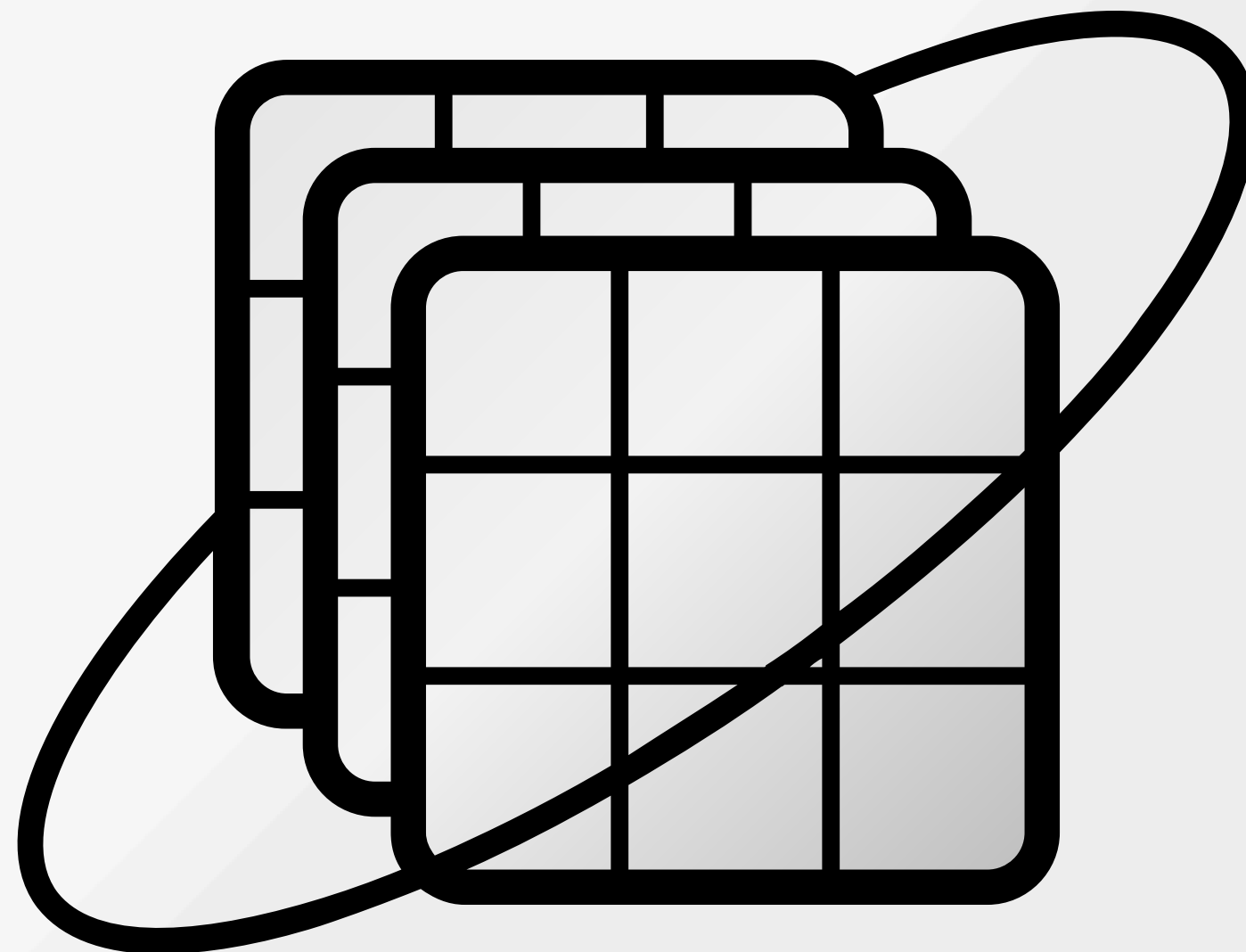


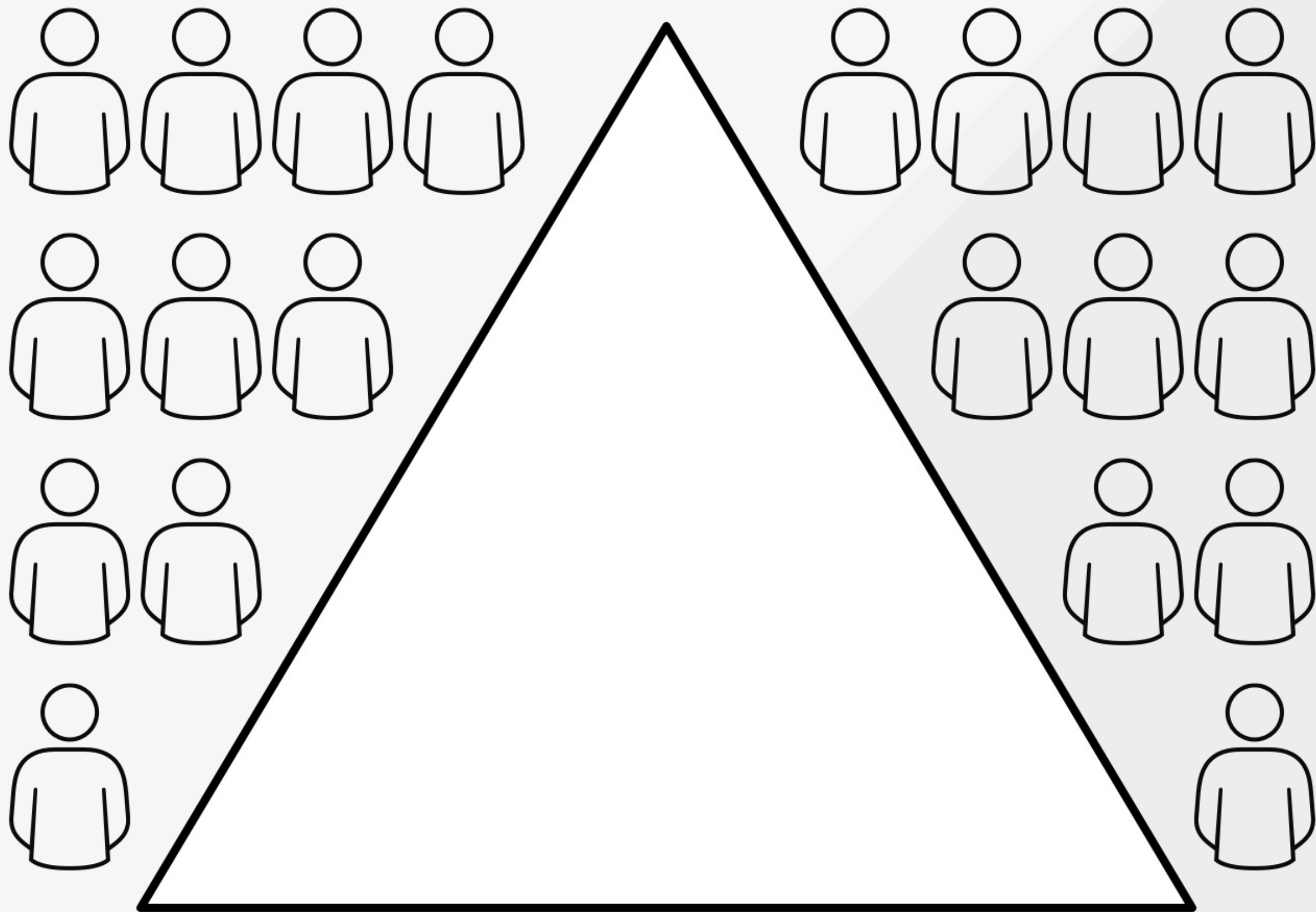


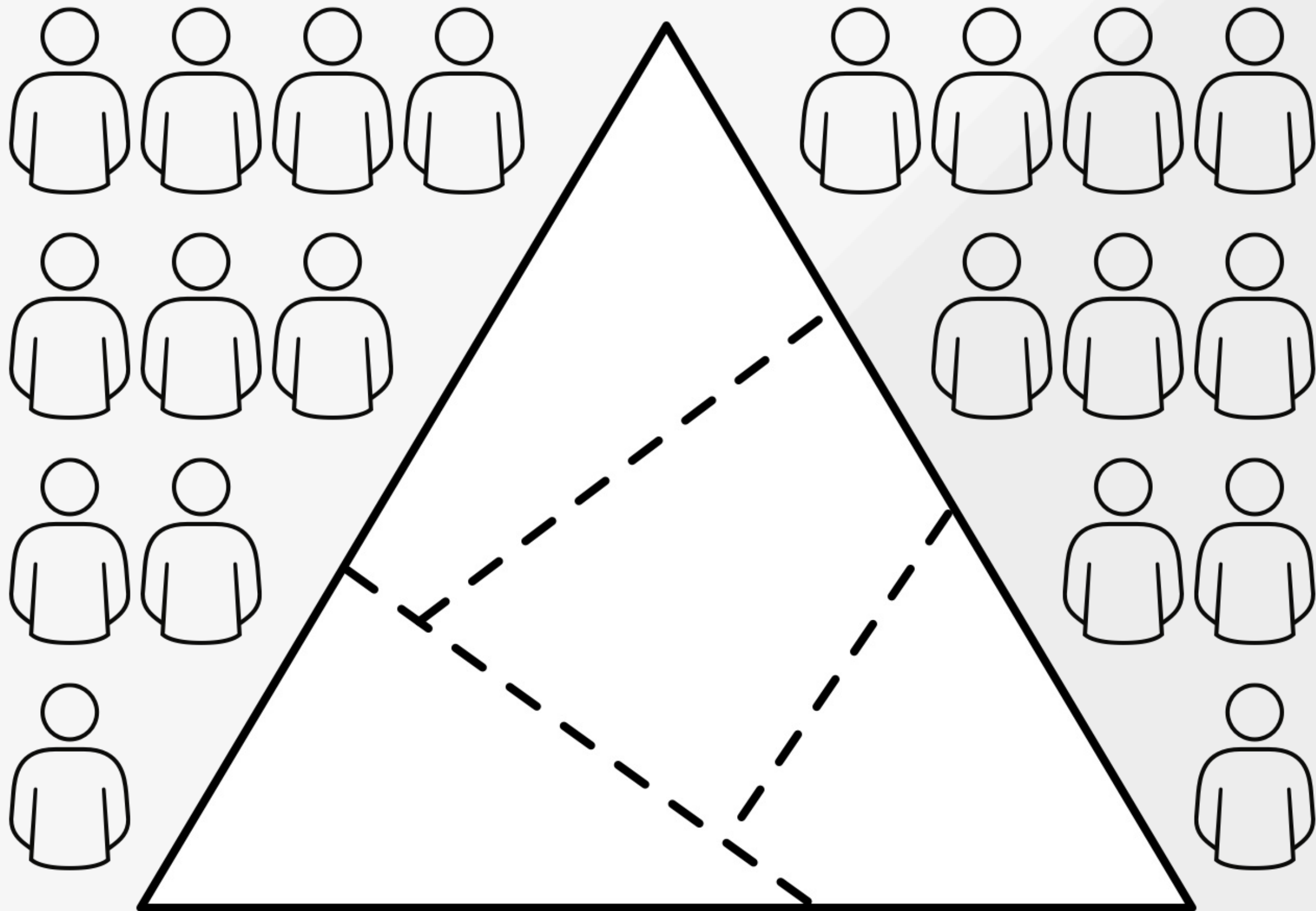
# JAVA DEVELOPMENT IN THE AGE OF THE WHALE

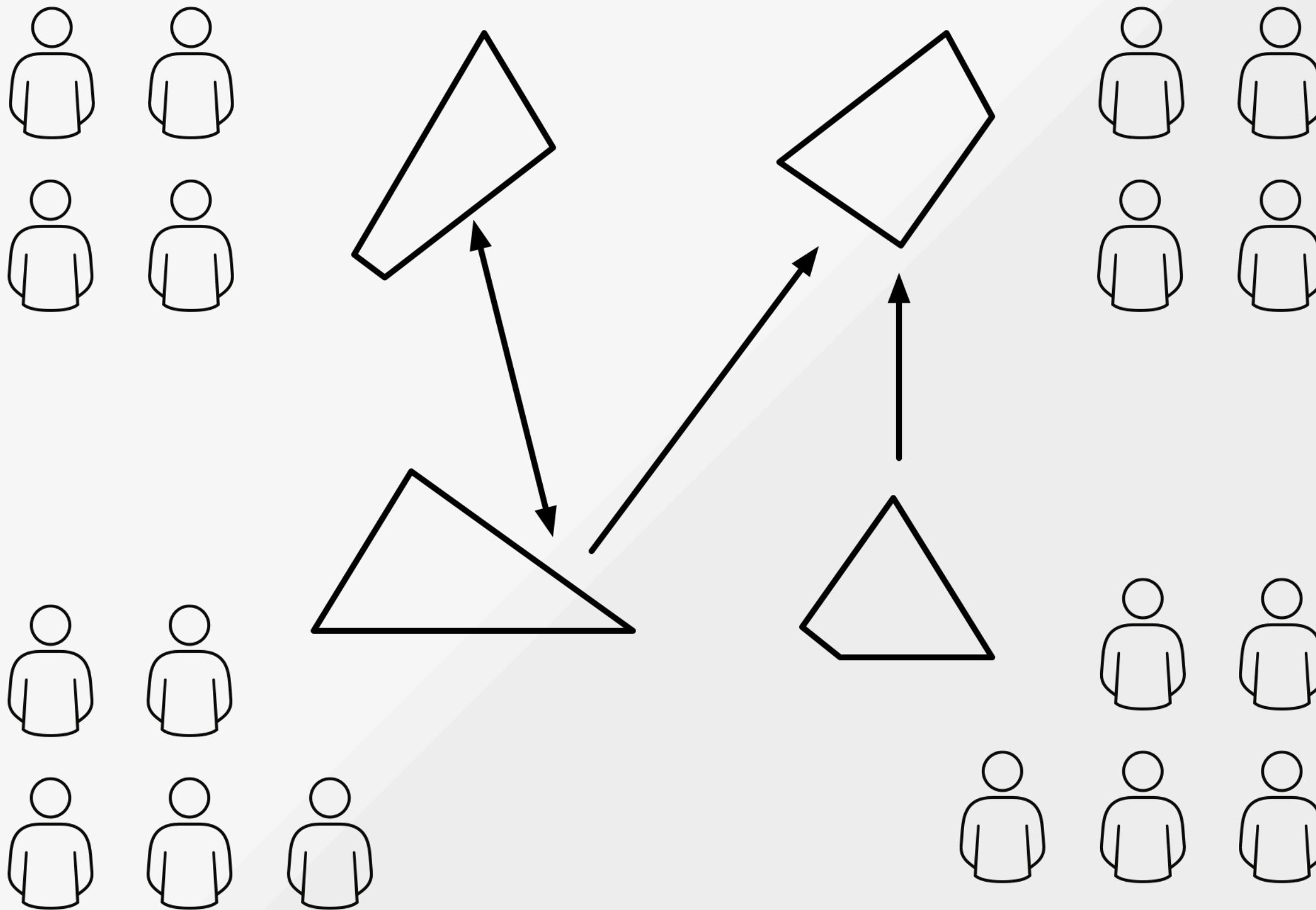
ObjektForum Mannheim, i06/2018

