



# Docker

---

Lass mal containern

Julian "mino"

GPN15 - 04.06.2015



## Interessen:

- Netzwerke
- Hardware
- Cocktails
- Hacking
- Musik- und Lichttechnik



Karlsruhe



[gpn15@lab10.de](mailto:gpn15@lab10.de)



[twitter.com/julianklinck](https://twitter.com/julianklinck)



## Docker:

- Beruflich seit 2014
- SDN Cloud



Karlsruhe



[jklinck@ocedo.com](mailto:jklinck@ocedo.com)



1

## Software Deployment

2

## VM vs. Container

3

## Basics

Dockerfile, Docker Daemon,  
Docker Hub

4

## Container Format

Layers

5

## Tools

Compose, Swarm, Kitematic

6

## Sicherheit

Angriffsvektoren

7

## Zukunft

LXD, Rocket, ClearLinux

8

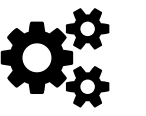
## Q & A



# Software Deployment

---

Vergangenheit und Jetzt

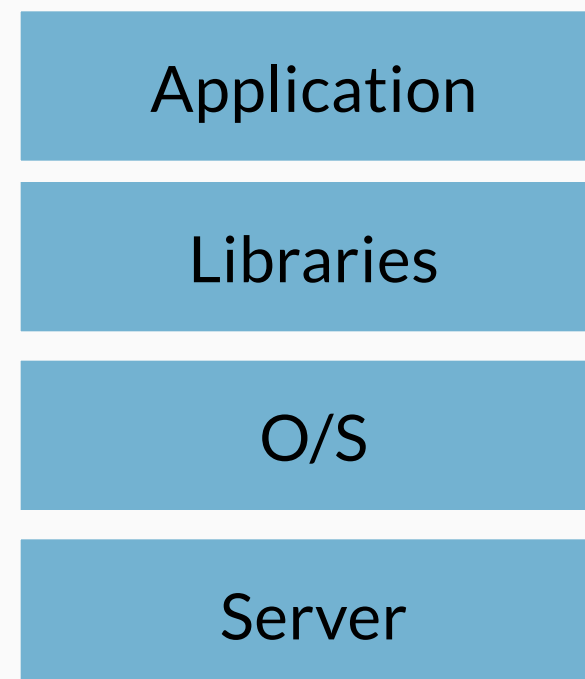


# Klassisch

Hast du ein Backup davor gemacht?

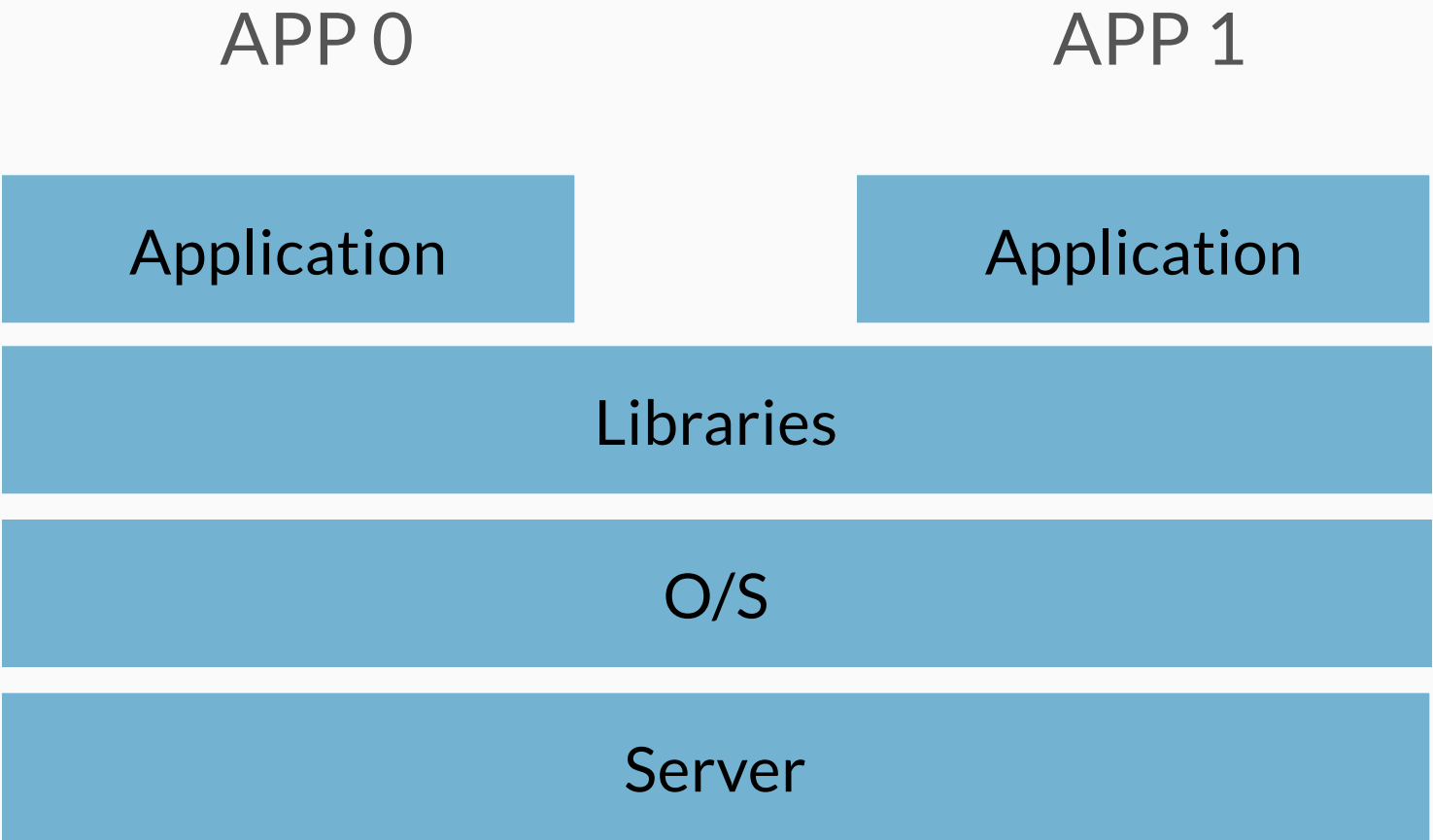


## Monolithisch



# Klassisch

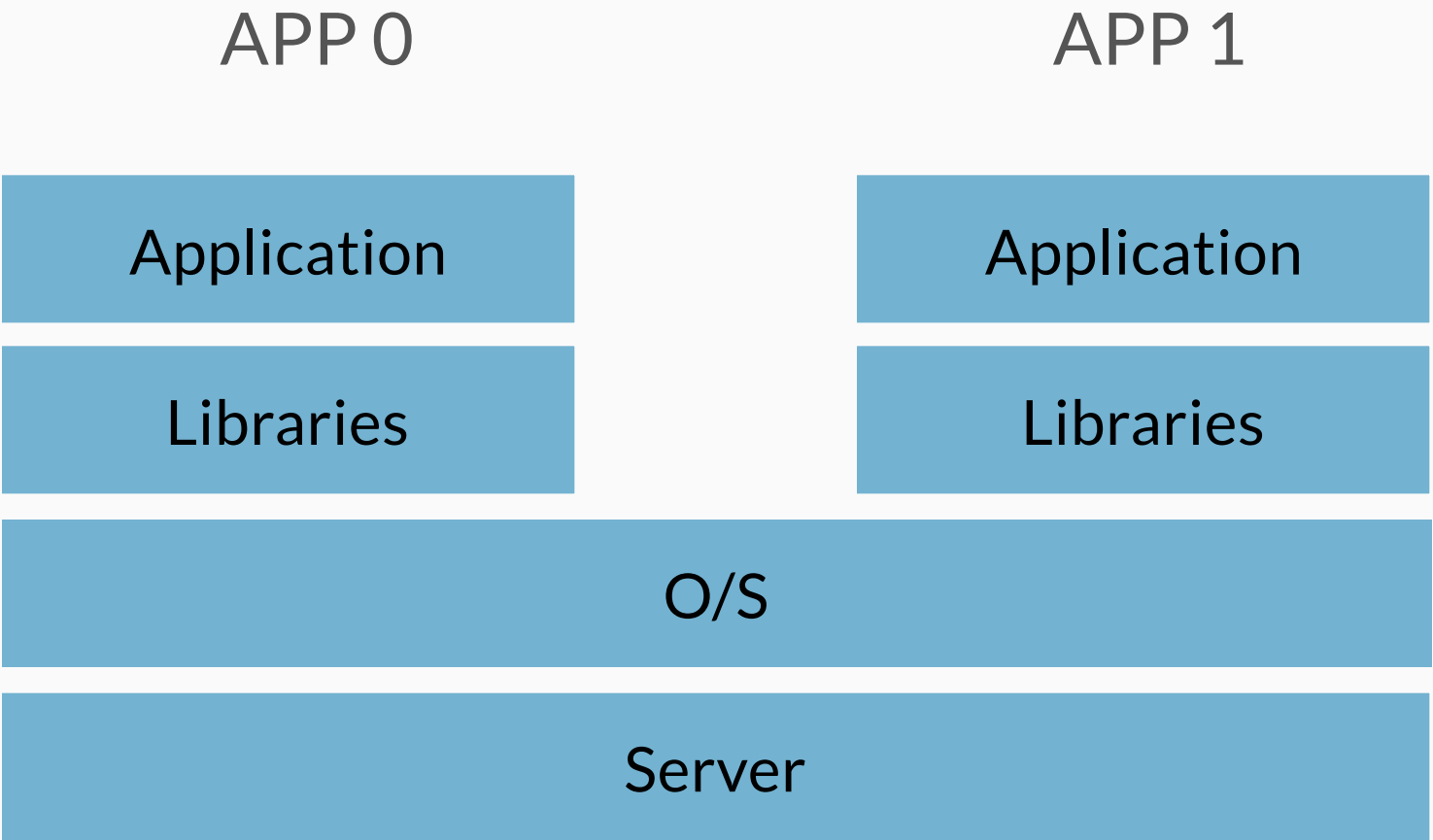
Wir brauchen da noch...





