
How To Query A Stream?
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

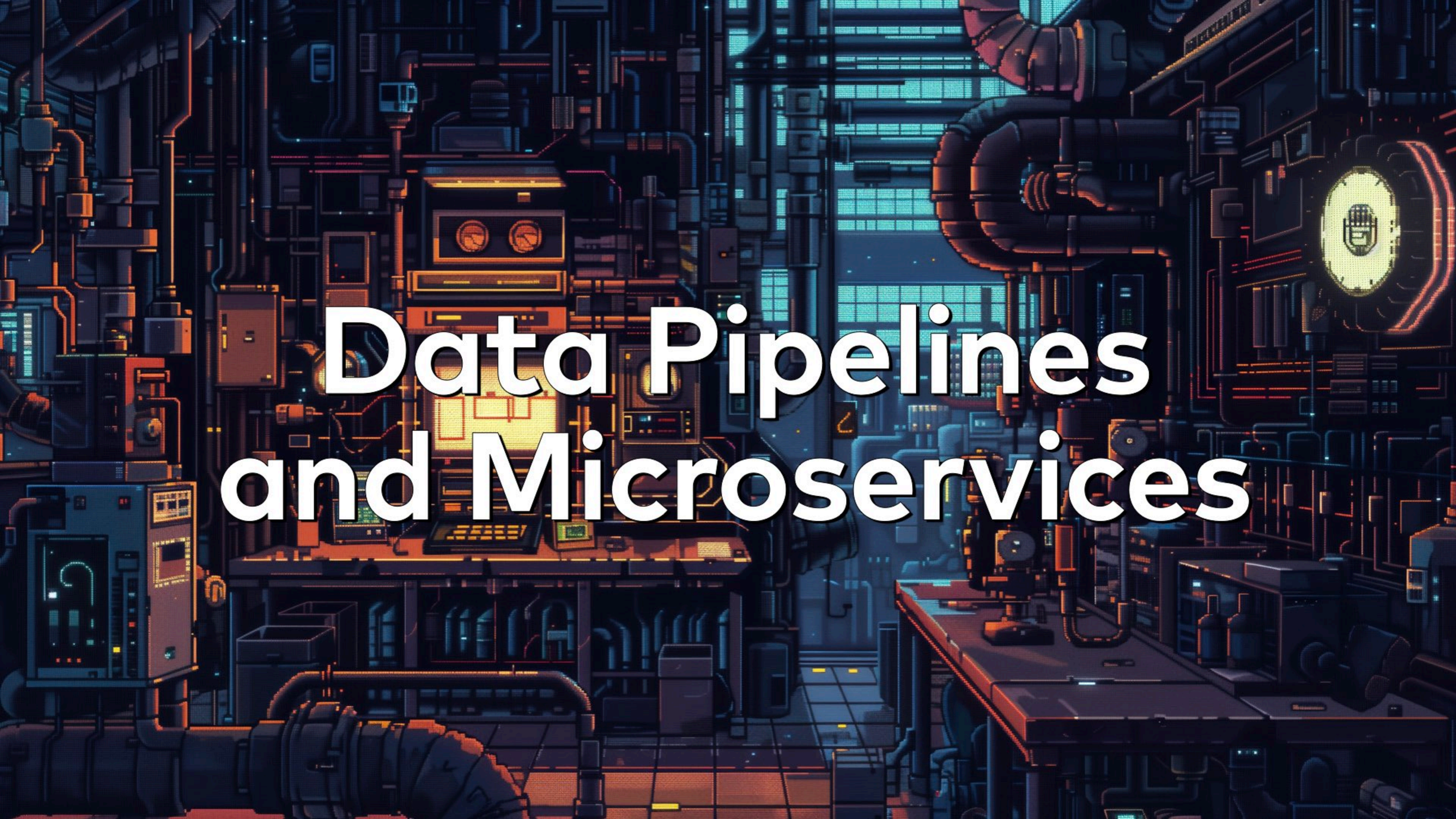
Principal Developer Advocate | Confluent

