



University of Bahrain  
College of Information Technology  
Department of Information Systems  
B.Sc. In Cybersecurity

**SENIOR PROJECT**  
**ACADEMIC YEAR 2025-2026-SEMESTER 2**  
**SECURE NATURAL LANGUAGE TO BASH**  
**EXECUTION - SHELLSENTRY**

**19 May 2026**

Prepared by:

Aljazi Ali Almuhammad - 202202227

Dana Hussain Alhayki - 202203431

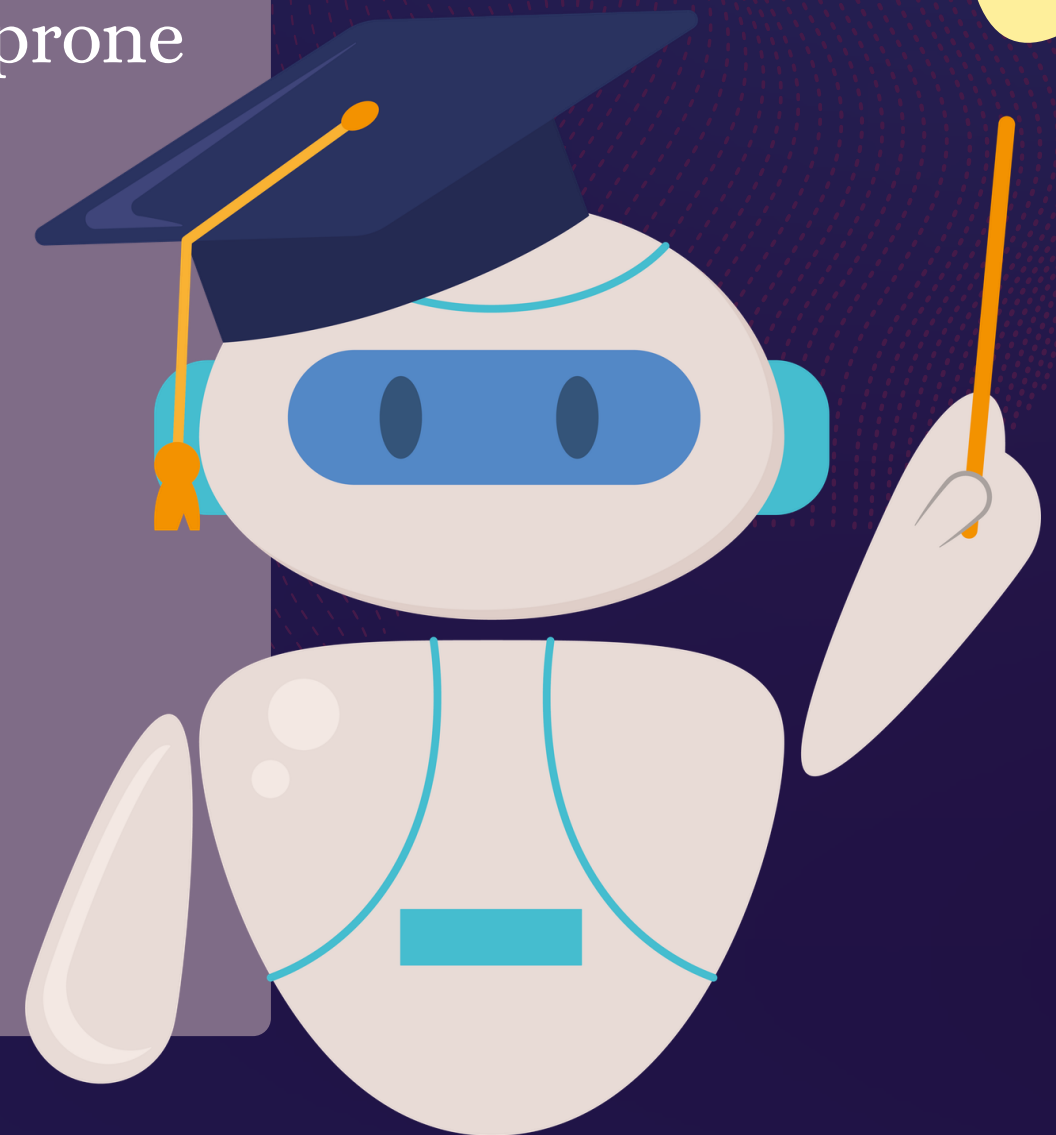
Supervised By :

Dr. Abdulla Khalifa Aldoseri

# PROBLEM STATEMENT

- Linux administration requires Bash expertise
- Manual execution across multiple servers is repetitive and error-prone
- Human mistakes can cause outages or security incidents
- LLMs can generate Bash, but outputs may be unsafe:
- Hallucinated destructive commands
- Prompt injection
- Privilege abuse
- Direct remote execution risks

Need for a secure AI-assisted execution platform



# OBJECTIVES & CONTRIBUTIONS

Secure NL → Bash  
execution

Multi-server SSH  
orchestration

Deterministic  
command validation

Host-aware command  
generation

RAG grounding with  
trusted examples

Audit logging and  
reporting

Script archive + Safe  
scheduling

A major contribution is shifting from “AI assistant” toward a secure operational execution platform.

# RELATED WORK SUMMARY

Study	Focus	Limitation
<b>ScriptSmith</b>	Bash generation + refinement	No secure execution
<b>NL→Bash LLM Study</b>	Functional correctness evaluation	No deployment platform
<b>LLM-as-a-Judge</b>	Validation/refinement	Relies on LLM judgment
<b>Shell-GPT</b>	Real-world command assistant	No execution safety enforcement

