

A pixel art illustration of a mountain valley. In the foreground, a stream flows through a lush green valley, surrounded by rocks and tall grass. A person in a light blue coat stands on a path to the right, looking towards the stream. In the background, there are steep, rocky mountains under a blue sky with white clouds. A small house is visible on a cliffside to the right.

One Does Not Simply Query a Stream!

Viktor Gamov, Confluent
@gamussa

Krakow, Poland 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 we use IQ vs something like [@ApachePinot](#) ?

7:32 AM · Jun 25, 2022

[@gamussa](#) | [@confluentinc](#) | [@apacheflink](#)



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

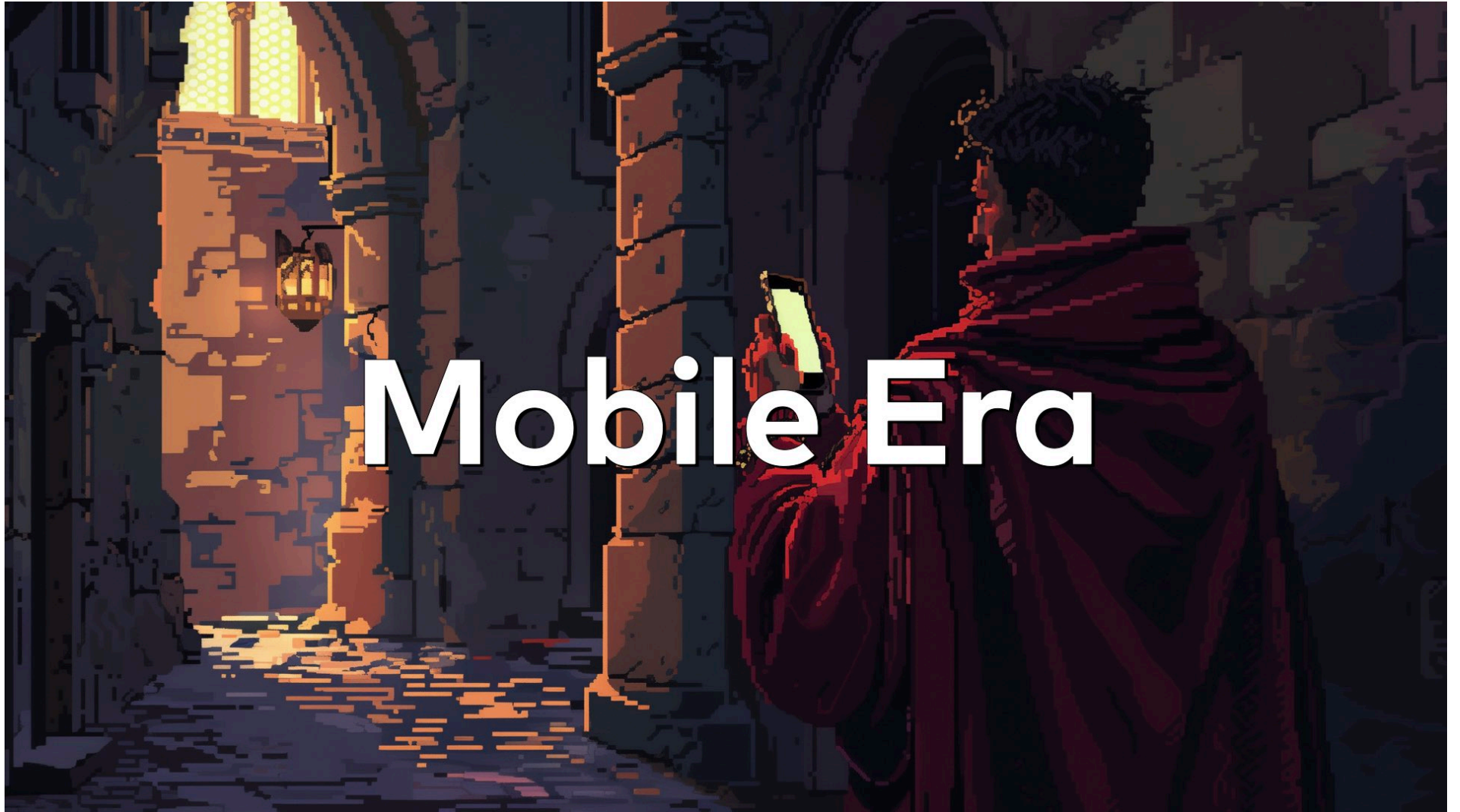




Monolith

ETL and CDC





Mobile Era




Data Pipelines and Microservices



LOG



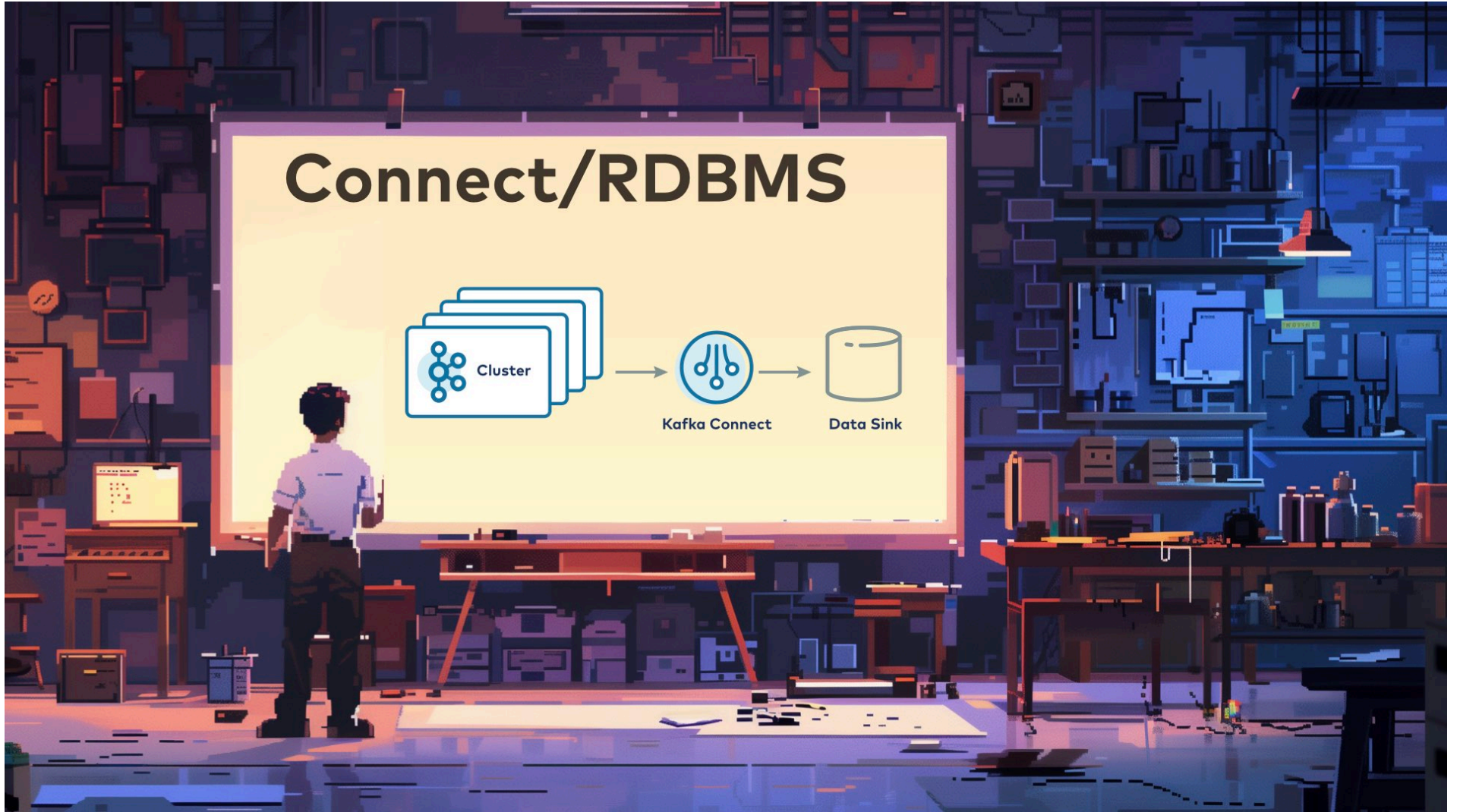
OLTP stream vs OLAP streams

- 
- A person in a white shirt and dark pants stands with their back to the camera, pointing at a large projection screen. The screen displays a bulleted list of data technologies. The background is a complex, futuristic control room with various monitors, shelves, and equipment, illuminated with blue and purple light.
- Connect/Relational DB
 - Kafka Streams
 - Streaming SQL
 - Cloud Data Warehouse
 - Data Lake
 - Real-Time OLAP Database



