BEYOND THE BASICS

THE ROLE OF LLMS IN THREAT INTELLIGENCE

Thomas Roccia | **@fr0gger_**
Roberto Rodriguez | **@Cyb3rWard0g**

SANS

WHO ARE WE?

Thomas Roccia

Sr. Threat Researcher at Microsoft

@fr0gger_

http://SecurityBreak.io

Roberto Rodriguez

Principal Security Researcher and Manager at Microsoft

@Cyb3rWard0g

https://blog.openthreatresearch.com/

IMPORTANT

DISCLAIMER

The presentation/slides/information we share today represent our own personal views. We are speaking for ourselves and not on behalf of Microsoft Corporation

WHAT WILL BE COVERED?

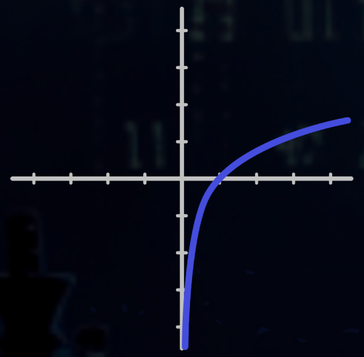
- ✓ Threat Intelligence Challenges
- ✓ Fundamentals of Generative AI
- ✓ Challenges and Pitfalls of LLMs
- ✓ Practical Usecases for TI
- ✓ What next?

THREAT INTELLIGENCE CHALLENGES

Time-consuming understanding

Overwhelming data

New analysts' learning curve



Fast-evolving threats

Leverage AI for enhanced capabilities.

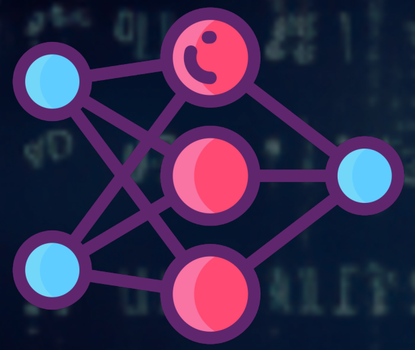
FUNDAMENTALS OF GENERATIVE AI

Back to the basics!

ARTIFICIAL INTELLIGENCE (AI)

ARTIFICIAL

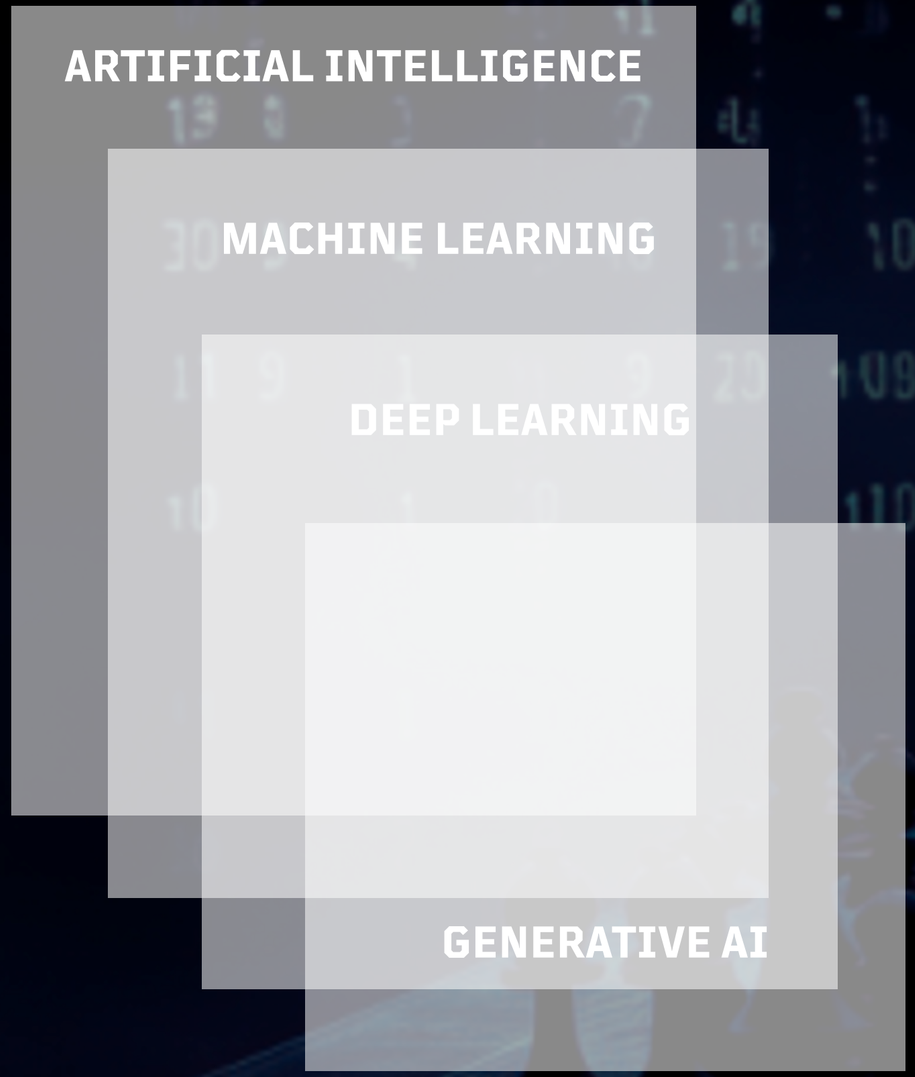
Algorithms that enable machines to simulate or replicate human-like intelligence



INTELLIGENCE

The ability to process information and make decisions or take actions to achieve a desired outcome

WHERE ARE WE?



