

# InsightAI:

## Log and Trace Analysis using an Agentic RAG Framework



Roozbeh Aghili



Hersh Hajare



Prince Kapoor



Anurag Prakash

May 2025

# Sherlog: Finding a needle in a haystack



# First, let's introduce Sherlog!



In-depth log analysis  
Visited: 9824

Live Debugging

Upload

Bookmark List

# First, let's introduce Sherlog!



In-depth log analysis  
Visited: 9824

Live Debugging

Project Select

RSP

6500

Waverouter

RLS

10.184.69.47

Connect

# First, let's introduce Sherlog!



In-depth log analysis

Visited: 9824

Total visits: ~10k

Unique users: +200

Live Debugging

Project Select

RSP

6500

Waverouter

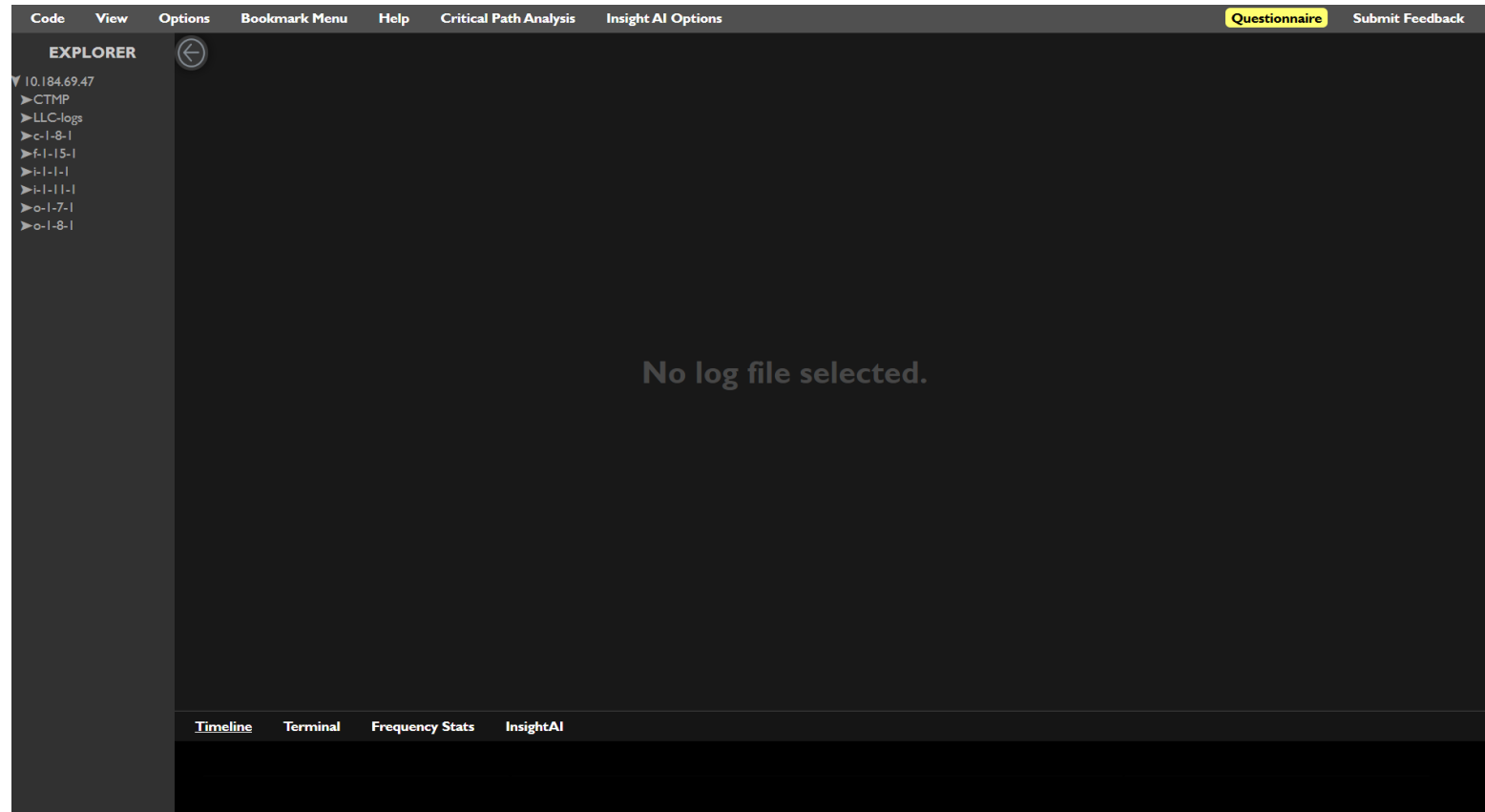
RLS

10.184.69.47

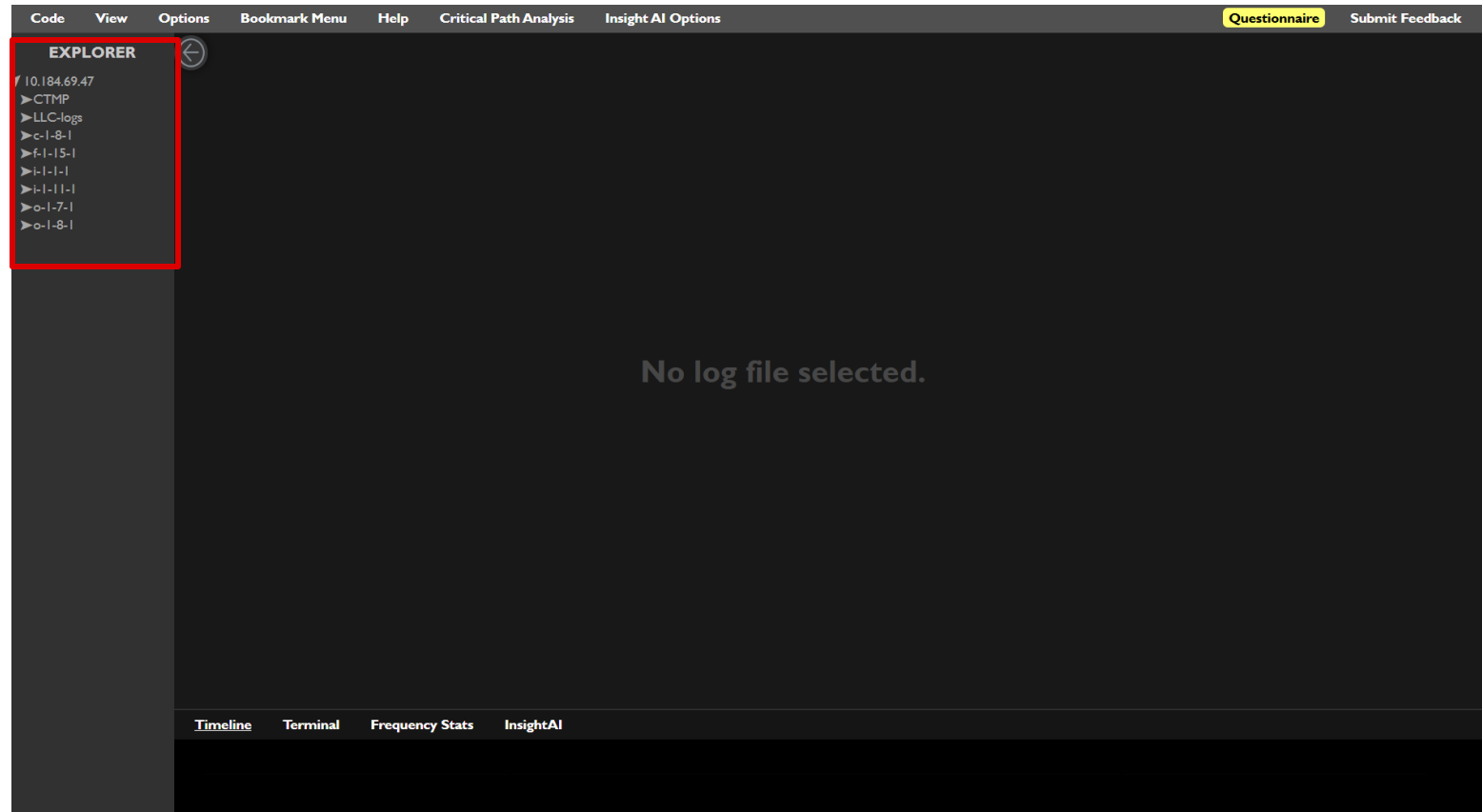
Connect



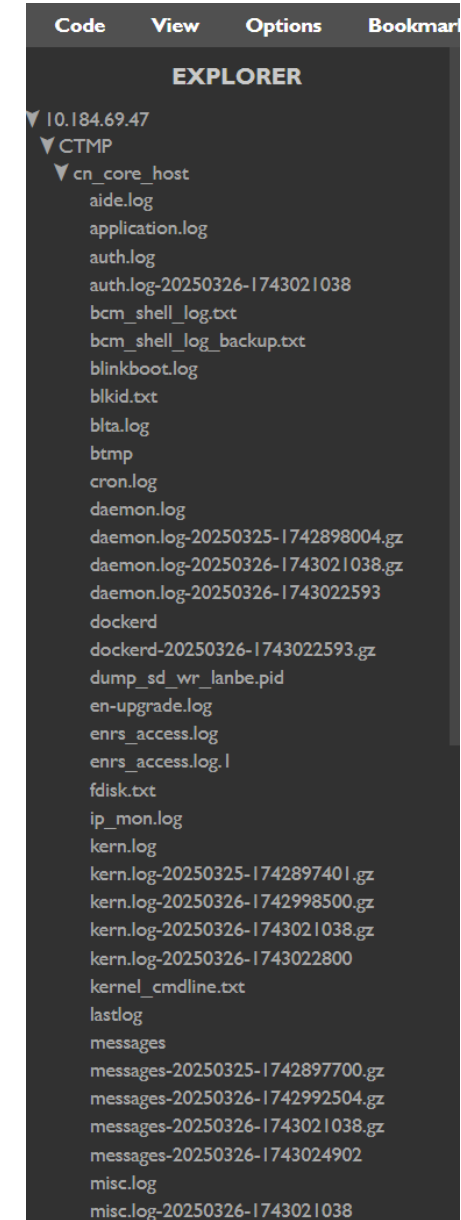
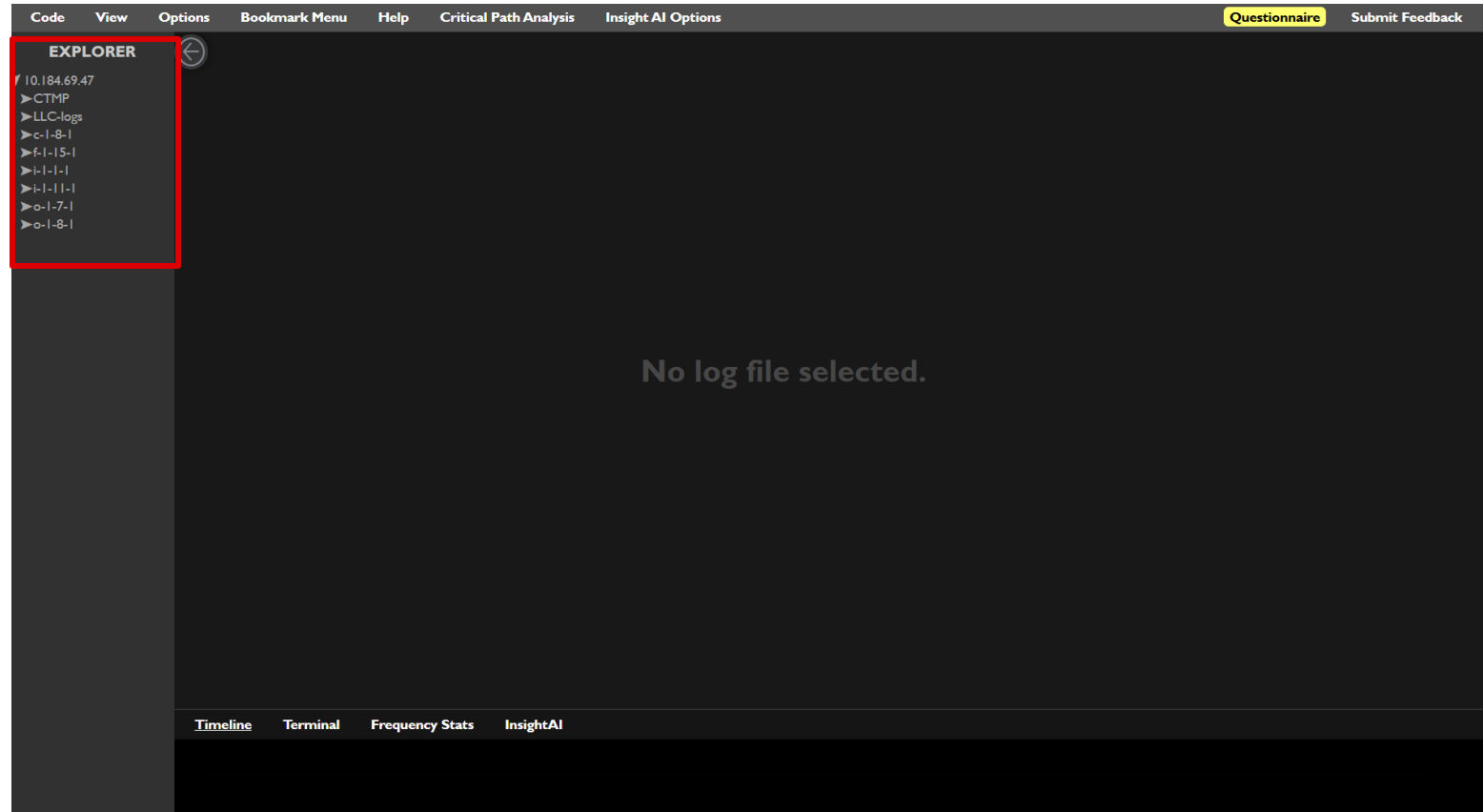
# Sherlog: A log & trace analysis tool



