

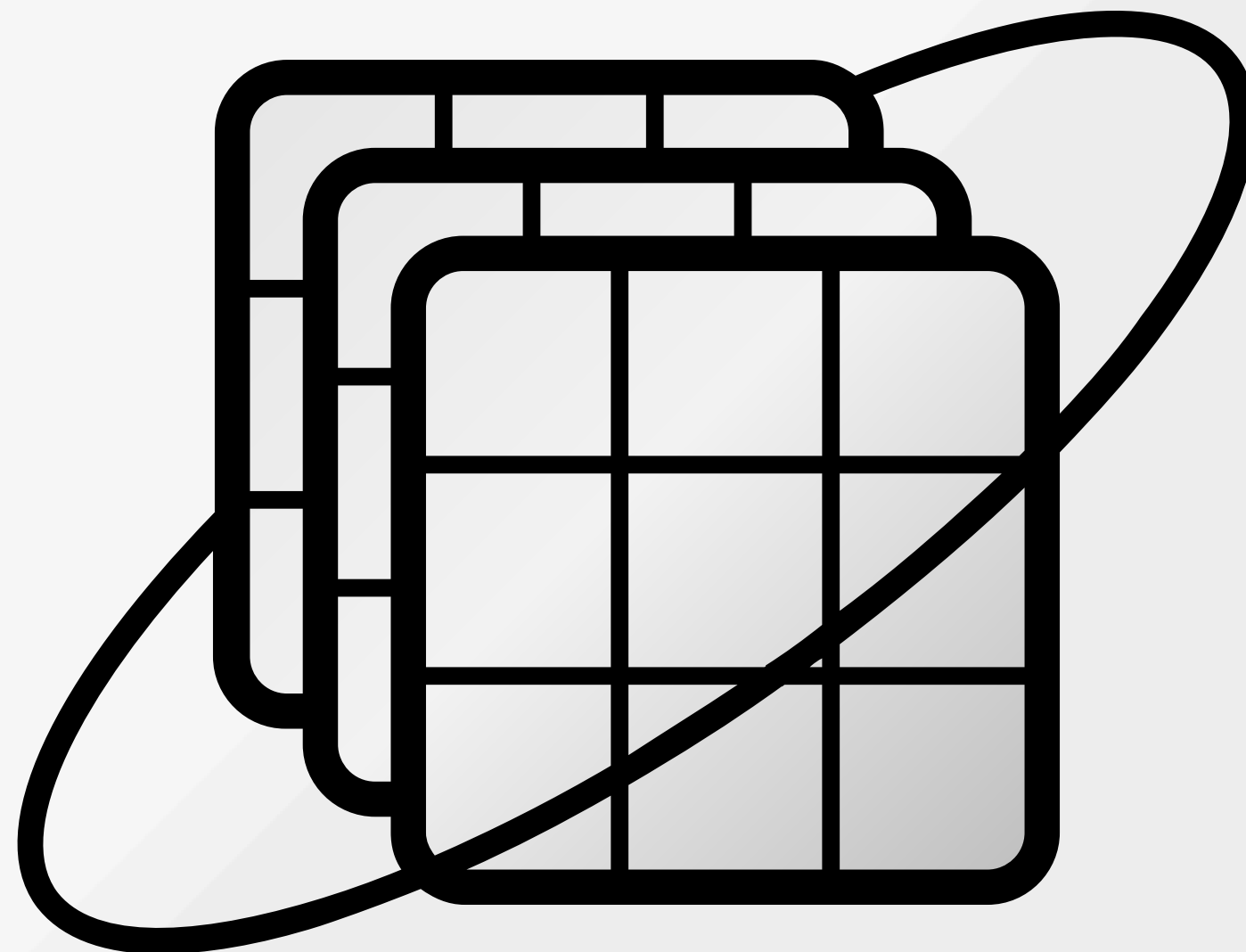


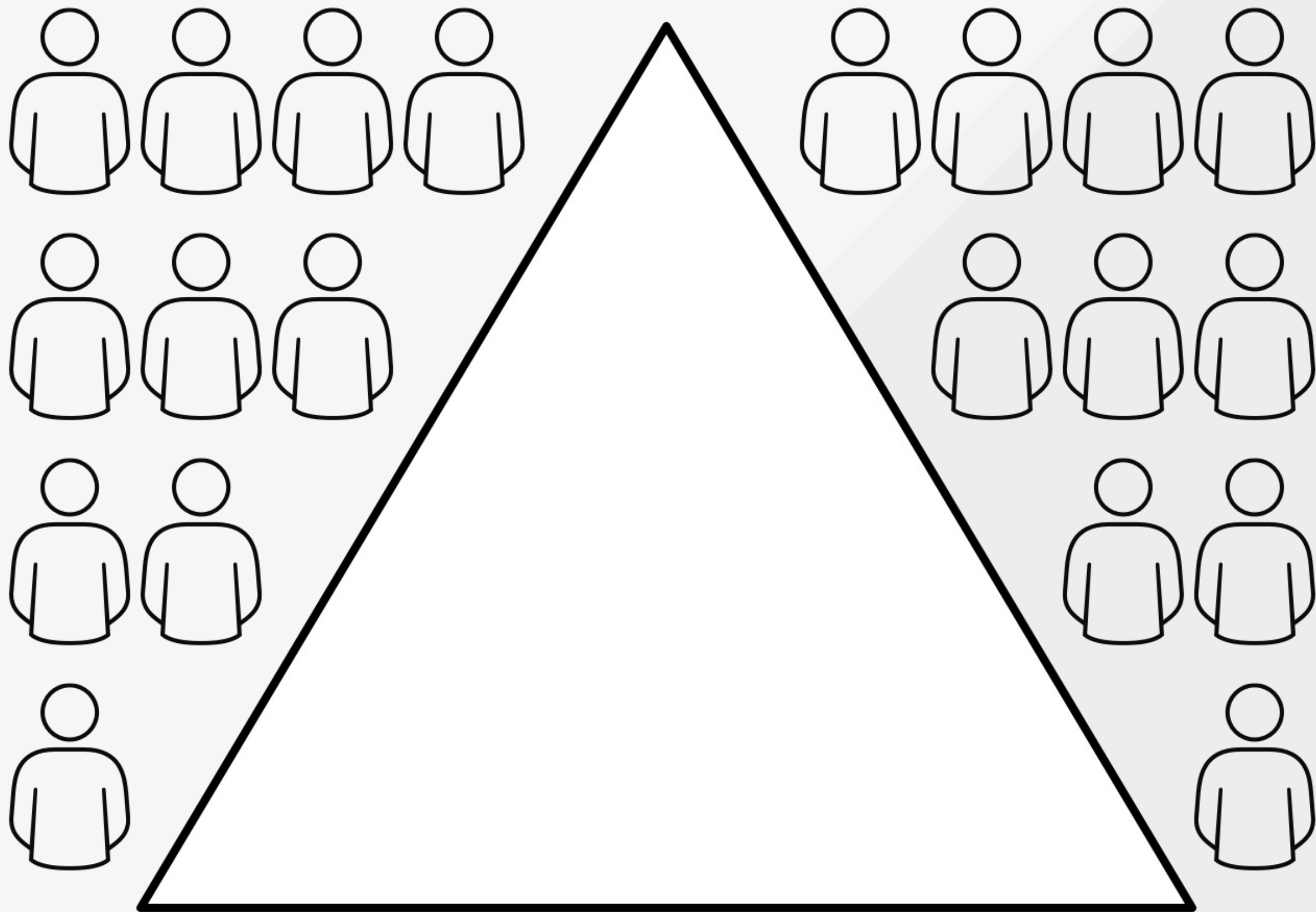
# KUBERNETES FOR JAVA DEVELOPERS

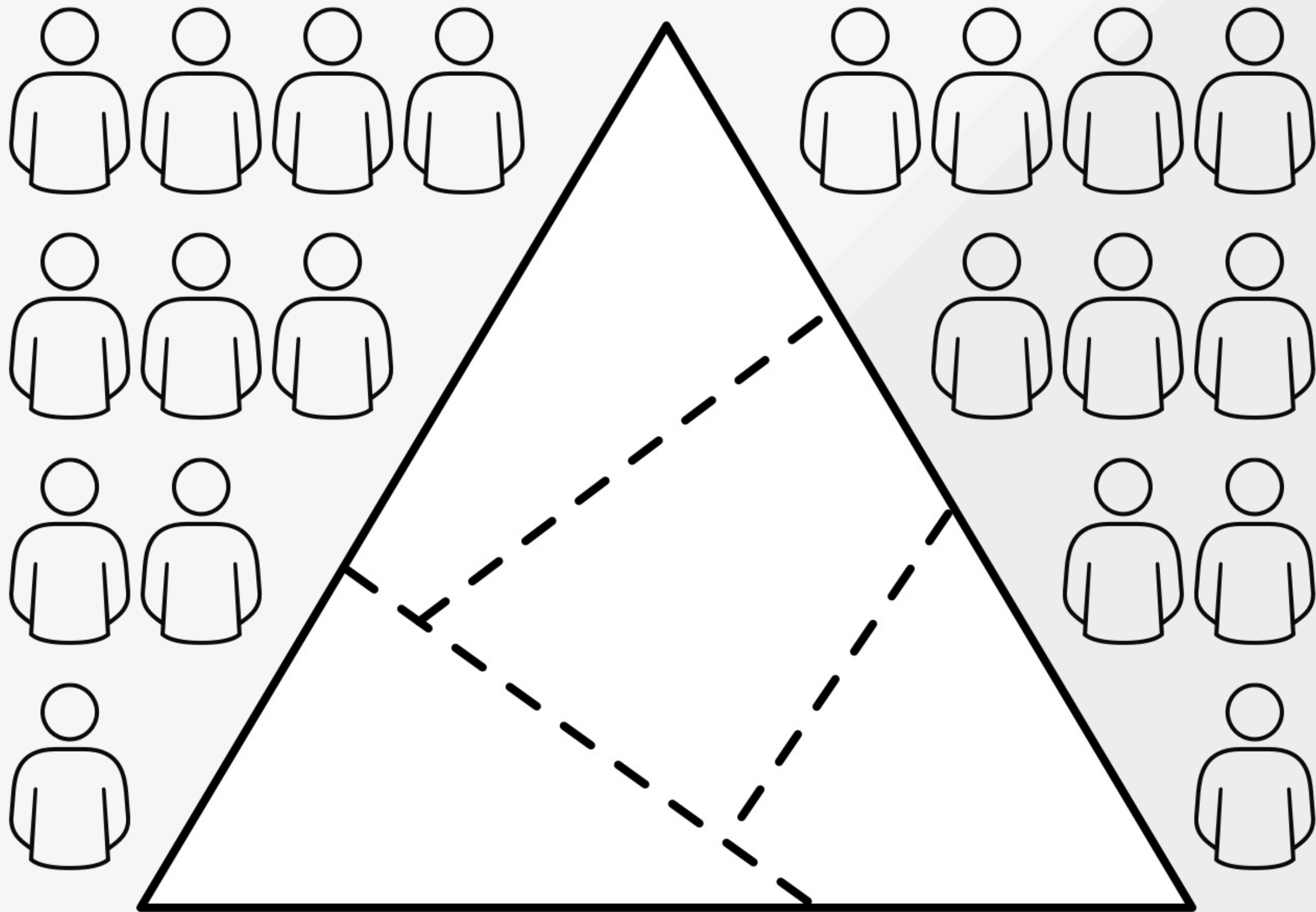
Dr. Roland Huß, Red Hat, @ro14nd