**Research gap:** Secure operational execution

# RESEARCH GAP

## Existing solutions provide:

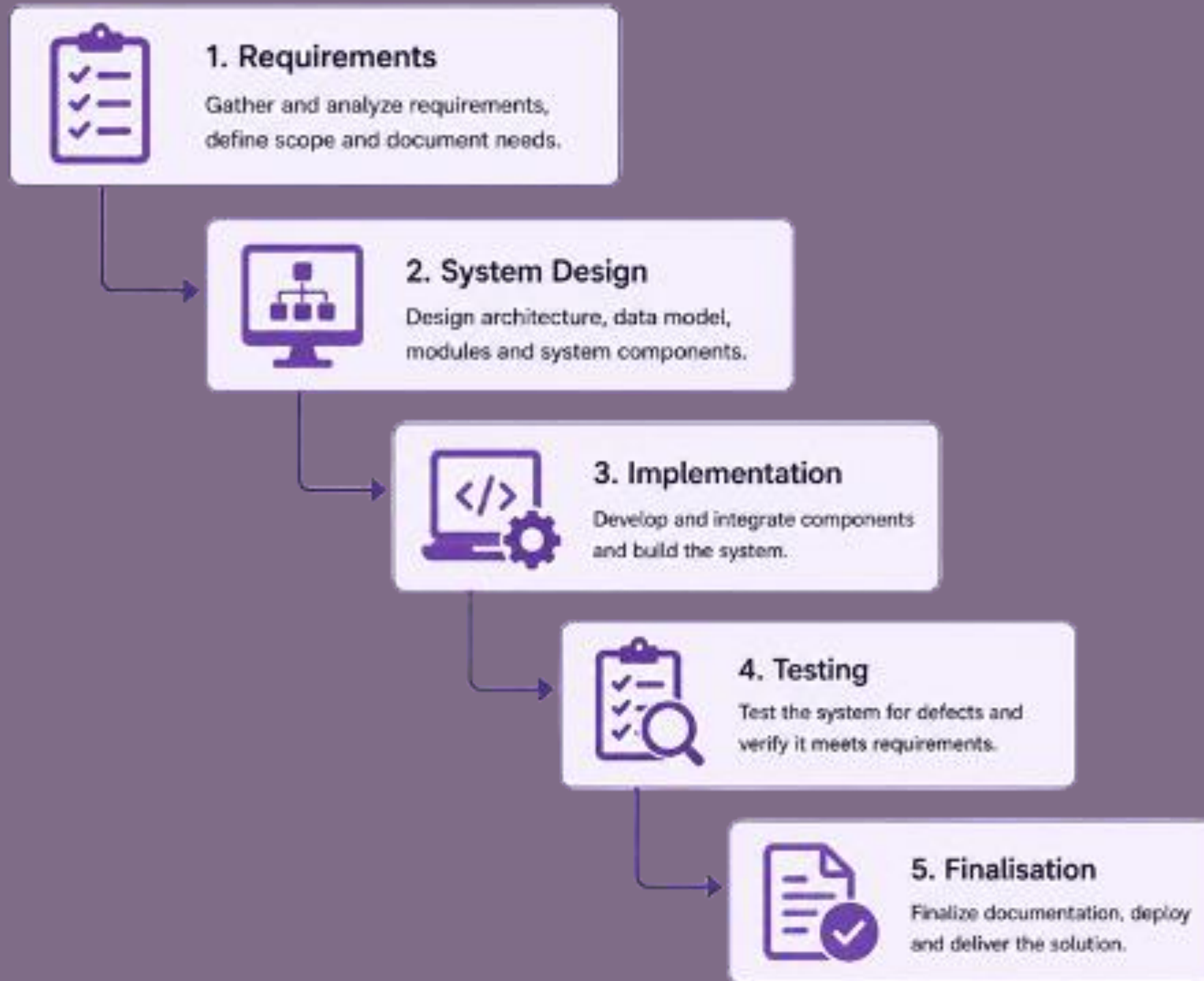
- AI assistance
- Bash generation
- Command refinement

## But lack:

- Audit logging
- Safe cron automation
- Deterministic validation
- Authenticated execution
- SSH multi-host orchestration

**Then:** ShellSentry addresses these gaps

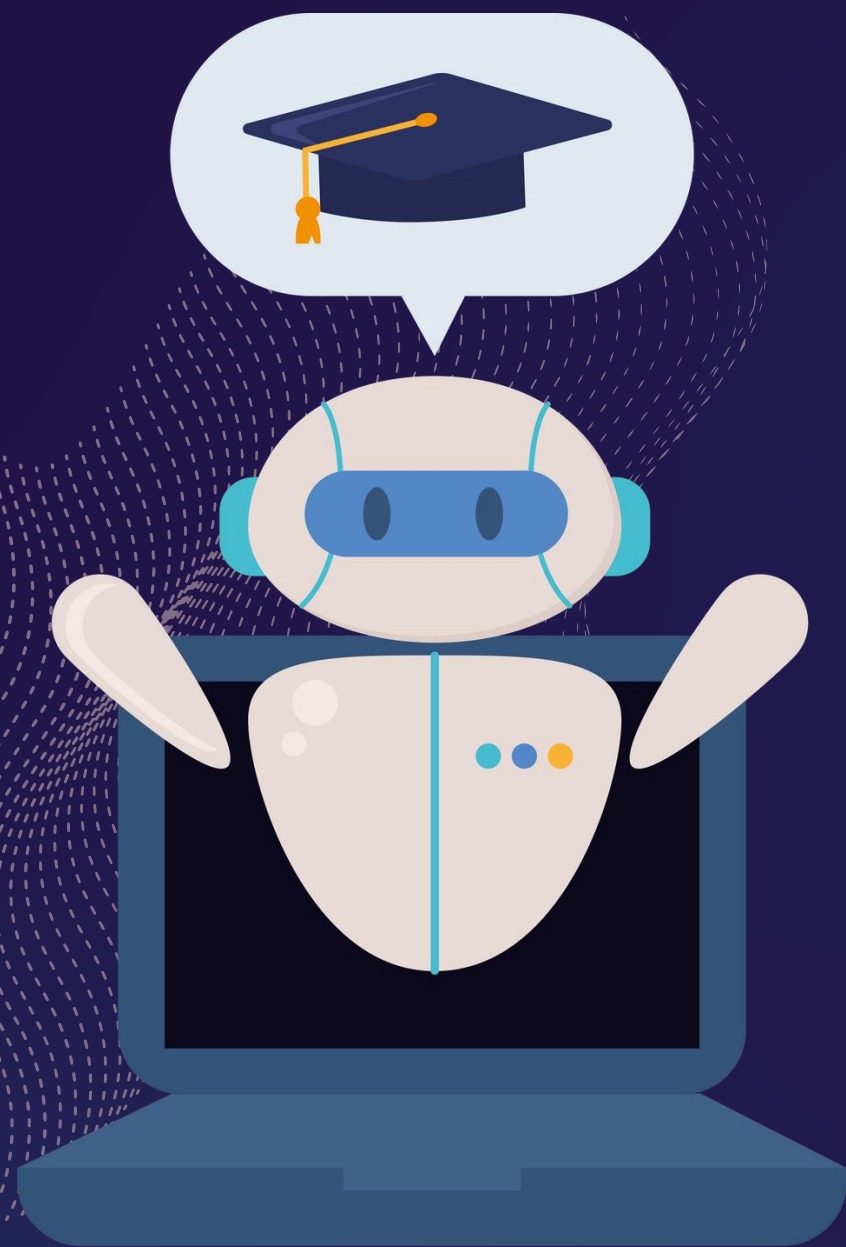
# DEVELOPMENT METHODOLOGY



**Reason:**  
**Structured**  
**academic**  
**development**

# REQUIREMENT COLLECTION

Interviews with 2 cybersecurity specialist



## Key findings:

- Read-only by default
- Validation is mandatory
- Auditability required
- Least privilege
- Unsafe LLM output cannot be trusted

