Introducing dA Platform 2

Including Application Manager and Apache Flink®

Patrick Lucas and
Robert Metzger

dataArtisans

What we’ve learned over the last three years

Stateful Stream Processing with Flink

- As of today, Flink is the most advanced stateful stream processor available
- **Stateful streaming is a hot topic, and it's here to stay**

Features:

- Unified Batch & Streaming SQL
- Complex Event Processing Library
- Rich Windowing API
- Event-Time semantics
- Versatile APIs
- Exactly-once fault tolerance
- Queryable State
- Fully scalable and distributed processing

Integrations:

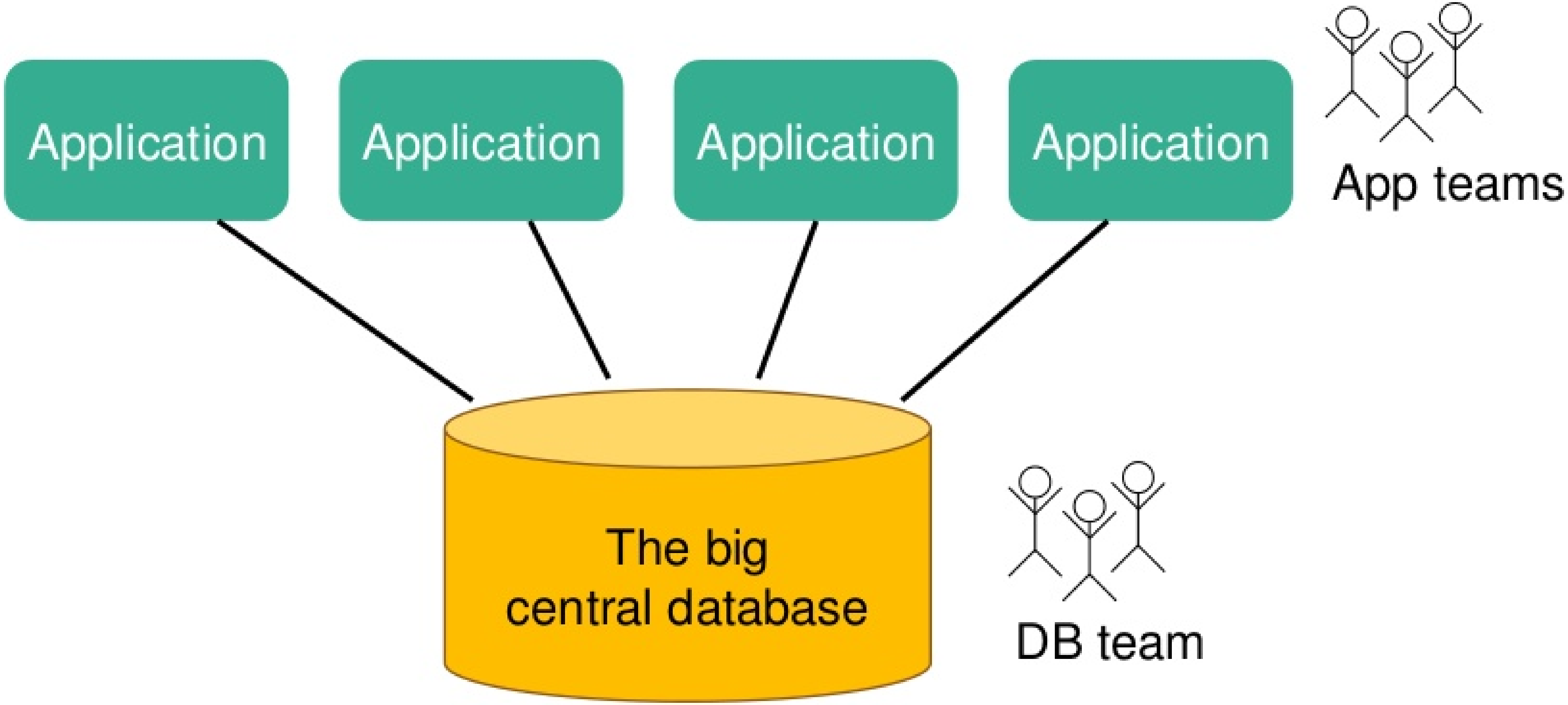
- Apache Kafka (with exactly-once)
- Apache Hadoop YARN
- Apache Mesos (and DC/OS)
- AWS Kinesis
- Docker & Kubernetes
- ElasticSearch & Cassandra & HBase
- Legacy message queues
- Hadoop-supported file-systems
- Apache Beam Runner

Operational Features:

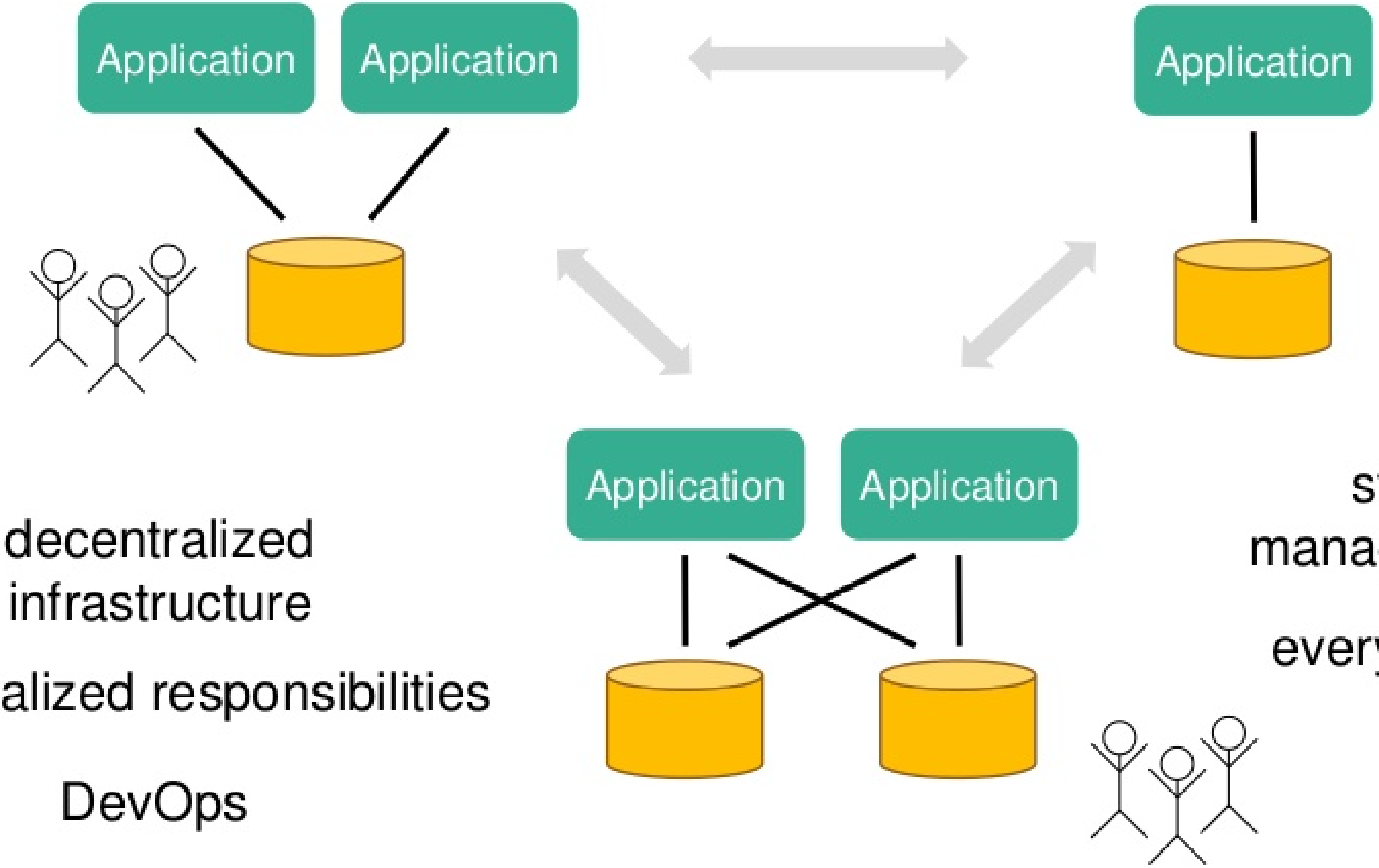
- Incremental Checkpointing
- Pluggable, fully asynchronous Statebackends
- RocksDB file-based state backend
- High-Availability
- Savepoints
- Kerberos Authentication
- SSL data encryption
- Backwards-compatibility for state and APIs
- Metrics