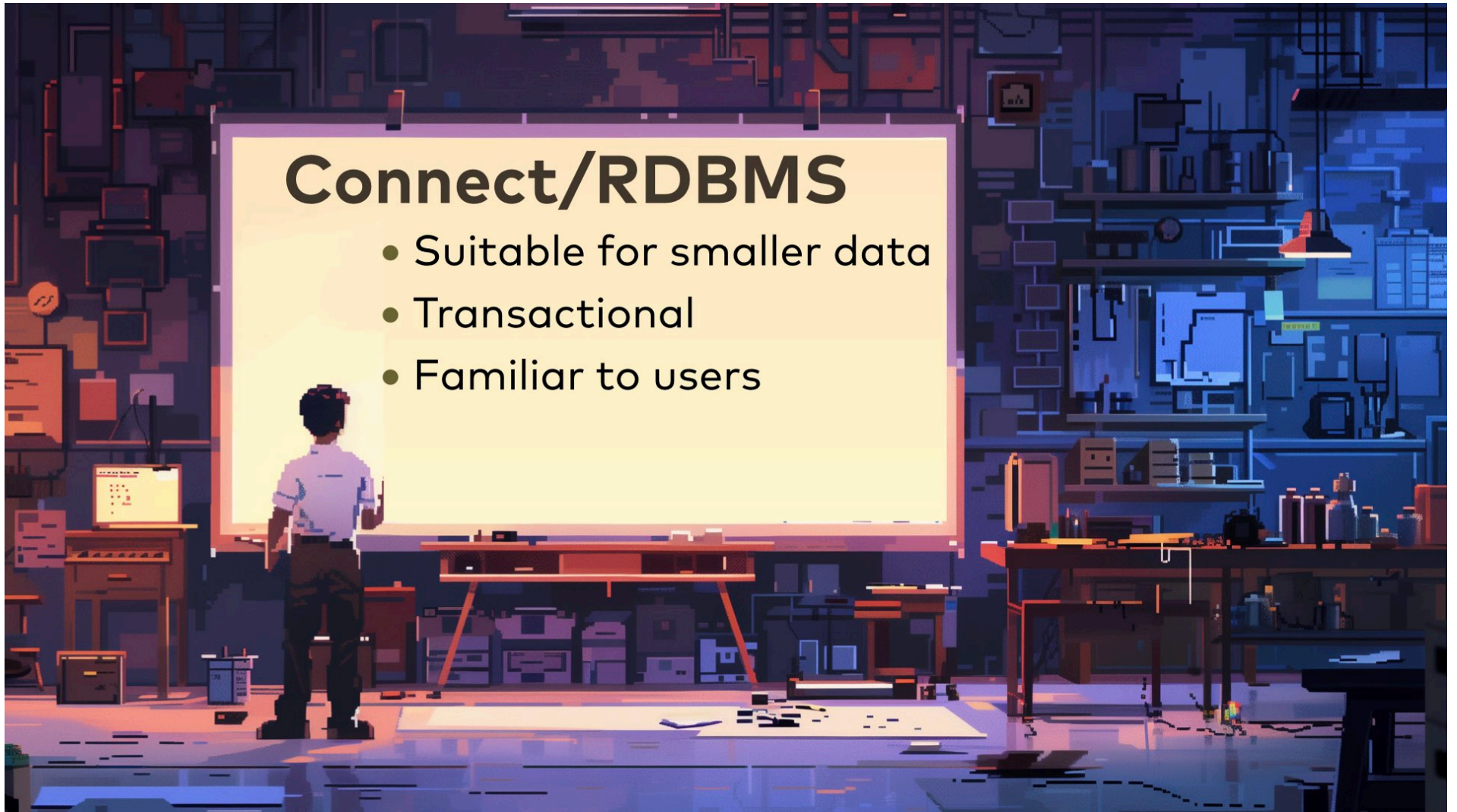
Kafka Connect

Connect/RDBMS



Connect/RDBMS

- Suitable for smaller data
- Transactional
- Familiar to users



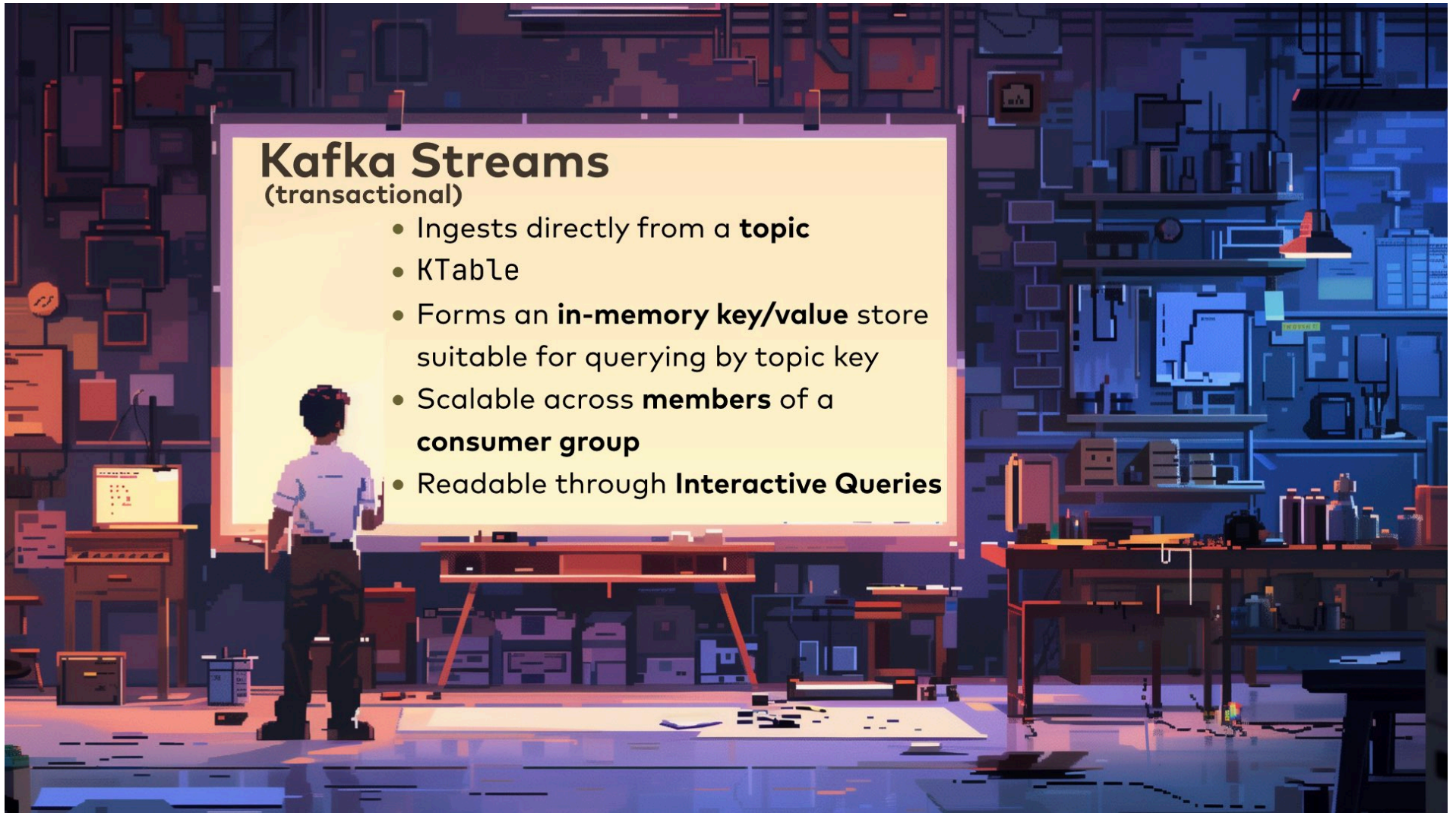


