A Microservices Framework for Real time Model Scoring using Structured Streaming

#SAISStreaming1

Vedant Jain

October 2018

databricks

About me

- Solutions Architect @ Databricks
- x-Hortonworks, JPMC

What will we talk about?

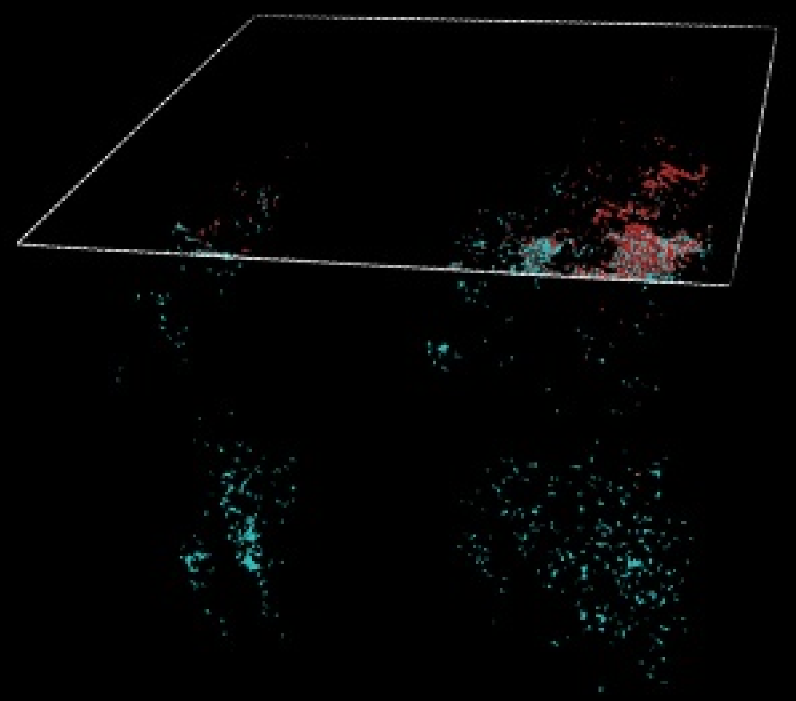
- **Machine learning**
- **Model scoring**
- **Microservices**
- **Structured Streaming**
- **Demo**