Architectures are changing …

From centralized architectures …



… to Microservices …



decentralized infrastructure

decentralized responsibilities

DevOps

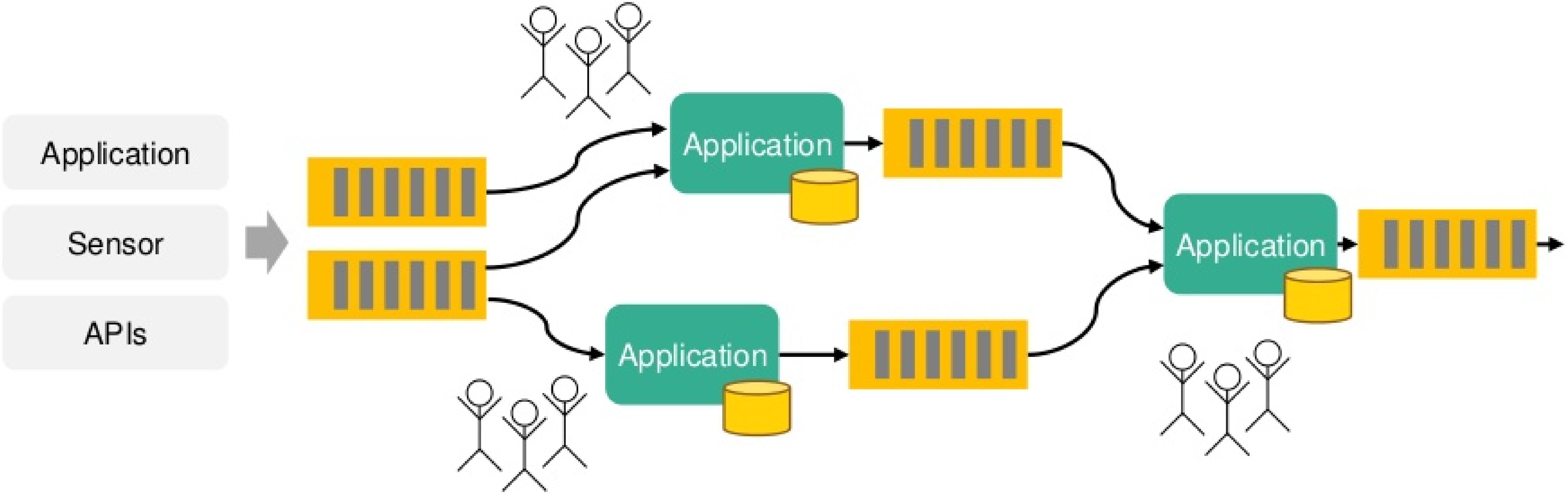
still involves managing databases

everyone is a mini-DBA

… 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

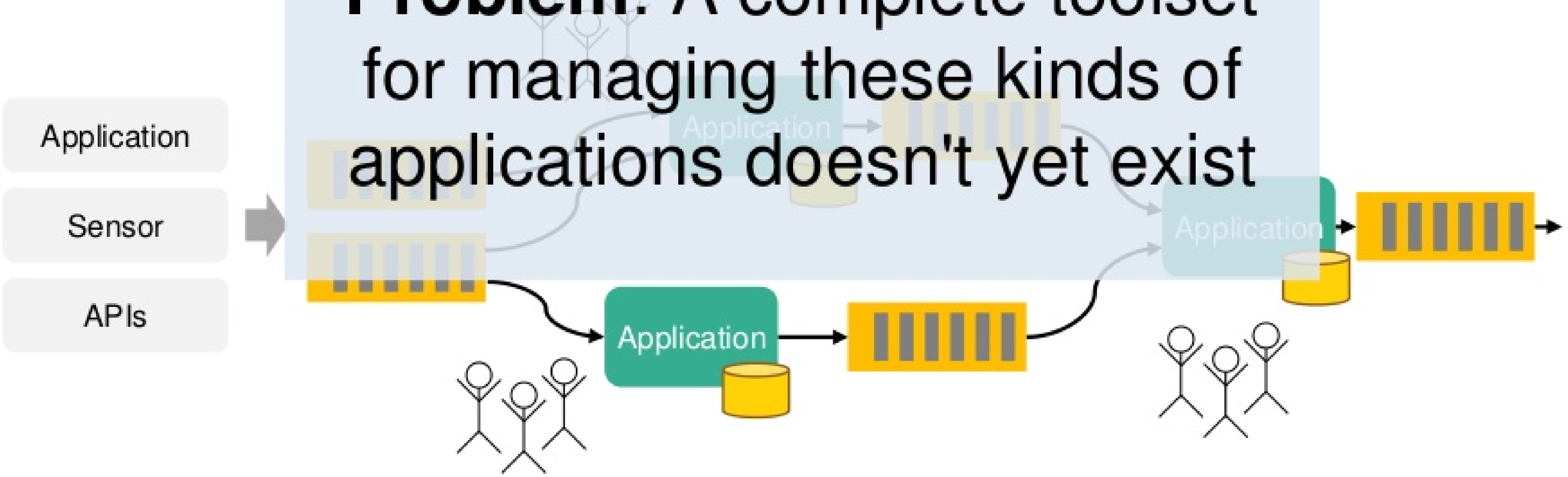


… and Stateful Stream Processing

very simple: state is just part of the application

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

Problem: A complete toolset for managing these kinds of applications doesn't yet exist



The rise of streaming platforms

- To solve these problems, companies started building internal streaming platforms
- For example, **Netflix** presented its Flink-based SPaaS (**Stream Processing as a Service**) platform at Flink Forward San Francisco 2017
- There is a need for self-service tools for stateful streaming applications

Netflix SPaaS: https://www.slideshare.net/mdaxini/flink-forward2017netflix-keystonespaas

Lessons learned

1. Apache Flink is here to stay
2. The stateful streaming architecture has been widely adopted
3. There's a gap to fill in tooling for this new architecture

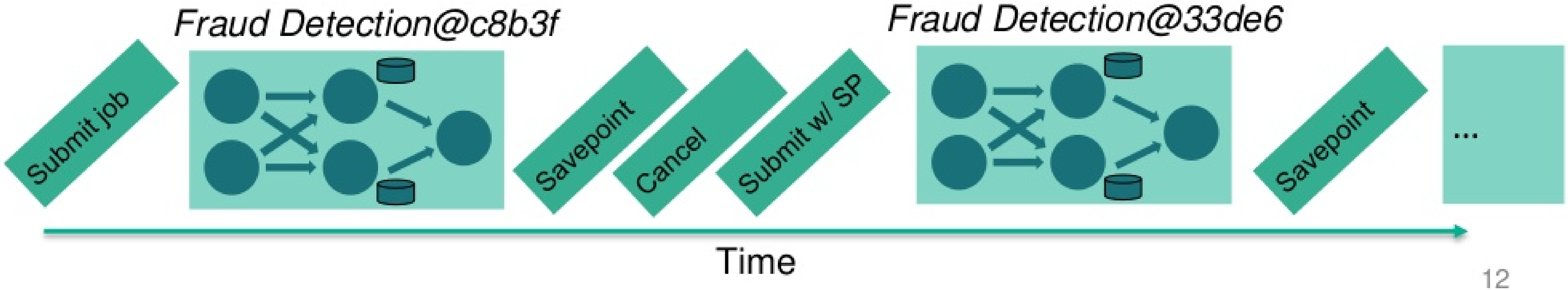
Introducing dA Platform 2

dA Platform 2

- Manage applications and state together
  - Instead of maintaining separate tools for applications (e.g. container environment) and state (e.g. databases), use one tool to manage their stateful streaming applications.
- Reduce time to production
  - dA Platform 2 comes with **all the infrastructure needed** to reliably operate streaming applications
  - It provides a **self-service platform** to operate streaming apps
  - Easily adopt streaming within an organization

