Managing Distributed Workflows at Scale
- *Kubernetes Jobs in Action*

Abhishek Kumar Singh
Software Engineer, Unbxd

UNBXD

About UNBXD

Search

VERTICAL-SPECIFIC RELEVANCE

RICH & VISUAL AUTOSUGGEST

MOBILE OPTIMIZED

Browse

1:1 PERSONALIZED BROWSING

A/B TESTING & ATTRIBUTE PAGES

BEHAVIORAL TARGETING

Recommendations

EASY EXPERIMENTATION

CAMPAIGN MANAGEMENT

ADVANCED AI MODELS

Product Information Management

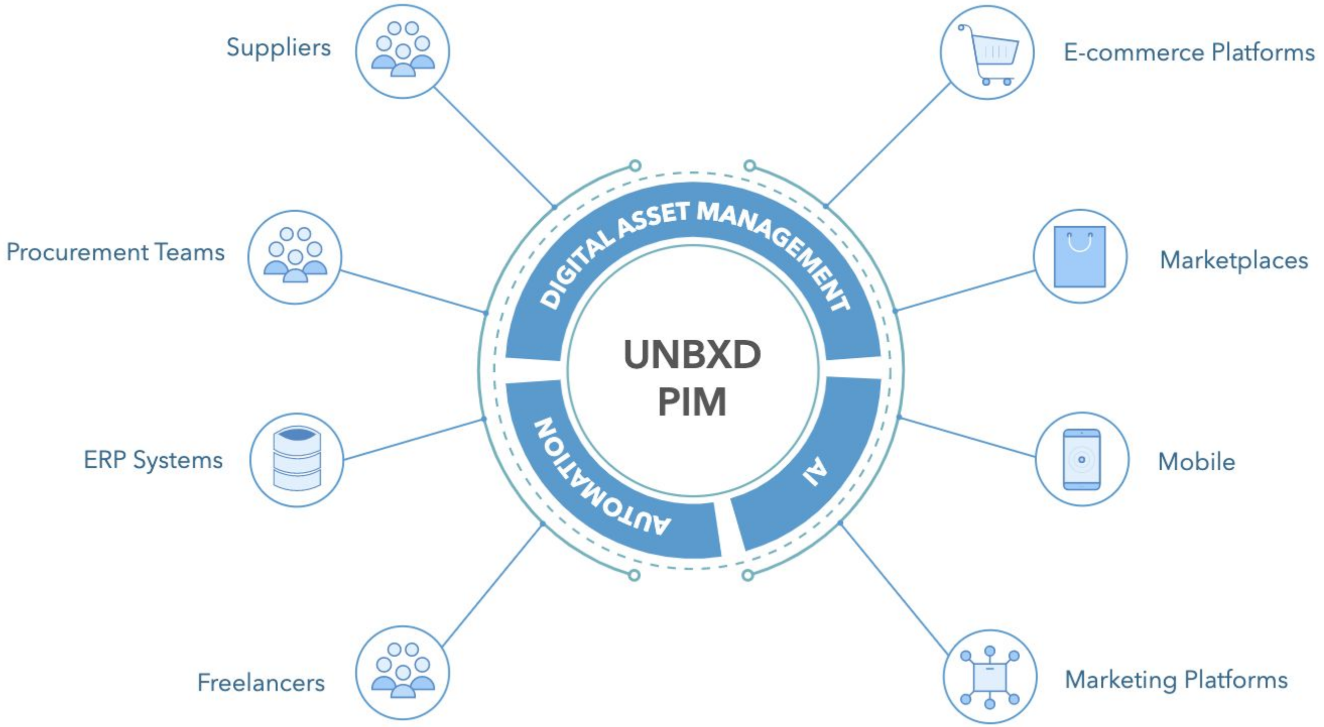
TASK AUTOMATION

EASY WORKFLOWS

GRANULAR CONTROLS

UNBXD

About Unbxd PIM



UNBXD

Agenda

1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Final Architecture
3. *Kubernetes and it’s features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