# FUNCTIONAL REQUIREMENTS

NL → Bash  
generation

Host-aware context  
collection

RAG retrieval

Command validation

SSH authentication

Multi-server  
execution

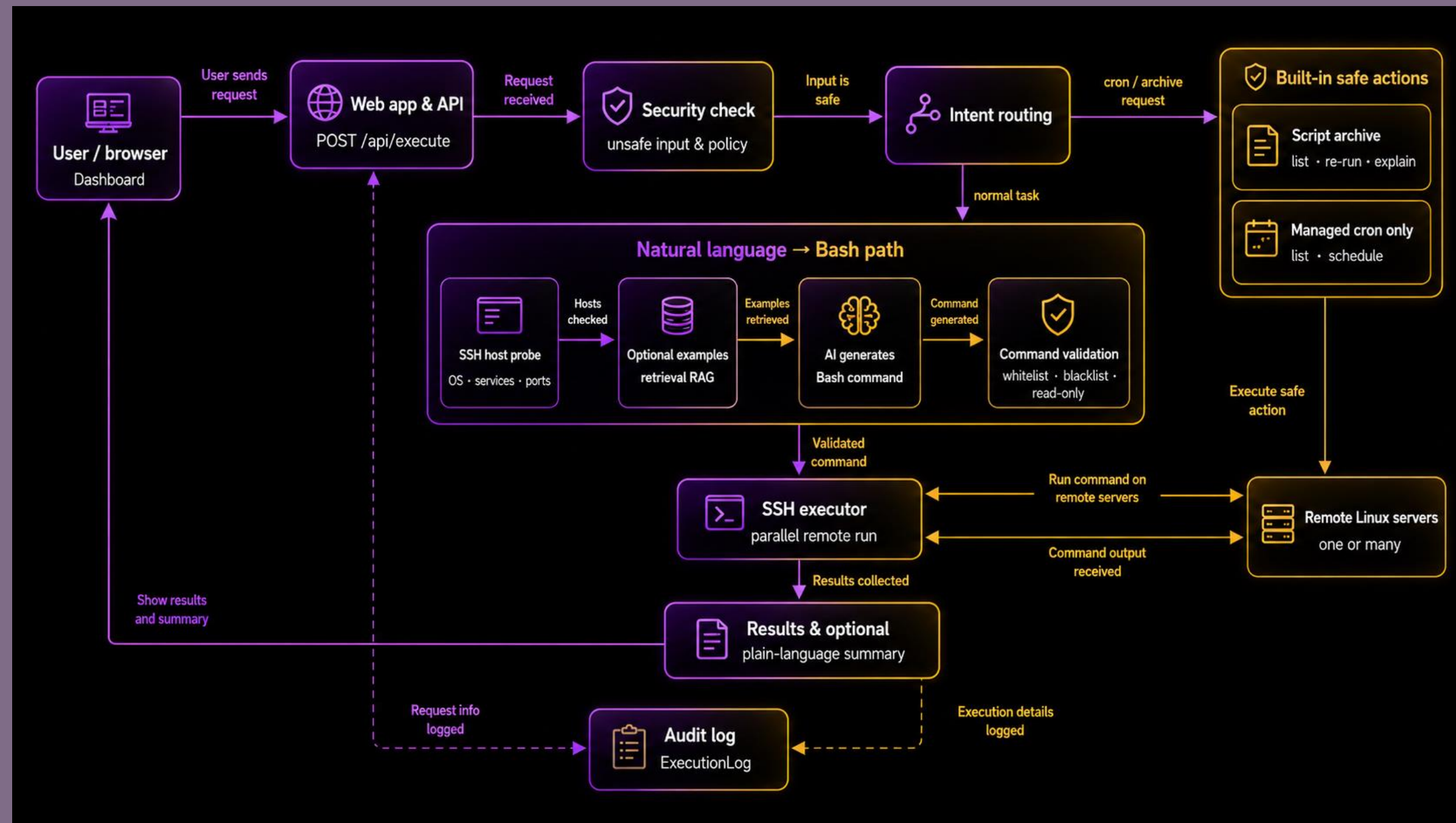
Script archive

Safe scheduling via  
Cron

Audit logging

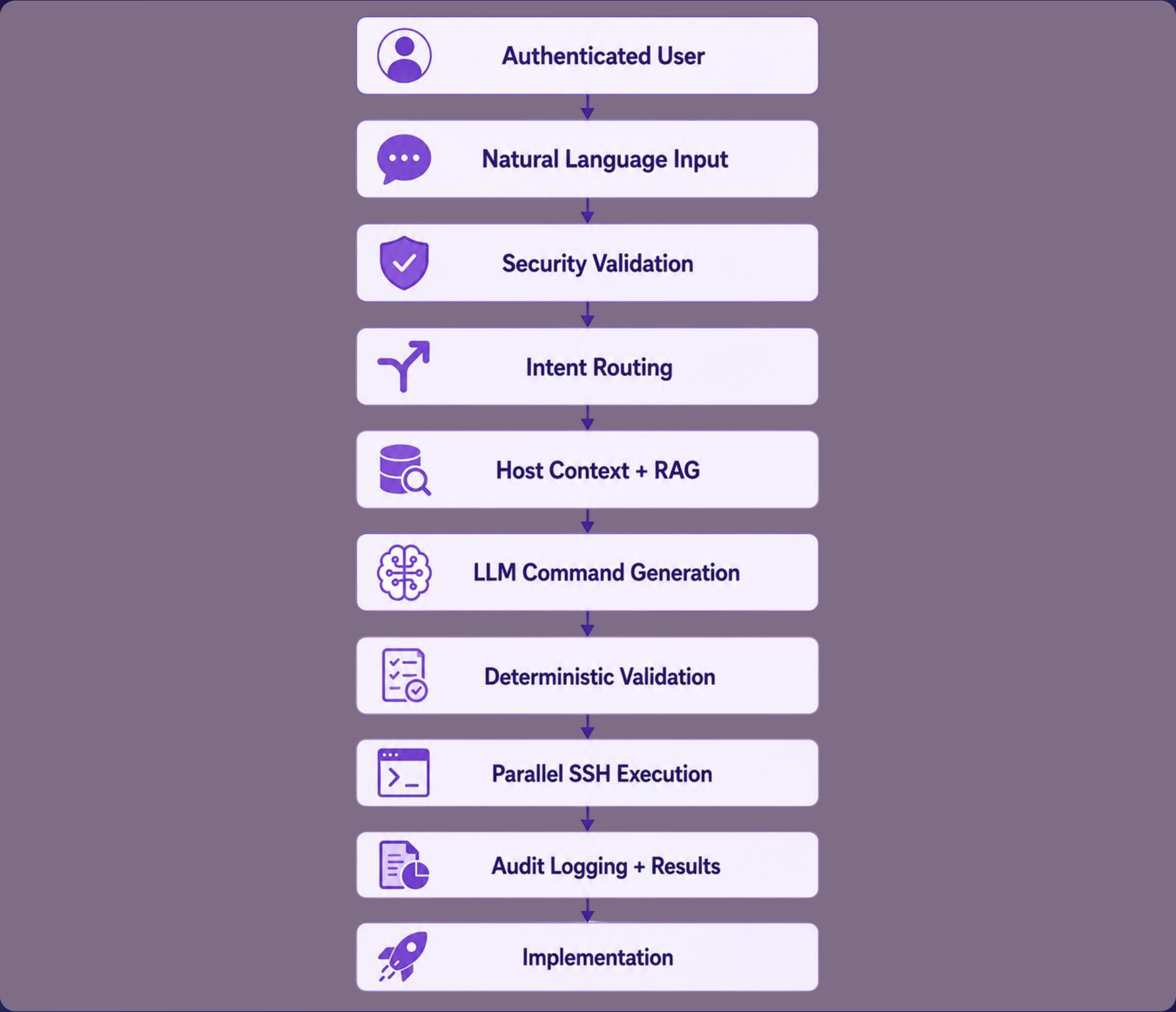
Configurable security  
policies

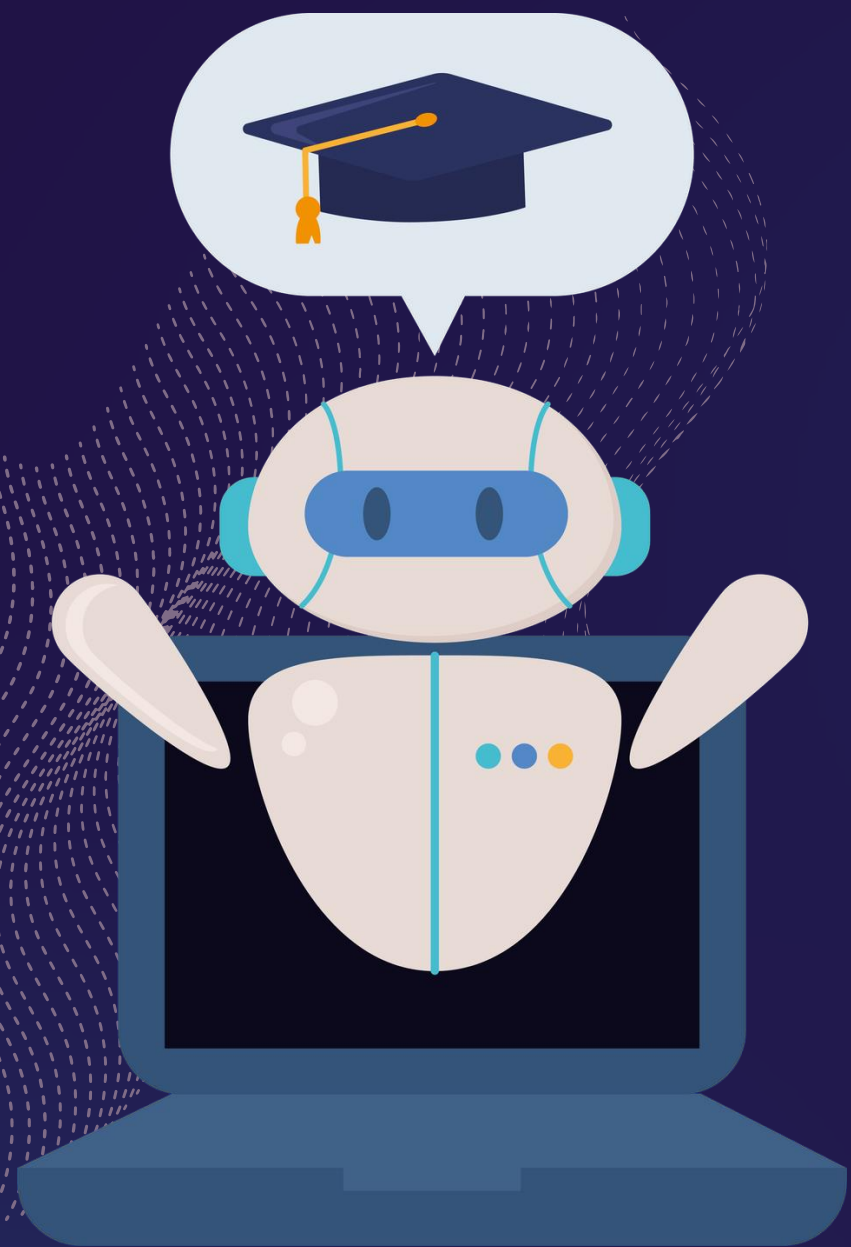
# SYSTEM ARCHITECTURE



Explain flow: User → Flask → Security → Routing → RAG → LLM → Validator → SSH → Results → Database

# EXECUTION PROTOCOL





# SYSTEM DEMONSTRATION

# IMPLEMENTATION STACK

Component	Technology
Frontend	Jinja/HTML/CSS/JS
Backend	Flask
Authentication	Flask-Login
Database	SQLite
SSH	Paramiko
AI model	llama-3.1-8b-instant / llama-3.3-70b versatile
RAG	Sentence Transformers + FAISS

# FUNCTIONAL TESTING RESULTS

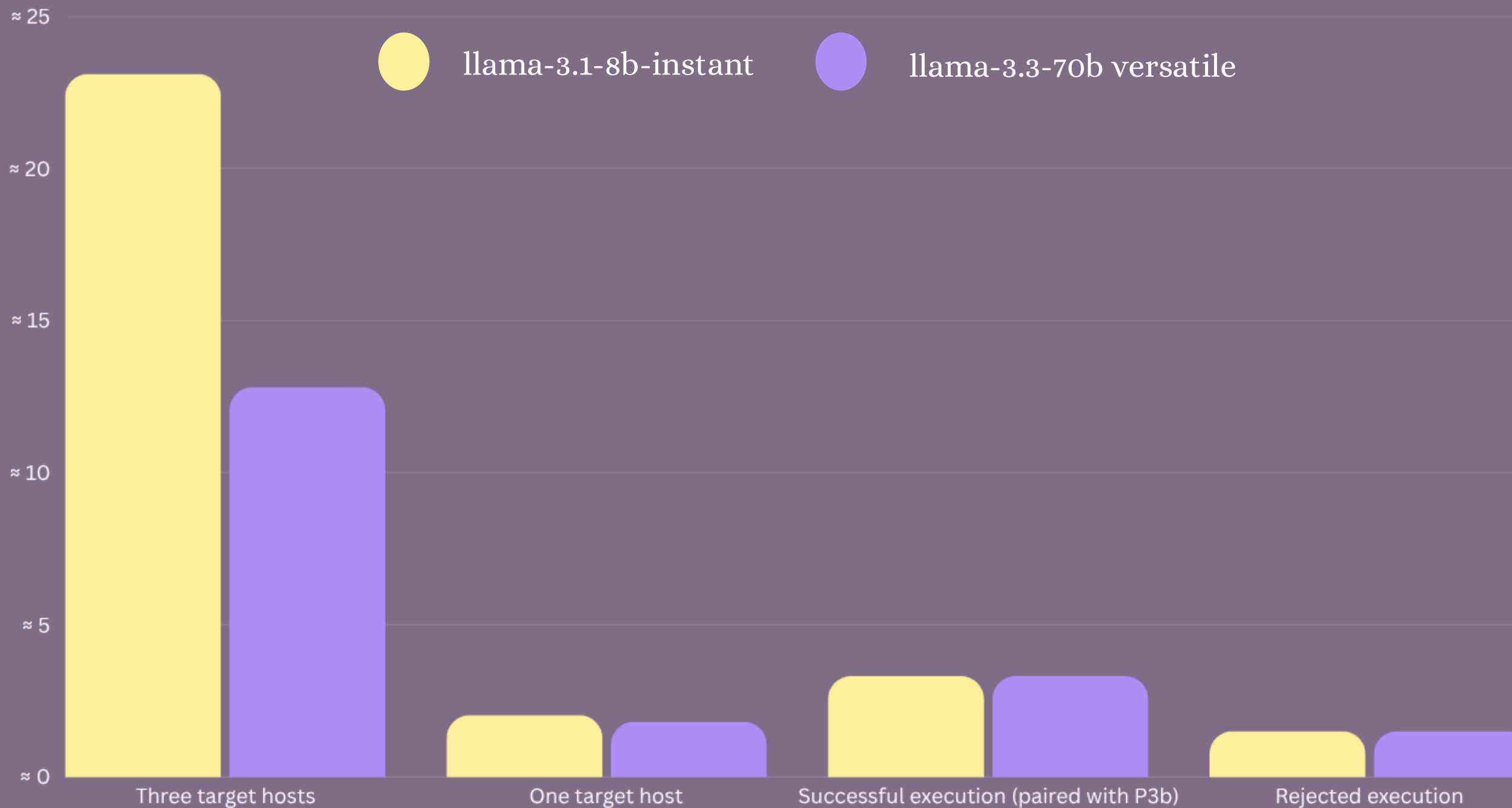
Test	Result
Authentication	Pass
NL→Bash	Pass
Multi-host execution	Pass
Archive scripts	Pass
Safe scheduling via Cron	Pass

# SECURITY TESTING RESULTS

<b>Security Test</b>	<b>Result</b>
<b>SQL Injection</b>	Pass
<b>Prompt Injection</b>	Pass
<b>Unsafe Command</b>	Blocked
<b>Root Execution</b>	Blocked
<b>Cron Deletion</b>	Blocked
<b>SSH Credential Failure</b>	Handled

# PERFORMANCE RESULTS

## Parallel execution improves scalability



# COMPARISON WITH Shell-GPT

ShellSentry focuses on  
secure execution, not  
just generation

Feature	ShellSentry	Shell-GPT
Bash generation	Yes	Yes
Validation	Yes	Limited
SSH execution	Yes	No
Multi-host	Yes	No
Audit logging	Yes	No
Safe scheduling via Cron	Yes	No

# CONCLUSION

- Secure AI-assisted Linux administration prototype
- Validated NL→Bash execution
- Multi-host SSH orchestration
- Strong security enforcement
- Successful testing results

# FUTURE Work

- RBAC
- Human approval workflow
- Dry-run mode
- SIEM integration
- Production deployment



# THANK YOU FOR YOUR ATTENTION

We sincerely appreciate your time, interest, and support throughout our presentation.

