



➤ Build a ChatGPT RAG Data Pipeline with RisingWave Stream Processor and Vector Store

Mary Grygleski
AI Practice Lead, Callibrity
X [@mgrygles](#)

Rayees Pasha
Chief Product Officer, RisingWave
X [@ProductPasha](#)

November 2024



Agenda

- Who are we?
- A Brief Overview of GenAI / ChatGPT
 - How does Data Pipeline fit in?
 - Techniques - RAG
- What is RisingWave?
 - How does Stream processing help with Real-time RAG?
- What is Vector DB?
- Demo & Building a ChatGPT Data Pipeline
 - An architectural conceptual walkthrough
- Resources/Q&A



Who are we?



Passionate
Advocate



Java Champion



[mgrygles](#)



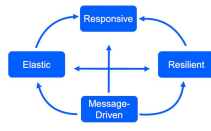
[mary-grygleski](#)



[mgrygles](#)



[mgrygles](#)



AI Practice Lead @  callibrity

- Generative AI
- Streaming
- Distributed Systems
- Reactive Systems
- IoT/MQTT
- Real-Time AI/ML



Rayees Pasha

Chief Product Officer, RisingWave Labs

- Expertise in Database Management and Stream processing Technologies
- Prior PM leadership positions at TigerGraph, Workday, Hitachi and HP
- MS in Computer Science from University of Memphis



RisingWave Labs



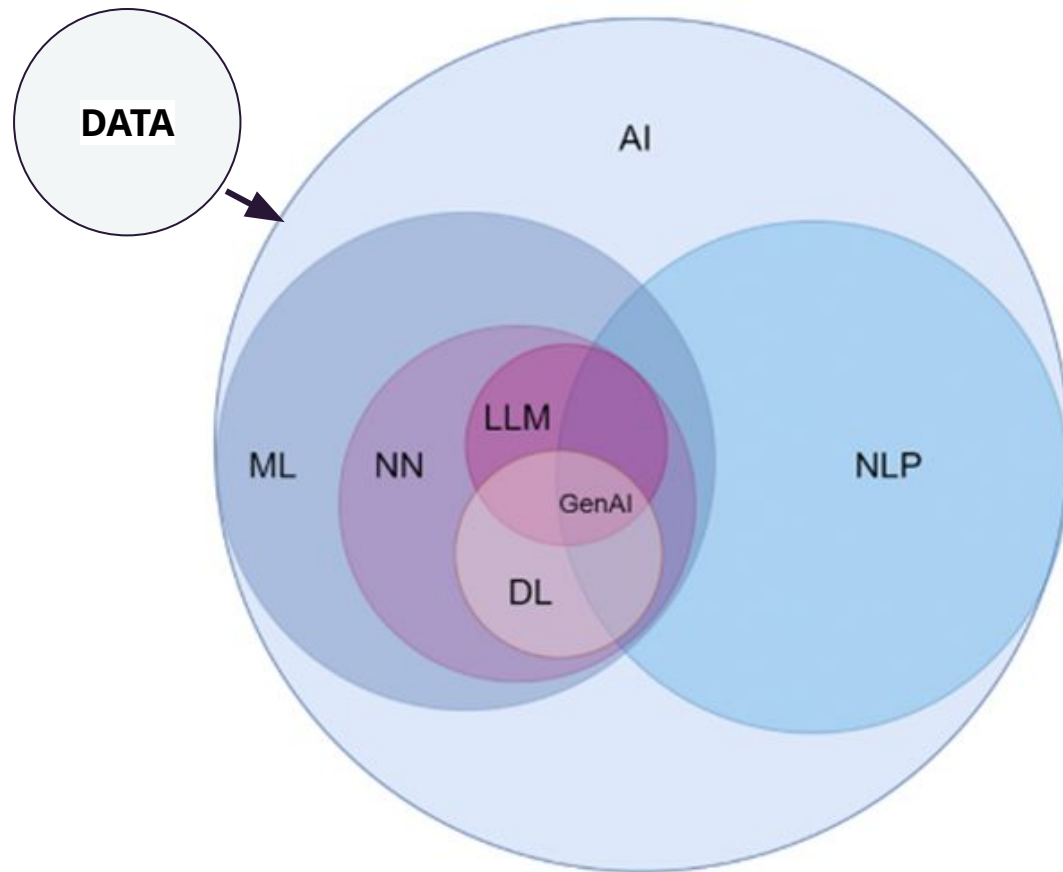
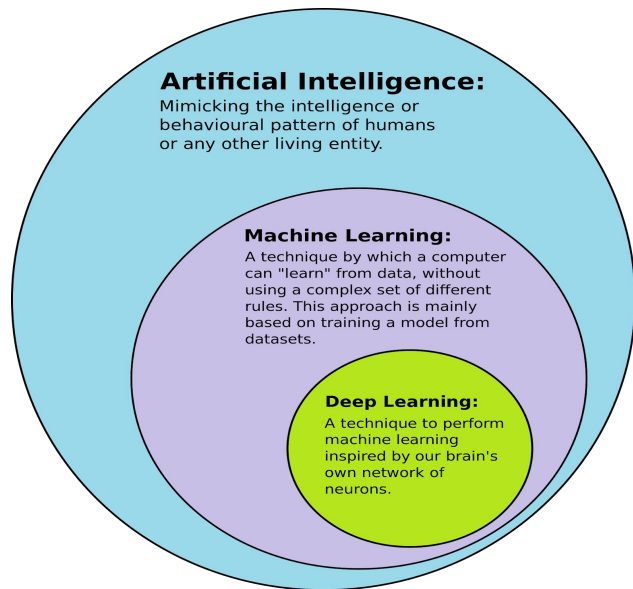
rayees-pasha



