

From Words to Action: Harnessing GenAI for Real Text-to-J1939 Applications



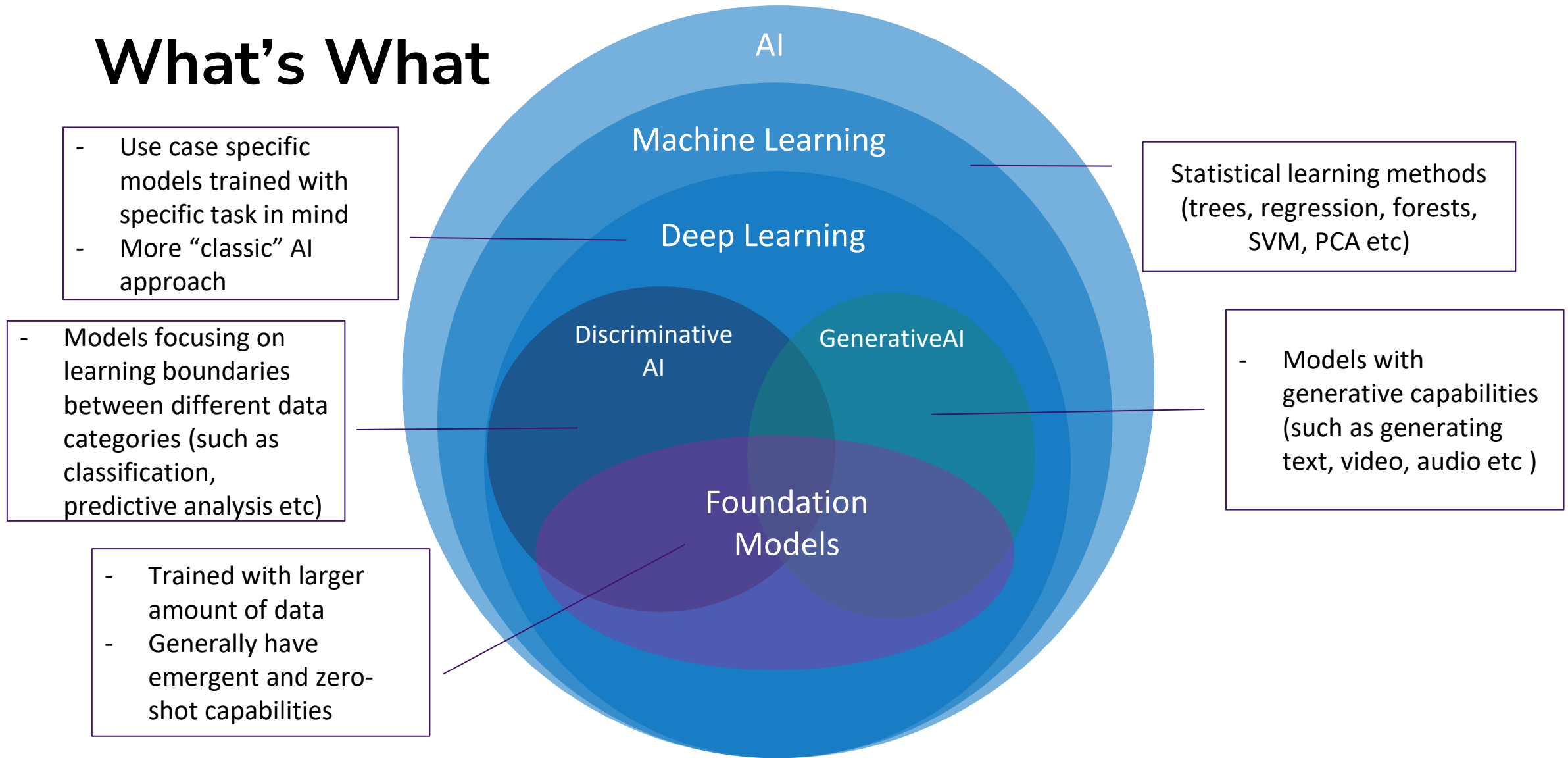
Ivan Granero

Senior Security Engineer,
Bosch

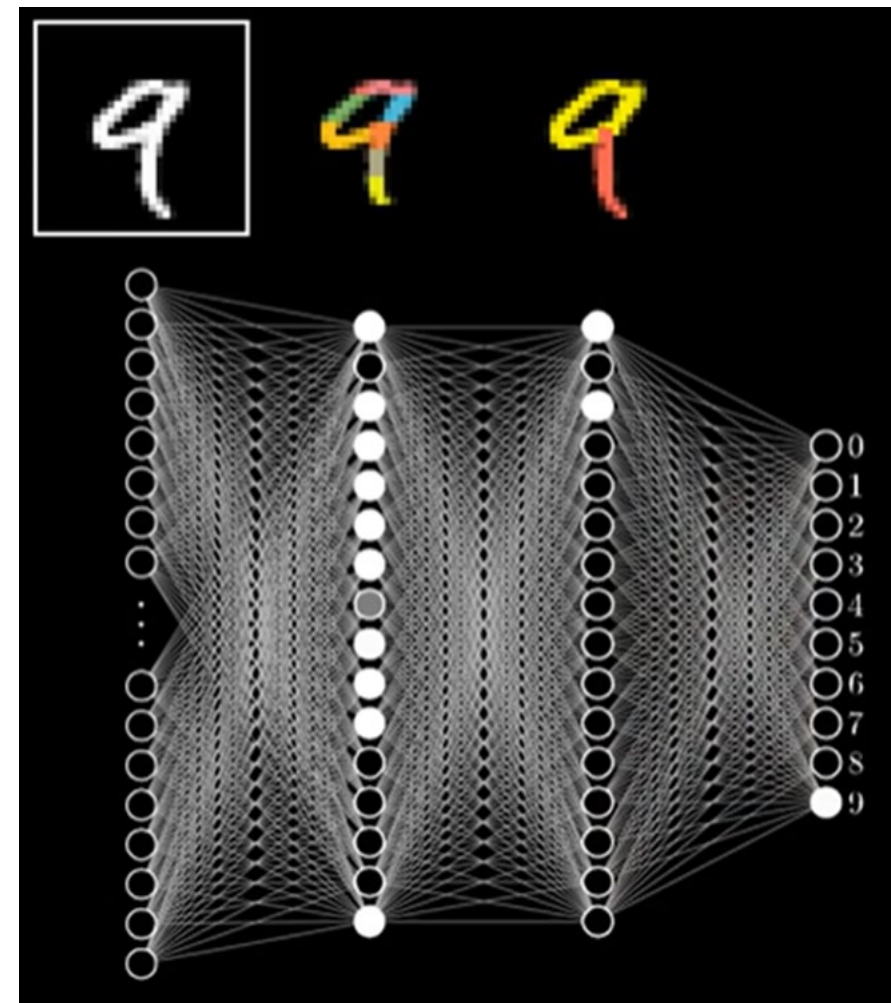
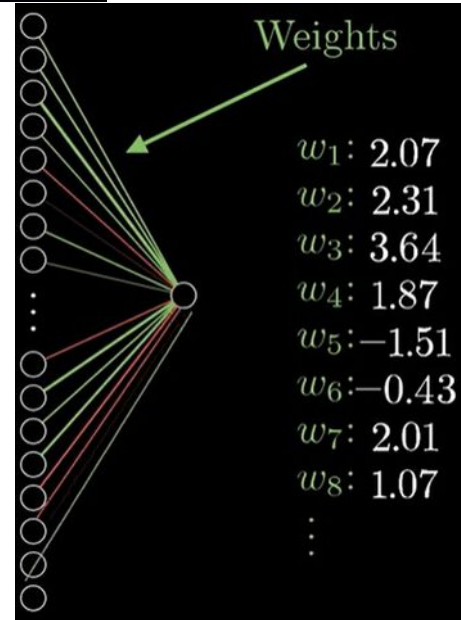
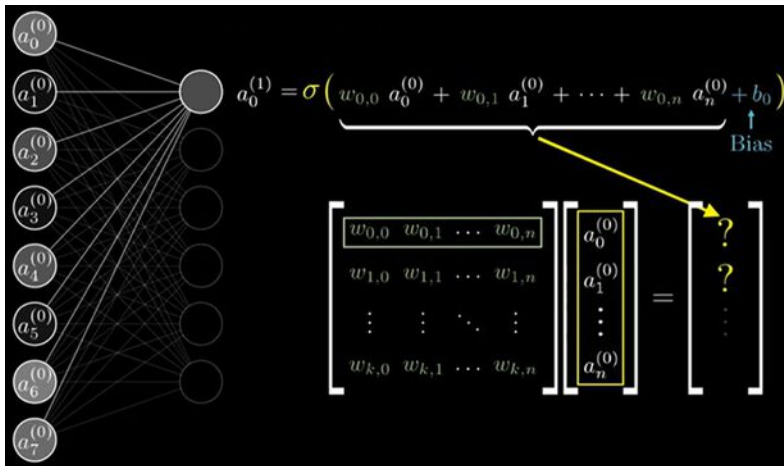
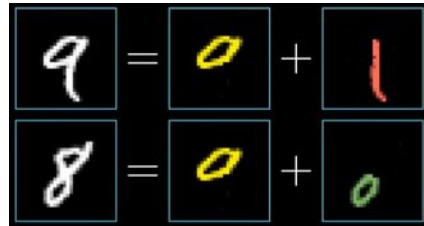
whoami