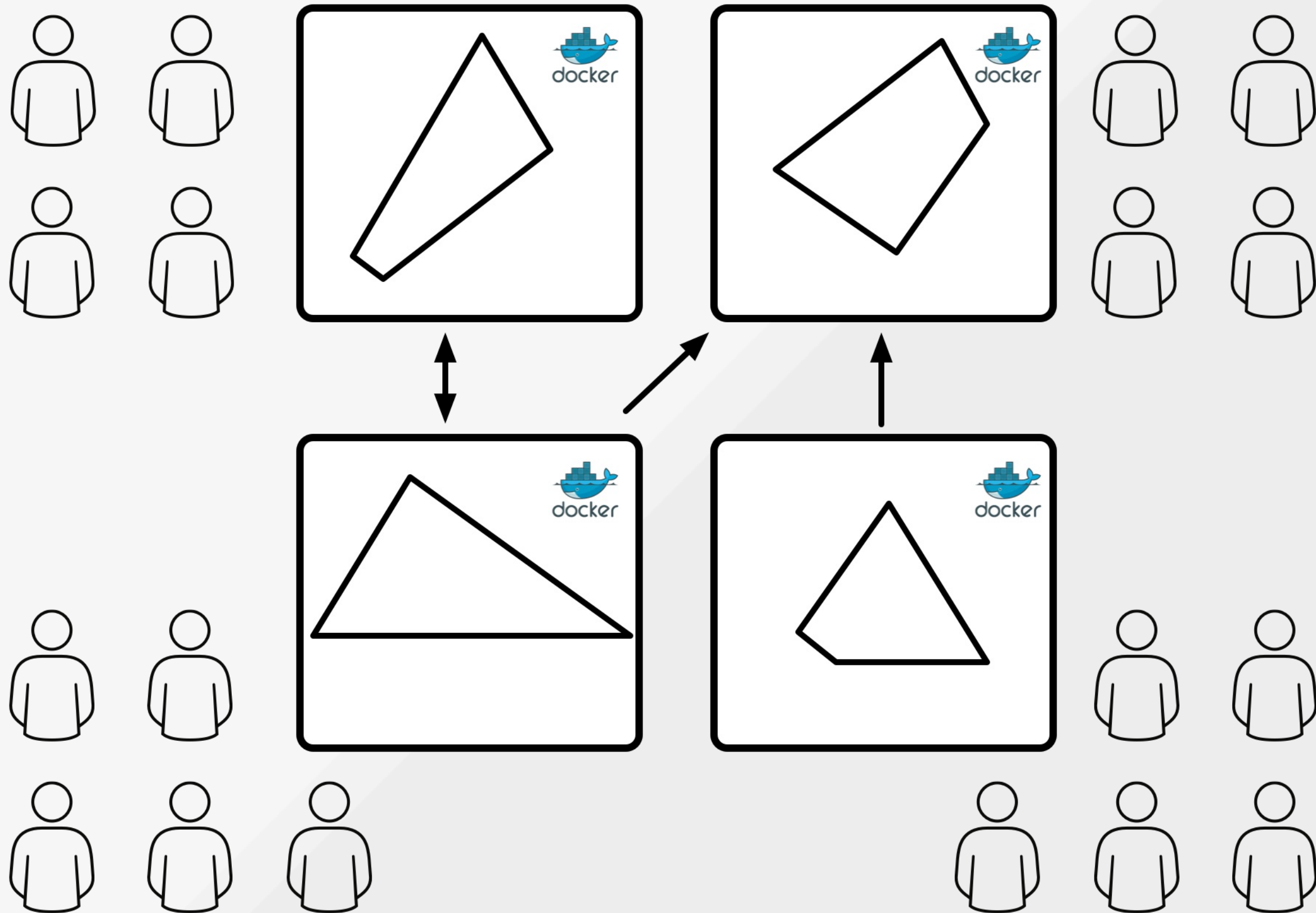
Roland Huß, Red Hat, @ro14nd

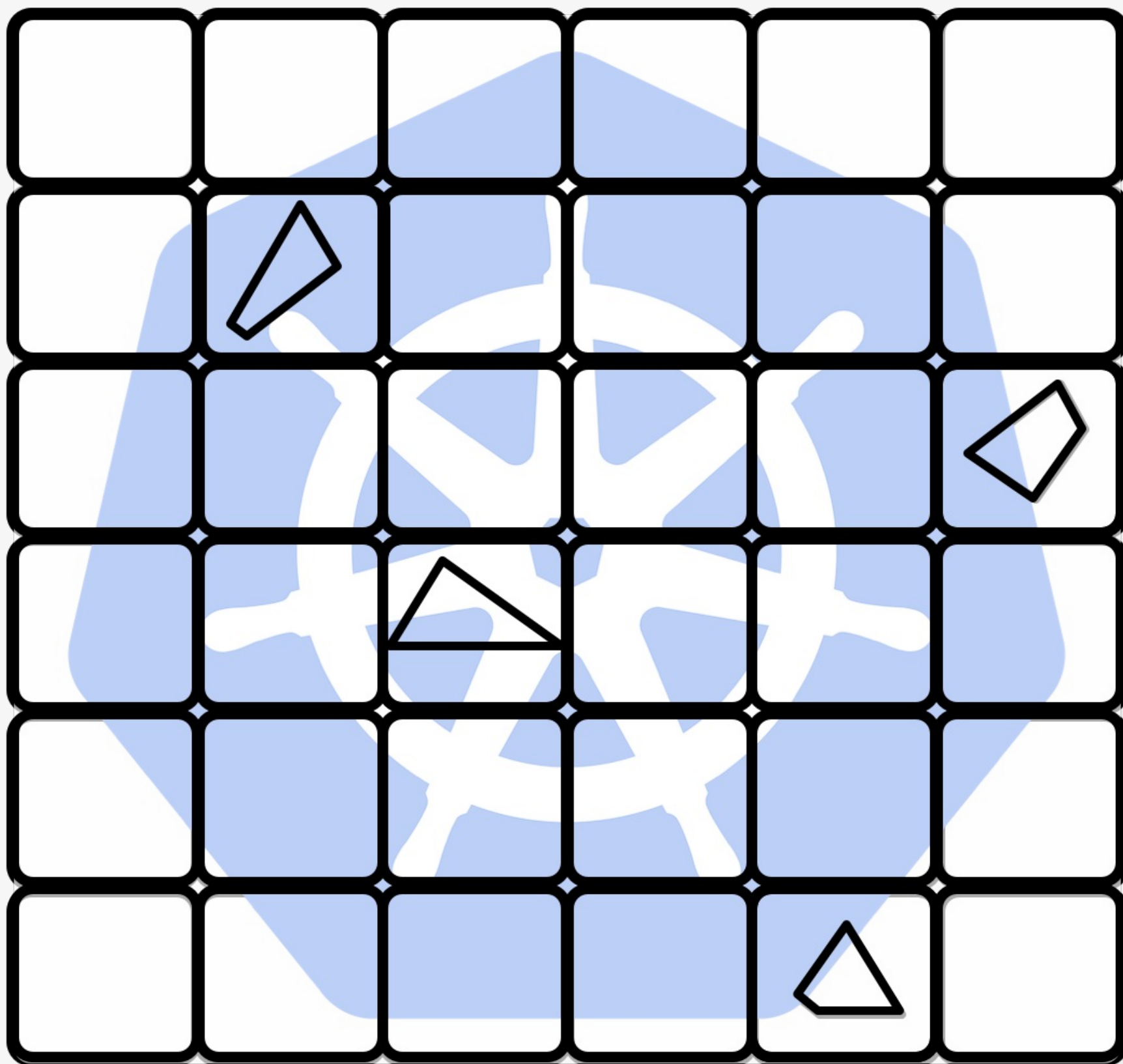














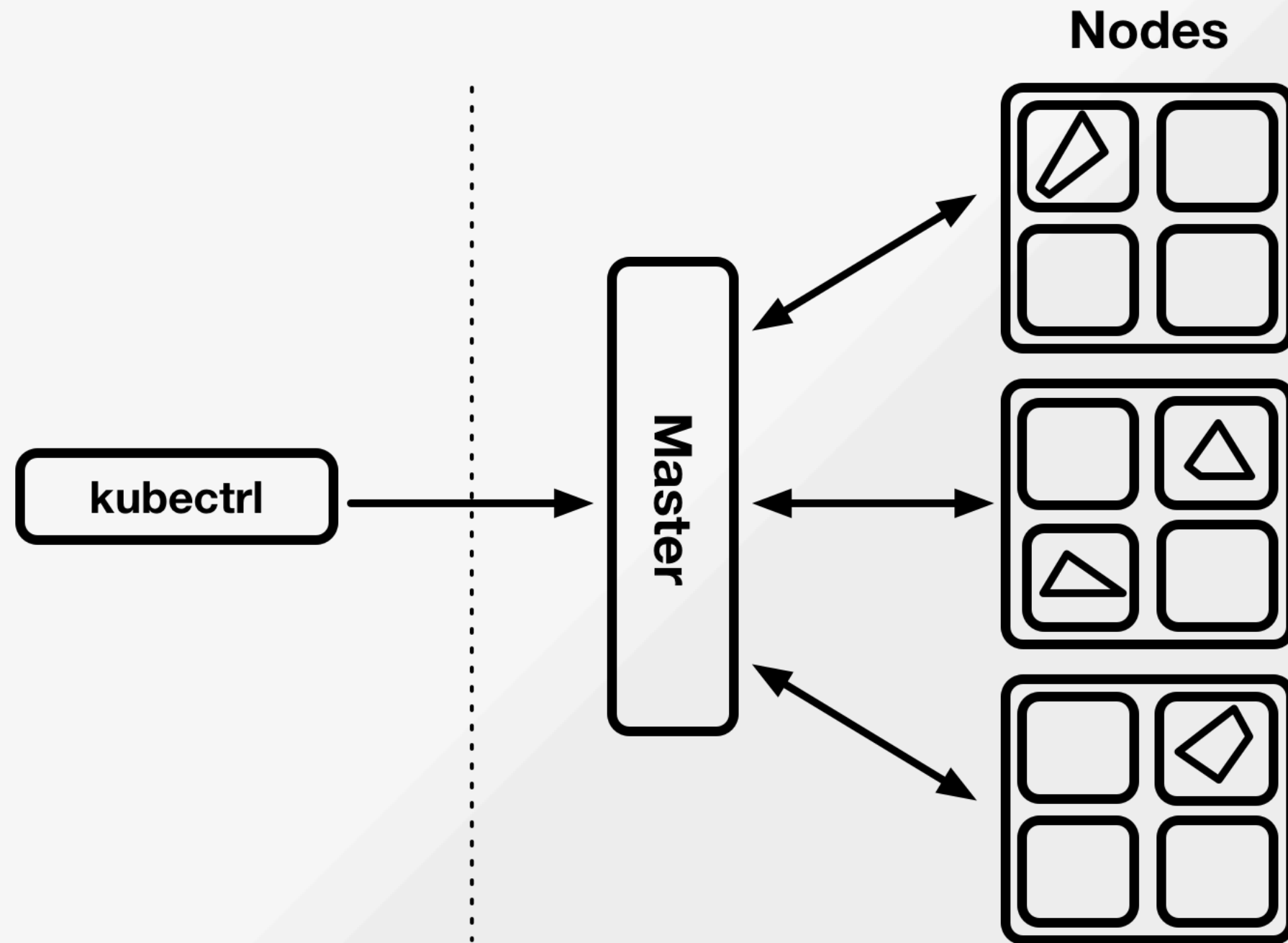




# KUBERNETES

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

# ARCHITECTURE



# KUBERNETES IN THE CLOUD

- Google Kubernetes Engine (GKE)
- Azure Container Service (AKS)
- OpenShift Online
- AWS EC2, Digital Ocean, ...
  - Stackpoint.io
  - CoreOS Tectonic
  - Kubernetes Operations (kops)

# RASPI CLUSTER

- 4 Raspberry Pi 3
- Wifi Router
- 6 Port USB charger
- 32 GB SD-Cards
- Costs: ~ 300 €
- Install via Ansible
- **kubeadm**



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

# ANSIBLE

- Ansible Playbooks for setting up Kubernetes
  - based on kubeadm
  - Flannel overlay network
  - Admin UI, Ingress Controller (Traefik)
- Soon:
  - Registry
  - OpenShift

<https://github.com/Project31/ansible-kubernetes-openshift-pi3>

# 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

- **Microservices Platform** for Kubernetes & OpenShift
- Upstream projects for [openshift.io](https://openshift.io)
- Themes:
  - Continuous Delivery
  - Management UI
  - Quickstarts
  - Tooling

