The Streaming Mindset

… what, why, how?

Marta Paes (@morsapaes)
Developer Advocate

ververica

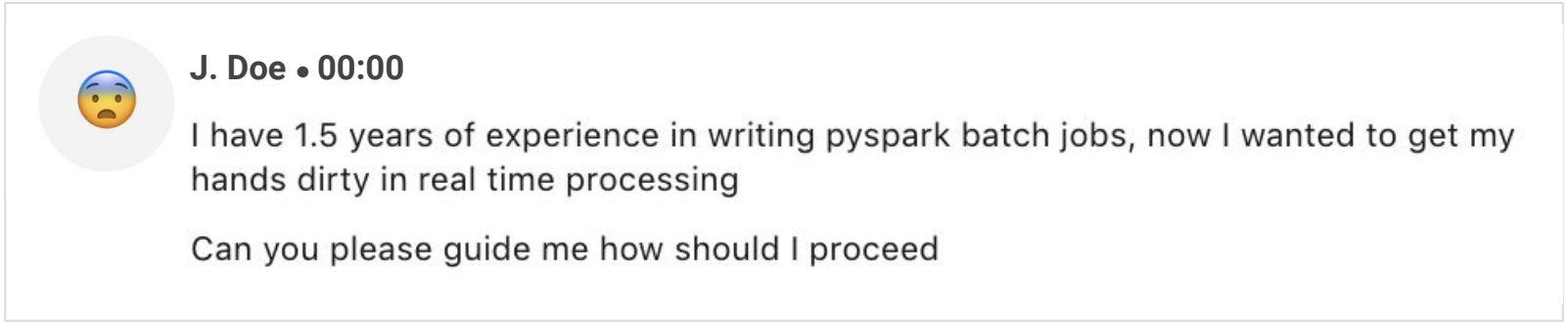
About Ververica

Original Creators of
Apache Flink®

Enterprise Stream Processing
With Ververica Platform

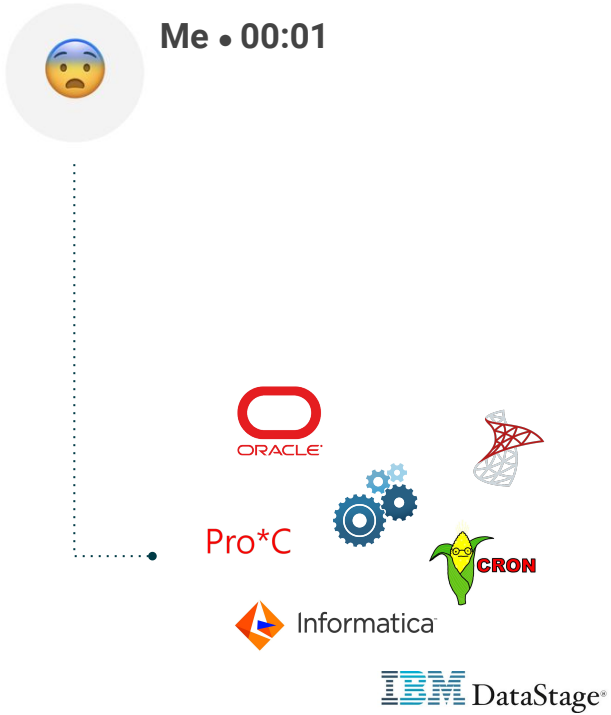
Part of
Alibaba Group

Working in DevRel



Working in DevRel

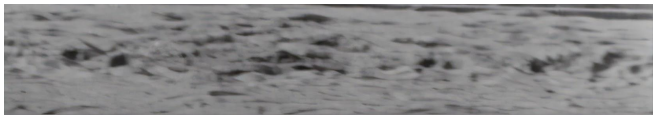
Working in DevRel



Working in DevRel

