POWERING MICROSERVICES WITH DOCKER

Phyo Min Htun

ITIL-V3 , RHCE , RHCSA , LPIC , CompTIA Linux+ , SUSE CLA , SUSE CLP

Solution Engineer and SysOps/DevOps Engineer
Core Member @ Myanmar Open Source Community (MOSC)

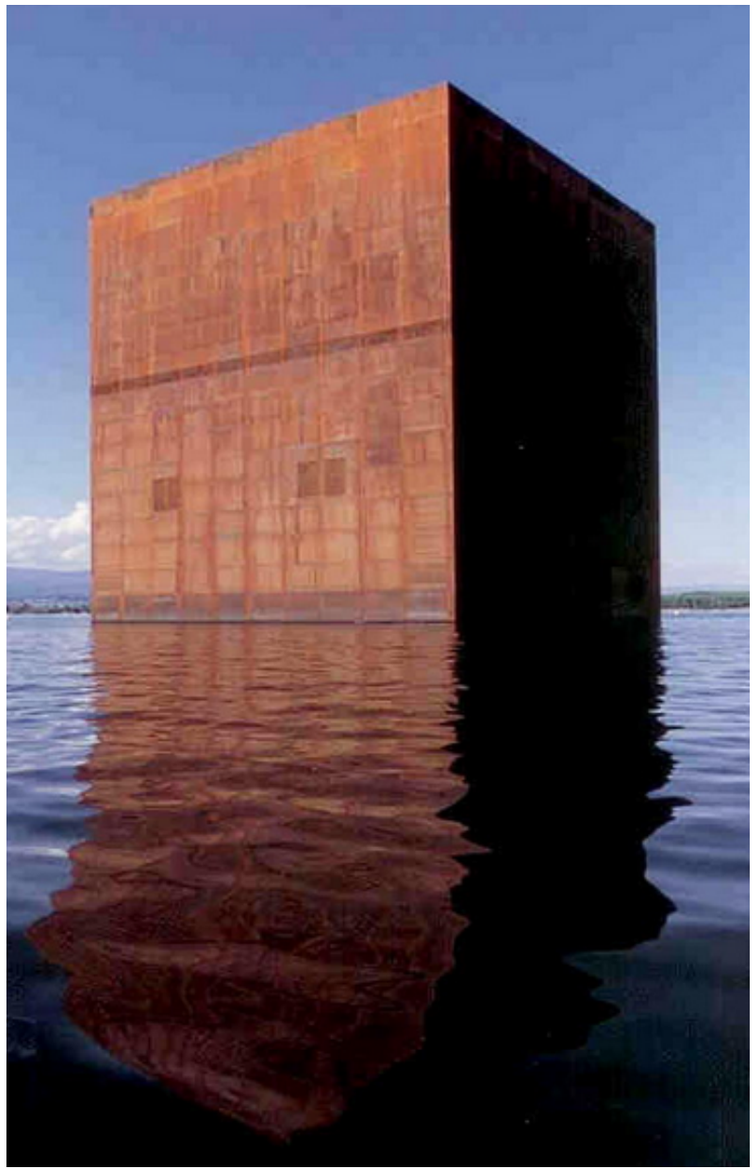
@Channel - https://phyominhtun.slack.com
@Key - pub 2048R/90373DB4 2015-10-21
uid Phyo Min Htun (Phyo Min Htun PGP Key ...) <phyominhtun@protonmail.com>
sub 2048R/DEE94BB9 2015-10-21

AGENDA

- What is Monolithic Architecture?
- What is Microservices Architecture?
- Different between Monolithic and Microservices Architecture
- Pros / Cons of Monolithic Architecture
- Pros / Cons of Microservices Architecture
- What is Container?
- Comparing Containers and Virtual Machines
- Digitally Transform Your Business with Containers
- Pets vs Cattle Analogy
- What is Docker?
- What is Docker Hub (cloud-based registry service)?
- Powering Microservices with Docker
- Docker Fits into the DevOps (Development + Operations)
- Service Providers
- Demonstration

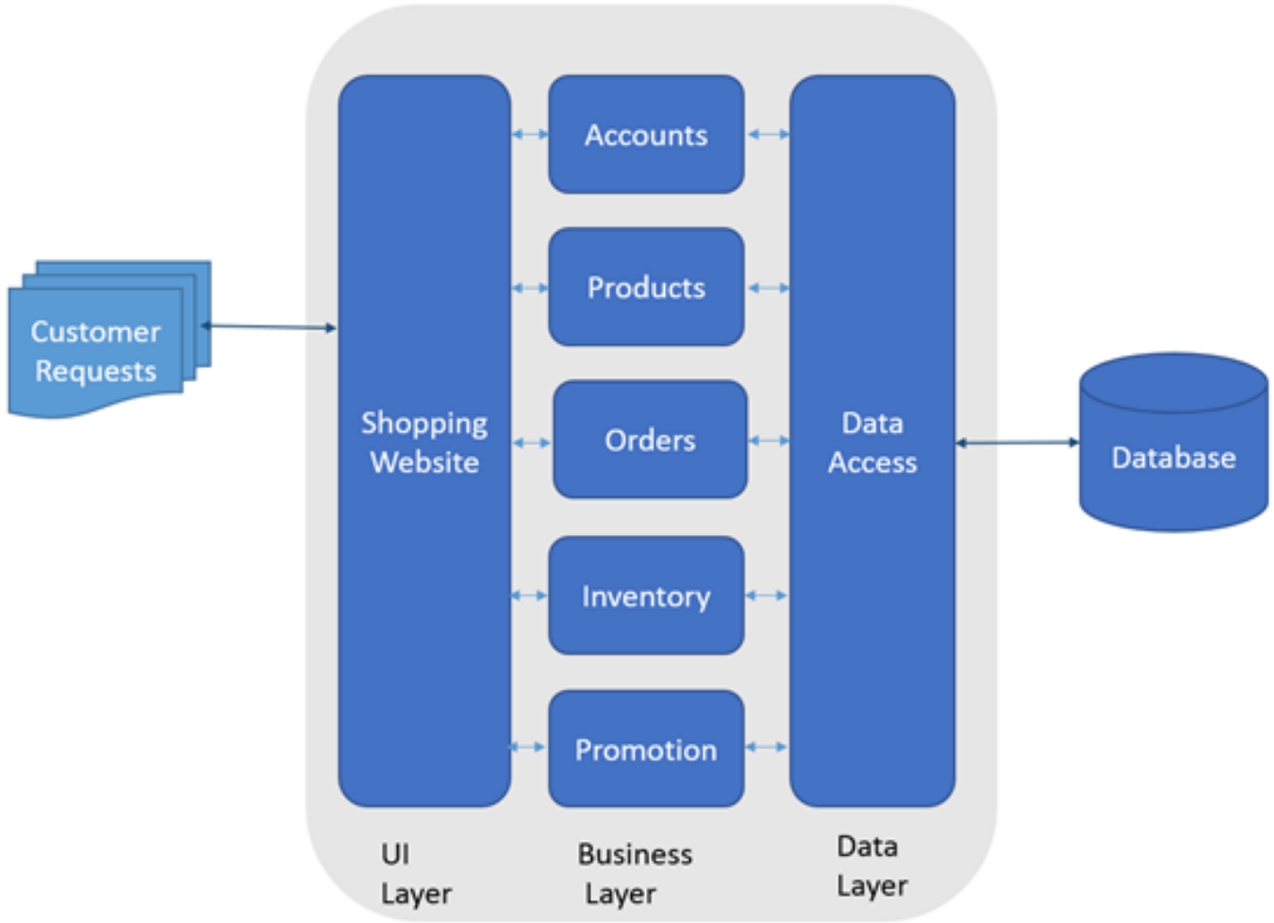
What is Monolithic Architecture?

