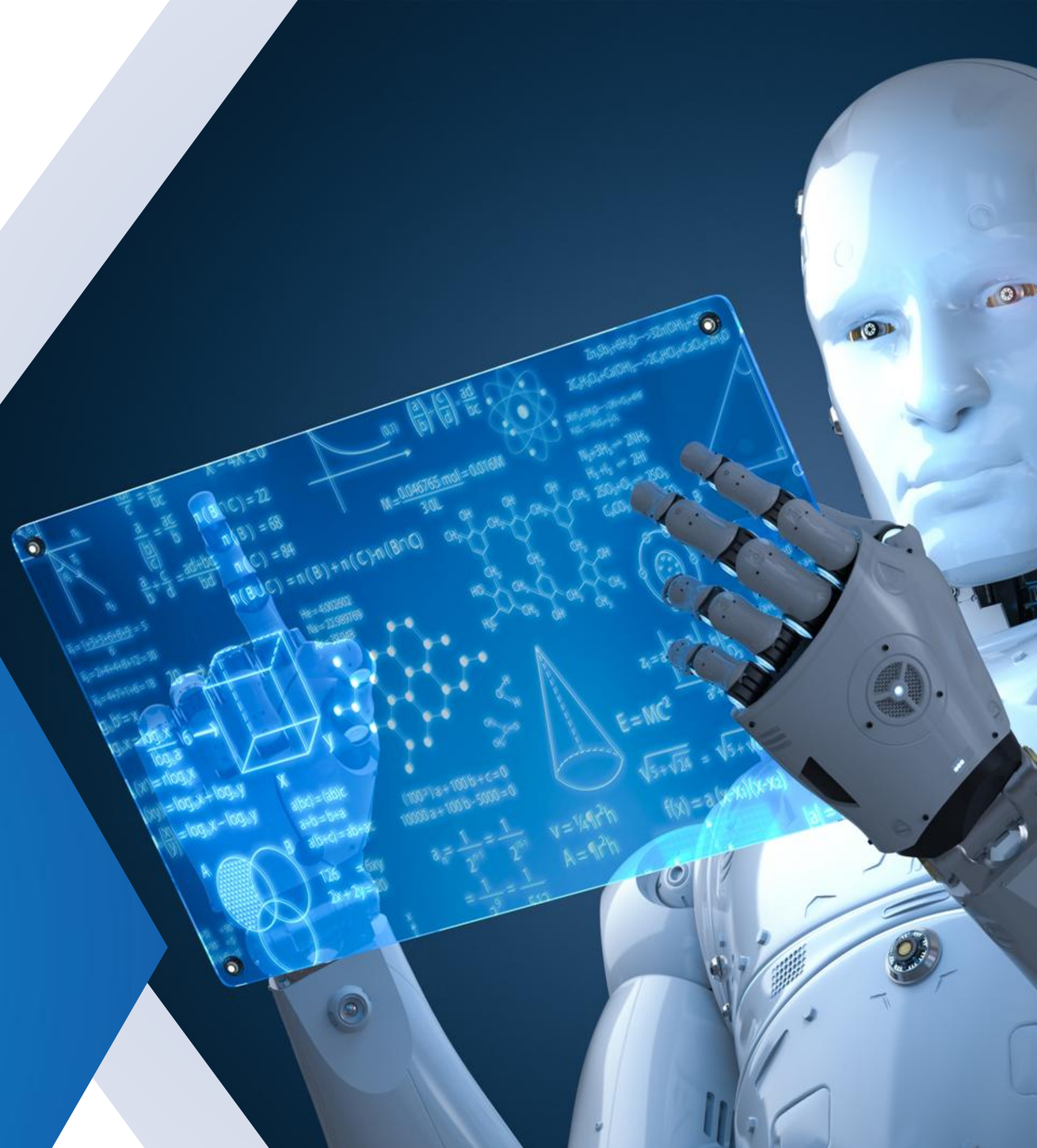
Vietnam National University, Ho Chi Minh City
University of Information Technology

MACHINE LEARNING in Cybersecurity



Group 08

MỤC TIÊU

Các công cụ phát hiện lỗ hổng trong smart contract hiện nay chủ yếu dựa vào ***rà soát thủ công*** và kỹ thuật ***đối chiếu theo mẫu.***

=> Mục tiêu của Project là xây dựng một hệ thống tự động phát hiện lỗ hổng trong smart contracts bằng LLMs, đem lại hiệu suất cao hơn, độ chính xác tốt hơn.

TỔNG QUAN

Retrieval-Augmented Generation (RAG)

Step-back prompting

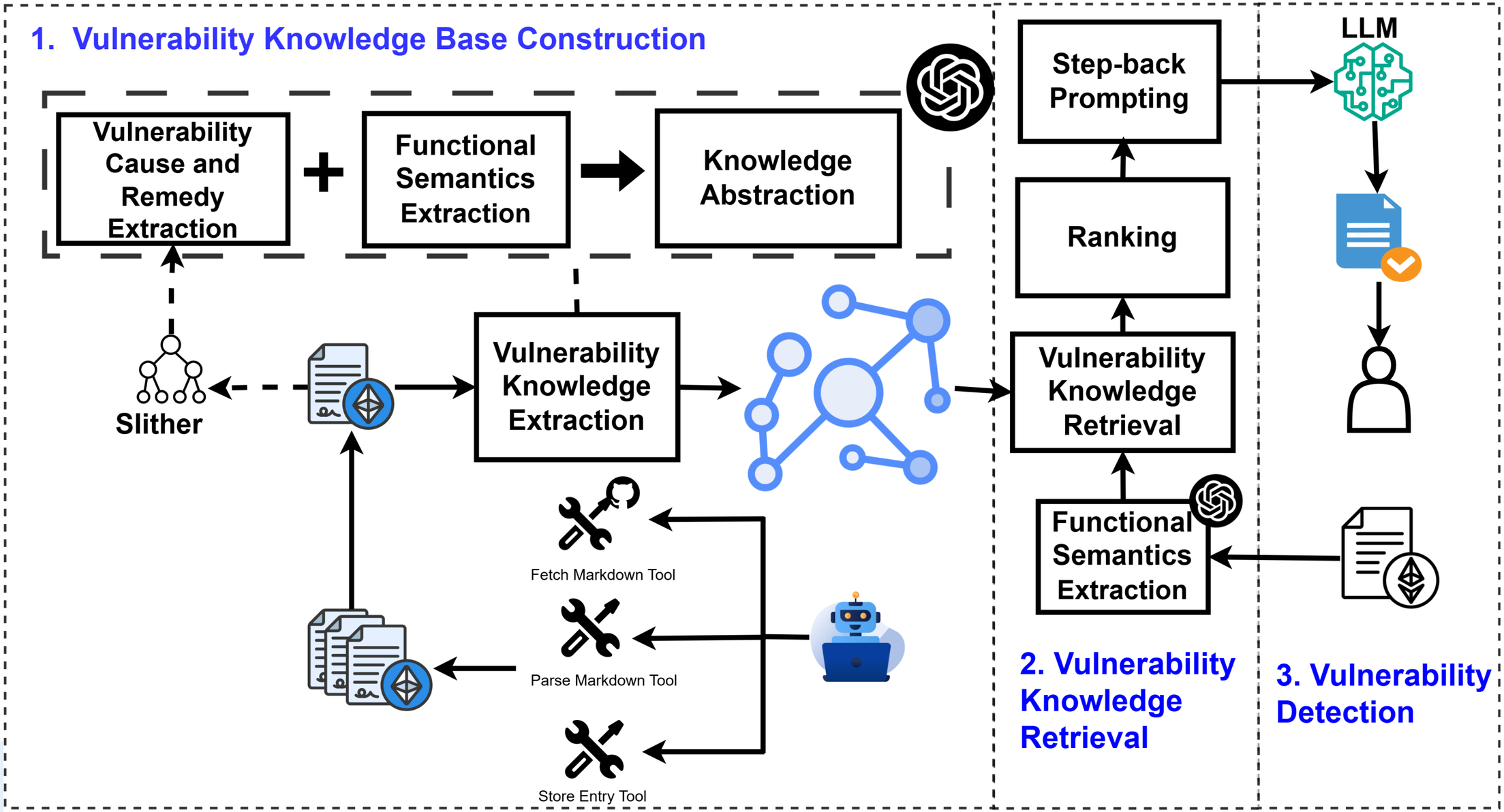
Fine-tuning LLMs

```
train_prompt_style = """
    Read the below context to enhance your knowledge about the vulnerability in the detected code.

    ### Context:
    {context}

    ### Instruction:
    You are a smart contract security assistant with deep expertise in Ethereum, Solidity, and DeFi security.

    Please follow these steps to analyze the provided Solidity code:

    **Step 1**: Summarize what the code is intended to do.
    **Step 2**: Identify and explain any suspicious or potentially unsafe logic patterns.
    **Step 3**: Based on your analysis, determine the type of vulnerability, describe the issue in detail, and recommend mitigation strategies.

    Provide your output in the following structured format, do not explicit your thoughts or reasoning:

    **Type of Vulnerability**:
    **Description**:
    **Recommendation**:

    ### Code:
    {query}

    ### Response:
"""
```

