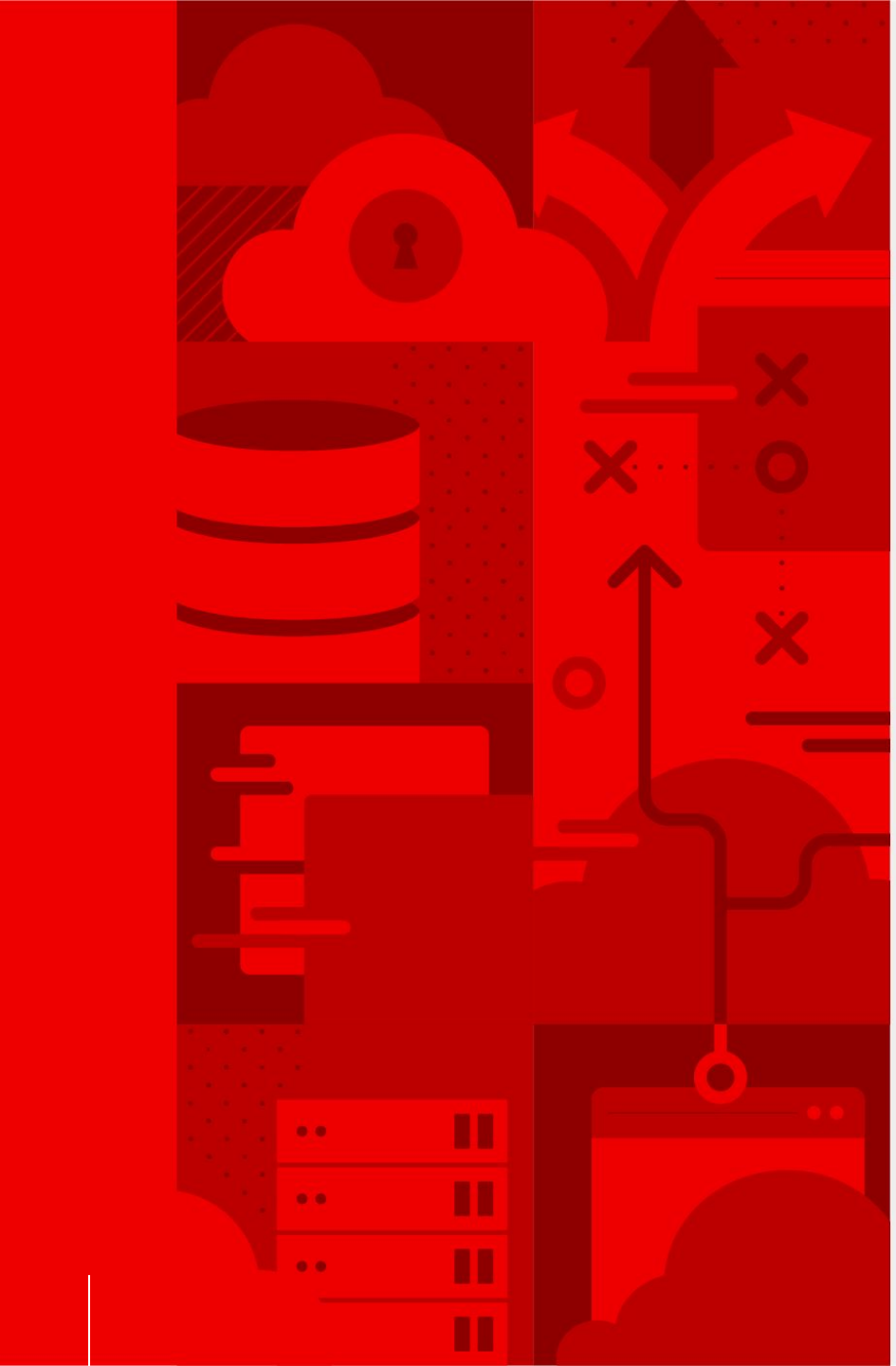
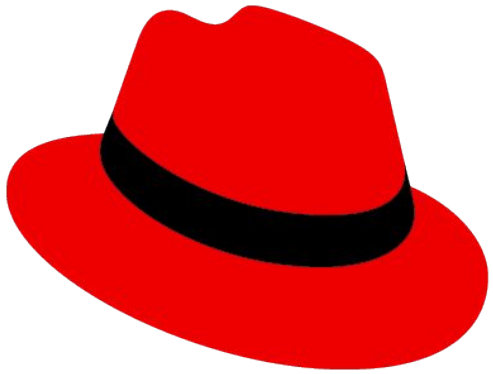


Kubernetes Patterns

Reusable Elements for Designing Cloud-Native Applications

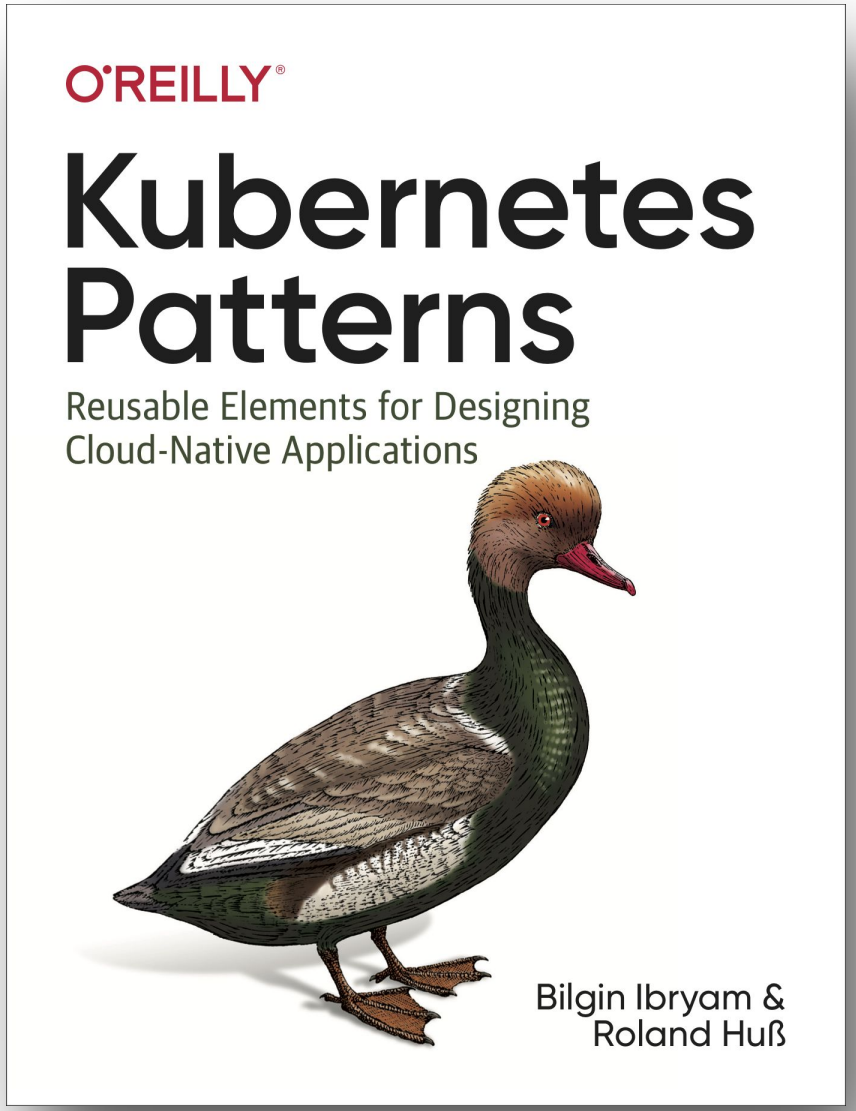
Dr. Roland Huß
Principal Software Engineer
@ro14nd
https://k8spatterns.io

Red Hat



Red Hat

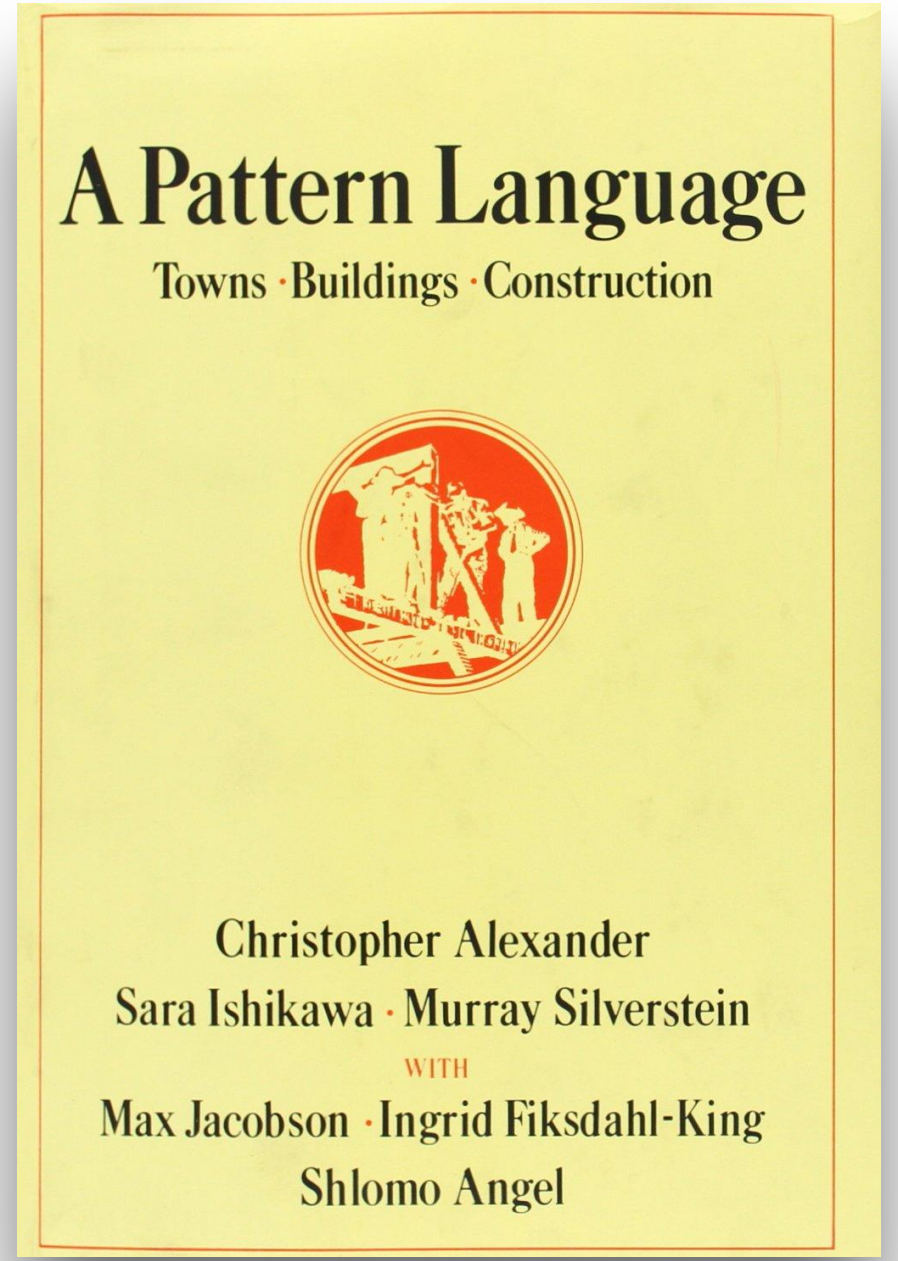
New Logo !



