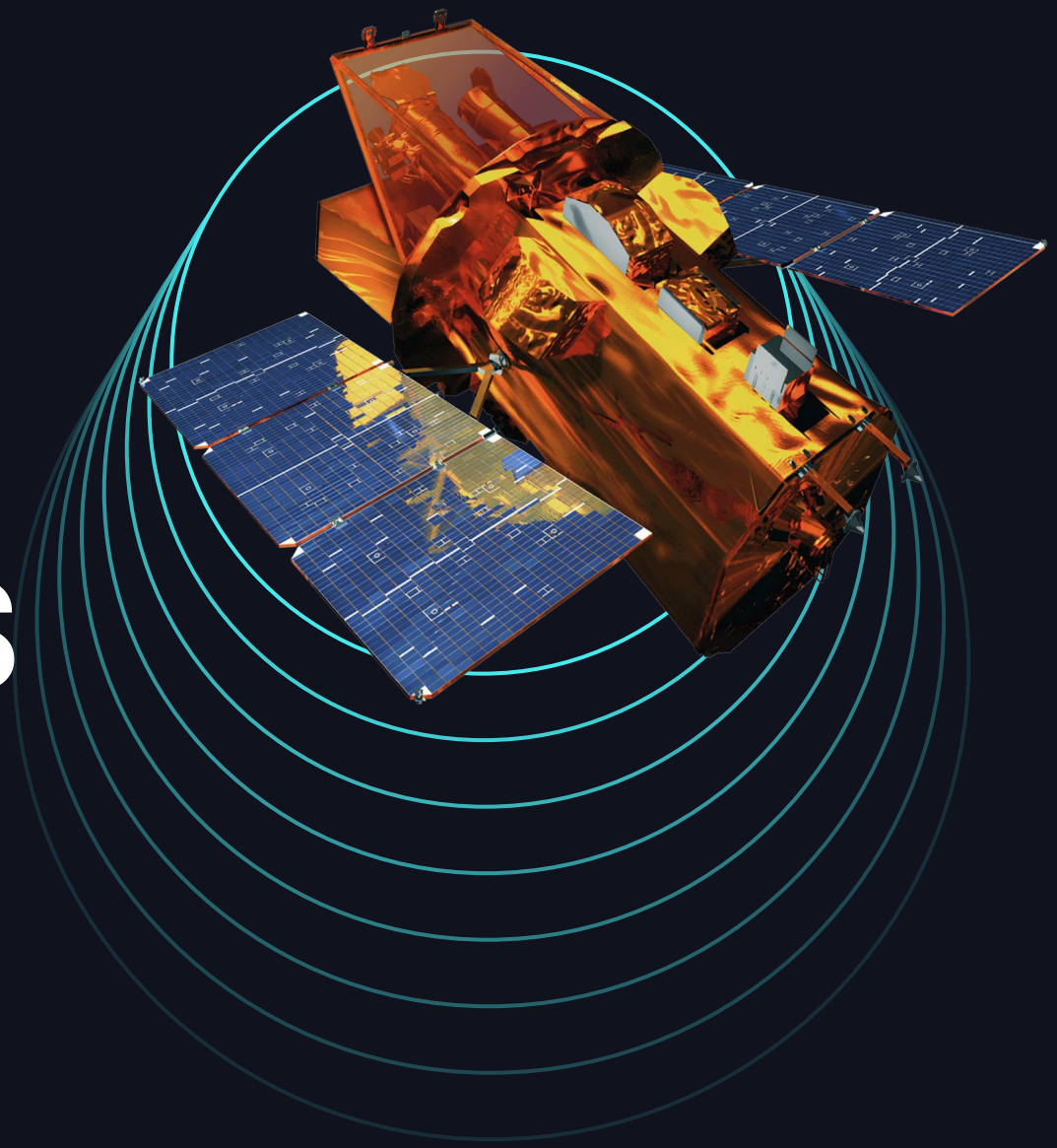
DATA+AI SUMMIT
BY databricks



# STREAMING DATA PIPELINES FROM SUPERNOVAS TO LLMS

Frank Munz, Databricks
June 2024

# Product safe harbor statement

This information is provided to outline Databricks' general product direction and is for **informational purposes only**. Customers who purchase Databricks services should make their purchase decisions relying solely upon services, features, and functions that are currently available. Unreleased features or functionality described in forward-looking statements are subject to change at Databricks discretion and may not be delivered as planned or at all

# Supernovas, Black Holes and GRBs

**1 Gamma Ray Burst (GRB) ~ energy of the sun over its lifetime**

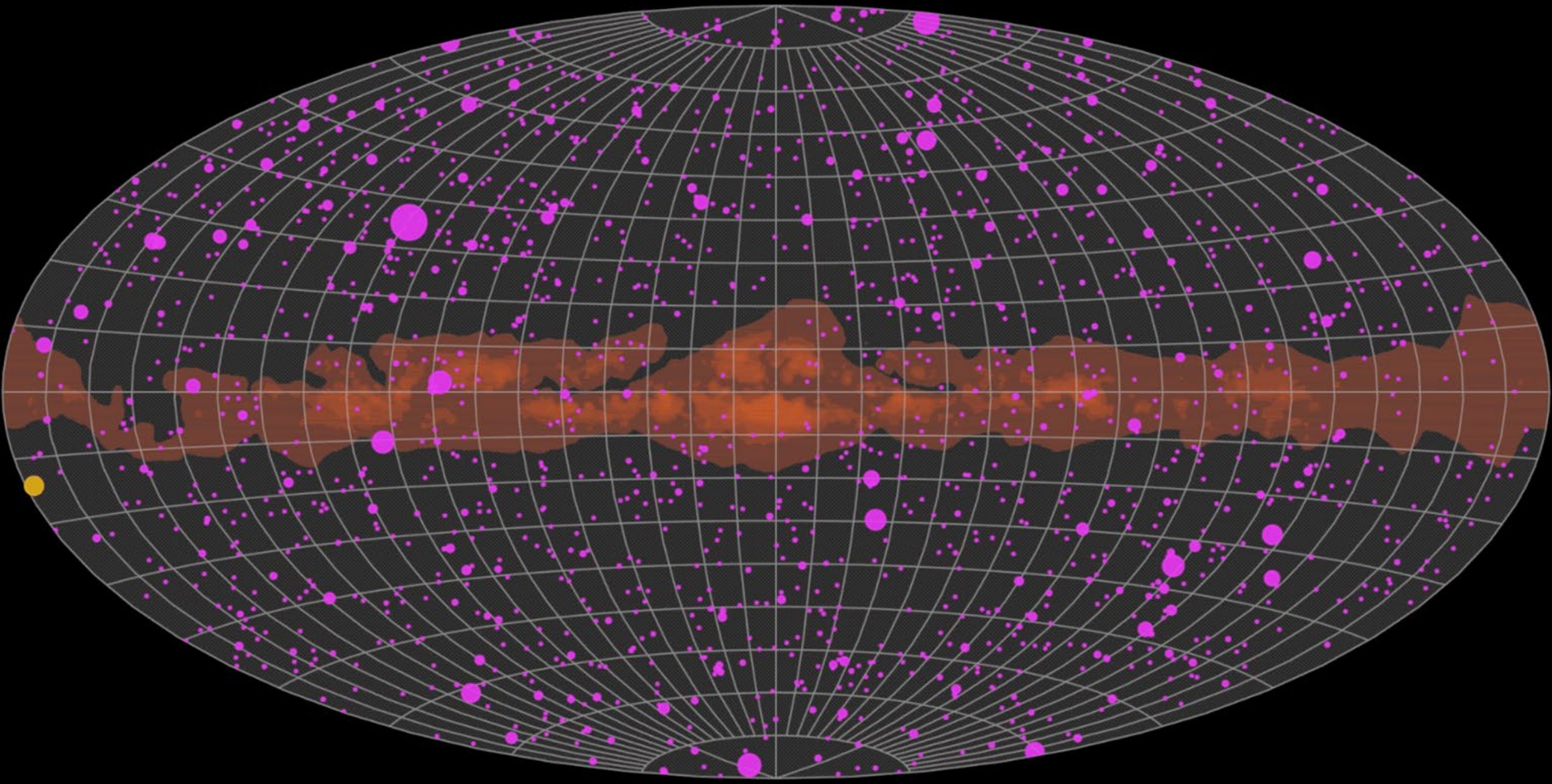
< 2 seconds: **merger** of neutron stars or a **neutron star and a black hole**
> 2 seconds: **collapse** of a **massive star** (> 30 solar masses)