~~Twitter~~ X: @gamussa

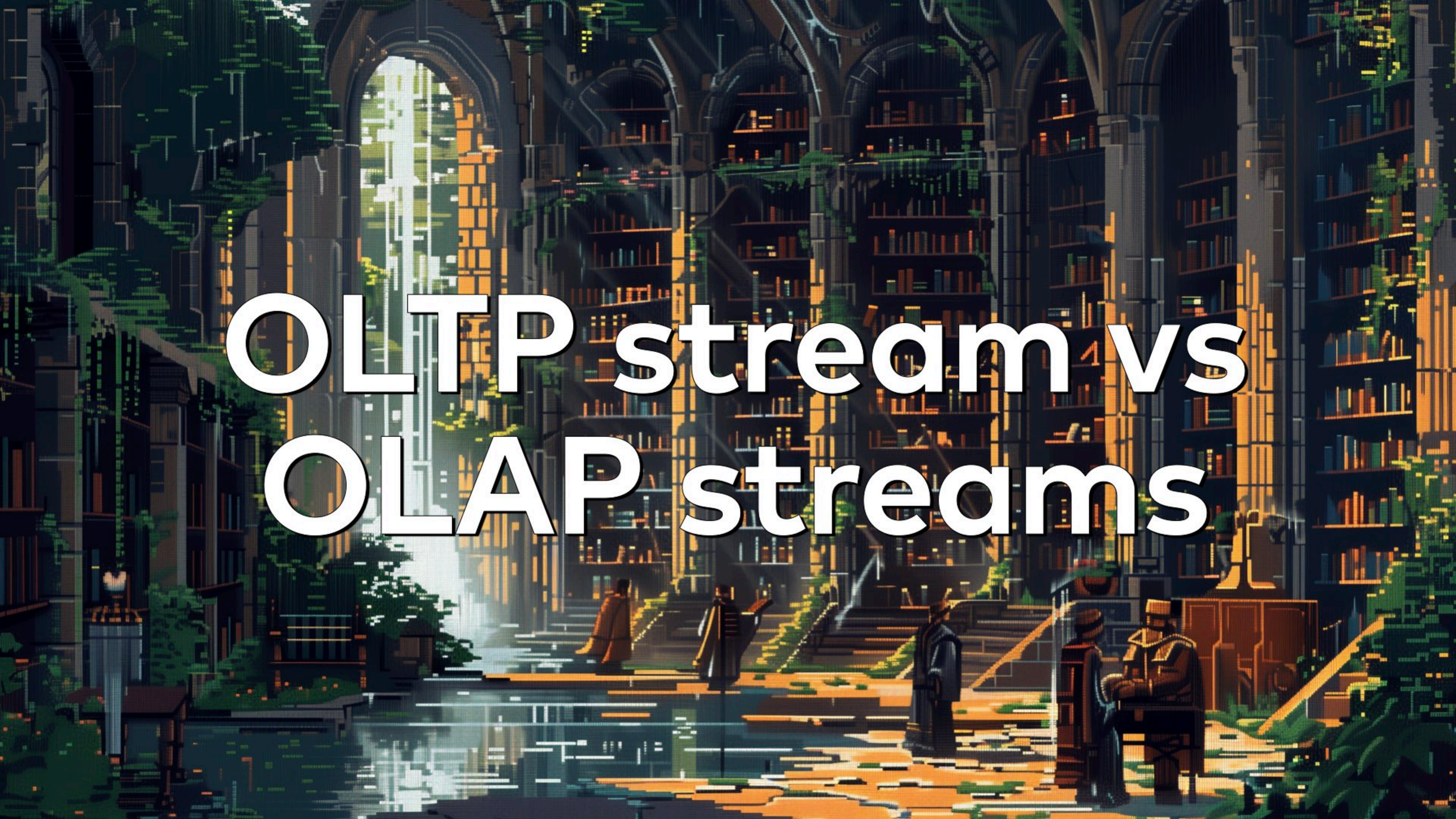

Monolith

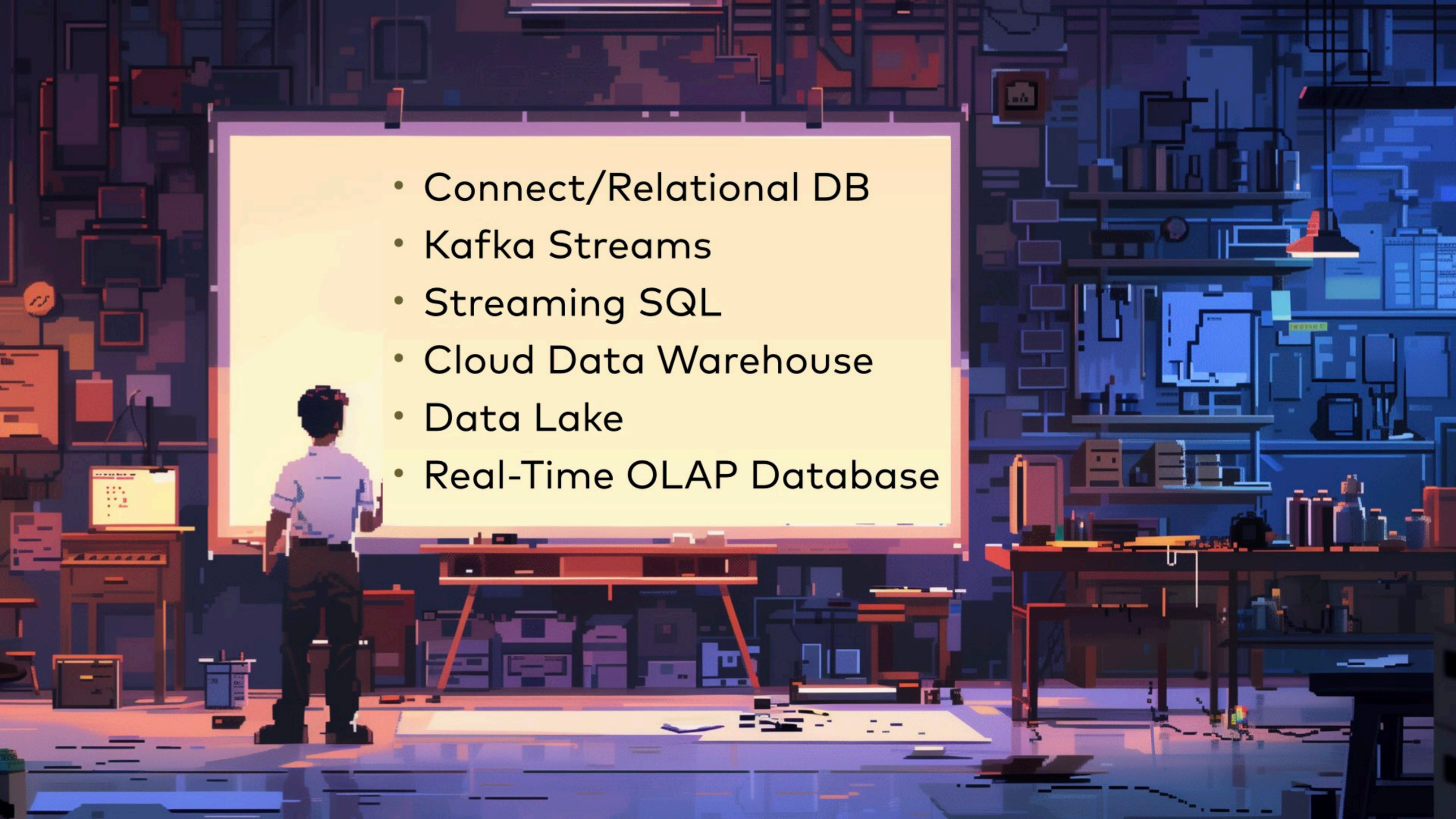

ETL and CDC


Mobile Era

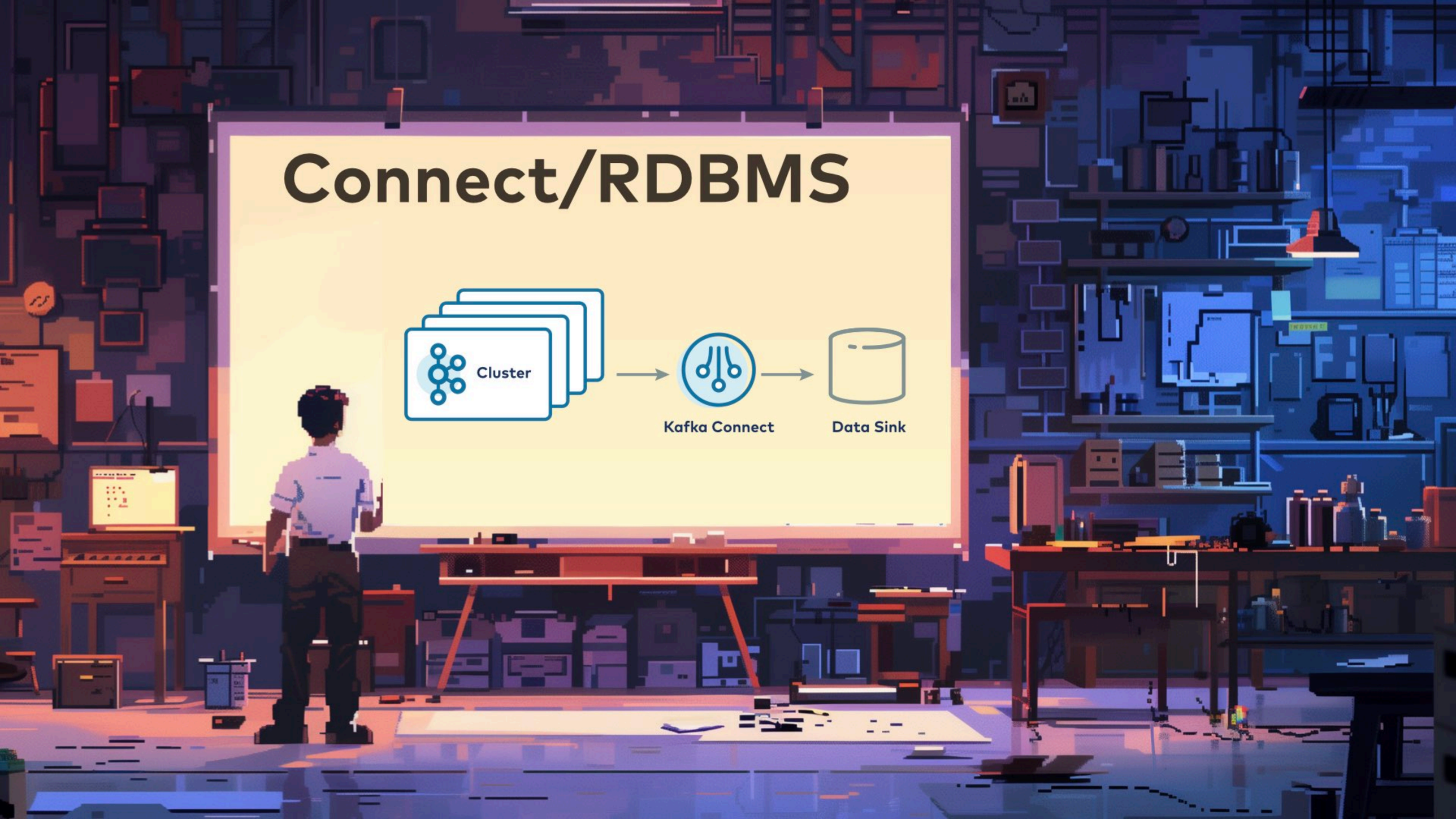

Data Pipelines
and Microservices

