



FLINK FORWARD

Berlin
11-13 Sep 2017

The premier conference
for Apache Flink®

Organized by **dataArtisans**

Some Practical Information



Network name: Flink Forward Berlin
Password: #FlinkForward



Twitter handle: @flinkforward
Hashtag: #FlinkForward



All talks will be recorded and can be found on our YouTube channel "Flink Forward" after the conference



Flink Fest today starting at 6.30 pm at Willner Brauerei

A big Thank You! to our Sponsors



RADICALBIT

Deep, Different, Disruptive.



eventador.io



getindata



THE
APACHE
SOFTWARE FOUNDATION

A big Thank You! to our Program Committee



Fabian Hueske
data Artisans



Kostas Kloudas
data Artisans



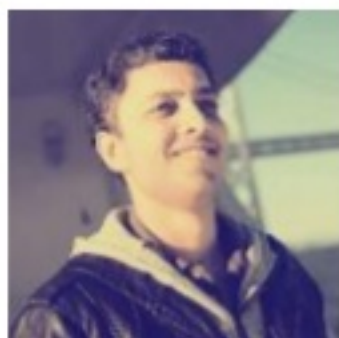
Dean Wampler
Lightbend



Tyler Akidau
Google



Xiaowei Jiang
Alibaba

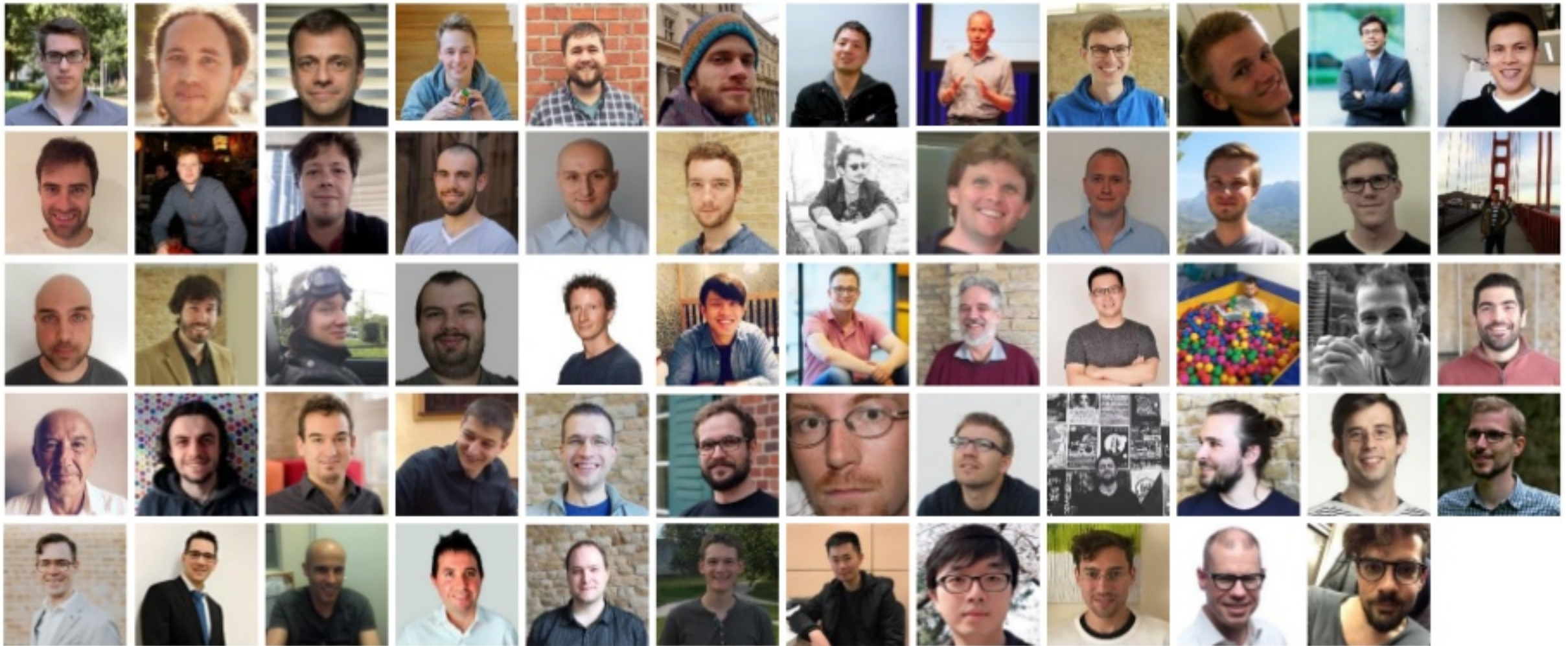


Chinmay Soman
Uber



Flavio Junqueira
Pravega by DellEMC

A big Thank You! to our Speakers



A big Thank You! to our Speakers



NETFLIX



bol.com



MESOSPHERE

PARALLEL
MACHINES

ING



dataArtisans

getindata

zalando



THE UNIVERSITY
of EDINBURGH



HanSight 瀚思



MTA SZTAKI
Hungarian Academy of Sciences
Institute for Computer Science and Informatics



GoDataDriven
proudly part of Rabla Group

noris network



redhat



accenture



UNIVERSITÄT LEIPZIG

ResearchGate



TECHNISCHE
UNIVERSITÄT
DRESDEN



German
Research Center
for Artificial
Intelligence

relayr.
enabling business outcomes



A warm Welcome to all of You!

We hope you enjoy the conference

An example of conference and community @ Flink Forward 2016



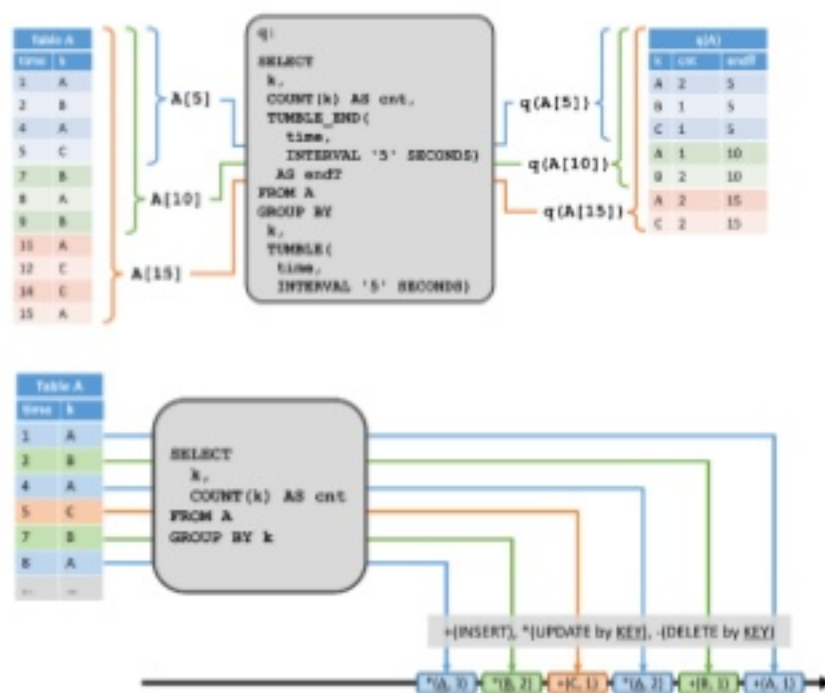
Last minute
talk cancellation



Impromptu discussion
session on
Streaming and SQL



Stream SQL in Flink (and possibly others)





A Flink Review of 2017

@StephanEwen
dataArtisans

Flink Forward, Berlin, 2017



2,100,000,000,000

That the number of events processed by
users attending today that filled out the survey
(~50% of you here)



Use case highlights
from this year so far...



@ UBER



- Apache Flink deployed as a streaming platform service
- Billions messages / petabytes of data per day
- Incrementally realizing more and more services
 - Growth: "how much did we earn in SF in the last 5 mins"
 - Intelligent alerting: Ban driver/rider if suspicious activity
 - Intelligent forecasting: Increase accuracy of ETA models



@



Blink in Alibaba Production

- ✓ In production for almost one year
- ✓ Run on thousands of nodes
 - hundreds of jobs
 - The biggest cluster is more than 1000 nodes
 - the biggest job has 10s TB states and thousands of subtasks
- ✓ Supported key production services on last Nov 11th, China Single's Day
 - China Single's Day is by far the biggest shopping holiday in China, similar to Black Friday in US
 - Last year it recorded \$17.8 billion worth of gross merchandise volumes in one day
 - Blink is used to do real time machine learning and increased conversion by around 30%




@ **NETFLIX**



- Various use cases
 - Example: Stream ingestion, routing
 - Example: Model user interaction sessions
- Mix of stateless / moderate state / large state
- Stream Processing as a Service
 - Launching, monitoring, scaling, updating

4000+ Kafka brokers, 50+ clusters

100's of Data Streams (Flink Jobs)

