

# How To Query A Stream?

A pixelated illustration of a mountain valley. In the foreground, a stream flows through a lush green valley. A person wearing a white coat and a hat stands on a path to the right, looking towards the stream. In the background, there are large, brown mountains under a blue sky with white clouds. A small house is visible on a hillside to the right.

Viktor Gamov, Confluent  
@gamussa

Dallas/Allen, TX, 2024





**Robert Zych** ✓

@zychr



Had a great Q&A with John Roesler yesterday about IQ (ie Kafka streams interactive queries) v2 will it much easier to perform more complex queries. My question was basically when should use we use IQ vs something like [@ApachePinot](#) ?

7:32 AM · Jun 25, 2022



**Jase**

@jasonbelldata



It does.... Or a podcast. Or a blog.

8:47 PM · Jun 25, 2022



# Viktor

## GAMOV

---

Principal Developer Advocate | Confluent

Twitter X: @gamussa





# Monolith





# ETL and CDC





A stylized illustration of a person in a red robe using a smartphone in a medieval street. The scene is set in a narrow, cobblestone street with stone buildings and arches. A bright light source, possibly a window or a lantern, is visible in the distance, casting a warm glow. The person in the foreground is seen from the back, holding a smartphone. The overall style is reminiscent of a digital painting or a high-quality illustration.

# Mobile Era





# Data Pipelines and Microservices



A pixelated illustration of a library. In the center, the word "LOG" is written in large, white, sans-serif capital letters. To the left, a long, yellow scroll with black text is unrolled across the floor. The background features tall wooden bookshelves filled with books, a globe on a stand, and a person in a blue coat standing near a desk. The floor is covered with papers and books, suggesting a place of study or research.