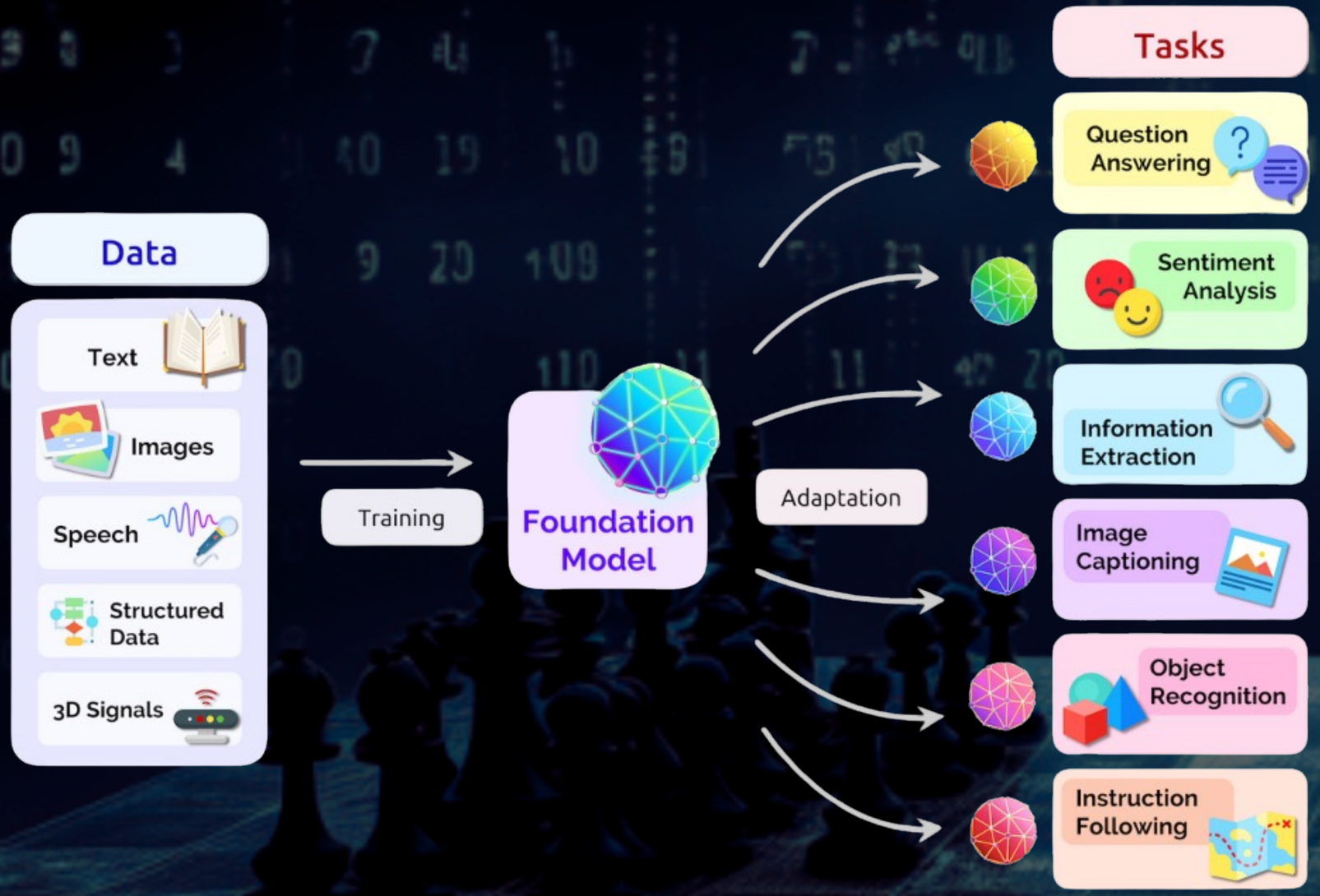
1950's — ARTIFICIAL INTELLIGENCE

1990's — MACHINE LEARNING

2010's — DEEP LEARNING

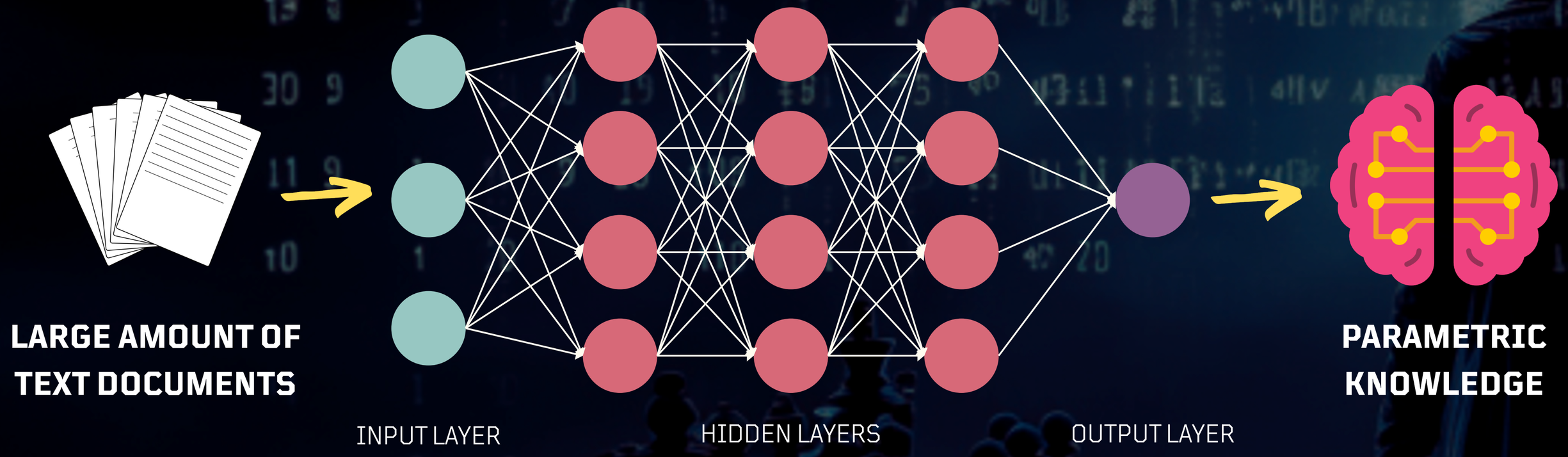
2020's — FOUNDATION MODELS (GENERAL PURPOSE)