**Supernova**

- Massive stellar explosions at the end of a star's life
- Can leave behind a black hole or neutron star

**Black Hole**

- Can form from the merger of 2 neutron stars or 2 black holes
- Extremely dense regions of space with immense gravitational pull

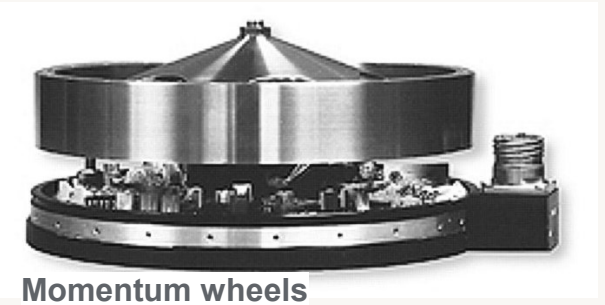


# Neil Gehrels Swift Observatory

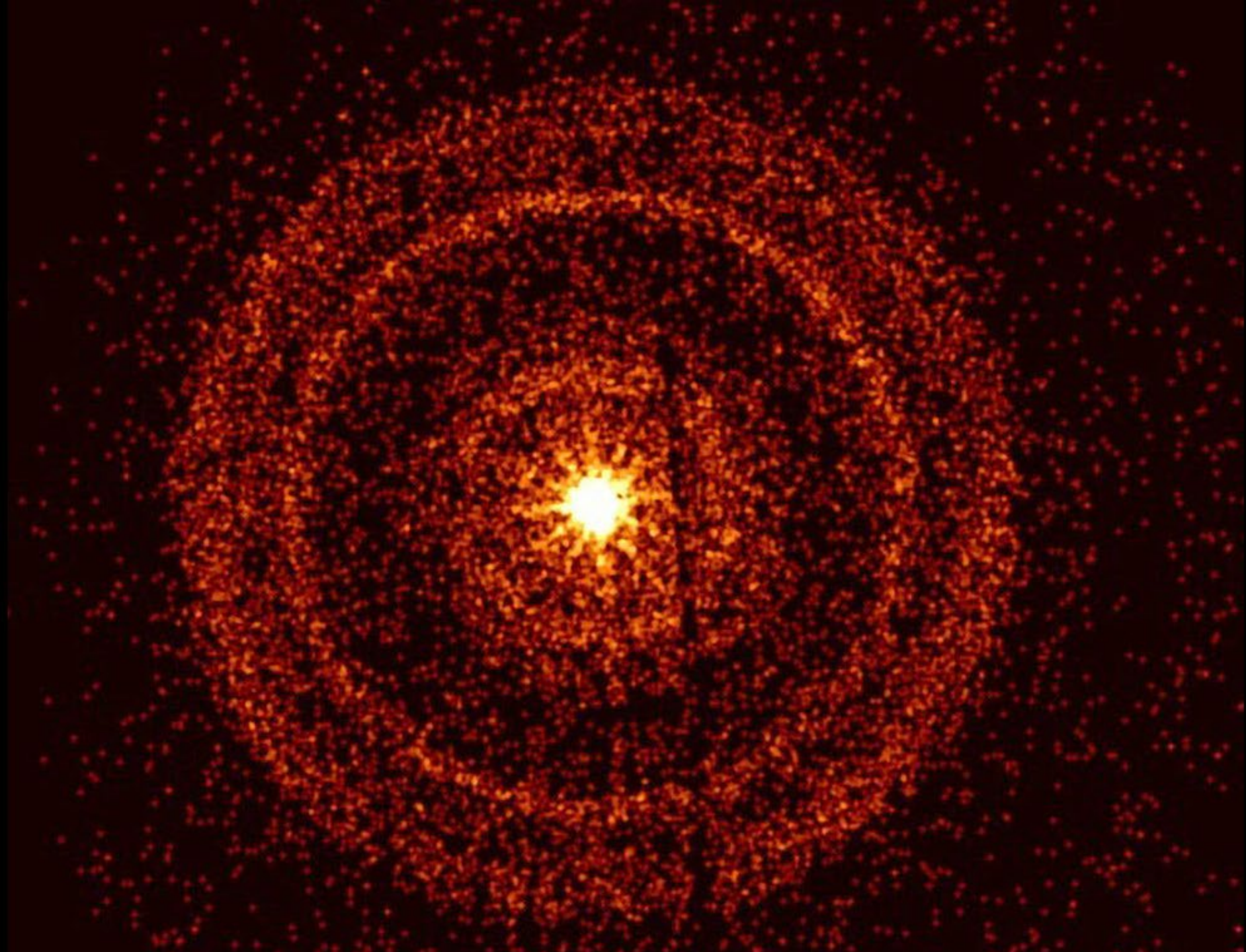
**Launched 2004,** Data transmitted via
**Gamma-ray Coordinates Network (GCN)**

**Key Instruments:**

- **Burst Alert Telescope (BAT)**:
  Locates GRBs across a wide field of view.
- **X-ray Telescope (XRT)**:
  Observes afterglow of GRBs in X-ray wavelengths
- **Ultraviolet/Optical Telescope (UVOT)**:
  Captures optical and ultraviolet emissions



Momentum wheels



# BOAT – GRB 221009A

## (Closest and) Brightest Of All Times  Gamma Ray Burst

- Detected on Oct 9, 2022 simultaneously by **Swift** and **Fermi** telescopes
- Originated 2.4 billion light-years away (**1.9 bn ago**) in Sagitta
- Lasted over **10 hours**, with 10 minute initial burst
- 5,000 VHE photons detected, previous record was ~100
- Brightest GRB afterglow ever recorded
- Thought to occur only **once every ~10,000 years**

Event time vs ingestion time?

