FLINK AS A LIBRARY (AND STILL AS A FRAMEWORK)

- GARY YAO

dataArtisans

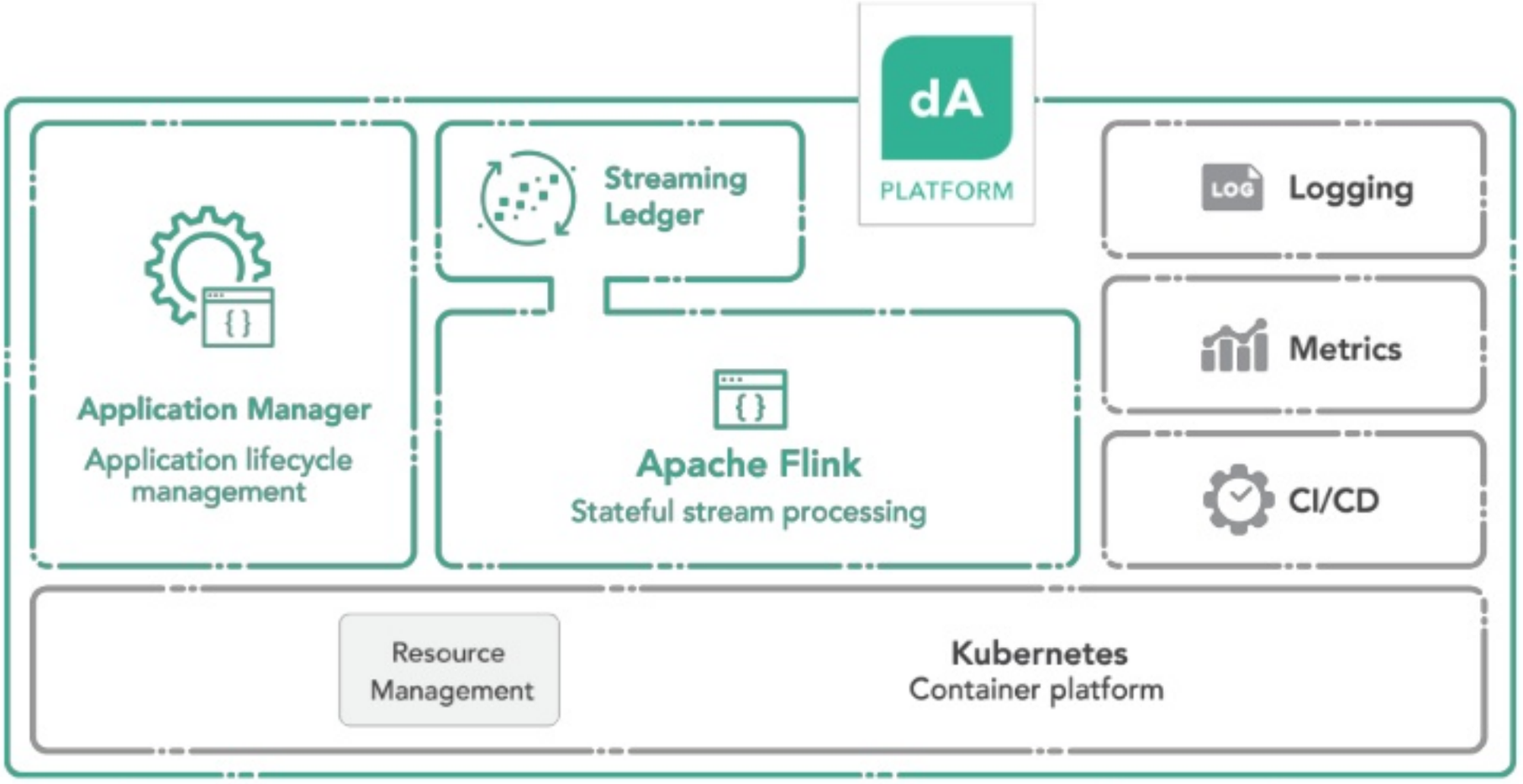
ABOUT DATA ARTISANS

Original Creators of Apache Flink®

dA PLATFORM

Real Time Stream Processing Enterprise Ready

FREE TRIAL DOWNLOAD



data-artisans.com/download

MOTIVATION

FLINK DEPLOYMENT OPTIONS & LIMITATIONS

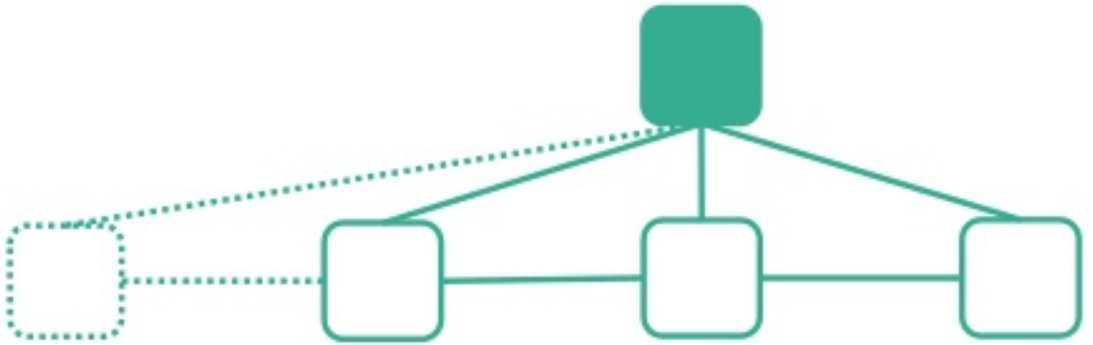
Before Apache Flink 1.5

	Session cluster	Job cluster
Standalone	✓	
YARN	✓	✓
Mesos	✓	
Kubernetes	✓	

- Shortcomings:
  - Fixed number of TaskManagers
  - Difficult to support autoscaling
  - Supporting new cluster managers difficult
  - Akka for client—cluster communication
  - Undisciplined about Job clusters

FLINK AS A LIBRARY

- Make deployments easier
- Application-oriented instead of cluster-oriented
- Just a start one or more Job jars, and automatically get distributed execution
- Remove distinction between different roles in the cluster (JM vs. TM)
- Participants of the cluster should determine roles on their own



RESOURCE MANAGEMENT & AUTOSCALING

- Short-lived & high latency
- Resources only exist for the duration of the job
- Different stages in a job may require different resources
- Flink should be in control of allocating/releasing resources

- Long-lived & low latency
- Dependent on external systems & exposed to varying load
- External system should drive resource allocation to meet SLO/SLAs
- Flink should react to available resources

CONTAINERIZED DEPLOYMENT

Why Flink on containers?

- Self-contained runtime environment that includes application code, libraries, dependencies, and configuration files
- Ease of operations by clean separation of concerns
- Enables dynamic resource allocation
- Fits the *"Flink as a Library"* model

CONTAINERIZED DEPLOYMENT (CONT.)

Brief introduction to Kubernetes

- De-facto industry standard for container orchestration
- Resource-oriented with declarative configuration
  - You tell K8s the desired state, and a background process asynchronously makes it happen
  - *"3 replicas of this container should be kept running"*
  - *"a load balancer should exist, listening on port 443, backed by containers with this label"*
- Core resource types:
  - Pod: a group of one or more containers running on a node
  - Deployment: keep n pods running

REWORKING FLINK'S RUNTIME

FLIP-6 – Flink Deployment and Process Model (Flink 1.5 and up)

- Revamped distributed architecture
  - Resource elasticity
  - Support for different environments
  - RESTful API for client-cluster communication
- Building blocks:

