

P.PORTO

Methods and Techniques for  
Software Development

Ricardo Santos | 2019/2020  
[rjs@estg.ipp.pt](mailto:rjs@estg.ipp.pt)

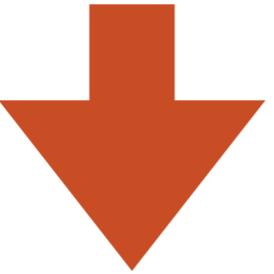
P.PORTO

# *Virtualization* Aula 2

Methods and Techniques for  
Software Development

2019/2020

*Businesses must innovate faster*



*Build better software, faster*

HOW?

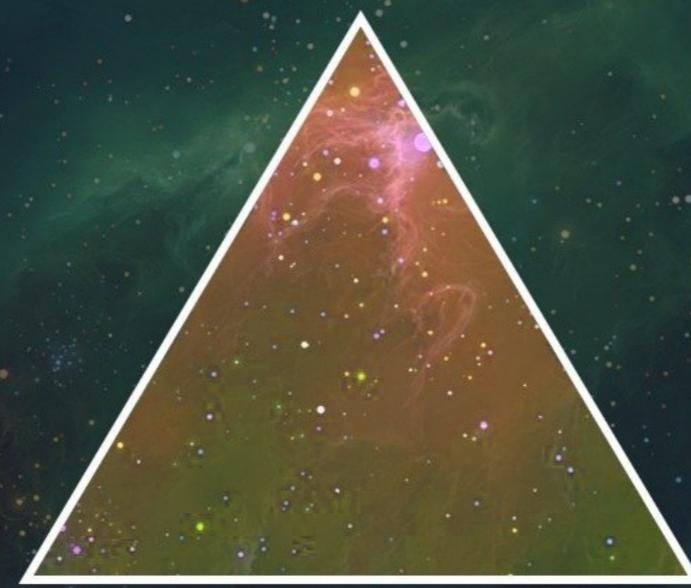
***Reducing lead time***

***Increasing deployment frequency***

# MODERN SOFTWARE DEVELOPMENT

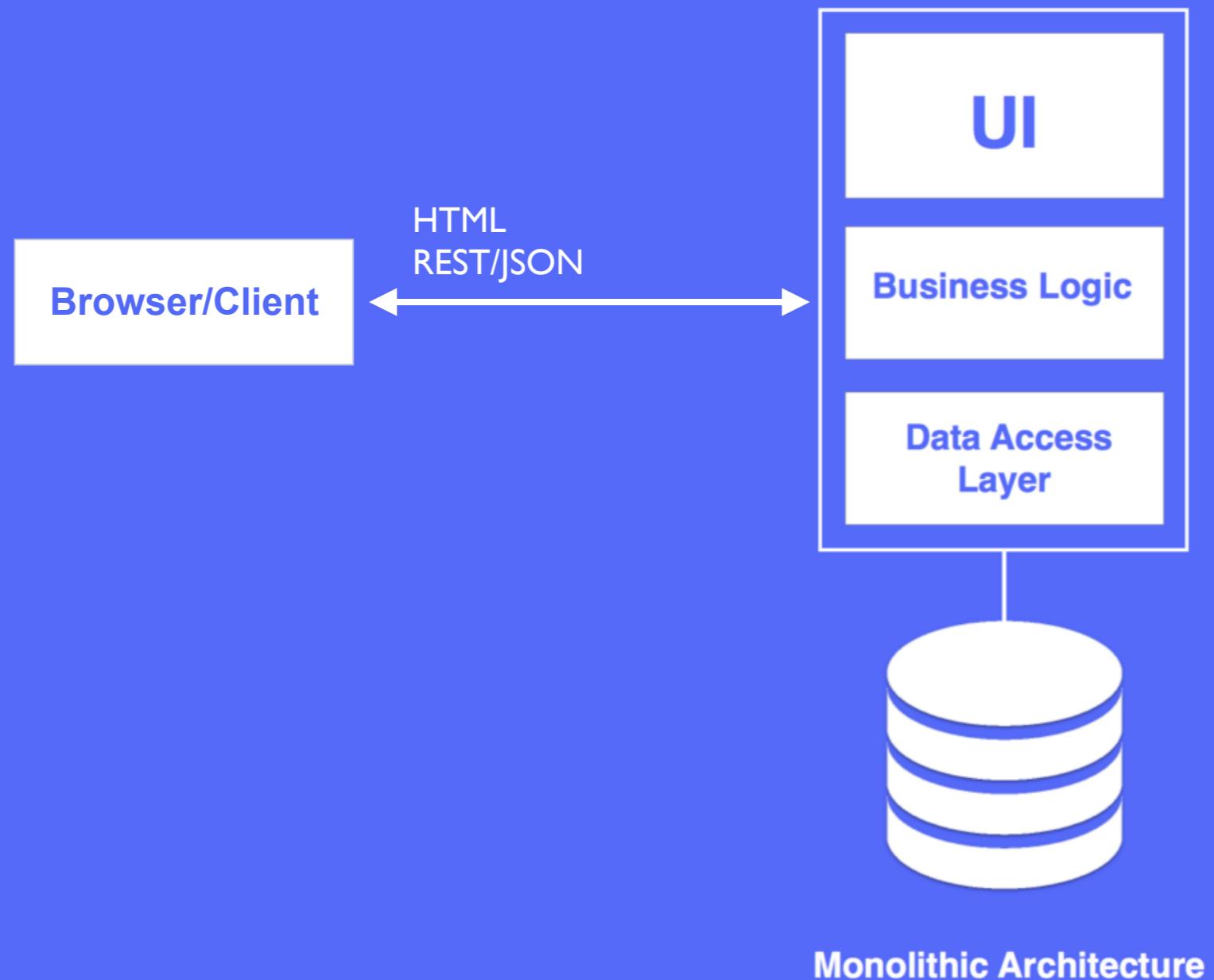


*Process: Continuous delivery/deployment*



*Organization:  
Small, autonomous  
teams*

*Architecture: ???*



*The monolithic architecture is an architectural style that structures the application as a single executable component*

# -ilities of Small Monoliths

- ▶ Maintainability
- ▶ Evolvability
- ▶ Testability
- ▶ Deployability
- ▶ Scalability



*But successful applications keep  
growing...*

Development  
Team

Application

*...and growing*

Development  
Team A

Development  
Team B

Development  
Team C

Application

A photograph of a massive stone pyramid, likely the Great Pyramid of Giza, under a blue sky with scattered white clouds. Two small figures of people are visible at the very top edge of the pyramid's steps.

