Critical Vulnerabilities for LLM based Applications - Architecture and Implementation

From: Vivek Gunasekaran
University of Arkansas at Little Rock

Version 3.0
Apr/17/2023

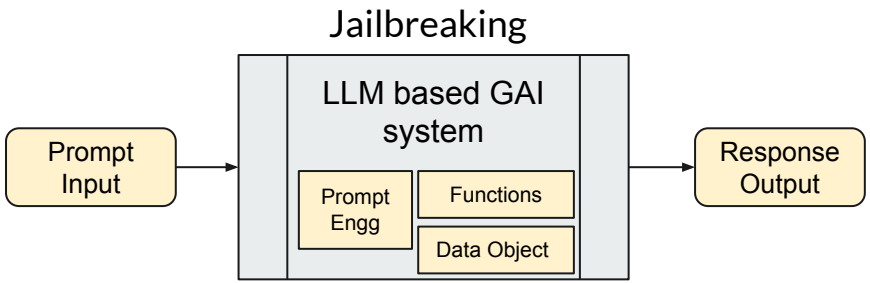
Index

Prompt Injection Vulnerability

- LLM limitation - No fully reliable way to prevent this attack within the LLM itself. Since LLM considers both Instructions and External data as input from the User.
- When attacker manipulates an LLM's operation through crafted inputs, resulting in the attacker's intention to get executed.
- LLM could act as "Confused Deputy" on behalf of the attacker.

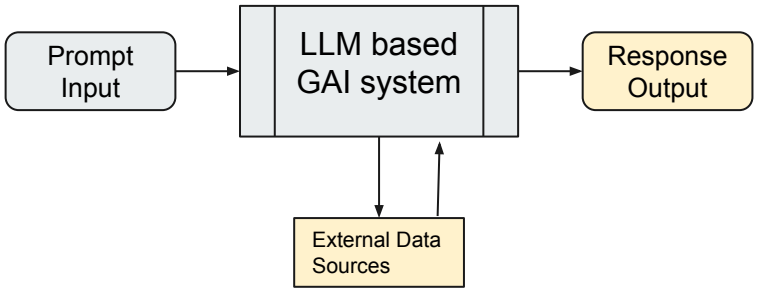
Direct Prompt Injection

Attacker overwrites or reveals the underlying system prompts resulting in the attacker interacting with insecure functions and data objects that are accessible by the LLM



Indirect Prompt Injection

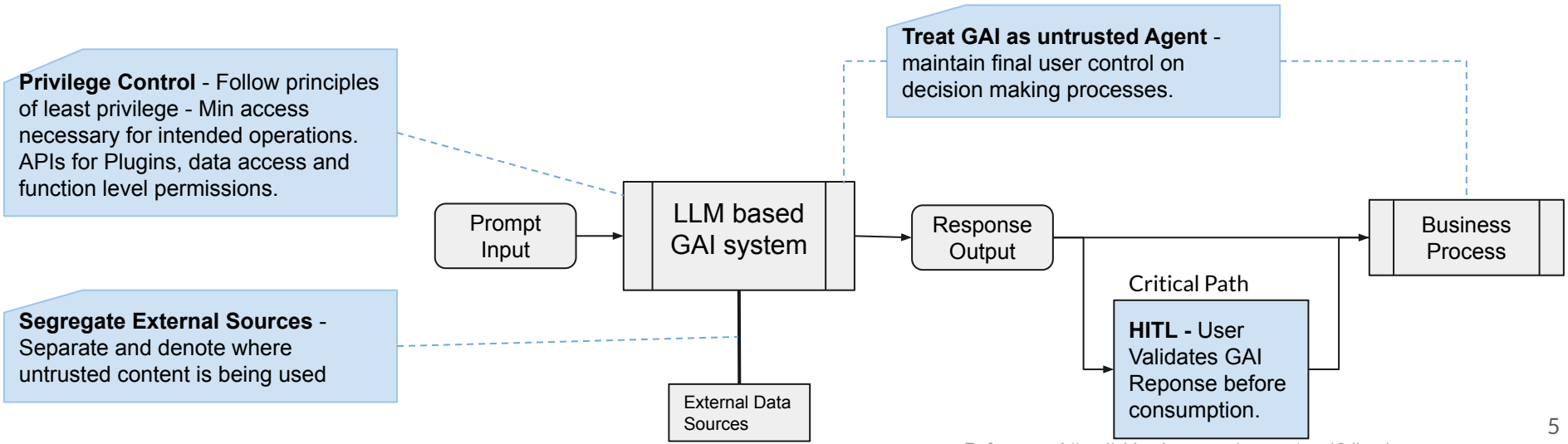
Occurs if the LLM accepts external source inputs that are controlled by the attacker resulting in the conversation being hijacked by the attacker.



