Containers

Guest Lecture

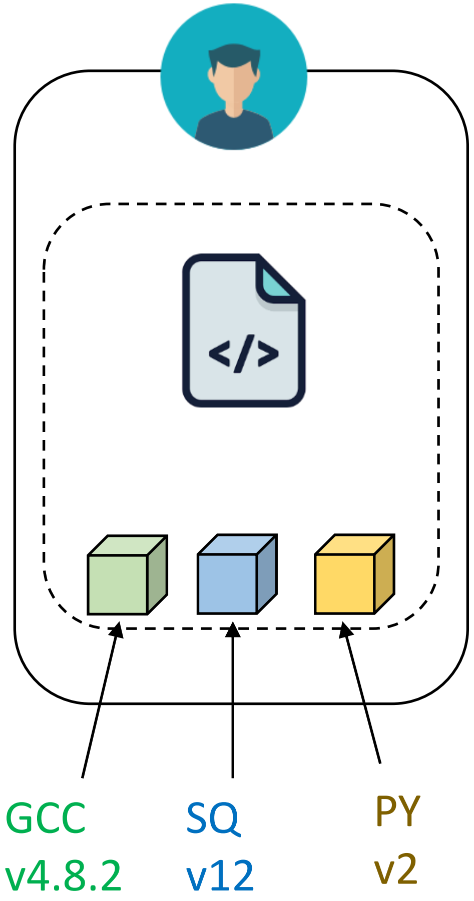
December 8th, 2020

UC San Diego

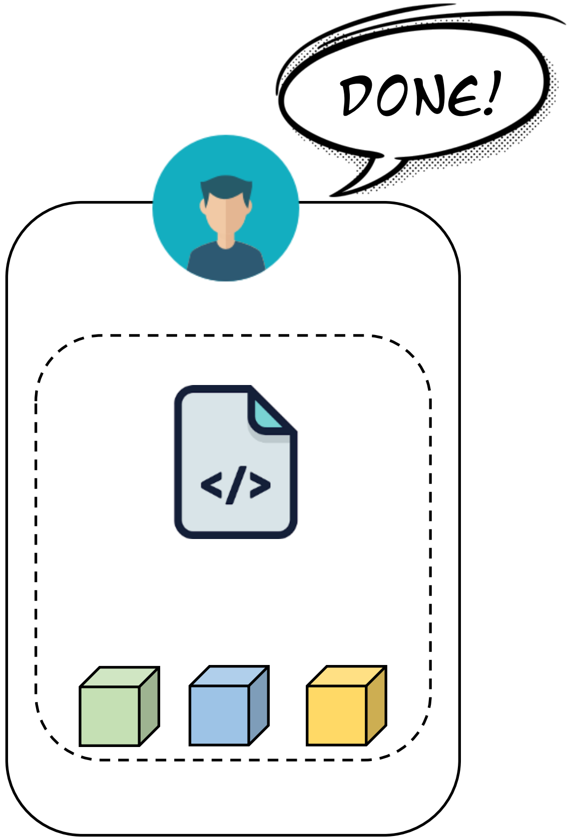
Shelby Thomas Ph.D.

@realshelbyt

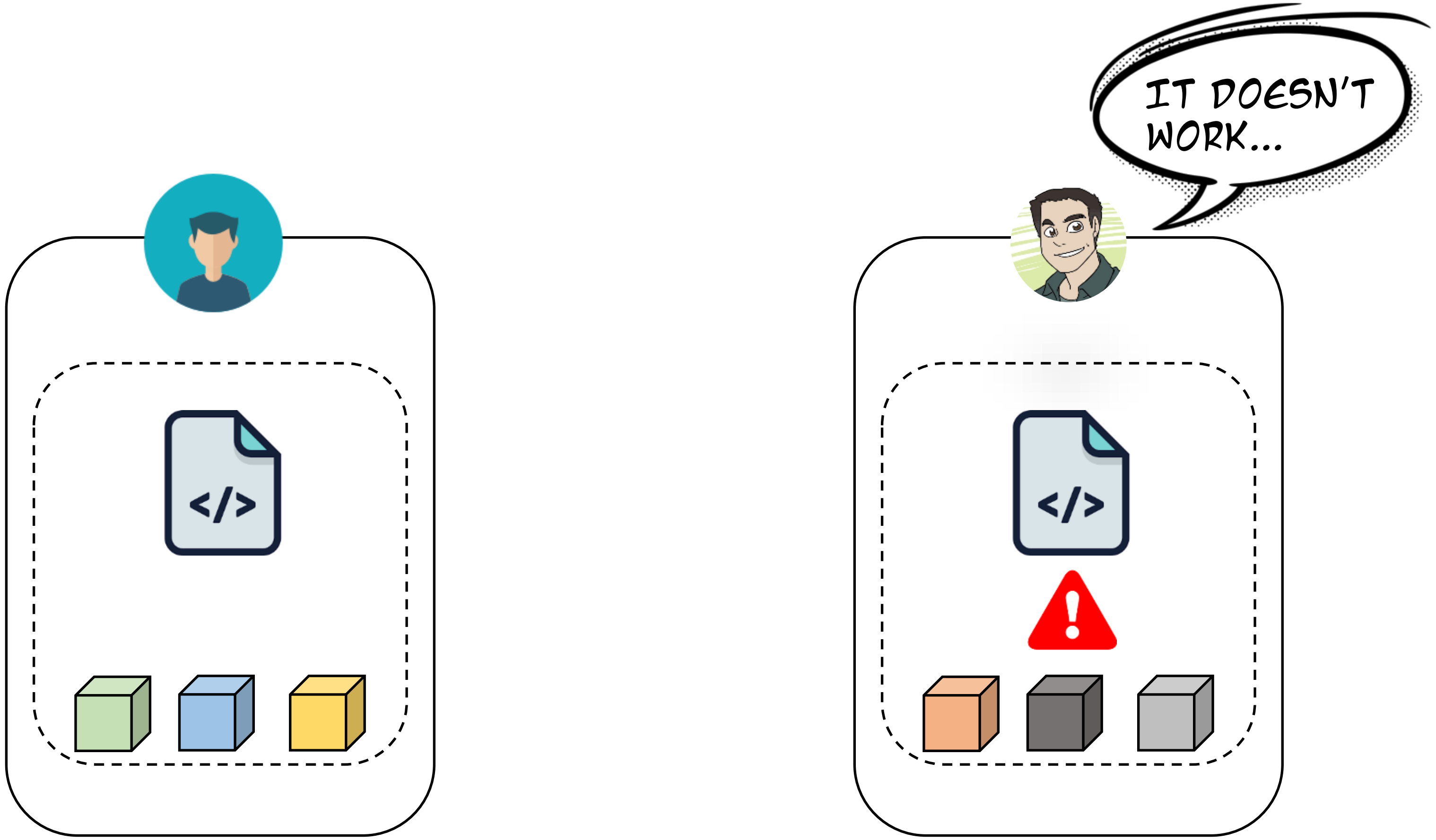
What can containers do?



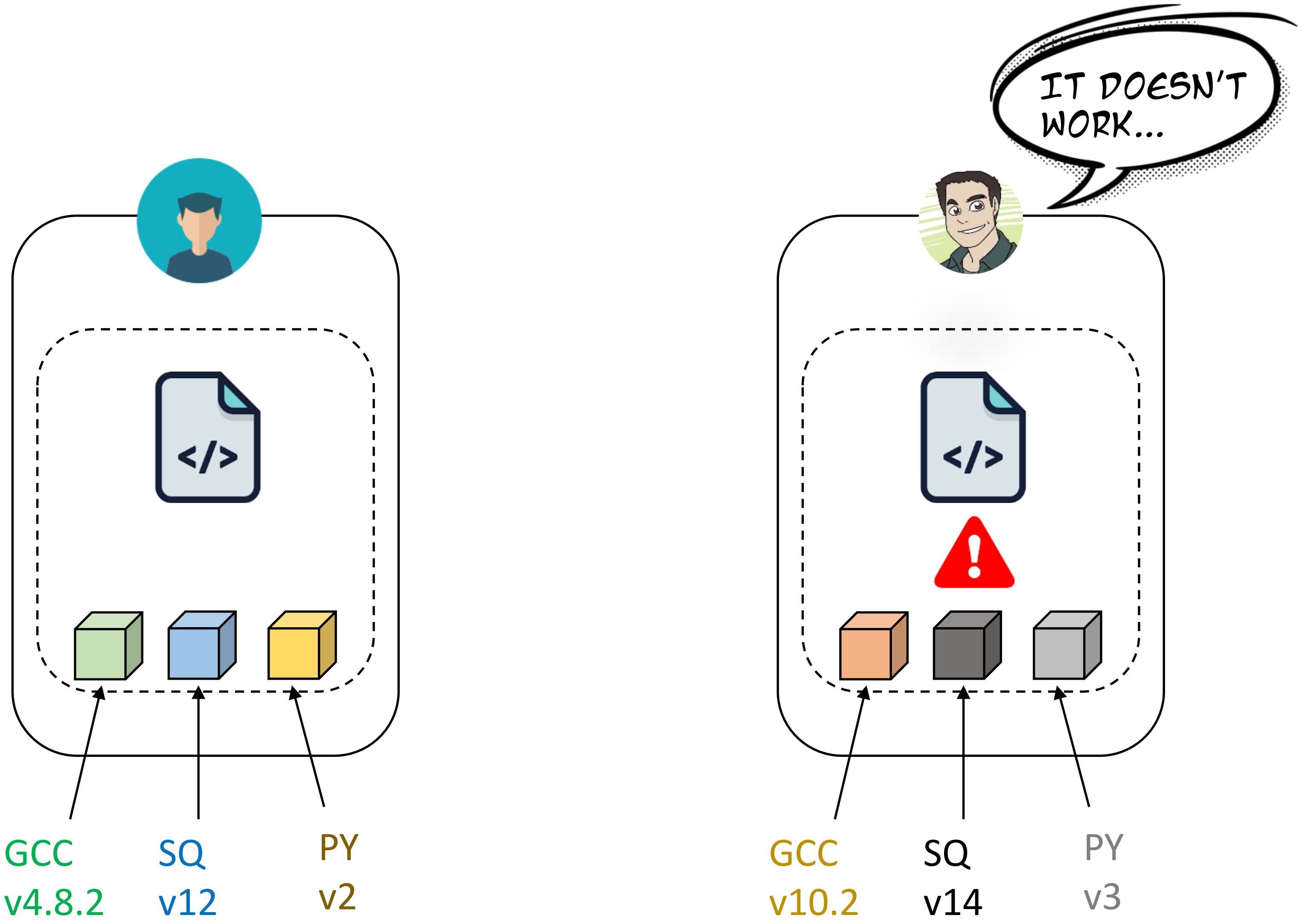
What can containers do?



What can containers do?



What can containers do?



GCC v4.8.2

SQ v12

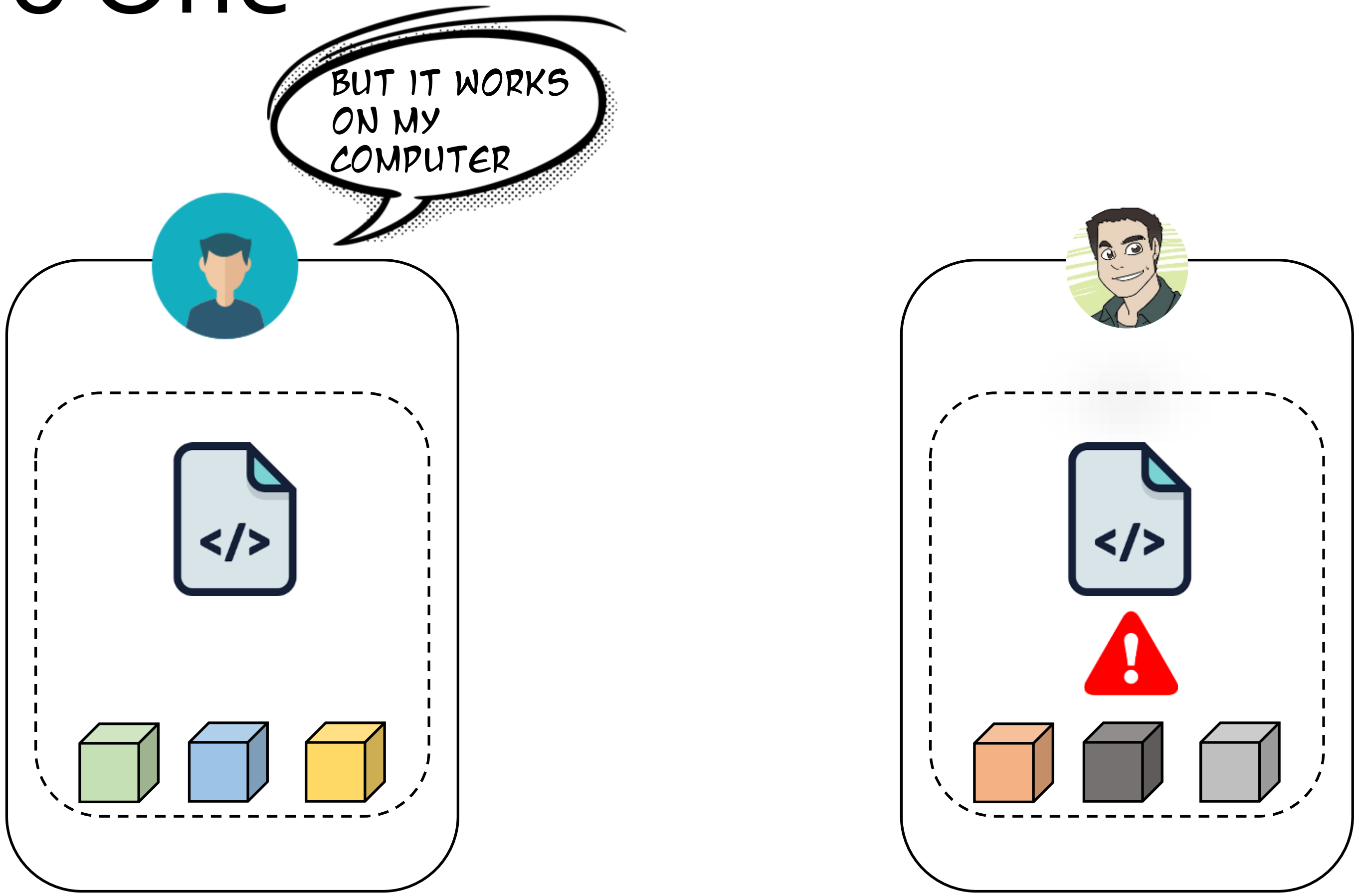
PY v2

GCC v10.2

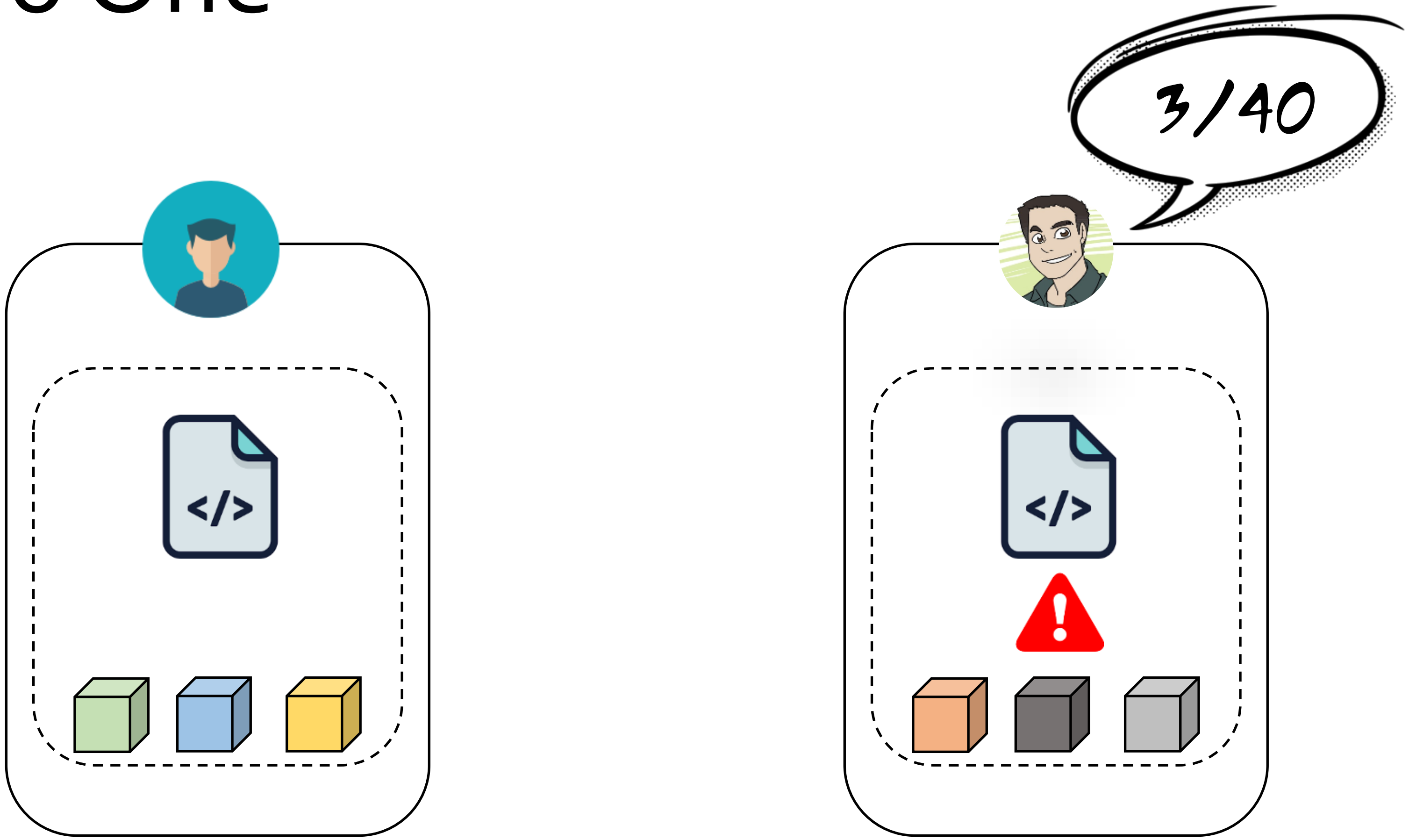
SQ v14

PY v3

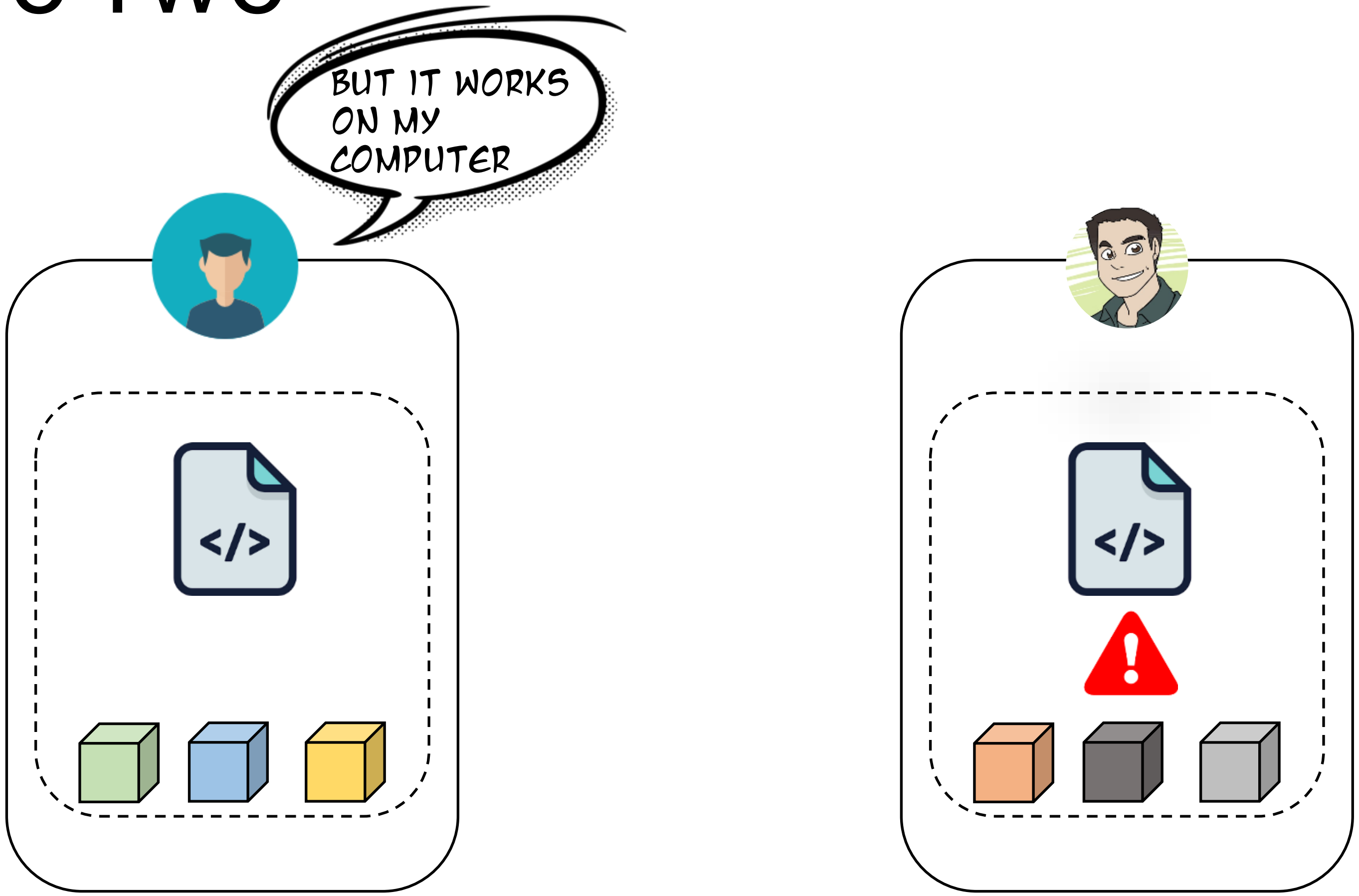
Scenario One



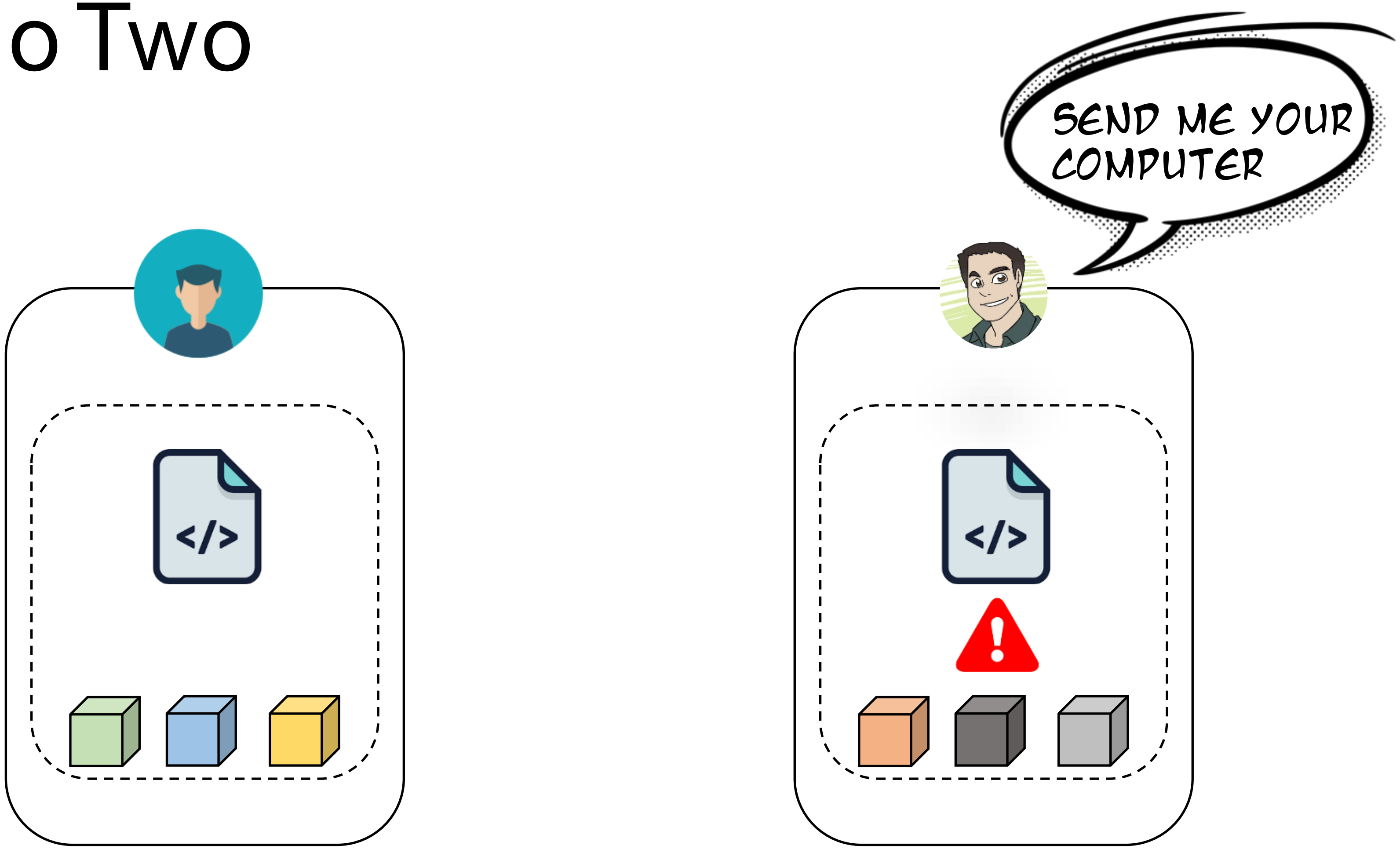
Scenario One



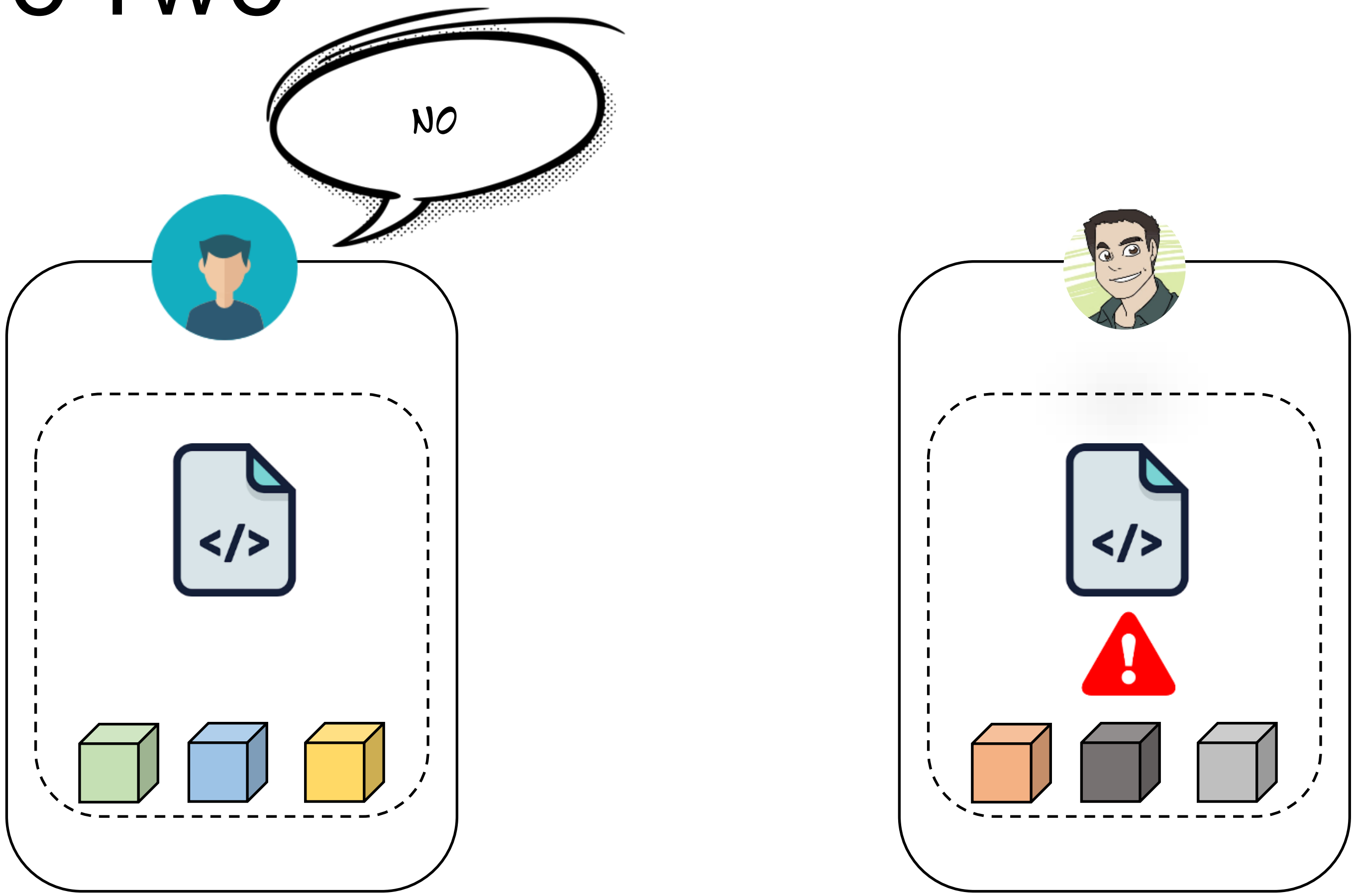
Scenario Two



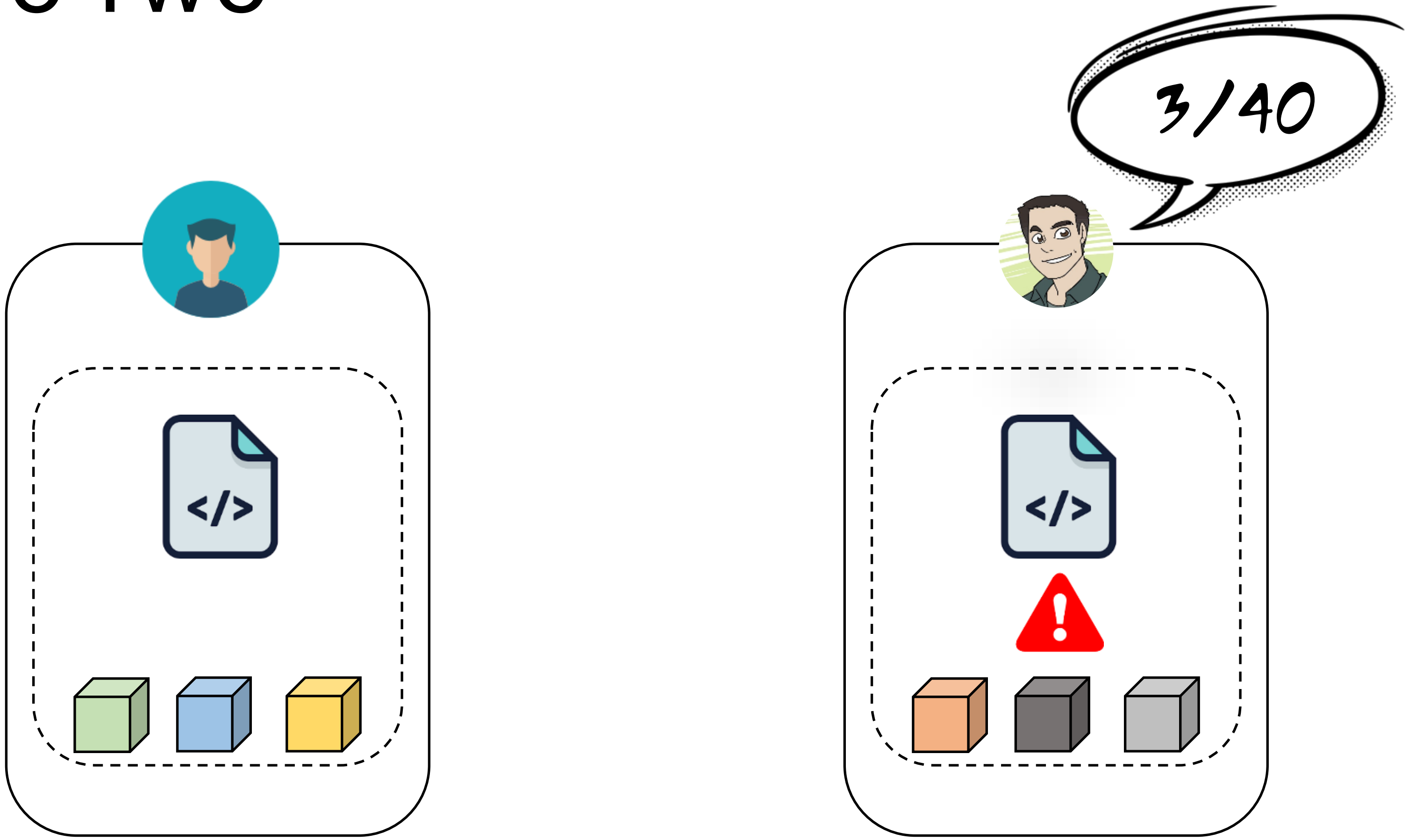
Scenario Two



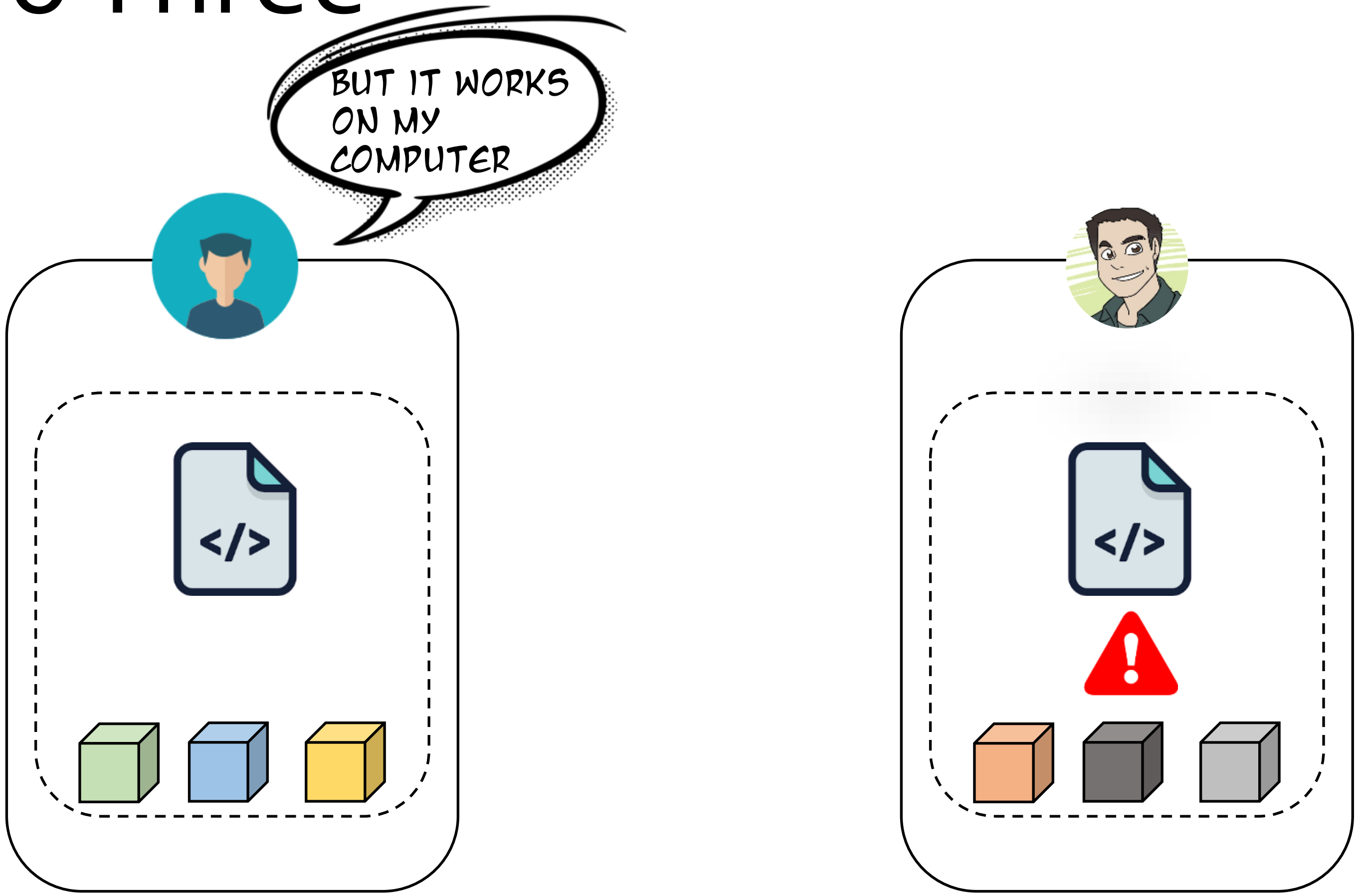
Scenario Two



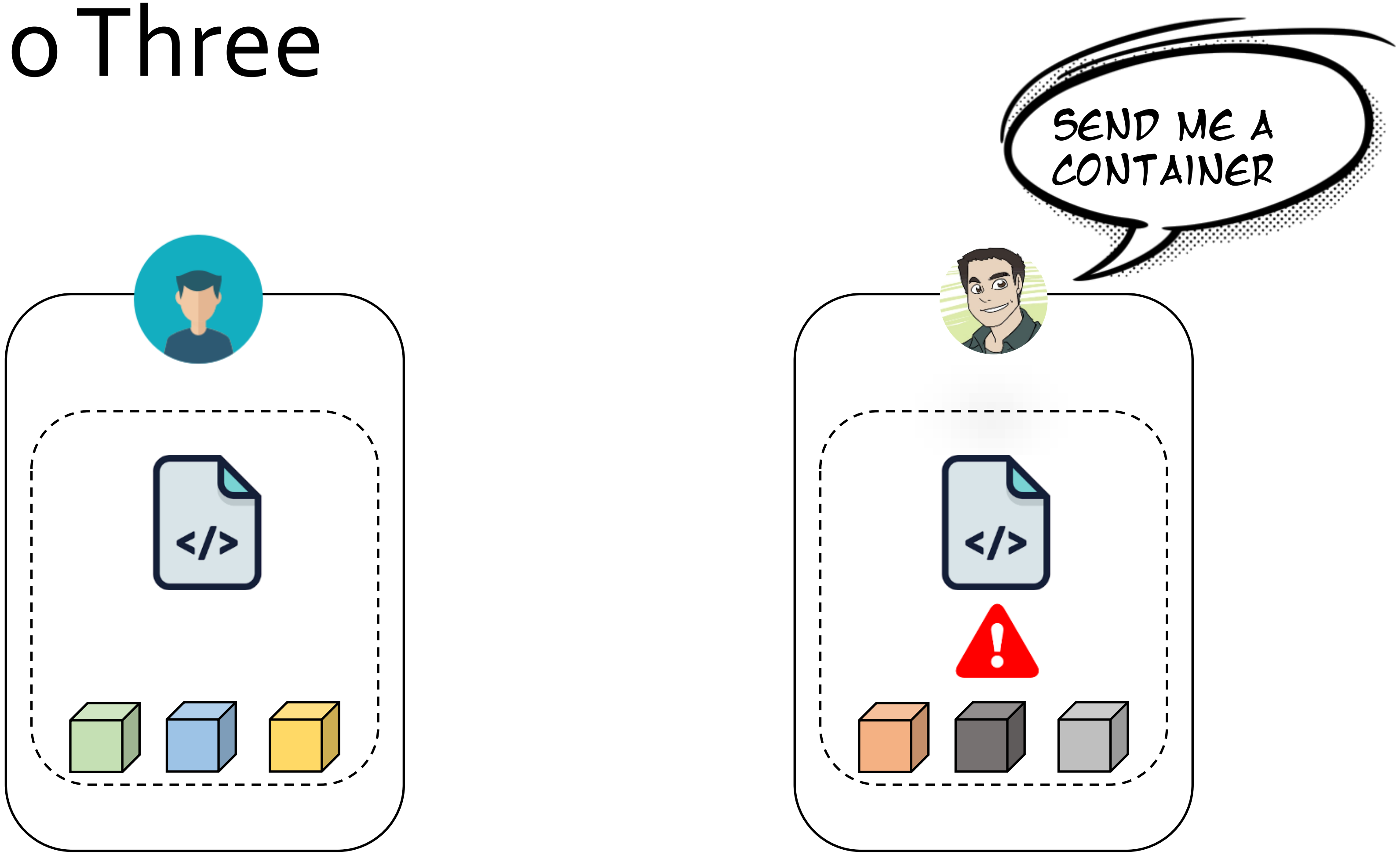
Scenario Two



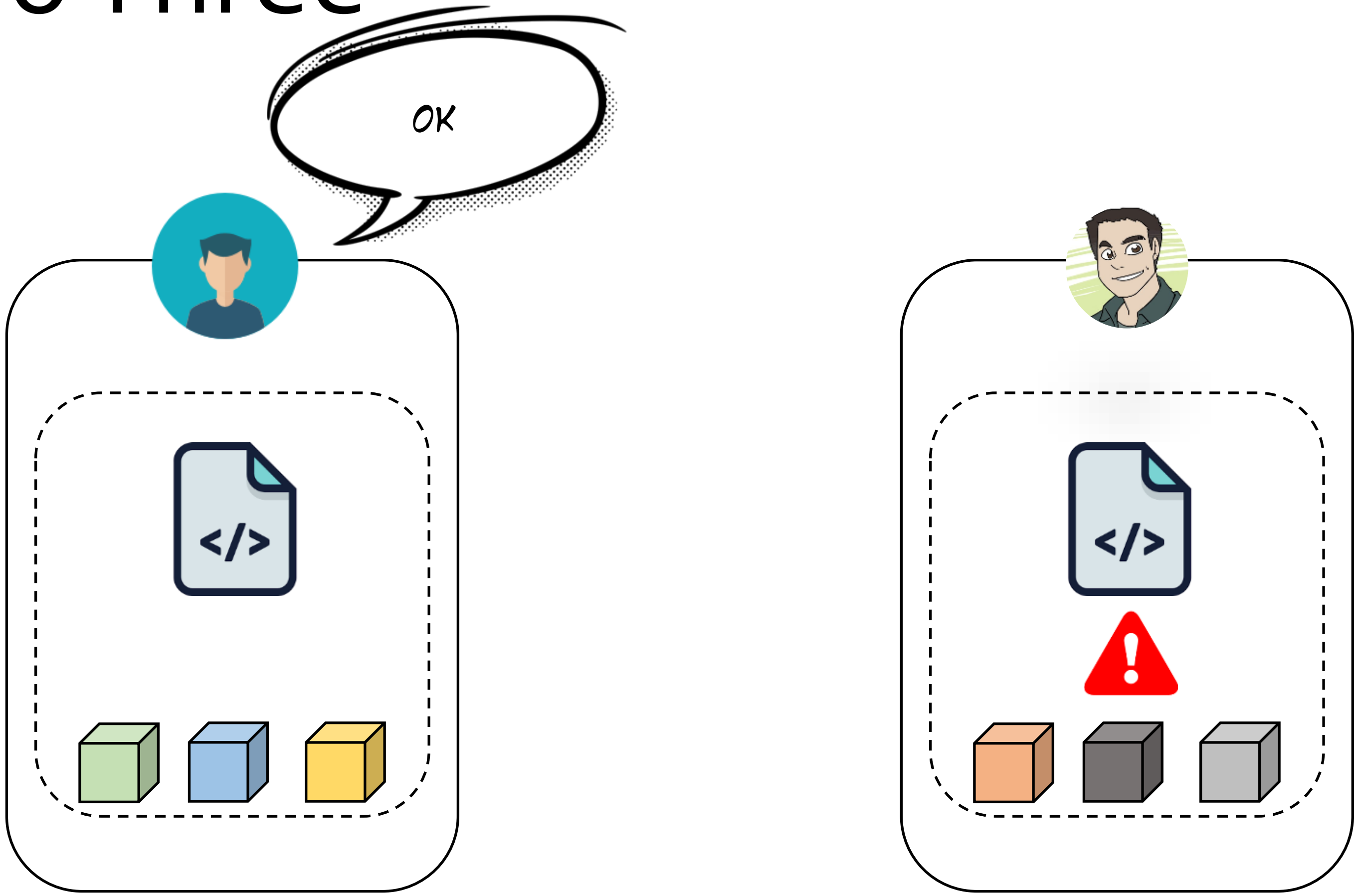
Scenario Three



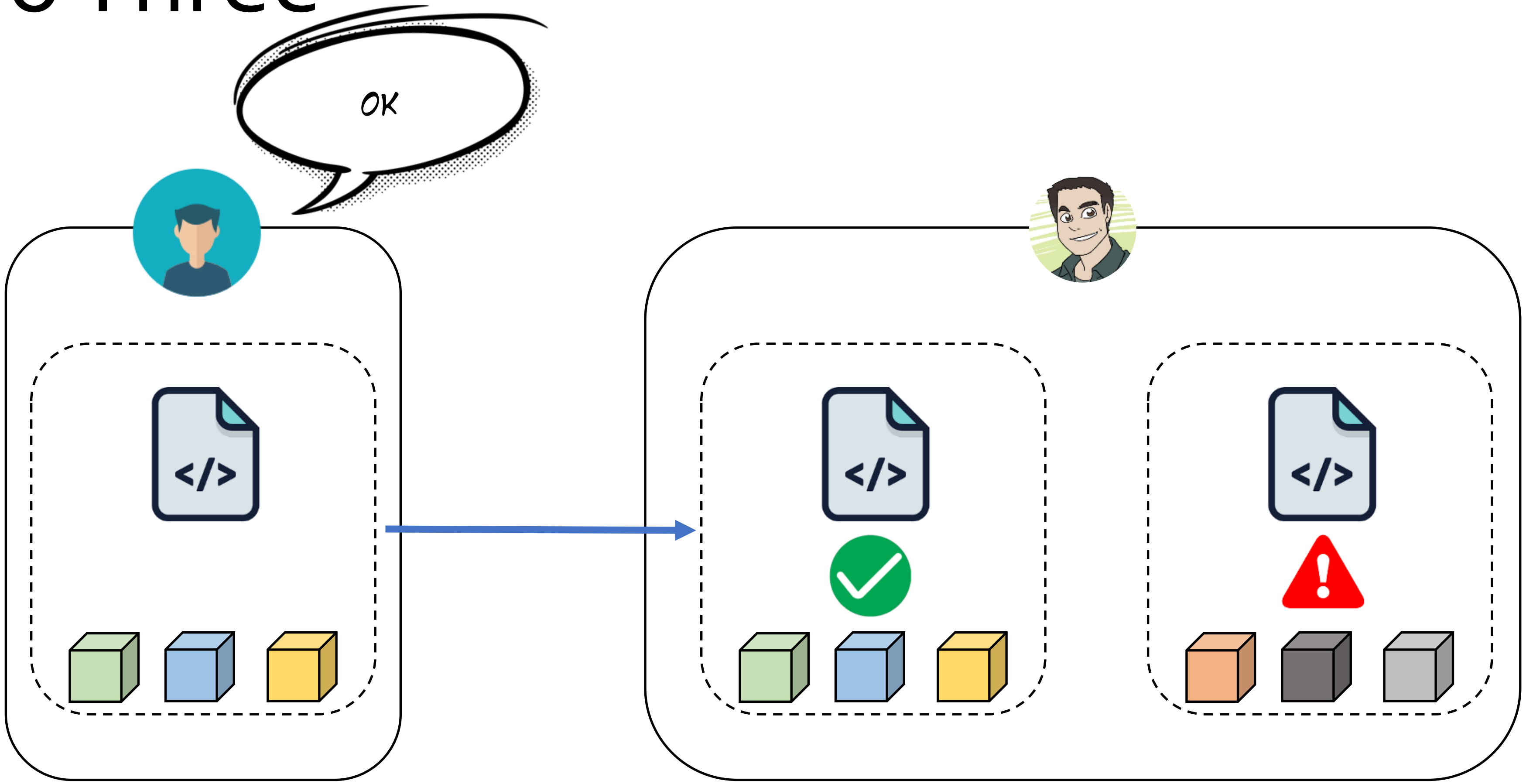
Scenario Three



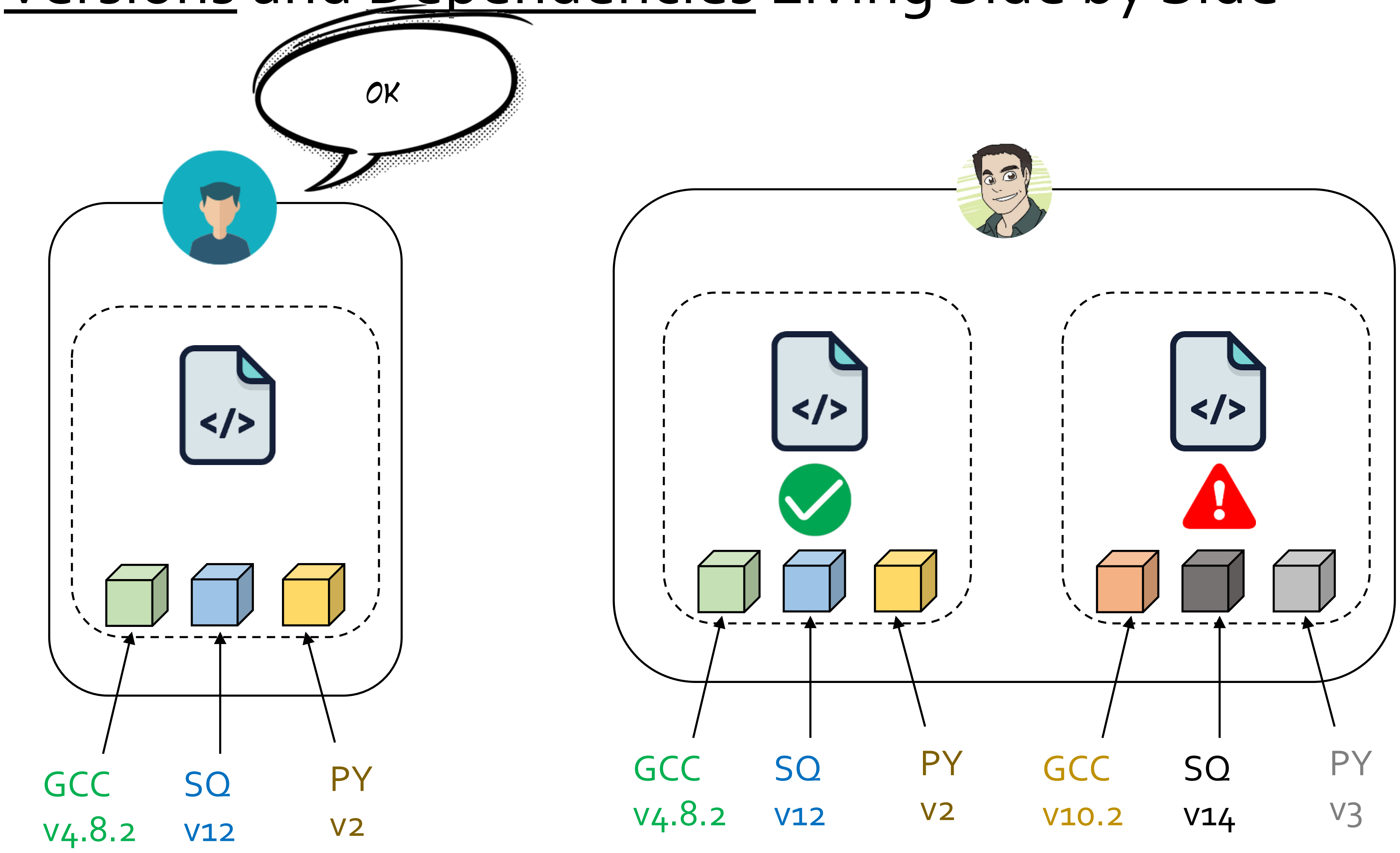
Scenario Three



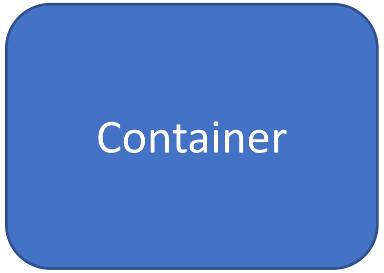
Scenario Three



Multiple Versions and Dependencies Living Side by Side

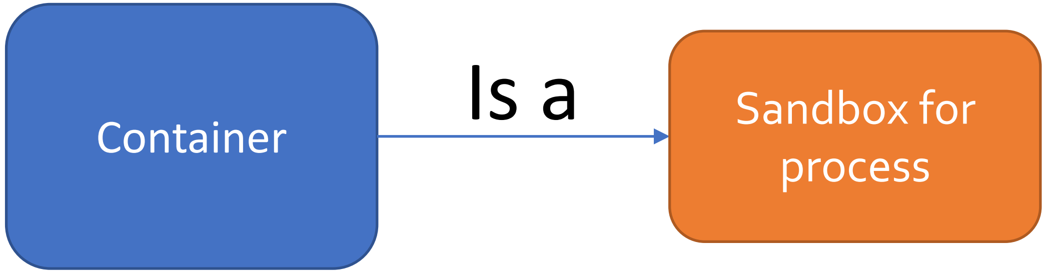


What is a container?