agile development and  
deployment becomes  
impossible

monolithic hell

*Technology stack becomes increasingly  
obsolete*

**BUT**

*A rewrite is not feasible*

# Challenges

- ▶ Setting up the work environment
- ▶ Distributing work environment changes

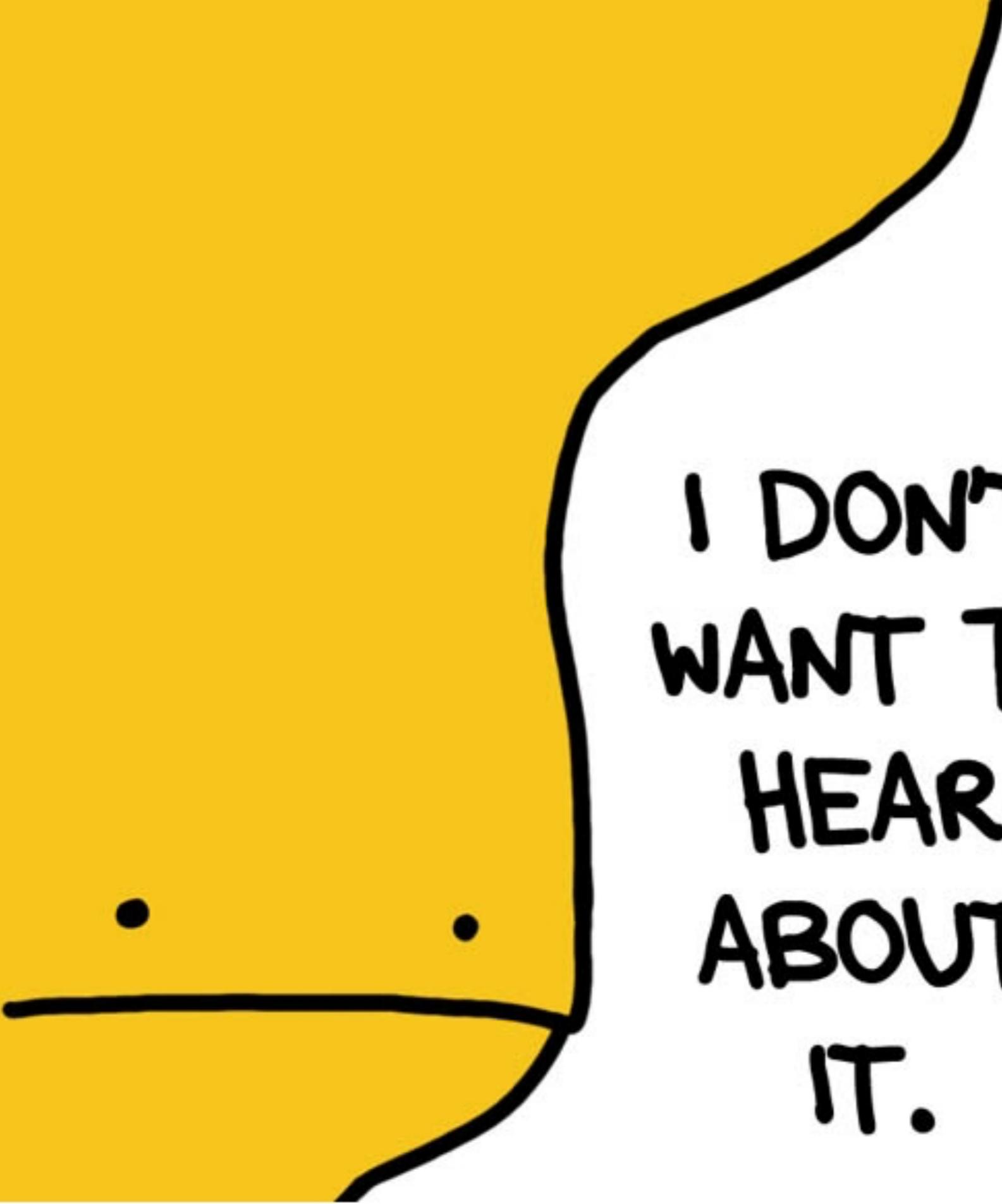
**“works on my machine”**

- ▶ Deployment

...up until now

*Solution Approach*

*Golden Image*



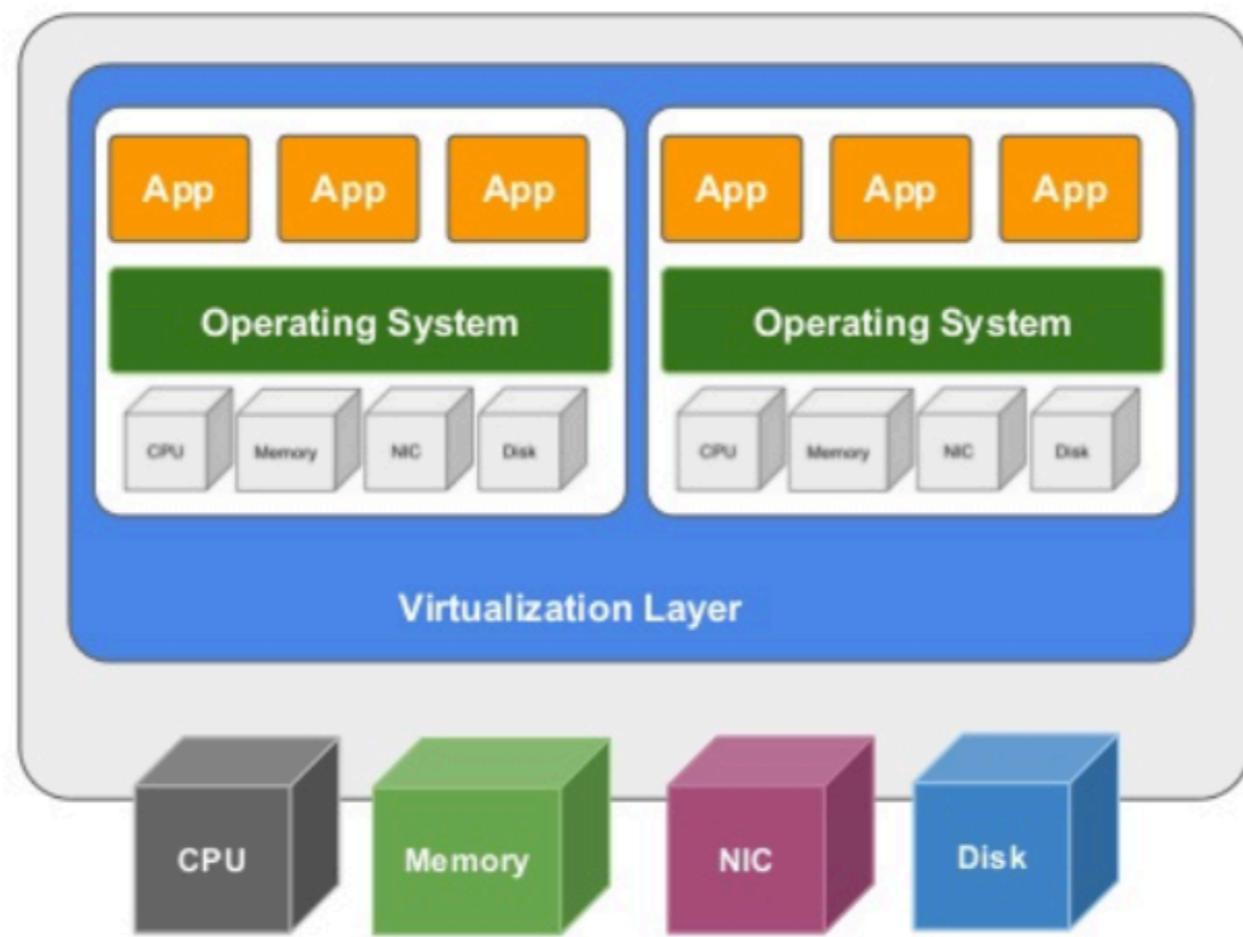
I DON'T  
WANT TO  
HEAR  
ABOUT  
IT.