Kafka Streams

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**

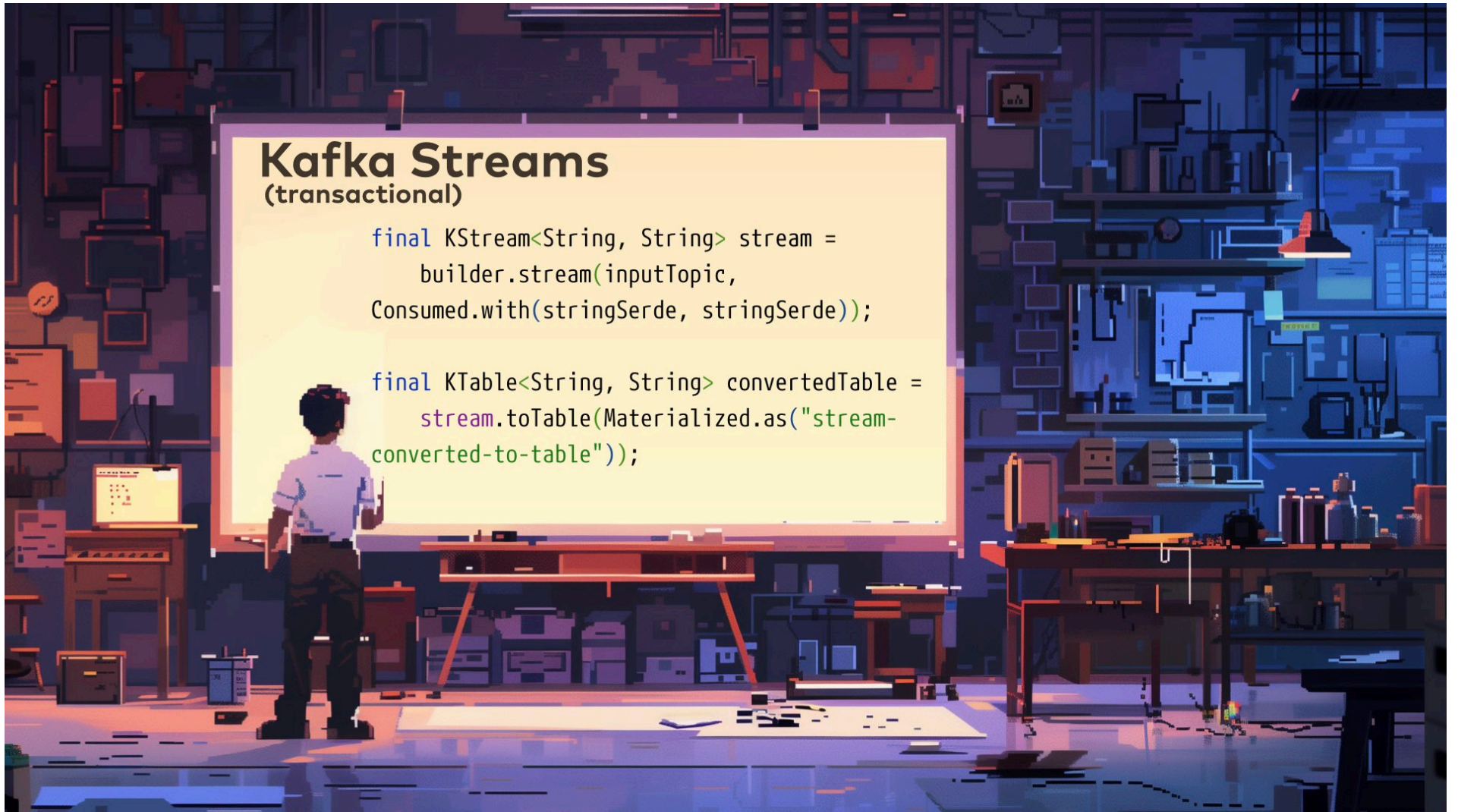


Kafka Streams

(transactional)

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