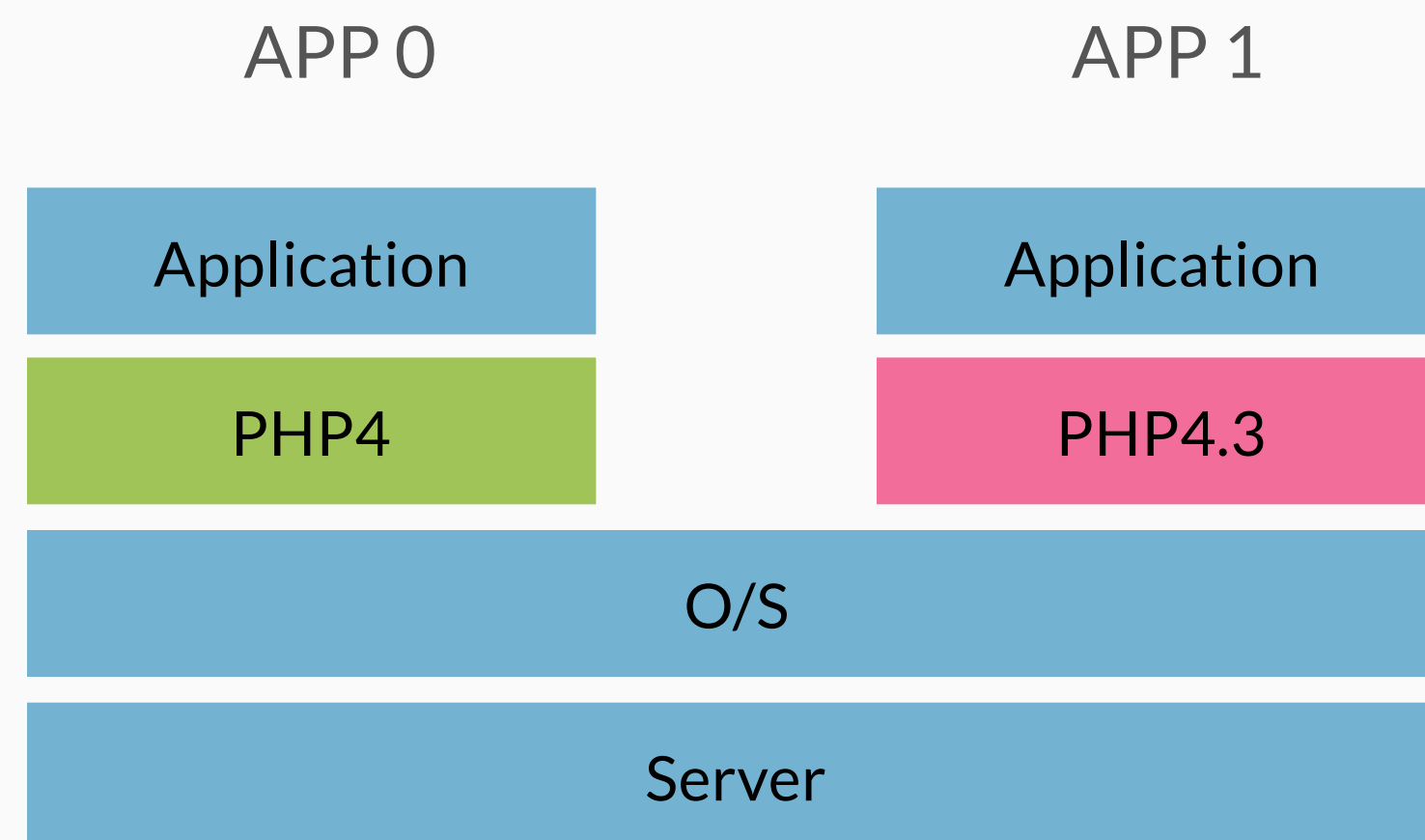
# Klassisch

Wir brauchen da noch...



# Klassisch

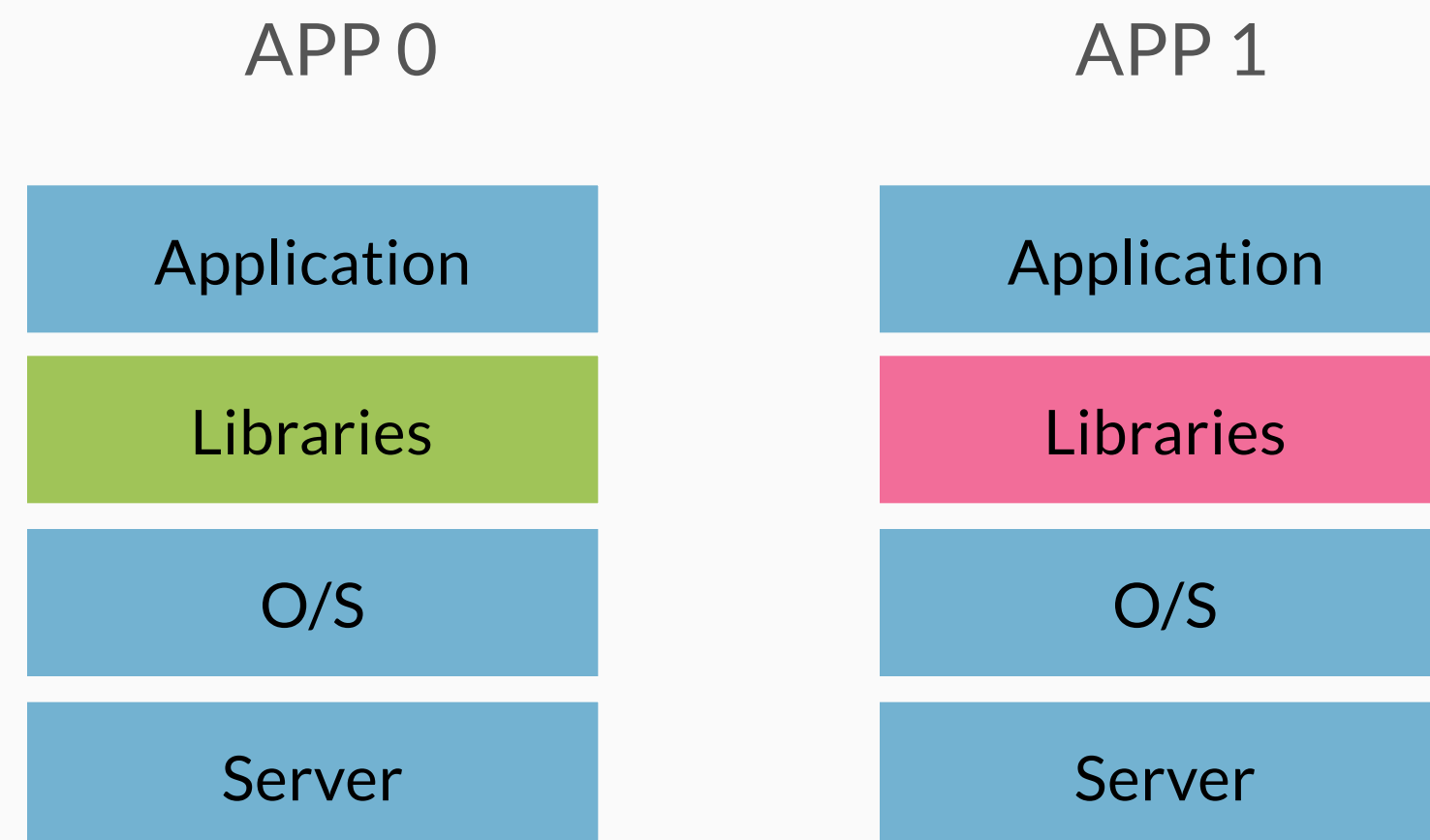
Wir müssen bei PHP4.0 bleiben!