Non-Technical Definition - Monolithic architecture is something that build from single piece of material, historically from rock.



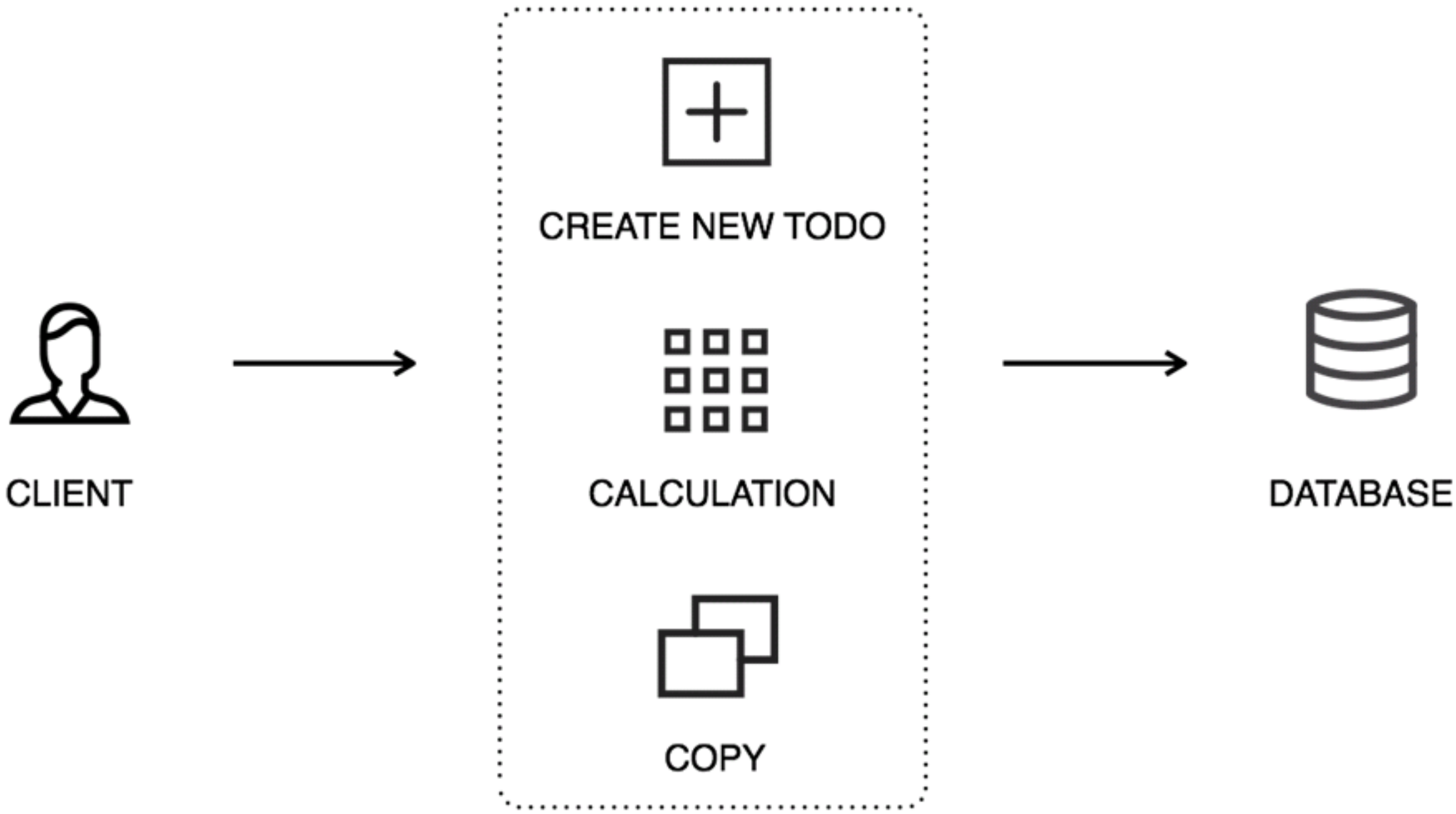
Technical Definition - Monolithic application has single code base with multiple modules.It has single build system which build entire application and/or dependency.