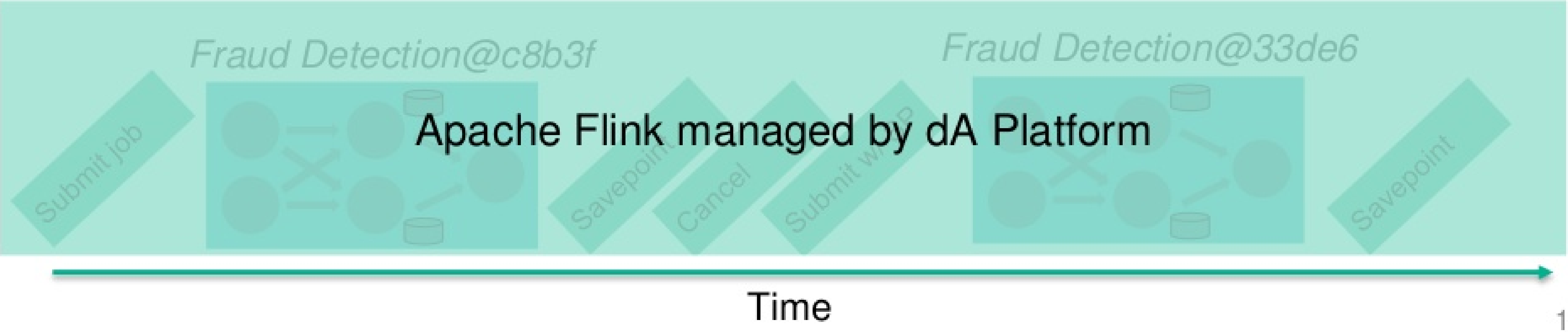
Instead of managing Flink streaming jobs manually …

- Requires users to manually call the APIs in Flink at the right time
- Handling any unexpected issues on the way
- Manual bookkeeping of savepoints, streaming job versions, configurations



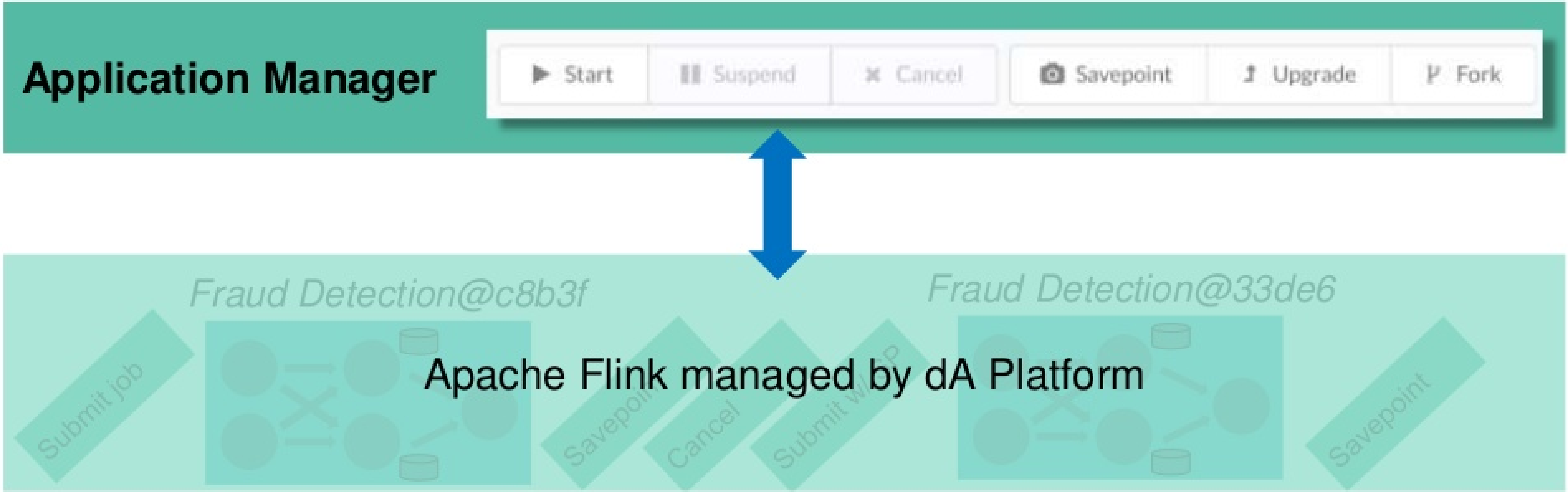
… dA Platform manages Flink

- dA Platform operates on a new concept: **Applications**, abstracting away the low-level details of Flink



Application Manager Intro

- Management layer within dA Platform 2, taking care of application lifecycle and metadata

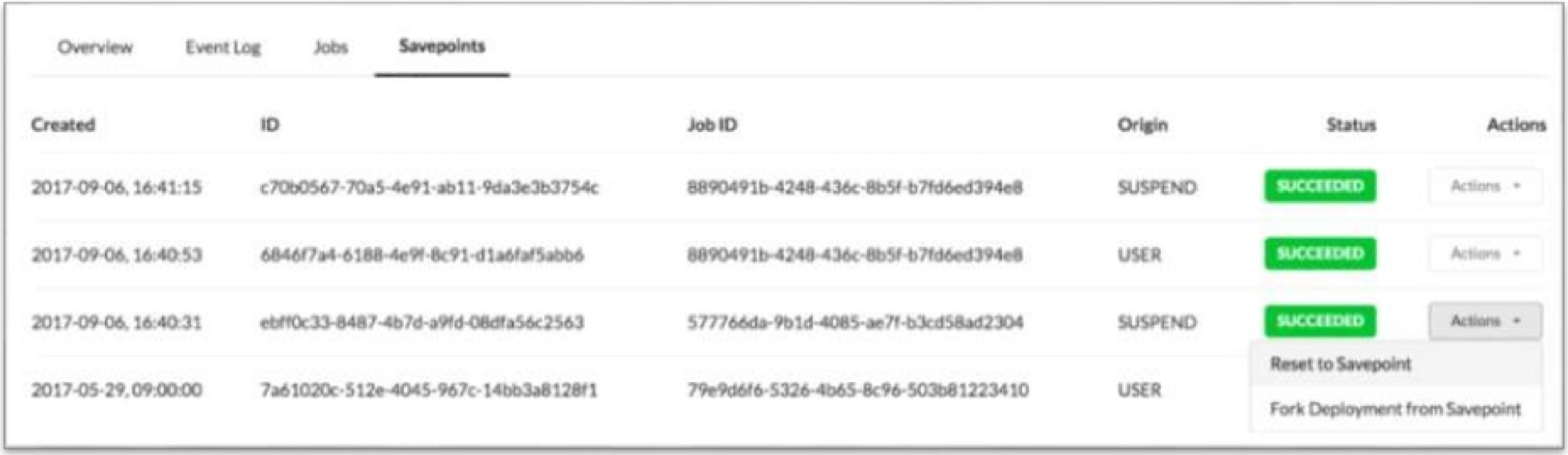


Application Manager

Apache Flink managed by dA Platform

Lifecycle Management

- **Start, suspend** (without state-loss) or **cancel** an application
- Manually Trigger a **savepoint**, **restore** to any savepoint