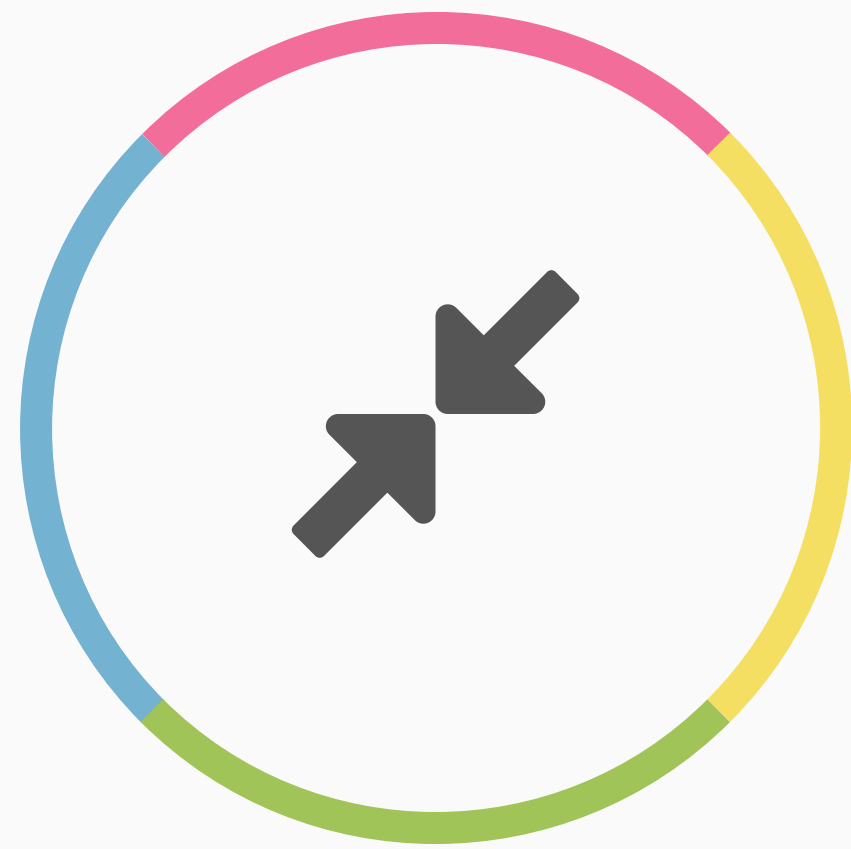
Dependency Hell



## Monolithisch



- Teuer
- Verschwenderisch
- Schwer zu managen

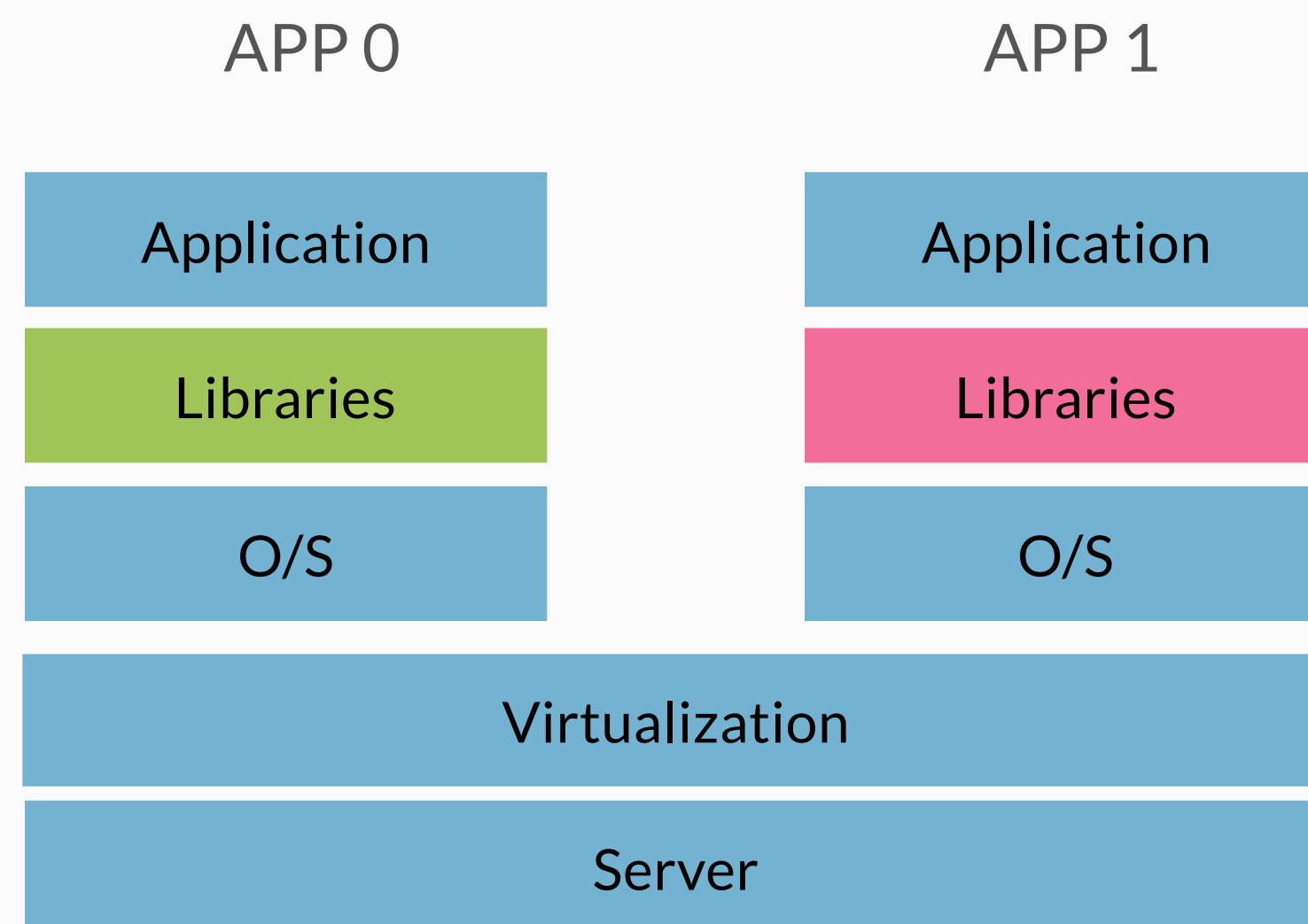


# VM vs. Container

---



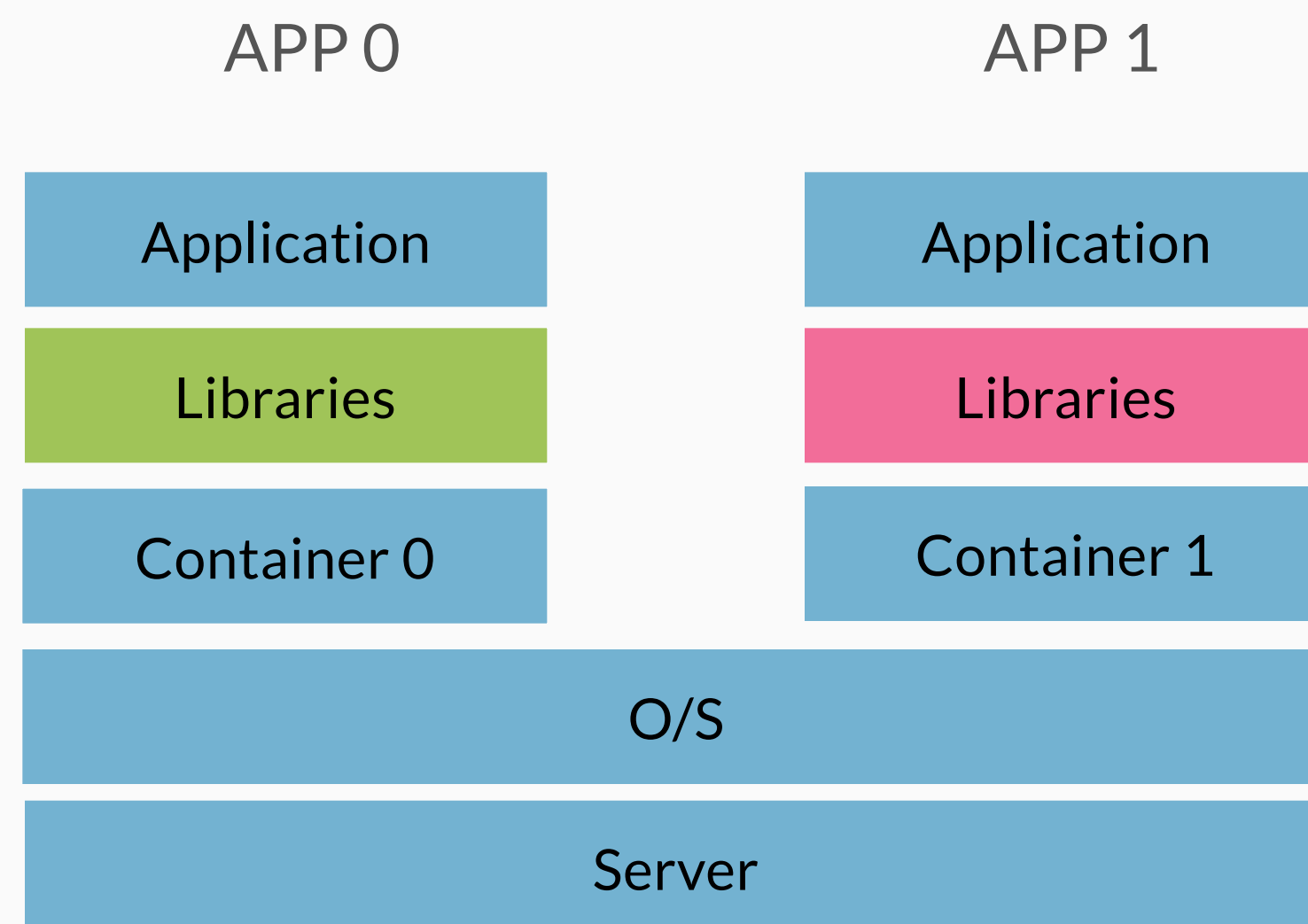
## Monolithisch



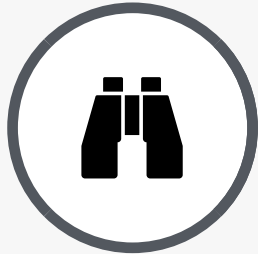
- Komplexität
- Overhead



## Monolithisch

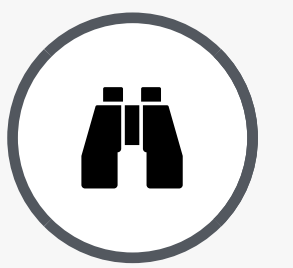


- Komplexität



APP 0

Applikation



APP 0

Webshop Code

Webserver

