdf

[41] B. Kitchenham, "Procedures for performing systematic reviews," Keele University, UK, Tech. Rep. 2004, 2004. [Online]. Available: http://artemisa.unicauca.edu.co/~ecaldon/docs/spi/kitchenham_2004.pdf

[42] A. Paez, "Gray literature: An important resource in systematic reviews," *Journal of Evidence-Based Medicine*, vol. 10, no. 3, pp. 233–240, 2017.

[43] K. Godin, J. Stapleton, S. I. Kirkpatrick, R. M. Hanning, and S. T. Leatherdale, "Applying systematic review search methods to the grey literature: a case study examining guidelines for school-based breakfast programs in canada," *Systematic reviews*, vol. 4, pp. 1–10, 2015.

[44] L. Schmidt, A. Finnerty Mutlu, R. Elmore, B. Olorisade, J. Thomas, and J. Higgins, "Data extraction methods for systematic review (semi)automation: Update of a living systematic review," *F1000Research*, vol. 10, no. 401, 2023. [Online]. Available: https://doi.org/10.12688/f1000research.51117.2

[45] V. Rawte, A. Sheth, and A. Das, "A survey of hallucination in large foundation models," *arXiv preprint arXiv:2309.05922*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2309.05922

[46] OpenAI, "OpenAI safety update," 2024. [Online]. Available: https://openai.com/index/openai-safety-update/

[47] S. Torkington, "These are the 3 biggest emerging risks the world is facing," World Economic Forum, Tech. Rep., 2024. [Online]. Available: https://www.weforum.org/agenda/2024/01/ai-disinformation-global-risks/

[48] C. Hutton, "Silicon valley self-regulates for AI misinformation in 2024 elections while government lags," 2024. [Online]. Available: https://www.washingtonexaminer.com/news/2803412/silicon-valley-self-regulates-ai-misinformation-in-2024-government-lags/

[49] S. Goyal, M. Hira, S. Mishra, S. Goyal, A. Goel, N. Dadu, D. Kirushikesh, S. Mehta, and N. Madaan, "LLMGuard: guarding against unsafe LLM behavior," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38(21), 2024, pp. 23 790–23 792. [Online]. Available: https://doi.org/10.1609/aaai.v38i21.30566

[50] S. Ray, "Samsung bans ChatGPT among employees after sensitive code leak," 2023, News article, Last accessed on Jul.-2024. [Online]. Available: https://www.forbes.com/sites/siladityaray/2023/05/02/samsung-bans-chatgpt-and-other-chatbots-for-employees-after-sensitive-code-leak/

[51] W. Du, Q. Li, J. Zhou, X. Ding, X. Wang, Z. Zhou, and J. Liu, "Finguard: A multimodal AIGC guardrail in financial scenarios," in *Proceedings of the 5th ACM International Conference on Multimedia in Asia*, Taiwan, 2024, pp. 1–3. [Online]. Available: https://doi.org/10.1145/3595916.3626351

[52] T. Zemčík, "Failure of chatbot Tay was evil, ugliness and uselessness in its nature or do we judge it through cognitive shortcuts and biases?" *AI & SOCIETY*, vol. 36, pp. 361–367, 2021. [Online]. Available: https://doi.org/10.1007/s00146-020-01053-4

[53] Z. Yuan, Z. Xiong, Y. Zeng, N. Yu, R. Jia, D. Song, and B. Li, "RigorLLM: Resilient guardrails for large language models against undesired content," *arXiv preprint arXiv:2403.13031*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.13031

[54] S. Banerjee, S. Layek, R. Hazra, and A. Mukherjee, "How (un) ethical are instruction-centric responses of LLMs? unveiling the vulnerabilities of safety guardrails to harmful queries," *arXiv preprint arXiv:2402.15302*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.15302

[55] J. Zhao, K. Chen, X. Yuan, Y. Qi, W. Zhang, and N. Yu, "Silent guardian: Protecting text from malicious exploitation by large language models," *arXiv preprint arXiv:2312.09669*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2312.09669

[56] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, ""do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," *arXiv preprint arXiv:2308.03825*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2308.03825

[57] M. Liffiton, B. E. Sheese, J. Savelka, and P. Denny, "Codehelp: Using large language models with guardrails for scalable support in programming classes," in *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, Finland, 2024, pp. 1–11. [Online]. Available: https://doi.org/10.1145/3631802.3631830

[58] Z. Wang, F. Yang, L. Wang, P. Zhao, H. Wang, L. Chen, Q. Lin, and K.-F. Wong, "SELF-GUARD: Empower the LLM to safeguard itself," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Mexico, 2024, pp. 1648–1668. [Online]. Available: https://aclanthology.org/2024.naacl-long.92