GENERAL PURPOSE AI MODELS



HTTPS://ARXIV.ORG/ABS/1706.03762

LARGE LANGUAGE MODEL (LLM)

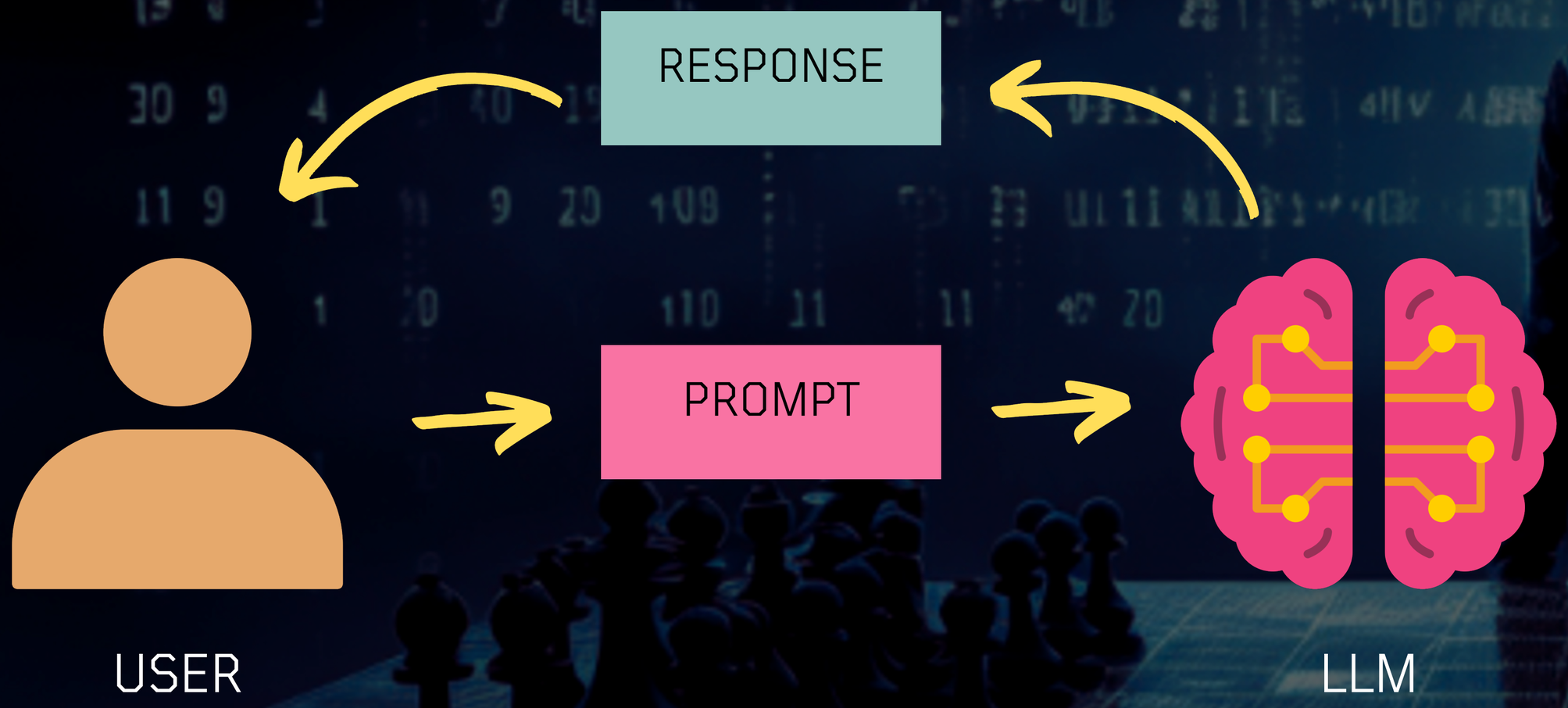


"Parameters represent the model's understanding of the statistical relationships between words and phrases in a language (Knowledge gained during training)"

ORGANIZING OUT OF THE BOX NLP ABILITIES

NLP Area	Task
Natural Language Understanding (NLU)	Entity Recognition
	Sentiment Analysis
	Language Parsing
Natural Language Generation (NLG)	Text Summarization
	Dialogue Systems
	Text Generation
	Text Translation

PROMPT-BASED INTERACTIONS



ENCODING LANGUAGE FOR A LLM

1 TOKENIZATION

THE PAYLOAD IS MALICIOUS → TOKENIZER → THE | PAYLOAD | IS | MALICIOUS →

ID	Token
5	The
3	payload
7	is
1	malicious

ENCODING LANGUAGE FOR A LLM

2 EMBEDDING MAPPING

EVERY WORD/TOKEN IN THE MODEL VOCABULARY HAS A vector **word embedding**!

