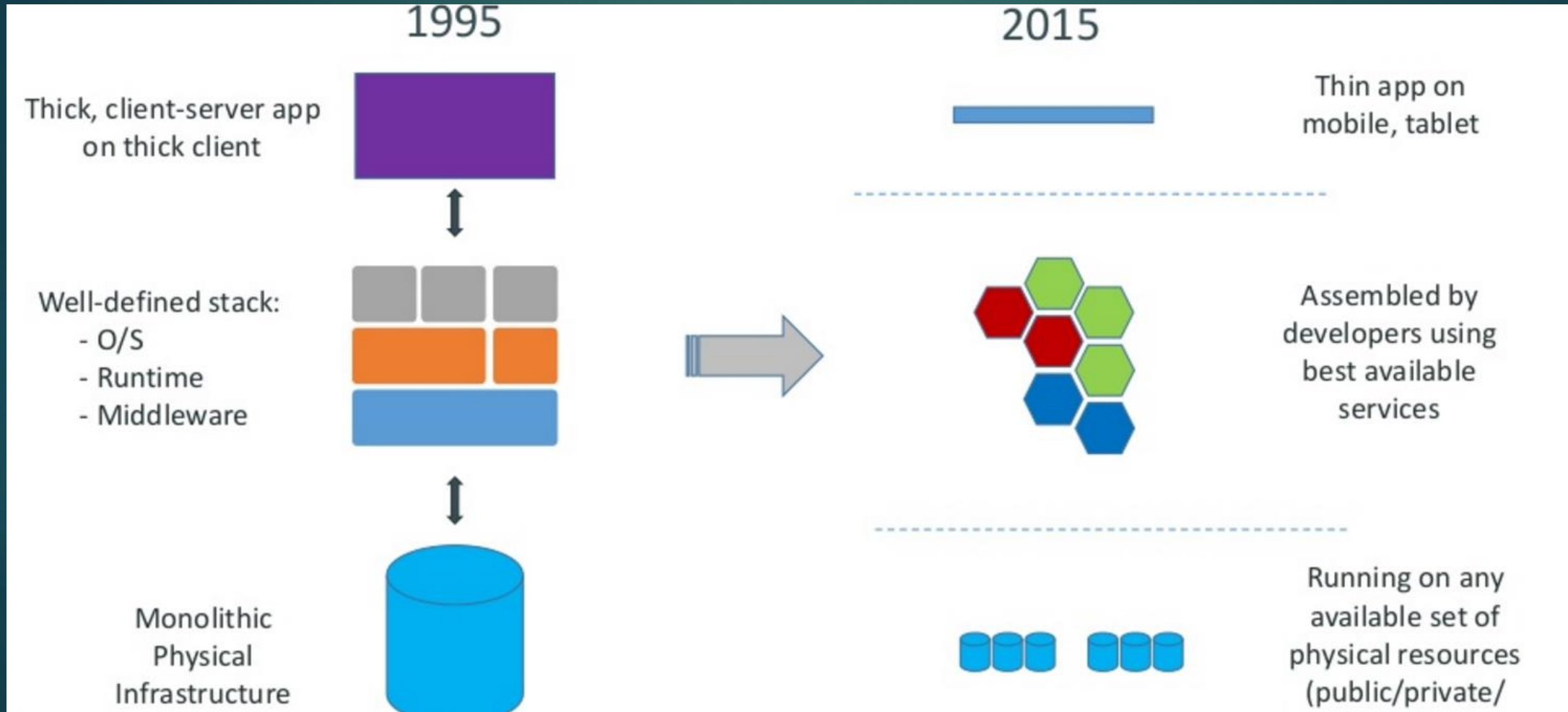




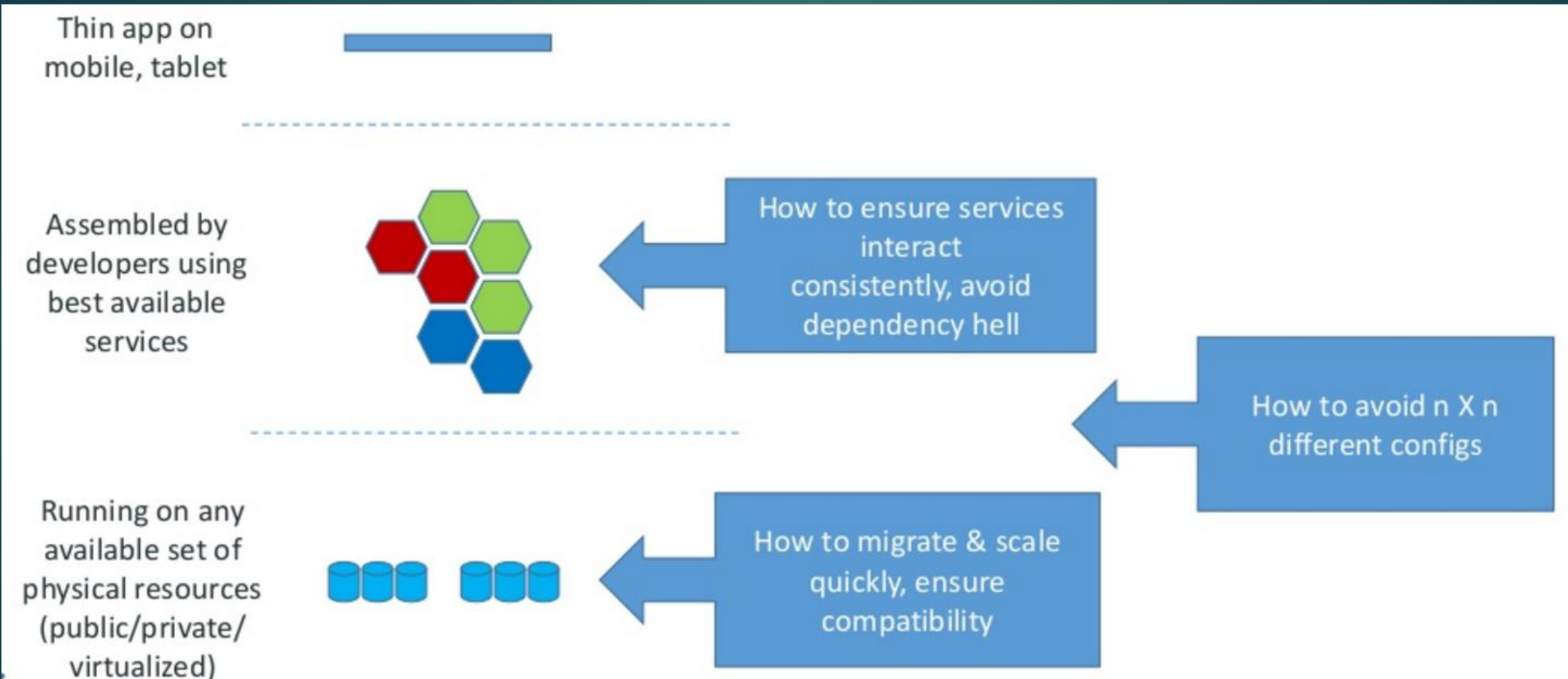
# Motivation

2



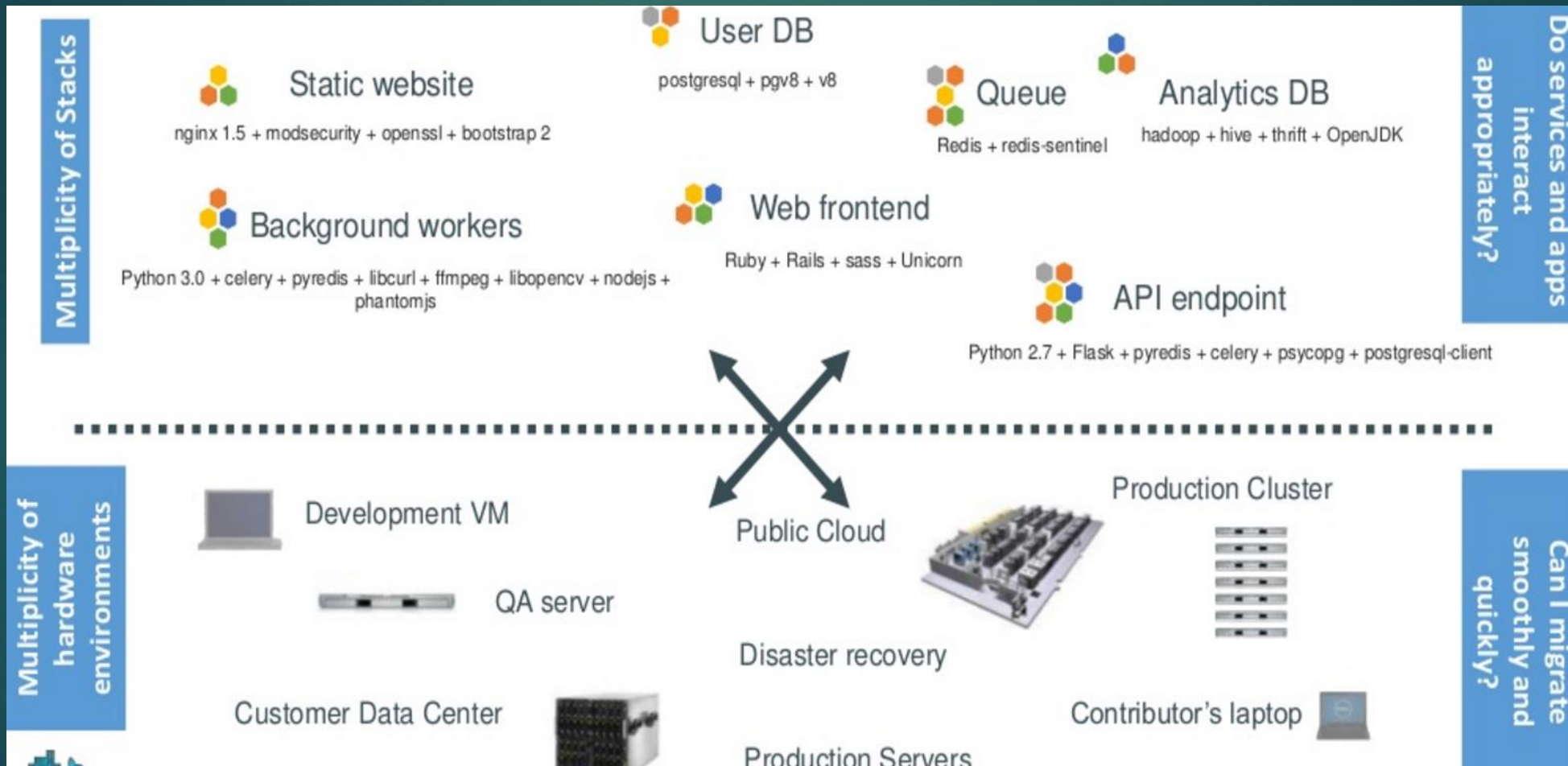
# Motivation –the challenge

3



# The challenge continued

4



# Looking for all kinds of solutions...

5

- ▶ Too many to consider

Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers

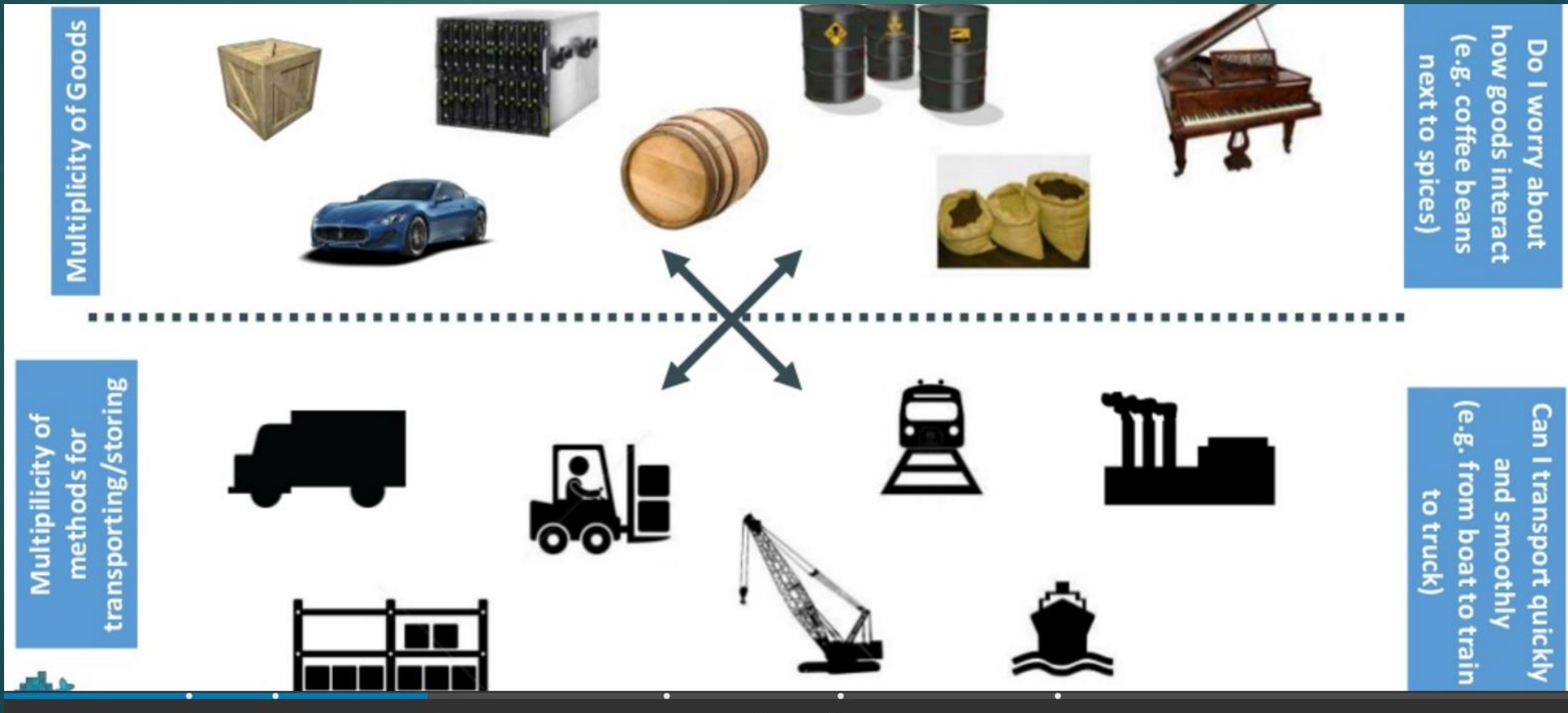




# Understanding...an analogy














6

## ...cargo transport pre-1960



# What are the possibilities

7

	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

