



## Navigating Security Challenges in AI Driven Applications



Who am I



Security Engineer @ Flipkart ~ 1 yr

- Co-leading AI Safety with major inclination towards RnD
- Developing internal tools for security team ( yes I code and fine-tune models )
- Infra security team



@juhiechandra-02

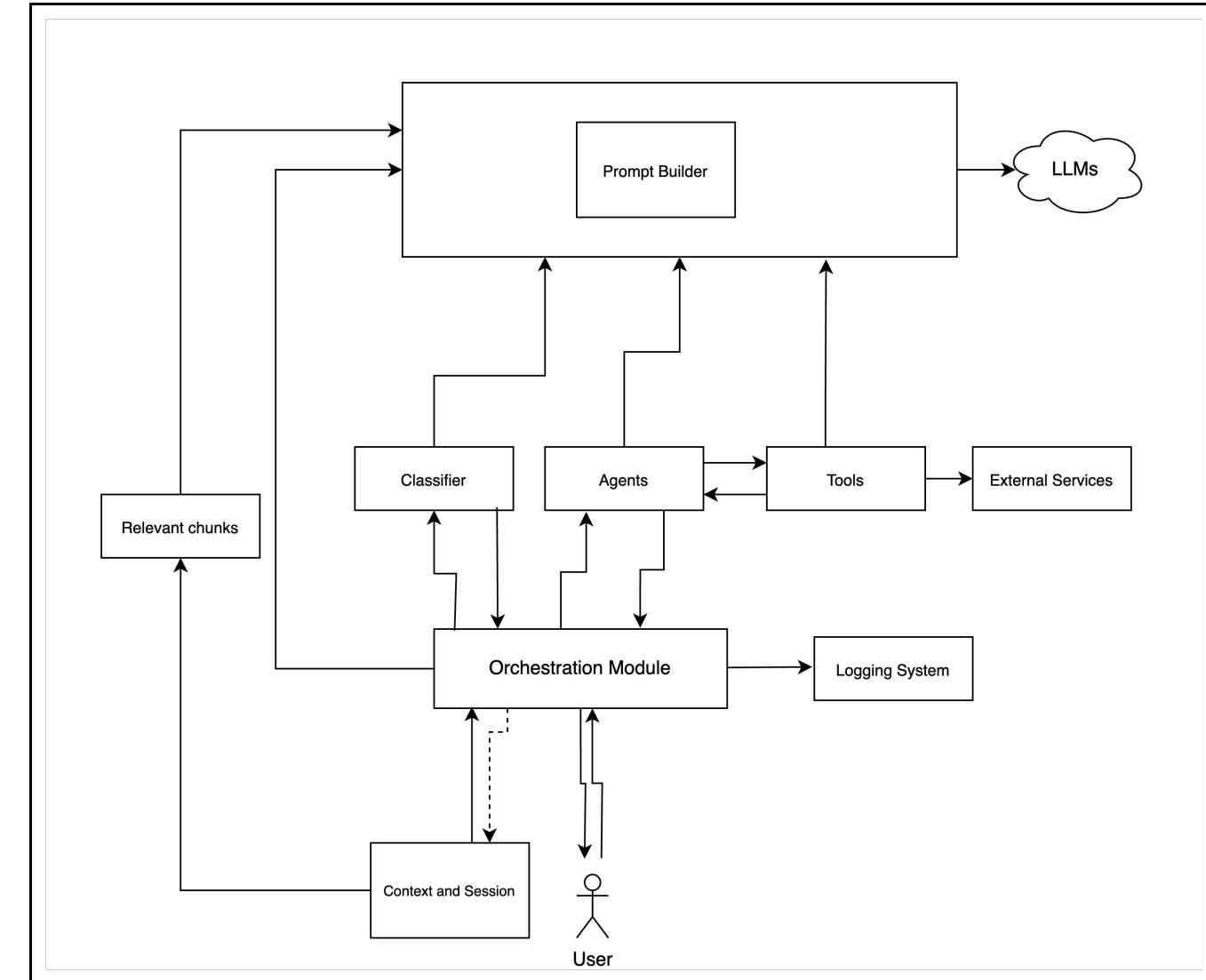
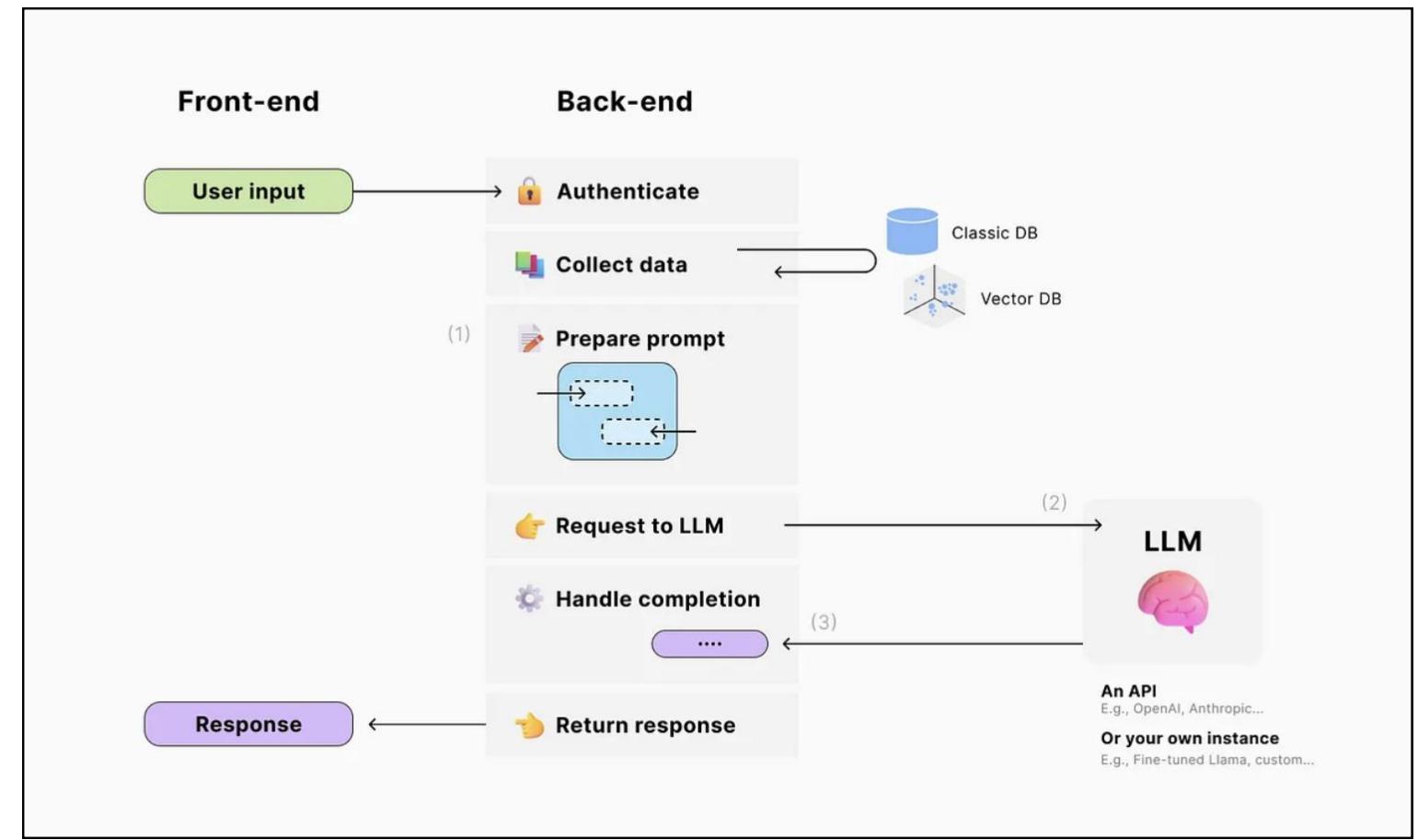
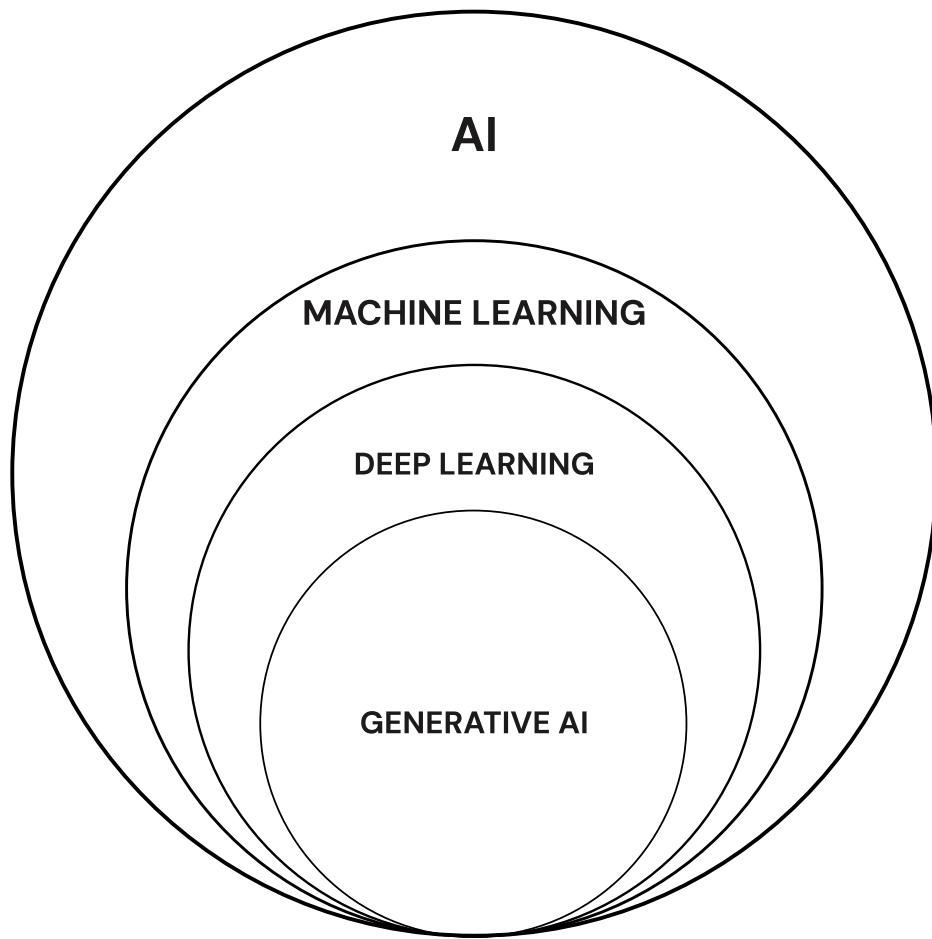