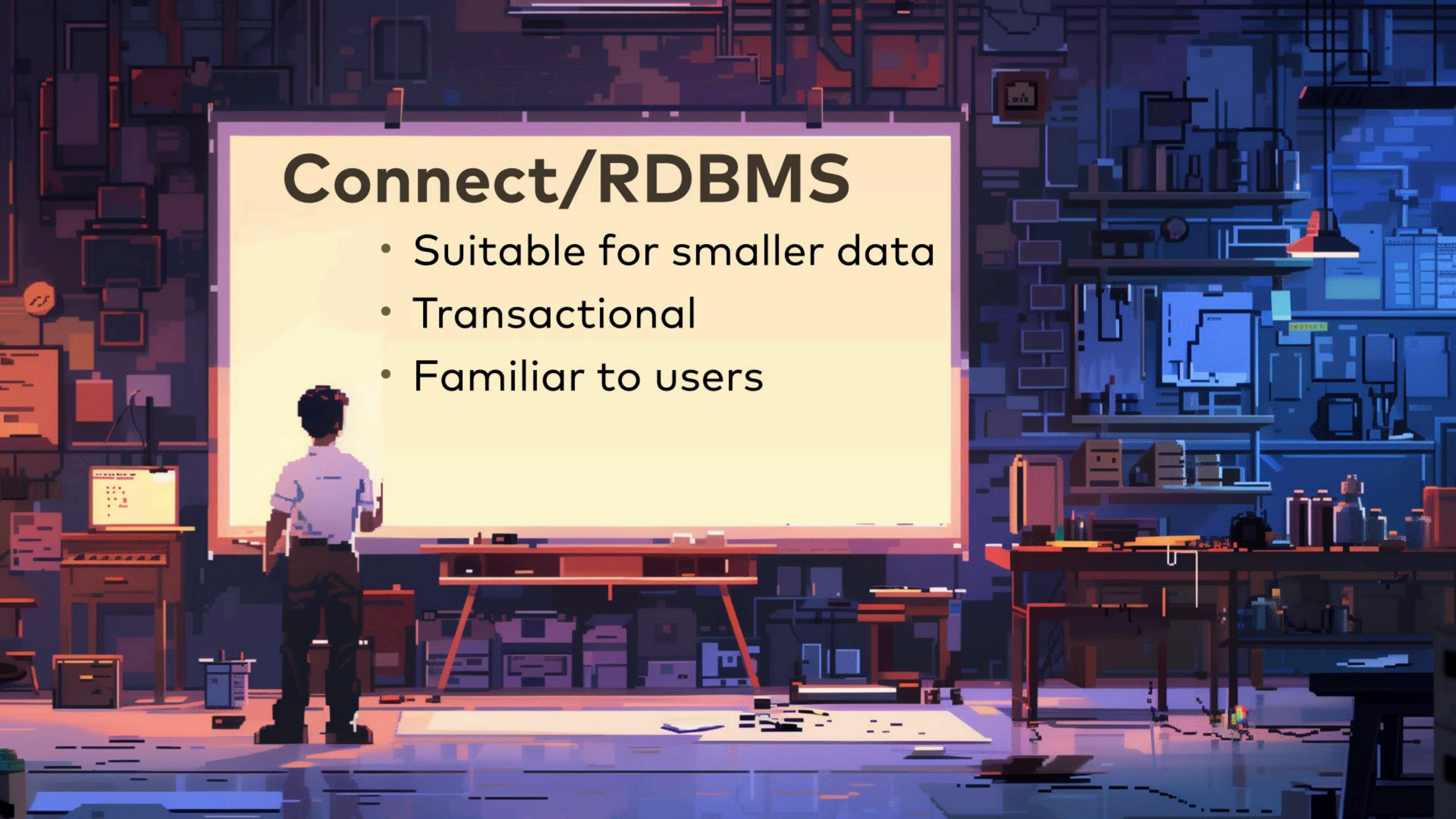

LOG


OLTP stream vs
OLAP streams

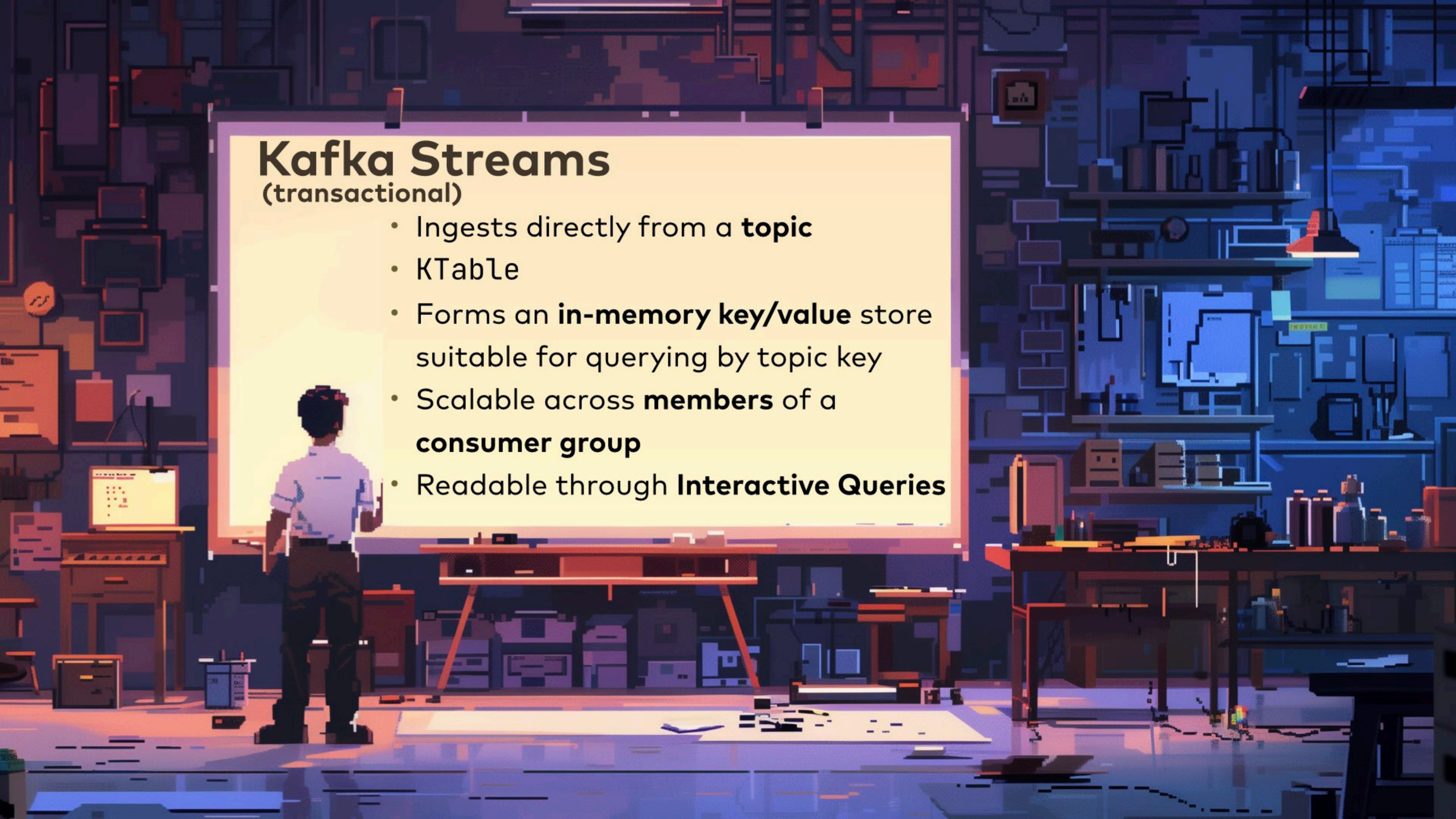

• Connect/Relational DB
• Kafka Streams
• Streaming SQL
• Cloud Data Warehouse
• Data Lake
• Real-Time OLAP Database