# IceCube Neutrino Observatory

## Located in Anartica

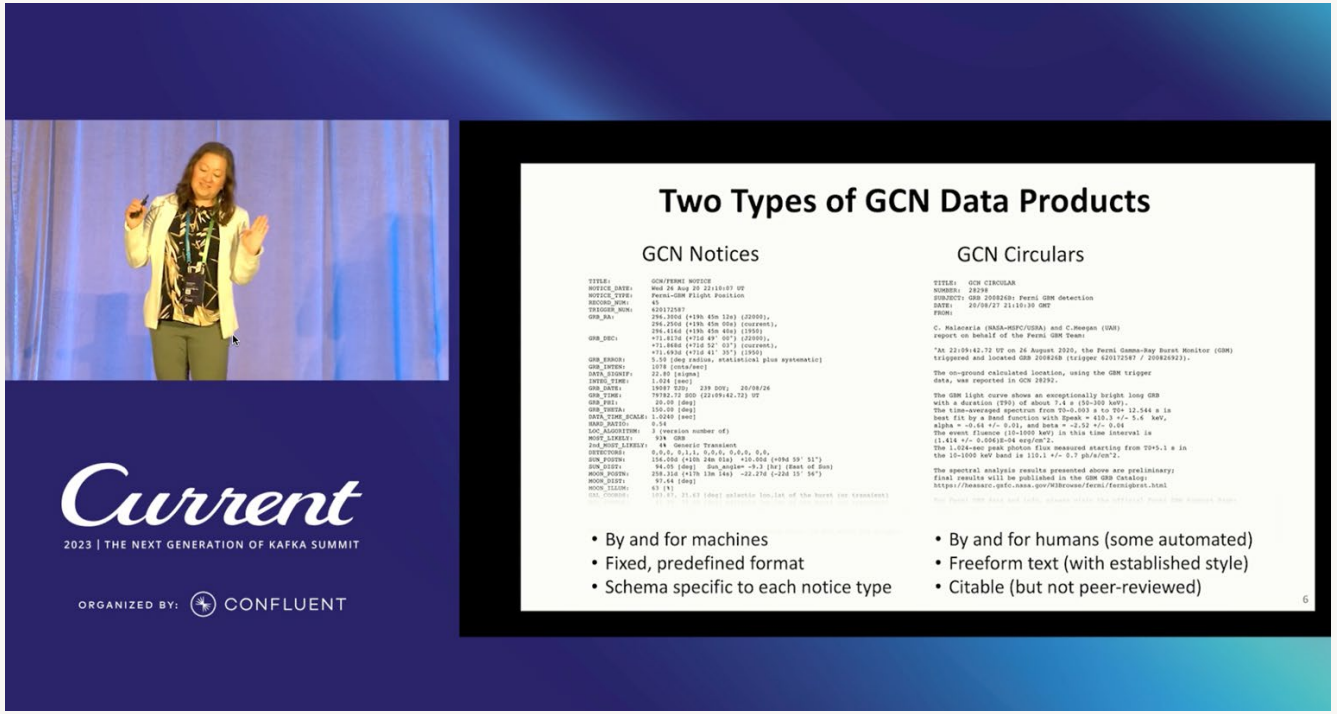
- Detects high-energy neutrinos from extreme cosmic environments
- 5,160 digital optical modules (DOMs)
- Embedded in **a cubic km of ice**
- Ice serves as detector medium and background radiation shield
- Neutrinos produce **Cherenkov radiation, detected by DOM**s

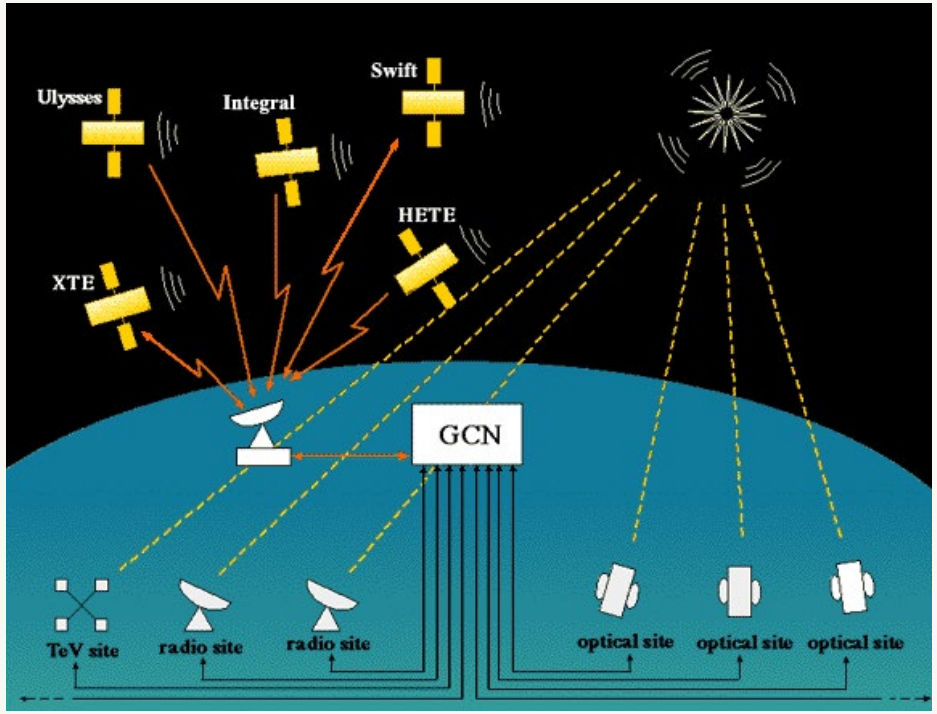
- Data transmitted via **Gamma-ray Coordinates Network (GCN)**
  -> alerts astronomers for quick follow-up observations

# NASA uses Apache Kafka: GCN Project



# Judith Rascusin's (NASA) Talk @ Current.io 2023



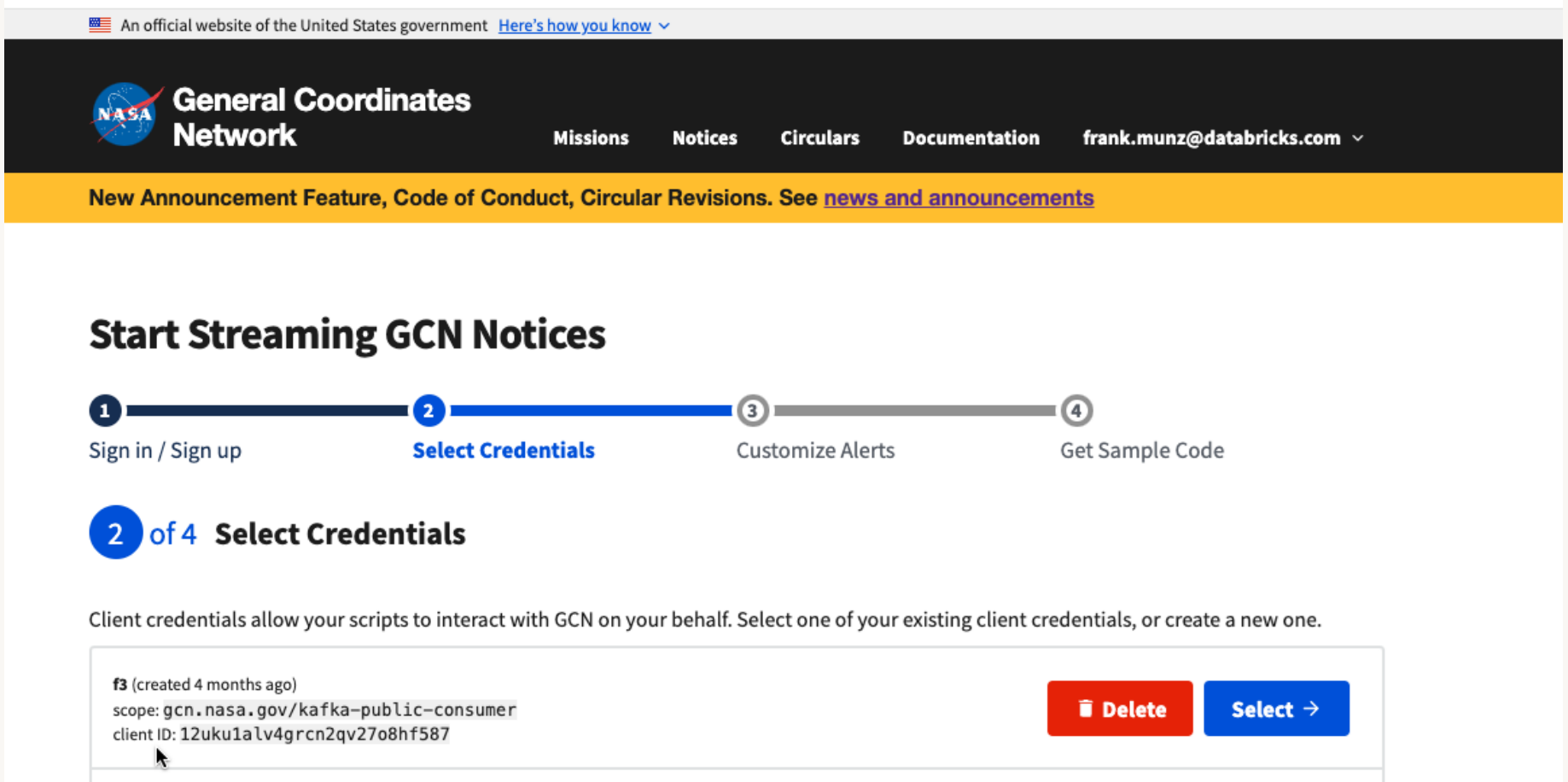

GCN **Notices**: machine generated

\+ **Circulars**: human generated

[Link to Judith's talk]