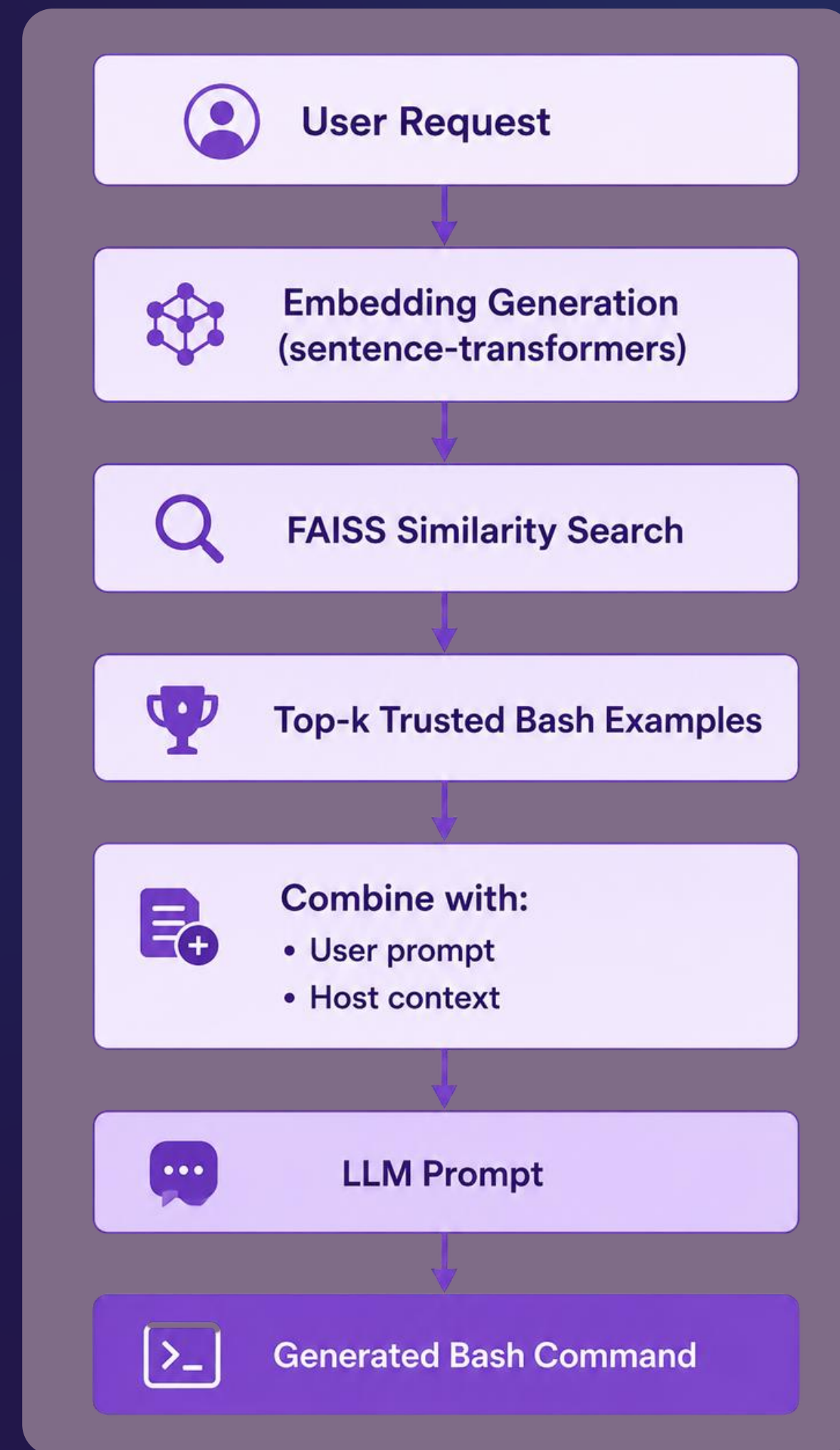
(وَأَخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ)

اللهم لك الحمد حتى ترضى ولك الحمد إذا رضيت ولك الحمد بعد الرضى.