# Sherlog: A log & trace analysis tool

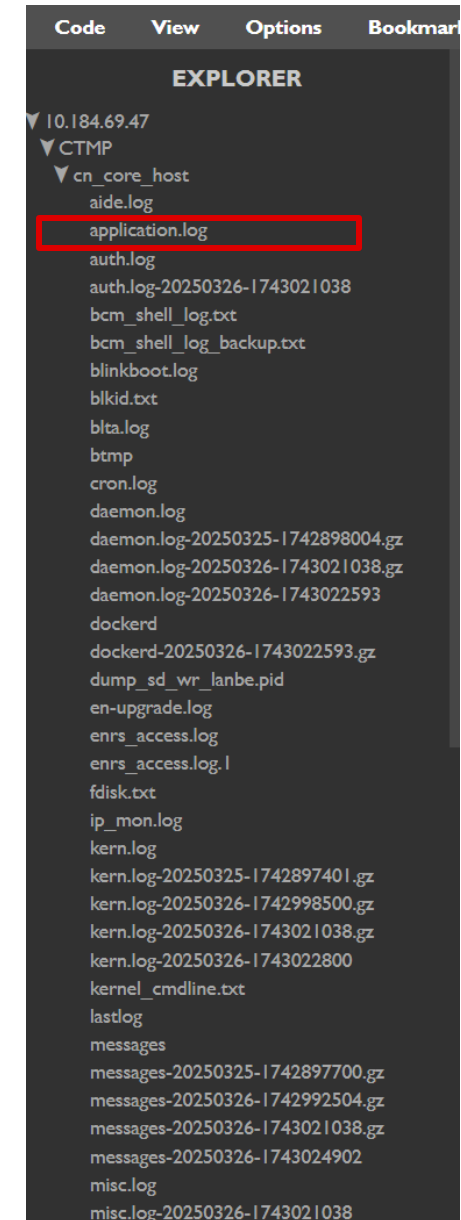
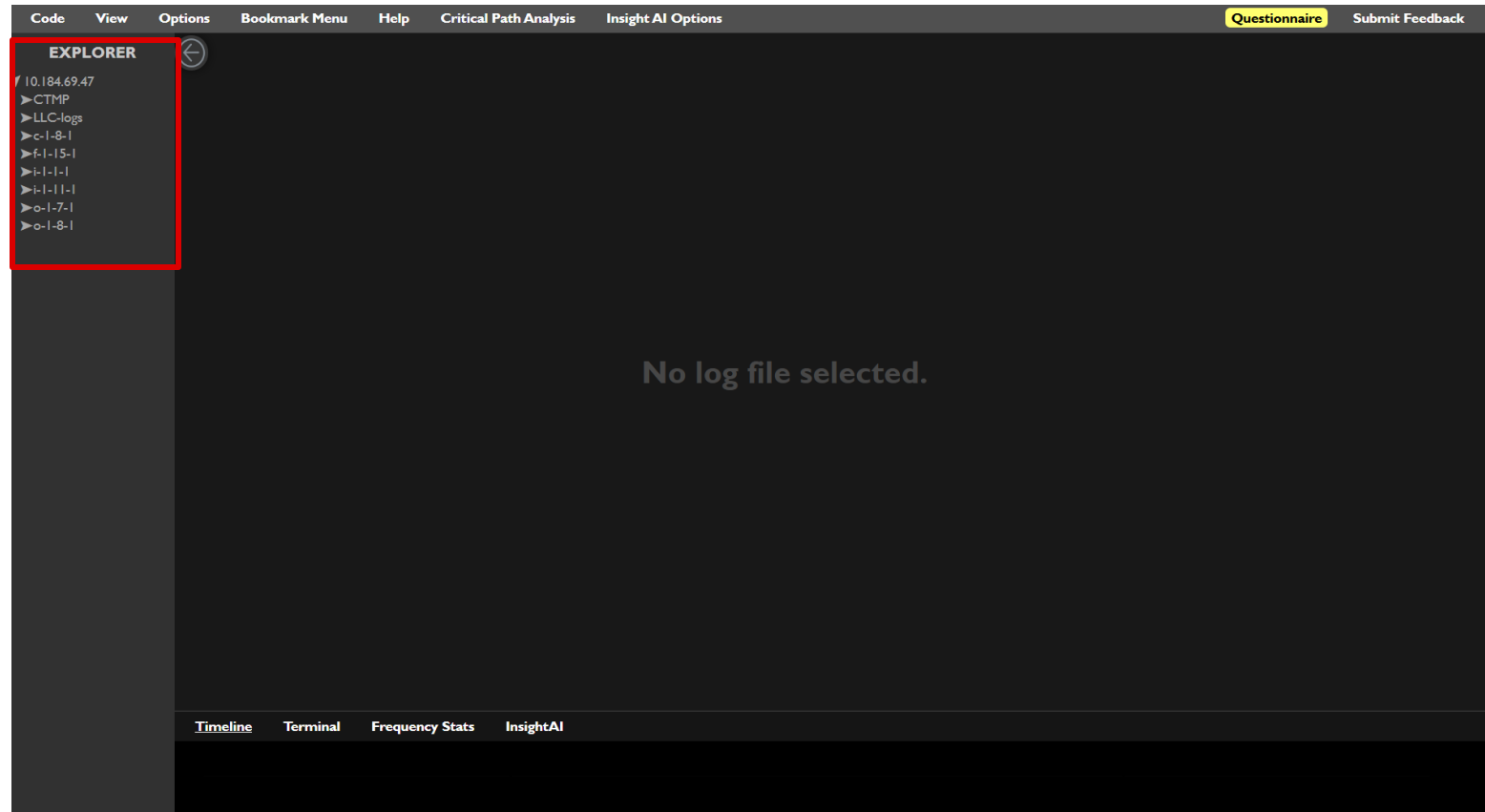