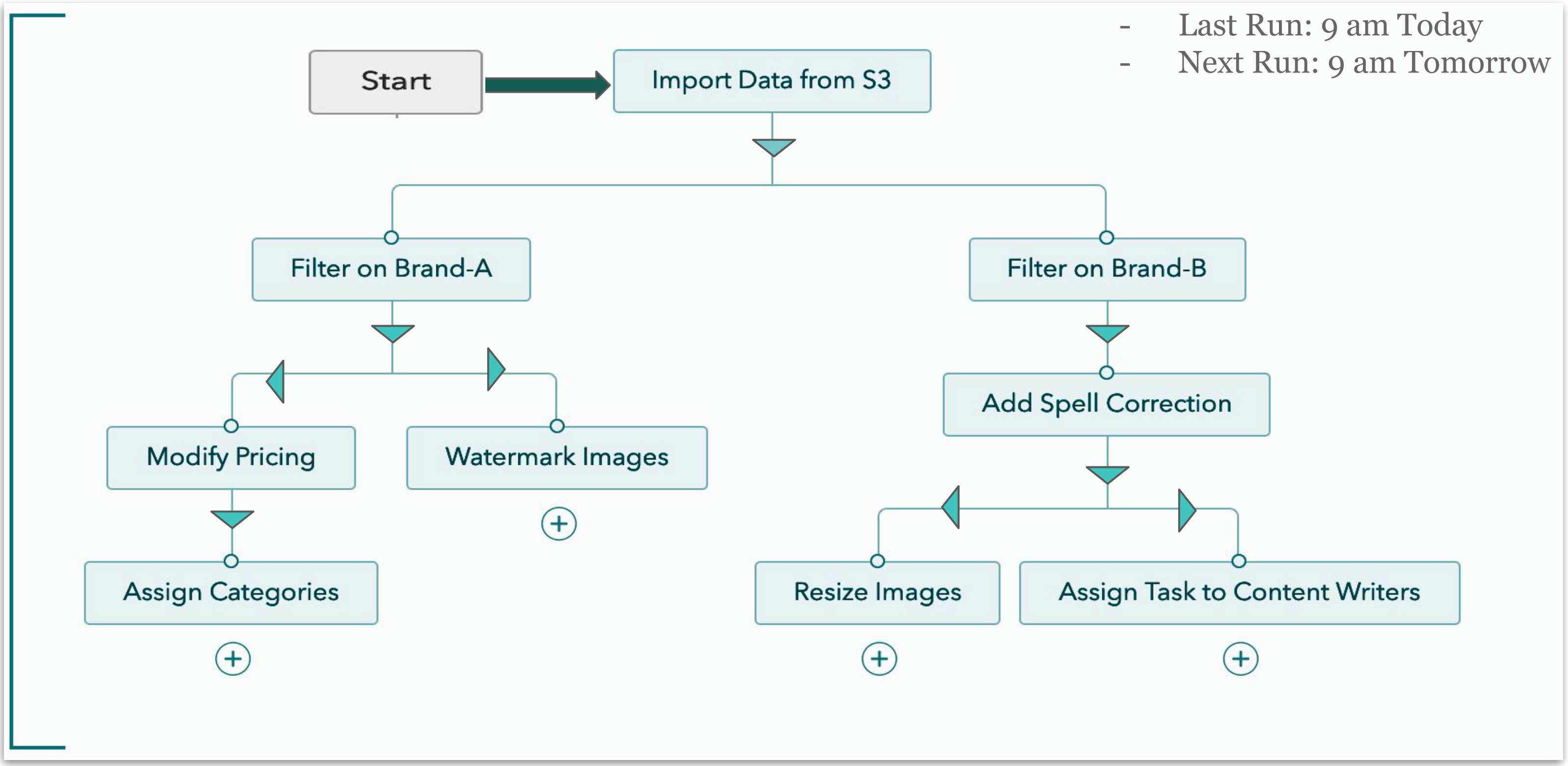
UNBXD

Next..

1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Final Architecture
3. *Kubernetes and it’s features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

UNBXD

Workflow in E-Commerce - An Overview



- Last Run: 9 am Today
- Next Run: 9 am Tomorrow

UNBXD

Next..

1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Final Architecture
3. *Kubernetes and it's features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

Workflow Orchestration Engine: Objectives

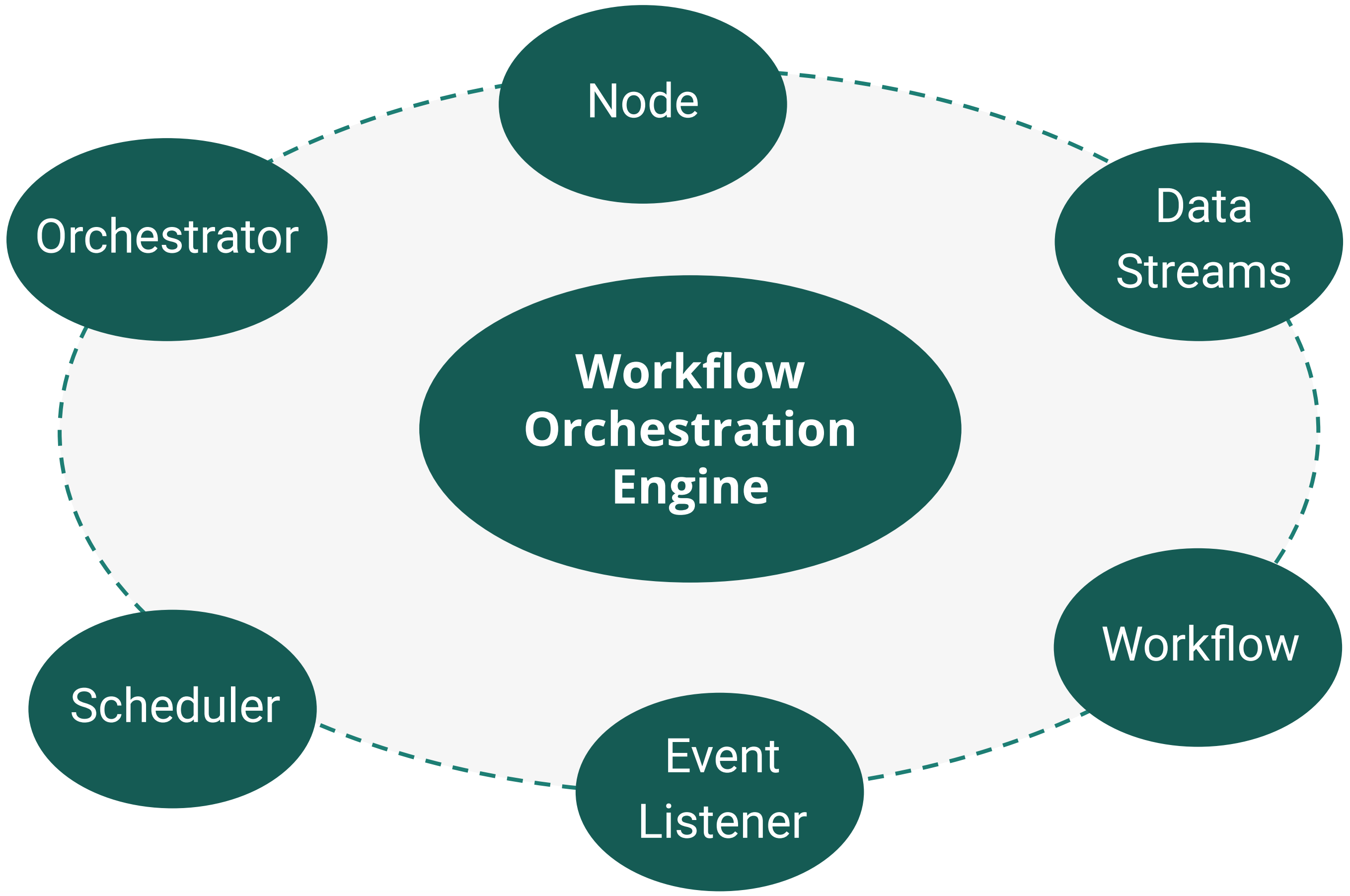
- Scalable
- Fault Tolerant
- Time or Event Based Triggers
- REST APIs for Configuration

UNBXD

Next..