ResourceManager

- Cluster manager specific
- Acquires/releases resources

Dispatcher

- Touch point for job submissions
- Spawns JobManagers

JobManager

- Single job only
- Deploys/monitors job execution

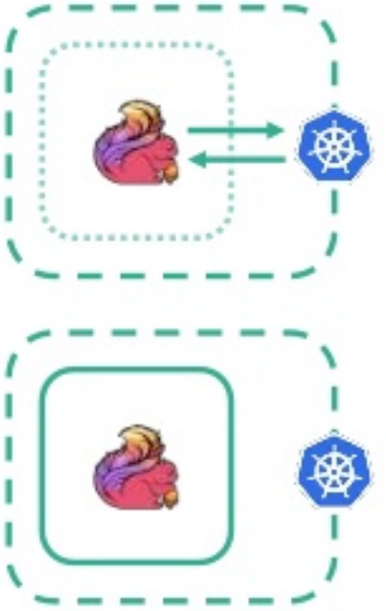
TaskManager

- Offers resources to ResourceManager
- Gets tasks from JobManagers

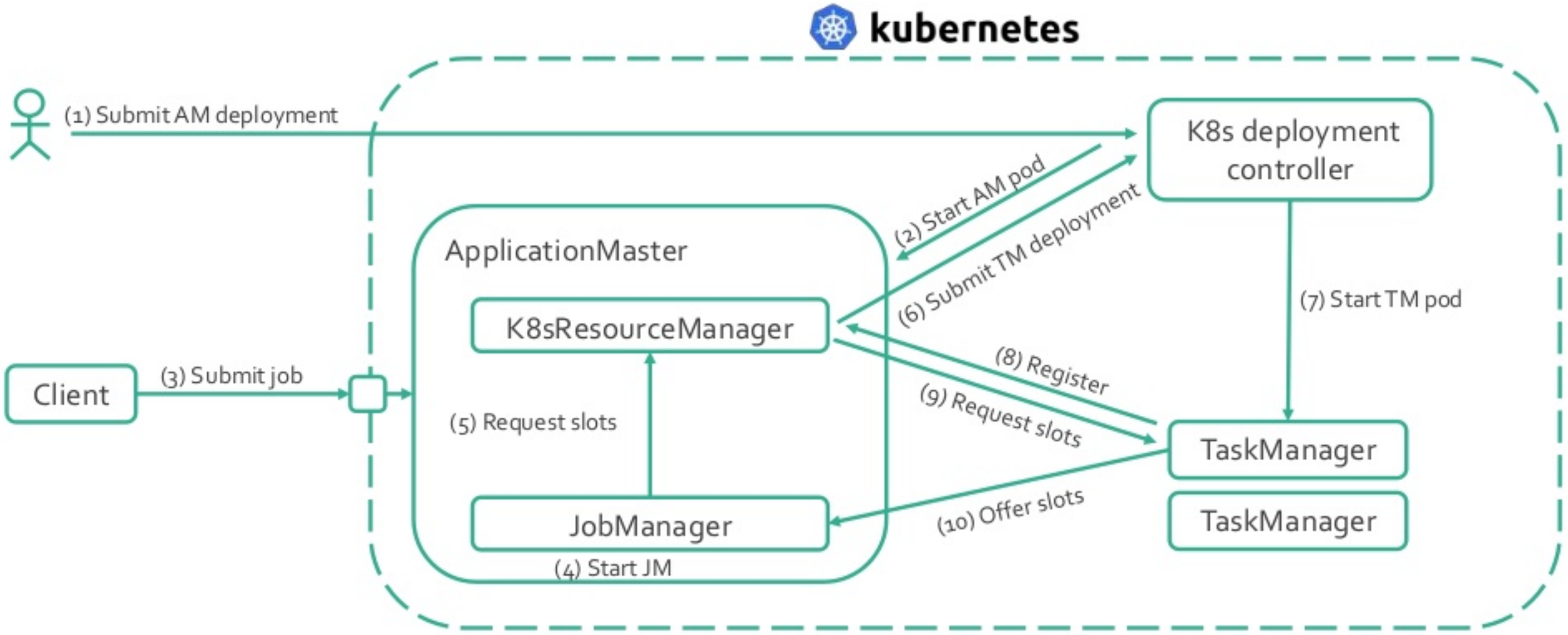
UPCOMING CHANGES

CONTAINER MODES

- Different levels of interaction with cluster manager
- Active Container Mode
  - Flink is aware about the cluster manager that it is running on, and interacts with it
  - Already exists, e.g., FLIP-6 YARN
- Reactive Container Mode
  - Flink is oblivious to its environment
  - Similar to standalone deployment
  - Flink may react on new resources by scaling up job