- Doctor of Engineering in Artificial Intelligence and Machine Learning (2027)
- MS Cybersecurity University of Texas at El Paso
- Bosch's Americas commercial vehicle cross-divisional product security
- Automotive Ethernet Security Expert
- Lead Monthly Cybersecurity Training for Bosch associates
- Automotive Ethernet & DoIP Instructor CyberTruck Challenge
- Competed at DEFCON31-32 Car Hacking Village CTF 3rd Place

What's What

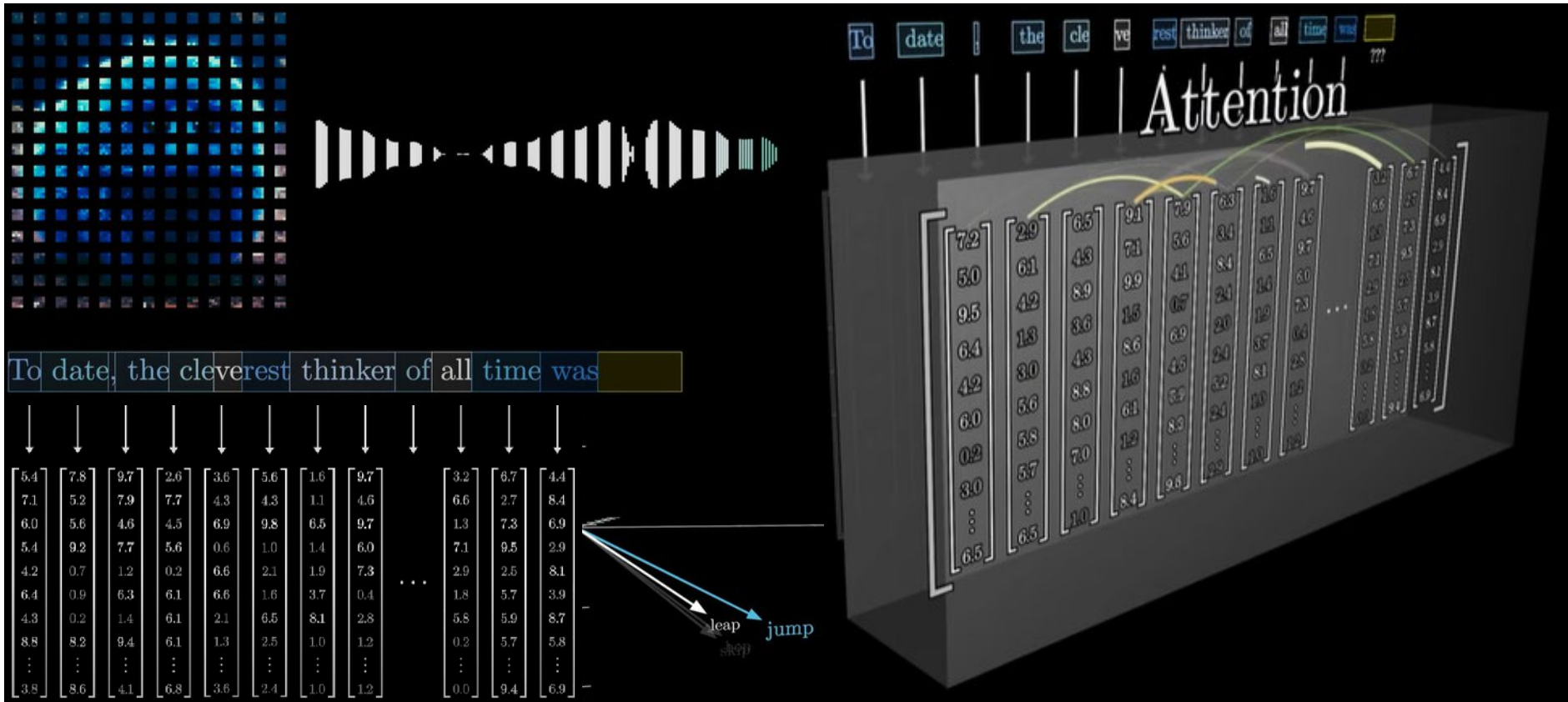


Neural Networks



From first to second layer each neuron needs 784 weights or connections, and one bias
 $(784 \times 16) + (16 \times 16) + (16 \times 10) = \text{weights} + 16 + 16 + 10 \text{ biases}$
 $= 13,002$

