

Building Agentic Systems



Gou Rao, Co-Founder

Evolution of GenAI in the Enterprise

Enterprise Chat Bots

Software Applications like Salesforce, Mongo etc will add chat bots

Enterprise Search Applications like Glean can enable chats with your enterprise data.

GenAI platforms

LLM platforms from OpenAI, Accenture, Hugging Face allow enterprises to build new AI powered applications to provide enhanced products to their customers



Agenting Systems

Startups are enhancing the enterprise workforce with specific, purpose build reasoning systems for specific job functions

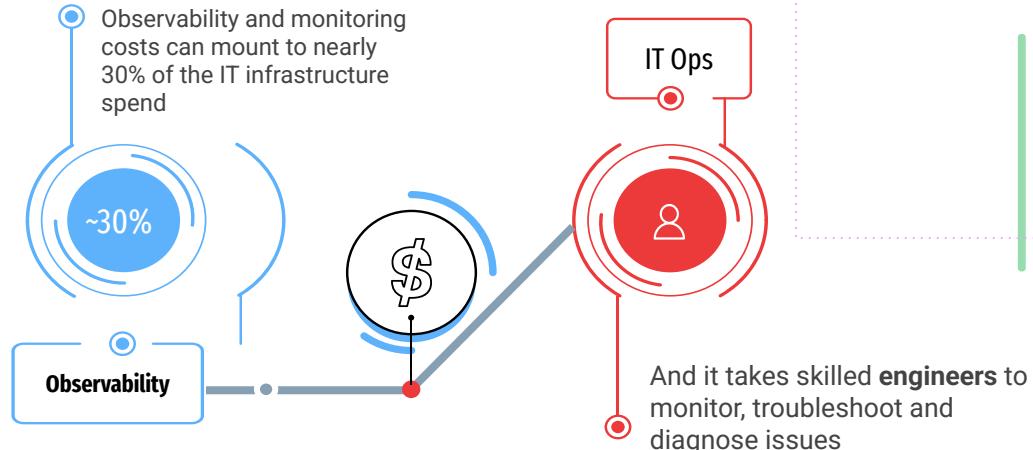


The Business Impact of slow IT Operations



neubird.ai

Running **production** IT operations is a hard business



Modern IT stack too **complex** to keep up with
Tickets keep piling up
Impedes Business Productivity

The Current IT Stack is too **Complex**

Too Many Data Sources

Log Files, Traces, Metrics, Alerts, Infra Config



```
sql
SELECT
    books.title,
    authors.author_name,
    COUNT(book_loans.book_id) AS total_loans
FROM
    books
INNER JOIN
    book_authors ON books.book_id = book_authors.book_id
INNER JOIN
    authors ON book_authors.author_id = authors.author_id
LEFT JOIN
    book_loans ON books.book_id = book_loans.book_id
WHERE
    books.publication_year >= 2000
GROUP BY
    books.title, authors.author_name
HAVING
    COUNT(book_loans.book_id) > 10
ORDER BY
    total_loans DESC, authors.author_name ASC;
```

Extract Transform
and Load into
Analytics DB (Elastic)