THE

PAYLOAD

IS

MALICIOUS

INPUT TOKENS

ID	Vector Embedding		
5	0.2	0.3	0.4
3	0.4	0.5	0.1
7	0.5	0.1	0.4
1	0.3	0.7	0.6

MODEL TOKEN EMBEDDINGS

LLM

PROMPT ENGINEERING

- **What is a "Prompt"?**
  - A question or instruction we give to LLMs.
- **What is "Prompt Engineering"?**
  - Crafting clear, structured inputs for effective LLM responses.
- **Prompting Principles:**
  - Write clear and specific instructions
  - Give the model time to "think"

ROLE	YOU ARE A HELPFUL THREAT INTELLIGENCE ANALYST
INSTRUCTION	LIST THE TOP 5 PHISHING TECHNIQUES USED BY THREAT ACTORS ATTACKING THE FINANCIAL SECTOR IN 2023
CONTEXT	USE THESE NEW THREAT INTEL REPORTS FROM 2023 …
OUTPUT	PROVIDE YOUR OUTPUT ONLY AS A JSON OBJECT FOLLOWING THIS SCHEMA …

HTTPS://LEARN.DEEPLEARNING.AI/CHATGPT-PROMPT-ENG/LESSON/2/GUIDELINES

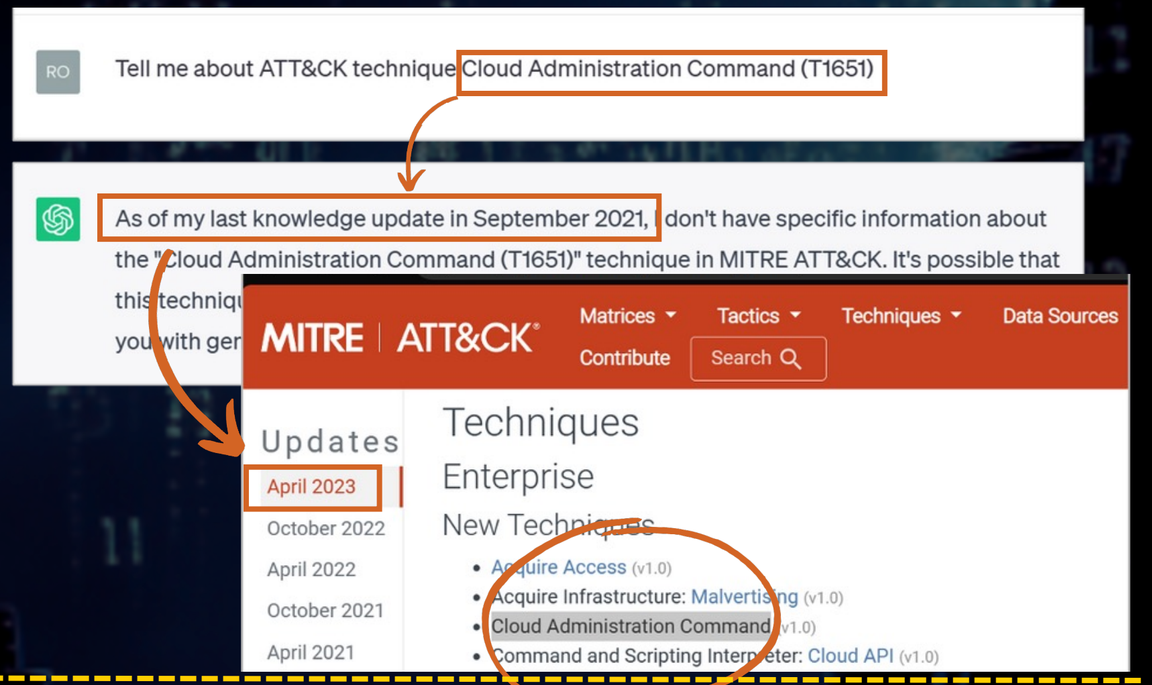
LLMS CHALLENGES

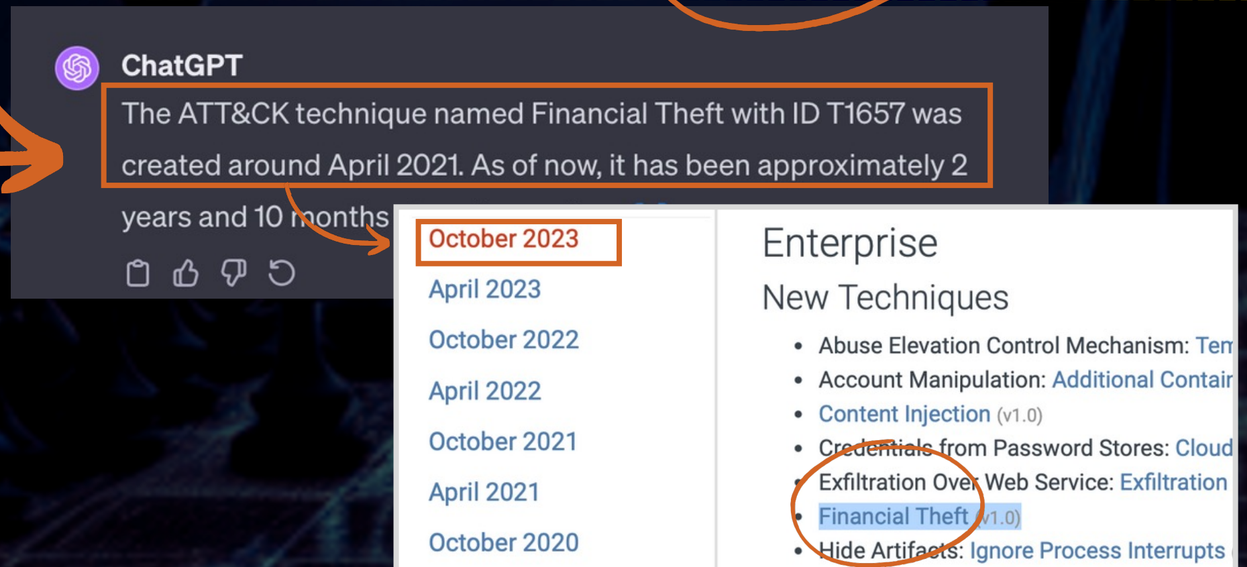
Personalization: The inability to provide private or personalized data.