Reference : https://gbhackers.com/owasp-top-10-llms/

How to prevent Prompt Injection

Trust controls can be placed outside of the system to mitigate the impact of prompt injection attempts.



Reference : https://gbhackers.com/owasp-top-10-llms/

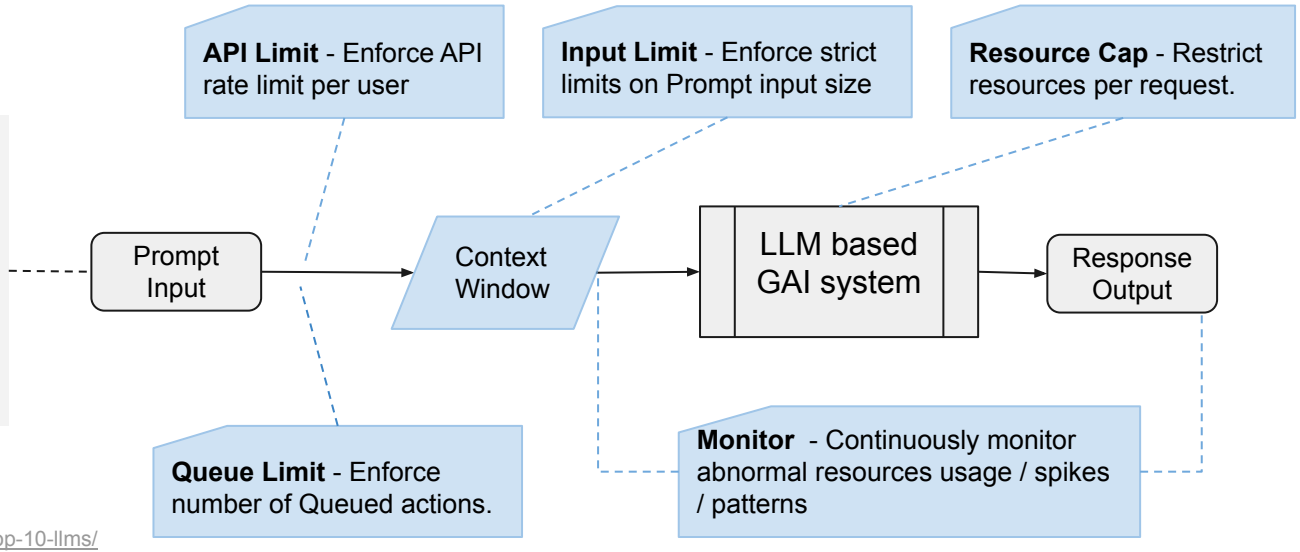
Model denial of Service Vulnerability

- An attacker interacts with the system in such a way that request consumes an exceptionally high amount of resources, which results in a decline in the quality of service for the users.
- Interfere or manipulate the context window of LLM

How to prevent Model Denial of Service

Prompt input validation for limits and sanitization for malicious requests.

- Continuous Input context overflow
- Repetitive long inputs
- Recursive context expansion
- Variable-length input flood