# SOLUTION—shipping containers

8

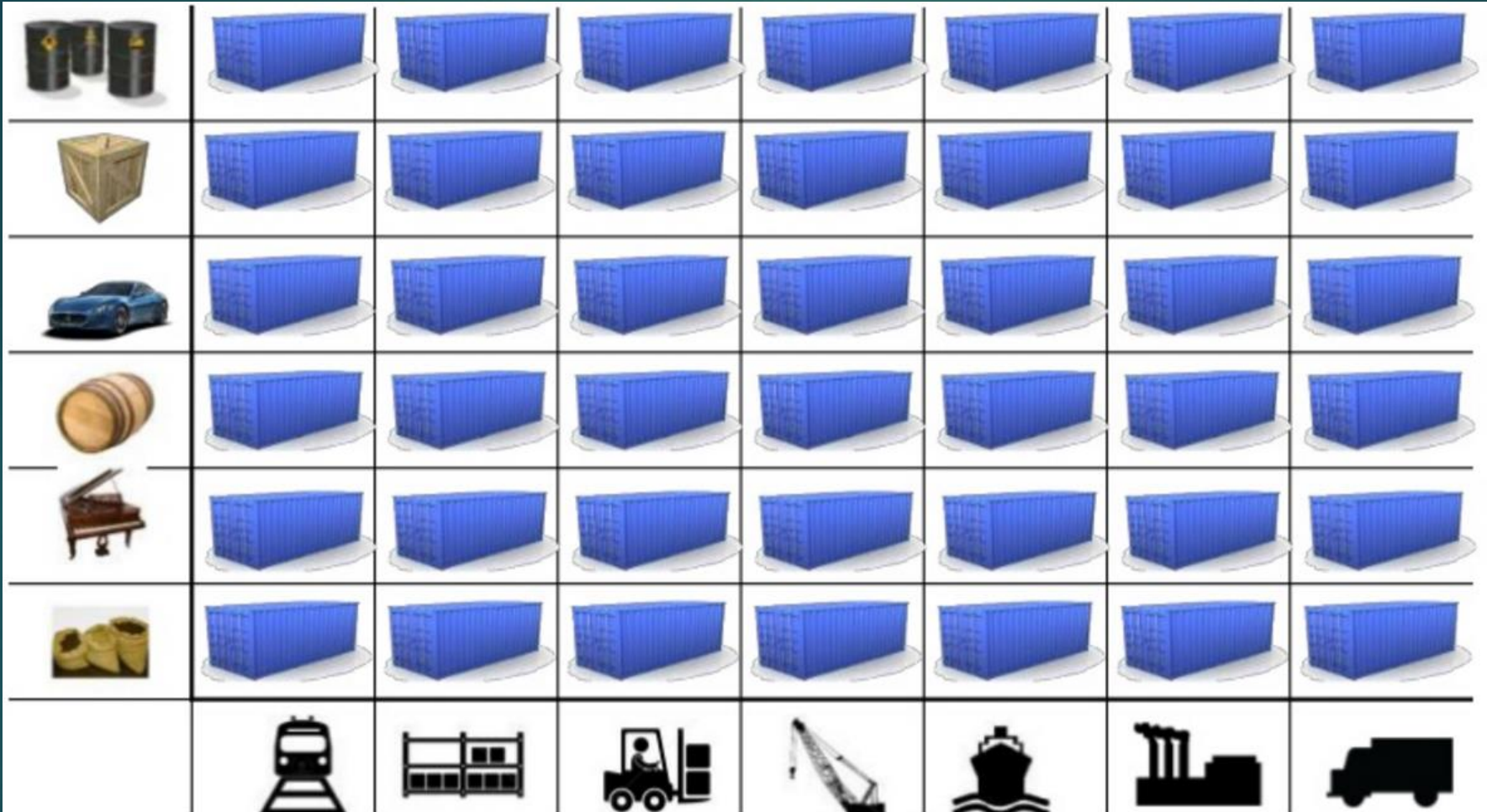
© 2018 CHAITANYA GAJULA - ALL RIGHTS RESERVED





# This solved the problem

9



# Today shipping is done with containers

10



- 90% of all cargo now shipped in a standard container
- Order of magnitude reduction in cost and time to load and unload ships
- Massive reduction in losses due to theft or damage
- Huge reduction in freight cost as percent of final goods (from >25% to <3%)
- massive globalizations
- 5000 ships deliver 200M containers per year