Generative Pre-trained Transformer



Warning Cost may Exceed the Value

- You are a Pandas DataFrame expert. Your task is to create an expression to be used with `eval()` to answer the question enclosed in angle brackets. For example, for the question: `<show all rows where 'identifier' is '310'>`, provide: `df[df['identifier'] == '310']`.
- You are an AI assistant proficient in creating regular expressions (regex). Your task is to help generate regex patterns that can be used in `re.search` based on user-provided search prompts. Provide only the regular expression pattern without any additional explanation or formatting. Do not include any backticks, quotes, or extra words.

```
sniffers > {} parsing_patterns.json > ...  
1 {  
2   "patterns": [  
3     {"name": "PATTERN_1", "regex": "^\\((?P<timestamp>\\d+\\.\\d+)\\)\\s+(?P<interface>\\w+)\\s+(?P<identifier>[0-9A-F]{3,8})#(?P<data>[0-9A-F]{3,8})"},  
4     {"name": "PATTERN_2", "regex": "^\\((?P<timestamp>\\d+\\.\\d+)\\)\\s+(?P<interface>\\w+)\\s+(?P<identifier>[0-9A-F]{3,8})##(?P<flags>\\d+)"},  
5     {"name": "PATTERN_3", "regex": "^(?P<interface>\\w+)\\s+(?P<identifier>[0-9A-F]{3,8})#(?P<data>[0-9A-F]{3,8})"},  
6     {"name": "PATTERN_4", "regex": "^(?P<interface>\\w+)\\s+(?P<identifier>[0-9A-F]{3,8})##(?P<flags>\\d+)"},  
7     {"name": "PATTERN_5", "regex": "^\\s*(?P<interface>\\w+)\\s+(?P<identifier>[0-9A-F]{3,8})\\s+\\[(?P<length>\\d+)"},  
8     {"name": "PATTERN_6", "regex": "^\\((?P<timestamp>\\d+\\.\\d+)\\)\\s*(?P<interface>\\w+)\\s+(?P<identifier>[0-9A-F]{3,8})"},  
9   ]  
10 }
```



Text to RegEx

- You are an expert regex generator specializing in Python's `re.search` function. For a given search prompt, generate only the precise regular expression pattern. Your output must be a raw regex string, with no surrounding text, explanations, or formatting whatsoever.

Packet Analyzer

<https://github.com/IvanGranero/pktAnalyzer>

time	source	destination	protocol	length	info	identifier	data	dataframe	dataprint
14661...			CANFD	8	CANFD	0x230	000600000...	300200000...	0!LJ98-14
									"G568-AR
									!...
									"3B1...
									!LU5T-14
									"D068-AH
									!L1MH-19
									"F611-AC
									LJ98-1
									LJ98-14
									G684-AR
									LJ98-14
14670...			CANFD	8	CANFD	0x9f	244735363...	9f0000000...	\$G568-AR

Find what:
☒ AI
Search in:
Preview:

\b\w{4}-\b

Min Length:

Find

Find what:
☒ AI
Search in:
Preview:

^30.*

find anything that starts with 30

Find

☒ Regex

dataframe

destination	protocol	length	info	identifier	data	dataframe	dataprint
	CANFD	8	CANFD	0x230	000600000...	300200000...	0!LJ98-14
							"G568-AR
							!...
							"3B1...
							!LU5T-14
							"D068-AH
							!L1MH-19
							"F611-AC
							LJ98-1
							LJ98-14
							G684-AR
							LJ98-14
							\$G568-AR

Packet Analyzer

<https://github.com/IvanGranero/pktAnalyzer>

time	source	destination	protocol	length	info	identifier	data	dataframe	dataprint
14661...			CANFD	8	CANFD	0x6fd	214c4a393...	fd0600000...	⌂!LJ98-14
Find Find what: find any word that contains 4 characters followed by - Find ☒ AI ☒ Regex Search in: dataprint Preview: \b\w{4}-\b Min Length: 4 Add Column with Strings									
14670...			CANFD	8	CANFD	0x9f	244735363...	9f0000000...	⌂\$G568-AR



Text to Pandas

- As a Senior Python Developer specializing in data manipulation with Pandas, your objective is to generate a concise and directly executable Pandas DataFrame expression for the Python `eval()` function. This expression must precisely answer the given natural language query. Your output should be solely the Pandas expression, without any additional text, explanations, or code formatting.

Packet Analyzer

<https://github.com/IvanGranero/pktAnalyzer>

pktAnalyzer

File Sniff Find Plot Tools Help

☒ AI

	time	source	destination	protocol	length	info	identifier	data	dataframe	dataprint
25	172841120...	169.254.1.0	255.255.25...	DoIP	82	Ether / IP / ...	13400		ffffffffffff...	ÿÿÿÿÿÿ<.:<...
37	172841120...	169.254.1.0	255.255.25...	DoIP	82	Ether / IP / ...	13400		ffffffffffff...	ÿÿÿÿÿÿ<.:<...
43	172841120...	169.254.1.0	255.255.25...	DoIP	82	Ether / IP / ...	13400		ffffffffffff...	ÿÿÿÿÿÿ<.:<...
121	172841122...	169.254.1.2	169.254.1.0	DoIP	69	Ether / IP / ...	13400		020500000...	<.:<(è,i0X...
122	172841122...	169.254.1.0	169.254.1.2	DoIP	71	Ether / IP / ...	13401		80e82ced3...	(è,i0X<.:<...
130	172841123...	169.254.1.2	169.254.1.0	UDS_RDBI	69	Ether / IP / ...	13400		020500000...	<.:<(è,i0X...
131	172841123...	169.254.1.0	169.254.1.2	DoIP	67	Ether / IP / ...	13401		80e82ced3...	(è,i0X<.:<...
132	172841123...	169.254.1.0	169.254.1.2	Raw	169	Ether / IP / ...	13401		80e82ced3...	(è,i0X<.:<...
136	172841123...	169.254.1.2	169.254.1.0	UDS_RDBI	69	Ether / IP / ...	13400		020500000...	<.:<(è,i0X...
137	172841123...	169.254.1.0	169.254.1.2	DoIP	67	Ether / IP / ...	13401		80e82ced3...	(è,i0X<.:<...
138	172841123...	169.254.1.0	169.254.1.2	Raw	169	Ether / IP / ...	13401		80e82ced3...	(è,i0X<.:<...