Machine Learning

*"…what we want is a **machine** that can **learn from experience**."*

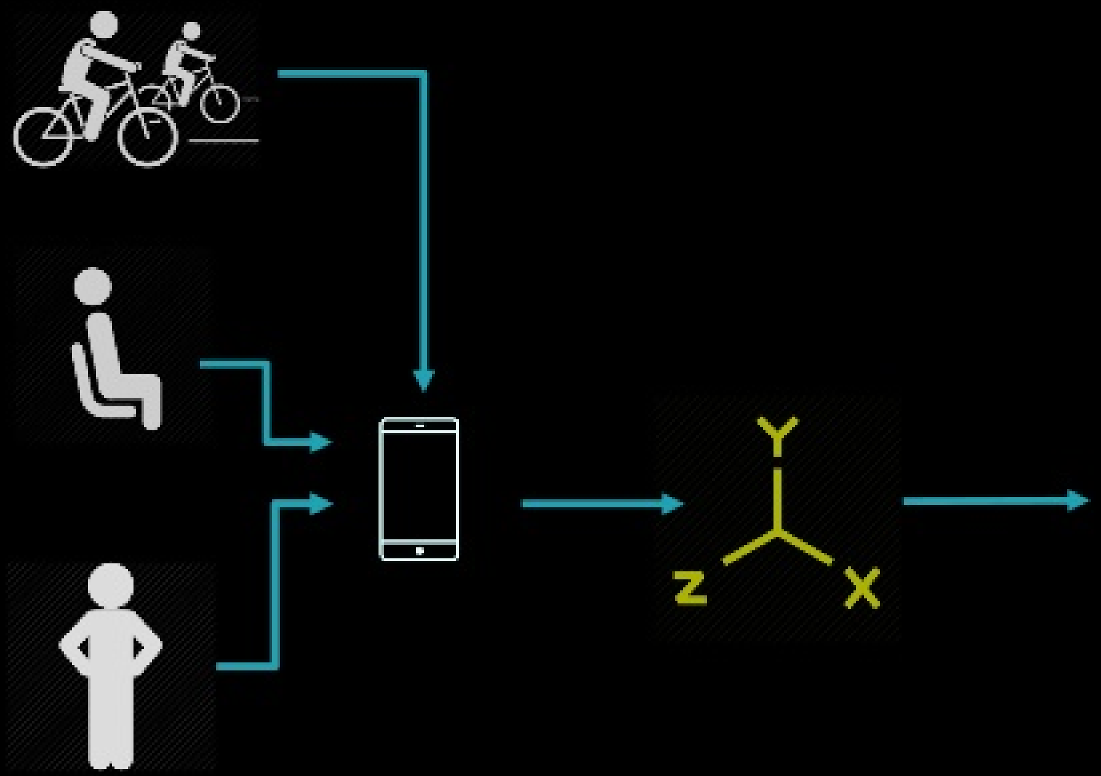
Types of Machine Learning

- **Unsupervised** Learning

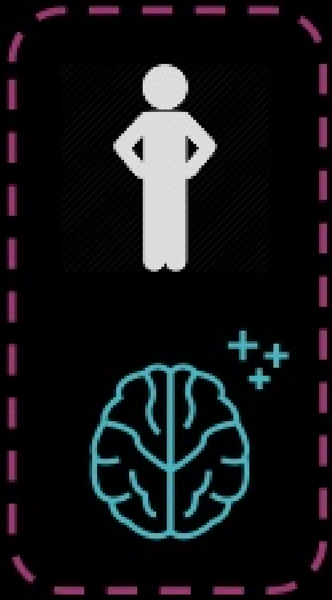


Types of Machine Learning

- **Supervised Learning**



X	Y	Z
0.5	-3.4	9.2
-2.3	2.9	3.2
1.3	2.3	-4.5

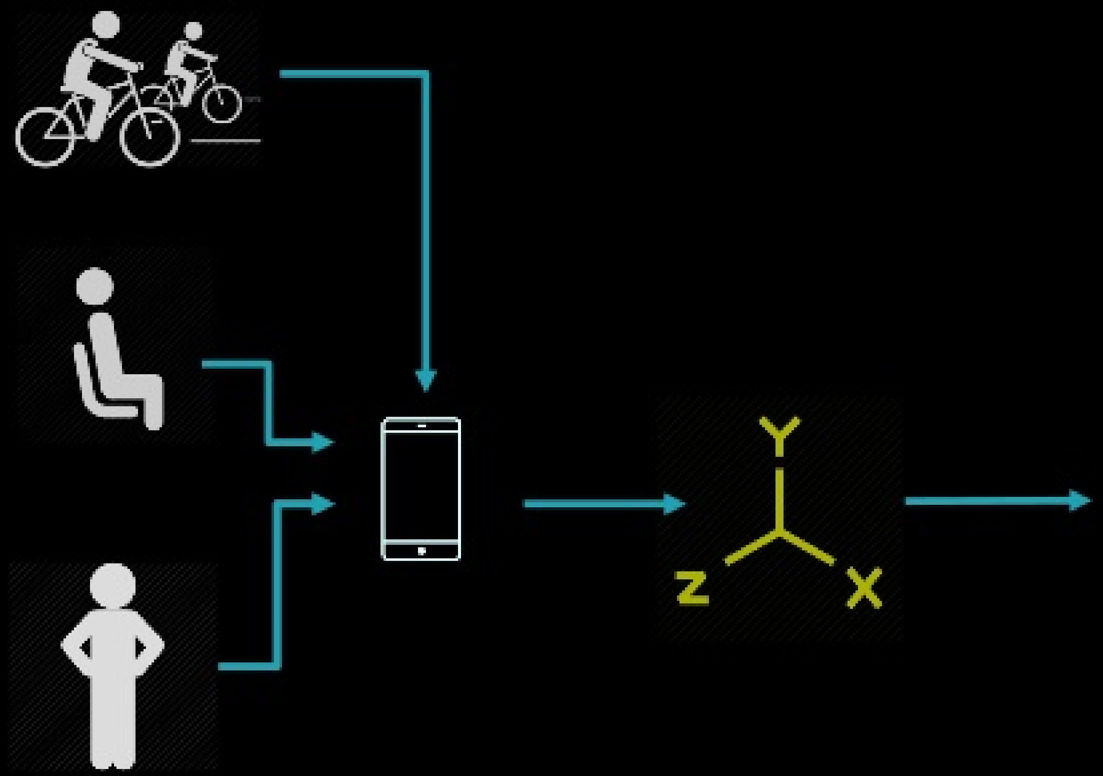


Labels

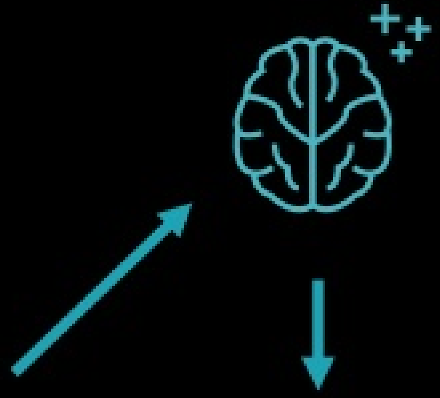
Activity
Biking
Standing
Sitting

Types of Machine Learning

- **Supervised Learning**