# How does this container idea translate to our problem

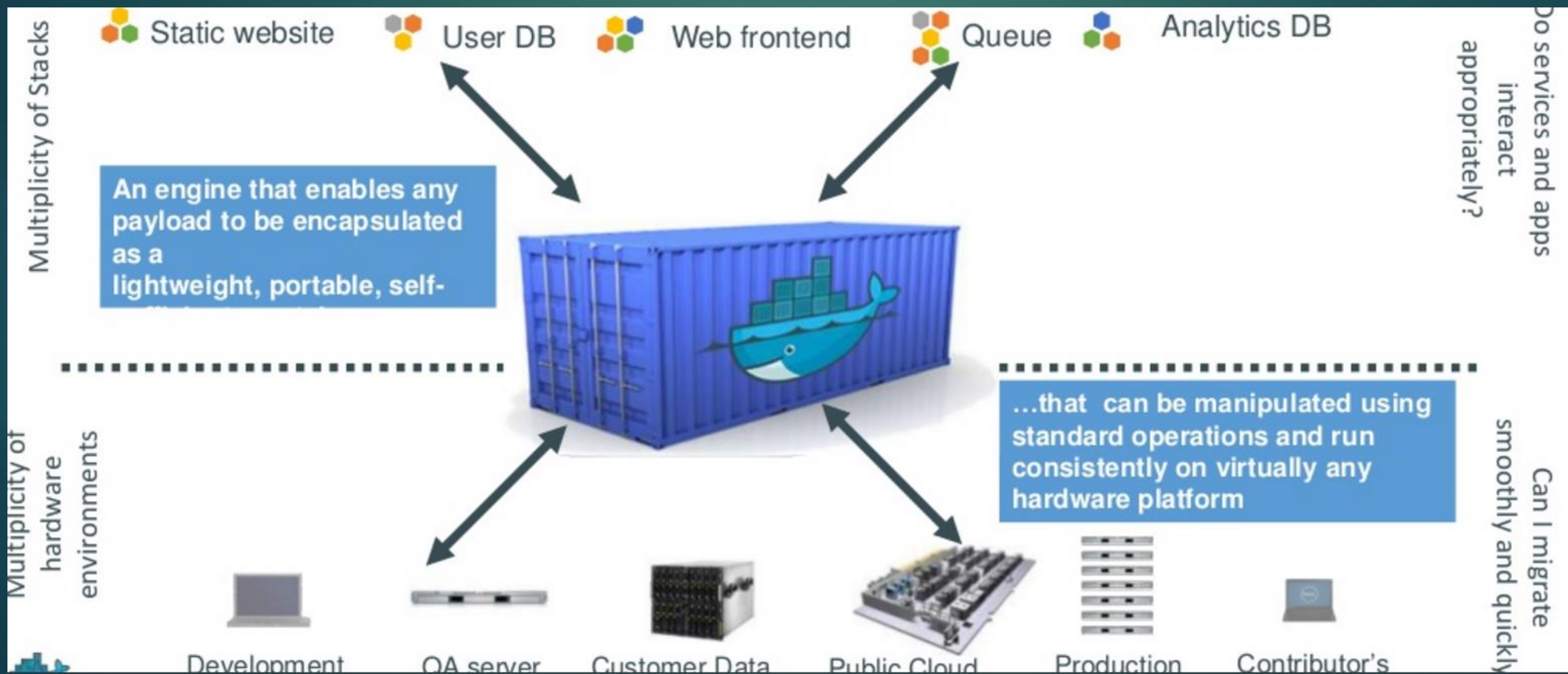
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers





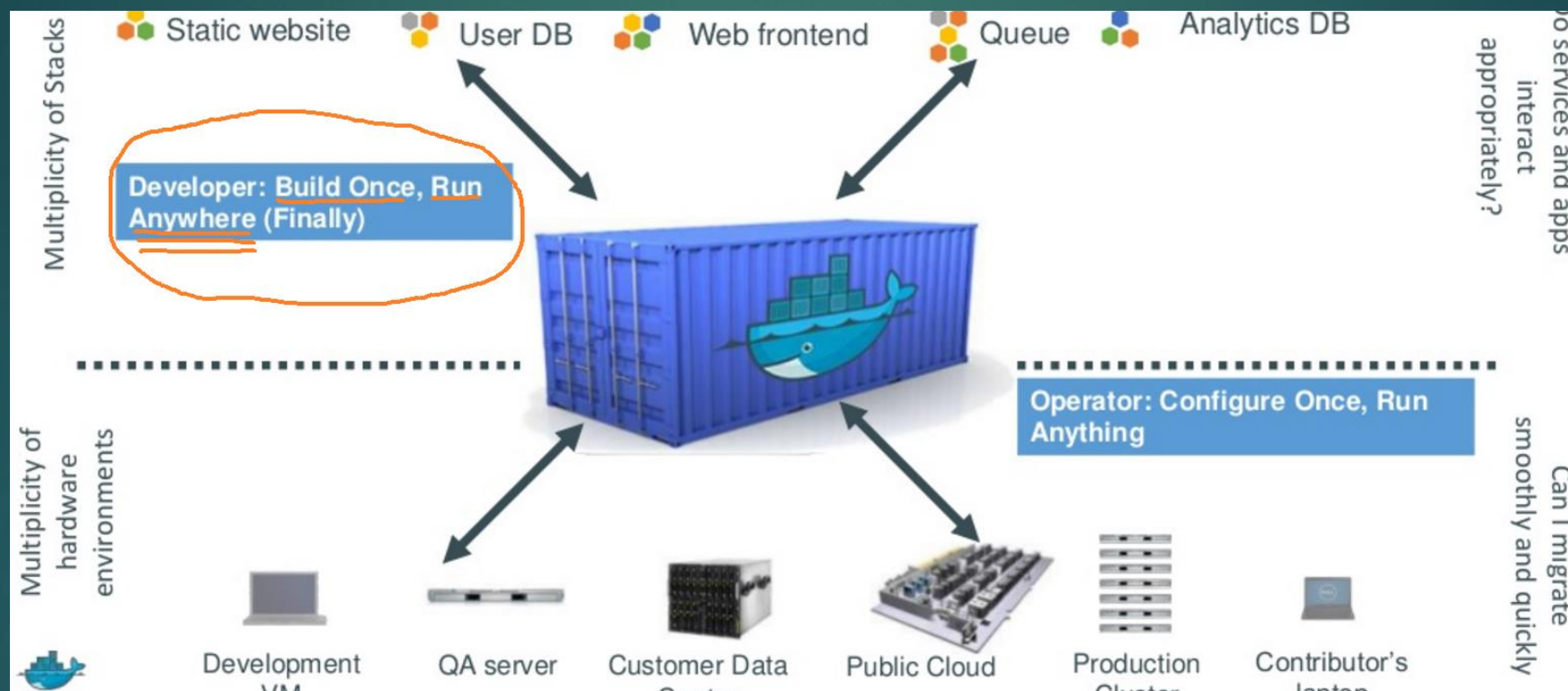
# How does this container idea translate to our problem—container for code????