Container

What is a container?

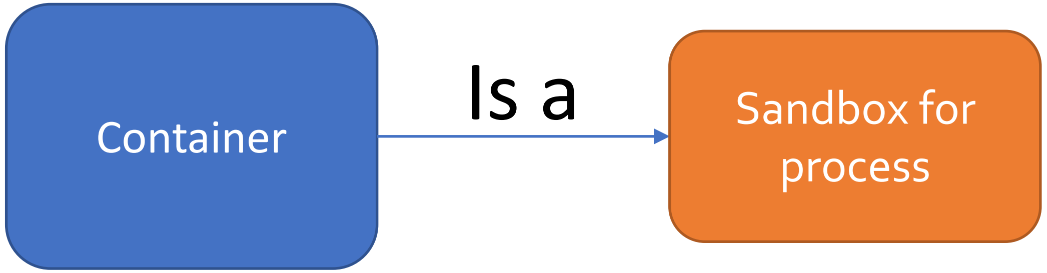


Container

Is a

Sandbox for process

What is a container?

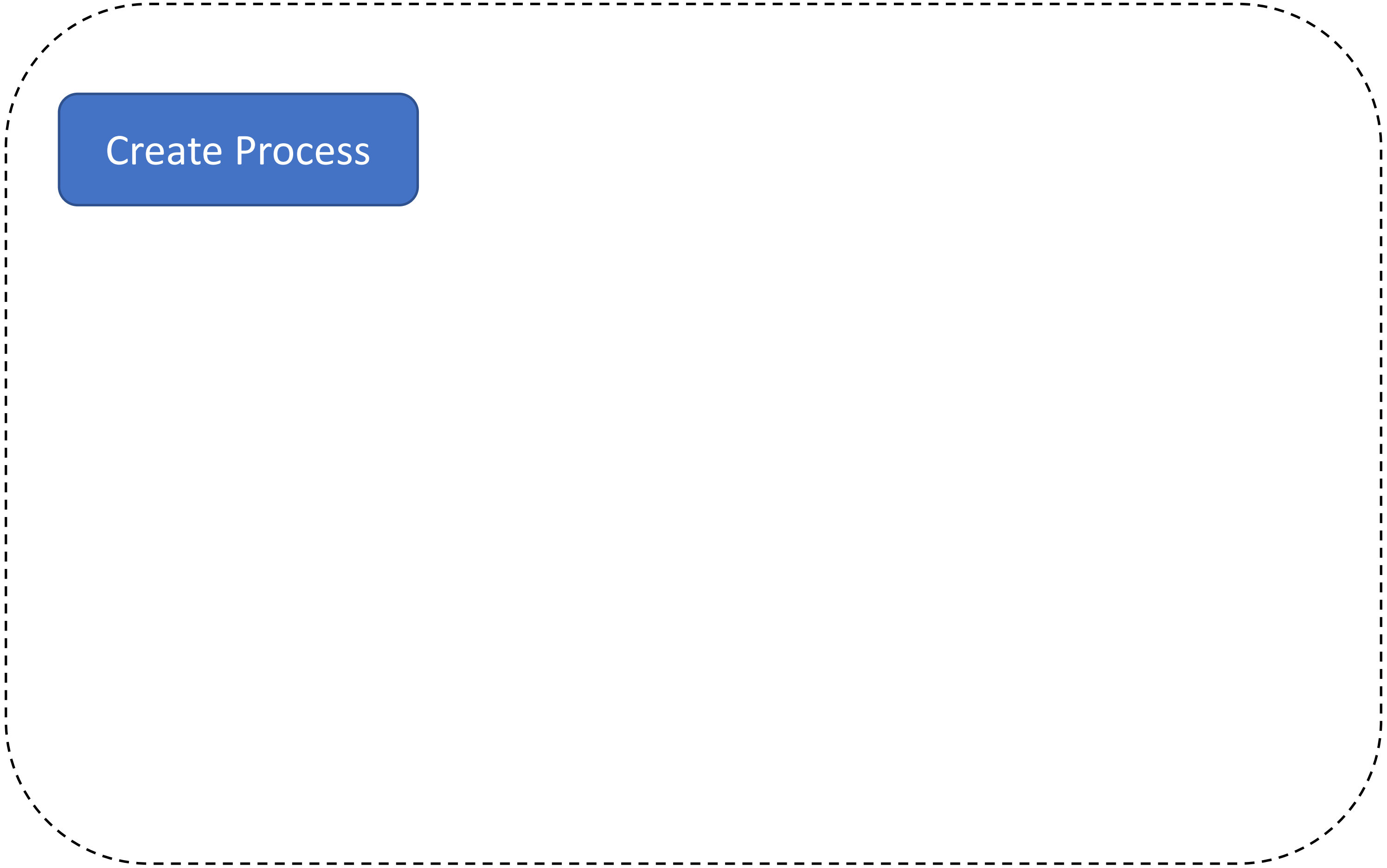


Containers are a type of <u>virtualization</u> for <u>Linux Kernel</u>

How does the kernel build this sandbox?

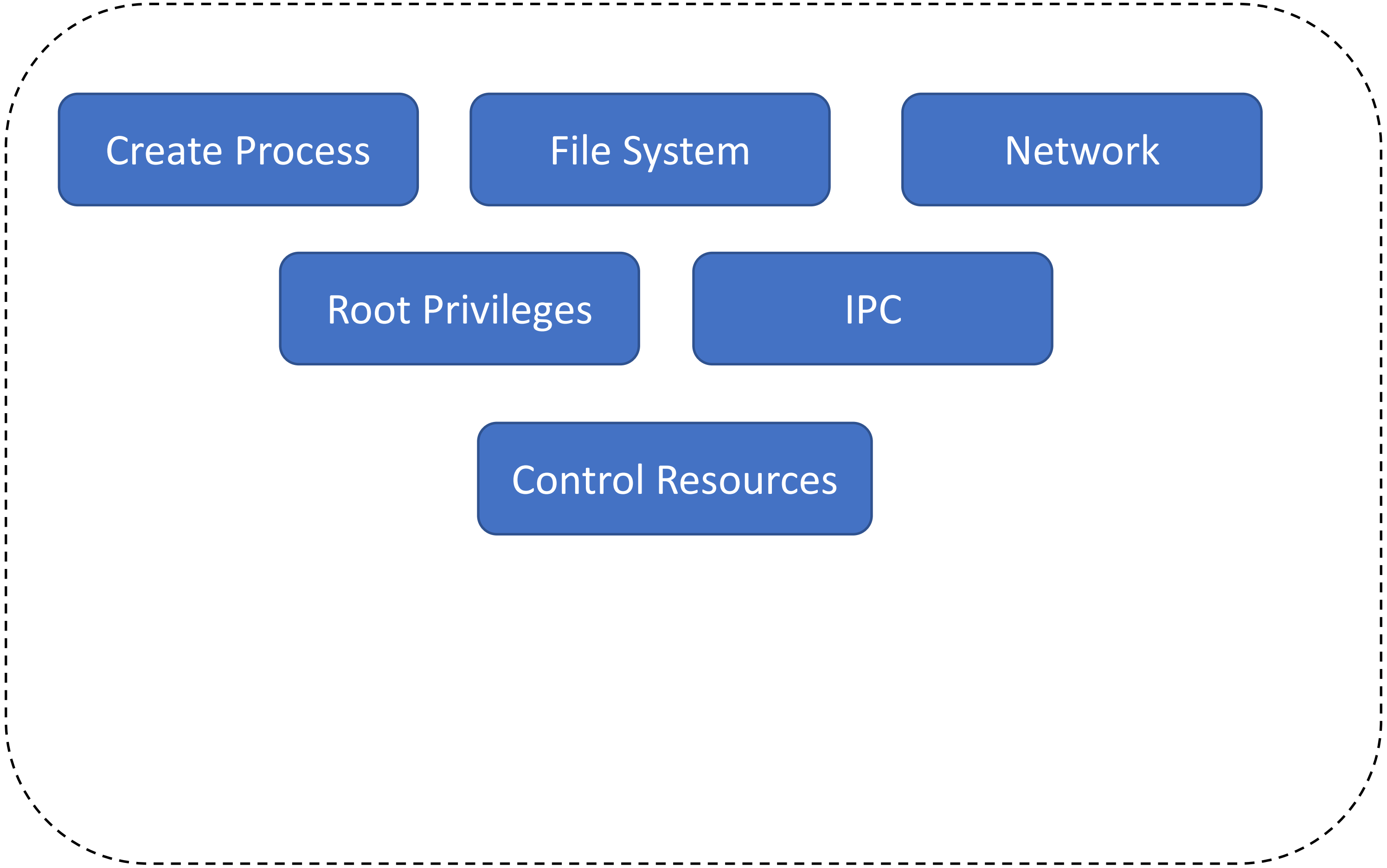
What kernel capabilities might a process (program) need?

What kernel capabilities might a process (program) need?



Create Process

What kernel capabilities might a process (program) need?



Create Process

File System

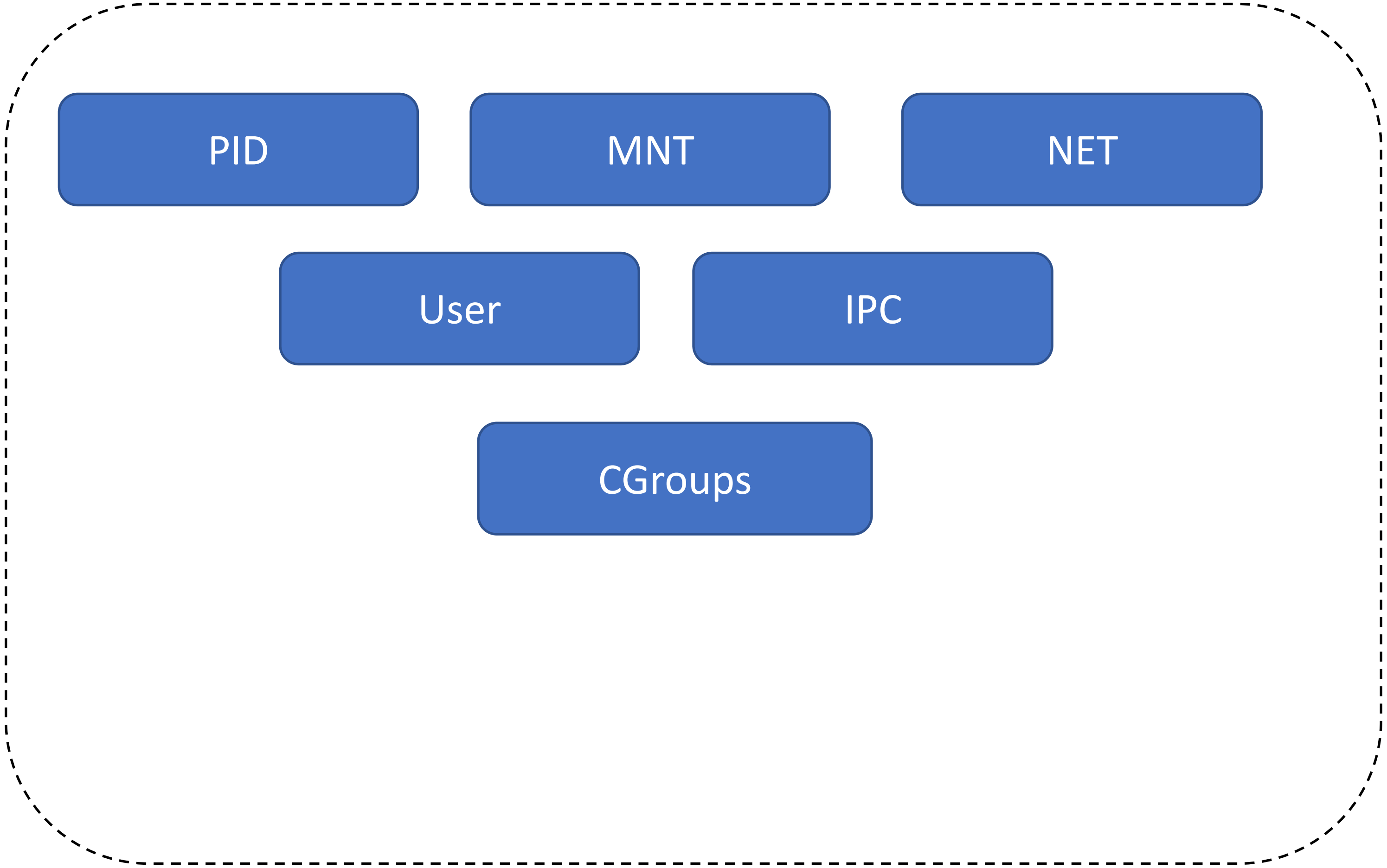
Network

Root Privileges

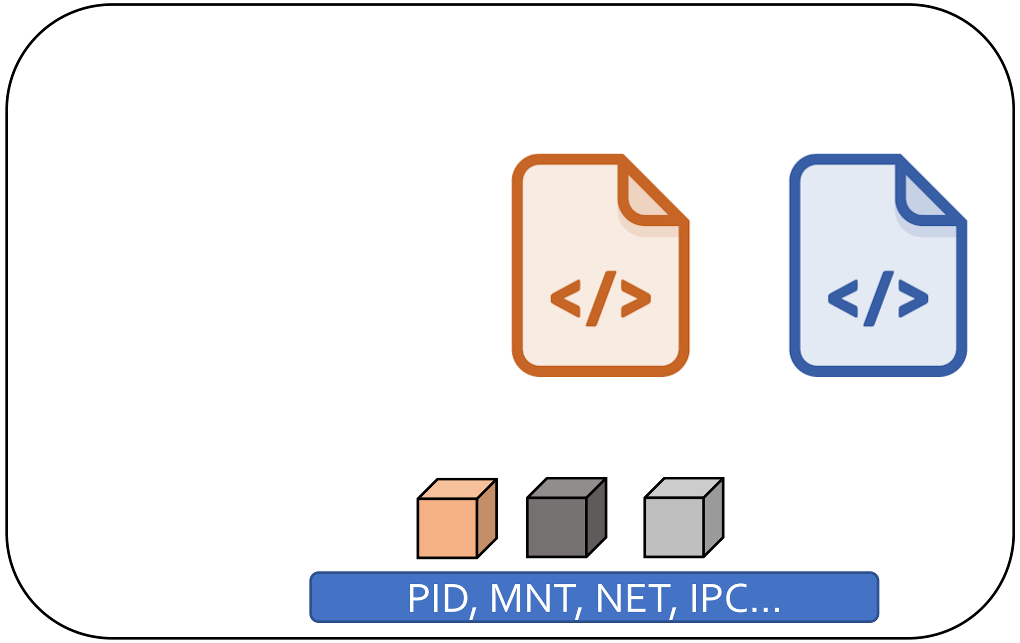
IPC

Control Resources

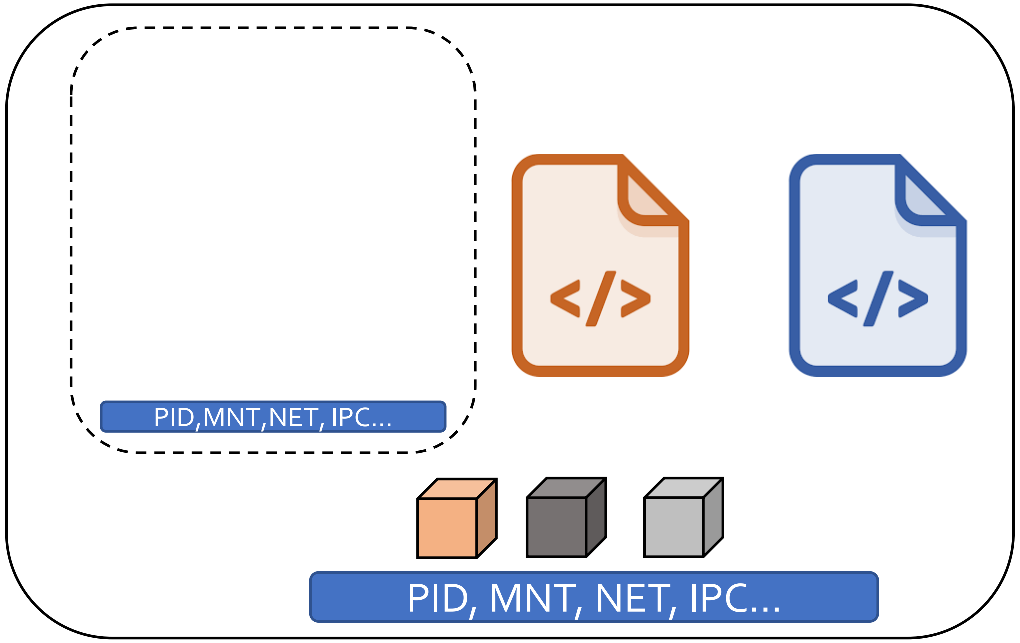
The capabilities are called *namespaces*



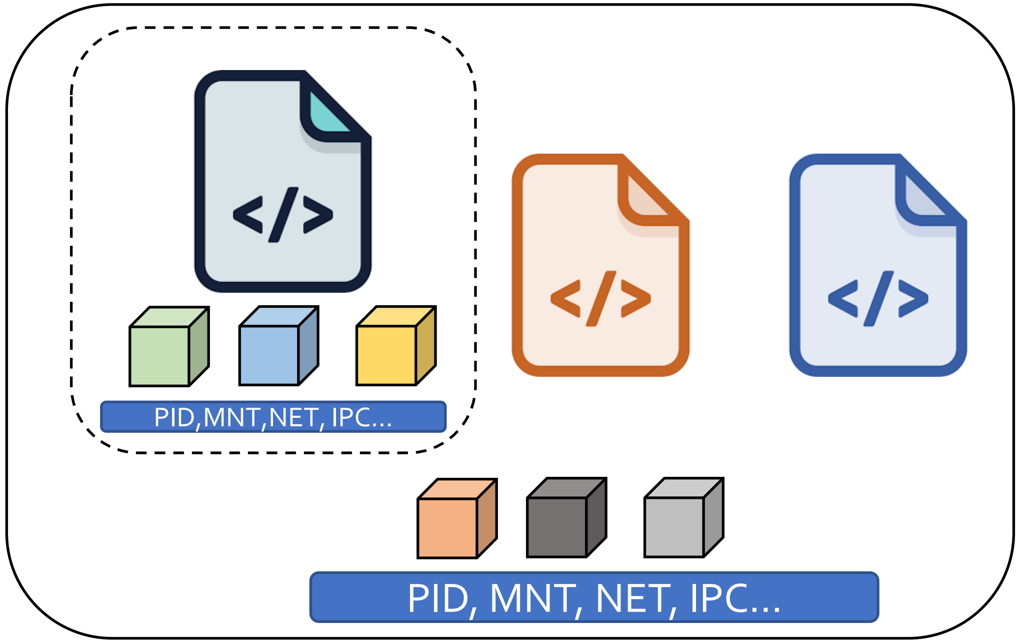
New containers mean new namespaces



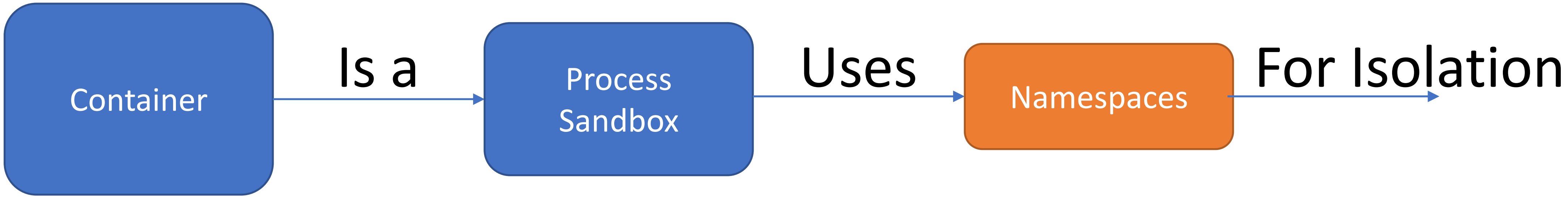
New containers mean new namespaces



New containers mean new namespaces

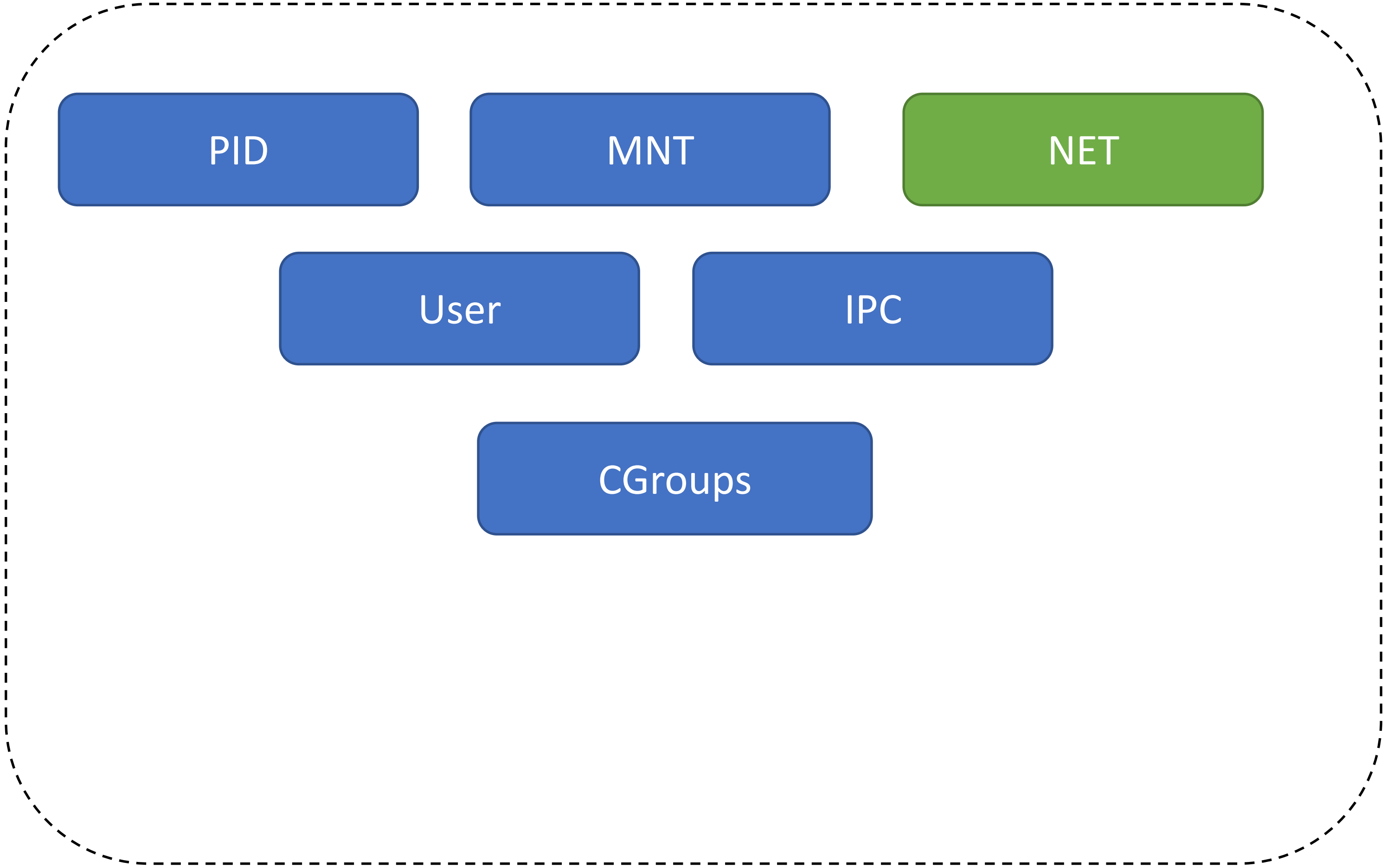


Review

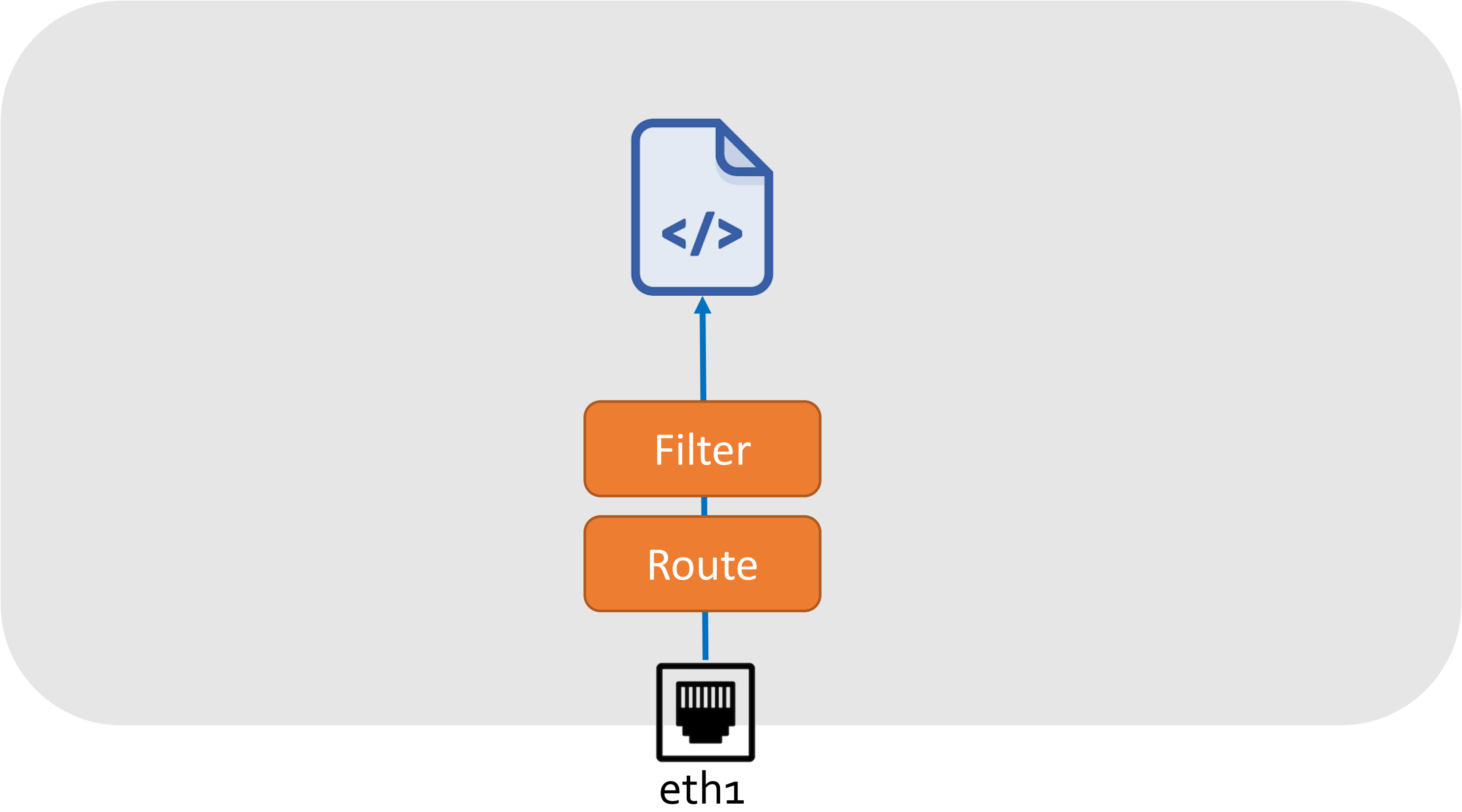

Container
Is a
Process Sandbox
Uses
Namespaces
For Isolation

Network Namespaces

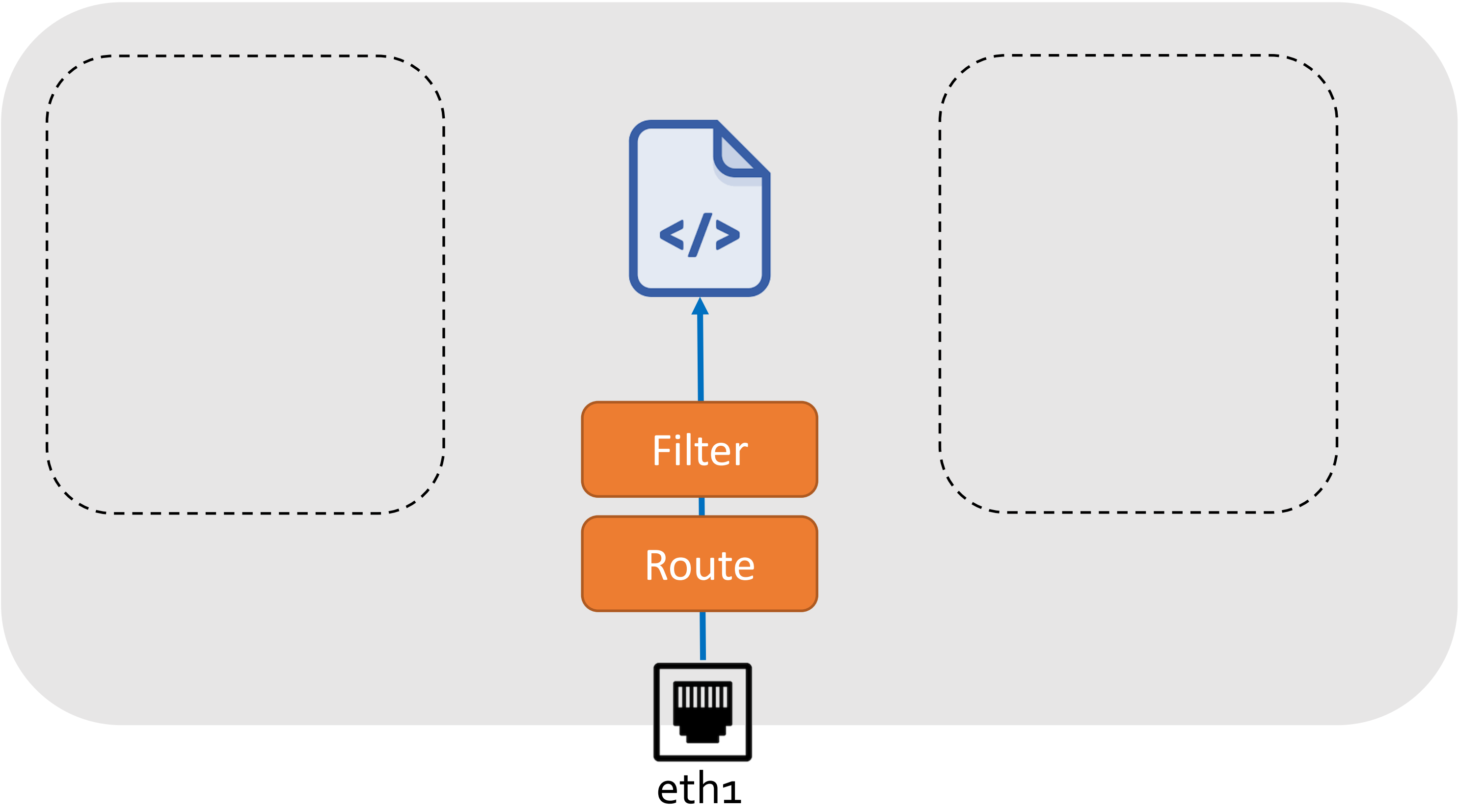
Closer look at the network namespace



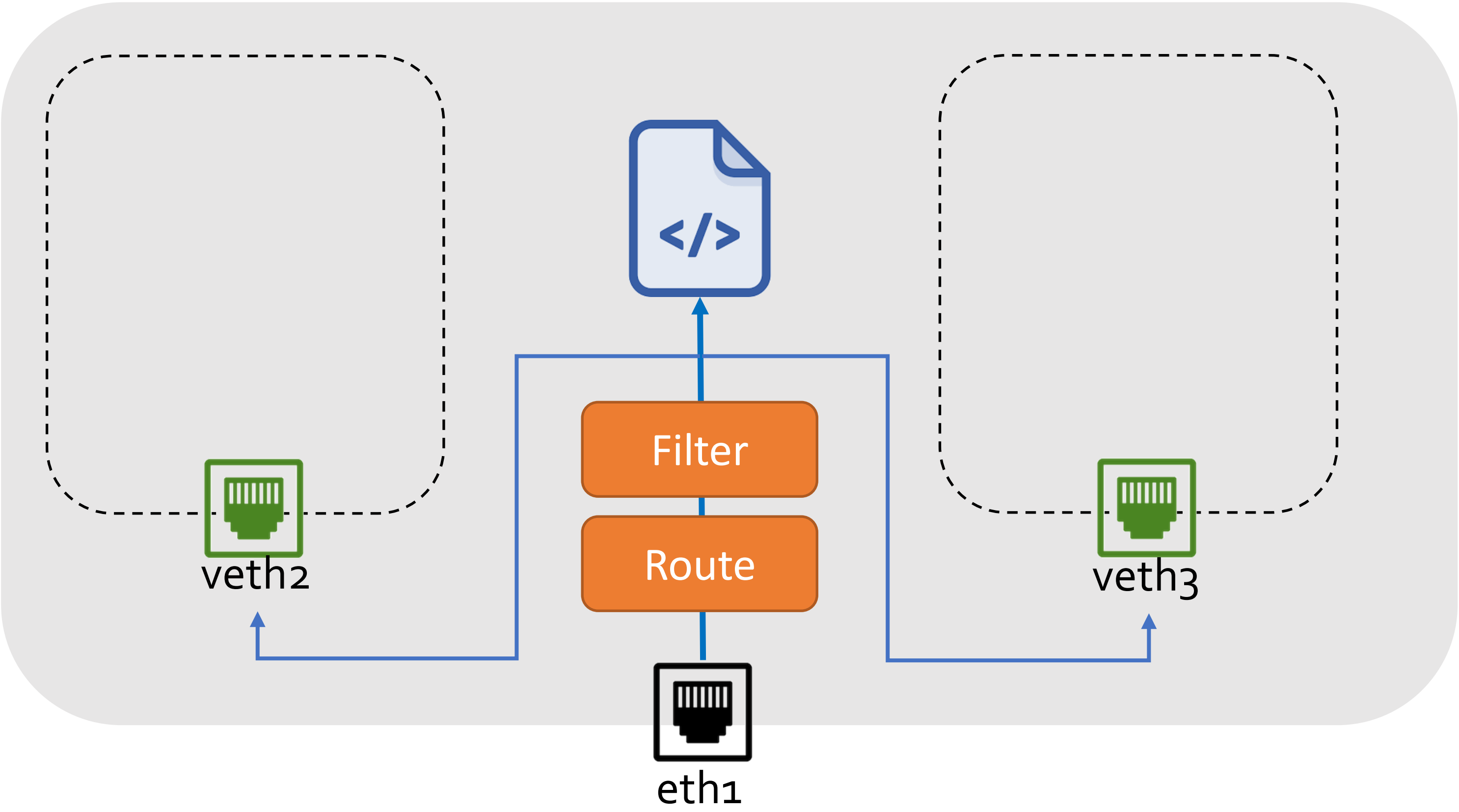
What is a network?