Me • 00:01

Pro*C



Working in DevRel

Me • 00:01

ORACLE

Pro*C

CRON

Informatica

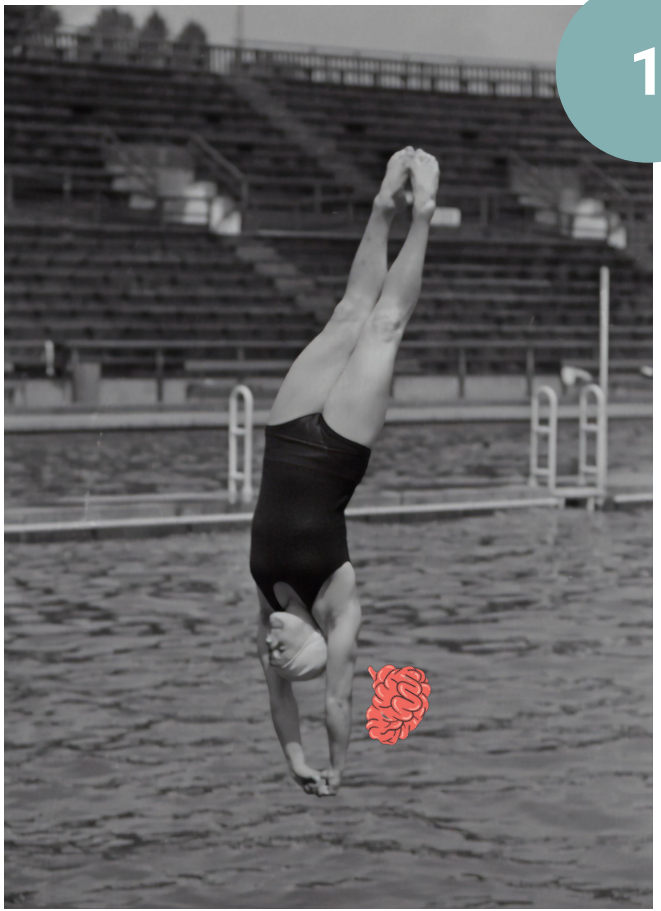
IBM DataStage

hadoop MapReduce

STORM

APACHE Spark

Where do you start?

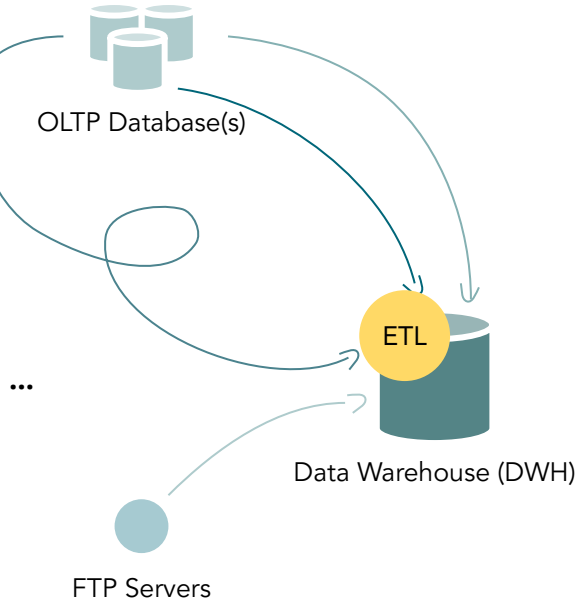


1

Go Headfirst

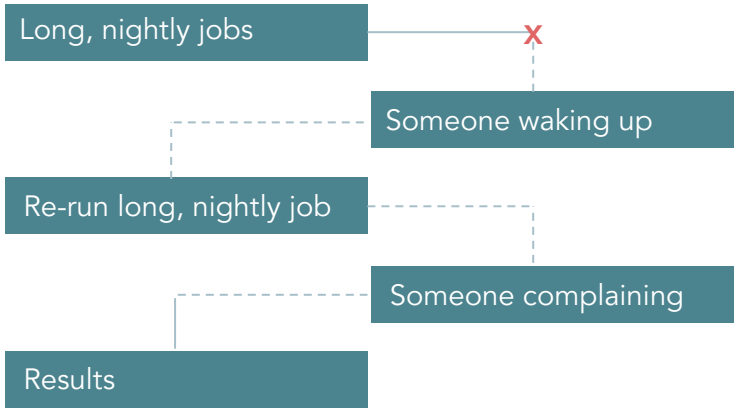
- Stream Processing 101

Analytics...Not that Long Ago



Analytics...Not that Long Ago

The quest for data...



