

SpringOne Platform

by Pivotal.

Real-Time Performance Analysis of Data-Processing Pipelines with Spring Cloud Data Flow

Christian Tzolov (@christtzolov)

Sabby Anandan (@sabbyanandan)

Plot

Open:

Data Intensive Applications // Orchestration + Operational Challenges

Pitch:

Spring Cloud Data Flow // Orchestration + Operationalization

Use:

Credit Cards as the theme

Show:

Architecture details, solution walkthrough, and demos



Developers



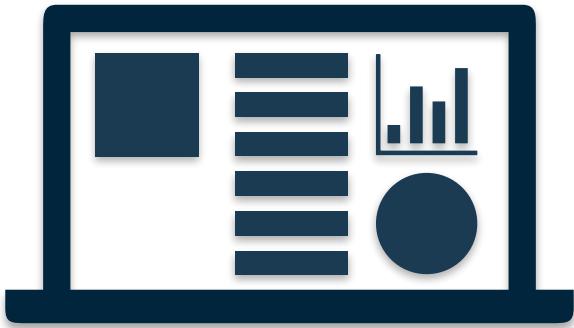
Applications



Application



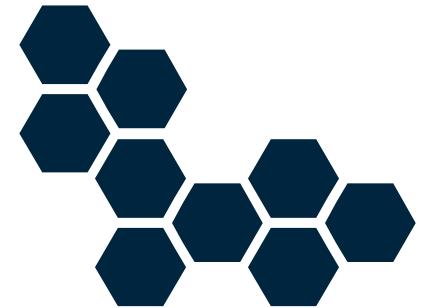
Monolith



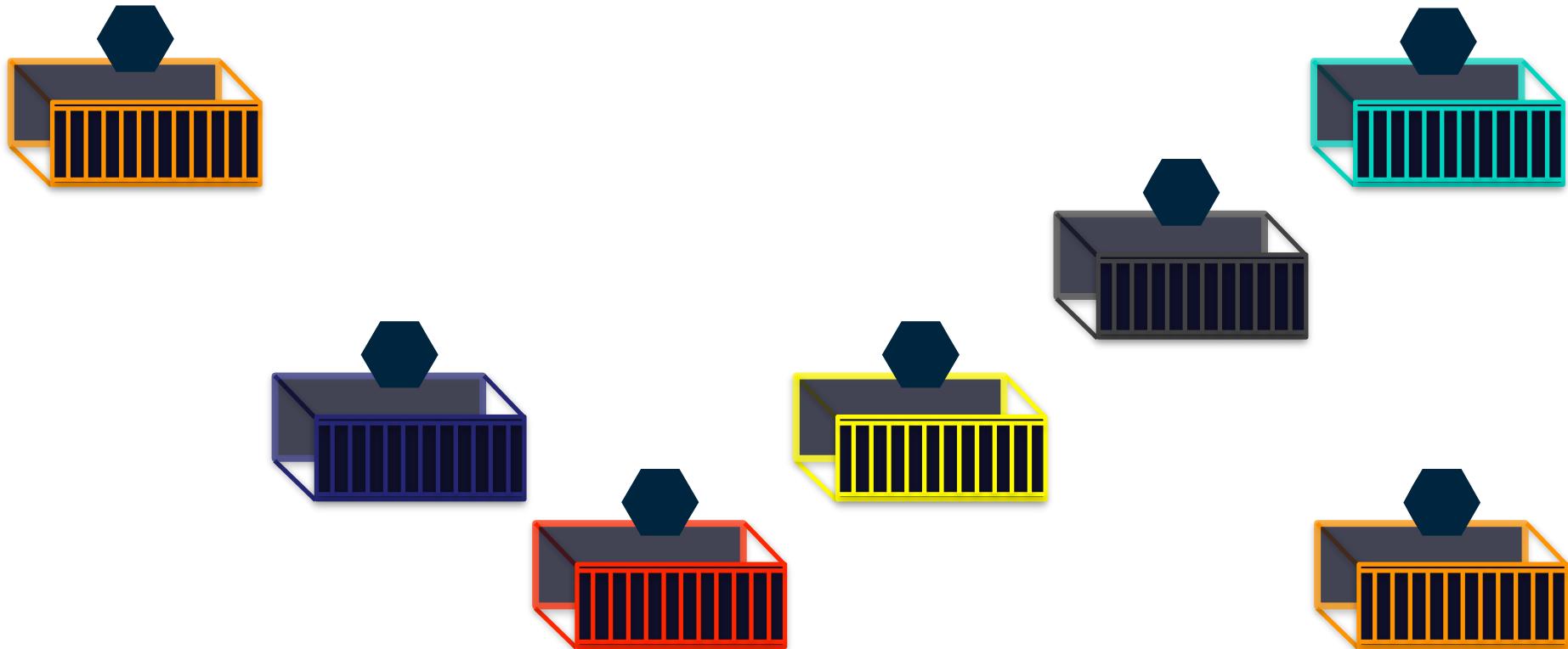
Application



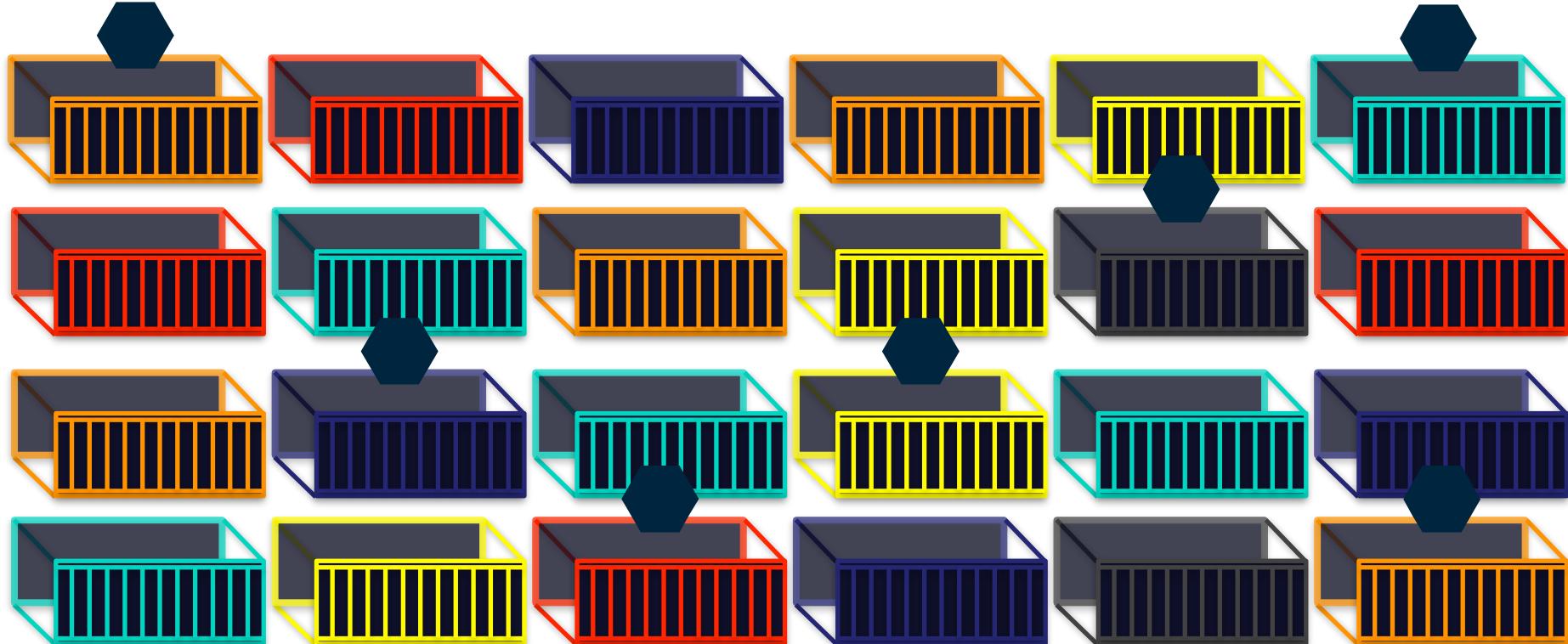
Monolith



Microservices



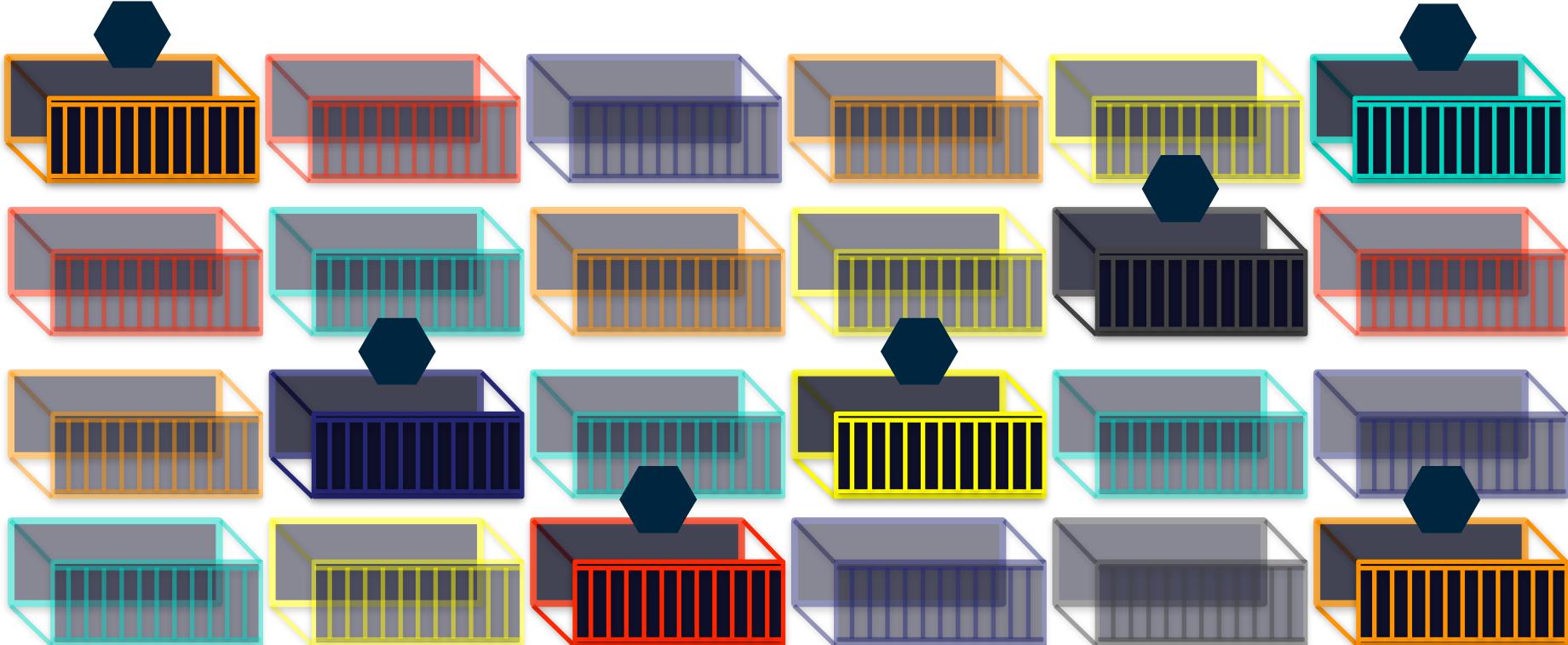
Apps, apps, and more apps



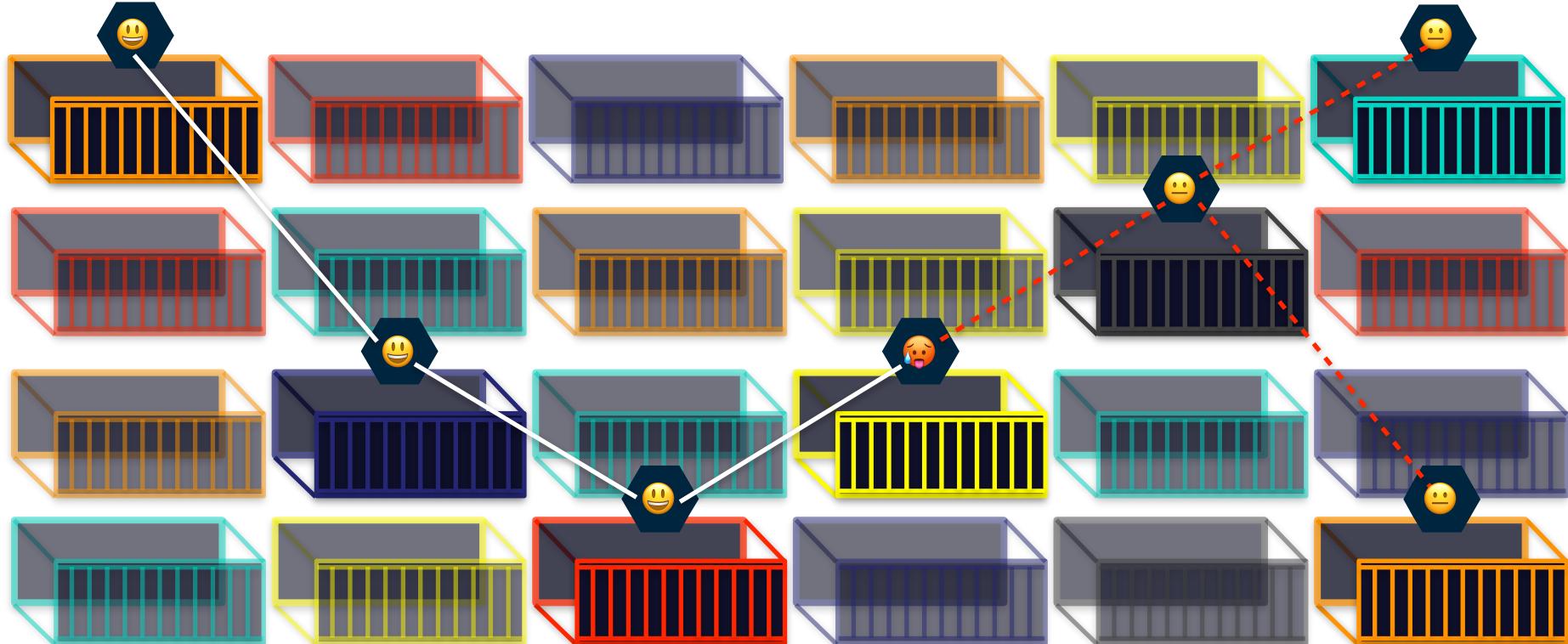