# Golden image: Problems

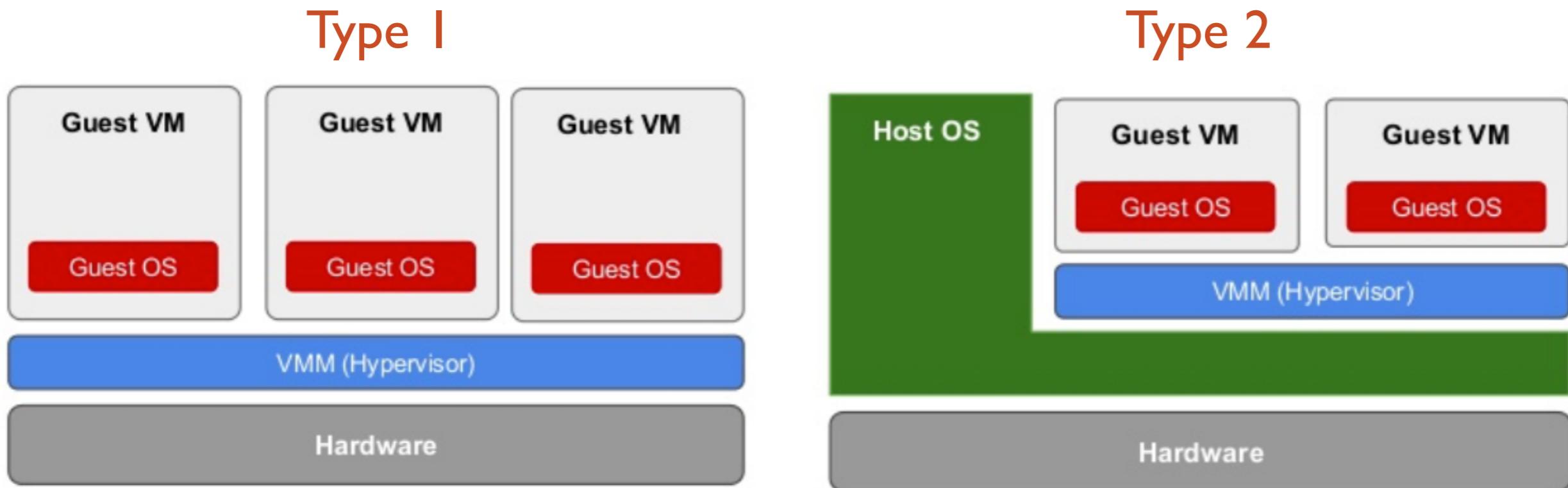
- ▶ Big
- ▶ Distribution takes long
- ▶ Simple customising is difficult
- ▶ Every small change leads to a big refactor and to a complete reinstallation
- ▶ No collaboration
- ▶ Versioning is difficult

*Familiar with Virtualization?*

The virtualization of computers as **complete hardware platforms**, certain logical abstractions of their components, or only the functionality required to run **various operating systems**



# Depending on what sits right on Hardware



Bare metal architecture:

- Mostly for server
- VMM by default
- OS-independent VMM

Hosted architecture:

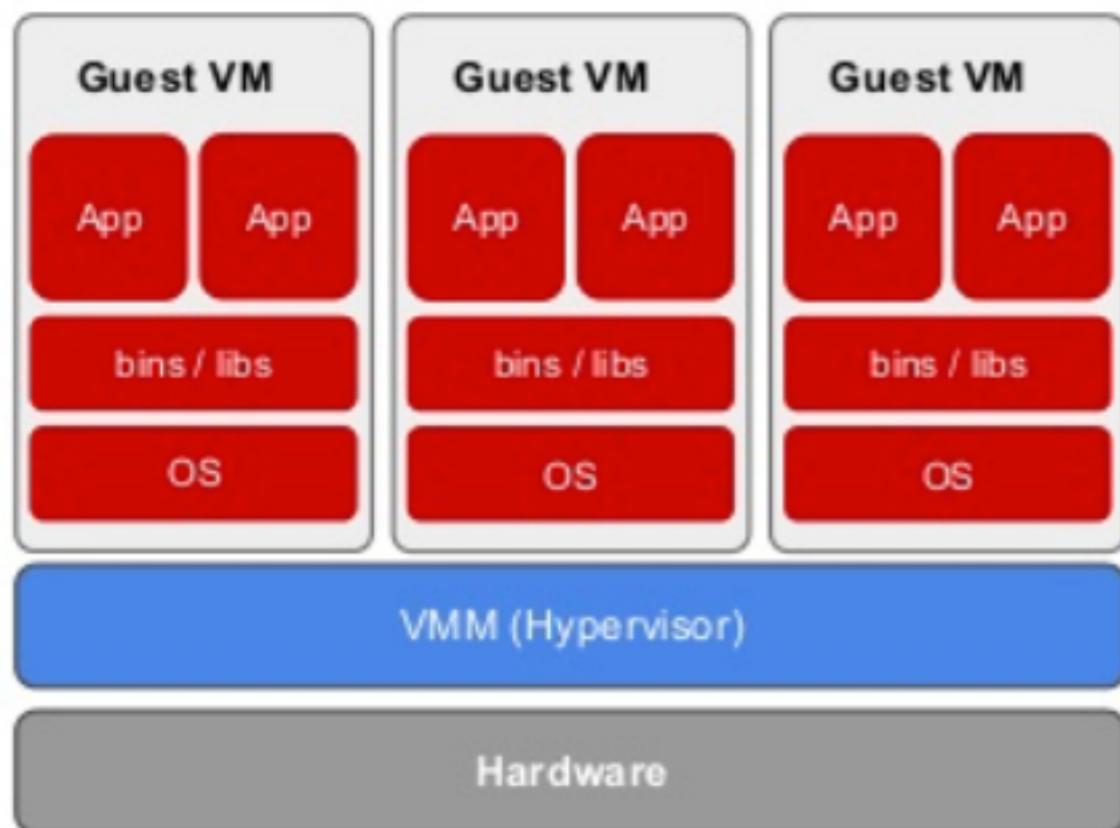
- Mostly for client devices
- VMM on demand
- OS-dependent VMM

# Linux Container

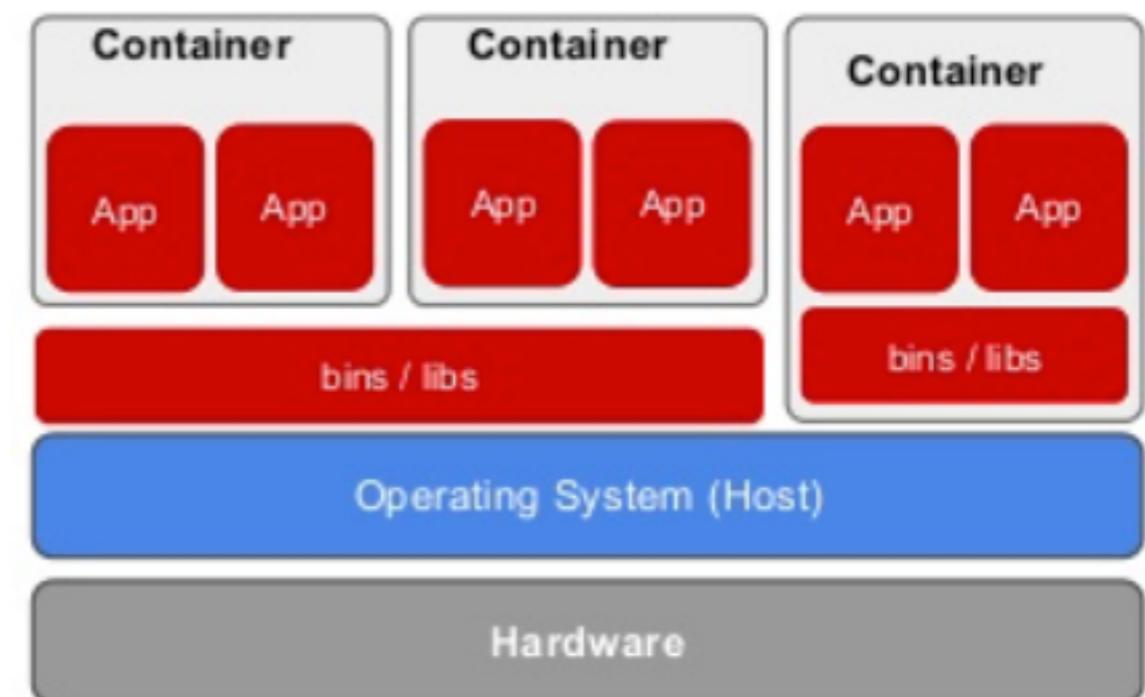
- ▶ Lightweight virtualization
- ▶ OS-level virtualization
- ▶ Allow single host to operate multiple isolated & resource-controlled Linux Instances
- ▶ includes in the linux kernel called LXC