3700+ Docker containers running  Flink

1400+ nodes with 22K+ cpu cores



@

ING



Credit card
transactions



Notifications
and alerts



Evolving fraud
models built by
data scientists

Detecting fraud in real time

As fraudsters get better, need
to update models without
downtime

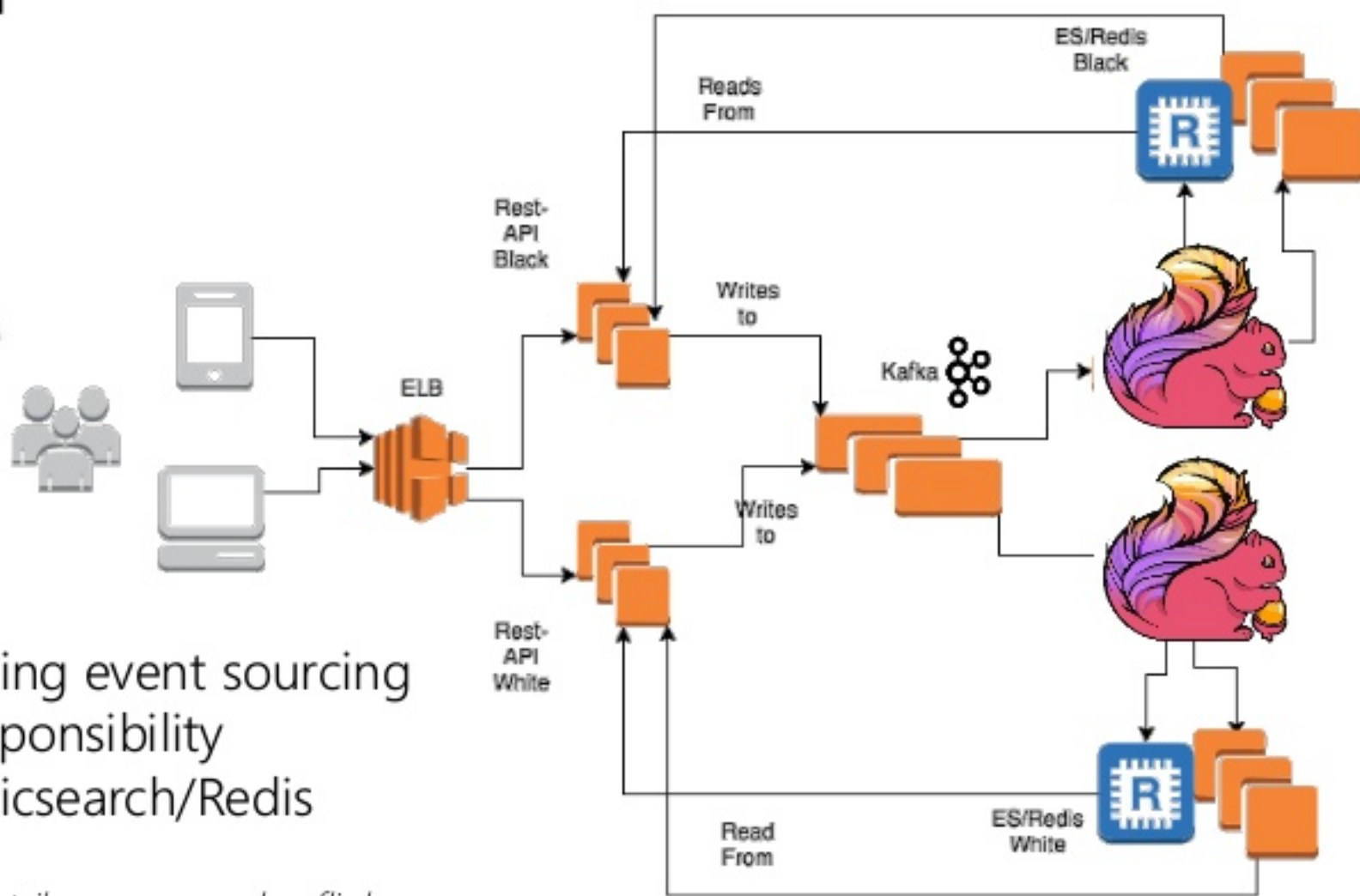
Live 24/7 service



@



THE SOCIAL NETWORK
FOR PETROLHEADS



Social network implemented using event sourcing and CQRS (Command Query Responsibility Segregation) on Kafka/Flink/Elasticsearch/Redis

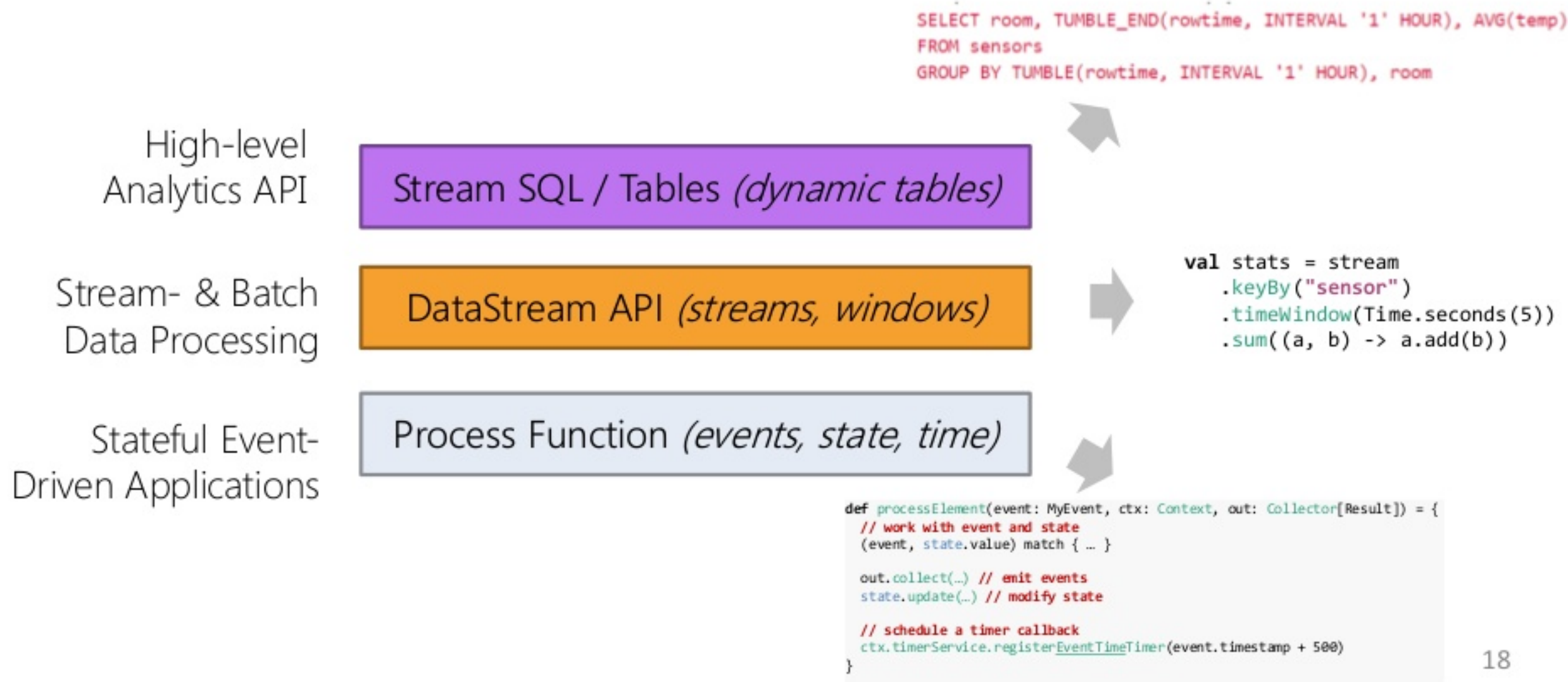
More: <https://data-artisans.com/blog/drivetribe-cqrs-apache-flink>

Streaming and Streaming Processing



- First wave for streaming was **lambda architecture**
 - Aid batch systems to be more real-time
- Second wave was **analytics (real time and lag-time)**
 - Based on distributed collections, functions, and windows
- The next wave is **much broader:**
A **new architecture** for a unified approach to **data analytics** and **event-driven applications**

Flink's APIs over the last year



Some Flink features in progress



Development and Deployment

Deployment
Versatility

Massive dependency
reduction

RESTified
ops APIs

Hadoop-free Flink

Engine

Auto-tuning
network latency

Faster checkpoints
and restore

Dynamic Resources

APIs & Languages

SQL / CEP integration

SQL connectors

SQL performance

Flink/Beam/Python

Connectors & Ecosystem

Pravega Integration

Kinesis

Kafka exactly-once

Eventually Consistent
FileSystems

Mesos

Kubernetes

(more)