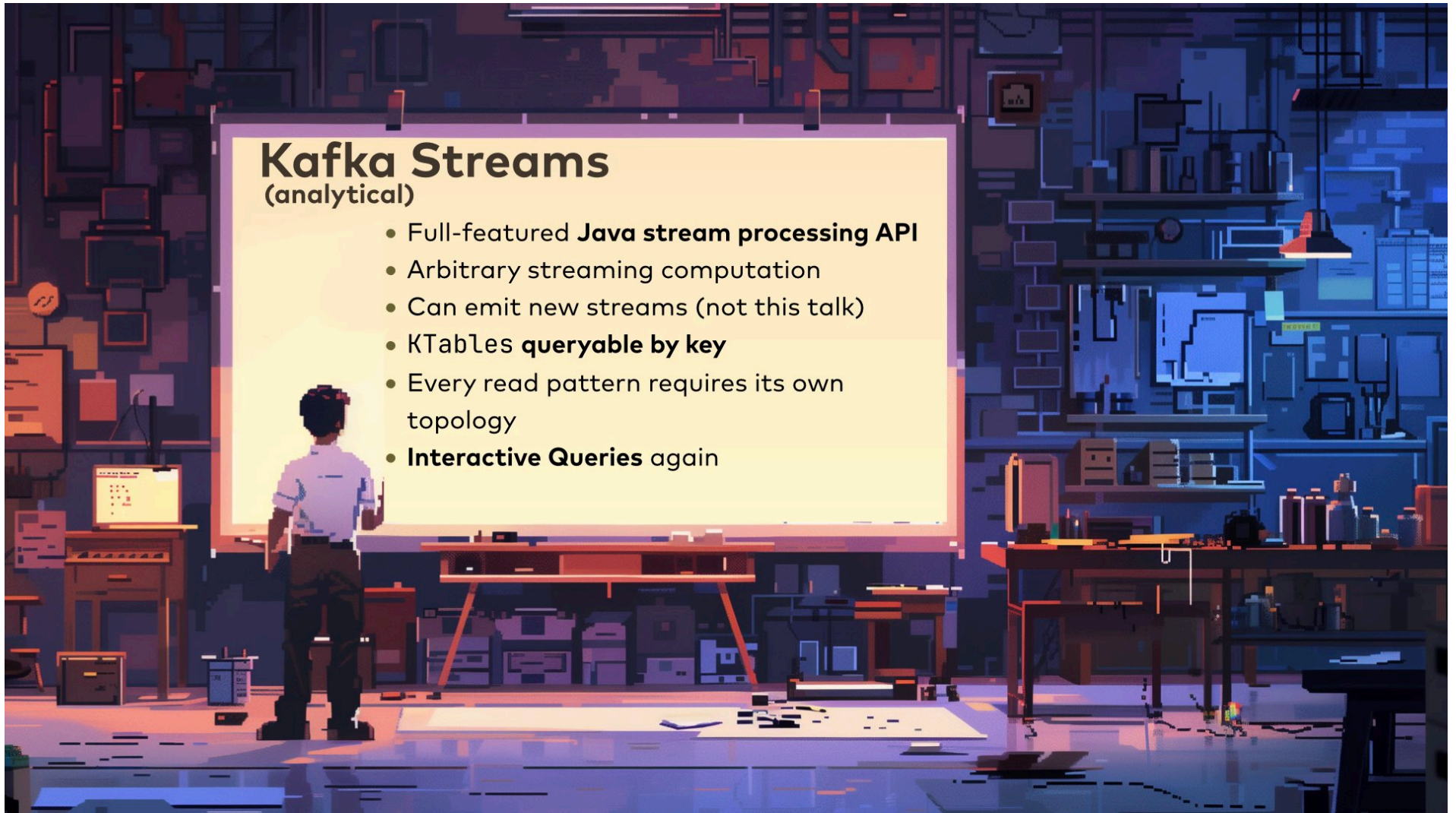
```
final KTable<String, String> convertedTable =  
    stream.toTable(Materialized.as("stream-  
converted-to-table"));
```



