Workflow Optimisation Project

Introduction

- LLMs is an opportunity to improve productivity.
- Productivity requires new workflows.
- Private enterprise data is the difference in the market.
- Make sense to focus on workflows related to private data.

Example Case

- Proof of Concept use private and public data.
  - Private data – reports of penetration tests
  - Public data – PCI DSS Cyber Security Framework

Technologies of Machine Learning

- Machine learning in enterprise settings should focus on private data.
- There are two major ways to implement ML:
  - RAG
  - LLM Tuning

Retrieval-Augmented Generation

- RAG advantages
  - Data manipulation portable between LLM providers.
  - Initial step for LLM tuning
  - Conversion of documents to structured data
- RAG disadvantages
  - Low performance
  - Low precision

LLM Tuning

- Tuning advantages
  - High performance
  - High precision
- Tuning disadvantages
  - Tuning expense
  - Periodic re-tuning required

Summary: RAG as an Entry Point

- RAG solution as an entry to LLM workflows
  - RAG allows to convert unstructured data to structured
  - RAG allows to implement effective authorisation mechanism
  - RAG allows to verify quality of data

Pydantic Agent

- Best of Agentic AI
  - Portable across all major LLM providers
  - Types based on JSON schema
  - Wraps iterative LLM calls
  - Serves as the foundation of many LLM SDKs

Architecture Summary

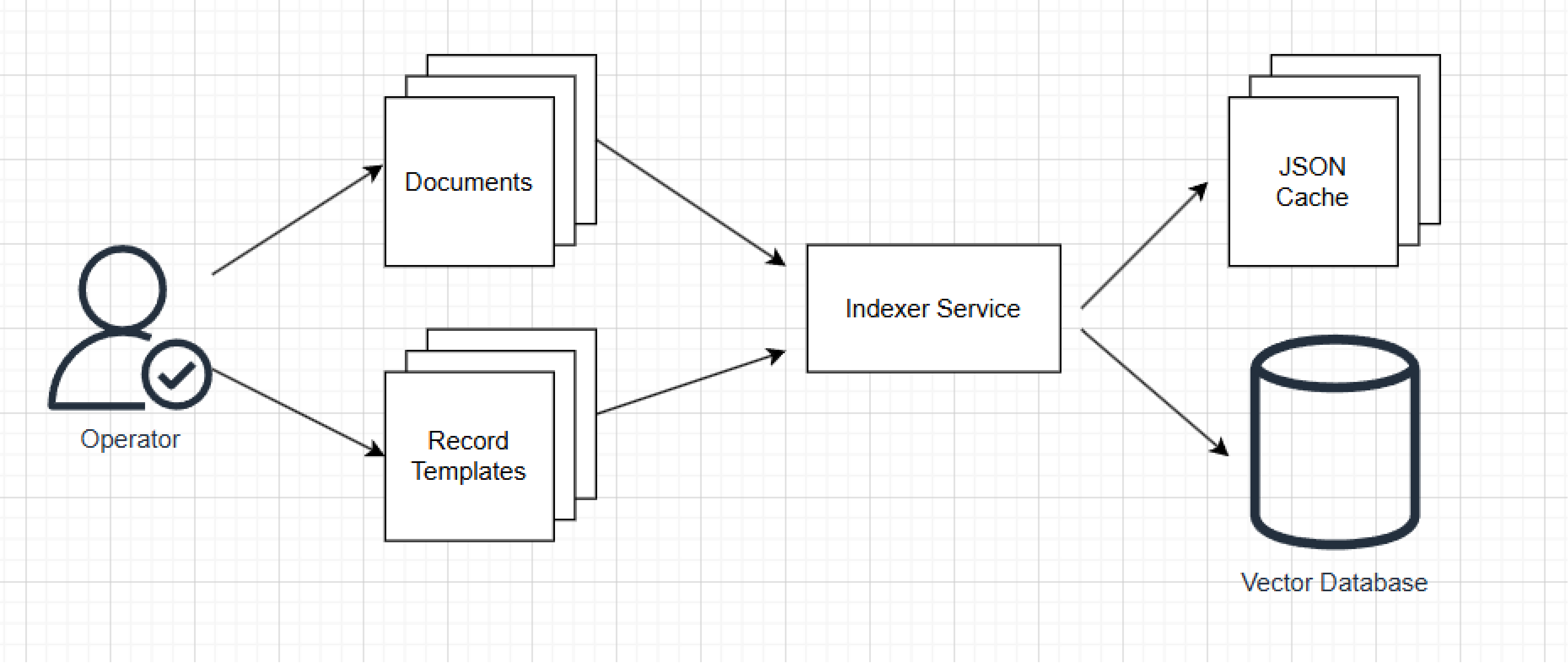
- Example app is a hybrid architecture
  - Cloud-based LLM accessed via REST API
  - Local vector database
  - Dual use web apps and command line scripts