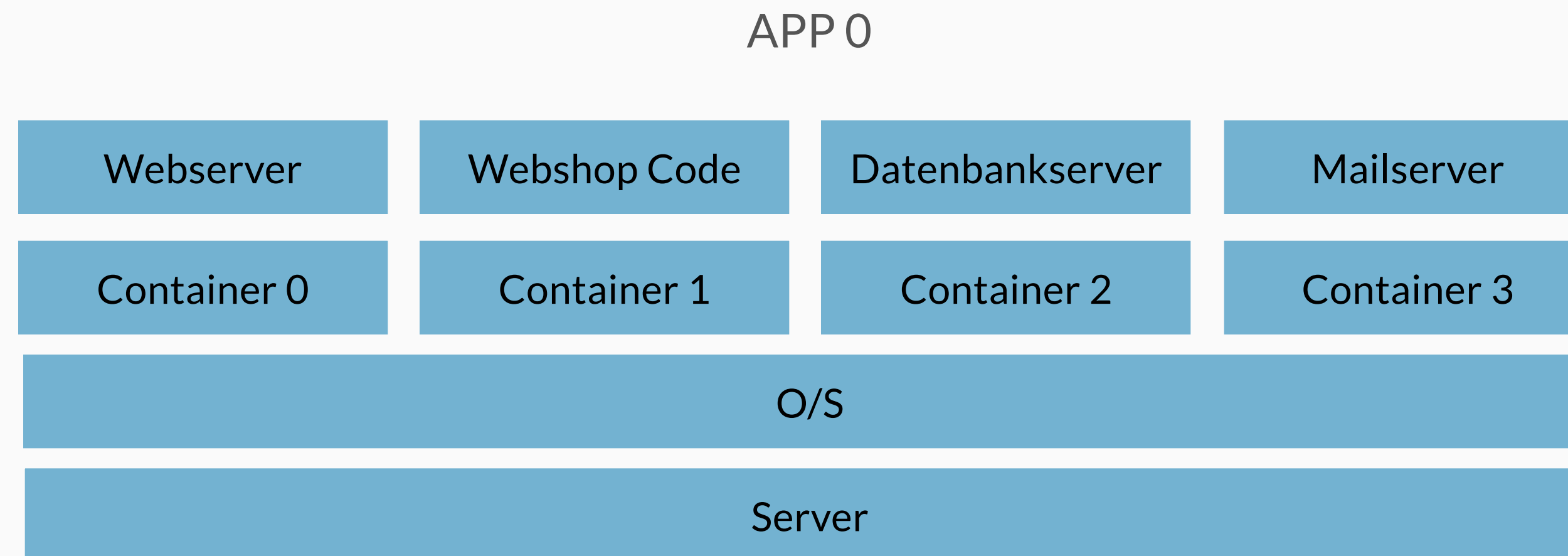
Datenbankserver

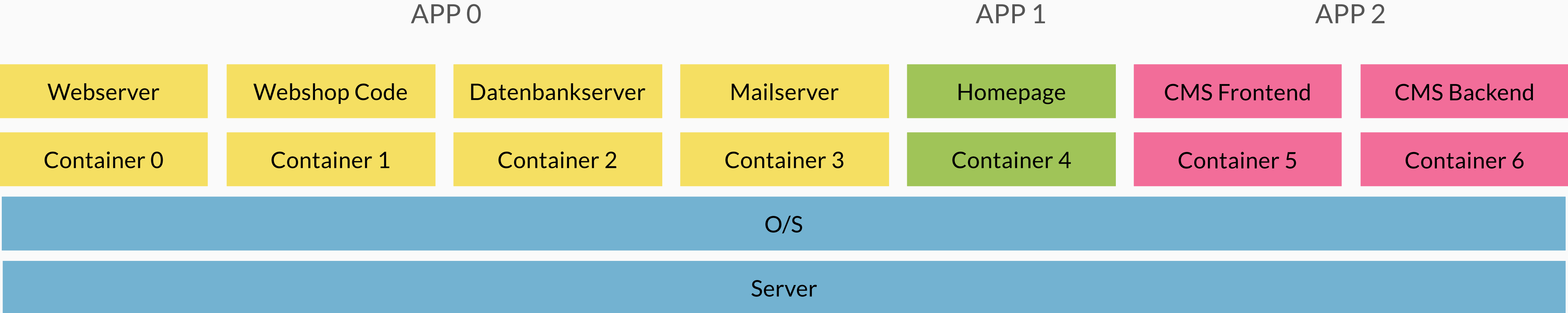
Mailserver

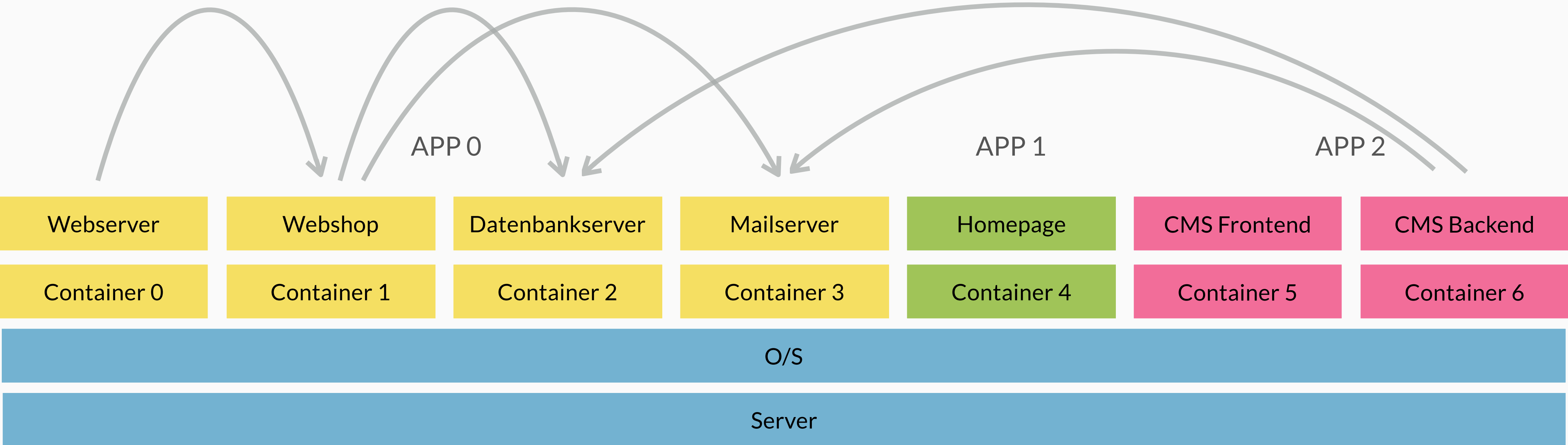




“One process per container”





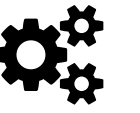




# Basics

---

Dockerfile, Docker Daemon, Docker Hub



Offene Plattform für:

- Entwickler
- Admins

“Deploy everything nearly everywhere reliably and consistently”

Deploy everything:

- Webapps
- Backends
- Databases
- Message Queues
- Proxys
- ...