# Sherlog: A log & trace analysis tool





# Sherlog: A log & trace analysis tool



# Sherlog: A log & trace analysis tool

The screenshot displays the Sherlog web application interface. The browser address bar shows `https://wrs-sherlog.ciena.com:8000`. The top navigation bar includes tabs for Code, View, Options, Bookmark Menu, Help, Critical Path Analysis, and Insight AI Options. On the right, there are buttons for Questionnaire and Submit Feedback.

The main interface is divided into three sections:

- EXPLORER:** A sidebar on the left showing a file tree. The selected file is `application.log` under the path `10.184.69.47/CTMP/cn...`. Other files listed include `cn_core_host`, `aide.log`, `auth.log`, `auth.log-20250326-17430`, `bcm_shell_log.txt`, `bcm_shell_log_backup.txt`, `blinkboot.log`, `blkid.txt`, `blta.log`, `btm`, `cron.log`, `daemon.log`, `daemon.log-20250325-174`, `daemon.log-20250326-174`, `daemon.log-20250326-174`, `dockerd`, `dockerd-20250326-17430`, `dump_sd_wr_lanbe.pid`, `en-upgrade.log`, `enrs_access.log`, `enrs_access.log.l`, `fdisk.txt`, `ip_mon.log`, `kern.log`, `kern.log-20250325-17428`, `kern.log-20250326-17429`, `kern.log-20250326-17430`, `kern.log-20250326-17430`, `kernel_cmdline.txt`, `lastlog`, `messages`, `messages-20250325-1742`, `messages-20250326-1742`, `messages-20250326-1743`, `messages-20250326-1743`, `misc.log`, and `misc.log-20250326-17430`.
- Log Viewer:** The central pane displays the content of `application.log`, which has 4584 lines. It features a search bar and several filter buttons: Full Search, rawLog, log-path, prefix, timestamp, system1, system2, system3, file, line, function, cnprefix, and msg. Below these are search input fields and a Reset button. The log entries are displayed in a table with columns for line number, timestamp, host, and log message. The first few lines show timestamps from 2025-03-25 05:38:42.122074 to 2025-03-25 05:38:49.789224, all from host `baremetal`. The messages include various system logs and service context entries.
- Timeline:** A bar chart at the bottom left shows the frequency of log entries over time. The x-axis represents time, and the y-axis represents the number of entries, ranging from 0 to 250. The chart shows several peaks, with the highest peak reaching approximately 250 entries.

# Sherlog: A log & trace analysis tool

The screenshot displays the Sherlog web application interface. The browser address bar shows `https://wrs-sherlog.ciena.com:8000`. The top navigation bar includes links for Code, View, Options, Bookmark Menu, Help, Critical Path Analysis, and Insight AI Options. On the right, there are buttons for Questionnaire and Submit Feedback.

The left sidebar, titled "EXPLORER", shows a tree view of files. The file `application.log` is selected, and its path `10.184.69.47/CTMP/cn...` is visible. Below the explorer, a list of files is shown, including `cn_core_host`, `aide.log`, `auth.log`, `auth.log-20250326-17430`, `bcm_shell_log.txt`, `bcm_shell_log_backup.txt`, `blinkboot.log`, `blkid.txt`, `blta.log`, `btm`, `cron.log`, `daemon.log`, `daemon.log-20250325-174`, `daemon.log-20250326-174`, `daemon.log-20250326-174`, `dockerd`, `dockerd-20250326-17430`, `dump_sd_wr_lanbe.pid`, `en-upgrade.log`, `enrs_access.log`, `enrs_access.log.l`, `fdisk.txt`, `ip_mon.log`, `kern.log`, `kern.log-20250325-17428`, `kern.log-20250326-17429`, `kern.log-20250326-17430`, `kern.log-20250326-17430`, `kernel_cmdline.txt`, `lastlog`, `messages`, `messages-20250325-1742`, `messages-20250326-1742`, `messages-20250326-1743`, `messages-20250326-1743`, `misc.log`, and `misc.log-20250326-17430`.

The main content area displays the selected file `application.log`, which contains 4584 lines. The interface includes search filters for `Full Search`, `rawLog`, `log path`, `prefix`, `timestamp`, `system1`, `system2`, `system3`, `file`, `line`, `function`, `cnprefix`, and `msg`. There are also buttons for `all`, `Seek`, and `Reset`.

The log entries are displayed in a table with columns for `Q`, `<187>1`, `2025-03-25`, `05:38:42.122074`, `baremetal`, `cn-node-hal`, `halMain`, `serviceContext.c`, `4995`, `svcStorageOwnEndpointSelect`, and `cncore`. The entries show various system messages and errors, including `halMain serviceContext.c 4995 svcStorageOwnEndpointSelect cncore`, `halMain serviceContext.c 5086 svcStorageOwnLocalEndpointSelect cncore`, `halDbgMsg serviceContext.c 4995 svcStorageOwnEndpointSelect cncore`, `halDbgMsg serviceContext.c 5086 svcStorageOwnLocalEndpointSelect cncore`, `halMainSvc serviceContext.c 4995 svcStorageOwnEndpointSelect cncore`, `halMainSvc serviceContext.c 5086 svcStorageOwnLocalEndpointSelect cncore`, `halOptIfSvc serviceContext.c 4995 svcStorageOwnEndpointSelect cncore`, `halOptIfSvc serviceContext.c 5086 svcStorageOwnLocalEndpointSelect cncore`, `halBcmTempSend serviceContext.c 4995 svcStorageOwnEndpointSelect cncore`, `halBcmTempSend serviceContext.c 5086 svcStorageOwnLocalEndpointSelect cncore`, `halMainSvc ctmdMain.c 239 halMainEnter cn-node-hal cn-node-hal ha`, `halMain ctmdMain.c 444 main cn-node-hal cn-node-hal checkpoint1`, `halMain ctmdMain.c 449 main cn-node-hal cn-node-hal checkpoint2`, `halMain ctmdMain.c 453 main cn-node-hal cn-node-hal checkpoint3`, `halMainSvc halDnxFaultsGet.c 410 halDnxPortStatsOverRestart cn-nod`, `halMainSvc halDnxHardwareInit.c 3251 halInit cn-node-hal cn-node-h`, `halMainSvc halDnxBoardWrNe2.c 65 drvWrNe2SalConfigDram valimar-hal`, `halMainSvc halDnxBoardWrNe2.c 67 drvWrNe2SalConfigDram valimar-hal`, `halMainSvc halBoardData.c 433 halBoardSystemMacGet cn-node-hal cn`, `optIfSvcApi.c 270 optIfProcessStart cn-node-hal cn-nod`, `halOptIfSvc halWrNe2XcvrOpsDispatch.c 519 halWrNe2XcvrOpsDispatchIn`, and `halOptIfSvc halWrNe2XcvrOpsDispatch.c 75 halWrNe2XcvrAoPortLedGet`.