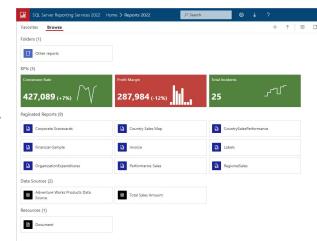
Yet Another DB

Create complex
queries

Database Engineering

Visualization

Integrate the results
into intuitive
dashboards

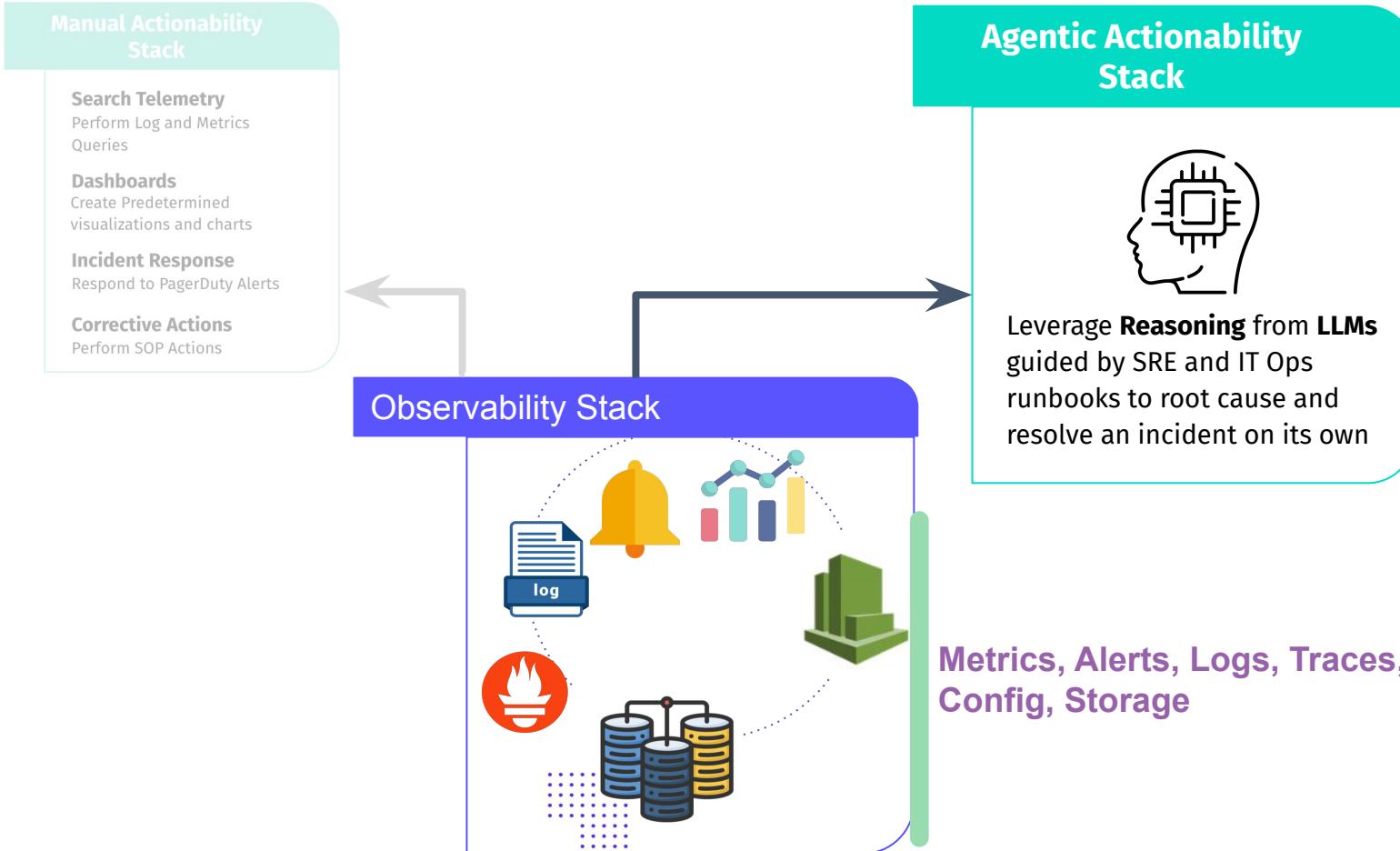


Too many layers of **complex**
telemetry

Bringing **Agentic** Workflows to IT Operations



neubird.ai

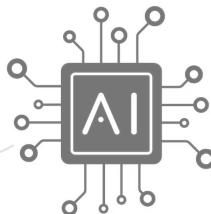
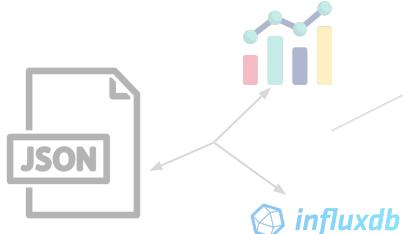


Use **REASONING** from the LLMs to build Agents



Use the **reasoning** capability of LLMs to solve IT operations issues in **real-time**

LLMs can mimic an experienced SRE's thought process in just minutes



“AI is an **amplifier** of human intelligence & when people are smarter, better things happen: people are more **productive**, happier & the economy strives.”

- Yann LeCun, Head of Meta AI, Turing Award

LLMs can query your observability tools directly



Meet **Hawkeye** - The First Agentic IT Ops Engineer



neubird.ai



Part of your IT workforce

Works alongside IT staff
as an assistant to help
take on mundane tasks

Autonomously or
Collaboratively resolve IT
tickets and issues

01

Predetermined Dashboards

Dashboards and problem scenarios are **prematurely** determined ahead of time

02

Uncorrelated

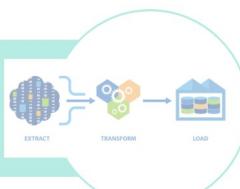
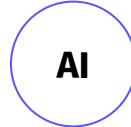
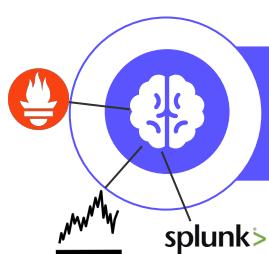
Information is from unrelated data sources - **causality** has to be inferred visually