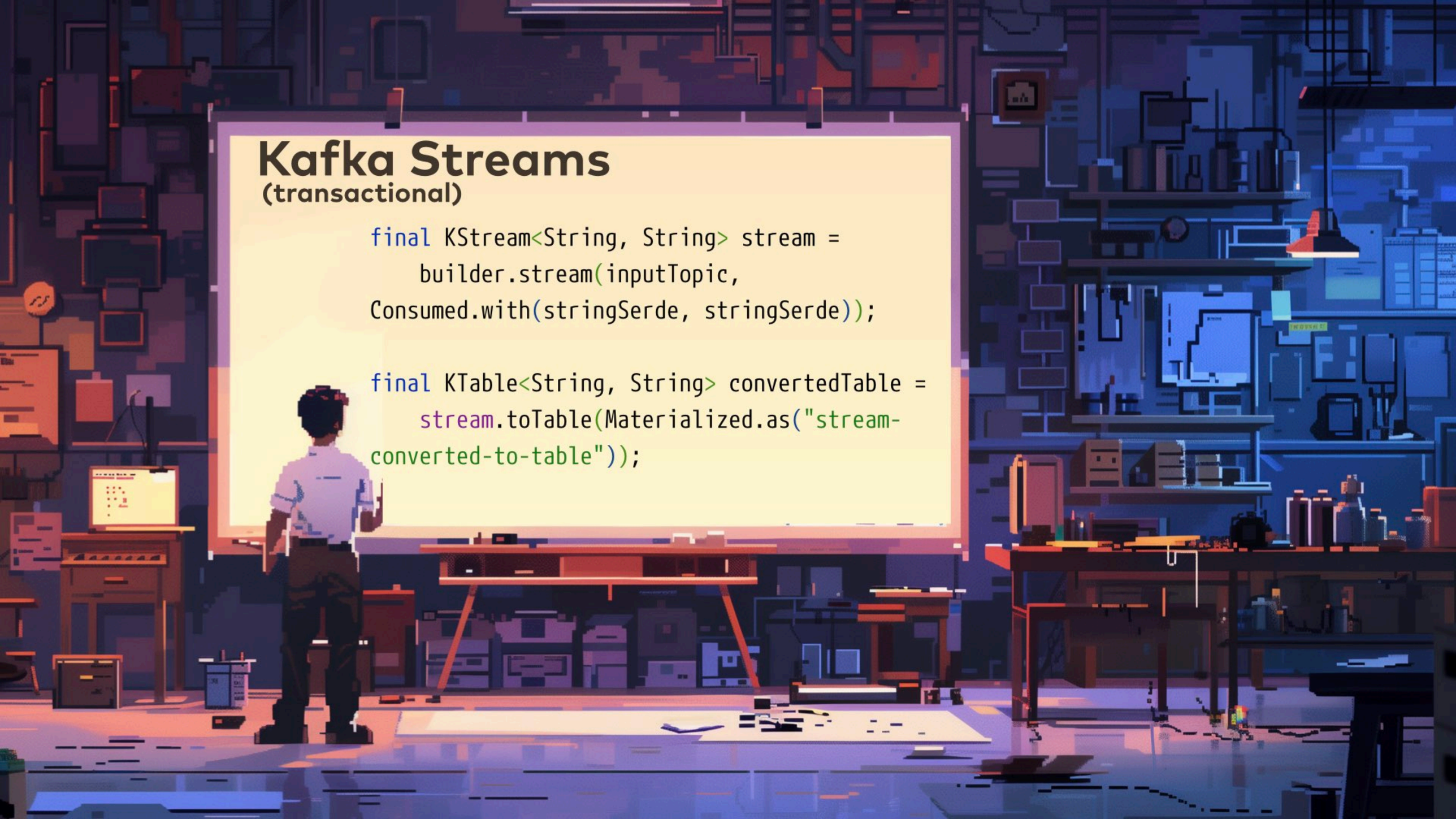

Kafka Connect


Connect/RDBMS
Cluster
Kafka Connect
Data Sink


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"));

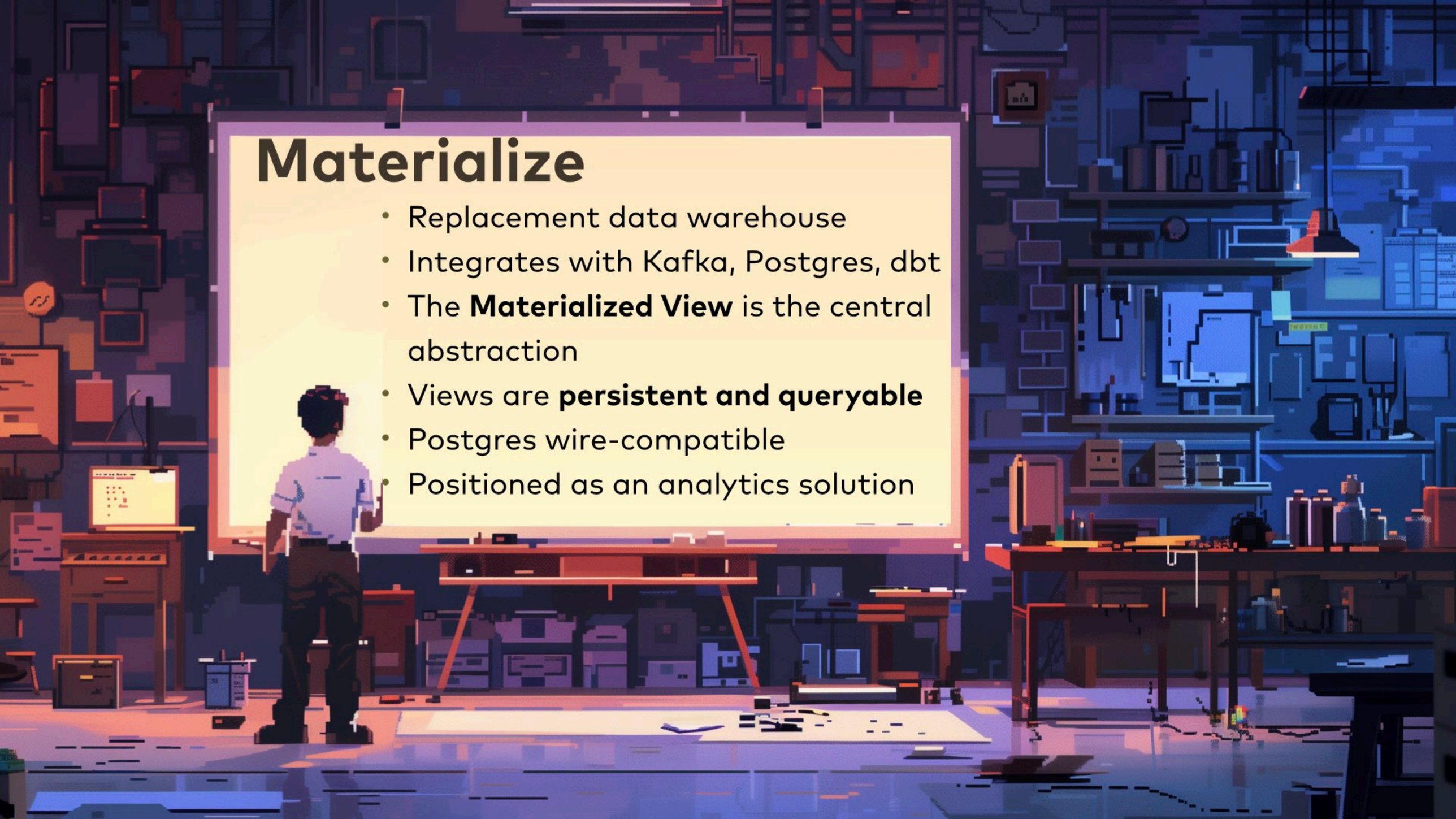

Kafka Streams
(analytical)
• Full-featured **Java stream processing API**
• Arbitrary streaming computation
• Can emit new streams (not this talk)
• `KTables` **queryable by key**
• Every read pattern requires its own topology
• **Interactive Queries** again

