

# **Bounded and Unbounded Data for RAG Systems: PostgreSQL Is All You Need**

**Samuel Hu**  
**RisingWave Labs, Jan. 2024**

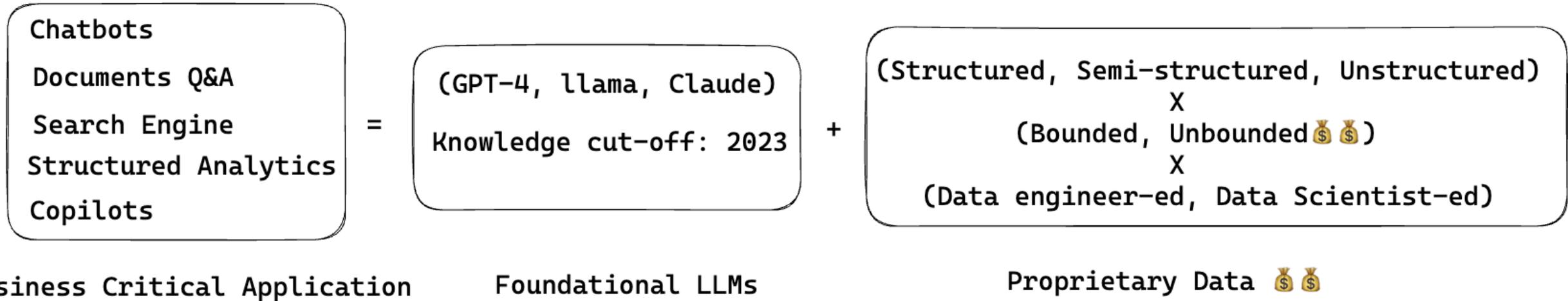
## › About Me

- Wearing multiple hats at RisingWave Labs – a \$40M funded startup – kernel lead, cloud lead to customer lead.
- Used to be a PostgreSQL kernel hacker (shared storage, many-core performance optimization, etc..)
- Founding engineer of Azure PostgreSQL and MySQL services.

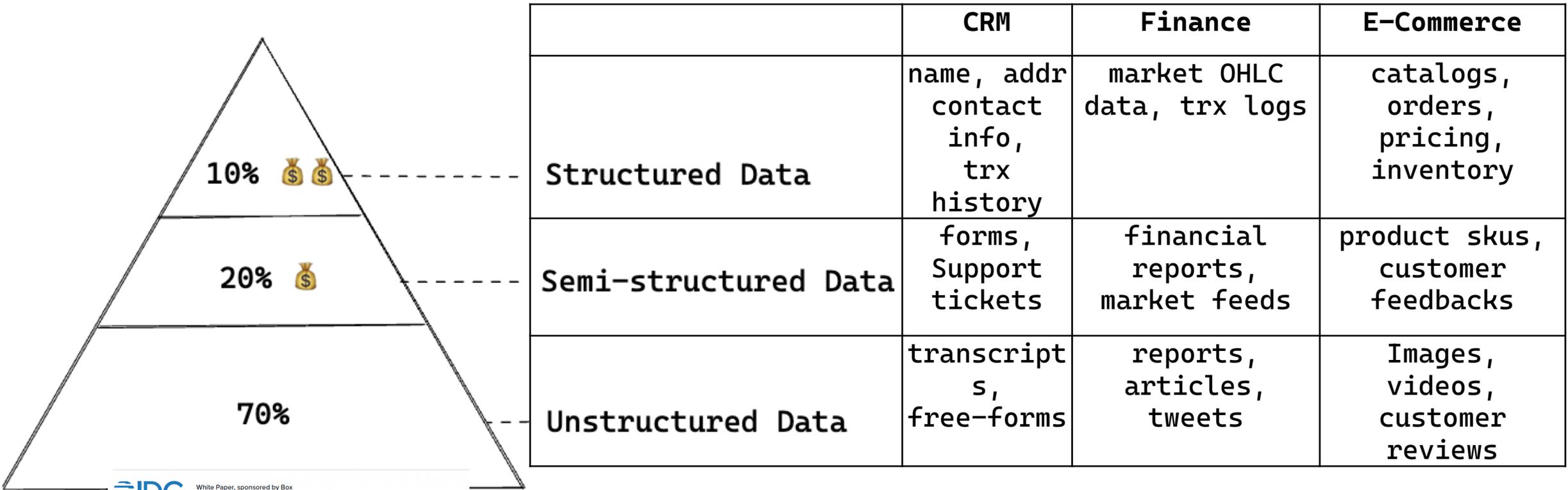


1. Bounded Data, Static RAG and PostgreSQL
2. Unbounded Data, Real Time RAG and PostgreSQL
3. The Future of PostgreSQL in AI and RAG Systems