03

Decisions Not in Real Time

Data has to be manually correlated and analyzed increasing the time to response

Traditional Observability



Generative IT Ops

01

Inplace and Interactive Analytics

Data is analyzed at the source, in **real time**

02

Generative AI assisted

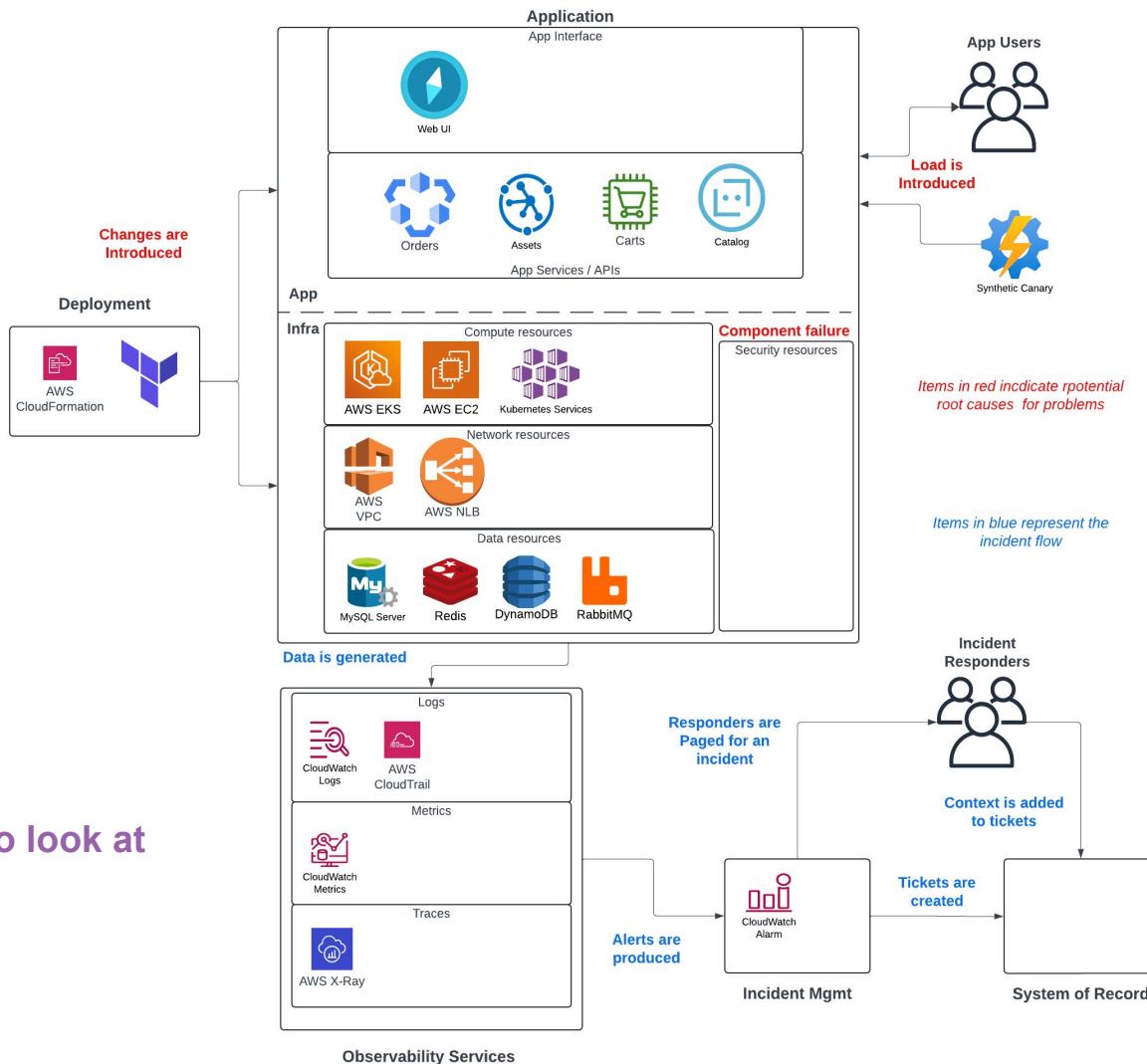
Schematic inferences and queries are **cognitively** generated - with visual feedback

03

Multi Modal Capable

AI generated queries are federated across multiple data sources and the results are further combined by **LLMs**