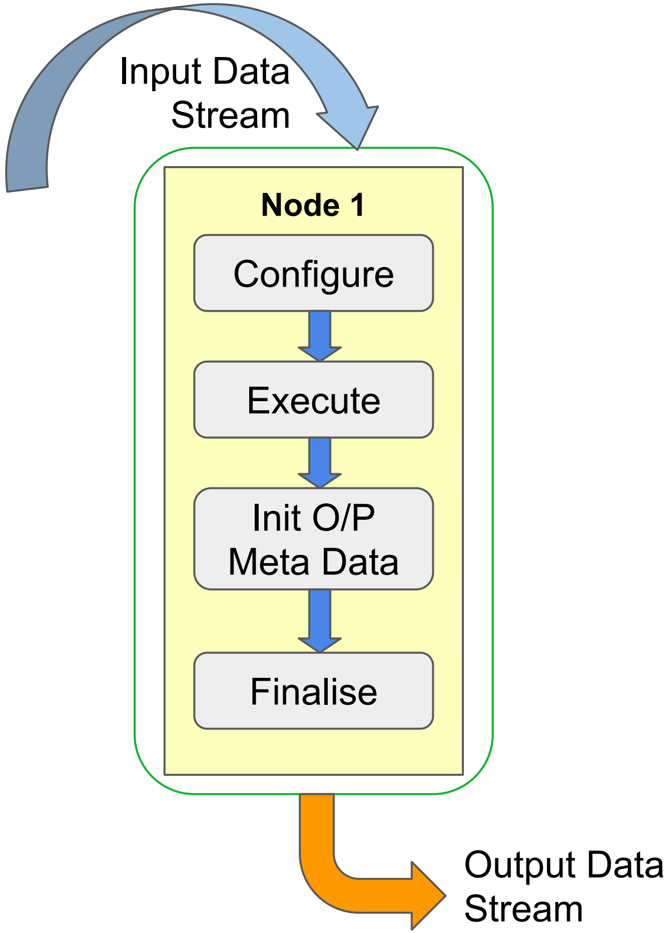
1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Final Architecture
3. *Kubernetes and it's features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