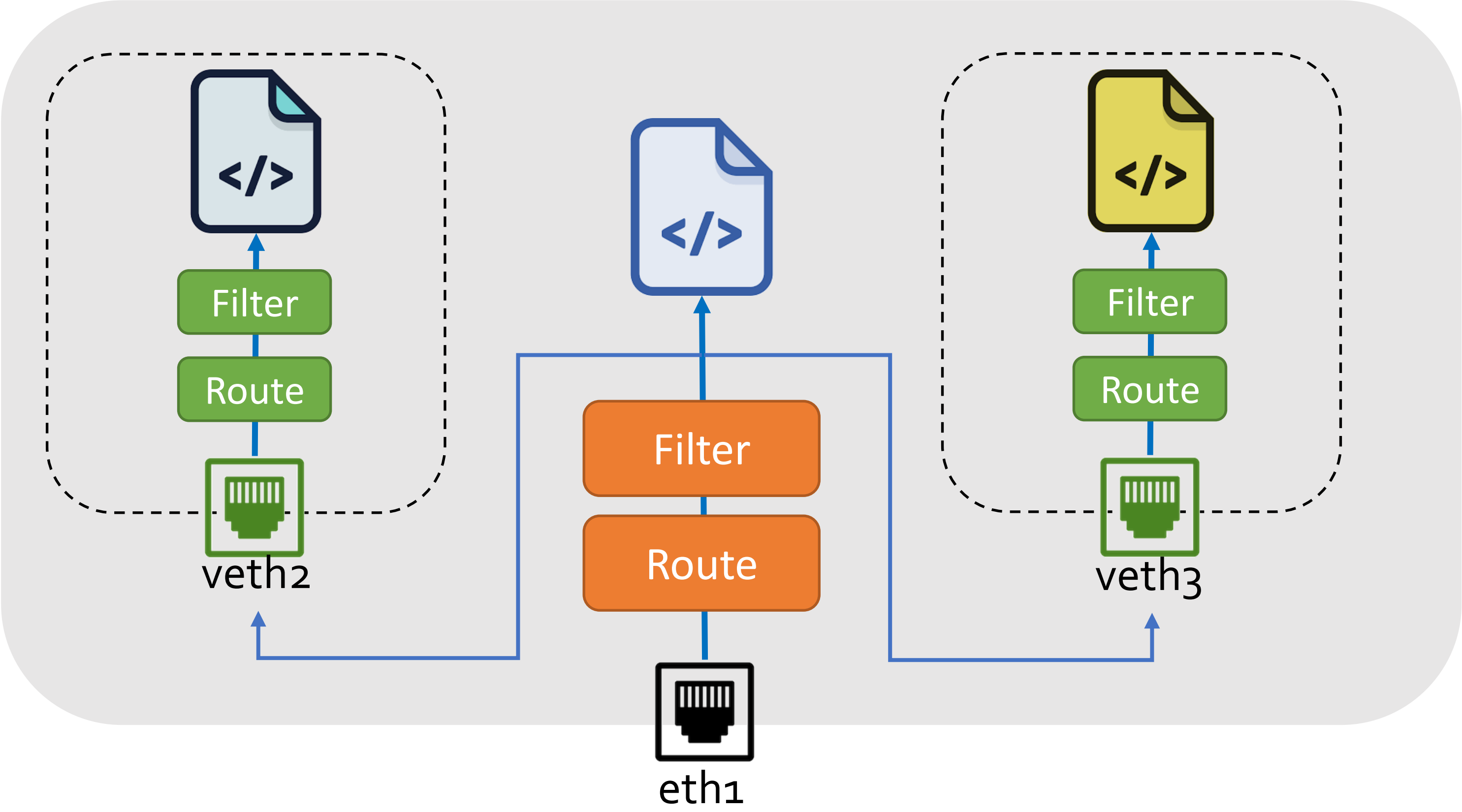
Network namespace isolation



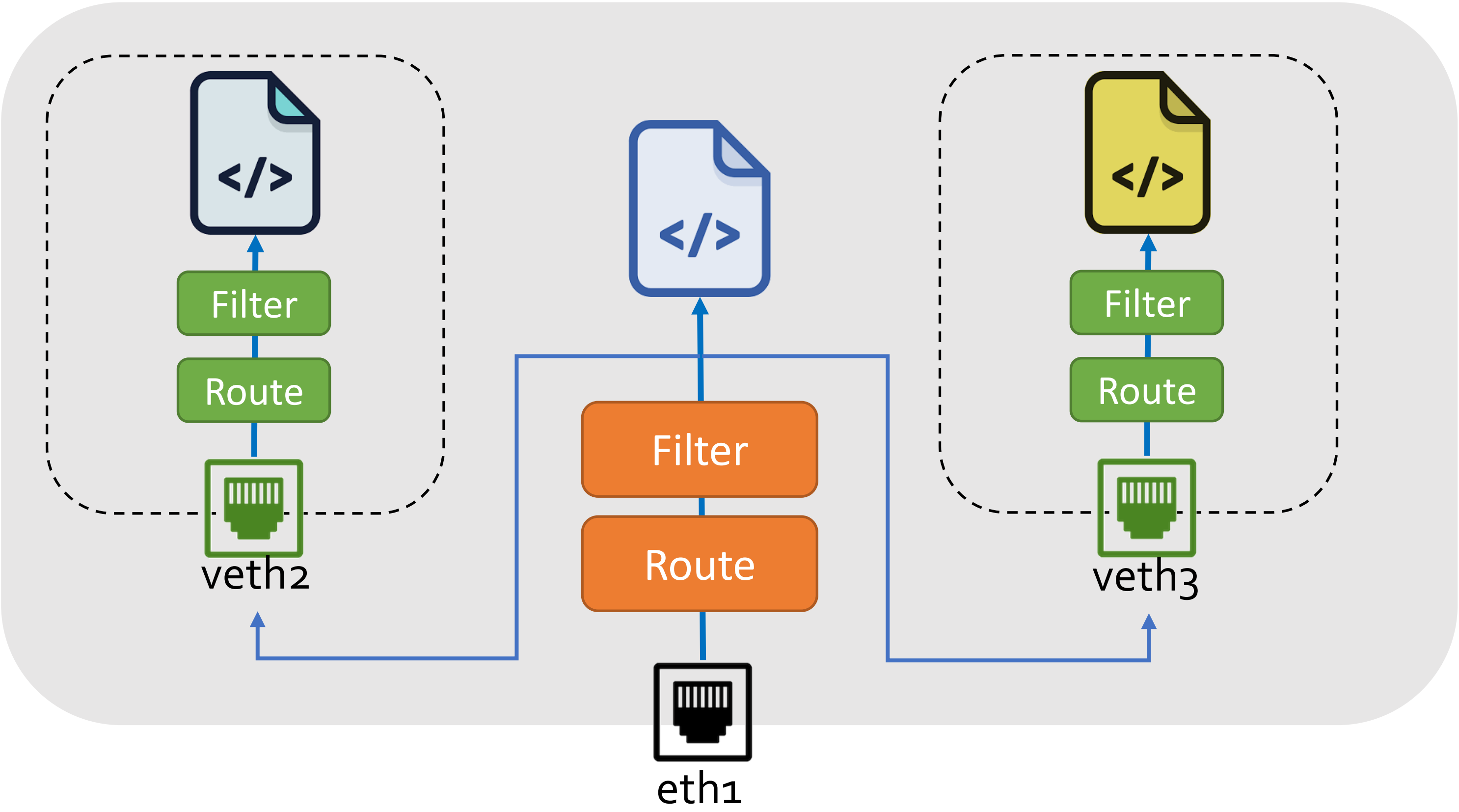
Network namespace isolation



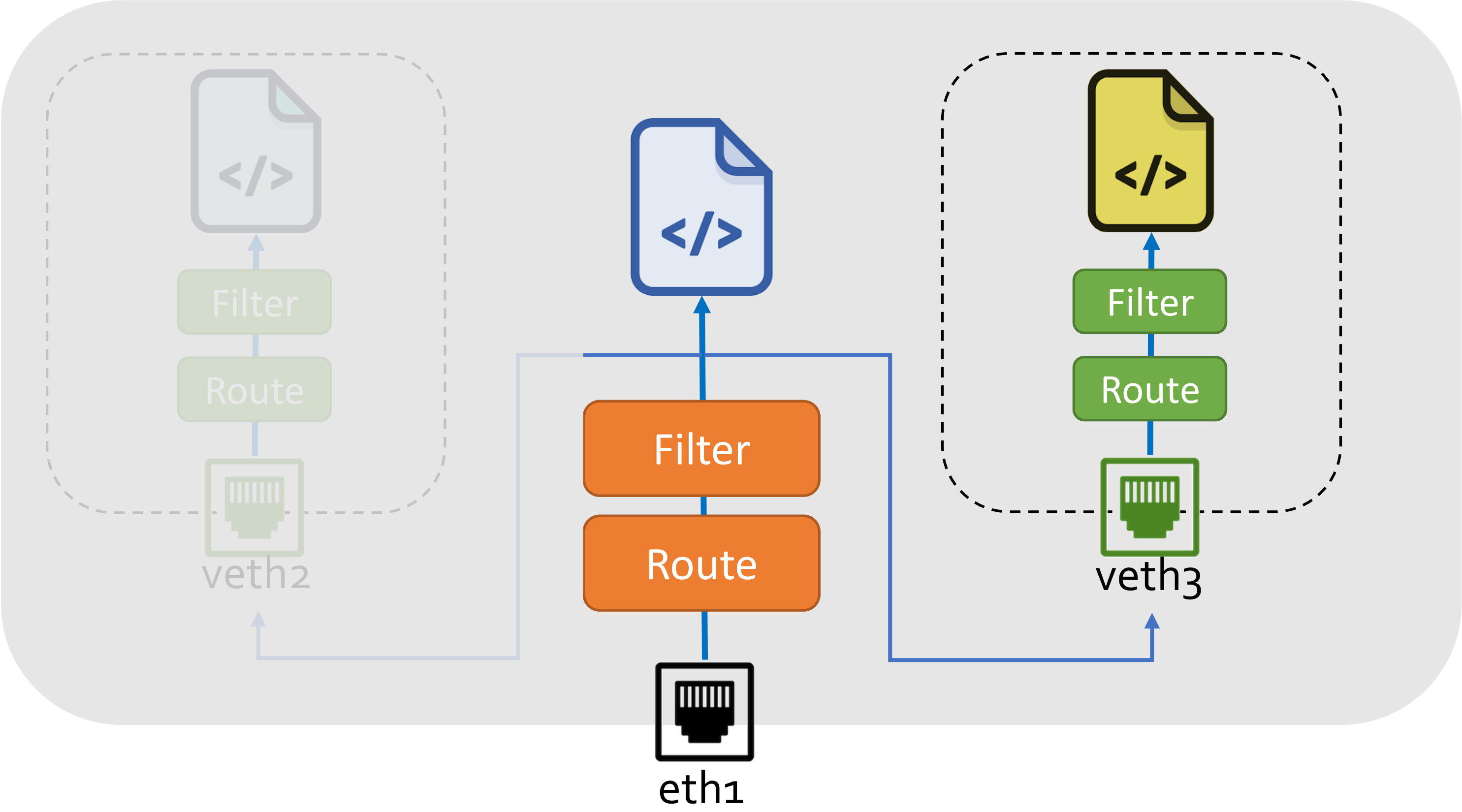
Network namespace isolation



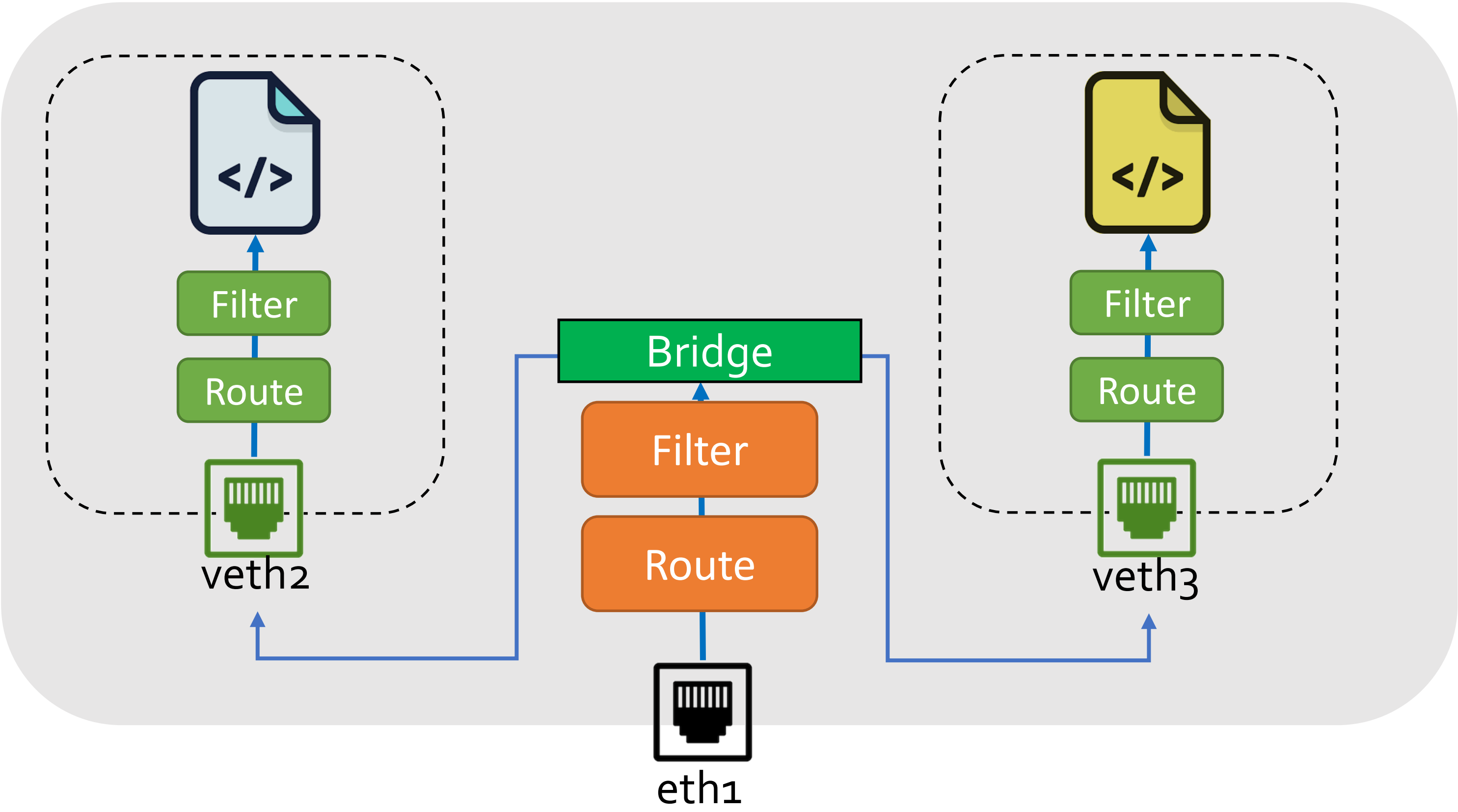
Operational Benefit: Easy to experiment with rules because they can be tied to the container



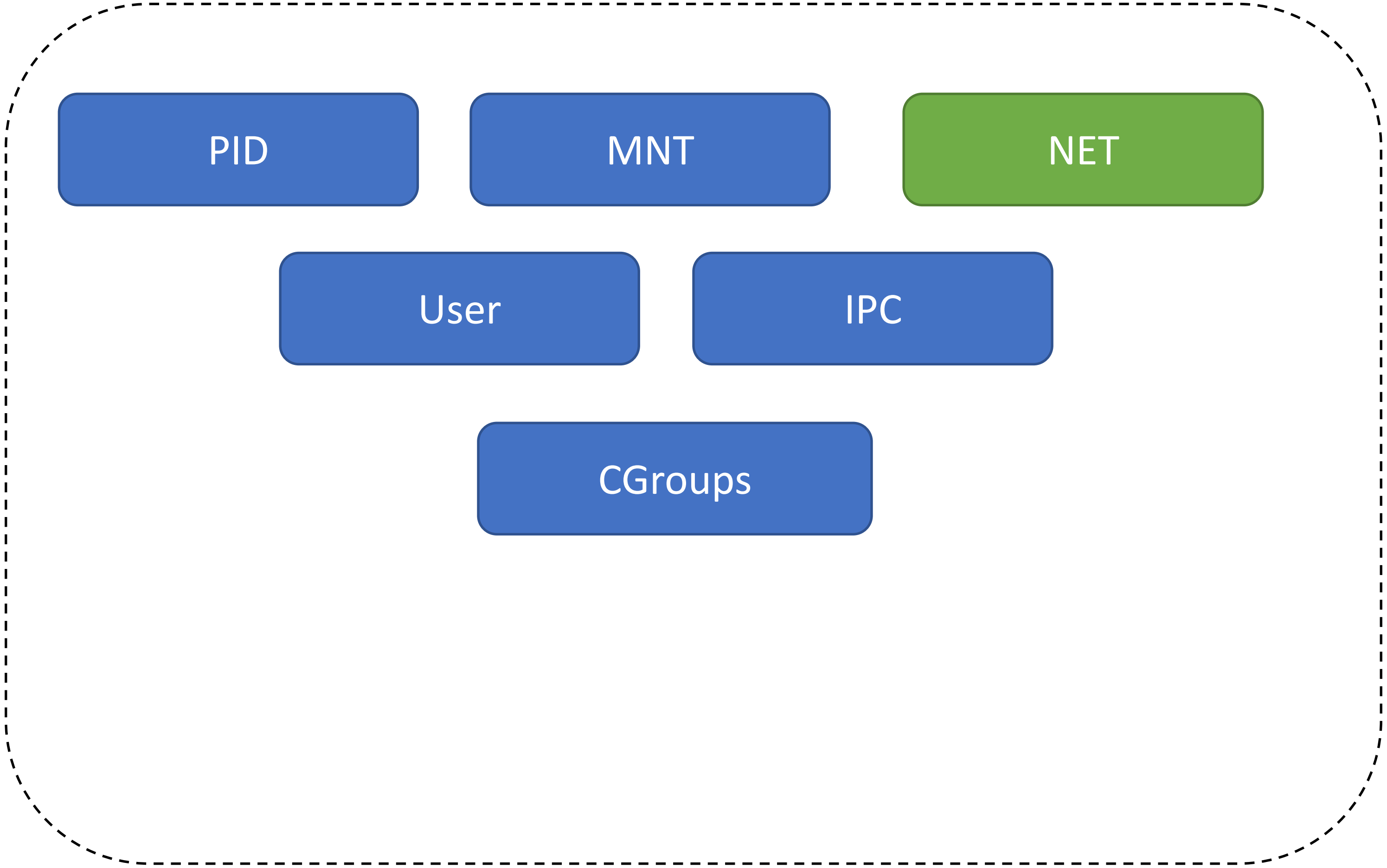
Operational Benefit: Easy to experiment with rules because they can be tied to the container



Container Network is often managed through a virtual bridge/switch



Other namespaces provide isolation in similar ways



PID

MNT

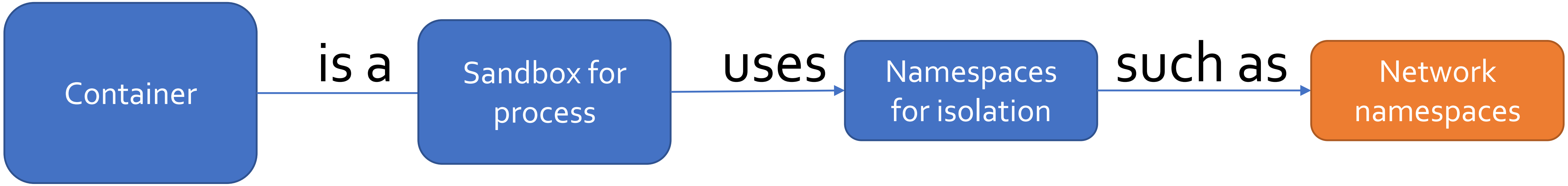
NET

User

IPC

CGroups

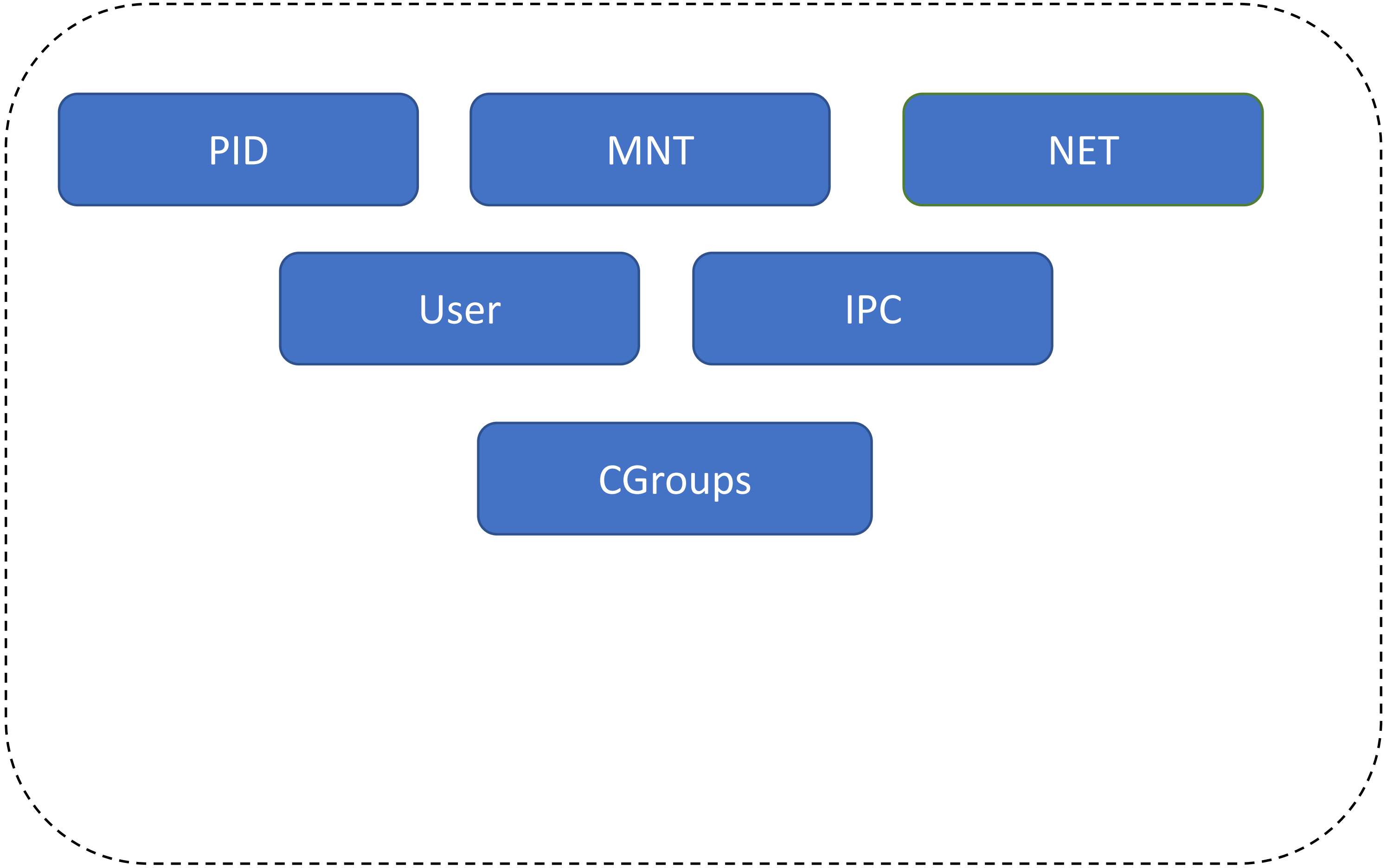
Review



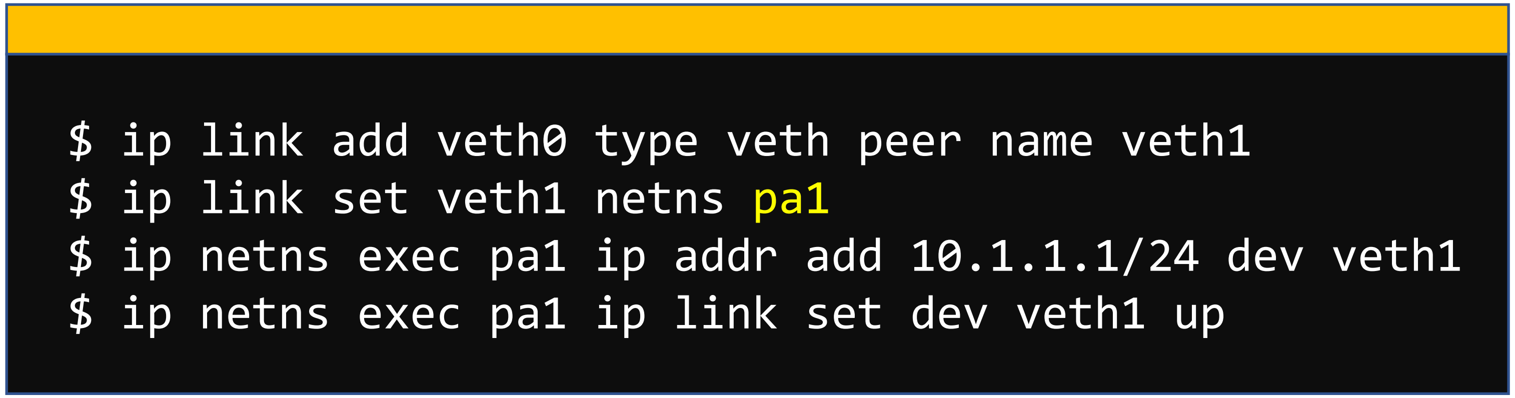
Container is a Sandbox for process uses Namespaces for isolation such as Network namespaces

Container Platforms and Docker

These are all kernel capabilities
i.e. you can build a container right now without additional software



Struggles of Creating Namespaces

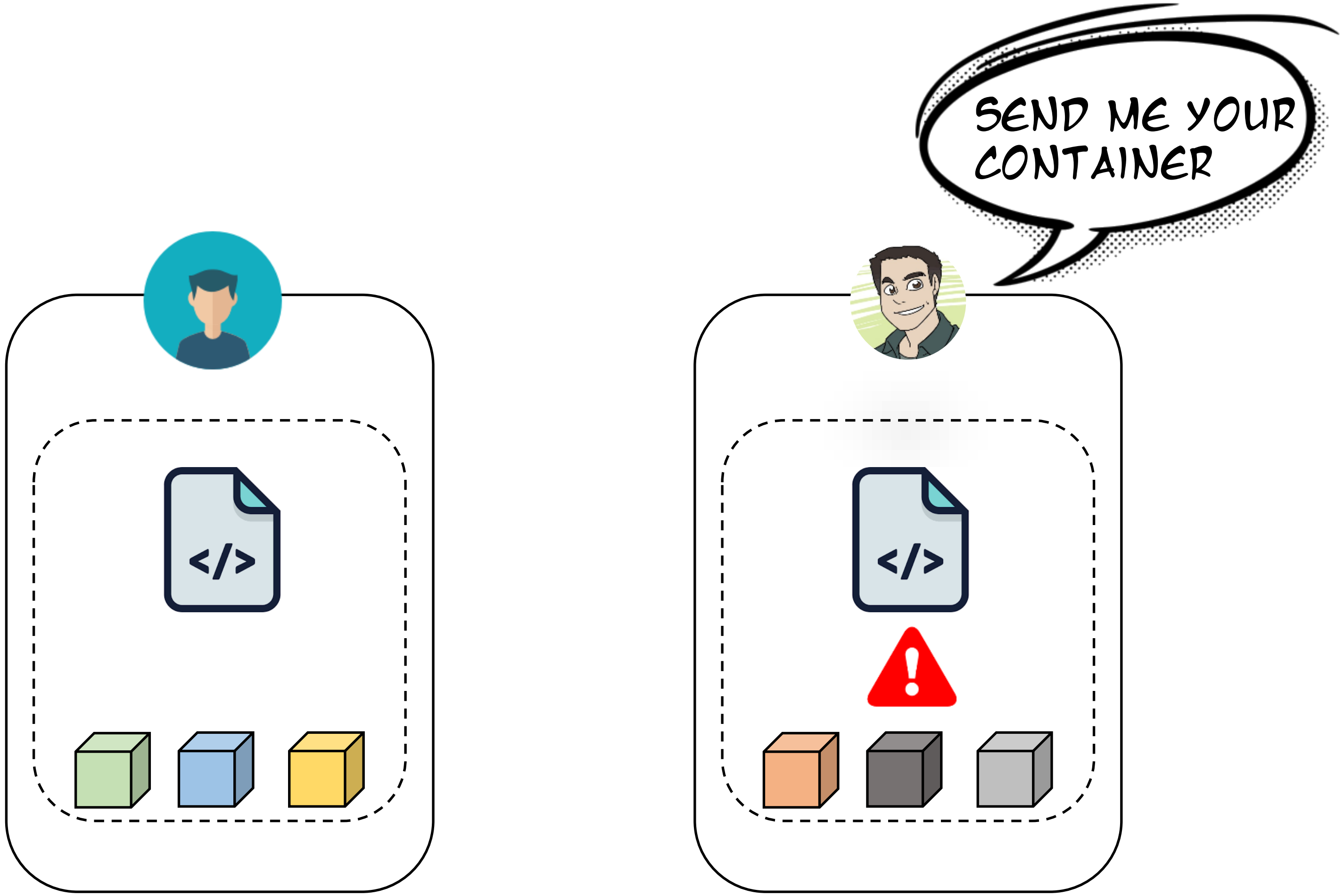


```
$ ip link add veth0 type veth peer name veth1
$ ip link set veth1 netns pa1
$ ip netns exec pa1 ip addr add 10.1.1.1/24 dev veth1
$ ip netns exec pa1 ip link set dev veth1 up
```

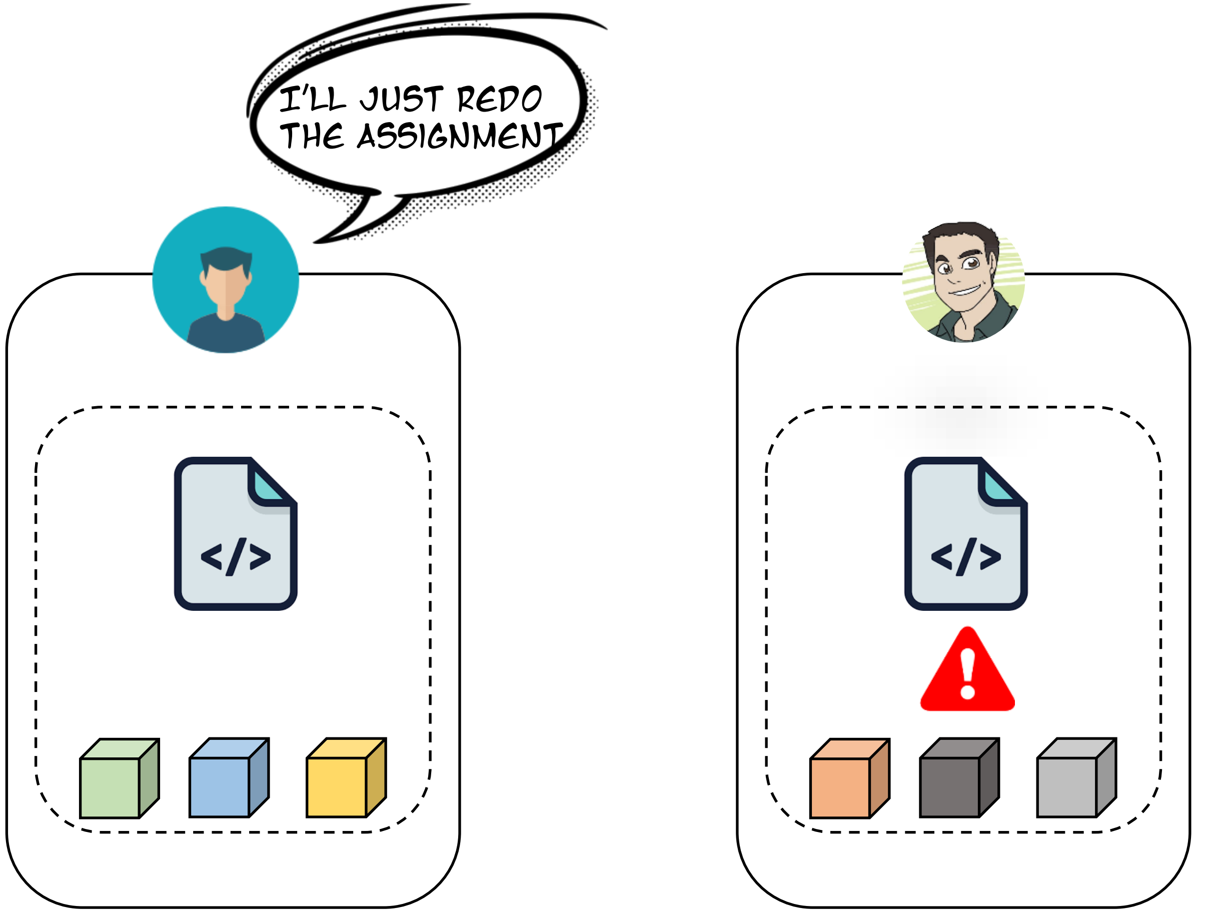
Repeat for additional isolation and you’ve created *one container*

```
$ ip link add veth0 type veth peer name veth1
$ ip link set veth1 netns pa1
$ ip netns exec pa1 ip addr add 10.1.1.1/24 dev veth1
$ ip netns exec pa1 ip link set dev veth1 up
```

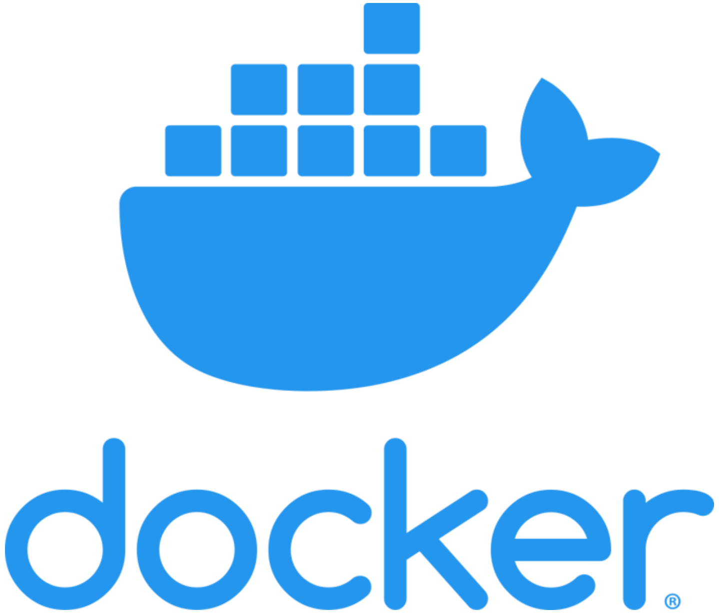
Scenario Three



Scenario Three



Container software makes it easy to manage containers and namespaces



Docker abstracts underlying namespaces

Create Dockerfile

```
FROM ubuntu
RUN apt-get update
RUN apt-get install mysql:5.6.50 -y
COPY /usr/shelbyt/cse224/pa1/
/usr/cse224/pa1/
```

Docker abstracts underlying namespaces

Create Dockerfile

```
FROM ubuntu
RUN apt-get update
RUN apt-get install mysql:5.6.50 -y
COPY /usr/shelbyt/cse224/pa1/
/usr/cse224/pa1/
```

Build an Image

```
docker build –t  pa1 .
```

Docker abstracts underlying namespaces

Create Dockerfile

```
FROM ubuntu
RUN apt-get update
RUN apt-get install mysql:5.6.50 -y
COPY /usr/shelbyt/cse224/pa1/
/usr/cse224/pa1/
```

Build an Image

```
docker build –t  pa1 .
```

Run

```
docker run –it pa1
% (pa1-shell):
```

Docker abstracts underlying namespaces

Create Dockerfile

```
FROM ubuntu
RUN apt-get update
RUN apt-get install mysql:5.6.50 -y
COPY /usr/shelbyt/cse224/pa1/
/usr/cse224/pa1/
```

Build an Image

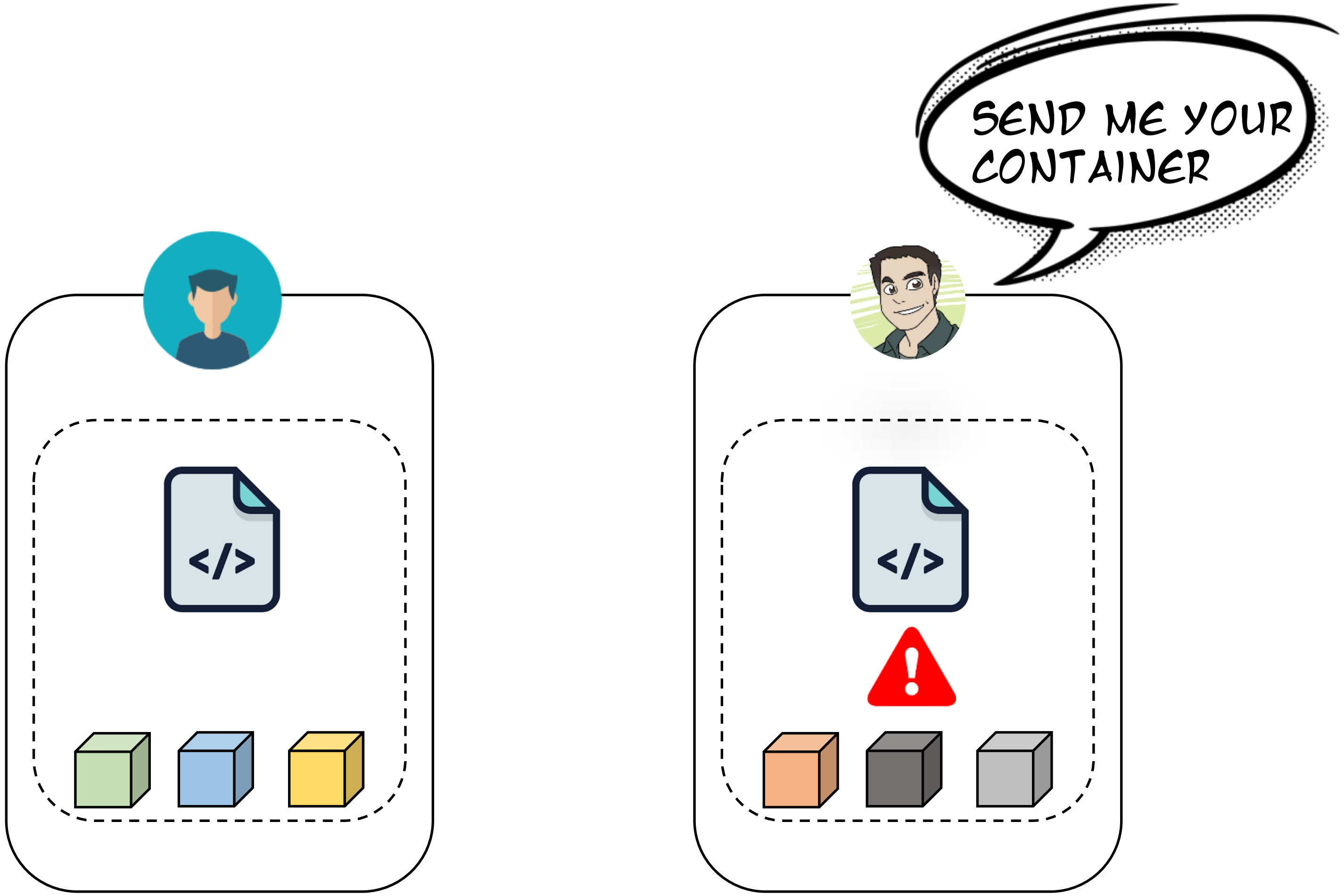
```
docker build –t  pa1 .
```

Run

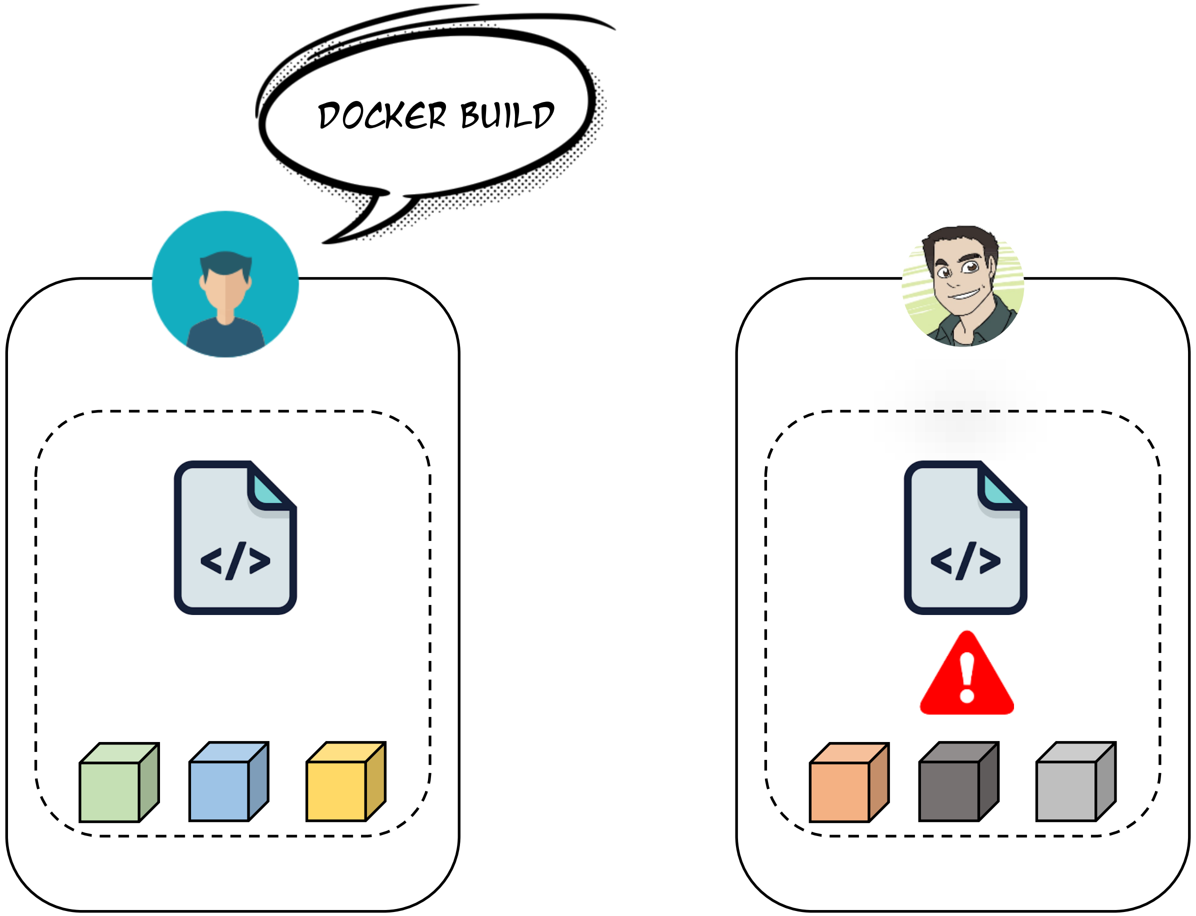
```
docker run –it pa1
% (pa1-shell):
```

Isolated shell, no manual configuration of namespaces needed

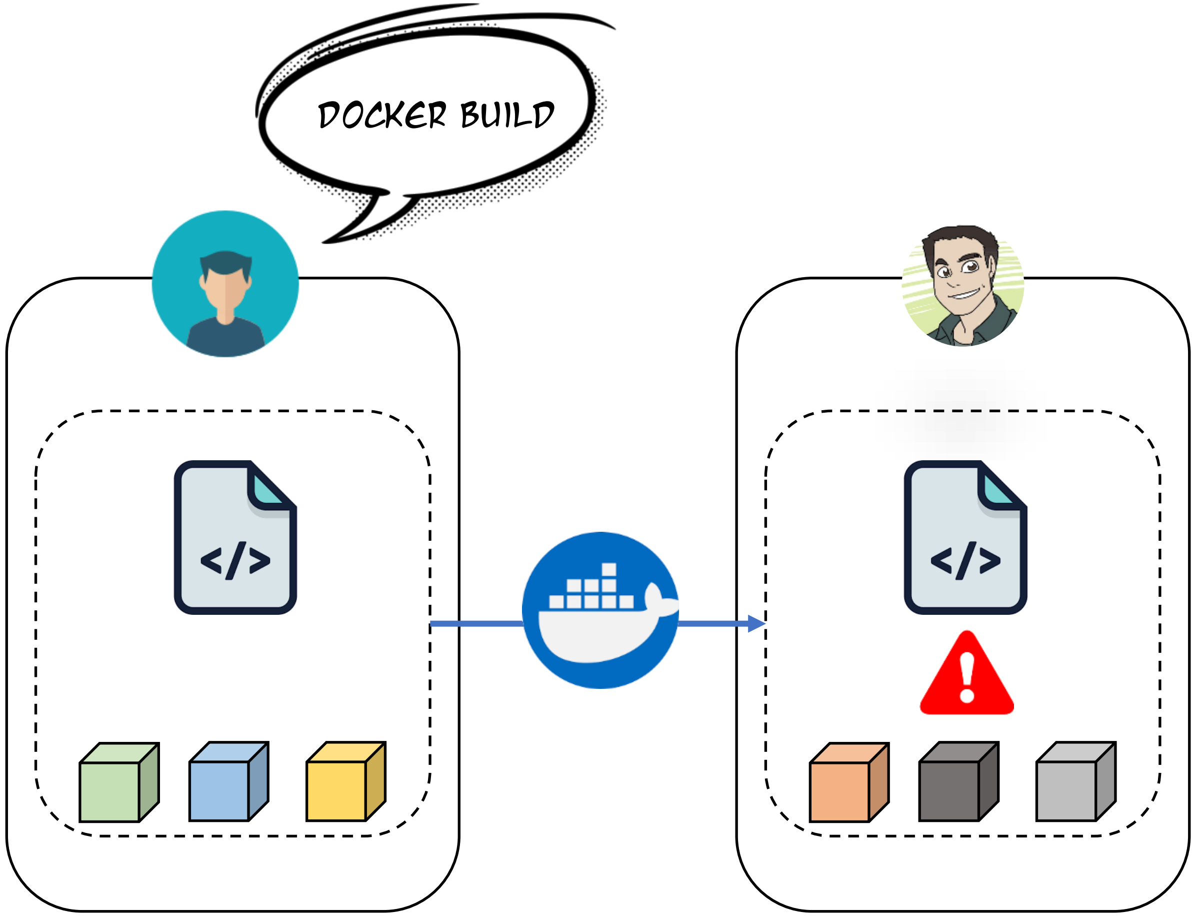
Scenario Three



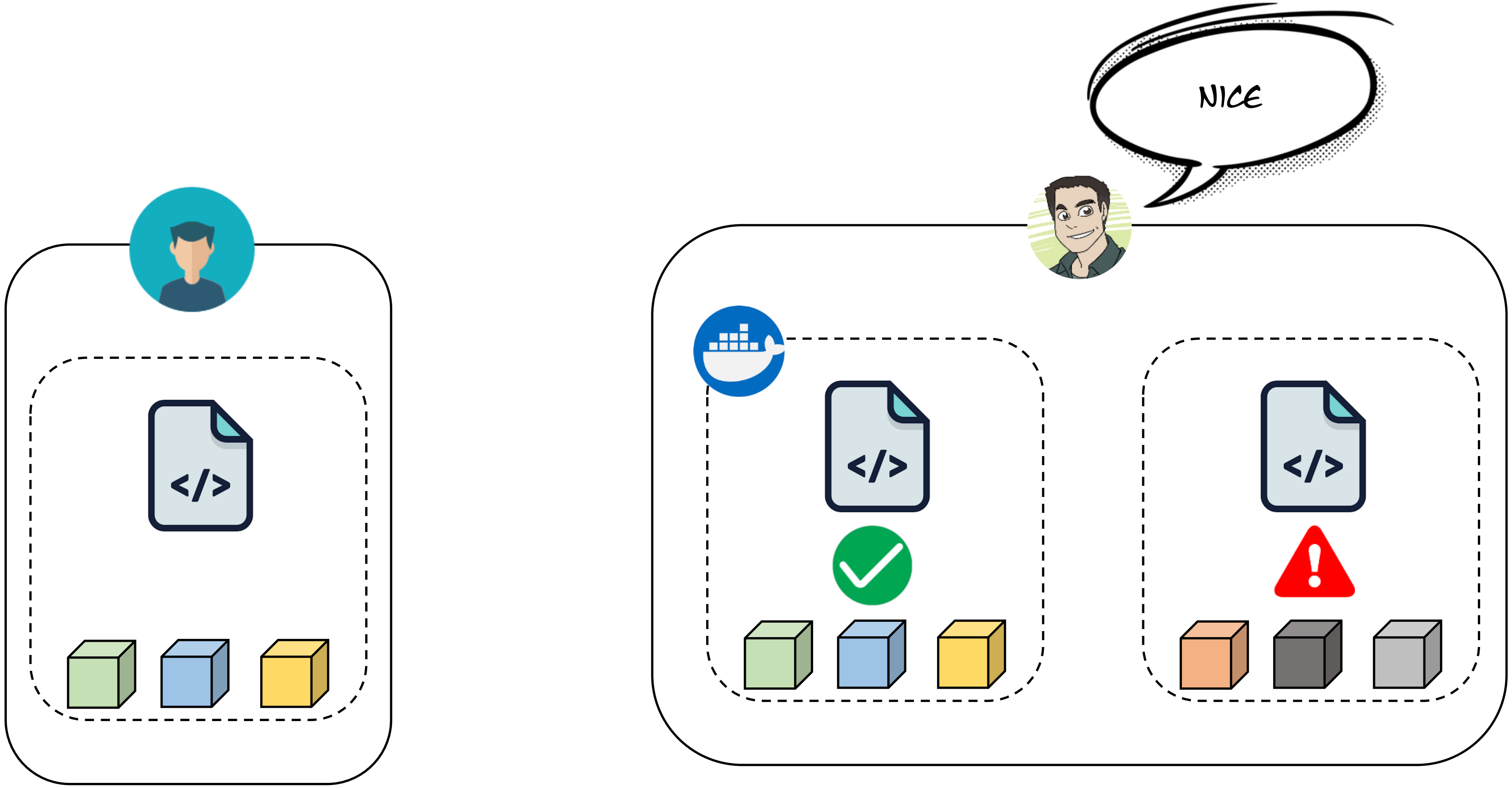
Scenario Three



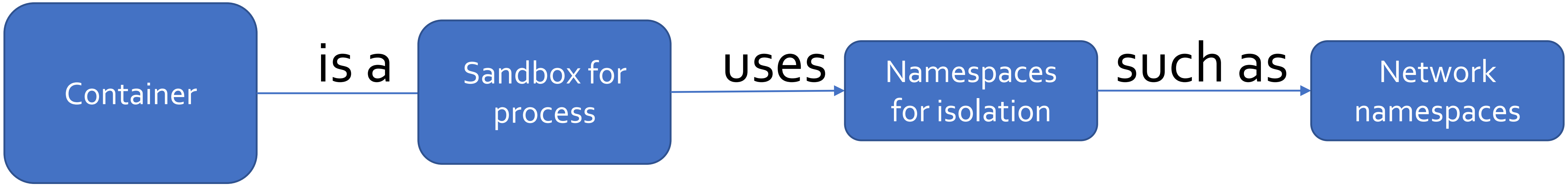
Just send the image



Just send the image



Review



Container

is a

Sandbox for process

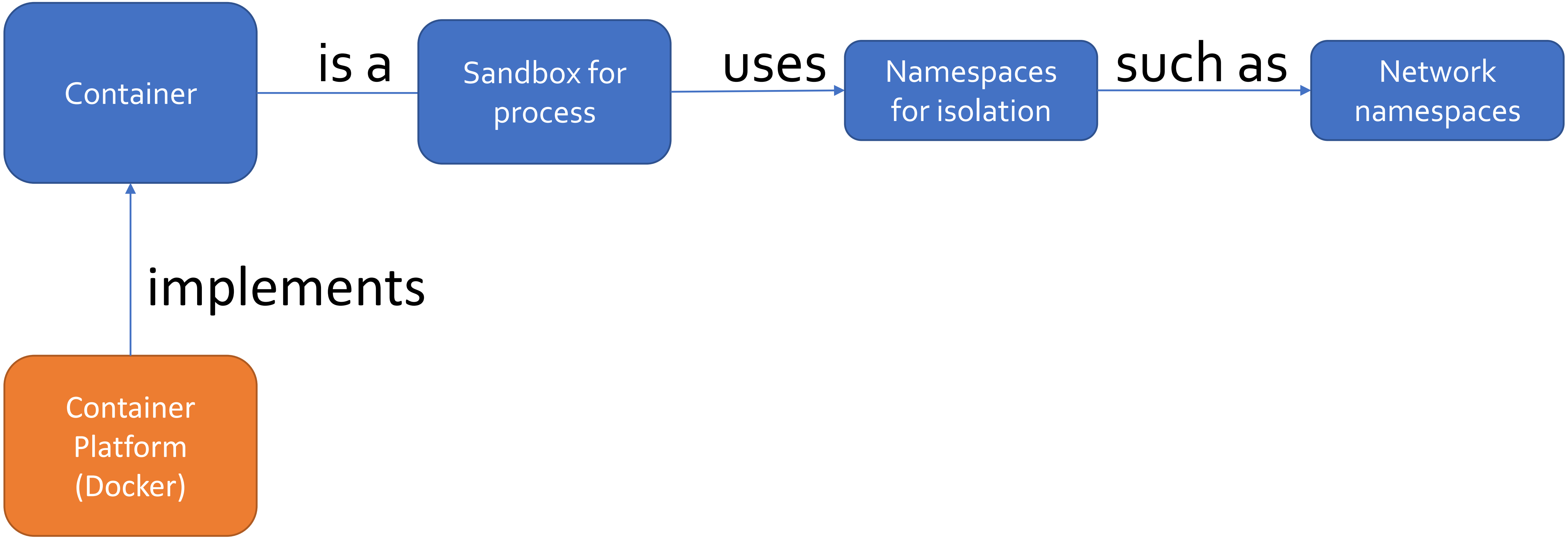
uses

Namespaces for isolation

such as

Network namespaces

Review

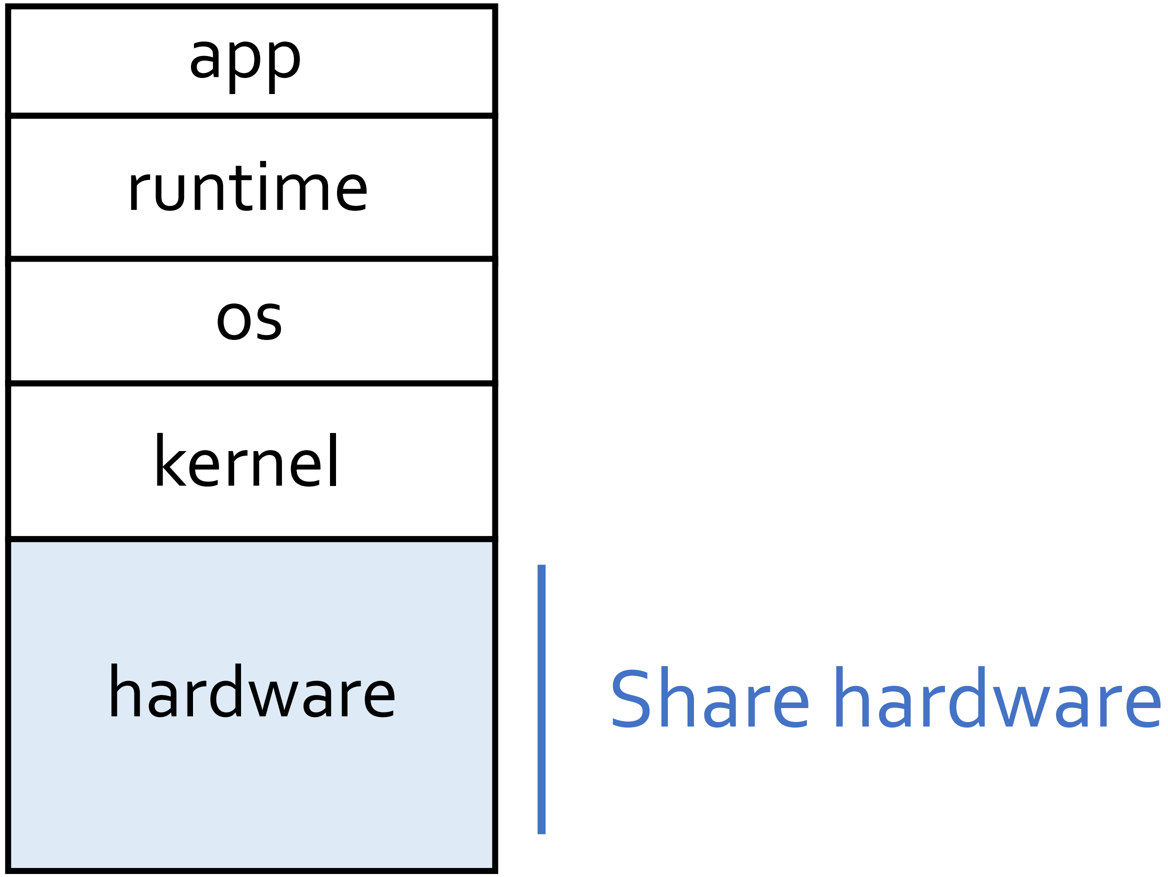


Container is a Sandbox for process uses Namespaces for isolation such as Network namespaces