Demo



Example - E-Commerce SRE @ ACME.com



neubird.ai

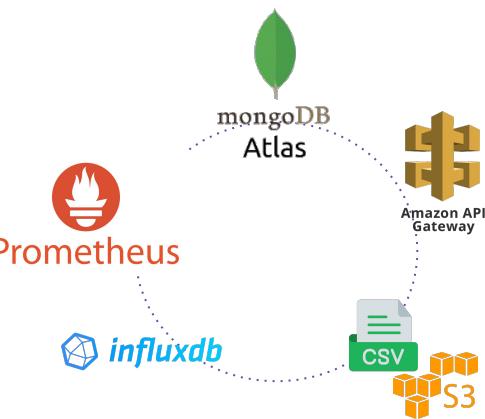
Problem: SRE notices that the ACME.com website is **performing poorly** on populating shopping cart suggestions

Causality: AI recognizes increased Daily Active Users (DAU)

Collaborative: AI suggests Mongo Atlas is a problem

Conclusive: AI provides a possible solution

• **Data Sources Consulted:**



Customer Example:

SRE notices slow website responses

 Kroger

New Chat 

website slow 

Home 

Projects 

Chat 

Datasource 

why is my website slow 

← | why is my website slow 

Summary:
This week's Daily Active Users (DAU) have shown a notable increase of 18% compared to last week. This is overloading the mongo atlas cluster deployment in the US-WEST-1 region.

Detailed Breakdown:
This Week: 26,000 DAU  (↑18% from the previous week) 
Last Week: 22,000 DAU 

Key Insights:
Growth: An 18% increase in DAU compared to last week reflects a growing user base and heightened engagement. This in turn is causing an increased number of queries per second on the Mongo deployment.
Handling more queries in a given time window causes Mongo to reply with increased latencies. Since your shopping cart feature depends on the results from Mongo Atlas, this is the reason for slow response times on that feature of this project.



30K
20K
10K
0

May 1 May 2 May 3 May 4

Export  Regenerate  Sources 

How can I resolve this issue?  

Related queries 

Customer Example:

Wants to know how they can resolve the issue

 Kroger

New Chat 

website slow 

Home

Projects

Chat

Datasource

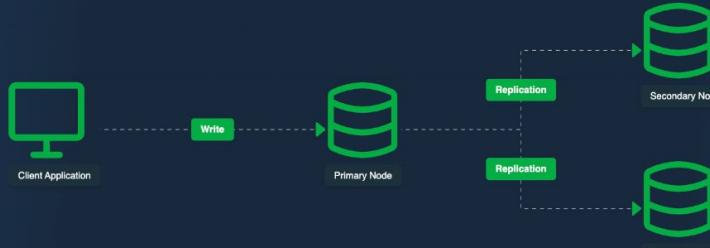
why is my website slow

Export Regenerate Sources

How can I resolve this issue? 

Summary:
After analyzing the atlas logs, it seems that the most active queries to Atlas happens between 3PM to 5PM PST. You need to increase the number of replication nodes to be able to service an increased number of queries per minute during these hours. You can utilize auto-scaling rules to add and remove replicas.

Detailed Breakdown:
This Week: 26,000 DAU ($\uparrow 18\%$ from the previous week)
Last Week: 22,000 DAU