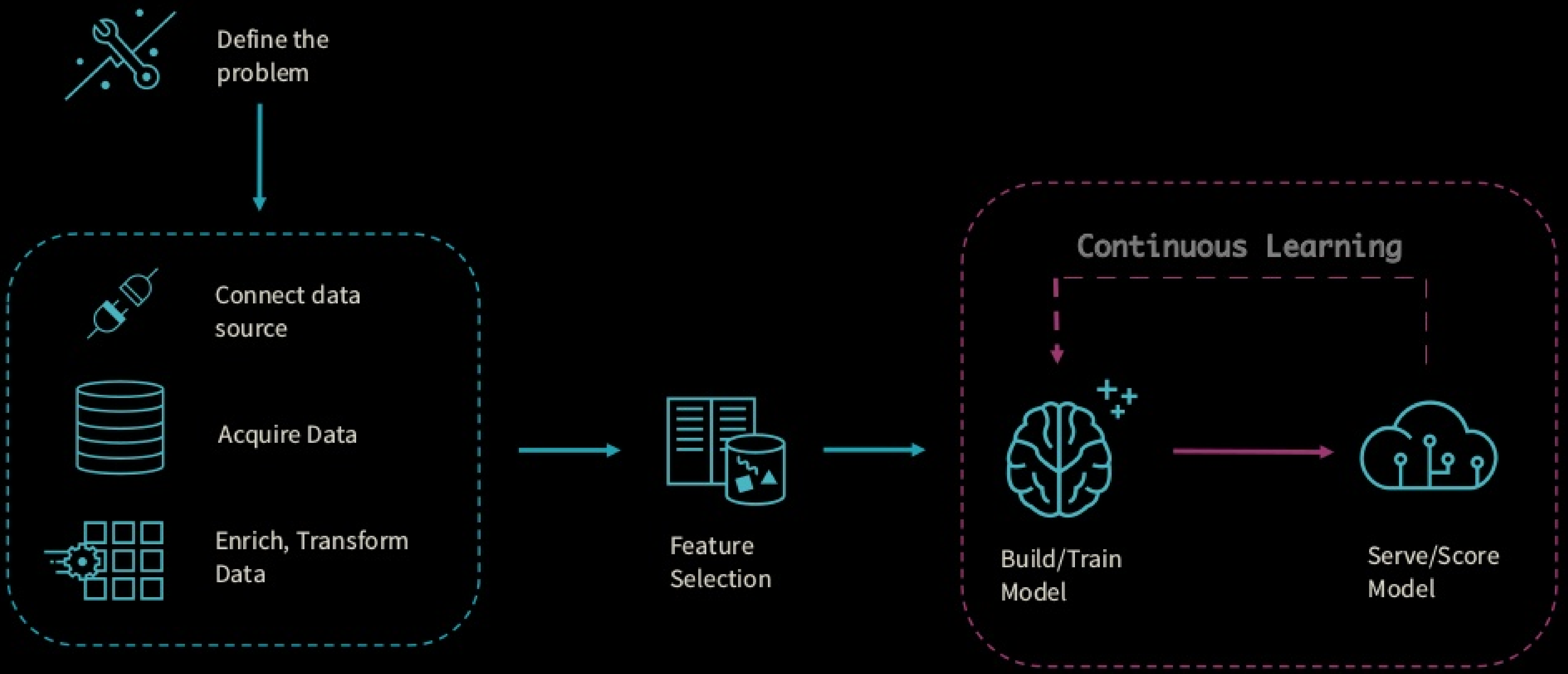


X	Y	Z
0.5	-3.4	9.2
-2.3	2.9	3.2
1.3	2.3	-4.5



Activity
Biking
Standing
Sitting

Machine Learning Process



What will we talk about?

- Machine learning
- **Model scoring**
- Microservices
- Structured Streaming
- Demo

Model Scoring

Propensity: Likelihood of a user to commit a certain action

Lead: How closely matched lead is to target profile

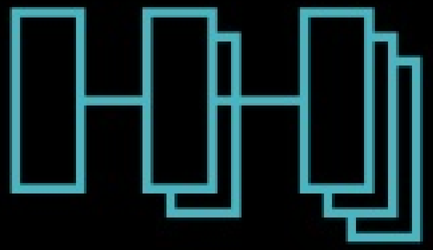
Credit: Ability of the user to keep promise if granted access

Affinity: How similar are two products or users etc.