At the bottom, there is a "Timeline" section with a bar chart showing the frequency of log entries over time. The chart has a y-axis labeled "Frequency Stats" and an x-axis labeled "Insight AI". The bars represent the frequency of log entries for different time intervals.

# Sherlog: A log & trace analysis tool

The screenshot displays the Sherlog web interface for log analysis. The browser address bar shows `https://wrs-sherlog.ciena.com:8000`. The interface includes a top navigation bar with tabs like 'Code', 'View', 'Options', 'Bookmark Menu', 'Help', 'Critical Path Analysis', and 'Insight AI Options'. A 'Questionnaire' button and 'Submit Feedback' link are also present.

The main content area is divided into three sections:

- EXPLORER**: A sidebar on the left showing a file tree with folders like 'CTMP' and 'cn\_core\_host', and various log files such as 'aide.log', 'auth.log', 'bcm\_shell\_log.txt', 'blinkboot.log', 'blkid.txt', 'blta.log', 'btmpt', 'cron.log', 'daemon.log', 'dockerd', 'dump\_sd\_wr\_lanbe.pid', 'enr\_upgrade.log', 'enrs\_access.log', 'fdisk.txt', 'ip\_mon.log', 'kern.log', and 'kernel\_cmdline.txt'.
- application.log 4584 lines**: The main log viewer. It features a search bar and several filter buttons: 'Full Search', 'rawLog', 'log-path', 'prefix', 'timestamp', 'system1', 'system2', 'system3', 'file', 'line', 'function', 'cnprefix', and 'msg'. Below these are 'all' and 'Seek' buttons, and a 'Reset' button. The log entries are displayed in a table with columns for line number, timestamp, host, and log message. The 'Insight AI' tab is highlighted in the bottom navigation bar.
- Timeline**: A bar chart at the bottom showing the frequency of log events over time. The y-axis represents frequency, ranging from 0 to 250. The x-axis represents time, with labels for various log files and timestamps.

# InsightAI: Demo!

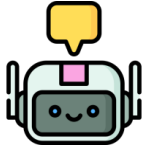


Live Debugging

Upload

Bookmark List

# InsightAI: Why do we need it?



Log & trace analysis using AI



Massive amount of log data for each product



Reduce manual effort of the users

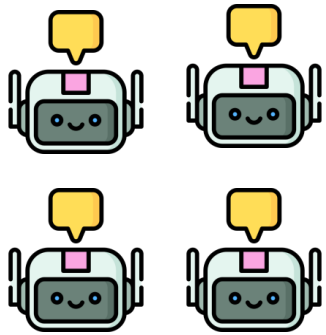


Faster root cause analysis



# InsightAI: How does it work?

Agentic AI



RAG

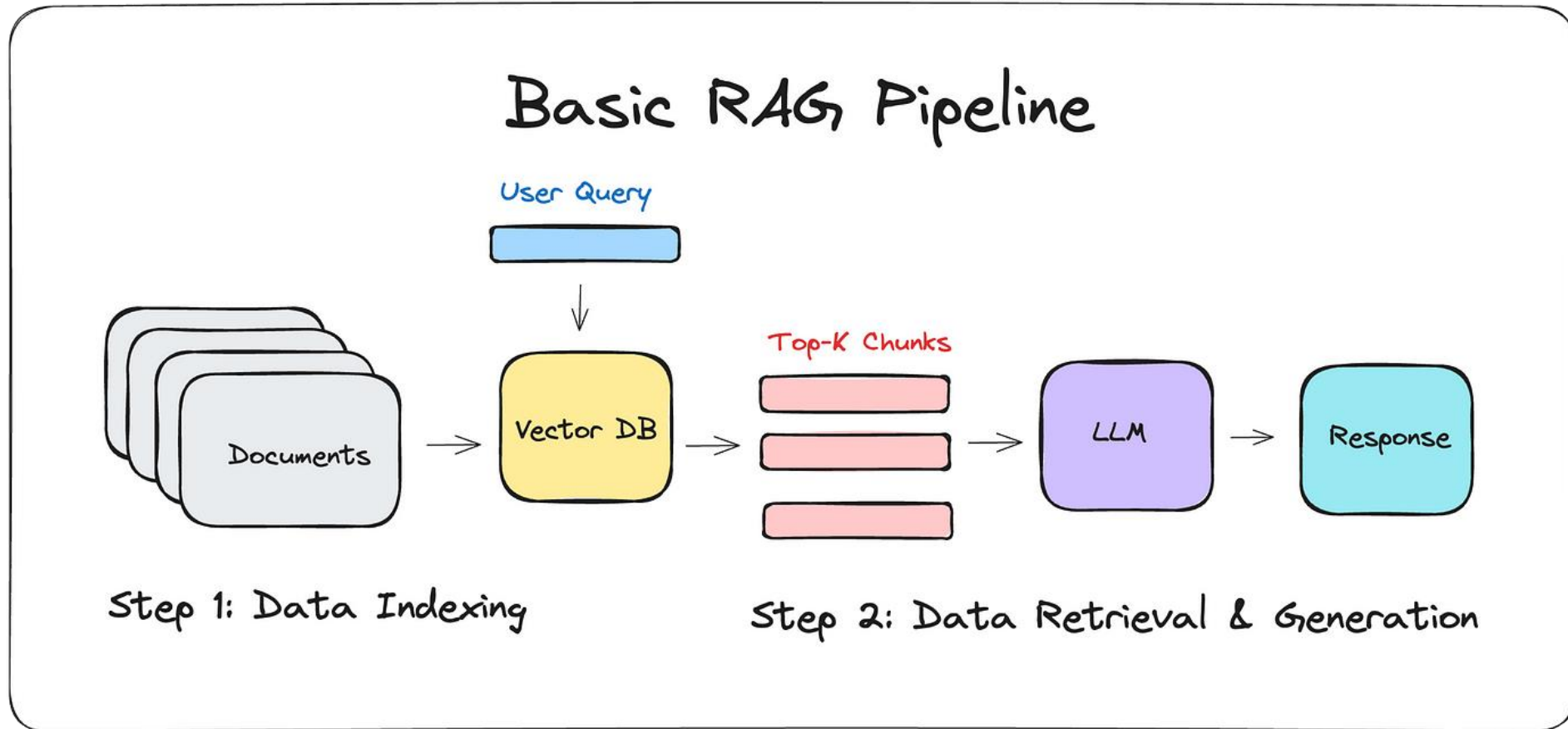
(Retrieval Augmented Generation)