# RAG empowers LLM, Data empowers RAG.



# Structured/Semi-Structured Data are the Core Value



White Paper, sponsored by Box  
August 2023 | IDC #US5128223

# Status Quo: Bounded Data, Static RAG and PostgreSQL

↓ INSERT  
UPDATE  
DELETE  
COPY TO  
TRUNCATE

↑ Text to SQL

↓ vector calculation  
vector indexing

↑ similarity search

**Rich Data Types:**  
Integers, floats, datetime,  
JSONB, array, xml  
geometric, vector

**Full-fledged SQLs:**  
Nested queries,  
Window functions,  
(Recursive)CTEs,  
Expressions

**Fast Retrieval:**  
B-Tree Index,  
GIN Index,  
GiST Index  
Parallel Query

**Plugins:**  
pgvector ↔ Pinecone  
pgvecto.rs ↔ pgvector  
ParadeDB ↔ ElasticSearch  
TimescaleDB ↔ InfluxDB

**RisingWaveCopilot**

**You**  
How to convert jsonb array to array type in RisingWave? something like  
`select jsonb_array_to_text_array(['1, 2, 3']); -- -> ARRAY['1', '2', '3']`

**RisingWaveCopilot**  
In RisingWave, you can convert a JSONB array to a text array using the `'jsonb_array_elements_text'` function. This function expands the top-level JSON array into a set of text (varchar) values. For example, you can use it as follows:

```
sql  
SELECT * FROM jsonb_array_elements_text(['1, 2, 3']::jsonb);
```

This will result in individual rows with values '1', '2', and '3' as text. To combine these into an array, you can use array aggregation:

```
sql  
SELECT array_agg(value) FROM jsonb_array_elements_text(['1, 2, 3']::jsonb) AS value;
```

This should give you an array of text values as desired: `'ARRAY['1', '2', '3']'`.

**Knowledge**  
If you upload files under Knowledge, conversations with your GPT may include file contents. Files can be downloaded when Code Interpreter is enabled

risingwave.dev.pdf  
PDF

Upload files

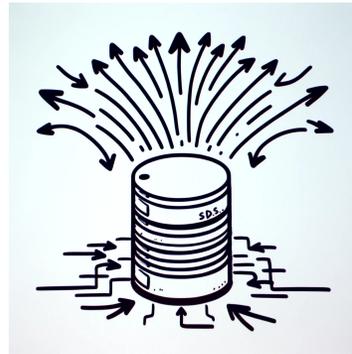
# Postgres Limitation in Unbounded Data and Real-Time RAG



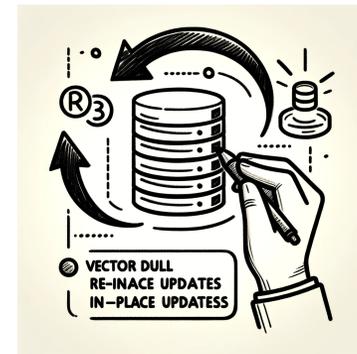
Real-time  
Data  
Streaming  
and  
Processing



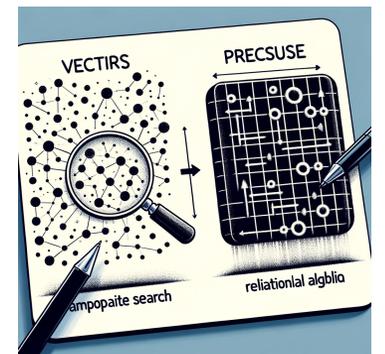
Scalability  
Challenges  
with Very  
Large  
Datasets



Write and  
Update  
Intensive  
Workloads



Vector: require  
full re-  
indexing with  
in-place  
updates.



Vector:  
approximate  
searching lack  
the precision  
of relational  
algebra.