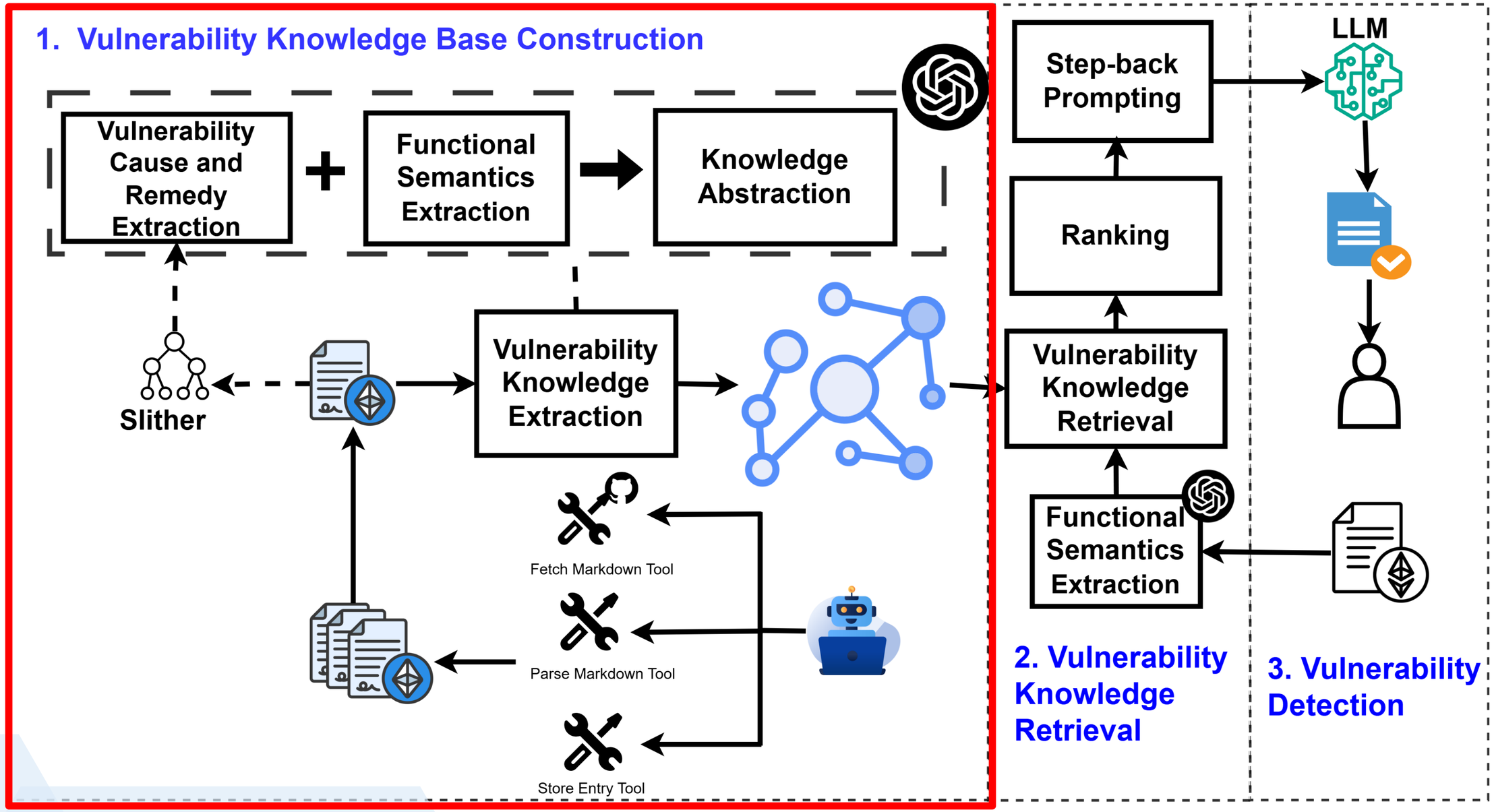
CÔNG CỤ TRIỂN KHAI

- Graph Database: Neo4j
- Open-source LLMs: Deepseek-R1-Distill-Llama-8B, Qwen-Coder-2.5-14B-Solidity
- Closed-source LLMs: GPT 3.5-Turbo
- Embedding model: all-MiniLM-L6-v2
- Data: SWC Registry, SmartBugs Curated, Solidi-FI
- Static analysis tool: Slither, Mythril

MÔ HÌNH TRIỂN KHAI



Vulnerability Knowledge Base Construction



Vulnerability Knowledge Base Construction

Processing SWC-126...

New run

Fetch, parse, and store SWC-126

InferenceClientModel - Qwen/Qwen2.5-7B-Instruct

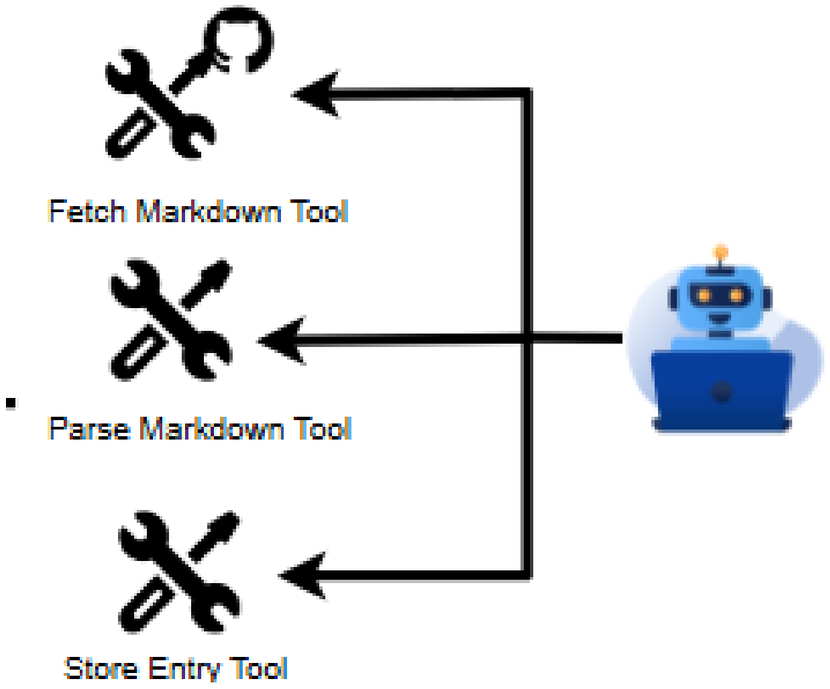
Step 1

Executing parsed code:

```
swc_id = "SWC-126"
markdown_content = fetch_markdown(swc_id)
print(markdown_content)
```