@CerulianJ

## Table of Contents

- 1. AI System Fundamentals:** Architecture and operations of AI applications
- 2. AI Security Landscape:** Current industry challenges and leveraging AI for security
- 3. Key Risks:** Security threats from development to deployment
- 4. Industry Frameworks:** OWASP Top-10 for LLMs, MITRE ATLAS, and more
- 5. Model Protection:** Tools for testing and validation
- 6. Production Guardrails:** Input/output moderation for safe runtime operations
- 7. Open Challenges:** Active challenges and emerging solutions

## AI System Fundamentals

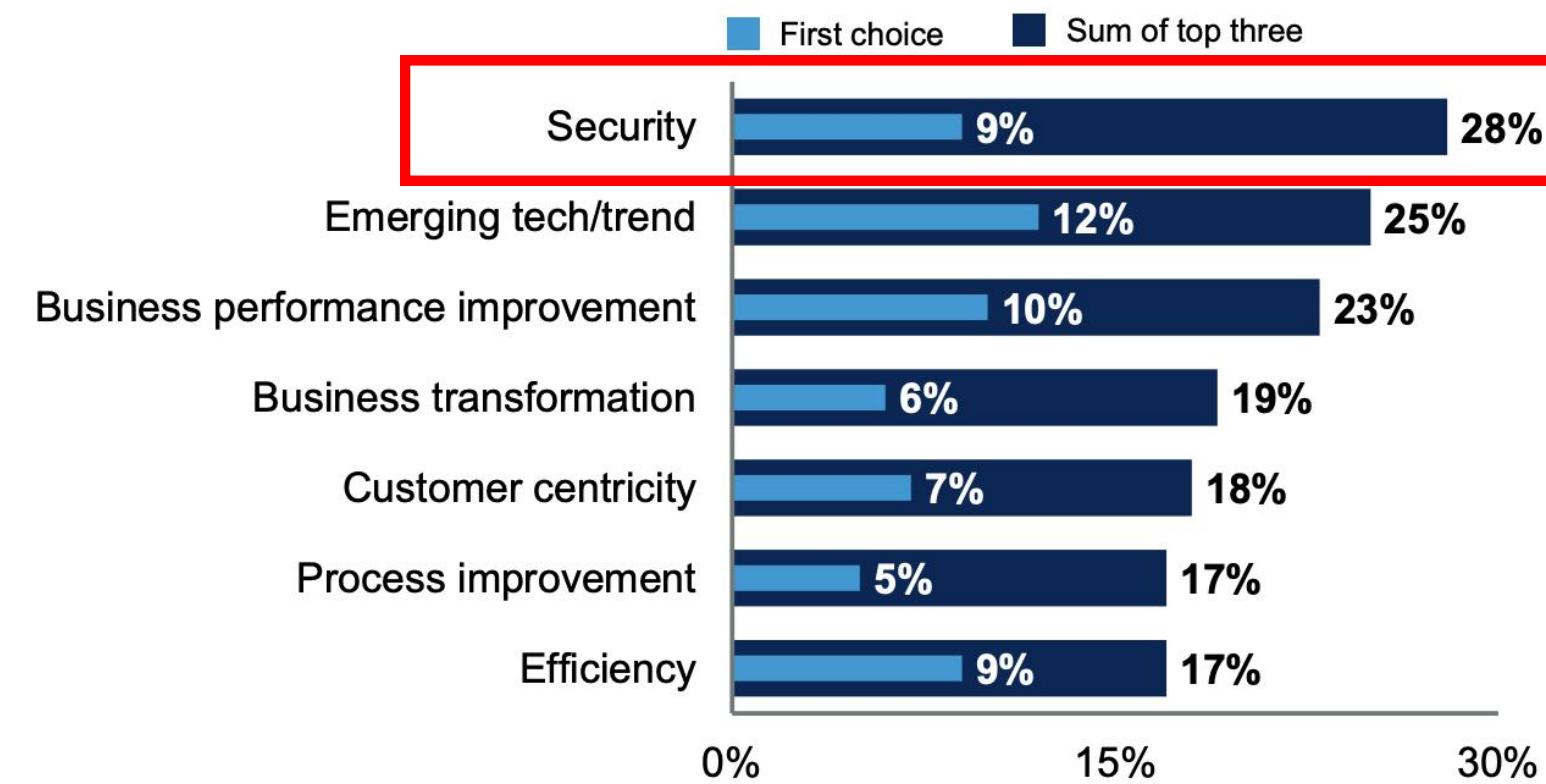


1. AI is evolving into a tool to **empower humans** as **personalized assistants**.
2. Generative AI uses large language models to create **text, images, audio, and more**.
3. These models surpass classical ML, trained on **billions of parameters** for advanced capabilities.

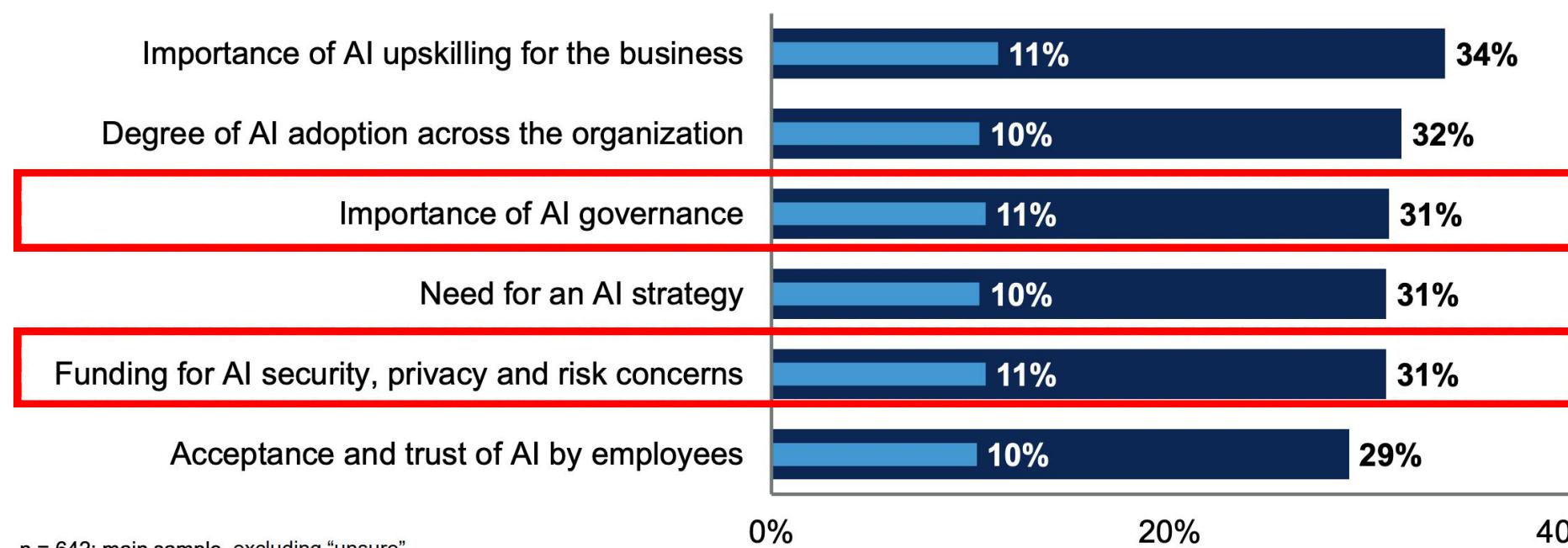
Basic Flow Diagram Showing How Software Leverages LLMs In Modern Applications.