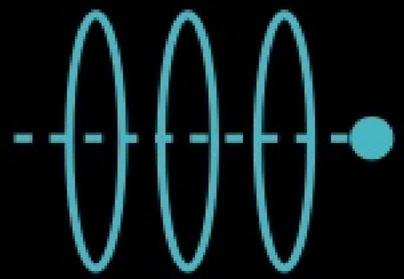
Attrition/Churn: Likelihood of a customer to drop a service and/or start using a competitor's service

Anomaly Detection: Identification of rare or invalid transaction

Model Scoring on “Big Data”



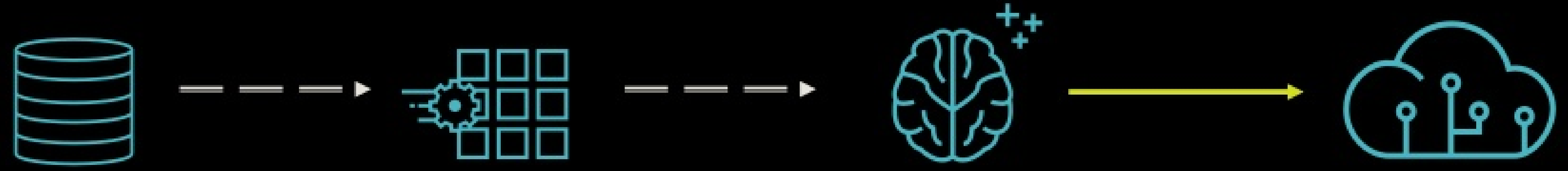
Scale



Streaming

Dynamic

Machine Learning Pipeline



- Phone/Watch
- Gyroscope/Accelerometer
- **CSV**

- Drop **null values**
- Calculate **moving averages** on 10 minute window
- Convert to single **parquet**

- OneHot**Encoder**
- RF **Classifier**
- **Cross Tabulation**
- K-Means **Clustering**
- **Hyperparameter** Tuning
- **Pipelines**

- Deploy Pipelines to **Docker**
- Serve the models using **Python Flask**

What will we talk about?

- Machine learning
- Model scoring
- **Microservices**
- Structured Streaming
- Demo