Apps Crash



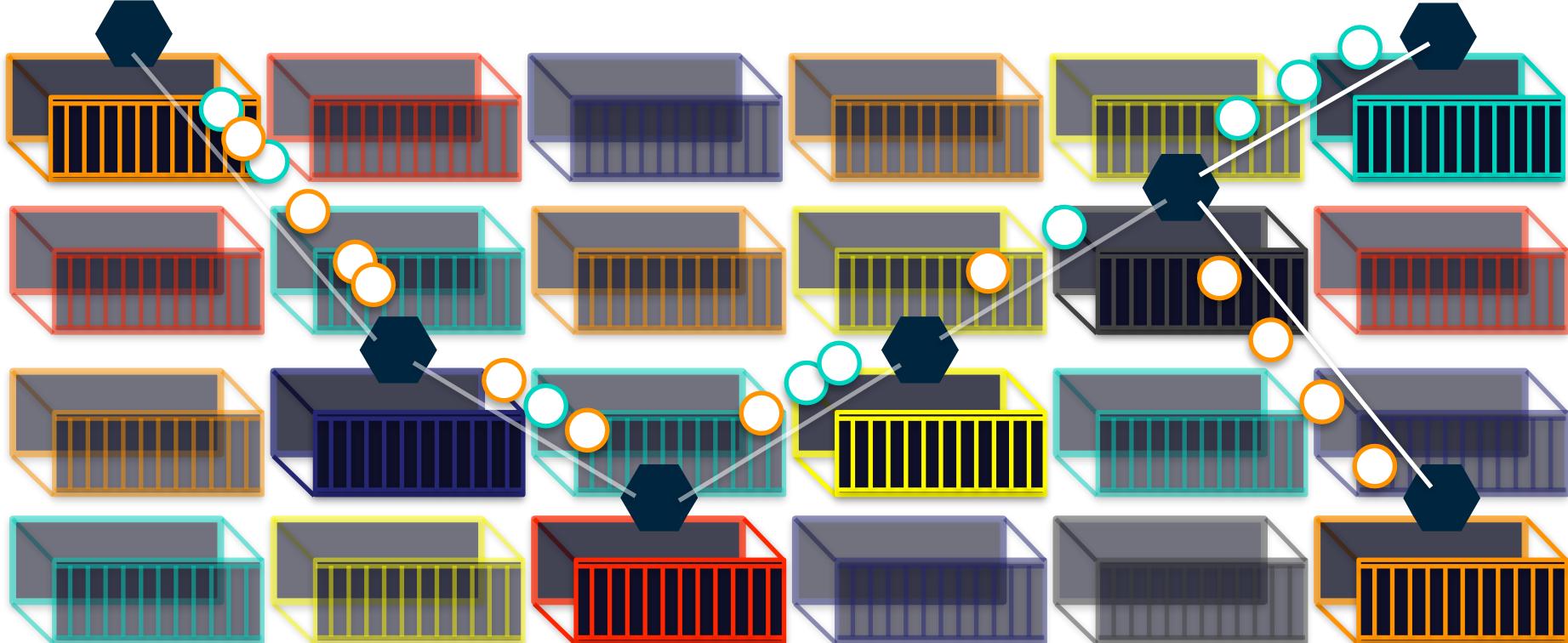
Apps are Hungry/Slow



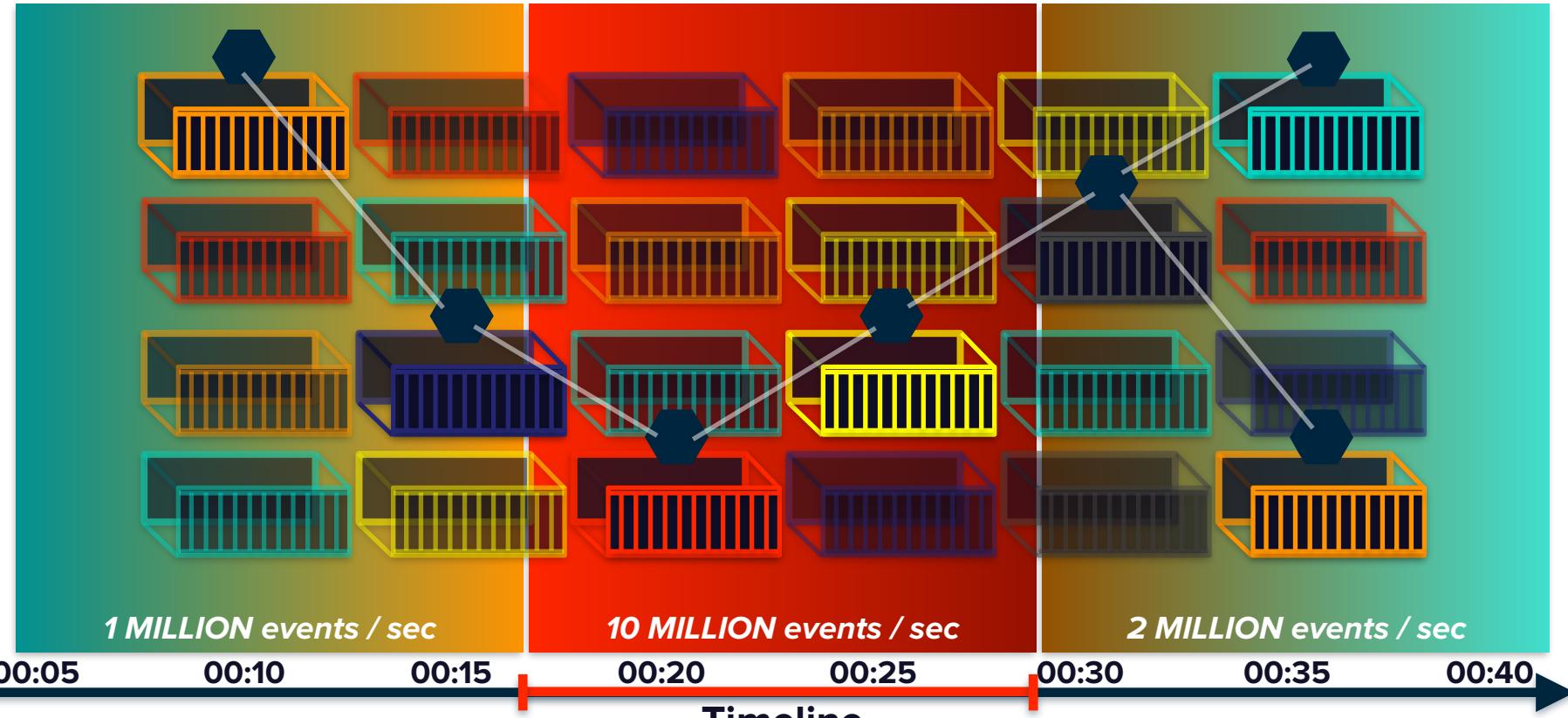
Apps in the Critical Path



Apps and Partitioned Data



Apps and Data Processing Volume



Apps and Multiple Platforms



Kubernetes

Cloud Foundry

**Deployment
Topologies**

Active-Passive

Active-Active

Orchestrating data-intensive applications at scale is *tough*



“Simple things should be simple, complex things should be possible.”

- Alan Kay

Spring Cloud Data Flow

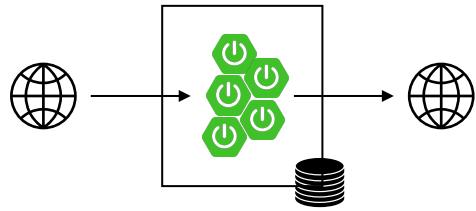
*A microservices based
Streaming and Batch
data processing in
Cloud Foundry and
Kubernetes*



Microservices for Data Processing

Spring Cloud Task

Short-lived Spring Boot microservices for batch data processing

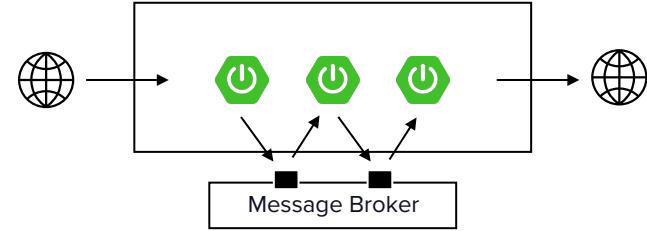


Use-cases:

- Scheduled data migration jobs
- Extract, Transform, and Load (ETL)
- Offline machine learning and model training

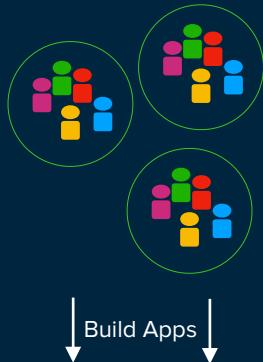
Spring Cloud Stream

Event-driven Spring Boot microservices for real-time data processing



Use-cases:

- Enterprise data integration (EAI/EIP)
- Event-driven architectures
- IoT and real-time predictive analytics



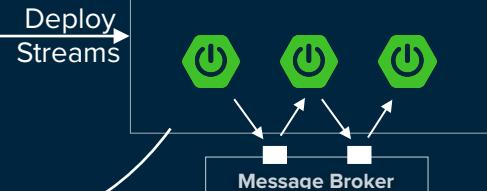
Spring Cloud Stream Spring Cloud Task



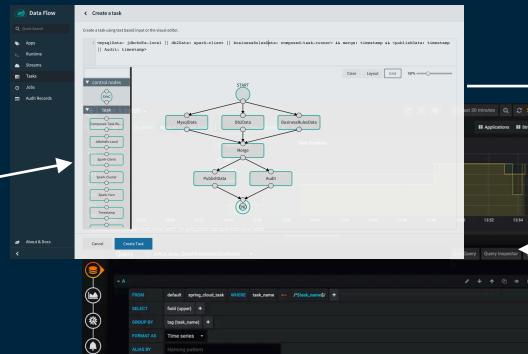
Spring Cloud Data Flow



Kubernetes / Cloud Foundry



Monitor Performance



Launch Batch Jobs

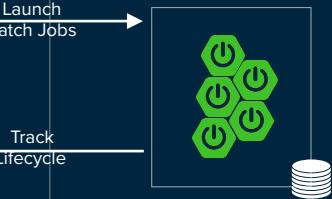
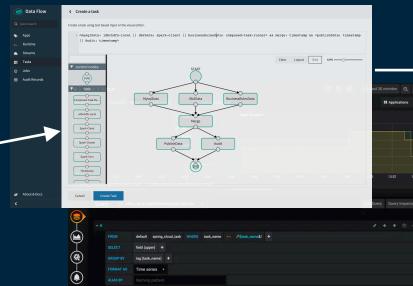
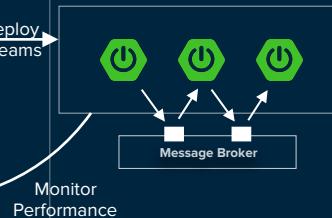
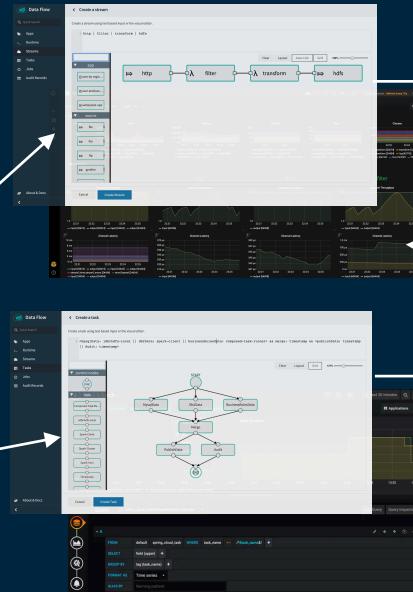