Deploy everywhere:

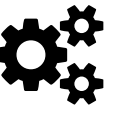
- Linux Server
- VMs or Bare-Metal
- Any distro
- Kernel 3.8+
- x86\_64

# Dockerfile

Die Baunaleitung



```
1 FROM fedora:20
2 MAINTAINER Mark Lamourine <markllama@gmail.com>
3
4 RUN yum install -y mongodb-server && yum clean all
5
6 RUN mkdir -p /var/lib/mongodb && \
7     touch /var/lib/mongodb/.keep && \
8     chown -R mongodb:mongodb /var/lib/mongodb
9
10 ADD mongodb.conf /etc/mongodb.conf
11
12 VOLUME [ "/var/lib/mongodb" ]
13
14 EXPOSE 27017
15
16 USER mongodb
17 WORKDIR /var/lib/mongodb
18
19 ENTRYPOINT ["/usr/bin/mongod", "--config", "/etc/mongodb.conf"]
20 CMD ["--quiet"]
```



## Download a pre-built image

```
# Download an ubuntu image
$ sudo docker pull ubuntu
```

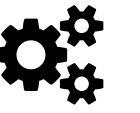
## Starting a long-running worker process

```
# Start a very useful long-running process
$ JOB=$(sudo docker run -d ubuntu /bin/sh -c "while true; do echo Hello world; sleep 1; done")

# Collect the output of the job so far
$ sudo docker logs $JOB

# Kill the job
$ sudo docker kill $JOB
```

Quelle: \_\_\_\_\_



## Controlling containers

```
# Start a new container
$ JOB=$(sudo docker run -d ubuntu /bin/sh -c "while true; do echo Hello world; sleep 1; done")

# Stop the container
$ sudo docker stop $JOB

# Start the container
$ sudo docker start $JOB

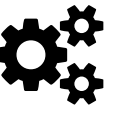
# Restart the container
$ sudo docker restart $JOB

# SIGKILL a container
$ sudo docker kill $JOB

# Remove a container
$ sudo docker stop $JOB # Container must be stopped to remove it
$ sudo docker rm $JOB
```