Microservices

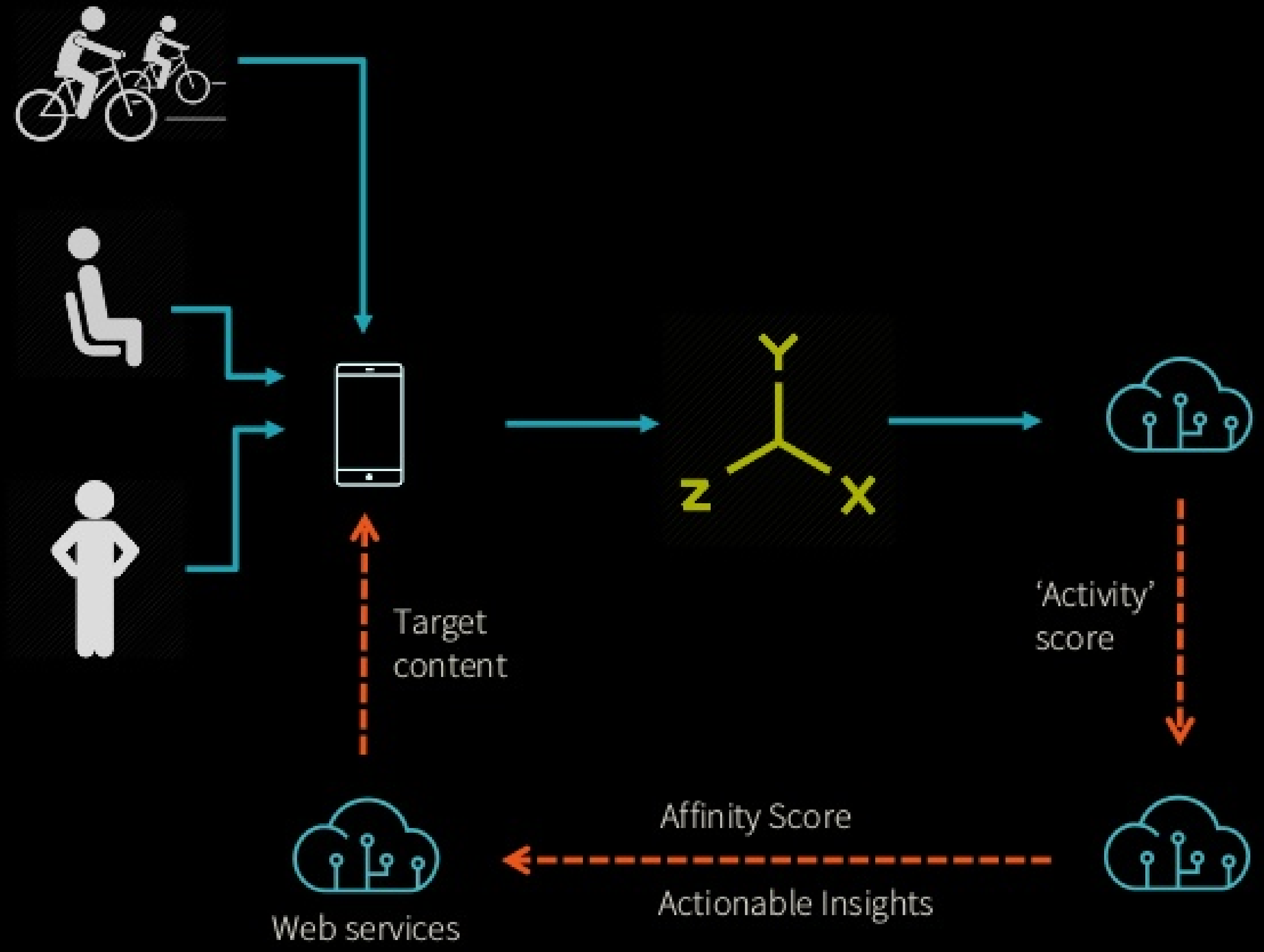
- **Properties**:
  - ➢ ***Decomposition***: Logic is broken down into multiple independent components
  - ➢ ***Isolation***: Component services are deployed and maintained independently of one another
- **Benefits**:
  - ✓ **Reduced** Regression Testing time
  - ✓ Organizational **Autonomy**
  - ✓ Cloud/On-premise **Agnostic**
  - ✓ **Scalability/Reusability**

Microservices

Example:

Company A tracks user's activity through smart devices and wants to provide tailored content to the users based on their behavior.