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



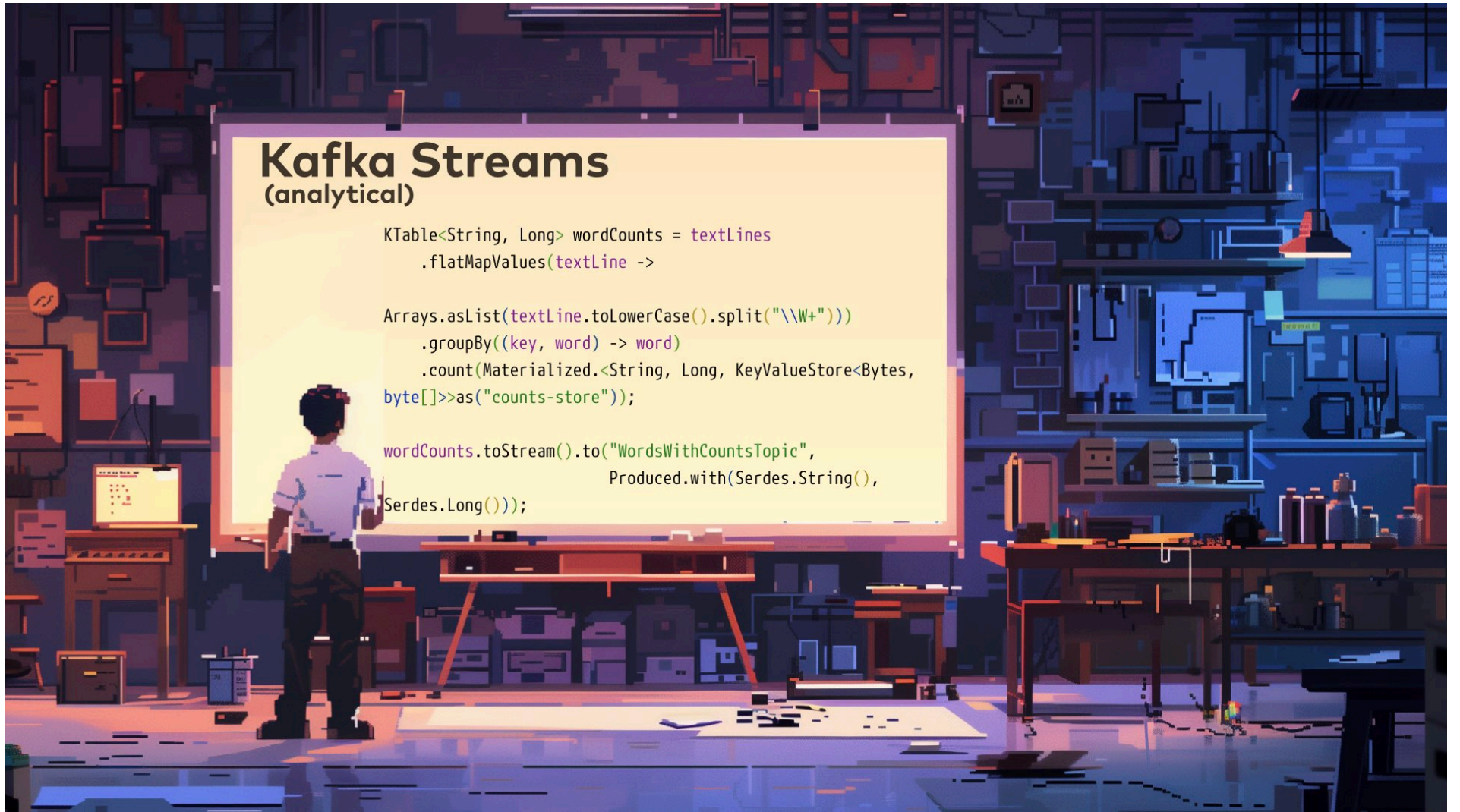
Kafka Streams

(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()));
```

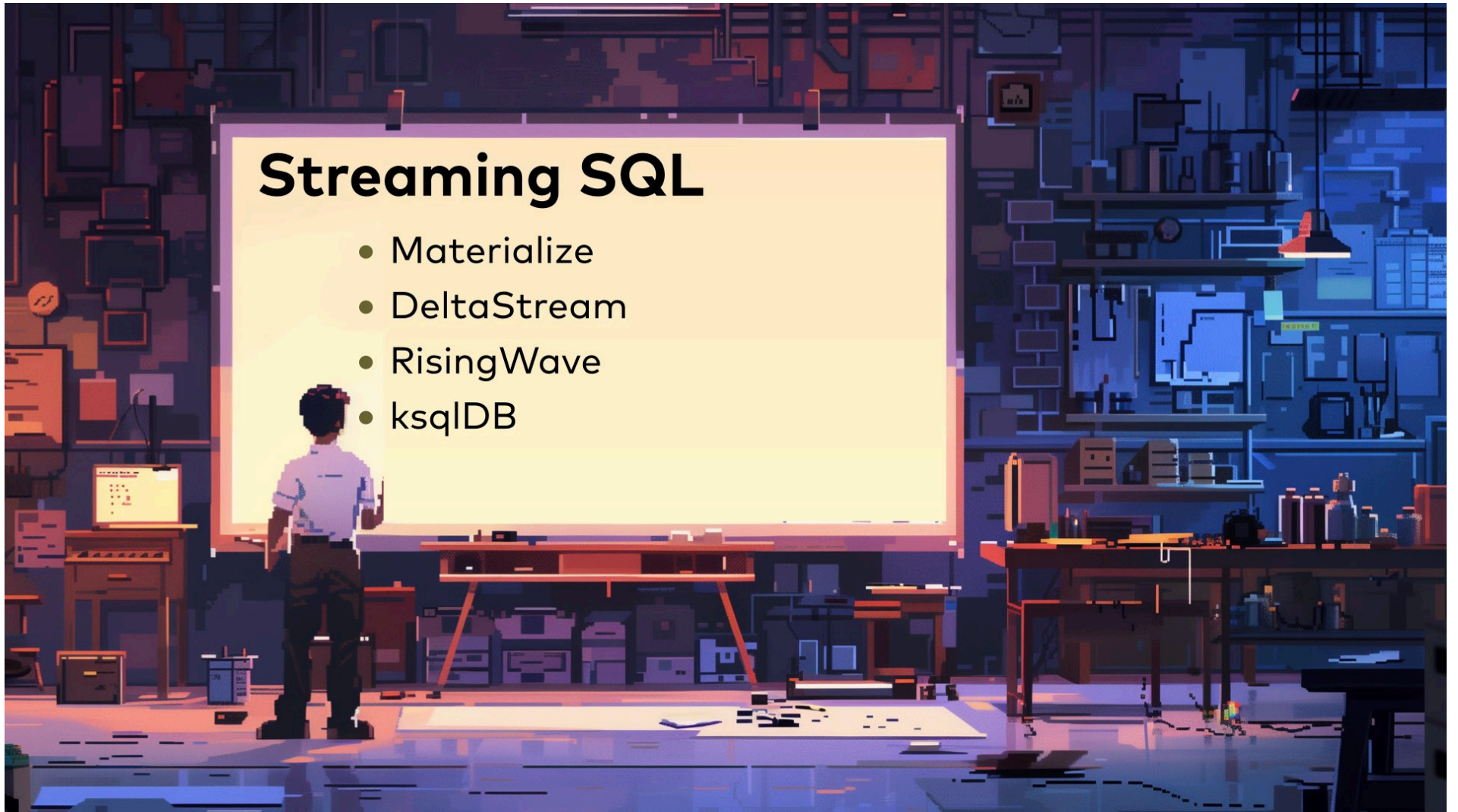


Streaming SQLs



Streaming SQL

- Materialize
- DeltaStream
- RisingWave
- ksqlDB

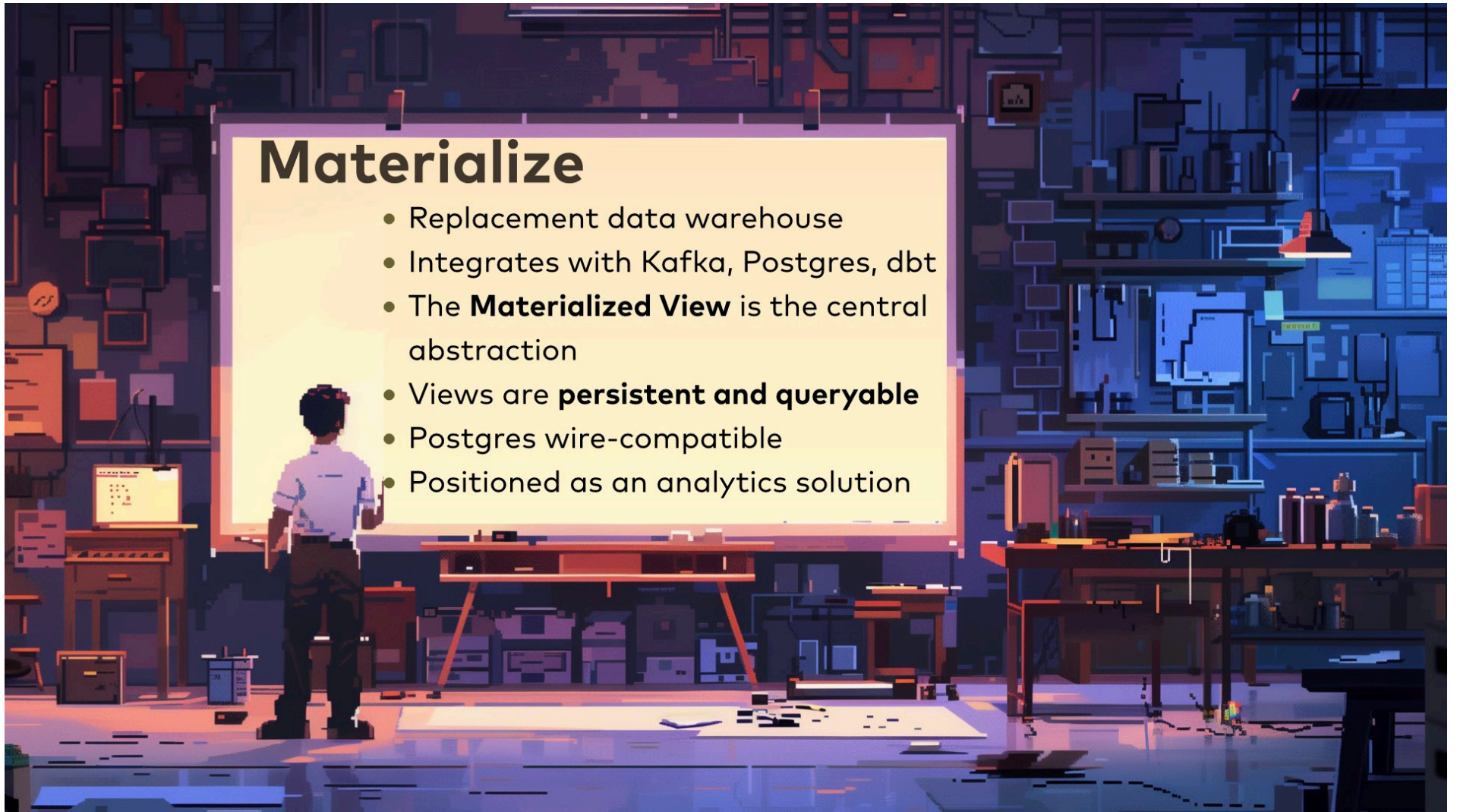






Materialize

- 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



A stylized illustration of a person standing in a server room, presenting a slide titled 'Delta Stream'. The person is seen from behind, wearing a white shirt and dark pants. The room is filled with server racks, desks, and various pieces of equipment, all rendered in a dark, moody color palette with blue and purple tones. The slide is the central focus, displaying the title and a list of features.