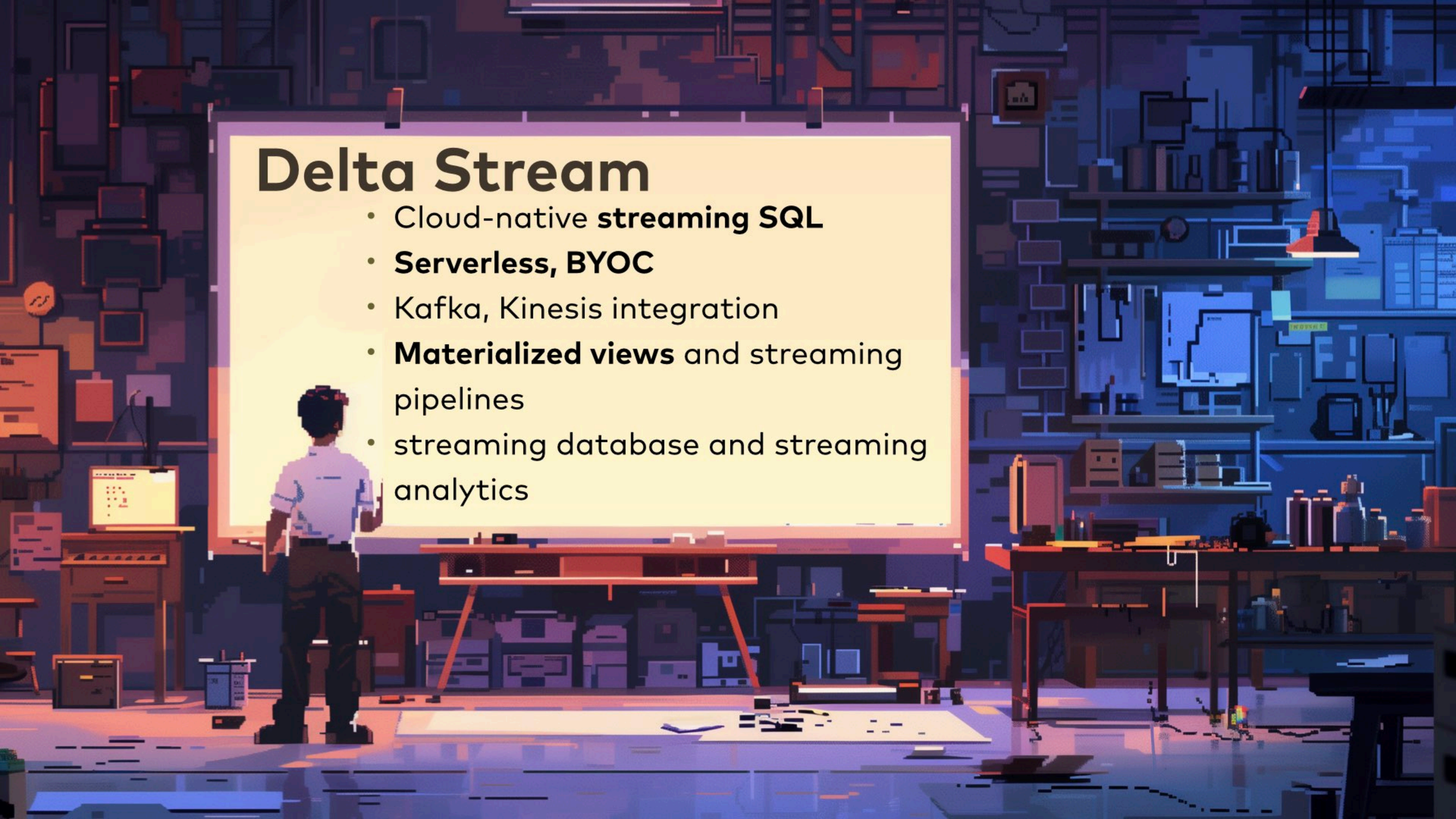

Kafka Streams
(analytical)
KTable<String, Long> wordCounts = textLines
.flatMapValues(textLine ->
Arrays.asList(textLine.toLowerCase().split("\\W+")))
.groupBy((key, word) -> word)
.count(Materialized.<String, Long, KeyValueStore<Bytes,
byte[]>>as("counts-store"));
wordCounts.toStream().to("WordsWithCountsTopic",
Produced.with(Serdes.String(),
Serdes.Long()));