Workflow Orchestration Engine: Components



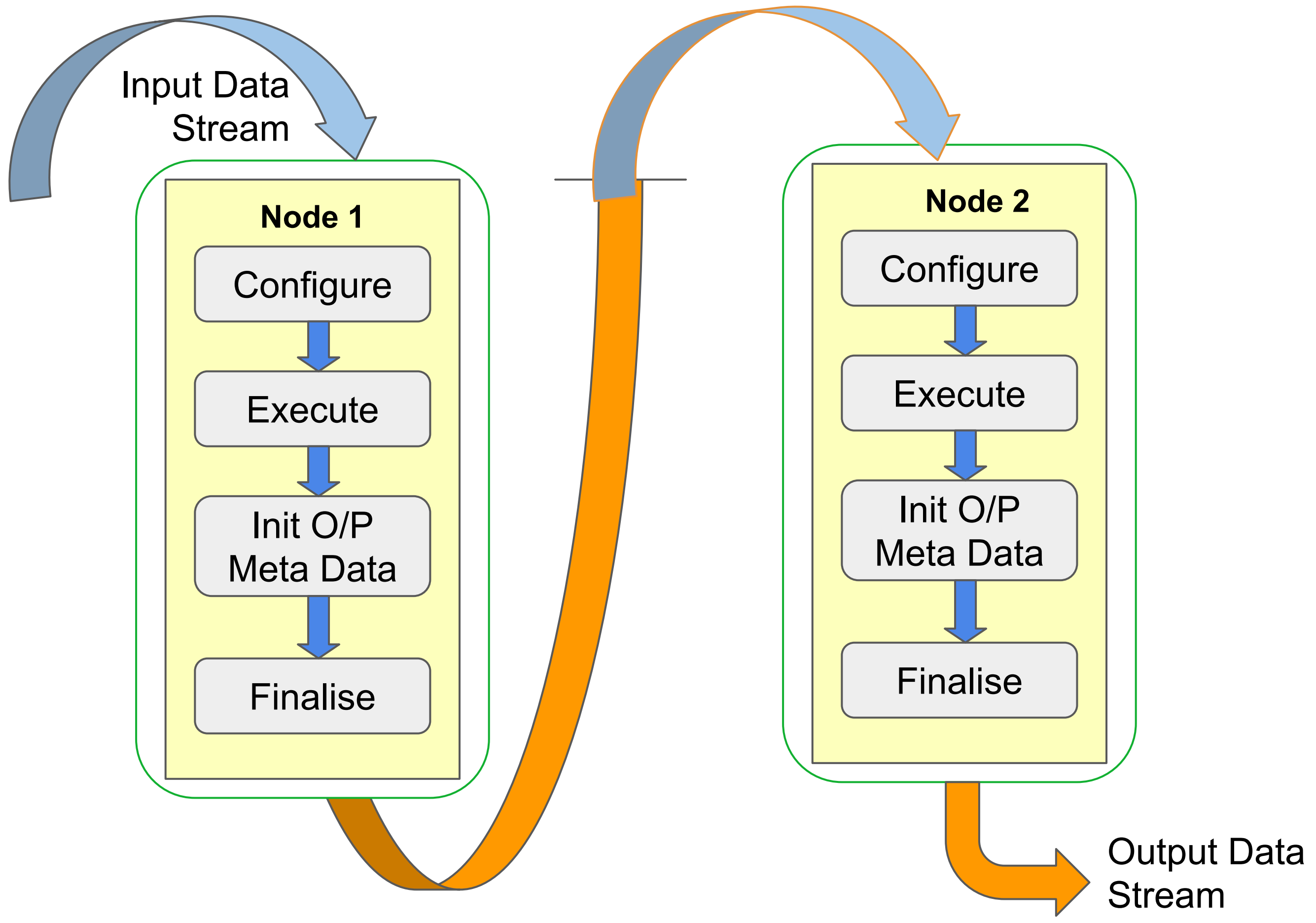
UNBXD

Workflow *Node*



1. A step in the workflow
2. Deployed as a Docker Image
3. Checkpoints states after every step
4. Check Pointed Resume

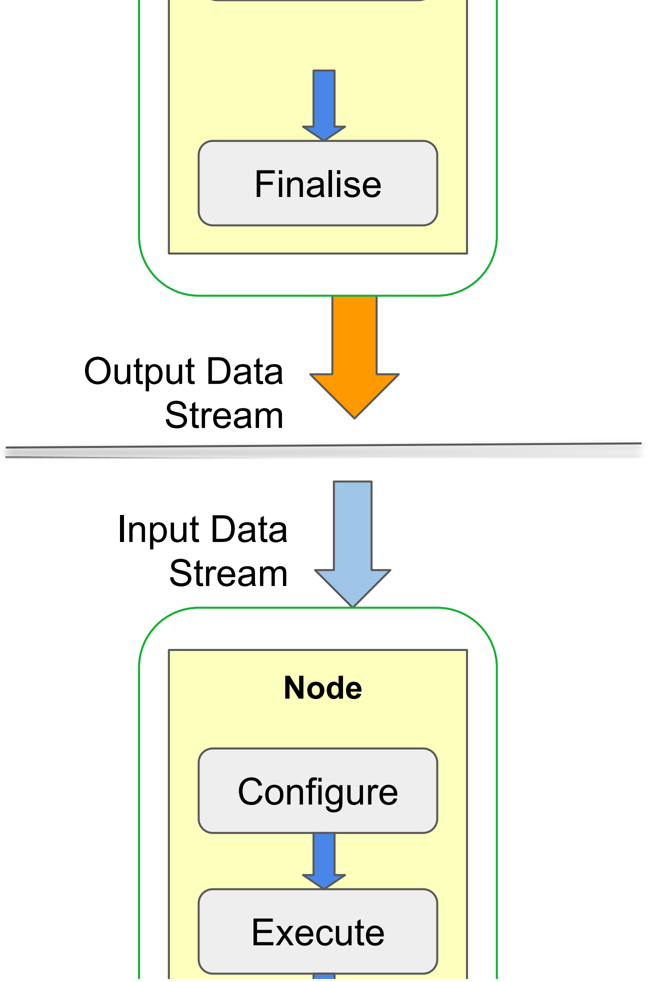
Workflow *Node*



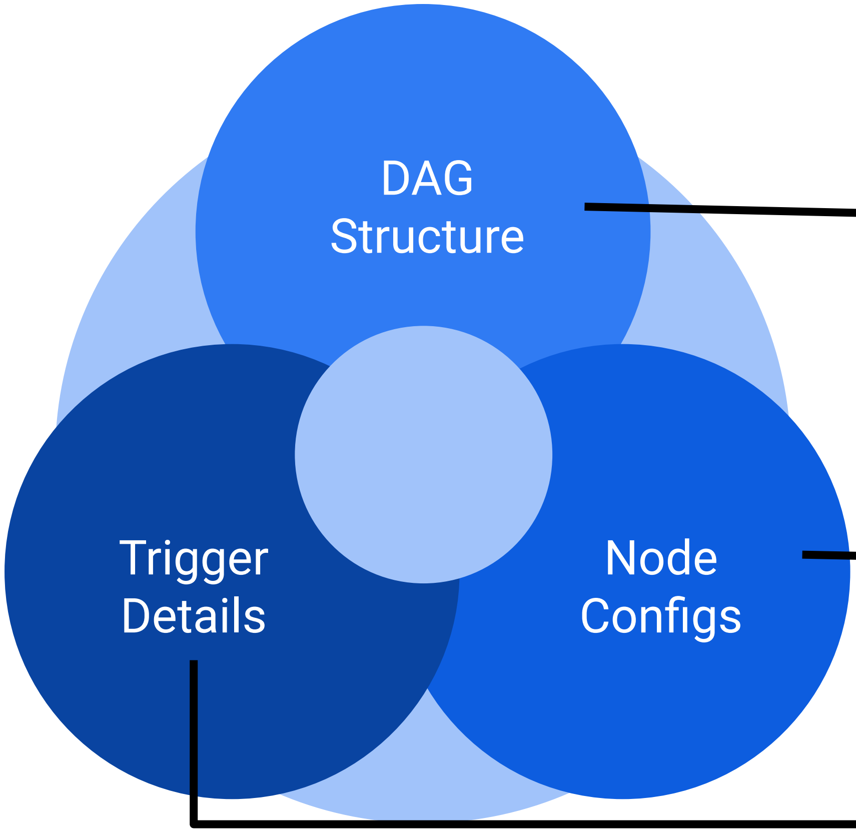
UNBXD

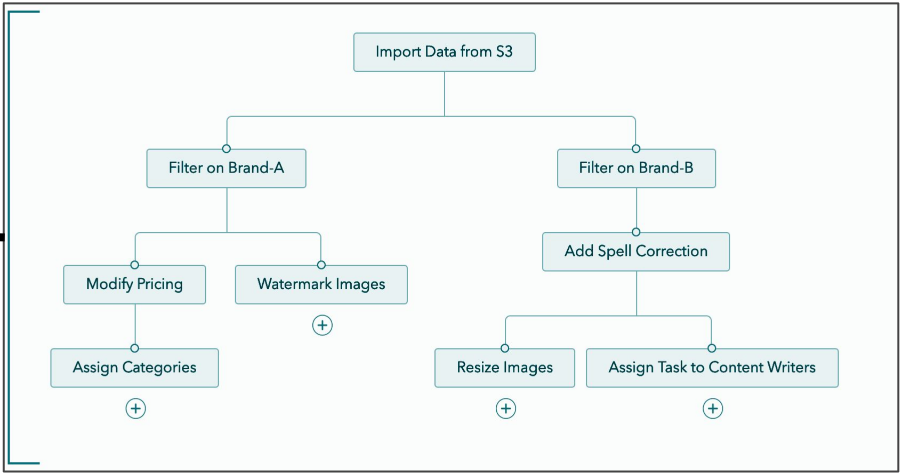
I/O Data Stream

1. A set of meta-data and logic to derive a stream of data.
2. The stream of data should be finite (Bounded)
3. O/P-Data-Stream is encoded in **Init O/P Meta Data** step of a node
4. Decoded in **Configure** step