Microservices

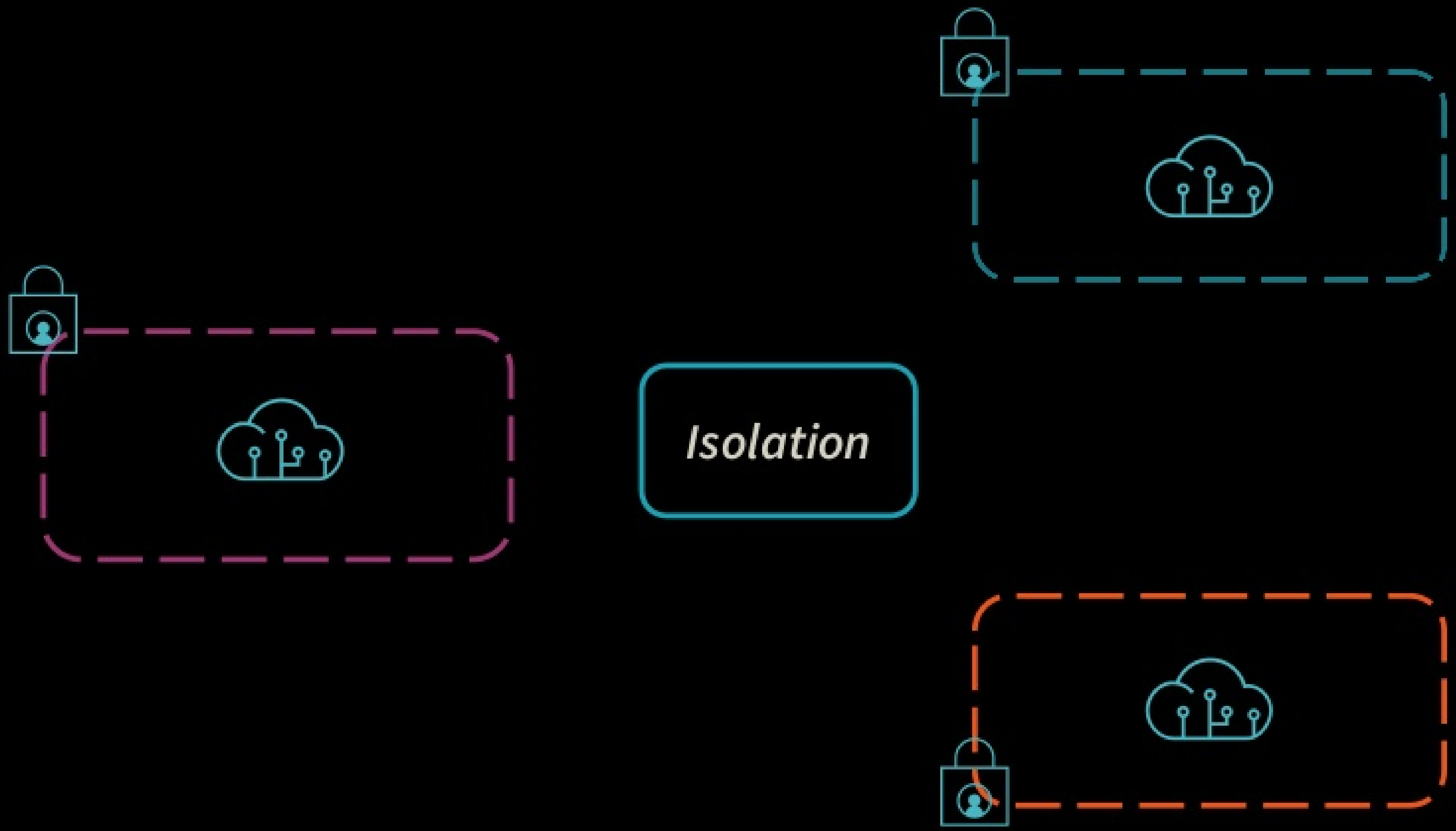


X	Y	Z	Activity
0.5	-3.4	9.2	Biking
-2.3	2.9	3.2	Standing
1.3	2.3	-4.5	Sitting

User	Bike	Stand	Sit	Cluster
A	0.1	0.3	0.6	0
B	0.0	0.3	0.7	1

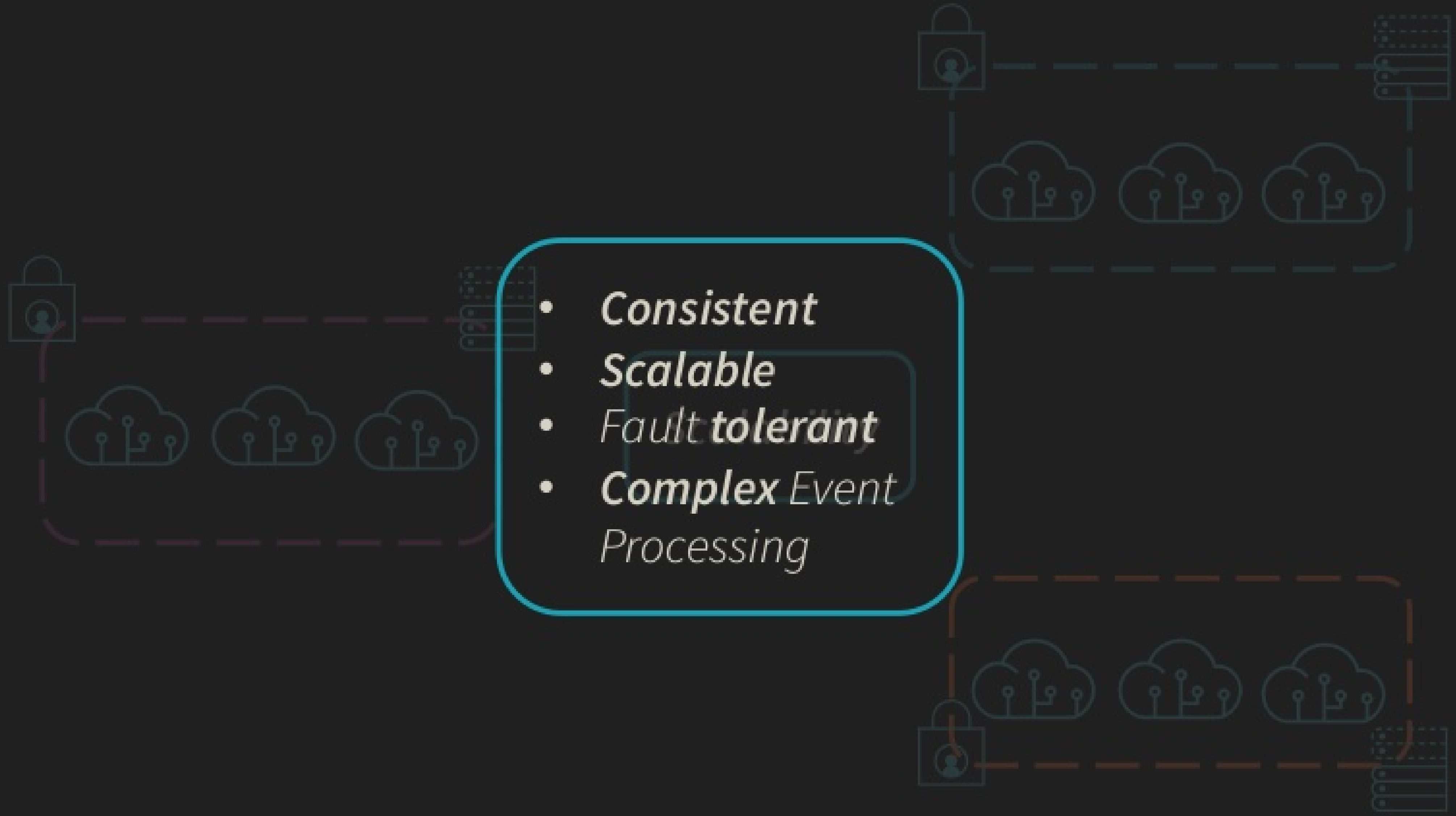
Microservices

Microservices



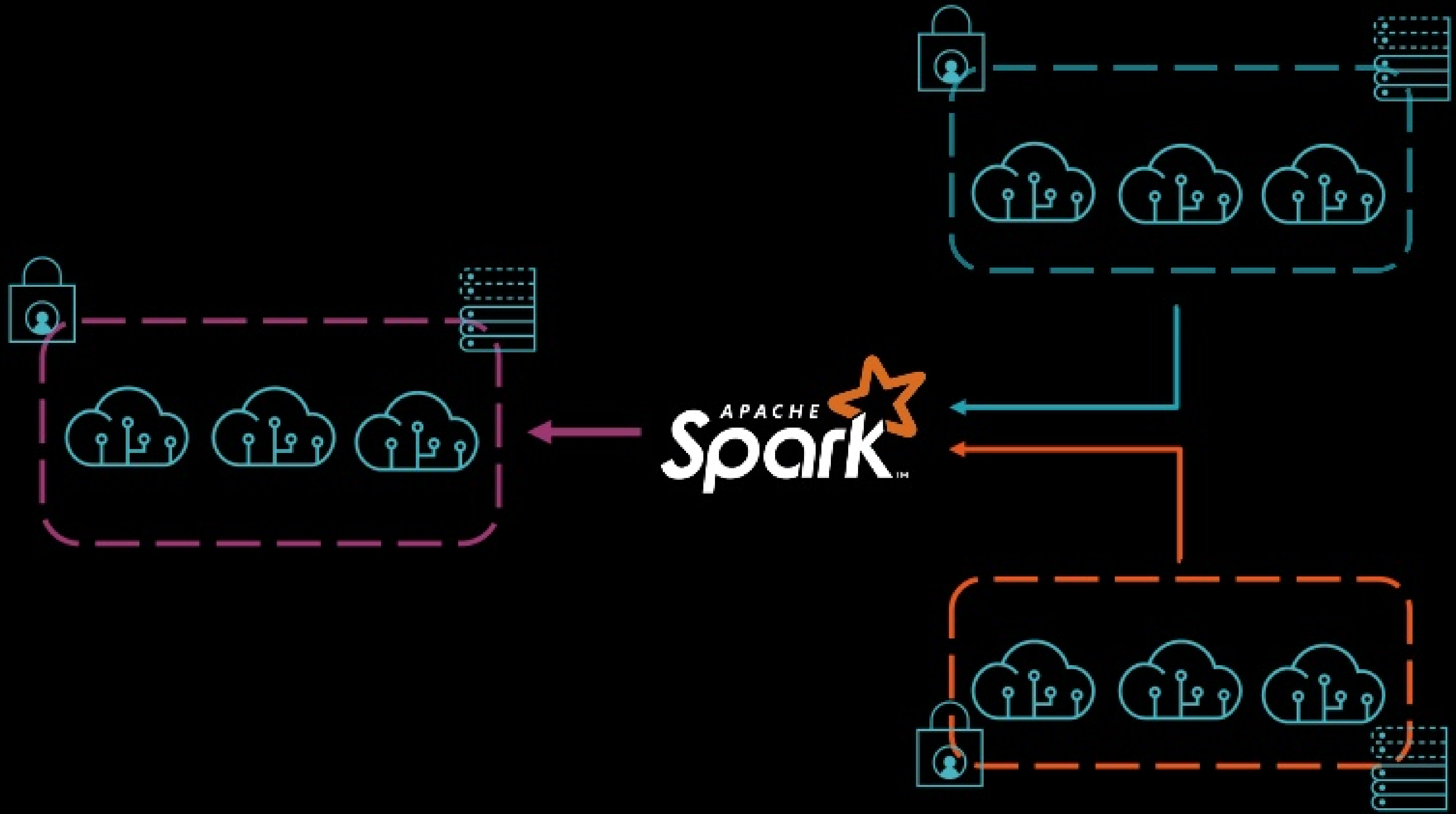
databricks

Microservices



- ***Consistent***
- ***Scalable***
- *Fault* ***tolerant***