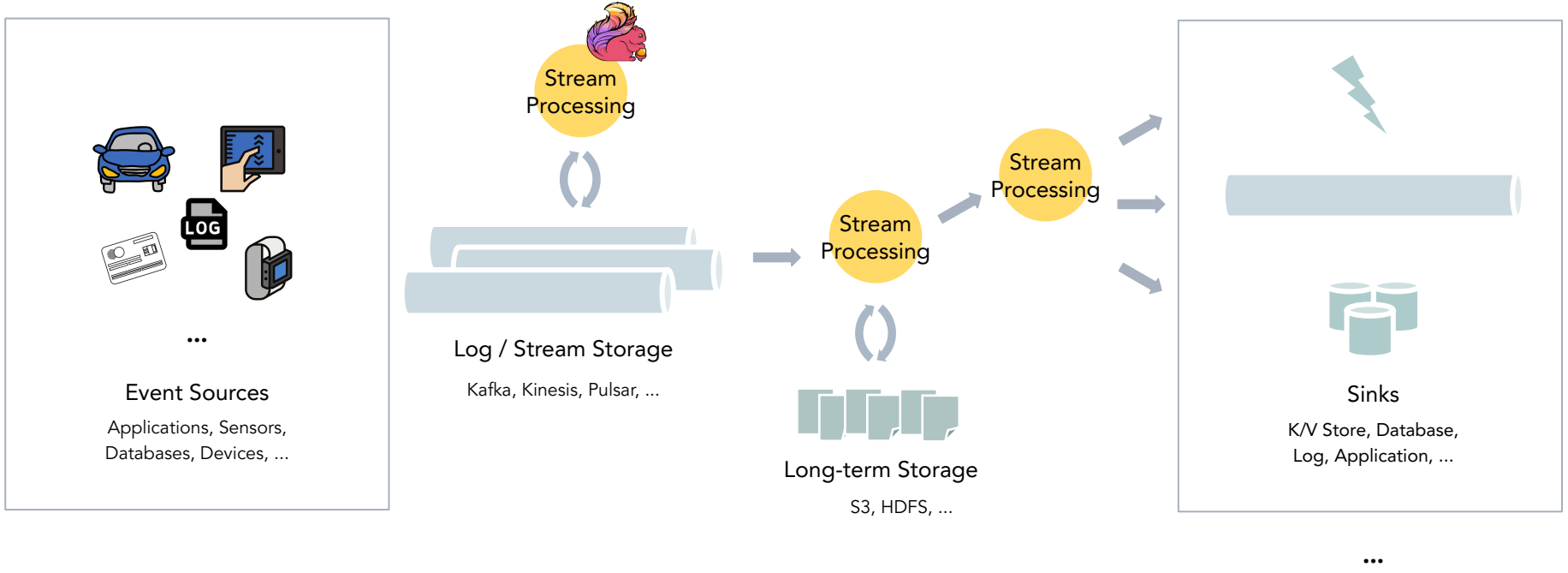
But in the end...

- Most source data is continuously produced
- Not everyone can wait for yesterday's data
- Most logic is not changing that frequently

Everything is a Stream

Everything is a Stream

Your static data records become **events** that are **continuously produced** and should be **continuously processed**.



Stream Processing 101

Batch Processing	Continuous Streaming
query/logic changes fast	data changes fast
data changes slowly	query/logic changes slowly
E.g: Ad-hoc queries, data exploration, ML model training	E.g: Most business logic nowadays

A good starter: Streaming 101: the World Beyond Batch

Stream Processing 101

Batch Processing	Continuous Streaming
query/logic changes fast	data changes fast
data changes slowly	query/logic changes slowly
E.g: Ad-hoc queries, data exploration, ML model training	E.g: Most business logic nowadays

Offline ML
Model Training

Data Warehousing
OLAP / BI / Reporting

Unified Offline/
Online Analytics

Continuous
ETL

Real-time Behavior Modeling
(e.g. recommenders, pricing)