Traditional 3-tier or N-tire monolithic architecture model



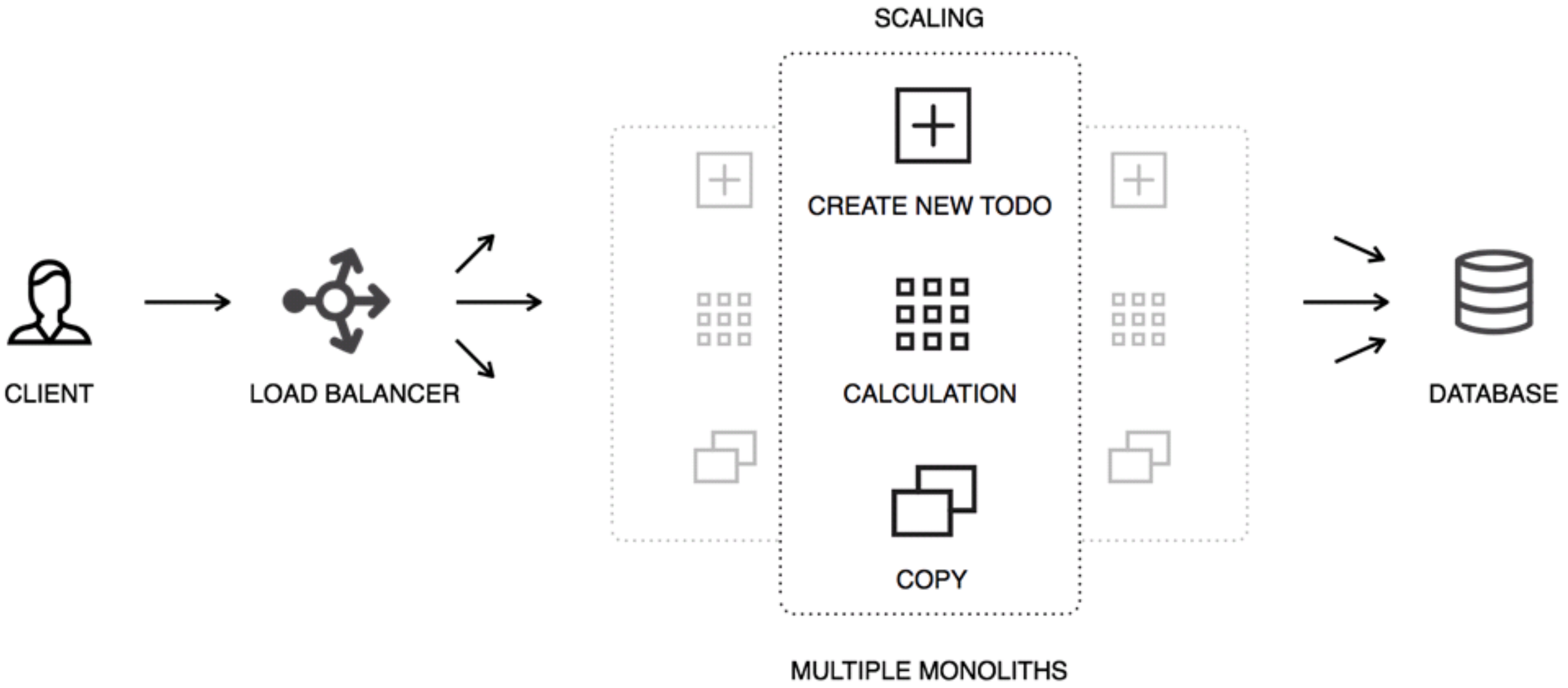
- Tightly Coupled
- Any failure could affect the whole system.
- Scale Up
- Impact the efficiency and agility

The Monolithic Architecture



All services combined into one build,
written in the same language and application framework

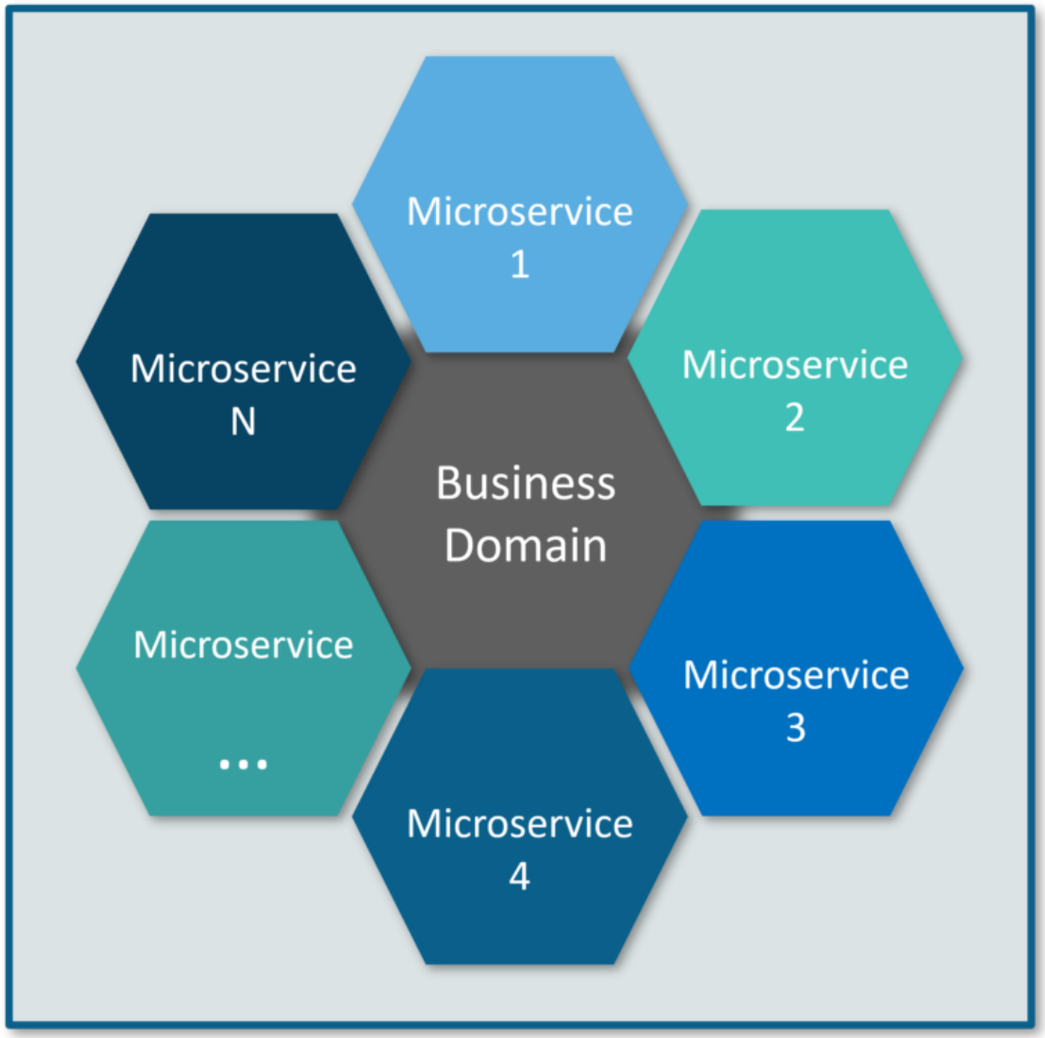
The Monolithic Architecture



What is Microservices Architecture?

