GitOps for Machine Learning

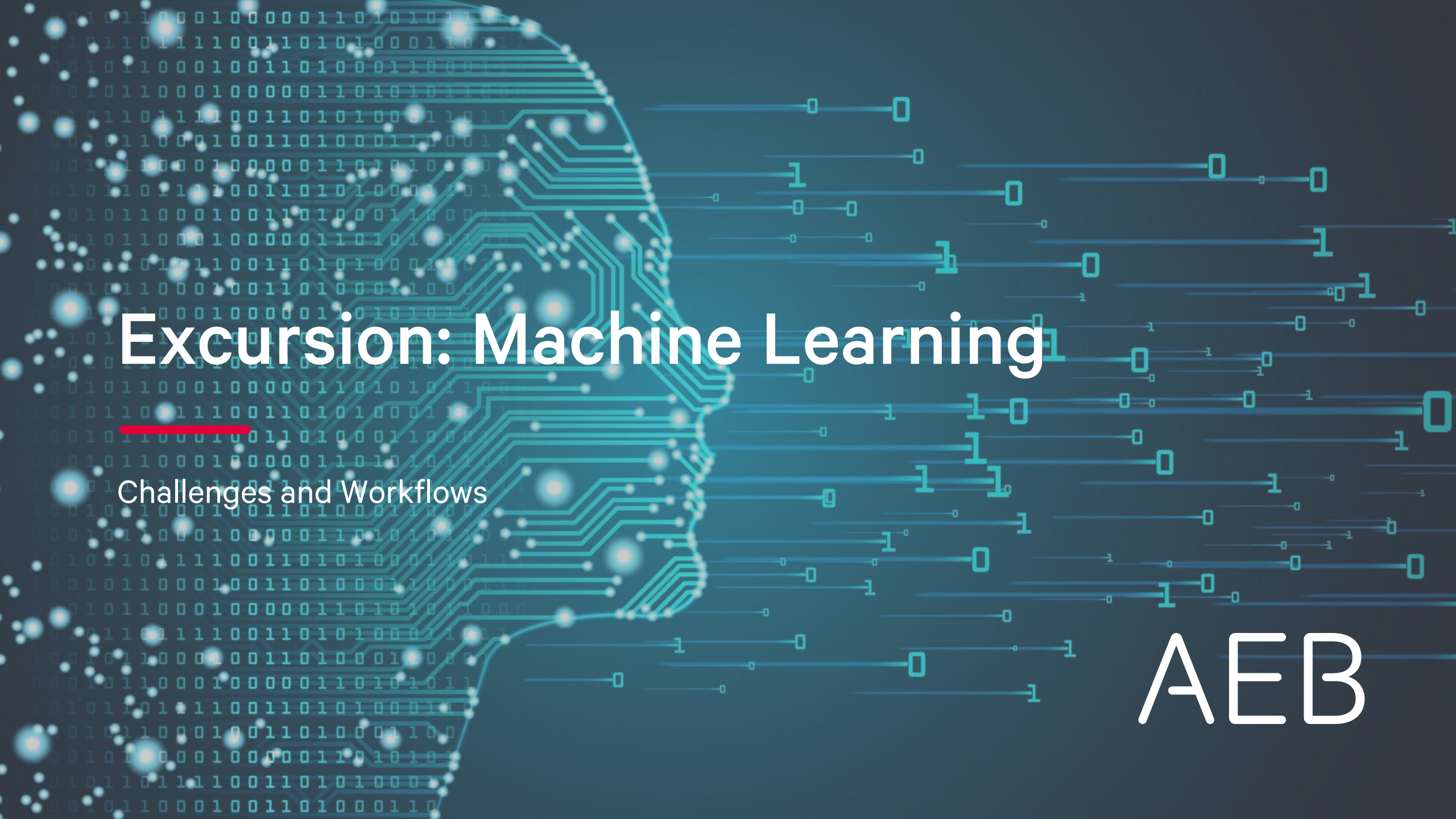
Sebastian Moritz

AEB SE

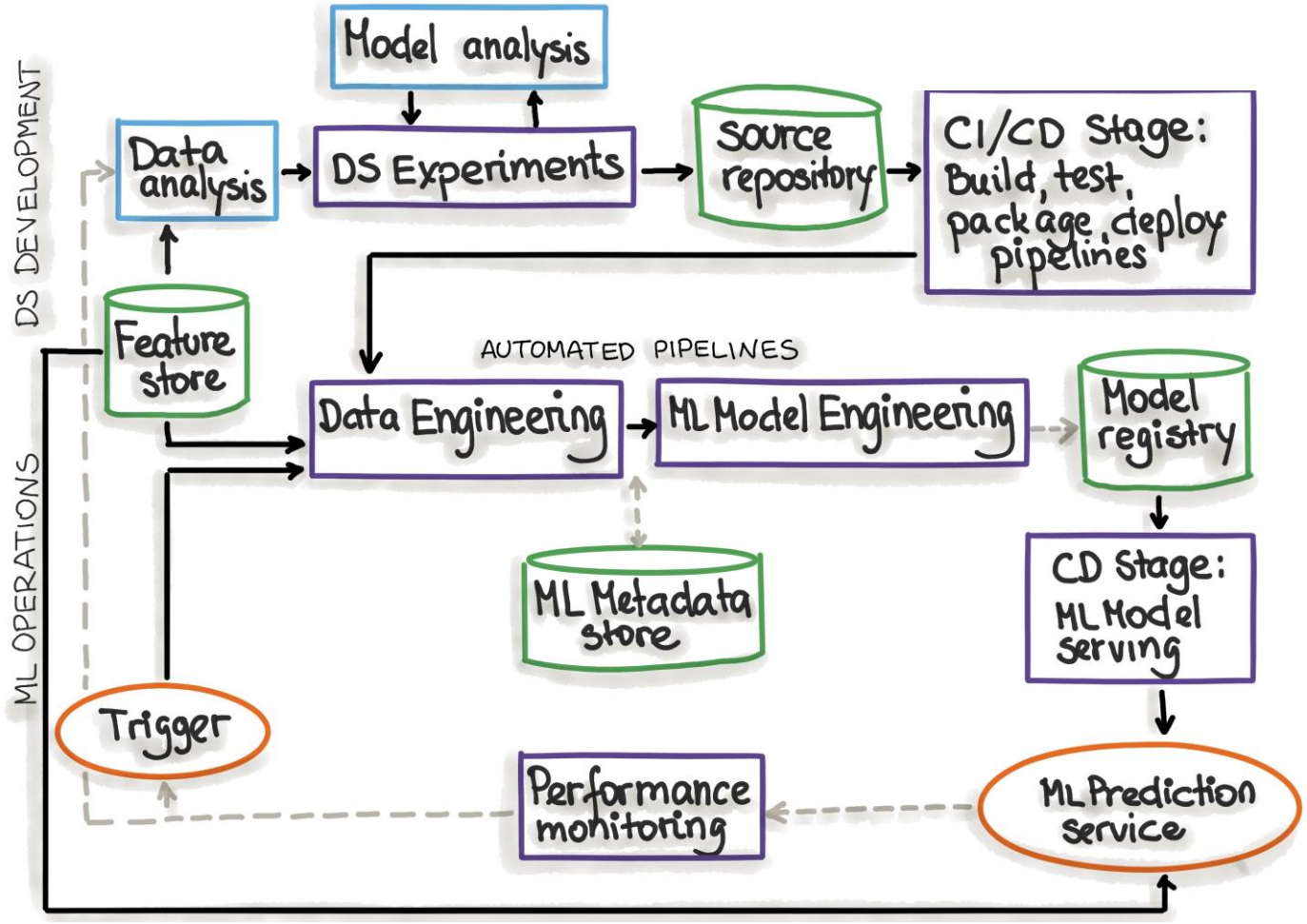
DevOps Engineer

Java Forum Stuttgart 2022

AEB


Excursion: Machine Learning
Challenges and Workflows
AEB

Machine Learning to Production



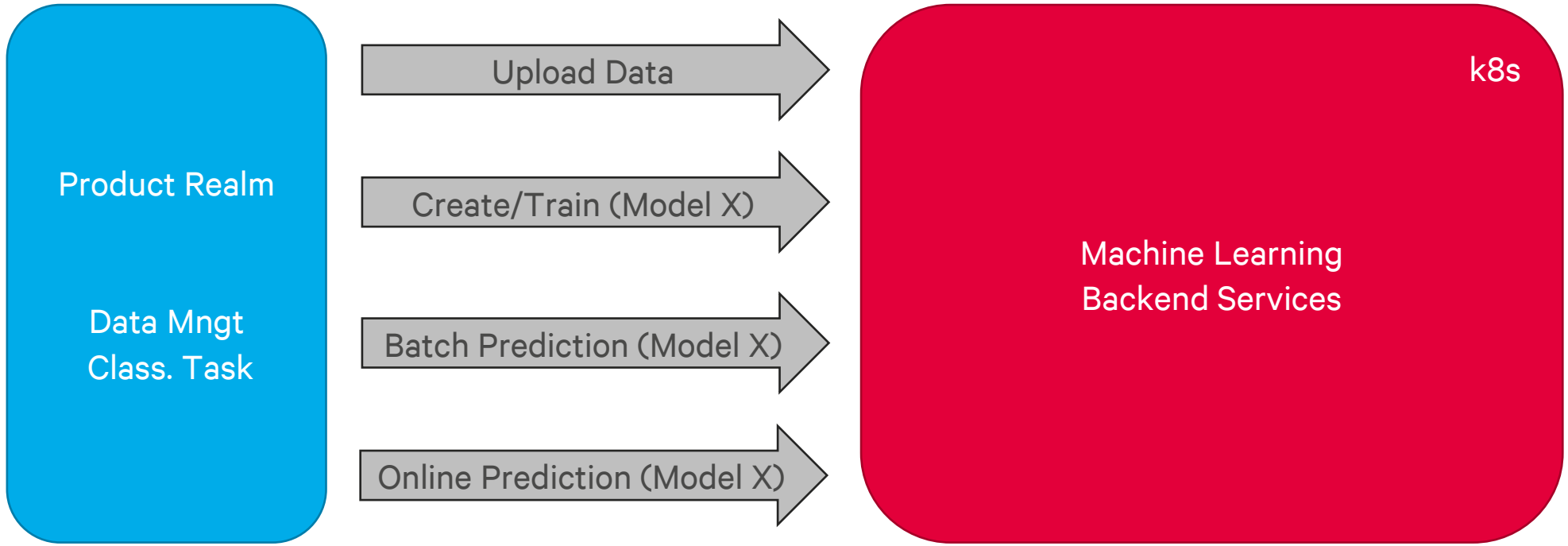
Different

- Teams & Processes
- Tools & Frameworks
- Technologies
- Programming Languages
- Product Integrations
- Operation Model

[https://ml-ops.org/content/mlops-principles]

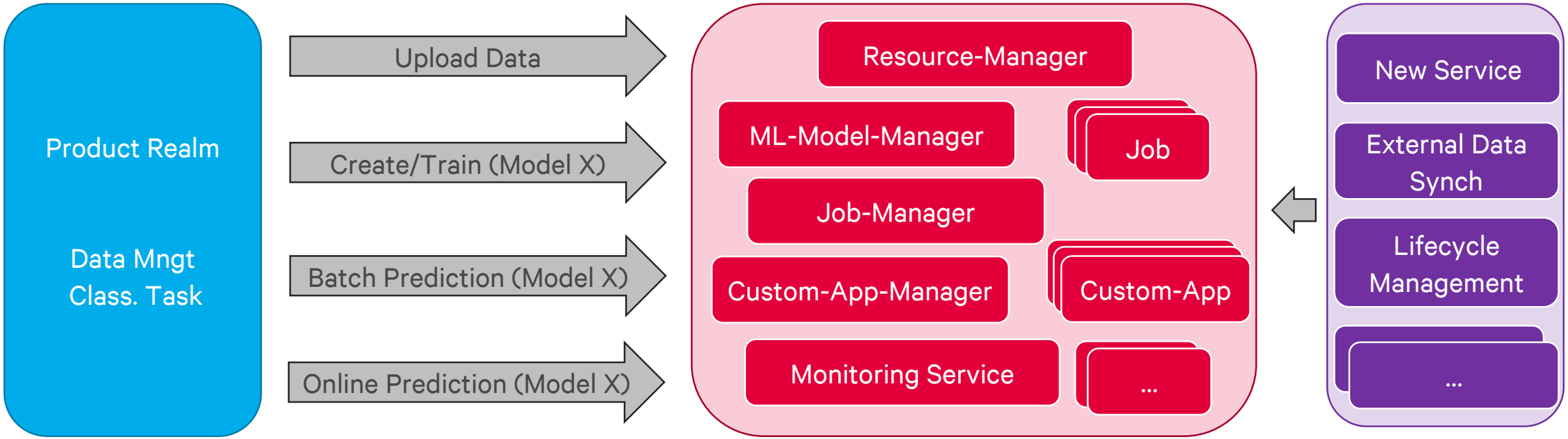
Examplary System Architecture

- Everything assessable via REST api
- Avoid manual onboarding and configuration



Microservices

How to keep things running?



01

GitOps

Theory & Experiences

AEB