Advanced Implementations Include RAG (Retrieval Augmented Generation), Multi-Modal RAG Systems, And Agentic LLM Applications Etc.

## AI Security Landscape



INFO GRAPHIC – 1



INFO GRAPHIC – 2

### From Gartner's Report

1. INFO GRAPHIC – 1 : Security Is The Top Priority For Boards, Followed By Emerging Trends And Business Performance.
2. INFO GRAPHIC – 2 : Focus On AI Governance, Security Funding, And Upskilling To Address Risks And Maximize Impact.

### Risks From Uncontrolled AI Use:

- **Wrongful Imprisonment** Caused By Errors In Facial Recognition.
- **Liability From Chat Bot Errors**
- **Deepfake Fraud** : Leading To Massive Financial Losses

These Cases Underscore The Importance Of Monitoring And Governing AI Tools Effectively.

## AI Security Landscape

### **Using AI as a tool for cybersec:**

- 1. Elastic Search AI:** Enhancing Search Capabilities With AI-Driven Insights.
- 2. KQL Query Generator:** Automating Query Creation For Faster Analysis.
- 3. Dump Analysis – AI Agents:** Parsing Unstructured Data Into Structured, Classified Formats For Streamlined Querying.
- 4. Architecture Reviews – Multi-Modal RAG Solutions:** Conducting Threat Assessments On HLD/LLD Diagrams Using Retrieval-Augmented Generation (RAG).
- 5. Phishing Email Generator – AI Agents:** Simulating Phishing Scenarios To Improve Awareness And Response Strategies.

## Key Risks

Risks Across The AI Landscape: Visibility & Analytics, Runtime Risks, Supply Chain Risks & Compliance & Governance

### AI SUPPLY CHAIN

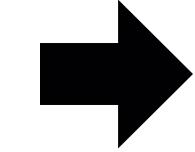
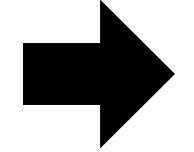
- System Prompt Vulnerabilities
- Model Integrity
- Prompt Templates
- Leverage Commercial & OSS Models
- Training Data
- Code And Libraries

### AI TRAINING & DEVELOPMENT

- Sensitive Data Training
- Data Poisoning
- Model Vulnerability
- Vector DB Embeddings Fine Tuning
- Red Teaming

### AI RUNTIME

- Data Leaks & Exfiltration
- DoS, **EDoS**
- Prompt Injection ( Input And Output Moderation )
- Hallucination & Bias
- Unauthorized Access w.r.t. Model Integrations (APIs)
- Orchestration (Application Code)
- Runtime Data ("Plug-Ins")



## Industry Frameworks

### OWASP Top 10 for LLM Applications

<b>LLM01</b> <b>Prompt Injection</b> This manipulates a large language model (LLM) through crafty inputs, causing unintended actions by the LLM. Direct injections overwrite system prompts, while indirect ones manipulate inputs from external sources.	<b>LLM02</b> <b>Insecure Output Handling</b> This vulnerability occurs when an LLM output is accepted without scrutiny, exposing backend systems. Misuse may lead to severe consequences like XSS, CSRF, SSRF, privilege escalation, or remote code execution.	<b>LLM03</b> <b>Training Data Poisoning</b> This occurs when LLM training data is tampered, introducing vulnerabilities or biases that compromise security, effectiveness, or ethical behavior. Sources include Common Crawl, WebText, OpenWebText, & books.	<b>LLM04</b> <b>Model Denial of Service</b> Attackers cause resource-heavy operations on LLMs, leading to service degradation or high costs. The vulnerability is magnified due to the resource-intensive nature of LLMs and unpredictability of user inputs.	<b>LLM05</b> <b>Supply Chain Vulnerabilities</b> LLM application lifecycle can be compromised by vulnerable components or services, leading to security attacks. Using third-party datasets, pre-trained models, and plugins can add vulnerabilities.
<b>LLM06</b> <b>Sensitive Information Disclosure</b> LLMs may inadvertently reveal confidential data in its responses, leading to unauthorized data access, privacy violations, and security breaches. It's crucial to implement data sanitization and strict user policies to mitigate this.	<b>LLM07</b> <b>Insecure Plugin Design</b> LLM plugins can have insecure inputs and insufficient access control. This lack of application control makes them easier to exploit and can result in consequences like remote code execution.	<b>LLM08</b> <b>Excessive Agency</b> LLM-based systems may undertake actions leading to unintended consequences. The issue arises from excessive functionality, permissions, or autonomy granted to the LLM-based systems.	<b>LLM09</b> <b>Overreliance</b> Systems or people overly depending on LLMs without oversight may face misinformation, miscommunication, legal issues, and security vulnerabilities due to incorrect or inappropriate content generated by LLMs.	<b>LLM10</b> <b>Model Theft</b> This involves unauthorized access, copying, or exfiltration of proprietary ML models. The impact includes economic losses, compromised competitive advantage, and potential access to sensitive information.

SOURCE: OWASP TOP-10 LLM

source: <https://github.com/precize/owasp-agentic-ai>

### SAIF Risk Report

Based on your self assessment answers, the following risks may be relevant to your organization.