Building a Stream Processing Infrastructure

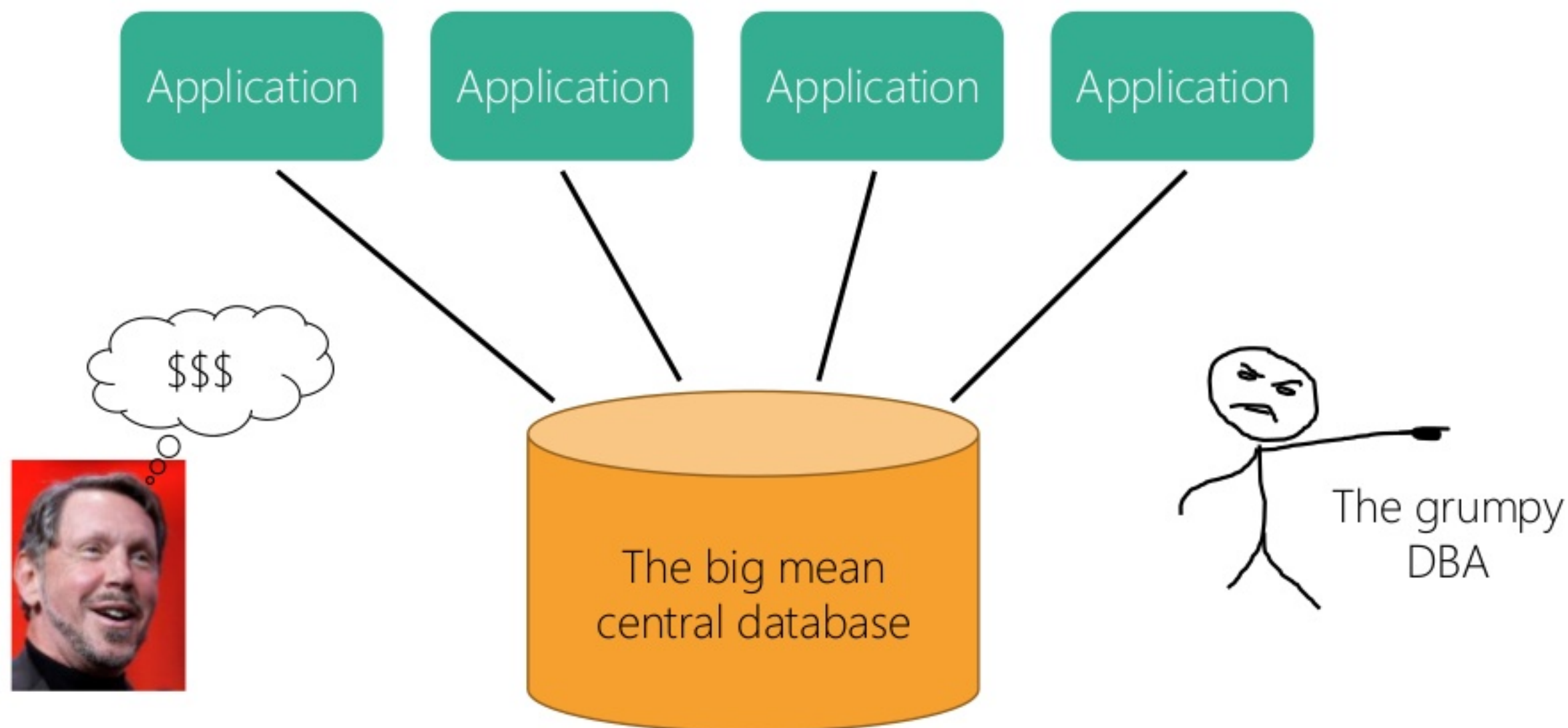


Flink Forward, Berlin, 2017

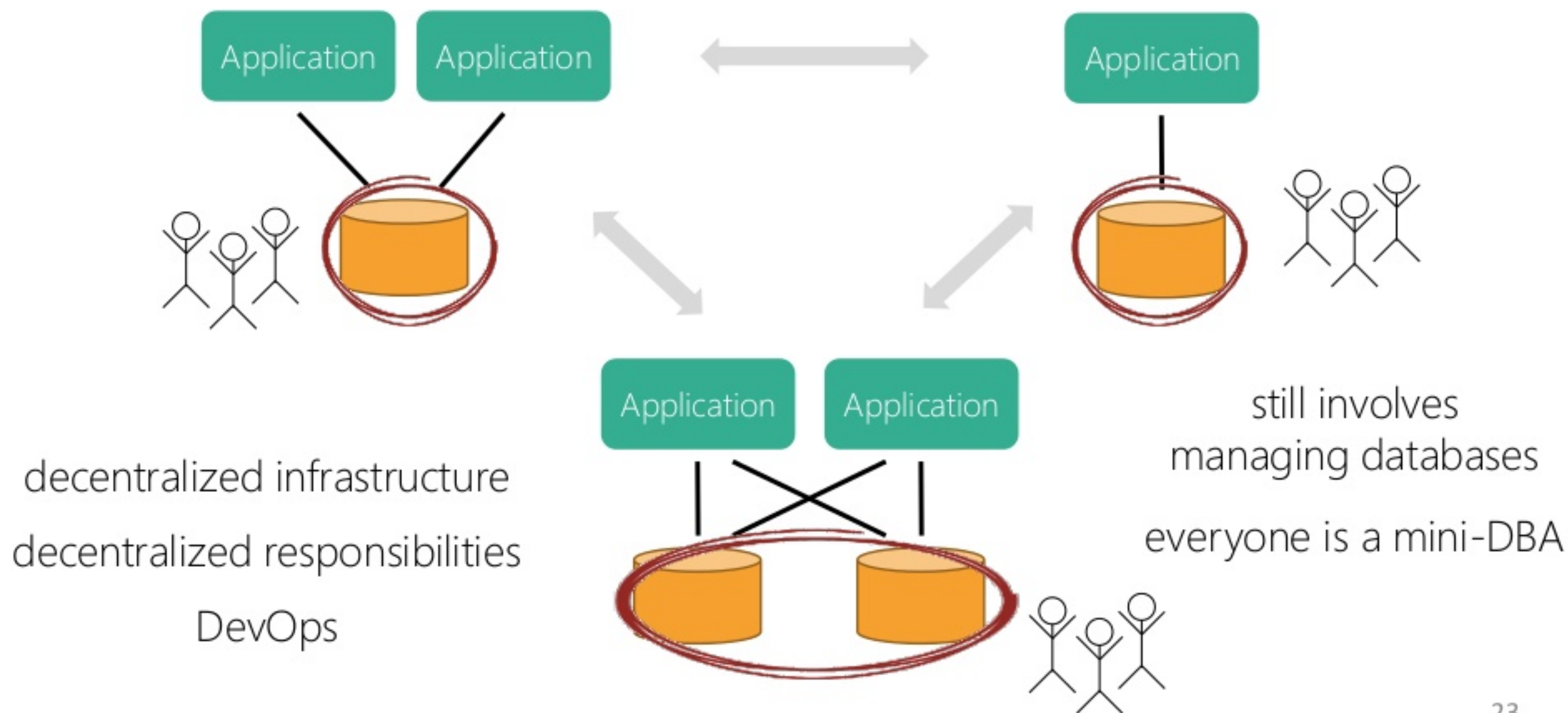


Let's take a step back...

Good old centralized architecture



Enter Microservices...



Thinking about State and Databases



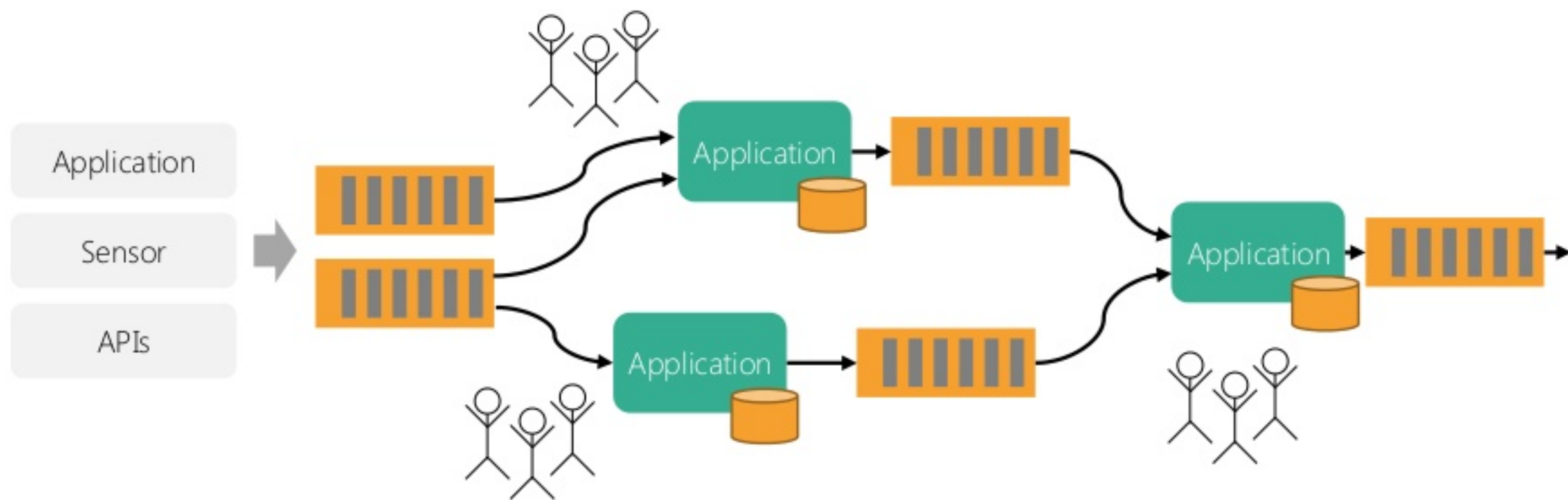
Kudos to Kiki Carter
for the Broccoli
Metaphor

... and Stateful Stream Processing



very simple: **state is just part of the application**

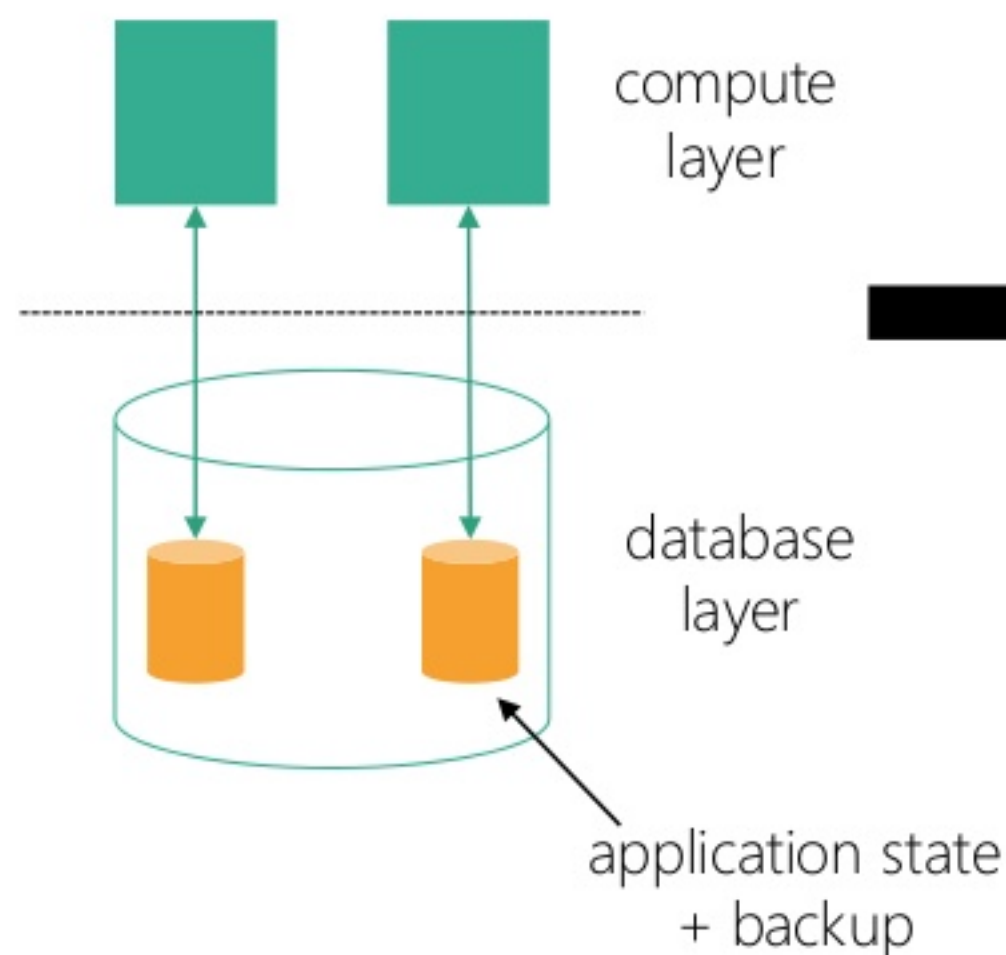
micro services on steroids!
encourages to build even more
lightweight and specialized apps