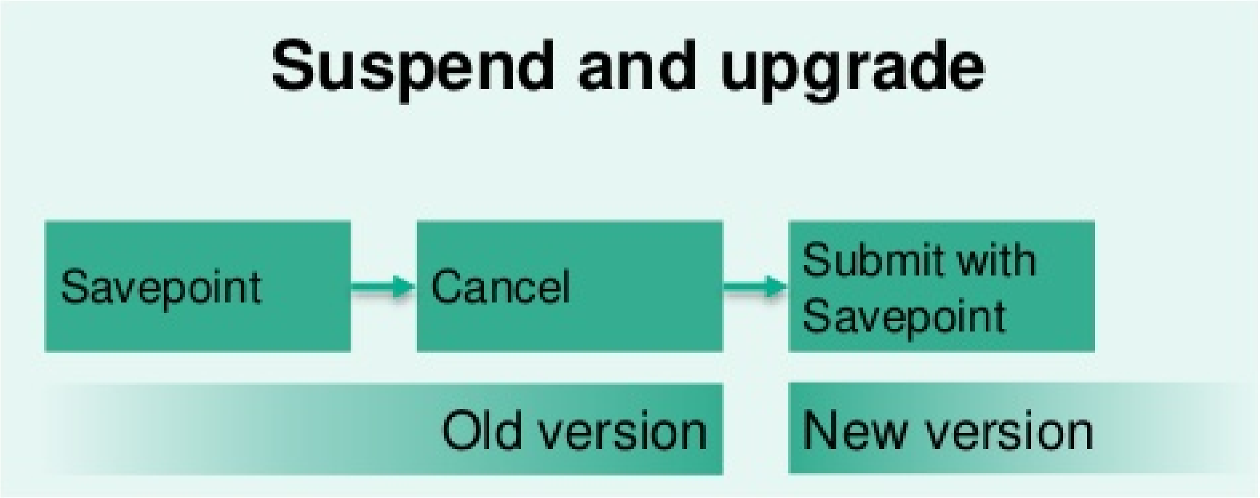


Created	ID	Job ID	Origin	Status	Actions
2017-09-06, 16:41:15	c70b0567-70a5-4e91-ab11-9da3e3b3754c	8890491b-4248-436c-8b5f-b7fd6ed394e8	SUSPEND	SUCCEEDED	Actions ▾
2017-09-06, 16:40:53	6846f7a4-6188-4e9f-8c91-d1a6faf5abb6	8890491b-4248-436c-8b5f-b7fd6ed394e8	USER	SUCCEEDED	Actions ▾
2017-09-06, 16:40:31	ebff0c33-8487-4b7d-a9fd-08dfa56c2563	577766da-9b1d-4085-ae7f-b3cd58ad2304	SUSPEND	SUCCEEDED	Actions ▾
2017-05-29, 09:00:00	7a61020c-512e-4045-967c-14bb3a8128f1	79e9d6f6-5326-4b65-8c96-503b81223410	USER		

Upgrading an application

- Deploy a newer application version
- Upgrade Flink
- Change configuration

- Upgrade modes:

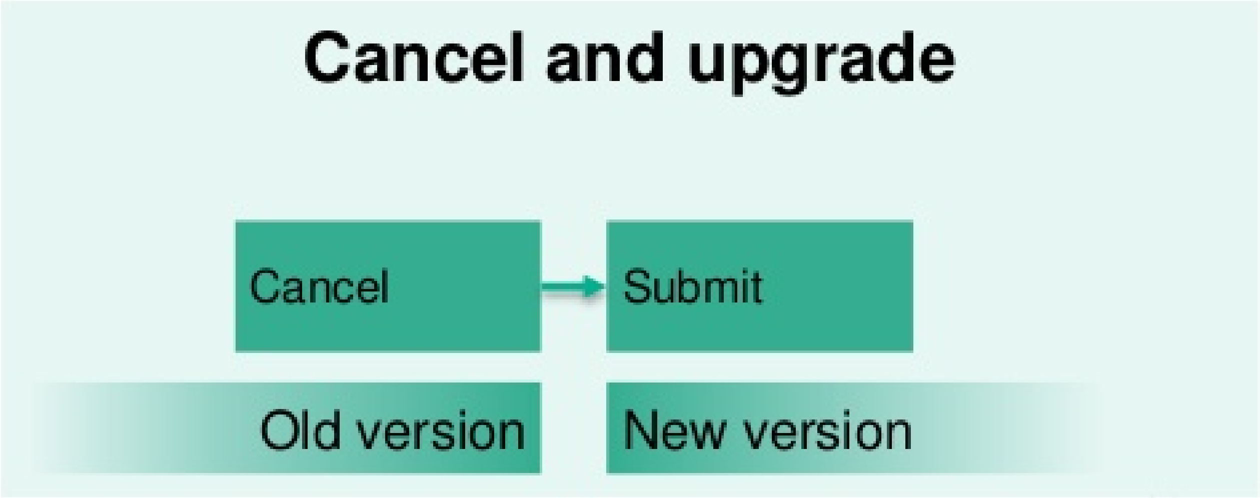


Suspend and upgrade

Savepoint → Cancel → Submit with Savepoint

Old version

New version



Cancel and upgrade

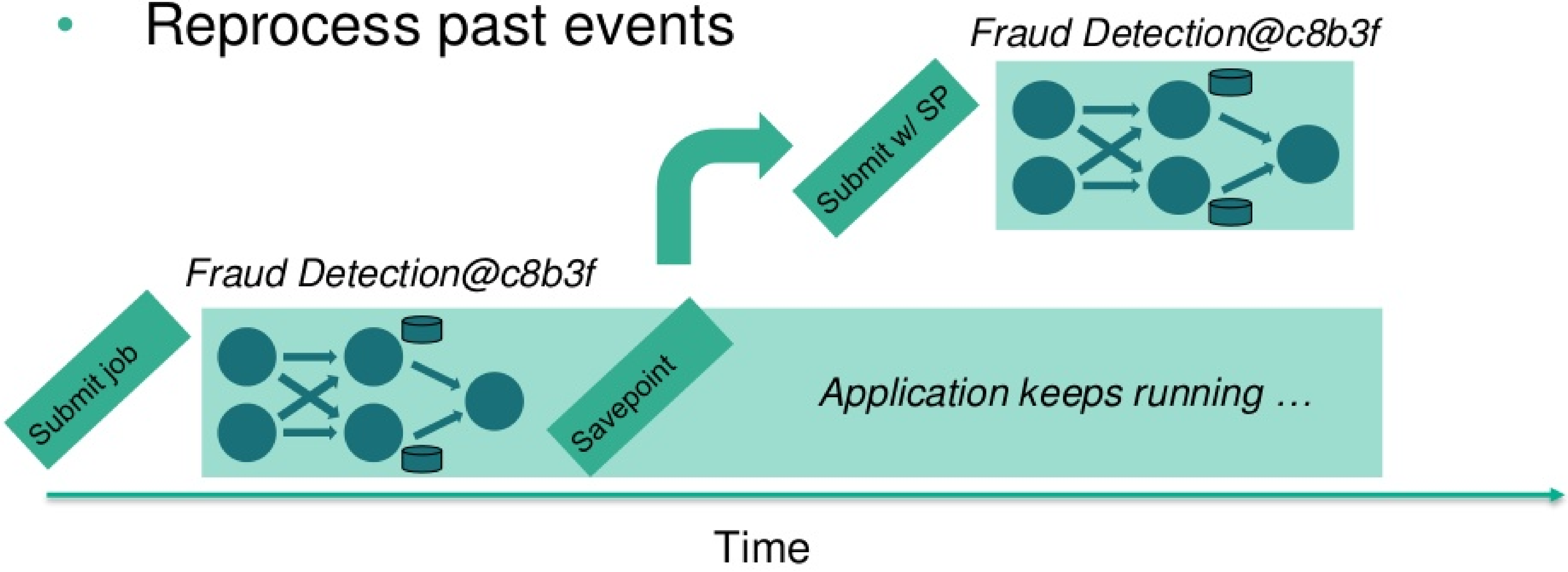
Cancel → Submit

Old version

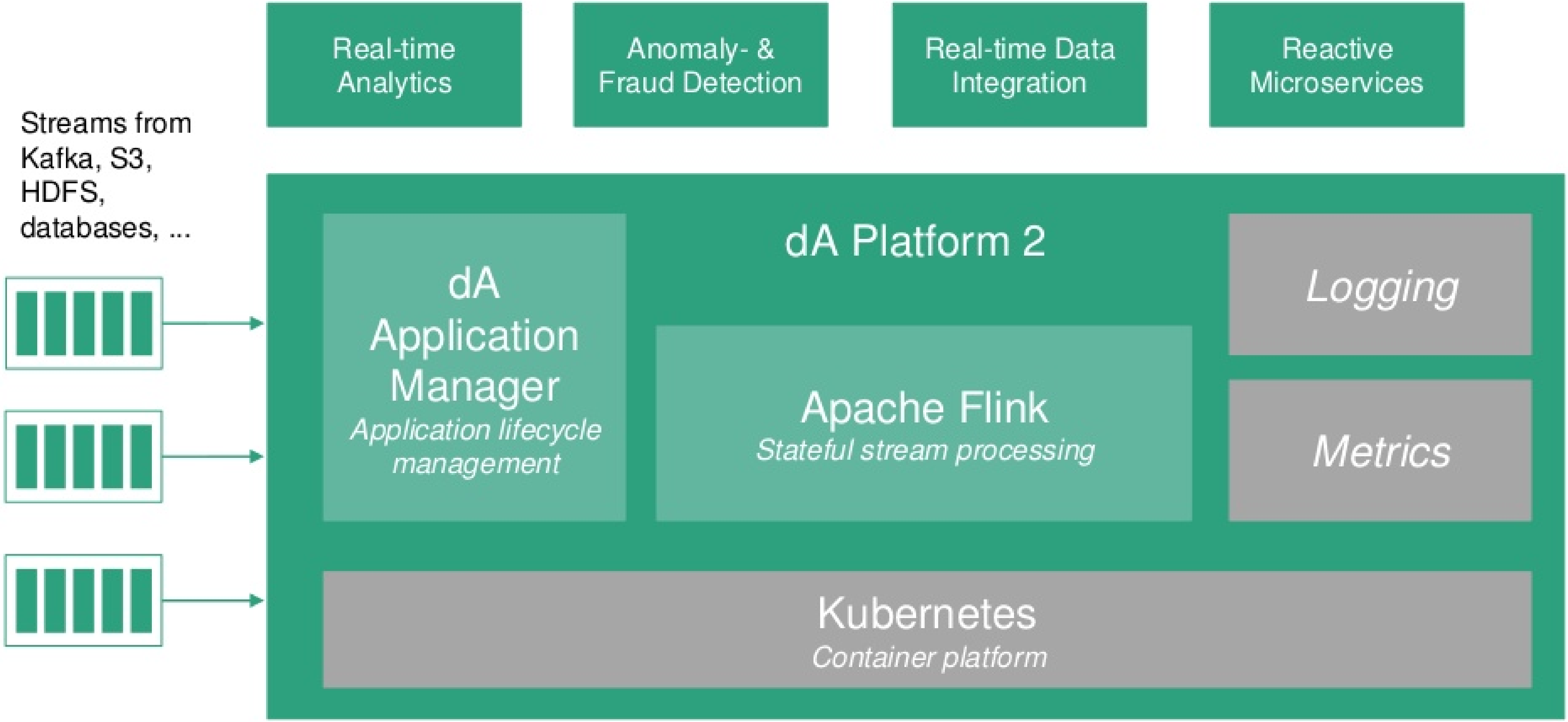
New version

Forking an application

- Stage changes in a pre-production environment
- Run experiments (a/b tests)
- Reprocess past events



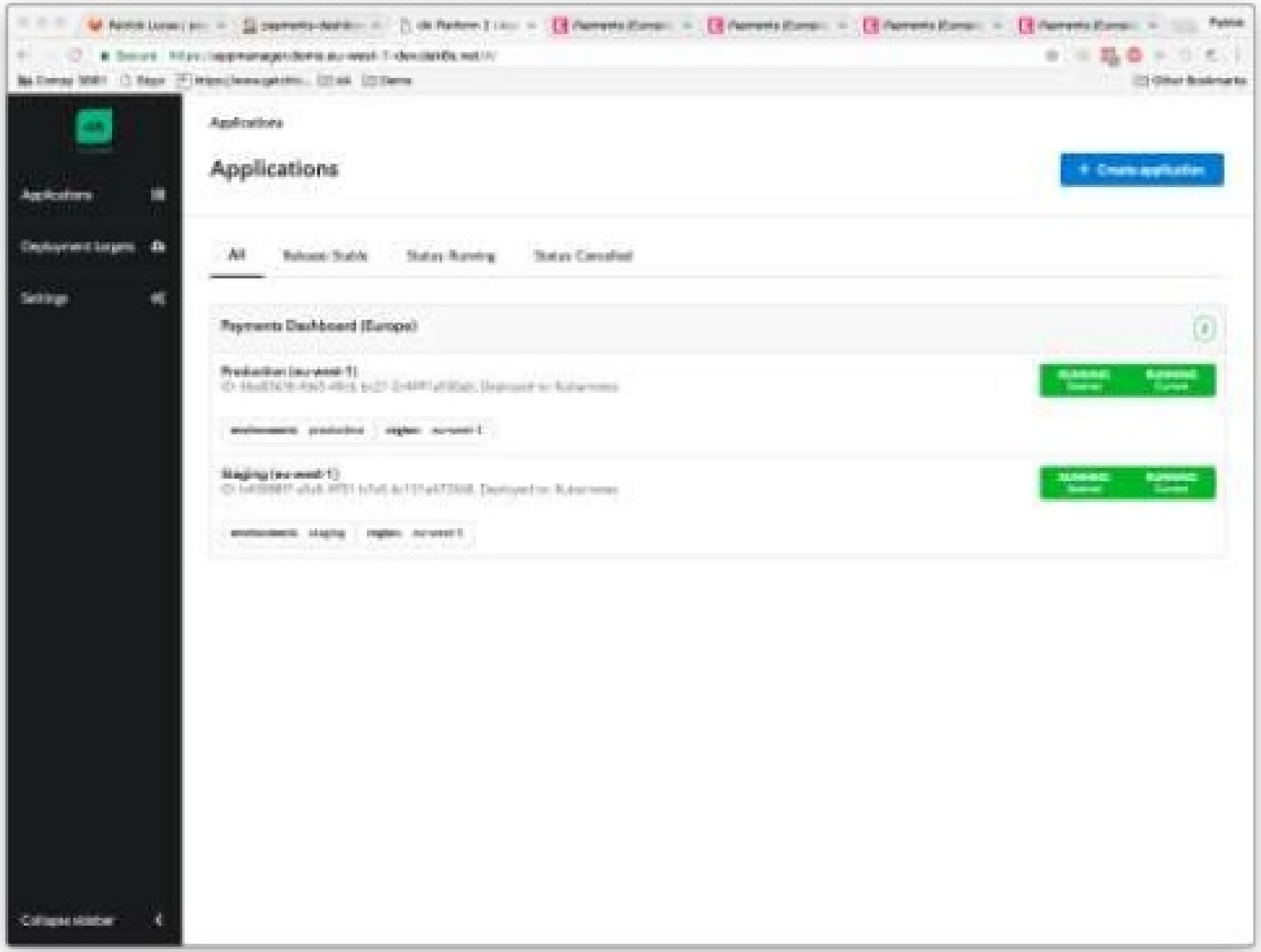
Architecture



Real-time Analytics
Anomaly- & Fraud Detection
Real-time Data Integration
Reactive Microservices
Streams from Kafka, S3, HDFS, databases, ...
dA Platform 2
dA Application Manager
Application lifecycle management
Apache Flink
Stateful stream processing
Logging
Metrics
Kubernetes
Container platform

Demo

Demo Components



Applications

Payments Dashboard (Europe)

Production (eu-west-1)
ID: [illegible], Deployed to: Kubernetes
environment: production region: eu-west-1

Staging (eu-west-1)
ID: [illegible], Deployed to: Kubernetes
environment: staging region: eu-west-1

dA Platform 2

Application Manager

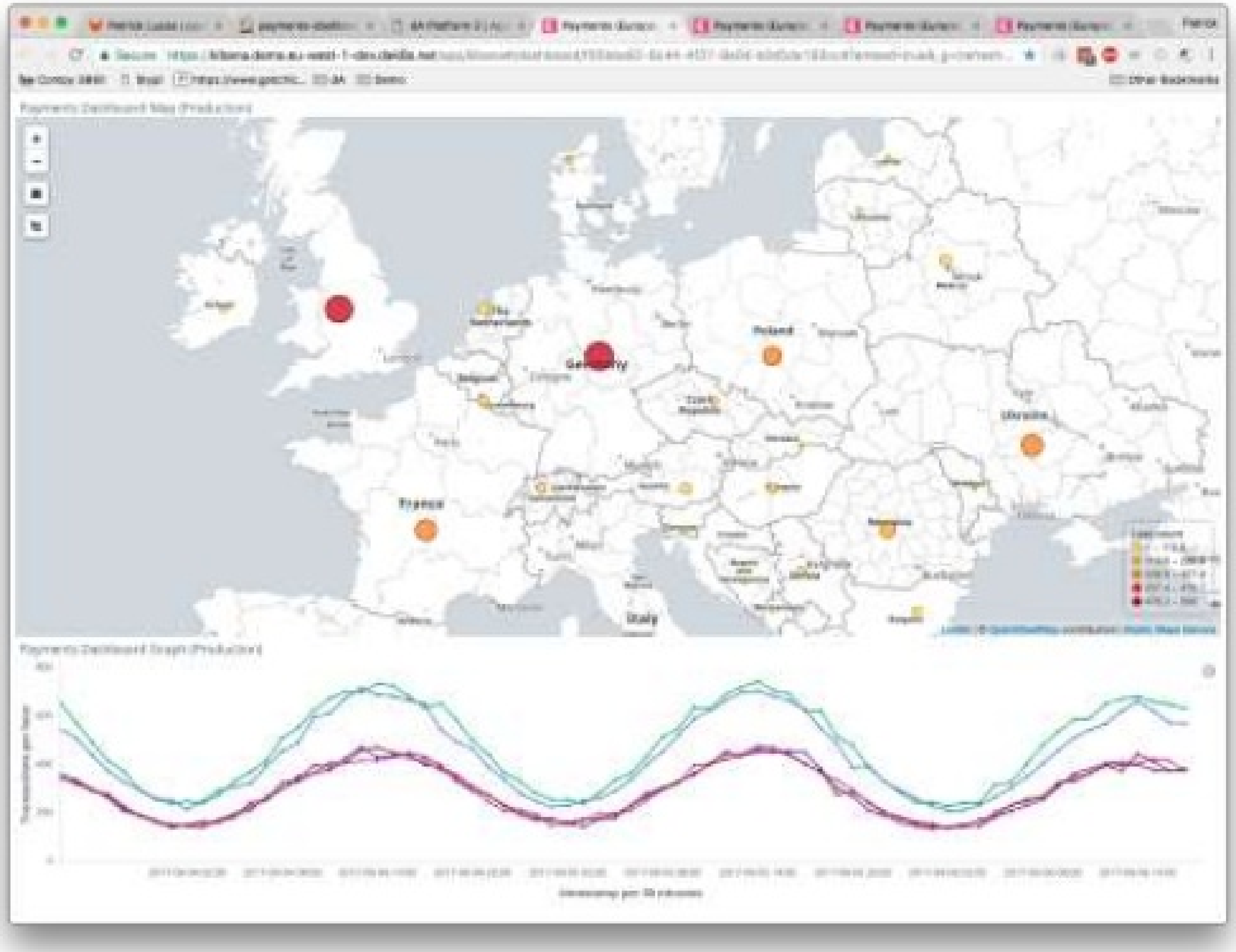
- One **Application**
  - Payments Dashboard (Europe)
- Two **Deployments**
  - Staging (eu-west-1)
  - Production (eu-west-1)