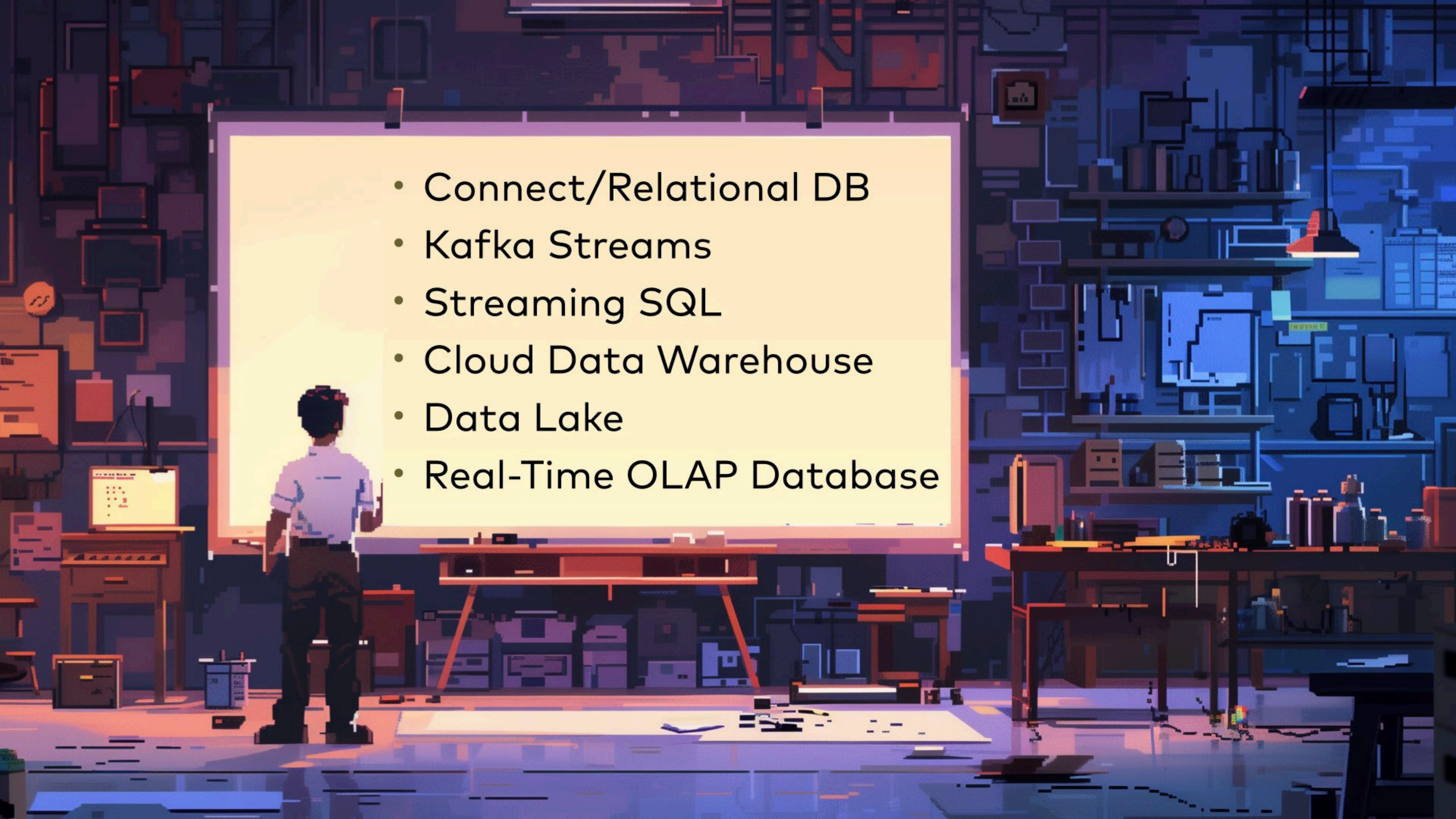
LOG





# OLTP stream vs OLAP streams



- 
- Connect/Relational DB
  - Kafka Streams
  - Streaming SQL
  - Cloud Data Warehouse
  - Data Lake
  - Real-Time OLAP Database



# Kafka Connect



# Connect/RDBMS





# Connect/RDBMS

- Suitable for smaller data
- Transactional
- Familiar to users







# Kafka Streams



# Kafka Streams

(transactional)

- Ingests directly from a **topic**
- KTable
- Forms an **in-memory key/value** store suitable for querying by topic key
- Scalable across **members** of a **consumer group**
- Readable through **Interactive Queries**





# Kafka Streams

(transactional)

```
final KStream<String, String> stream =  
    builder.stream(inputTopic,  
        Consumed.with(stringSerde, stringSerde));
```

```
final KTable<String, String> convertedTable =  
    stream.toTable(Materialized.as("stream-  
        converted-to-table"));
```



# Kafka Streams

(analytical)

- Full-featured **Java stream processing API**
- Arbitrary streaming computation
- Can emit new streams (not this talk)
- KTables **queryable by key**
- Every read pattern requires its own topology
- **Interactive Queries** again





# Kafka Streams

## (analytical)

```
KTable<String, Long> wordCounts = textLines
    .flatMapValues(textLine ->

Arrays.asList(textLine.toLowerCase().split("\\W+")))
    .groupBy((key, word) -> word)
    .count(Materialized.<String, Long, KeyValueStore<Bytes,
byte[]>>as("counts-store"));

wordCounts.toStream().to("WordsWithCountsTopic",
    Produced.with(Serdes.String(),
Serdes.Long()));
```



# Streaming SQLs





# Streaming SQL

- Materialize
- DeltaStream
- RisingWave
- ksqlDB





I love you, little squirrel.





# Materialize

- Replacement data warehouse
- Integrates with Kafka, Postgres, dbt
- The **Materialized View** is the central abstraction
- Views are **persistent and queryable**
- Postgres wire-compatible
- Positioned as an analytics solution





# Delta Stream

- Cloud-native **streaming SQL**
- **Serverless, BYOC**
- Kafka, Kinesis integration
- **Materialized views** and streaming pipelines
- streaming database and streaming analytics





# Rising Wave

- Distributed SQL Streaming database
- Cloud and OSS versions
- Implementation of **Flink in Rust**
- **Kafka, Pulsar, Kinesis** integrations
- Flink+persistent views
- Postgres wire-compatible



# ksqlDB

- «*Streaming Database*»
- Provides persistent TABLE abstraction
- Pull and Push queries
- Like **KafkaStreams**, but in **SQL**