Source Code

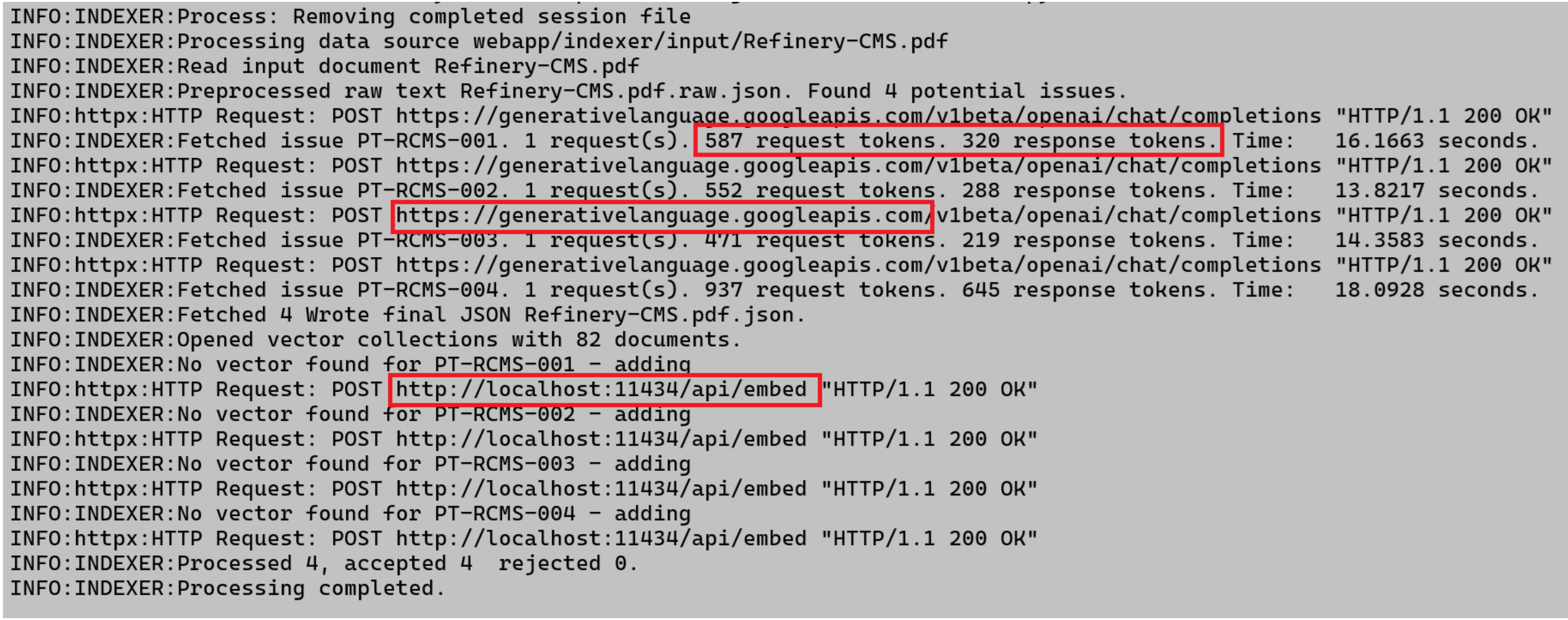
- Public repository on GitHub
  - github.com/DarlowieTechnology/a_bridge_too_far
  - Python, Pydantic, Django, ChromaDB
  - 3 web applications (generator, indexer, query)