GitOps Principles

1. Declarative 

A system managed by GitOps must have its desired state expressed declaratively.

2. Versioned and Immutable 

Desired state is stored in a way that enforces immutability, versioning and retains a complete version history.

3. Pulled Automatically 

Software agents automatically pull the desired state declarations from the source.

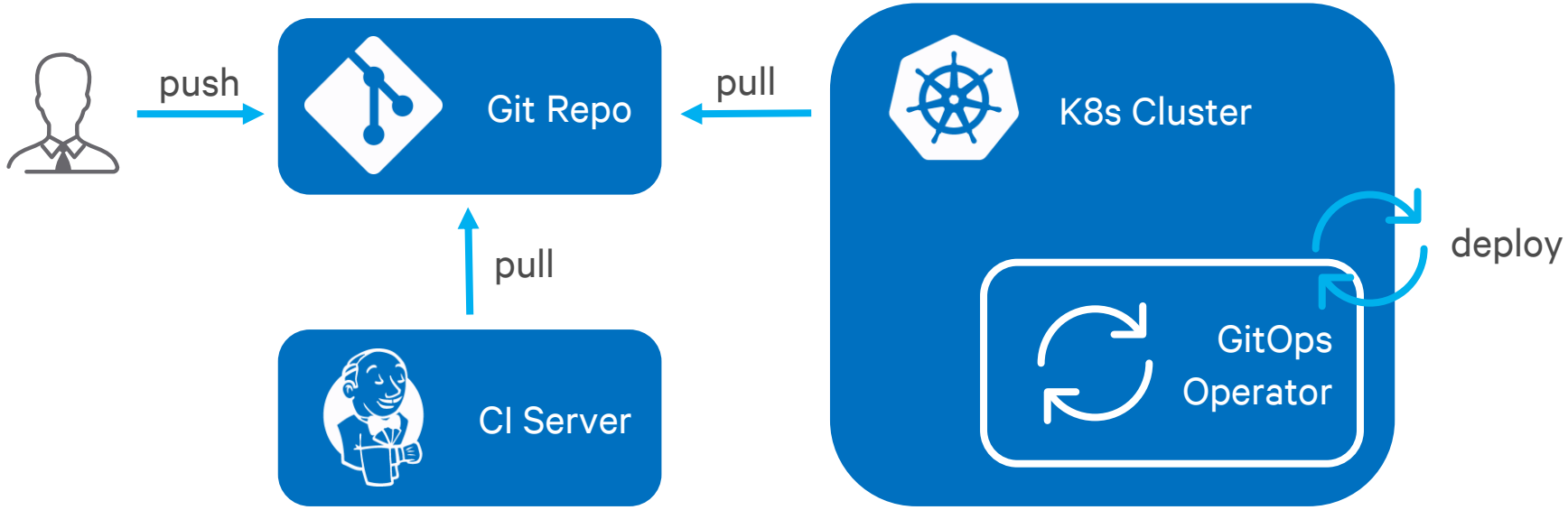
4. Continuously Reconciled 

Software agents continuously observe actual system state and attempt to apply the desired state.

[https://opengitops.dev]

GitOps

- Git as single source of truth for the desired state of the system
- Control loop compares desired and actual state to pull changes and enforce convergent atomic updates