# Cloud Data Warehouses



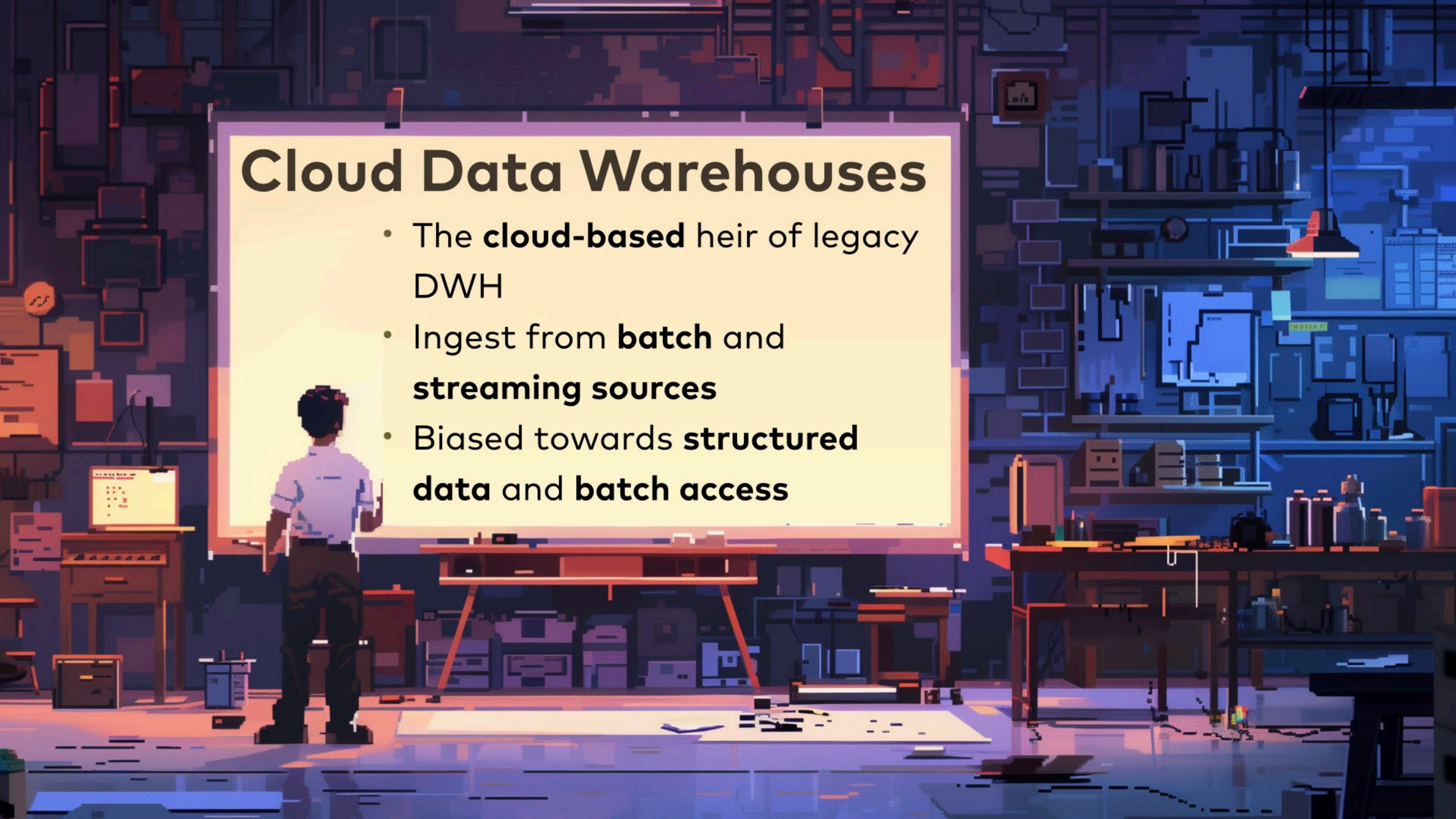
# Cloud Data Warehouses





# Cloud Data Warehouses

- The **cloud-based** heir of legacy DWH
- Ingest from **batch** and **streaming sources**
- Biased towards **structured data** and **batch access**