Execution logs:

```
# Please note, this content is no longer actively maintained.
```



Step 2

Executing parsed code:

```
entry = parse_markdown(markdown_content, swc_id)
print(entry)
store_entry(entry, "swc_entries.json")
```

Execution logs:

```
{'id': 'SWC-126', 'description': ["Insufficient gas griefing attacks can be performed on contracts which accept data and use it in a sub-call on another contract. If the sub-call fails, either the whole transaction is reverted, or execution is continued. In the case of a relayer contract, the user who executes the transaction, the 'forwarder', can effectively censor transactions by using just enough gas to execute the transaction, but not enough for the sub-call to succeed."], 'relayer.sol': '/*\n * @source: https://consensys.github.io/smart-contract-best-practices/known_attacks/#insufficient-gas-griefing\n * @author: ConsenSys Diligence\n * Modified by Kaden Zipfel\n */\n\npragma solidity ^0.5.0;\n\ncontract Relayer {\n    uint transactionId;\n\n    struct Tx {\n        bytes data;\n        bool executed;\n    }\n\n    mapping (uint => Tx) transactions;\n\n    function relay(Target target, bytes memory _data) public returns(bool) {\n        // replay protection; do not call the same transaction twice\n        require(transactions[transactionId].executed == false, \'same transaction twice\');\n        transactions[transactionId].data = _data;\n        transactions[transactionId].executed = true;\n        transactionId += 1;\n\n        (bool success, ) = address(target).call(abi.encodeWithSignature("execute(bytes)", _data));\n        return success;\n    }\n}\n\n// Contract called by Relayer\ncontract Target {\n    function execute(bytes memory _data) public {\n        // Execute contract code\n    }\n}', 'relayer_fixed.sol': '/*\n * @source: https://consensys.github.io/smart-contract-best-practices/known_attacks/#insufficient-gas-griefing\n * @author: ConsenSys Diligence\n * Modified by Kaden Zipfel\n */\n\npragma solidity ^0.5.0;\n\ncontract Relayer {\n    uint transactionId;\n\n    struct Tx {\n        bytes data;\n        bool executed;\n    }\n\n    mapping (uint => Tx) transactions;\n\n    function relay(Target target, bytes memory _data, uint _gasLimit) public {\n        // replay protection; do not call the same transaction twice\n        require(transactions[transactionId].executed == false, \'same transaction twice\');\n        transactions[transactionId].data = _data;\n        transactions[transactionId].executed = true;\n        transactionId += 1;\n\n        address(target).call(abi.encodeWithSignature("execute(bytes)", _data, _gasLimit));\n    }\n}\n\n// Contract called by Relayer\ncontract Target {\n    function execute(bytes memory _data, uint _gasLimit) public {\n        require(gasleft() >= _gasLimit, \'not enough gas\');\n        // Execute contract code\n    }\n}'}
```

Out: Saved SWC-126 to swc_entries.json

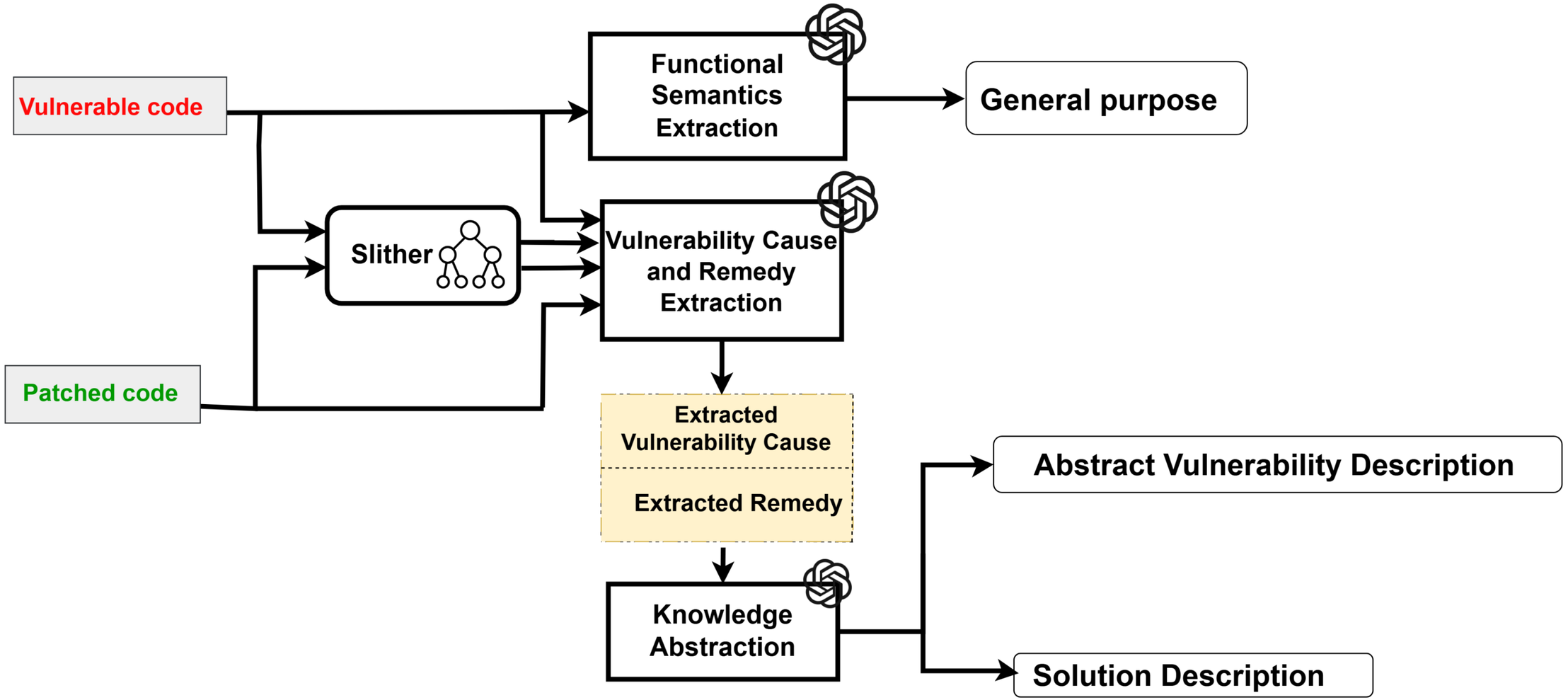
[Step 2: Duration 0.75 seconds| Input tokens: 5,613 | Output tokens: 204]