**Containers are not a new technology...**

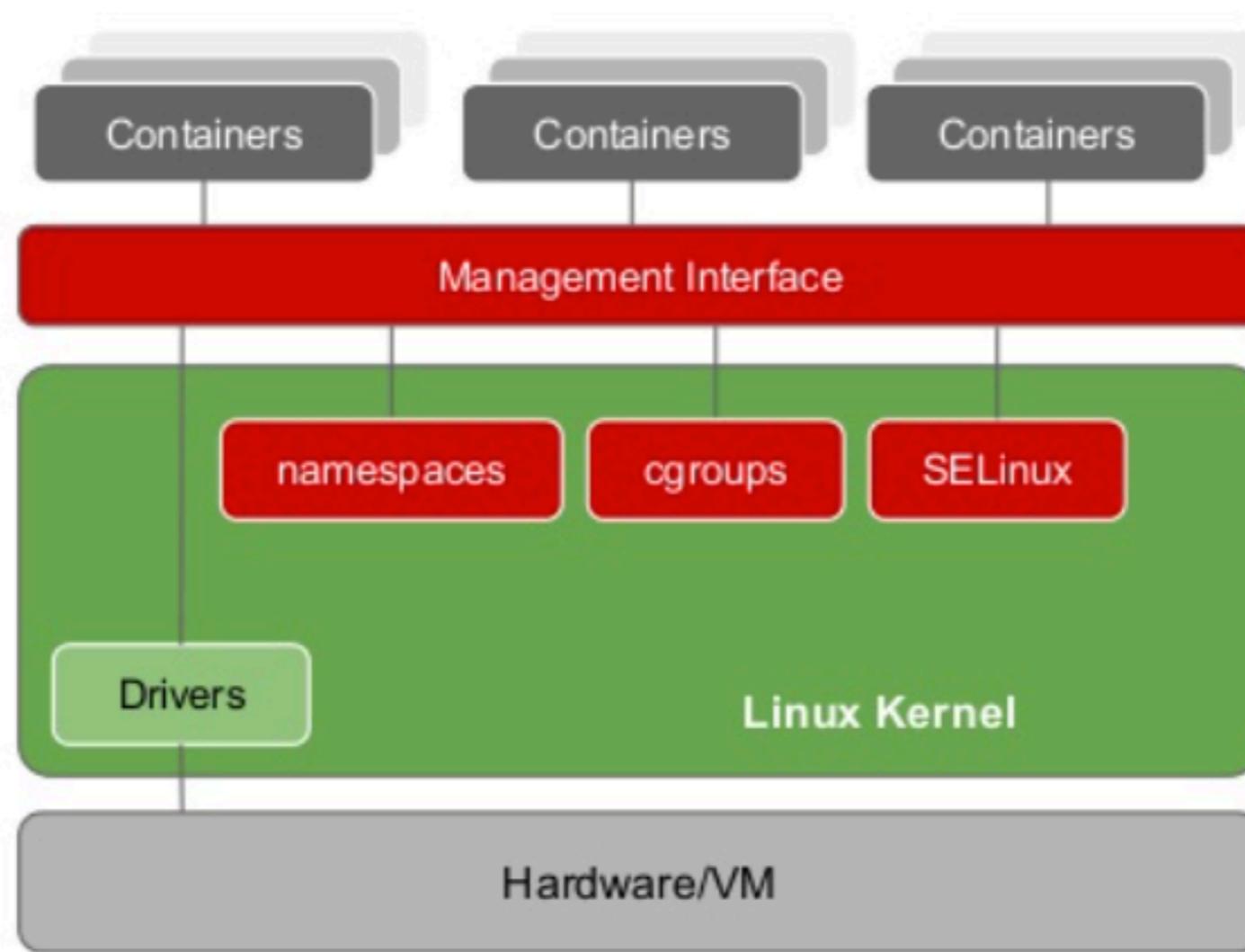
# Type I Hypervisor



# Linux Container



# Container Architecture



*Container supports separation of various resources. They are internally realized with different technologies called "namespace"*