Track Lifecycle

Build

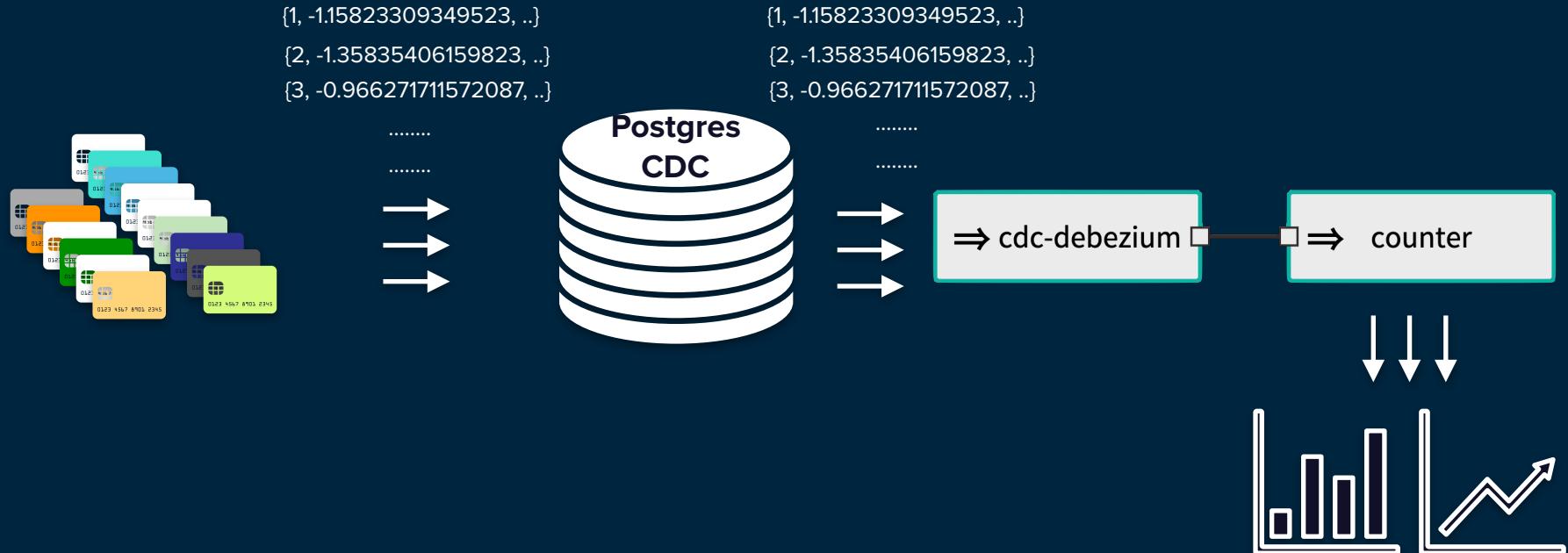


Run



Monitor

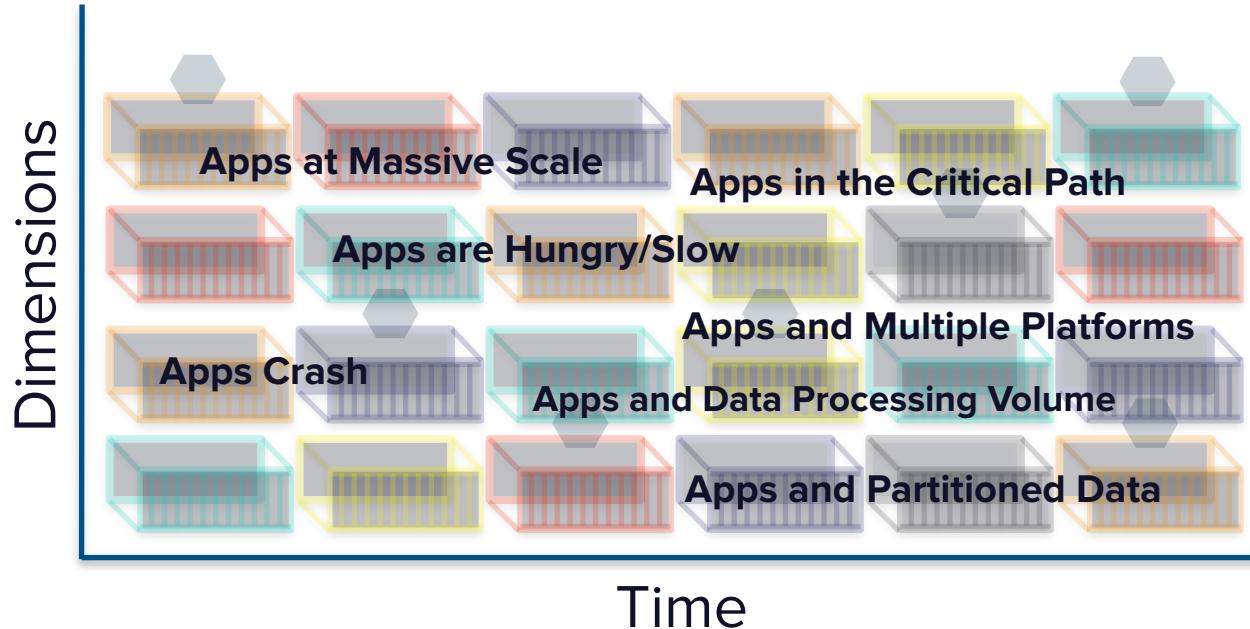
Demo 1: Credit Card Data + Change Data Capture



Let's Recap



Time Dimensions



Why Time Dimensions?

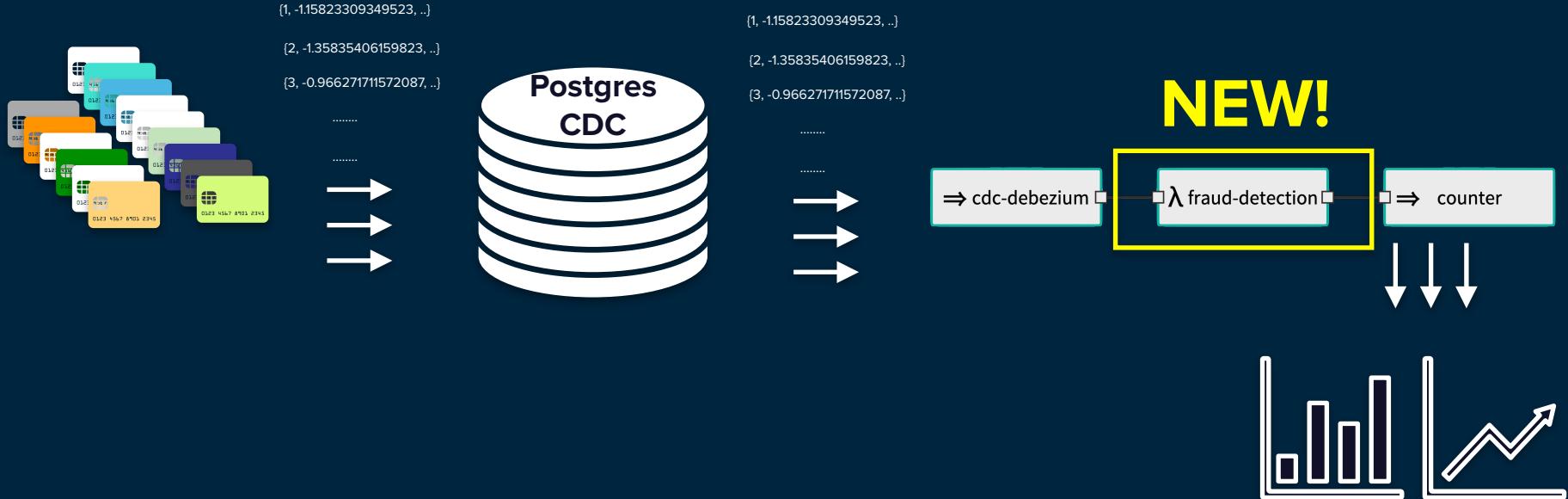
- Sequence of metrics data ordered by timestamp
- Identifiable by labels and tag dimensions
- Multi-dimensional time range aggregation
- Focus on the recent view of the metrics