What is Microservices Architecture?

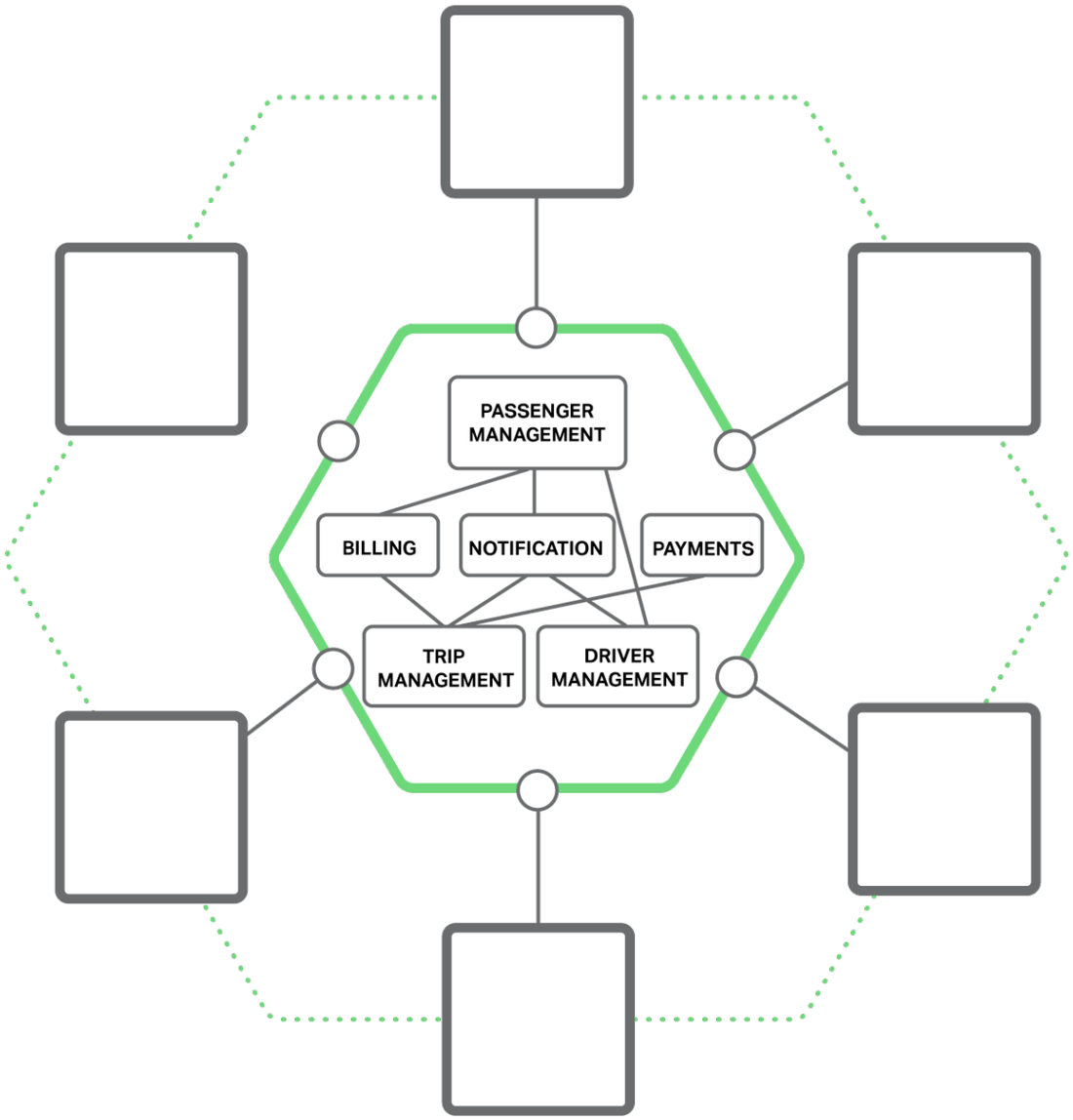
Microservices - an architectural style that structures an application as a collection of small autonomous services, modeled around a business domain.