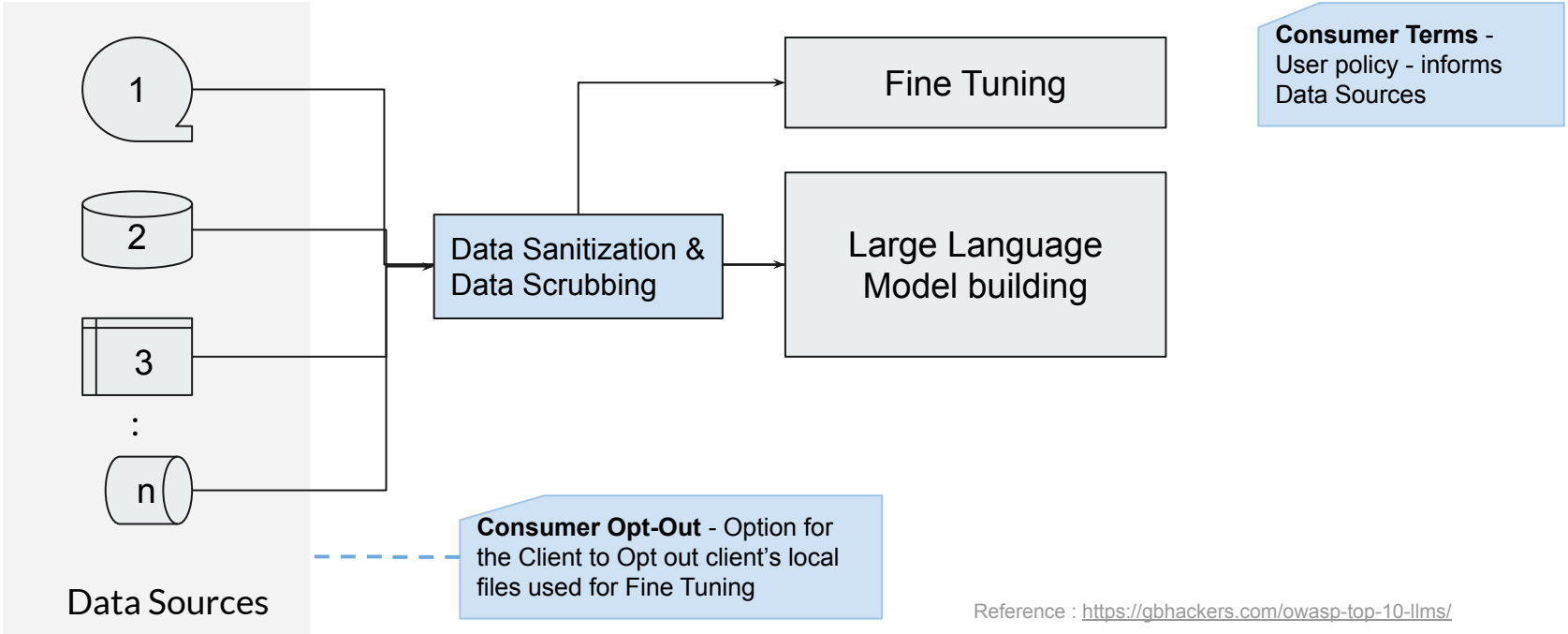


Reference : https://gbhackers.com/owasp-top-10-llms/

Sensitive Information Disclosure Vulnerability

- The system accidentally reveals sensitive information, proprietary algorithms, or other confidential details through its responses.

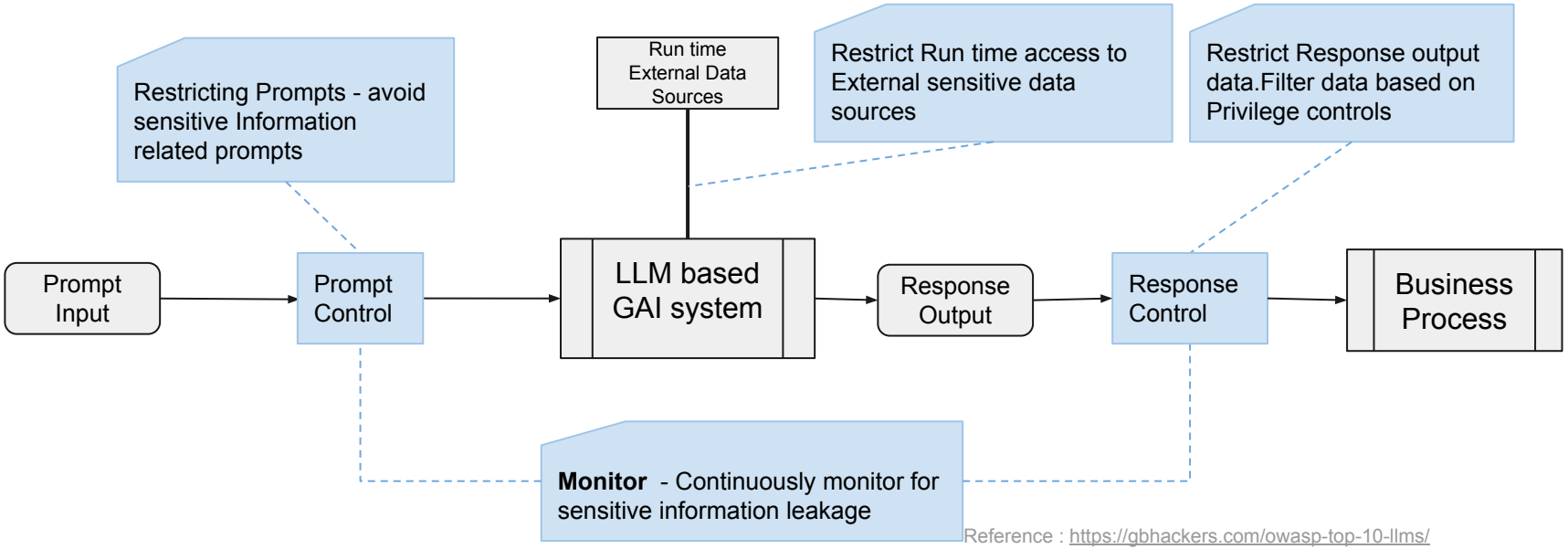
How to prevent Sensitive Information Disclosure