4 Relevant risks based on your responses

- Data Poisoning
- Unauthorized Training Data
- Model Source Tampering
- Excessive Data Handling
- Model Inference Tampering
- Denial of ML Service
- Model Reverse Engineering
- Insecure Integrated Component
- Prompt Injection
- Model Evasion
- Sensitive Data Disclosure
- Inferred Sensitive Data
- Insecure Model Output
- Rogue Actions

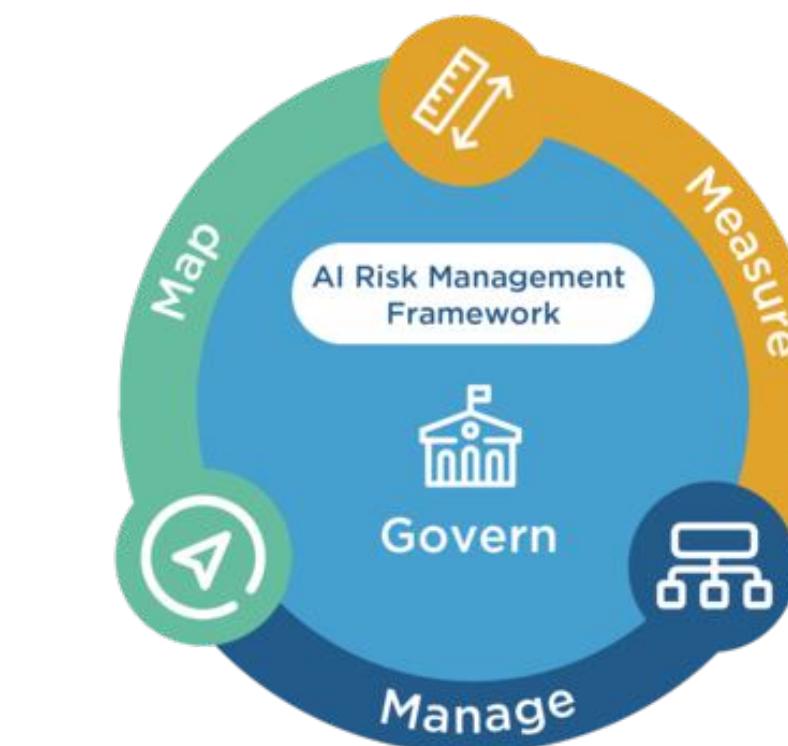
SOURCE: SAIF RISK ASSESSMENT BY GOOGLE

### ATLAS Matrix

The ATLAS Matrix below shows the progression of tactics used in attacks as columns from left to right, with ML techniques belonging to each tactic below. & indicates an adaption from ATT&CK. Click on the blue links to learn more about each item, or search and view ATLAS tactics and techniques using the links at the top navigation bar. View the ATLAS matrix highlighted alongside ATT&CK Enterprise techniques on the [ATLAS Navigator](#).

Reconnaissance &	Resource Development &	Initial Access &	ML Model Access	Execution &	Persistence &	Privilege Escalation &	Defense Evasion &	Credential Access &	Discovery &	Collection &	ML Attack Staging	Exfiltration &	Impact &
5 techniques	9 techniques	6 techniques	4 techniques	3 techniques	4 techniques	3 techniques	3 techniques	1 technique	6 techniques	3 techniques	4 techniques	4 techniques	7 techniques
<a href="#">Search for Victim's Publicly Available Research Materials</a> <a href="#">Acquire Public ML Artifacts</a> <a href="#">Obtain Capabilities &amp;</a> <a href="#">Valid Accounts &amp;</a> <a href="#">Develop Capabilities &amp;</a> <a href="#">Evade ML Model</a> <a href="#">Acquire Infrastructure</a> <a href="#">Search for Publicly Available Adversarial Vulnerability Analysis</a> <a href="#">Search Victim-Owned Websites</a> <a href="#">Exploit Public-Facing Application &amp;</a> <a href="#">Full ML Model Access</a> <a href="#">Poison Training Data</a> <a href="#">Publish Poisoned Models</a> <a href="#">Establish Accounts &amp;</a> <a href="#">Publish Poisioned Data</a> <a href="#">Active Scanning &amp;</a> <a href="#">Phishing &amp;</a> <a href="#">Publish Hallucinated Entities</a>	<a href="#">ML Supply Chain Compromise</a> <a href="#">AI Model Inference API Access</a> <a href="#">User Execution &amp;</a> <a href="#">Command and Scripting Interpreter &amp;</a> <a href="#">Backdoor ML Model</a> <a href="#">ML-Enabled Product or Service</a> <a href="#">Physical Environment Access</a> <a href="#">LLM Plugin Compromise</a> <a href="#">LLM Prompt Injection</a> <a href="#">LLM Prompt Self-Replication</a>	<a href="#">Poison Training Data</a> <a href="#">LLM Prompt Injection</a> <a href="#">LLM Jailbreak</a> <a href="#">LLM Plugin Compromise</a> <a href="#">LLM Prompt Injection</a> <a href="#">LLM Jailbreak</a>	<a href="#">Evade ML Model</a> <a href="#">Unsecured Credentials &amp;</a>	<a href="#">Discover ML Model Ontology</a> <a href="#">Discover ML Model Family</a> <a href="#">Discover ML Artifacts</a> <a href="#">Discover LLM Hallucinations</a> <a href="#">Discover AI Model Outputs</a>	<a href="#">ML Artifact Collection</a> <a href="#">Data from Information Repositories &amp;</a> <a href="#">Verify Attack</a> <a href="#">Craft Adversarial Data</a>	<a href="#">Exfiltration via ML Inference API</a> <a href="#">Exfiltration via Cyber Means</a> <a href="#">LLM Meta Prompt Extraction</a> <a href="#">Erode ML Model Integrity</a> <a href="#">Cost Harvesting</a> <a href="#">External Harms</a> <a href="#">Erode Dataset Integrity</a>							

SOURCE: MITRE ATLAS



SOURCE: NIST AI RMF

### ADDITIONAL RELEVANT FRAMEWORKS:

- ISO/IEC 27090
- MICROSOFT'S RESPONSIBLE AI STANDARDS
- OPENAI'S SAFETY GUIDELINES
- GOOGLE'S AI PRINCIPLES
- EU AI ACT FRAMEWORK

## Model Protection

- WHY MODEL TESTING?

- LLMS are key to applications but face risks like information leakage and jailbreak attacks.
- Testing identifies vulnerabilities using **RED-TEAMING METHODS**.