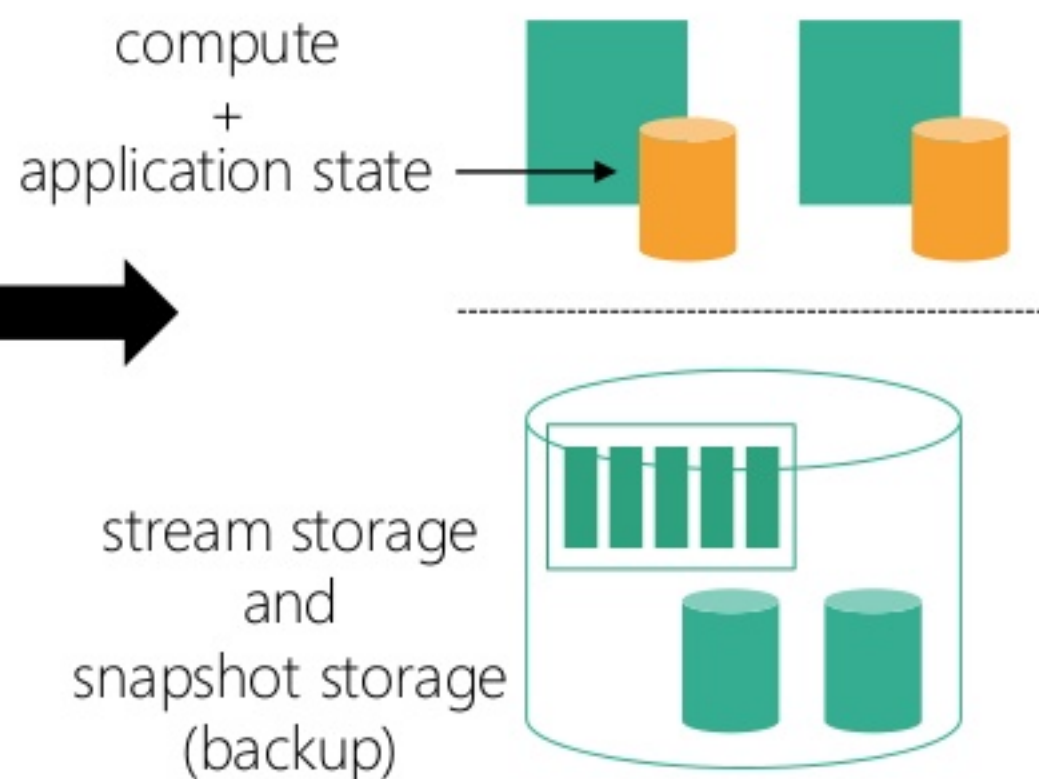
Compute, State, and Storage



Classic tiered architecture



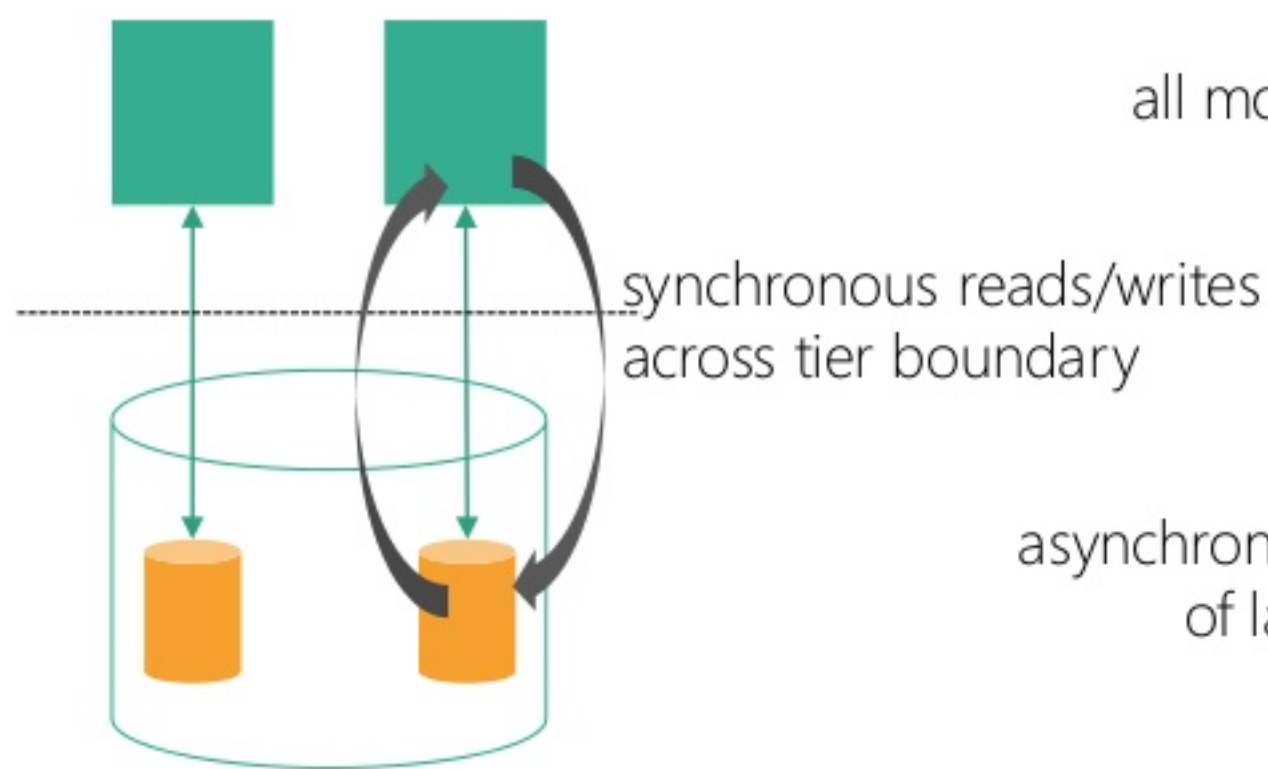
Streaming architecture



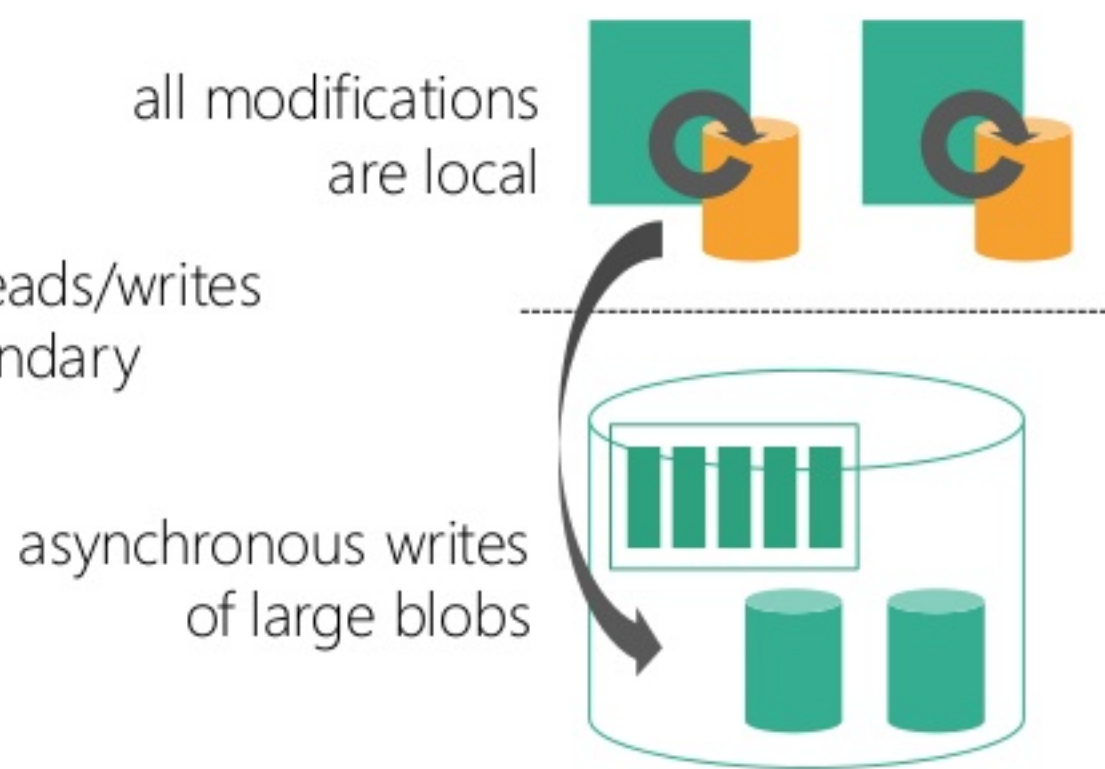
Performance



Classic tiered architecture



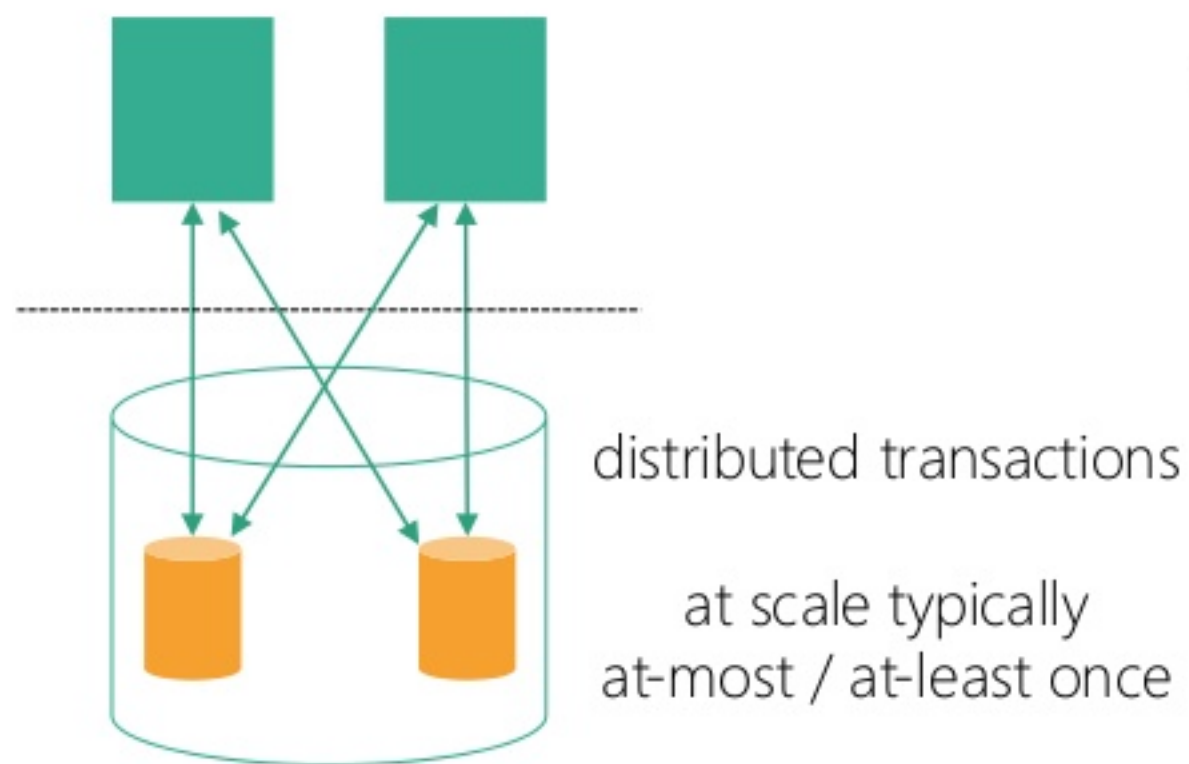
Streaming architecture



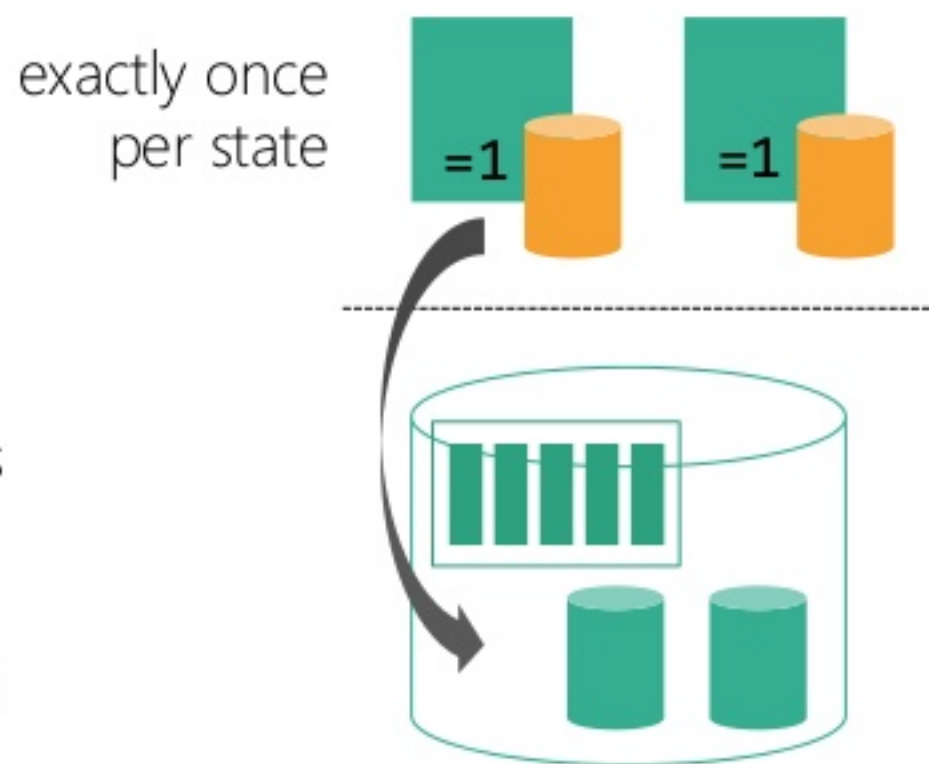
Consistency



Classic tiered architecture



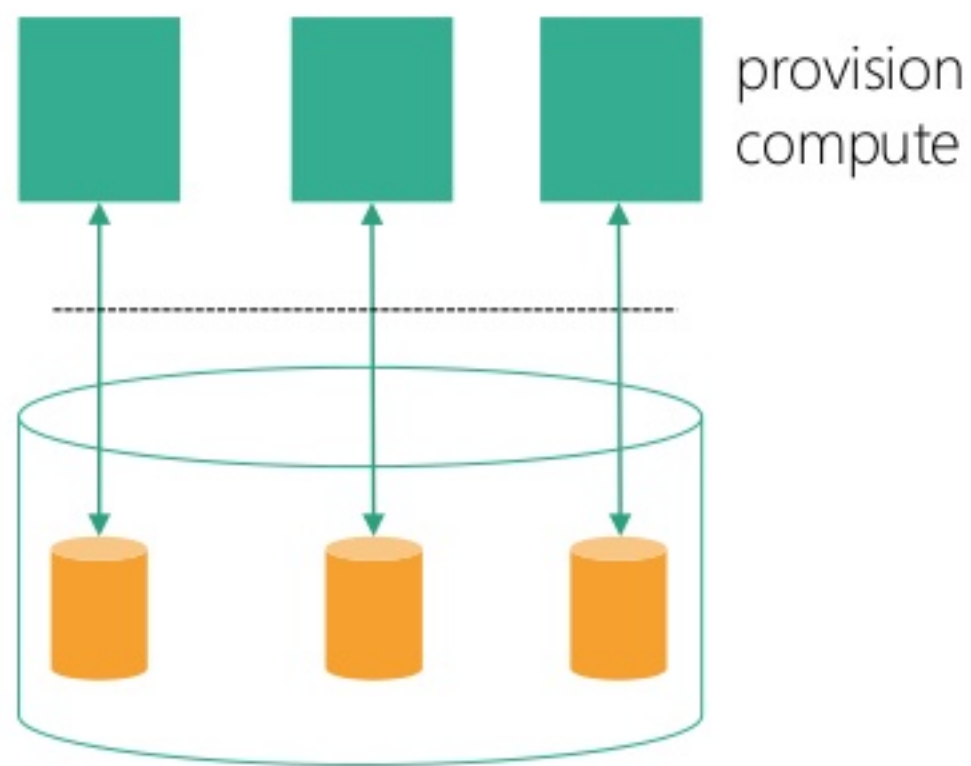
Streaming architecture



Scaling a Service

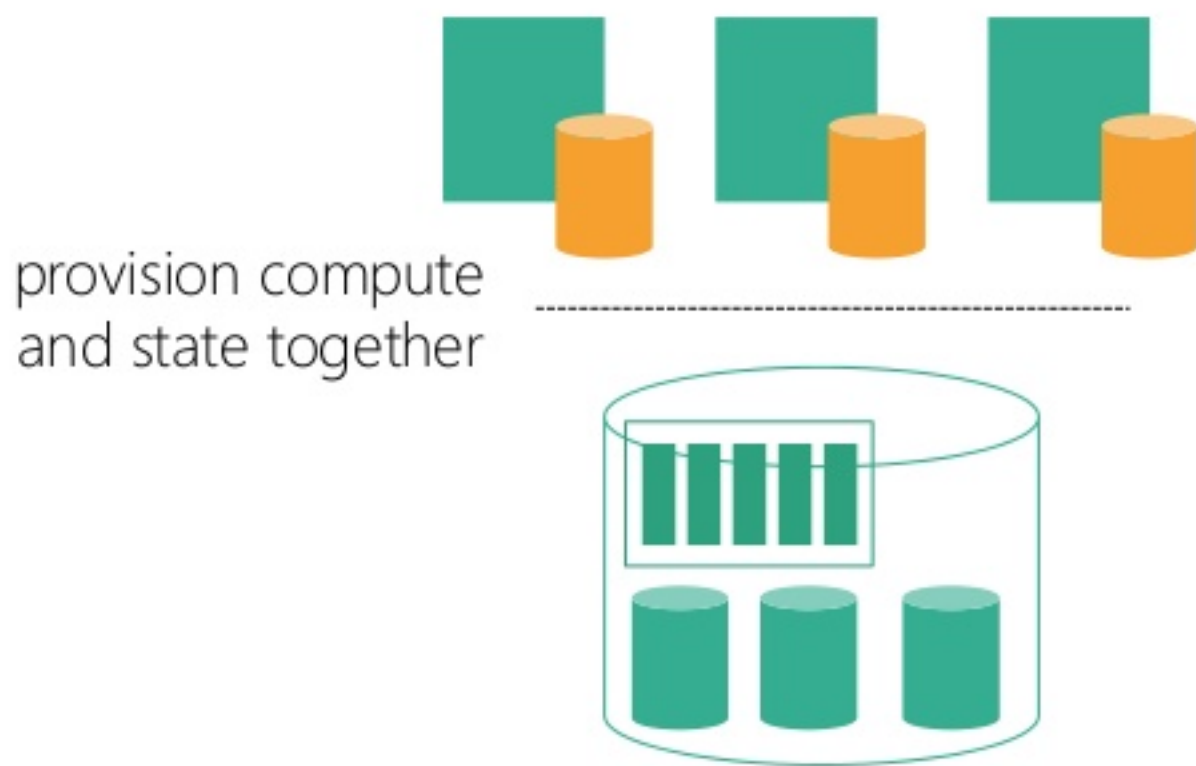


Classic tiered architecture



separately provision additional