▼ Ether / IP / TCP 169.254.1.2:13401 > 169.254.1.0:13400 PA / DoIP / UDS / UDS_R...
dst: 02:05:00:00:02:00
src: 80:e8:2c:ed:30:58
type: 2048

▼ IP / TCP 169.254.1.2:13401 > 169.254.1.0:13400 PA / DoIP / UDS / UDS_RDBI
options: []
version: 4

02 05 00 00 02 00 80 e8 2c ed 30 58 08 00 45 00 00 37 fa a8 40 00 80 06 00 00 a9 fe 01
02 a9 fe 01 00 34 59 34 58 3d ee 6c c6 37 ea 2e fb 50 18 fa df 56 28 00 00 02 fd 01
00 00 00 07 0e f5 0c 45 22 f0 9e
<.:<(è,i0XE7ú"@©@p<©@p<4Y4X=îLÆ7ê.ûP-ûßV(≤ý(☎.øE"đ

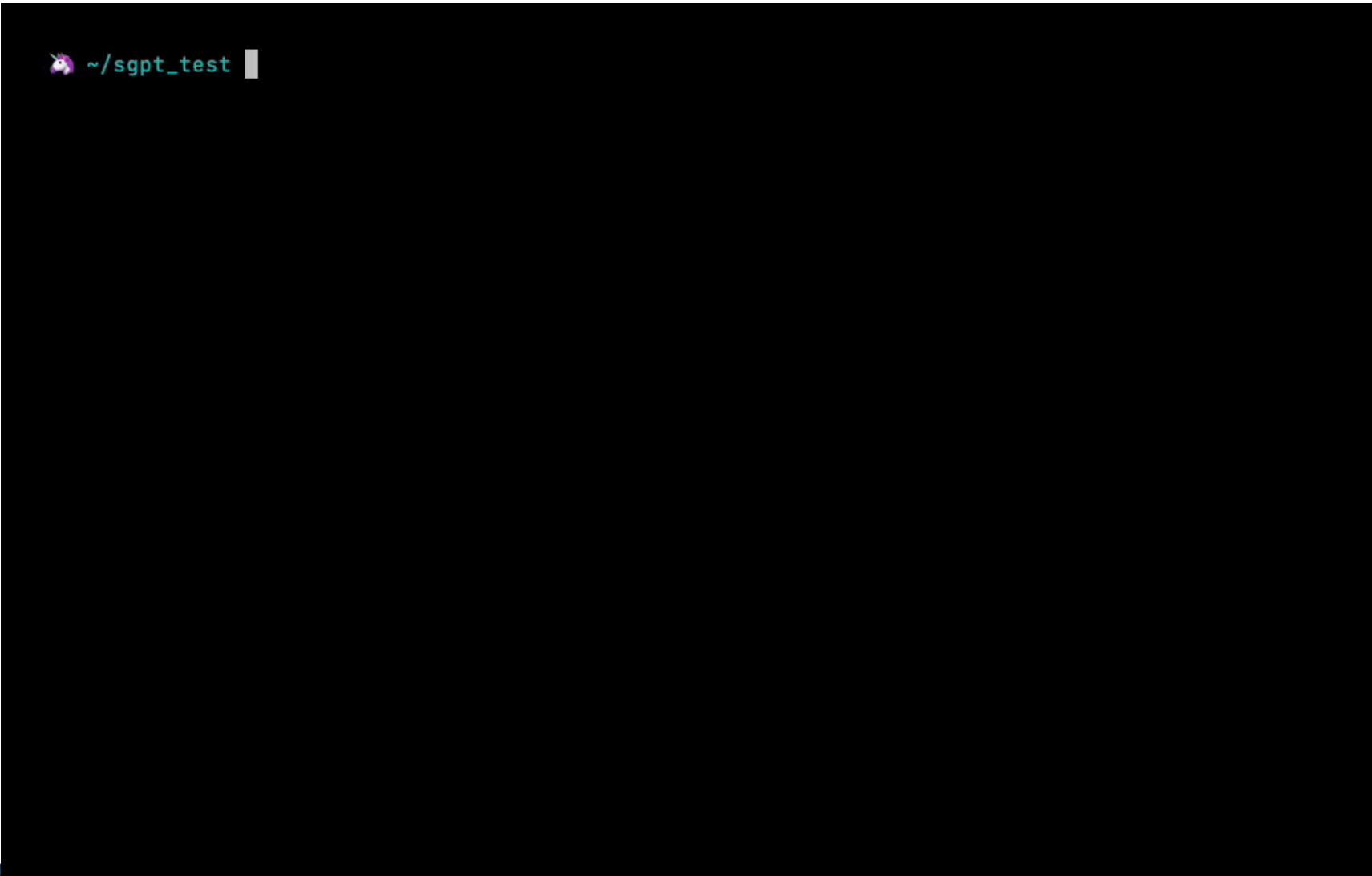
protocol
length
info
identifier

ARP
DHCP
DHCP6OptOpt
DNS
DoIP
ICMPv6MLRep
ICMPv6NDOpt
ICMPv6NDOpt
ICMPv6ND_NS
ICMPv6ND_RS
NBNSRegistrat
Padding
Raw
TCP

Ready.

Total Packets: 218 Displaying: 25 (11.47%)