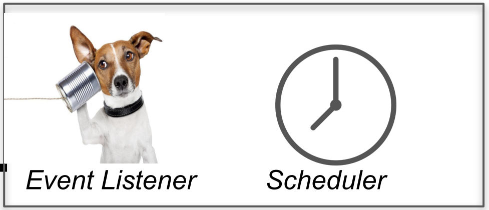


Workflow Configurations



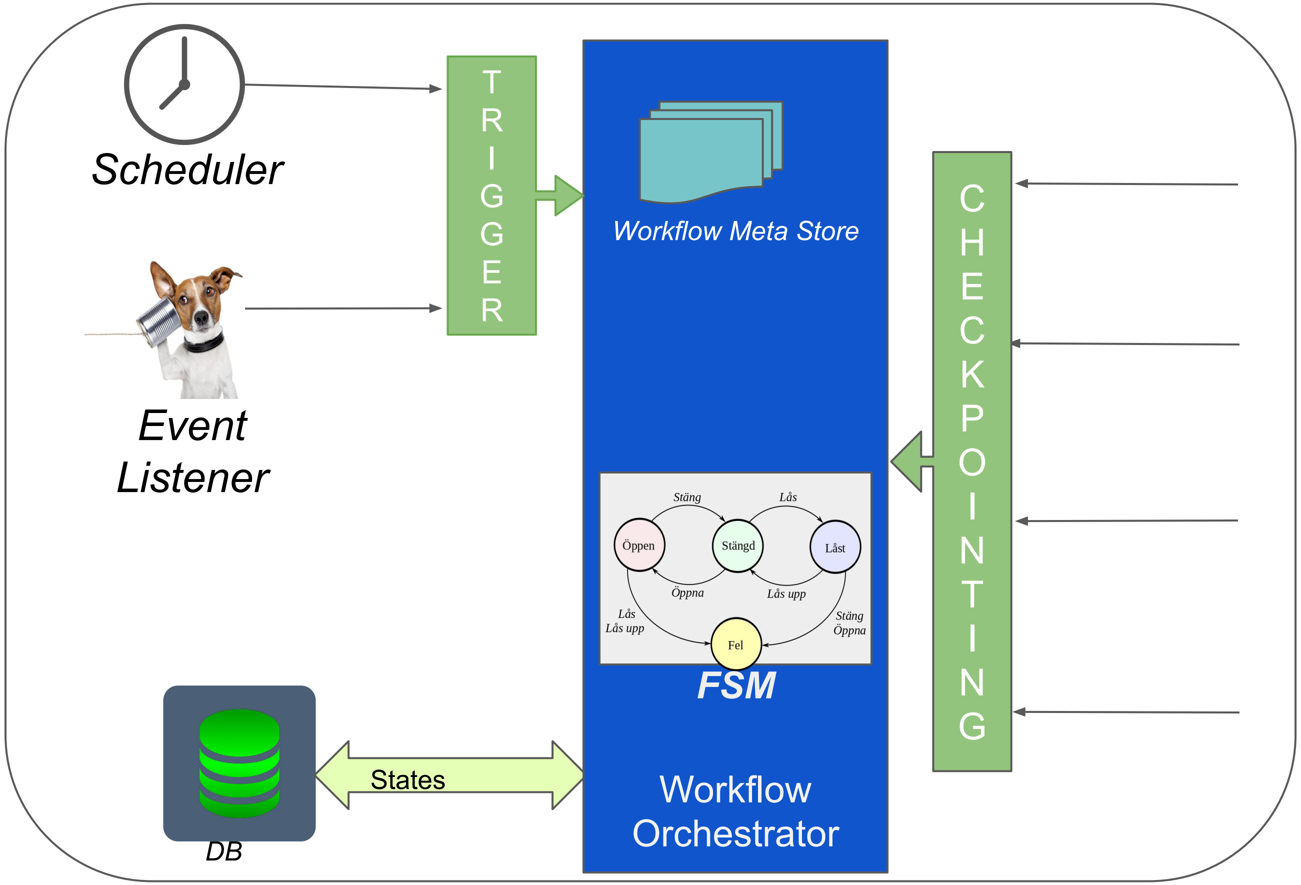


{"Node1":"S3Location", "Dam Endpoint", Start WIth}



UNBXD

Workflow Orchestrator



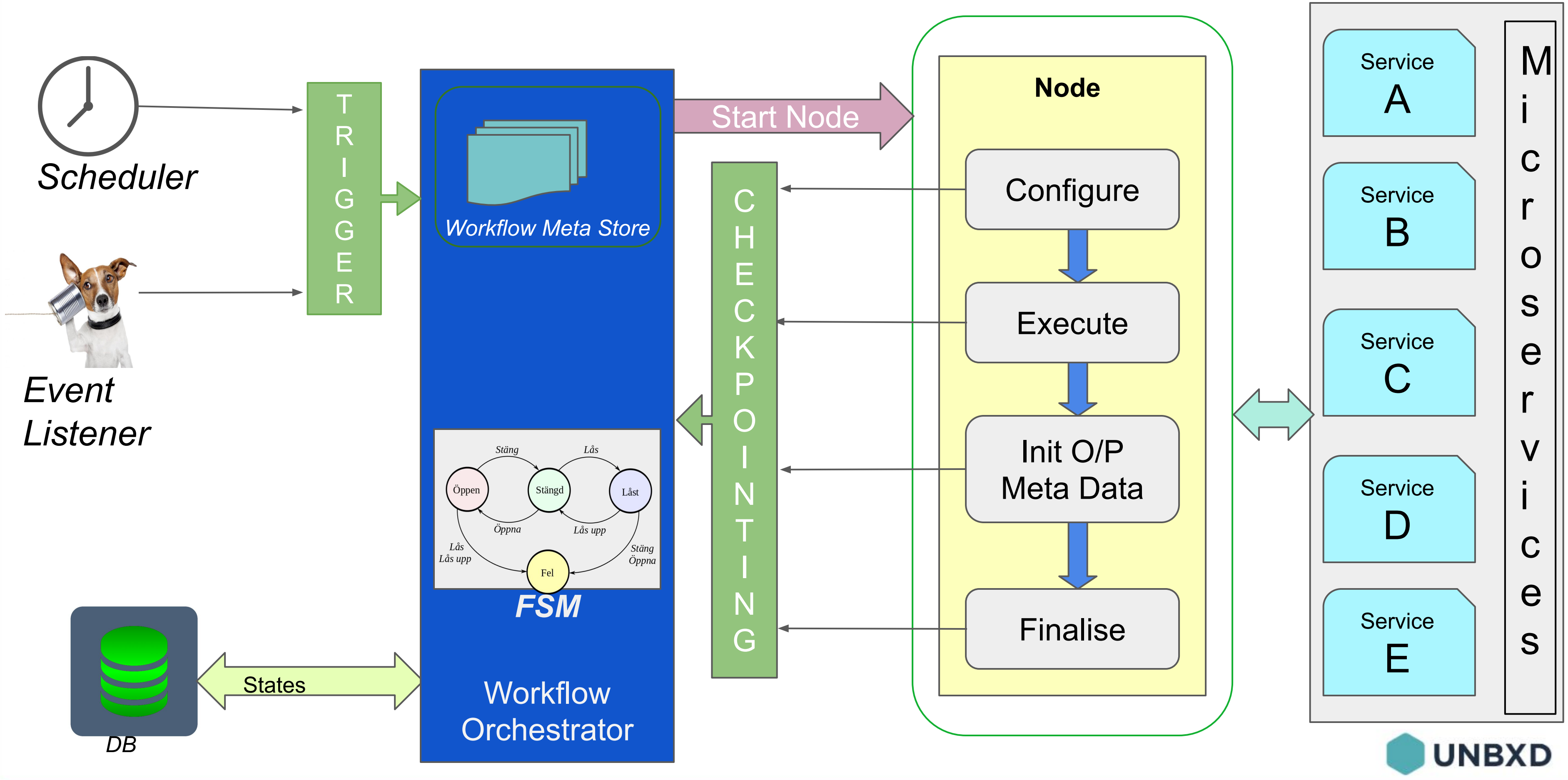
UNBXD

Next..