Continuous Monitoring
(e.g. position, risk)

Online ML Model
Training/Evaluation

Real-time Alerting
(e.g. fraud, security)

Distributed
OLTP-style Apps

Stream Processing Use Cases

Examples

Large-scale Data Pipelines

ML-Based Fraud Detection

Service Monitoring & Anomaly Detection

Stream Processing Use Cases

Examples

NETFLIX	ING	aws
Large-scale Data Pipelines	ML-Based Fraud Detection	Service Monitoring & Anomaly Detection
	Uber	criteo
Unified Online/Offline Model Training	E2E Streaming Analytics Pipelines	ML Feature Generation

2

Bridge Concepts

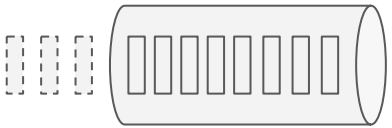
- Bounded vs. Unbounded data
- Event time vs. Processing time
- Fault tolerance

Bounded vs. Unbounded Data

Batch Processing

- Data “at rest”
- Hard boundaries (e.g. process 1 day of data)

Continuous Streaming



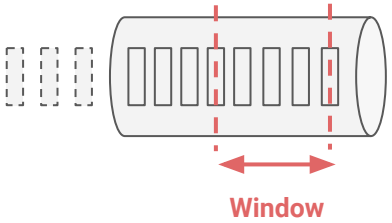
- Data “on the fly”
- Ever-growing, infinite data set

Bounded vs. Unbounded Data

Batch Processing

- Data “at rest”
- Hard boundaries (e.g. process 1 day of data)

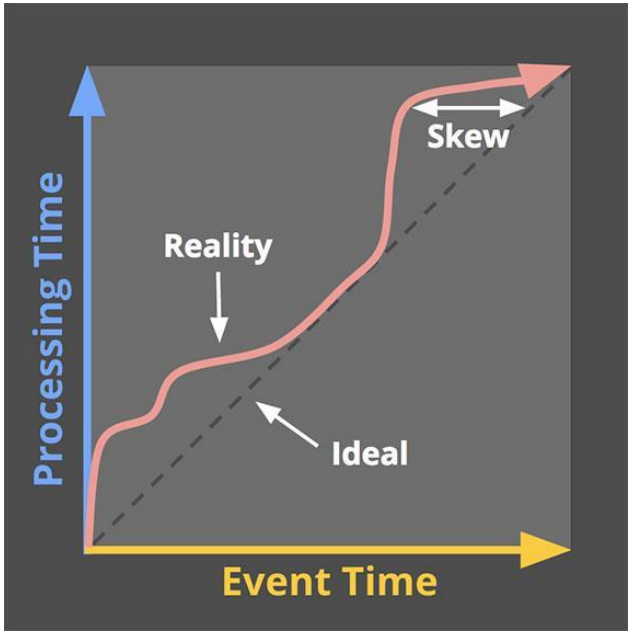
Continuous Streaming



- Data “on the fly”
- Ever-growing, infinite data set

Windows split the stream into buckets of finite size, over which you can apply computations

Event Time vs. Processing Time



Event time

- Deterministic results
- Handle out-of-order or late events
- Trade-off result completeness/correctness and latency

Processing time

- Non-deterministic results
- Best performance and lowest latency
- Speed > completeness/correctness

Fault Tolerance

Batch Processing	Continuous Streaming
pipelines run on a fixed schedule	Long-running pipelines