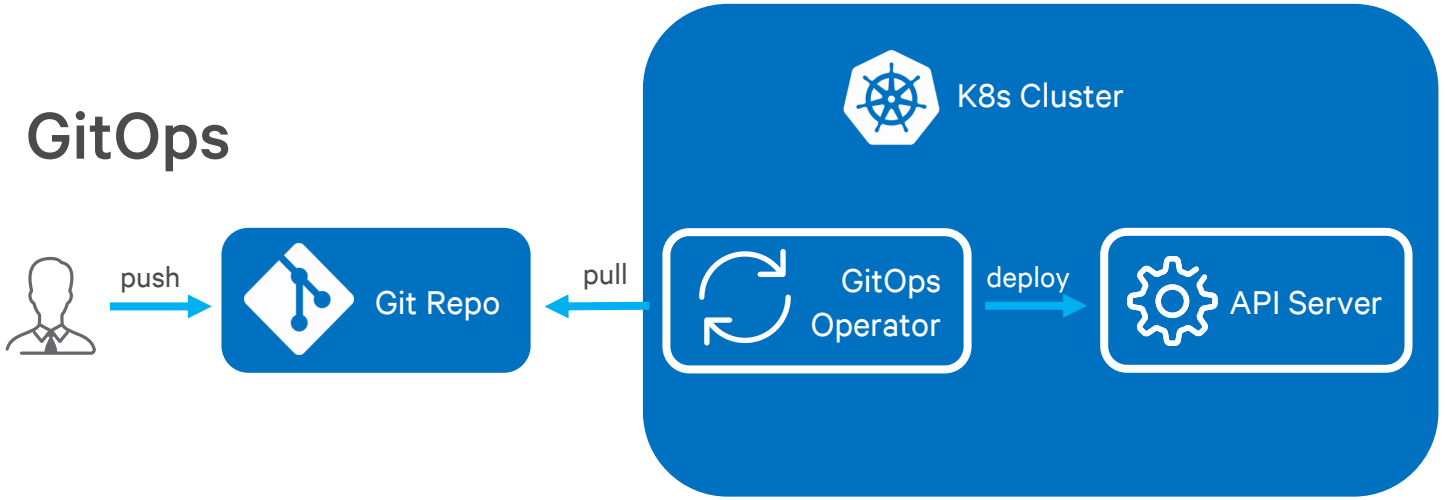
Classic CI/CD vs. GitOps

CI-Ops

Push

- Imperative
- One-shot

GitOps



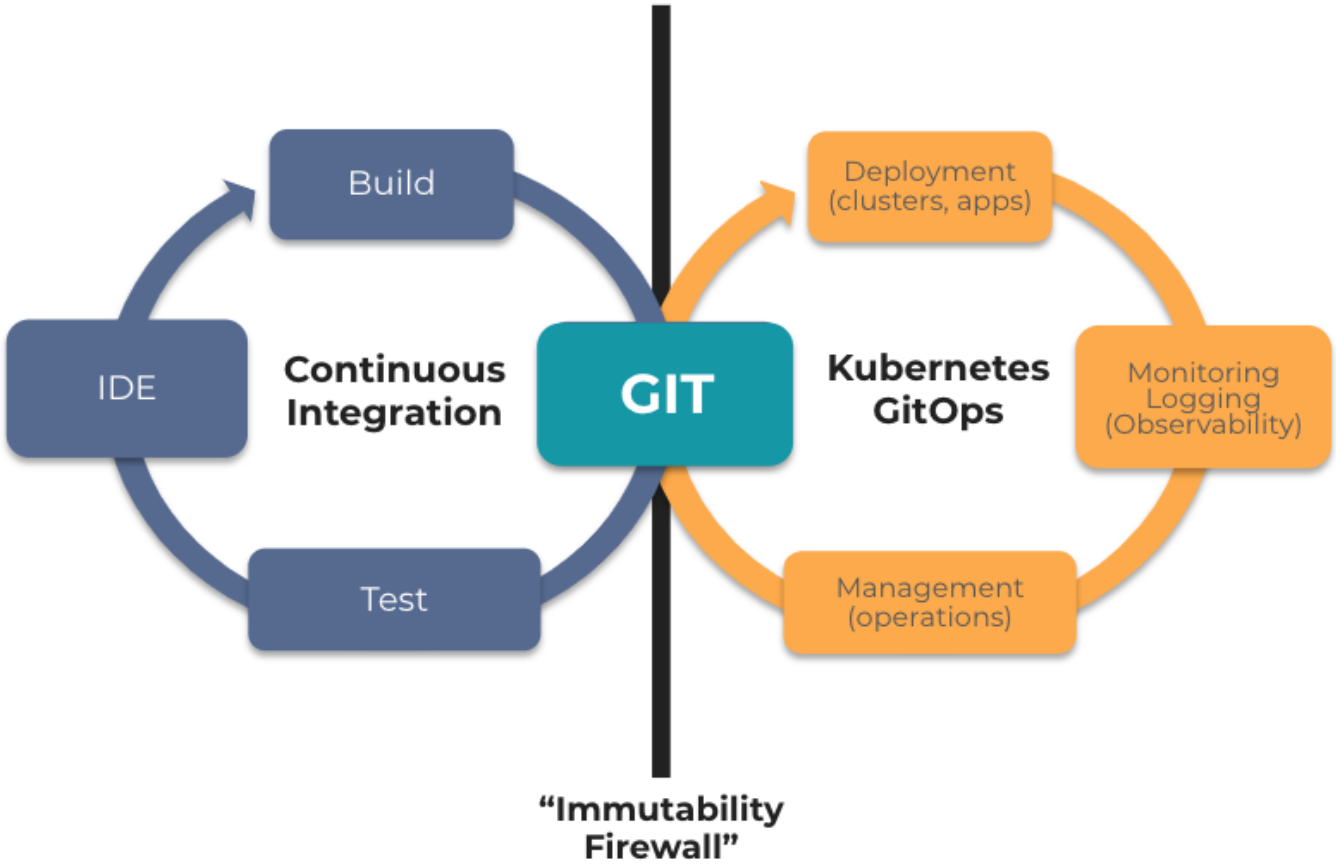
Pull

- Declarative
- Reconcile loop

Advantages of GitOps

- Forces declarative description
  Scalability, automation, self-healing, recoverable...
- Transparency
  Cluster updates as sequence of atomic transactions, rollback...
  Auditable, succeed or fail cleanly
- Developer experience, DevOps velocity
- CI tooling has no production system access
- CD tooling retains credentials inside the cluster

GitOps - Cloud Native Operating Model



Unifying Deployment, Monitoring and Management.

Git as the single source of truth of a system's desired state

ALL intended operations are committed by pull request

ALL diffs between intended and observed state with automatic convergence

ALL changes are observable, verifiable, and auditable

[https://www.weave.works/blog/migrating-legacy-apps-to-kubernetes-with-gitops]

DevOps vs. GitOps

- DevOps: It’s about a mindset

  Collaboration of formerly separate groups/teams/responsibilities
- GitOps: It’s about operations

  Git as developer tool, but focus on operating model
- GitOps could be used with or without DevOps

„The right way to do DevOps“

Alexis Richardson

Excursion: Kubernetes

A practical view

AEB

What is kubernetes?

- Container orchestrator

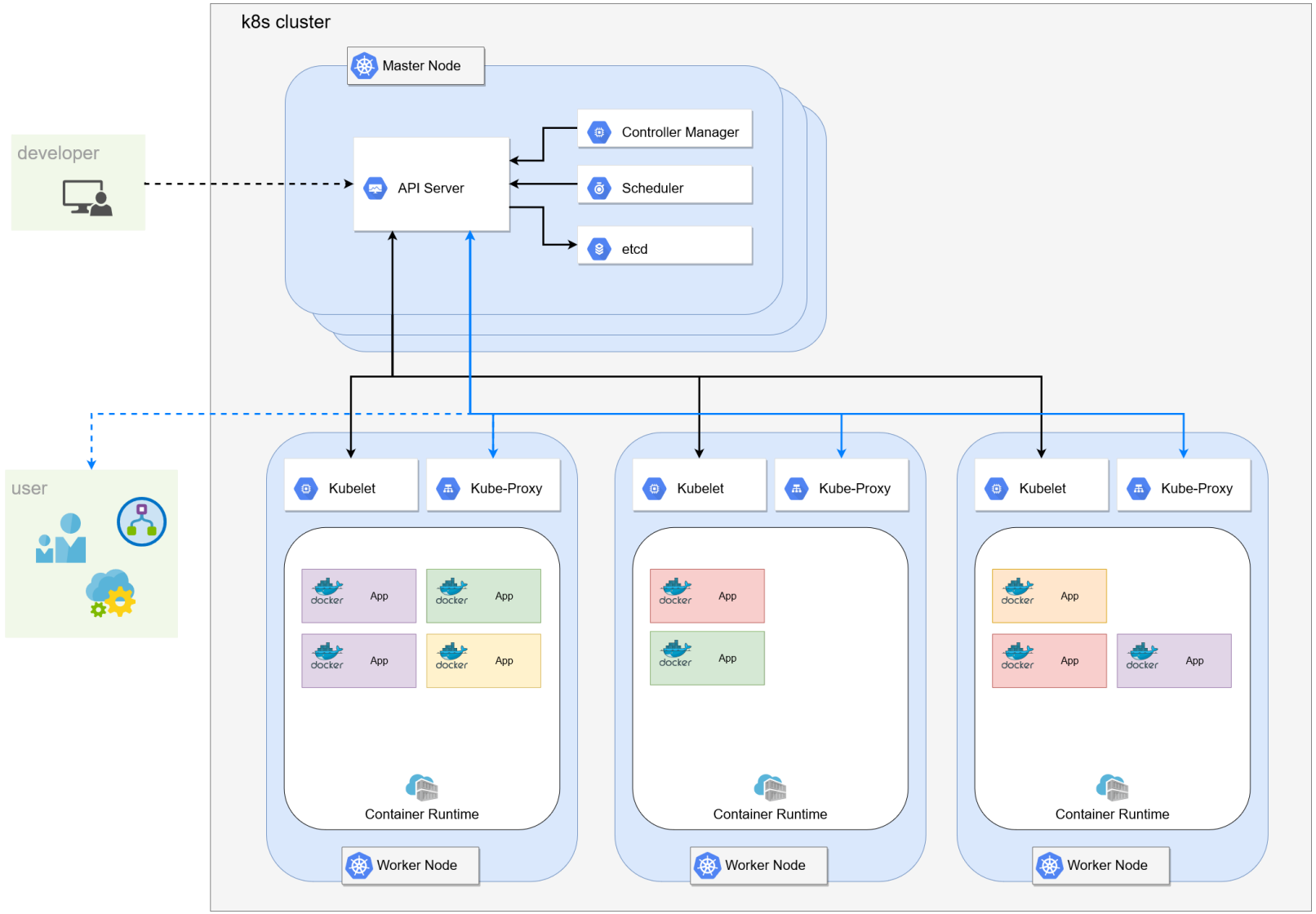
  Defacto standard for cloud operations
- Open source, CNCF