database capacity

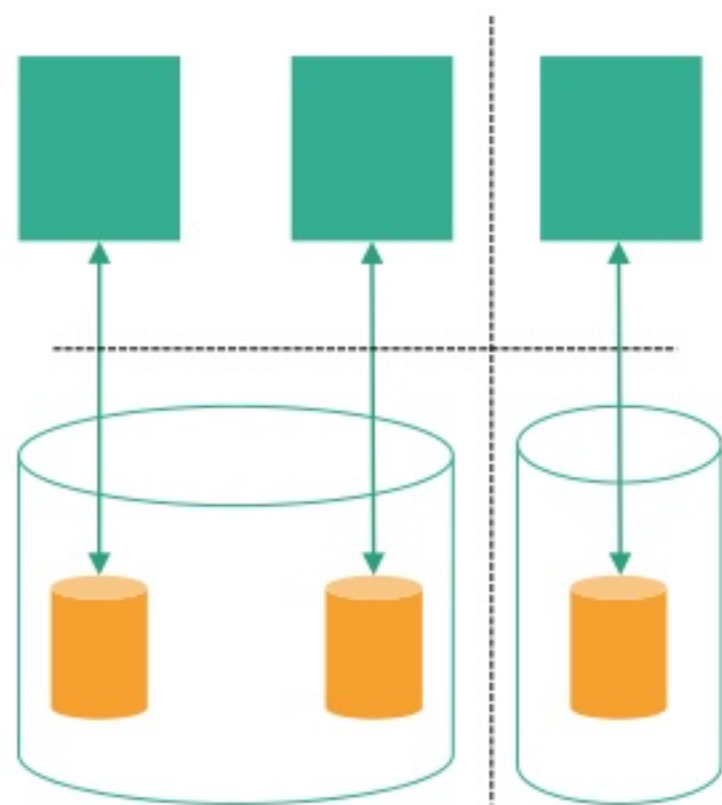
Streaming architecture



Rolling out a new Service

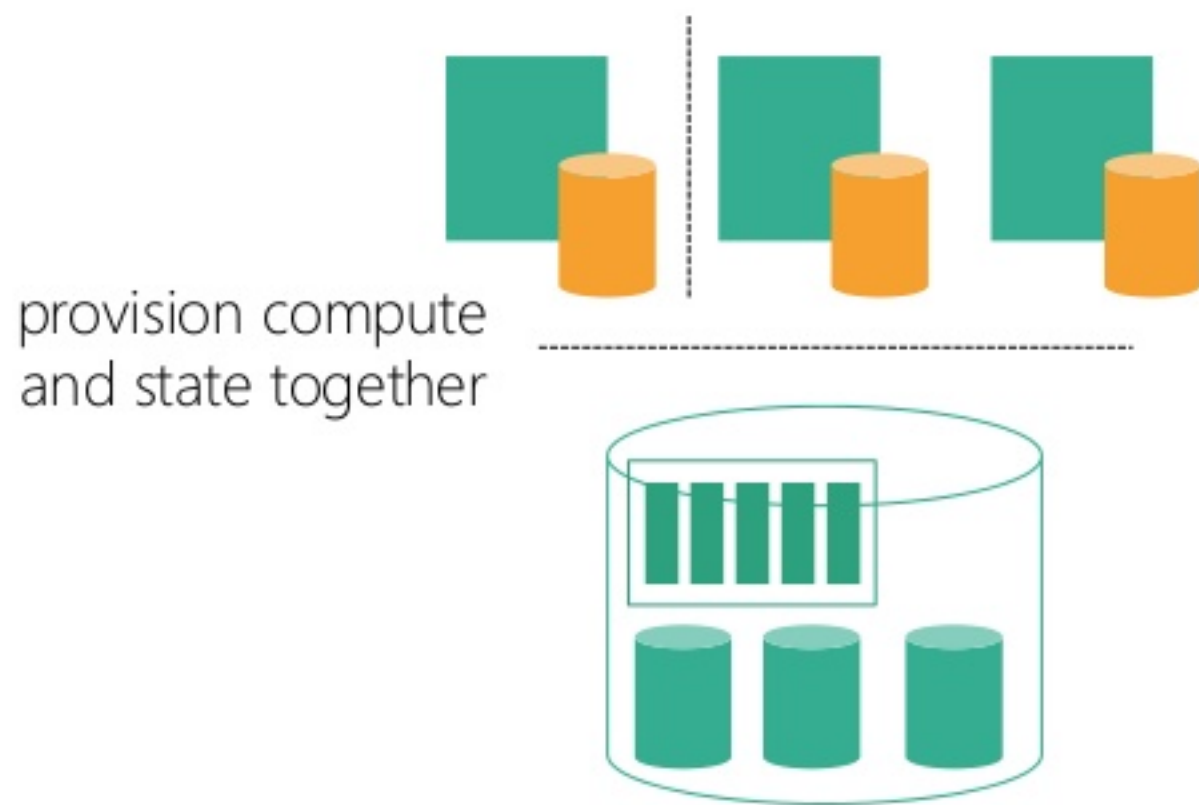


Classic tiered architecture



provision a new database
(or add capacity to an existing one)

Streaming architecture



provision compute
and state together

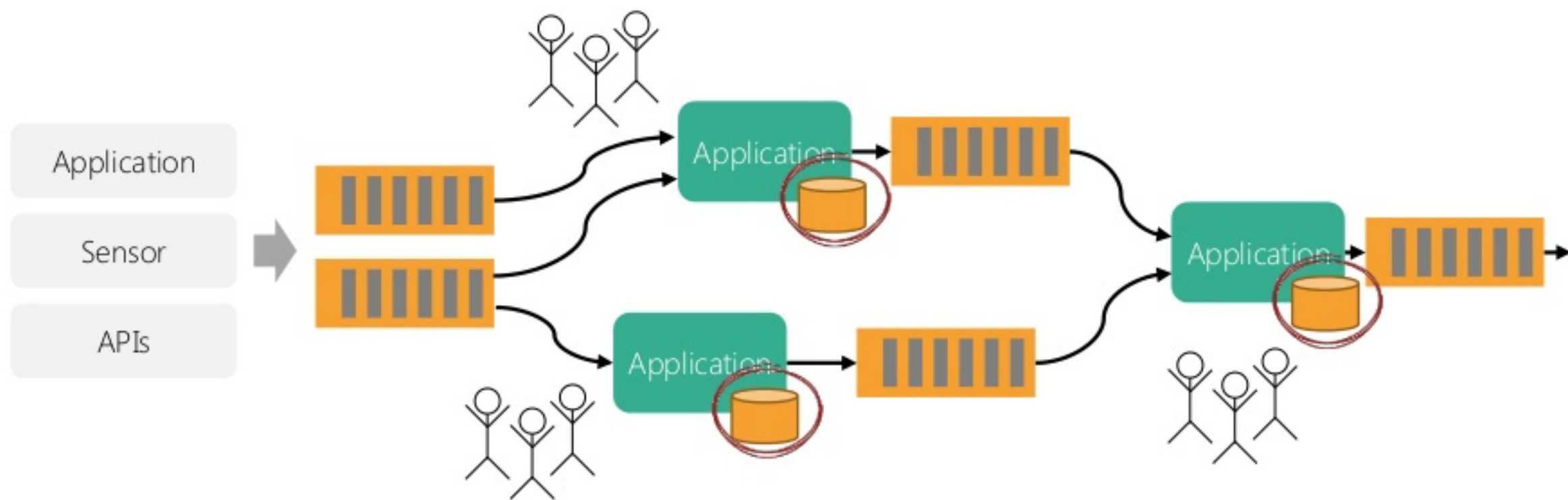
simply occupies some
additional backup space

Stateful Stream Processing



very simple: state is just part
of the application

micro services on steroids!
encourages to build even more
lightweight and specialized apps





Does that solve
everything?

(of course not)

How does one easily do...?



- consistent stateful upgrades
 - application evolution and bug fixes
- migration of application state
 - cluster migration, A/B testing
- re-processing and reinstatement
 - fix corrupt results, bootstrap new applications
- state evolution (schema evolution)



Consistent Distributed
Snapshots

(a.k.a. savepoints)

Continuous Applications

versioning, upgrading,
rollback, duplicating,
migrating, ...



PLATFORM

Consistent Distributed
Snapshots

(a.k.a. savepoints)

Continuous Applications

versioning, upgrading,
rollback, duplicating,
migrating, ...