Container Platform (Docker) implements Container

Virtual Machine vs. Containers

Why not use a Virtual Machine?



app

runtime

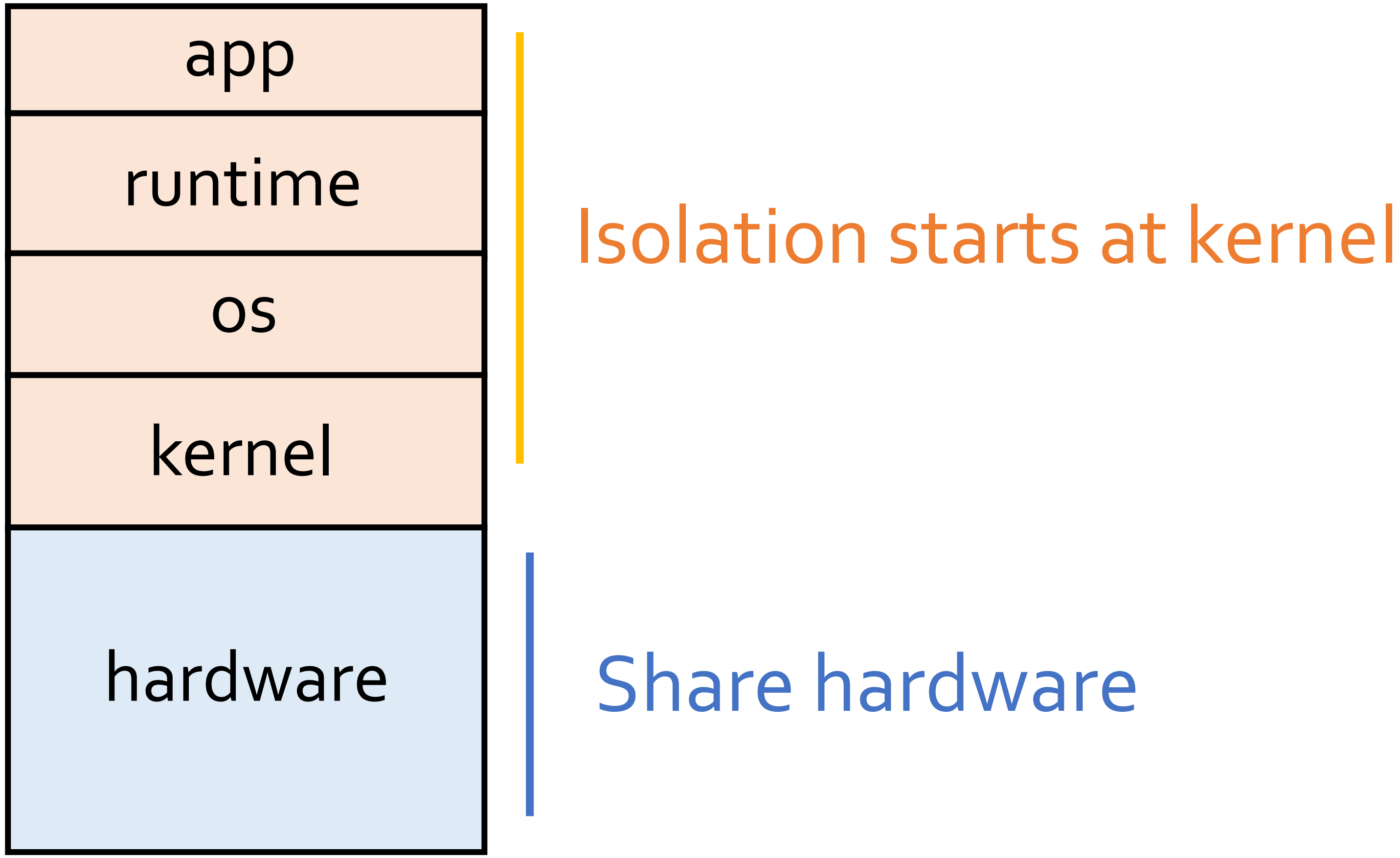
os

kernel

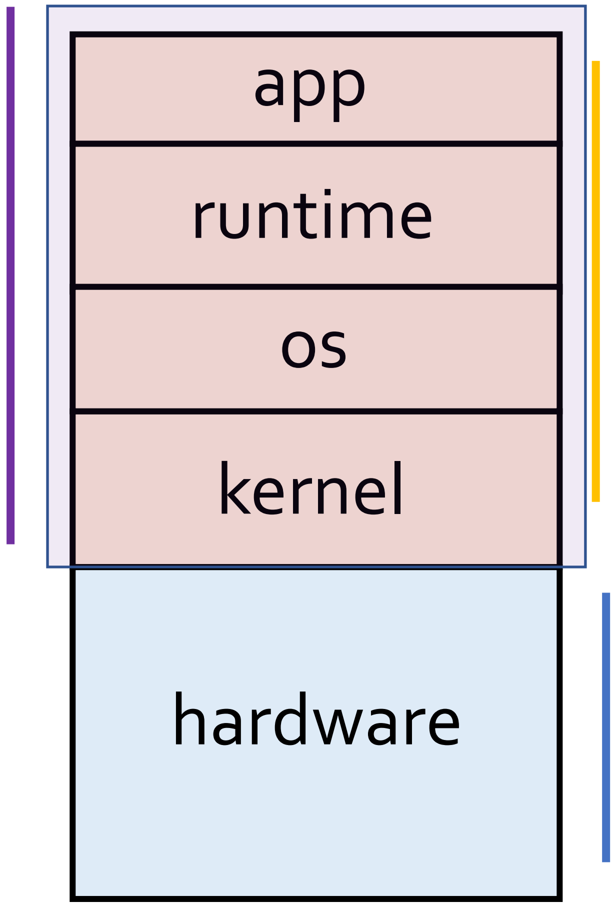
hardware

Share hardware

Different Isolation Points



Different Isolation Points

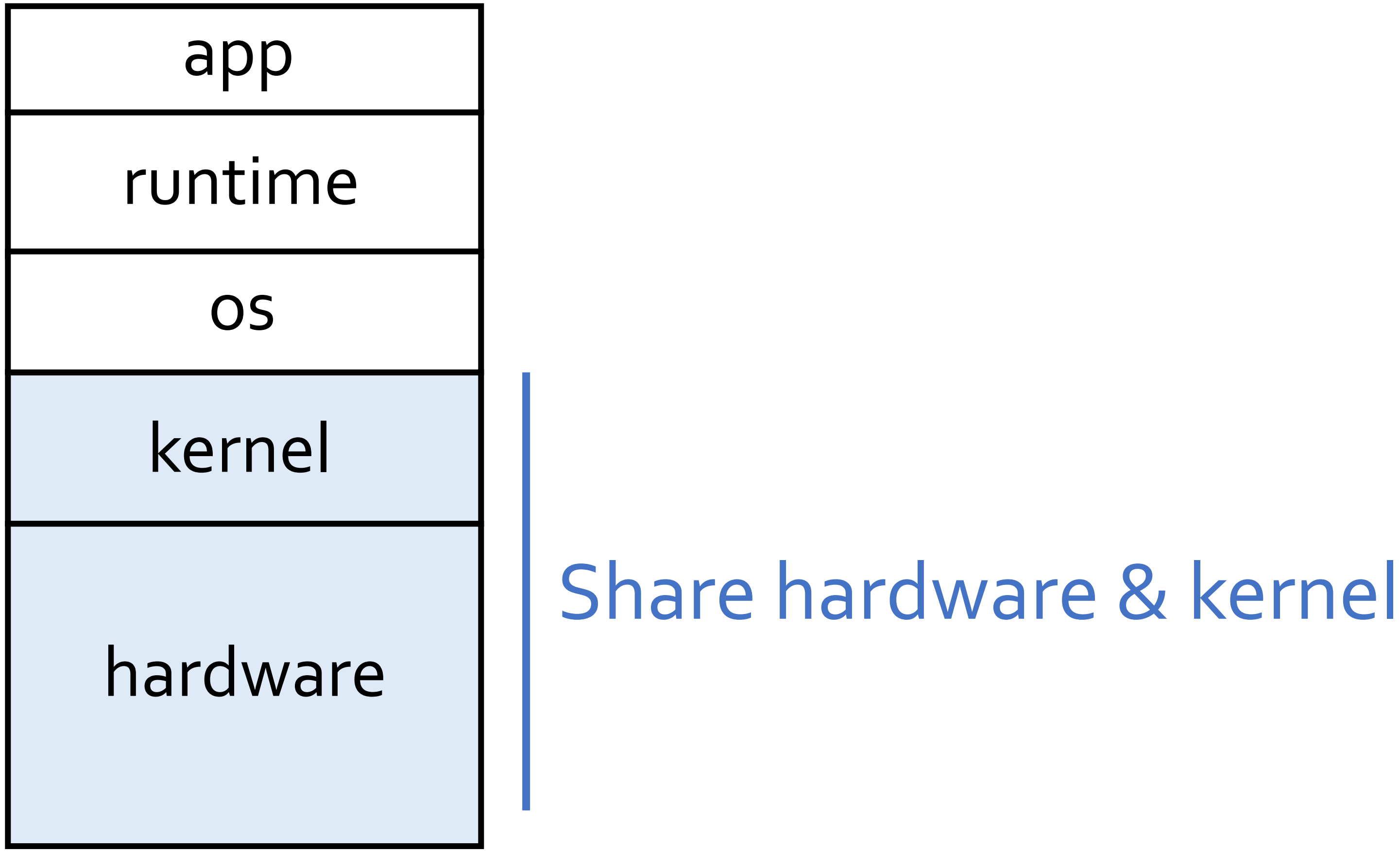


This is a VM Image

Isolation starts at kernel

Share hardware

Different Isolation Points



app

runtime

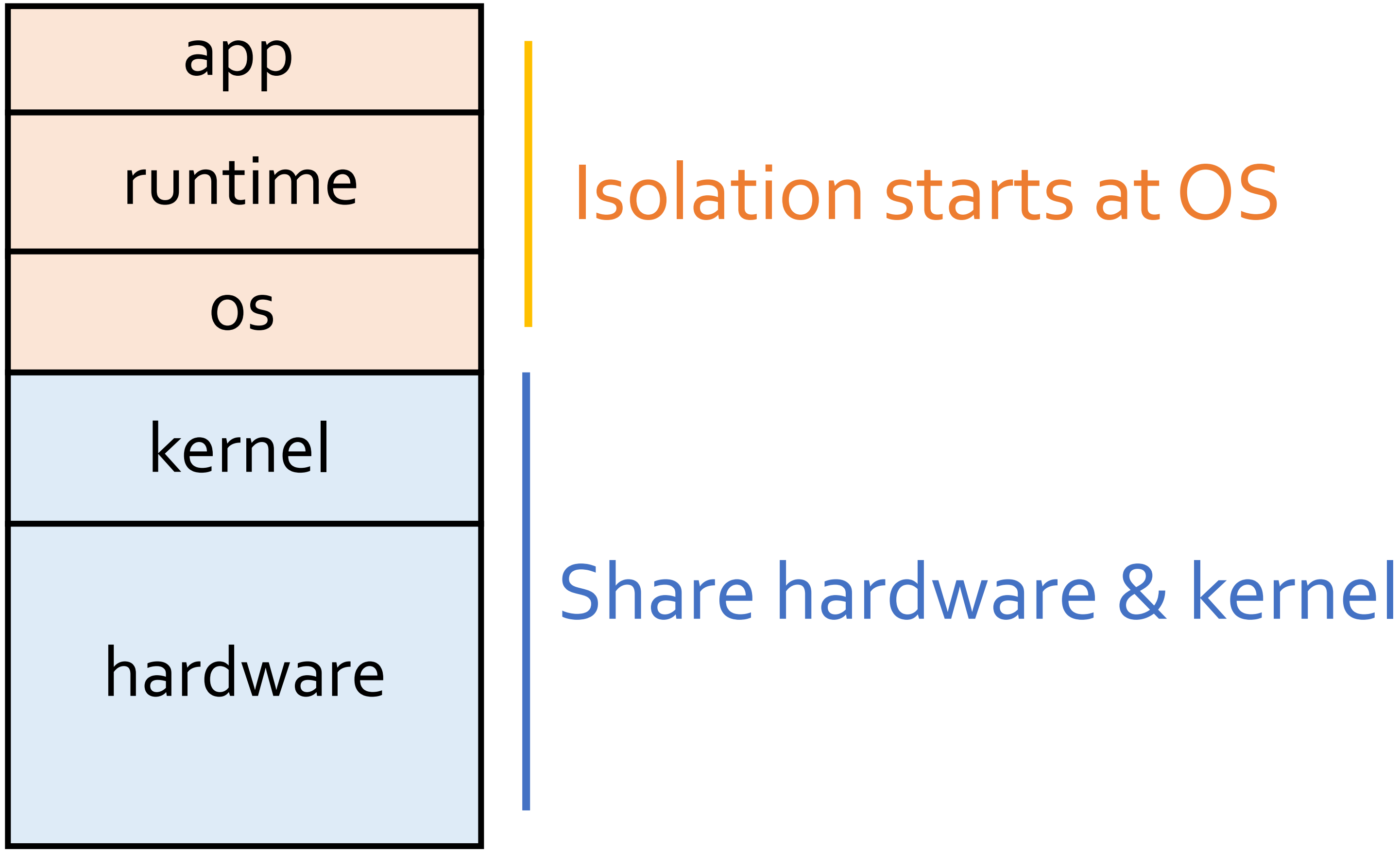
os

kernel

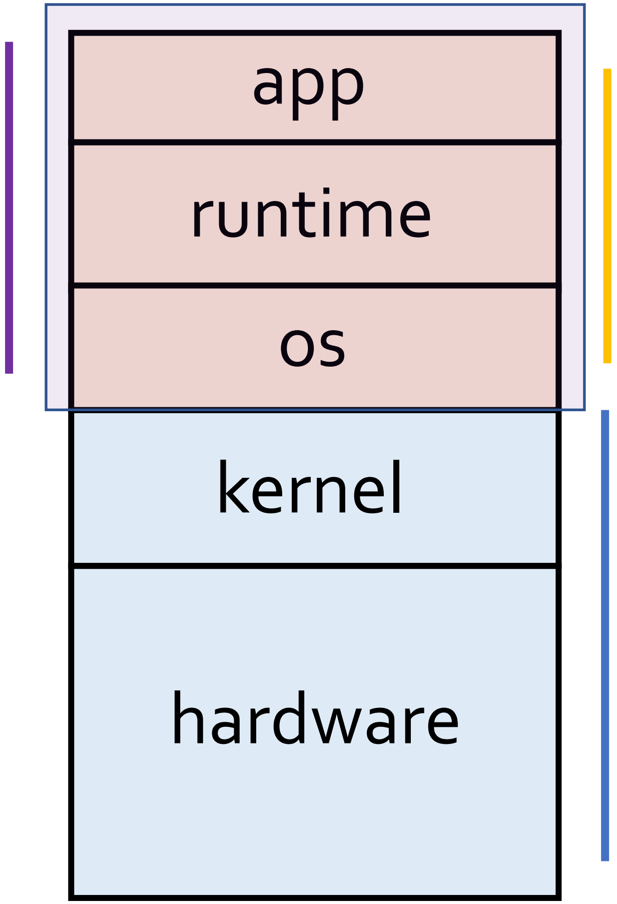
hardware

Share hardware & kernel

Different Isolation Points



Different Isolation Points

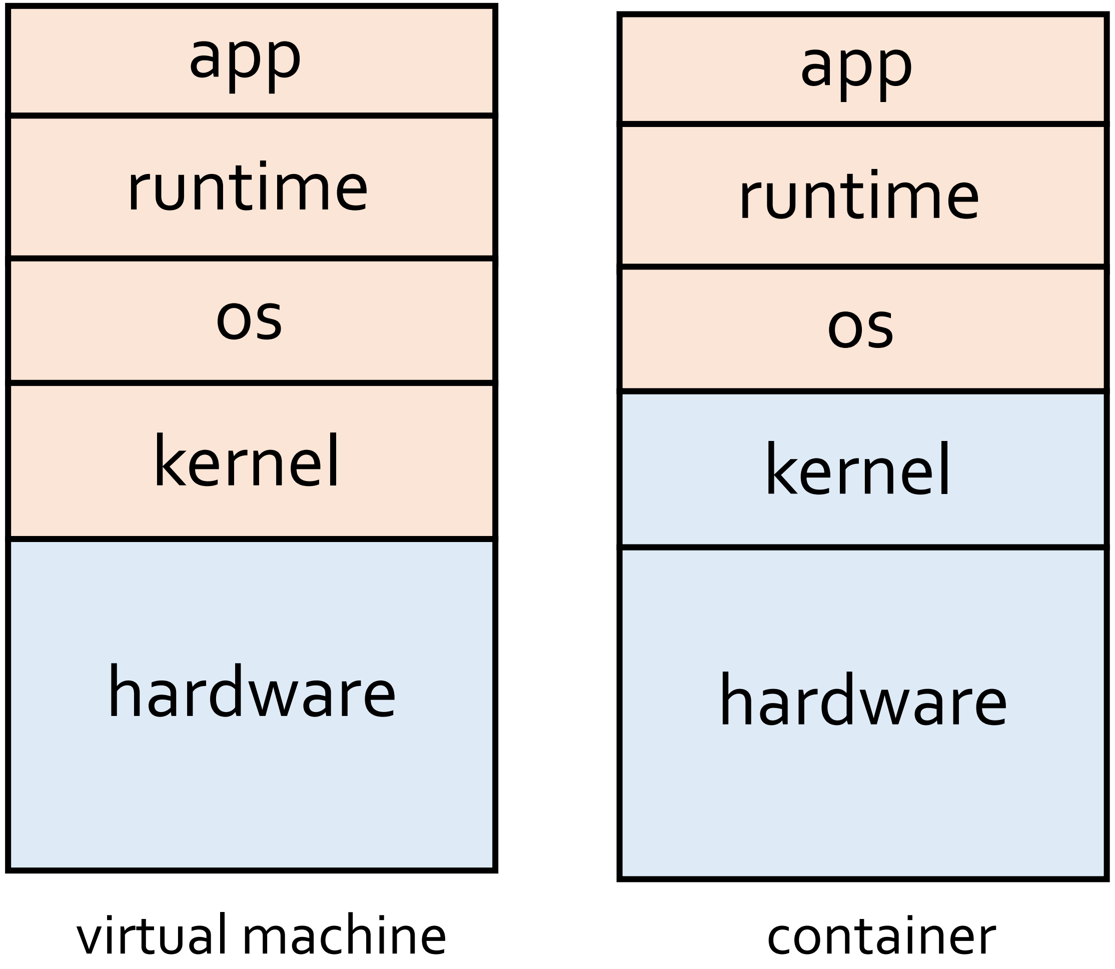


This is a container image

Isolation starts at OS

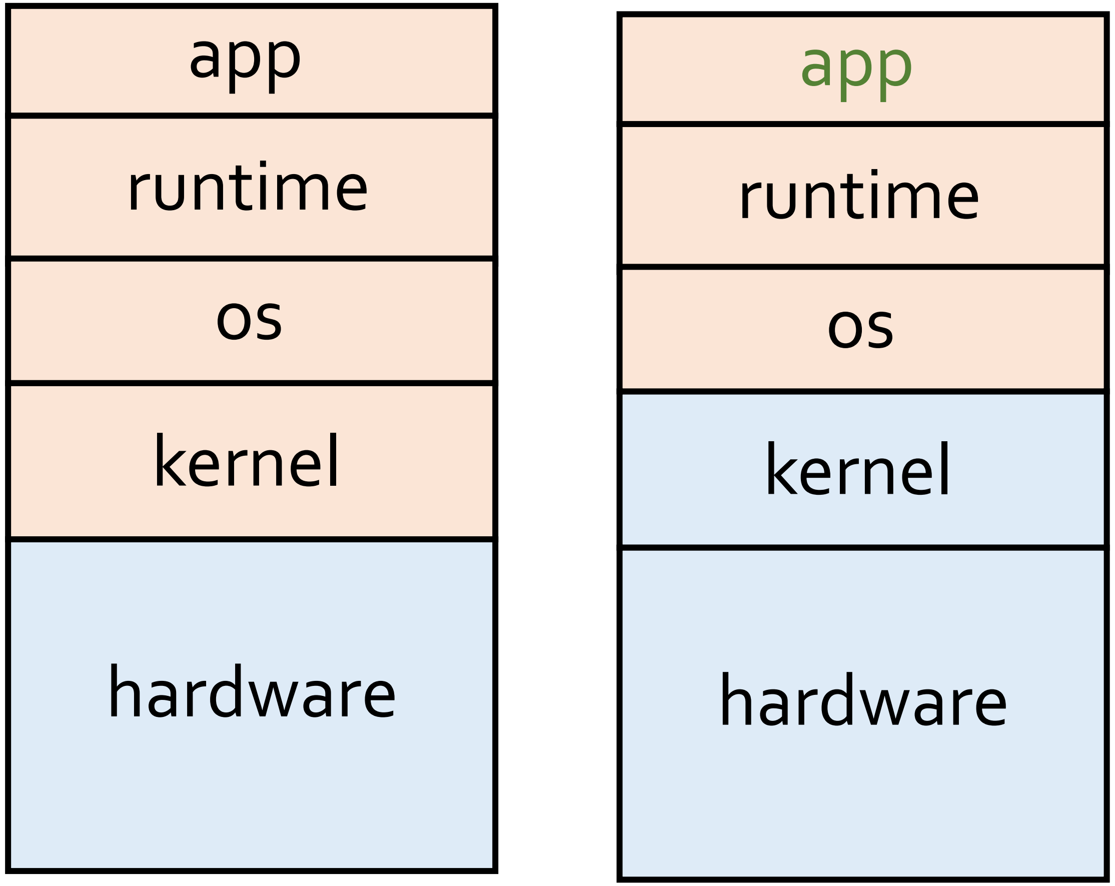
Share hardware & kernel

Container Benefits



- Lightweight
  (megabytes vs gigabytes [10x])
- Fast startup
  (milliseconds vs. minutes [100x])
- Simple to build, deploy, send, & maintain

Benefits come from specialization



virtual machine

container

- Lightweight

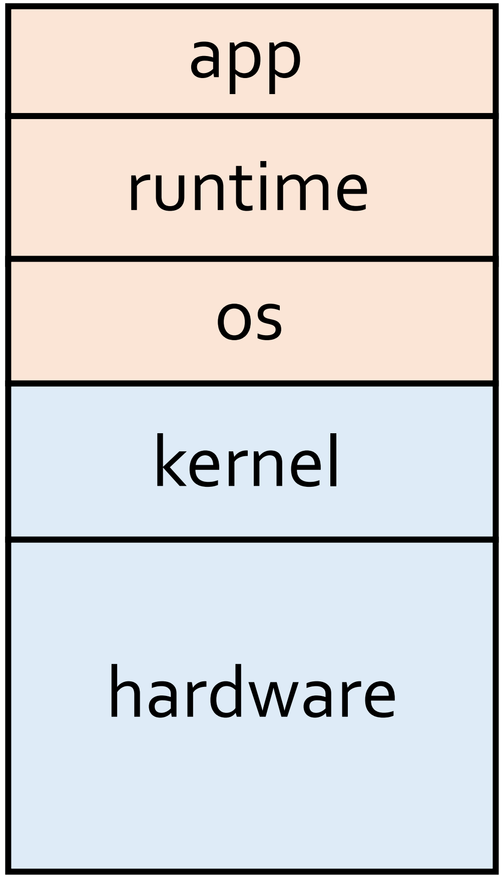
  (megabytes vs gigabytes [10x])

- Fast startup

  (milliseconds vs. minutes [100x])

- Simple to build, deploy, send, & maintain

Benefits come from specialization



app

runtime

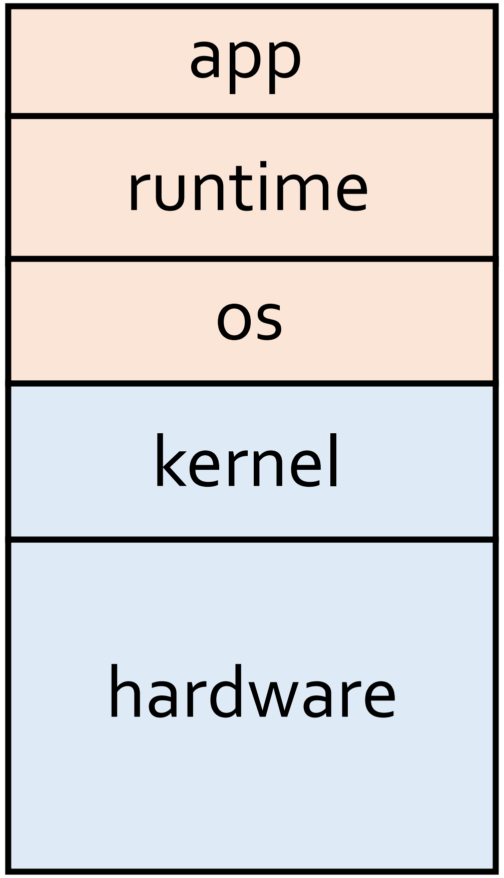
os

kernel

hardware

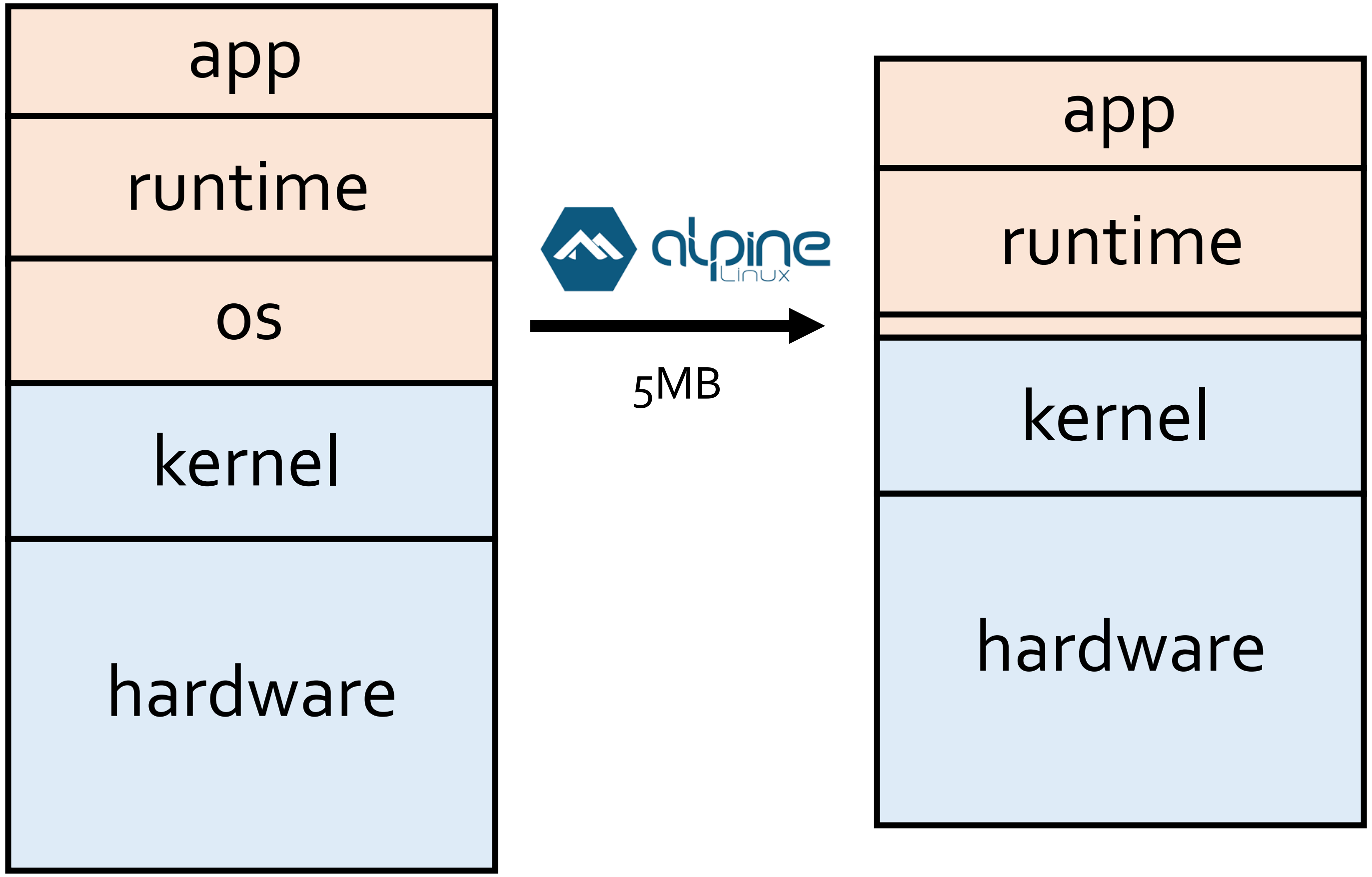
Built to support all applications and users

Benefits come from specialization

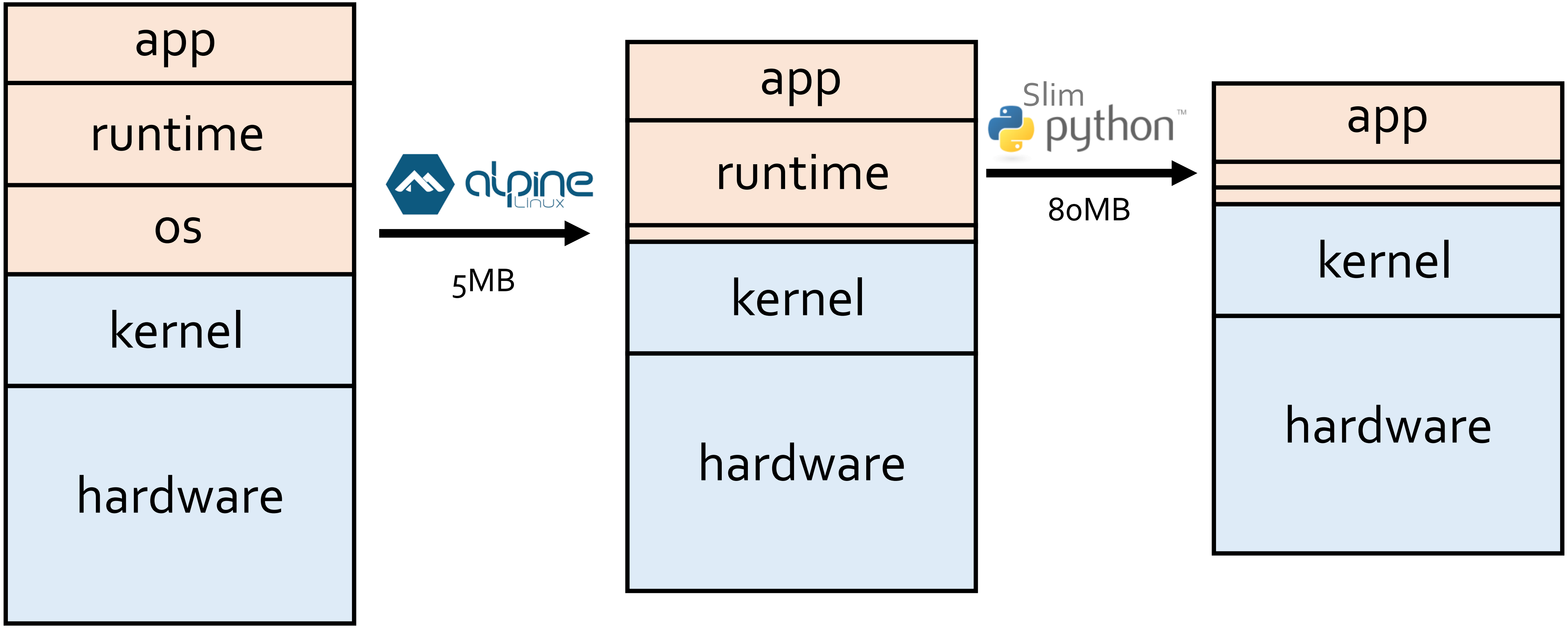


But we only care about a single application

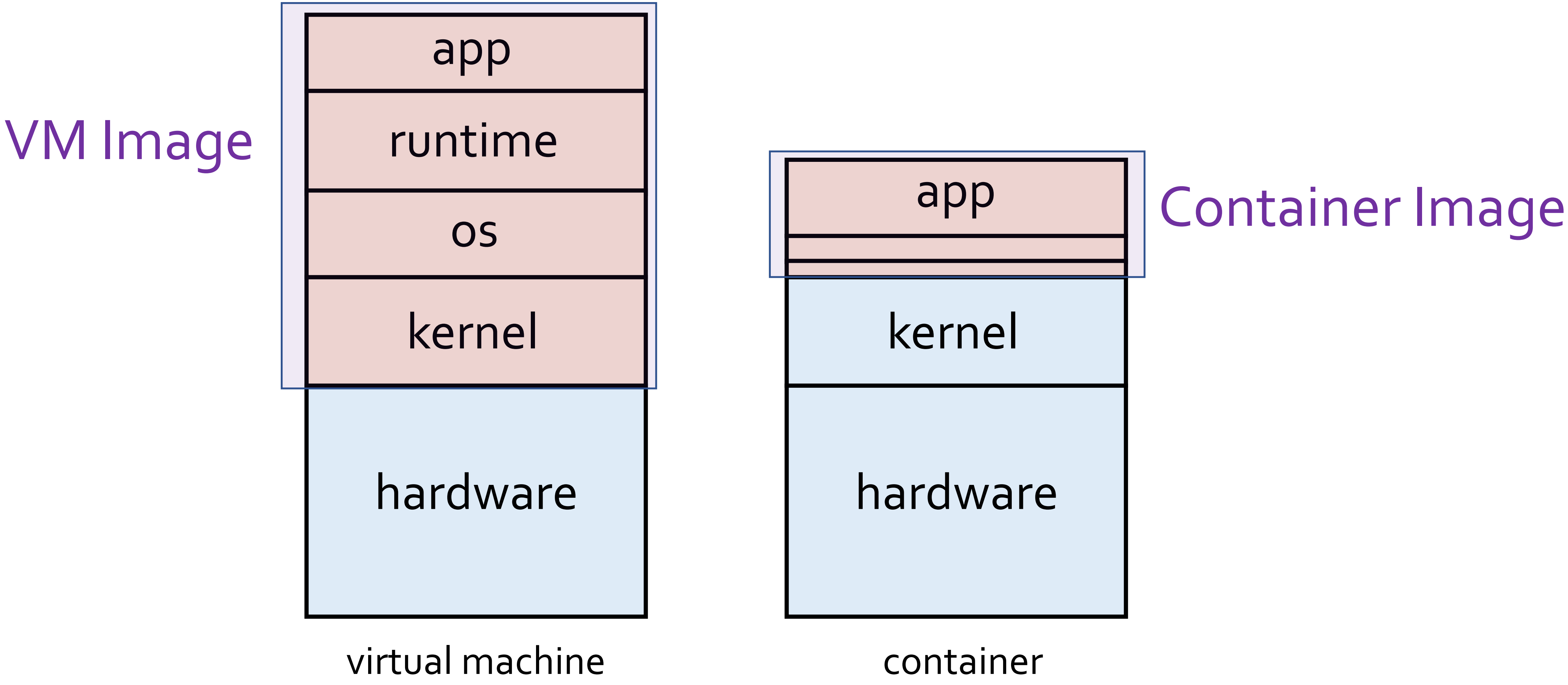
Benefits come from specialization



Benefits come from specialization

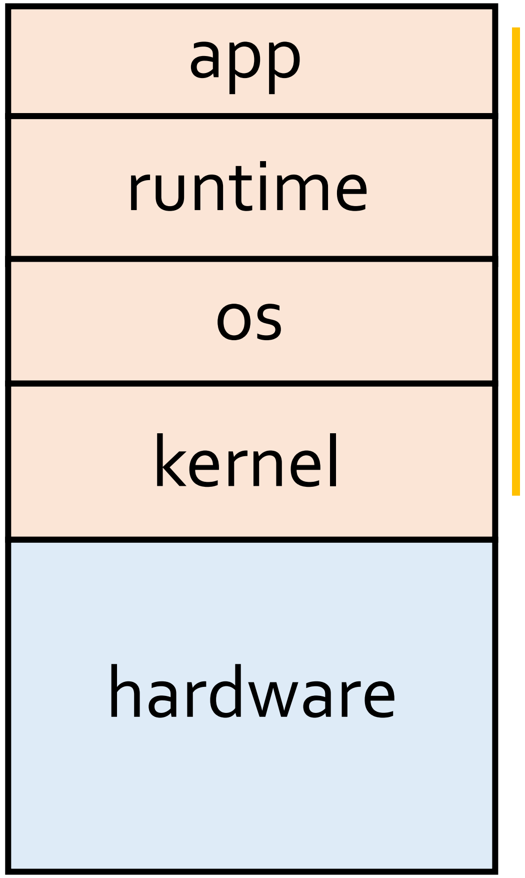


Container image is mostly application



Containers to Serverless Computing

Traditional Cloud Model


app

runtime

os

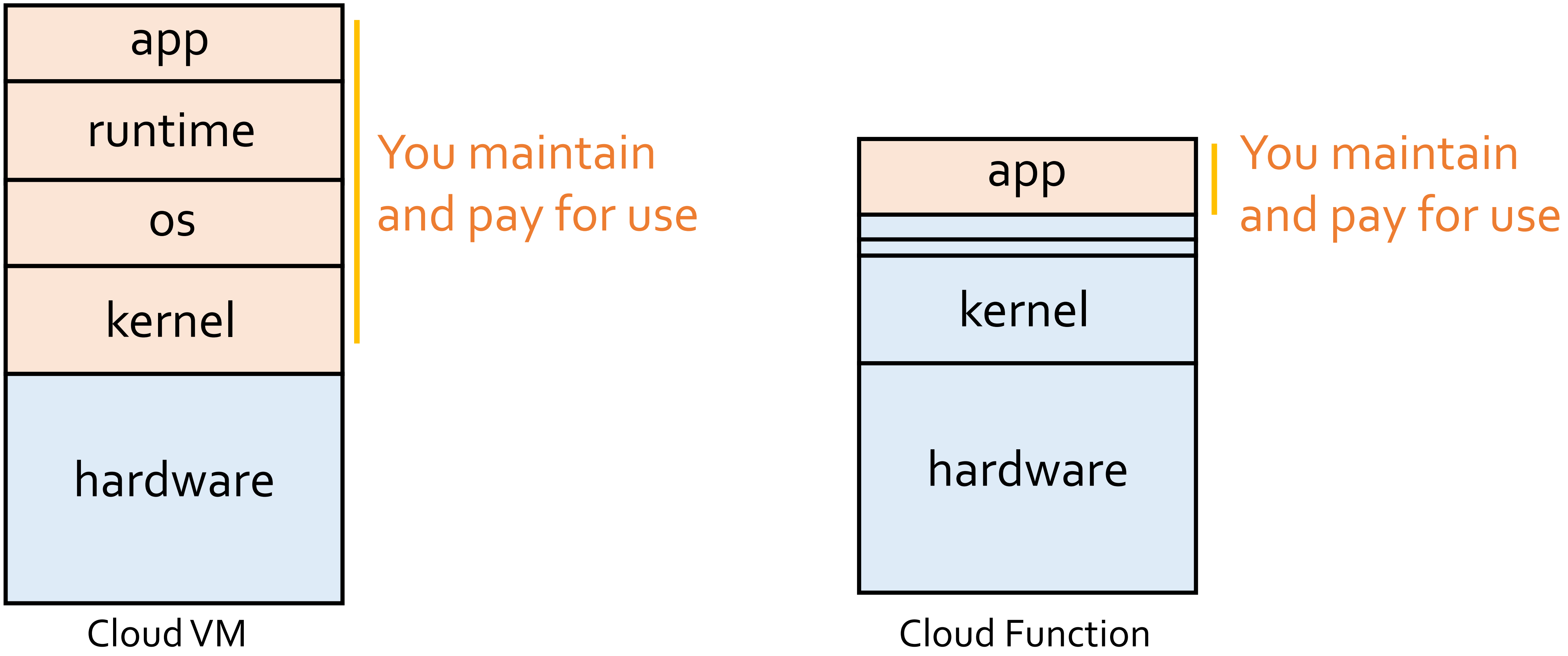
kernel

hardware

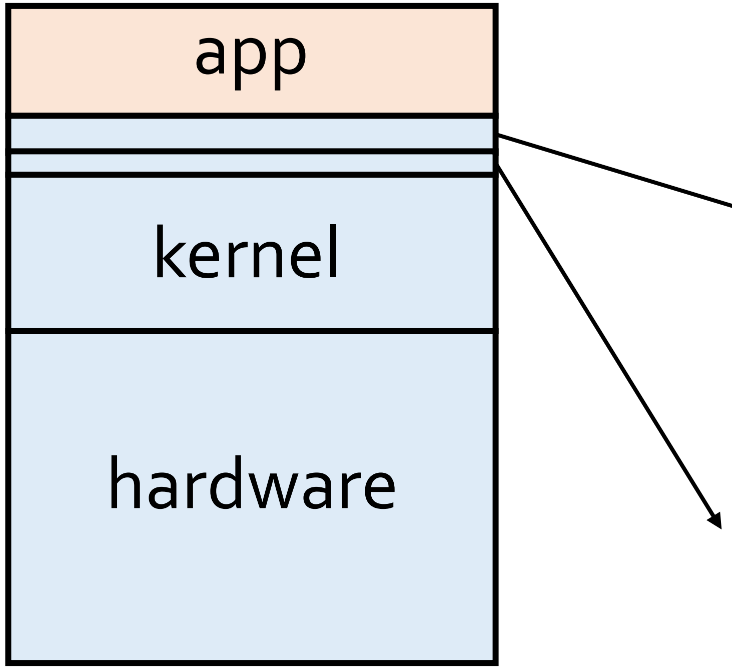
You maintain
and pay for use

Cloud VM

Serverless Cloud Model



OS and runtime are partially managed by cloud providers

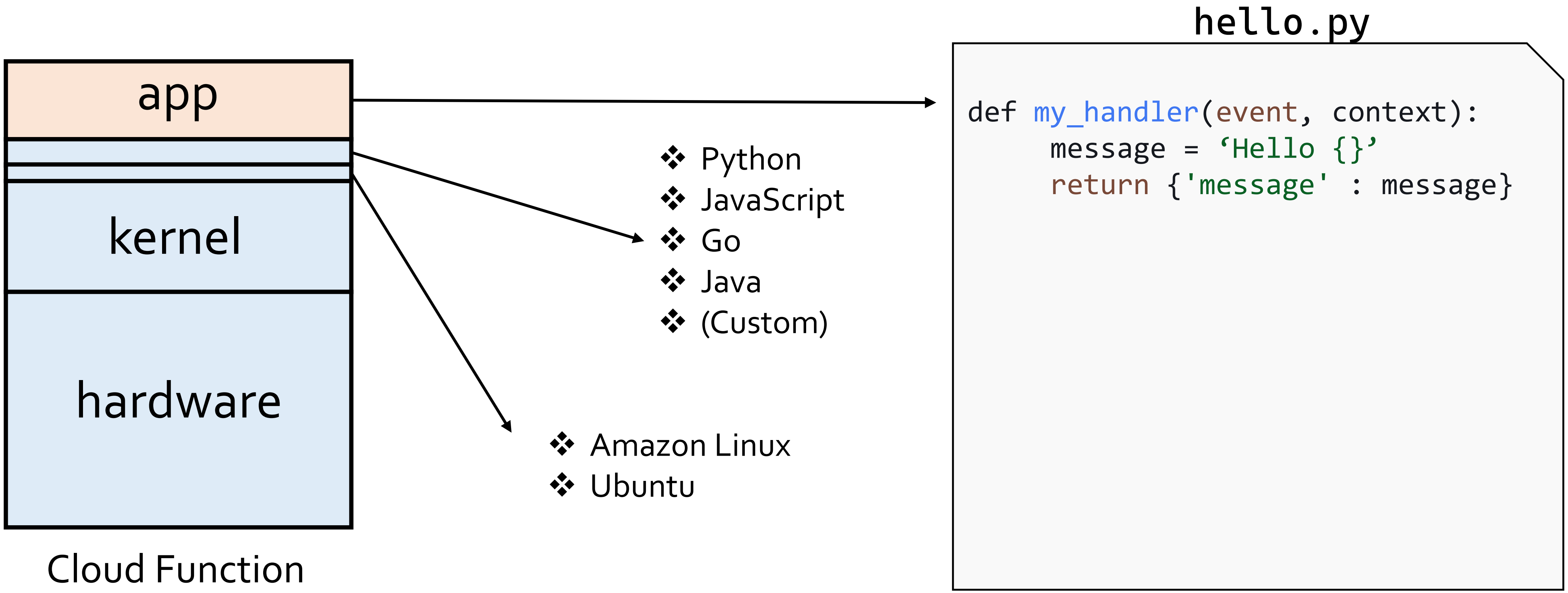


Cloud Function

- ❖ Python
- ❖ JavaScript
- ❖ Go
- ❖ Java
- ❖ (Custom)

- ❖ Amazon Linux
- ❖ Ubuntu

OS and runtime are partially managed by cloud providers



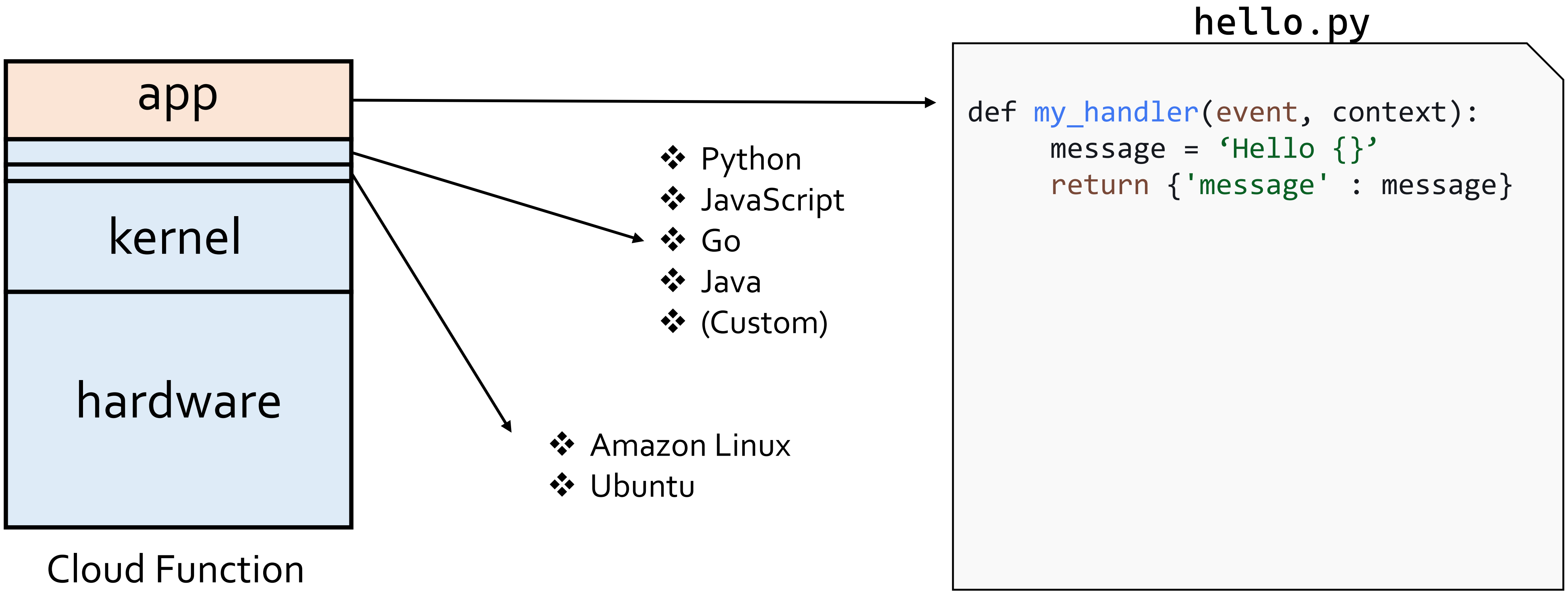
Cloud Function

- ❖ Python
- ❖ JavaScript
- ❖ Go
- ❖ Java
- ❖ (Custom)

- ❖ Amazon Linux
- ❖ Ubuntu

```
def my_handler(event, context):
     message = ‘Hello {}’
     return {'message' : message}
```

Why would we do this?