# Container Technologies

cloudfoundry/heroku/dotcloud/appfog/openshift

docker

LXC

libvirt

warden

lmcfy

openvz

grsec  
apparmor  
SELinux

namespace/cgroups/netfilter/tc/veth/quota/union fs



VAGRANT

development environment on

- ▶ VM
- ▶ container
- ▶ cloud

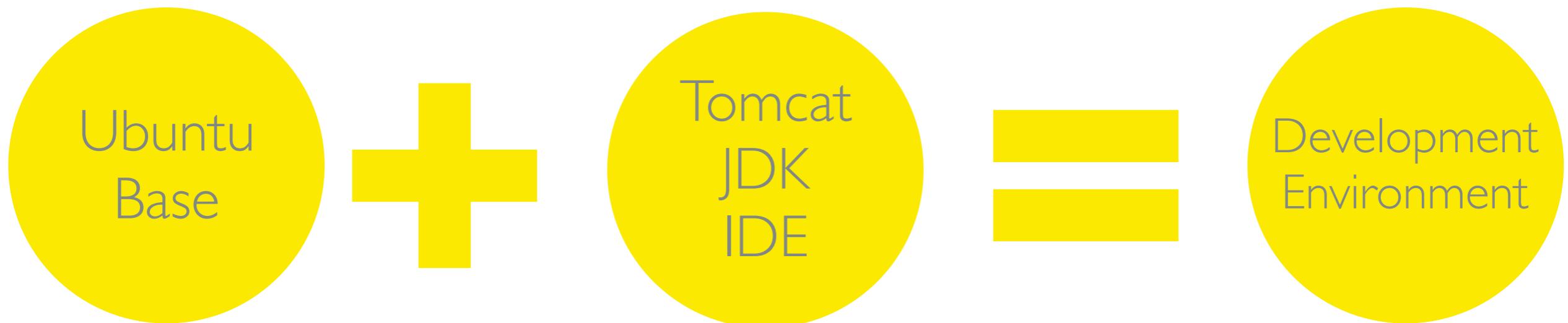
same

- ▶ among team members
- ▶ among production and development

## Base Box

- ▶ Pre-assembled Vagrant VM image, ready-to-run
- ▶ Custom build possible
- ▶ Base Box is base for further provisioning

# Provisioning



# Provisioning

Small Base Box

High flexibility

Long duration  
of provisioning

Bigger Base Box

Low flexibility

Shorter duration  
of provisioning



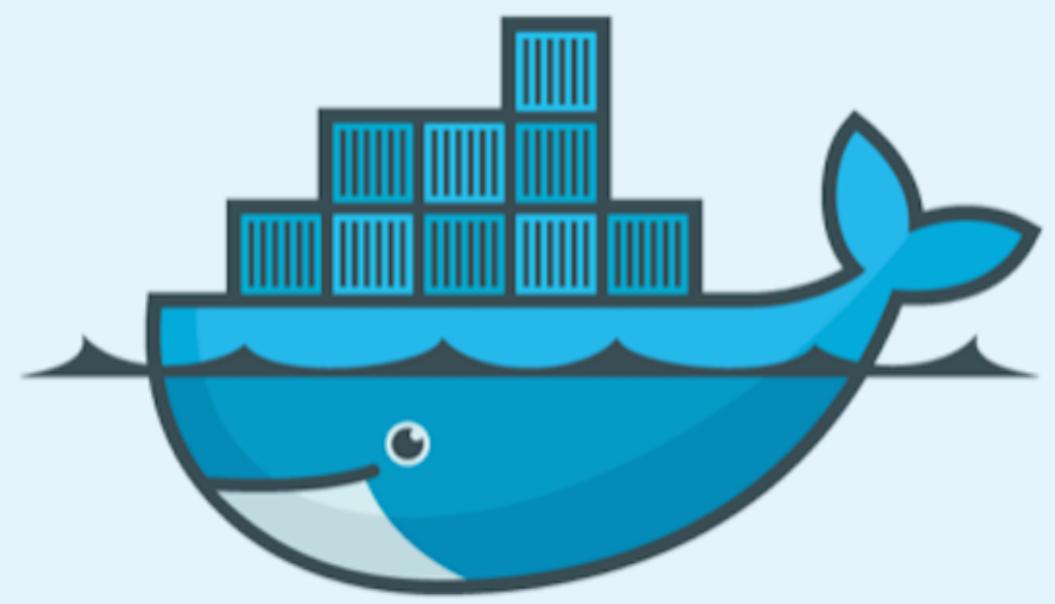


VAGRANT

## *Hands on lab*

<https://www.vagrantup.com/intro/getting-started/>

```
$ vagrant up (start or create the VM)
$ vagrant ssh (ssh to the VM)
$ vagrant halt (stop the VM)
$ vagrant destroy (destroy the VM)
$ vagrant package (snapshot the VM)
```