Delta Stream

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

A stylized illustration of a person standing in a workshop or office, presenting a slide titled "Rising Wave". The scene is dimly lit with blue and orange tones. The person is seen from behind, wearing a white shirt and dark pants. The slide is the central focus, displaying a list of features for Rising Wave. The background is filled with various pieces of equipment, shelves, and a desk with a computer monitor.

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

A stylized illustration of a person in a white shirt and dark pants standing in a server room, presenting to a large screen. The room is filled with server racks, desks, and various pieces of equipment, illuminated by warm and cool lights. The screen displays the title 'ksqlDB' and a list of features.

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

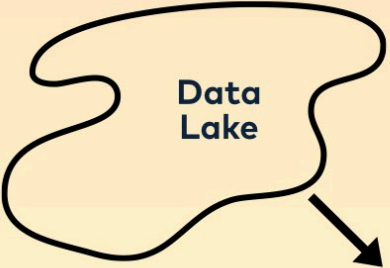
- The **cloud-based** heir of legacy DWH
- Ingest from **batch** and **streaming sources**
- Biased towards **structured data** and **batch access**

Data Lake





Anything
else



Data
Lake

We'll figure
this out

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



Real-Time Analytics Database

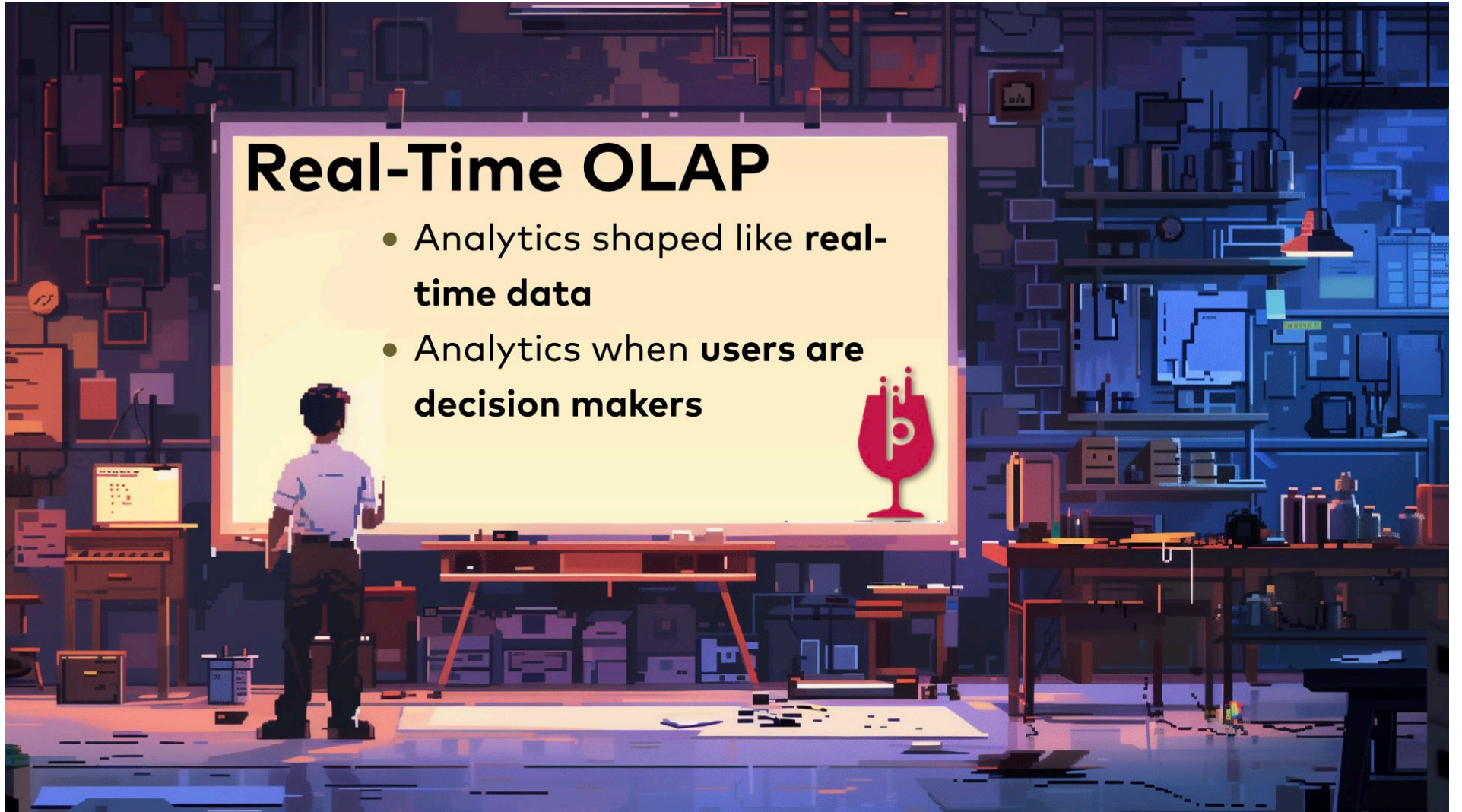


Real-Time OLAP

- 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

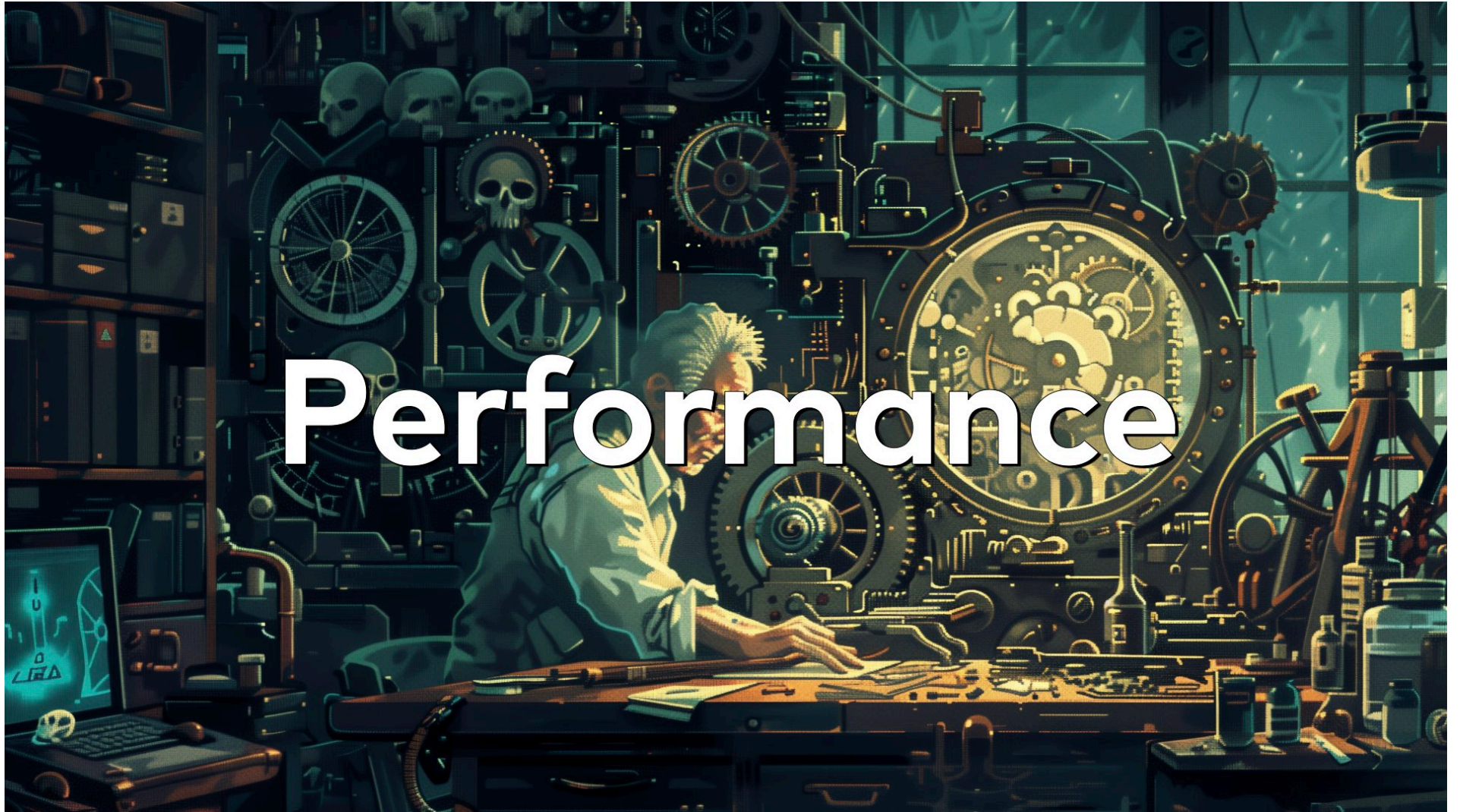


Sometimes you go
with what you know



This is not bad!

Performance



Community



The background is a complex, hand-drawn illustration on a grid. It depicts a fantastical world with various creatures and elements. In the foreground, there are several dragons: a large green dragon with a red eye on the right, a brown dragon with a long neck on the left, and a blue dragon at the top. There are also sea monsters, including a large green sea serpent and a blue sea monster. A sailing ship is visible in the background. The overall style is reminiscent of a fantasy map or a detailed illustration for a game or book. The text is overlaid on blue semi-transparent banners.

Differentiated Application Code

Area of
Exploration

Kafka