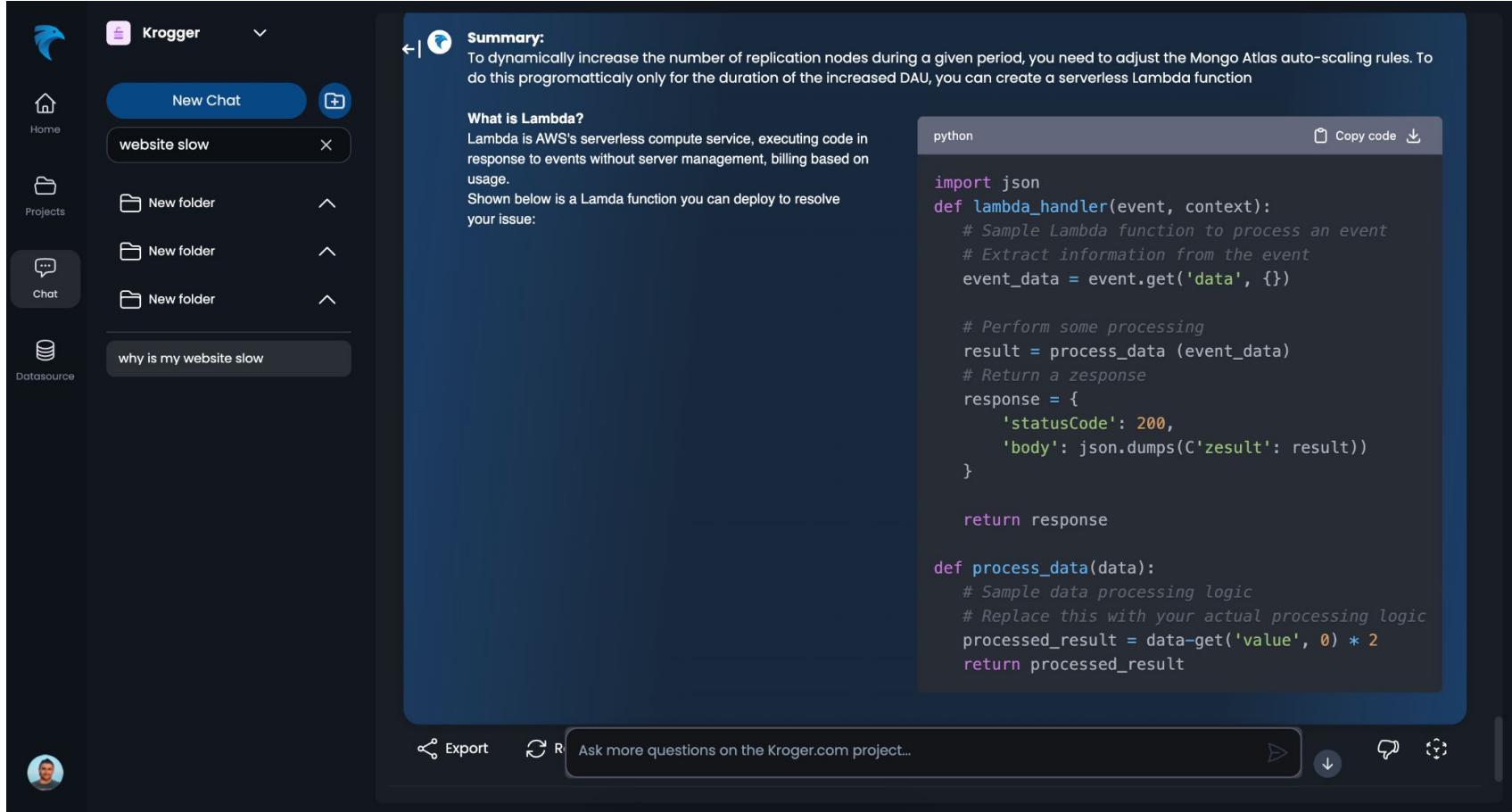
```
graph LR; CA[Client Application] -- Write --> PN[Primary Node]; PN --> SN1[Secondary Node 1]; PN --> SN2[Secondary Node 2];
```

Related queries

show me how to setup autoscaling for that period 

Customer Example:

And gets a real time solution



Kroger

New Chat

website slow

Summary:

To dynamically increase the number of replication nodes during a given period, you need to adjust the Mongo Atlas auto-scaling rules. To do this programmatically only for the duration of the increased DAU, you can create a serverless Lambda function

What is Lambda?

Lambda is AWS's serverless compute service, executing code in response to events without server management, billing based on usage.

Shown below is a Lambda function you can deploy to resolve your issue:

```
python
import json
def lambda_handler(event, context):
    # Sample Lambda function to process an event
    # Extract information from the event
    event_data = event.get('data', {})

    # Perform some processing
    result = process_data(event_data)
    # Return a response
    response = {
        'statusCode': 200,
        'body': json.dumps(result)
    }

    return response

def process_data(data):
    # Sample data processing logic
    # Replace this with your actual processing logic
    processed_result = data.get('value', 0) * 2
    return processed_result
```