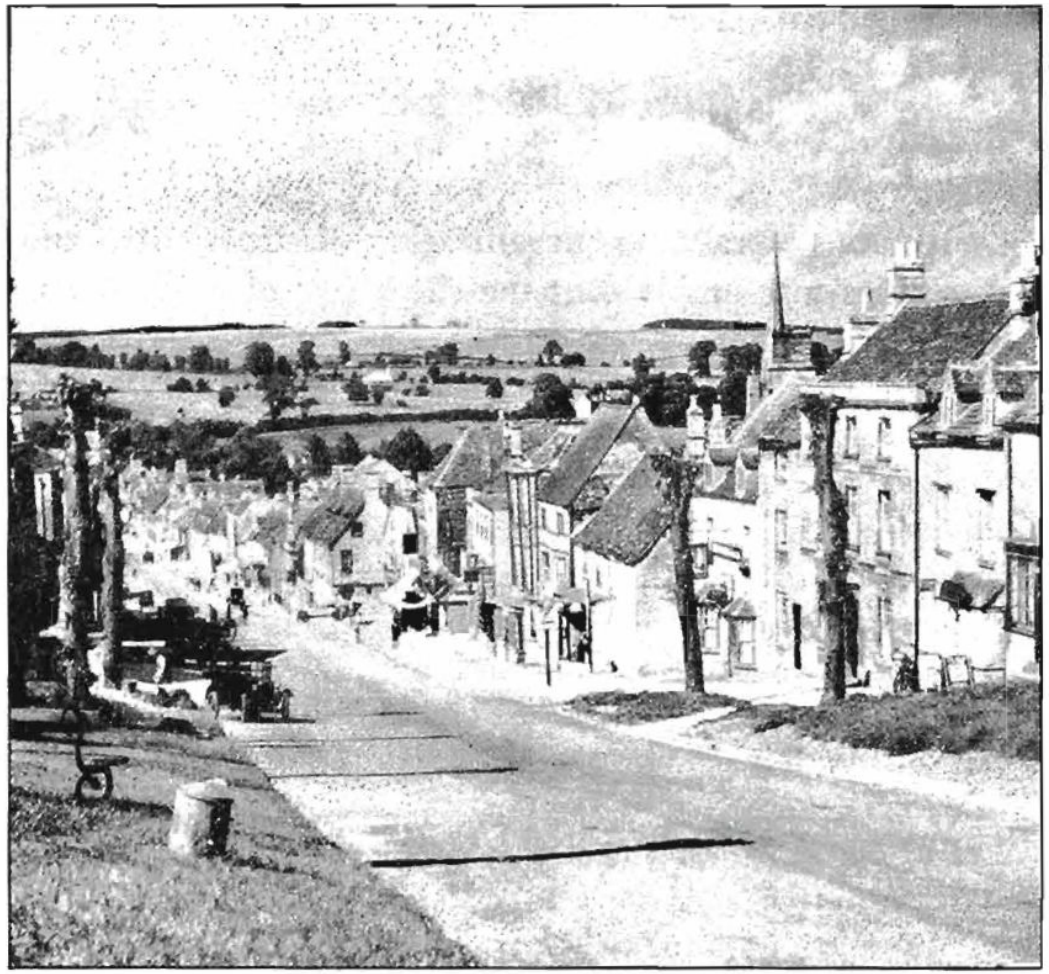
https://k8spatterns.io

Red Hat

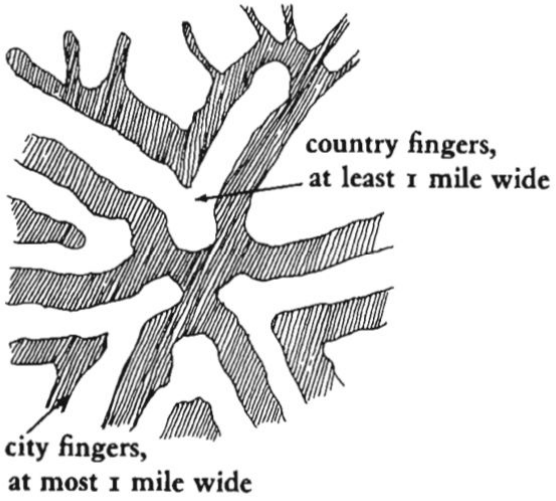

Patterns


A Pattern Language
Towns · Buildings · Construction
Christopher Alexander
Sara Ishikawa · Murray Silverstein
WITH
Max Jacobson · Ingrid Fiksdahl-King
Shlomo Angel

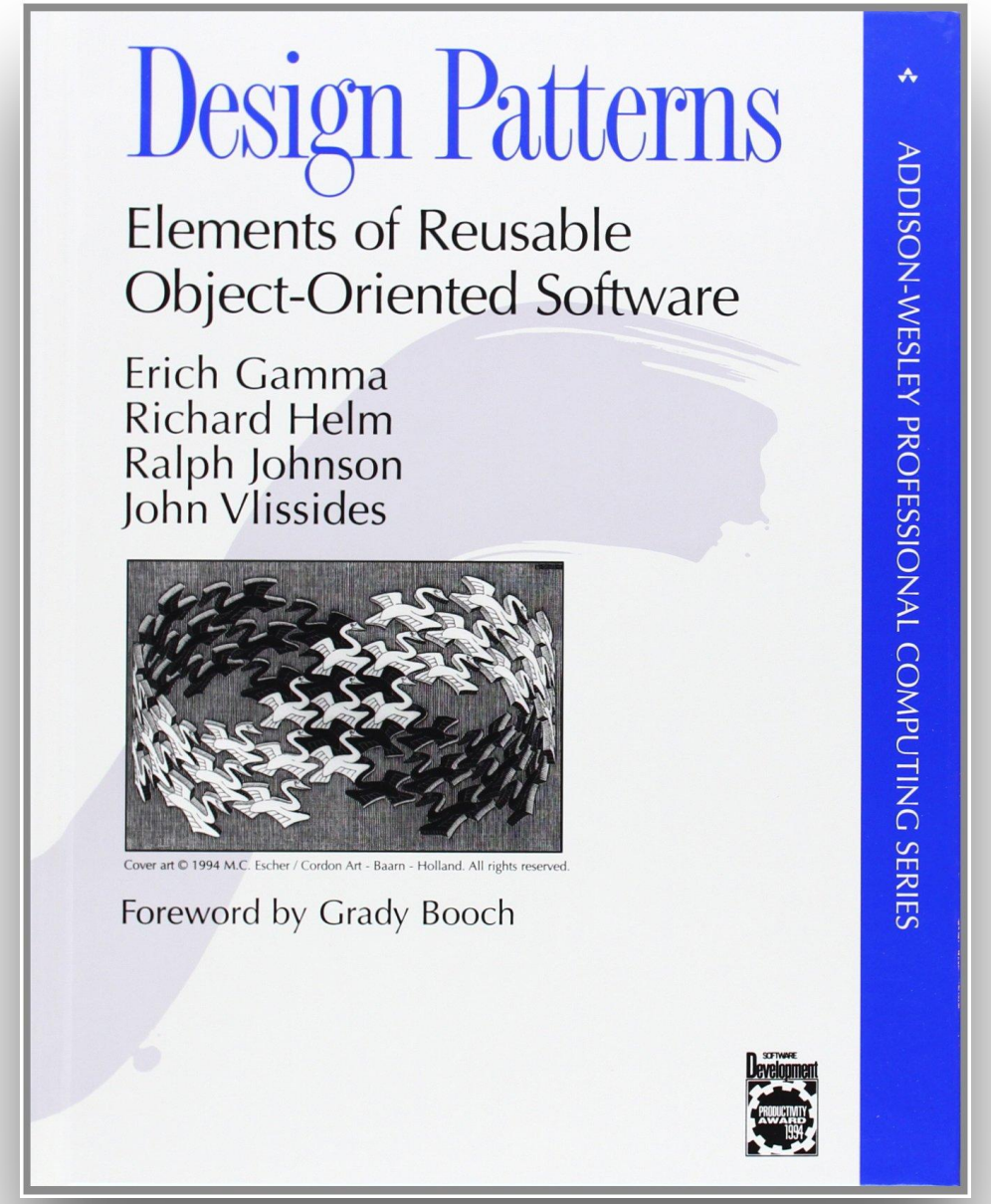
3 CITY COUNTRY FINGERS**



When the countryside is far away the city becomes a prison.



A Pattern Language, Christopher Alexander et. al, 1977, pp. 21


Design Patterns
Elements of Reusable
Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Red Hat

Patterns

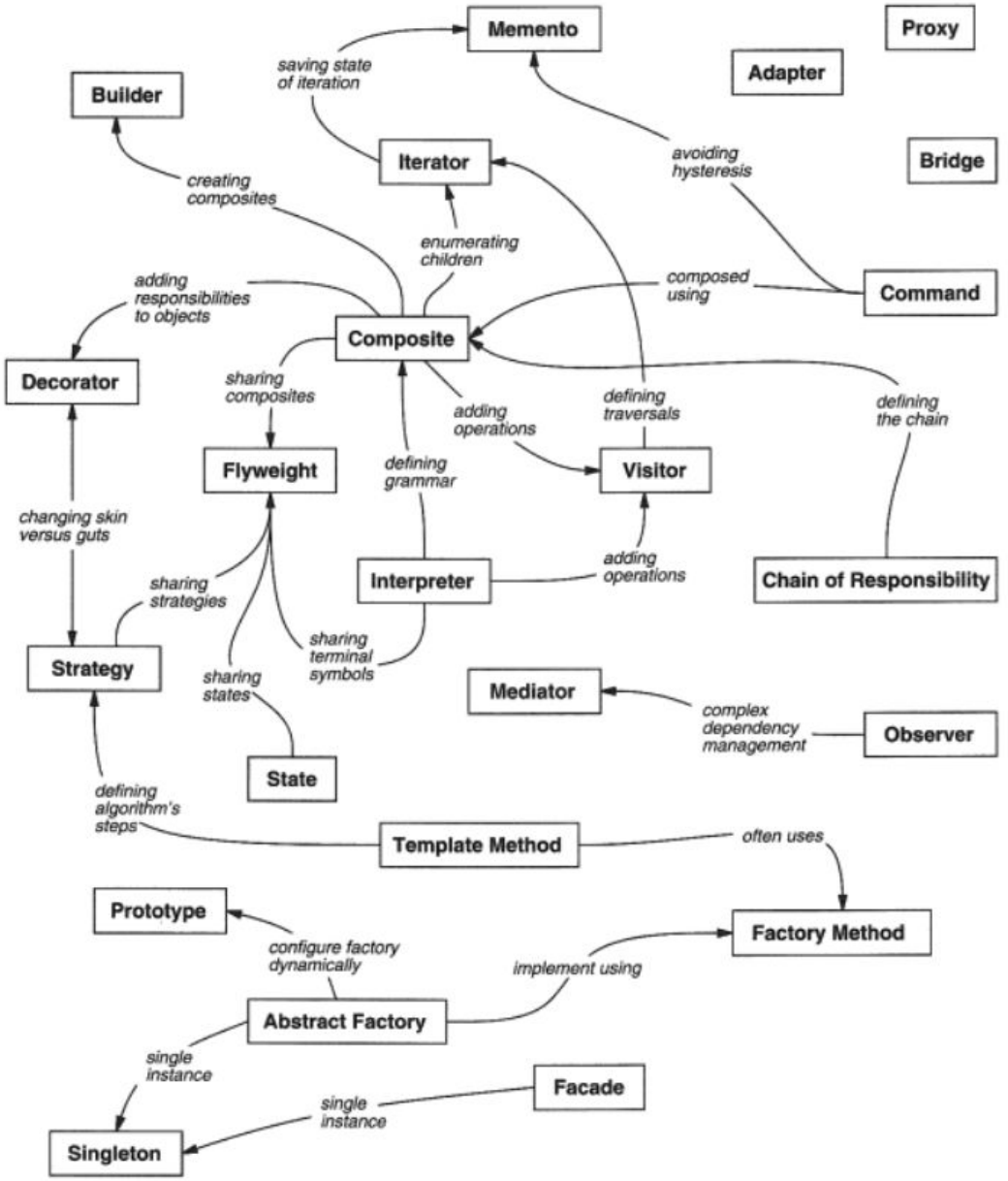
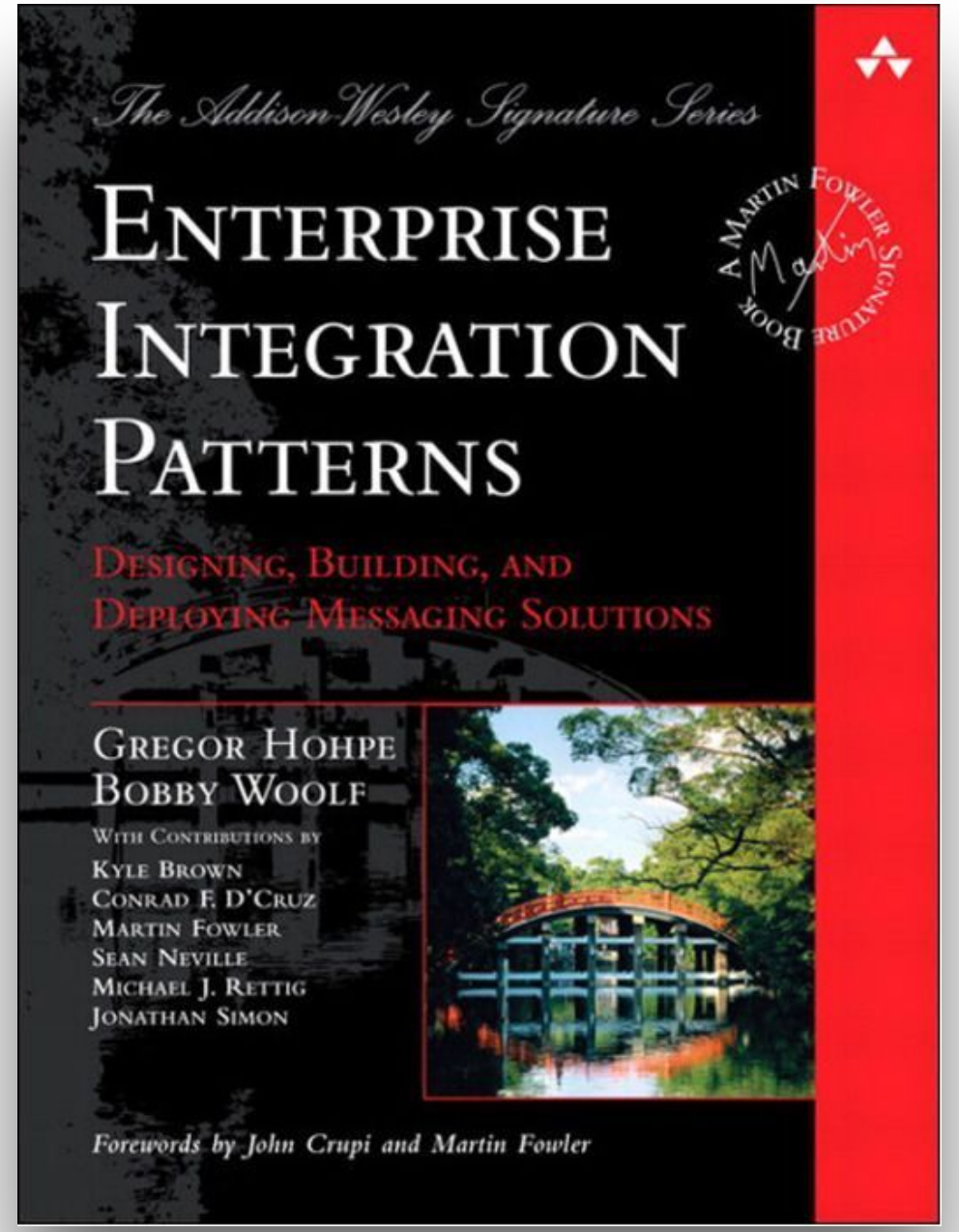