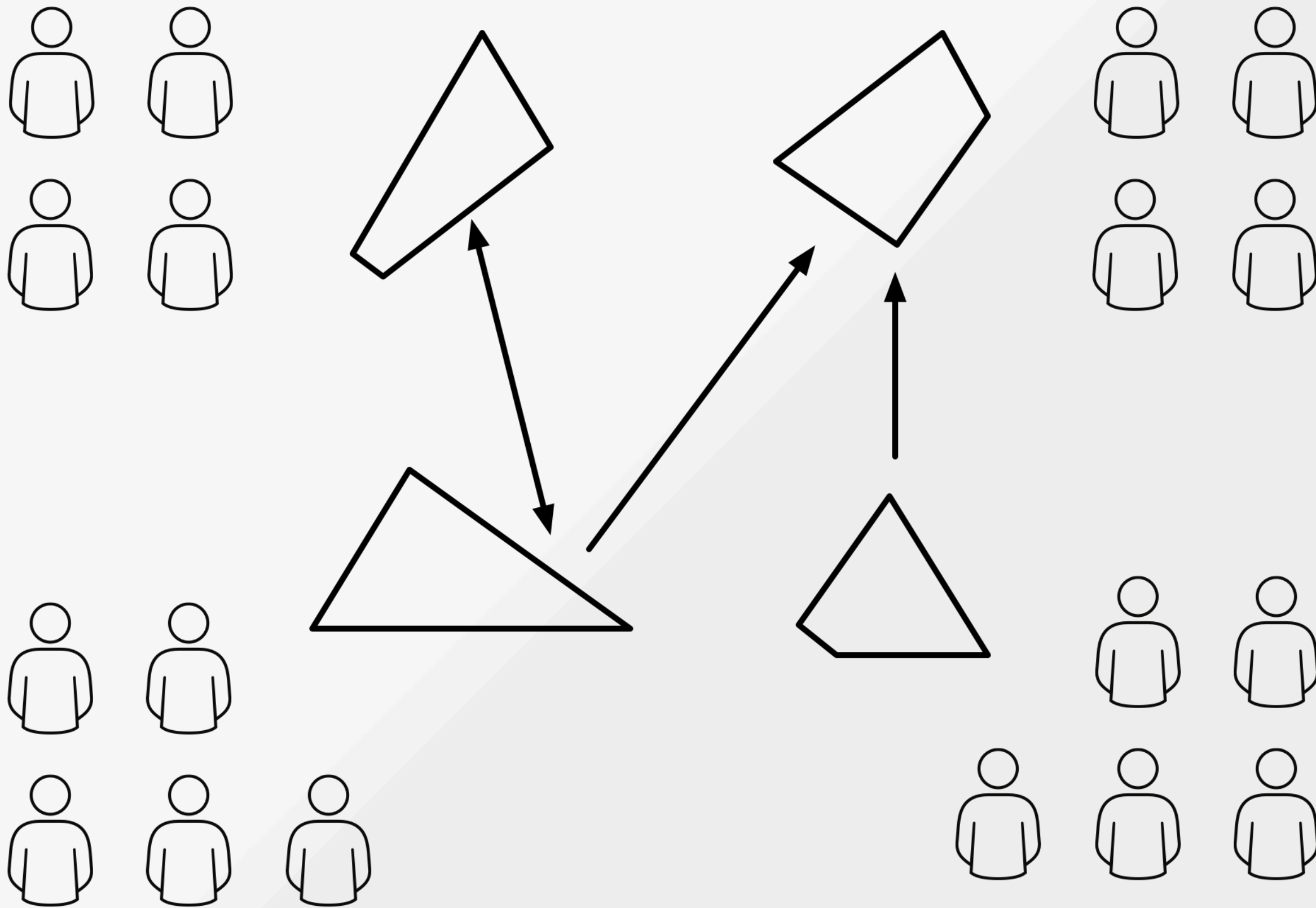
"m" for menu, "?" for other shortcuts

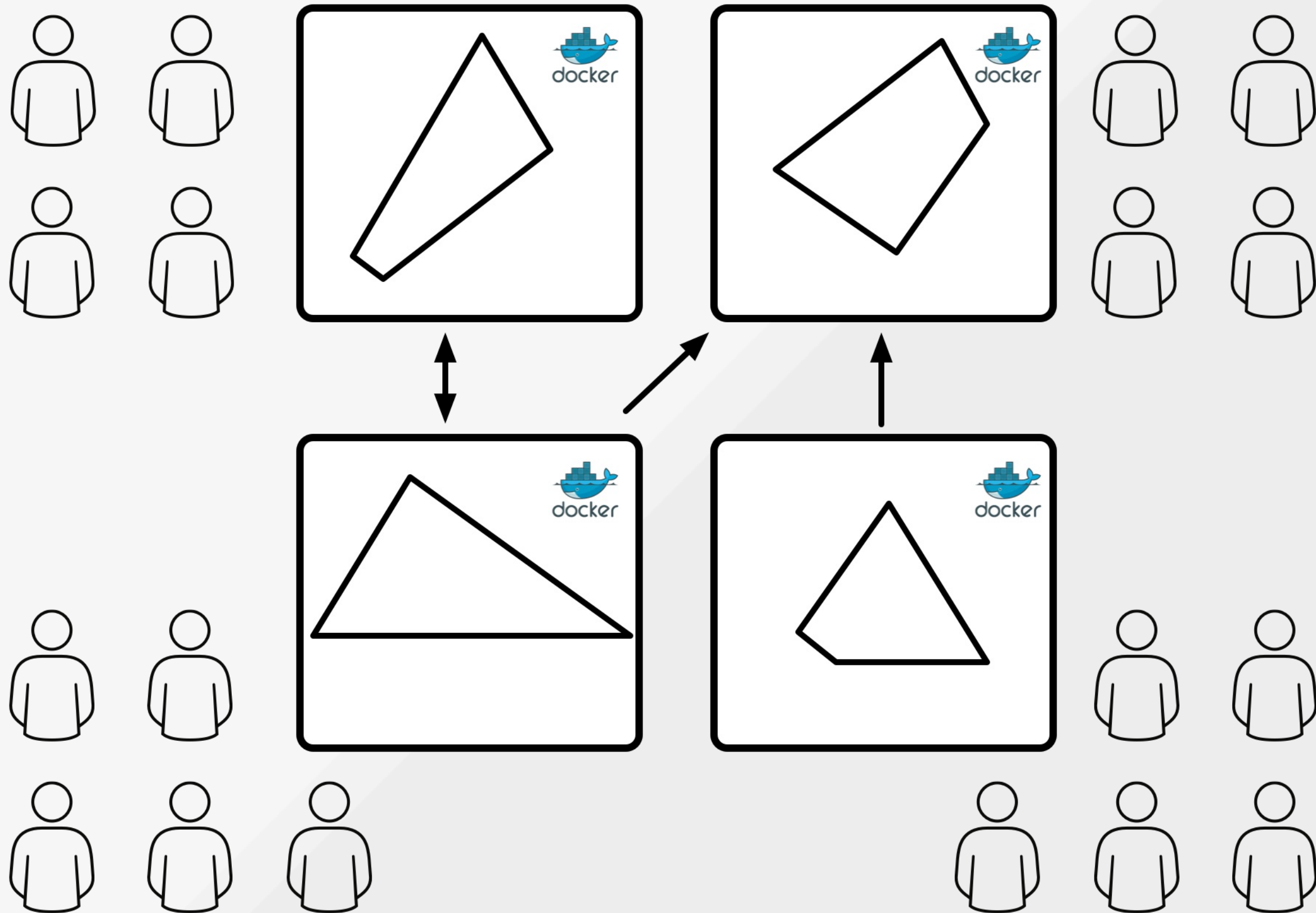


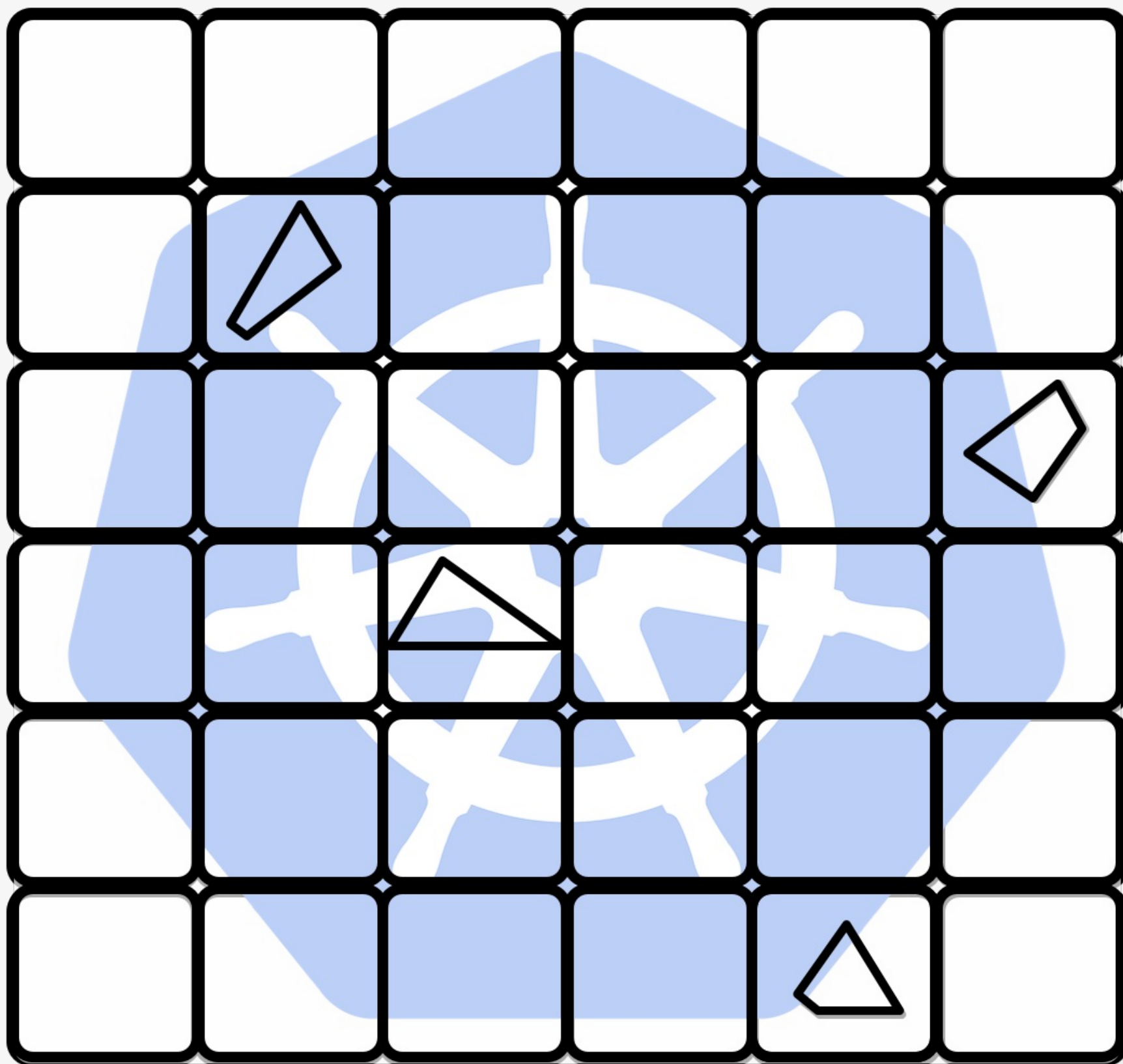