Indexer Application

High-Level Indexer Service Design

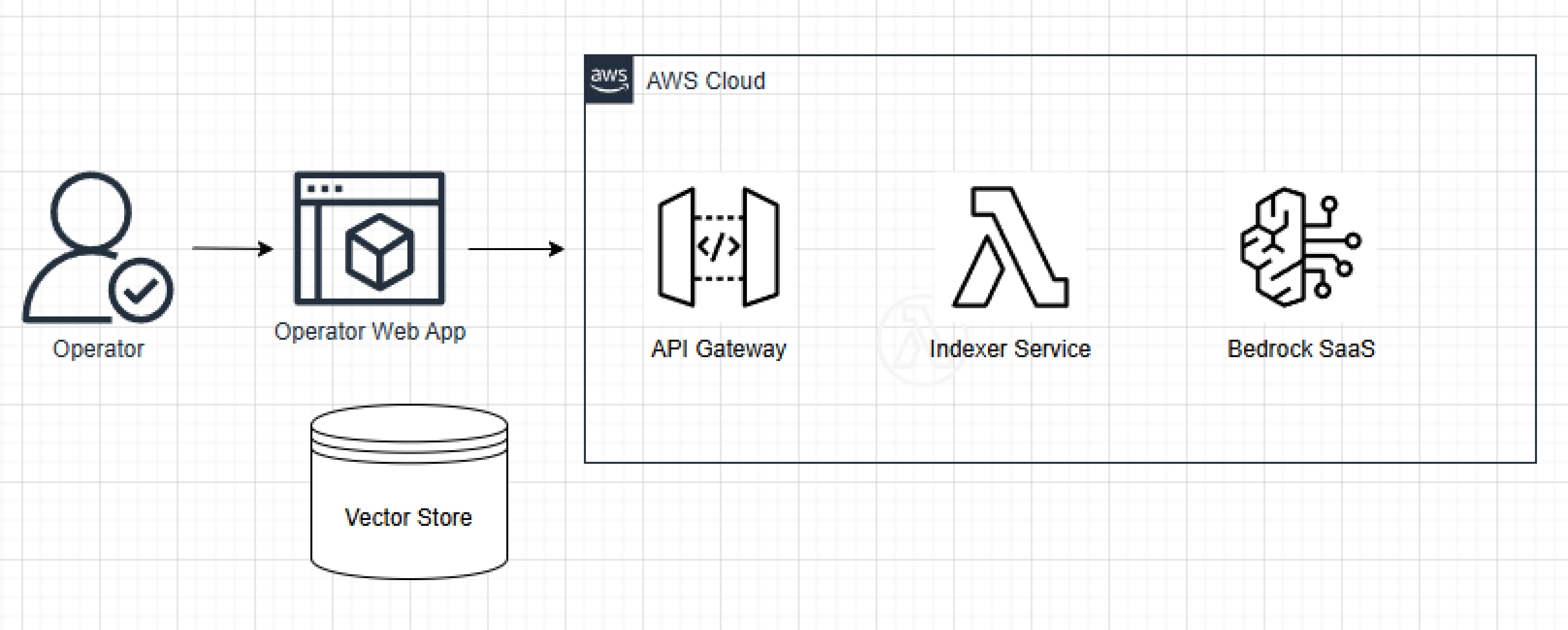


Indexer Output



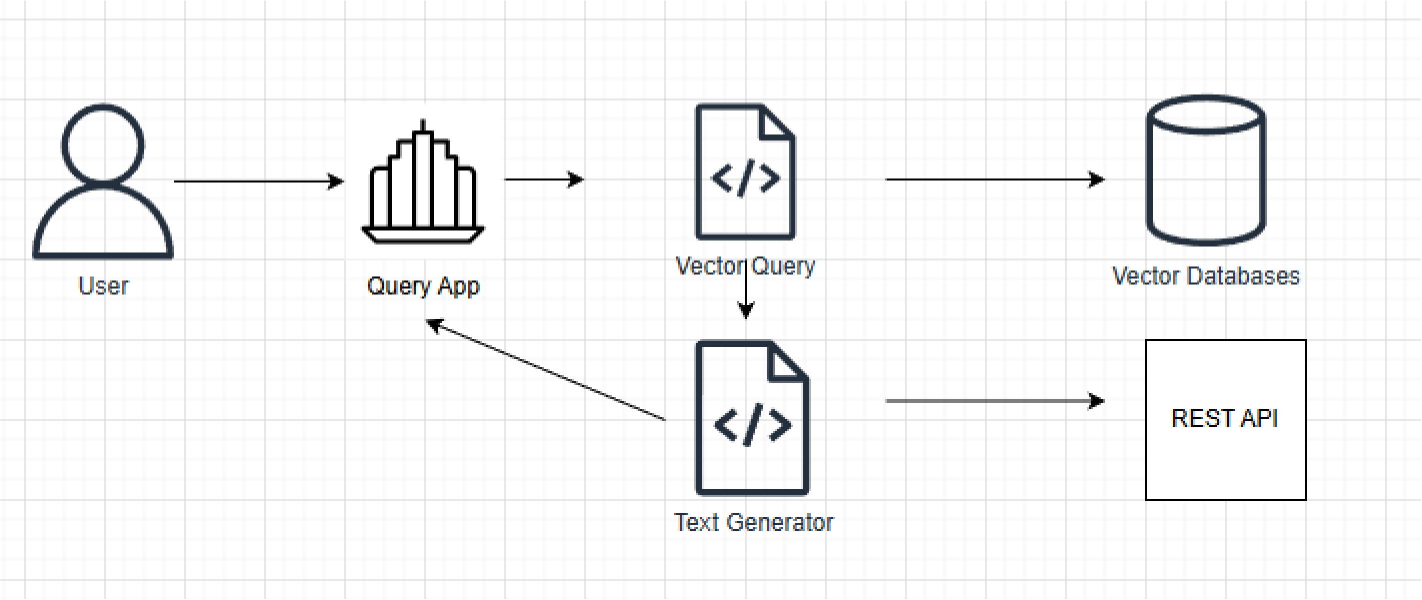
```
INFO:INDEXER:Process: Removing completed session file
INFO:INDEXER:Processing data source webapp/indexer/input/Refinery-CMS.pdf
INFO:INDEXER:Read input document Refinery-CMS.pdf
INFO:INDEXER:Preprocessed raw text Refinery-CMS.pdf.raw.json. Found 4 potential issues.
INFO:httpx:HTTP Request: POST https://generativelanguage.googleapis.com/v1beta/openai/chat/completions "HTTP/1.1 200 OK"
INFO:INDEXER:Fetched issue PT-RCMS-001. 1 request(s). 587 request tokens. 320 response tokens. Time:   16.1663 seconds.
INFO:httpx:HTTP Request: POST https://generativelanguage.googleapis.com/v1beta/openai/chat/completions "HTTP/1.1 200 OK"
INFO:INDEXER:Fetched issue PT-RCMS-002. 1 request(s). 552 request tokens. 288 response tokens. Time:   13.8217 seconds.
INFO:httpx:HTTP Request: POST https://generativelanguage.googleapis.com/v1beta/openai/chat/completions "HTTP/1.1 200 OK"
INFO:INDEXER:Fetched issue PT-RCMS-003. 1 request(s). 471 request tokens. 219 response tokens. Time:   14.3583 seconds.
INFO:httpx:HTTP Request: POST https://generativelanguage.googleapis.com/v1beta/openai/chat/completions "HTTP/1.1 200 OK"
INFO:INDEXER:Fetched issue PT-RCMS-004. 1 request(s). 937 request tokens. 645 response tokens. Time:   18.0928 seconds.
INFO:INDEXER:Fetched 4 Wrote final JSON Refinery-CMS.pdf.json.
INFO:INDEXER:Opened vector collections with 82 documents.
INFO:INDEXER:No vector found for PT-RCMS-001 - adding
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:INDEXER:No vector found for PT-RCMS-002 - adding
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:INDEXER:No vector found for PT-RCMS-003 - adding
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:INDEXER:No vector found for PT-RCMS-004 - adding
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:INDEXER:Processed 4, accepted 4  rejected 0.
INFO:INDEXER:Processing completed.
```