# Imperatives of Real-Time RAG with Unbounded Data:

## Real-time reflections:

Staying attuned to global events without a lag.

## Scalability:

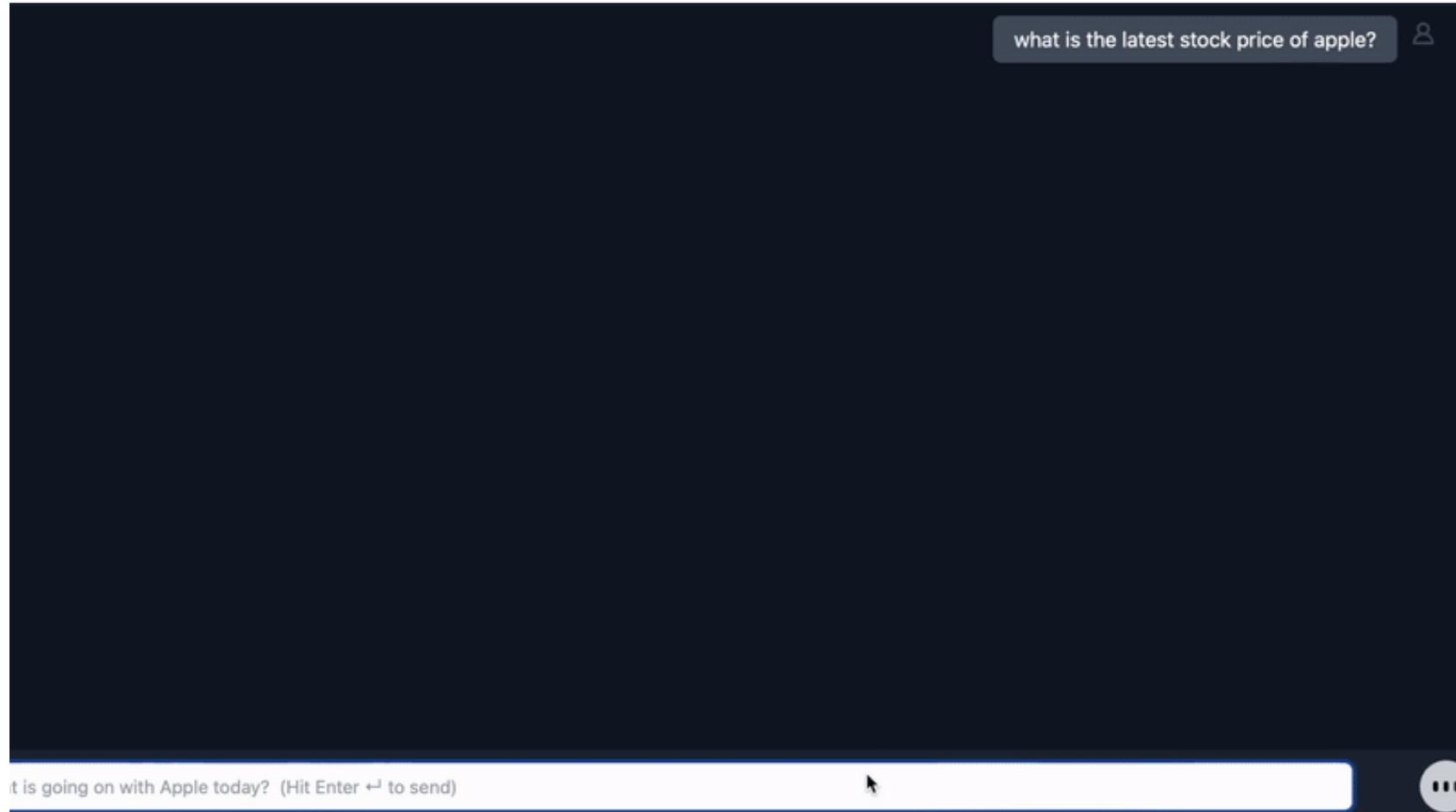
Handle increasing volumes of data.

## Data Integrity and Accuracy:

Consistency and algebra semantics.

## Interoperability:

Integrate with various data sources, systems, and data exchange standards.