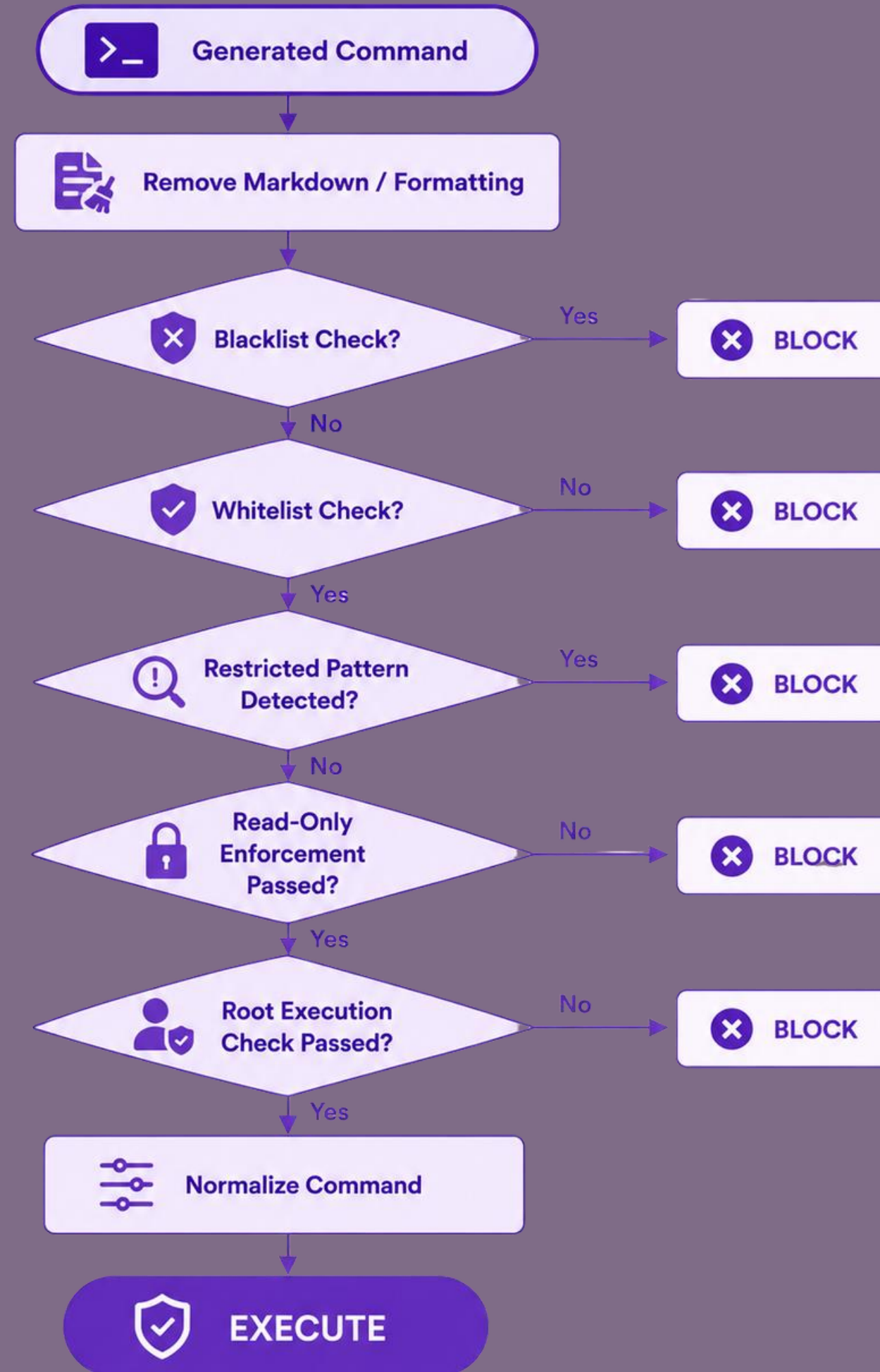
**ANY  
QUESTIONS?**



# RETRIEVAL AUGMENTED GENERATION PIPELINE



# DETERMINISTIC COMMAND VALIDATION



# SYSTEM CONFIGURATION CONTROLS

- environment variables
- no hardcoded secrets
- least privilege

Setting	Default
<b>READ_ONLY_EXECUTION</b>	TRUE
<b>ALLOW_ROOT_EXECUTION</b>	FALSE
<b>SSH authentication</b>	Password / Key / Agent
<b>REMOTE_SERVERS</b>	Configurable
<b>SERVER_CREDENTIALS</b>	Per-host supported
<b>LLM provider</b>	Configurable

# SYSTEM INTERFACE

ShellSentry  
LLM-TO-BASH · SECURE

Secure by design

## Welcome back, operator.

Run natural-language requests across your servers with policy checks, SSH execution, and a clear audit trail—without leaving this console.

- LLM-to-bash with validation before anything runs remotely
- Multi-host execution with per-server results you can trust
- Session-based access—your credentials stay on your terms

### Sign in

Use your account to open the command dashboard.

Username  
dalhayki

Password  
\*\*\*\*\*

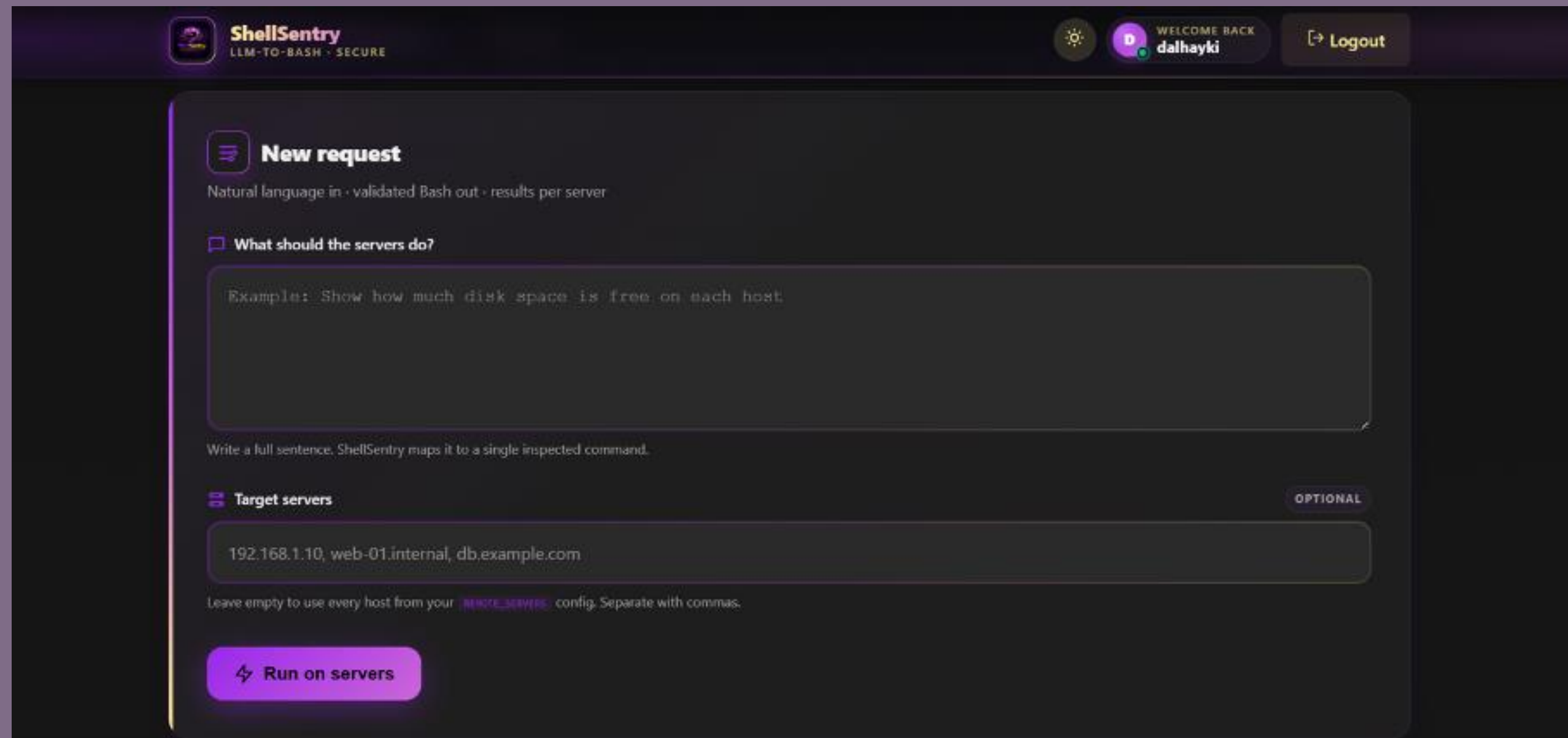
Continue to dashboard →

New here?  
[Create an account](#)

ShellSentry — secure natural-language command execution.