Figure 1.1: Design pattern relationships

Design Patterns, Elements of Reusable Object-Oriented Software, E. Gamma et. al., 1994, p. 12

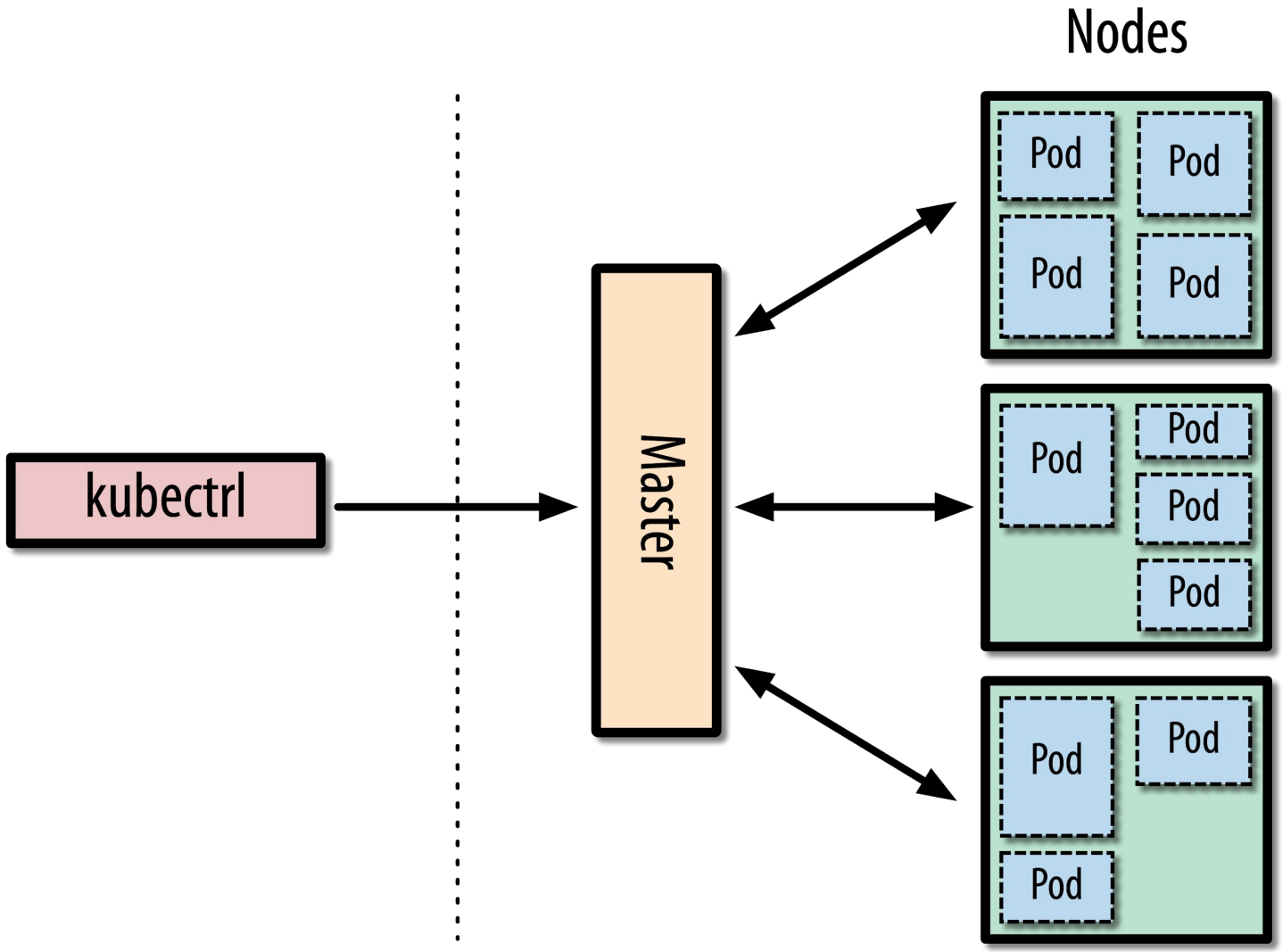

The Addison Wesley Signature Series
A Martin Fowler Signature Book
Enterprise Integration Patterns
Designing, Building, and Deploying Messaging Solutions
Gregor Hohpe
Bobby Woolf
With Contributions by
Kyle Brown
Conrad F. D'Cruz
Martin Fowler
Sean Neville
Michael J. Rettig
Jonathan Simon
Forewords by John Crupi and Martin Fowler


Kubernetes

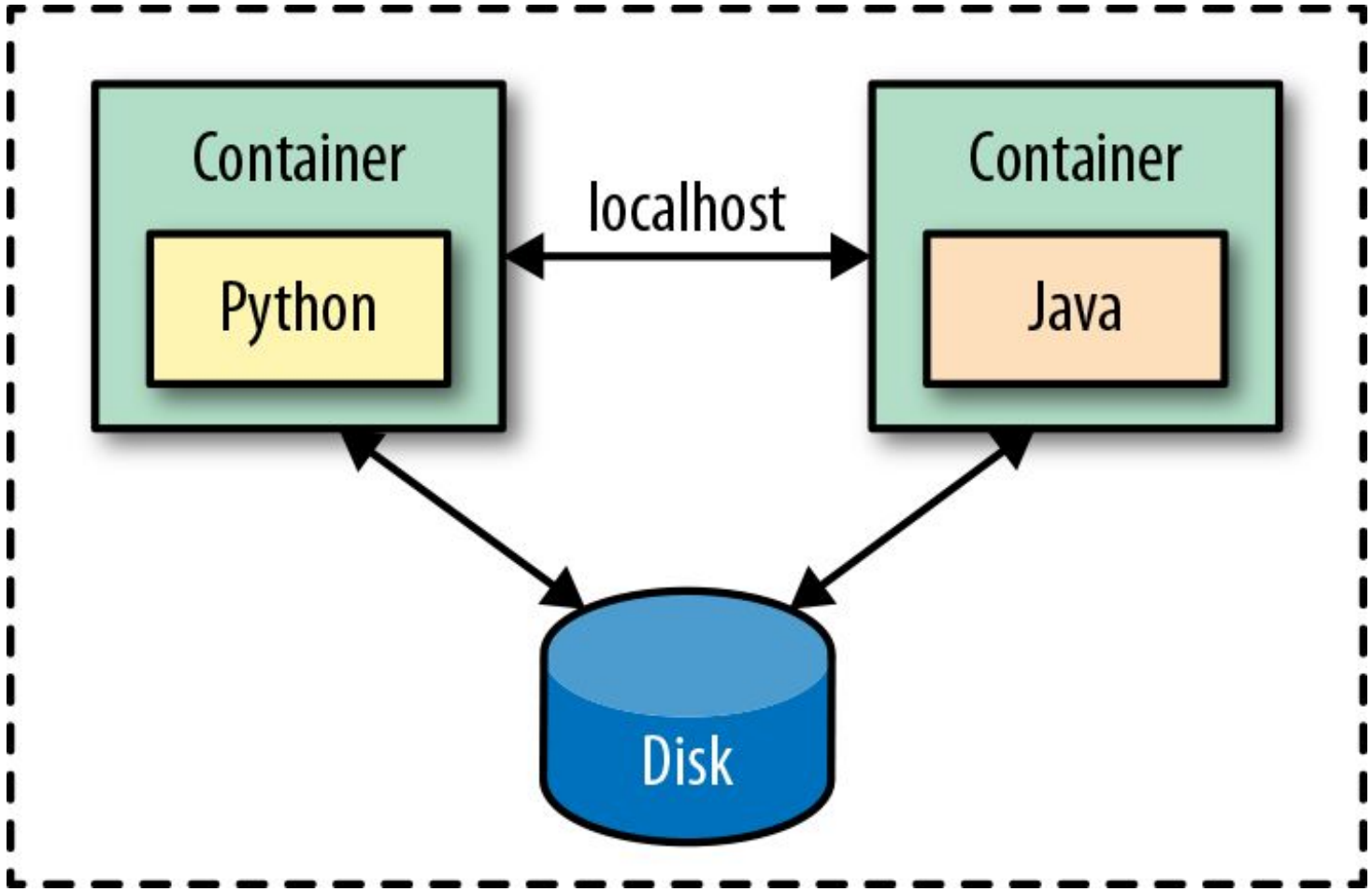
Kubernetes

- Open Source container orchestration system started by Google in 2014
  - Scheduling
  - Self-healing
  - Horizontal and vertical scaling
  - Service discovery
  - Rollout and Rollbacks
- Declarative resource-centric REST API