Stateful Stream Processing for the Real-time Enterprise

Announcing dA Platform 2 with Application Manager and open source Apache Flink®

[Learn more](#)

The screenshot displays the dA Platform 2 Application Manager interface. On the left is a dark sidebar with the 'dA' logo and navigation links for 'Applications', 'Deployment targets', and 'Settings'. The main content area has a breadcrumb trail: 'Applications > Application: app2 > Deployment: Fraud Detection'. The title 'Fraud Detection' is prominently displayed, with a status indicator showing two 'RUNNING' states (Desired and Current). Below the title is the deployment ID: 'ID: 093c1309-c880-41af-825d-da08176a2024, Application: app2, Deployed to: Kubernetes'. A row of action buttons includes Start, Suspend, Cancel, Savepoint, Upgrade, Fork, Metrics, Logs, and Delete. Below these are tabs for Overview, Event Log, Jobs, and Savepoints. The 'Overview' tab is active, showing two panels: 'Deployment' and 'Deployment Target'. The 'Deployment' panel lists 'Run' (3), 'Revision' (0), and 'Started' (2017-08-24, 18:01:17, 3 minutes ago). The 'Deployment Target' panel shows 'ID' (57b4c290-73ad-11e7-8cf7-a6006ad3dba0), 'Name' (Kubernetes), and 'Configuration' (environment: production).

Applications > Application: app2 > Deployment: Fraud Detection

Fraud Detection

ID: 093c1309-c880-41af-825d-da08176a2024, Application: app2, Deployed to: Kubernetes

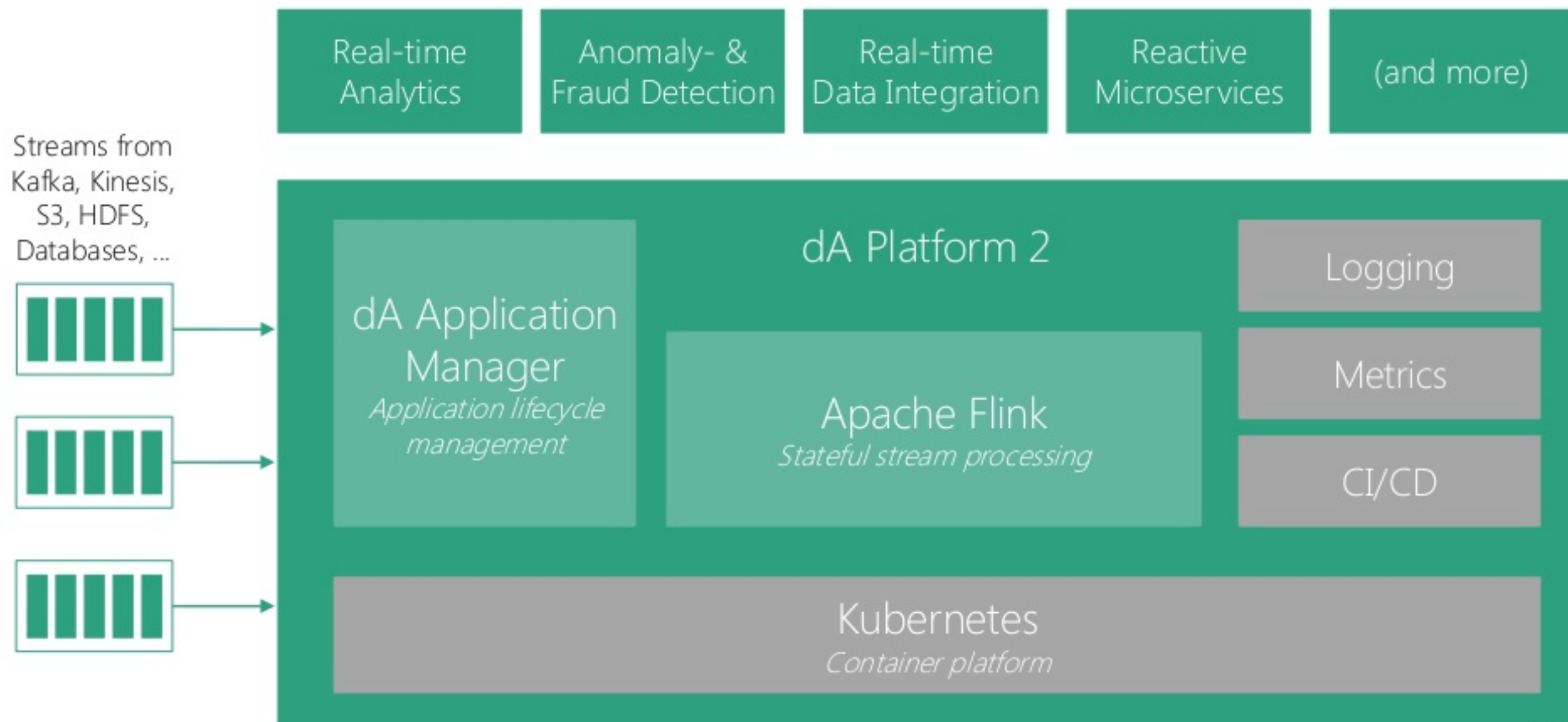
[Start](#) [Suspend](#) [Cancel](#) [Savepoint](#) [Upgrade](#) [Fork](#) [Metrics](#) [Logs](#) [Delete](#)

[Overview](#) [Event Log](#) [Jobs](#) [Savepoints](#)

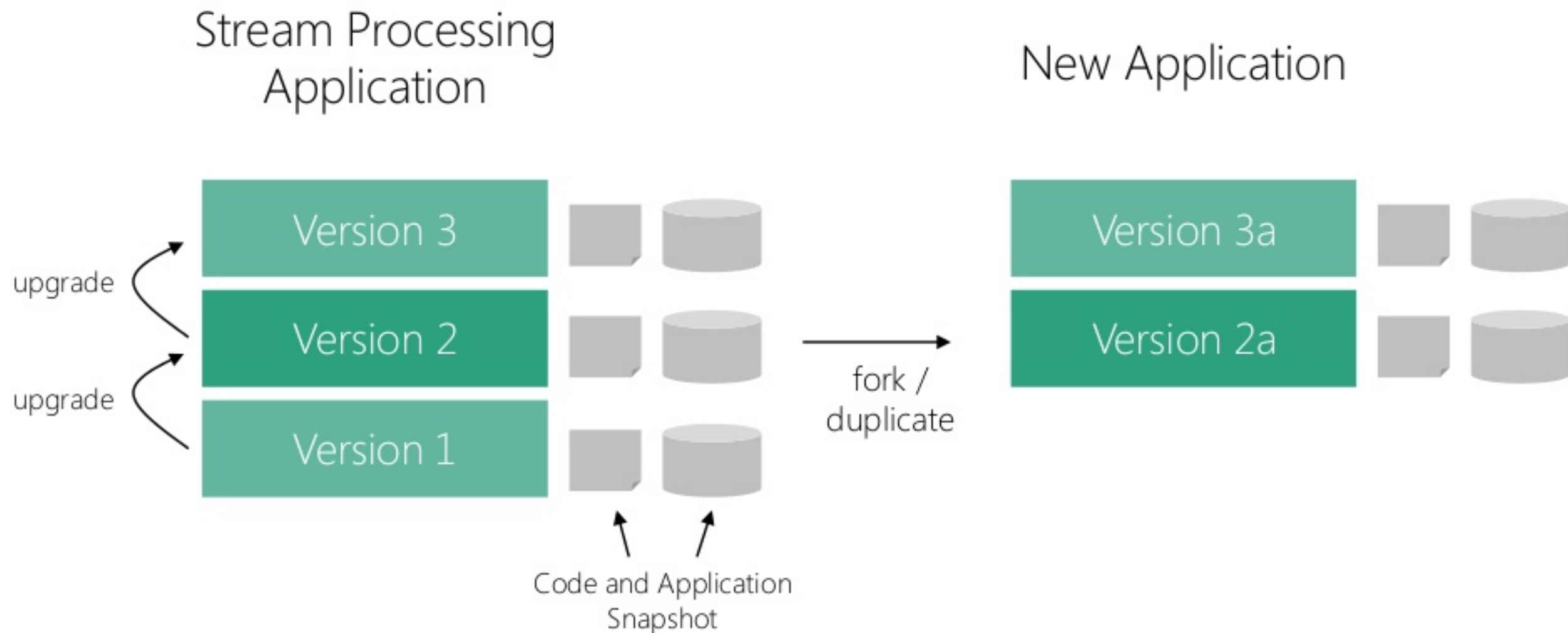
Deployment	
Run	3
Revision	0
Started	2017-08-24, 18:01:17 3 minutes ago

Deployment Target	
ID	57b4c290-73ad-11e7-8cf7-a6006ad3dba0
Name	Kubernetes
Configuration	environment: production