Demo Components

GitLab to host the code repository and trigger builds in Jenkins

Jenkins to build and test the code and initiate upgrades via the Application Manager’s **HTTP API**

Demo Components



Elasticsearch and Kibana

to store and visualize the dashboard’s data

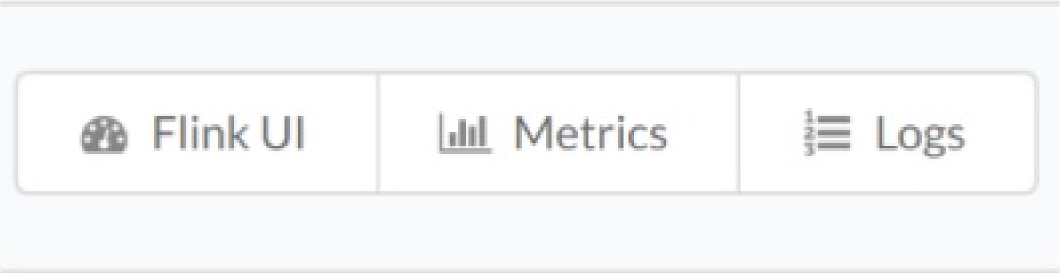
- Data is simulated payments coming in from around Europe
- Upper pane visualizes the relative proportion of payments from each country
- Lower pane plots the rate over time of the five highest volume sources

dA Platform 2 Architecture

Integrations

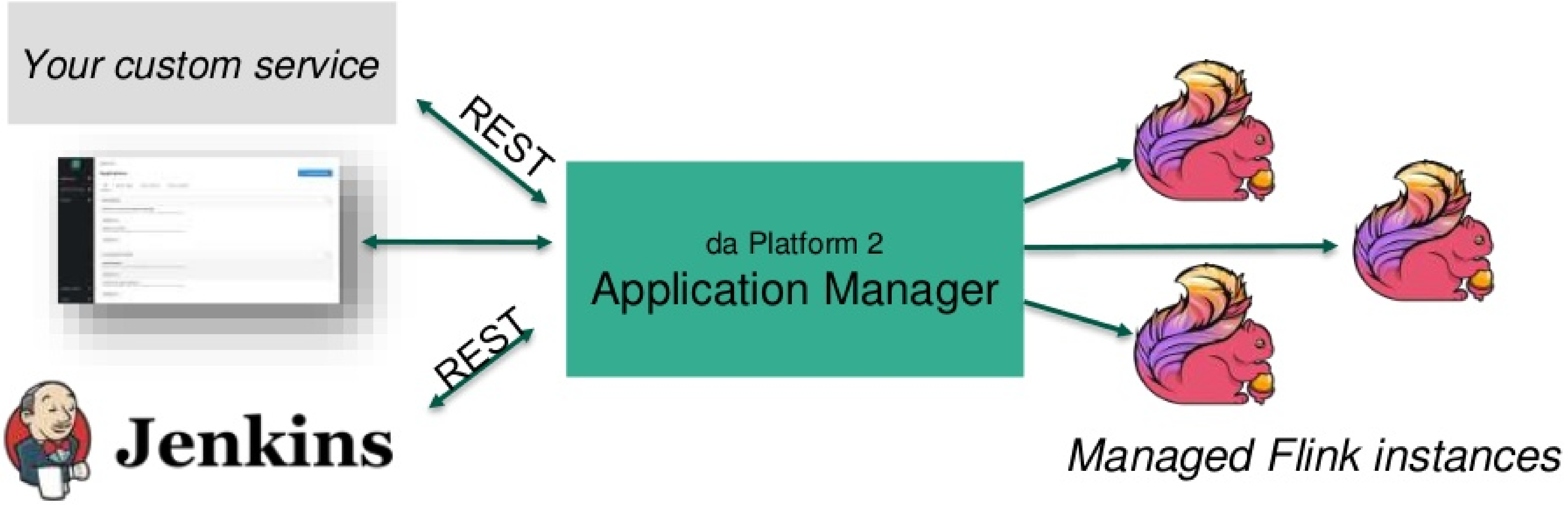
- Application Manager integrates with **centralized logging** and **metrics services**
- Access log of application for any point in time

→ Make debugging and monitoring as easy as possible from day one



Connectivity

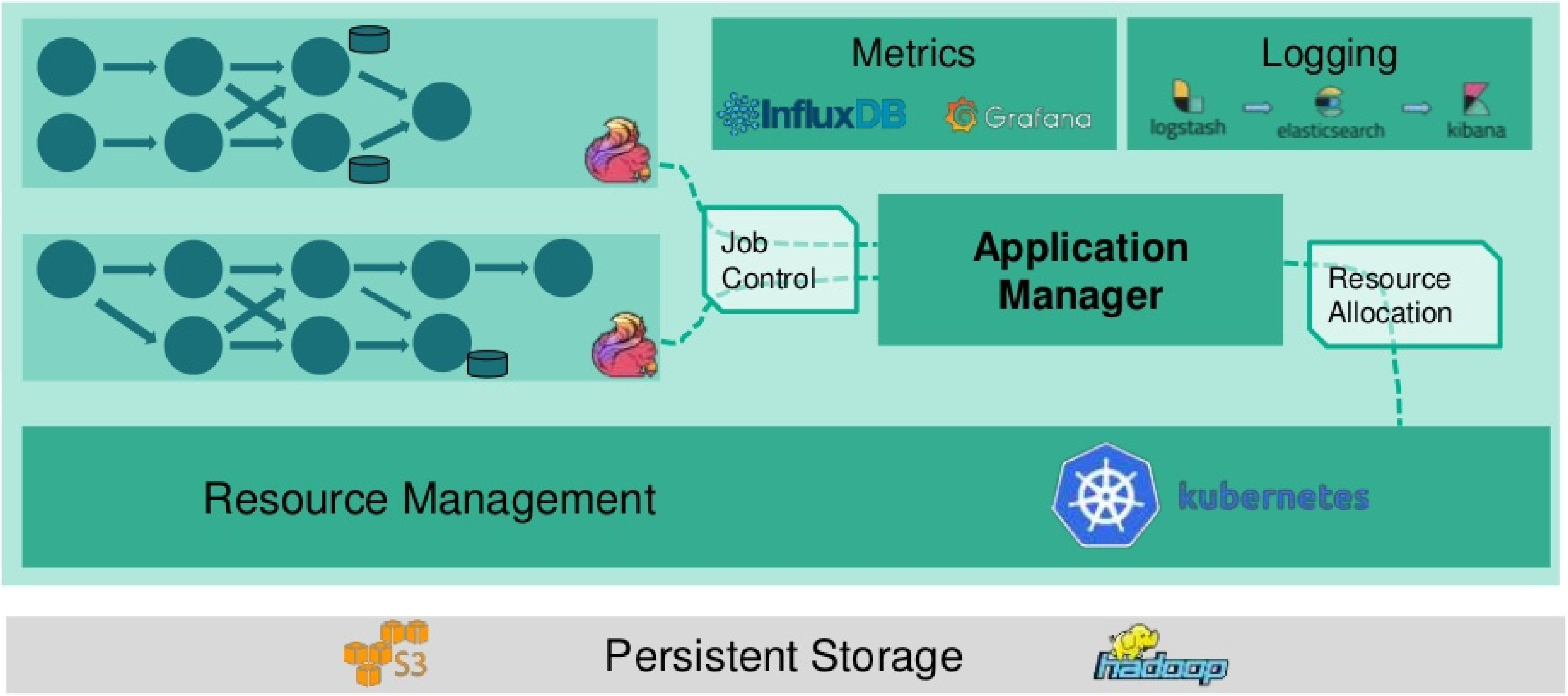
- **REST API** as first class citizen for custom integrations
- **Web-based** user interface and **command line** interface