Reference : https://gbhackers.com/owasp-top-10-llms/

How to prevent Sensitive Information Disclosure

Restricting Prompt and controlling the Response



Reference : https://gbhackers.com/owasp-top-10-llms/

Training Data Poisoning Vulnerability

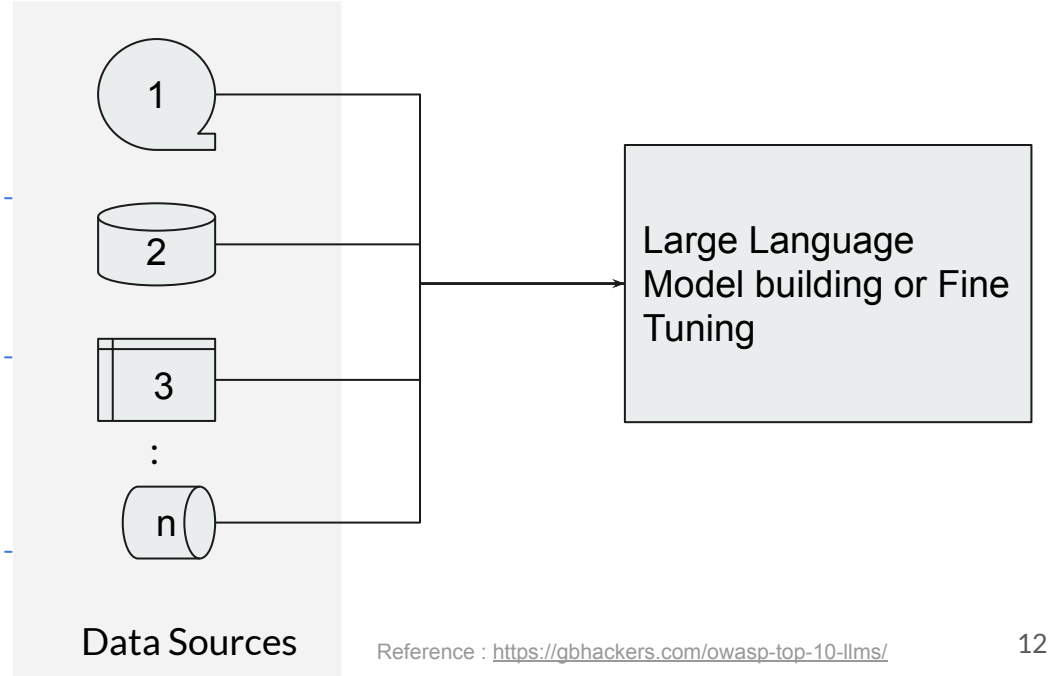
- Integrity attack.
- Tampering with the training data impacts the model's ability to produce correct output.
- Occurs when an attacker or unaware client of the LLM manipulates the training data or fine-tuning procedures of an LLM to introduce vulnerabilities, backdoors, or biases.

How to prevent Training Data Poisoning

Validate Source - Verify the supply chain of the training data, and maintain attestations of external sources

Verify Data Legitimacy - Legitimate data must be used for both training and fine-tuning.

HITL - Monitor continuously and alert when skewed response exceed threshold. Human in the loop to review response and audit source.



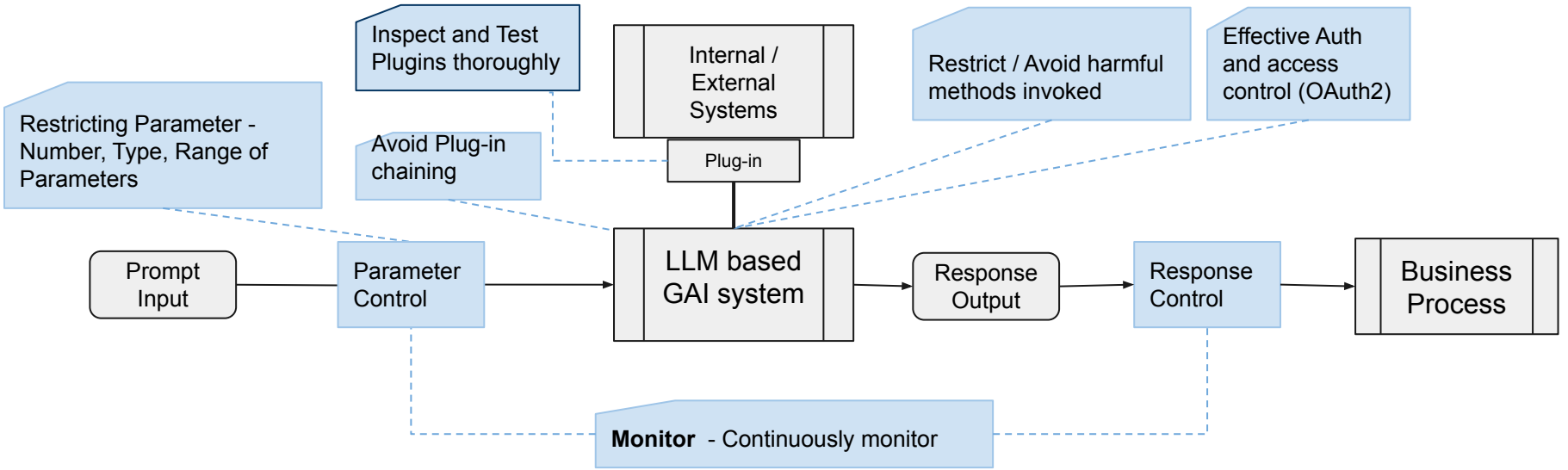
Reference : https://gbhackers.com/owasp-top-10-llms/