# Get Your OIDC Credentials

https://gcn.nasa.gov/quickstart

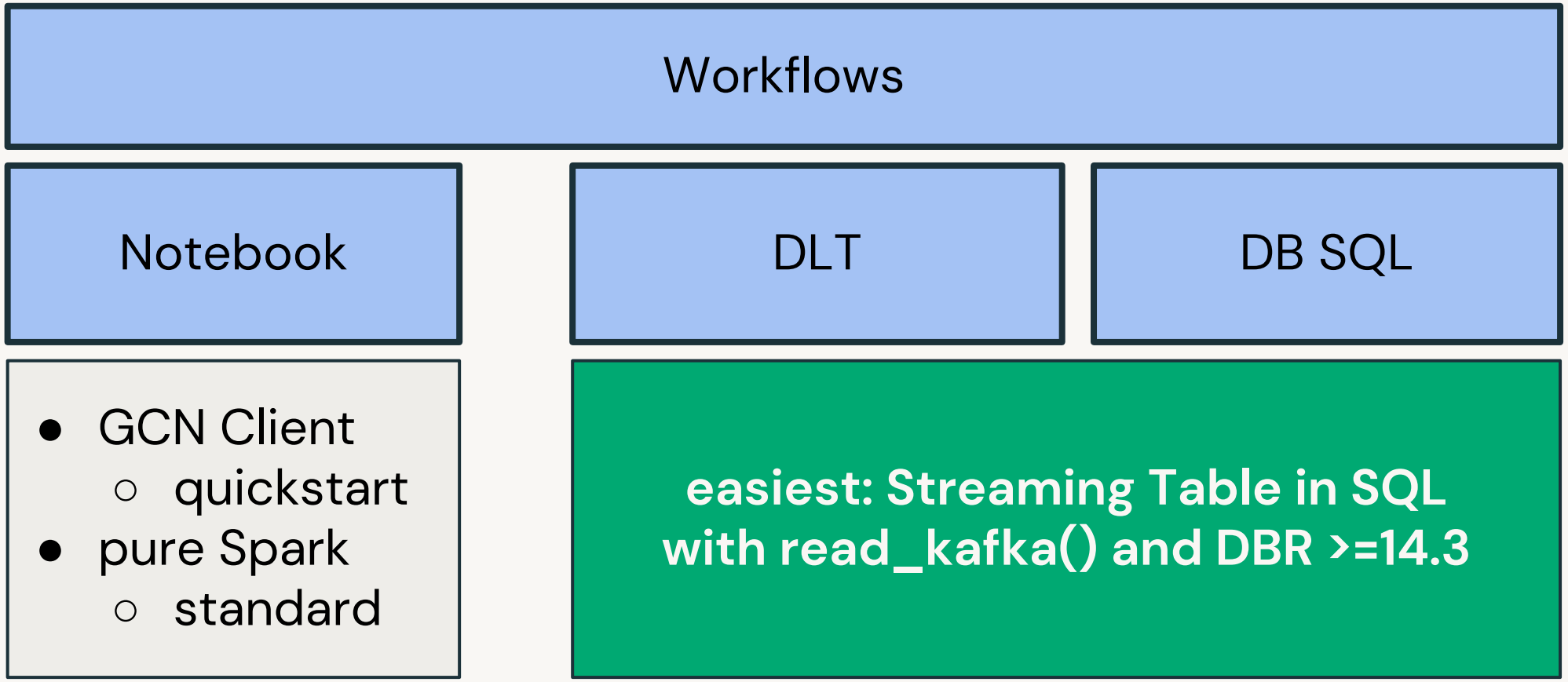


# The Data Intelligence Platform supports streaming data from the ground up

# The main actors (Ingest)

# Ingest Streaming Data from Apache Kafka

(we cover the human written circulars later…)



# Now, show me the code

# Notebook with GCN Kafka Wrapper

## Wraps Confluent Kafka Client

```
from gcn_kafka import Consumer

topics = ['gcn.classic.text.SWIFT_POINTDIR']
config = {'auto.offset.reset': 'earliest'}

consumer = Consumer(config,
                    client_id='abc…',
                    client_secret='xyz…',
                    domain='gcn.nasa.gov')
consumer.subscribe(topics)

while True:
    for message in consumer.consume(timeout=1):
```

**KAFKA message**

msg = TITLE: GCN/SWIFT NOTICE
NOTICE_DATE: Fri 03 May 24 04:16:31 UT
NOTICE_TYPE: SWIFT Pointing Direction
**NEXT_POINT_RA:** **213.407d** {+14h 13m 38s} (J2000)
**NEXT_POINT_DEC:** **+70.472d** {+70d 28' 20"} (J2000)
NEXT_POINT_ROLL: 2.885d
**SLEW_TIME:** **15420.00** SOD {04:17:00.00} UT
SLEW_DATE: 20433 TJD; 124 DOY; 24/05/03
OBS_TIME: 900.00 [sec] (=15.0 [min])
**TGT_NAME:** **RX J1413.6+7029**
TGT_NUM: 3111759, Seg_Num: 10
**MERIT:** **60.00**
INST_MODES: BAT=0=0x0 XRT=7=0x7 UVOT=12525=0x30ED
**SUN_POSTN:** 40.78d {+02h 43m 07s} +15.81d {+15d 48' 31"}
SUN_DIST: 93.68 [deg] Sun_angle= -11.5 [hr] (East of Sun)
**MOON_POSTN:** 338.61d {+22h 34m 27s} -12.48d {-12d 28' 49"}
MOON_DIST: 113.09 [deg]
MOON_ILLUM: 31 [%]
GAL_COORDS: 113.36, 45.10 [deg] galactic lon,lat of the pointing direction
ECL_COORDS: 143.56, 69.70 [deg] ecliptic lon,lat of the pointing direction
COMMENTS: SWIFT Slew Notice to a preplanned target.
COMMENTS: Note that **preplanned targets** are overridden by any new BAT Automated Target.
COMMENTS: Note that preplanned targets are overridden by any TOO Target if the TOO has a higher Merit Value.
COMMENTS: The **spacecraft longitude,latitude at Notice_time is 247.70,10.86** [deg].
COMMENTS: This Notice was ground-generated -- not flight-generated.

What that SWIFT notice means:

Swift Alert: Pointing towards RX J1413.6+7029

On Friday, May 3rd, 2024, at 04:16:31 UT, the Swift telescope is scheduled to point towards a preplanned target, RX J1413.6+7029. This celestial object is located at a Right Ascension of 213.407 degrees (or 14 hours, 13 minutes, and 38 seconds) and a Declination of +70.472 degrees (or +70 degrees, 28 minutes, and 20 seconds).

The telescope will begin its slew to this target location at 04:17:00.00 UT, which will take approximately 15 minutes to complete. Once in position, Swift will observe RX J1413.6+7029 for 900 seconds, or 15 minutes, using its Burst Alert Telescope (BAT), X-ray Telescope (XRT), and Ultraviolet/Optical Telescope (UVOT).