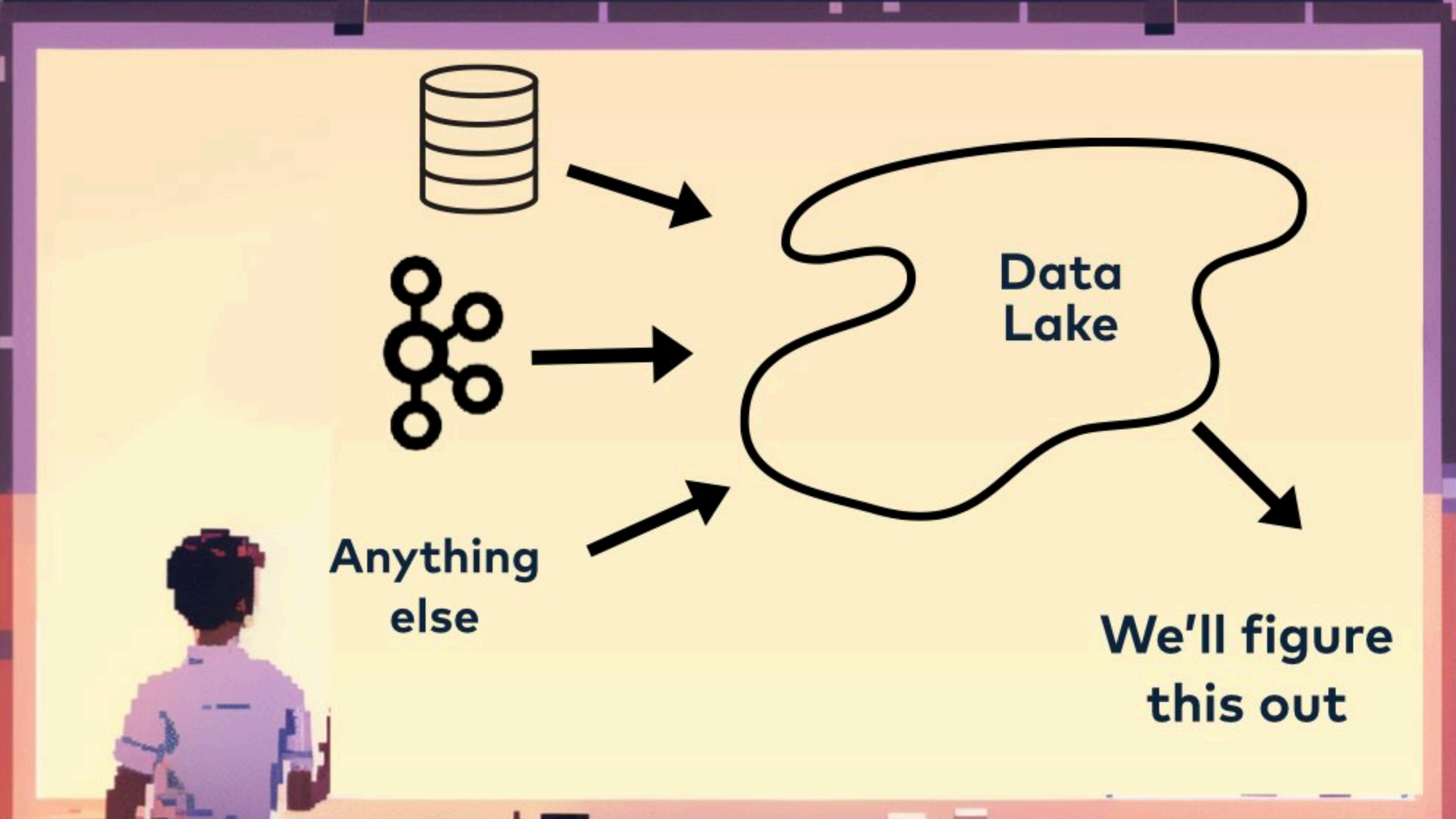




# Data Lake









# Data Lakes

- Started as the HDFS cluster
- Became S3
- That didn't help...
- ELT vs. ETL
- Iceberg/Hudi/DeltaLake





# Data Lakes

- Storage and compute are radically decoupled
- Structure is relatively less important
- Reads are slow
- Streaming is historically difficult





# Data Lakes

- Storage and compute are radically decoupled
- Structure is relatively less important
- Reads are slow
- Streaming is historically difficult







# Real-Time Analytics Database



# Real-Time OLAP

- Designed for **high concurrency**, **low latency** queries
- Ingests from **streaming** and **batch** sources
- Intimate **integration with Kafka**
- Conventional **tables and SQL**





# Real-Time OLAP

- Analytics shaped like **real-time data**
- Analytics when **users are decision makers**







**No Solutions  
only Trade Offs**





Sometimes you go  
with what you know



A vibrant, stylized illustration of a man with dark hair and a mustache, wearing a yellow shirt and blue pants, sitting in a large, patterned armchair. He is smiling and reading a book with a sun-like symbol on the cover. The setting is a library with tall bookshelves filled with books. A large window in the background shows a bright, stylized sun with rays and a face. A large green plant is visible on the left side of the frame.

**This is not bad!**



A detailed illustration of a steampunk laboratory. In the center, a scientist with a large, spiky white afro hairstyle is seated at a wooden desk, looking down at papers. The desk is cluttered with various tools, a microscope, and several glass bottles. Behind the scientist, a large, complex mechanical apparatus with a prominent circular dial and numerous gears is visible. The background is filled with more machinery, including large gears and several human skulls mounted on a wall. To the left, a computer monitor displays a glowing blue interface with a circular diagram. The overall lighting is a mix of warm yellow and cool blue tones, creating a futuristic yet industrial atmosphere.

# Performance



# Community





# Differentiated Application Code

The background is a complex illustration of a fantastical world. It features several dragons of various colors (green, brown, blue) and sizes, some breathing fire or water. A large sailing ship is visible in the upper center. In the bottom right corner, there's a depiction of a city or industrial area with buildings and a vehicle. The entire scene is overlaid with a grid pattern, suggesting a map or a technical drawing. Three semi-transparent blue text boxes are placed over the illustration, containing the text 'Differentiated Application Code', 'Area of Exploration', and 'Kafka'.

Area of  
Exploration

Kafka