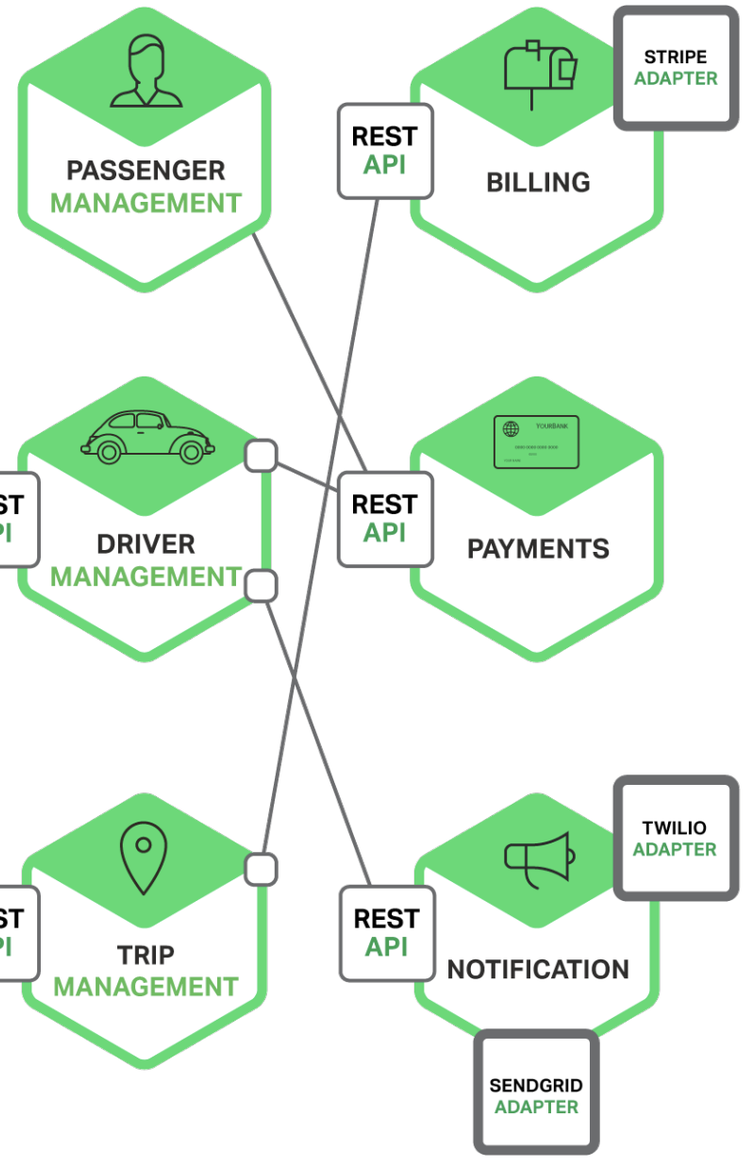


An application is divided into services.
Each service runs a unique process and usually manages its own databases. A service can generate alerts, log data, support user interfaces, handle user identification or authentication, and perform various other tasks.

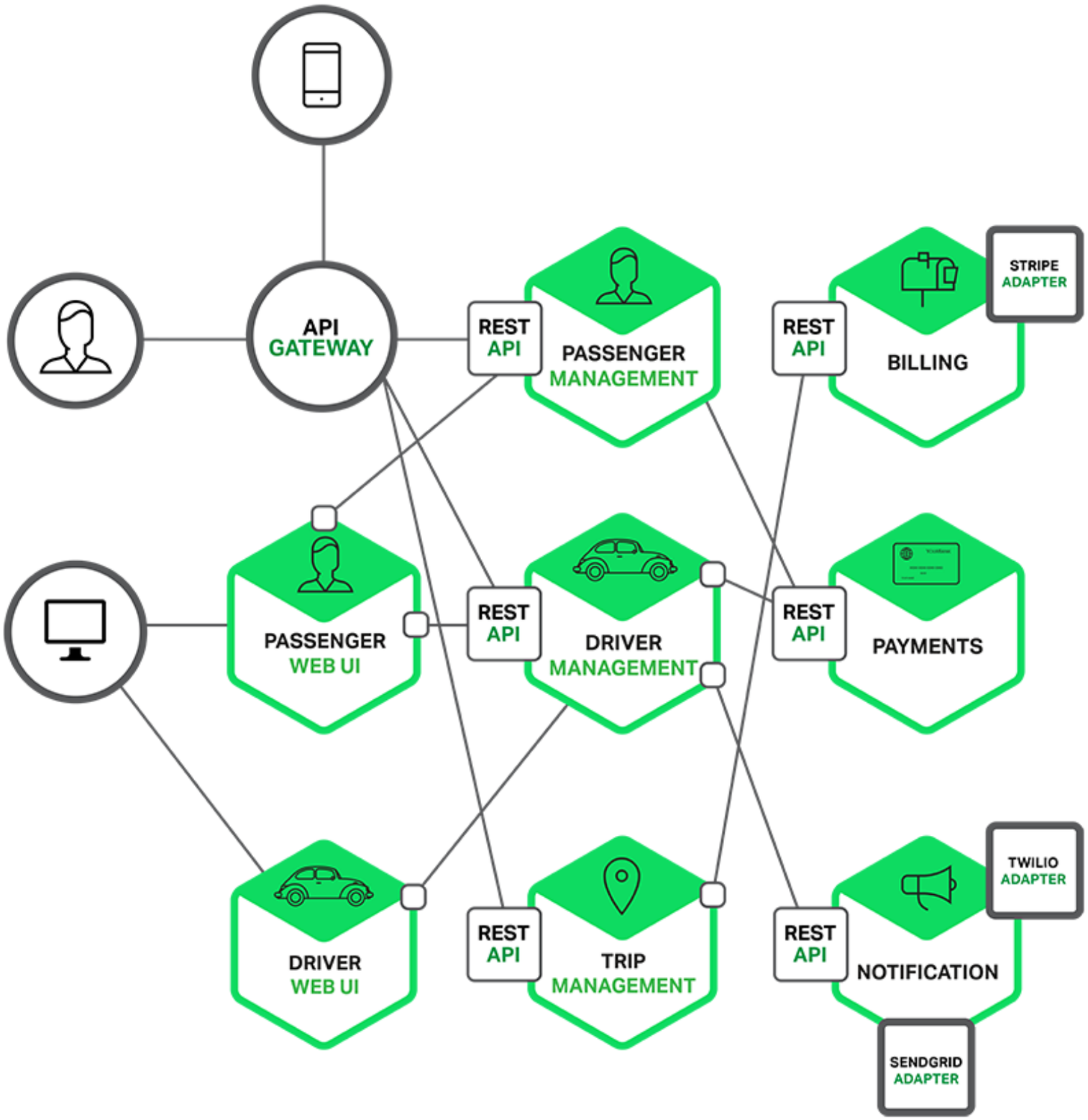
Microservices is basically a self contained process that provides a unique business capability.

Monolith to Microservice





Microservices Architecture



Why Use Microservices?

Speed

Maintain

Change

Scale

Empower

Microservices Features

Decoupling - Services within a system are largely decoupled. So the application as a whole can be easily built, altered, and scaled.