Configuration and Deployments

- Advanced **configuration management**
  - Default configs + deployment specific configuration
  - Configuration history
- Support for deploying to multiple **deployment targets**
- A deployment target is the abstraction for any resource manager supported by Flink

dA Platform: Detailed Architecture



Architecture notes

- All components are chosen to be **cloud-ready**. dA Platform runs on public clouds and on-premise
- All components are **pluggable**. In particular metrics and logging integrations
- We plan to support more deployment targets than just Kubernetes in the future

Closing

dA Platform 2

- Manage applications and state together
- Reduce time to production by relying on the best practices from the original creators of Apache Flink
- Manage streaming application lifecycle easily
- Make streaming technologies accessible as self-service platform

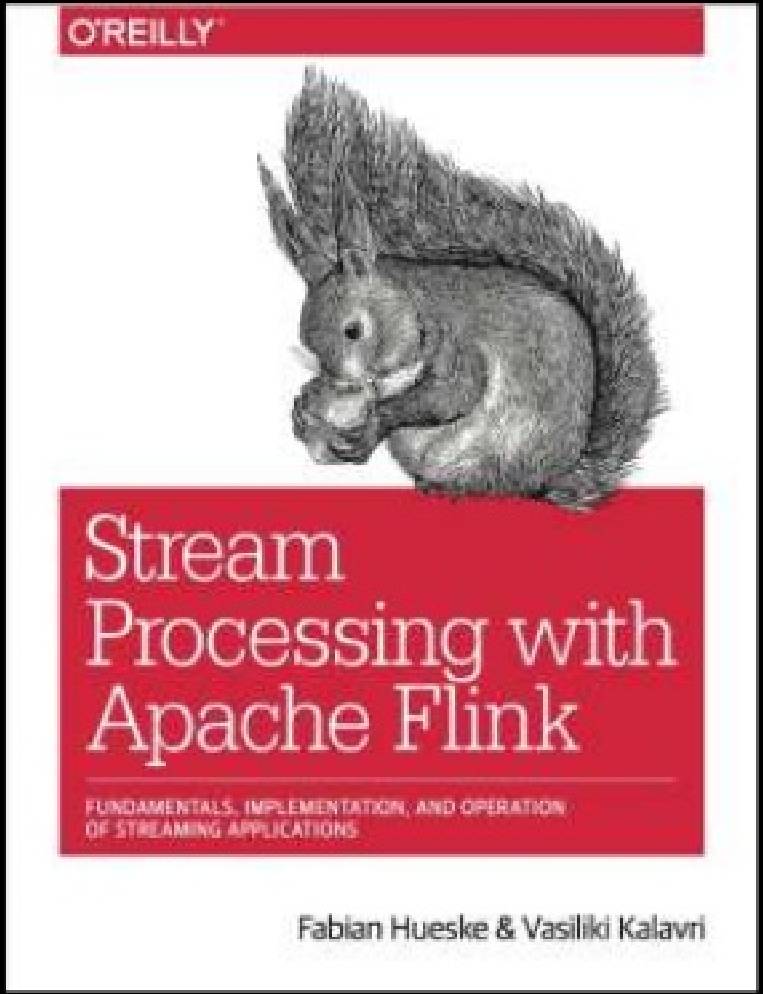
dA Platform 2: Roadmap

- Signup on the data Artisans website for a **product newsletter** and **Early Access Program**.
- General Availability is planned for end of 2017 / early 2018
- Visit the **data Artisans booth** to learn more
- Reach out at *platform@data-artisans.com*

dA Platform 2 with Application Manager and Apache Flink®

Q & A

Reach out to us at *platform@data-artisans.com*



O'REILLY

Stream Processing with Apache Flink

FUNDAMENTALS, IMPLEMENTATION, AND OPERATION OF STREAMING APPLICATIONS

Fabian Hueske & Vasiliki Kalavri

Thank you!

@rmetzger | @theplucas
@dataArtisans

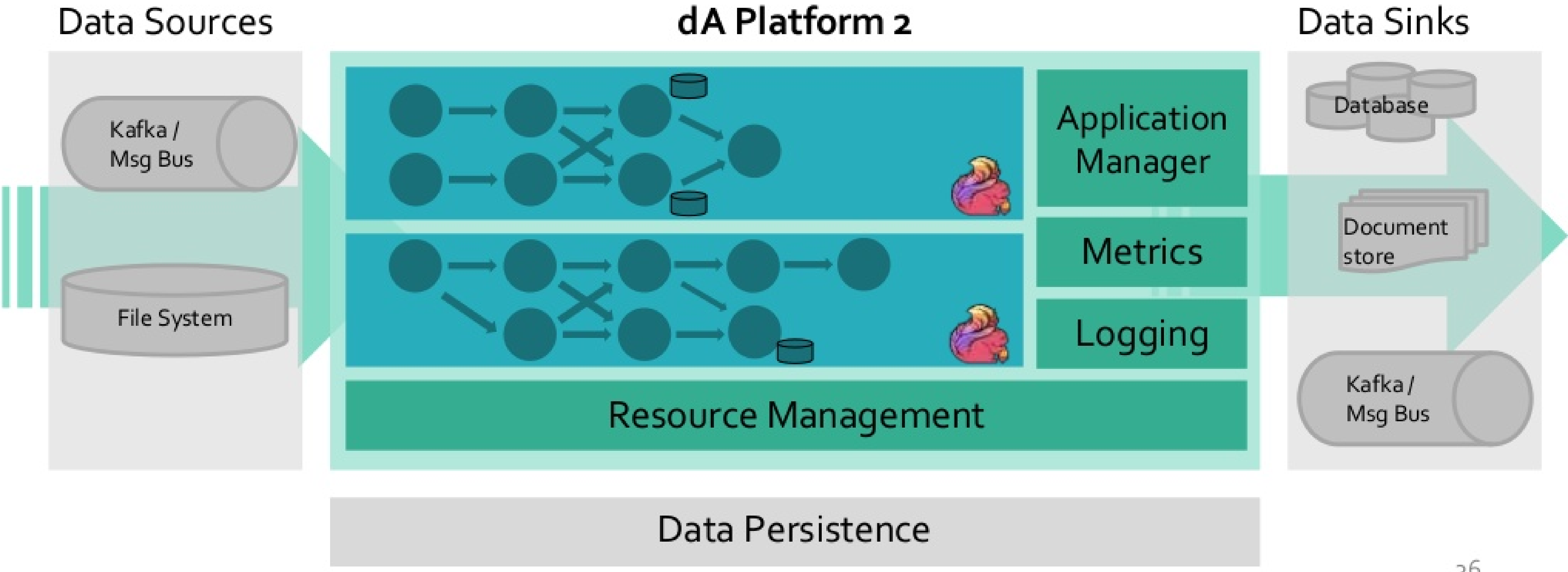
dataArtisans

We are hiring!