- ***Complex*** *Event Processing*

Microservices

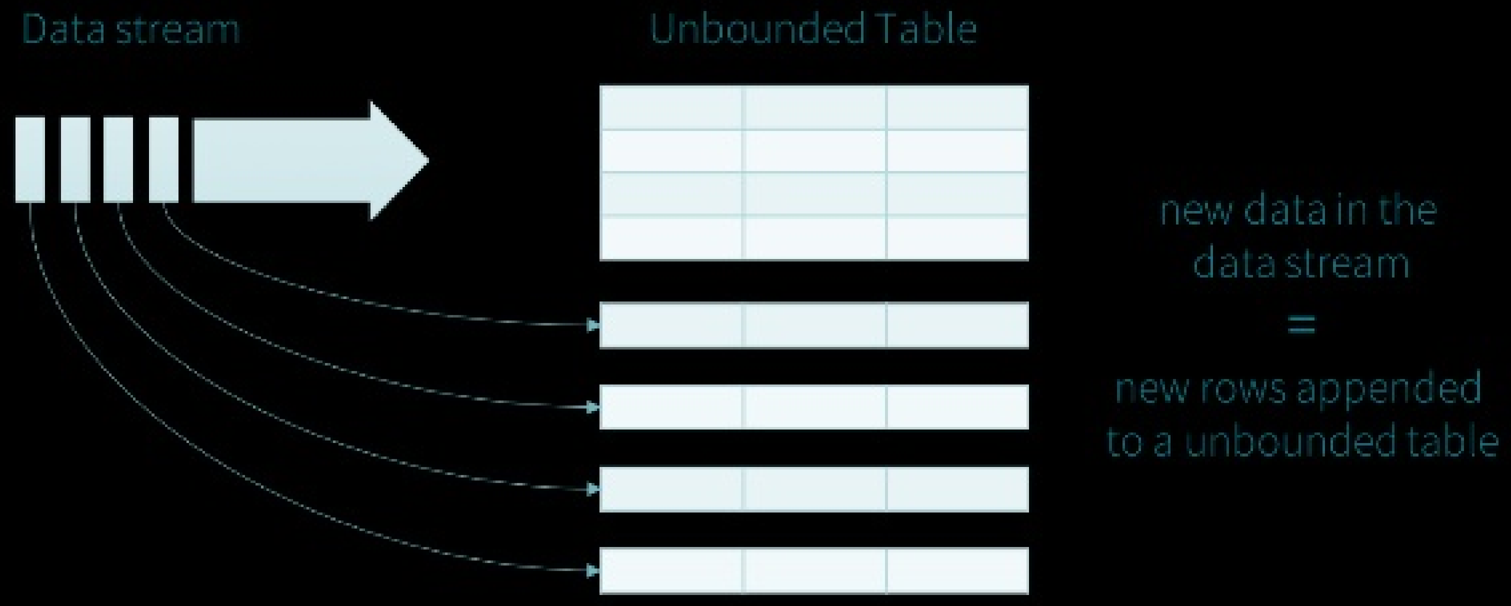


What will we talk about?

- Machine learning
- Model scoring
- Microservices
- **Structured Streaming**
- Demo

Structured Streaming

- **High-level** streaming API built on **Spark SQL** engine
- Runs the **same computation as batch queries** in Datasets/DataFrames
- Event time, **windowing**, sessions, sources & sinks
- End-to-end **exactly once** semantics
- **Late Data** Handling



Data stream as an unbounded table

ML Limitations in Streaming

- Many models/transformers/estimators are not supported
- Limited to only models built *in Spark MLLib*
- Not ideal for ***Continuous Learning***

What will we talk about?

- Machine Learning
- Model scoring
- Microservices
- Structured Streaming
- **Demo**

  https://github.com/vedantja/eu_summit_demo