ProductPasha



➤ A Brief Overview of GenAI / ChatGPT, Streaming



What is Generative AI (GenAI) ?

- A “disruptive” field in AI
- Has the potential to change the way we create and consume content
- Generate new contents based on prompts
- Uses a combination of machine learning and deep learning to produce contents
- Tends to be on the “creative” side: generating code, writing an article, designing new fashion, composing a new song... especially when compared with Predictive AI which tends to be more strictly about business, marketing, and weather forecasting.

What is Generative Pre-Trained Transformer (GPT) ?

- Takes simple prompts (in natural human language) as input
- Pattern matching (also commonly being called as “search”)
- Answers questions for the **prompts**
- Produce contents such as: a new essay, a blog post, a new computer program

What is Natural Language Processing (NLP)?

- An interdisciplinary sub-field of linguistics of computer science
- Primary concern is to process natural language datasets (as such text corpora or speech corpora)
- Uses rule-based or probabilistic machine learning approaches
- Enables computer to learn from contents, including the contextual nuances of the language itself
- Ideally to draw insights from the documents

What is Large Language Models (LLMs)

- A type of Machine Learning Model
- Foundation type of model
- Typically the pre-training consumes a humongous amount of resources: \$\$\$\$\$\$ GPUs, multiple weeks of processing
- Performs NLP tasks
- Generates and classifies texts
- Answers questions (**prompts**) just like a human: analyze sentiments, chatbot conversations, etc.



The Event-Driven World and AI ?

› At the Heart of AI:

DATA and the
***FLOW* of Data** in
many directions



Working with Data

- Architecture level: Event-Driven Architecture
- Handling Data in different States:
 - At Rest
 - In Motion
 - In Use
- Techniques:
 - Event Streaming
 - Event Messaging
 - Event Sourcing & CQRS

Generative AI Streaming usages

- GenAI App Input
 - Prompts
 - Command-line
 - Browser (different devices)
 - Mobile App
- GenAI Intermediary steps:
 - During RAG operations:
 - different sources (DB, filesystem, socket?, message queue, Kafka topic...)
 - Sending requests to LLMs
- GenAI App Output
 - Responses sent from LLMs

Retrieval-Augmented Generation (RAG)

What is RAG?

- A hybrid framework that integrates the 2 components of the RAG models:
 - Retrieval model
 - Generative model
- The purpose is to produce text that is not only contextually accurate but also information-rich

Role of the Retrieval Model

- In a nutshell, the retrieval model acts as a specialized 'librarian', pulling in relevant information from a database or a corpus of documents.
- This information is then fed to the generative model, which acts as a 'writer,' crafting coherent and informative text based on the retrieved data. The two work in tandem to provide answers that are not only accurate but also contextually rich.

Role of the Generative Model

- Act as creative writers, synthesizing the retrieved information into coherent and contextually relevant text.
- Usually built upon Large Language Models (LLMs), generative models have the capability to create text that is grammatically correct, semantically meaningful, and aligned with the initial query or prompt.
- They take the raw data selected by the retrieval models and give it a narrative structure, making the information easily digestible and actionable.
- In the RAG framework, generative models serve as the final piece of the puzzle, providing the textual output we interact with.