# What is RAG?

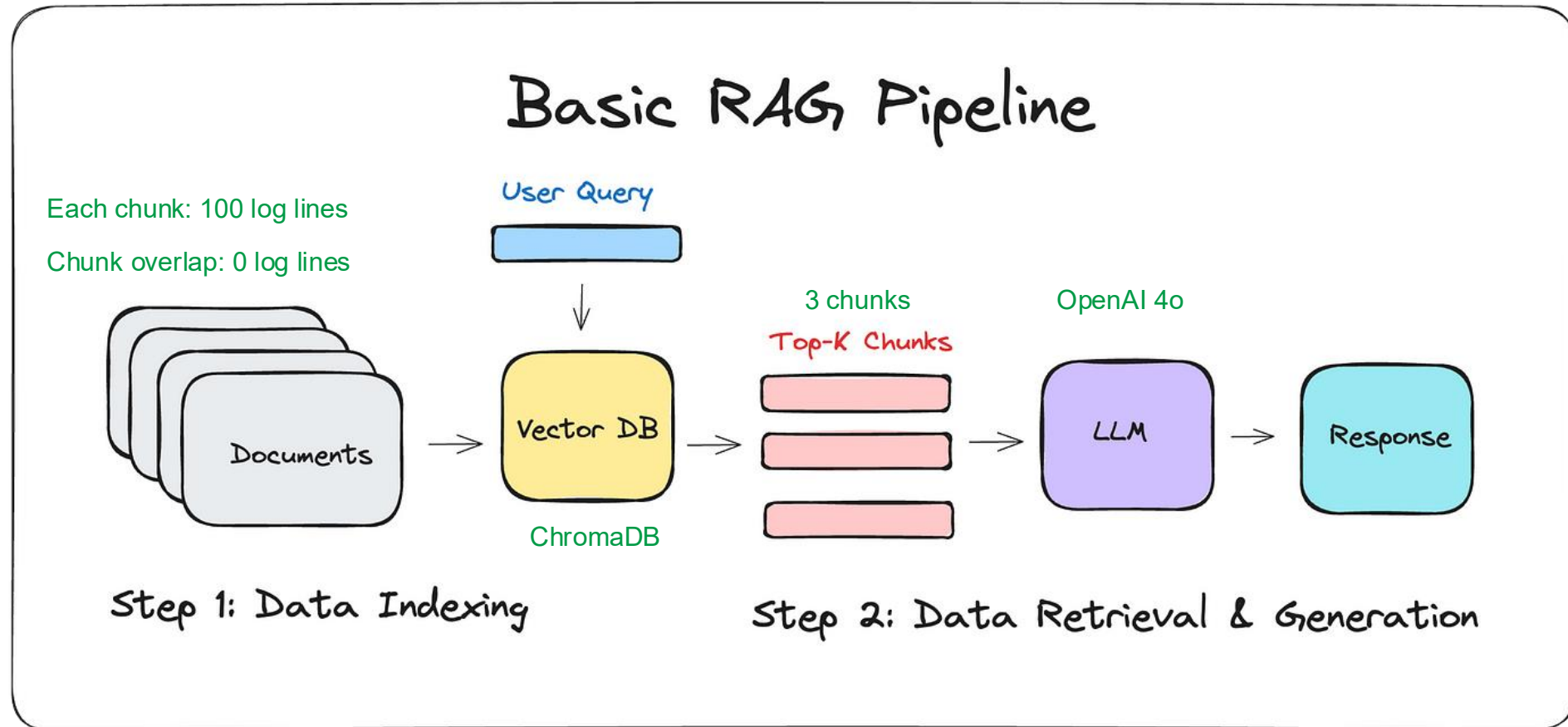


# What is RAG?



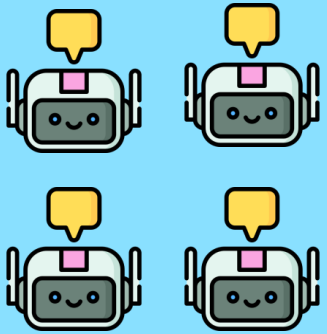
<https://medium.com/@drjulija/what-is-retrieval-augmented-generation-rag-938e4f6e03d1>

# InsightAI: Our RAG implementation



<https://medium.com/@drjulija/what-is-retrieval-augmented-generation-rag-938e4f6e03d1>