Timeliness: How current the information is?

Hallucinations: Content that lacks factual basis due to guessing.

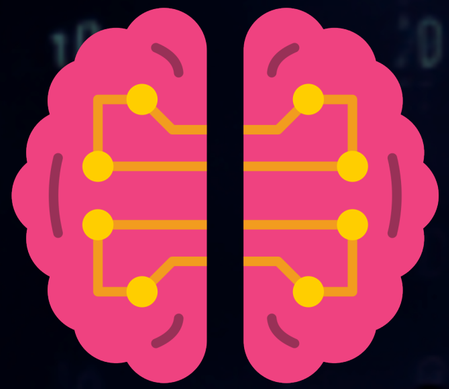
Knowledge Coverage: How much LLMs know about different topics.know about different topics?





ENTER RETRIEVAL AUGMENTED GENERATION (RAG)

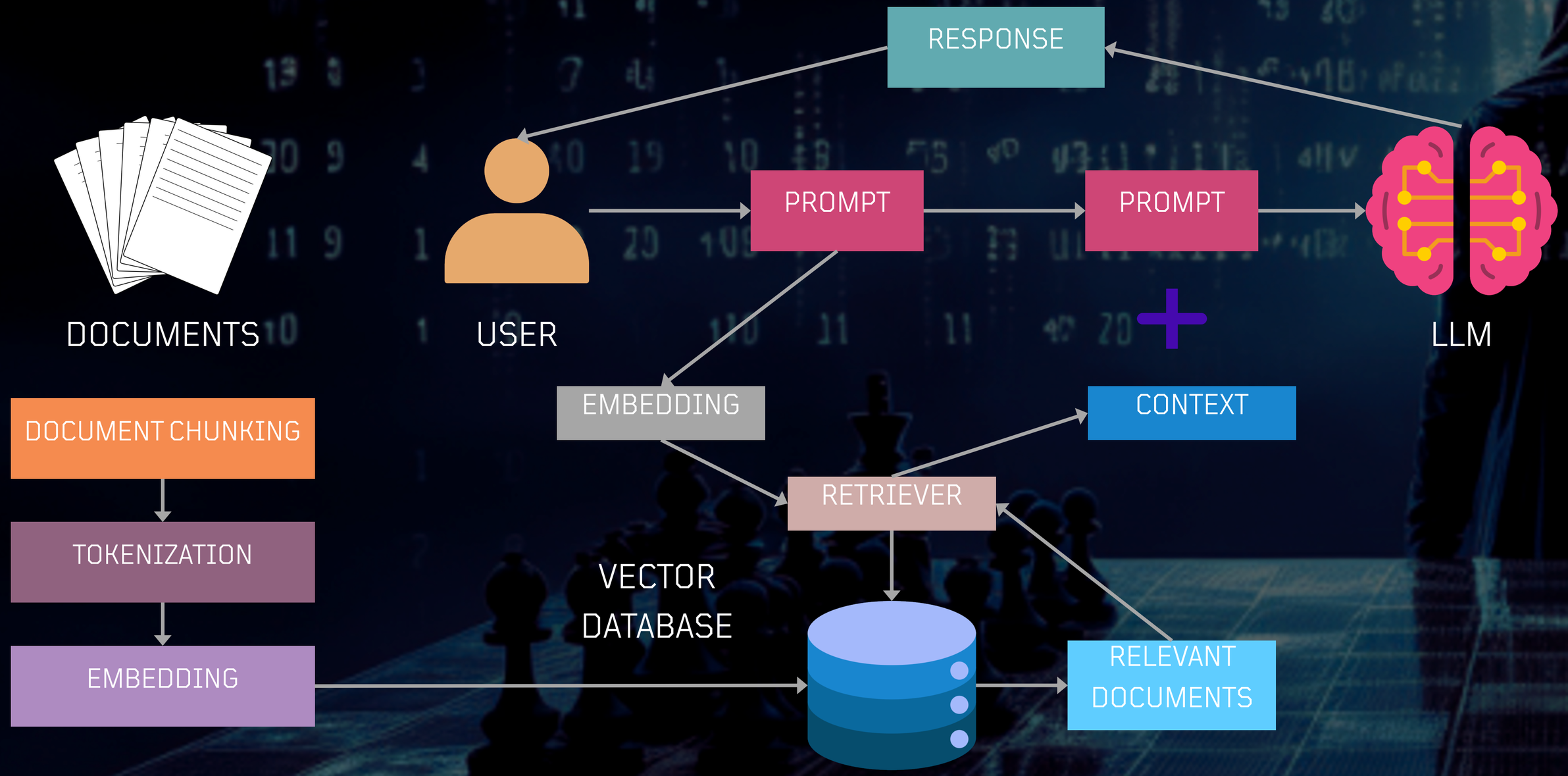
"A TECHNIQUE THAT COMBINES **INFORMATION RETRIEVAL** WITH **TEXT GENERATION,** ALLOWING A LANGUAGE MODEL TO USE EXTERNAL INFORMATION AS ADDITIONAL CONTEXT."