1. Workflow Overview
2. Workflow Orchestration Engine:-
   - Objectives
   - Components
   - Architecture
3. *Kubernetes and it's features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

UNBXD

The Architecture



UNBXD

Next..

1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Architecture
3. *Kubernetes and it’s features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

UNBXD

Kubernetes

- Container Orchestrator
- Runs and Manages Containers
- Open Source
- Manage Applications not Machines

Kubernetes Jobs

- Represents a finite task
    - ❖ Tasks run to completion
- Supports parallel execution of *Pods*
- Useful for *Large Computations* & Batch-oriented tasks
- Fault Tolerant
    - ❖ Restarts a pod if it fails before completion

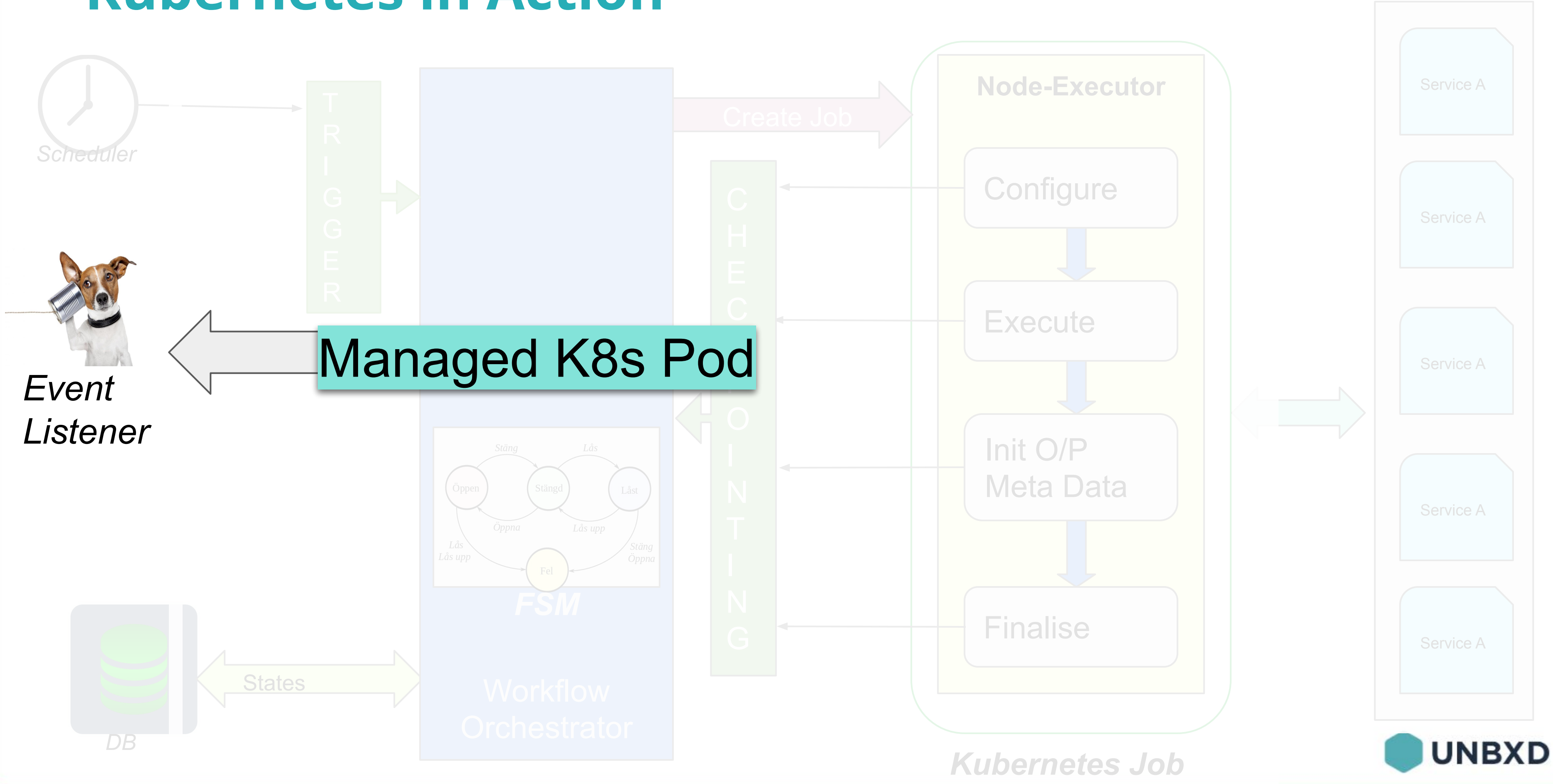
Kubernetes Cron-jobs

- Jobs with a Time Based Scheduling
- Accepts Linux Crontab Expressions
- Exhibits similar Fault Tolerance as Kubernetes Jobs
- Useful for *Repeated Actions*

Next..