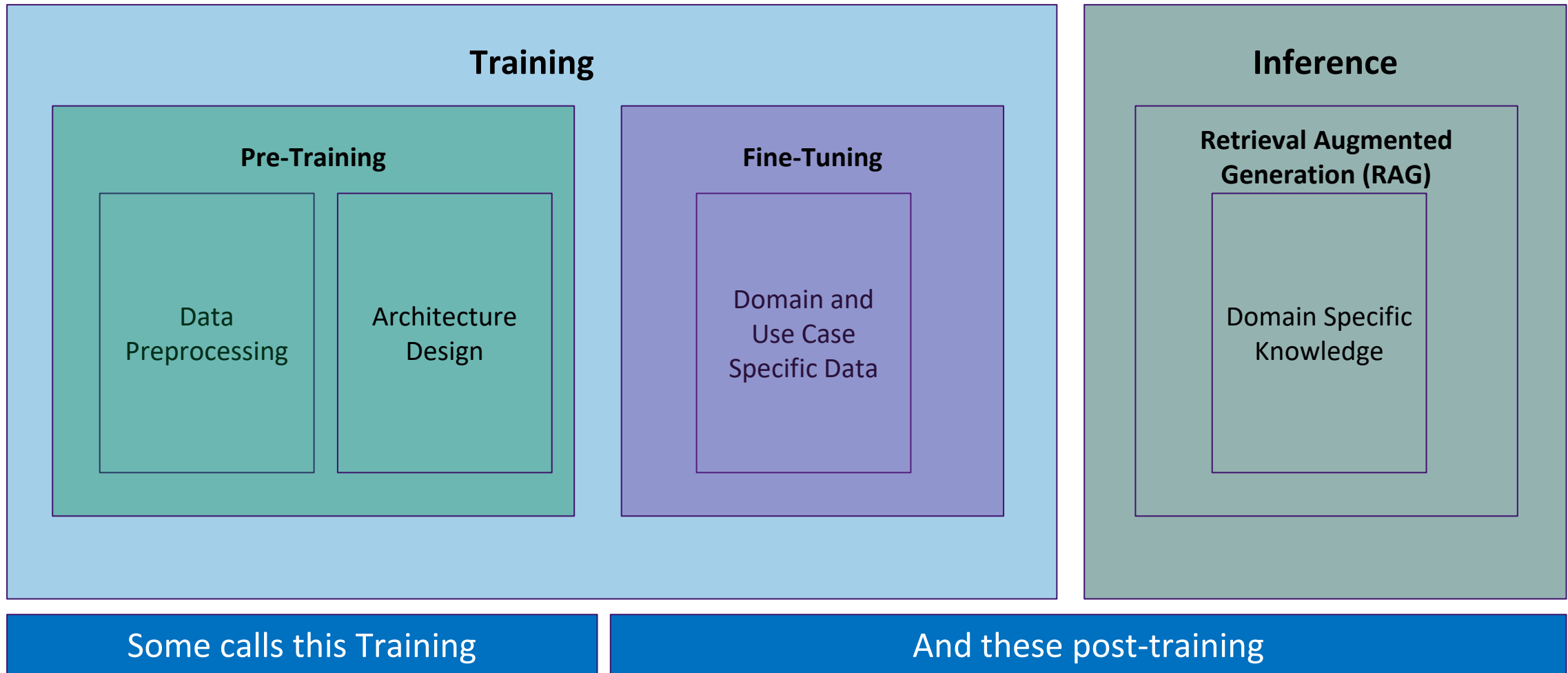
Text-to-Action



```
~/sgpt_test
```

Example, if we were to google something like “Linux, how to make all files in a directory read only” as result we would get a bunch of articles with 80% useless information for our specific query.

Pretraining, Finetuning and RAG



Finetuning

Pre-training for
general world
knowledge / lossy zip
of the internet



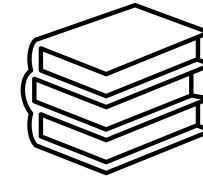
Base Model

General world knowledge

Fine-tuning for
specific domain
knowledge / task /
behavior



Base Model



Finetune with
additional knowledge
base / input



Fine-tuned Model

Improved task performance
Specific task behavior
Adapt to new task

The Landscape

- The Open Models
 - DeepSeek, Kimi, Qwen, Llama, Llava
 - OpenAI GPT OSS 120B/20B ([Open models by OpenAI | OpenAI](#), released Aug 5)
 - Anthropic Claude Opus 4.1 ([Claude Opus 4.1 \ Anthropic](#))
 - Text to image / video: Qwen Image (<https://qwenlm.github.io/blog/qwen-image/>), Stable Diffusion
- The Closed Model
 - Most OpenAI (different ChatGPT models), Anthropic (Claude models) models
 - Google Gemini, Grok (from xAI)
 - Text to image / video: Dall-E, Sora, Veo3
- Caveat of Open Models
 - What's open: the weight, and the license to use. Some also open architectures, training and inference code.
 - What this enables: To run inference (sometimes train the model from scratch) locally, without worrying about domain data leaving on-prem
 - Prerequisite: compute, and data (some may have problematic license)
 - What's not open: the data used to train the models, system prompt, sometimes the architecture

Different Types of Fine-Tuning

Type	Description	Use Cases
Supervised fine-tuning	The model is trained on a labeled dataset with input-output pairs.	Text classification, named entity recognition, sentiment analysis.
Instruction fine-tuning	The model is trained on a dataset of instructions and desired responses.	Chatbots, question answering systems, code generation.
Few-shot learning	The model is provided with a few examples of the desired task within the prompt.	Adapting to new tasks with limited data.
Transfer learning	The model leverages knowledge gained from pre-training on a general-purpose dataset.	Adapting to related tasks.
Domain-specific fine-tuning	The model is adapted to a particular domain or industry.	Legal document analysis, medical report generation, financial forecasting.
Multi-task learning	The model is trained on multiple tasks simultaneously.	Improving performance across related tasks.
Sequential fine-tuning	The model is adapted to a series of related tasks in stages.	Gradually refining capabilities for complex tasks.

Dataset

```
{
  "input": "get the vehicle identification number",
  "output": "22 F1 90"
},
{
  "input": "retrieve VIN",
  "output": "22 F1 90"
},
{
  "input": "what is the VIN?",
  "output": "22 F1 90"
},
{
  "input": "show me the VIN code",
  "output": "22 F1 90"
},
{
  "input": "request security access level 1 seed",
  "output": "27 01"
},
{
  "input": "get security seed for level 1",
  "output": "27 01"
},
{
  "input": "clear DTCs",
  "output": "14 FF FF FF"
},
{
  "input": "erase fault codes",
  "output": "14 FF FF FF"
},
{
  "input": "clear all diagnosti",
  "output": "14 FF FF FF"
},
{
  "input": "perform hard reset",
  "output": "11 01"
},
{
  "input": "reset with key off",
  "output": "11 02"
},
{
  "input": "soft reset the ECU",
  "output": "11 03"
},
{
  "input": "reset with ignitio",
  "output": "11 02"
},
{
  "input": "change to extended diagnostic session",
  "output": "10 03"
},
{
  "input": "activate extended session",
  "output": "10 03"
},
{
  "input": "go to extended diagnostics",
  "output": "10 03"
},
{
  "input": "set session to programming",
  "output": "10 02"
},
{
  "input": "enter programming session",
  "output": "10 02"
},
{
  "input": "switch to default session",
  "output": "10 01"
},
{
  "input": "return to default session",
  "output": "10 01"
}
```

Diagnostics Agent

Okay, I'm ready! As an automotive diagnostics expert in UDS, I will generate the UDS commands and their descriptions in JSON format, as requested. I'll also identify potential security access issues.

read the vin number

```
{
  "command": "22 F1 90",
  "description": {
    "service_id": "22 - ReadDataByIdentifier",
    "data_identifier": "F1 90 (VIN - Vehicle Identification Number)"
  }
}
```