PARAMETRIC KNOWLEDGE

SOURCE KNOWLEDGE

RAG - FLOW EXAMPLE



AGENTS? AI AGENTS?

AGENT

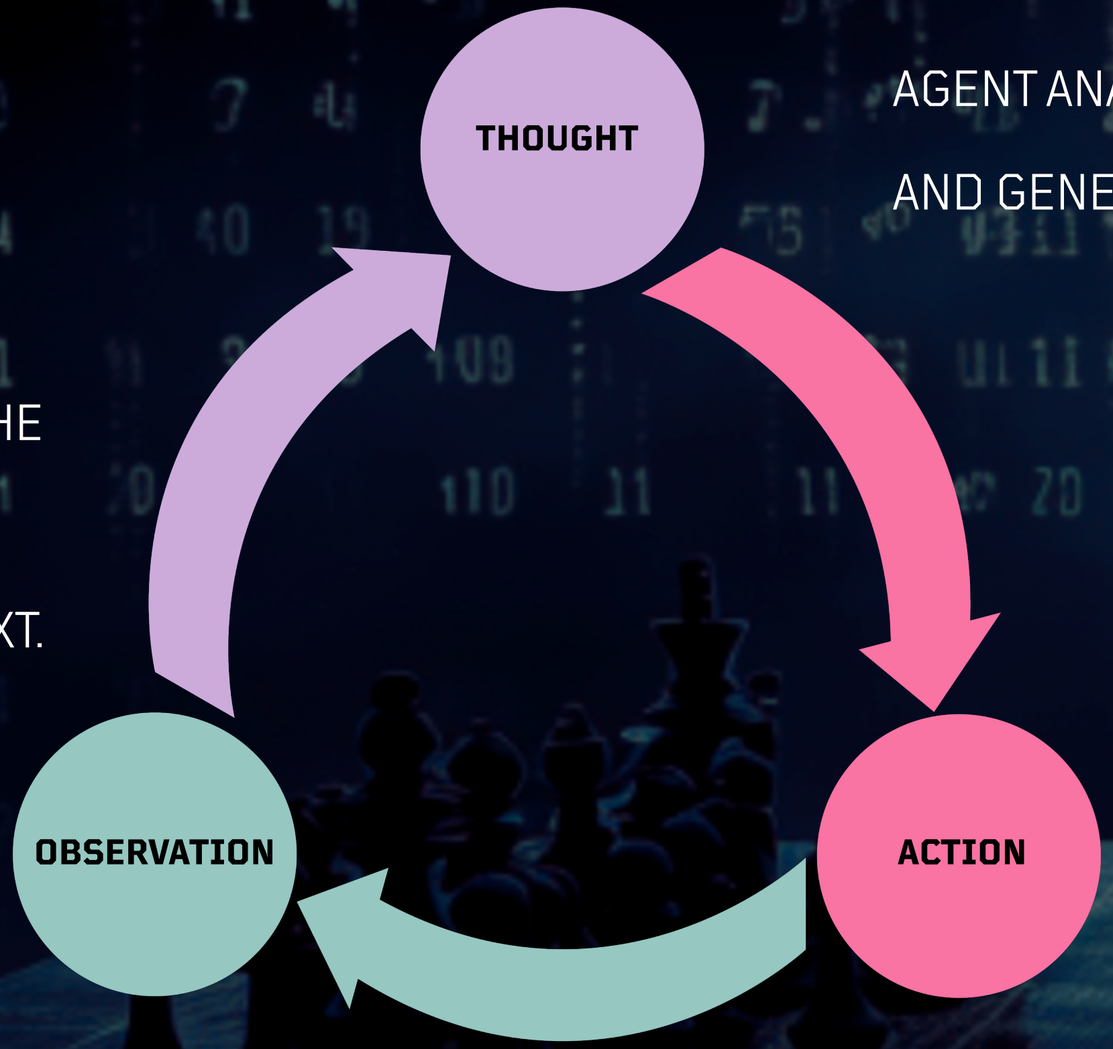
ANYTHING THAT CAN ACT ON ITS OWN OR MAKE DECISIONS

AI AGENT

A COMPUTER PROGRAM OR MACHINE THAT CAN ACT ON ITS OWN, MAKING DECISIONS SIMILAR TO HOW HUMANS DO.

IT CAN USE TOOLS TO EXECUTE ACTIONS AND GATHER NEW INFORMATION, ENHANCING ITS DECISION-MAKING PROCESS

REACT (REASON + ACT)