At the time of observation, the Sun will be at a position of 40.78 degrees (or 2 hours, 43 minutes, and 7 seconds) and +15.81 degrees (or +15 degrees, 48 minutes, and 31 seconds), with a Sun angle of -11.5 hours (or East of the Sun). The Moon will be at a position of 338.61 degrees (or 22 hours, 34 minutes, and 27 seconds) and -12.48 degrees (or -12 degrees, 28 minutes, and 49 seconds), with a Moon illumination of 31%.

It's worth noting that this observation is part of a preplanned target list, but it may be overridden by a new BAT Automated Target or a Target of Opportunity (TOO) with a higher merit value. Additionally, the spacecraft's longitude and latitude at the time of observation will be 247.70 degrees and 10.86 degrees, respectively.

# Ingest and Transform Easily with Delta Live Tables Pipelines

## The best way to do ETL on the Databricks Data Intelligence Platform

```
-- incrementally ingest
CREATE STREAMING TABLE raw_data
AS
SELECT *
FROM cloud_files ("/raw_data",
"json")


-- incrementally transform
CREATE MATERIALIZED VIEW clean_data
AS
SELECT timestamp, id, target
FROM LIVE.raw_data
```

**Accelerate ETL development**

Declare **SQL or Python** and DLT automatically orchestrates the DAG, handles retries, changing data

**Automatically manage your infrastructure**

Automates complex tedious activities like **recovery, auto-scaling, and performance optimization**

**Ensure high data quality**

Deliver reliable data with built-in **quality controls, testing, monitoring, and enforcement**

**Unify batch and streaming**

Get the simplicity of SQL with freshness of streaming with one **unified API**

# Delta Live Tables with serverless compute

## The simplest way to build data pipelines

| | |
|---|---|
| **Simplicity** | • Simple development<br>• Simple operations |
| **Performance** | • End-to-end incremental processing<br>• Parallelized ingestion |
| **Low TCO** | • Serverless metering<br>• Efficient data processing |

# Delta Live Tables

## Ingest: Streaming Table in SQL with read_kafka()

```sql
CREATE OR REPLACE STREAMING TABLE raw_space_events AS
  SELECT offset, timestamp, value::string as msg
   FROM STREAM read_kafka(
    bootstrapServers => 'kafka.gcn.nasa.gov:9092',
    subscribe => 'gcn.classic.text.SWIFT_POINTDIR',
    startingOffsets => 'earliest',

    -- params kafka.sasl.oauthbearer.client.id
    `kafka.sasl.mechanism` => 'OAUTHBEARER',
    `kafka.security.protocol` => 'SASL_SSL',
    `kafka.sasl.oauthbearer.token.endpoint.url` => 'https://auth.gcn.nasa.gov/oauth2/token',
    `kafka.sasl.login.callback.handler.class` => 'kafkashaded.org.apache.kafka.common.security.oauthbearer.secured.
    OAuthBearerLoginCallbackHandler',


    `kafka.sasl.jaas.config` =>
        '
         kafkashaded.org.apache.kafka.common.security.oauthbearer.OAuthBearerLoginModule required
         clientId="7u2rpivvxxxxxxxxxxxxxxxxxx"
         clientSecret="1errfm2jdgl0uolkb78kjnf8v94eyyyyyyyyyyyyyyy" ;
        '
  );
```

# Delta Live Tables

## Transformation: Materialized View using PIVOT and type casts

```sql
CREATE OR REPLACE MATERIALIZED VIEW split_events
COMMENT "Split Swift event message into individual rows"
AS
  WITH extracted_key_values AS (
    SELECT
      timestamp,
      split_part(line, ':', 1) AS key,
      TRIM(SUBSTRING(line, INSTR(line, ':') + 1)) AS value
    FROM (
      SELECT
        timestamp,
        explode(split(msg, '\\n')) AS line
      FROM (LIVE.raw_space_events)
    )
    WHERE line != ''
  ),
  pivot_table AS (
    SELECT *
    FROM (
      SELECT key, value, timestamp
      FROM extracted_key_values
    )
    PIVOT (
      MAX(value) FOR key IN ('TITLE', 'NOTICE_DATE', 'NOTICE_TYPE', 'NEXT_POINT_RA', 'NEXT_POINT_DEC',
      'NEXT_POINT_ROLL', 'SLEW_TIME', 'SLEW_DATE', 'OBS_TIME', 'TGT_NAME', 'TGT_NUM', 'MERIT',
      'INST_MODES', 'SUN_POSTN', 'SUN_DIST', 'MOON_POSTN', 'MOON_DIST', 'MOON_ILLUM', 'GAL_COORDS',
      'ECL_COORDS', 'COMMENTS')
    )
  )
  SELECT timestamp, TITLE, CAST(NOTICE_DATE AS TIMESTAMP) AS NOTICE_DATE, NOTICE_TYPE, NEXT_POINT_RA,
  NEXT_POINT_DEC, NEXT_POINT_ROLL, SLEW_TIME, SLEW_DATE, OBS_TIME, TGT_NAME, TGT_NUM, CAST(MERIT AS
  DECIMAL) AS MERIT, INST_MODES, SUN_POSTN, SUN_DIST, MOON_POSTN, MOON_DIST, MOON_ILLUM, GAL_COORDS,
  ECL_COORDS, COMMENTS
  FROM pivot_table
```

# Demo Swift DLT Pipeline with Data Intelligence

# Data Rooms

# AI/BI Genie

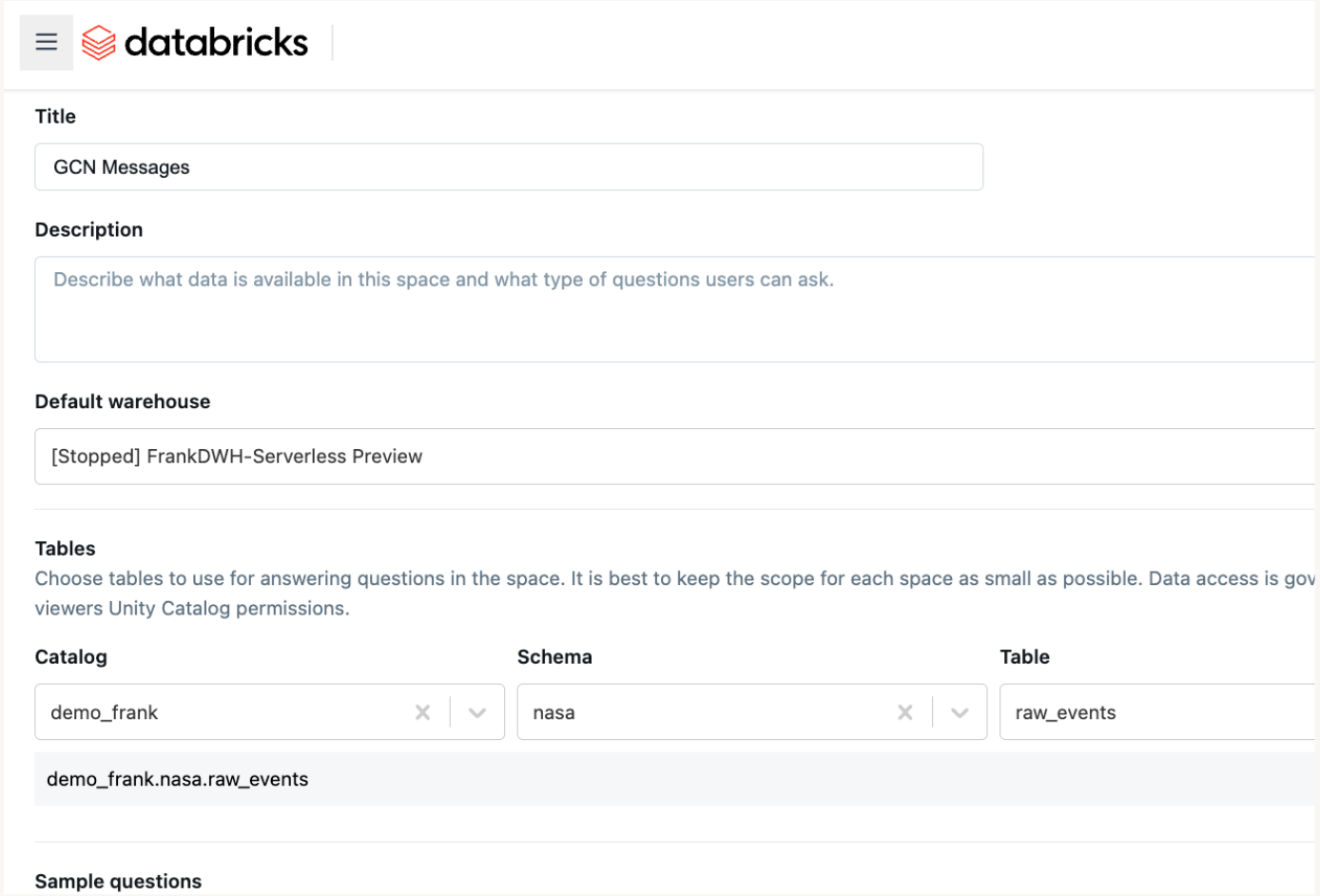
## Enable business users to interact with data with LLM-powered Q&A