Hybrid Architecture

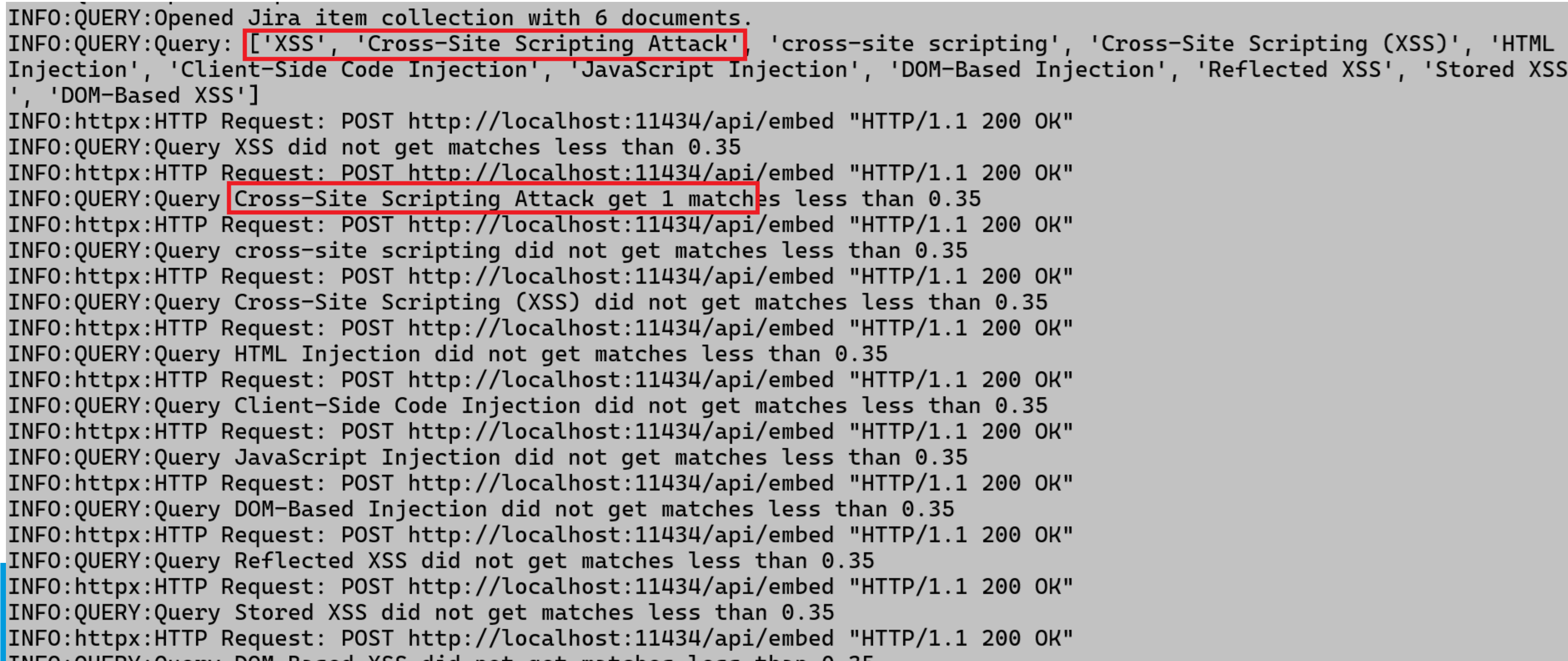


Query Application

High-Level Query App Design

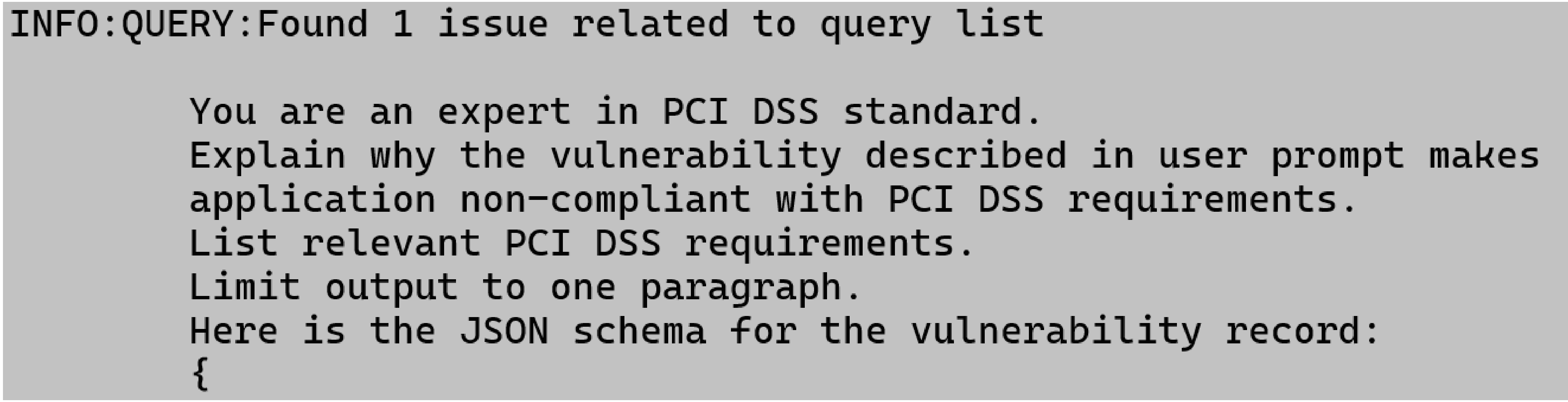


Query RAG Query



```
INFO:QUERY:Opened Jira item collection with 6 documents.
INFO:QUERY:Query: ['XSS', 'Cross-Site Scripting Attack', 'cross-site scripting', 'Cross-Site Scripting (XSS)', 'HTML
Injection', 'Client-Side Code Injection', 'JavaScript Injection', 'DOM-Based Injection', 'Reflected XSS', 'Stored XSS
', 'DOM-Based XSS']
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query XSS did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query Cross-Site Scripting Attack get 1 matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query cross-site scripting did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query Cross-Site Scripting (XSS) did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query HTML Injection did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query Client-Side Code Injection did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query JavaScript Injection did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query DOM-Based Injection did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query Reflected XSS did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query Stored XSS did not get matches less than 0.35
INFO:httpx:HTTP Request: POST http://localhost:11434/api/embed "HTTP/1.1 200 OK"
INFO:QUERY:Query DOM-Based XSS did not get matches less than 0.35
```