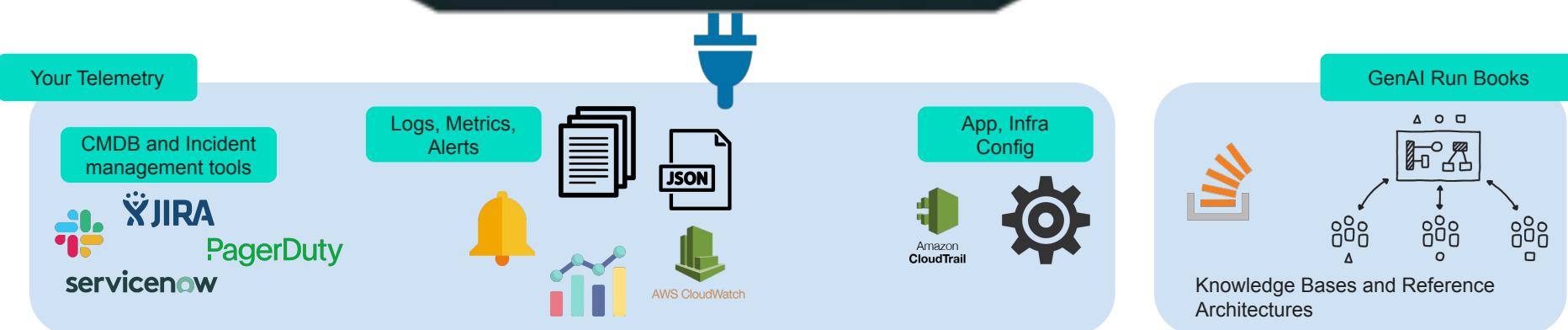
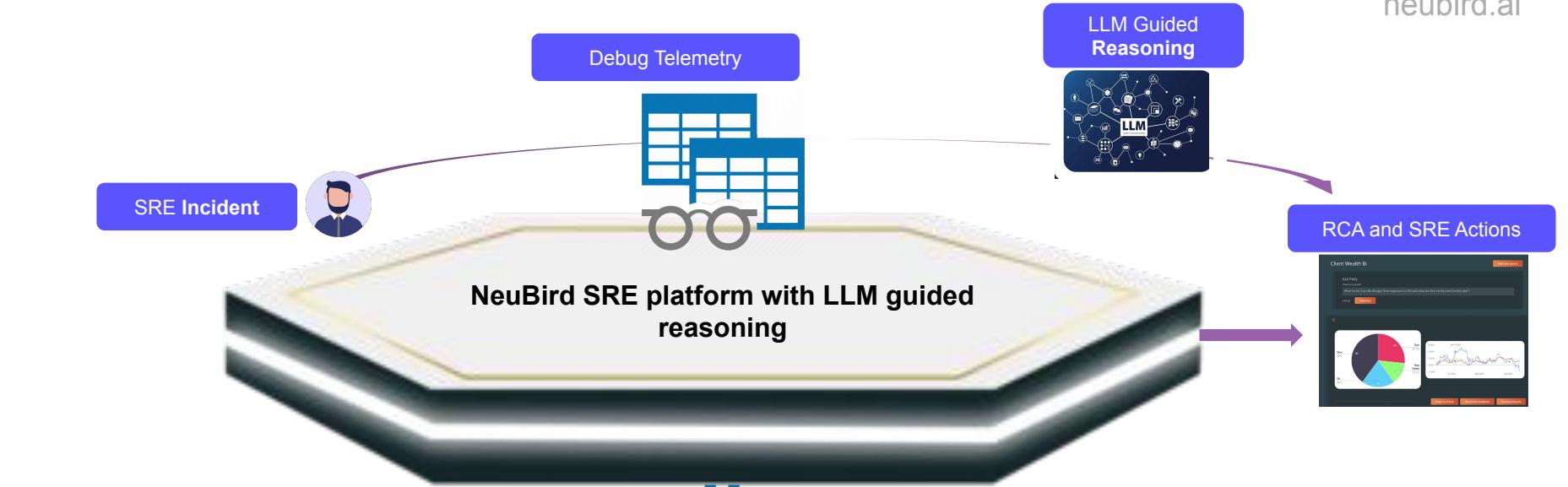
Export

Ask more questions on the Kroger.com project...