Natural language ->
answers in text and visualizations

Curate dataset-specific experiences with custom instructions

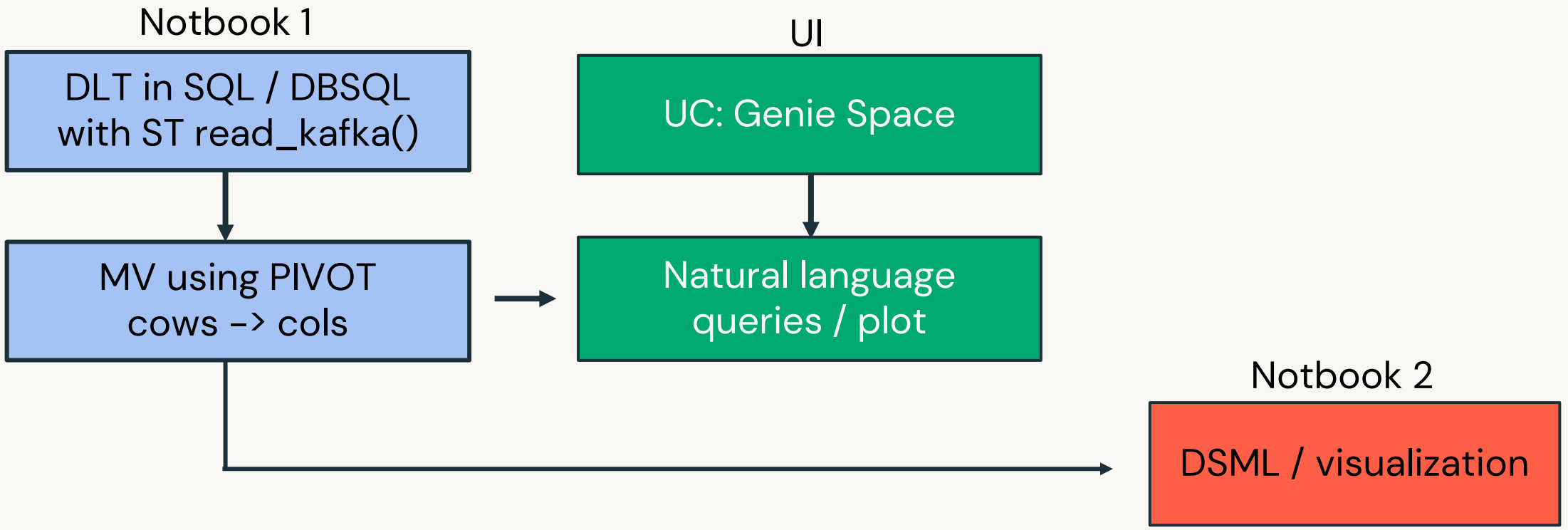
Powered by Databricks SQL & DatabricksIQ

Works with DLT Streaming Tables and Materialized Views



# Demo Genie

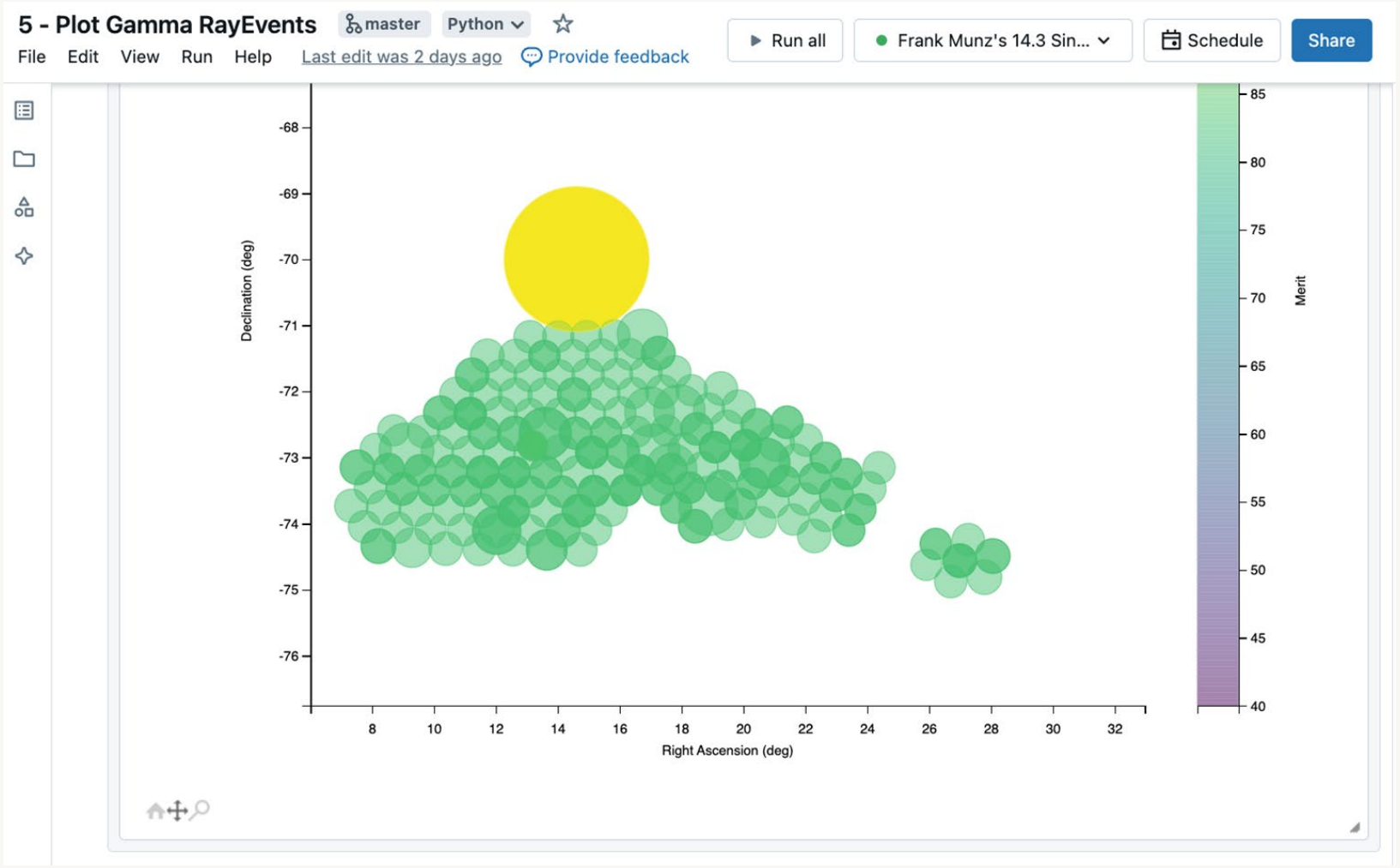
# SWIFT Analytics - Back of an envelope architecture