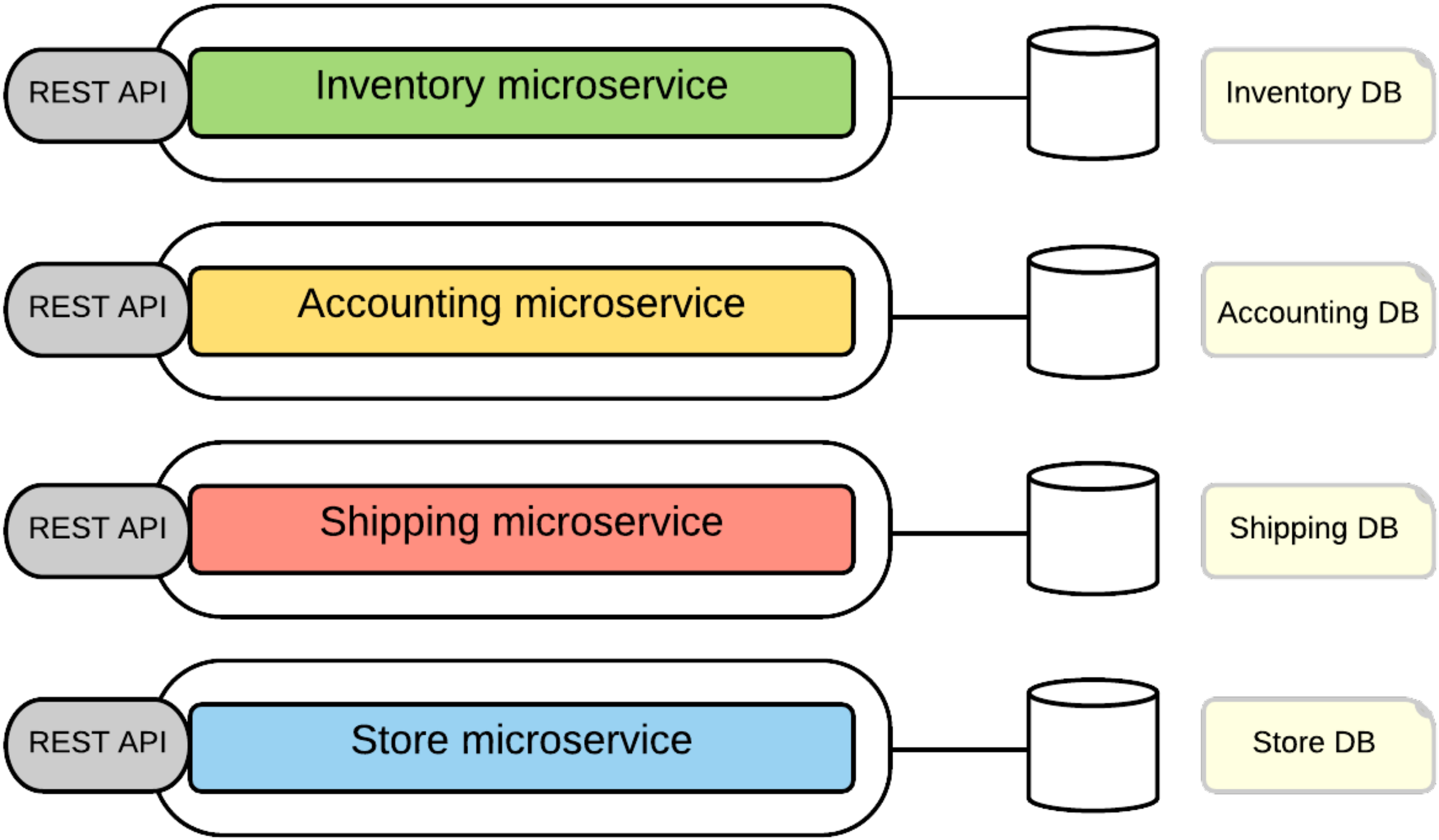
Componentization - Microservices are treated as independent components that can be easily replaced and upgraded

Business Capabilities - Microservices are very simple and focus on a single capability

Decentralized Governance - Developers have the freedom to choose the best useful tools to solve their problems

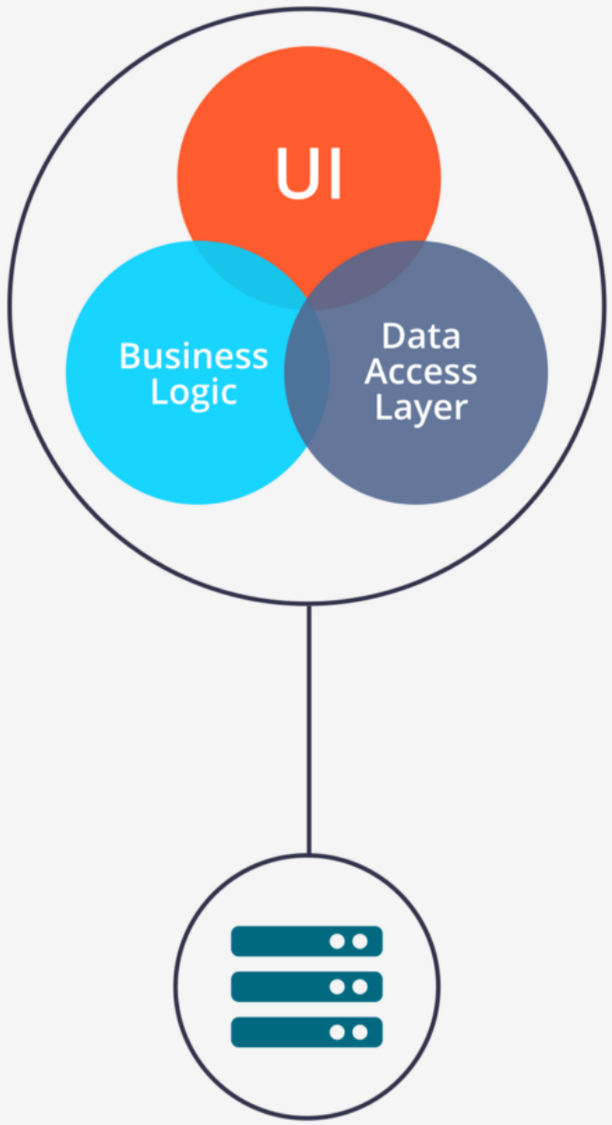
Autonomy - Developers and teams can work independently of each other, thus increasing speed

Agility - Microservices support agile development. Any new feature can be quickly developed and discarded again