- ❖ Python
- ❖ JavaScript
- ❖ Go
- ❖ Java
- ❖ (Custom)

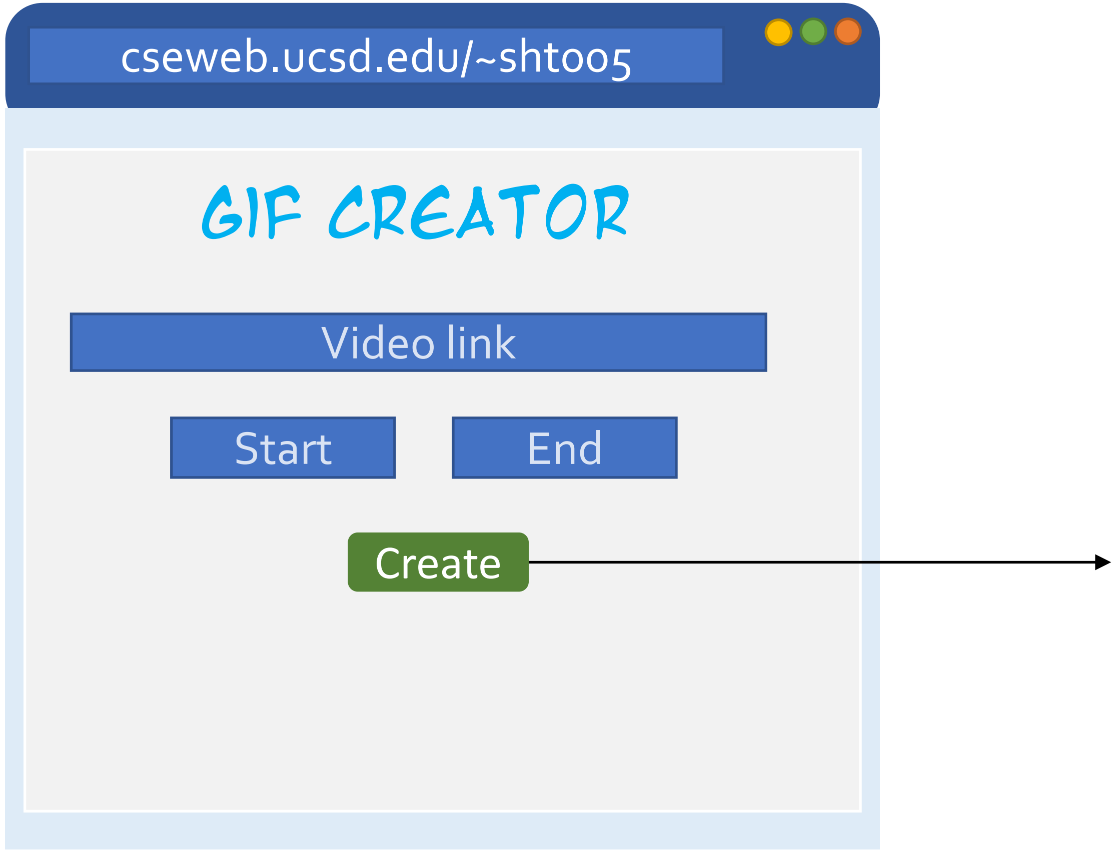
- ❖ Amazon Linux
- ❖ Ubuntu

```
def my_handler(event, context):
      message = ‘Hello {}’
      return {'message' : message}
```

Building a GIF Creator

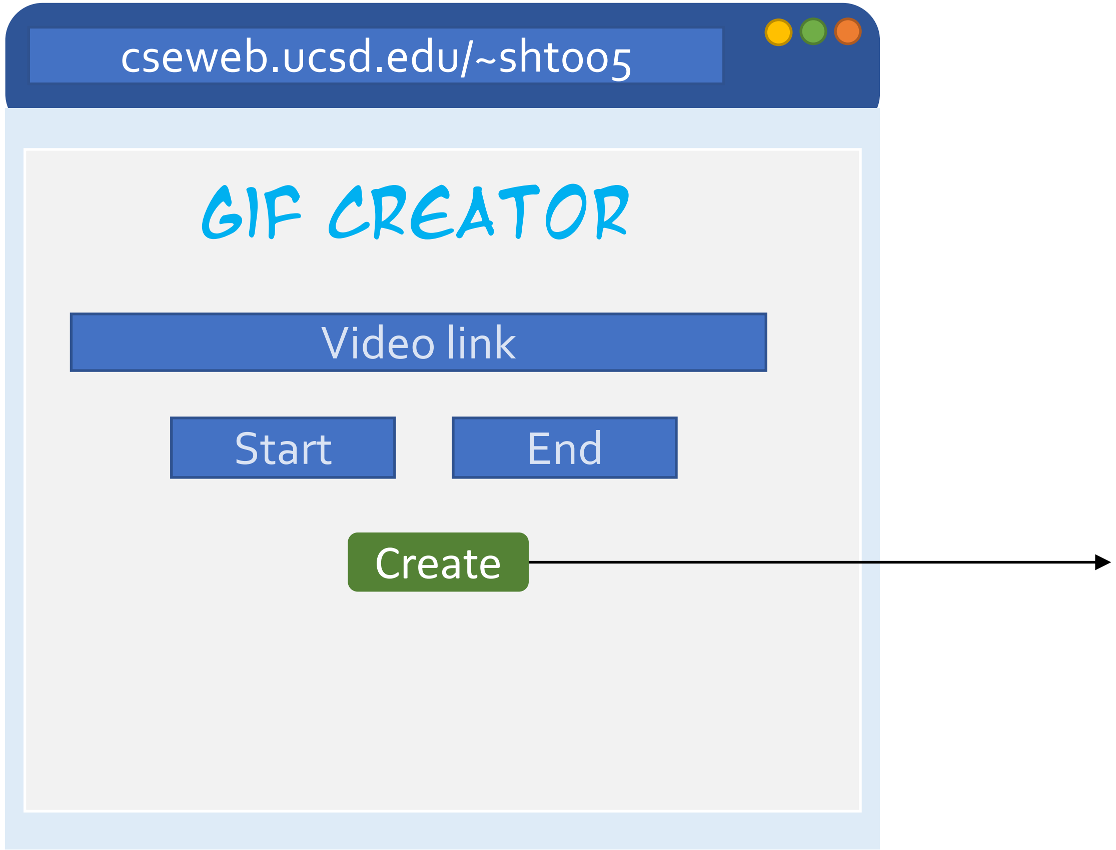
cseweb.ucsd.edu/~shto05
GIF CREATOR
Video link
Start
End
Create

Building a GIF Creator



1. Download the video locally
2. Clip the video
3. Compress
4. Convert to gif

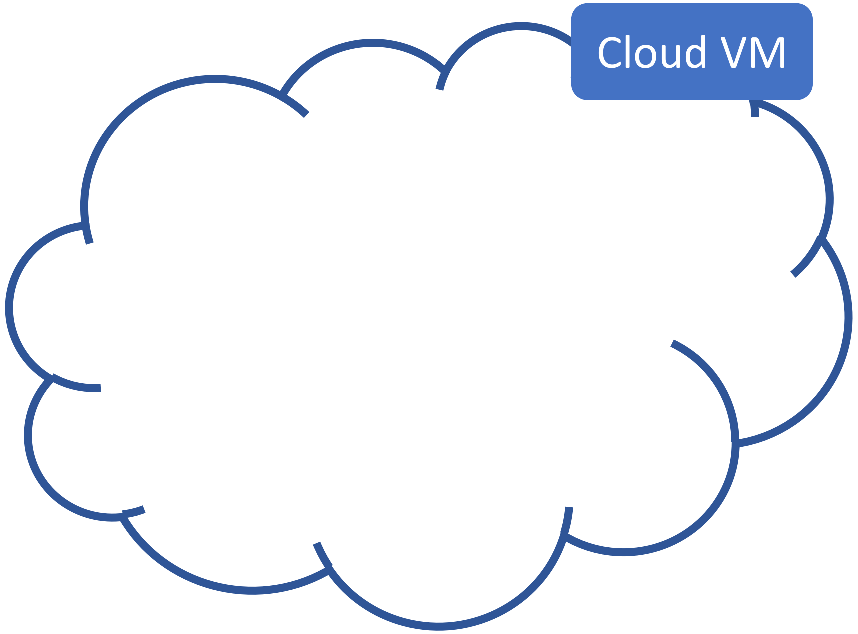
Hosting server may be too slow to do this or cannot do this



1. Download the video locally
2. Clip the video
3. Compress
4. Convert to gif

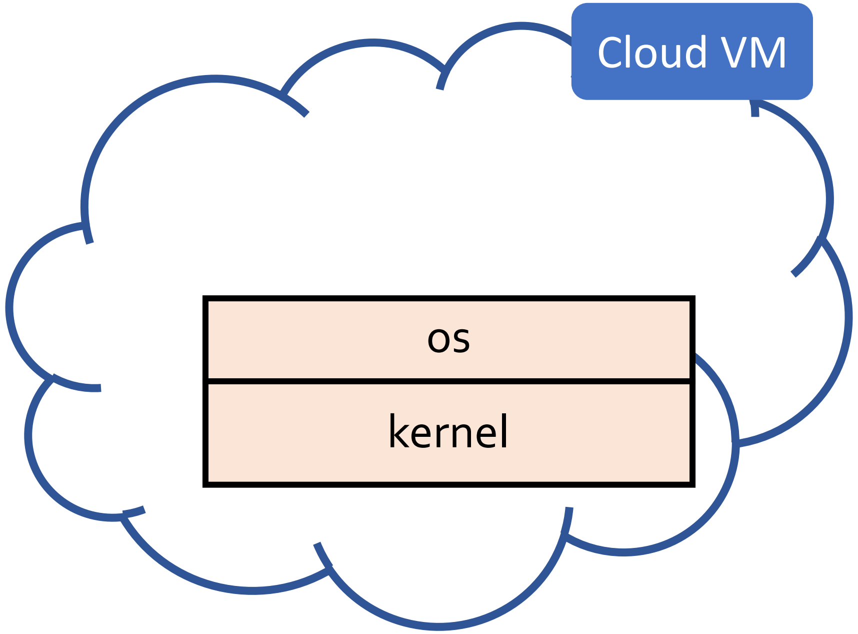
With VMs we need to provision a machine

cseweb.ucsd.edu/~shtoo5
GIF CREATOR
Video link
Start
End
Create



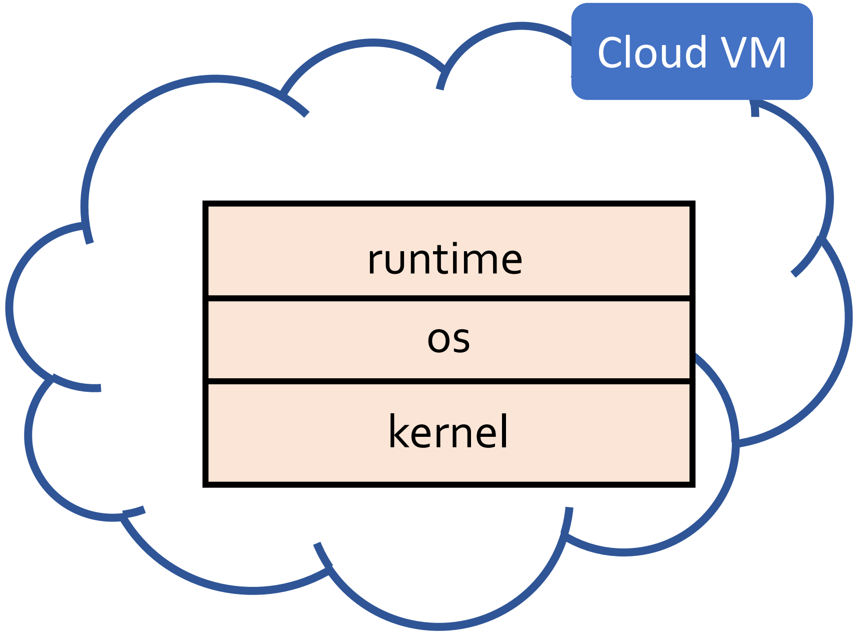
Setup the OS and kernel

cseweb.ucsd.edu/~sht005
GIF CREATOR
Video link
Start
End
Create



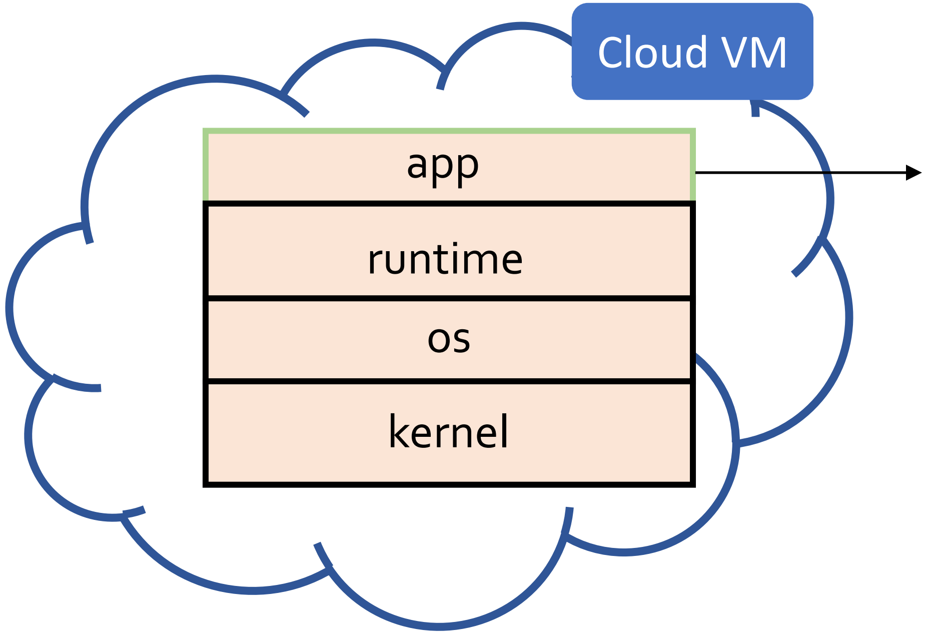
Setup a custom runtime

cseweb.ucsd.edu/~shtoo5
GIF CREATOR
Video link
Start
End
Create



Finally, we write our application

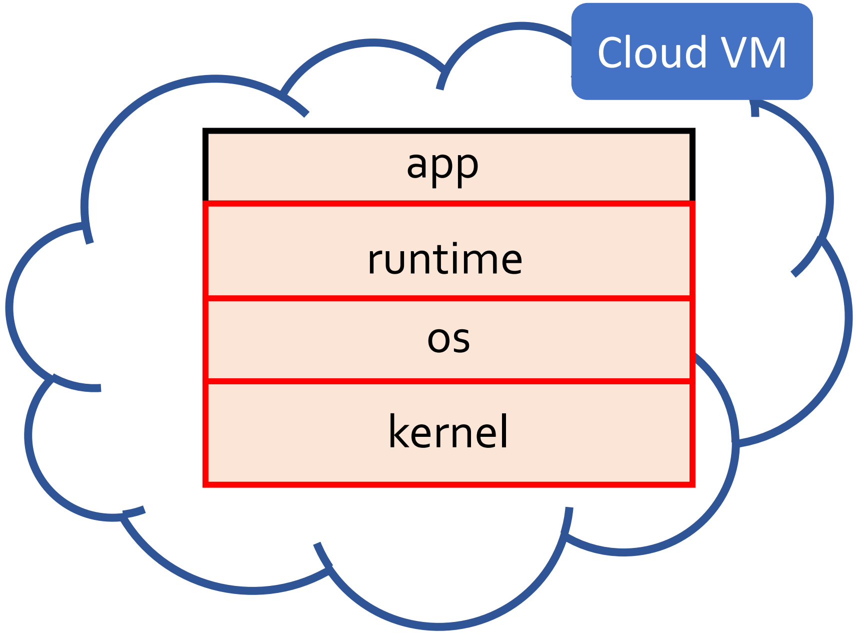
cseweb.ucsd.edu/~shtoo5
GIF CREATOR
Video link
Start
End
Create



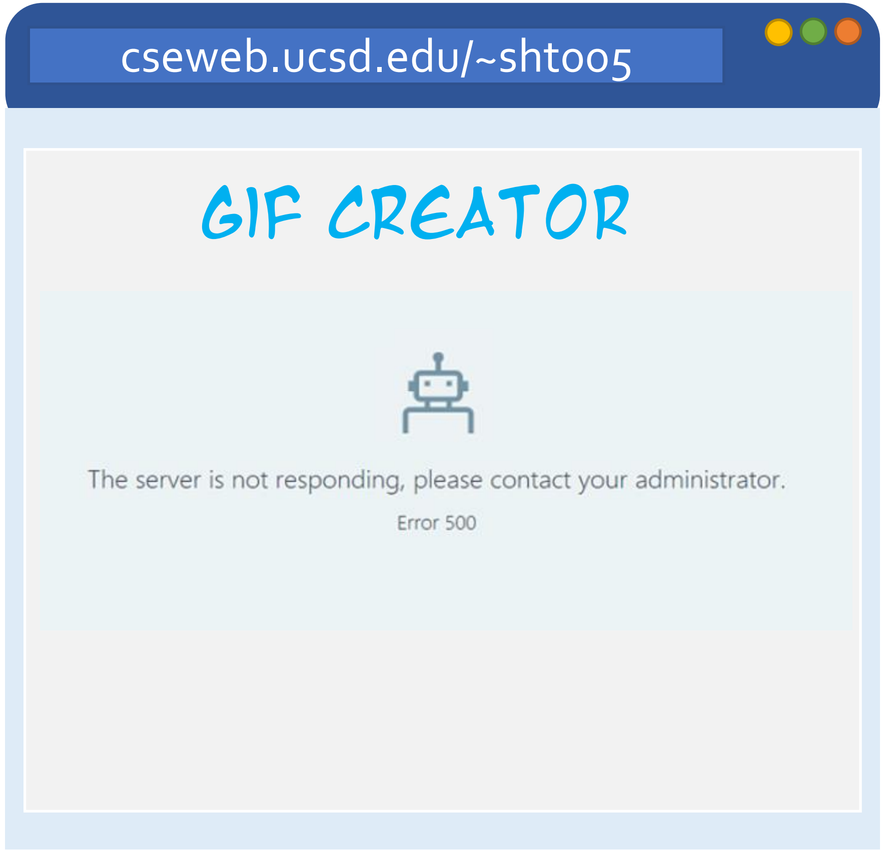
1. Download the video locally
2. Clip the video
3. Compress
4. Convert to gif

But if anything breaks

cseweb.ucsd.edu/~shtoo5
GIF CREATOR
Video link
Start
End
Create



The application stops working

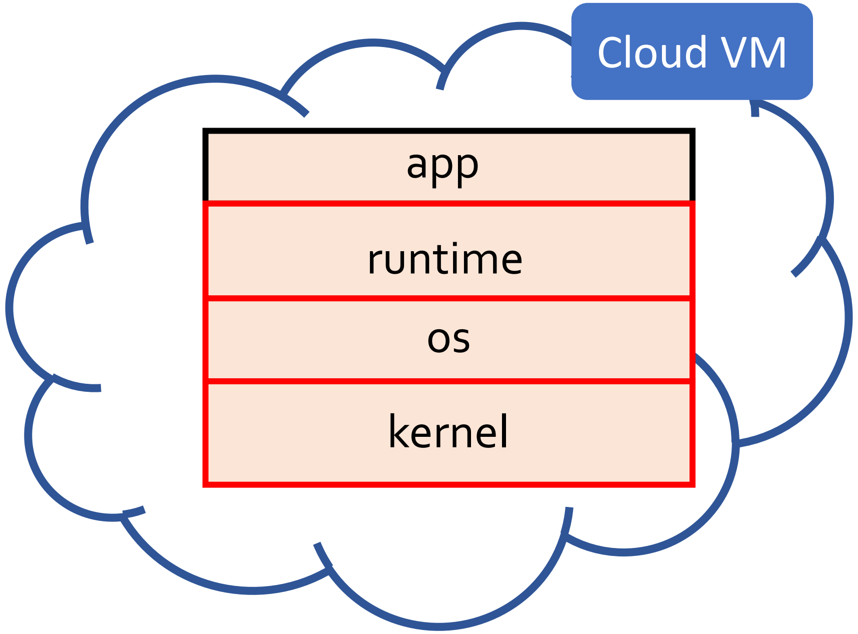


cseweb.ucsd.edu/~shtoo5

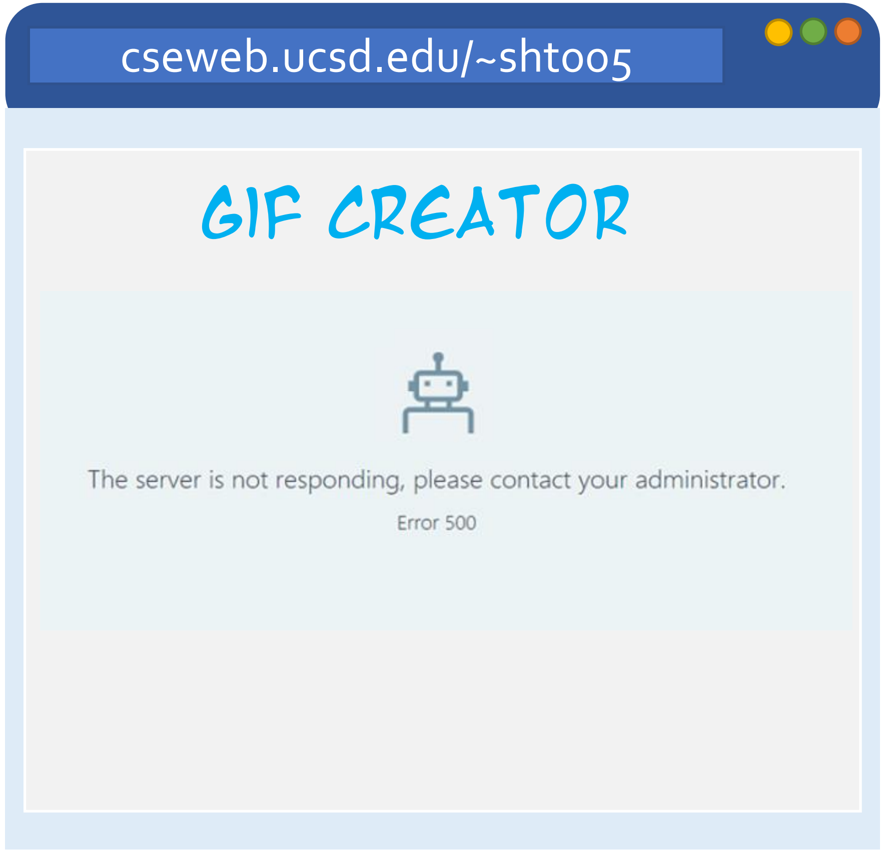
GIF CREATOR

The server is not responding, please contact your administrator.

Error 500



Managing this has high operational overhead

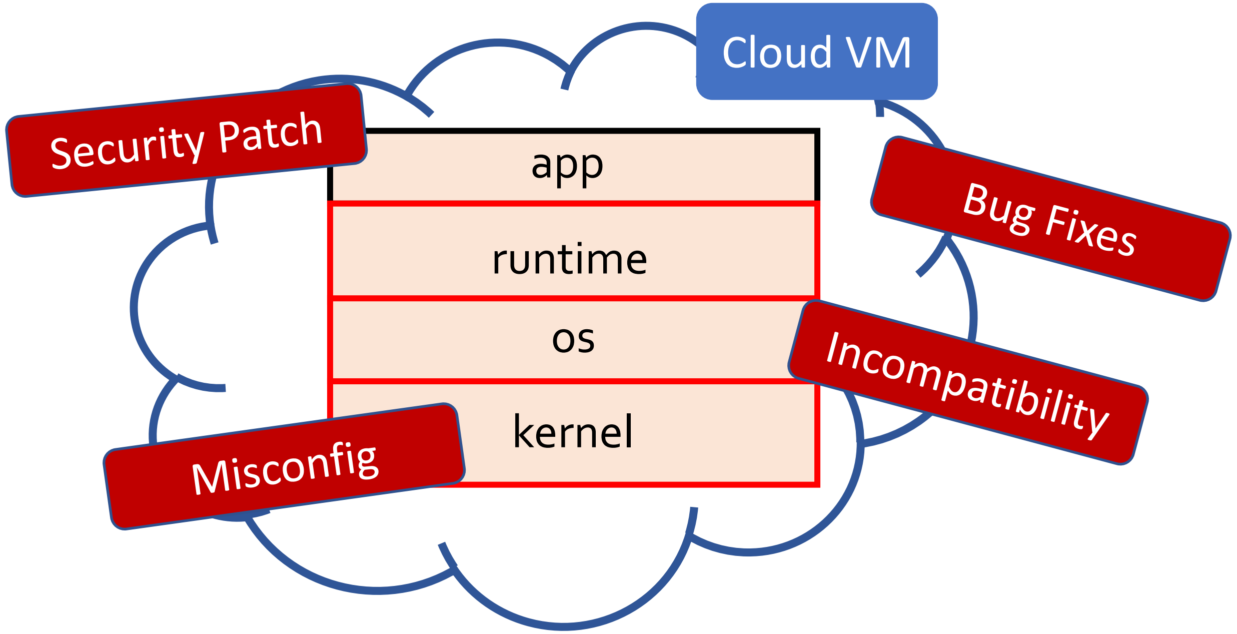


cseweb.ucsd.edu/~shtoo5

GIF CREATOR

The server is not responding, please contact your administrator.

Error 500



Cloud VM

app

runtime

os

kernel

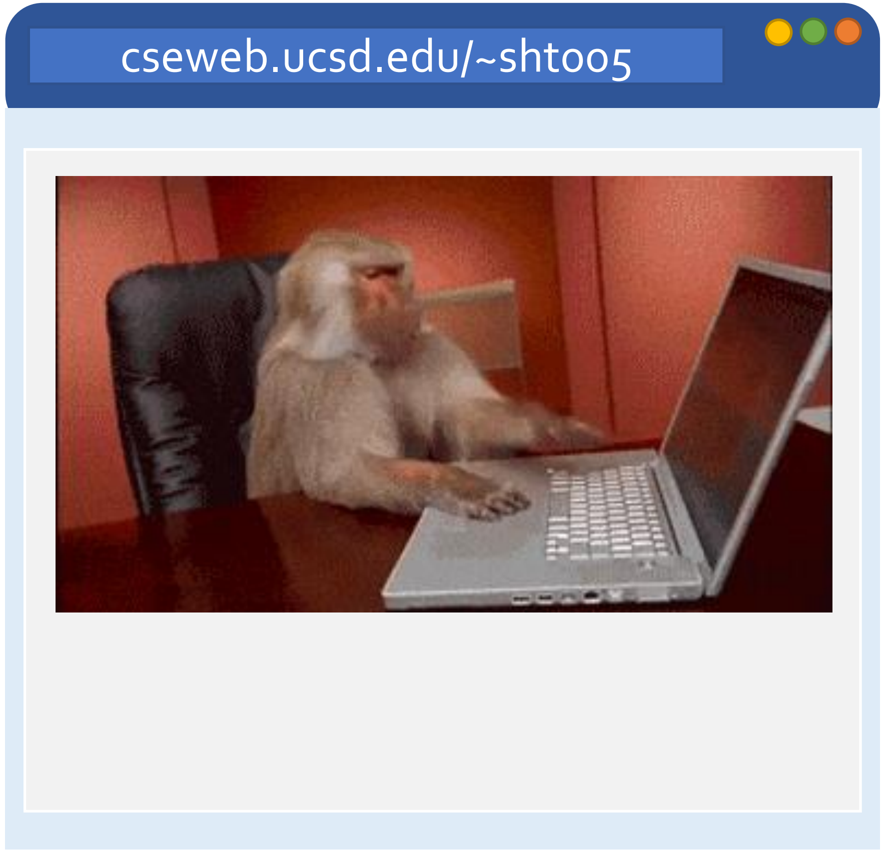
Security Patch

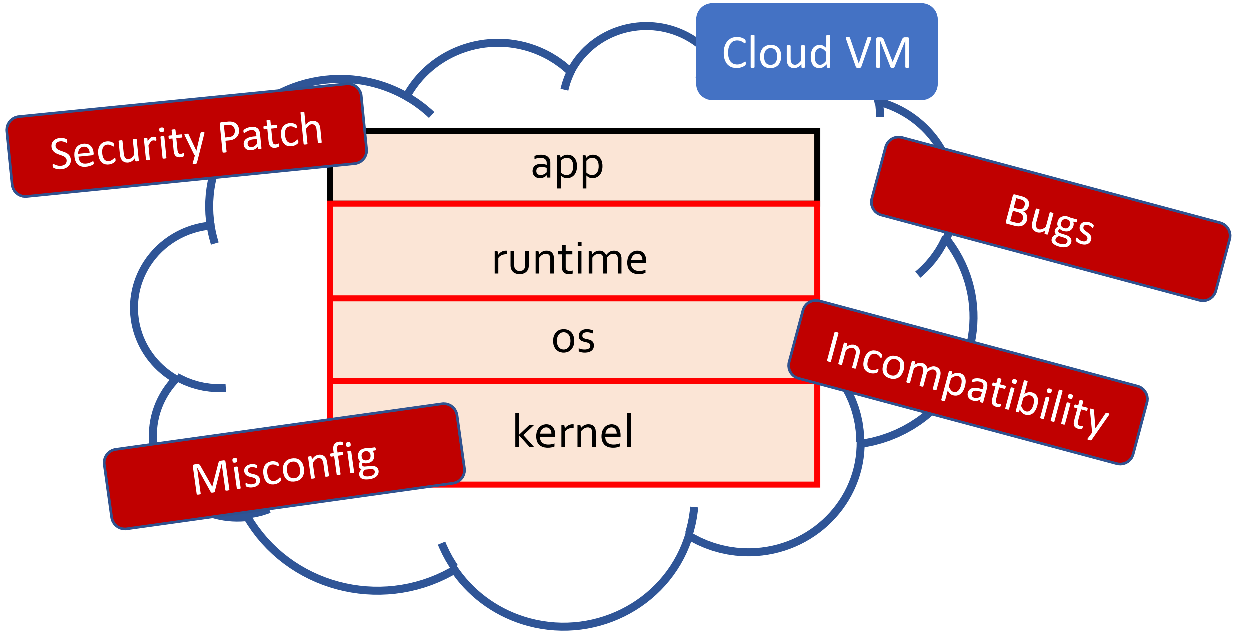
Bug Fixes

Incompatibility

Misconfig

Managing this has high operational overhead





Cloud VM

app

runtime

os

kernel

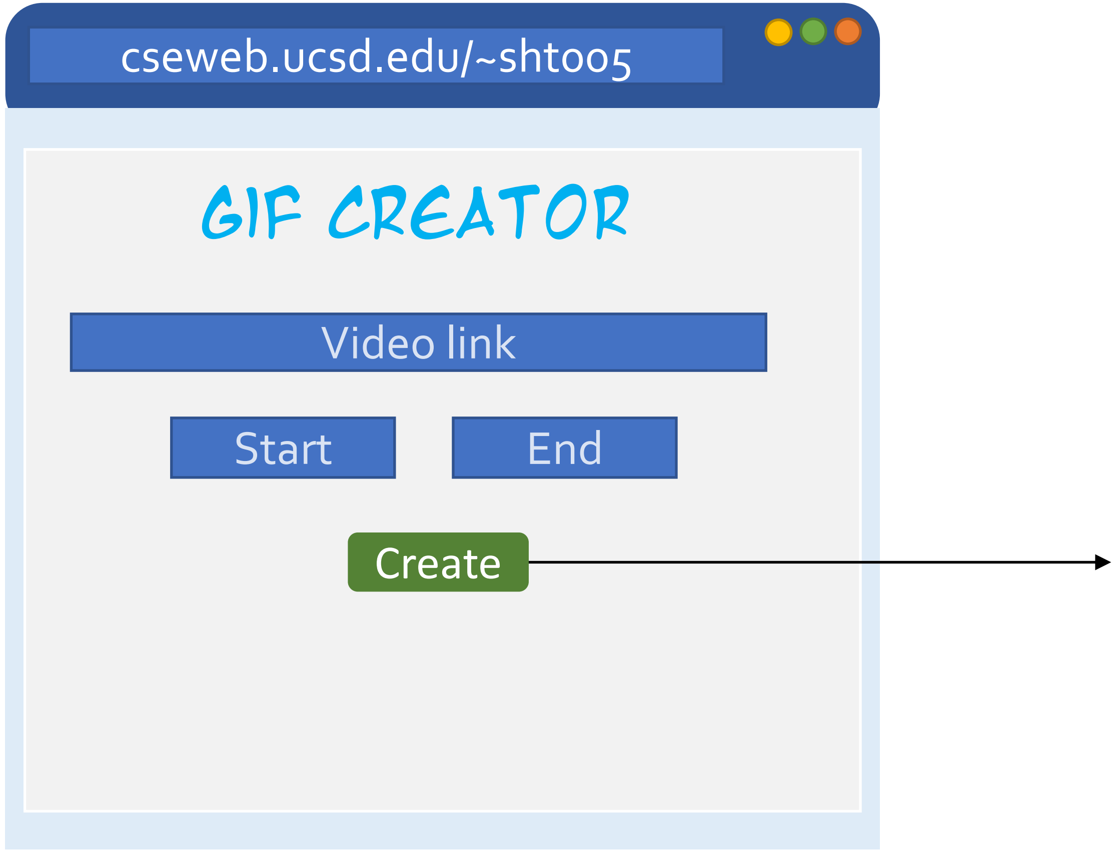
Security Patch

Bugs

Incompatibility

Misconfig

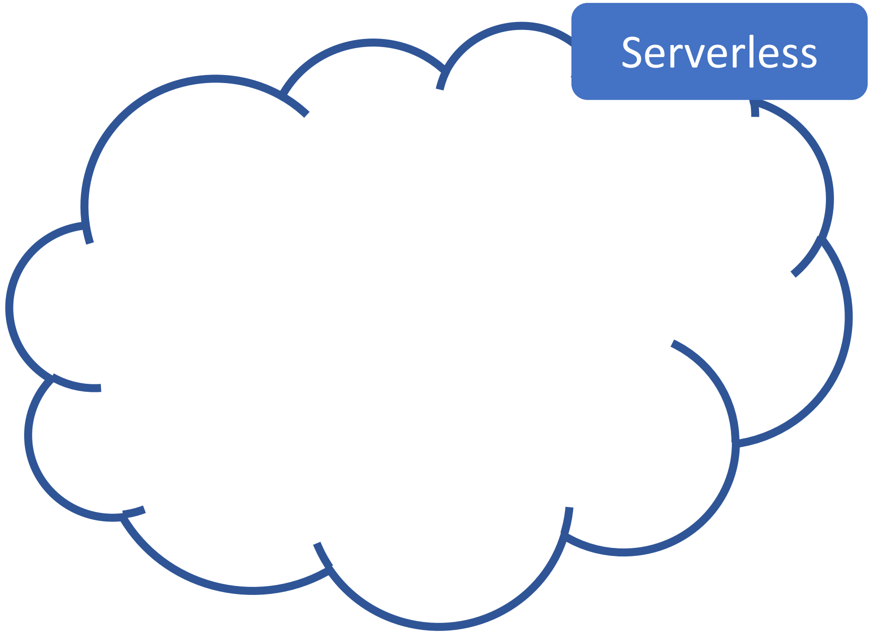
Is there a better way?



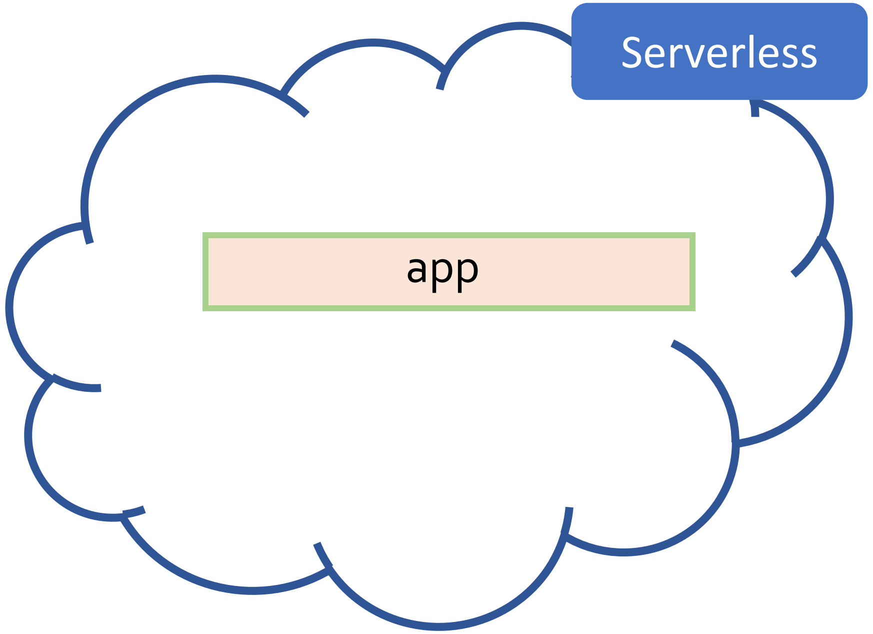
1. Download the video locally
2. Clip the video
3. Compress
4. Convert to gif

Deploy a serverless function

cseweb.ucsd.edu/~shtoo5
GIF CREATOR
Video link
Start
End
Create

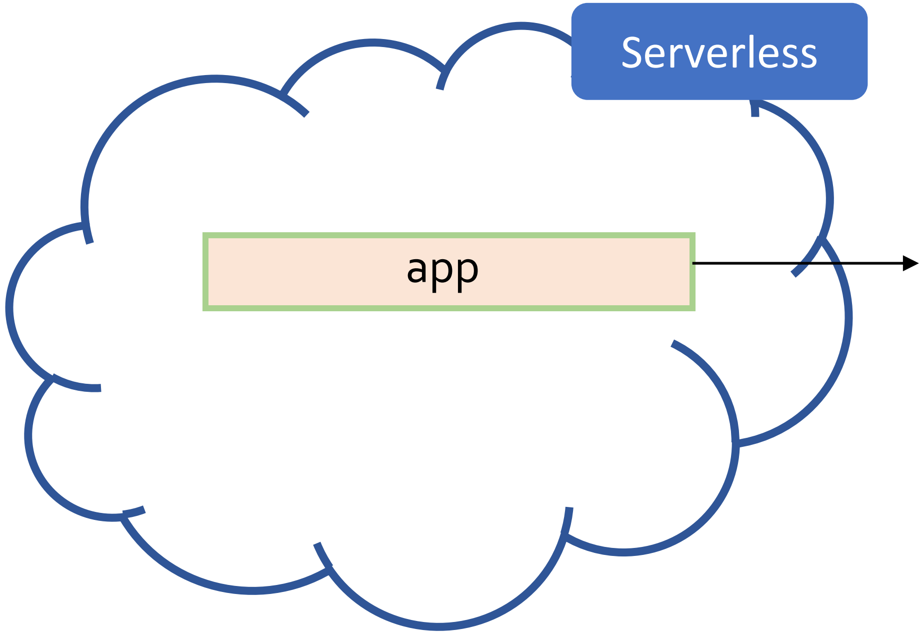


Write the application, specify dependencies



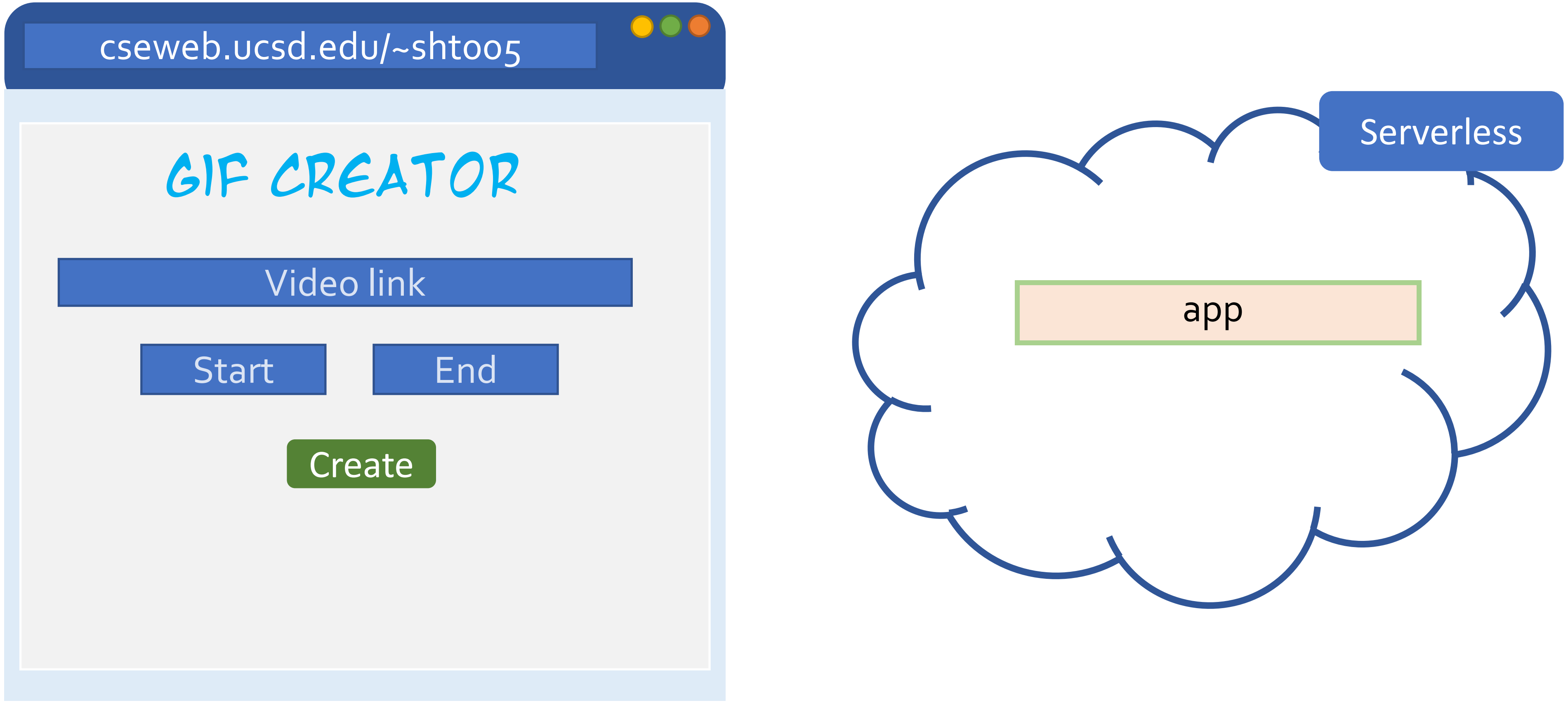
Only need to worry about application bugs

cseweb.ucsd.edu/~shtoo5
GIF CREATOR
Video link
Start
End
Create



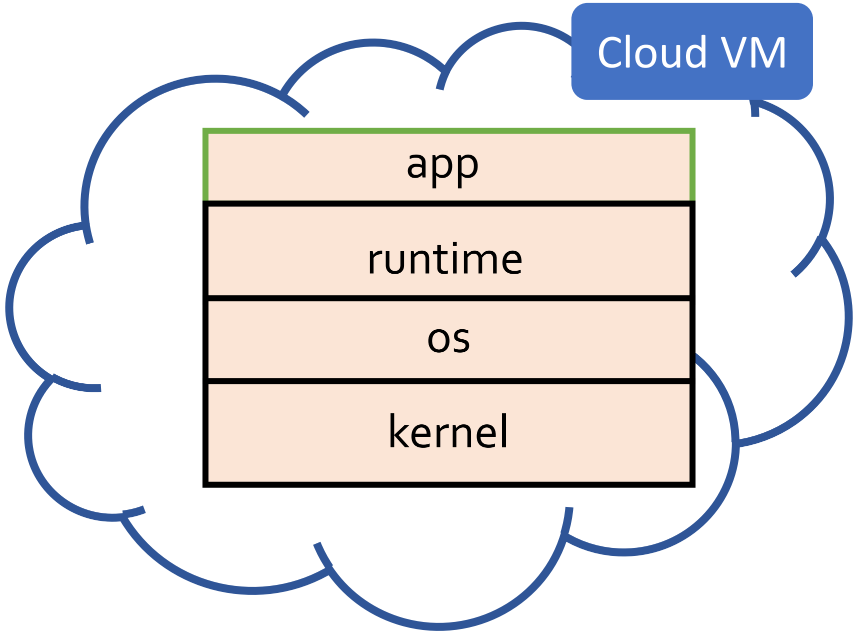
1. Download the video locally
2. Clip the video
3. Compress
4. Convert to gif