Micrometer

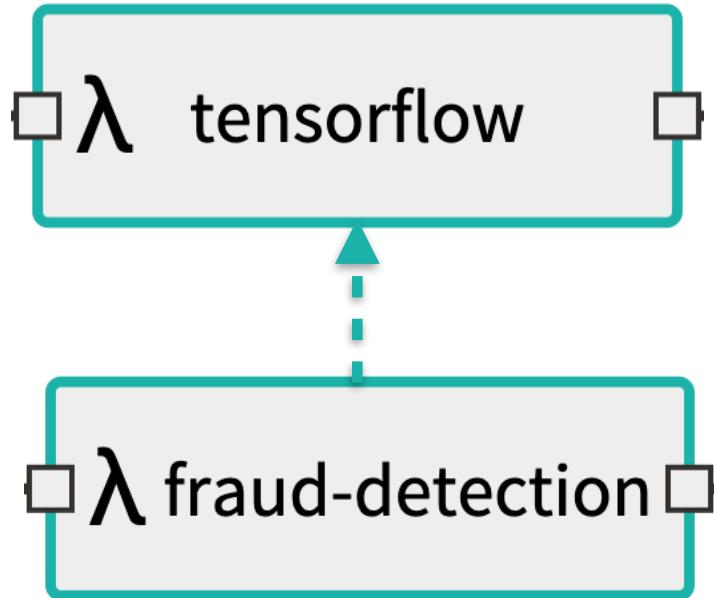
*A simple facade over
the instrumentation clients
for the most popular
monitoring systems*

Demo 2: Credit Card Fraud Detection

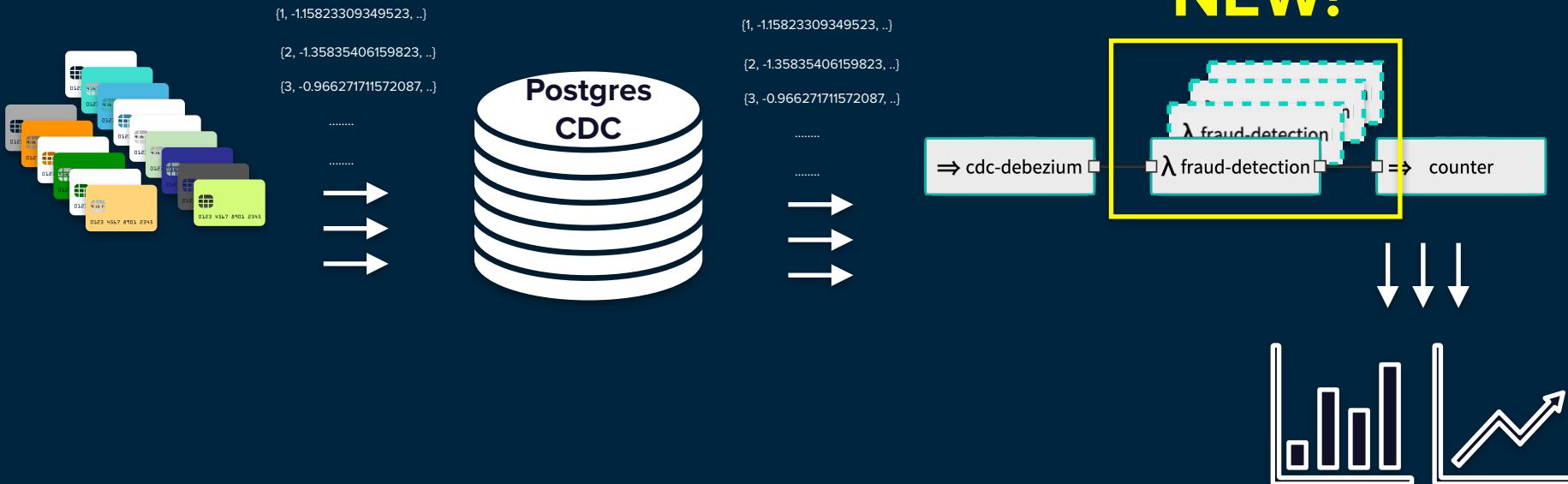


Fraud Detection Genesis

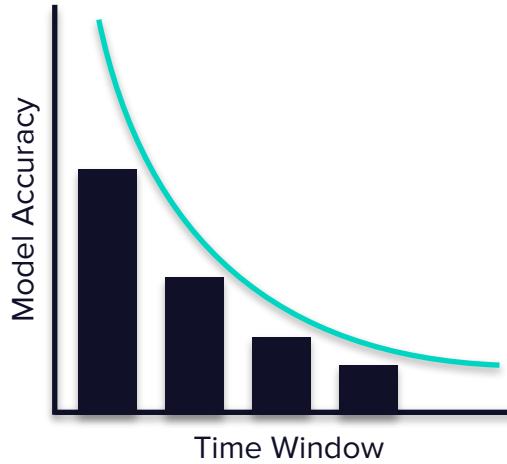
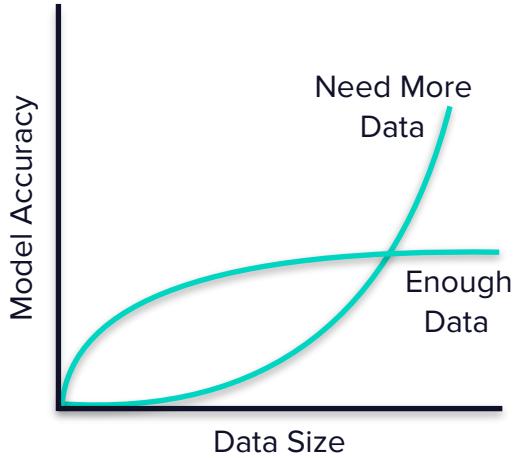
- Studied popular fraud-detection solutions from Kaggle
- Trained the dataset to detect fraudulent transactions
- Generated pre-trained model for real-time inferences
- Developed a Tensorflow based ***fraud-detection*** Spring Cloud Stream processor