  Avoid vendor lock-in
- One communication central

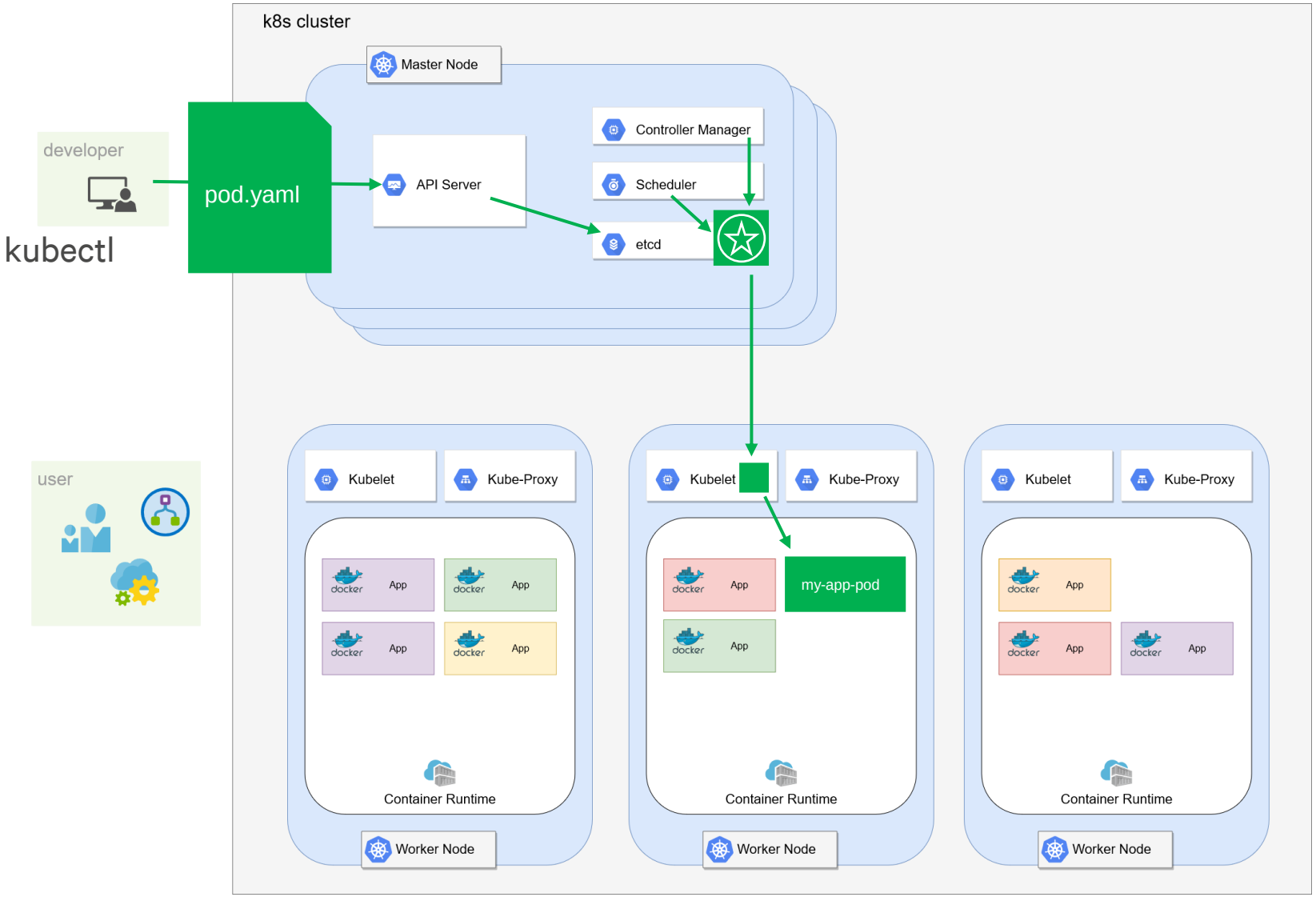
  kube-api server, YAML format
- Controller pattern

  Atomic update approach to establish desired state of system

Kubernetes Architecture



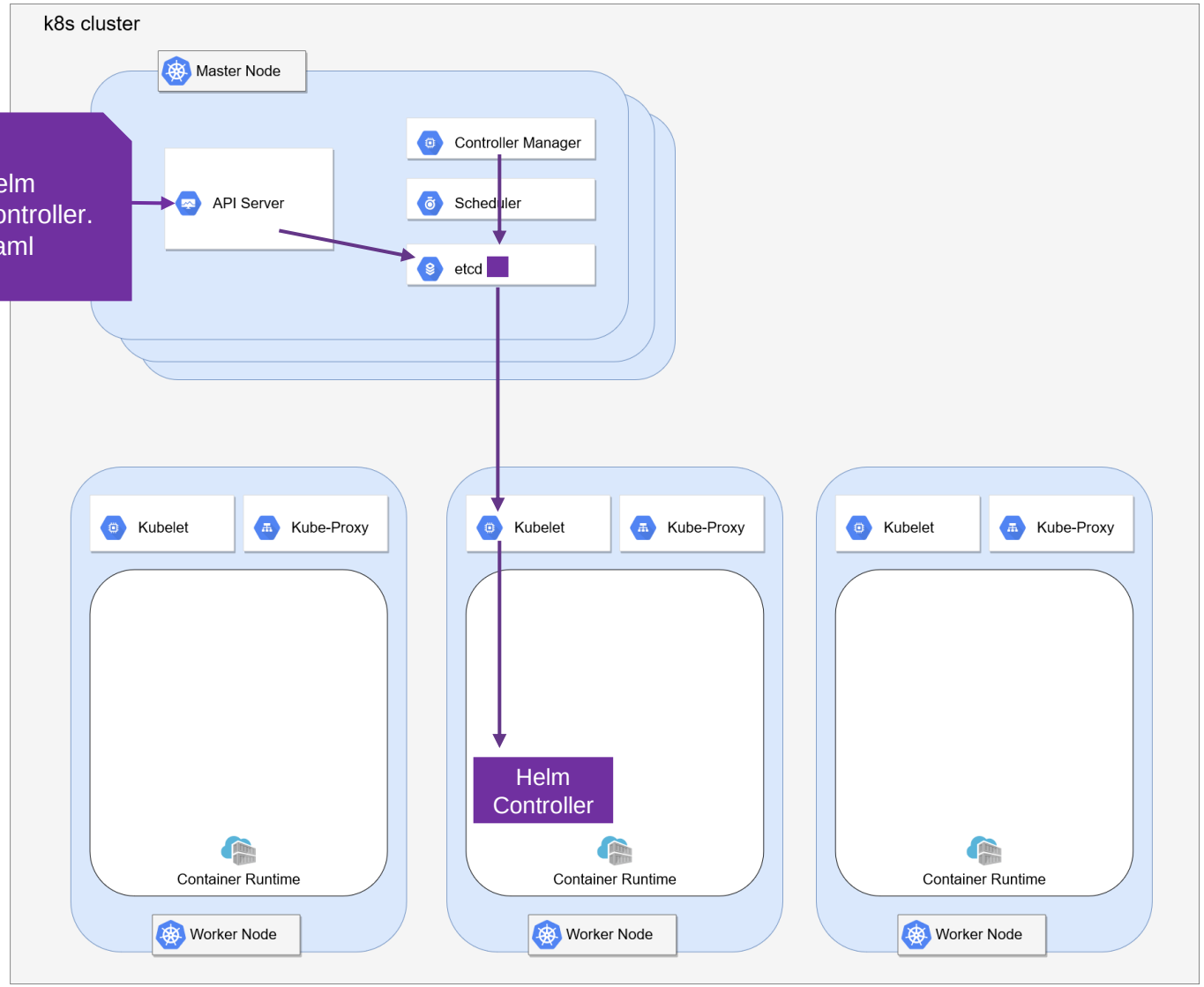
Kubernetes Architecture




```
apiVersion: v1
kind: Pod
metadata:
  name: my-app-pod
  labels:
    foo: bar

spec:
  containers:
  - name: my-app
    image: my-app:2.4
```