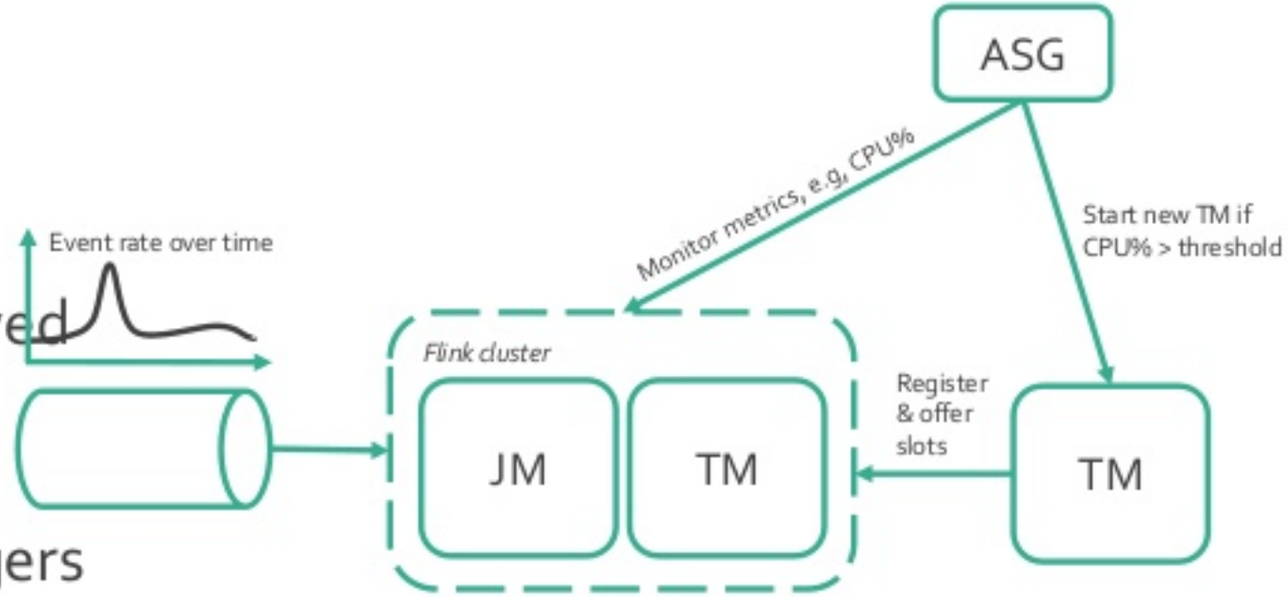
ACTIVE K8S INTEGRATION



 |

REACTIVE CONTAINER MODE

- Relies on external system to start/release TaskManagers, e.g.,
  - Kubernetes Horizontal Pod Autoscaler
  - GCP Autoscaling
  - AWS Auto Scaling Group
- Re-scale job if resources are added/removed (take savepoint and resume job with new parallelism automatically)
- By definition works with all cluster managers



DOES ANY OF THIS WORK?

Demo: Autoscaling in Active & Reactive Container Mode

CURRENT STATE

- Recommendations on how to deploy job and session cluster on K8s
  - Docker image build script
  - K8s resource configs
- StandaloneJobClusterEntryPoint to start a (containerized) job cluster
- Work in progress/planned:
  - Reactive Container Mode with automatically rescaling
    - Design documents will be published soon
  - Active K8s integration: prototype is available
    - git clone -b nativeKubernetes git@github.com:tillrohrmann/flink.git

SUMMARY

- Thinking in terms of jobs & application code as an alternative to deploying Flink clusters
- Active K8s integration & Reactive Container Mode
- Existing code already enables job and session clusters on K8s
- Call to action:
  - Umbrella tickets: FLINK-9953, FLINK-7087
  - Follow dev@flink.apache.org to stay informed about latest state of development

THANK YOU!

@gjyao
@dataArtisans
@ApacheFlink

WE ARE HIRING

data-artisans.com/careers

dataArtisans