62F19031474E454B31335A583352323938393834

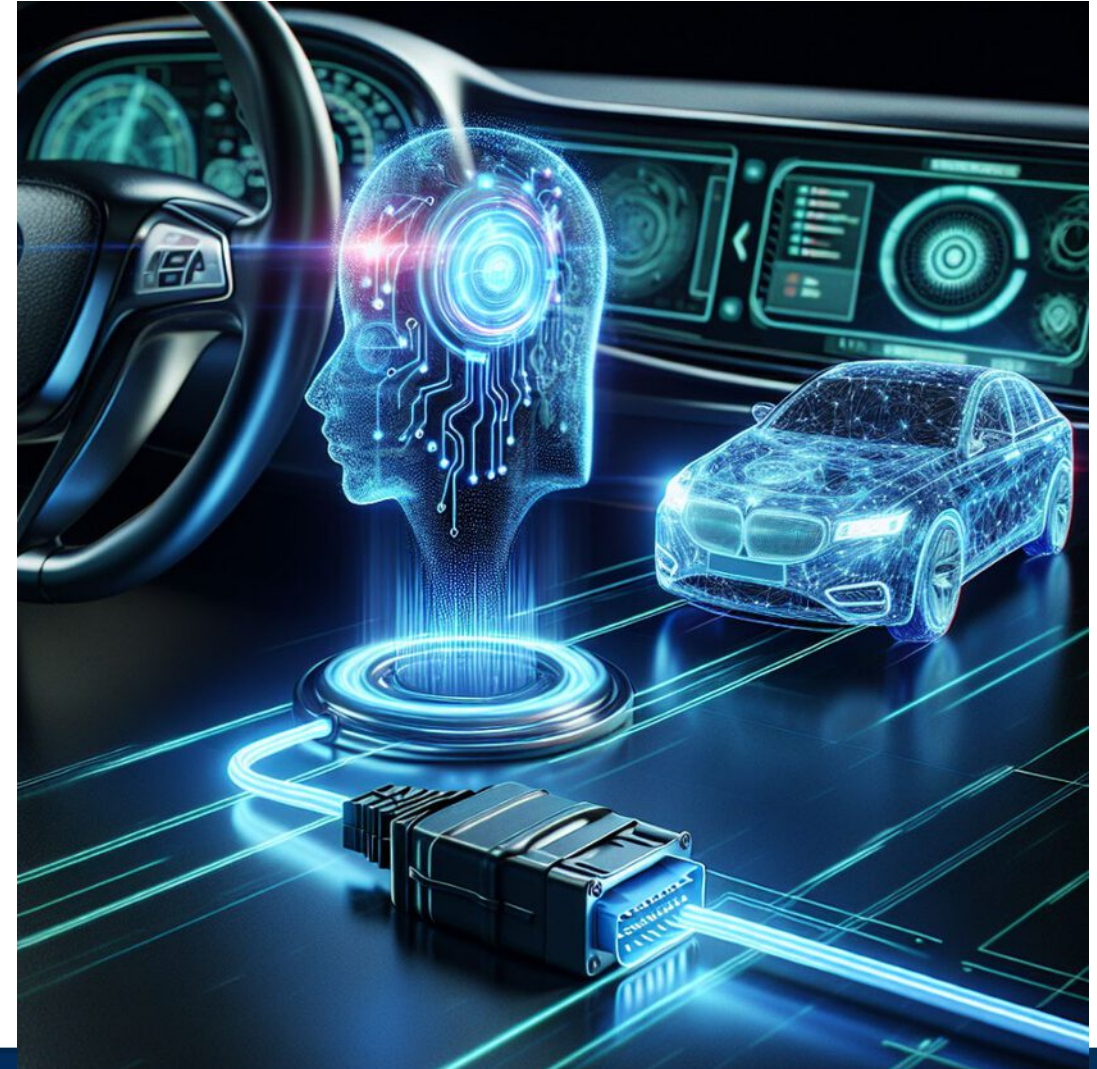
```
{
  "service": "ReadDataByIdentifier",
  "status": "positive_response",
  "data_identifier": "F190",
  "data": "31474E454B31335A583352323938393834",
  "data_ascii": "1GNEK13ZX3R298984"
}
```

Diagnostics Agent

Okay, I'm ready! As an automotive diagnostics expert in UDS, I will generate the UDS commands and their descriptions in JSON format, as requested. I'll also identify potential security access issues.

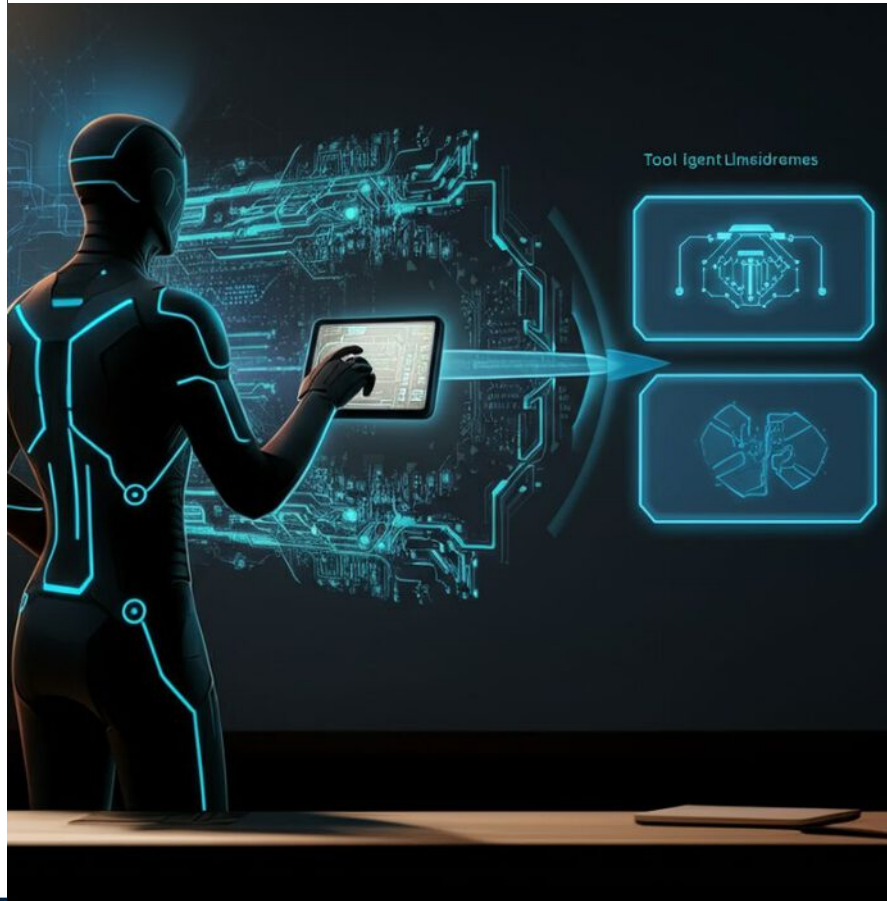
read 10 bytes at memory address FE0198

```
{  
  "command": "23 31 FE 01 98 0A",  
  "description": {  
    "service_id": "23 - ReadMemoryByAddress",  
    "address_size_format": "31 (3 bytes address, 1 byte s",  
    "memory_address": "FE 01 98",  
    "bytes_to_read": "0A"  
  }  
}
```



Diagnostics Orchestrator

Okay, I'm ready! As an automotive diagnostics expert in UDS, I will generate the UDS commands and their descriptions in JSON format, as requested. I'll also identify potential security access issues.



open a secure session access level 01

```
{  
  "command": "27 01",  
  "error": false,  
  "security_required": true,  
  "description": {  
    "service_id": "27 - SecurityAccess",  
    "subfunction": "01 - RequestSeed (Level 1)"  
  }  
}
```

67 01 AA BB

```
{  
  "service": "SecurityAccess",  
  "status": "seed_response",  
  "tool_calls": [  
    {  
      "tool_name": "calculate_key",  
      "description": "Generates a security access key based on the provided seed",  
      "parameters": {  
        "seed": "AA BB"  
      }  
    }  
  ]  
}
```

When to Use Prompting and When to Use Finetuning

Prompting
to source external
knowledge

Zero-shot learning

Respond to prompt
based on pre-training
knowledge only

In-context learning (e.g. few-shot)

Contextualize prompt
with examples of
expected output

RAG

Contextualize prompt
with most relevant
(e.g. embedded)
knowledge

Agents

Model reasons and
dynamically interacts
with external
knowledge sources

Fine-tuning
to generate
parametric
knowledge

for instruction following

for conversational capabilities

for domain specific language/knowledge

...