# Unbounded Data, Real Time RAG and PostgreSQL

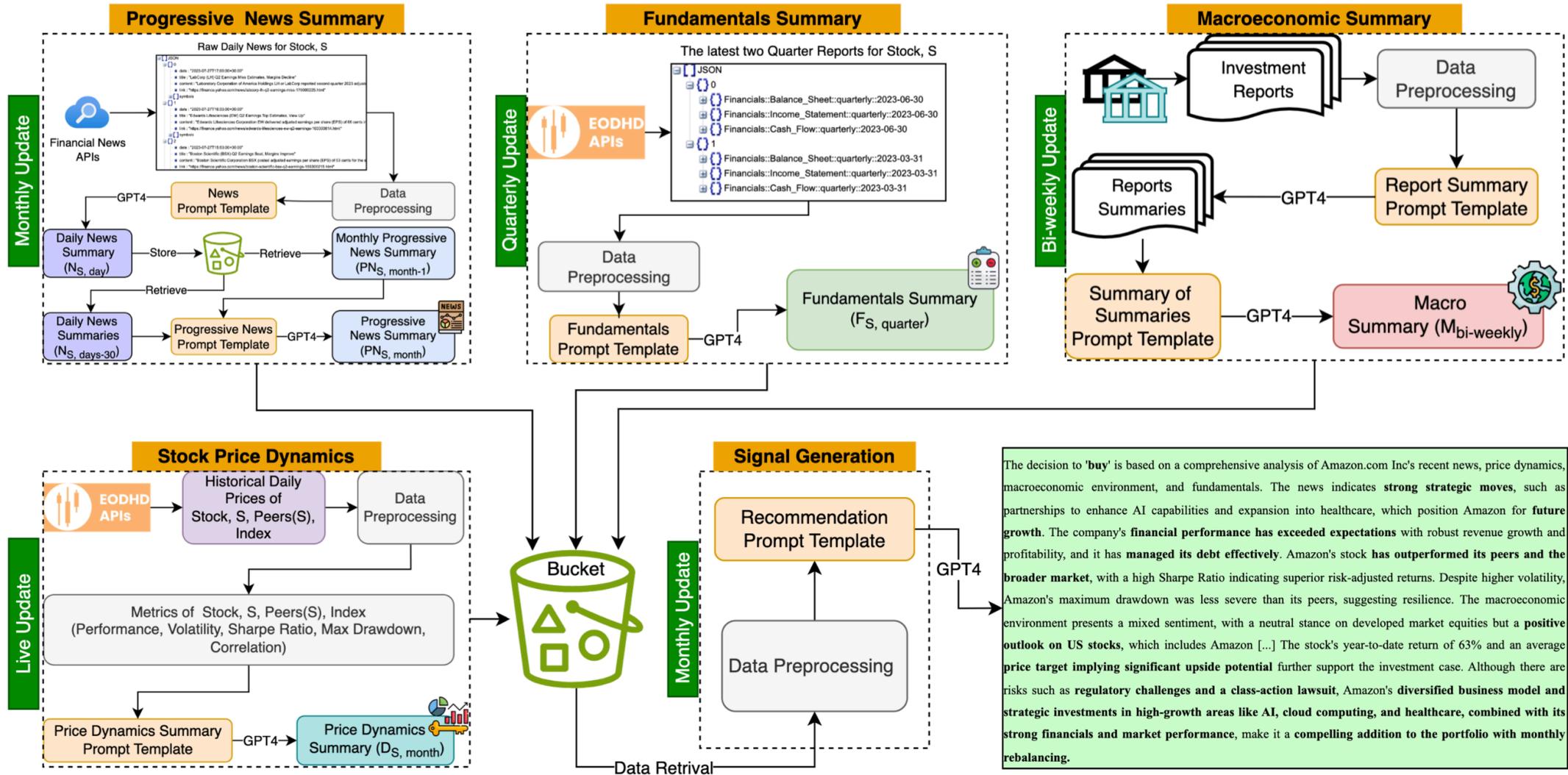
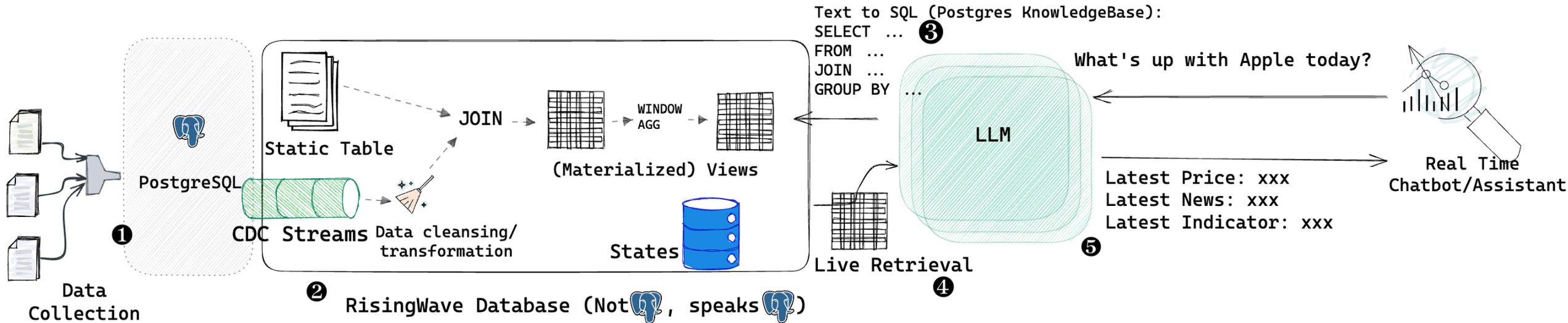