12



# Do once run anywhere

13





# Docker's container ---the concept (and relation to our shipping container)

14

	Physical Containers	Docker
Content Agnostic	The same container can hold almost any type of cargo	Can encapsulate any payload and its dependencies
Hardware Agnostic	Standard shape and interface allow same container to move from ship to train to semi-truck to warehouse to crane without being modified or opened	Using operating system primitives (e.g. LXC) can run consistently on virtually any hardware—VMs, bare metal, openstack, public IAAS, etc.—without modification
Content Isolation and Interaction	No worry about anvils crushing bananas. Containers can be stacked and shipped together	Resource, network, and content isolation. Avoids dependency hell
Automation	Standard interfaces make it easy to automate loading, unloading, moving, etc.	Standard operations to run, start, stop, commit, search, etc. Perfect for devops: CI, CD, autoscaling, hybrid clouds
Highly efficient	No opening or modification, quick to move between waypoints	Lightweight, virtually no perf or start-up penalty, quick to move and manipulate
Separation of duties	Shipper worries about inside of box, carrier worries about outside of box	Developer worries about code. Ops worries about infrastructure.

# Docker supported in many Cloud platforms

15



# Docker container—developer viewpoint

16

## Build once...run anywhere

- A clean, safe, hygienic and portable runtime environment for your app.
- No worries about missing dependencies, packages and other pain points during subsequent deployments.
- Run each app in its own isolated container, so you can run various versions of libraries and other dependencies for each app without worrying
- Automate testing, integration, packaging...anything you can script
- Reduce/eliminate concerns about compatibility on different platforms, either your own or your customers.
- Cheap, zero-penalty containers to deploy services? A VM without the overhead of a VM? Instant replay and reset of image snapshots? That's the power of Docker



## Developer viewpoint---(doesn't this quote remind you of Java virtual machine)

17

“Docker interests me because it allows simple environment isolation and repeatability. I can create a run-time environment once, package it up, then run it again on any other machine. Furthermore, everything that runs in that environment is isolated from the underlying host (much like a virtual machine). And best of all, everything is fast and simple.”