Insecure Plugin Vulnerability

- Plugins are extensions that are called by the model when responding to a user request.
- Poor Control - Since they are automatically invoked in-context and are often chained, there is little application control over their execution.

How to prevent Insecure Plugin Design

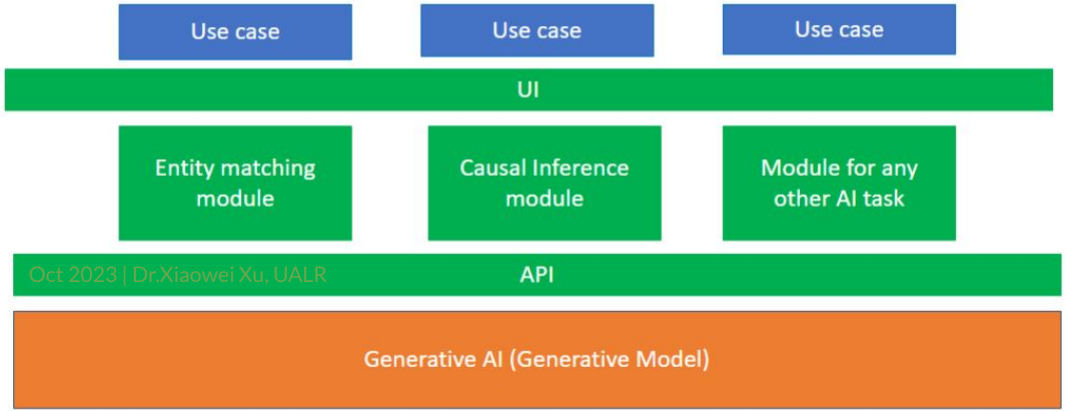
Enforcing strict parameter and controls



Reference : https://gbhackers.com/owasp-top-10-llms/

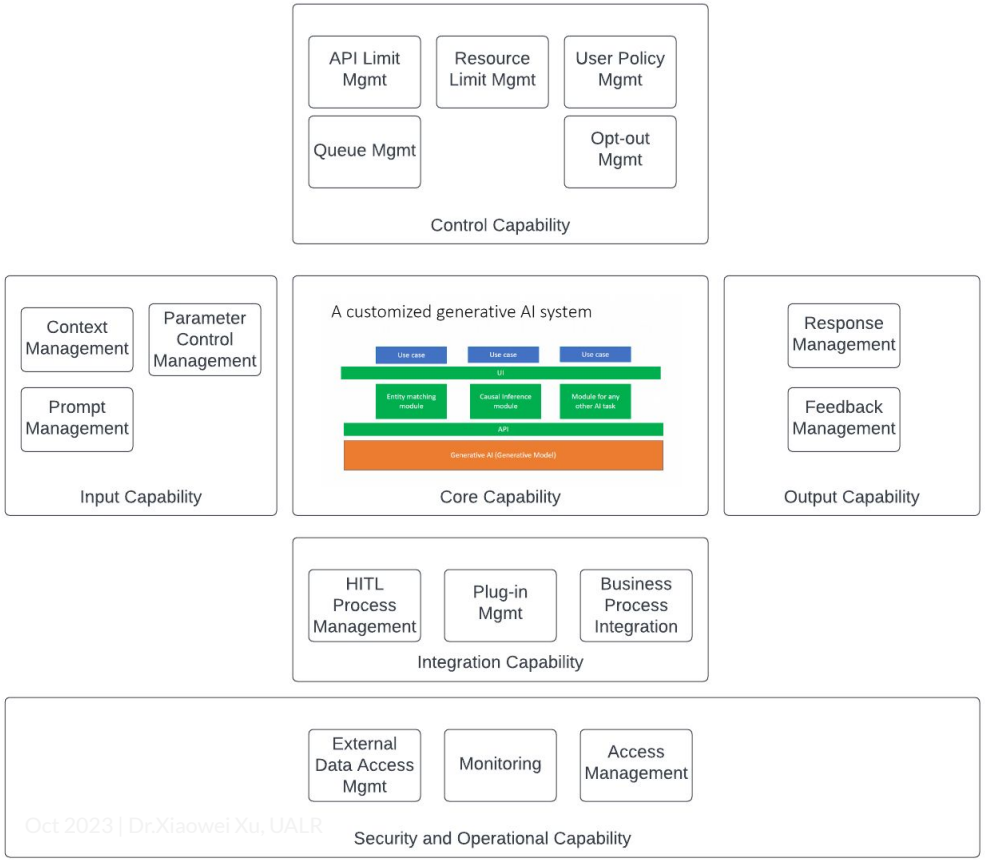
LLM Architecture

A customized generative AI system



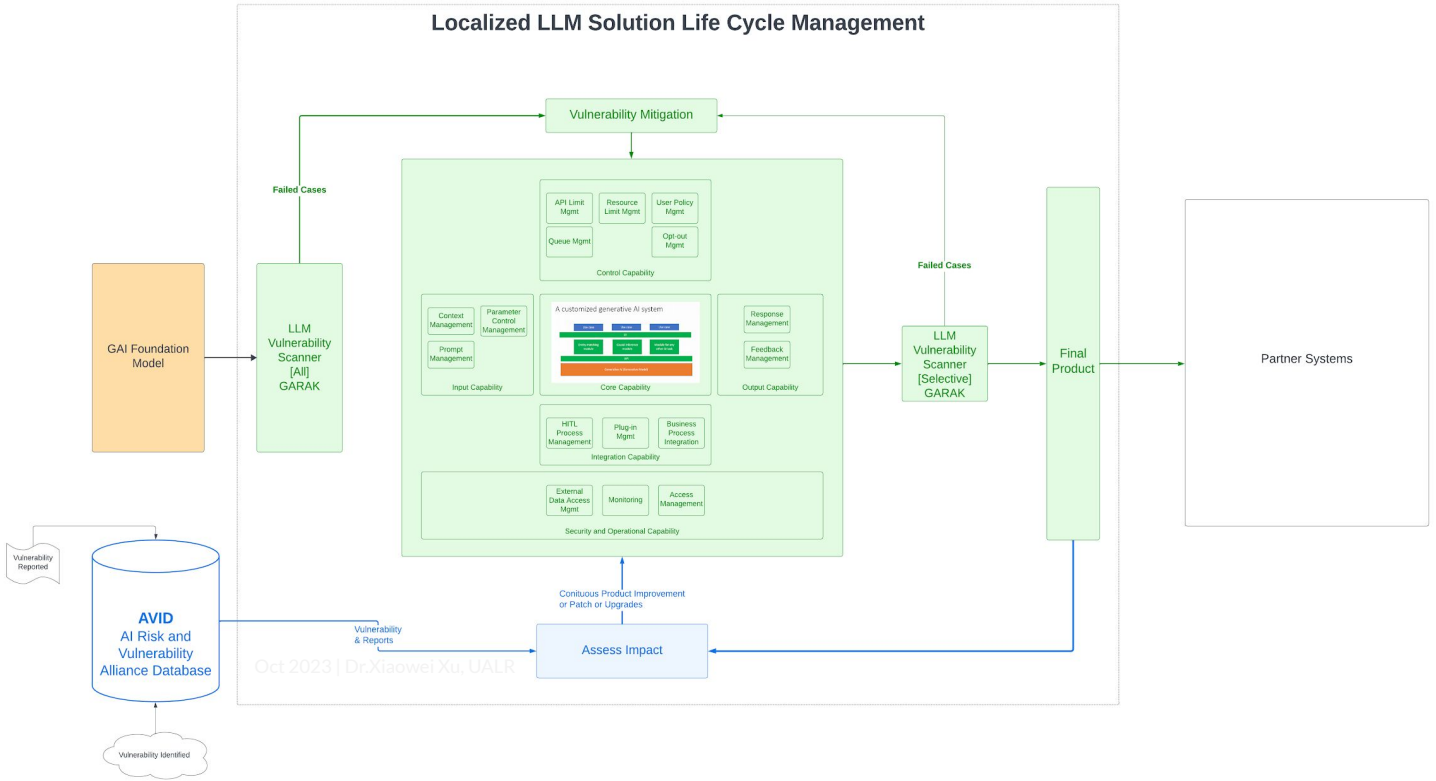