RisingWave

An Open source Distributed PostgreSQL
Stream Processing Database

What really powers AI? -> Data

RAG Business Applications

- Chatbots
- Co-pilots
- Enhanced Search Capabilities
- Knowledge mining



Foundational Models



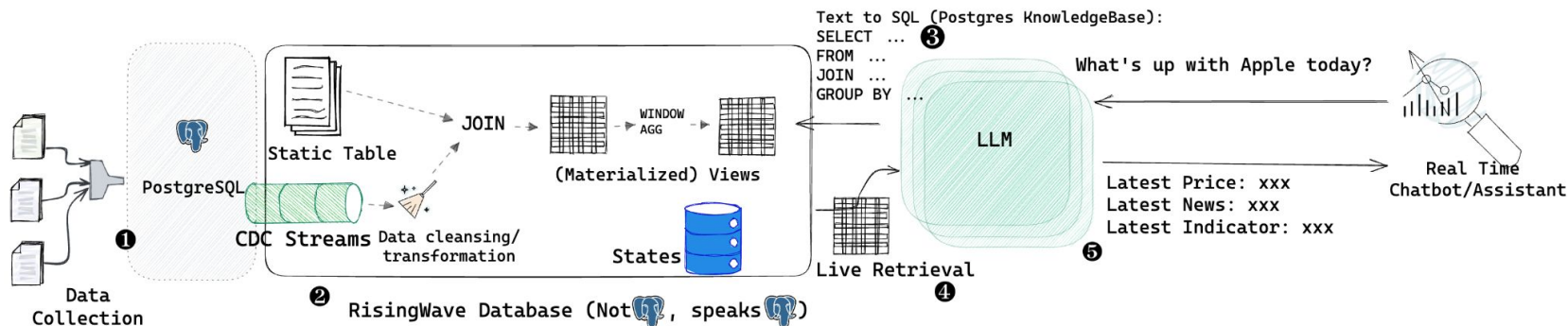
Enterprise Proprietary Data

Challenges with LLMs:

- Based on Bounded data - Knowledge cutoff
- Lacks Relevance - No access to proprietary data to provide context
- Data Quality issues - Hard to ensure consistency and validation

	CRM	Finance	E-Commerce
- Structured Data	name, addr contact info, trx history	market OHLC data, trx logs	catalogs, orders, pricing, inventory
- Semi-structured Data	forms, Support tickets	financial reports, market feeds	product skus, customer feedbacks
- Unstructured Data	transcripts, free-forms	reports, articles, tweets	Images, videos, customer reviews

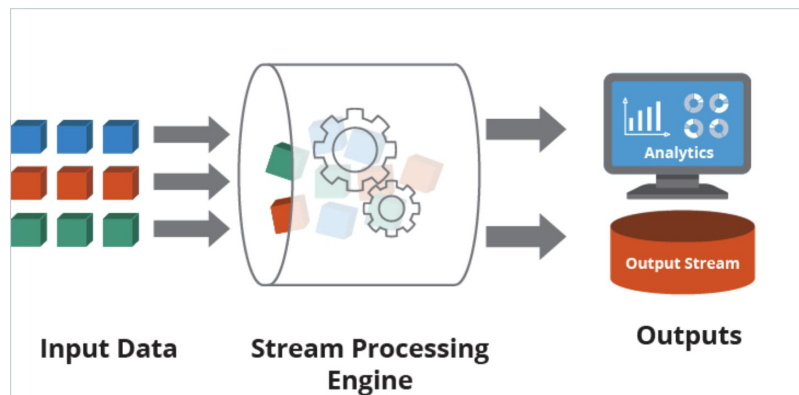
Real Time RAG with Streaming Database



- ① Data Collection:** Clickstream events mapping user activities (append-only) and Operational databases showcasing real-time business operations (upsert)
- ② Data Transformation:** Streaming databases enable data pipelines all-in-PostgreSQL: Real-time data transformations and cleanups, Efficient data enrichment like joining and aggregation, Real-time indexing, in consistency.
- ③ Prompt Engineering (Text to SQL):** English is the new Postgres client interface (connector).
- ④ Live Data Retrieval:** Everything is up-to-date through Incremental View Maintenance.
- ⑤ Real-time RAG:** Real-time data interpretation.

What is Stream Processing?

- Is a data processing approach that deals with continuously flowing (streaming) data.
- It involves the real-time analysis, transformation, and computation of data as it arrives..
- Is used to extract valuable insights, detect anomalies, and trigger actions based on incoming data.



What is RisingWave?

RisingWave is ..