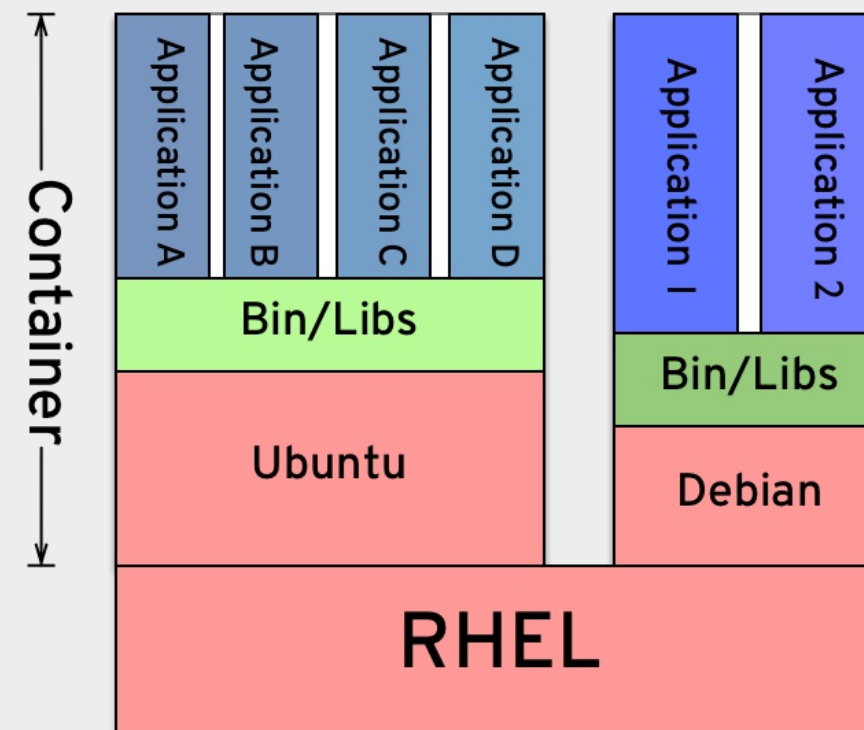
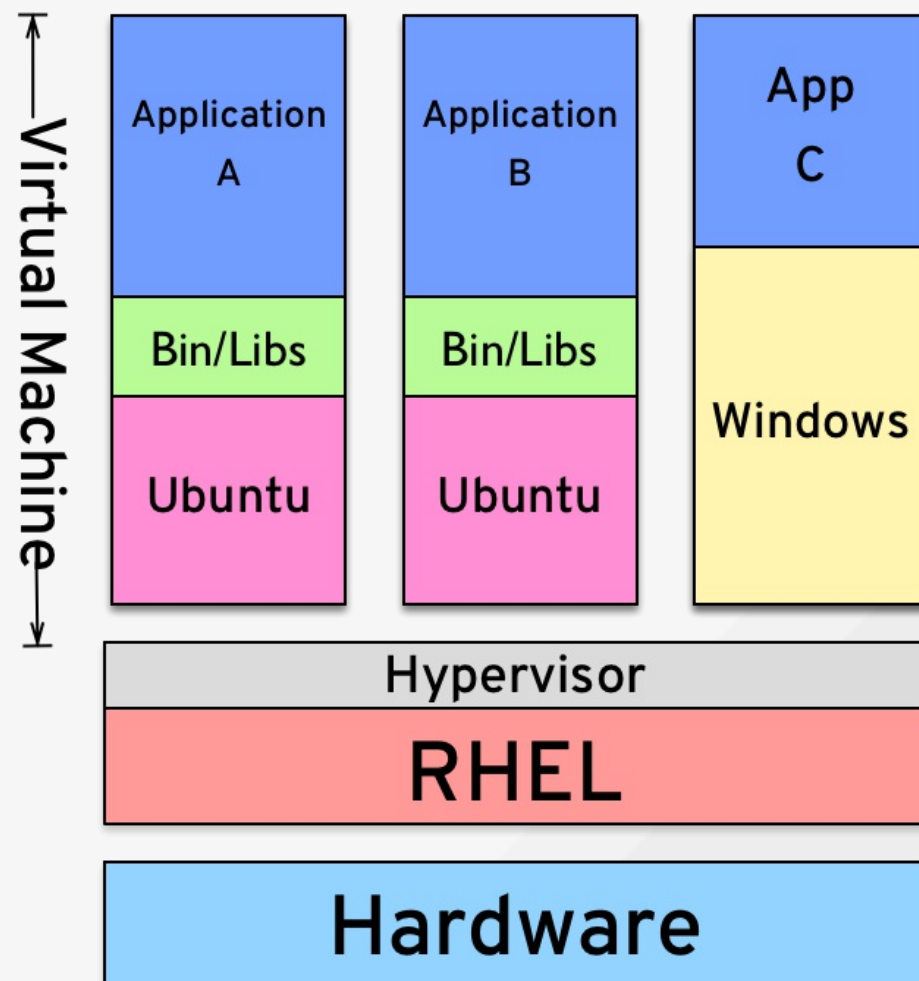


# docker



# VM VS. CONTAINER

Containers are **isolated**, but share the kernel and (some) files  
➔ **lighter** and **faster**



# DOCKERFILE

```
FROM java:alpine-jre-8  
RUN apk update  
ADD microservice.jar /  
EXPOSE 8080  
CMD java -jar /microservice.jar
```

Build:

```
docker build -t rhuss/microservice:1.3.1 .
```

# DOCKER-MAVEN-PLUGIN

- Building images within Maven build
- Versioned
- Assembly with dependencies
- No Docker client required