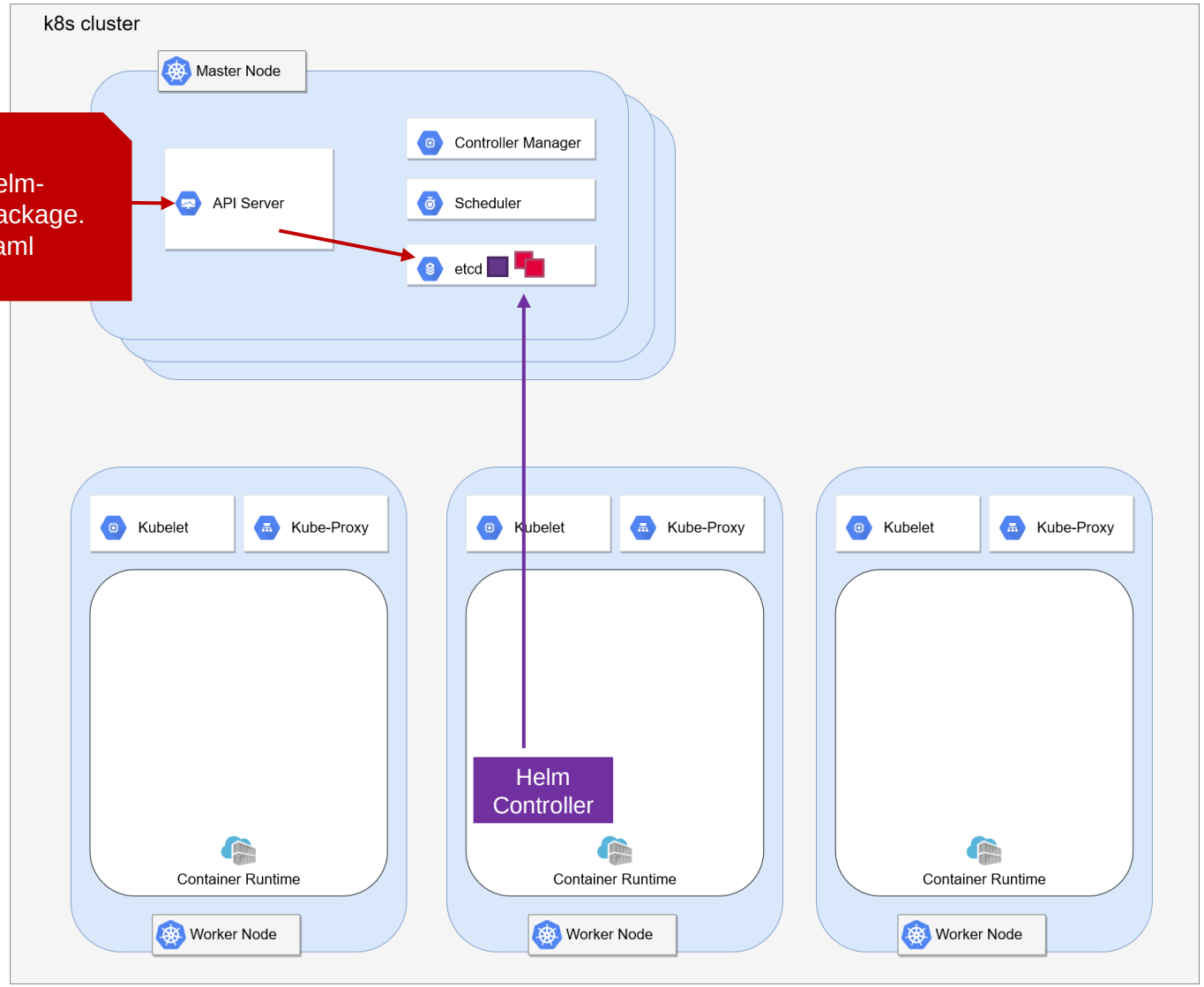
GitOps Controller




```
apiVersion: v1
kind: Pod
metadata:
  name: helm-controller
  labels:
    controls: HelmPackage

spec:
  containers:
  - name: helm-controller
    image: helm-cntl:3.1
```

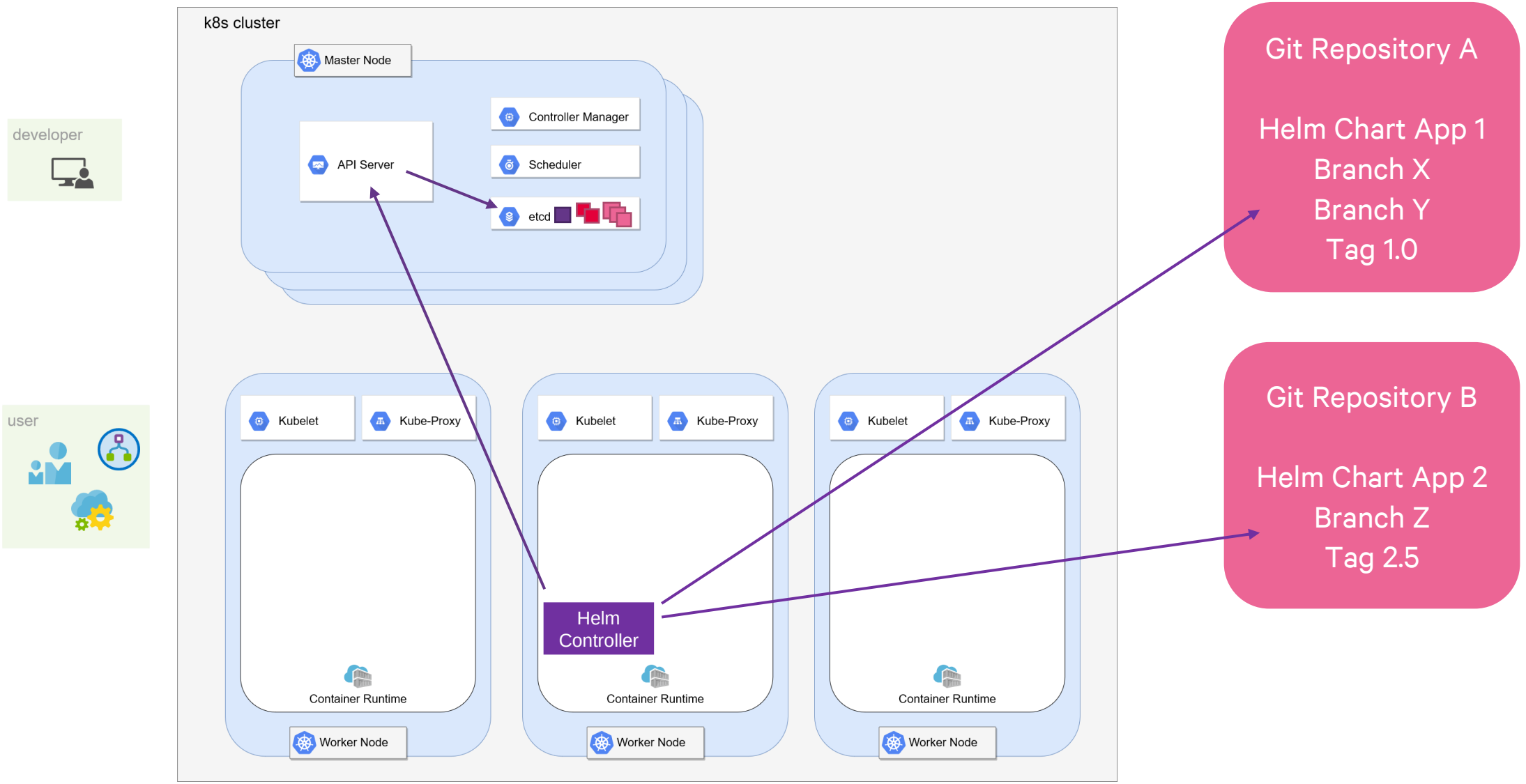
GitOps Controller




```
apiVersion: v1
kind: HelmPackage
metadata:
  name: my-app-package
  labels:
    use: helm-controller

spec:
 chart:
    git: //app1-chart.git
    version: release-3.12
  values:
    foo: bar
```

