



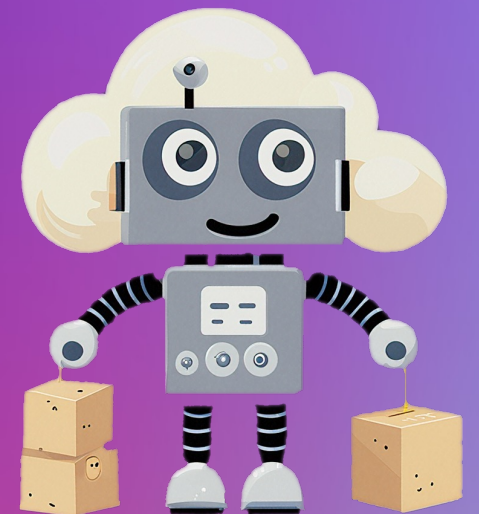
LIG202

Call My Agent

Enable LLMs Talking with your Database

Puria Izady

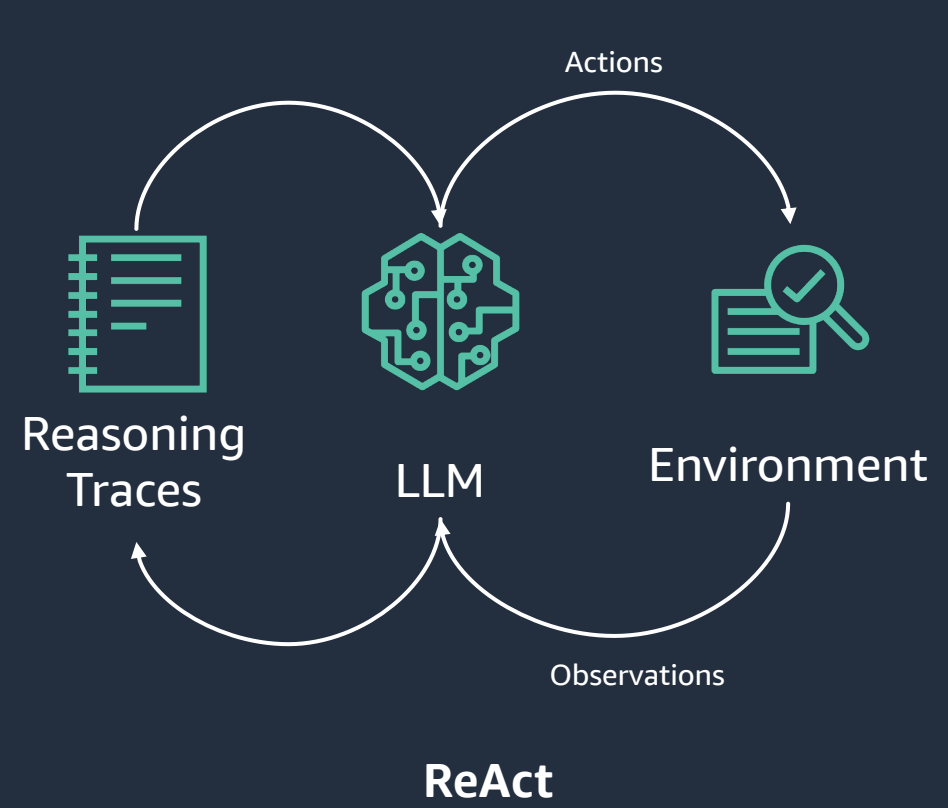
Solutions Architect
AWS



Calling An Agent 📞 – But why?



Reason and Act – ReAct



Use ReAct to create Agents leveraging Tools



Prompt

“How many active devices do we currently have in our fleet?”

“How do I configure your product if I want to use feature A & B at the same time?”

“Find our most popular item for January and start a 10% discount campaign for it.”



ReAct Agent



Tools



Knowledge



Find our most popular item for January and start a 10% discount campaign for it.

Thought 1: In the Database SalesDB I can access the needed data. Next I should discover the schema to find the right tables.

Action 1: LookUpSchema[SalesDB]

Observation 1: The tables ItemSales and Campaigns look relevant.

INTERNAL



Thought 2: I need to run the following SQL "SELECT * .." to find the most sold item in ItemSales.

Action 2: SQL["SELECT * .."; SalesDB]

Observation 2: Item with id 123456 sold the most in January.

INTERNAL



Thought 3: Next I should add a 10% discount for item id 123456 in Campaigns. I can leverage this SQL for it "INSERT INTO ..."
Action 3: SQL["INSERT INTO ...", SalesDB]
Observation 3: Task completed successfully.