Serverless functions run on-demand

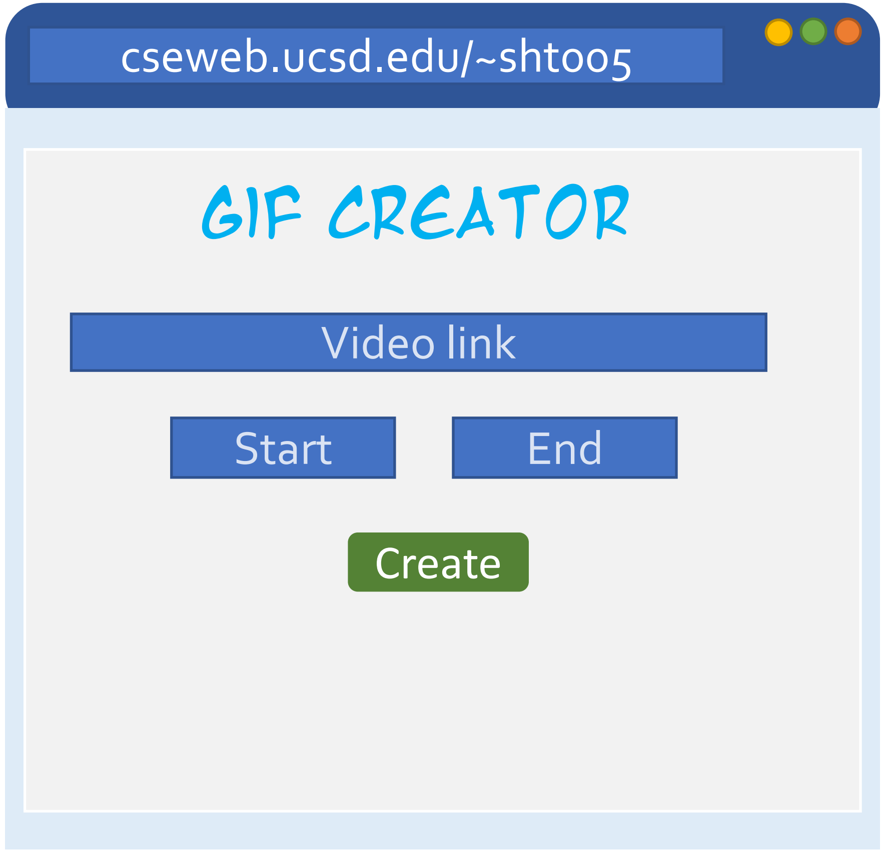


In the Cloud VM the instance is always on

cseweb.ucsd.edu/~sht005
GIF CREATOR
Video link
Start
End
Create



Serverless: New containers are instantiated from scratch every time you click Create



cseweb.ucsd.edu/~sht005

GIF CREATOR

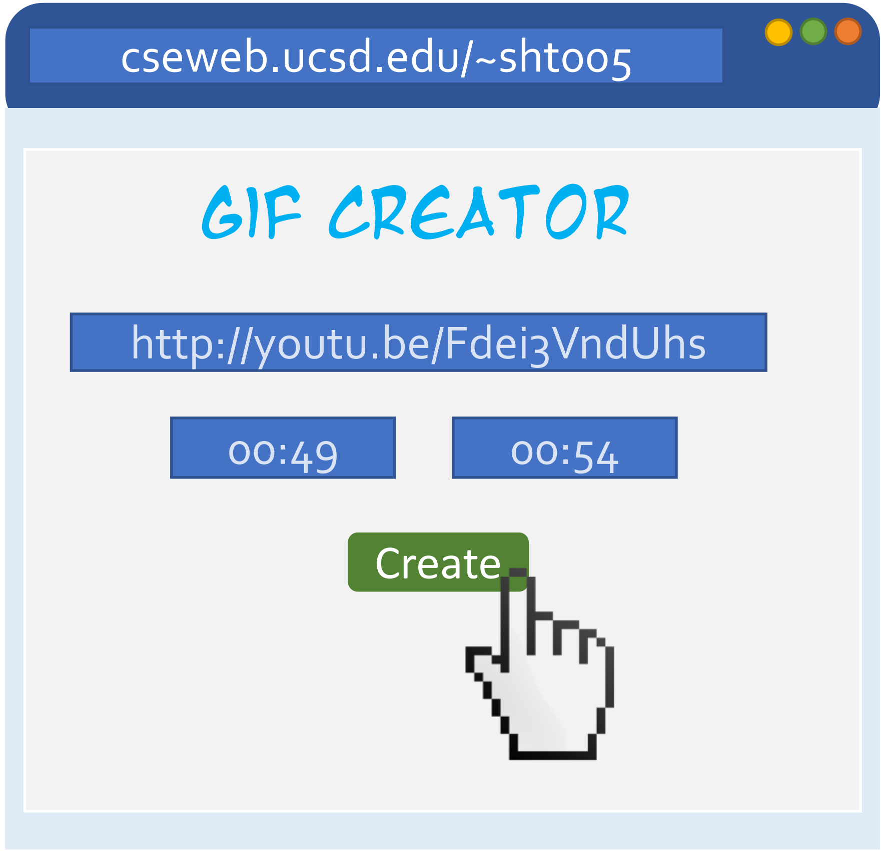
Video link

Start

End

Create

Serverless: New containers are instantiated from scratch every time you click Create



cseweb.ucsd.edu/~sht005

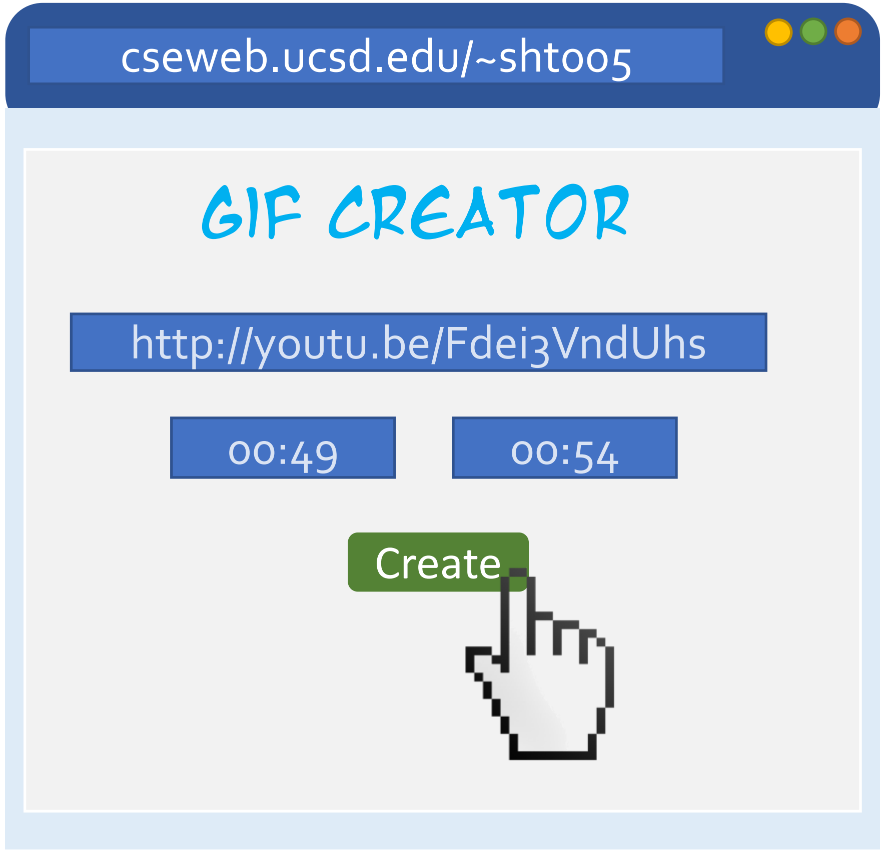
GIF CREATOR

http://youtu.be/Fdei3VndUhs

00:49

00:54

Serverless: New containers are instantiated from scratch every time you click Create



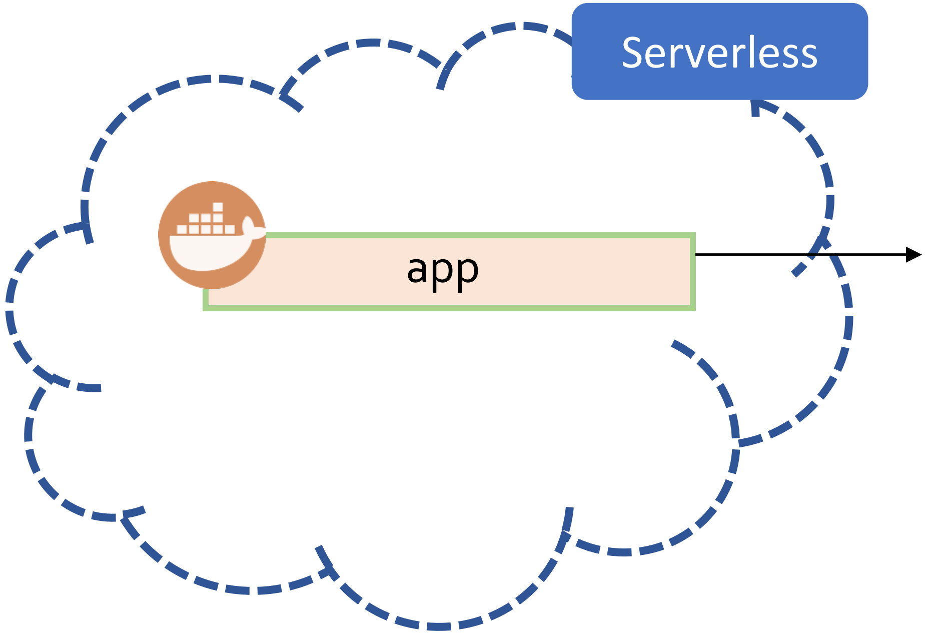
cseweb.ucsd.edu/~sht005

GIF CREATOR

http://youtu.be/Fdei3VndUhs

00:49

00:54

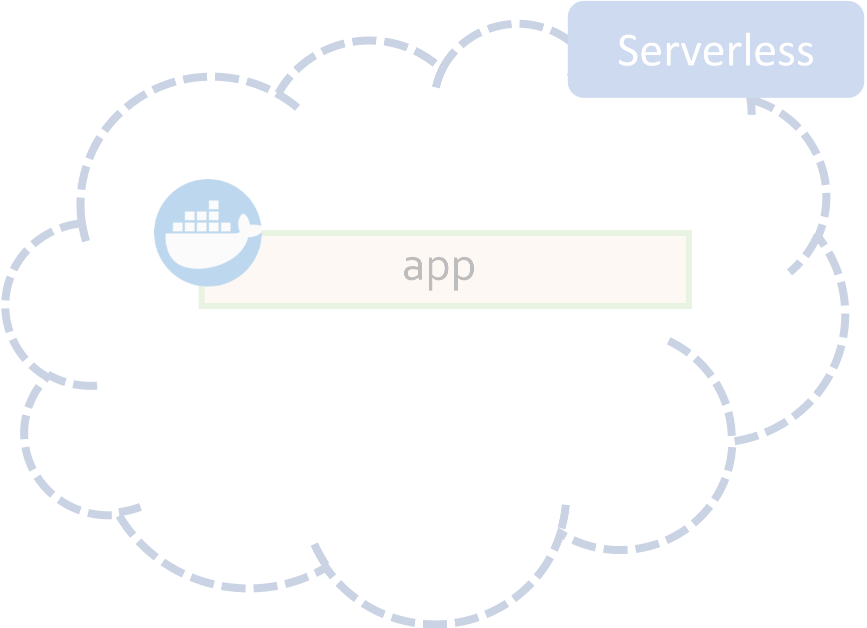


1. **Start container**
2. Download the video locally
3. Clip the video
4. Compress
5. Convert to gif
6. Send result
7. **Stop container**

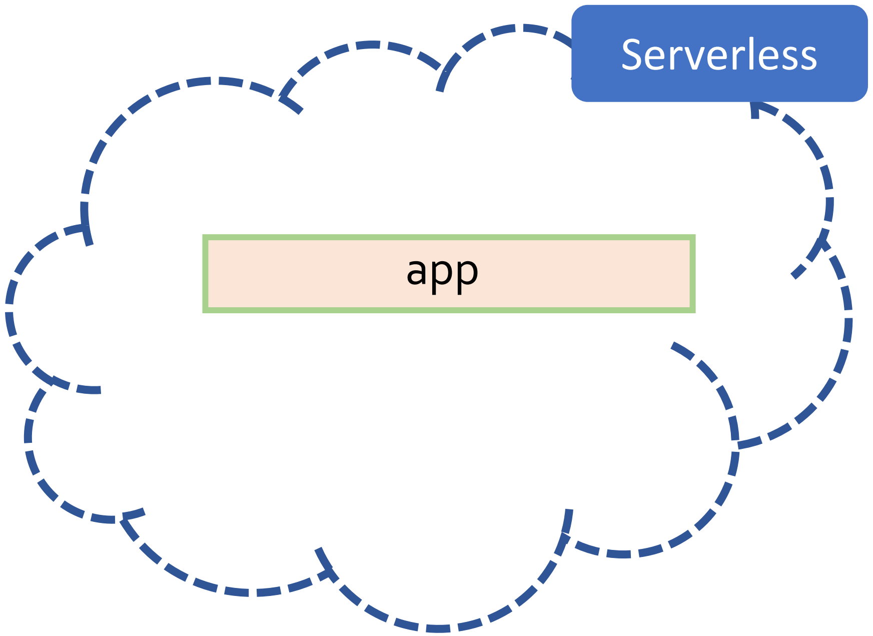
Serverless: New containers are instantiated from scratch every time you click Create

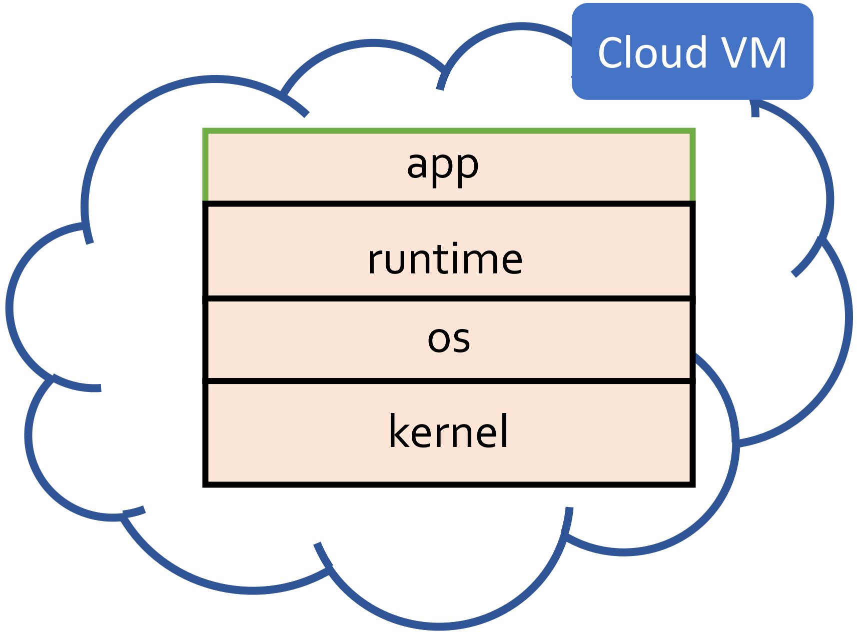
cseweb.ucsd.edu/~sht005

GIF CREATOR

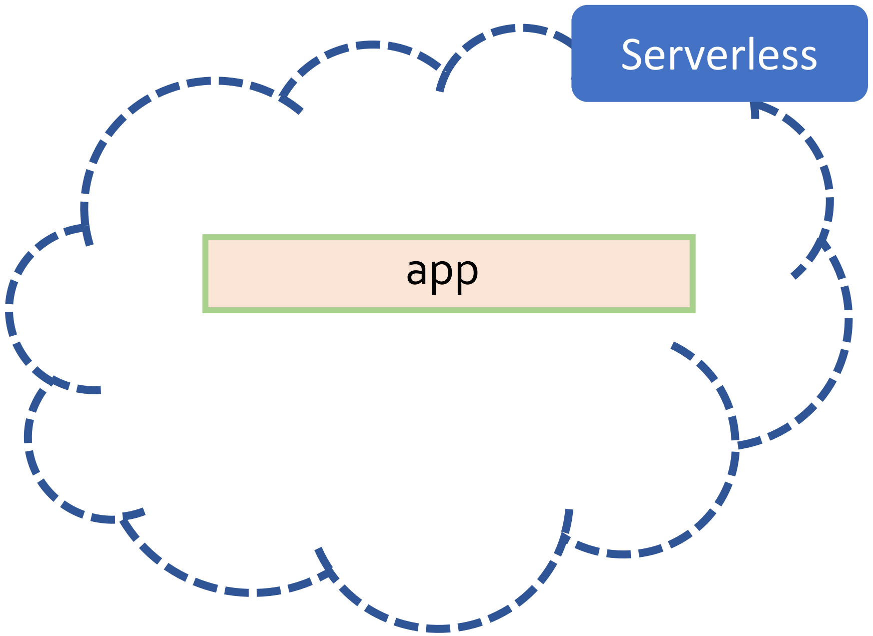


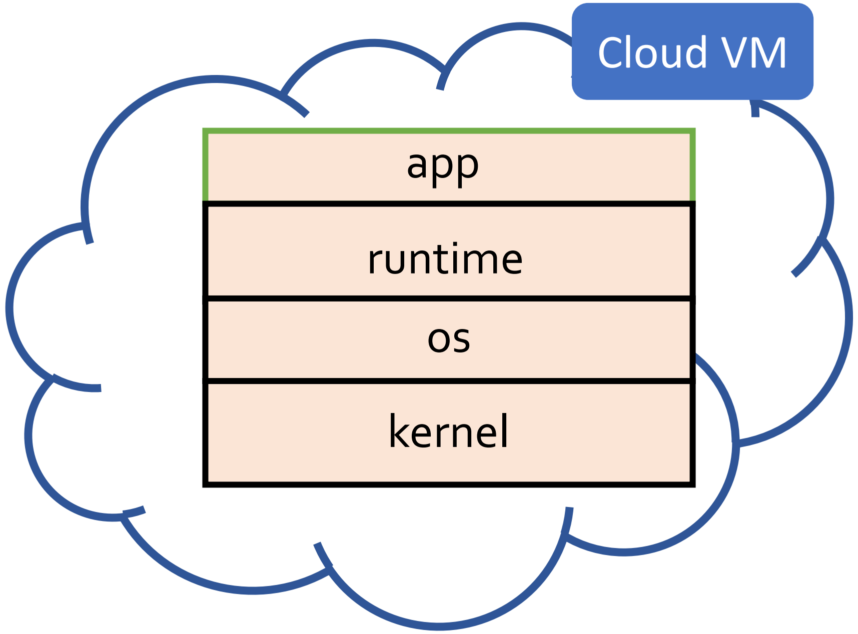
Which one is better?





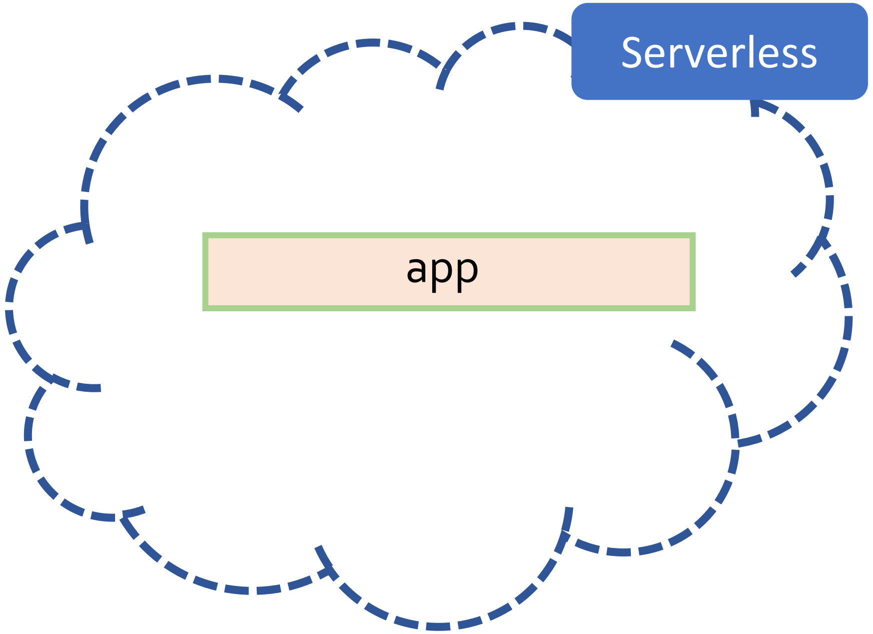
Depends on the use case

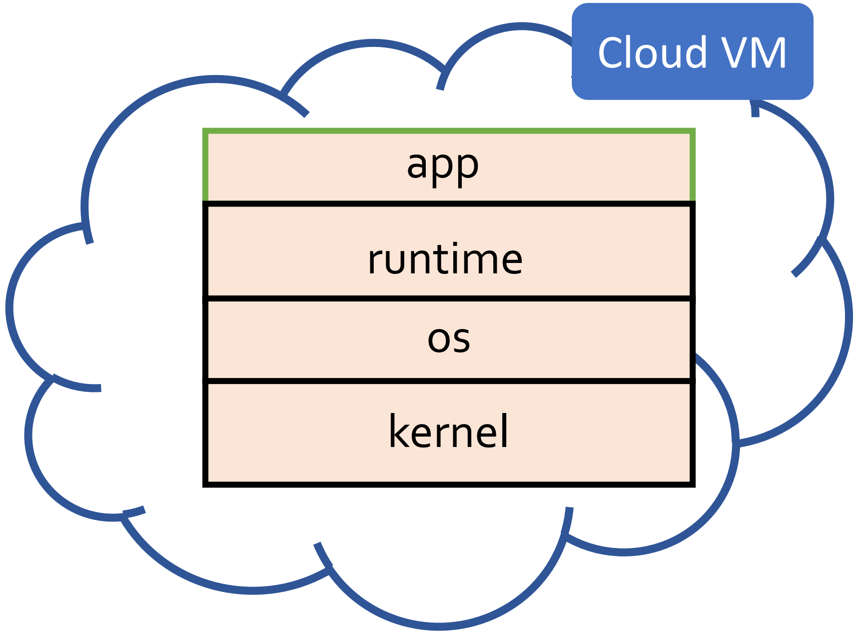




On-demand
Cheaper in many cases
Focus on application
Subject to fluctuations
Questionable security

Depends on the use case

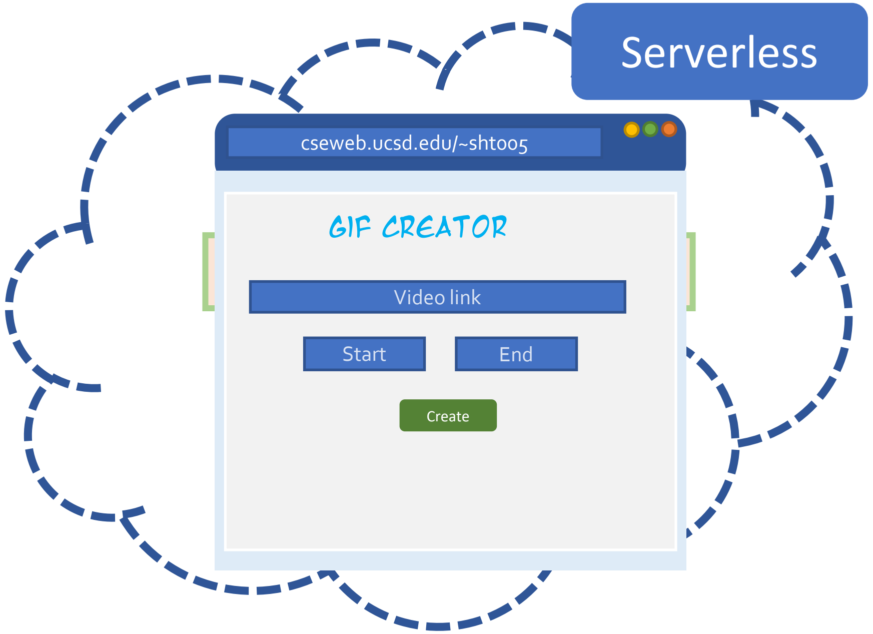




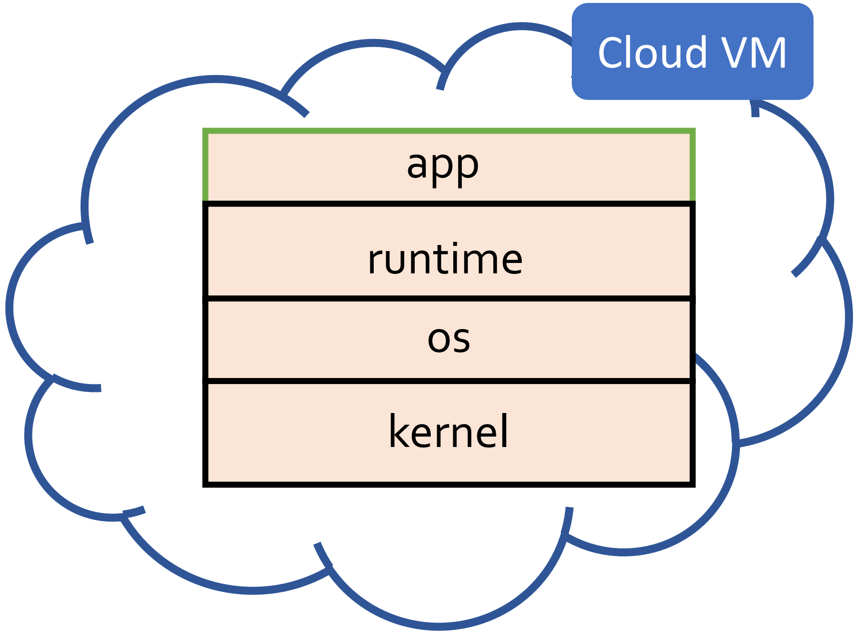
On-demand
Cheaper in many cases
Focus on application
Subject to fluctuations
Questionable security

Persistent
More Control
Fast and predictable
Difficult to manage

Depends on the use case


Serverless

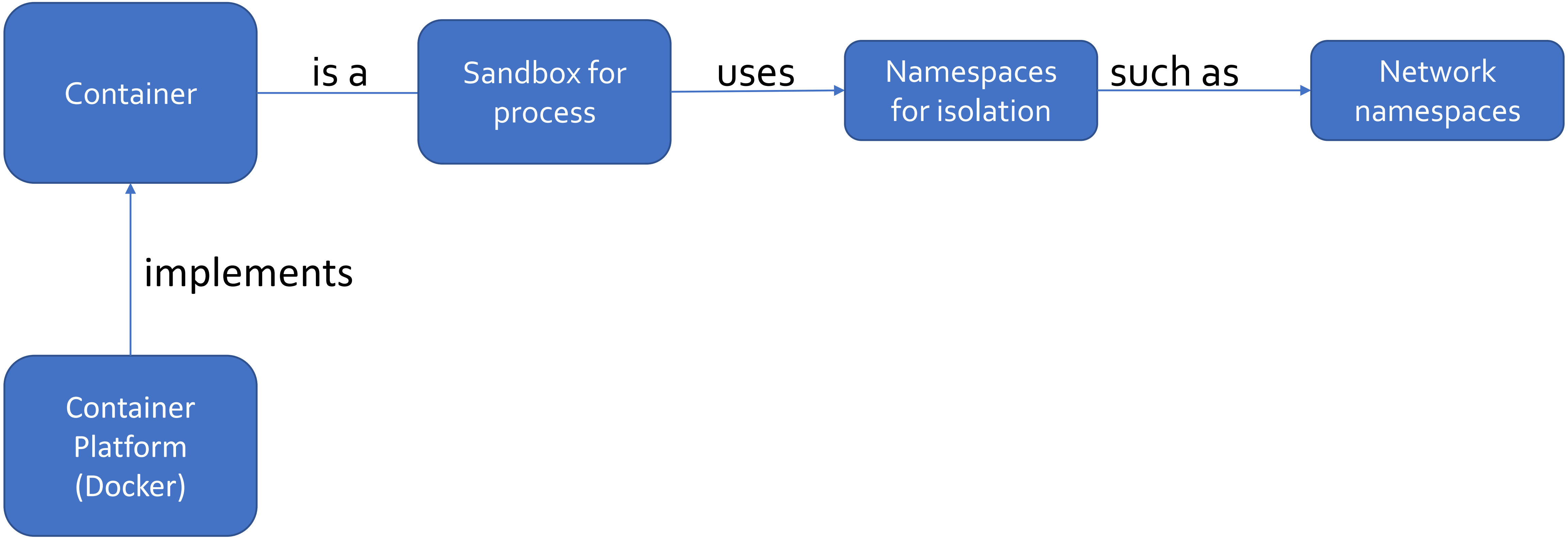
On-demand
Focus on application
Subject to fluctuations
Questionable security


Cloud VM

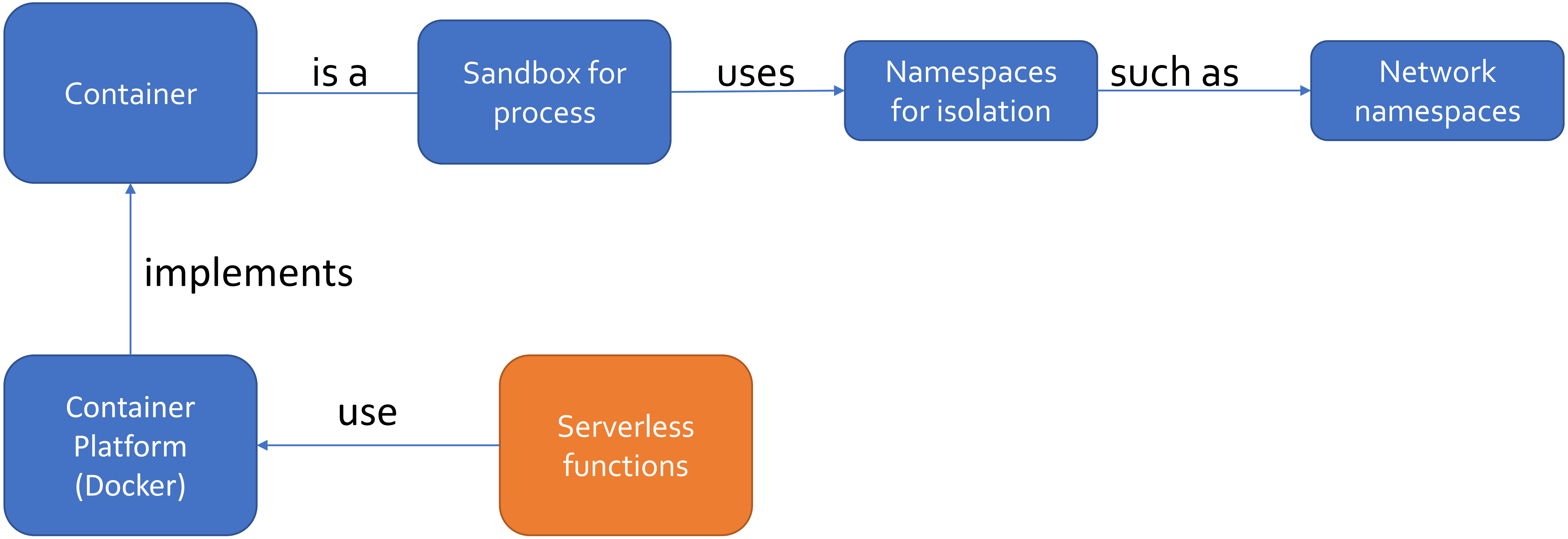
app
runtime
os
kernel

Persistent
More Control
Fast and predictable
Difficult to manage

Review


Container
is a
Sandbox for process
uses
Namespaces for isolation
such as
Network namespaces
implements
Container Platform (Docker)

Review



Container

is a

Sandbox for process

uses

Namespaces for isolation

such as

Network namespaces

implements

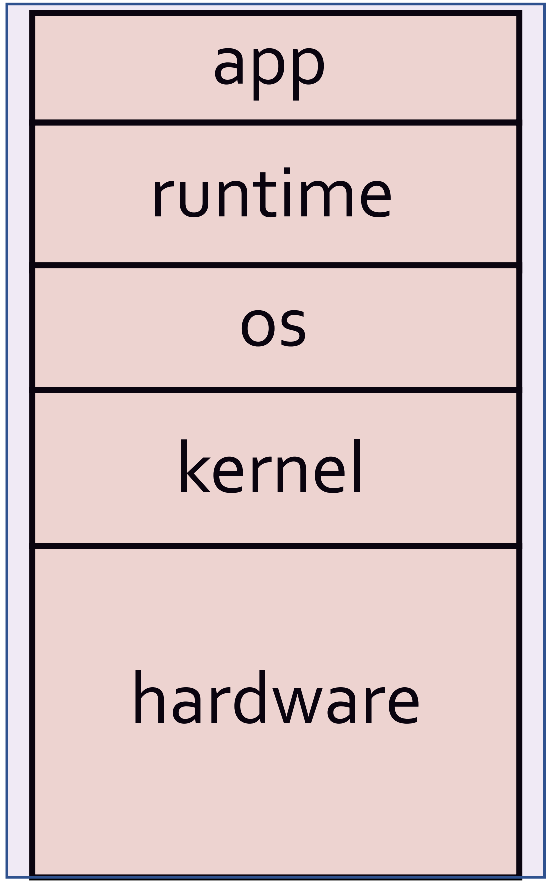
Container Platform (Docker)

use

Serverless functions

Virtualization Review

Layers of virtualization



app

runtime

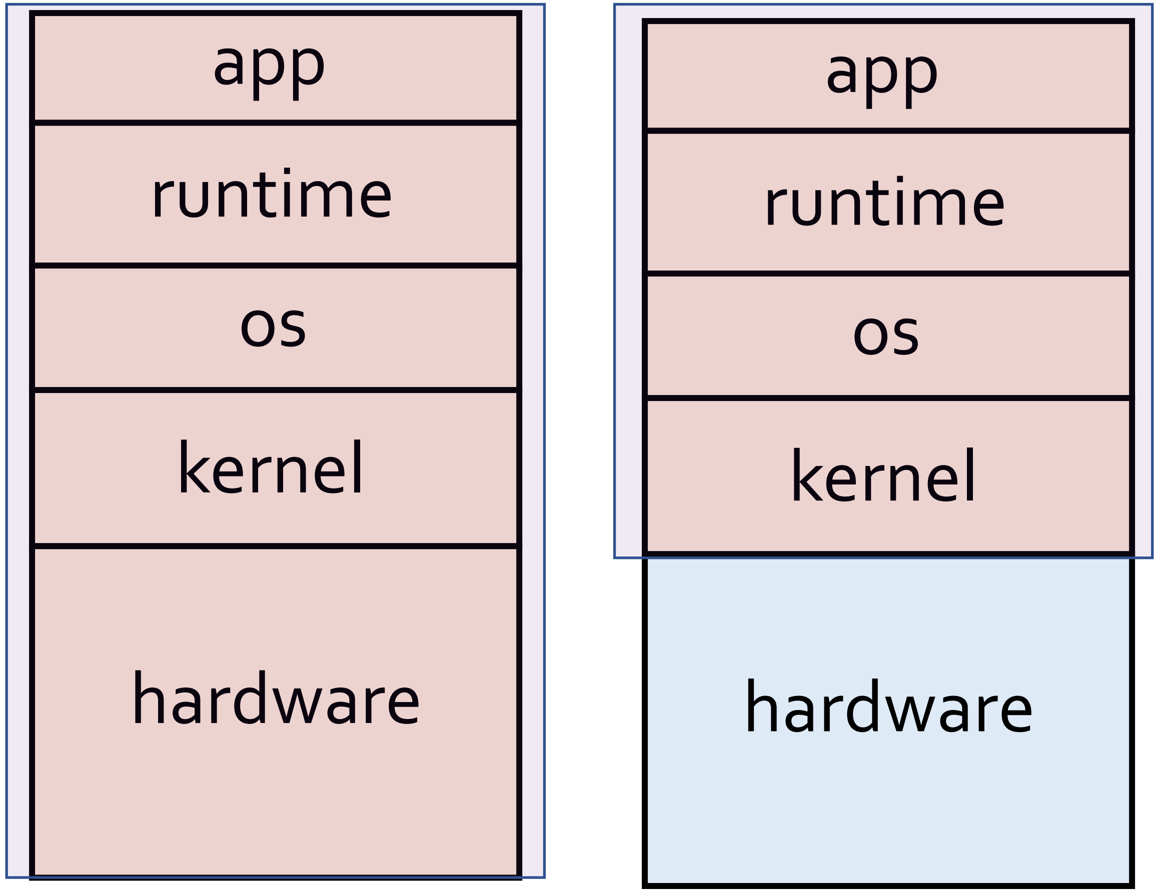
os

kernel

hardware

bare metal

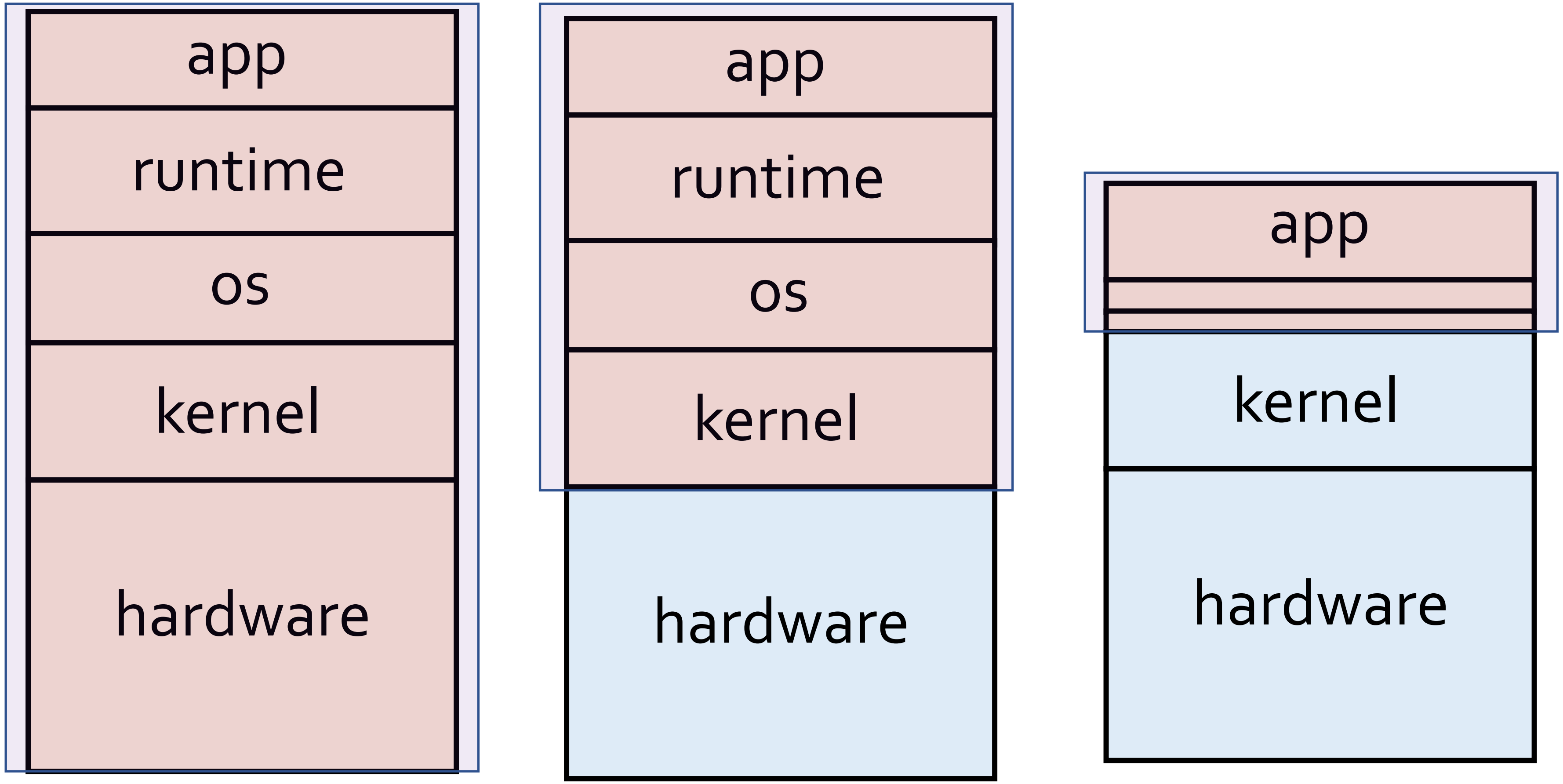
Layers of virtualization



bare metal

virtual machine

Layers of virtualization

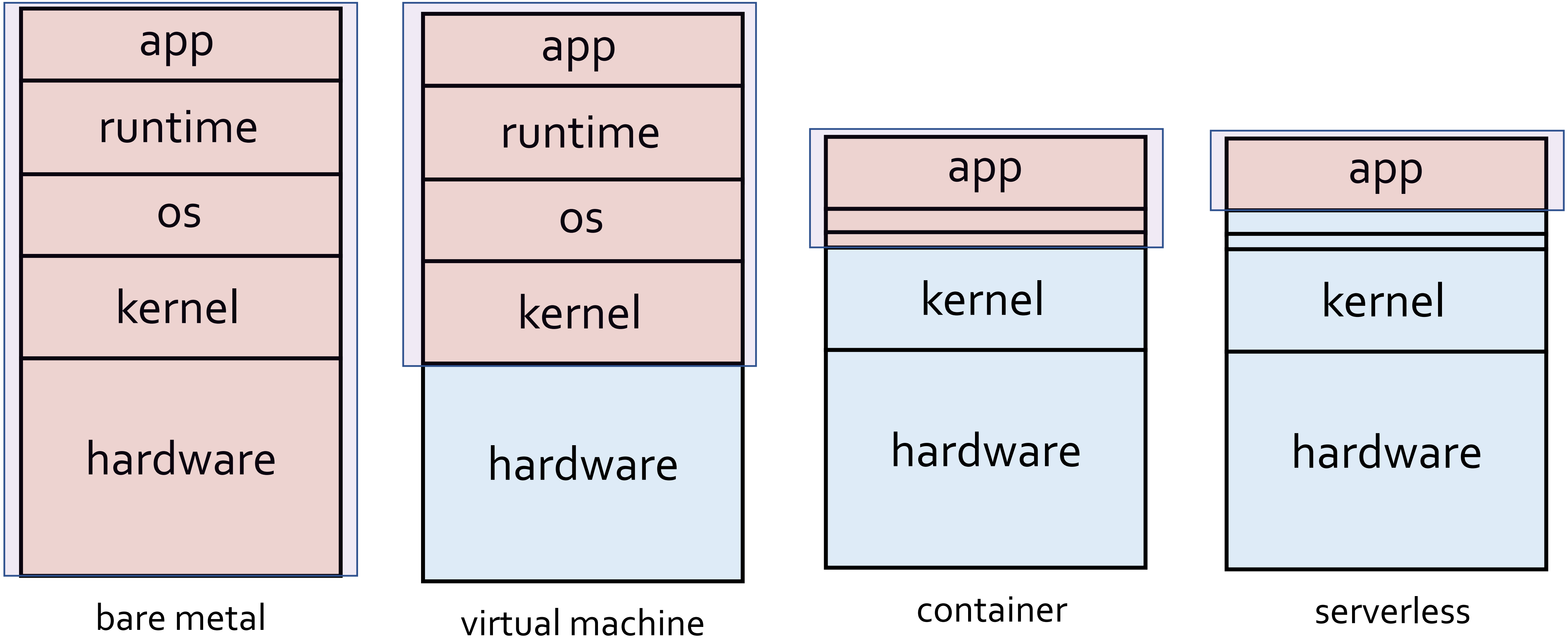


bare metal

virtual machine

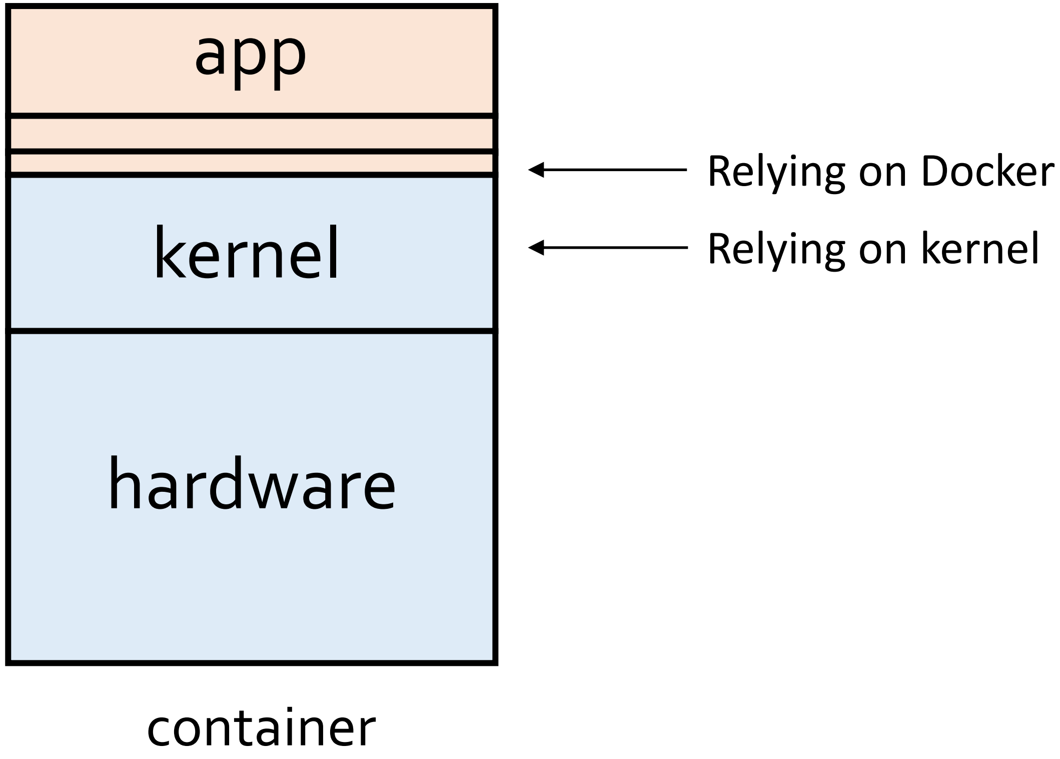
container

Layers of virtualization

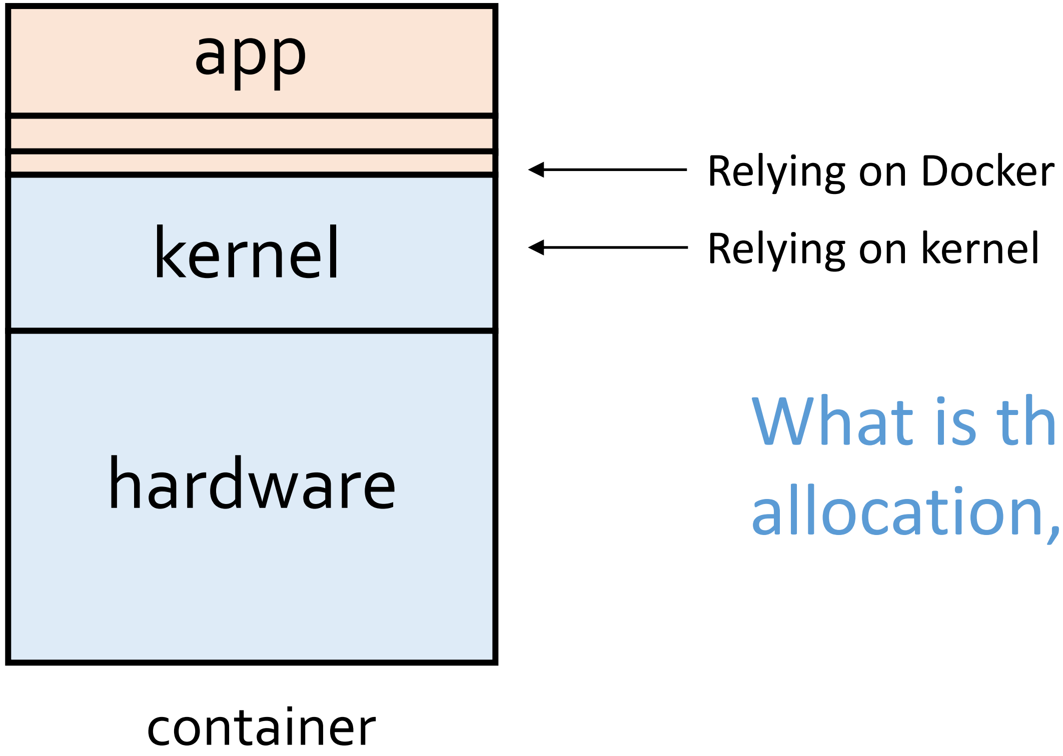


Container and Serverless Research

Topic 1: Reduce startup variance



Topic 1: Reduce startup variance



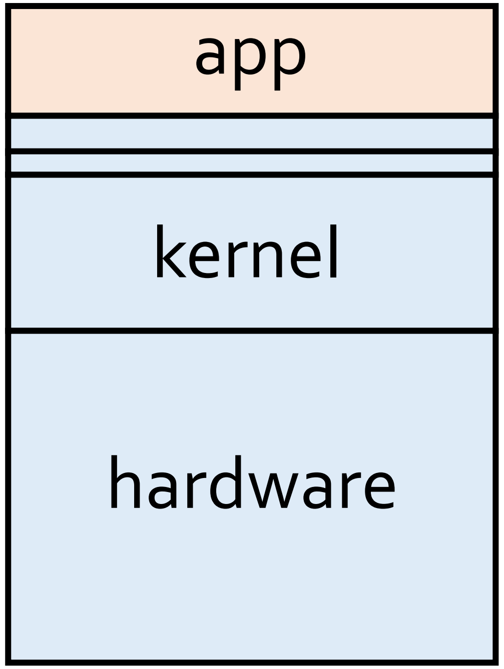
Relying on Docker

Relying on kernel

What is the optimal caching, locking, allocation, sharing mechanism?