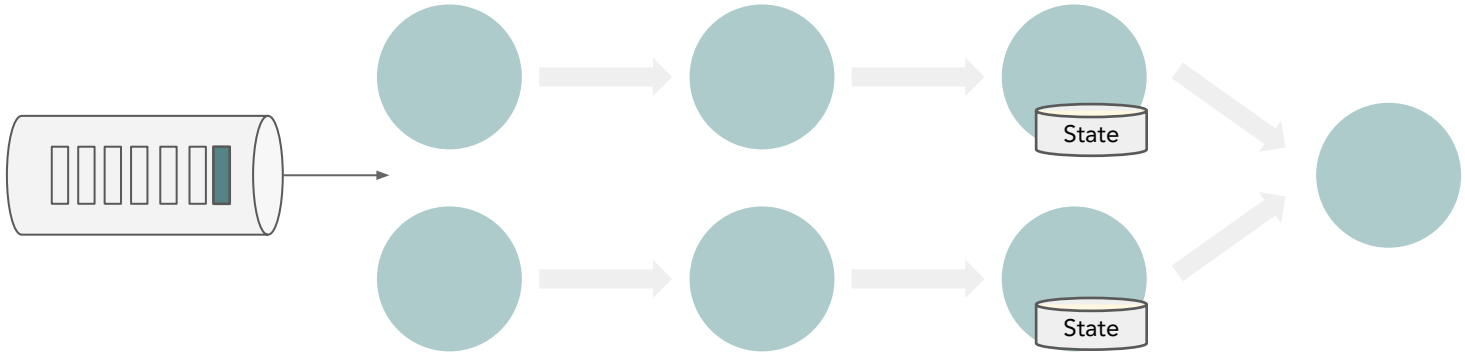
Fault Tolerance

Batch Processing

pipelines run on a fixed schedule

Continuous Streaming

Long-running pipelines



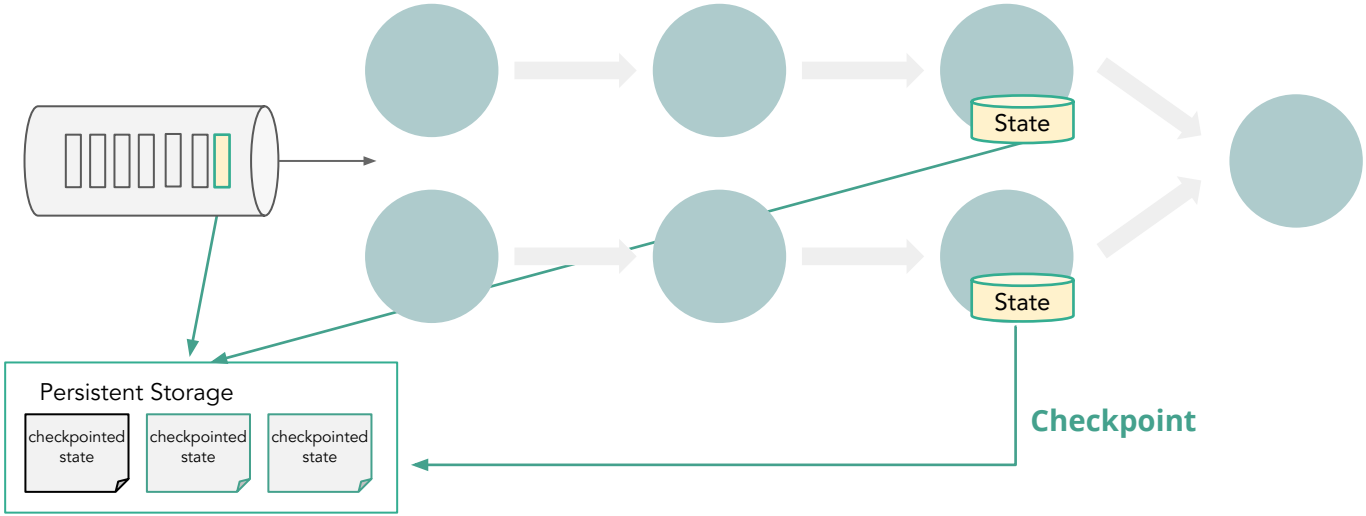
Fault Tolerance

Batch Processing

pipelines run on a fixed schedule

Continuous Streaming