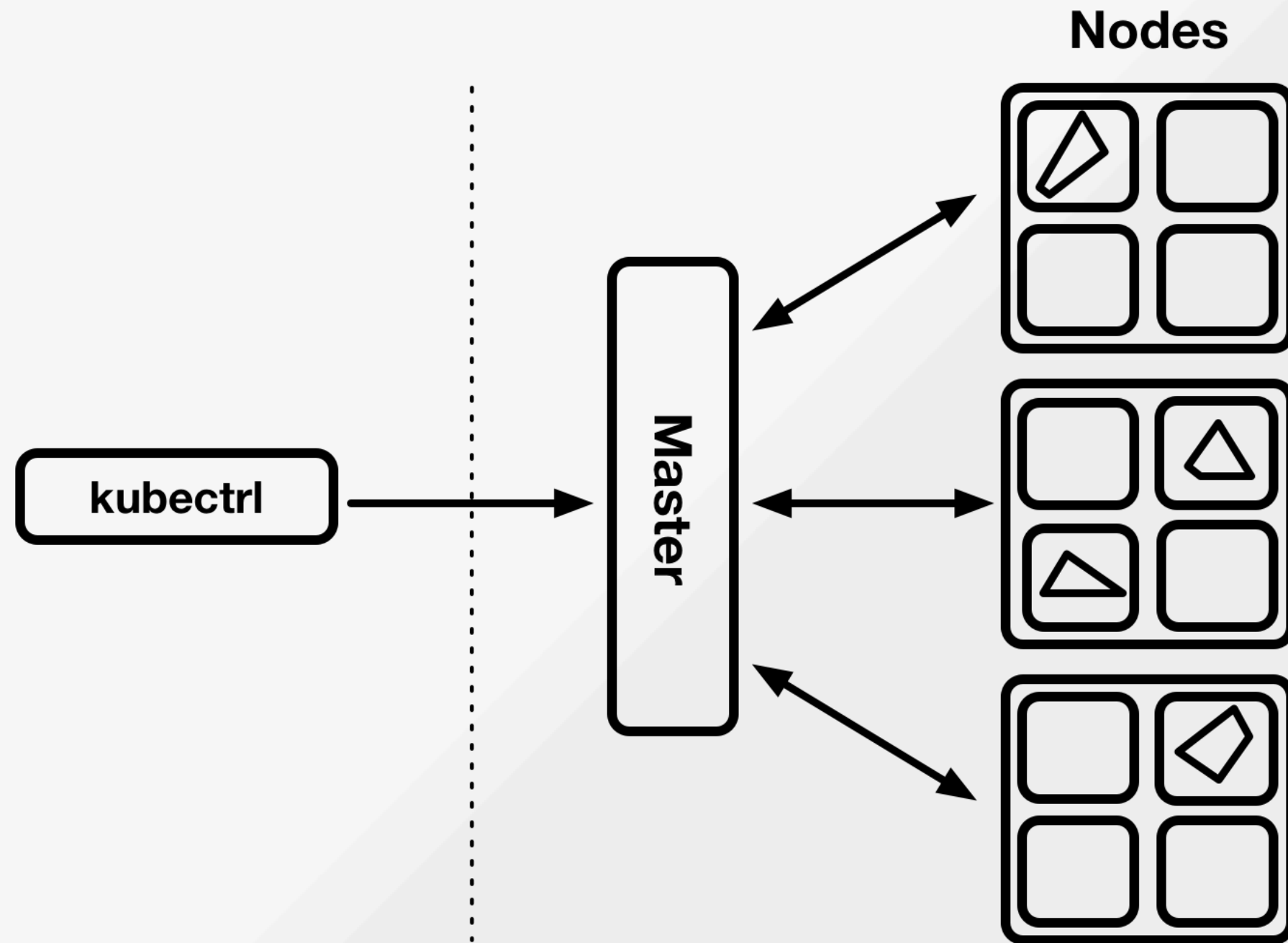
**<https://github.com/fabric8io/docker-maven-plugin>**



# KUBERNETES

- Open Source orchestration system for Docker containers
  - Scheduling
  - Horizontal scaling
  - Self-healing
  - Service discovery
  - Automated rollout and rollbacks

# ARCHITECTURE



# MINIKUBE

- Single-node Kubernetes cluster inside a VM
- No Docker daemon required
- Ideal for local development
- Supports DNS, NodePorts, Volumes, ...
- <https://github.com/kubernetes/minikube>





# fabric8

# FABRIC8

- Open Source **Microservices Platform**
- Supports Kubernetes & OpenShift
- Themes:
  - Continuous Delivery
  - Management UI
  - Funktion
  - Quickstarts
  - Tooling

# FABRIC8-MAVEN-PLUGIN

- Embeds docker-maven-plugin
- Creates resource descriptors from build configuration
- Support for descriptor templates which can be **enriched**
- <https://maven.fabric8.io>

# GOALS

<code>fabric8:build</code>	Build application images (Docker, S2I binary, S2I source)
<code>fabric8:resource</code>	Create Kubernetes and OpenShift resource descriptors
<code>fabric8:apply</code>	Apply resource descriptors to a running cluster

# CONFIGURATION

- Zero Config
  - Opinionated Defaults
  - Limited configuration options
- XML Configuration
  - Restricted configuration syntax
- Resource Fragments
  - Most powerful
  - Verbose

# ZERO CONFIG

- **Generators** for Image generation

```
<build>
  <plugins>
    <plugin>
      <groupId>io.fabric8</groupId>
      <artifactId>fabric8-maven-plugin</artifactId>
      <version>3.3.0</version>
    </plugin>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

# RESOURCE FRAGMENTS

- Resource fragment

src/main/fabric8/pong-rc.yml