- A distributed SQL streaming database
- Purpose-built for processing streaming data
- Open sourced in April 2022 under Apache 2.0 License



~7K GitHub stars



~2K Slack members

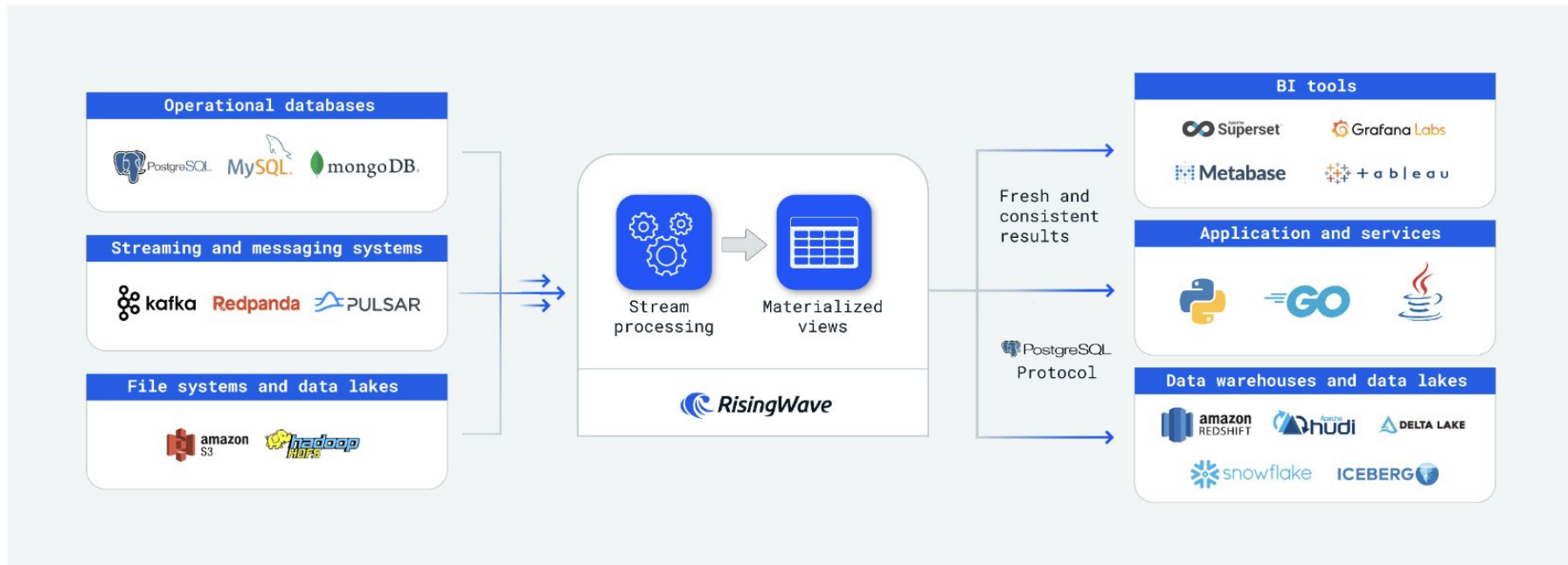


~ 9K+ Community Members



~150K K8s deployments

RisingWave Ecosystem



RisingWave Core Features



Distributed Streaming Engine

- Native streaming design for speed and efficiency
- Rust implementation for high performance
- Use of tiered caches for performant data access