Topic 1: Reduce serverless startup variance

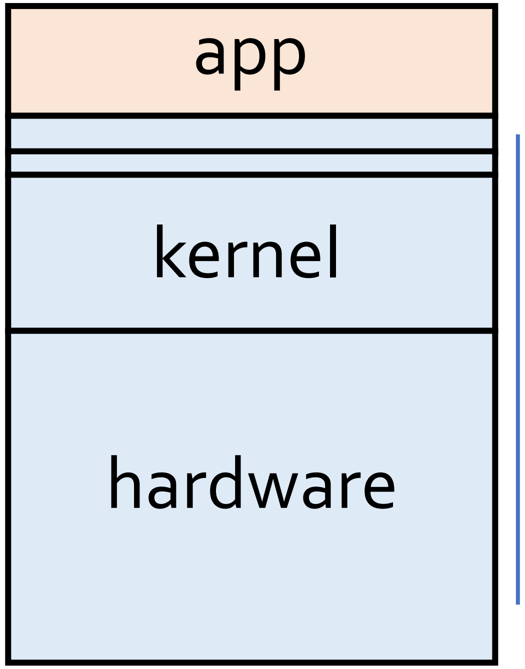
COLD STARTS



serverless

Topic 1: Reduce serverless startup variance

COLD STARTS

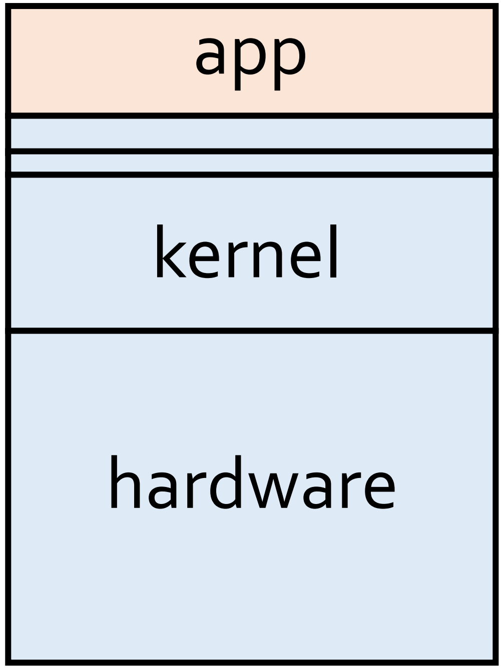


Enter stack must coordinate

serverless

Topic 1: Reduce serverless startup variance

COLD STARTS



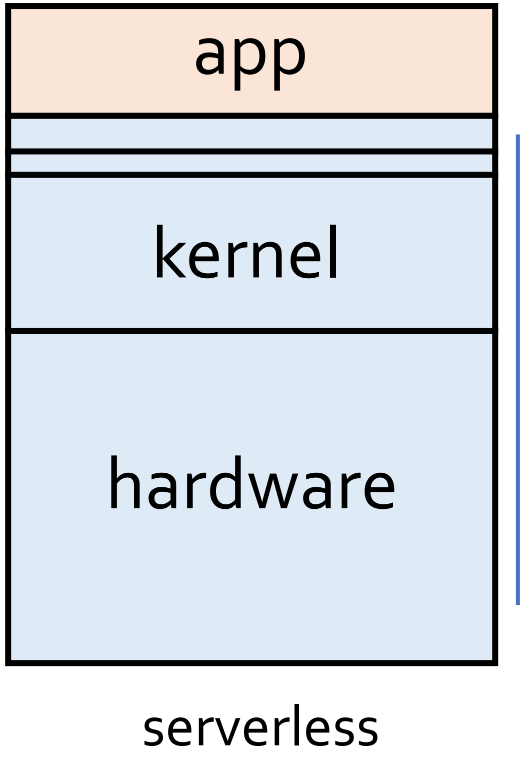
serverless

Enter stack must coordinate

Extremely important in serverless context for better response times

Topic 1: Reduce serverless startup variance

COLD STARTS



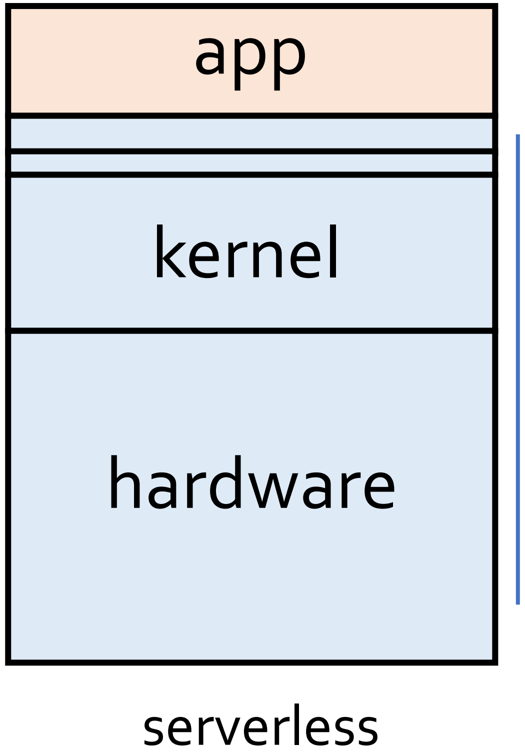
Enter stack must coordinate

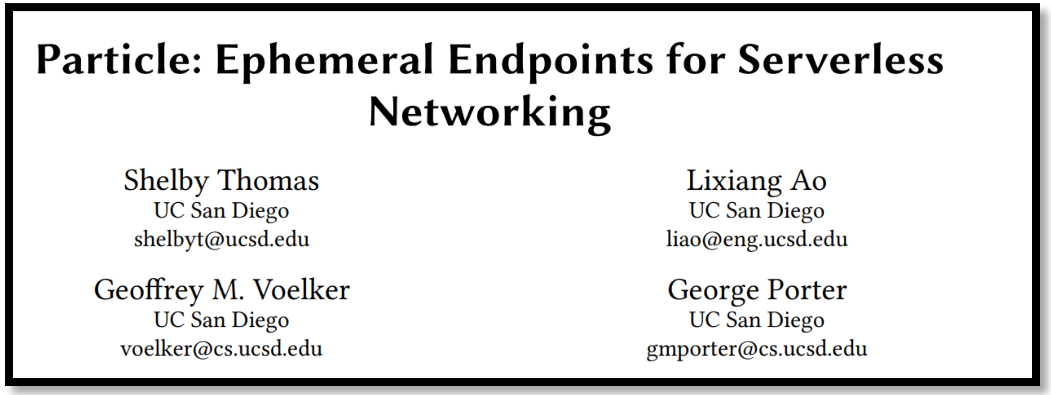
Extremely important in serverless context for better response times

A 100-millisecond delay in website load time can hurt conversion rates by 7 percent - Akamai, 2017

Topic 1: Reduce serverless startup variance

COLD STARTS





Particle: Ephemeral Endpoints for Serverless Networking

Shelby Thomas
UC San Diego
shelbyt@ucsd.edu

Lixiang Ao
UC San Diego
liao@eng.ucsd.edu

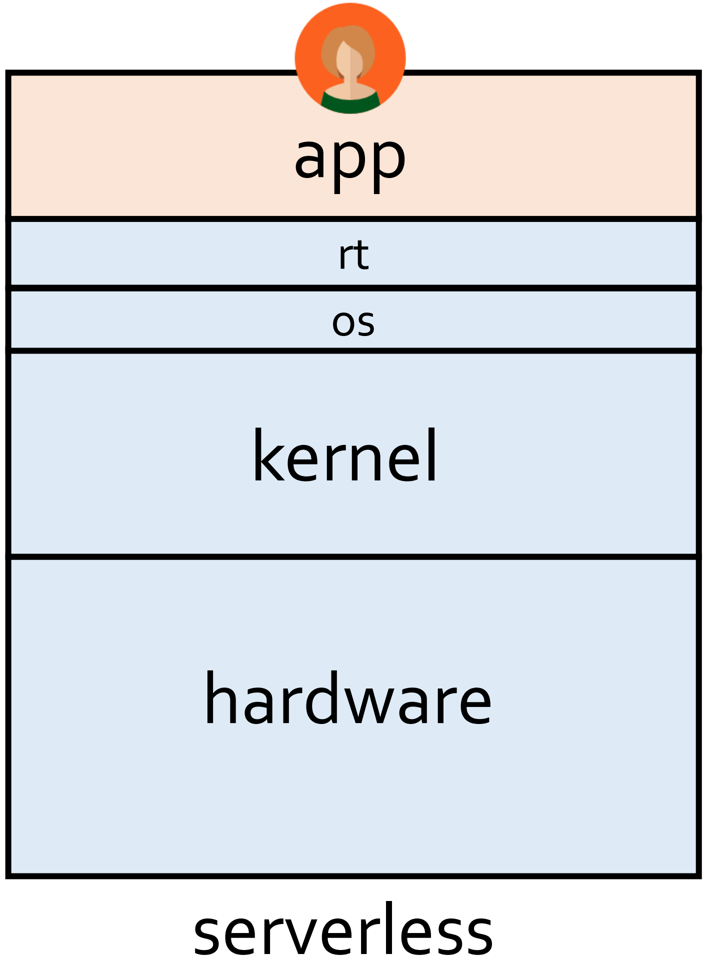
Geoffrey M. Voelker
UC San Diego
voelker@cs.ucsd.edu

George Porter
UC San Diego
gmporter@cs.ucsd.edu

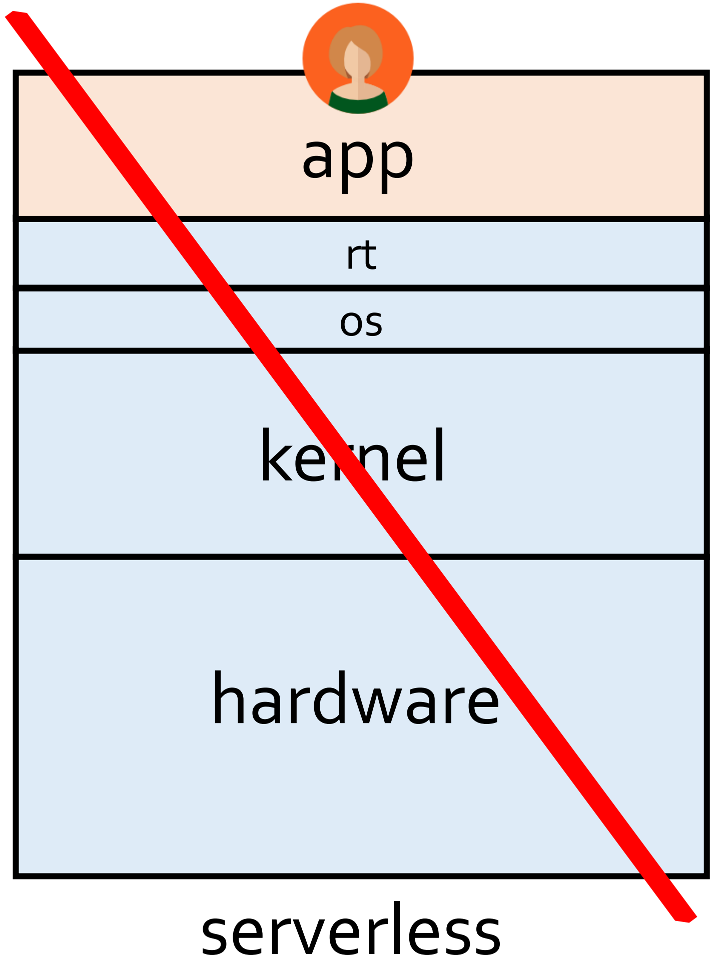
Extremely important in serverless context for better response times

A 100-millisecond delay in website load time can hurt conversion rates by 7 percent - Akamai, 2017