```
spec:  
  replicas: 1  
  template:  
    spec:  
      containers:  
        - name: pong  
          ports:  
            - containerPort: 8080
```

- **Enrichers** add missing pieces



# K8S & OPENSIFT

- Kubernetes:
  - Docker builds
  - Deployments
  - Ingress
- OpenShift
  - S2I & Docker Binary Builds
  - DeploymentConfig
  - ImageStream
  - Template

# MISC

<code>fabric8:install</code>	Install local development environment
<code>fabric8:cluster-start</code>	Start minikube or minishift
<code>fabric8:watch</code>	Watch for changes and redployments
<code>fabric8:debug</code>	Debug into pods

# RASPI CLUSTER

- 4 Raspberry Pi 3
- Wifi Router
- 6 Port USB charger
- 32 GB SD-Cards
- Costs: ~ 300 €
- Hypriot OS image



Full Story: <https://ro14nd.de/kubernetes-on-raspberry-pi3>

# ANSIBLE

- Ansible Playbooks for setting up Kubernetes
  - based on kubeadm
  - Flannel overlay network
  - SkyDns Addon
- Soon:
  - Registry
  - Ingress Controller (Router)
  - OpenShift

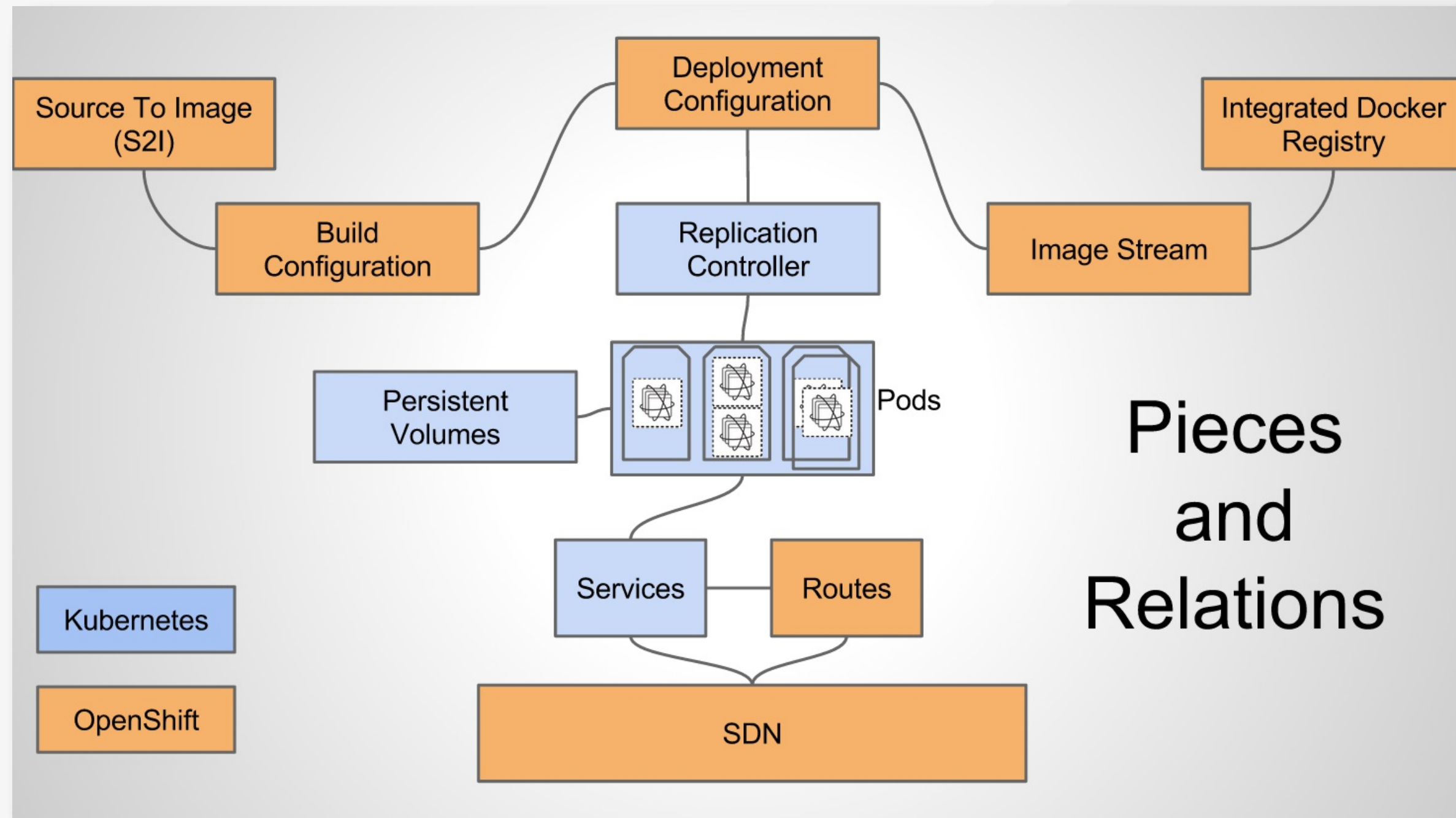


# OPENSHIFT

# OPENSSHIFT

- Adds the **BUILD** to Kubernetes
- Developer and Operation Tools
- Infrastructure Services
  - Registry, Router, OAuth2 Security
- Volume Management
- Multi tenancy
- Management UI

# OPENSHIFT EXTRAS





# MINISHIFT

- Single-node OpenShift Origin cluster inside a VM
- Based on oc cluster up
- Supports routes, registry, s2i builds, ...
- <https://github.com/minishift/minishift>

# WRAP UP

- Starting with Kubernetes can be almost as easy as with Docker ;-)