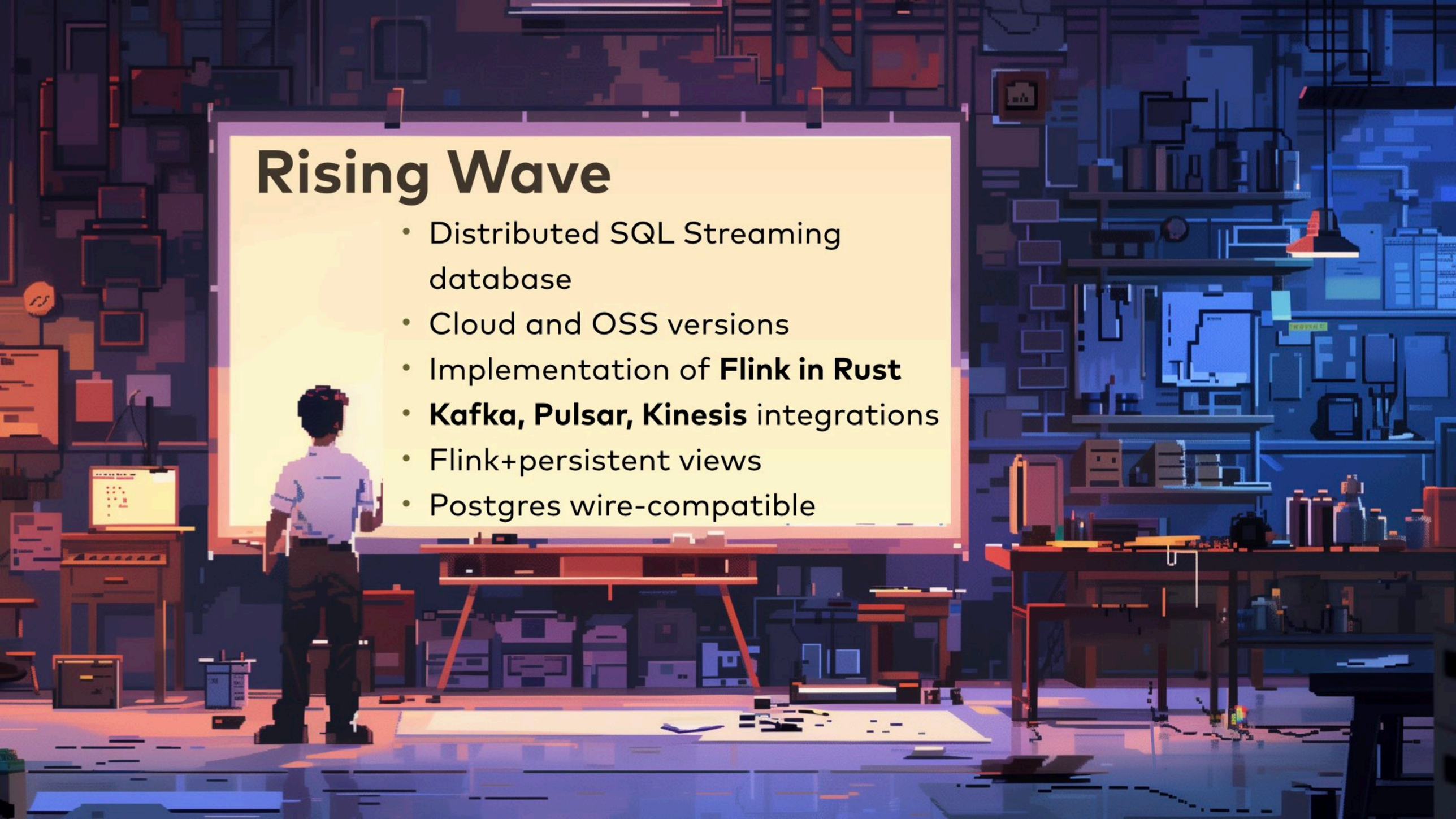

Streaming SQLs

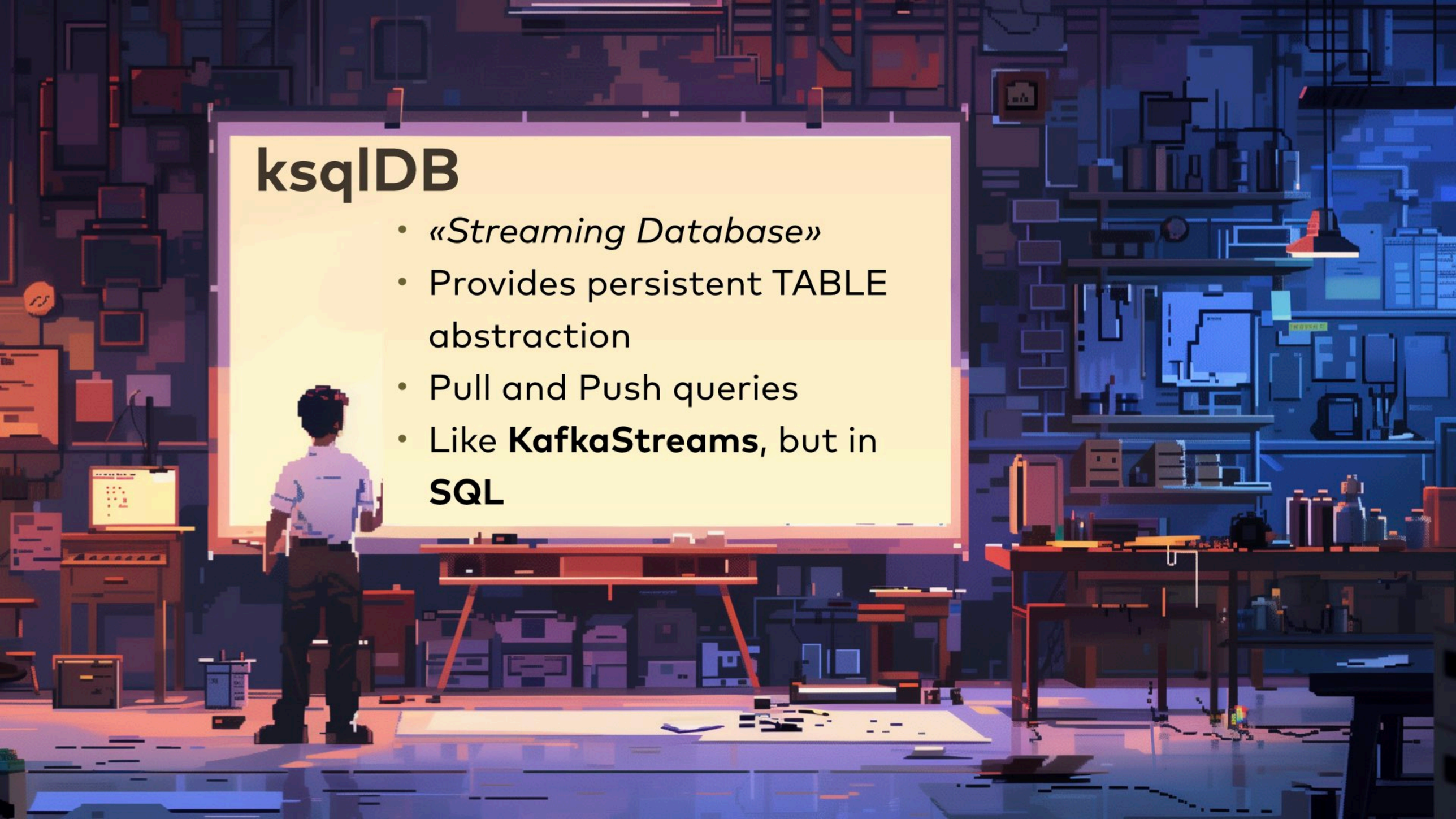

Streaming SQL
Materialize
DeltaStream
RisingWave
ksqlDB

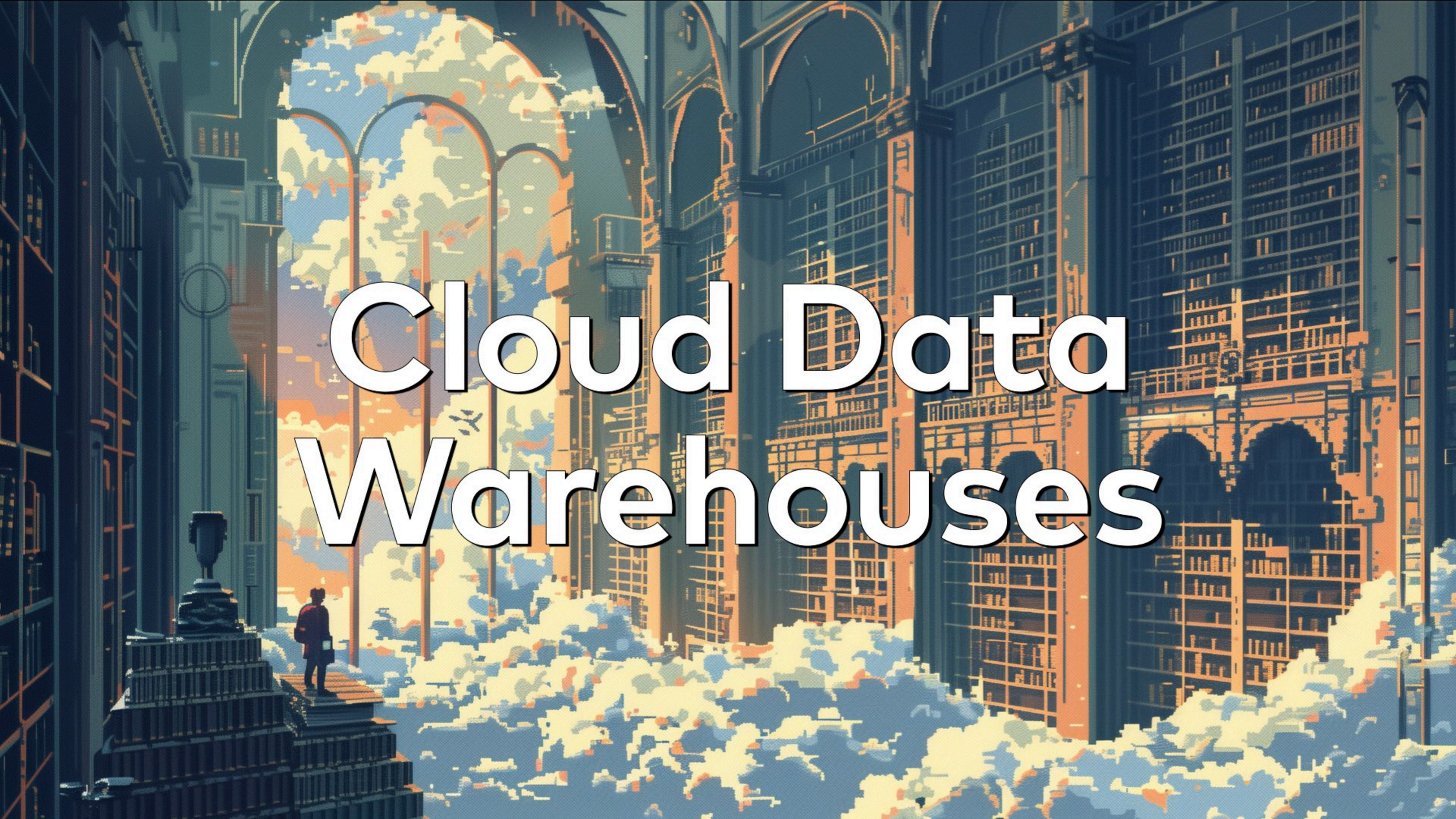

I love you, little squirrel.