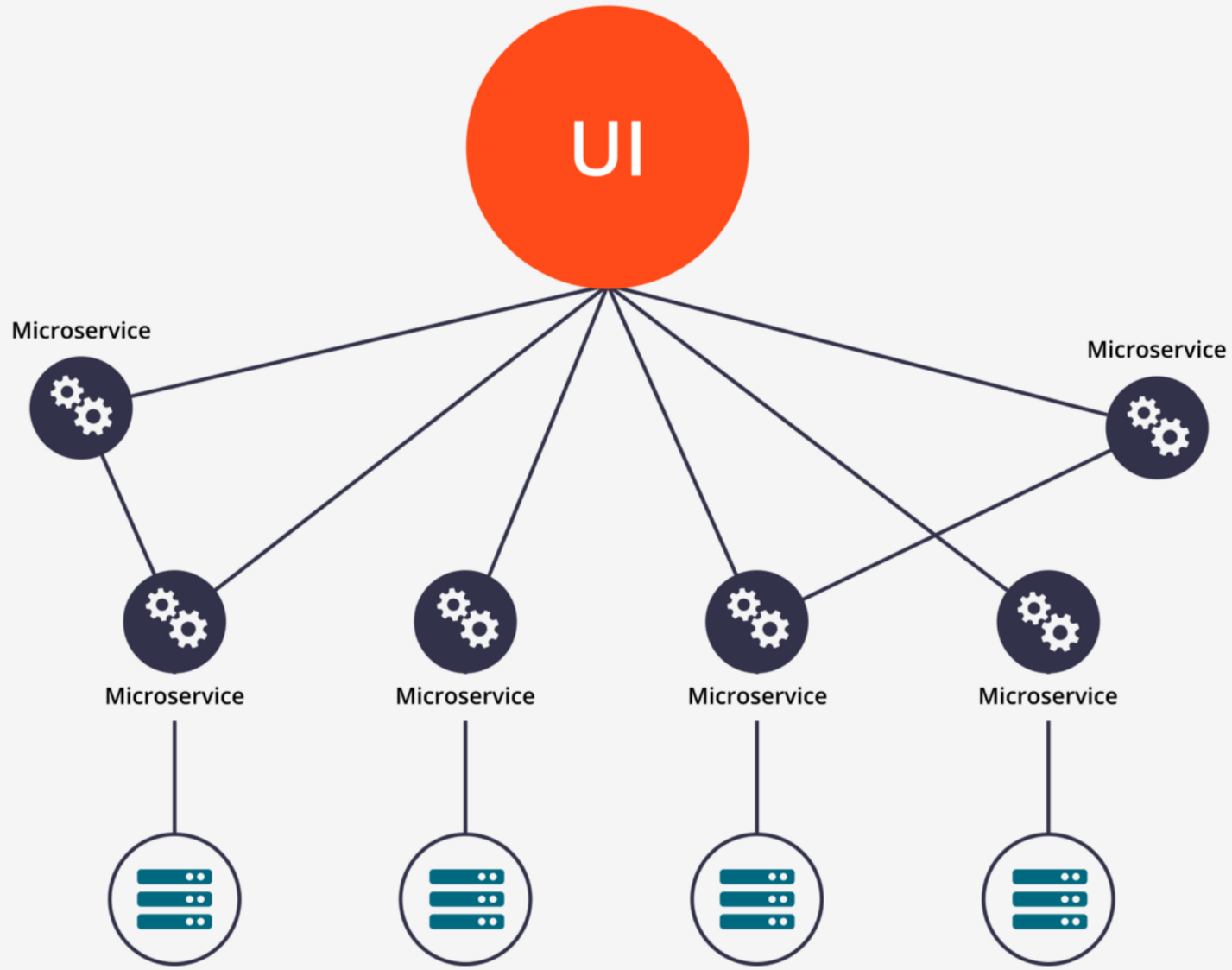


Best Practice is each microservice has its own database.