Step 3

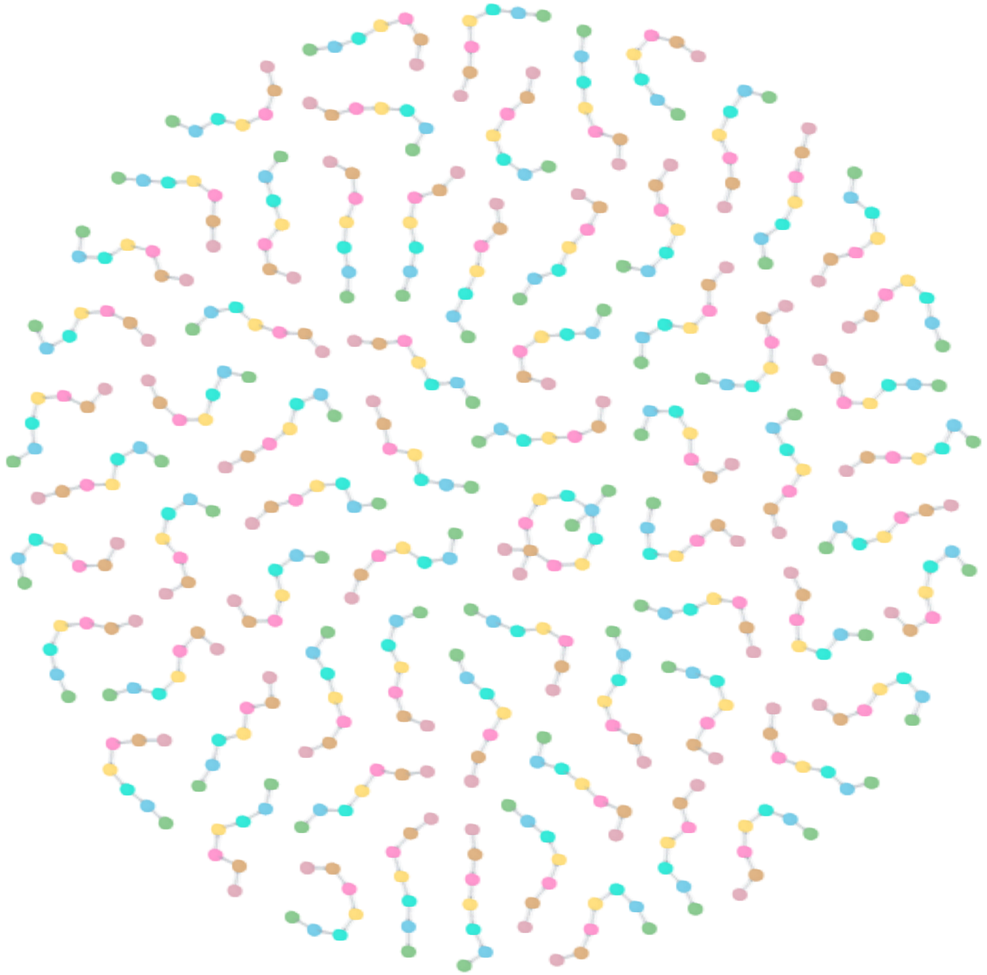
Executing parsed code:

```
final_answer("SWC-126 entry has been successfully fetched, parsed, and stored.")
```

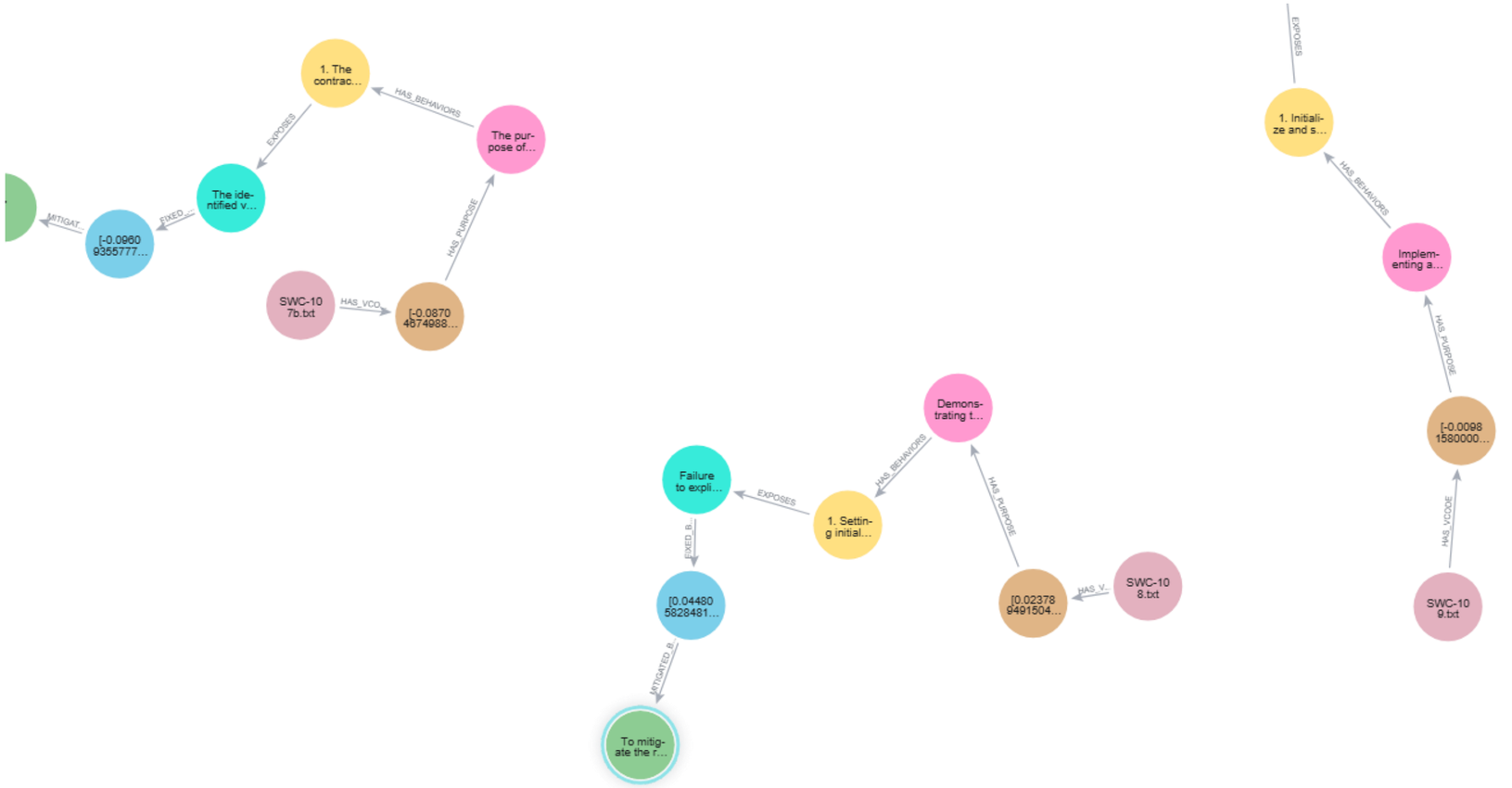
Vulnerability Knowledge Base Construction