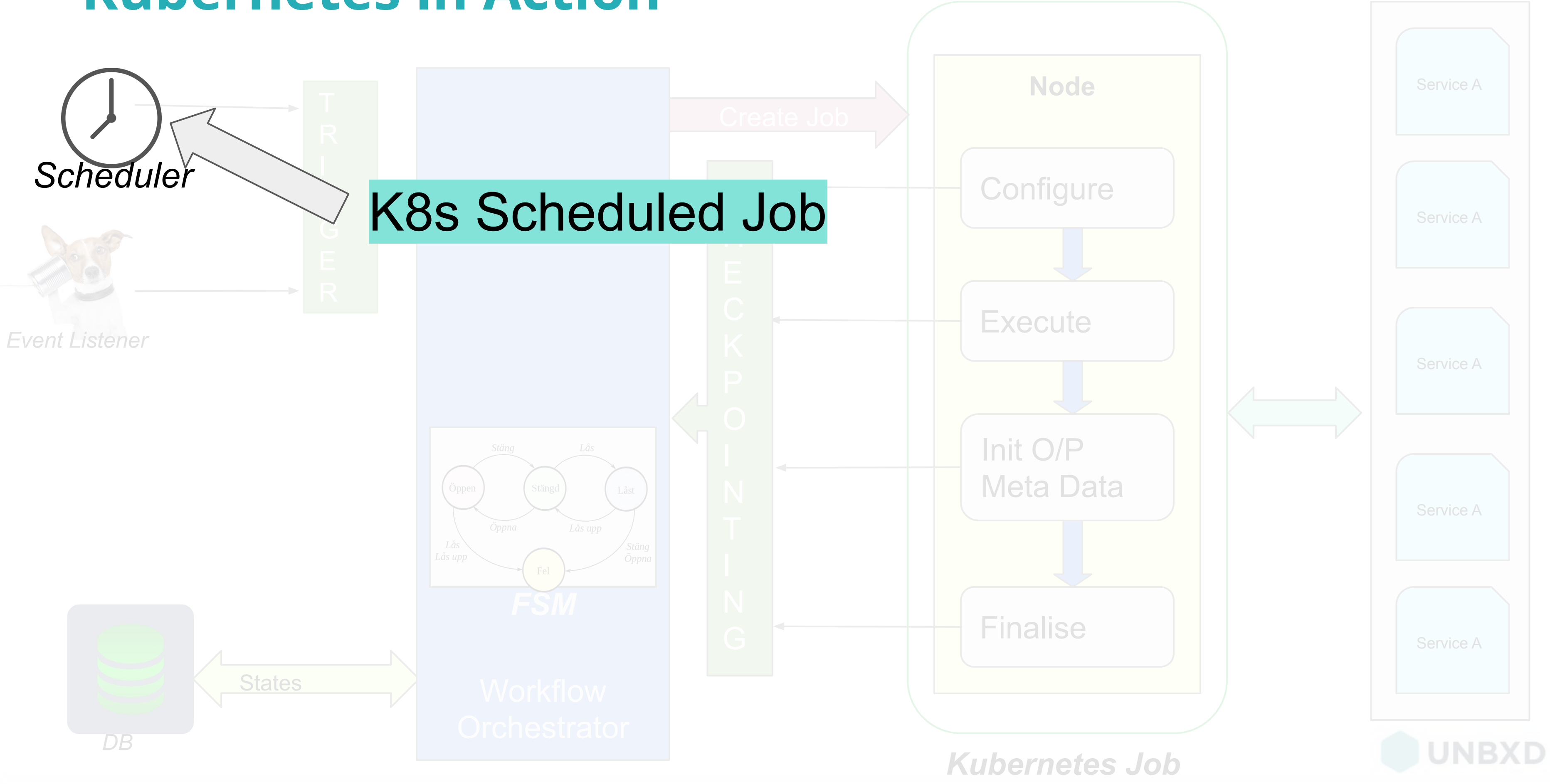
1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Architecture
3. *Kubernetes and it’s features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