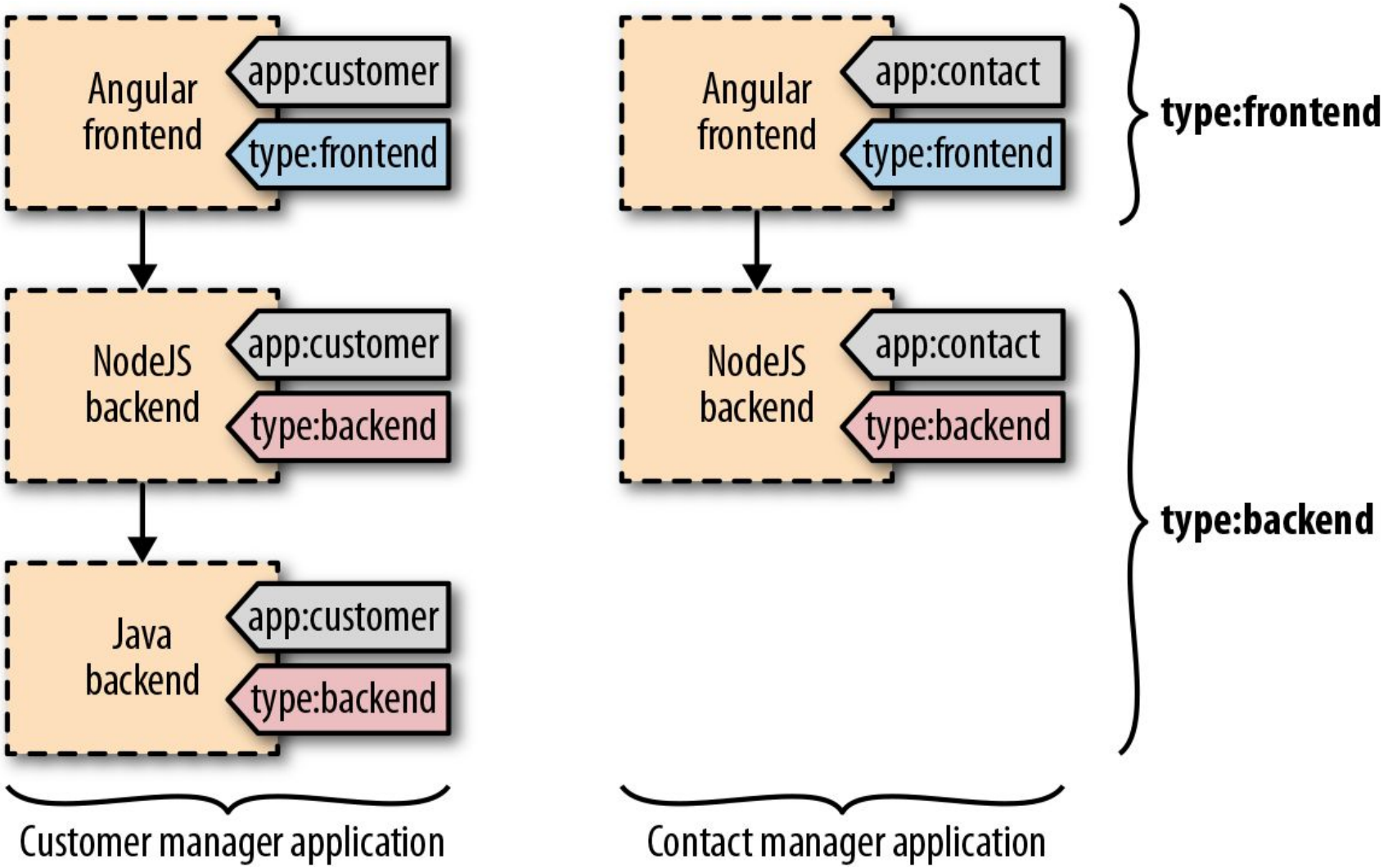
Architecture

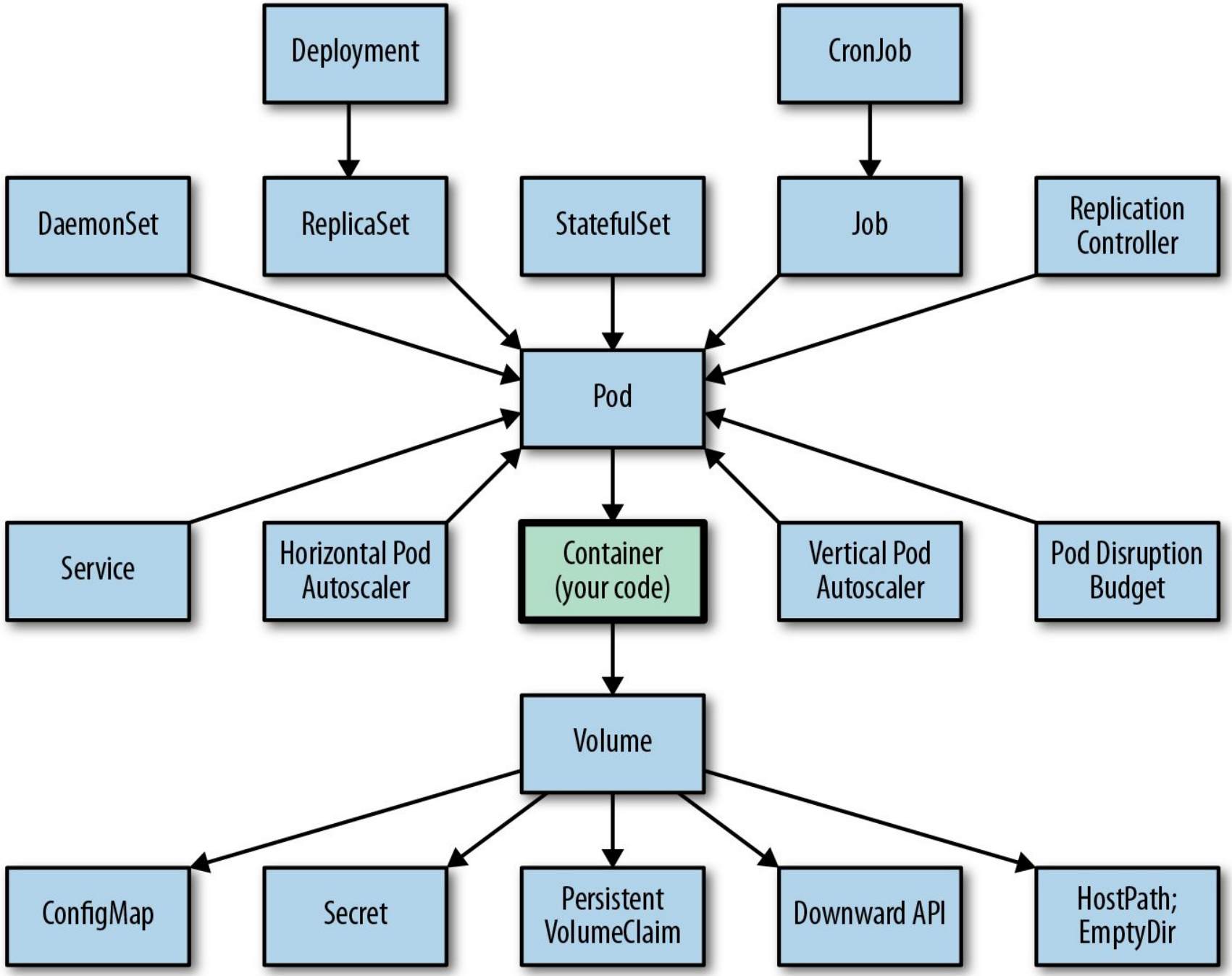


Pod



Labels




Deployment
CronJob
DaemonSet
ReplicaSet
StatefulSet
Job
Replication Controller
Pod
Service
Horizontal Pod Autoscaler
Container (your code)
Vertical Pod Autoscaler
Pod Disruption Budget
Volume
ConfigMap
Secret
Persistent VolumeClaim
Downward API
HostPath; EmptyDir


Predictable
Demands

How to declare application requirements.

Application Requirements

- Declared requirements help in
  - Scheduling decisions
  - Capacitivy planning
  - Matching infrastructure services
- Runtime dependencies
  - Persistent Volumes
  - Host ports
  - Dependencies on ConfigMaps and Secrets

Resource Profile

```
apiVersion: v1
kind: Pod
metadata:
  name: http-server
spec:
  containers:
  - image: nginx
    name: nginx
    resources:
      requests:
        cpu: 200m
        memory: 100Mi
      limits:
        cpu: 300m
        memory: 200Mi
```

Quality-of-Service Classes

- **Best Effort**
  - No requests or limits
- **Burstable**
  - requests < limits
- **Guaranteed**
  - requests == limits


Health
Probe

How to communicate an application's health state to Kubernetes.

Monitoring Health

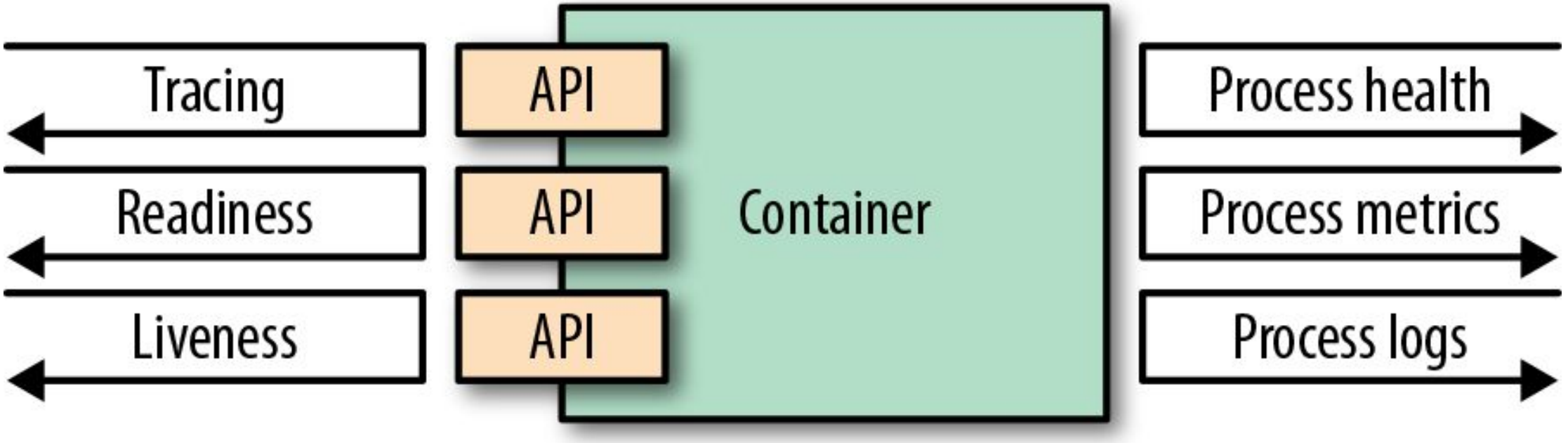
- Process health checks
  - Checks running process
  - Restarts container if container process died
- Application provided Health Probes
  - **Liveness Probe**: Check application health
  - **Readiness Probe**: Check readiness to process requests

Liveness & Readiness

- Liveness Probe
  - Restarting containers if liveness probes fail
- Readiness Probe
  - Removing from service endpoint if readiness probe fails
- Probe methods
  - HTTP endpoint
  - TCP socket endpoint
  - Unix command’s return value