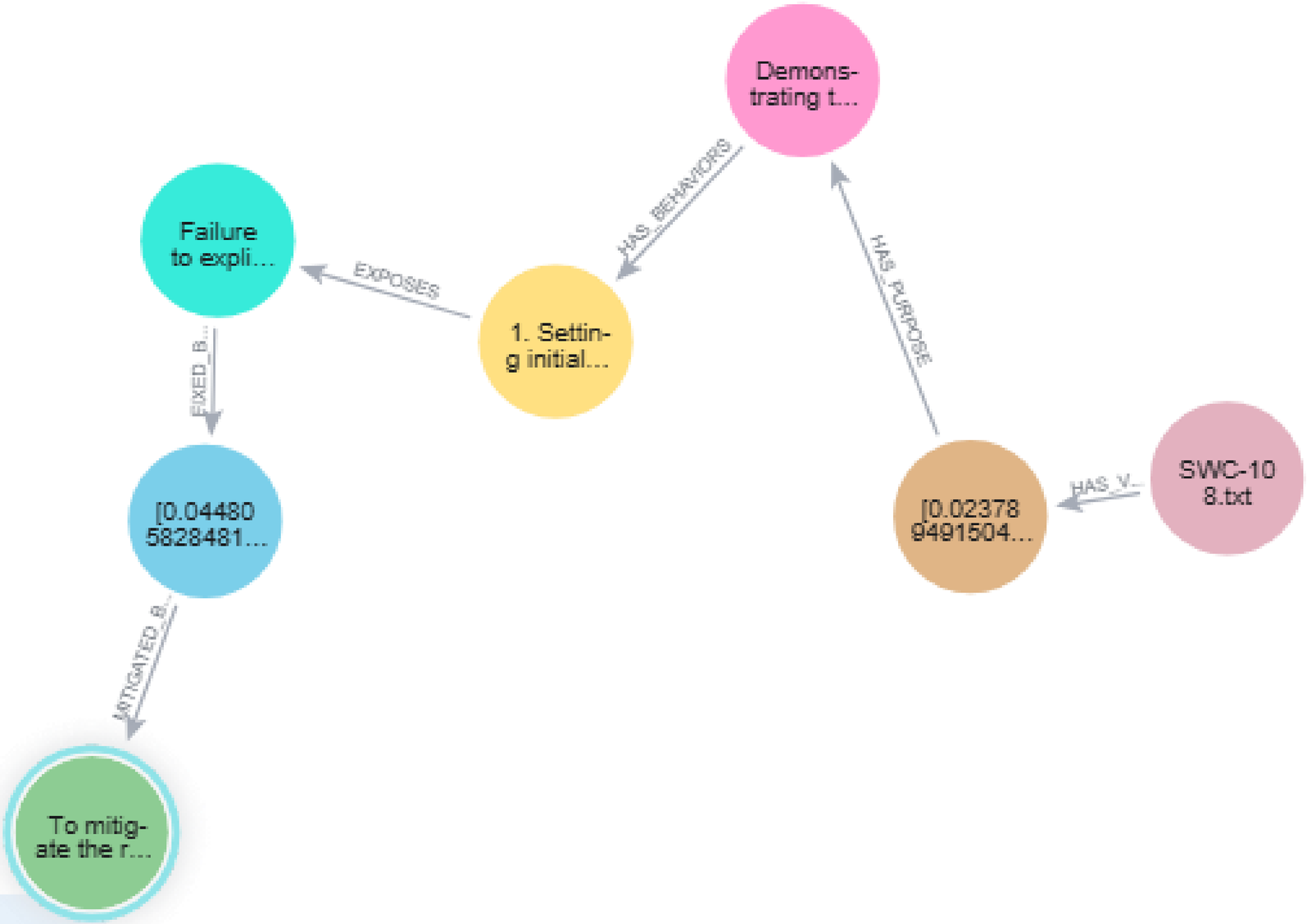
Vulnerability Knowledge Base Construction



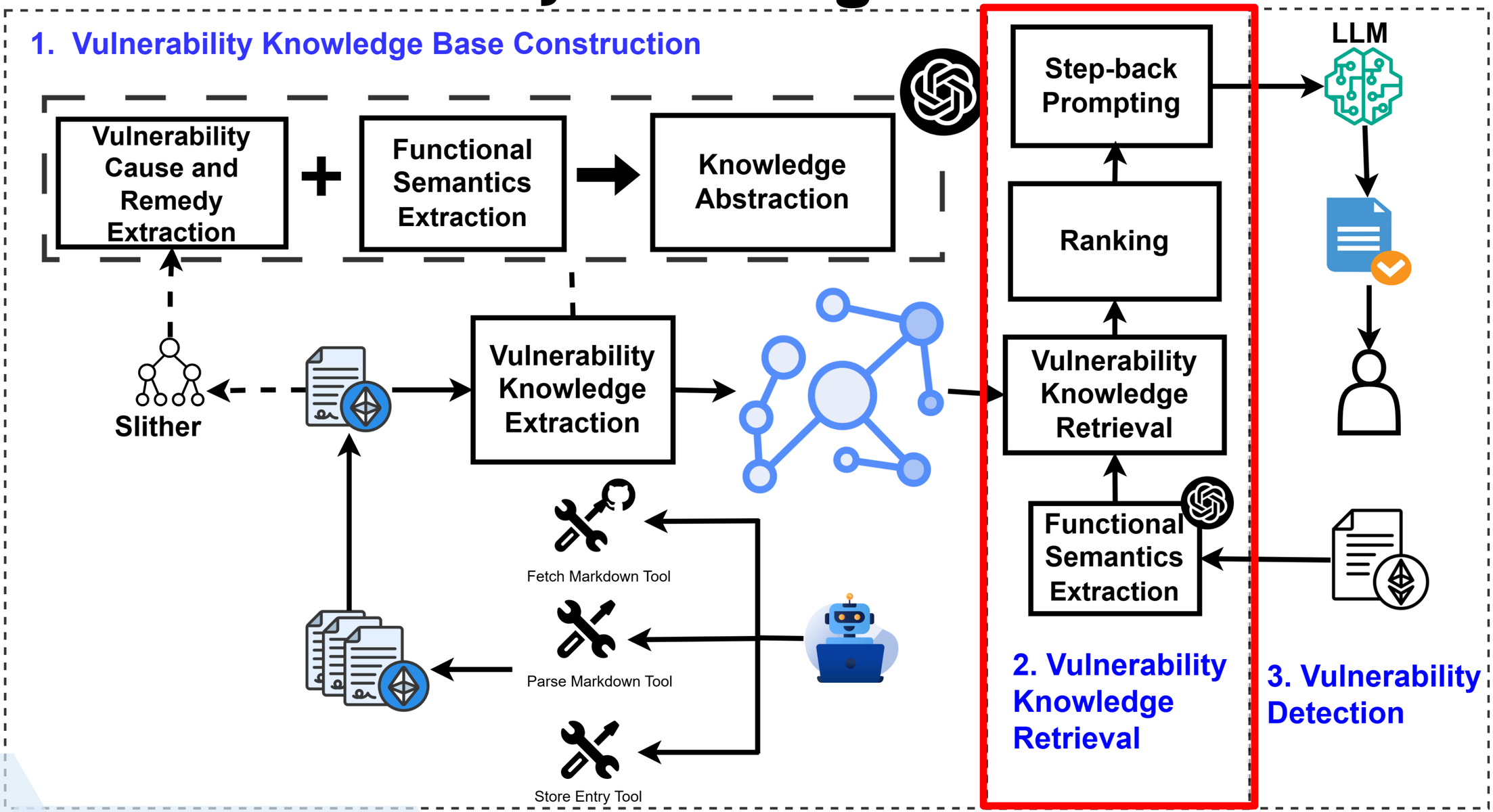
Vulnerability Knowledge Base Construction