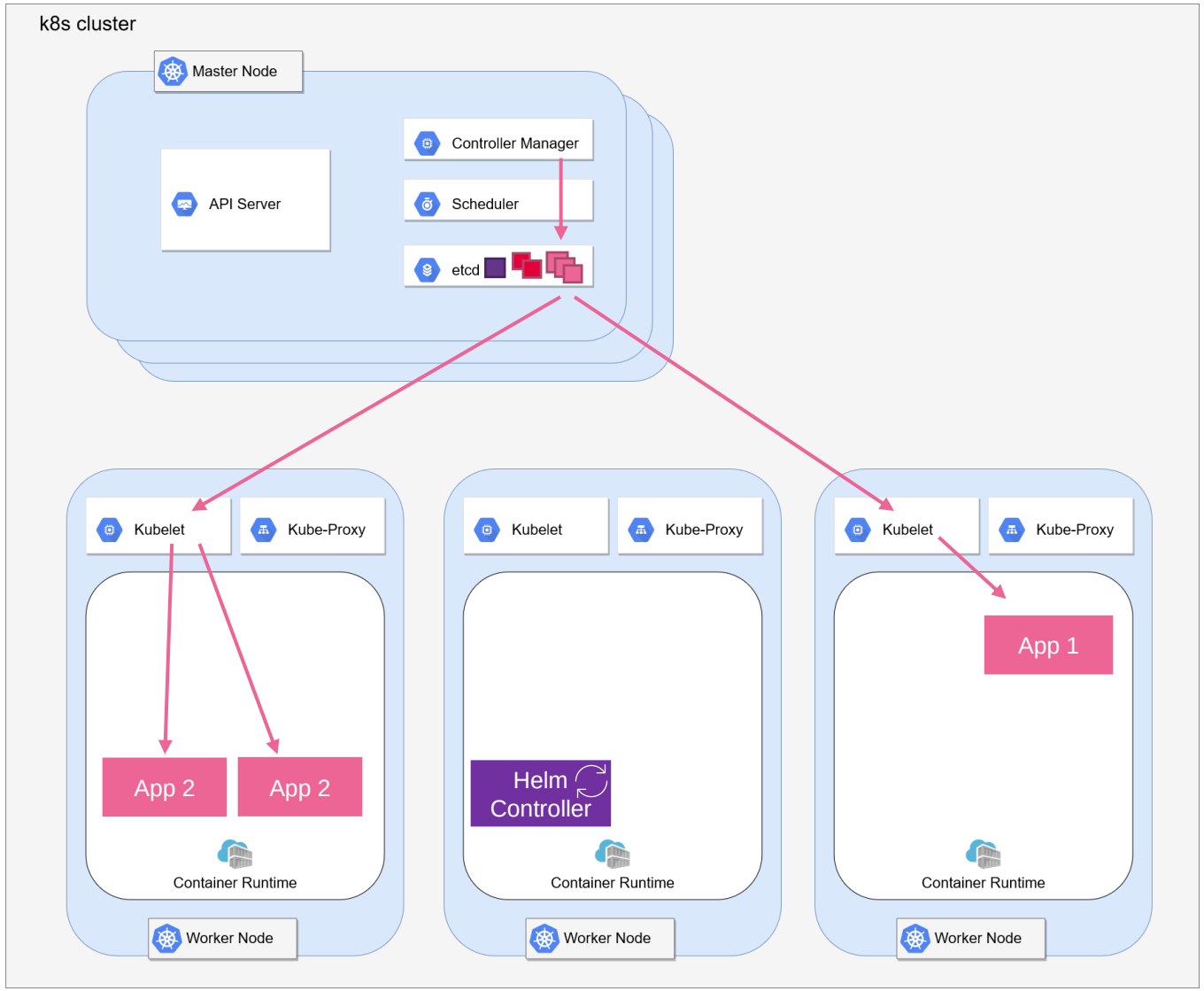
GitOps


developer
user
k8s cluster
Master Node
Controller Manager
API Server
Scheduler
etcd
Kubelet
Kube-Proxy
Kubelet
Kube-Proxy
Kubelet
Kube-Proxy
Helm Controller
Container Runtime
Container Runtime
Container Runtime
Worker Node
Worker Node
Worker Node
Git Repository A
Helm Chart App 1
Branch X
Branch Y
Tag 1.0
Git Repository B
Helm Chart App 2
Branch Z
Tag 2.5

GitOps

developer

user



Git Repository A

Helm Chart App 1
Branch X
Branch Y
Tag 1.0

Git Repository B

Helm Chart App 2
Branch Z
Tag 2.5

Controller Pattern

- Controller Pattern
  - Nginx: Ingress-Controller
  - Storage: S3/NFS-Controller
  - Jenkins Server as Controller of build agents
- Cluster GitOps:
  - Declarative description of a k8s cluster (i.e. ClusterAPI)
  - Deployment of „cluster controllers“ and rollout of resources (CRDs)
  - Continuous synchronization via GitOps Controller