Long-running pipelines



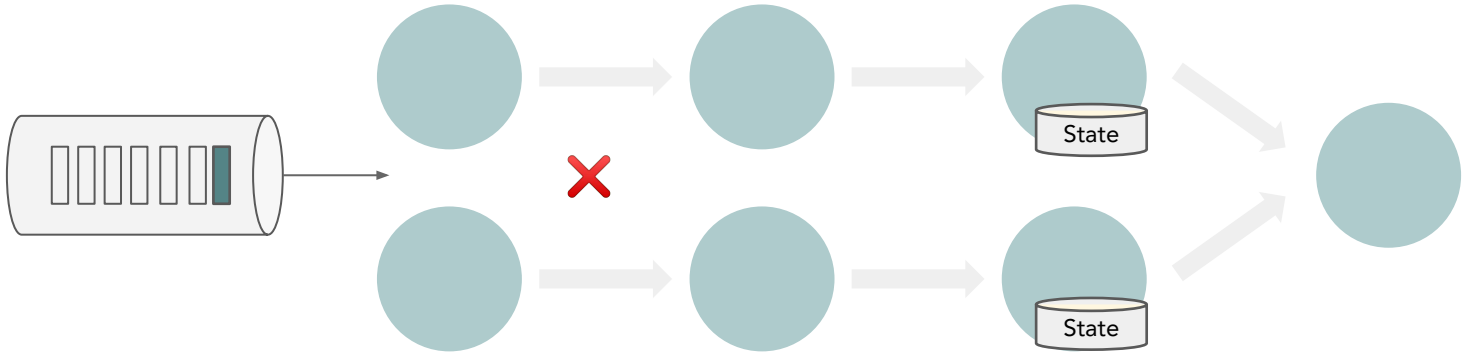
Fault Tolerance

Batch Processing

pipelines run on a fixed schedule

Continuous Streaming

Long-running pipelines



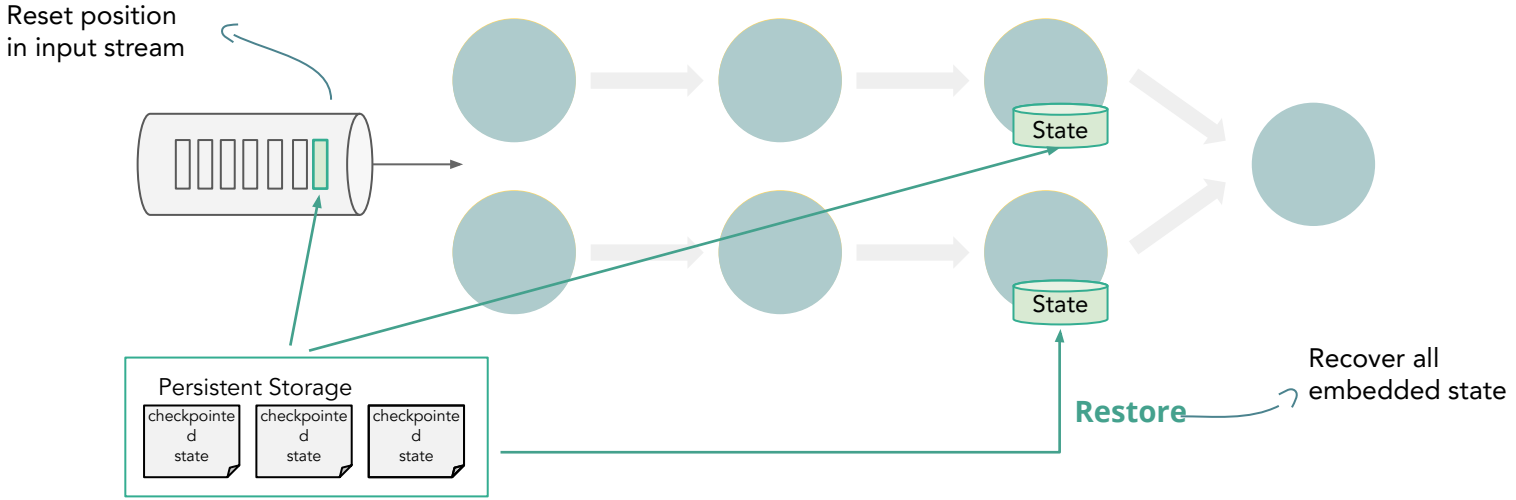
Fault Tolerance

Batch Processing

pipelines run on a fixed schedule