Cloud-native Platform

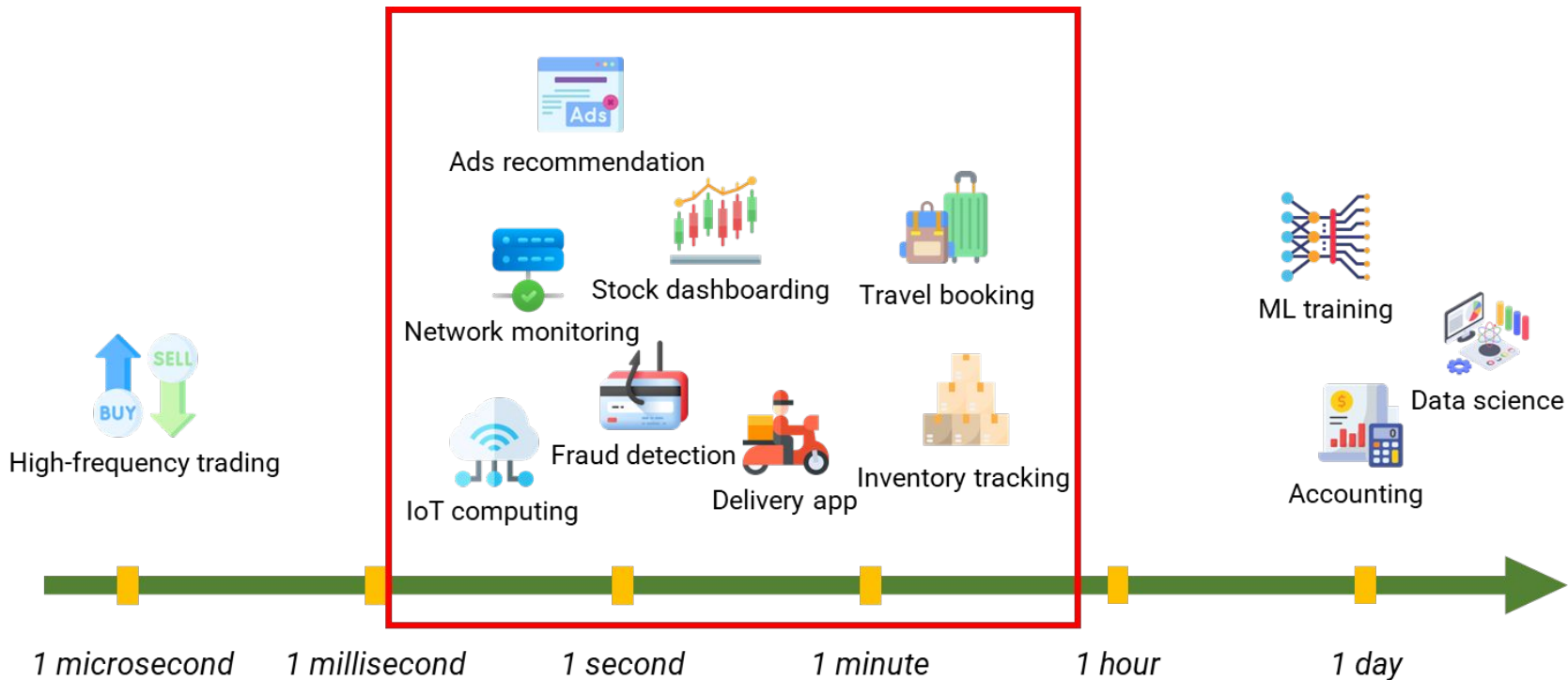
- Supports heterogeneous elastic scalability
- Separation of data and compute
- On-demand deployment model
- Independent from Zookeeper and other dependencies

Open Ecosystem

- SQL-based interface with full PostgreSQL wire compatibility
- Improved expressiveness with UDF support
- Cost-based query optimizer
- Integrates with open standards and computing frameworks



RisingWave – Real-world Use Cases





➤ Vector Search (PgVector)

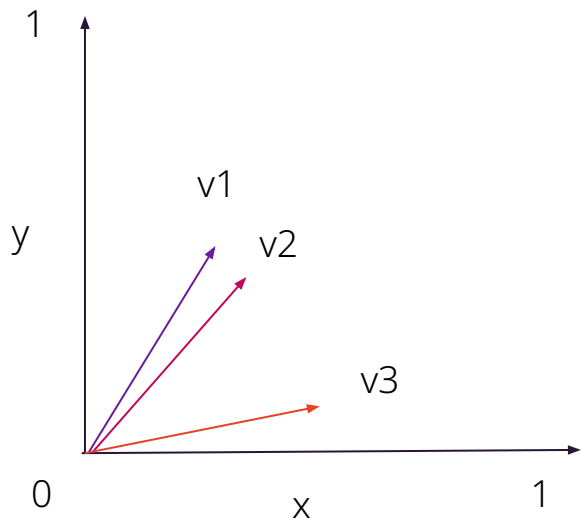
What is Vector Database (DB)

- A purpose-built database that serves up vector data type for complex machine learning purposes
- Relies on vector embeddings which are numerical representations of the data that are stored in vector DB
- An automatic “feature engineering”
- Approximate Nearest Neighbor (ANN)

What are vector embeddings being used for?