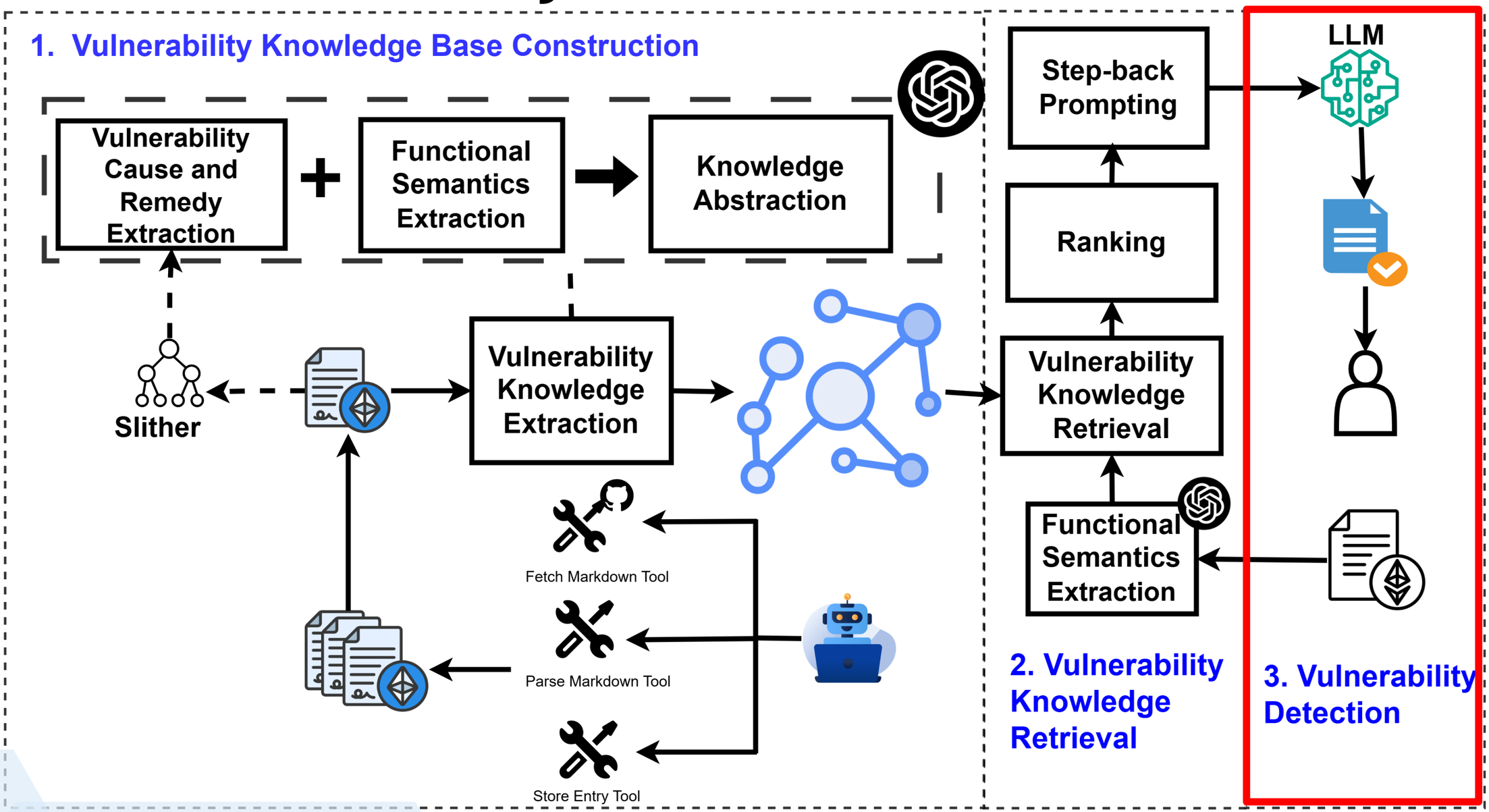
Vulnerability Knowledge Base Construction