Quelle: \_\_\_\_\_



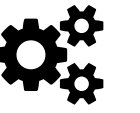


## Docker Daemon

- Root Prozess
- Docker API

## Docker Hub

- Webservice
- Image Speicher
- Public Registry/Repository



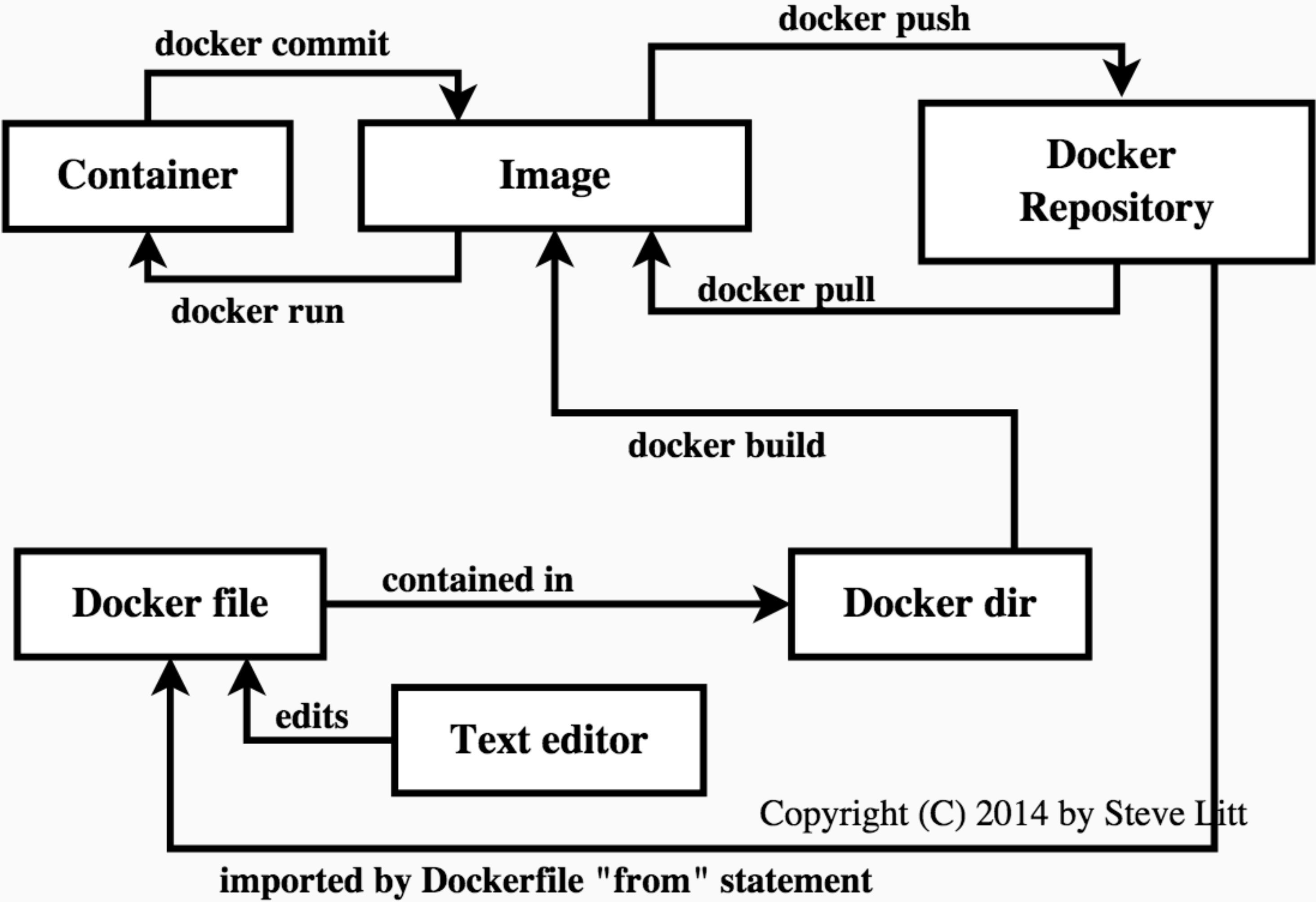
## Image

- Name
- Dockerfile step-by-step
- Applikations Abbild

## Container

- Name
- Image als Basis
- Runtime

# Docker Prozess



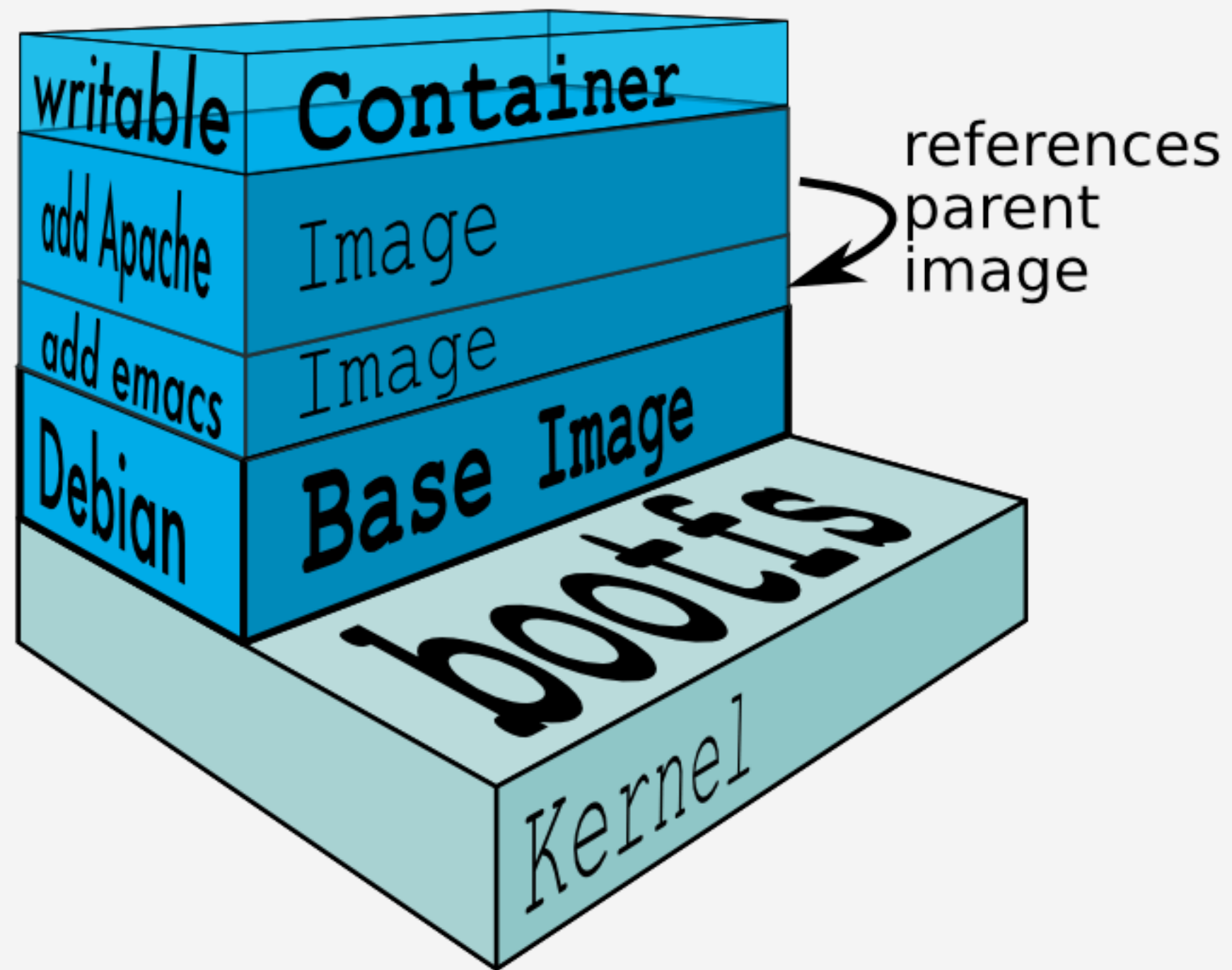
Quelle: \_\_\_\_\_



# Container Format

---

Layers

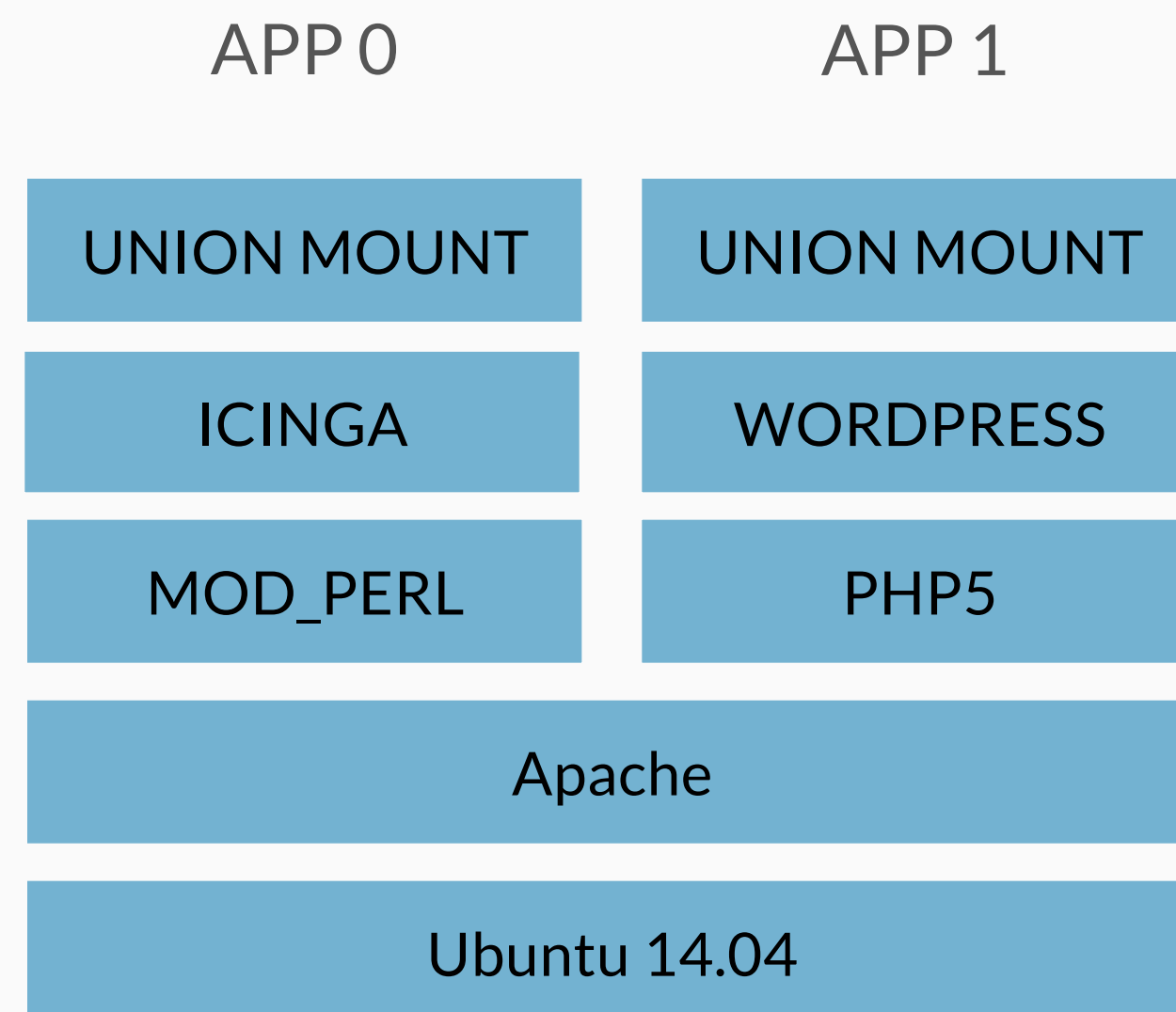


- Sub-Layers
- Top-Layer Union Mount
- Copy-On-Write

Quelle: \_\_\_\_\_

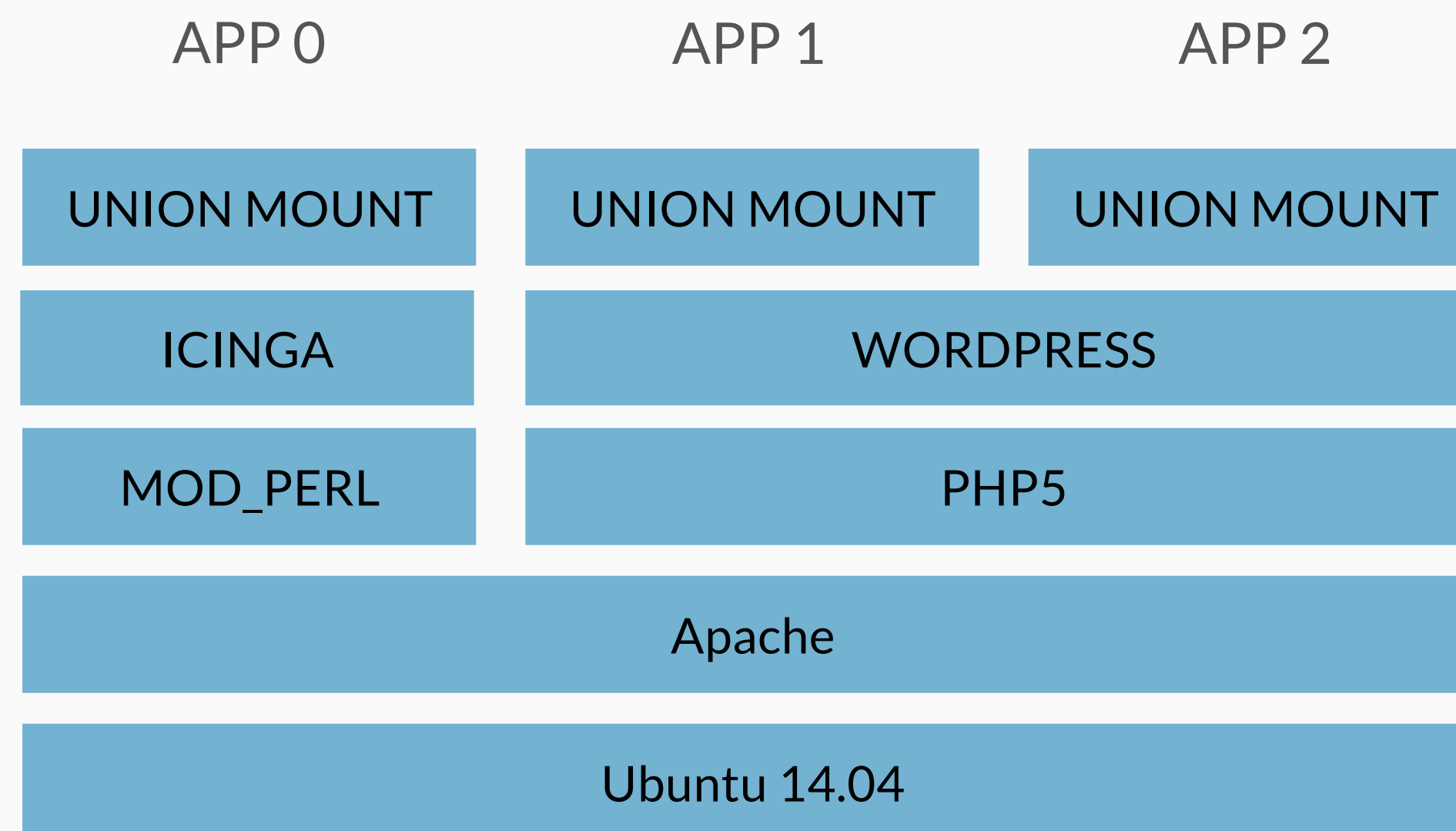