- Search (where results are ranked by relevance to a query string)
- Clustering (where text strings are grouped by similarity)
- Recommendations (where items with related text strings are recommended)
- Anomaly detection (where outliers with little relatedness are identified)
- Diversity measurement (where similarity distributions are analyzed)
- Classification (where text strings are classified by their most similar label)

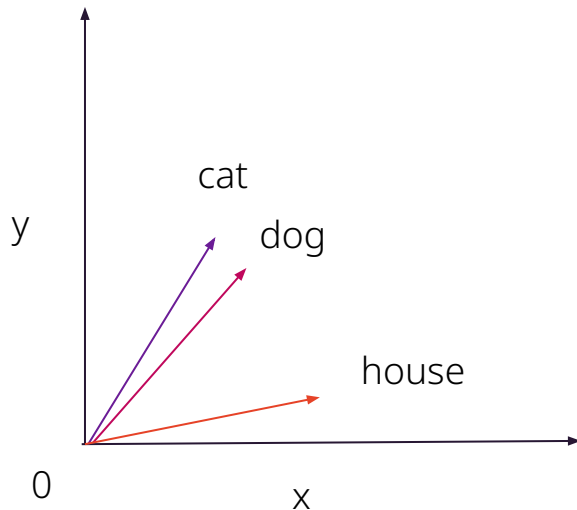
► Mechanism: What is a vector/embedding ?



2 dimensions normalised vectors

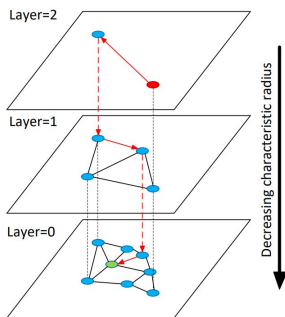
- An embedding model transforms a text into a vector called an embedding.
- The embedding can be N dimensions. For instance OpenAI's embeddings are 1536 dimensions.
- Similarity: v1 is more similar to v2 than v3. This is a simple mathematical formula.

► (cont'd): What is a vector/embedding ?



- The vector captures the essence of a word or a block of text within its context.
- The dimensions are the result of the LLM training.

➤ Vector search



Vector stores / vector databases

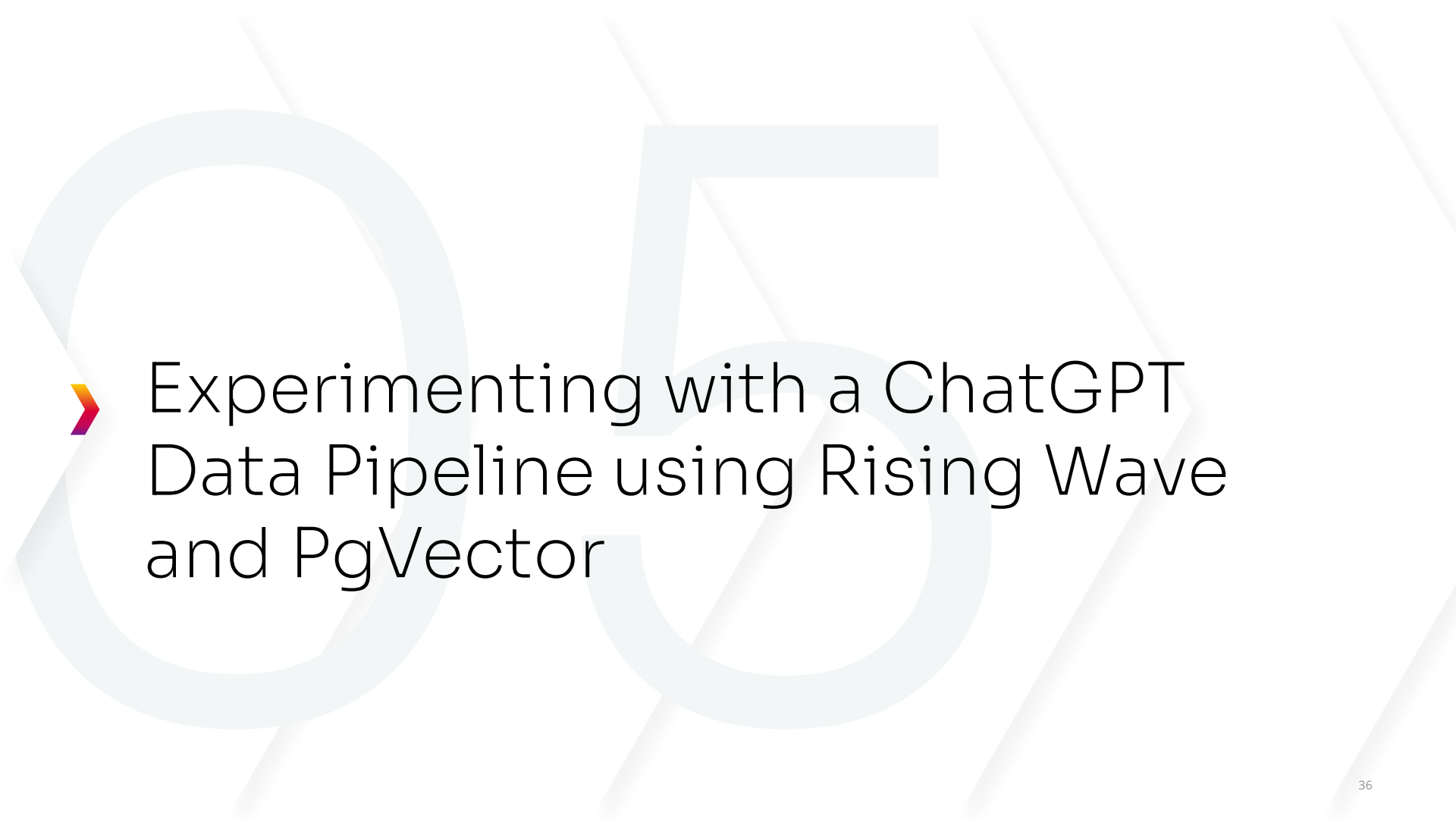
- Embeddings storage (with or without metadata depending on the DB)
- Built-in algorithms for fast retrieval of so-called “nearest-neighbors” embeddings (eg. HNSW, JVector, ...)
- Vectors are a new type of data supported in established databases (DataStax AstraDB, Cassandra, ...) and new specialized databases (Pinecone, Milvus, ...)