INTERNAL



Action 4: Finished.

INTERNAL



Item with id 123456 was the highest selling item and I have activated a discount campaign for it.

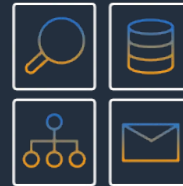


Thanks Buddy!

Calling An Agent 📞 – But why?



Natural
Language as
Interface



Build new
Applications
with Agents



Faster Time
to Insights &
Results

What are our ingredients? 🧂

Who will be our Agents? 🕵️🕵️

Foundation Models

CREATE OR IMPORT FMs

 Amazon SageMaker

SERVERLESS GENAI FMs

 Amazon Bedrock



Agent

 Amazon Bedrock Agent

 LangChain Agent




or

Ready to Use

GENERATIVE AI AGENT

 Amazon Q

BUSINESS INTELLIGENCE

 Amazon QuickSight Q

How can we group our data sources?

Unstructured



Amazon S3

Semi-Structured



Amazon DynamoDB



Amazon OpenSearch Service



Amazon DocumentDB



Amazon MemoryDB

Structured

RELATIONAL



Amazon Aurora



Amazon RDS

Data Warehouse



Amazon Redshift

GRAPH



Amazon Neptune

Vector Data



Amazon Aurora



Amazon RDS



Amazon OpenSearch Service



Amazon DocumentDB



Amazon MemoryDB

Connect the Agent to your Data

One number fits all? 📞



Unstructured

A white bucket icon, representing unstructured data. The entire box has a green border and a green header bar.

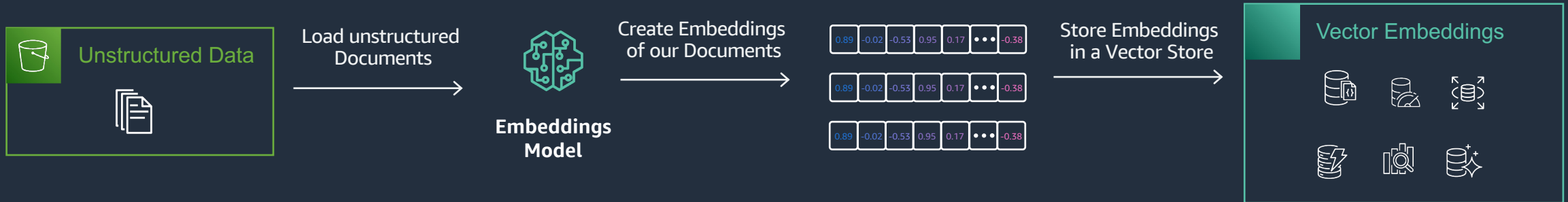
Semi-structured

Four icons representing semi-structured data: a document with arrows, a camera, a database with a plus sign, and a database with a pie chart. The entire box has a blue border and a blue header bar.