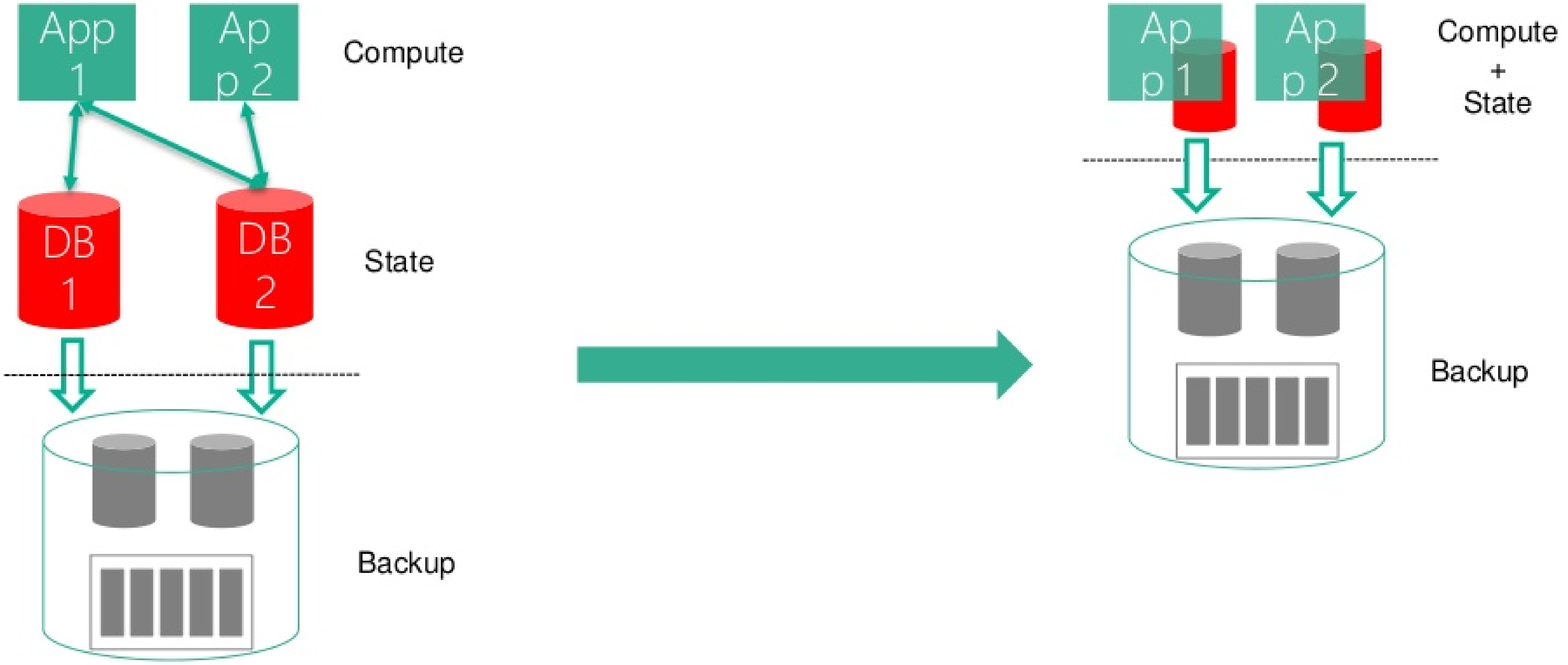
data-artisans.com/careers

backup slides

dA Platform Architecture



Architectures are changing

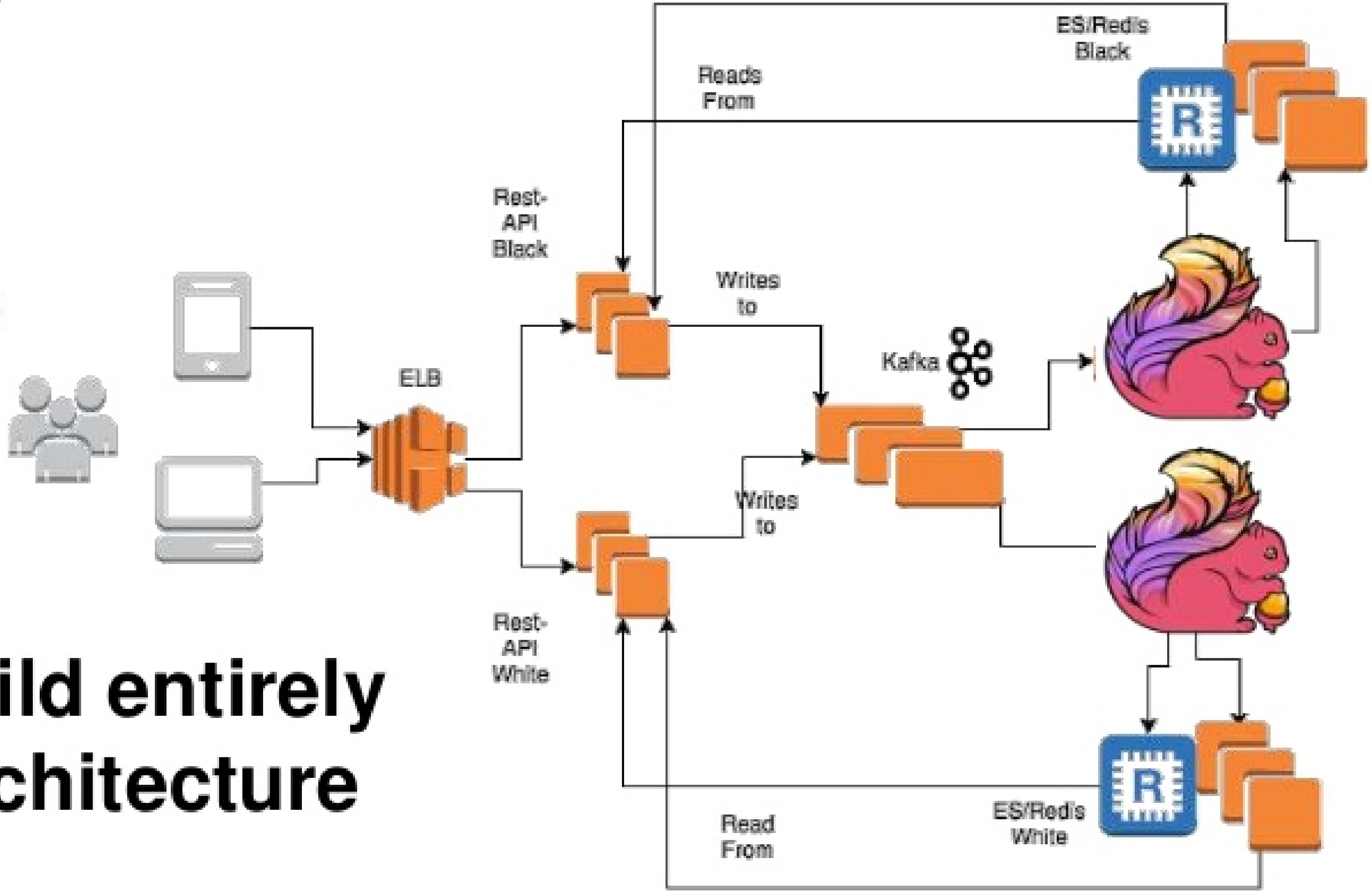


Traditional tiered architecture

Streaming architecture

@

THE SOCIAL NETWORK FOR PETROLHEADS



A social network build entirely on the streaming architecture

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

Building streaming applications is easy …

- **… productionizing them is hard**
  - Integration with existing infrastructures and processes
    - build pipeline
    - resource / cluster management
    - monitoring
    - data sources and sinks, persistent state storage
  - Figuring out which components to choose
- Feedback: More time spend on operations than on implementation

Self-service streaming platforms

- Companies are building their own Flink streaming platforms
- Integration with internal infrastructures
- Right now, Flink has limited integration capabilities

dA Platform 2: Making Flink easy

- dA Platform 2 solves the following problems:
  - Managing stateful Flink streaming jobs
  - Integrating Flink into infrastructures, and providing best practices for them
  - Providing a self-service Flink Platform

→ **Reduce to time production**

- You get the best tools from day one → more developer productivity

every team needs to solve…

- consistent stateful upgrades
  - ⑩ application evolution and bug fixes
- migration of application state
  - ⑩ cluster migration, A/B testing
- re-processing and reinstatement
  - ⑩ fix corrupt results, bootstrap new applications
- state evolution (schema evolution)

Rethinking data architectures

- The infrastructure requirements are changing with this new architecture
- Deployment, scaling, migrations, upgrades and debugging are easier -- because state and compute are in the same system.
- However, this **new architecture requires different tools** and systems.
- Feedback from users: Implementation of streaming applications is easier than deployment and operations