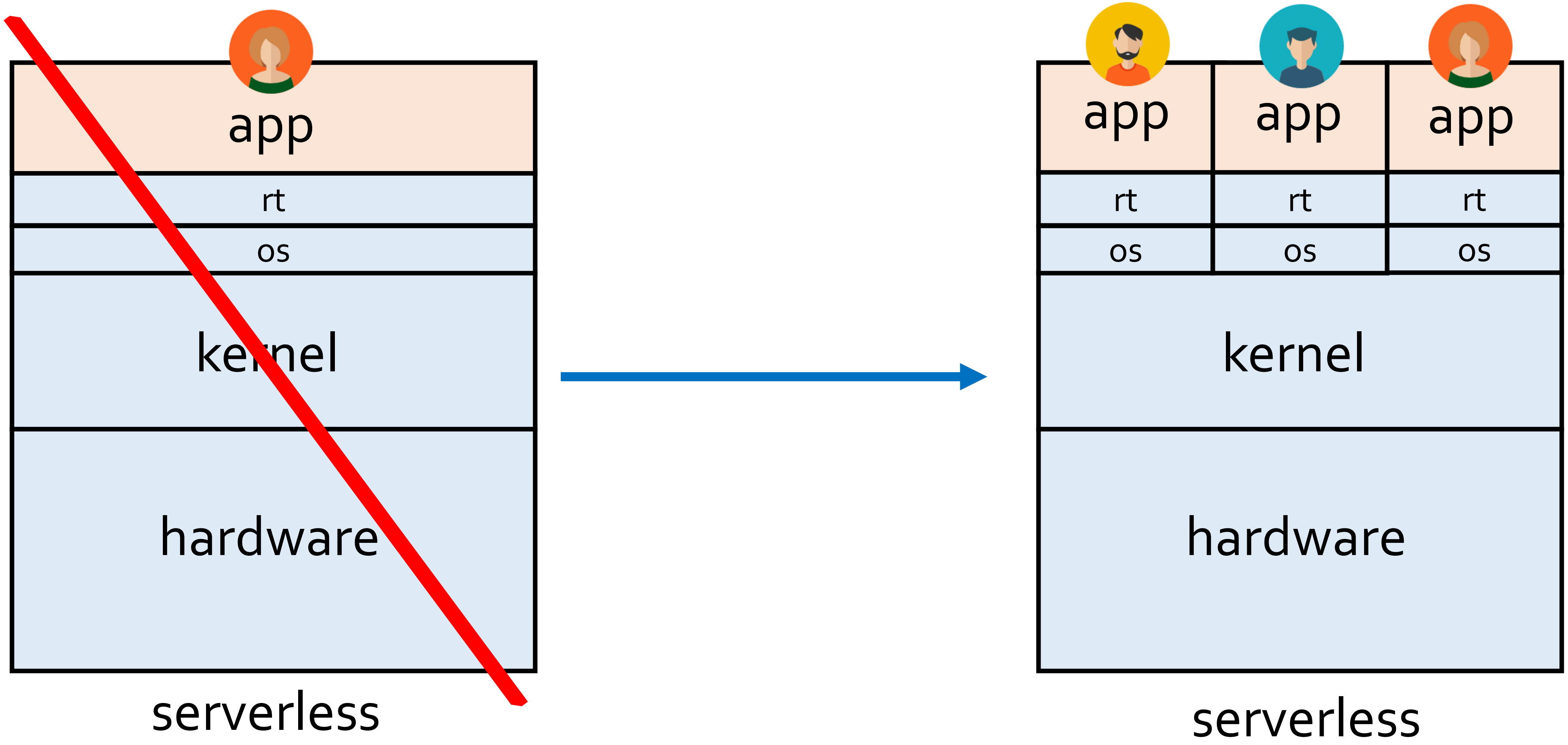
Topic 2: Improve security and control



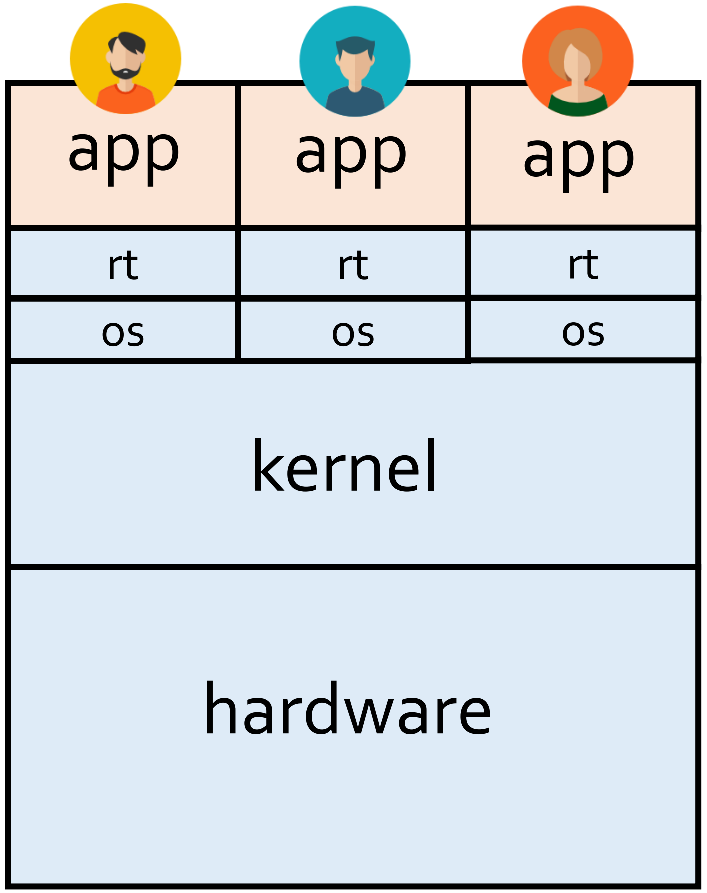
Topic 2: Improve security and control



Topic 2: Improve security and control

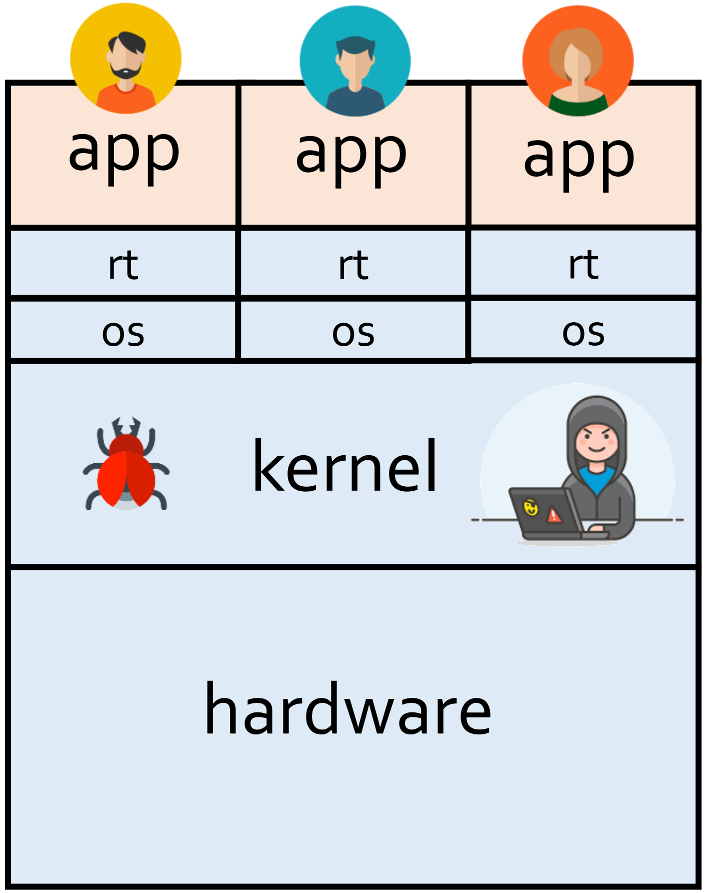


Topic 2: Improve security and control



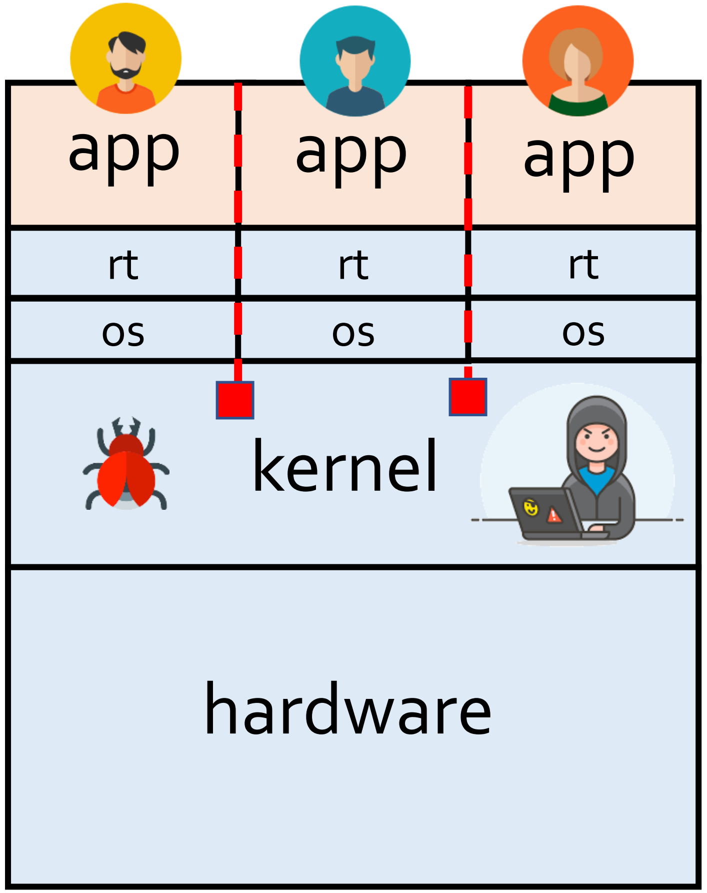
serverless

Topic 2: Improve security and control



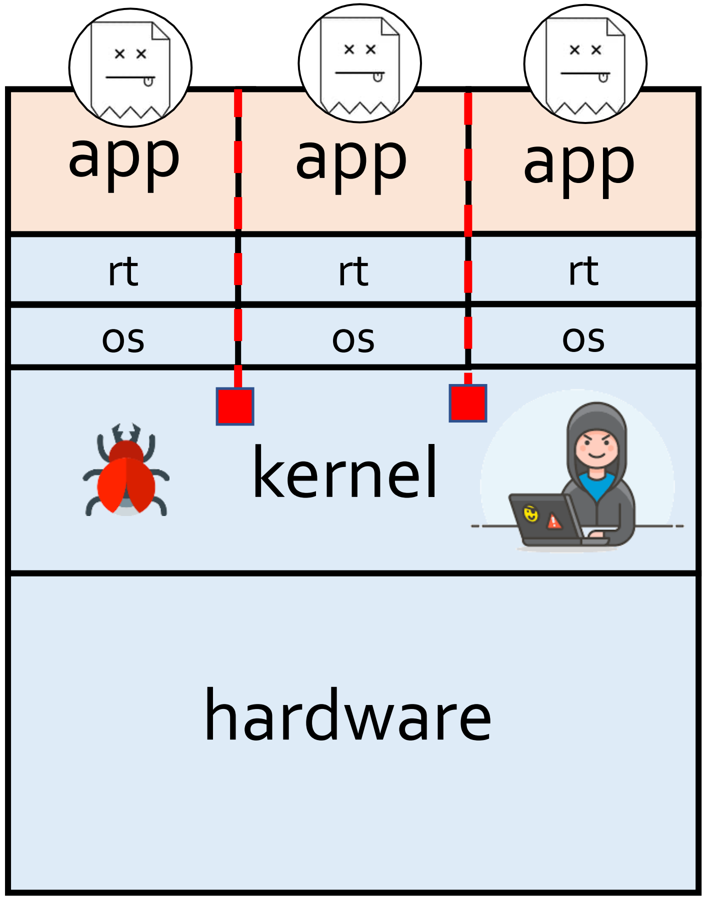
serverless

Topic 2: Improve security and control



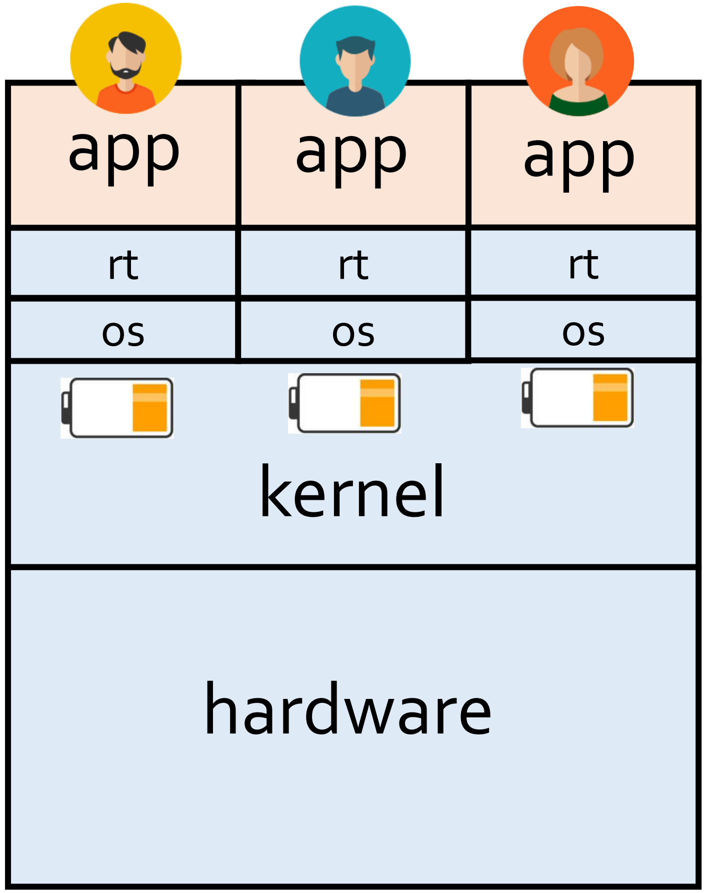
serverless

Topic 2: Improve security and control



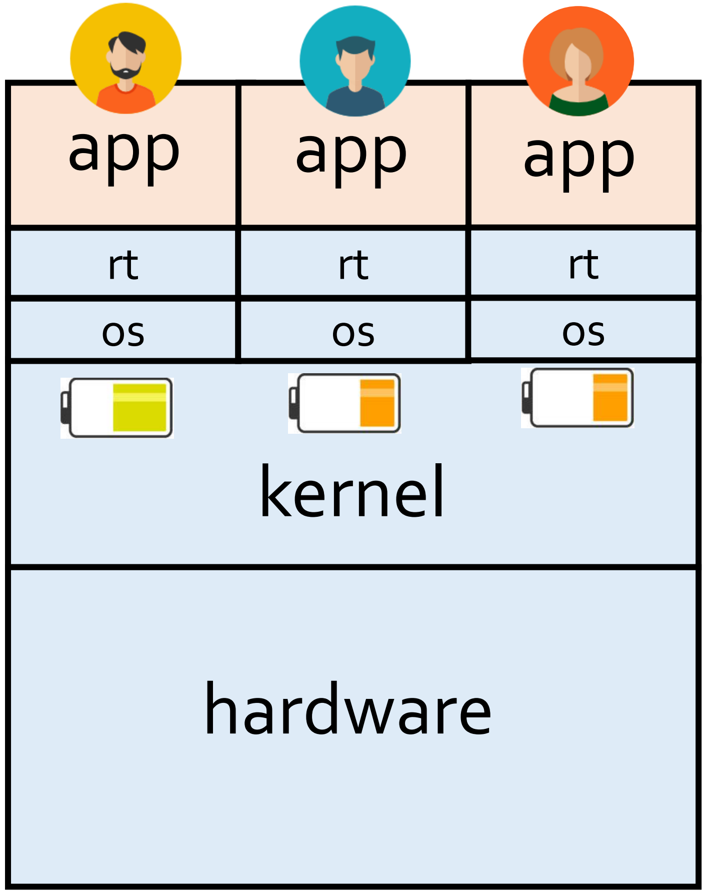
serverless

Topic 2: Improve security and control



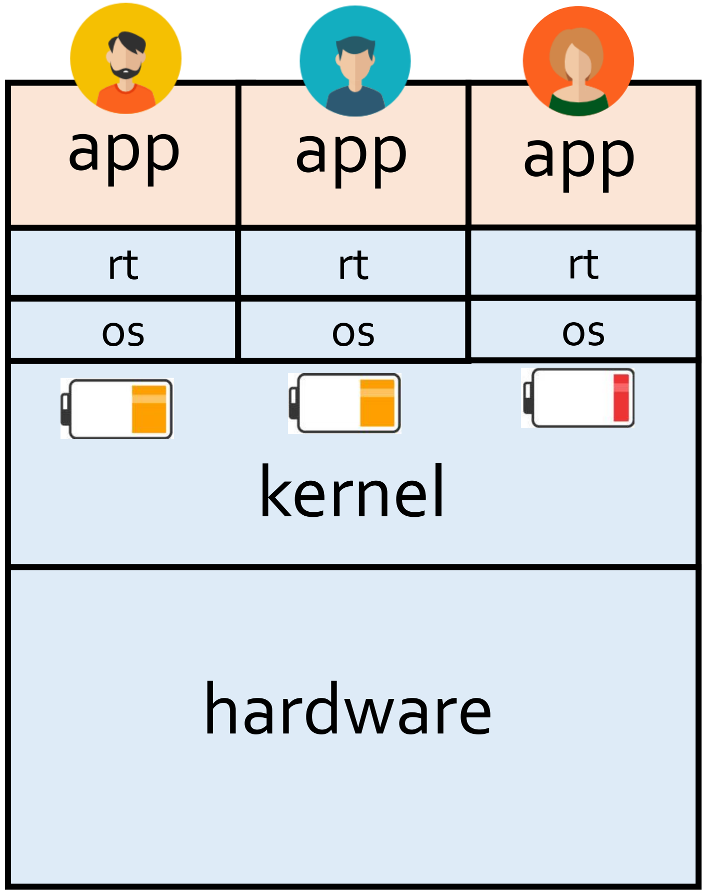
serverless

Topic 2: Improve security and control



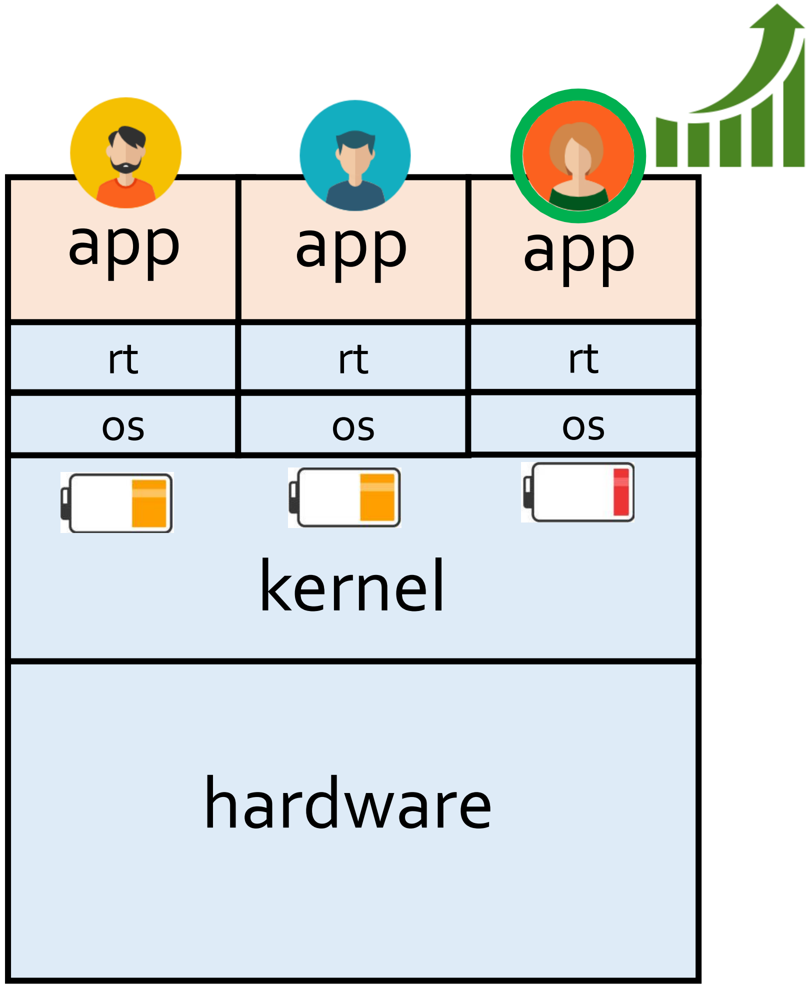
serverless

Topic 2: Improve security and control



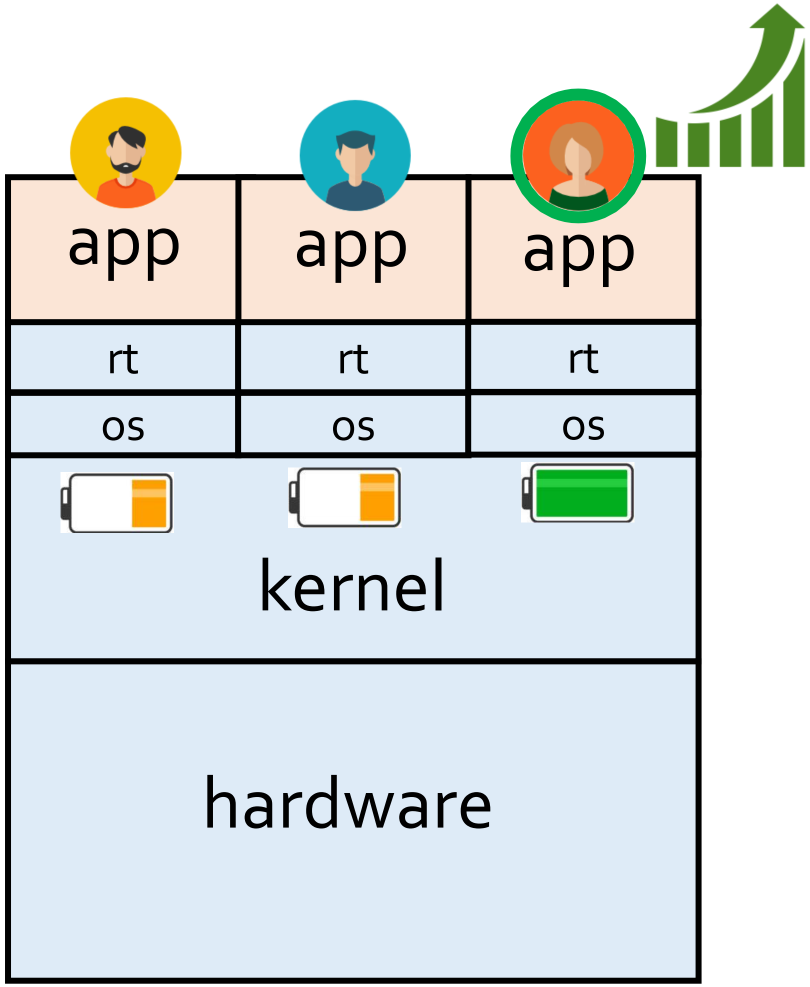
serverless

Topic 2: Improve security and control



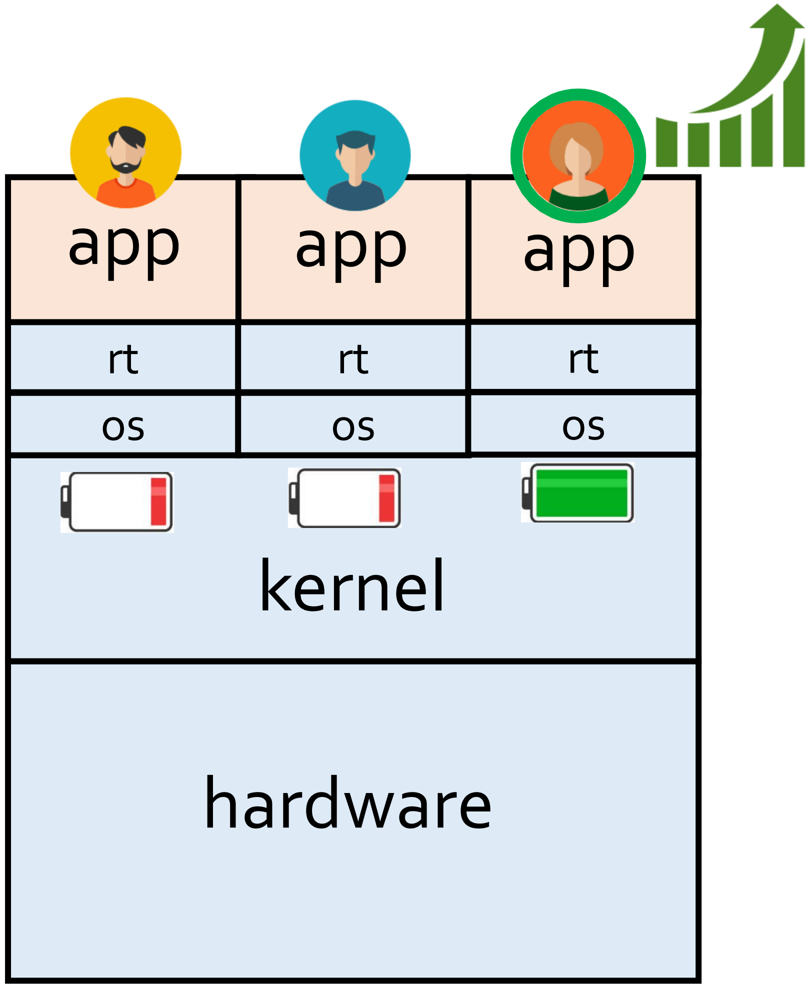
serverless

Topic 2: Improve security and control



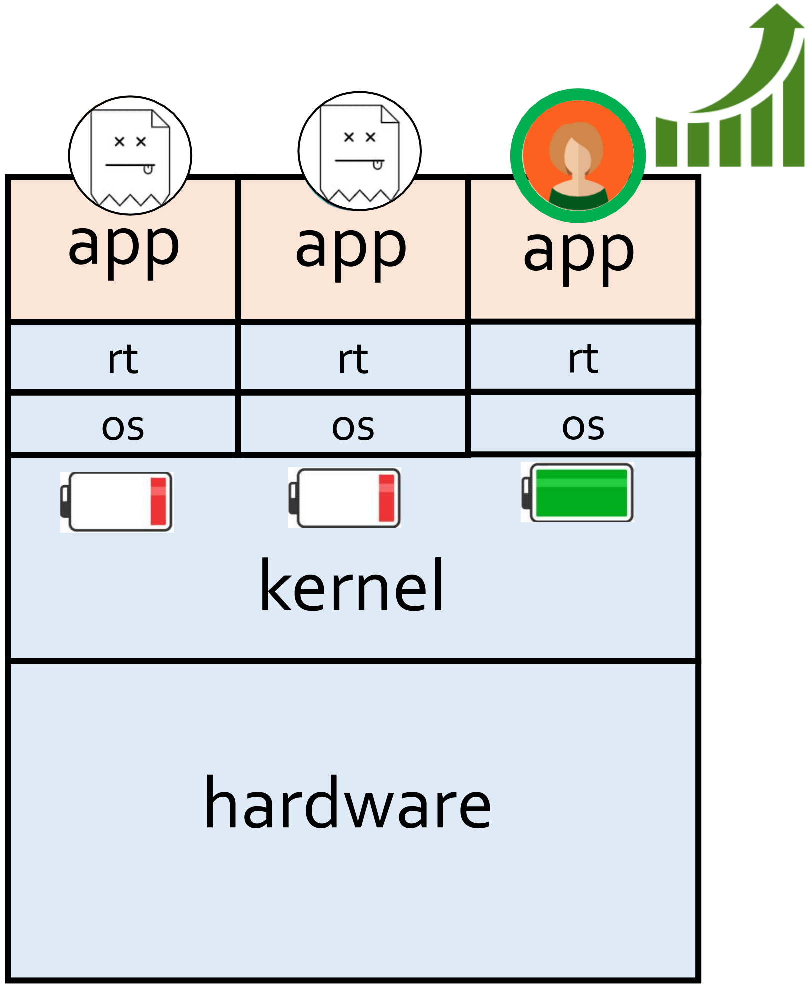
serverless

Topic 2: Improve security and control



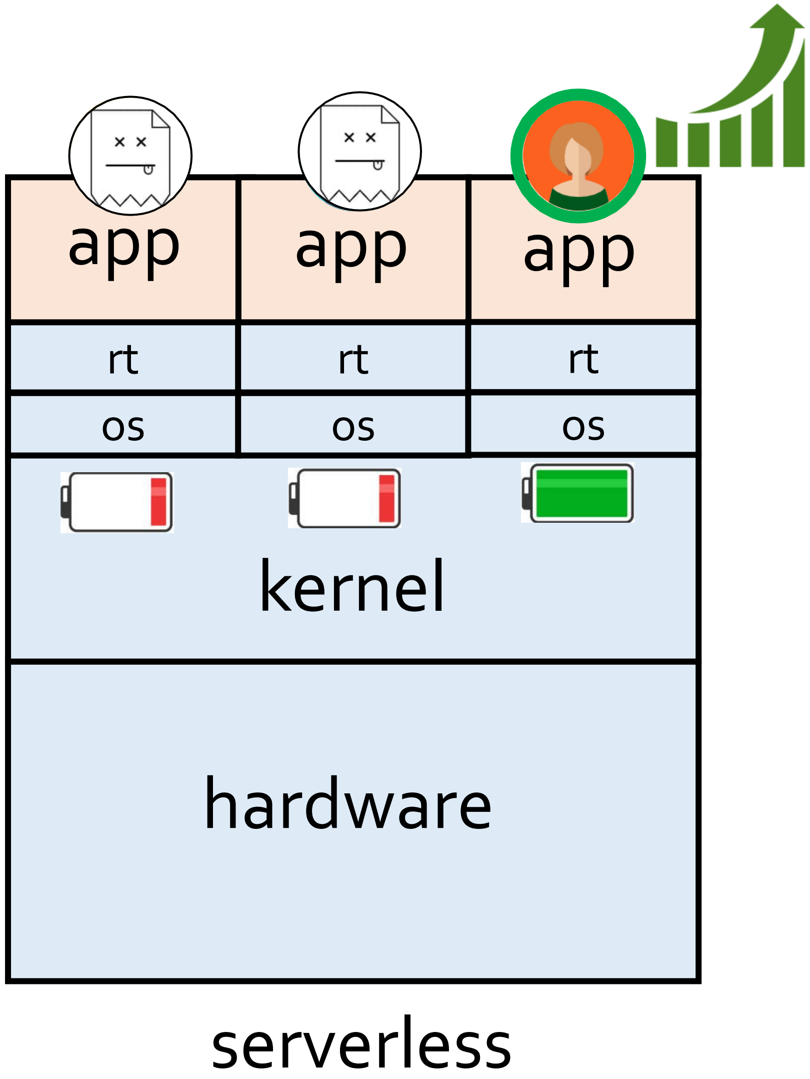
serverless

Topic 2: Improve security and control



serverless

Topic 2: Improve security and control



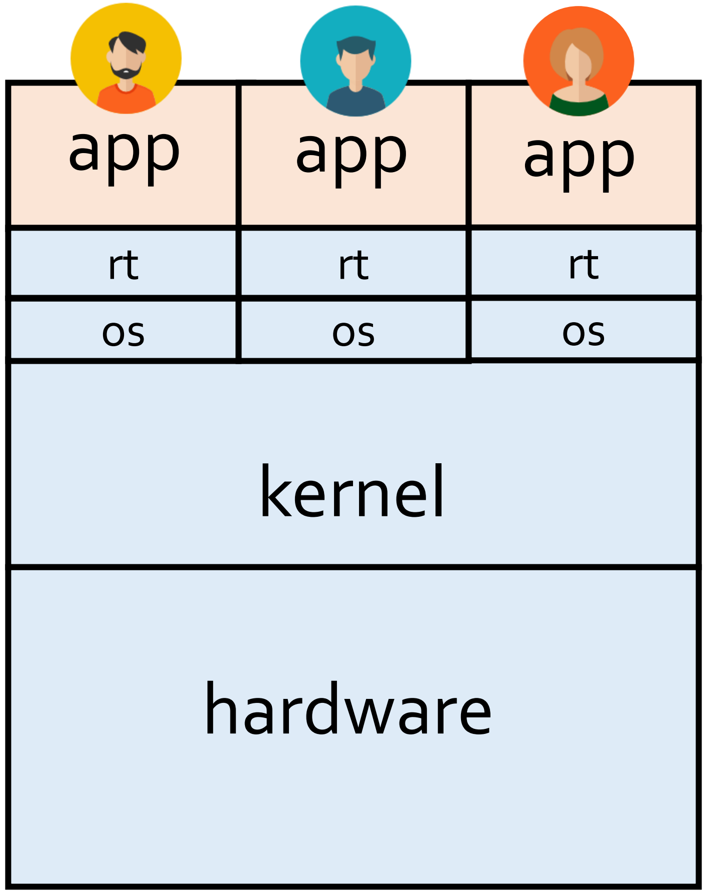
CPU Problem

Network Problem

Memory Works OK

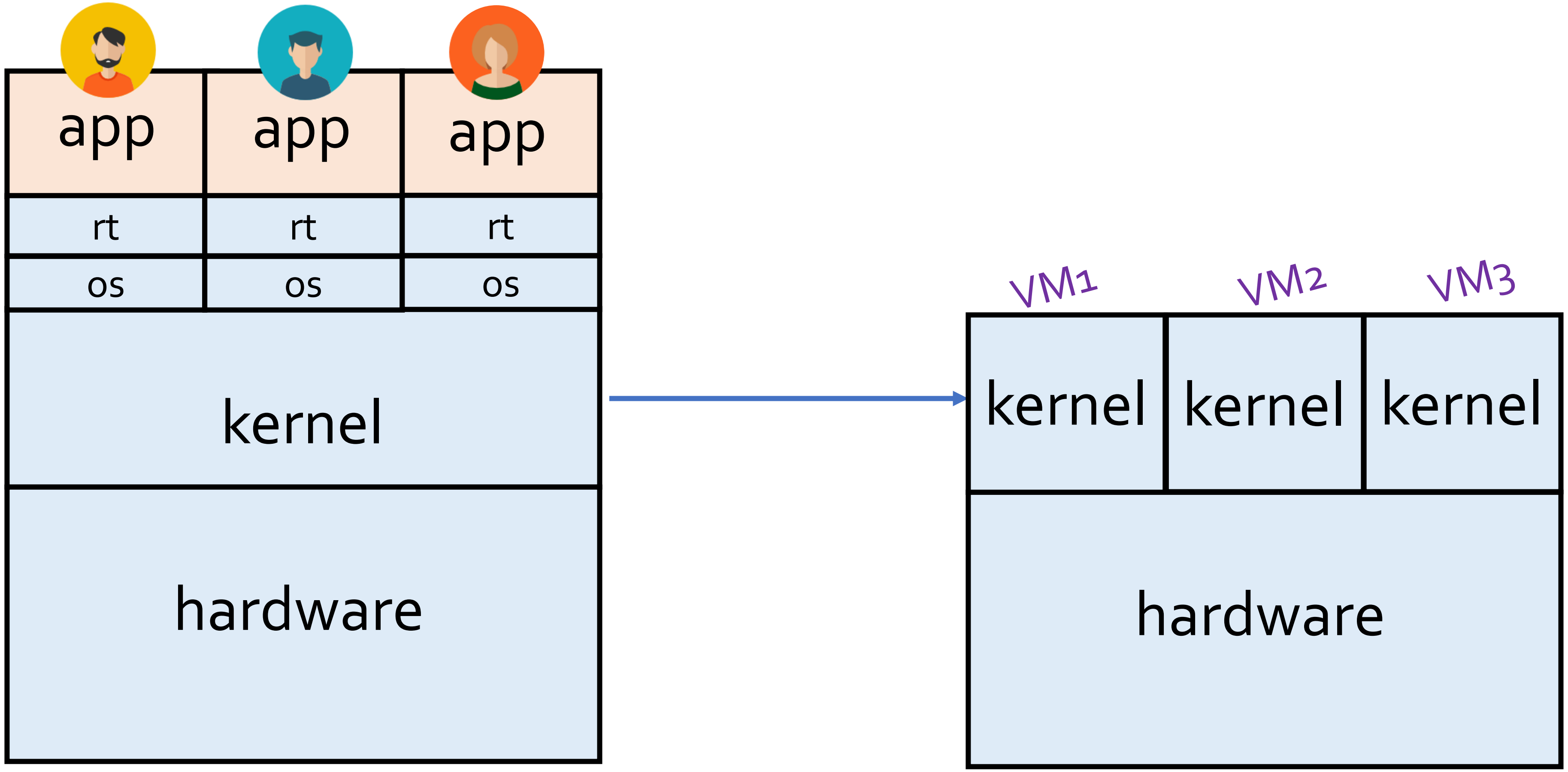
Topic 2: Improve security and control

Partially solved by running containers inside VMs



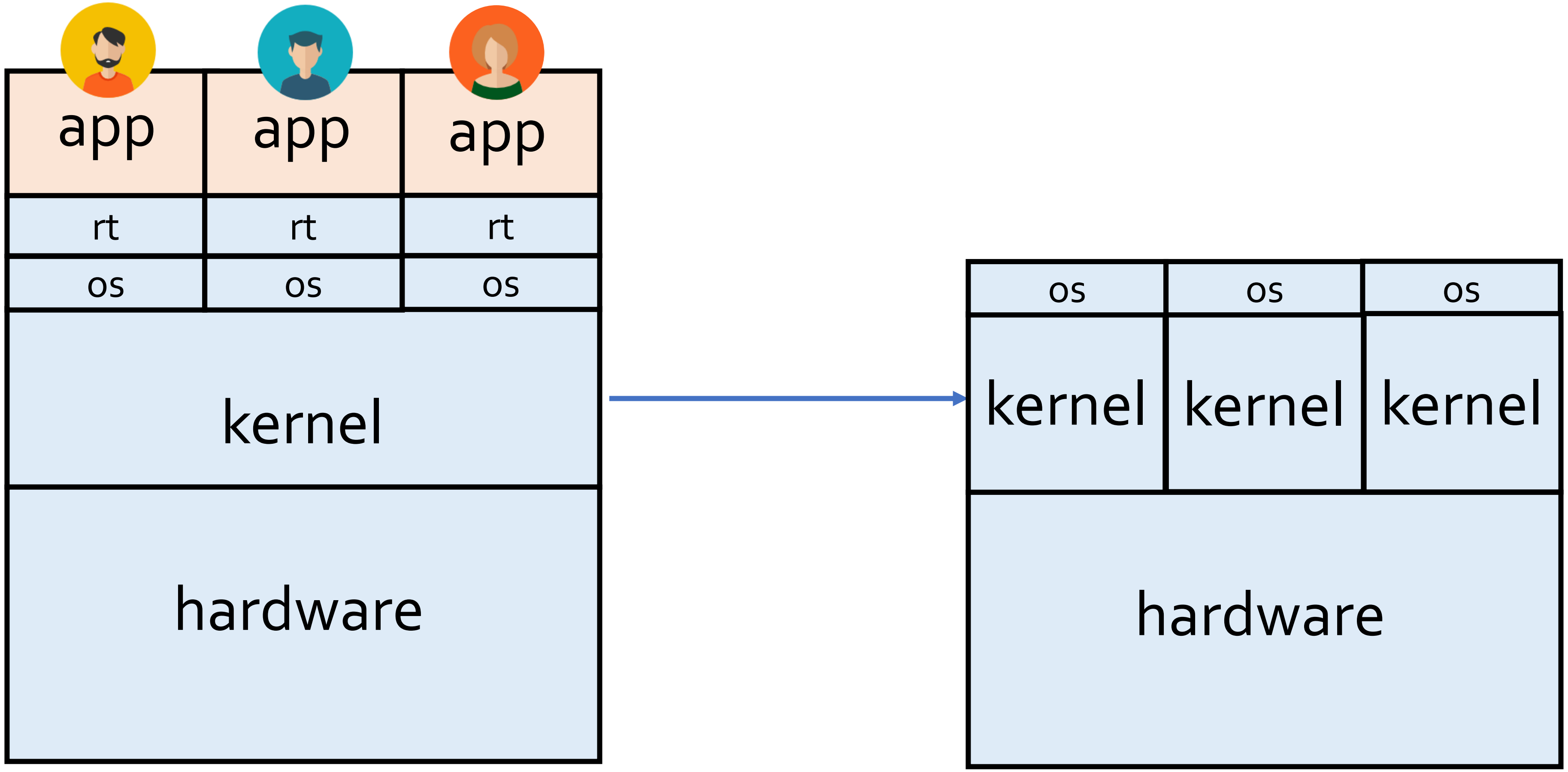
Topic 2: Improve security and control

Partially solved by running containers inside VMs



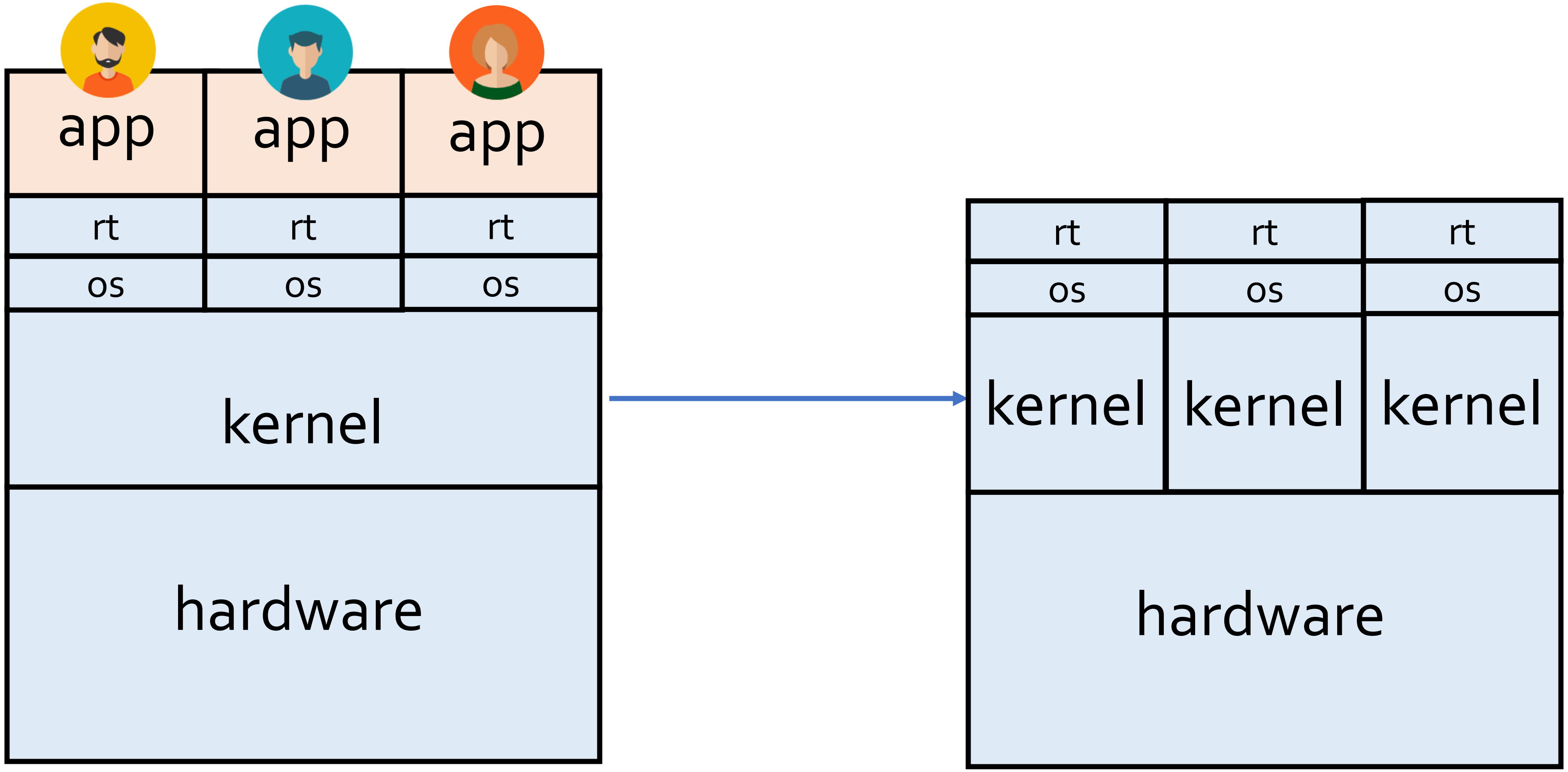
Topic 2: Improve security and control

Partially solved by running containers inside VMs



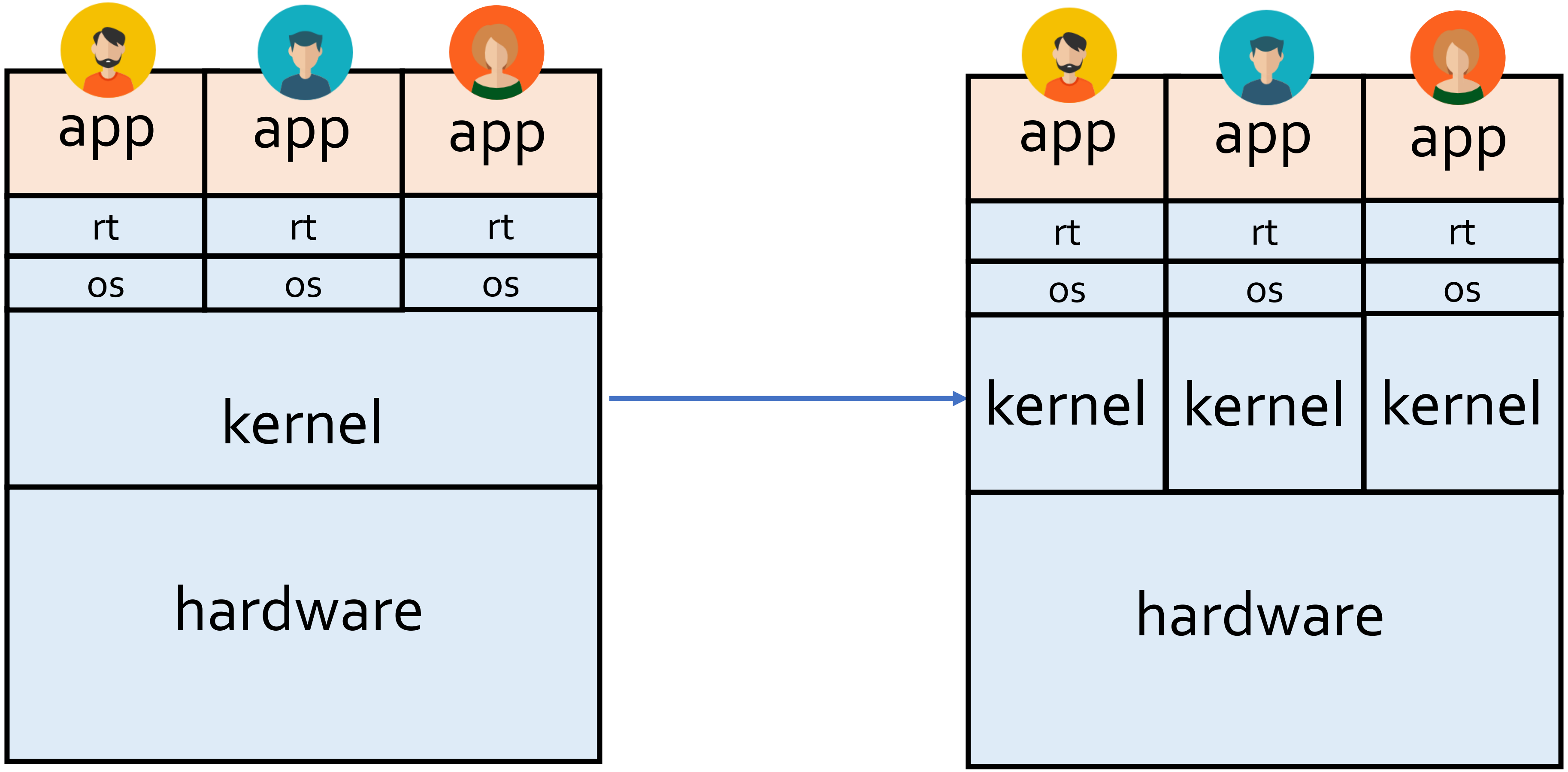
Topic 2: Improve security and control

Partially solved by running containers inside VMs



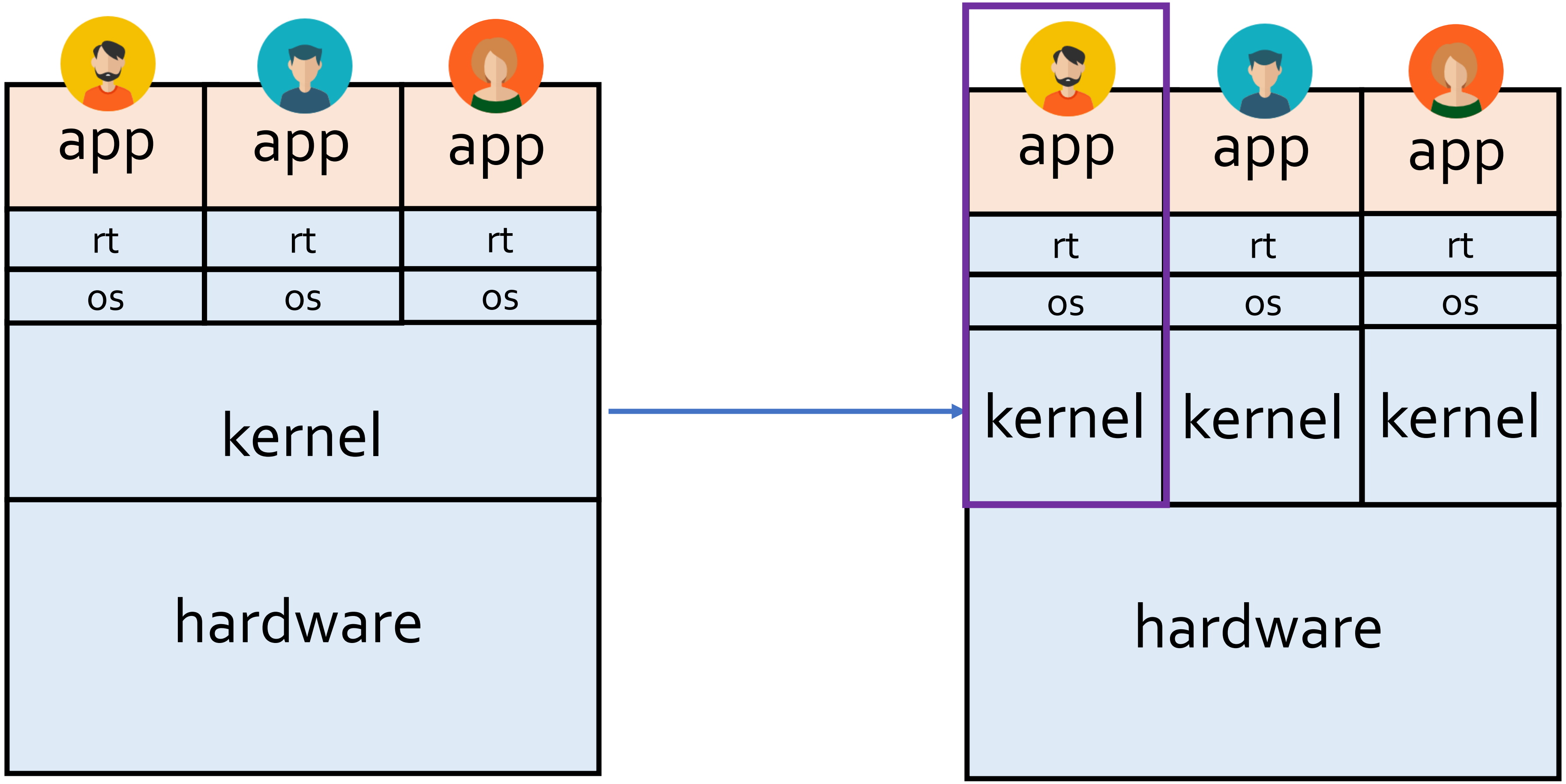
Topic 2: Improve security and control

Partially solved by running containers inside VMs



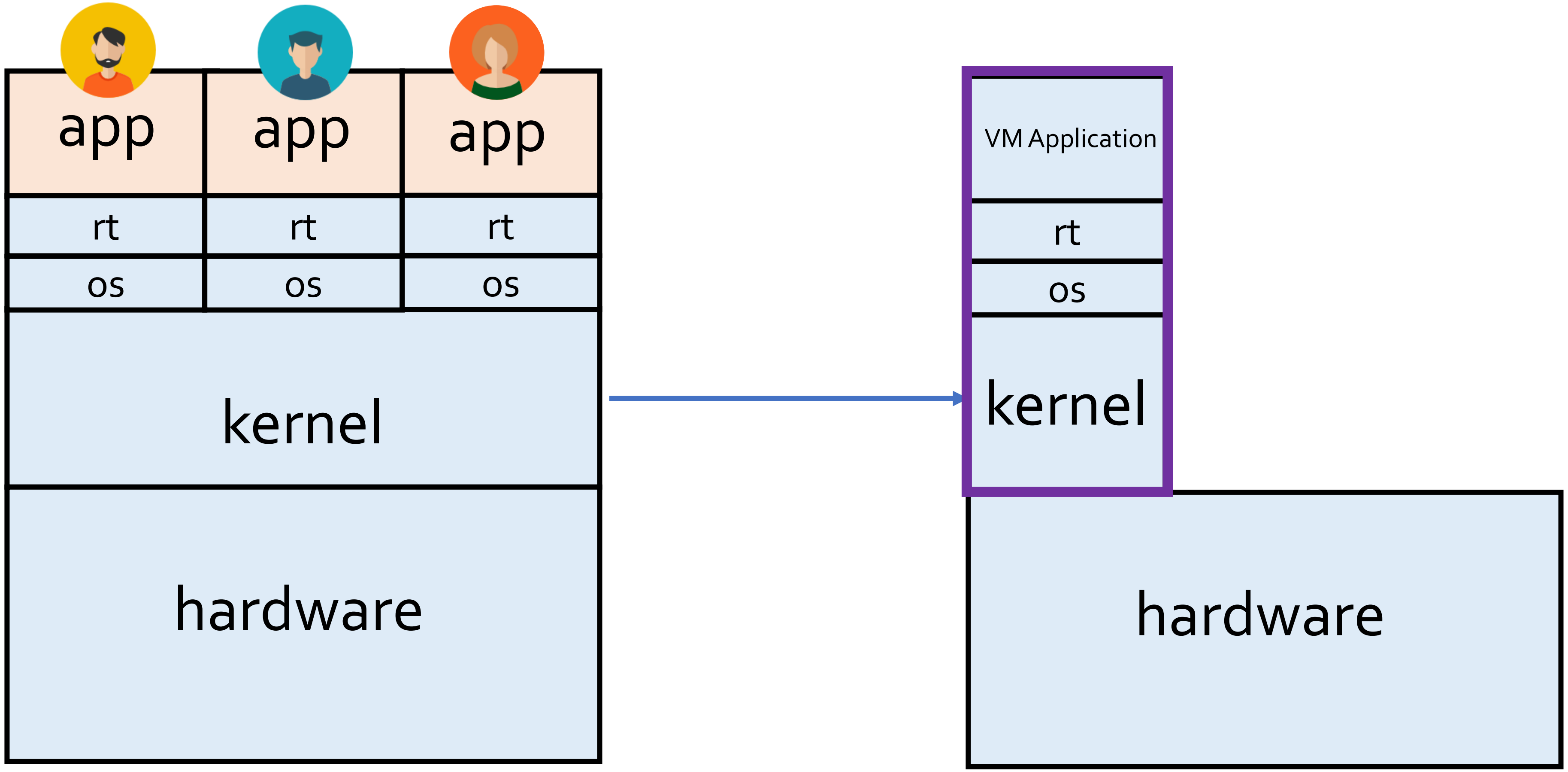
Topic 2: Improve security and control

Partially solved by running containers inside VMs



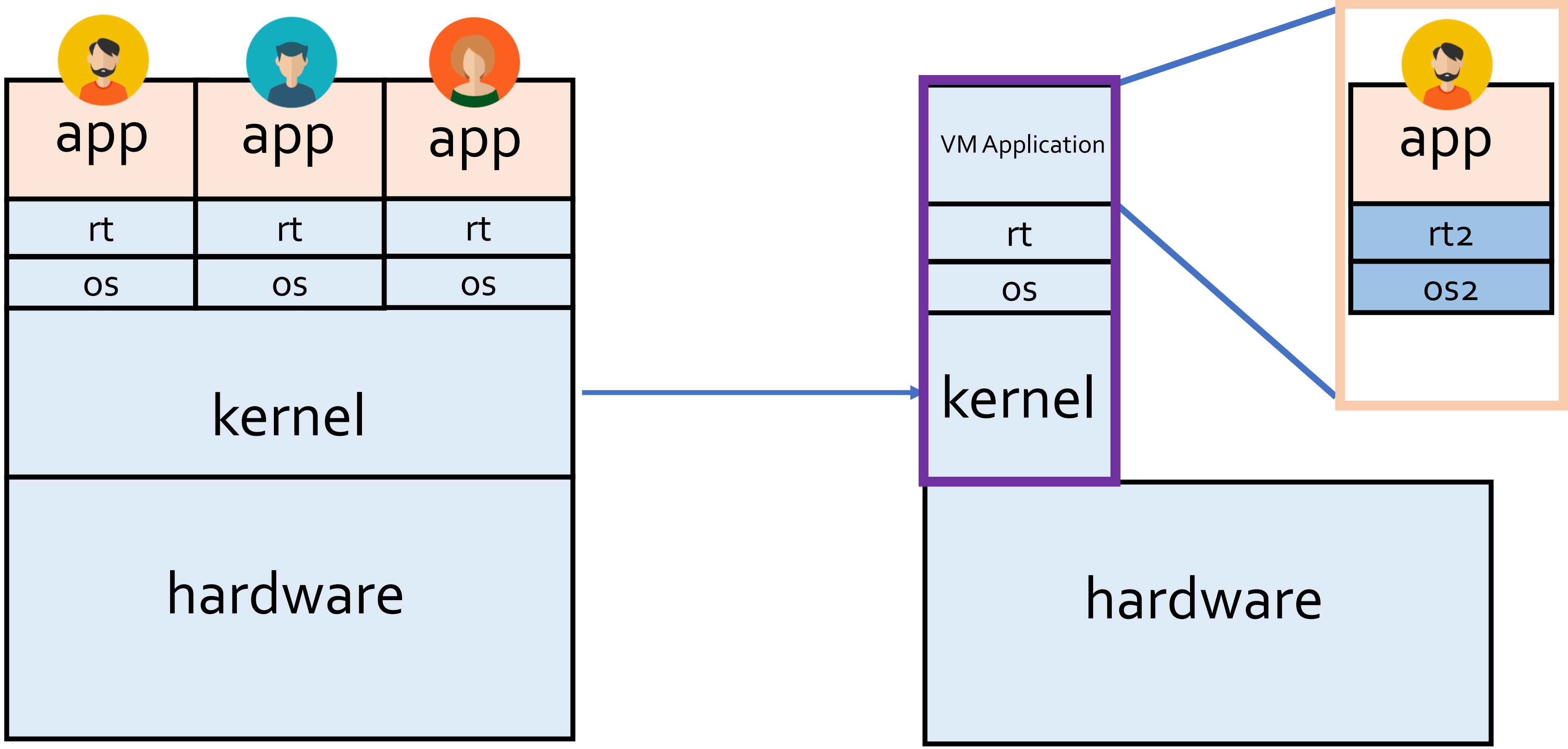
Topic 2: Improve security and control

Partially solved by running containers inside VMs



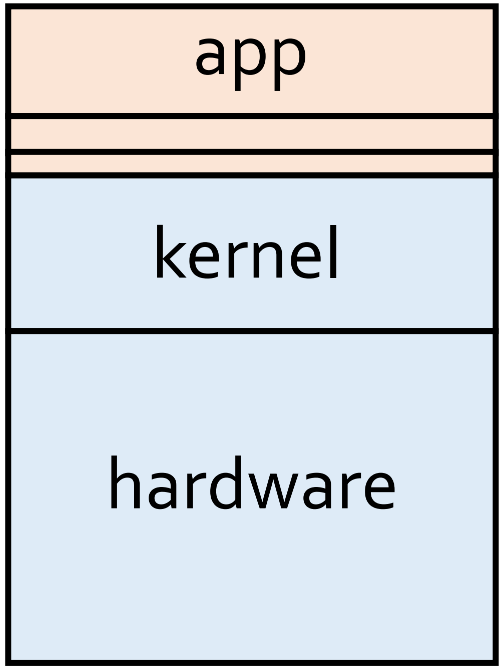
Topic 2: Improve security and control

Partially solved by running containers inside VMs



Topic 2: Improve security and control

MicroVMs

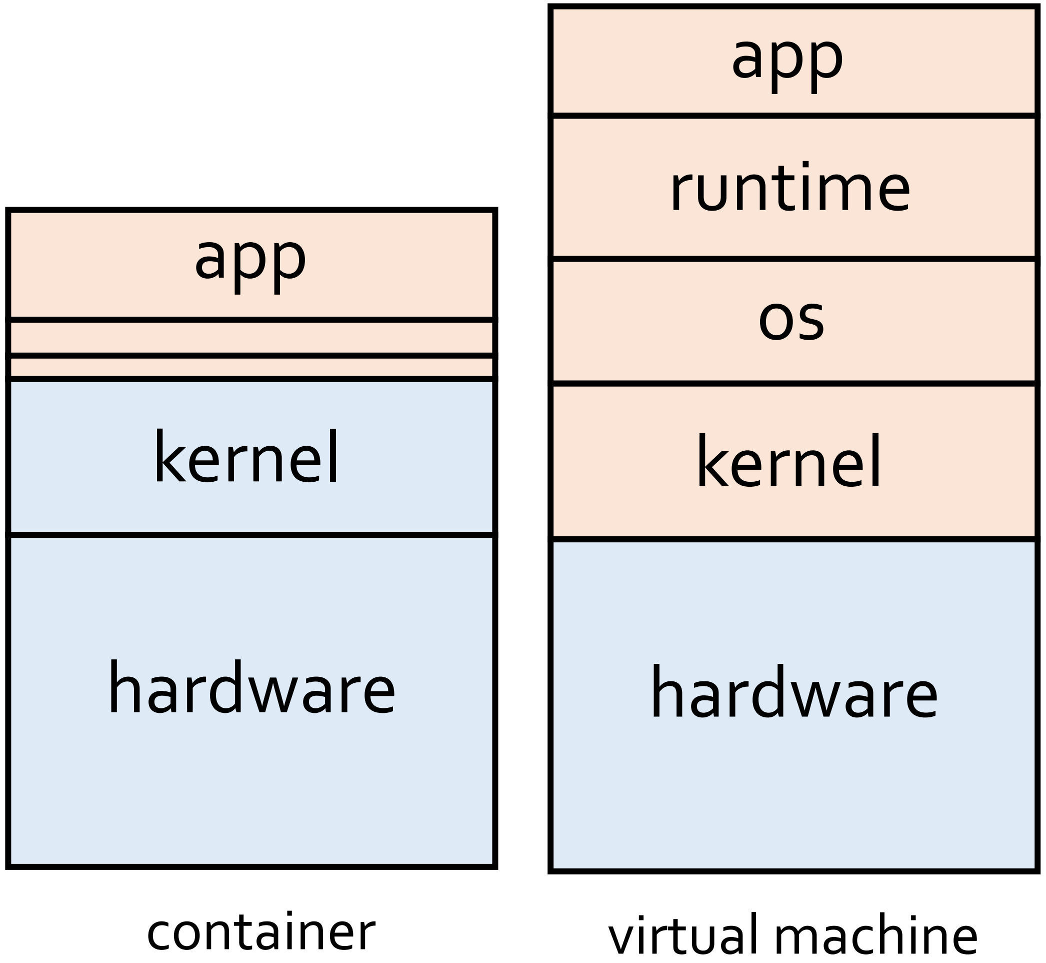


Tailor our app

container

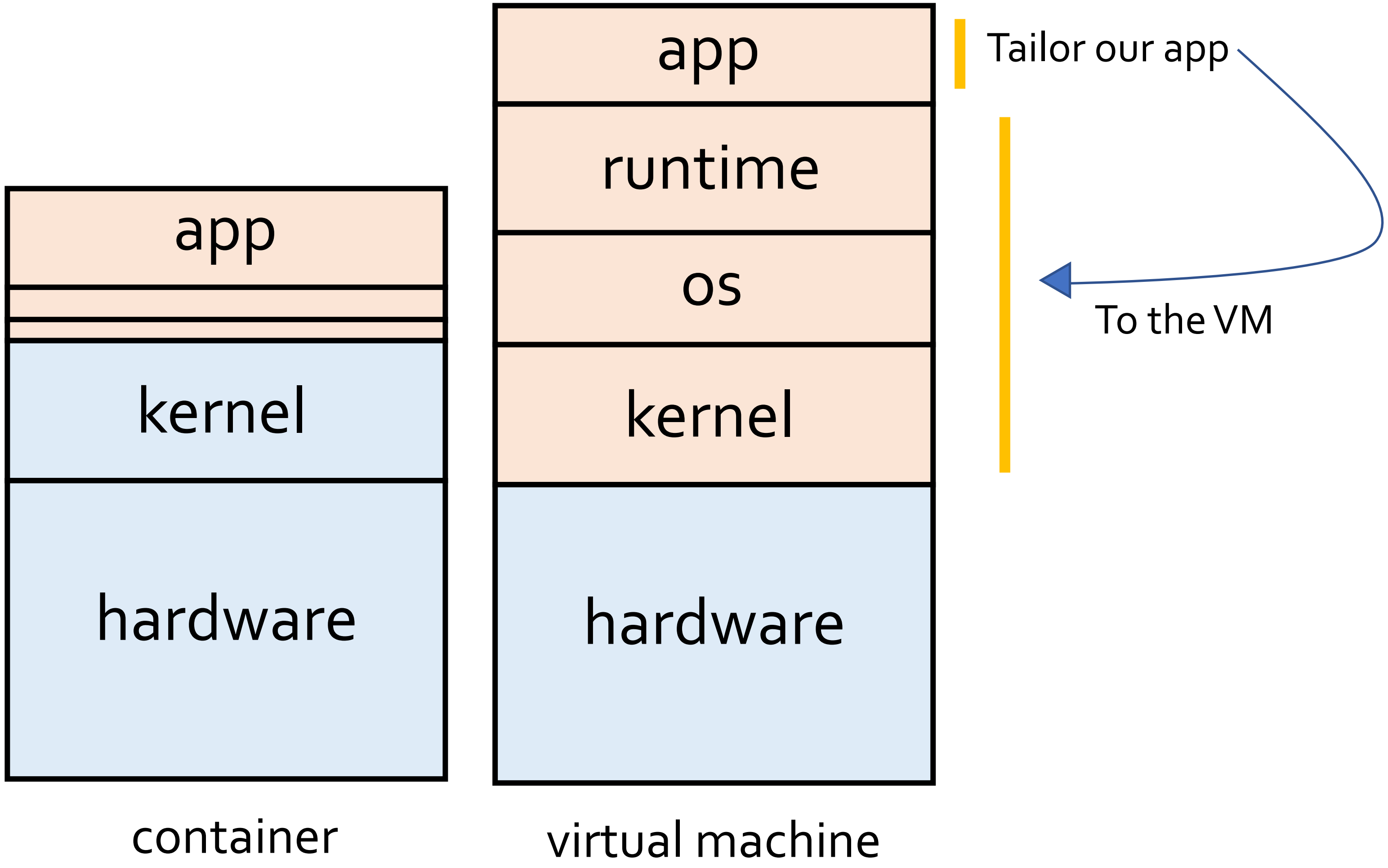
Topic 2: Improve security and control

MicroVMs

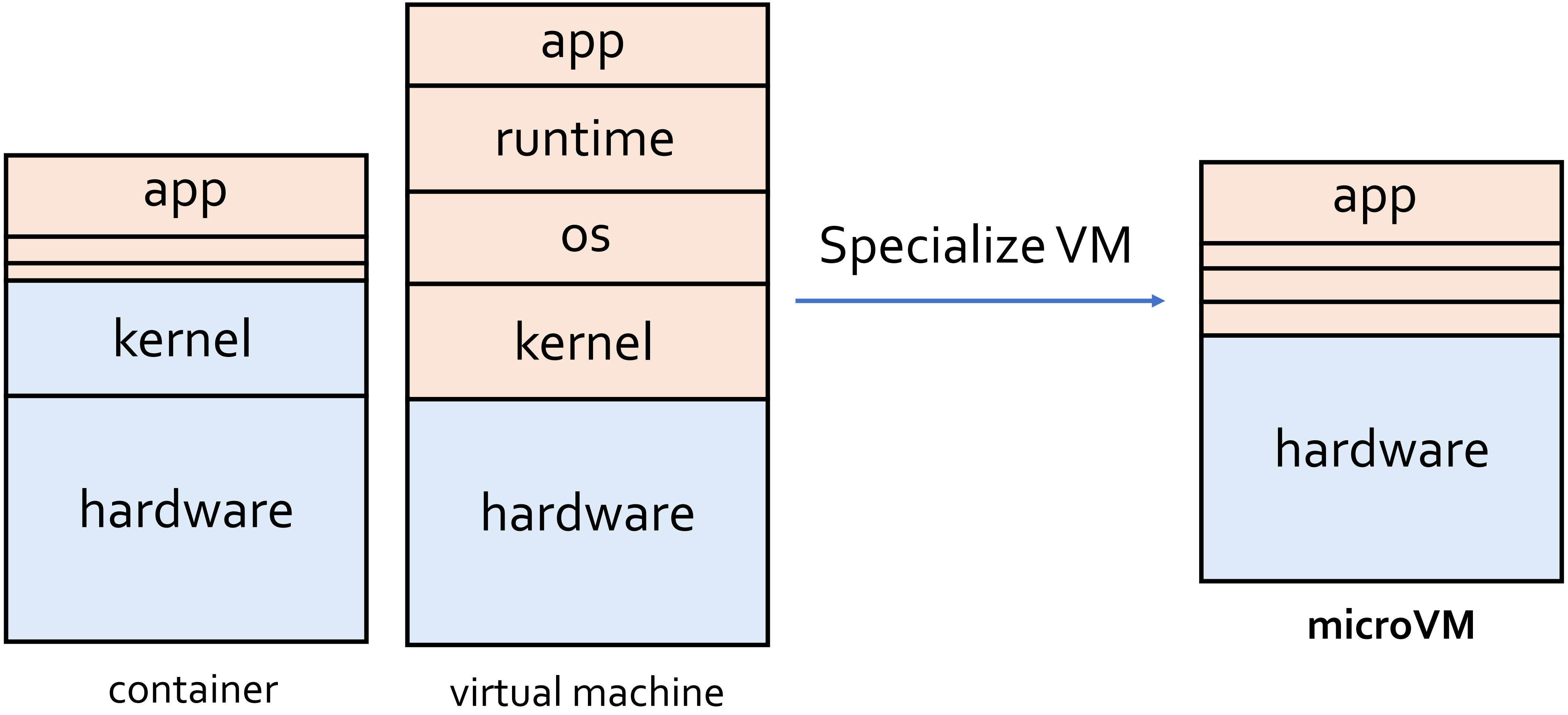


Tailor our app

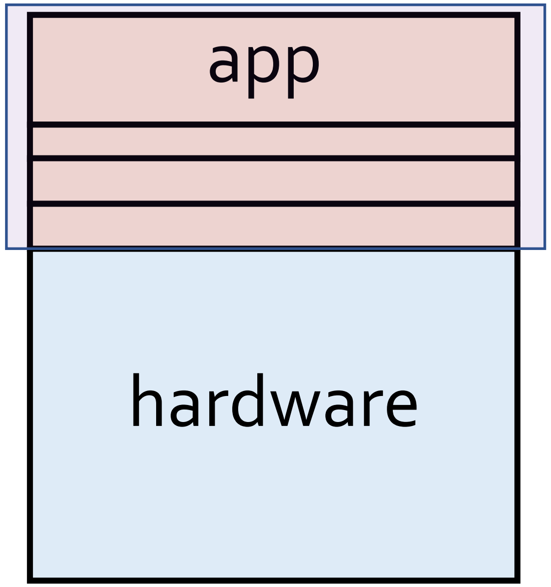
Topic 2: Improve security and control



Topic 2: Improve security and control



Topic 2: Improve security and control

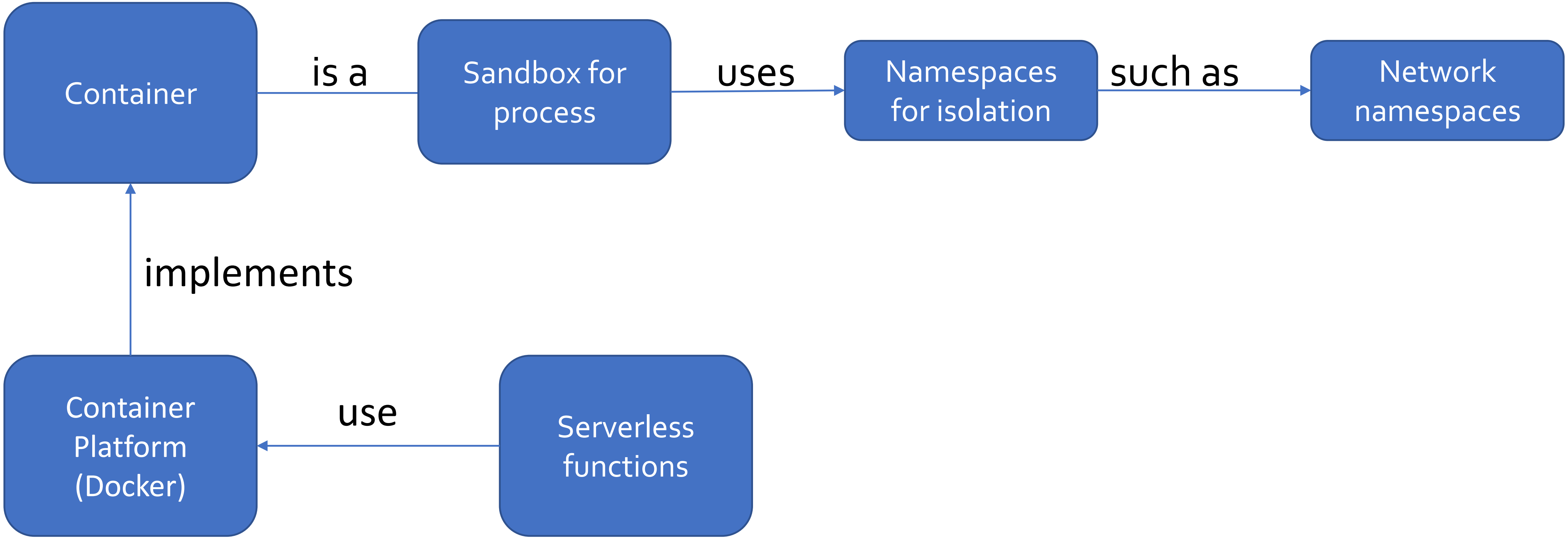


microVM

- Small image size
- Fast startup
- Better security
- Better resource control
- Harder to use/maintain
- Less portable

Conclusion

Review



Container

is a

Sandbox for process

uses

Namespaces for isolation

such as

Network namespaces

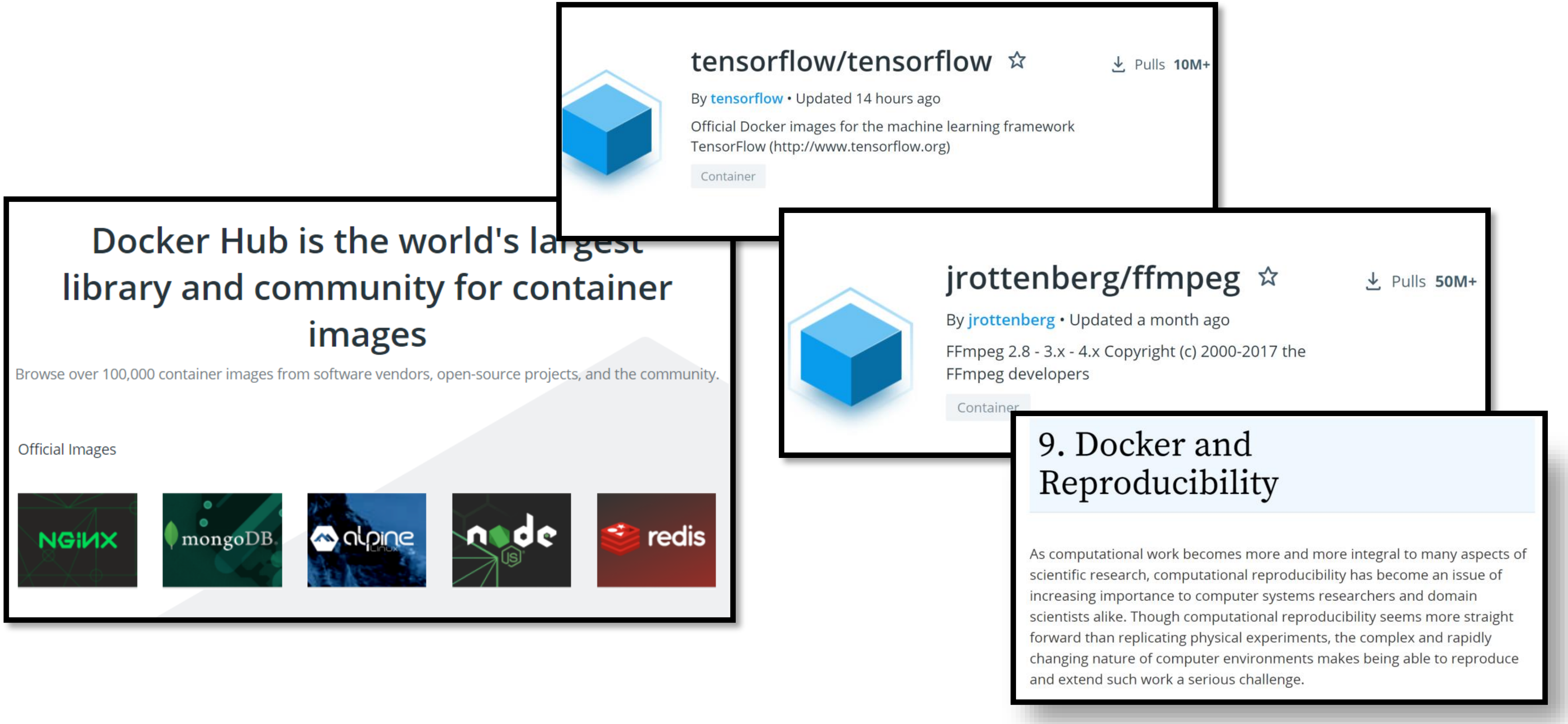
implements

Container Platform (Docker)

use

Serverless functions

Get exposure to software using containers



Docker Hub is the world's largest library and community for container images

Browse over 100,000 container images from software vendors, open-source projects, and the community.

Official Images

tensorflow/tensorflow ☆ Pulls 10M+

By tensorflow • Updated 14 hours ago

Official Docker images for the machine learning framework TensorFlow (http://www.tensorflow.org)

Container

jrottenberg/ffmpeg ☆ Pulls 50M+

By jrottenberg • Updated a month ago

FFmpeg 2.8 - 3.x - 4.x Copyright (c) 2000-2017 the FFmpeg developers

Container

9. Docker and Reproducibility

As computational work becomes more and more integral to many aspects of scientific research, computational reproducibility has become an issue of increasing importance to computer systems researchers and domain scientists alike. Though computational reproducibility seems more straight forward than replicating physical experiments, the complex and rapidly changing nature of computer environments makes being able to reproduce and extend such work a serious challenge.