Structured

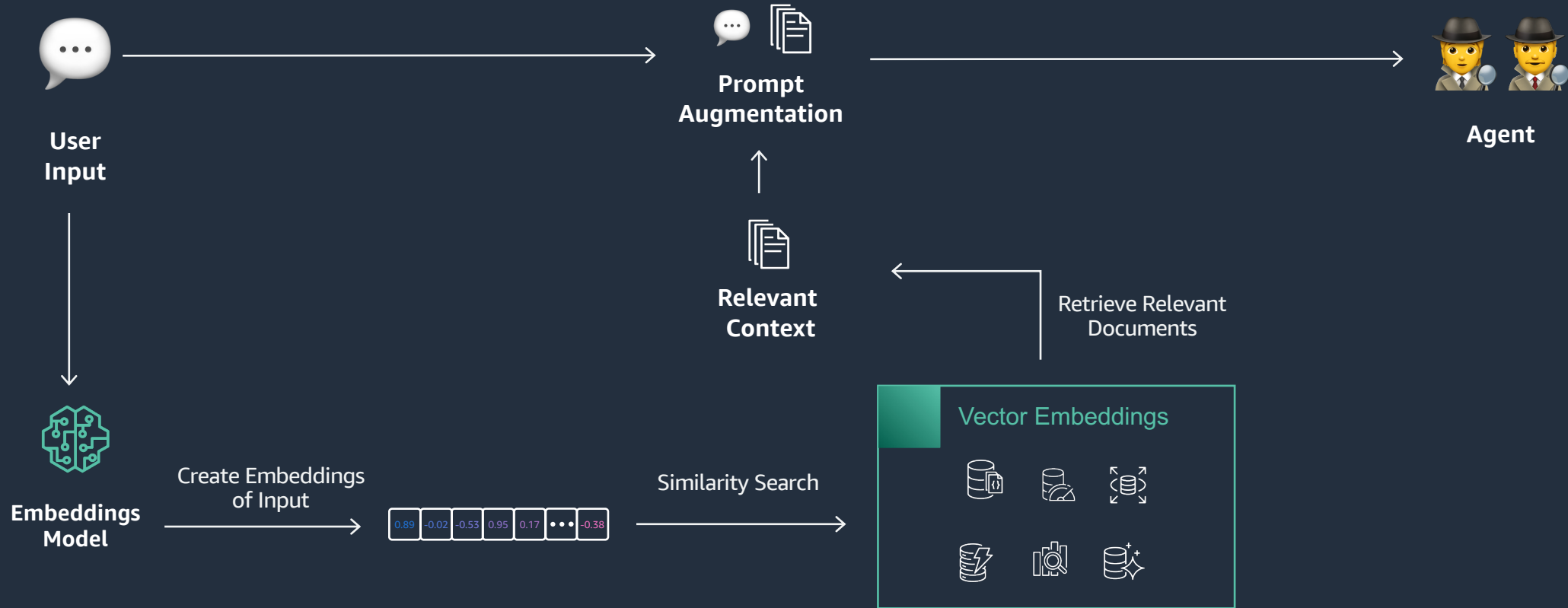
Four icons representing structured data: a database with a plus sign, a database with a network diagram, a database with arrows, and a database with a pie chart. The entire box has a red border and a red header bar.

Create searchable Embeddings

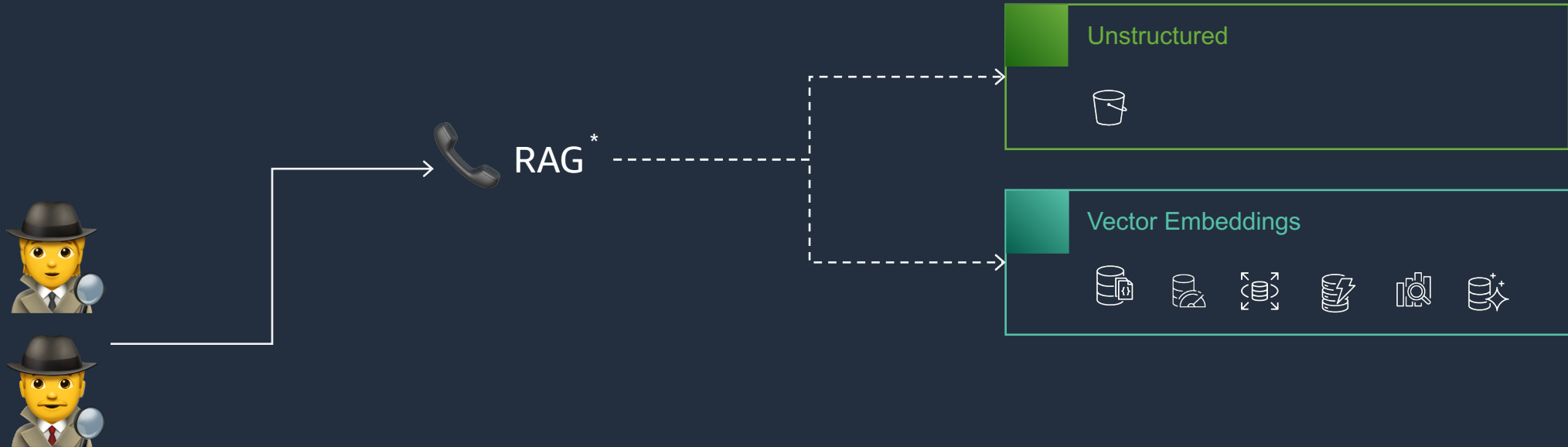


Use Embeddings for Retrieval Augmented Generation

*“How can I configure
Function A in your
Software Solution?”*



Let's dial the right numbers 📞



* Retrieval Augmented Generation is the general process of combining data with your prompt. It is not specific to Embeddings.

Using SQL to Discover and Create Queries for Relational Databases

“How many active devices do we currently have in our fleet?”