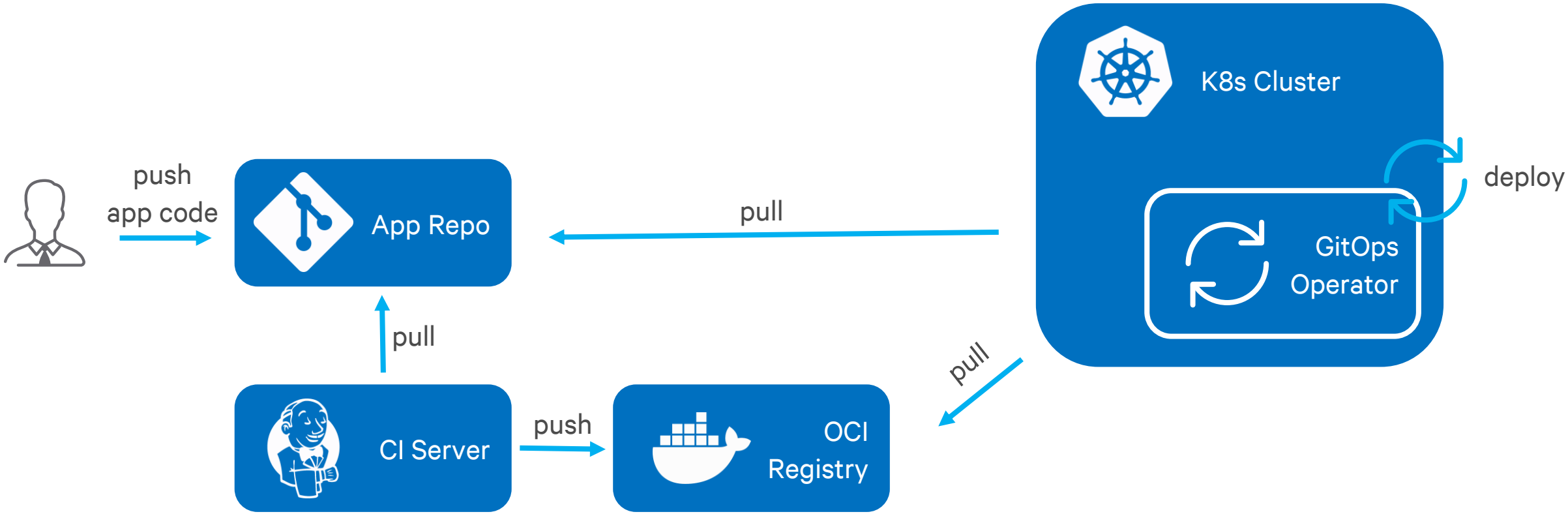
Back to GitOps

Now with a plan

AEB

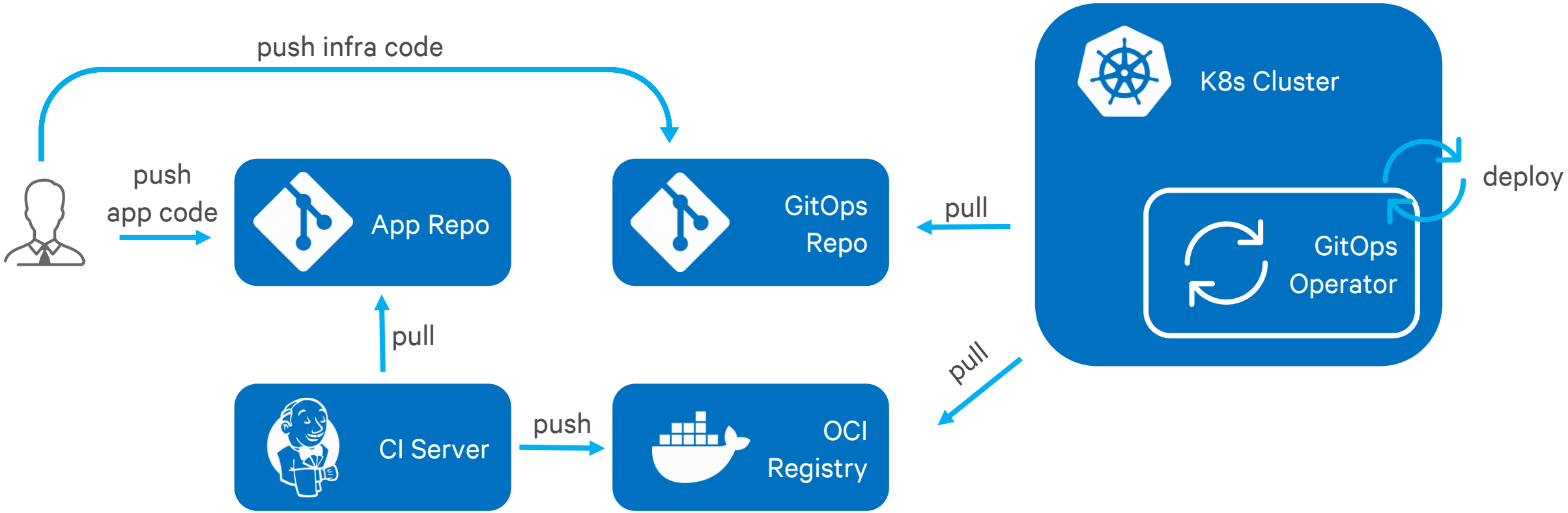
Evolution of GitOps Processes

Basic GitOps approach



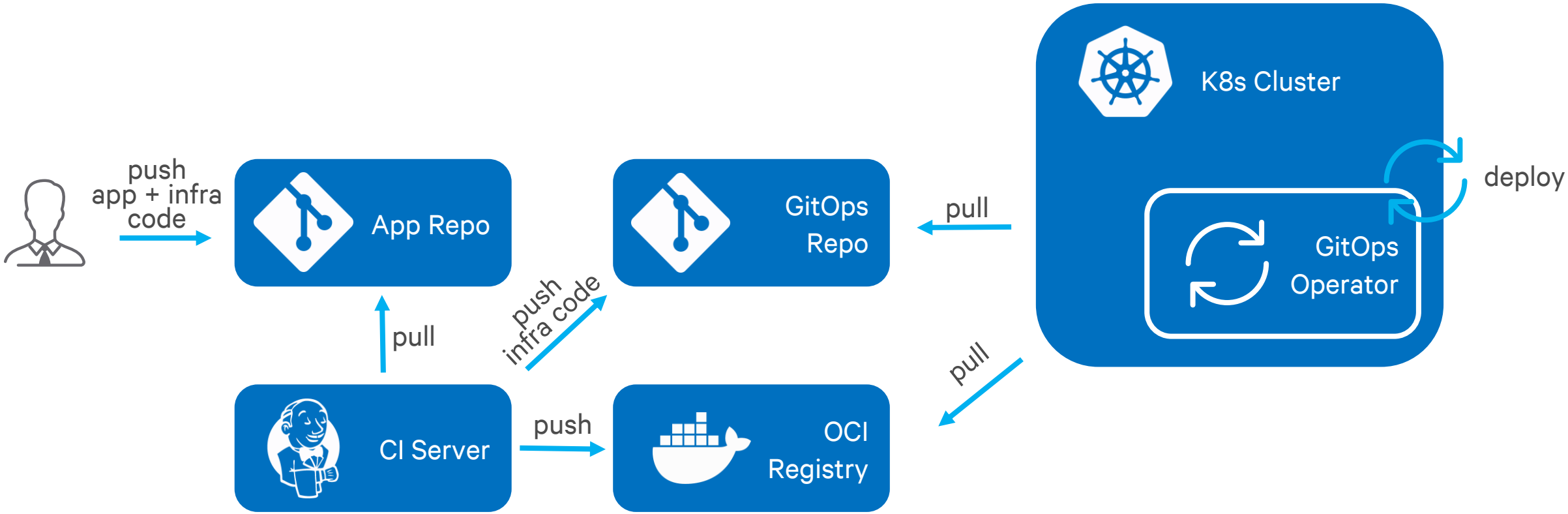
Evolution of GitOps Processes

Split configuration from application code in dedicated GitOps repo

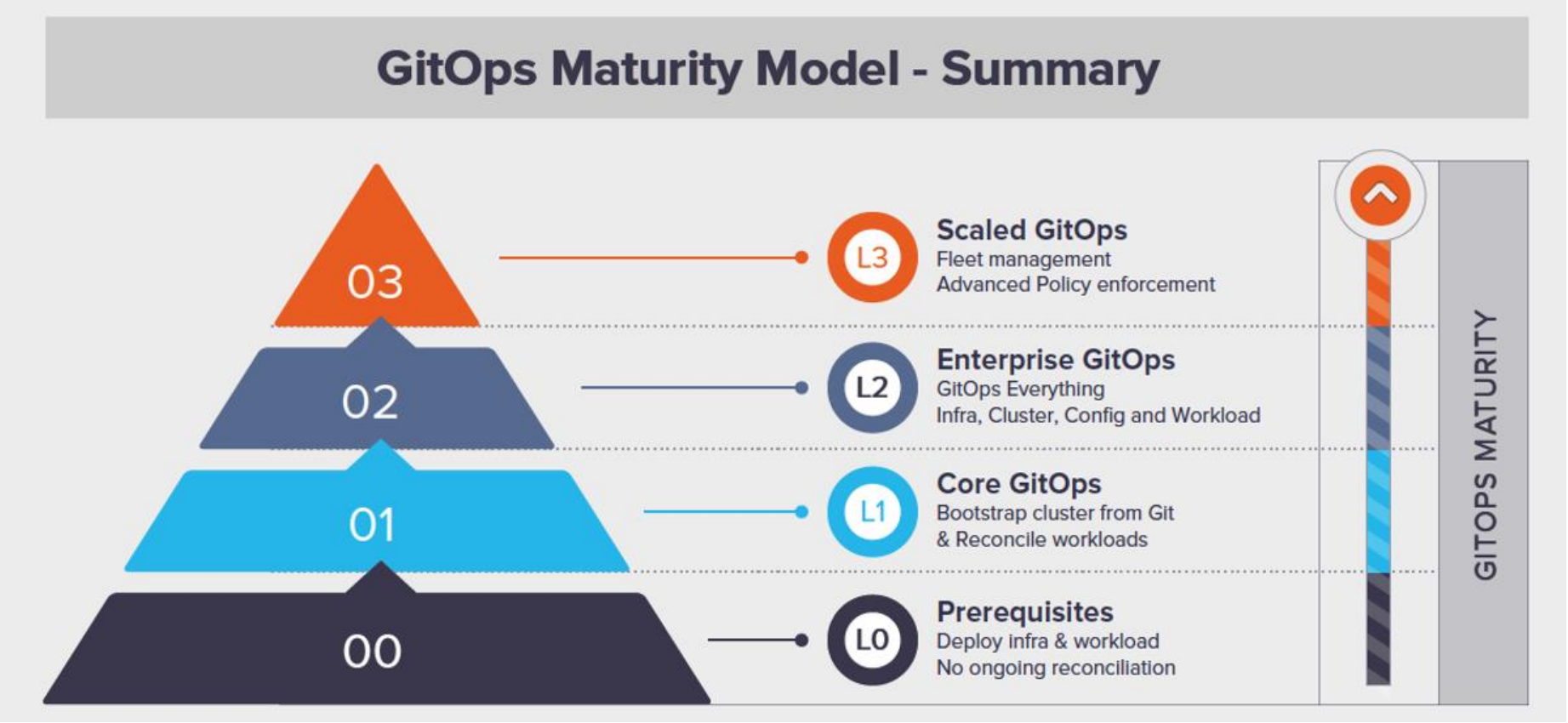


Evolution of GitOps Processes

Use CI Server to automate updates to dedicated GitOps repo



GitOps Maturity



[https://www.weave.works/blog/the-gitops-maturity-model]

Experiences

- How to structure your GitOps repo?

  According to your process needs
- Use separate code repo & config/GitOps repo
- Staging: Branches vs. Folders?

  Risk of environmental drift! Use branches you can actually merge!
- Local Development?
- Role of CI server?

02

GitOps Tools

(for kubernetes)

AEB

Tools: Overview

	Flux	Flux v2	ArgoCD	Fleet	PipeCD	JenkinsX	werf
Operation in Cluster	✓	✓	✓	✓	✓	✓	✗
Image Builds	✗	✗	✗	✗	✗	✓	✓
Pipelines	✗	✗	✗	✗	✗	✓	✗
Multi-Cluster	✗	✓	✓	✓	✓	✗	✗
Terraform Support	✗	✗	✗	✗	✓	✗	✗
Version (06/22)	1.25.2	0.31.2	2.4.2	0.3.9	0.33.0	3.2.385	1.2.117
1st commit	08/16	04/20	02/18	03/20	03/20	01/18	01/16
GitHub Stars	6.9k	3.5k	9.7k	1.1k	600	4.2k	3.2k
No. Contributors	275	106	690	57	42	34	43
Total Commits	5.200	2.200	4.300	750	3.000	11.700	10.300
Commits (05/22)	11	27	55	25	47	21	78
No. Issues	1.600	630	3.000	200	800	4.000	600
No. Pull Requests	1.900	1.000	4.300	300	2.700	4.200	3.700

[https://cloudogu.com/en/blog/gitops-tools updated 06/22 by S.Moritz]

Tools: Flux & ArgoCD

- CNCF Incubator Projects
- RBAC
- Observability (health status, metrics, notifications)
- CLI for automation and CI integration
- Webhooks
- Support for Helm and Kustomize
- Multi-cluster support

Tools: Flux vs. ArgoCD

ArgoCD features:

- Support for Ksonnet, Jsonnet, and others via the plug-in mechanism
- Web interface to administer and monitor applications in real time
- SSO integrations for UI and CLI

Flux v2 features:

- Support for SOPS
- Automatic updating of the image version in the Git repository
- Authentication of the CLI via Kubeconfig