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

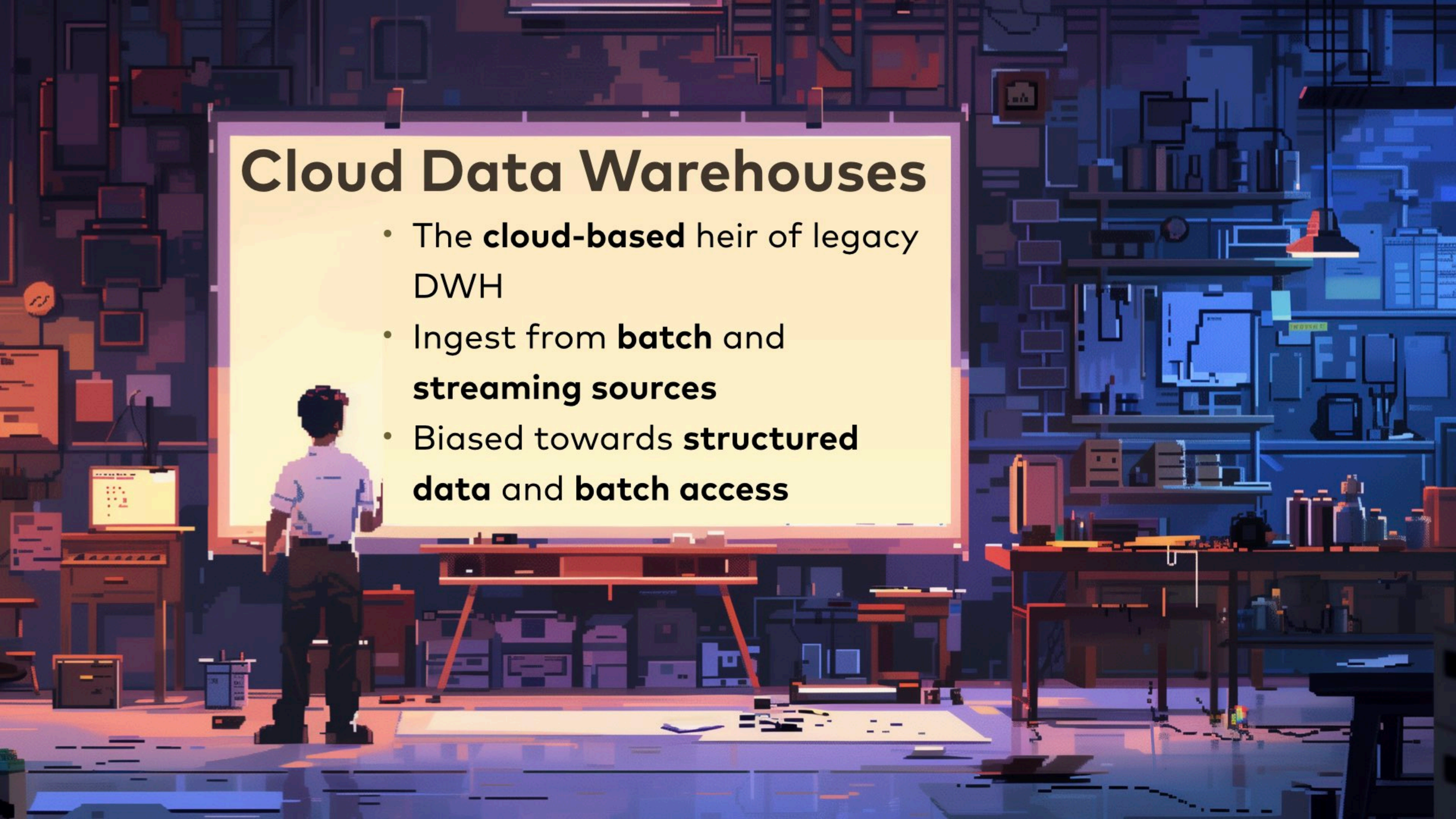

Delta Stream
Cloud-native **streaming SQL**
Serverless, BYOC
Kafka, Kinesis integration
Materialized views and streaming pipelines
streaming database and streaming analytics