# What is *Agentic* AI?



# What are agents?





# What are agents?



<https://huggingface.co/learn/agents-course/en/unit1/what-are-agents>



# What are agents?



<https://huggingface.co/learn/agents-course/en/unit1/what-are-agents>

# What are agents?



<https://huggingface.co/learn/agents-course/en/unit1/what-are-agents>

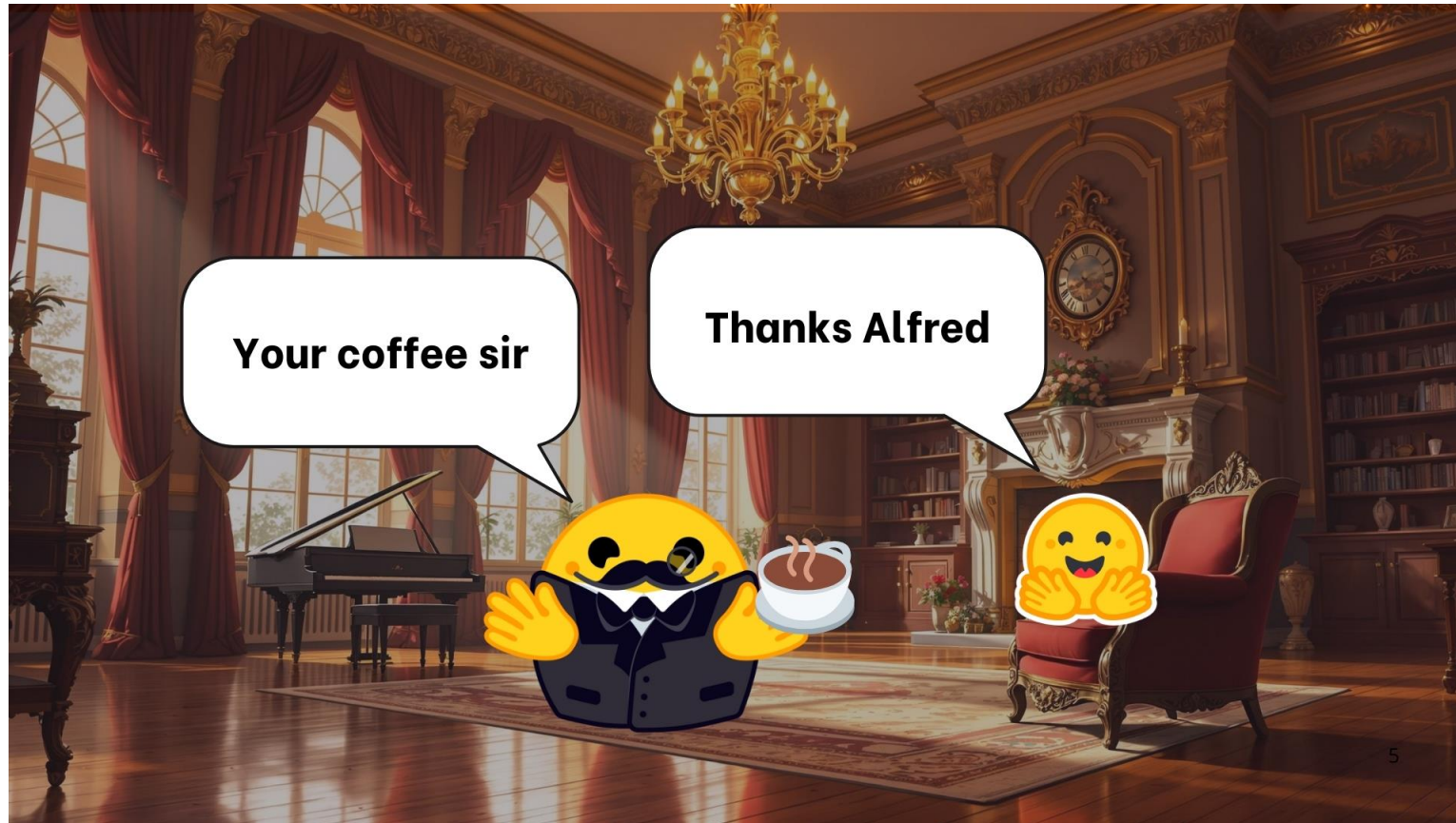
# What are agents?