The Problem with “Traditional” DB in AI

- Unable to handle the complex data that's required in AI to handle the dimensions, patterns and relationships
- Should function like human memories but not so
- Essentially we need to provide the context for GenAI processing
- Cannot be used to store and querying of high dimensional vector data

A typical pipeline that brings data to the Vector DB

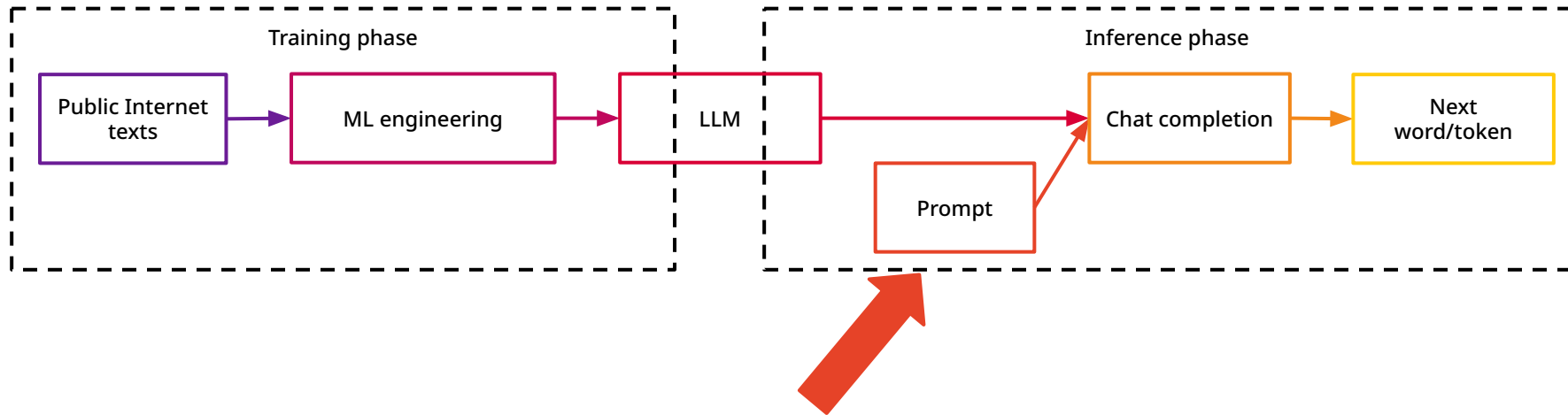
- Read data from a source (e.g. an S3 bucket, a WebSite, a Kafka topic, etc.)
- Process the data to extract the text to vectorise
- Split the text into chunks of a given size
- Compute vector embeddings for each chunk
- Write the chunks to the vector database
- Clean up obsolete data from the vector database



Experimenting with a ChatGPT Data Pipeline using Rising Wave and PgVector

› Lowering the barrier to AI

GPT (Large Language Model)



A very high-level description of the data flow of the sample

twitter feed

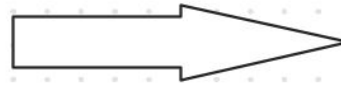


twitter messages
will come as a
streaming feed
(based on Rising
Wave's example)
from Kafka