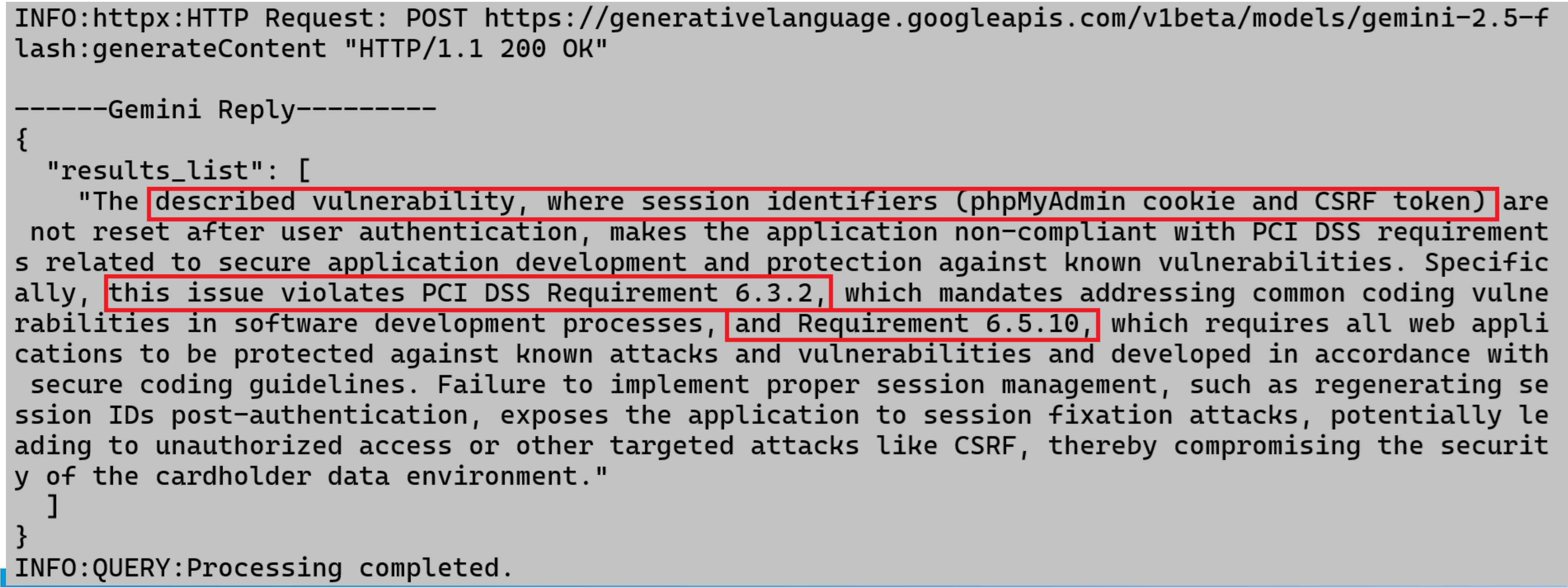
Query Final Query



```
INFO:QUERY:Found 1 issue related to query list

        You are an expert in PCI DSS standard.
        Explain why the vulnerability described in user prompt makes
        application non-compliant with PCI DSS requirements.
        List relevant PCI DSS requirements.
        Limit output to one paragraph.
        Here is the JSON schema for the vulnerability record:
        {
```

Query Final Query



```
INFO:httpx:HTTP Request: POST https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-f
lash:generateContent "HTTP/1.1 200 OK"

------Gemini Reply---------
{
  "results_list": [
    "The described vulnerability, where session identifiers (phpMyAdmin cookie and CSRF token) are
 not reset after user authentication, makes the application non-compliant with PCI DSS requirement
s related to secure application development and protection against known vulnerabilities. Specific
ally, this issue violates PCI DSS Requirement 6.3.2, which mandates addressing common coding vulne
rabilities in software development processes, and Requirement 6.5.10, which requires all web appli
cations to be protected against known attacks and vulnerabilities and developed in accordance with
 secure coding guidelines. Failure to implement proper session management, such as regenerating se
ssion IDs post-authentication, exposes the application to session fixation attacks, potentially le
ading to unauthorized access or other targeted attacks like CSRF, thereby compromising the securit
y of the cardholder data environment."
  ]
}
INFO:QUERY:Processing completed.
```

Query Semantic Search

- Query “*XSS issues*”
-
- Semantic search returns:
-
- *The installed version of Jenkins server was affected by a number of security vulnerabilities. Known vulnerabilities included CSRF and* ***<u>XSS</u>*** *weaknesses.*