docker

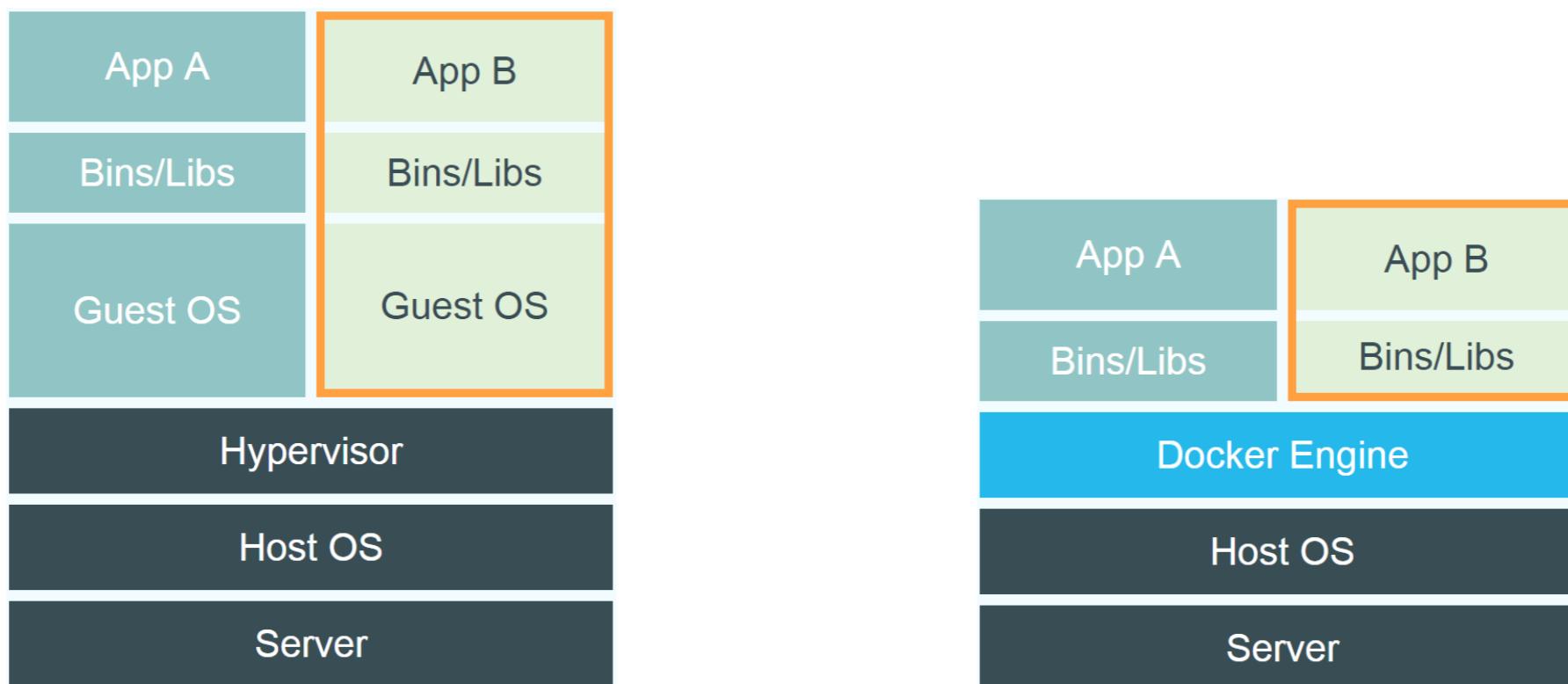
*Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system-level virtualization on Linux.*

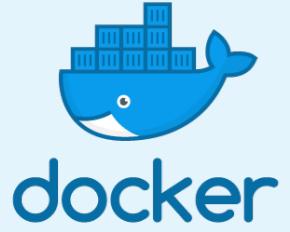
Provides a uniformed wrapper around a software package:

## *«Build, Ship and Run Any App, Anywhere»*

- Similar to shipping containers: The container is always the same, regardless of the contents and thus fits on all trucks, cranes, ships, ...

# Docker vs Virtual Machine





## *Hands on lab*

<https://github.com/docker/labs/tree/master/beginner>

development environment on

- ▶ VM
- ▶ container
- ▶ cloud

same

- ▶ among team members
- ▶ among production and development

P.PORTO

Polytechnic of  
Porto

ESTG - School of  
Management and Technology