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-with-readiness-check
spec:
  containers:
  - image: k8spatterns/random-generator:1.0
    name: random-generator
    livenessProbe:
      httpGet:
        path: /actuator/health
        port: 8080
      initialDelaySeconds: 30
    readinessProbe:
      exec:
        command: [ "stat", "/var/run/random-generator-ready" ]
```

Container Observability Options



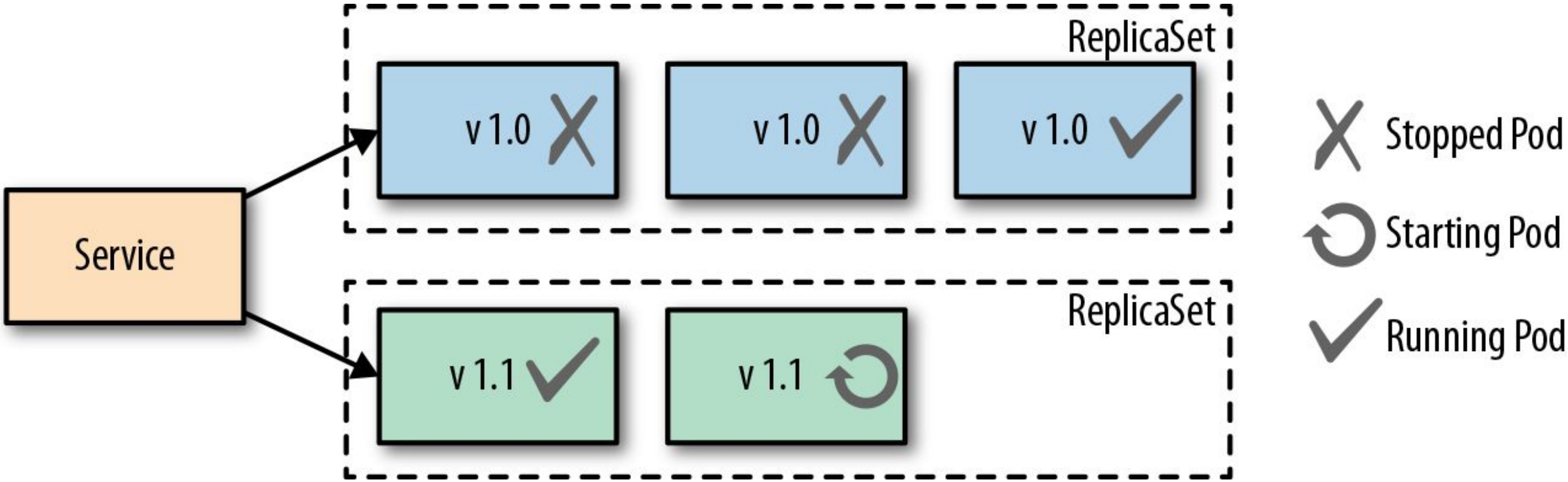
Declarative Deployment

How to perform application installations and upgrades by configuration.

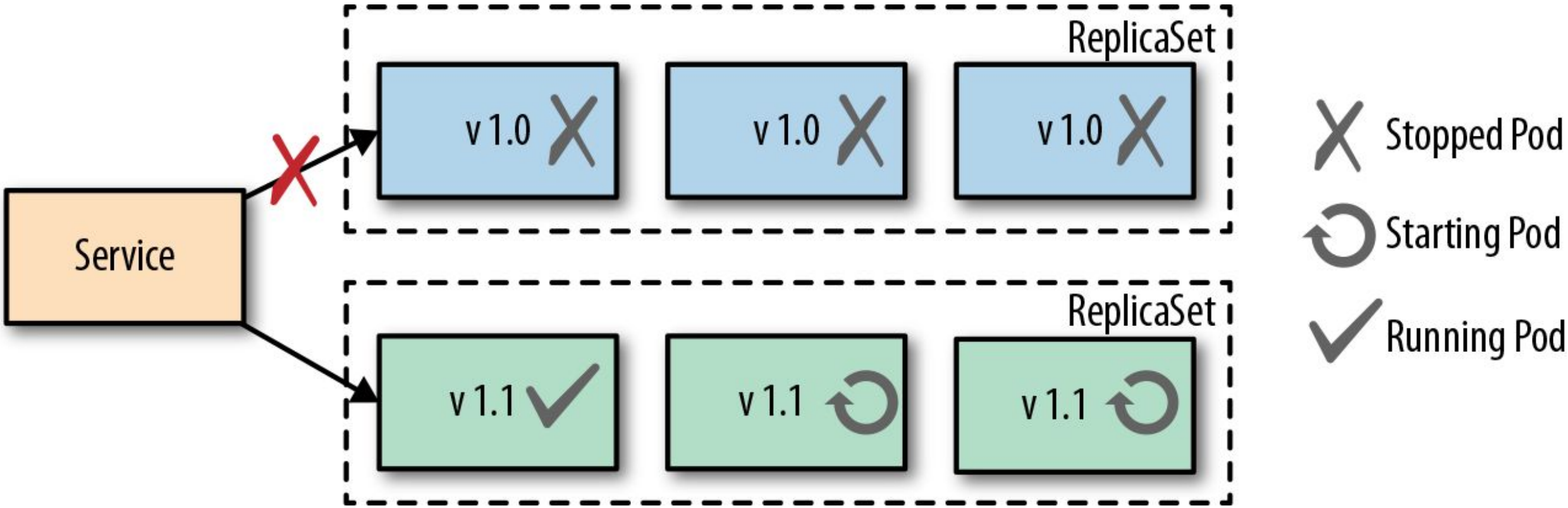
Deployment

- **Declarative** vs. **Imperative** deployment
- Deployment Kubernetes Resource:
  - Holds template for Pod
  - Creates ReplicaSet on the fly
  - Allows rollback
  - Update strategies are declarable