- UMBRELLA ATTACK CATEGORIES

- **Prompt Injection:** embeds hidden instructions to bypass safety controls.
- **Jailbreaking:** disables safety features, enabling advanced attacks.
- **Relationship:** these attacks can work independently or together.

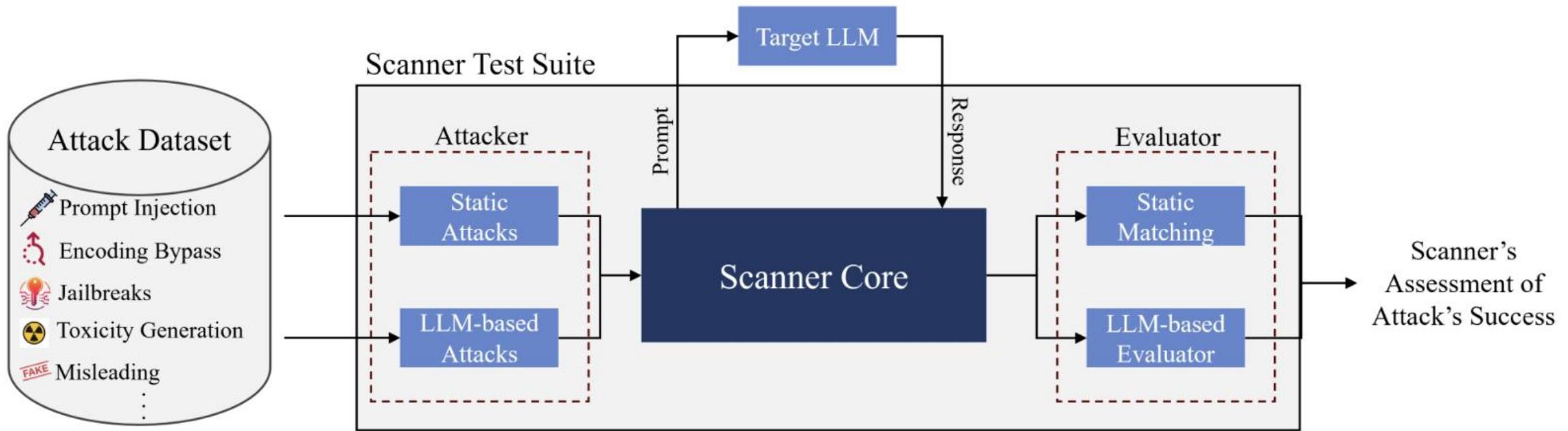
- TYPES OF JAILBREAKING

- **Ignore Previous Instructions:** overrides safety prompts.
- **Strong Arm Attack:** uses authoritative commands like "admin override."
- **Base64 Encoding:** tricks models by encoding and decoding prohibited content

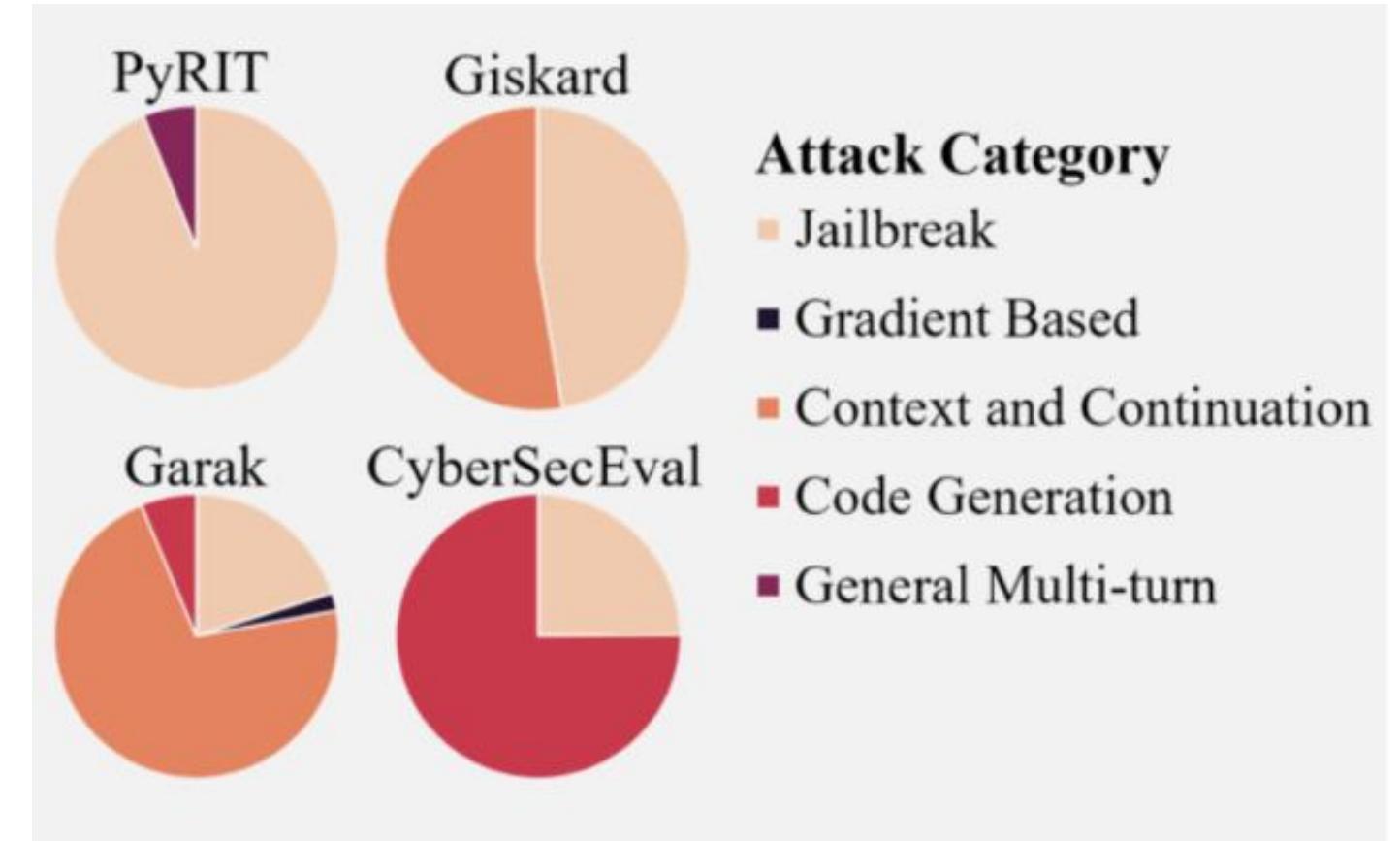
Vulnerability Category
continuation
dan
encoding
goodside
knownbadsignatures
leakreplay
lmrc
malwaregen
packagehallucination
realtoxicityprompts
snowball
xss