Continuous Streaming

Long-running pipelines

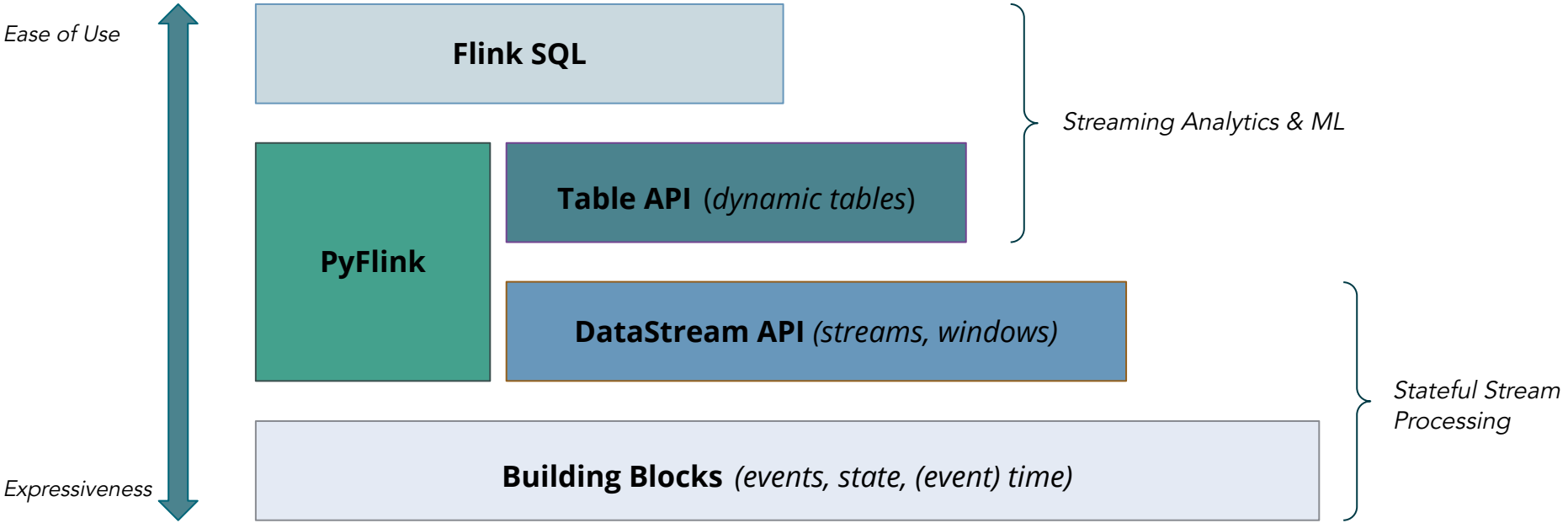


3

Pick a Flavour & Build

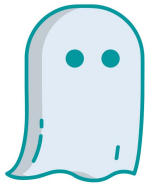
The Flink API Stack

Layered, with different tradeoffs for **expressiveness** and **ease of use**. You can mix and match all the APIs!