[59] A. Khorramrouz, S. Dutta, A. Dutta, and A. R. KhudaBukhsh, "Down the toxicity rabbit hole: Investigating PaLM 2 guardrails," *arXiv preprint arXiv:2309.06415*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2309.06415

[60] N. Mangaokar, A. Hooda, J. Choi, S. Chandrashekaran, K. Fawaz, S. Jha, and A. Prakash, "PRP: Propagating universal perturbations to attack large language model guard-rails," *arXiv preprint arXiv:2402.15911*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.15911

[61] P. Gajane and M. Pechenizkiy, "On formalizing fairness in prediction with machine learning," *arXiv preprint arXiv:1710.03184*, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1710.03184

[62] R. R. Llaca, V. Leskoschek, V. C. Paiva, C. Lupău, P. Lippmann, and J. Yang, "Student-teacher prompting for red teaming to improve guardrails," in *Proceedings of the ART of Safety: Workshop on Adversarial testing and Red-Teaming for generative AI*, 2023, pp. 11–23. [Online]. Available: http://dx.doi.org/10.18653/v1/2023.artofsafety-1.2

[63] W. Li *et al.*, "Segment anything model can not segment anything: Assessing AI foundation model's generalizability in permafrost mapping," *Remote Sensing*, vol. 16, no. 5, p. 797, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.08787

[64] Z. Zhang, Y. Lu, J. Ma, D. Zhang, R. Li, P. Ke, H. Sun, L. Sha, Z. Sui, H. Wang *et al.*, "ShieldLm: Empowering LLMs as aligned, customizable and explainable safety detectors," *arXiv preprint arXiv:2402.16444*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.16444

[65] H.-I. Kim, K. Yun, J.-S. Yun, and Y. Bae, "Customizing segmentation foundation model via prompt learning for instance segmentation,"

arXiv preprint arXiv:2403.09199, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2403.09199

[66] R. Y. Wong, A. Chong, and R. C. Aspegren, "Privacy legislation as business risks: How GDPR and CCPA are represented in technology companies' investment risk disclosures," *Proceedings of the ACM on Human-Computer Interaction*, vol. 7, no. CSCW1, pp. 1–26, 2023. [Online]. Available: https://doi.org/10.1145/3579515

[67] Z. Chu, Y. Wang, L. Li, Z. Wang, Z. Qin, and K. Ren, "A causal explainable guardrails for large language models," *arXiv preprint arXiv:2405.04160*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.04160

[68] M. Shanahan, "Talking about large language models," *Commun. ACM*, vol. 67, no. 2, p. 68–79, jan 2024. [Online]. Available: https://doi.org/10.1145/3624724

[69] Y. Zeng, H. Lin, J. Zhang, D. Yang, R. Jia, and W. Shi, "How johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing LLMs," *arXiv preprint arXiv:2401.06373*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2401.06373

[70] A. Kumar, C. Agarwal, S. Srinivas, S. Feizi, and H. Lakkaraju, "Certifying LLM safety against adversarial prompting," *arXiv preprint arXiv:2309.02705*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2309.02705

[71] D. Dalrymple *et al.*, "Towards guaranteed safe AI: A framework for ensuring robust and reliable AI systems," *arXiv preprint arXiv:2405.06624*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2405.06624

[72] M. Pawagi and V. Kumar, "Guardrails: Automated suggestions for clarifying ambiguous purpose statements," in *Proceedings of the 16th Annual ACM India Compute Conference*, 2023, p. 55–60. [Online]. Available: https://doi.org/10.1145/3627217.3627234

[73] W. Zou, R. Geng, B. Wang, and J. Jia, "PoisonedRAG: Knowledge poisoning attacks to retrieval-augmented generation of large language models," *arXiv preprint arXiv:2402.07867*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.07867

[74] J. Hua and P. Wang, "Security vulnerabilities in facebook data breach," in *International Conference on Information Technology-New Generations*. Springer, 2024, pp. 159–166. [Online]. Available: https://doi.org/10.1007/978-3-031-56599-1_22

[75] J. Larouzee and J.-C. Le Coze, "Good and bad reasons: The swiss cheese model and its critics," *Safety Science*, vol. 126, p. 104660, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753520300576

[76] T. Shabani, S. Jerie, and T. Shabani, "A comprehensive review of the swiss cheese model in risk management," *Safety in Extreme Environments*, vol. 6, no. 1, pp. 43–57, 2024.