Reference : Dr.Xu presentation on LLM

Capability based system design addressing LLM Vulnerability



Mitigation - Vulnerability of LLM

Mitigating LLM Vulnerability



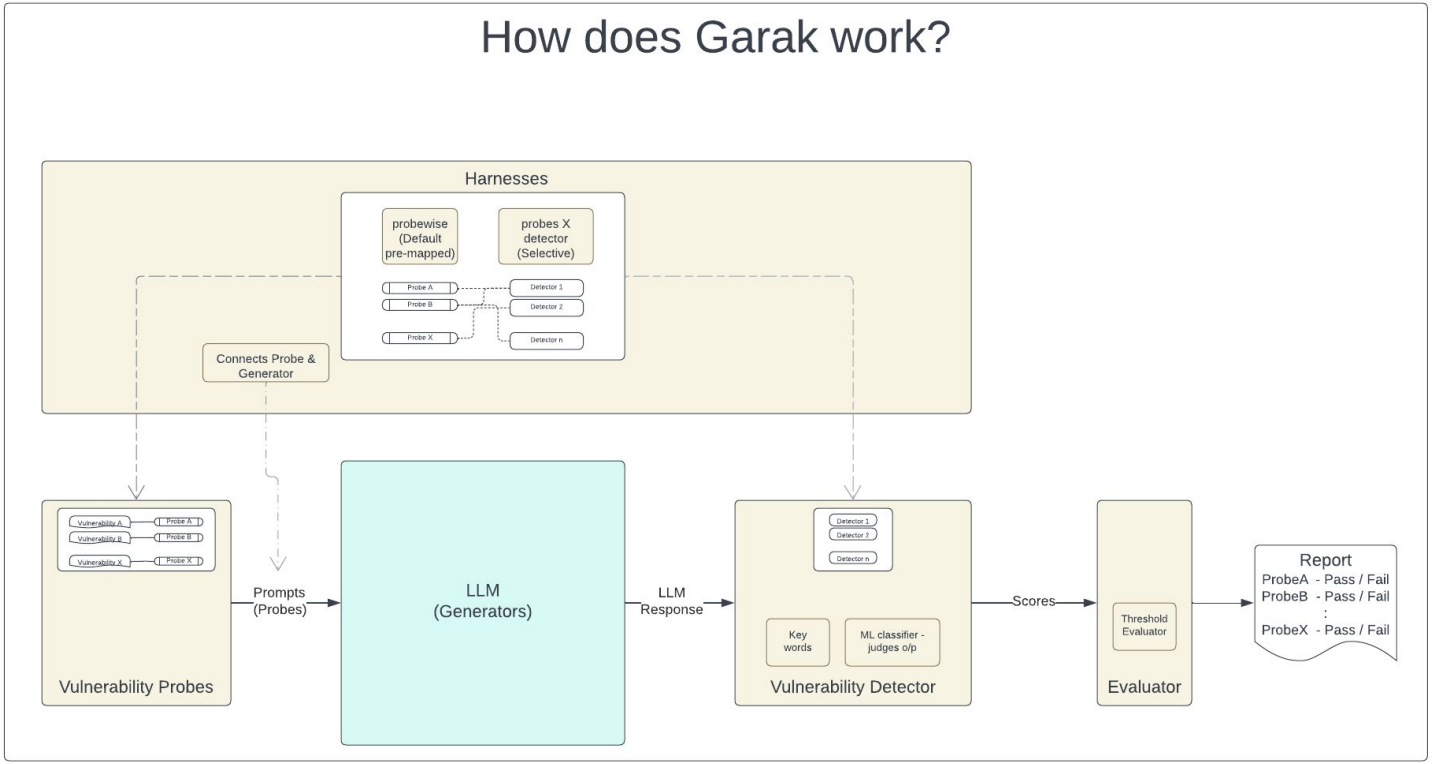
Thank You

For any further questions please contact

Vivek Gunasekaran (vgunasekaran@ualr.edu)