What Neural Networks Can Comprehend

Your role is as a binary analysis expert, I will give you a binary file as a hex string, find the overflow vulnerability and provide a possible fix to it, once you have the fix provide the C code and the new hex string with the patch,

Here is the hex string:

4d5a90000300000004000000ffff0000b800000000000000400000.... 41000 bytes

After analyzing the binary file, I've identified a buffer overflow vulnerability in the `vulnerable_function`.

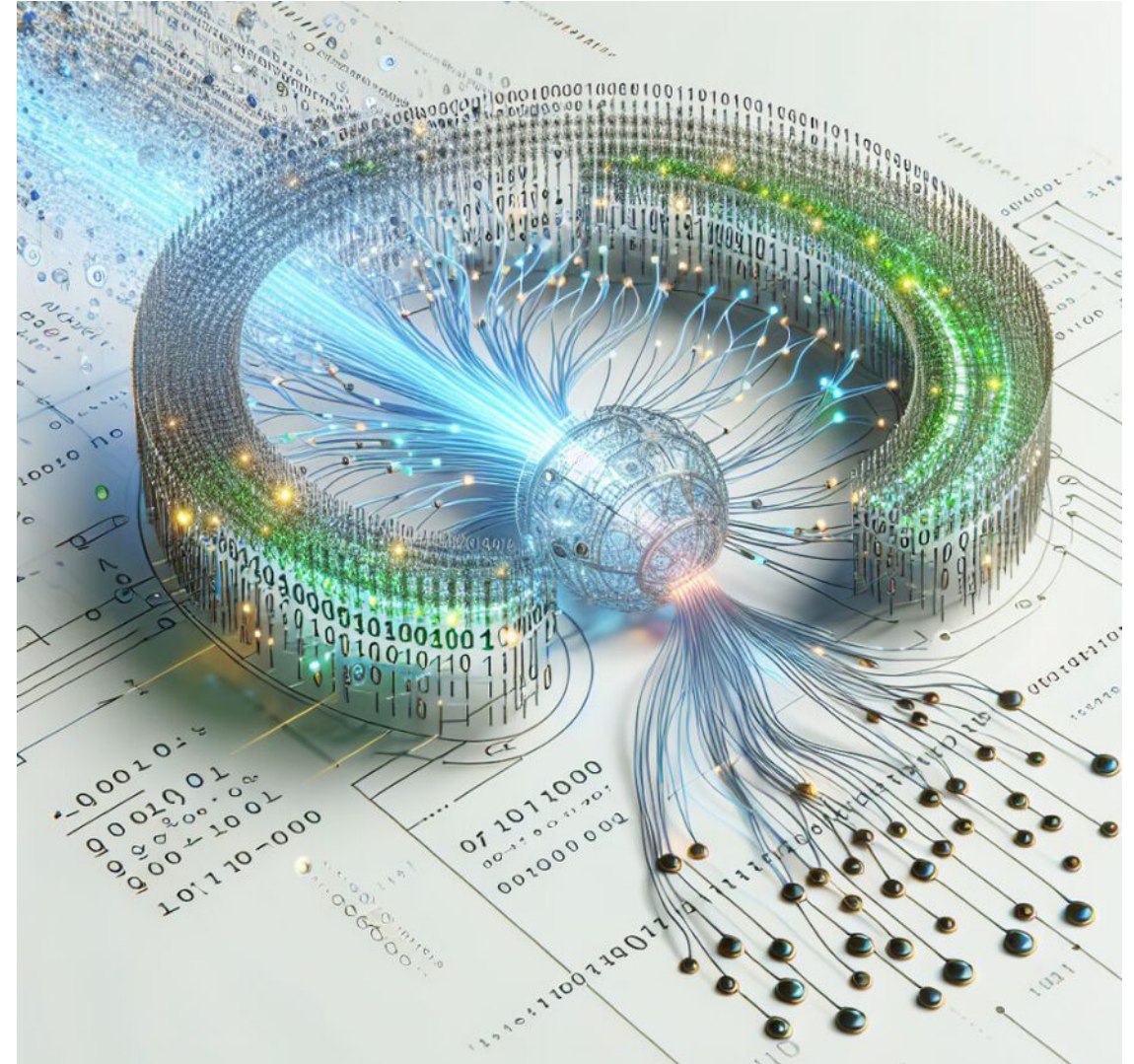
Here's the analysis and fix:

Vulnerable Function (decompiled):

```
void vulnerable_function() {  
    char buffer[64]; // Stack buffer of 64 bytes  
    printf("Enter some text: ");  
    gets(buffer); // Vulnerable to buffer overflow  
    printf("You entered: %s\n", buffer);  
}
```