Tools: Flux vs. ArgoCD

Flux

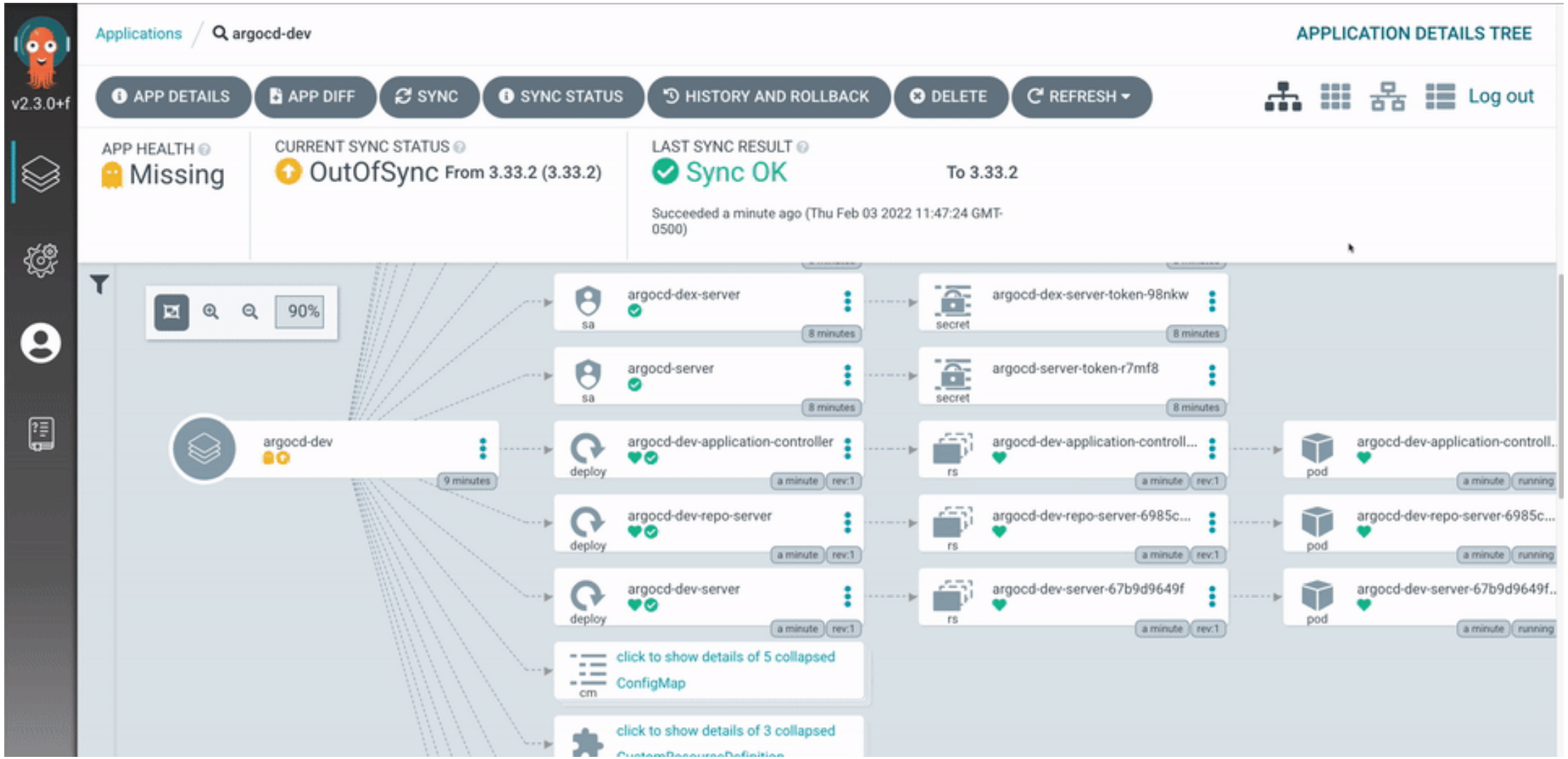
- Management Level: Git repo
- Coarsely RBAC
- CRDs: helmrelease and kustomization

ArgoCD

- Management Level: Projects and Applications (CRDs)
- Fine granular RBAC
- Helm templating to k8s manifests
- UI: Monitoring, transperency, simple to start


Excursion: ArgoCD
Get it out on the streets
AEB

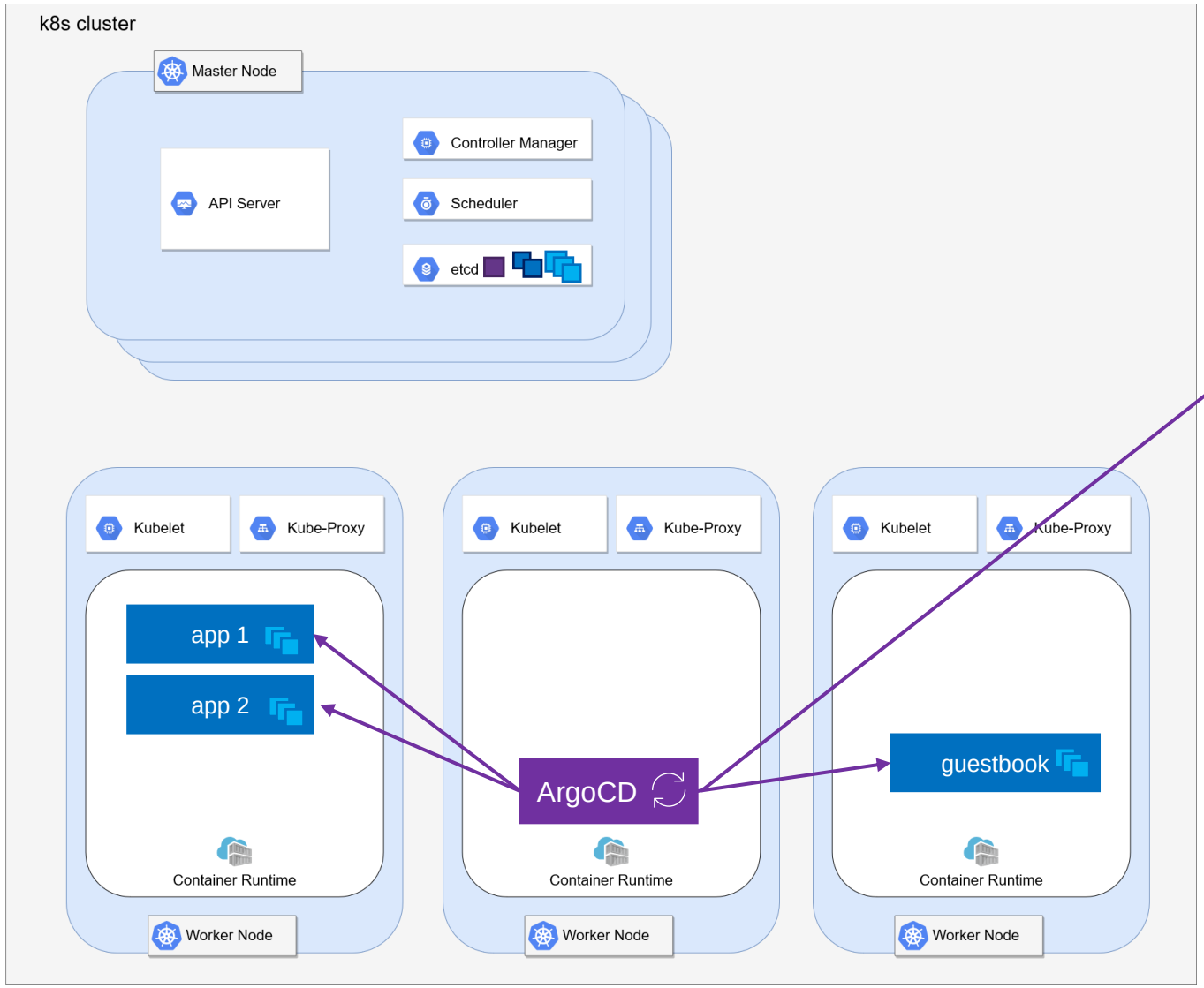
ArgoCD


v2.3.0+f
Applications
argocd-dev
APPLICATION DETAILS TREE
APP DETAILS
APP DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
Log out
APP HEALTH
Missing
CURRENT SYNC STATUS
OutOfSync From 3.33.2 (3.33.2)
LAST SYNC RESULT
Sync OK
To 3.33.2
Succeeded a minute ago (Thu Feb 03 2022 11:47:24 GMT-0500)
90%
argocd-dev
9 minutes
argocd-dex-server
8 minutes
argocd-dex-server-token-98nkw
8 minutes
argocd-server
8 minutes
argocd-server-token-r7mf8
8 minutes
argocd-dev-application-controller
a minute
rev:1
argocd-dev-application-controll...
a minute
rev:1
argocd-dev-application-controll..
a minute
running
argocd-dev-repo-server
a minute
rev:1
argocd-dev-repo-server-6985c...
a minute
rev:1
argocd-dev-repo-server-6985c...
a minute
running
argocd-dev-server
a minute
rev:1
argocd-dev-server-67b9d9649f
a minute
rev:1
argocd-dev-server-67b9d9649f..
a minute
running
click to show details of 5 collapsed ConfigMap
click to show details of 3 collapsed

The application.yaml

```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: guestbook
  namespace: argocd
spec:
  project: default
  source:
    repoURL: https://github.com/argoproj/argocd-example-apps.git
    targetRevision: HEAD
    path: guestbook
    helm:
      valueFiles:
      - values.yaml
  destination:
    server: https://kubernetes.default.svc
    namespace: guestbook
```

Application Deployment



developer

user

GitOps Repository

app 1
app 2
guestbook

```
apiVersion: argoproj.io/v1
kind: Application
metadata:
  name: guestbook
spec:
  project: default
  source:
    repoURL: https://gb.git
    targetRevision: 2.3.12
    path: guestbook
```

03

Summary

What you should take away

AEB

Summary

- Remember the GitOps Principles

  Versioned, declarative description of desired state

  Pull mechanism with continuous synchronization
- Move from CI-Ops to GitOps
- Checkout Flux & ArgoCD

  Pure, open source, k8s GitOps Controllers

„The right way to do DevOps“

~~Alexis Richardson~~ Sebastian Moritz

Thank you!
Questions?

Sebastian Moritz

AEB SE

DevOps Engineer

sebastian.moritz@aeb.com

AEB