Figure 1: Conceptual architecture of MarketSenseAI, highlighting the core components, data flow, and outcome for a selected stock (e.g., Amazon).

# Unbounded Data, Real Time RAG and PostgreSQL



## 1 Data Collection:

- Clickstream events mapping user activities (append-only)
- Operational databases showcasing real-time business operations (upsert)

## 2 Streaming Databases: Streaming databases like RisingWave, data pipelines all-in-PostgreSQL:

- Real-time data transformations and cleanups.
- Efficient data enrichment like joining and aggregation.
- Real-time indexing, in consistency.

## 3 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.

## 6 System Scalability:

- CQRS (Command and Query Responsibility Segregation).

# Unbounded Data, Real Time RAG and PostgreSQL

```
CREATE TABLE IF NOT EXISTS tickers_bars_1d_yfinance (  
    "c" DOUBLE,  
    "h" DOUBLE,  
    "l" DOUBLE,  
    "n" INTEGER,  
    "o" DOUBLE,  
    "t" BIGINT,  
    "v" DOUBLE,  
    "vw" DOUBLE,  
    "ticker" STRING,  
    PRIMARY KEY ("ticker", "t")  
) WITH (  
    connector = 'kafka',  
xxx  
) ROW FORMAT JSON;
```

```
CREATE TABLE IF NOT EXISTS news_currencies_stocks (  
    "title" TEXT,  
    "link" TEXT,  
    "pubDate" TIMESTAMP,  
    "publisher" TEXT,  
    "summary" TEXT,  
    "currencies_impacted" TEXT[],  
    "currencies_sentiment_impact" INTEGER[],  
    "currencies_impact_reasoning" TEXT[],  
    "currencies_impact_confidence_level" INTEGER[],  
    "stocks_impacted" TEXT[],  
    "stocks_sentiment_impact" INTEGER[],  
    "stocks_impact_reasoning" TEXT[],  
    "stocks_impact_confidence_level" INTEGER[],  
    PRIMARY KEY ("publisher", "link")  
) WITH (  
    connector = 'kafka',  
) ROW FORMAT JSON;
```

(Data Collection)