<https://huggingface.co/learn/agents-course/en/unit1/what-are-agents>

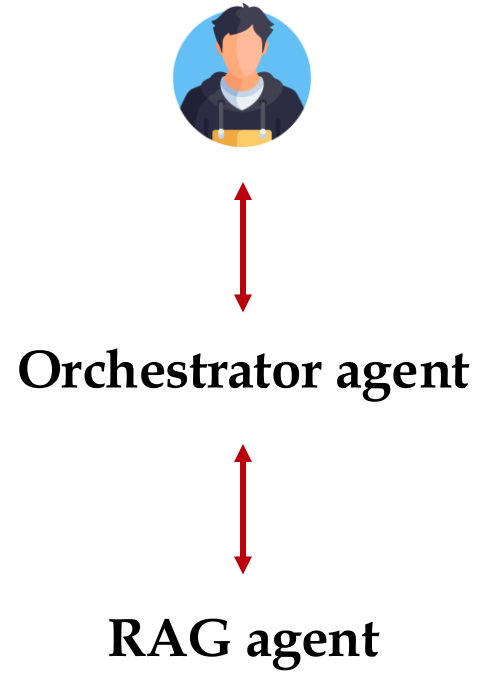


# What are agents?

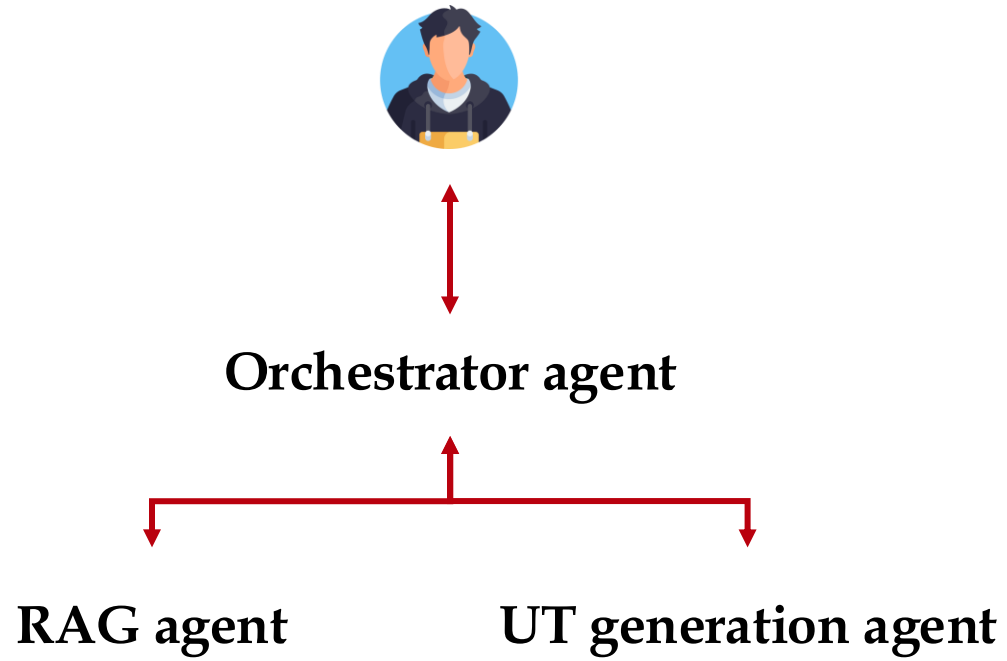


<https://huggingface.co/learn/agents-course/en/unit1/what-are-agents>

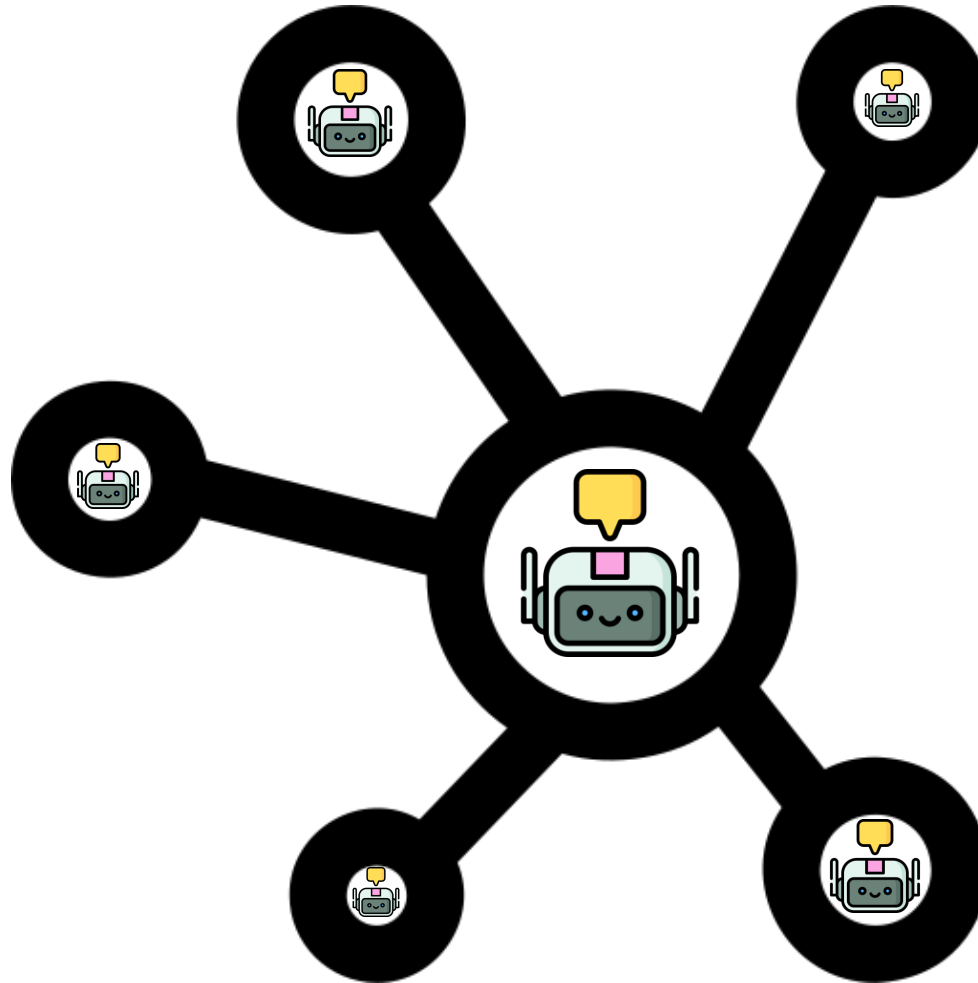
# Current agentic workflow in InsightAI



# Future agentic workflow in InsightAI

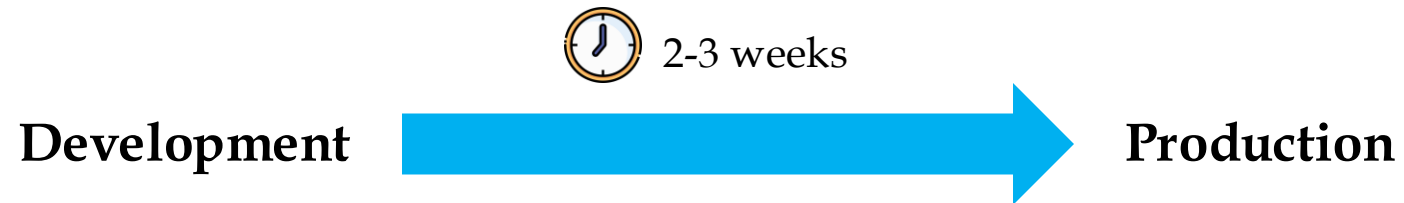






# Future agentic workflow in InsightAI





# InsightAI: Current stage



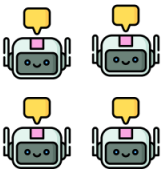
	One file	Multiple files
Live		
Statedump		

# Sherlog: Finding a needle in a haystack



## InsightAI: How does it work?

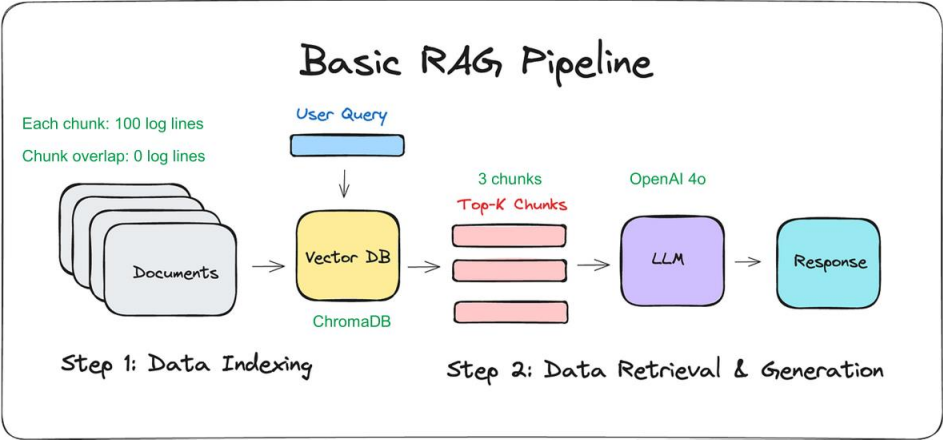
Agentic AI



RAG  
(Retrieval Augmented Generation)



## InsightAI: Our RAG implementation



<https://medium.com/@drjulija/what-is-retrieval-augmented-generation-rag-938e4f6e03d1>

## Future agentic workflow in InsightAI