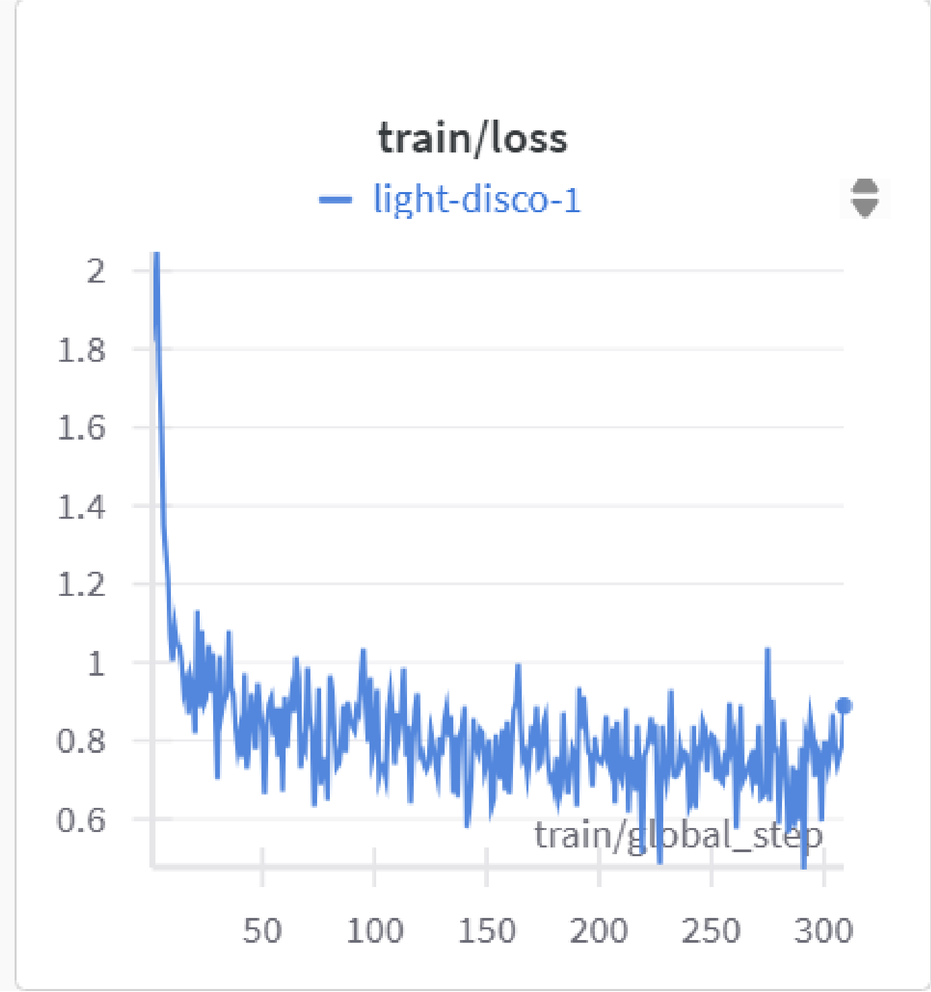
Vulnerability Knowledge Retrieval



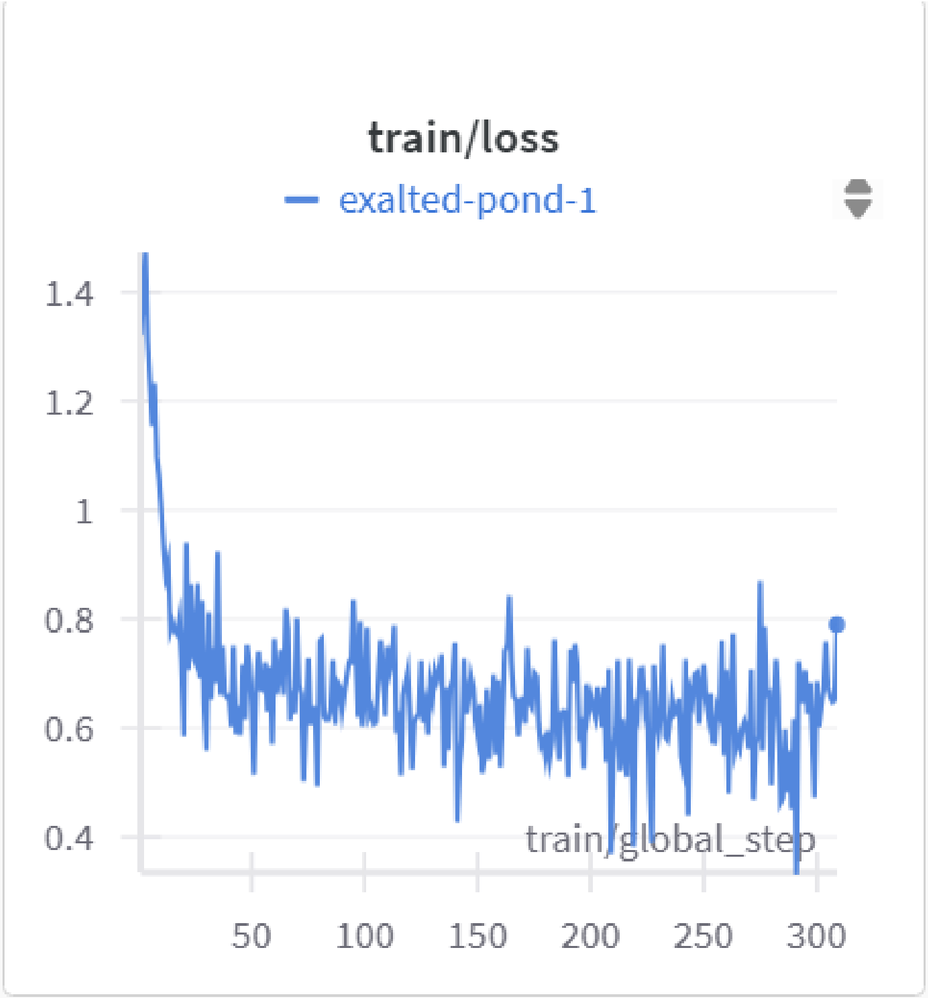
Vulnerability Detection



Vulnerability Detection



Qwen-Coder-2.5-14B-Solidity



DeepSeek-R1-Distill-Llama-8B-Solidity

Evaluation

DoS, Reentrancy, Arithmetic

Accuracy, Precision, Recall, F1-score

Evaluation

	SolidiFI				SmartBugs			
	Acc	***Pre***	***Rec***	***F1***	***Acc***	***Pre***	***Rec***	***F1***
GPT-3.5	0.491	0.590	0.502	0.542	0.519	0.75	0.523	0.585
GPT-3.5-RAG	0.573	0.623	0.501	0.554	0.538	0.75	0.531	0.593
FTQR	**0.832**	**0.713**	**0.523**	**0.6023**	**0.865**	**0.719**	**0.685**	**0.699**
FTDR	0.723	0.667	0.481	0.558	0.731	0.75	0.578	0.651

GPT-3.5 without RAG (GPT-3.5)
GPT-3.5 with RAG (GPT-3.5-RAG)
Fine-tuned Qwen-coder-2.5 with RAG (FTQR)
Fine-tuned Deepseek with RAG (FTDR)

DEMO

kaggle.com/code/nguynngnh/scalm-graph-done/edit

Scalm-graph-DONE Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Code Draft Session (50m)

Main

initialize instance pointing to index to query

```
behaviors_index = Neo4jVector.from_existing_graph(
    embedding=embeddings,
    url=url,
    username=username,
    password=password,
    index_name="Behavior_index",
    node_label="Behavior",
    text_node_properties=["text"],
    embedding_node_property="embedding"
)
```

+ Code + Markdown

[10]:

```
from neo4j import GraphDatabase

driver = GraphDatabase.driver(url, auth=(username, password))
```

Notebook

Input

+ Add Input Upload

DATASETS

private-dataset

private-dataset

Output (18.6MiB / 19.5GiB)