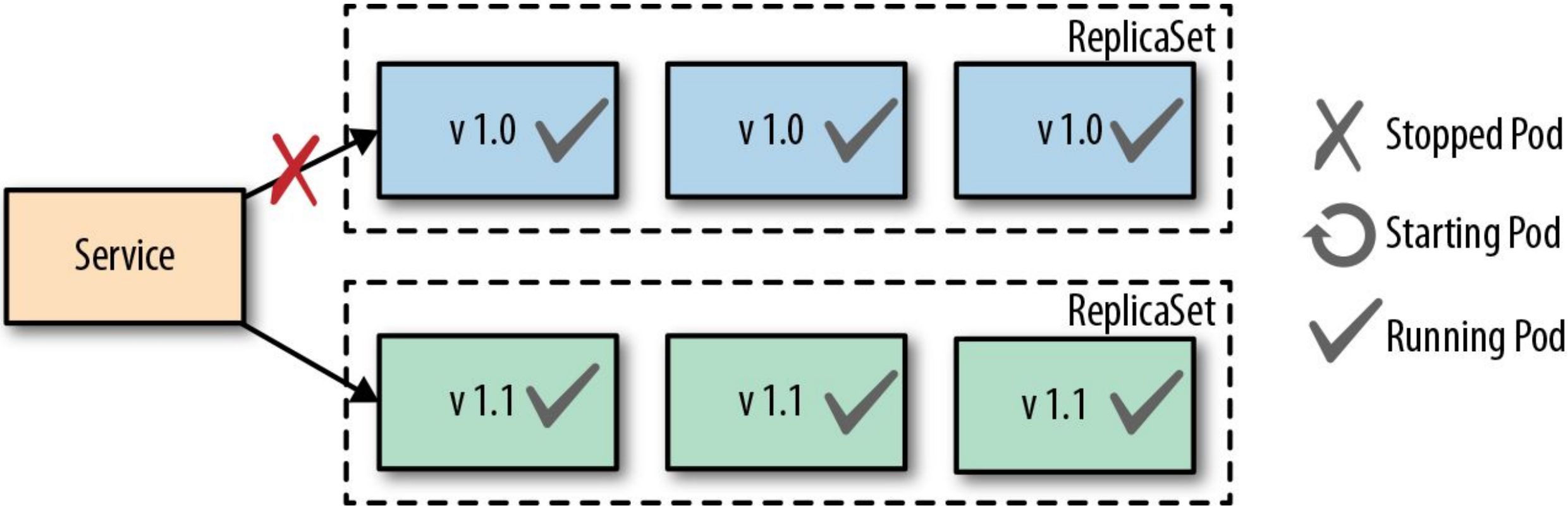
Rolling Deployment



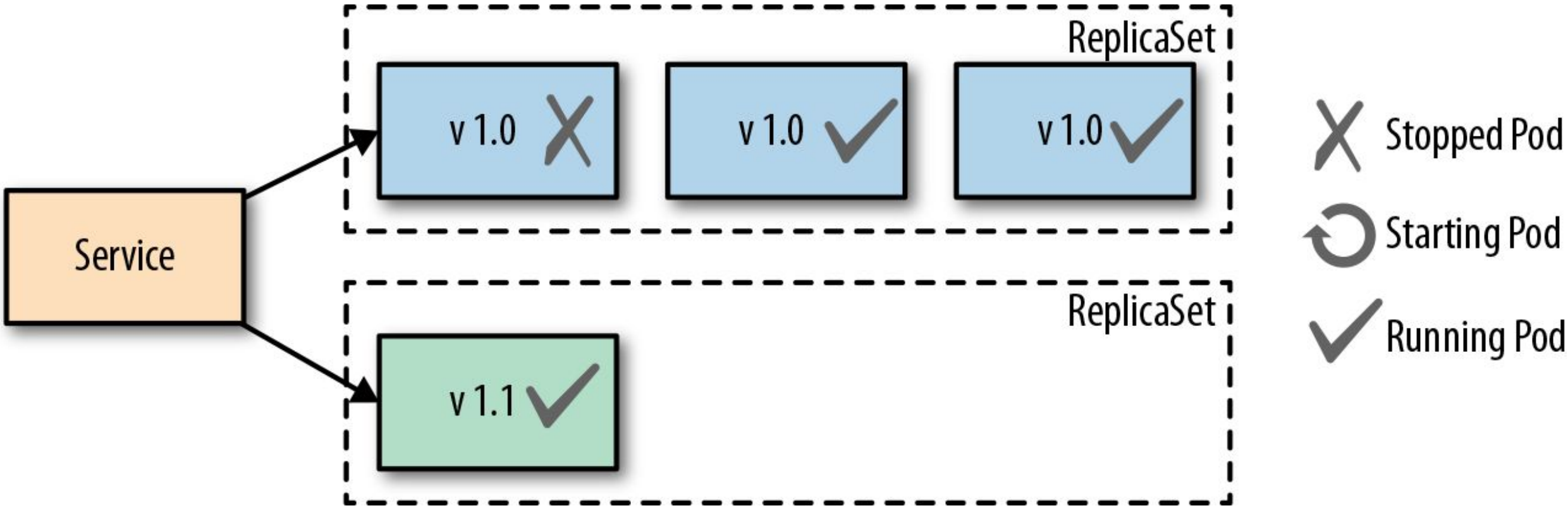
Fixed Deployment

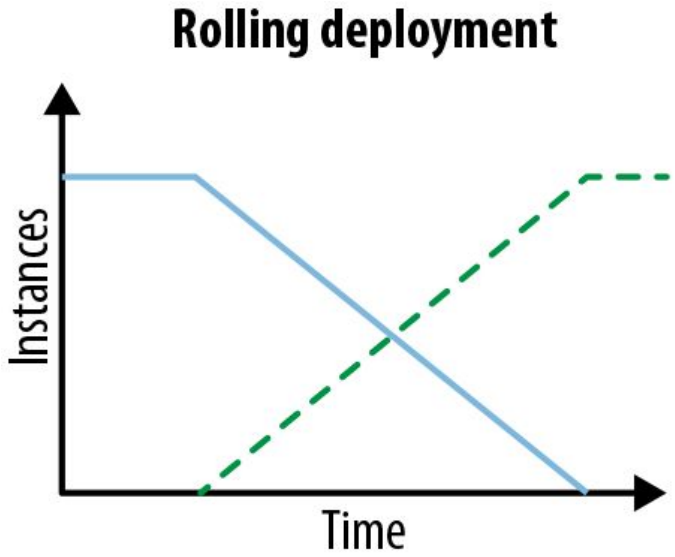


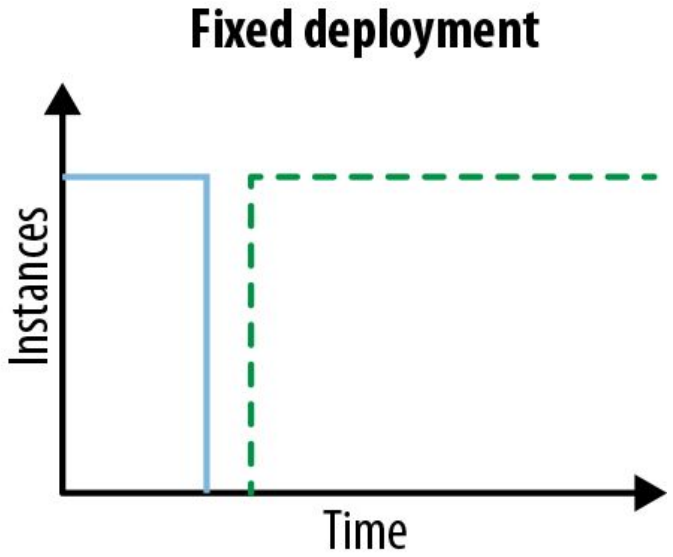
Blue-Green Release

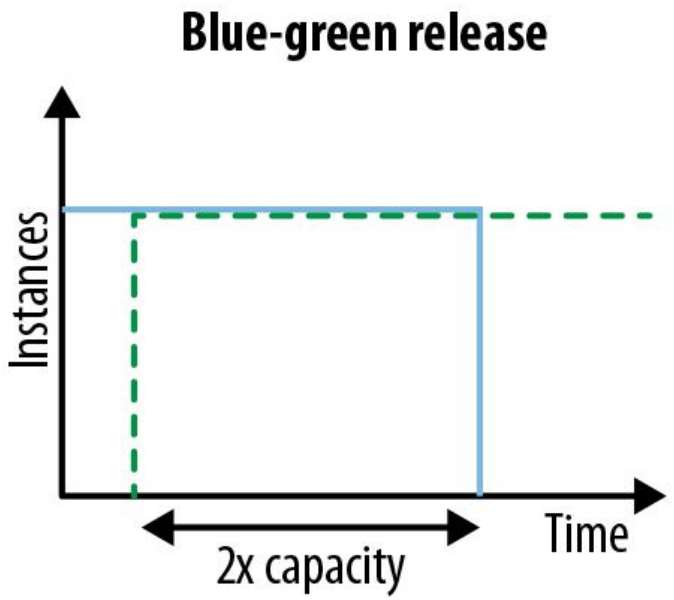


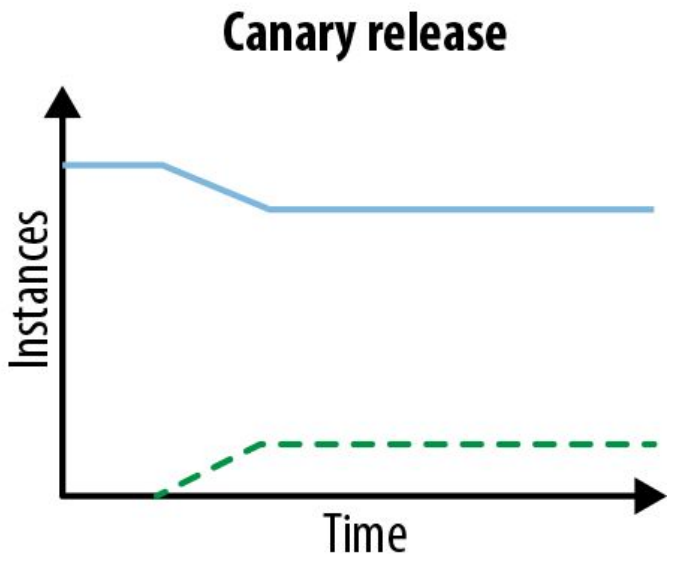
Canary Release

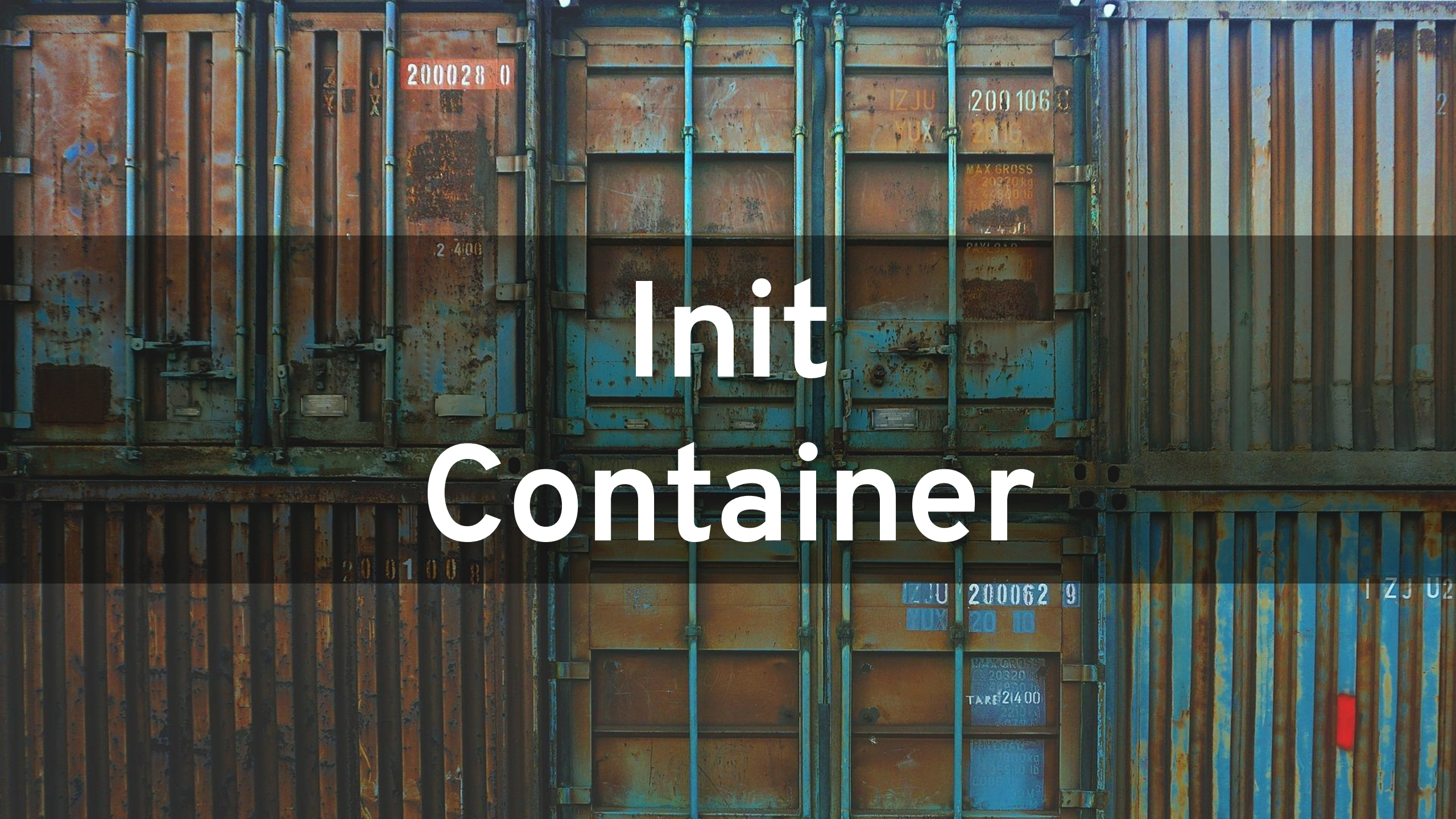



Rolling deployment
Instances
Time


Fixed deployment
Instances
Time


Blue-green release
Instances
Time
2x capacity


Canary release
Instances
Time



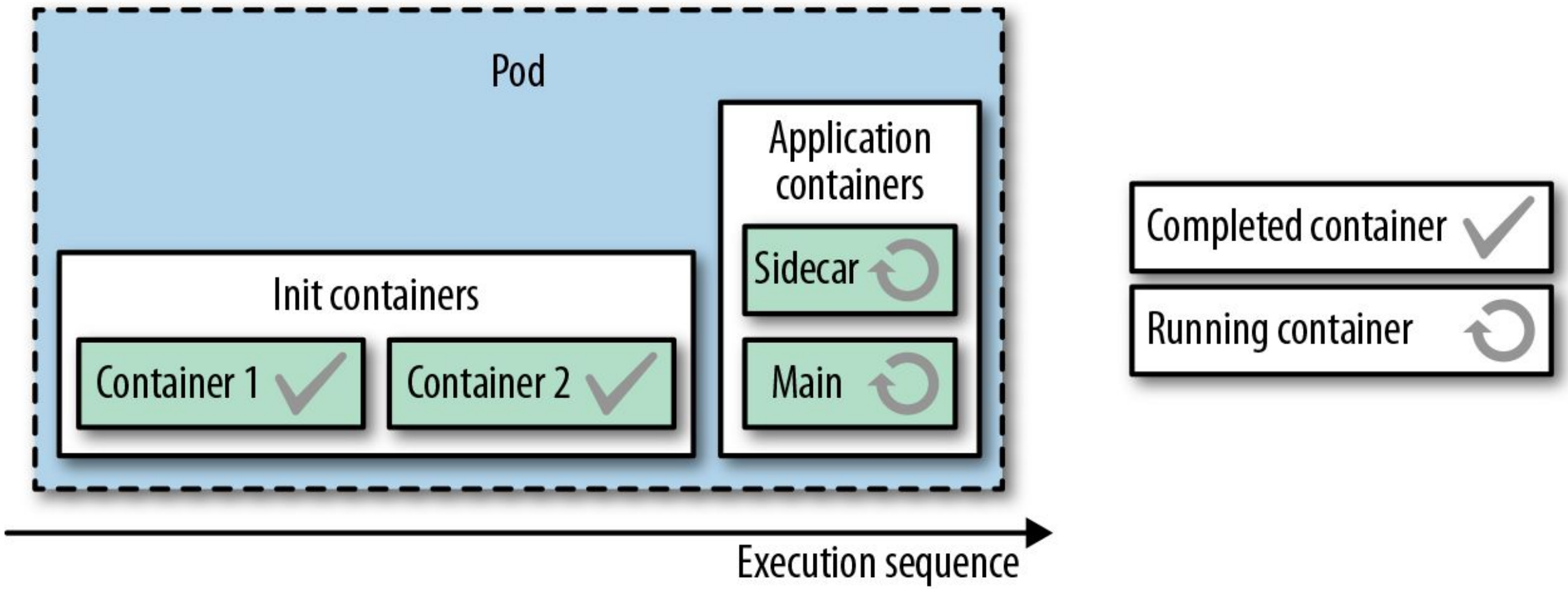
Init Container

How to separate initialization steps from the main application.

Init Container

- Part of a Pod
- One shot actions before application starts
- Needs to be idempotent
- Has own resource requirements

Init Container



Red Hat

Init Container