- Kubernetes and OpenShift are powerful orchestration platforms with enterprise grade features.
- Use fabric8-maven-plugin for Java apps

# Kubernetes Patterns



Patterns, Principles, and Practices  
for Designing Cloud Native Applications

Bilgin Ibryam & Roland Huss

<https://leanpub.com/k8spatterns>



# QUESTIONS ?

Blog <https://ro14nd.de>

---

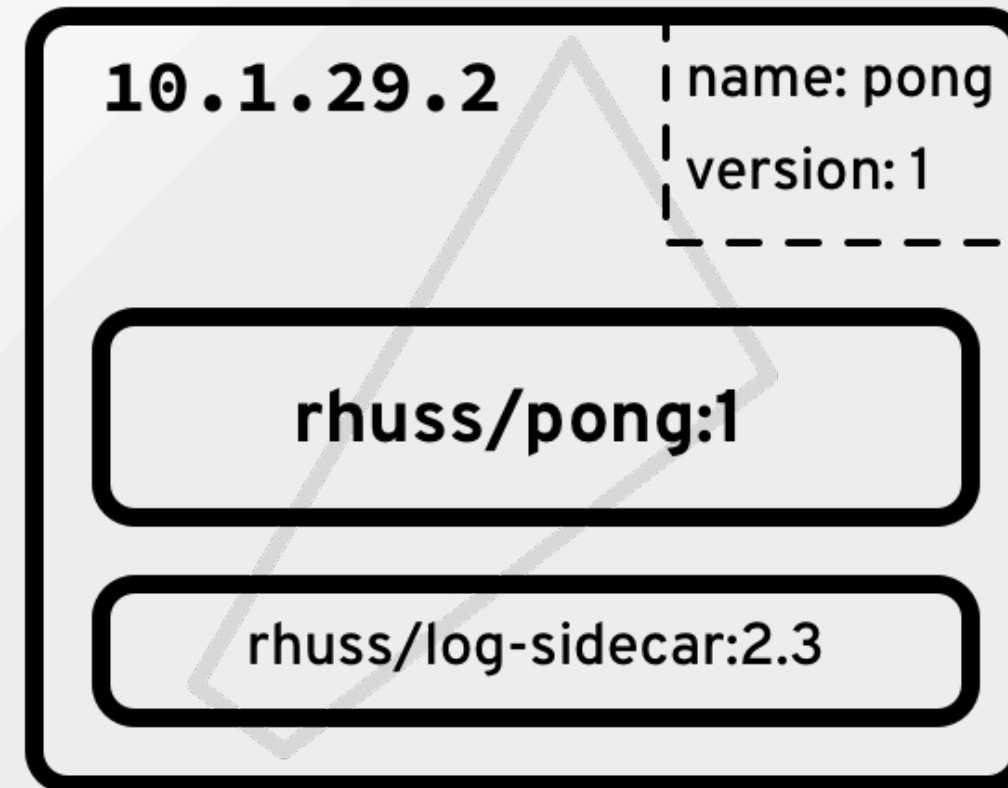
Slides `firefox $(curl -sL bit.ly/k8s-for-java-devs | sh)`





# POD

- Kubernetes Atom
- One or more containers sharing:
  - IP and ports
  - Volumes
- Ephemeral IP address



# LABELS

- Metadata attachable to every resource object
- Used to categorize stuff
- Important for selectors
- "Freeform"



# DEMO

# REPLICATION CONTROLLER

- Responsible for managing **Pods**
- **replicas** : Number of **Pod** copies to keep
- Label selector choose **Pods**
- Holds a template for creating new **Pods**

# Replication Controller

replicaCount:

**3**

Selector:

name: pong  
version: 1



**10.1.29.2**

name: pong  
version: 1

**10.1.29.3**

name: pong  
version: 1

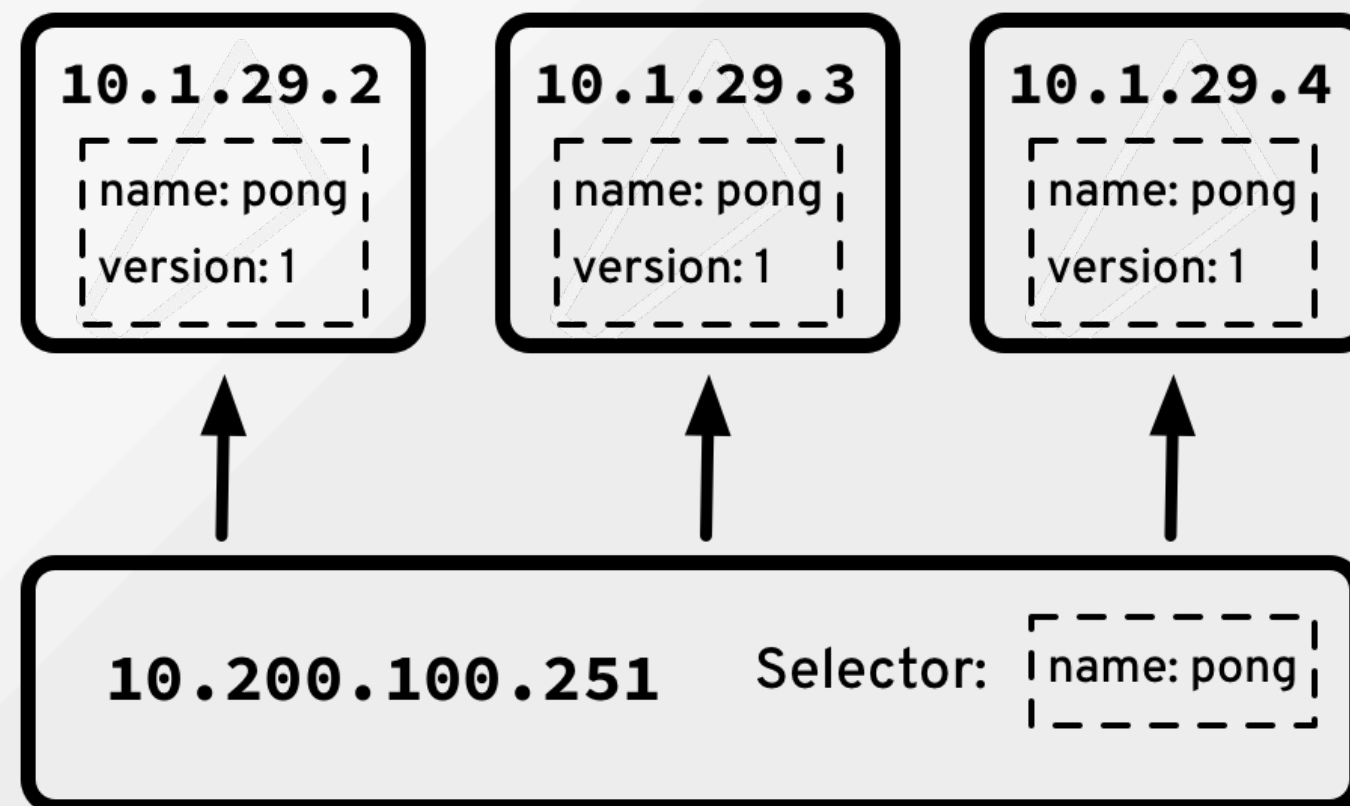
**10.1.29.4**

name: pong  
version: 1

# DEMO

# SERVICE

- Proxy for a set of **Pods**
- **Pods** selected by **Label** selector
- Permanent IP address



# DEMO

# ROLLING UPDATE

- kubectl rolling-update
- Downscale of old **replication controller**
- Upscale of new **replication controller**

# DEMO



# VOLUMES

- Distributed storage
- Support types:
  - Local
  - NFS
  - Gluster
  - Ceph
  - ...

# MISC FEATURES

- Secrets
- ConfigMaps
- ServiceAccounts
- Health & Liveness Checks
- Ingress