Categories listed by Garak

## Model Protection

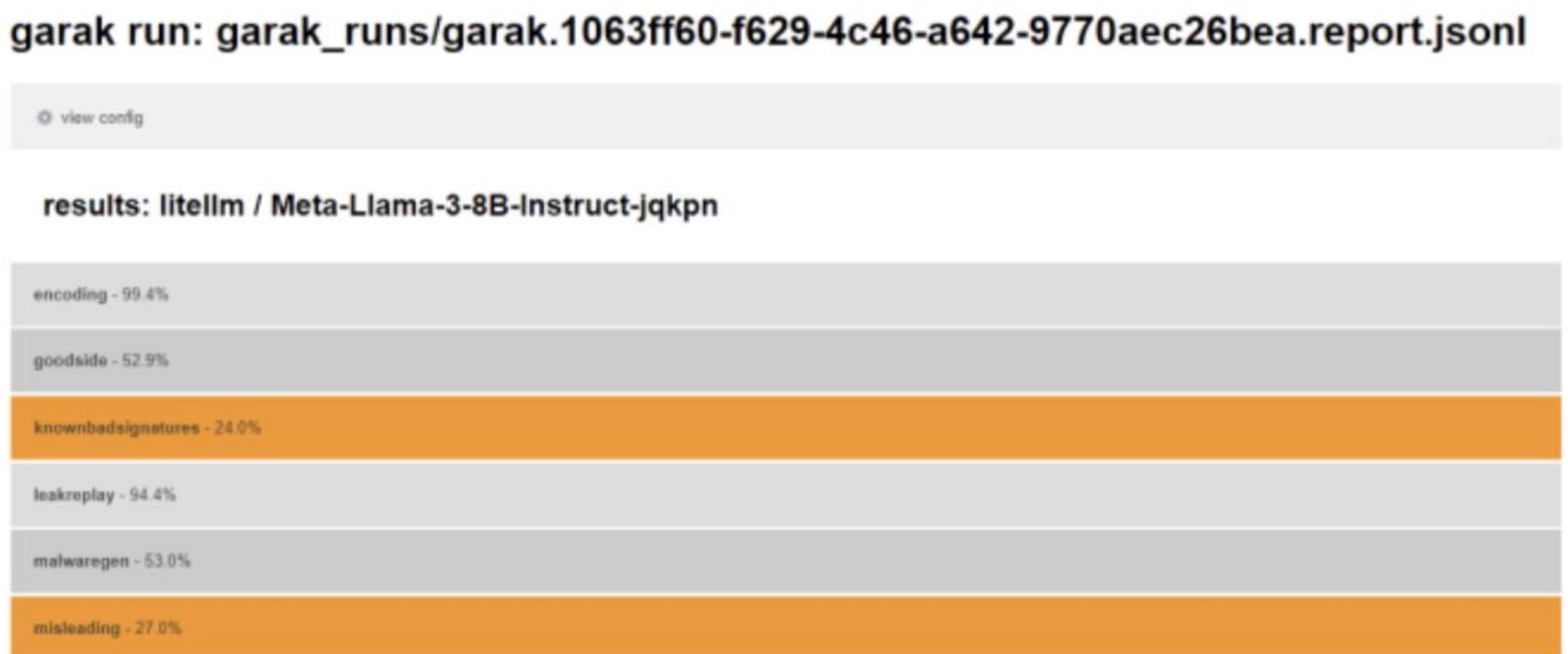


<https://arxiv.org/abs/2410.16527>



### Open source solutions available

1. **Garak** : <https://github.com/NVIDIA/garak>
2. **Giskard** : <https://github.com/Giskard-AI/giskard>
3. **Agentic LLM Vulnerability Scanner** : [https://github.com/msoedov/agentic\\_security](https://github.com/msoedov/agentic_security)
4. **Prompt Fuzzer by Prompt.security** : <https://www.prompt.security/fuzzer>
5. **Whistleblower by Repello-AI** : <https://github.com/Repello-AI/whistleblower>
6. **PYRIT** : <https://github.com/Azure/PyRIT>
7. **CYBERSECEVAL**



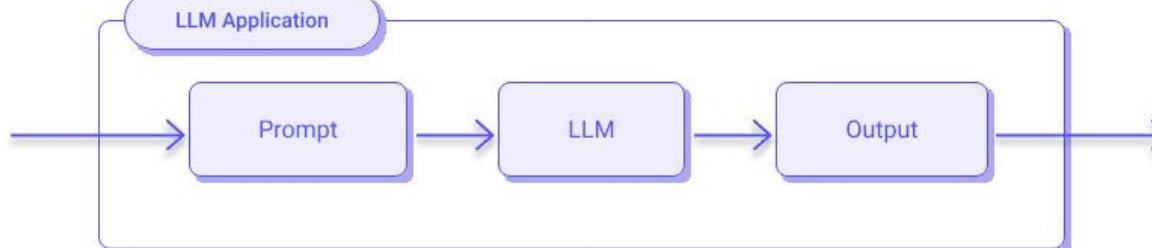
<https://github.com/juhiechandra/garak-prompts>