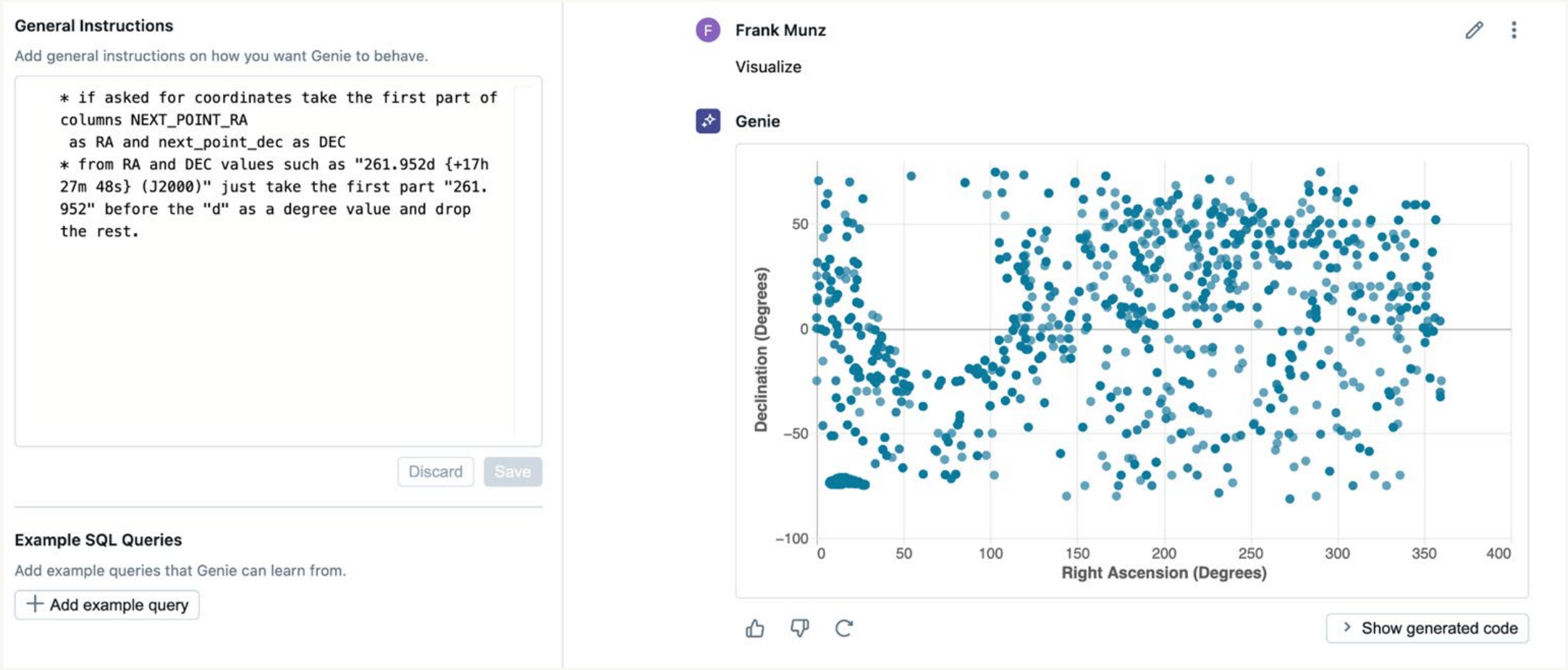
# Genie or Databricks Assistant?

| Databricks Assistant | Genie |
| --- | --- |
| **Technical User** | **Business User** |
| Developer with SQL / Python | No programming |
| Tabular data | Tabular data |
| Technical or data tasks<br>● Fix this Python code<br>● document this table<br>● write me a SQL query | Answer business questions such as<br>● Who were my fastest growing customers last quarter?<br>● Explain me this data set |

# Scientific Notebook Visualization



# Genie Visualization



# RAG with DBRX / LLama3

# Compound AI chat bot based on 36,000 NASA Circulars

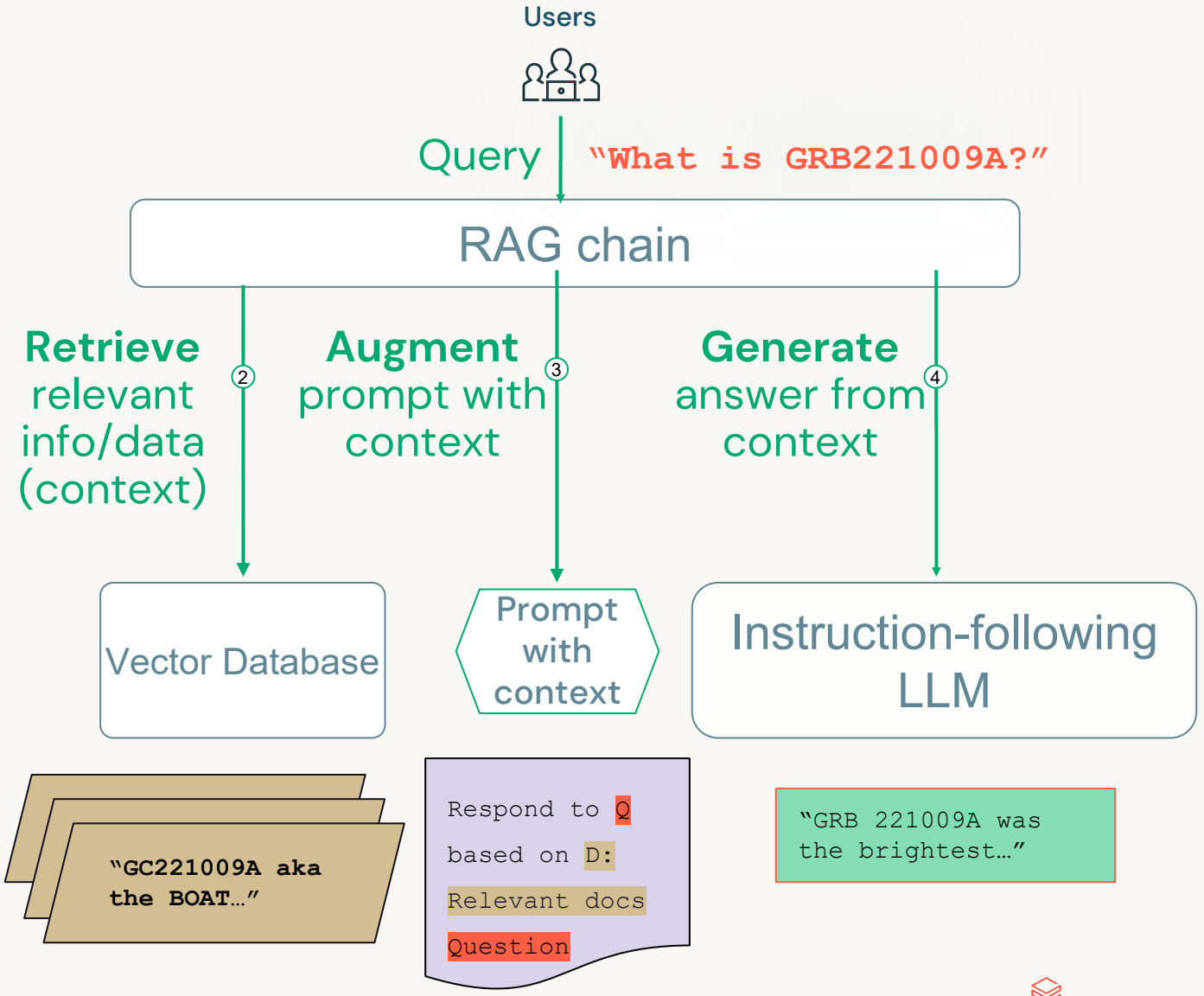
# Retrieval Augmented Generation (RAG)

RAG uses LLMs as *reasoning engines*, rather than as static models.