```
apiVersion: v1
kind: Pod
....
spec:
  initContainers:
  - name: download
    image: axeclbr/git
    command: [ "git","clone","https://github.com/myrepo","/data"]
    volumeMounts:
    - mountPath: /var/lib/data
      name: source
  containers:
  - name: run
    image: docker.io/centos/httpd
    volumeMounts:
    - mountPath: /var/www/html
      name: source
  volumes:
  - emptyDir: {}
    name: source
```

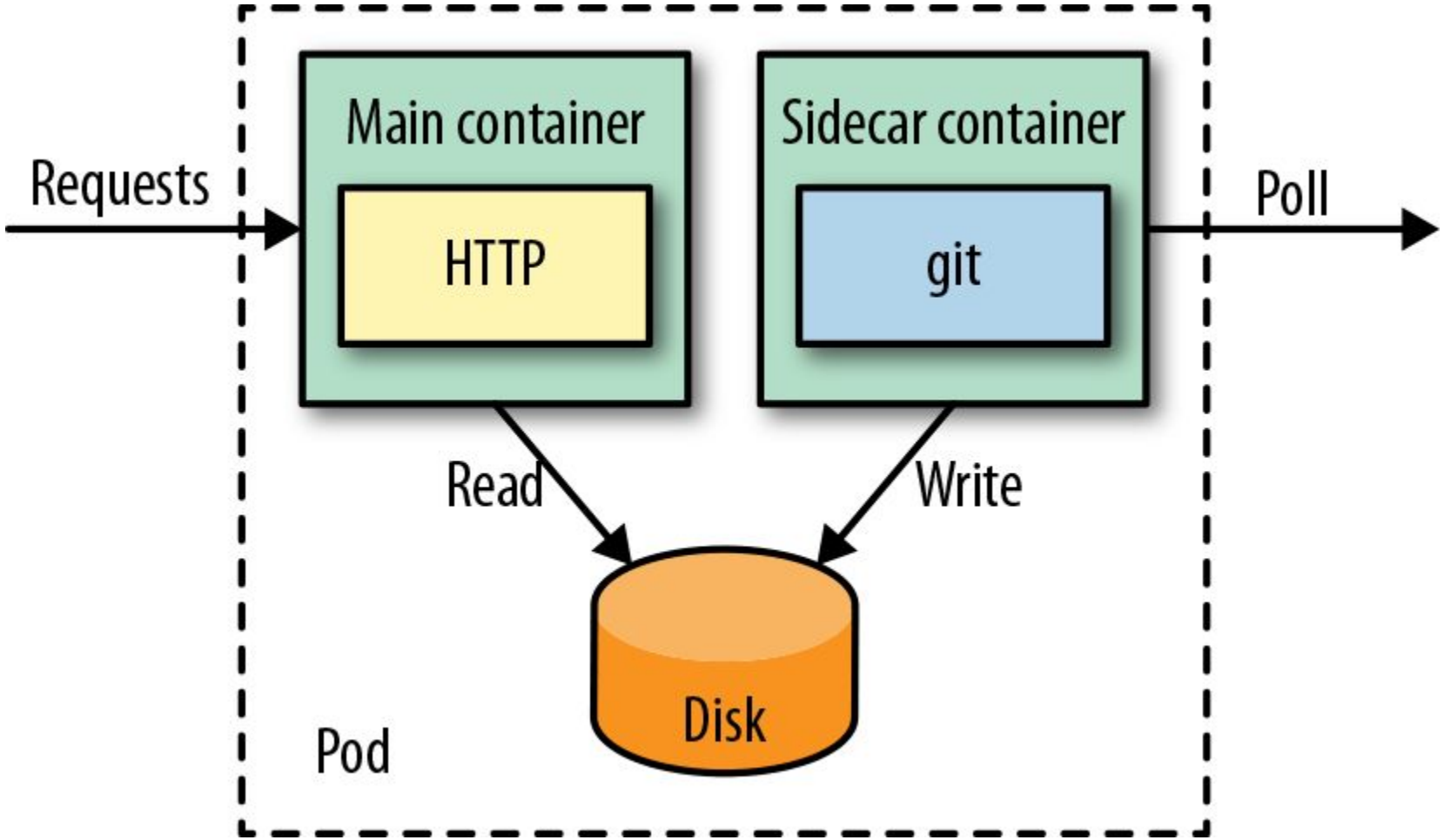
Sidecar

How to enhance the functionality of an application without changing it.

Sidecar

- Runtime collaboration of containers
- Connected via shared resources:
  - Network
  - Volumes
- Similar what AOP is for programming
- Separation of concerns

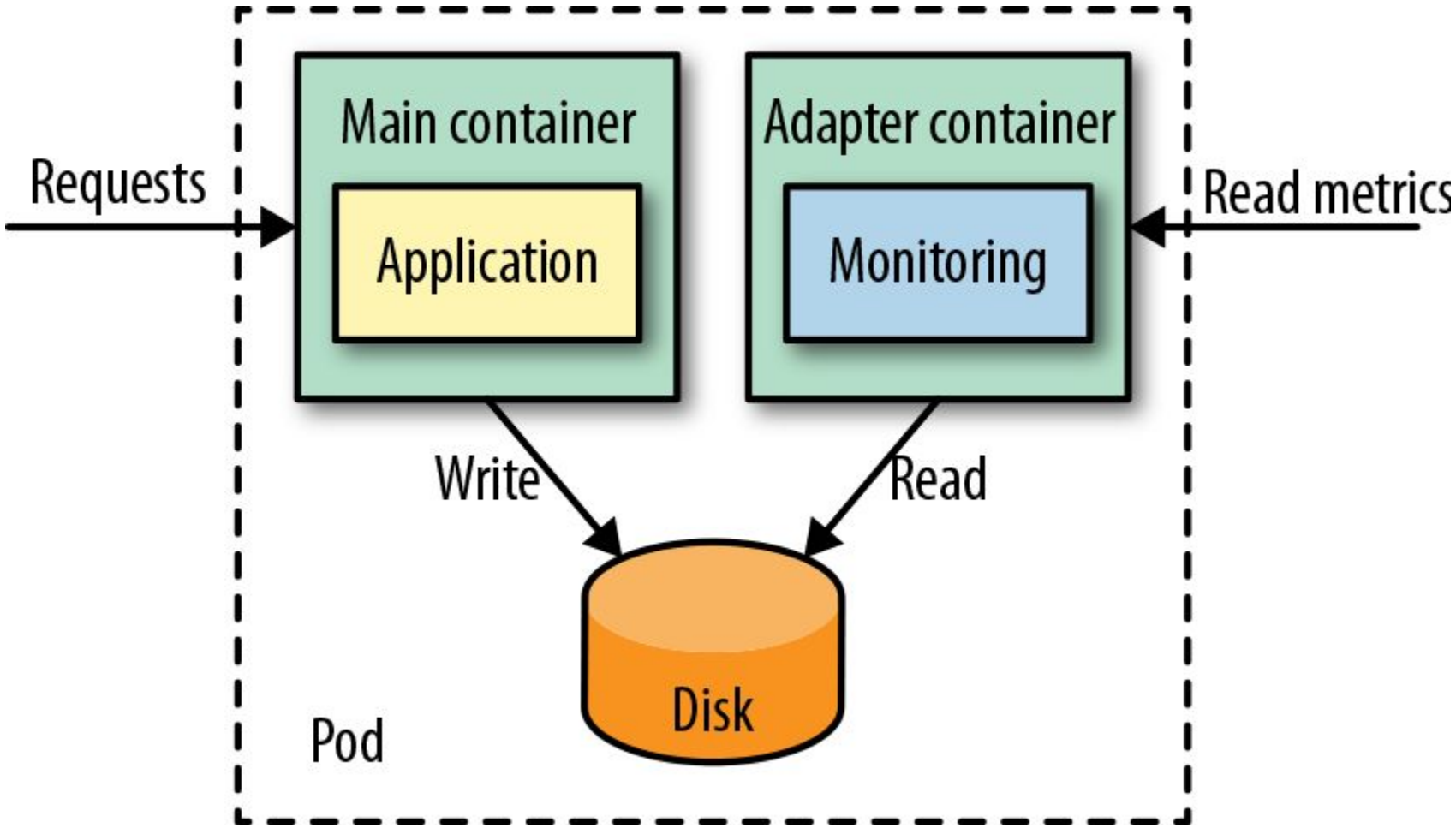
Sidecar



Adapter

How to decouple access to a container **from** the outside world.

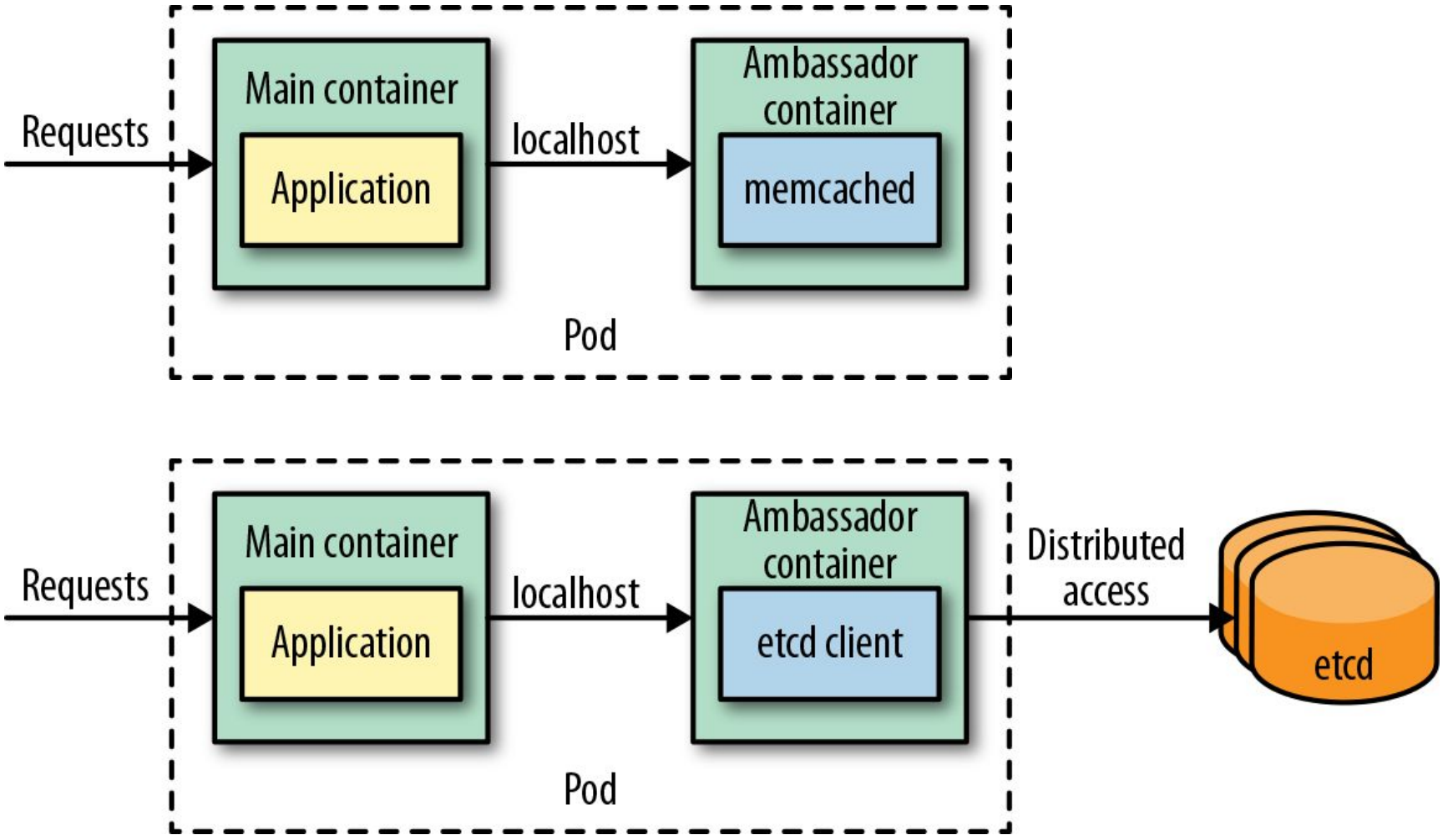
Adapter




Ambassador

How to decouple a container's access **to** the outside world

Ambassador



Controller

How to get from the current state to a declared target state.

State Reconciliation

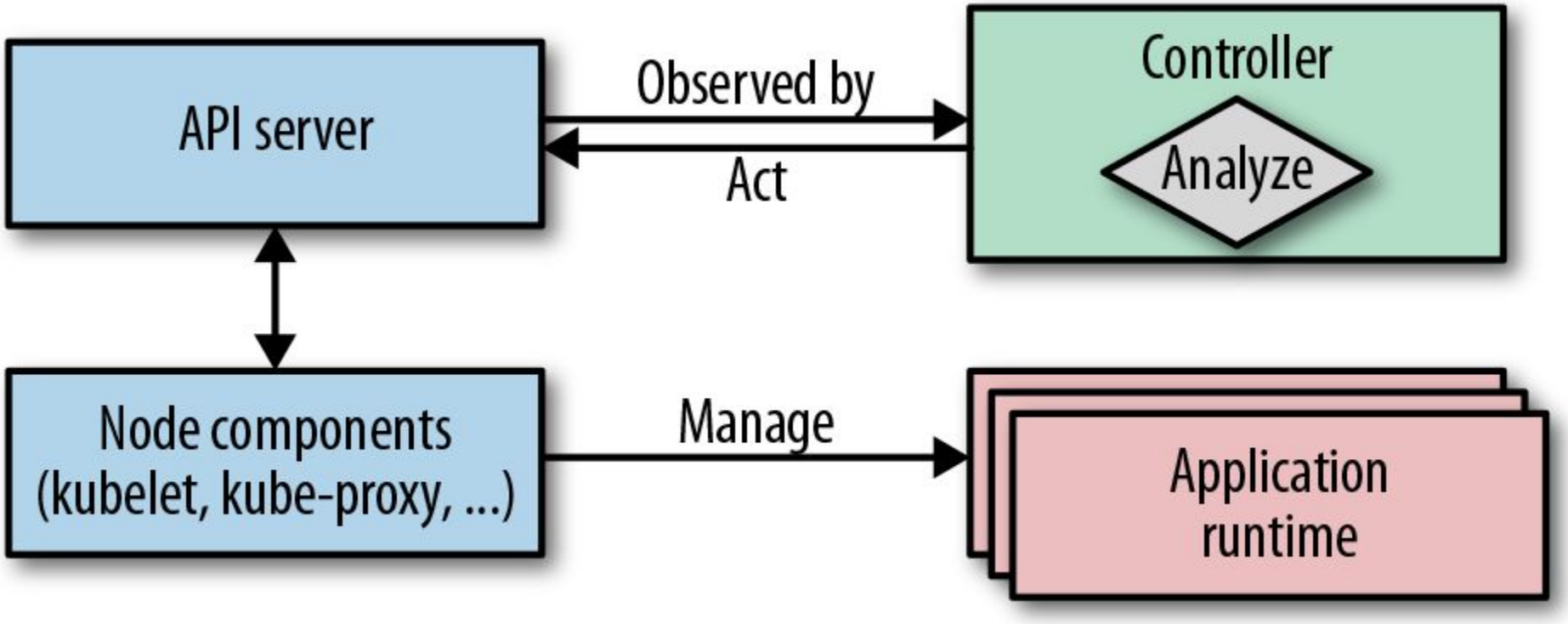
- Kubernetes as distributed state manager
- Make the **actual** state more like the declared **target** state.

🔎 ***Observe*** - Discover the actual state

🤔 ***Analyze*** - Determine difference to target state

🔨 ***Act-*** Perform actions to drive the actual to the desired state

Observe - Analyze - Act



Common Triggers

- Labels
  - Indexed by backend
  - Suitable for selector-like functionality
  - Limitation on characterset for names and values
- Annotations
  - No syntax restrictions
  - Not indexed
- ConfigMaps
  - Good for complex structured state declarations
  - Simple alternative to CustomResourceDefinitions

ConfigMap Watch Controller