Login page

# SYSTEM INTERFACE

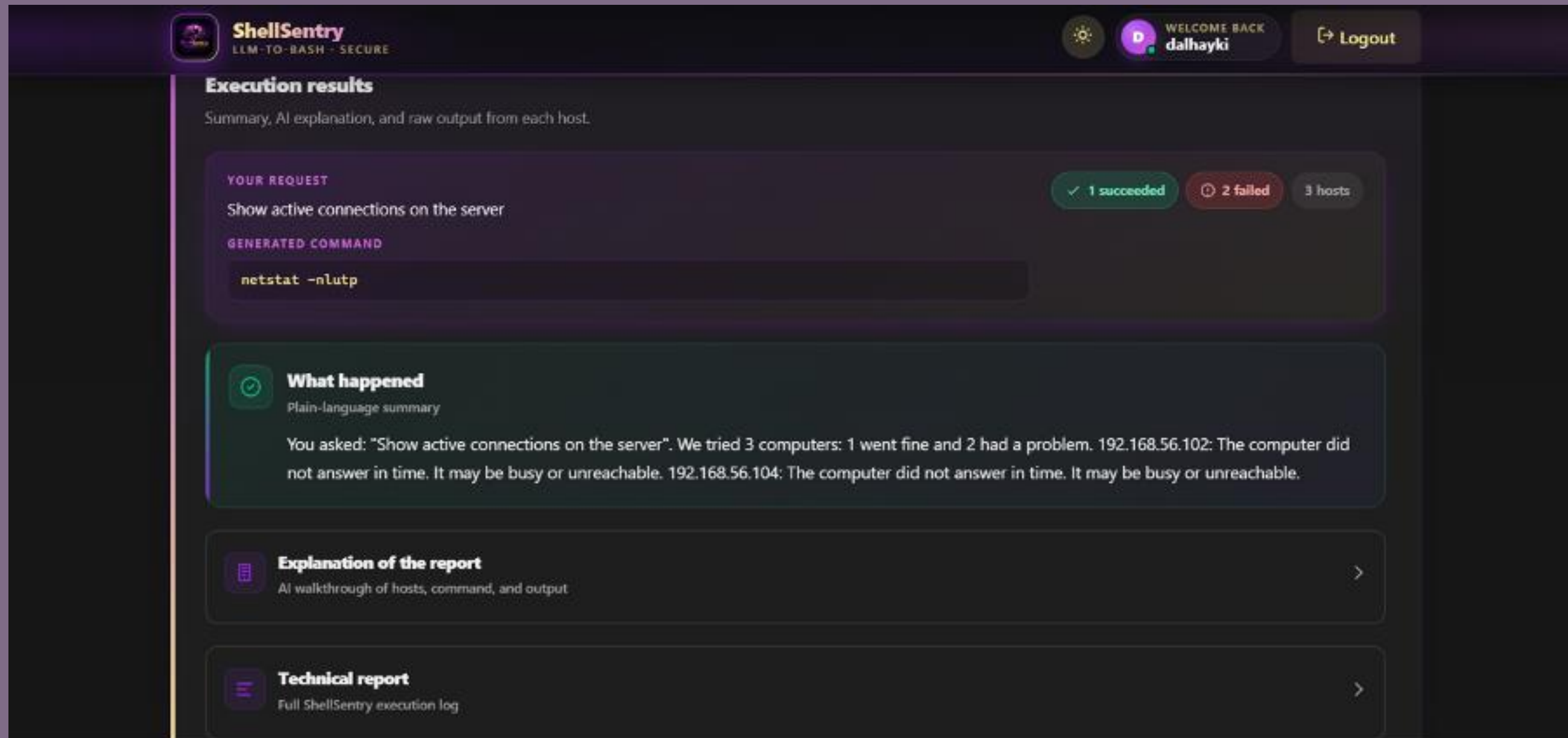


The screenshot displays the ShellSentry dashboard. At the top left is the ShellSentry logo with the tagline 'LLM-TO-BASH · SECURE'. At the top right, there is a settings icon, a user profile for 'dalhayki' with the text 'WELCOME BACK', and a 'Logout' button. The main content area is titled 'New request' and includes a sub-header 'Natural language in · validated Bash out · results per server'. Below this is a section 'What should the servers do?' with a text input field containing the example text: 'Example: Show how much disk space is free on each host.' A note below the input field reads: 'Write a full sentence. ShellSentry maps it to a single inspected command.' The next section is 'Target servers' with an 'OPTIONAL' label and a text input field containing '192.168.1.10, web-01.internal, db.example.com'. A note below this field reads: 'Leave empty to use every host from your `known_hosts` config. Separate with commas.' At the bottom of the form is a purple button labeled 'Run on servers'.

Dashboard

Extras

# SYSTEM INTERFACE

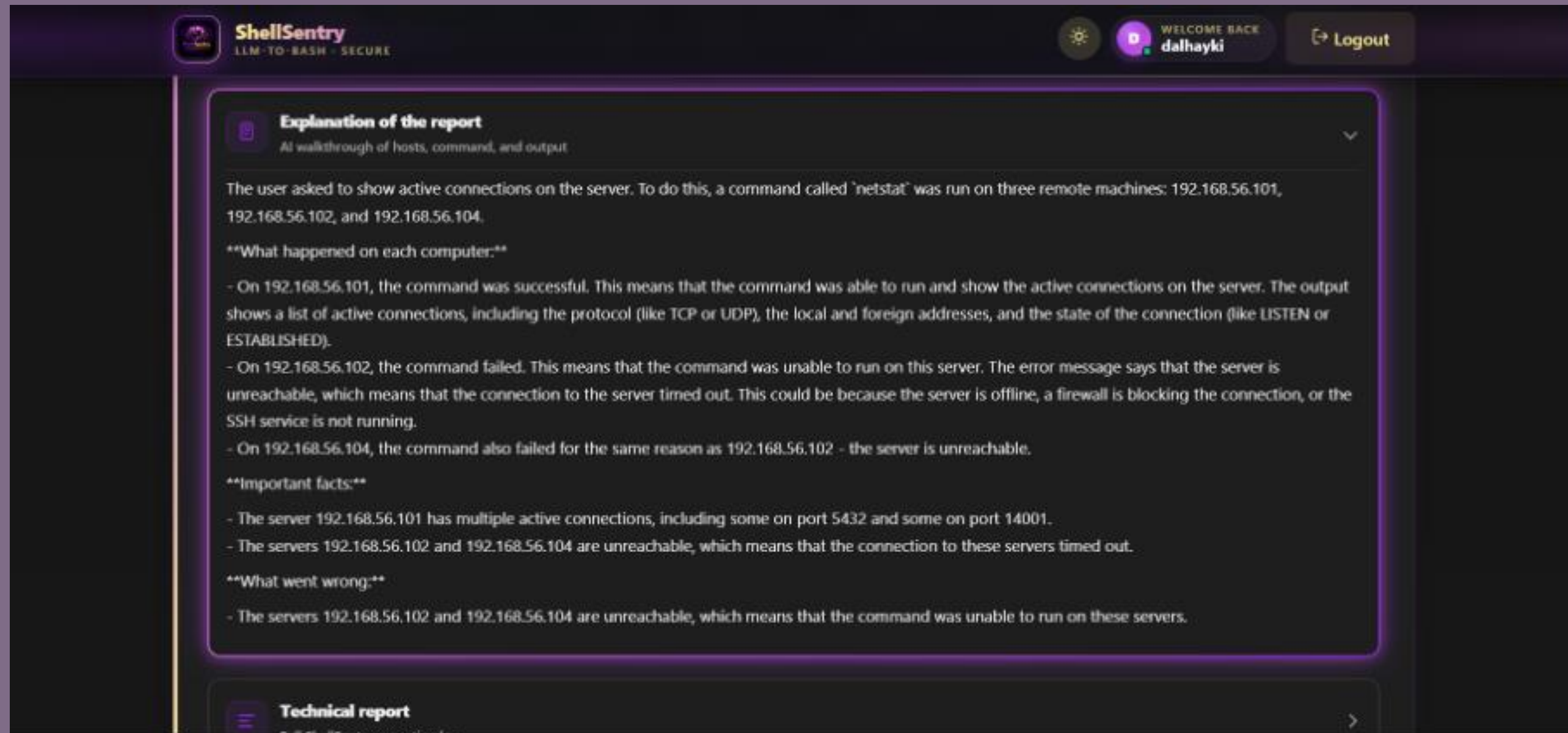


The screenshot displays the ShellSentry interface with the following components:

- Header:** ShellSentry logo (LLM-TO-BASH - SECURE), user profile (dalhayki), and a Logout button.
- Section: Execution results**
  - Summary: AI explanation, and raw output from each host.
  - YOUR REQUEST:** Show active connections on the server. Status: 1 succeeded, 2 failed, 3 hosts.
  - GENERATED COMMAND:** `netstat -nltup`
  - What happened:** Plain-language summary. Text: "You asked: 'Show active connections on the server'. We tried 3 computers: 1 went fine and 2 had a problem. 192.168.56.102: The computer did not answer in time. It may be busy or unreachable. 192.168.56.104: The computer did not answer in time. It may be busy or unreachable."
  - Explanation of the report:** AI walkthrough of hosts, command, and output.
  - Technical report:** Full ShellSentry execution log.