/kaggle/working

Table of contents

Main

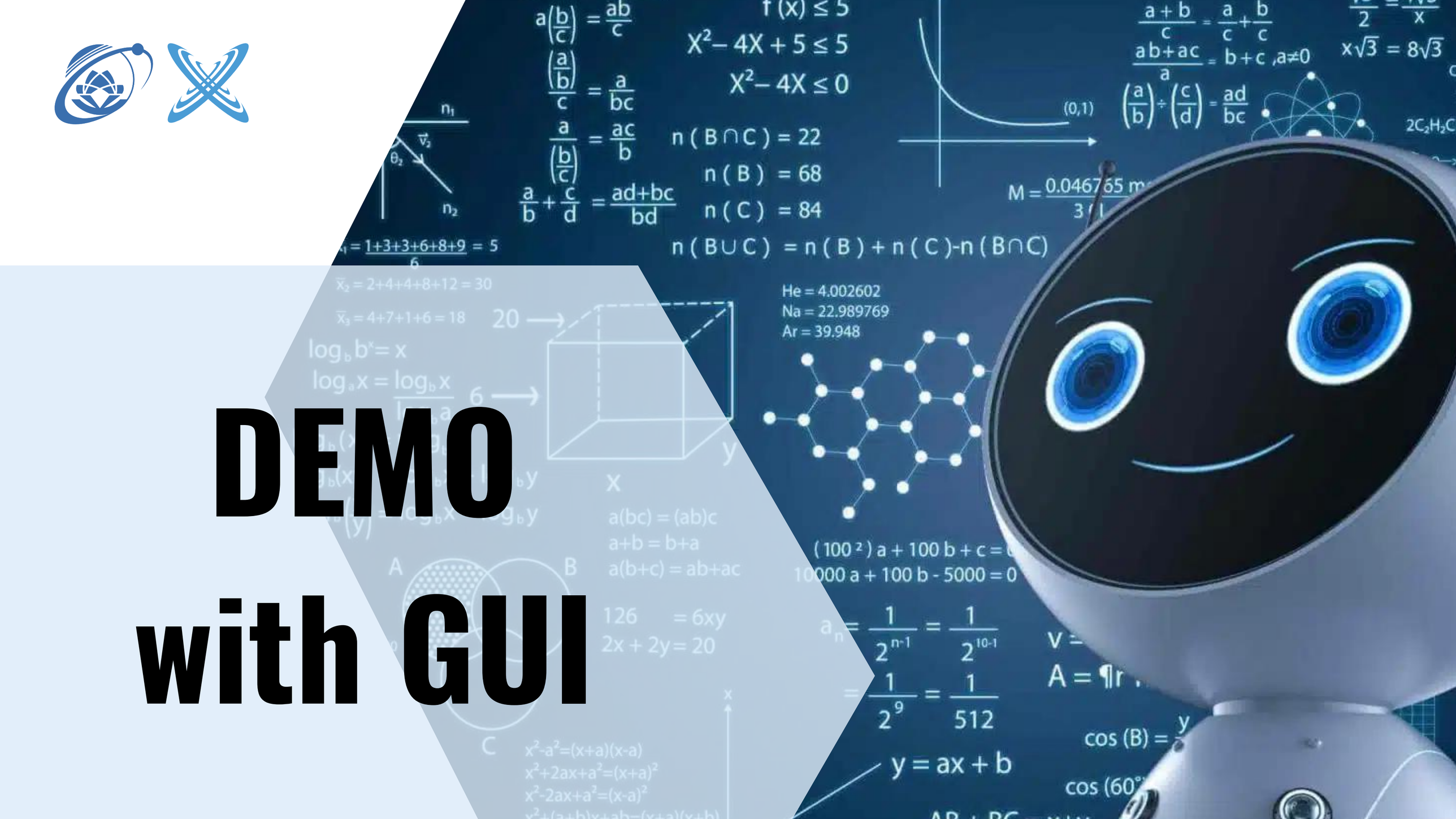
Session options

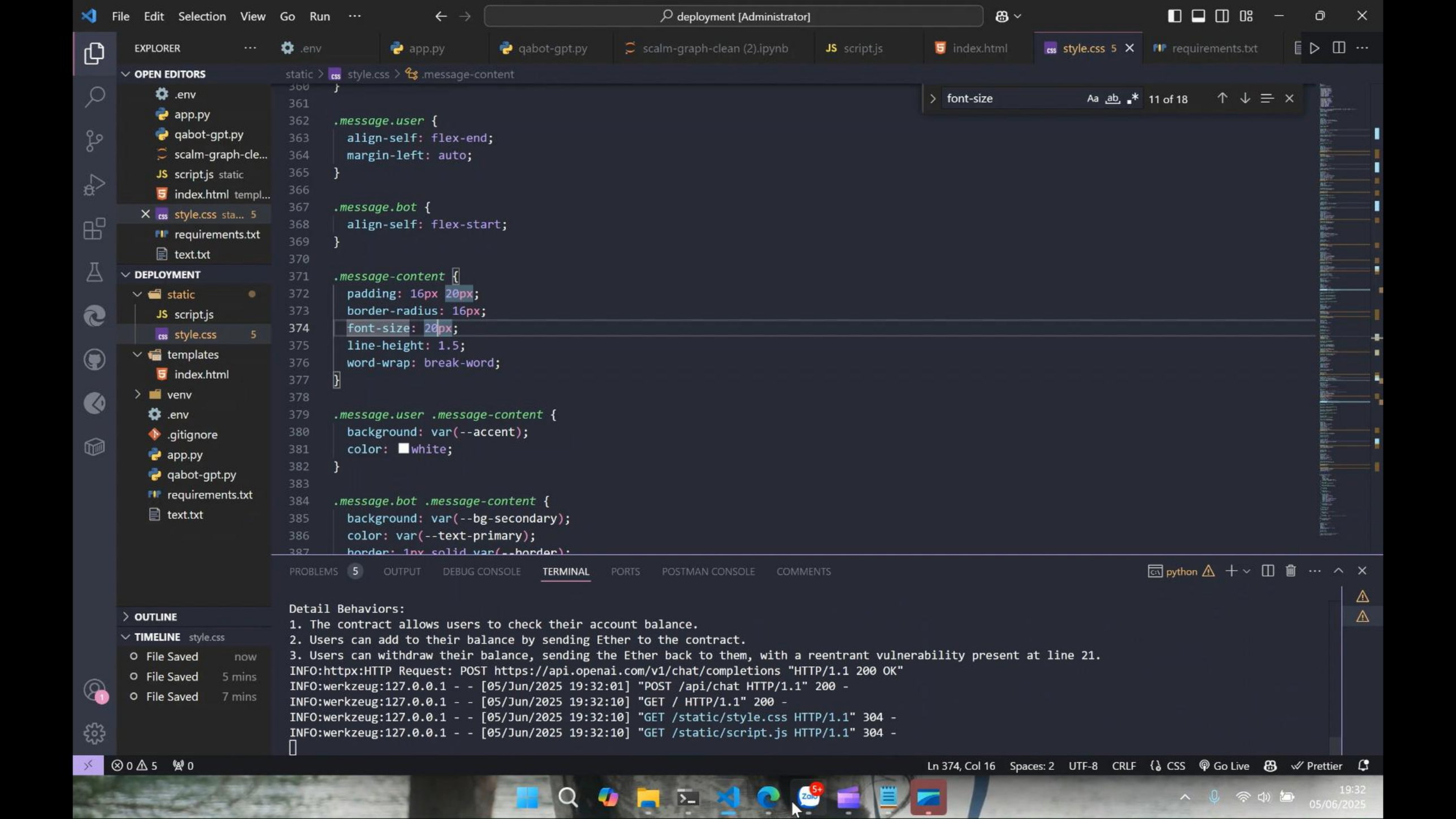
Schedule a notebook to run

Schedule this notebook to run and save a new version on a future date. View all your scheduled notebooks

Search 28°C Nhiều mây VIE 6:33 PM 6/3/2025

DEMO with GUI




File Edit Selection View Go Run
deployment [Administrator]
EXPLORER
.env
app.py
qabot-gpt.py
scalm-graph-clean (2).ipynb
script.js
index.html
style.css 5
requirements.txt
OPEN EDITORS
.env
app.py
qabot-gpt.py
scalm-graph-cle...
script.js static
index.html templ...
style.css sta... 5
requirements.txt
text.txt
DEPLOYMENT
static
script.js
style.css 5
templates
index.html
venv
.env
.gitignore
app.py
qabot-gpt.py
requirements.txt
text.txt
OUTLINE
TIMELINE style.css
File Saved now
File Saved 5 mins
File Saved 7 mins
static > style.css > .message-content
font-size
11 of 18
.message.user {
align-self: flex-end;
margin-left: auto;
}
.message.bot {
align-self: flex-start;
}
.message-content {
padding: 16px 20px;
border-radius: 16px;
font-size: 20px;
line-height: 1.5;
word-wrap: break-word;
}
.message.user .message-content {
background: var(--accent);
color: white;
}
.message.bot .message-content {
background: var(--bg-secondary);
color: var(--text-primary);
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
POSTMAN CONSOLE
COMMENTS
python
Detail Behaviors:
1. The contract allows users to check their account balance.
2. Users can add to their balance by sending Ether to the contract.
3. Users can withdraw their balance, sending the Ether back to them, with a reentrant vulnerability present at line 21.
INFO:httpx:HTTP Request: POST https://api.openai.com/v1/chat/completions "HTTP/1.1 200 OK"
INFO:werkzeug:127.0.0.1 - - [05/Jun/2025 19:32:01] "POST /api/chat HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [05/Jun/2025 19:32:10] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [05/Jun/2025 19:32:10] "GET /static/style.css HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [05/Jun/2025 19:32:10] "GET /static/script.js HTTP/1.1" 304 -
Ln 374, Col 16
Spaces: 2
UTF-8
CRLF
CSS
Go Live
Prettier
19:32
05/06/2025

THANK YOU

FOR YOUR LISTENING!!!