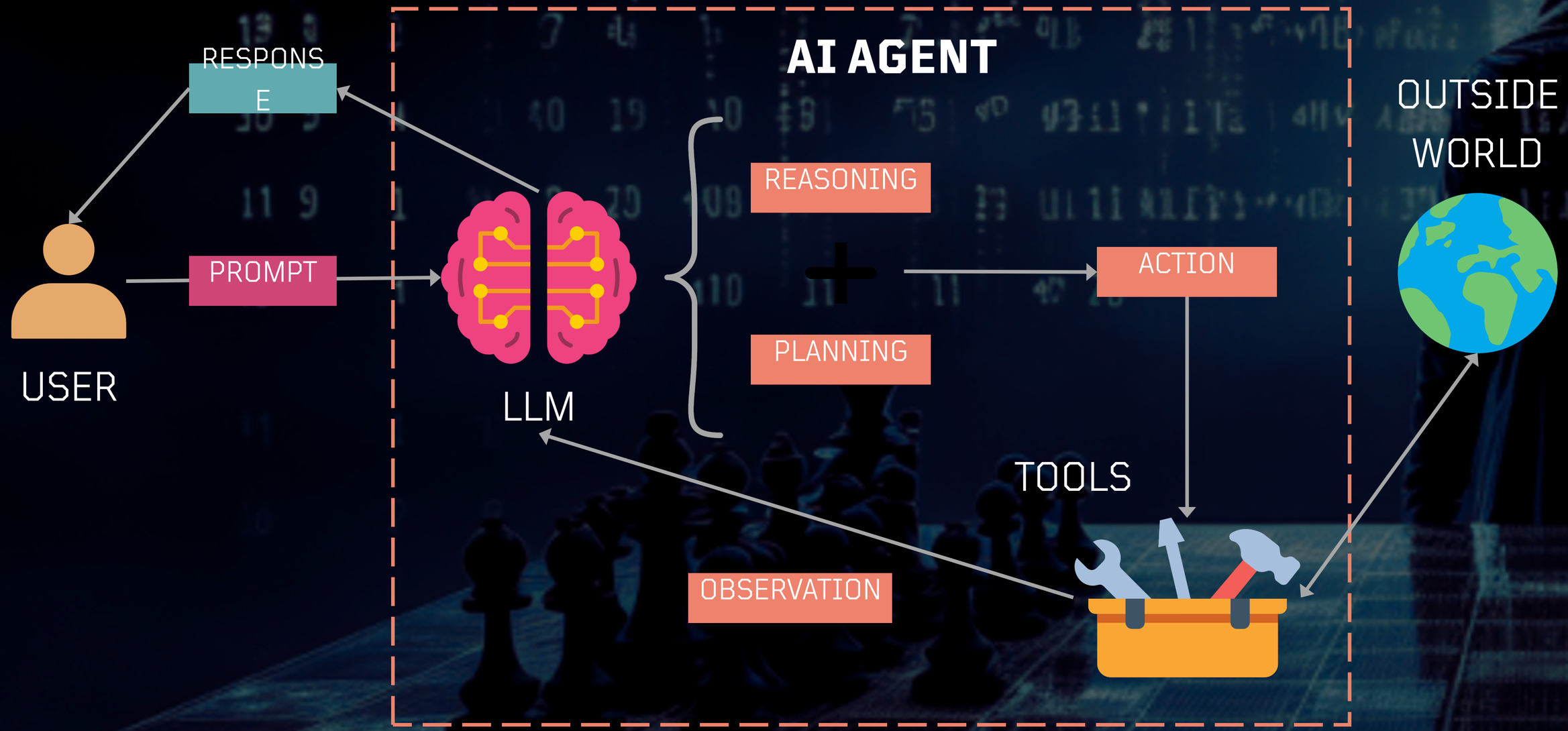
AGENT ANALYZES SITUATION AND GENERATES A THOUGHT.

BASED ON ITS THOUGHT, THE AGENT THEN TAKES AN ACTION.

AGENT OBSERVES THE OUTCOME, PROCESSES THE INFORMATION AND DECIDES WHAT TO DO NEXT.

HTTPS://ARXIV.ORG/PDF/2210.03629.PDF

LLM-BASED AGENTS - BASIC IDEA



LLMS AND THREAT INTEL
PRACTICAL USECASES

How you can leverage LLMs for your work

THREAT REPORT SUMMARIZATION USING FEW SHOT PROMPTING

What is Few-Shot prompting?

- Guides ML models using example prompts for task understanding.

Zero-Shot vs Few-Shot

- **Zero-Shot**: No example prompts.
- **Few-Shot**: Uses example prompts for clarity.

Why Use Few-Shot Prompting?