Result page

# SYSTEM INTERFACE



The screenshot displays the ShellSentry web interface. At the top left, the logo 'ShellSentry' is visible with the tagline 'LLM-TO-BASH - SECURE'. On the top right, there is a user profile for 'dalhayki' with the text 'WELCOME BACK' and a 'Logout' button. The main content area features a report titled 'Explanation of the report' with a sub-header 'AI walkthrough of hosts, command, and output'. The report text explains that the user requested active connections on the server, and a 'netstat' command was run on three remote machines: 192.168.56.101, 192.168.56.102, and 192.168.56.104. It details the success of the command on the first machine and the failure on the other two due to unreachability. Below this, 'Important facts' and 'What went wrong' sections provide further context. At the bottom of the report, a 'Technical report' section is partially visible.

**ShellSentry**  
LLM-TO-BASH - SECURE

WELCOME BACK  
dalhayki Logout

### Explanation of the report

AI walkthrough of hosts, command, and output

The user asked to show active connections on the server. To do this, a command called `netstat` was run on three remote machines: 192.168.56.101, 192.168.56.102, and 192.168.56.104.

**What happened on each computer:**

- On 192.168.56.101, the command was successful. This means that the command was able to run and show the active connections on the server. The output shows a list of active connections, including the protocol (like TCP or UDP), the local and foreign addresses, and the state of the connection (like LISTEN or ESTABLISHED).
- On 192.168.56.102, the command failed. This means that the command was unable to run on this server. The error message says that the server is unreachable, which means that the connection to the server timed out. This could be because the server is offline, a firewall is blocking the connection, or the SSH service is not running.
- On 192.168.56.104, the command also failed for the same reason as 192.168.56.102 - the server is unreachable.

**Important facts:**

- The server 192.168.56.101 has multiple active connections, including some on port 5432 and some on port 14001.
- The servers 192.168.56.102 and 192.168.56.104 are unreachable, which means that the connection to these servers timed out.

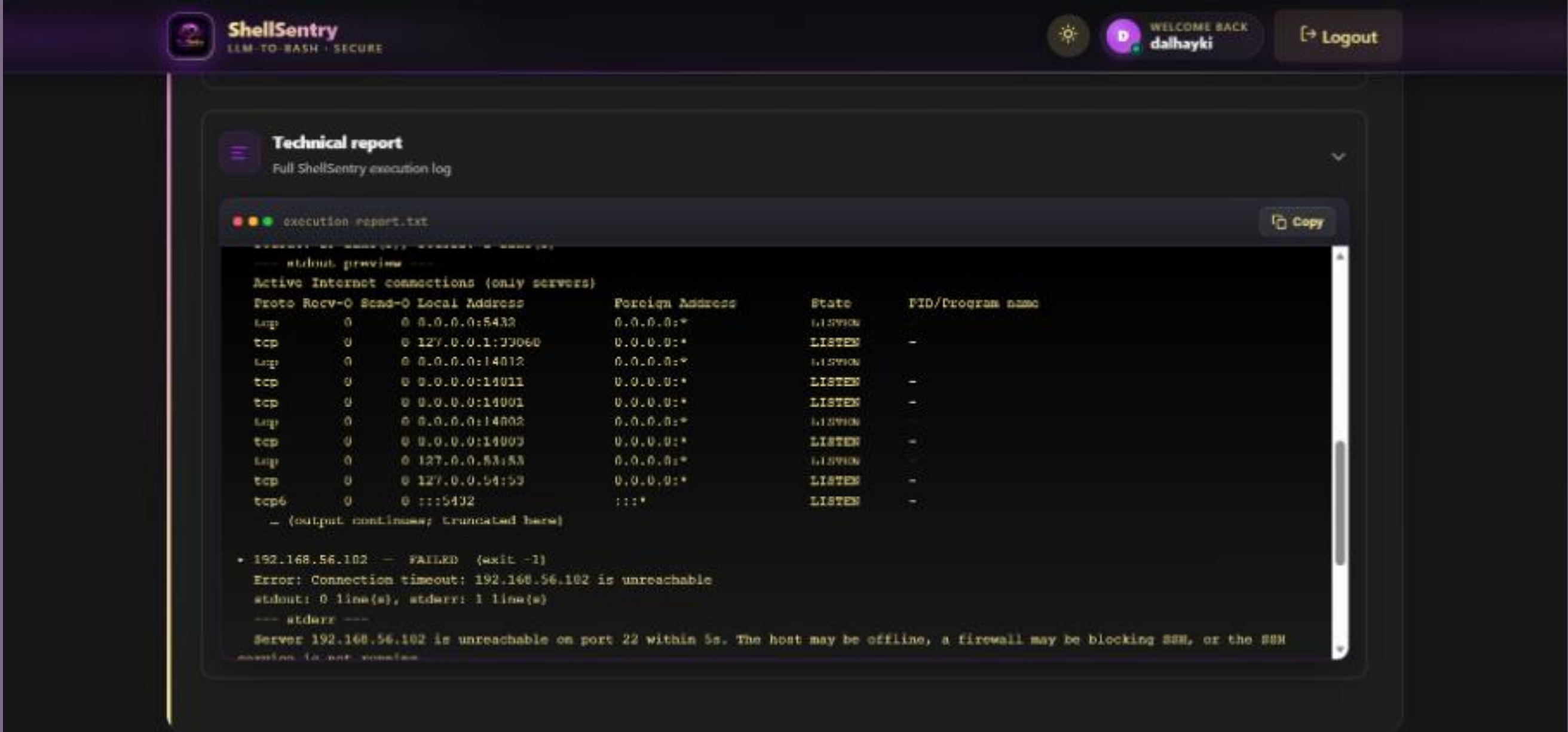
**What went wrong:**

- The servers 192.168.56.102 and 192.168.56.104 are unreachable, which means that the command was unable to run on these servers.

### Technical report

Result page

# SYSTEM INTERFACE



The screenshot displays the ShellSentry web interface. At the top left is the ShellSentry logo with the tagline "LLM-TO-BASH · SECURE". On the top right, there is a user profile for "dalhayki" with a "Logout" button. The main content area is titled "Technical report" and shows a "Full ShellSentry execution log". A terminal window titled "execution report.txt" is open, displaying the following output:

```
----- stdout preview -----  
Active Internet connections (only servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name  
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      -  
tcp        0      0 127.0.0.1:22060         0.0.0.0:*               LISTEN      -  
tcp        0      0 0.0.0.0:14012          0.0.0.0:*               LISTEN      -  
tcp        0      0 0.0.0.0:14011          0.0.0.0:*               LISTEN      -  
tcp        0      0 0.0.0.0:14001          0.0.0.0:*               LISTEN      -  
tcp        0      0 0.0.0.0:14002          0.0.0.0:*               LISTEN      -  
tcp        0      0 0.0.0.0:14003          0.0.0.0:*               LISTEN      -  
tcp        0      0 127.0.0.53:53          0.0.0.0:*               LISTEN      -  
tcp        0      0 127.0.0.54:53          0.0.0.0:*               LISTEN      -  
tcp6       0      0 :::5432                :::*                    LISTEN      -  
- (output continues; truncated here) -  
  
* 192.168.56.102 -- FAILED (exit -1)  
Error: Connection timeout: 192.168.56.102 is unreachable  
stdout: 0 line(s), stderr: 1 line(s)  
----- stderr -----  
Server 192.168.56.102 is unreachable on port 22 within 5s. The host may be offline, a firewall may be blocking SSH, or the SSH  
connection is not possible.
```

Result page

Extras