# FABRIC8-MAVEN-PLUGIN

- Creates **Docker images** and **resource descriptors**
- Zero-configuration mode with **opinionated defaults**
- <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>3.5.39</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
```

# GENERATORS

- Extract Docker image configuration from pom.xml
- Supported technologies:
  - Spring Boot
  - Wildfly Swarm
  - Fat Jars
  - Eclipse Vert.x
  - Webapps
  - Karaf



# ENRICHERS

- Add default Kubernetes resources
- Update existing resources
- E.g.
  - Default Deployment and Service
  - Git information as labels
  - Add healthchecks
  - Add OpenShift routes
  - ....

# PROFILES

- Named collection of enrichers and generators
- -Dfabric8.profile to select

|           |                                     |
|-----------|-------------------------------------|
| raw       | No enrichment                       |
| explicit  | Only default objects                |
| minimal   | Small enrichments                   |
| aggregate | Combine resources from dependencies |

# 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 redeployments   |
| <code>fabric8:debug</code>         | Debug into pods                       |

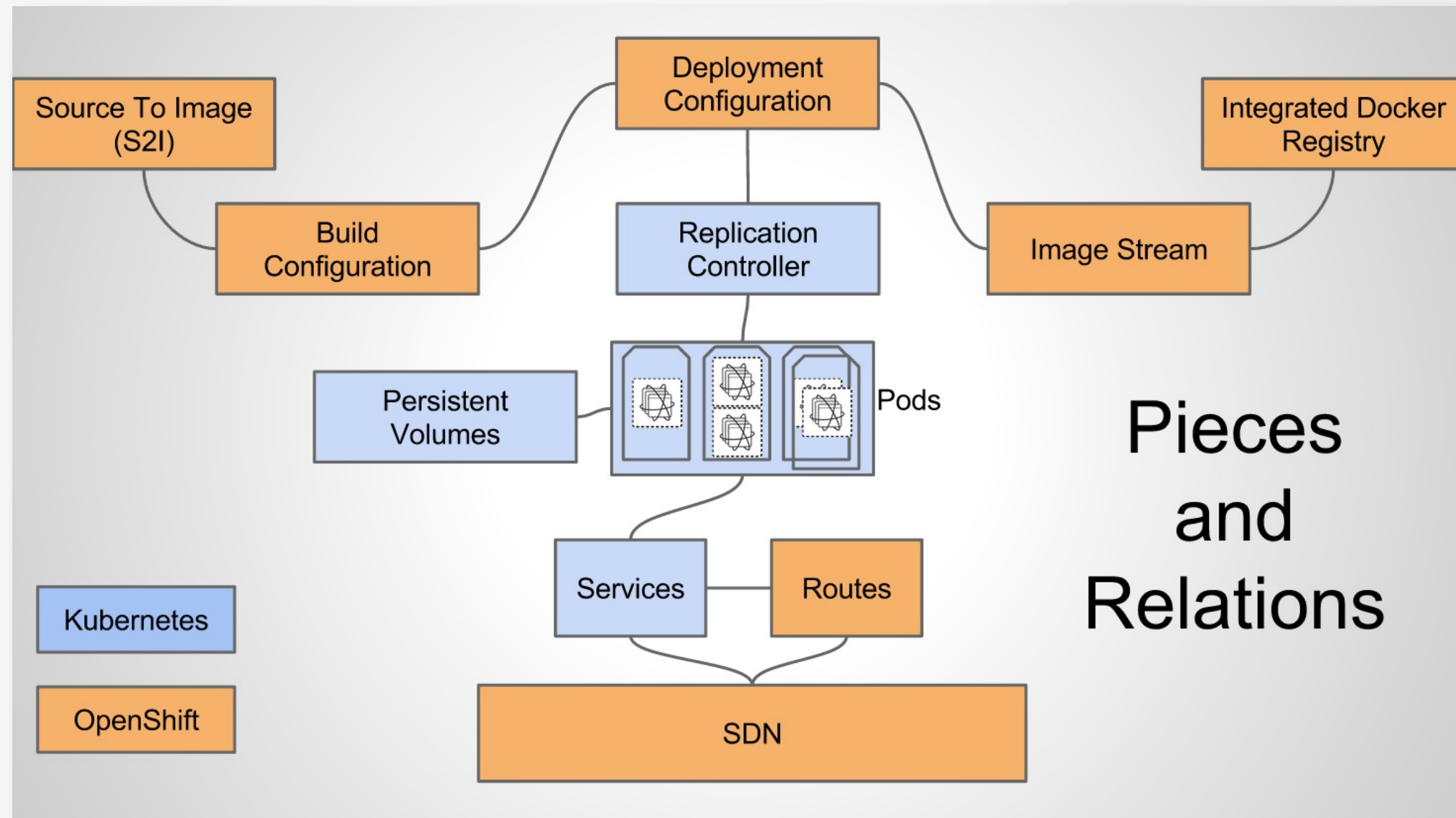


# OPENSHIFT

# OPENSSHIFT

- Adds the **BUILD** to Kubernetes
- Infrastructure Services
  - Registry
  - Router
  - OAuth2 SSO
- 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 ?

Twitter @ro14nd

---

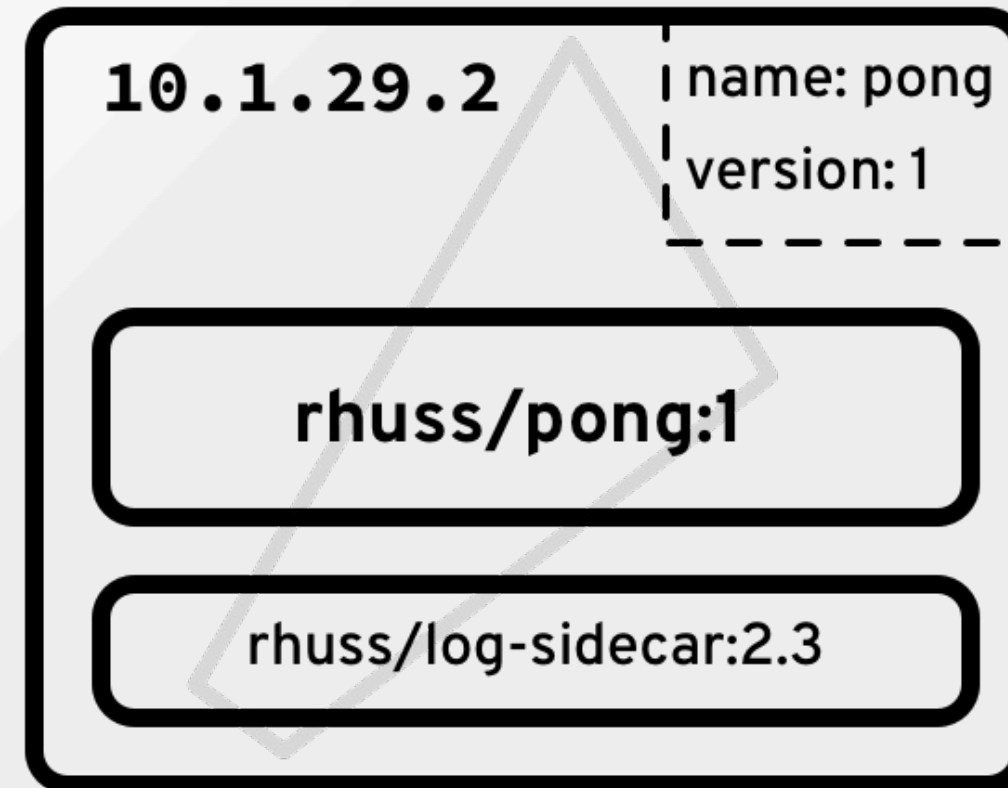
Slides <https://github.com/ro14nd-talks/kubernetes-for-java-developers>





# 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"

# 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**

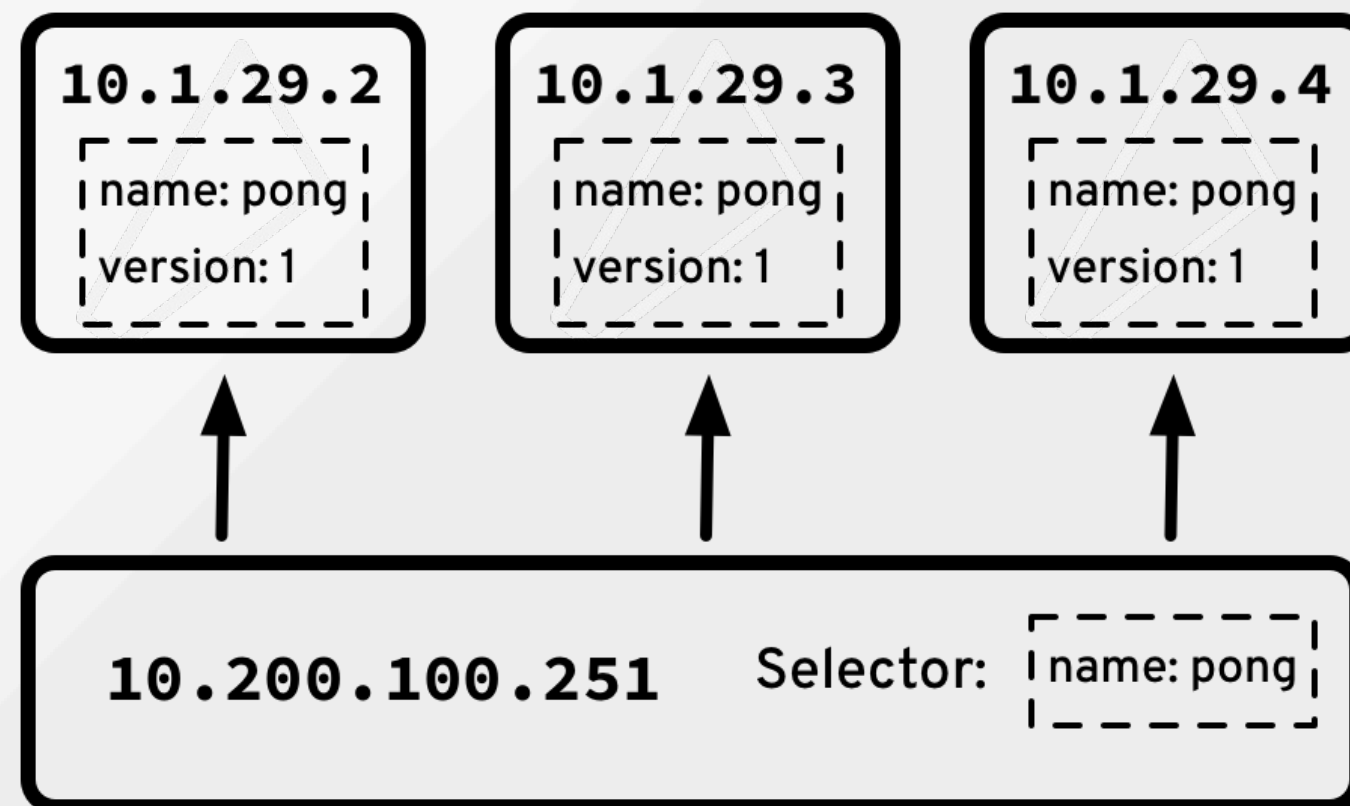
name: pong  
version: 1

**10.1.29.4**

name: pong  
version: 1

# SERVICE

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



# VOLUMES

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

# MISC FEATURES

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