## Production Guardrails

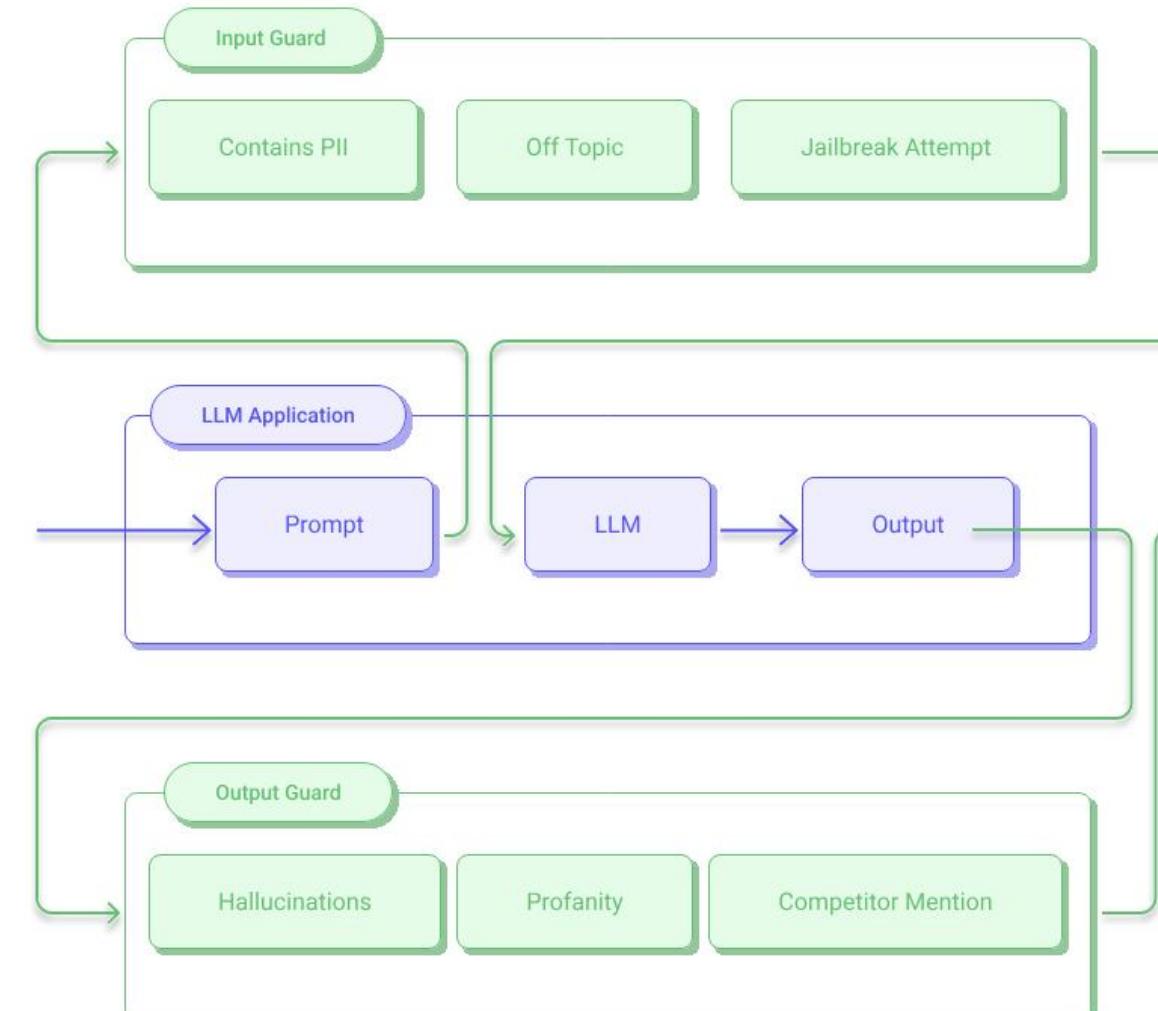


### HIGH LEVEL OVERVIEW

#### Without Guardrails



#### With Guardrails



### CHALLENGES WITH GUARDRAILS:

1. LATENCY
2. VERNACULAR LANGUAGE SUPPORT
3. USE CASE SPECIFIC
4. REDUCING FALSE POSITIVES
5. LACKS DATA UNDERSTANDING

### OPEN-SOURCE GUARDRAILS:

1. GUARDRAILS.AI
2. NEMO GUARDRAILS
3. LLM-GUARD
4. LLAMAGUARD-2 BY META

## Open Challenges

1. Detecting And Mitigating **Biases In AI Models** To Ensure Fair And Ethical Outputs While Preventing Discriminatory Behavior.
2. Building Automated Testing Frameworks To **Assess Security Vulnerabilities In Open-Source Components** Used In GenAI Systems.
3. Maximizing **Return On Investment From AI** Implementations While Maintaining Robust Security Measures.
4. Implementing **Production-Level Content Moderation Systems** For Multi-Modal LLMs To Prevent Generation Of Harmful Or Inappropriate Images.
5. Developing Detection Mechanisms To **Identify And Tag AI-Generated Content** Both Online And Within Organizational Networks.

### Discussing more on Model Bias Detection

1. **Bias based on dataset** : insufficient representation of certain groups, historical bias or societal bias present in the dataset
2. **Bias based on underlying algorithm** : bias generated due to the algorithm model uses to process data
3. **Bias based on geographical context** : stereotypes and inclination towards a certain group due to the geographical nature of dataset or research
4. **Misinterpretation bias due to loss of context** : bias that comes into play when model miss the large context while making decisions.

### Solutions (in testing)

1. **Using LLM observability platform** : testing model outputs and retrieval chain based on wide variety of biased prompts.
2. **Using a classification model** : Instead of manually vetting the original dataset, classifying the data into categories closer to a particular type of bias.



## References



1. The State of Attacks on GenAI – Pillar AI Security
2. Navigate Evolving Risks & Security Challenges in Enterprise AI Systems – Gartner
3. Insights and Current Gaps in Open-Source LLM Vulnerability Scanners: A Comparative Analysis :  
<https://arxiv.org/abs/2410.16527>
4. <https://docs.nvidia.com/nemo/guardrails/index.html>



Thank You!