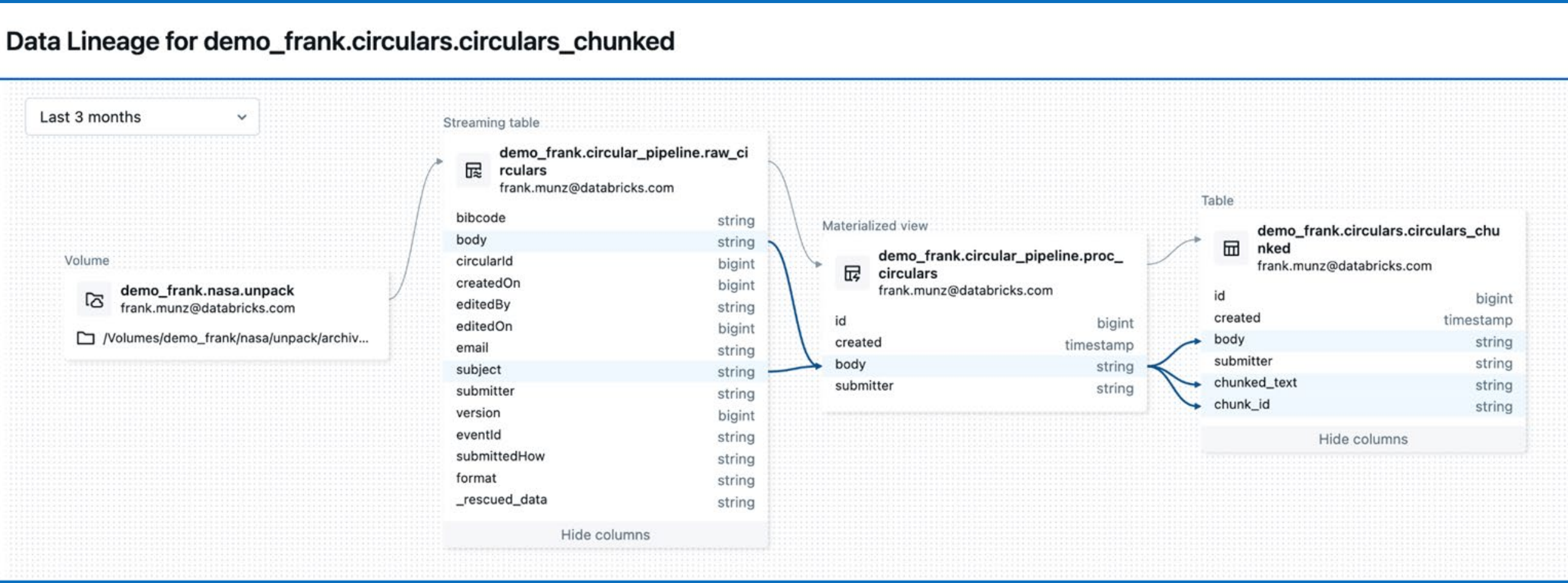
***Your data***

***+***

***an LLM "brain"***

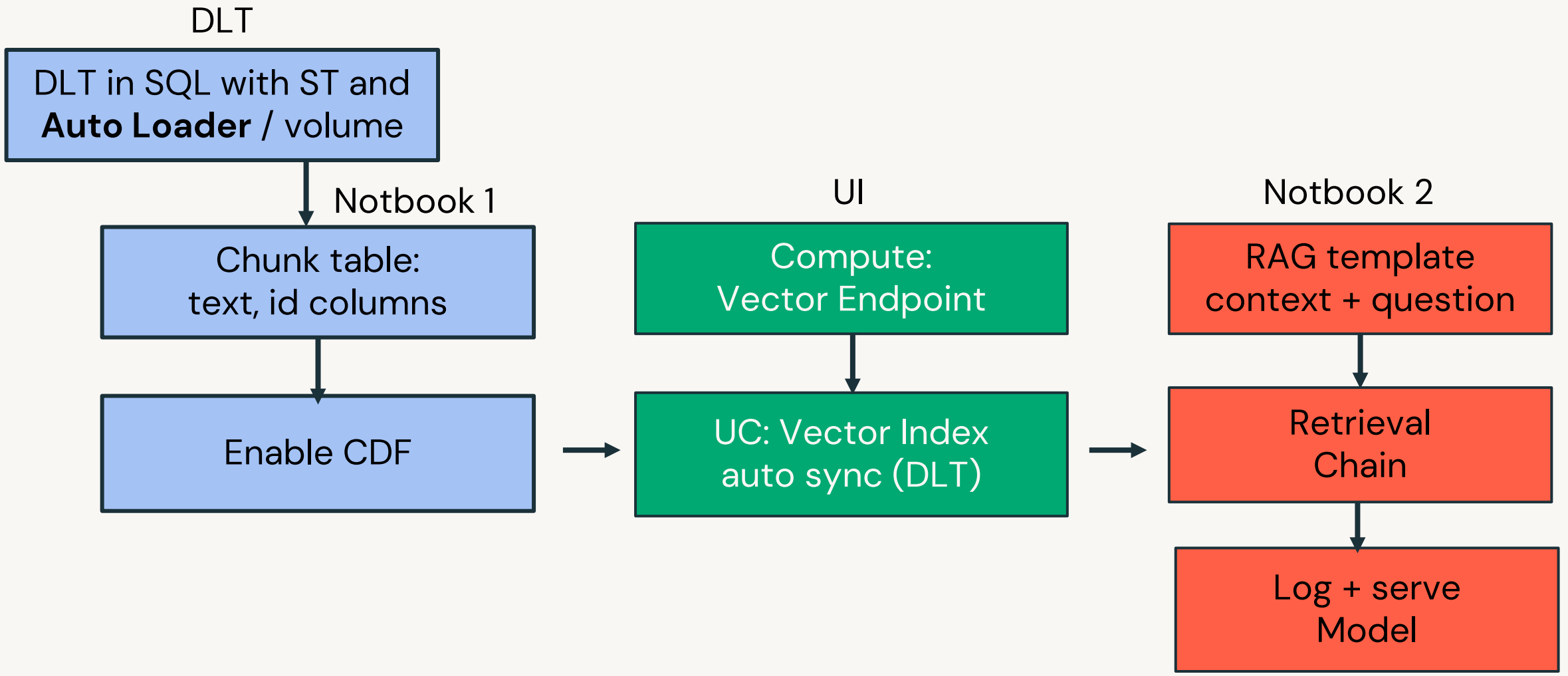


# Unity Catalog Lineage



# Demo RAG + LLM

# Circulars RAG – Back of envelope architecture



# There is no good model without good data

# Summary and Conclusion

# Conclusion

- You are just one copy and paste of a SQL command away from exploring streaming data from a NASA satellite.

- Simply enable Genie on any UC table,
  E.g. DLT Streaming Tables or Materialized Views

- Ask Genie natural language questions and create plots
  - Genie writes SQL for you
  - Add your own instructions (2 instructions made notebook obsolete)
  - Instructions work with functions

# Conclusion

- RAG adds (context based text) data context to an LLM query
  - The template matters a lot -> prompt engineering
  - Fresher data
  - Less hallucinations
- Use Data Intelligence:
  Assistant & DBRX and other LLMs for coding support!
- Explore the new RAG Framework and tooling
- TLDR: It's all about the platform

# THANK YOU!

Judith Rascusin (NASA)
Alex, Nicolas, Raghu, Praveen, Neil, Eric (Databricks)

# & all of YOU!