```
namespace=${WATCH_NAMESPACE:-default}
base=http://localhost:8001
ns=namespaces/$namespace

curl -N -s $base/api/v1/${ns}/configmaps?watch=true | \
while read -r event
do
  type=$(echo "$event" | jq -r '.type')
  config_map=$(echo "$event" | jq -r '.object.metadata.name')
  annotations=$(echo "$event" | jq -r '.object.metadata.annotations')

  if [ $type = "MODIFIED" ]; then
    # Restart Pods using this ConfigMap
    # ...
  fi
done
```


Operator

How to encapsulate operational knowledge into executable software.

Definition

> An **operator** is a Kubernetes **controller** that understands two domains: Kubernetes and *something else*. By combining knowledge of both areas, it can **automate** tasks that usually require a human operator that understands both domains.
>
> **Jimmy Zelinskie**
> http://bit.ly/2Fjlx1h

Technical:

Operator = Controller + CustomResourceDefinition

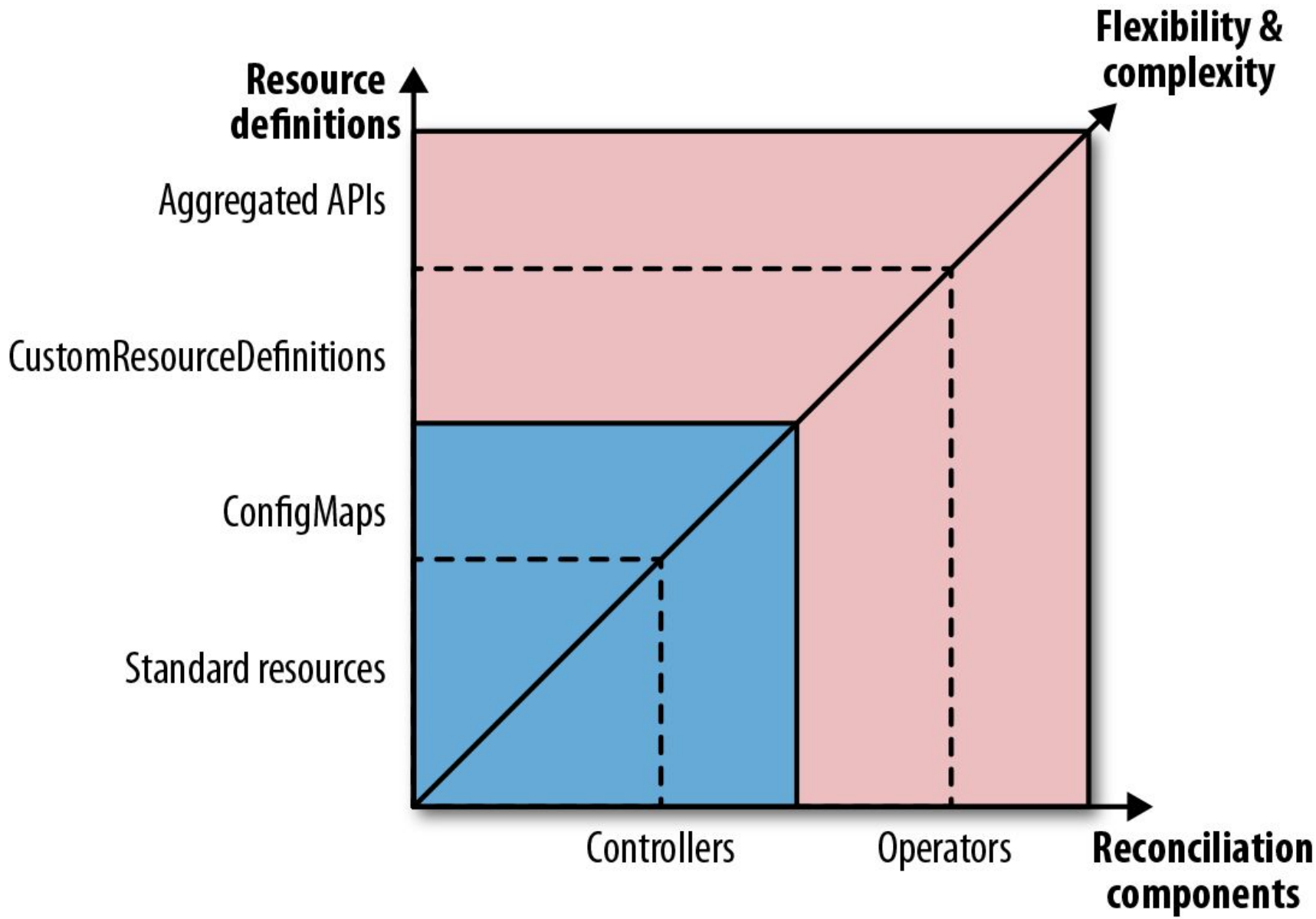
CustomResourceDefinition

Custom resource is modelling a custom domain and managed through the Kubernetes API

```
apiVersion: apiextensions.k8s.io/v1beta1
kind: CustomResourceDefinition
metadata:
  name: configwatchers.k8spatterns.io
spec:
  scope: Namespaced
  group: k8spatterns.io
  version: v1
  names:
    kind: ConfigWatcher
    plural: configwatchers
  validation:
    openAPIV3Schema:
    ...
```

Custom Resource

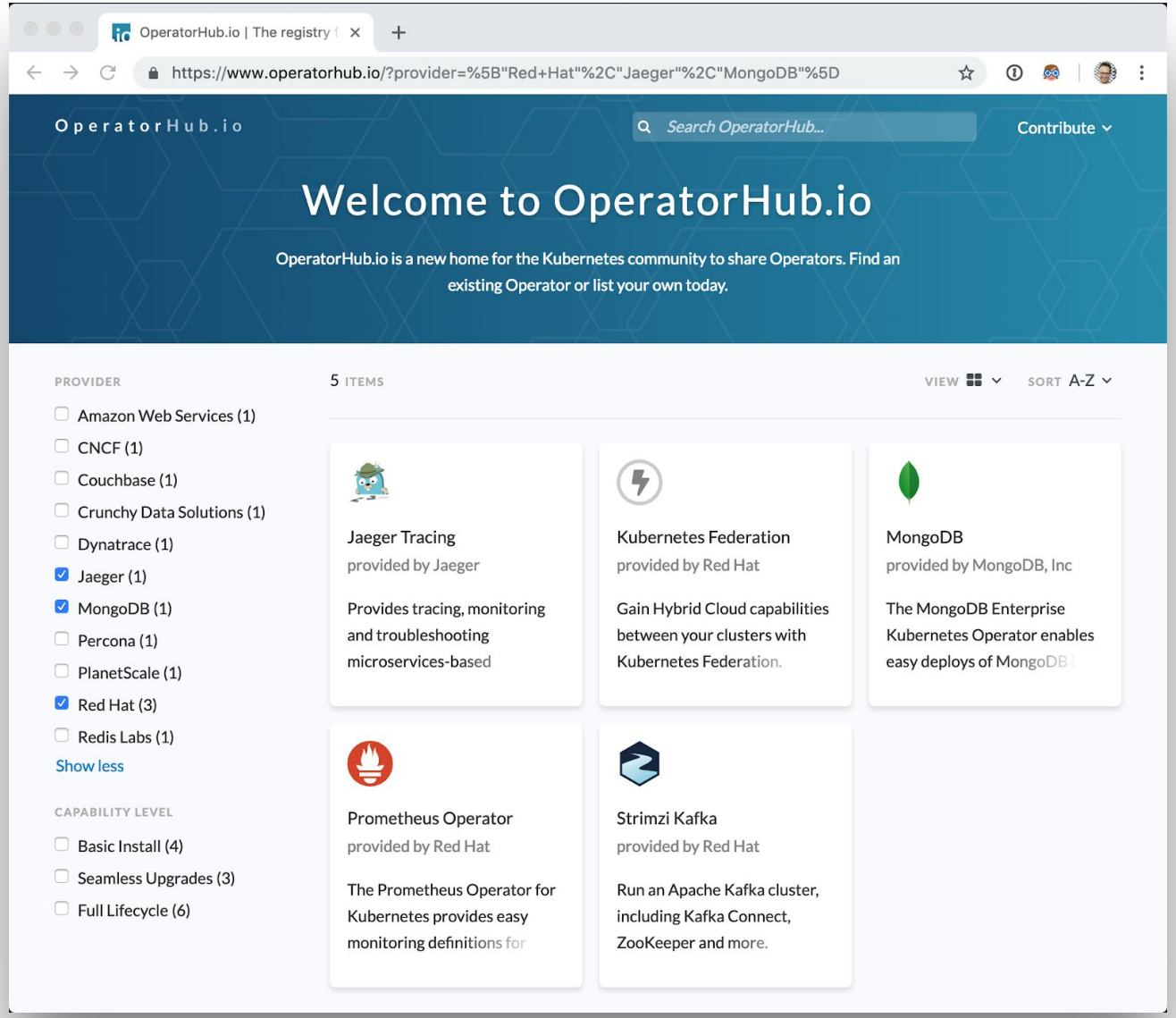
```
kind: ConfigWatcher
apiVersion: k8spatterns.io/v1
metadata:
  name: webapp-config-watcher
spec:
  configMap: webapp-config
  podSelector:
    app: webapp
```


Flexibility & complexity
Resource definitions
Aggregated APIs
CustomResourceDefinitions
ConfigMaps
Standard resources
Controllers
Operators
Reconciliation components

CRD Classification

- Installation CRDs
  - Installing and operating applications
  - Backup and Restore
  - Monitoring and self-healing
  - Example: Prometheus for installing Prometheus & components
- Application CRDs
  - Application specific domain concepts
  - Example: ServiceMonitor for registering Kubernetes service to be scraped by Prometheus

Operator Hub



Red Hat

Operator Development

- Operator can be implemented in any language
- Frameworks:
  - Operator Framework (Golang, Helm, Ansible) https://github.com/operator-framework
  - Kubebuilder (Golang) https://github.com/kubernetes-sigs/kubebuilder
  - Metacontroller (Language agnostic) https://metacontroller.app/
  - jvm-operators (Java, Groovy, Kotlin, ...) https://github.com/jvm-operators

Thank you


https://k8spatterns.io


https://twitter.com/ro14nd


https://twitter.com/bibryam


https://twitter.com/k8spatterns

Picture credits: http://bit.ly/k8spatterns-picture-credits

Red Hat

Picture Credits

https://www.pexels.com/photo/brown-and-black-pattern-2158386/

https://pixabay.com/photos/ship-helm-sunset-cutter-coast-guard-759954/

https://unsplash.com/photos/yo01Z-9HQAw

https://www.pexels.com/photo/turned-on-light-crane-1117211/

https://www.pexels.com/photo/shallow-focus-photography-of-black-ship-109581
4/

https://pixabay.com/photos/containers-storage-rusted-rusty-1209079/

https://pixabay.com/photos/motocross-sidecar-race-motorsport-1045661/

https://www.pexels.com/photo/golden-gate-bridge-san-francisco-california-1141
74/

https://www.pexels.com/photo/reflection-playstation-pad-gaming-181
74/

https://unsplash.com/photos/UJP_QpCKj7M

https://www.pexels.com/photo/grayscale-photo-of-person-holding-c
hess-piece-1498958/

https://pixabay.com/photos/cans-manufacturing-business-2888650/

https://unsplash.com/photos/IRoX0shwjUQ

https://www.freeimages.com/photo/poppy-in-wheat-1344010

https://pixabay.com/photos/telescope-field-glass-spyglass-1966366/

https://unsplash.com/photos/UHDx3BHIFvY

https://pixabay.com/photos/bake-advent-christmas-cookie-1786926/

https://pixabay.com/photos/balloons-colors-party-celebration-18697
90/

https://unsplash.com/photos/ymf4_9Y9S_A