Demo 3: Autoscale `fraud-detection` Processor



tl;dr: Model Training



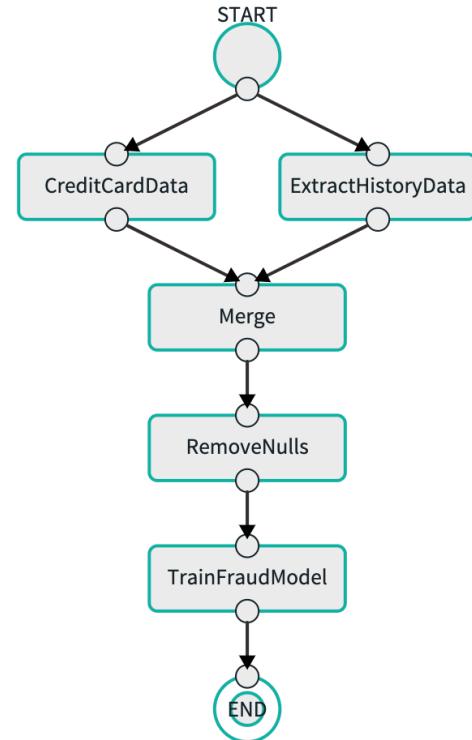
tl;dr: Model Training

More Data = More Accuracy

More Accuracy = More Time

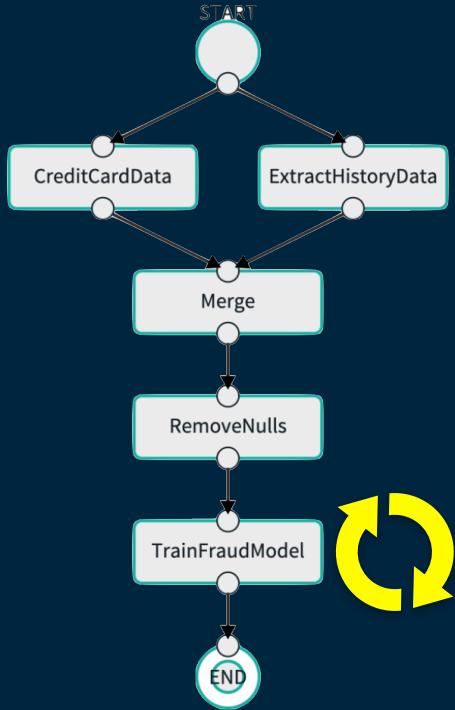
Requires a Dynamic Deployment Topology

Cloud-native Batch for Model Training



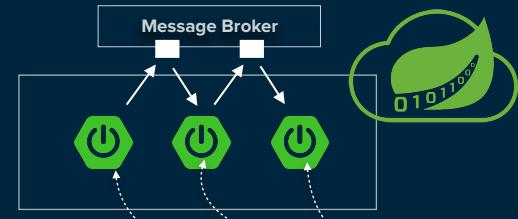
Each step is **short-lived**; in other words, each step runs as long as the business logic runs

Demo 4: Cloud-native Predictive Model Training



10,000ft Architecture

Streaming Data Pipeline



RSocket Bidirectional Connection

Batch Data Pipeline



Prometheus
RSocket Proxy



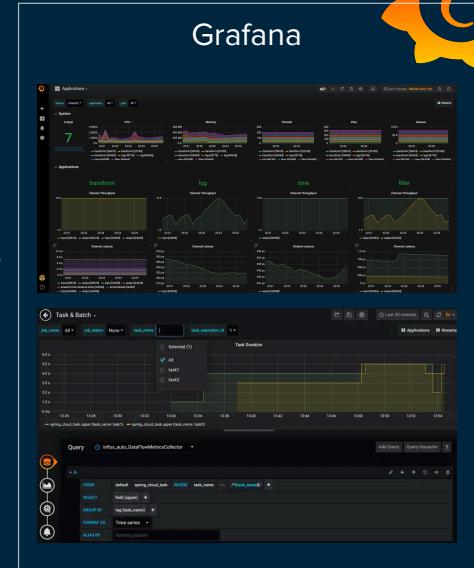
Scrape



Prometheus
TSDB



PromQL



Bringing Orchestration + Monitoring to Developers

Event-driven Streaming



+



=



Cloud-native **Batch**

Timelines / Next Steps

	Current Milestone	GA Milestone
Spring Cloud Task	2.2 M3	November 2019
Spring Cloud Stream	Hoxton / 3.0 M4	
Spring Cloud Data Flow	2.3 M2	

Metrics & Monitoring Roadmap

- *Monitoring stateful workloads (eg: Kafka Streams)*
- *SCDF-native `scale()` operation for metrics-driven autoscaling*

Resources

- Slides: <https://github.com/sabbyanandan/s1p2019>
- Fraud-detection in Action: <https://github.com/tzolov/cdc-fraud-detection-demo>
- SCDF Microsite: <https://dataflow.spring.io>
- SCDF Docs: <https://spring.io/projects/spring-cloud-dataflow#learn>
-

Q+A

SpringOne Platform

by Pivotal.



@christzolov / @sabbyanandan