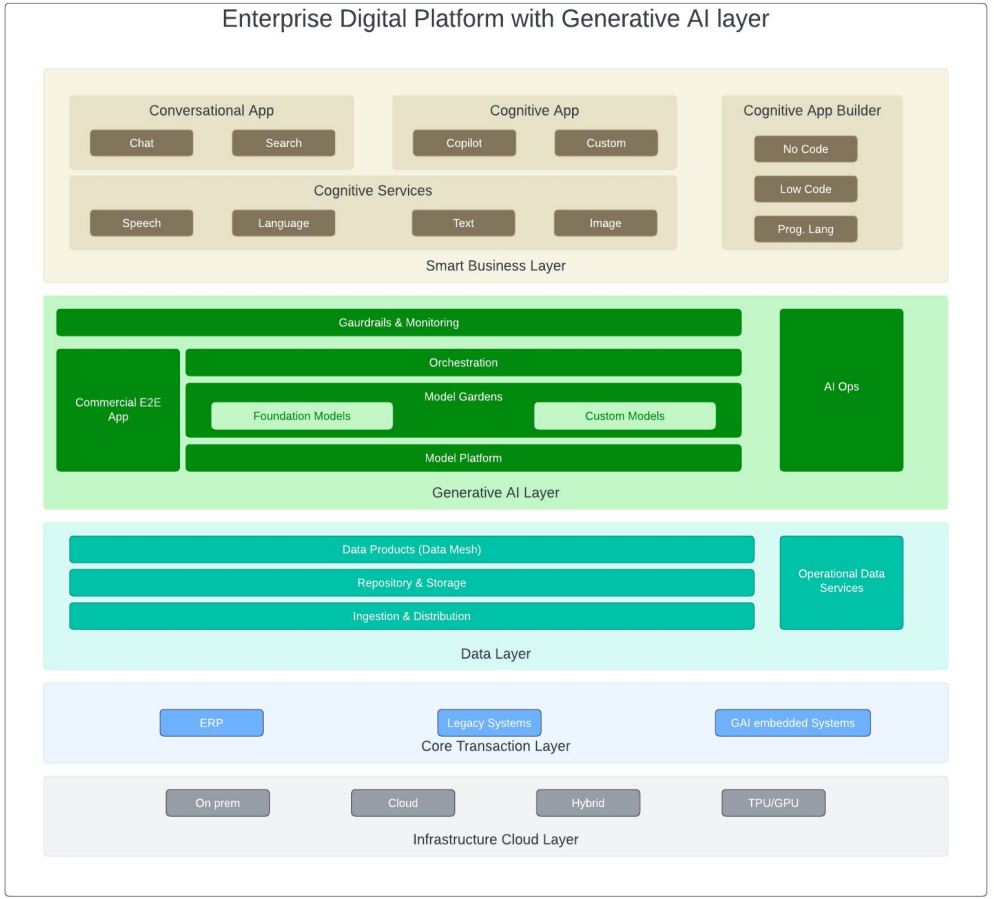
References

How Garak works?

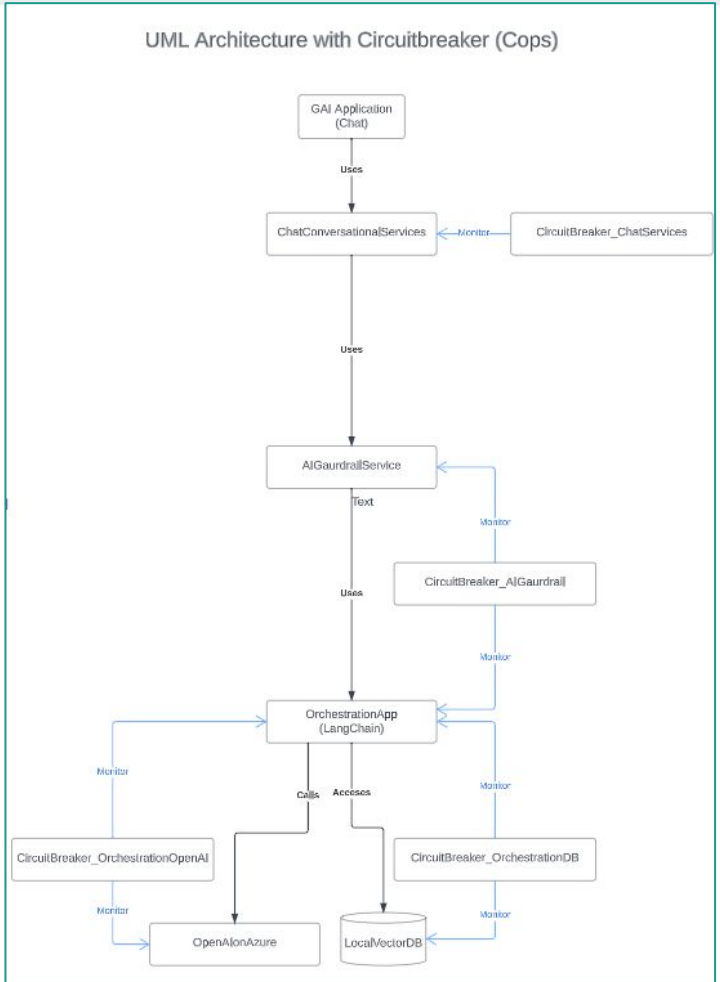


Reference : https://docs.garak.ai/garak/

GAI Digital Platform Architecture



Guardrails with Circuit Breaker Architecture



LLM Deployment Risk Assessment