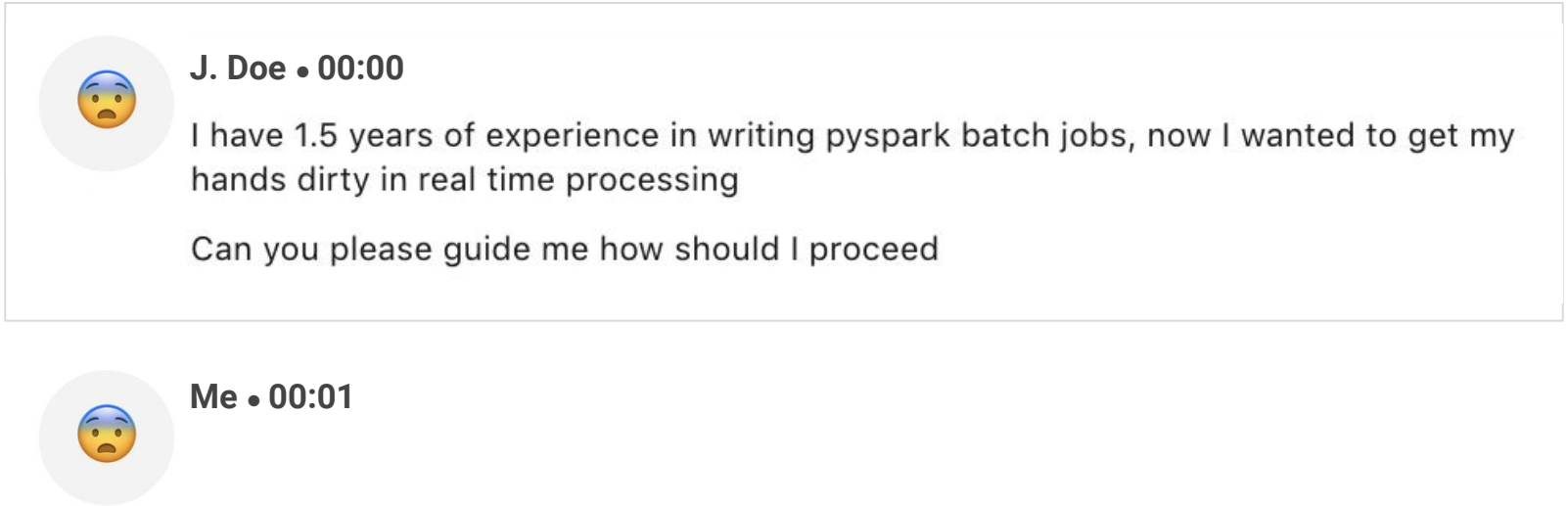
How to Get Hands-On?

Start with whatever language and/or abstractions are more familiar to you!

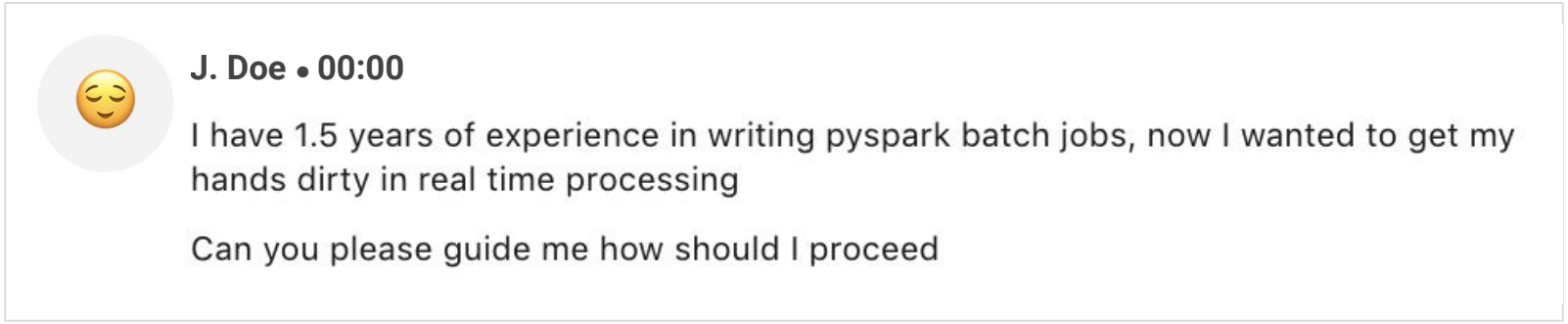
Java/Scala	SQL	Python
		
• Self-paced Training Course • DataStream API Walkthrough	• Flink SQL Cookbook • Table API Walkthrough	• PyFlink Walkthrough • Zeppelin Notebooks

Starting from the beginning

From being dumbfounded...



...to actually having a plan!



Me • 00:01

✅ Invest in learning the **Stream Processing 101**

✅ Take the time to **understand** how it differs from Batch Processing

✅ Start with something **familiar** and increase complexity gradually

✅ Ask **questions!**

* Where to ask questions: [How do I get help from the Apache Flink community?]

Thank you, Bristech!

Follow me on Twitter: @morsapaes

Learn more about Flink: https://flink.apache.org/

ververica