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

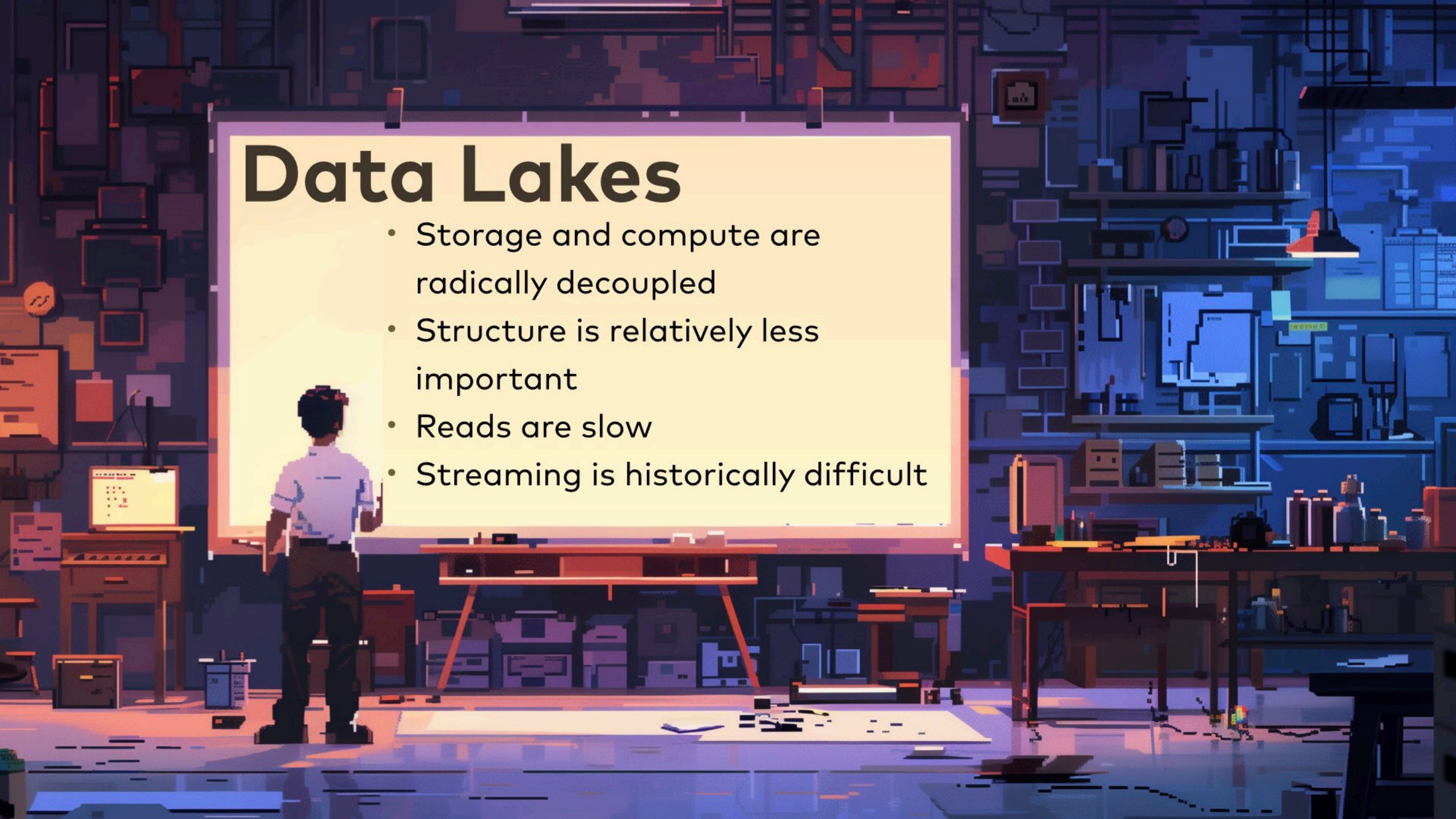

Cloud Data Warehouses


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

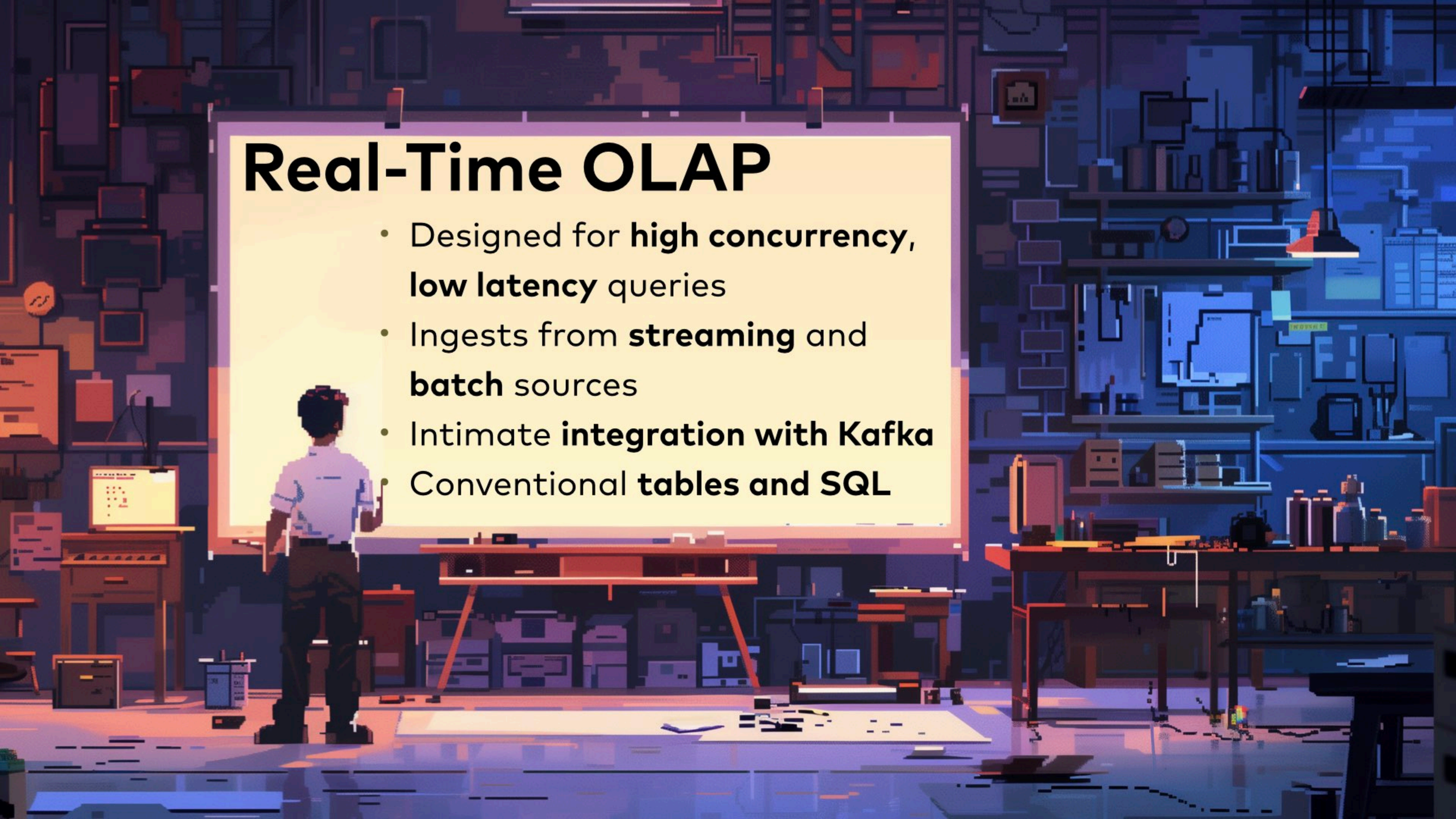

Data
Lake
Anything
else
We'll figure
this out


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


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


Real-Time Analytics
Database

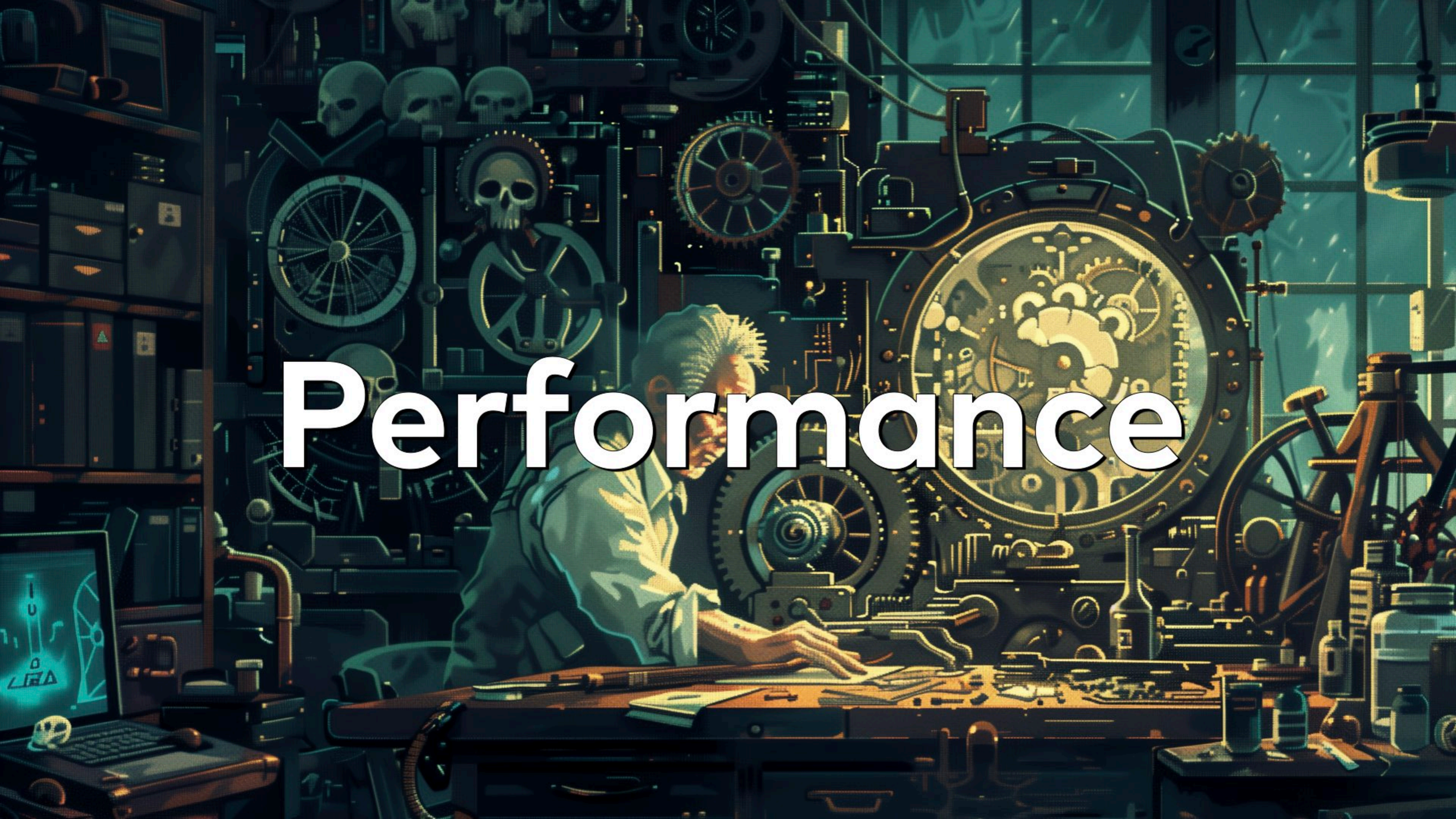

Real-Time OLAP
• Designed for **high concurrency**, **low latency** queries
• Ingests from **streaming** and **batch** sources
• Intimate **integration with Kafka**
• Conventional **tables and SQL**


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


Differentiated Application Code
Area of Exploration
Kafka