- Layer können wiederverbenutzt werden



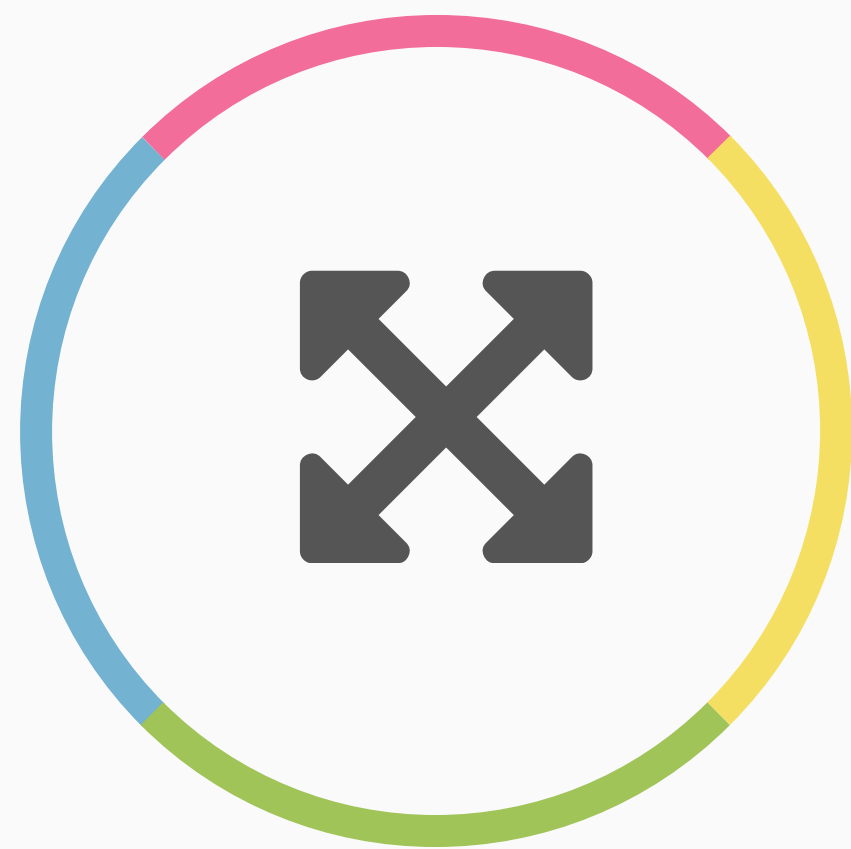
Quelle: \_\_\_\_\_

- Layer können wiederverbenutzt werden



- Speichereffizienz

Quelle: \_\_\_\_\_

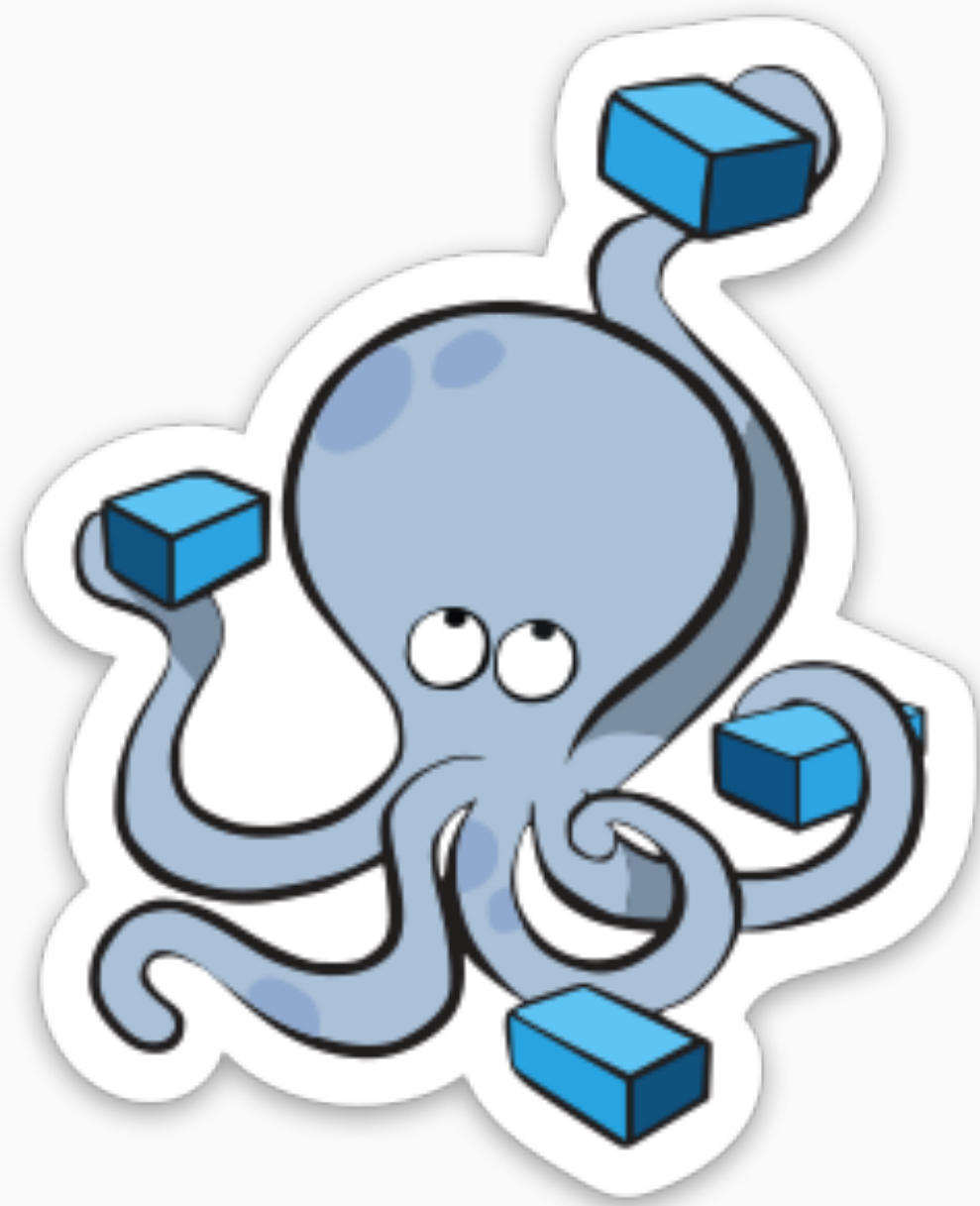


# Tools

---

Compose, Swarm, Kitematic





## Applikationen

- Multiple Container
- Abhängigkeiten

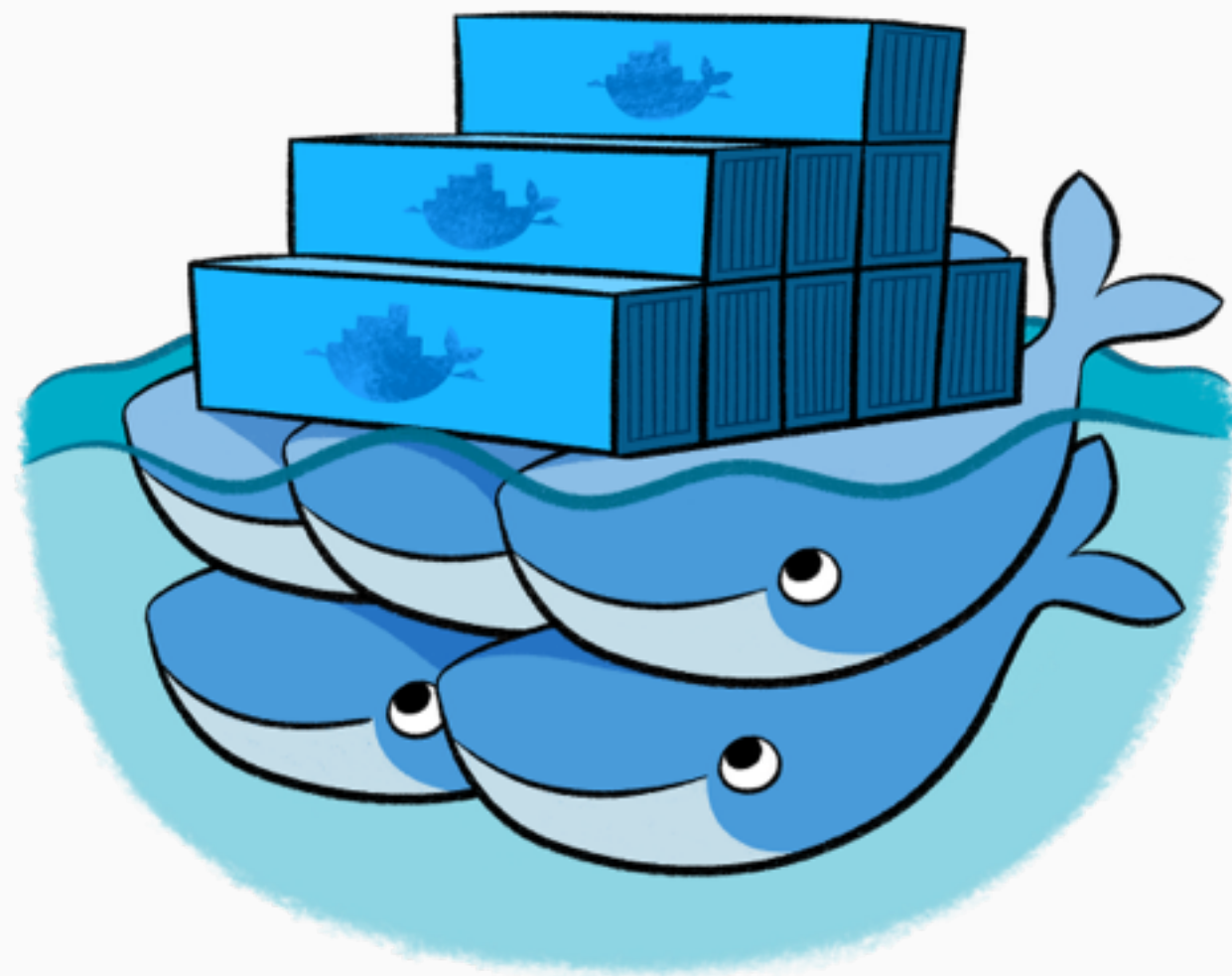
## Verwaltungswerkzeug

- “docker-compose.yml”
- Start, Stop, Rebuild...

Quelle: \_\_\_\_\_

# Swarm

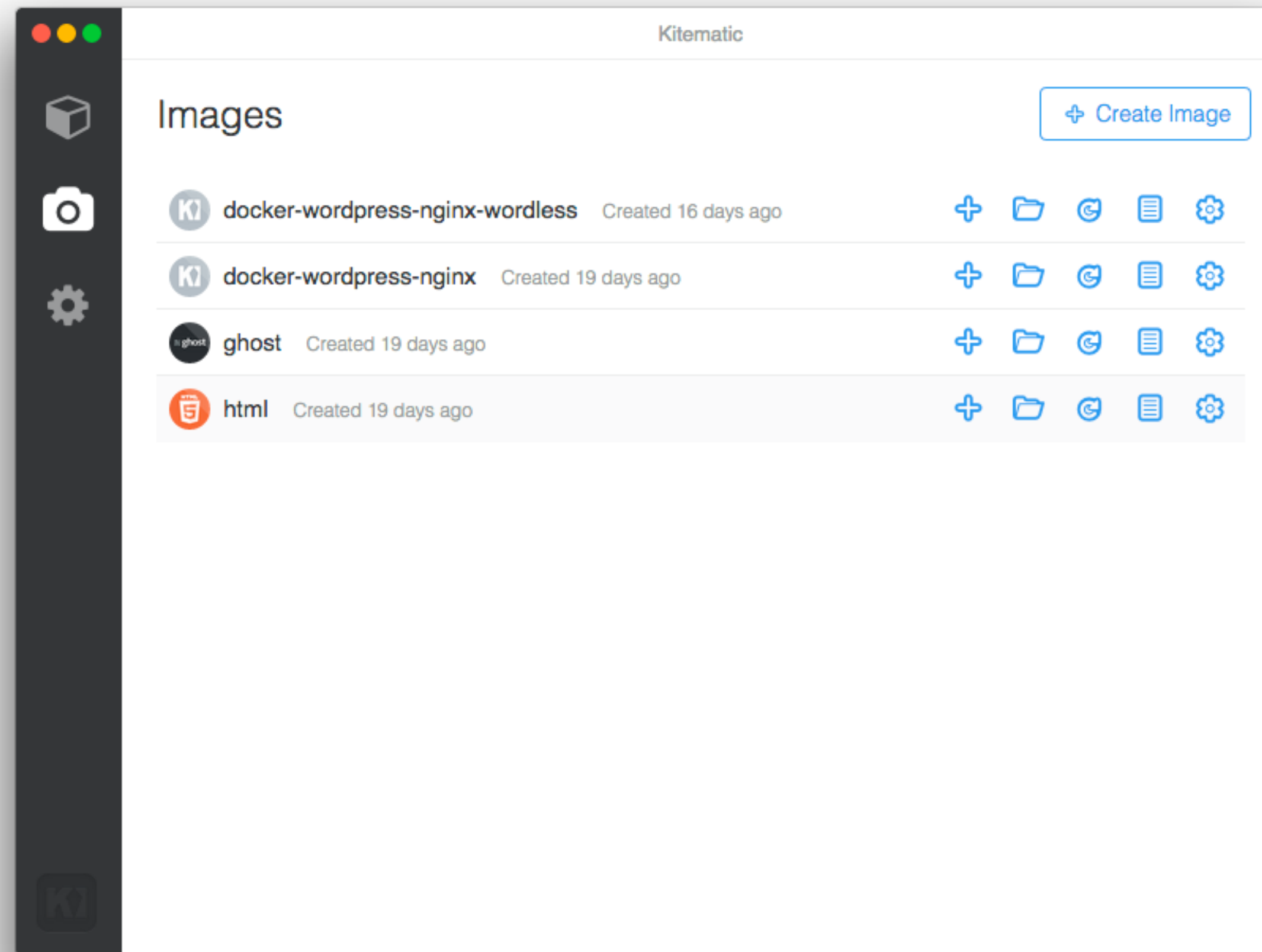
Docker Daemon Clustering



## Cluster

- Multiple Docker Daemons
- Verwaltung