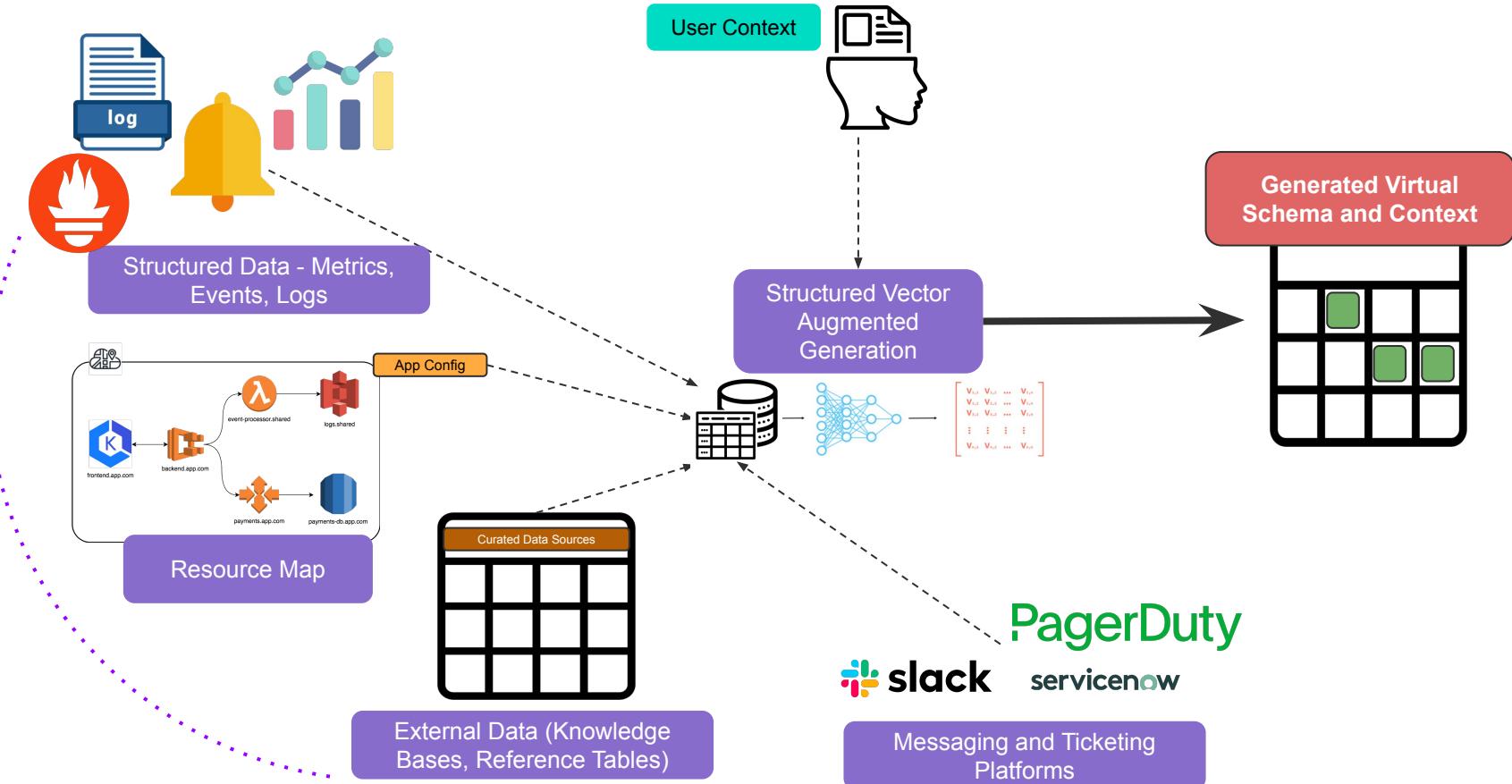
Datasource

why is my website slow

SRE as a Service using all your **IT** telemetry and observability tools



The Science Behind it: **Cognitive** Views for IT Telemetry



RAEL: A purpose built IT coding language for Agentic Systems



```
from: aws_cloudwatch
filter: "service=EKS AND logGroup=NodeStatus AND severity >= ERROR"
time_range: "last 15 minutes"
}

# Query Splunk for application error rates
query app_error_rate {
    from: splunk_observability
    filter: "app_name=my_microservice AND metric=error_rate AND status_code >= 500"
    time_range: "last 5 minutes"
}

# Query Datadog for high CPU usage
query cpu_alerts {
    from: datadog
    filter: "metric=cpu.utilization AND value > 90"
    time_range: "last 10 minutes"
}

# Decision logic
if eks_node_health.errors > 10 then
    action "Send alert to SRE team: EKS nodes experiencing frequent errors."

if app_error_rate.value > 5 then
    action "Auto-scale microservice: Increase instance count by 2."
```