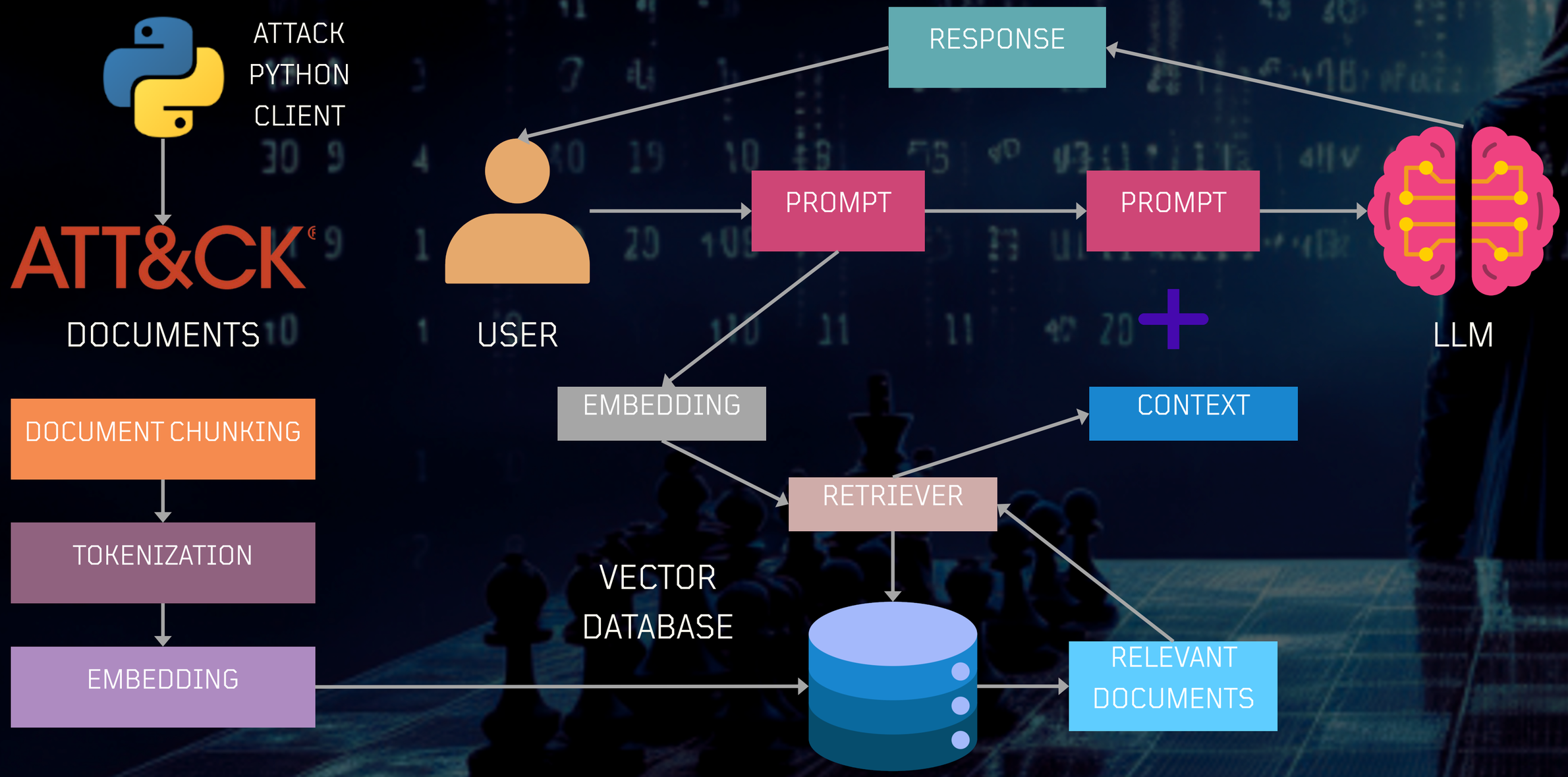
- Enhances task-specific accuracy.
- Ideal for complex or nuanced tasks.

```
[
    {
        "role": "system",
        "content": "Create a mindmap for threat analysts to organize
         key findings. Follow these guidelines:\n1. No hyphens in
         text.\n2. Max of four primary nodes.\n3. Use icons for
         readability.\n4. Single parentheses for rounded nodes."
    },
    {
        "role": "user",
        "content": "Your threat report content here."
    },
    {
        "role": "assistant",
        "content": "mindmap\nroot(Threat Report)\n  (Primary Node 1)\n
        ::icon(icon-name)\n    (Sub-node)\n  (Primary Node 2)\n
        ::icon(icon-name)\n    (Sub-node)\n  (Primary Node 3)\n
        ::icon(icon-name)\n    (Sub-node)\n  (Primary Node 4)\n
        ::icon(icon-name)\n    (Sub-node)"
    },
    {
        "role": "user",
        "content": "Your next input here."
    }
]
```

THREAT REPORT SUMMARIZATION
USING FEW SHOT PROMPTING

Source: https://store.securitybreak.io/theintelbrief

RAG AND ATT&CK



MSTICPY + AGENT = 🤯

```python
from msticpy.sectools.tilookup import TILookup
from langchain.chat_models import ChatOpenAI

llm = ChatOpenAI(model_name="gpt-4", temperature=0.3)

class TIVTLookup:
    def __init__(self):
        self.ti_lookup = TILookup()

    def ip_info(self, ip_address: str) -> str:
        result = self.ti_lookup.lookup_ioc(observable=ip_address,
        ioc_type="ipv4", providers=["VirusTotal"])
        details = result.at[0, 'RawResult']
        sliced_details = str(details)[:3500]
        return sliced_details

    def communicating_samples(self, ip_address: str) -> str:
        domain_relation = vt_lookup.lookup_ioc_relationships(observable =
        ip_address, vt_type = 'ip_address', relationship =
        'communicating_files', limit = "10")
        return domain_relation

    def samples_identification(self, hash: str) -> str:
        hash_details = vt_lookup.get_object(hash, "file")
        return hash_details
```

```python
ti_tool = TIVTLookup()

tools = [
    Tool(
        name="Retrieve_IP_Info",
        func=ti_tool.ip_info,
        description="Useful when you need to look up threat intelligence
        information for an IP address.",
    ),
    Tool(
        name="Retrieve_Communicating_Samples",
        func=ti_tool.communicating_samples,
        description="Useful when you need to get communicating samples from
        an ip or domain.",
    ),
    Tool(
        name="Retrieve_Sample_information",
        func=ti_tool.samples_identification,
        description="Useful when you need to obtain more details about a
        sample.",
    ),
]
```

MSTICPY
AI ASSISTANT

How can I assist your threat investigation today?

Enter your prompt:

Run investigation

TAKE AWAY

- ✓ We discussed the fundamentals of Generative AI
- ✓ Exploring prompt engineering, RAG and Agents
- ✓ Practical Generative AI application to Threat Intelligence
- ✓ The possibilities are endless!

https://blog.openthreatresearch.com/demystifying-generative-ai-a-security-researchers-notes/
https://blog.securitybreak.io/applying-llms-to-threat-intelligence-f3b8ba4463a4

BUT WE JUST SCRATCHED
THE SURFACE!

THANK YOU

Thomas Roccia | **@fr0gger_**
Roberto Rodriguez | **@Cyb3rWard0g**

SANS