Kubernetes in Action

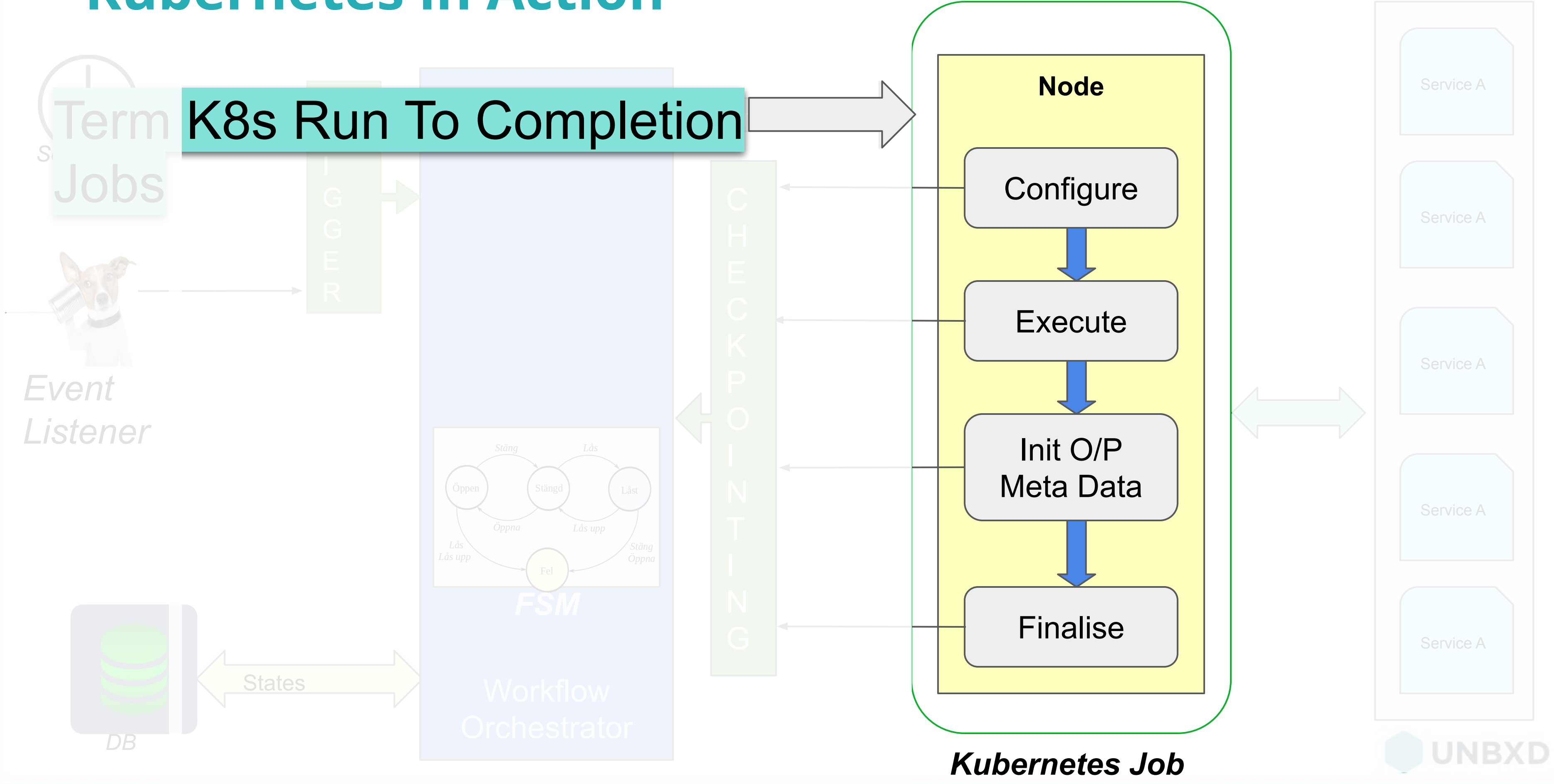


UNBXD

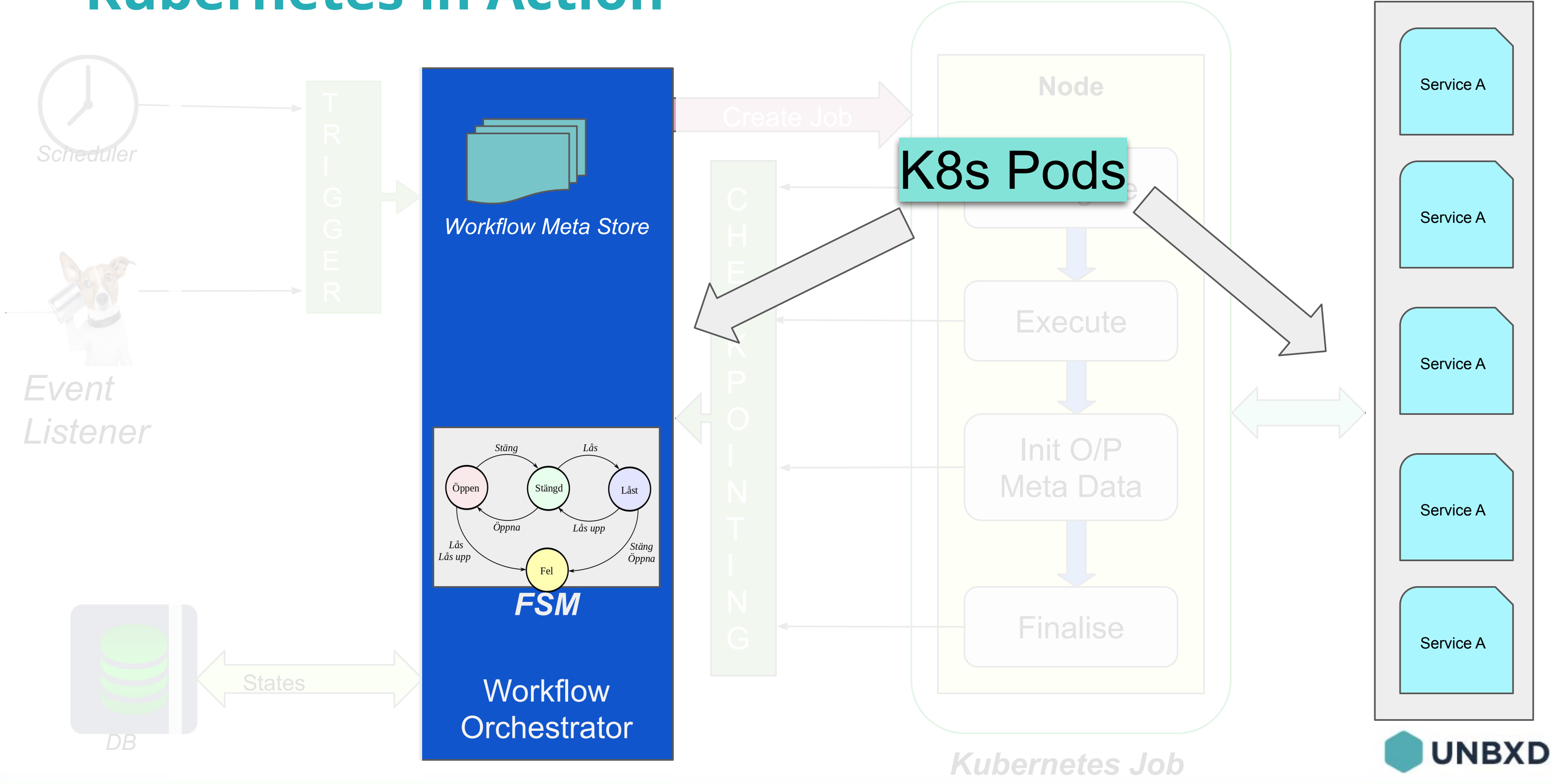
Kubernetes in Action



Kubernetes in Action



Kubernetes in Action



UNBXD

Next..

1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Architecture
3. About *Kubernetes and it's features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

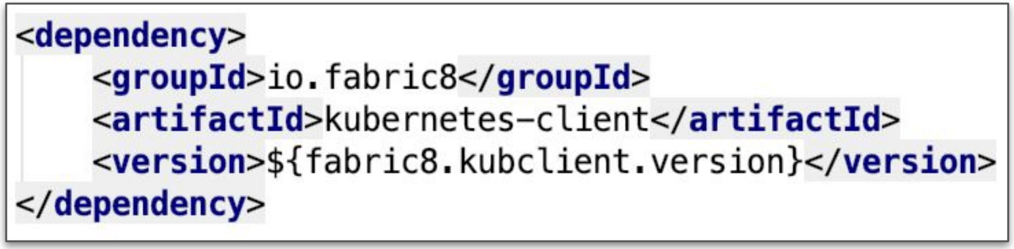
UNBXD

Controlling Kubernetes Programmatically

Fabric8 Kubernetes Client

Gives easy apis as a wrapper for Kubernetes REST APIs

- Create Jobs (One Time/Scheduled)
- List Jobs
- Filter Based On Tags
- Delete Jobs



```xml
<dependency>
    <groupId>io.fabric8</groupId>
    <artifactId>kubernetes-client</artifactId>
    <version>${fabric8.kubclient.version}</version>
</dependency>
```

UNBXD

What happens at scale?

- Every Node Executes Independently
- Kubernetes Cluster scales by adding new nodes
- Cluster Size ~ Number of Nodes Running in Parallel
- Scaling by Rate Limiting

UNBXD

Next..

1. Workflow Overview
2. Workflow Orchestration Engine:-
    - Objectives
    - Components
    - Architecture
3. About *Kubernetes and it’s features*
4. Kubernetes for *Workflow Orchestration Engine*
5. Controlling *Kubernetes Jobs* Programmatically
6. Best Practices

UNBXD

Some Good Practices

- Orchestration Engine should be made platform agnostic
- Clean up the pods in K8s, use TTL
    - See： TTL Controller, (*`.spec.ttlSecondsAfterFinished`*)
- For massively parallel jobs, use external engines like Spark, Flink

My Team

UNBXD

Thank You!

asingh2411@gmail.com

@asingh2411

UNBXD