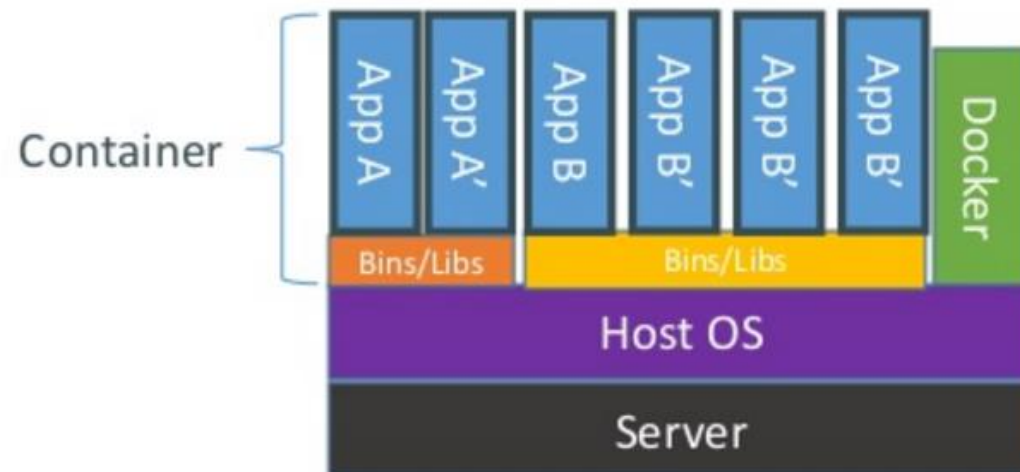
-Gregory Szorc, Mozilla Foundation

<http://gregoryszorc.com/blog/2013/05/19/using-docker-to-build-firefox/>

# How does Docker containers work?

18

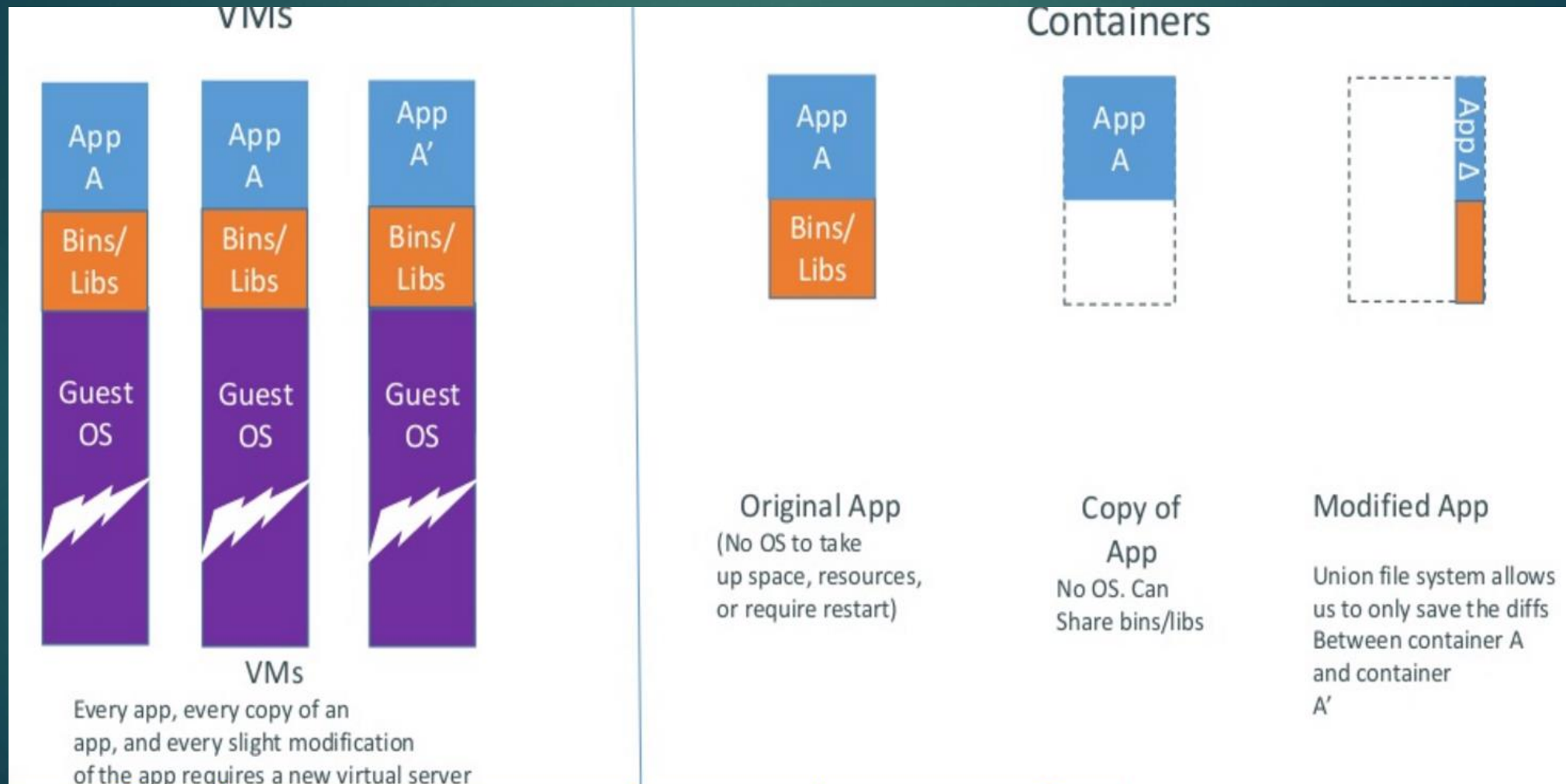
Containers are isolated,  
but share OS and, where  
appropriate, bins/libraries