User Input



Agent

2. Create SQL Query & Validate



```
SELECT COUNT(*) AS num_active_devices  
FROM devices  
WHERE status = 'active';
```



SQL Query


1. Discover Schema





3. Run SQL Query




Structured - Relational

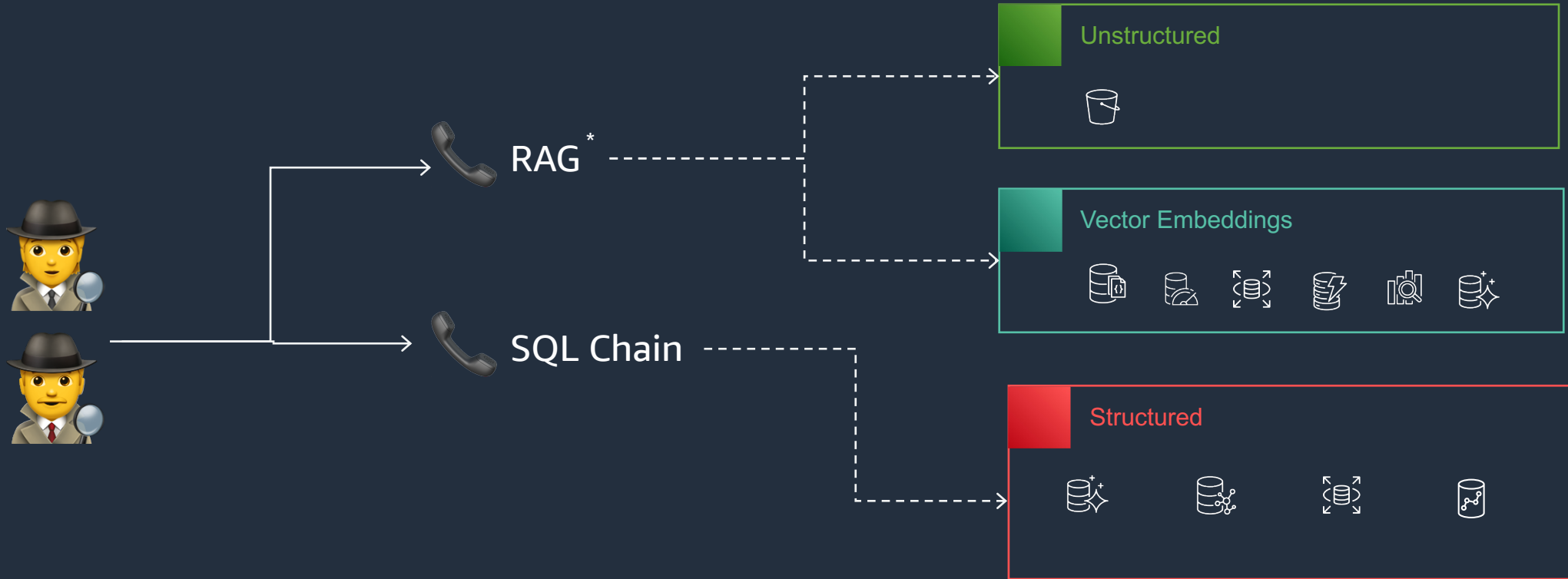

+


+


↕


↕

Let's dial the right numbers



* Retrieval Augmented Generation is the general process of combining data with your prompt. It is not specific to Embeddings.

Leverage GraphQL for Semi-Structured Data Sources

“Which Jackie Chan movies have a 5 ★ rating?”



User Input



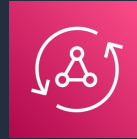
Agent

2. Create GraphQL Query



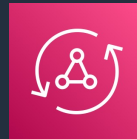
```
{
  actor(name: "Jackie Chan") {
    name
    movies(where: { rating: 5 }) {
      title
      rating
    }
  }
}
```