# Unbounded Data, Real Time RAG and PostgreSQL

```
CREATE MATERIALIZED VIEW
news_currencies_stocks_unnested AS
SELECT
  "title",
  "link",
  "pubDate" AS "pub_date",
  "publisher",
  "summary",
  unnest("currencies_impacted") AS
"currency_impacted",
  unnest("currencies_sentiment_impact") AS
"currency_sentiment_impact",
  unnest("currencies_impact_reasoning") AS
"currency_impact_reasoning",
  unnest("currencies_impact_confidence_level") AS
"currency_impact_confidence_level",
  unnest("stocks_impacted") AS "stock_impacted",
  unnest("stocks_sentiment_impact") AS
"stock_sentiment_impact",
  unnest("stocks_impact_reasoning") AS
"stock_impact_reasoning",
  unnest("stocks_impact_confidence_level") AS
"stock_impact_confidence_level"
FROM
  news_currencies_stocks;
```

```
CREATE MATERIALIZED VIEW daily_stocks_trends_view AS
SELECT lt.ticker,
  lt.c AS current_price,
  (lt.c - pt.c) / pt.c AS relative_price_movement,
  (lt.v - pt.v) / pt.v AS relative_trading_volume,
  CASE
    WHEN prev_ni.sentiment_impact = 0 THEN NULL
    ELSE (ni.sentiment_impact - prev_ni.sentiment_impact) /
prev_ni.sentiment_impact
  END AS relative_sentiment_impact,
  CASE
    WHEN prev_ni.c = 0 THEN NULL
    ELSE (ni.c - prev_ni.c) / prev_ni.c
  END AS relative_news_count,
  TO_CHAR(lt.t, 'YYYY-MM-DD') AS date_part
FROM latest_stock_tickersBars lt
JOIN prev_stock_tickersBars pt ON lt.ticker = pt.ticker
JOIN news_stock_impacts_view ni ON lt.ticker = ni.stock_impacted
AND CAST(ni.t AS DATE) = CAST(lt.t AS DATE)
JOIN prev_news_stock_impacts_view prev_ni ON lt.ticker =
prev_ni.stock_impacted
AND CAST(prev_ni.t AS DATE) = CAST(pt.t AS DATE);
```

(Real-time Data Transformation, Enrichment and Serving)

# Unbounded Data, Real Time RAG and PostgreSQL

```
~/works/others/Toyon/be | main *5 !3 ?1 | python -m aiagent.agent | base Py | 16:19:23
Welcome to the interactive financial analysis assistant.
Please enter your financial question (or type 'exit' to quit): How did the news released last week impact the stock prices of Tesla?
2024-01-22 16:19:35.799 | INFO | __main__:reason_and_break_down:88 - Reasoned sub-tasks: {'Financial Data Analyst': 'Assess the impact of news released last week on the stock prices of Tesla (ticker symbol: TSLA).', 'Financial News Analyst': 'Compile the major news released about Tesla and their corresponding impact on stock prices from last week.'}
Sample data:
2024-01-22 16:19:35.800 | INFO | __main__:answer_question:168 - Subtasks: {'Financial Data Analyst': 'Assess the impact of news released last week on the stock prices of Tesla (ticker symbol: TSLA).', 'Financial News Analyst': 'Compile the major news released about Tesla and their corresponding impact on stock prices from last week.'}
2024-01-22 16:19:37.326 | INFO | aiagent.data_retriever:text_to_sql:258 - Sending to GPT. Retries: 0
2024-01-22 16:19:52.003 | DEBUG | aiagent.data_retriever:get_data_for_conversation:221 - SQL Query: SELECT ns."title", ns."pub_date", ns."publisher", ns."summary", ns."stock_impacted", ns."stock_sentiment_impact", ns."stock_impact_reasoning", ns."stock_impact_confidence_level", db."c", db."h", db."l", db."o", db."v" FROM news_stocks ns JOIN daily_bars db ON ns."stock_impacted" = db."ticker" AND ns."pub_date"::DATE = db."t" WHERE ns."pub_date"::DATE BETWEEN current_date - INTERVAL '7 days' AND current_date - INTERVAL '1 day' AND ns."stock_impacted" = 'TSLA' ORDER BY ns."pub_date" DESC LIMIT 10;
Answer: # Impact Analysis of Tesla's (TSLA) Stock Prices Based on Recent News
## Overview
Last week's news has shown a mix of positive and negative sentiment impacting Tesla's stock prices. The most significant factors are Elon Musk's statements on ownership and AI projects and the collaboration with Uber to boost electric vehicle adoption.
## Positive Sentiment Impact
* **Collaboration with Uber**: Tesla's effort to incentivize Uber drivers to adopt Model 3 and Model Y electric vehicles reflects a strategic move, likely to boost electric vehicle sales. This collaboration aligns with the increasing demand for eco-friendly transportation options.
* **Child Tax Credit Expansion**: The proposal to expand child tax credits could indirectly benefit Tesla as it may lead to increased consumer spending, including in the automotive sector.
* **Car Idling Policies**: Local laws against car idling could potentially increase the attractiveness of Tesla's electric vehicles, which do not require idling in
```

(Real-time RAG)

Thank you!