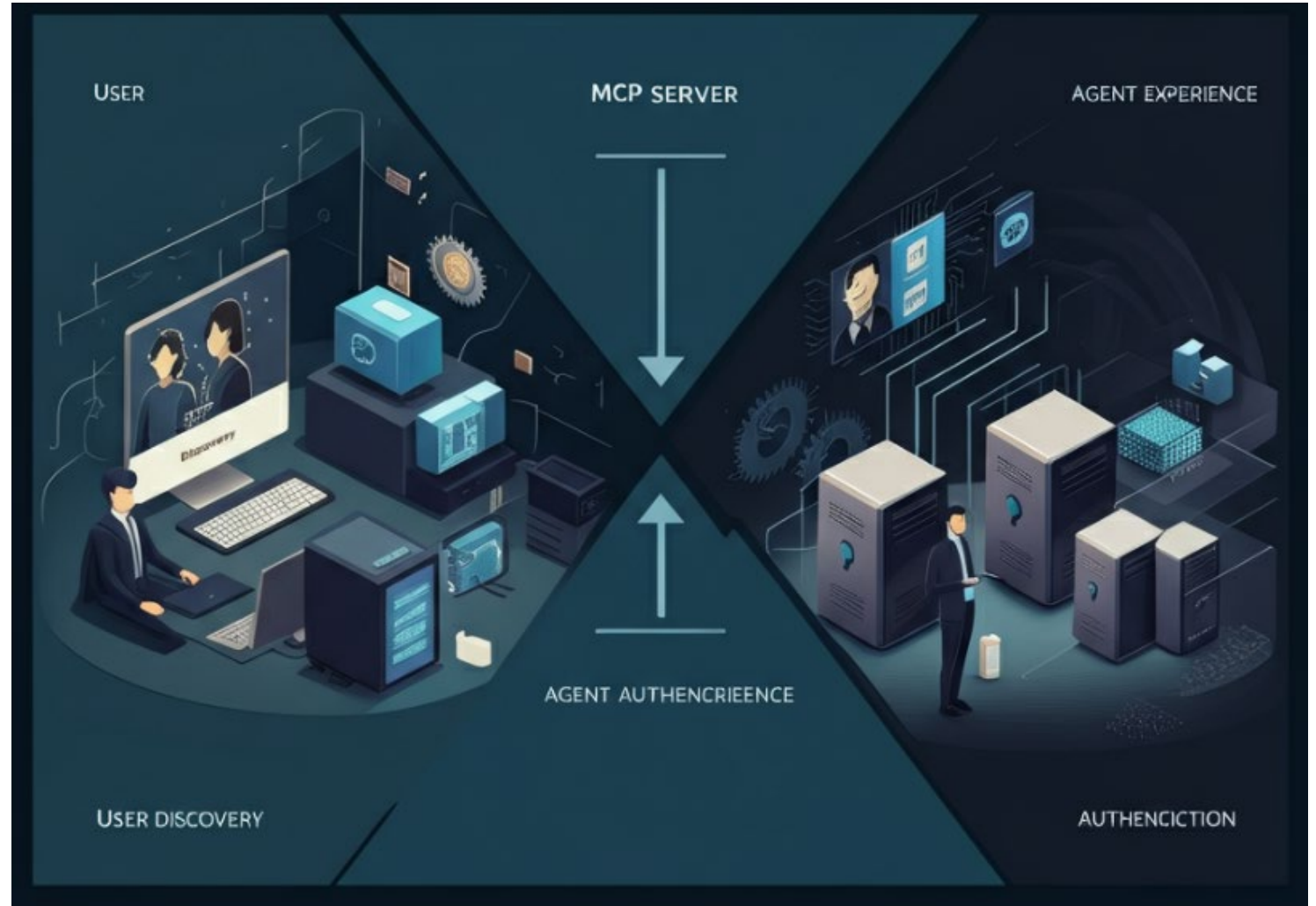
Further Considerations in Binary Analysis

- Instruction set for Embedded uC
- Context Window Limitation
- Checksums and Integrity Checks
- Obfuscated Firmware
- Validation needed



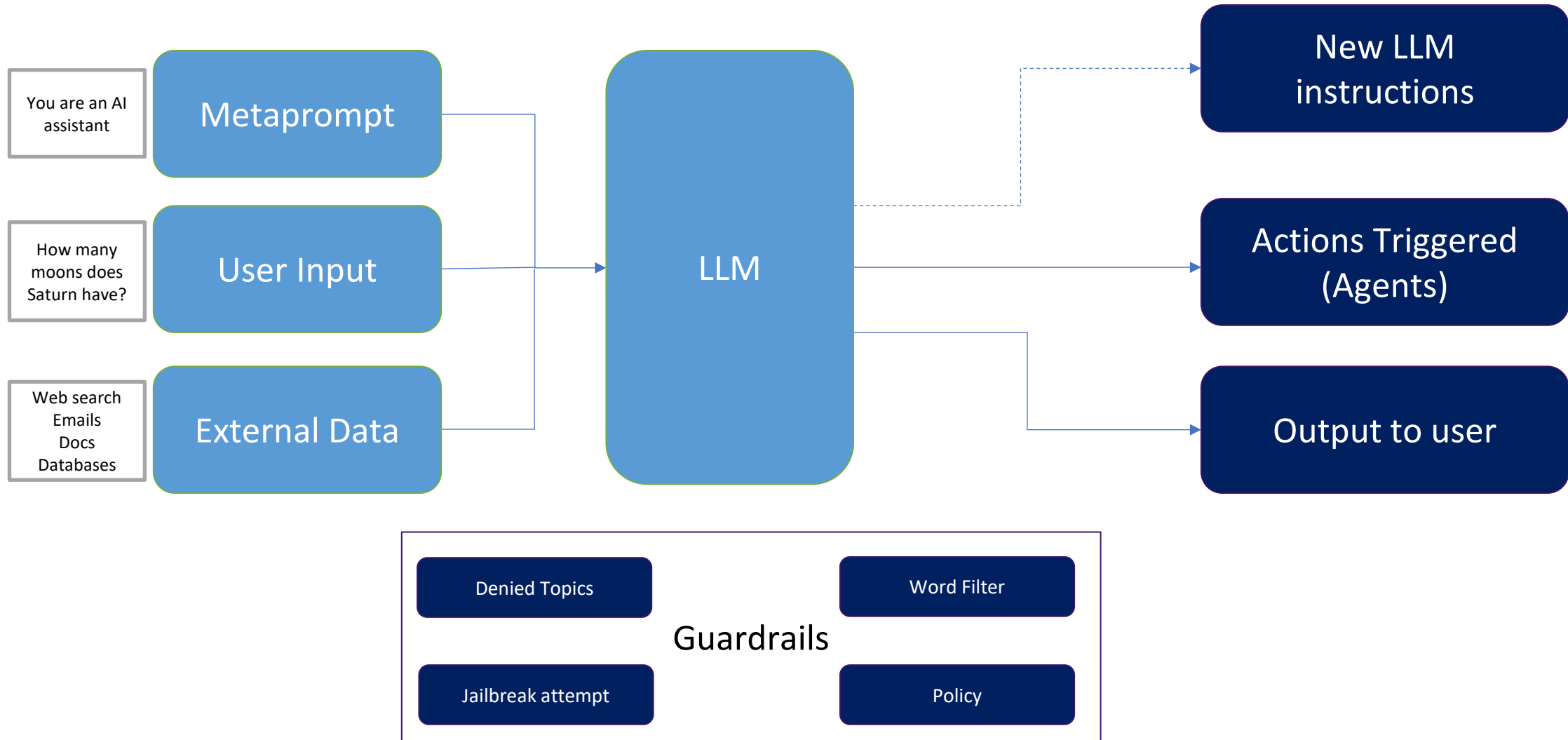
UX to AX Agent Experience

- MCP servers
- ONNX
- Interface your tools



Security of AI

<https://cybergranero.com>



Questions & Answers

- I'm happy to take your questions

Contact Info

- Thank you
 - Ivan Granero
 - Bosch
 - Ivan.Granero@us.bosch.com
 - www.linkedin.com/in/IvanGranero