1. Discover API Schema



AppSync

3. Query for 5 ★ Movies of Jackie Chan



AppSync

Introspection Query



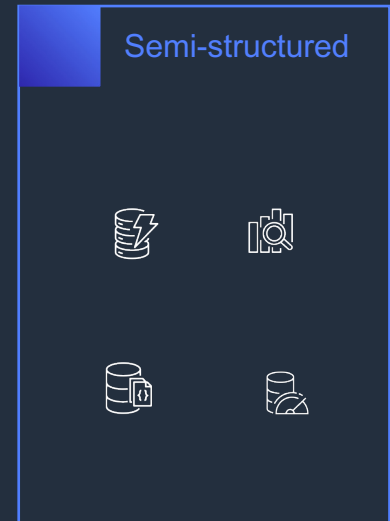
API Schema



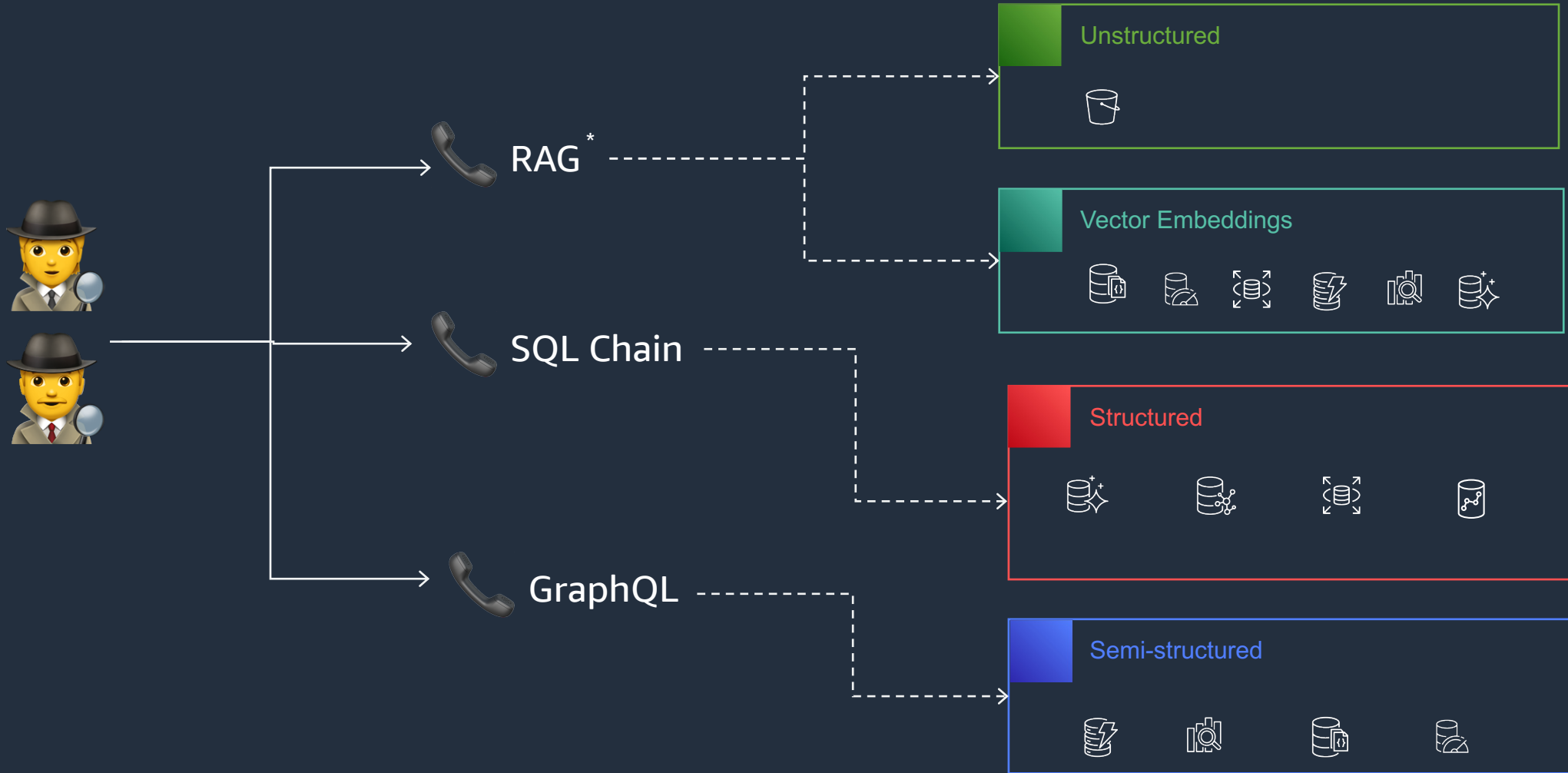
Run Query



Query Result



Let's dial the right numbers



* Retrieval Augmented Generation is the general process of combining data with your prompt. It is not specific to Embeddings.

Key Takeaways



**Connect Amazon Bedrock to
your data with
AWS AppSync and GraphQL**



**Amazon Bedrock, RAG,
and SQLDatabaseChain**



**Question answering using
Retrieval Augmented Generation**



Thank you!

Puria Izady

 puria-izady

pizady@amazon.de



Please complete the session survey.