- Docker API kompatibel

Quelle: \_\_\_\_\_



## Docker GUI + VM

- Klicki-Bunti
- Installer
  - VirtualBox
  - Minimal Docker OS
- Suche im Docker Hub

Quelle: \_\_\_\_\_



# Sicherheit

---

Angriffsvektoren



- Docker Daemon
  - Root Rechte
- Docker API
  - Lokaler socket
  - HTTP optional
  - HTTP(s) optional
- Kernel
  - Selinux, AppArmor
- User Fehler
  - Container Capabilities
  - Volume Mounts

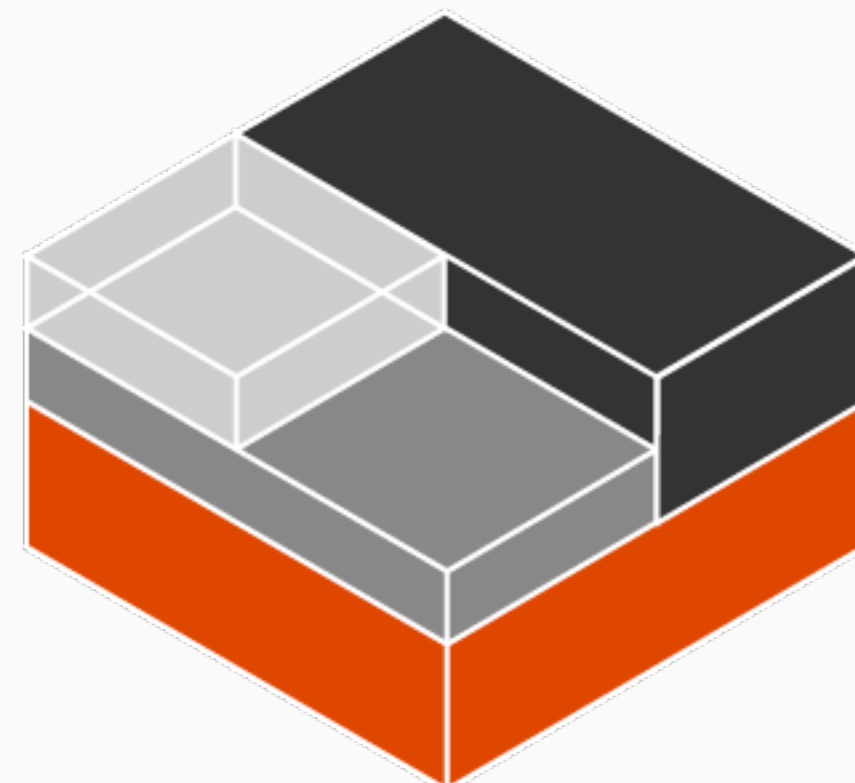


# Zukunft

---

LXD, Rocket, ClearLinux

Docker ist nicht der Weisheit letzter Schluss



LXD







DANKE FÜR DIE AUFMERKSAMKEIT

FRAGEN?



**VIEL SPASS AUF DER GPN15**