Grammar and Syntax

```
PROGRAM ::= { SOURCE_DEFINITION | QUERY_DEFINITION | DECISION_BLOCK } EOF

SOURCE_DEFINITION ::= "source" IDENTIFIER "{"
    "type:" STRING
    "region:" STRING?
    "role_arn:" STRING?
    "endpoint:" STRING?
    "token:" STRING?
    "api_key:" STRING?
    "app_key:" STRING?
  }"

QUERY_DEFINITION ::= "query" IDENTIFIER "{"
    "from:" IDENTIFIER
    "filter:" STRING
    "time_range:" STRING
  }"

DECISION_BLOCK ::= "if" CONDITION "then" ACTION_BLOCK

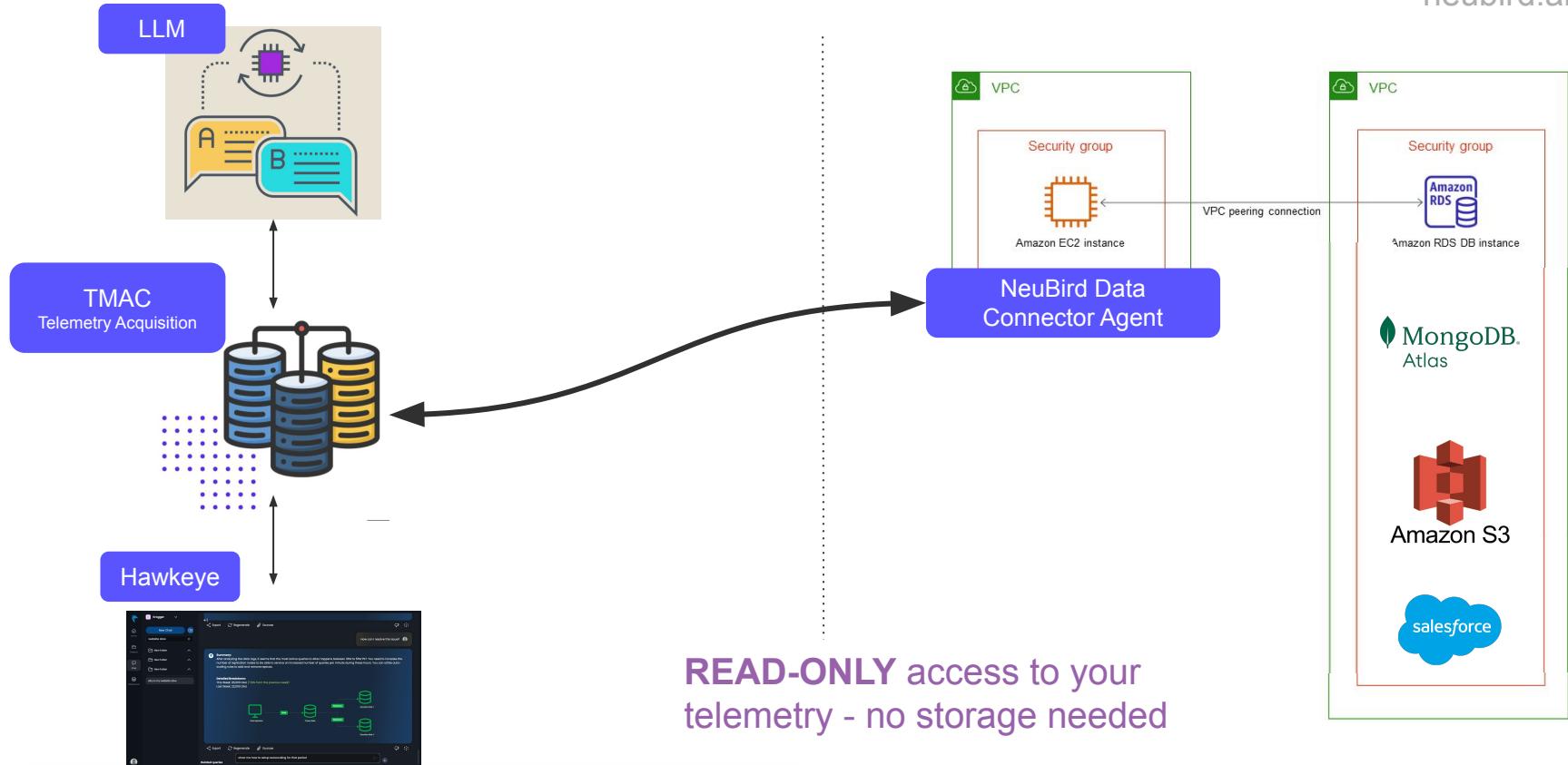
CONDITION ::= QUERY_RESULT "." FIELD COMPARATOR VALUE
            | QUERY_RESULT "." FIELD LOGICAL_OP CONDITION
            | CONDITION LOGICAL_OP CONDITION

COMPARATOR ::= ">" | "<" | ">=" | "<=" | "==" | "!="
LOGICAL_OP ::= "and" | "or"

QUERY_RESULT ::= IDENTIFIER

FIELD ::= IDENTIFIER
```

And It's **Simple** to Use: Plug and Play... on your telemetry



NeuBird.ai

Reimagine



IT Ops

Generative {IT SRE}.Ops

Manual Actionability Stack

Search Telemetry

Perform Log and Metrics Queries

Dashboards

Create Predetermined visualizations and charts

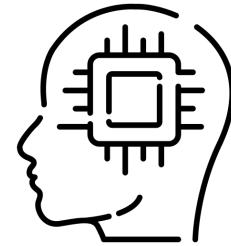
Incident Response

Respond to PagerDuty Alerts

Corrective Actions

Perform SOP Actions

Agentic Actionability Stack



Leverage **Reasoning** from LLMs guided by SRE and IT Ops runbooks to root cause and resolve an incident on its own