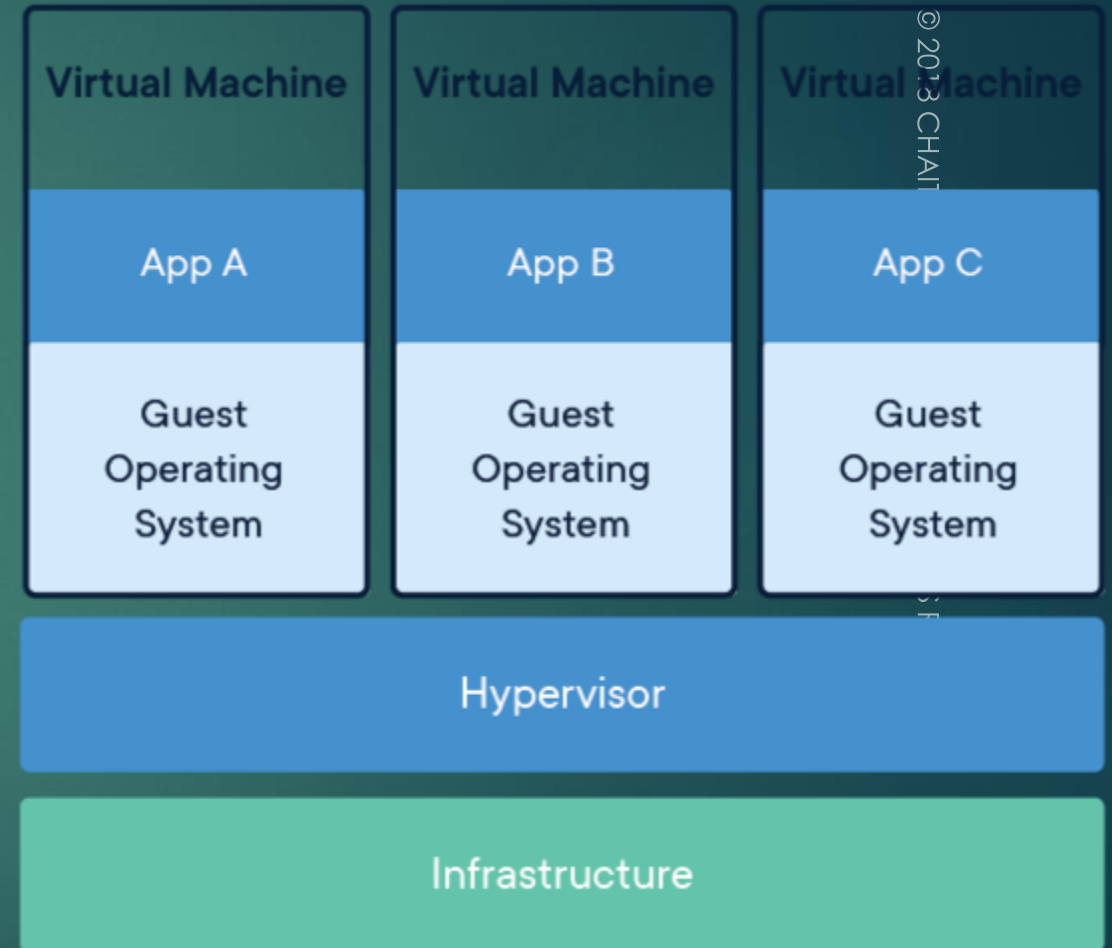
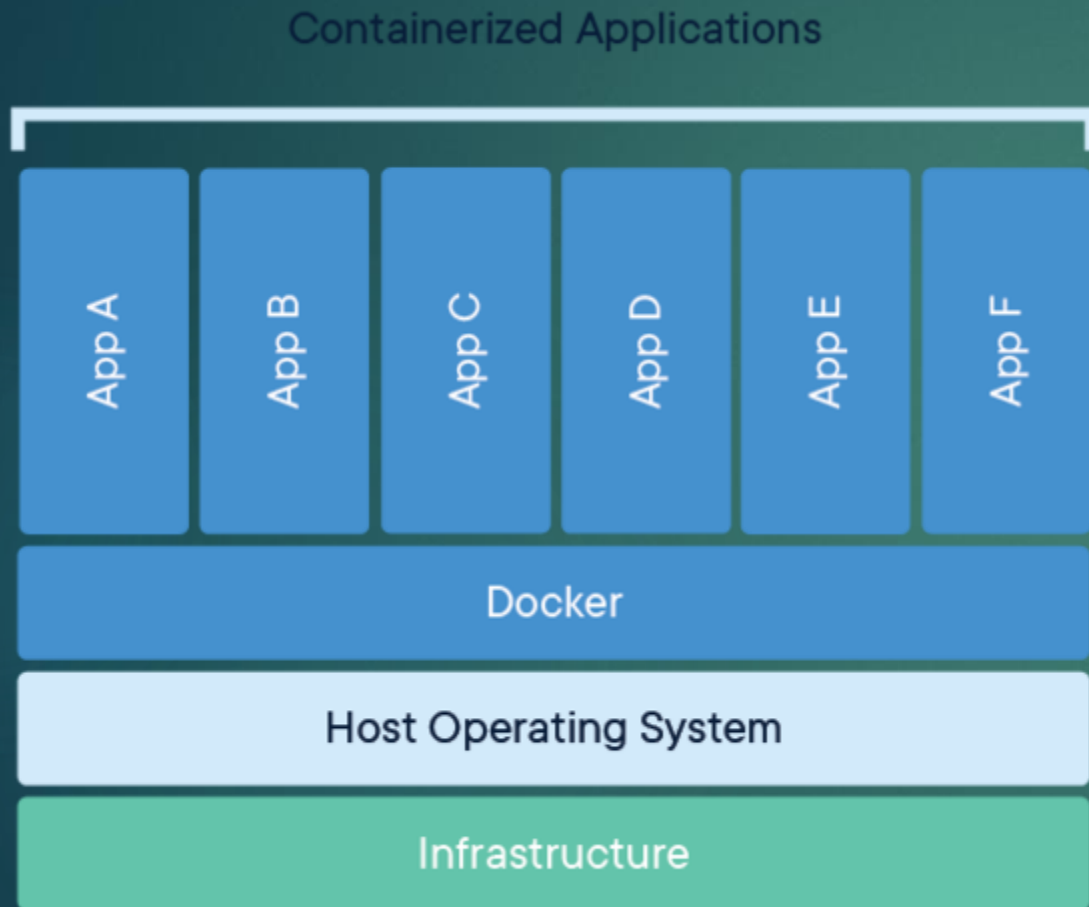
# Docker containers are lightweight

19



# Differences between Dockers and Virtual Machines

20



## Use Case

A developer will setup a JBoss software on his system



After the application is developed, it is examined by the testing team



Here, the tester repeats the installation process of JBoss

Once the application is tested, it will be deployed by the production team



To host the Java application, the system admin also has to install JBoss on his system

# Use Case

Let's take an example, where a company develops a Java Application

due to the difference in computer environments, Jboss doesn't work on the other systems, so installation is done separately on three different computers

Developer

Tester

Why not try Docker Containers?

But this process consumes a lot time and effort. can there be an alternate way to this?