Rising Wave

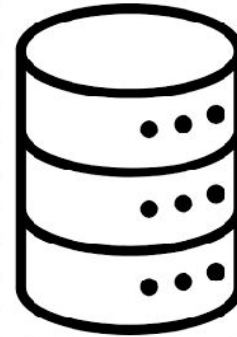


sort thru the feed and
find the tags of interest



Construct the chat
prompts based on
the tags
accordingly, and
send these prompts
to the chat bot
which also
leverages on Astra
DB vector

Vector DB
(PgVector,
Qdrant,
Weaviate,
Milvus, etc





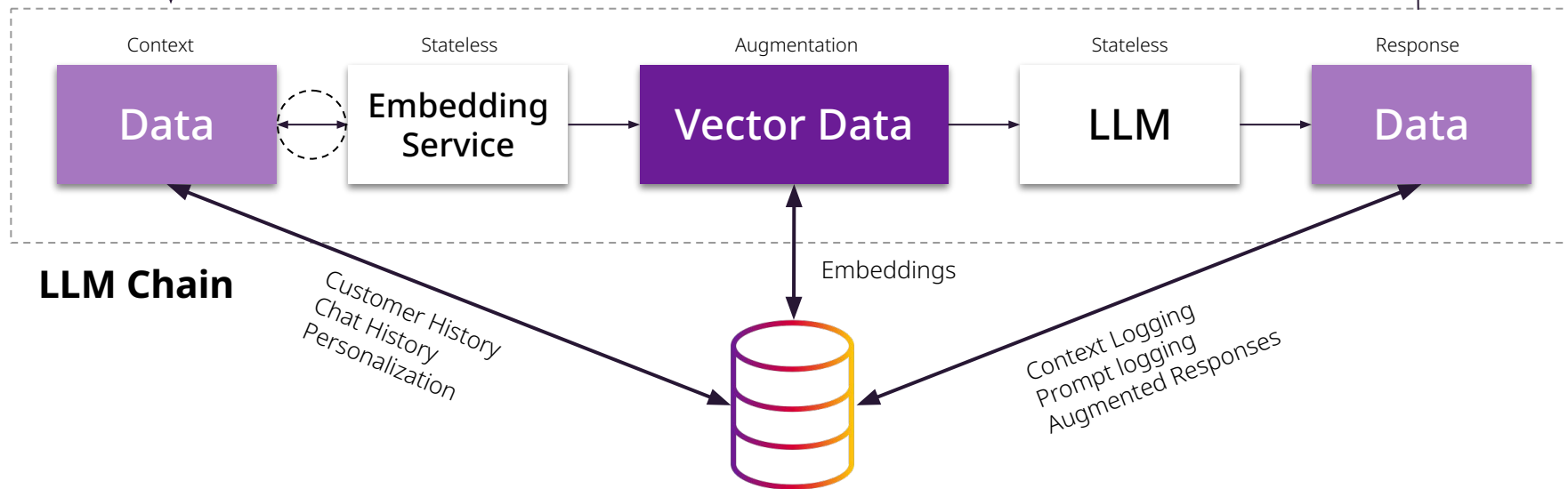
User
Input

Prompt

Generative (RAG) AI Apps

System
Response

Augmented
Response





Resources

This slide deck can be accessed here:

<https://bit.ly/3O5GhgU>





Get started with RisingWave

Keeping it simple for you guys (just like RisingWave is meant to be! 😊) :

RisingWave.com

docs.risingwave.com/docs/current/intro

RisingWave Fast X events example used for the presentation:

<https://docs.risingwave.com/docs/current/fast-twitter-events-processing/>

Follow Mary's Stream & The Elusive AI Podcast

[Different topics: GenAI/ChatGPT, Java, Python, JS/TS, Open Source, Distributed Messaging, Event-Streaming, Cloud, DevOps, etc]

Wed|Thurs|Fri-afternoon-US/CST



<https://twitch.tv/mgrygles>



<https://youtube.com/@marygrygleski9271>

Comments and
feedback here



 **Thank You**

Mary Grygleski



in <https://www.linkedin.com/in/mary-grygleski/>

X [@mgrygles](https://twitter.com/mgrygles)

 <https://discord.gg/RMU4Juw>



Rayees Pasha



in <https://www.linkedin.com/in/rayees-pasha/>

X [@ProductPasha](https://twitter.com/ProductPasha)

 <https://risingwave.com>