The dA Platform Architecture



Versioned Applications, not Jobs/Jars



Deployments, not Flink Clusters



Threat Metrics
App. Testing

Fraud Detection
App. Testing

Activity Monitor
Application

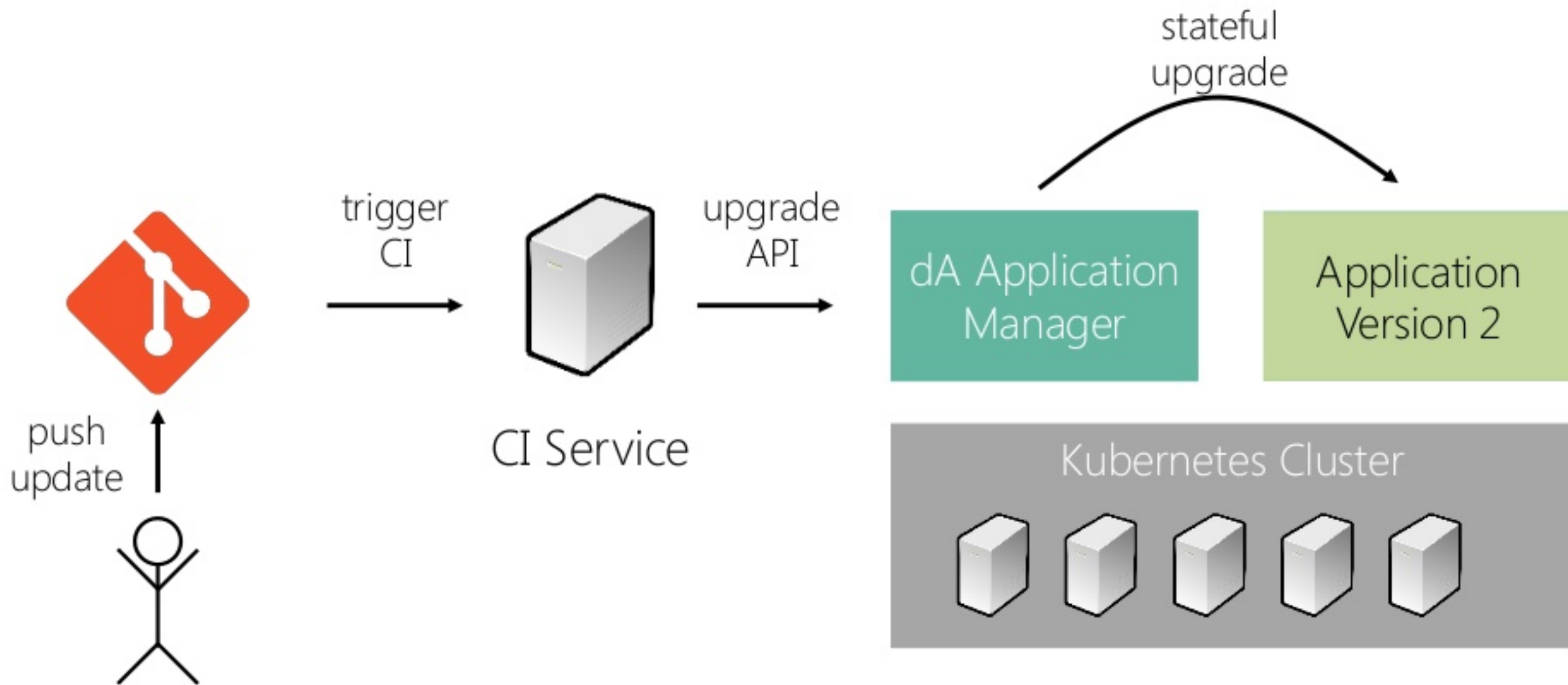
Testing / QA Kubernetes Cluster



Production Kubernetes Cluster

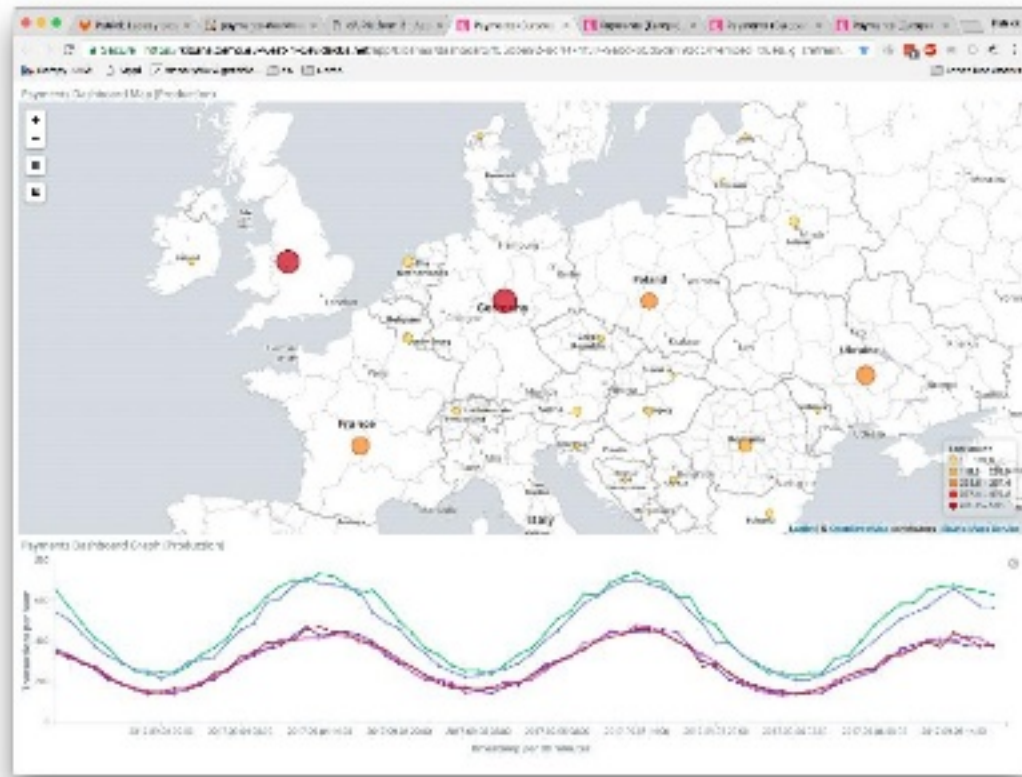
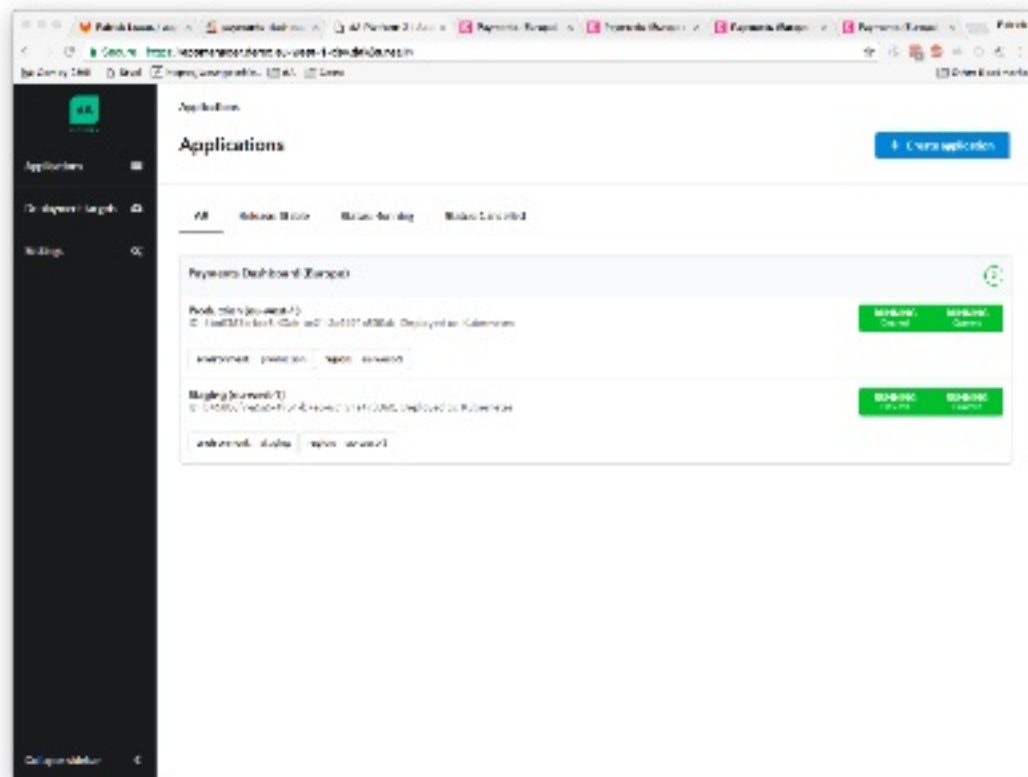


Hooks for CI/CD pipelines





Demo Time!



Payments Dashboard

Sign up for the early access program!



More information about dA Platform 2

Want to learn more about dA Platform 2? Or interested in early access? We'd love to hear from you.

First Name

Last Name

Email

Company

Role

SIGN UP

<https://data-artisans.com/da-platform-2>



Learn more at

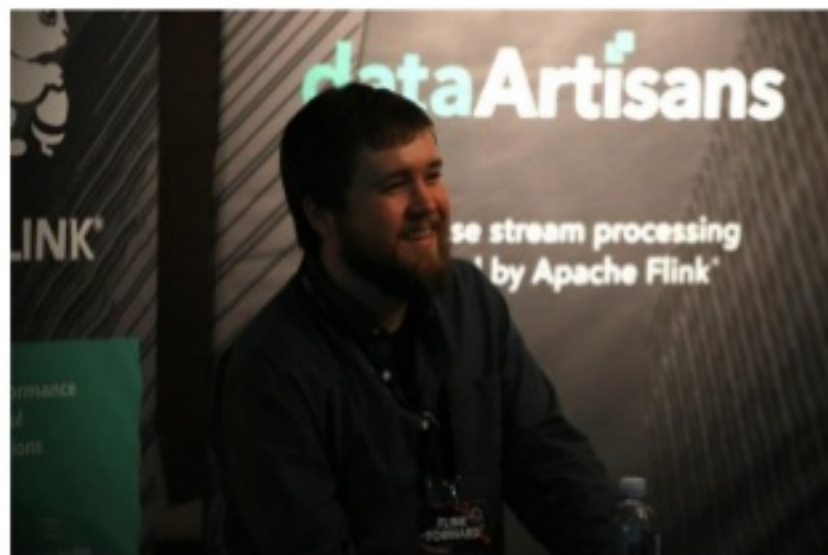
TIME

DAY 2 - SEPTEMBER 12
11:00 AM - 11:40 AM

LOCATION

KESSELHAUS

Talk at 11am today
@ Kesselhaus



At the booth

dataArtisans



Enjoy the Conference!