Different between Monolithic and Microservice Architecture



Monolithic Architecture



Microservice Architecture

Different between Monolithic and Microservice Architecture

Code

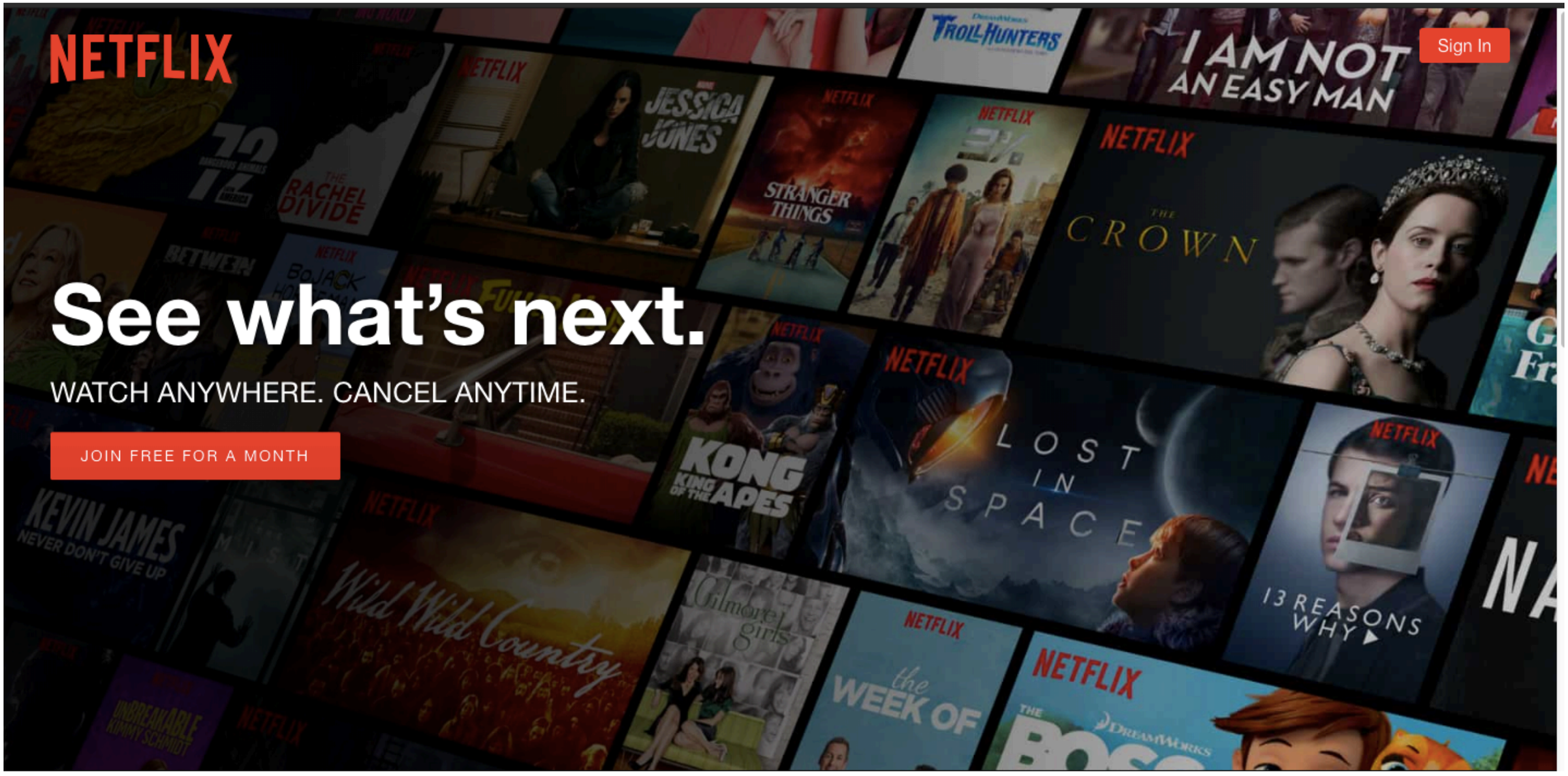
Clarity

Deployment

Language

Scaling

We Can’t Talk About Microservices Without Mentioning Netflix



https://www.netflix.com/mm/

Back in 2008, when Netflix was still operating as a monolith, a single missing semicolon brought down the entire Netflix website for several hours.

Monolithic Application
Pros / Cons

A large number of cross-cutting concerns	Easy to hook up components	Smoothly in operation	Inter-process communication
Independent scaling or code maintainability.	Dependencies	To Scale, Must scale entire monolithic.	Written in same language.

Microservice Application
Pros / Cons

Better architecture for large application	Better agility in the long term	Easy to learn	Scale service independently and written different language
More moving parts	**Complex operation**	**Configuration management**	**Monitoring**

Microservices Challenges

Rapid Provisioning : Ability and automation to spin up a server in hours.

Basic Monitoring : Many loosely coupled services in a production environment, you need visibility.

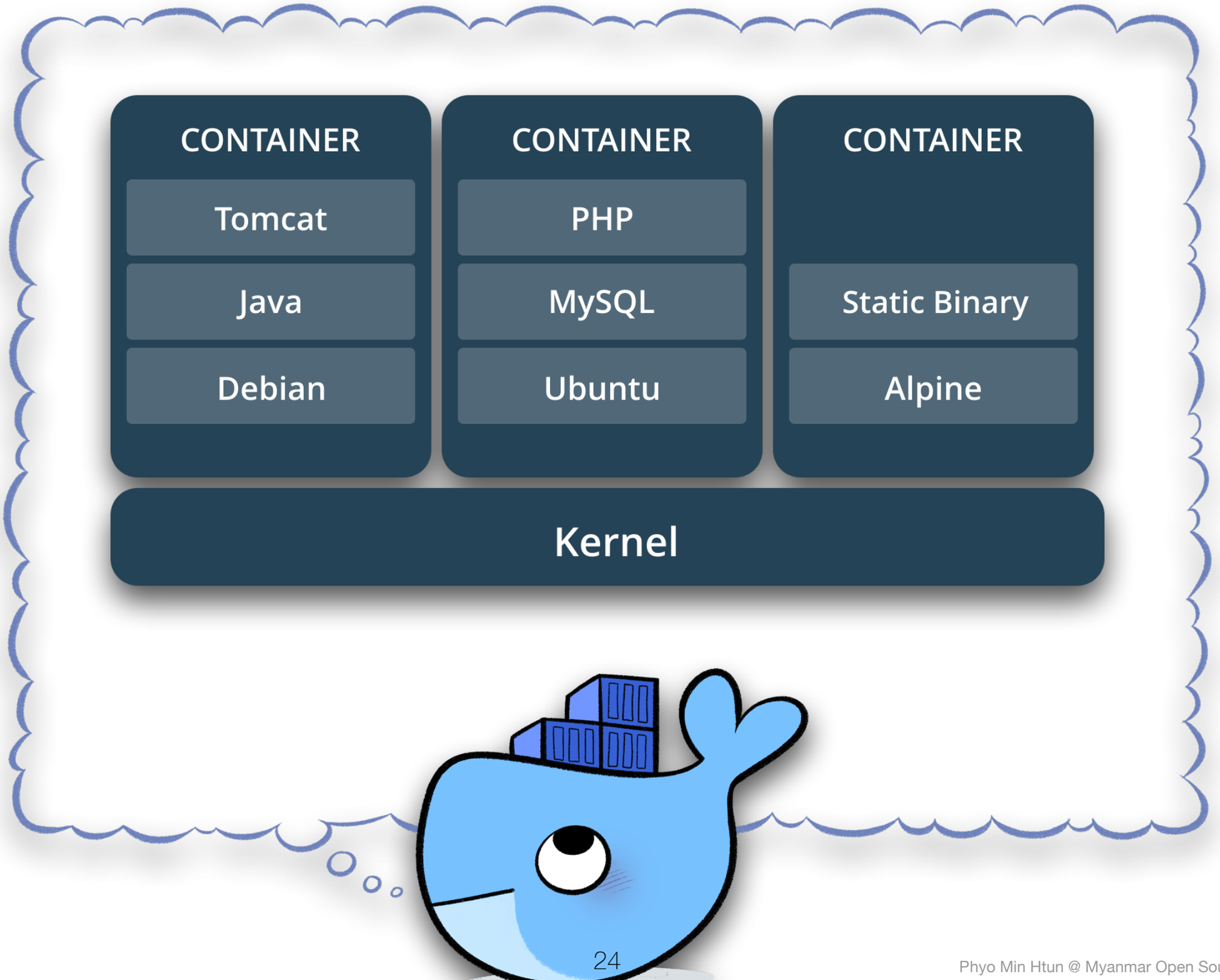
Rapid Application Deployment : Re-deploy quickly and automatically.

What is Container?

What is Container?

- A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, settings.
- Available for both Linux and Windows based apps, containerized software will always run the same, regardless of the environment.
- Containers isolate software from its surroundings, for example differences between development and staging environments and help reduce conflicts between teams running different software on the same infrastructure.

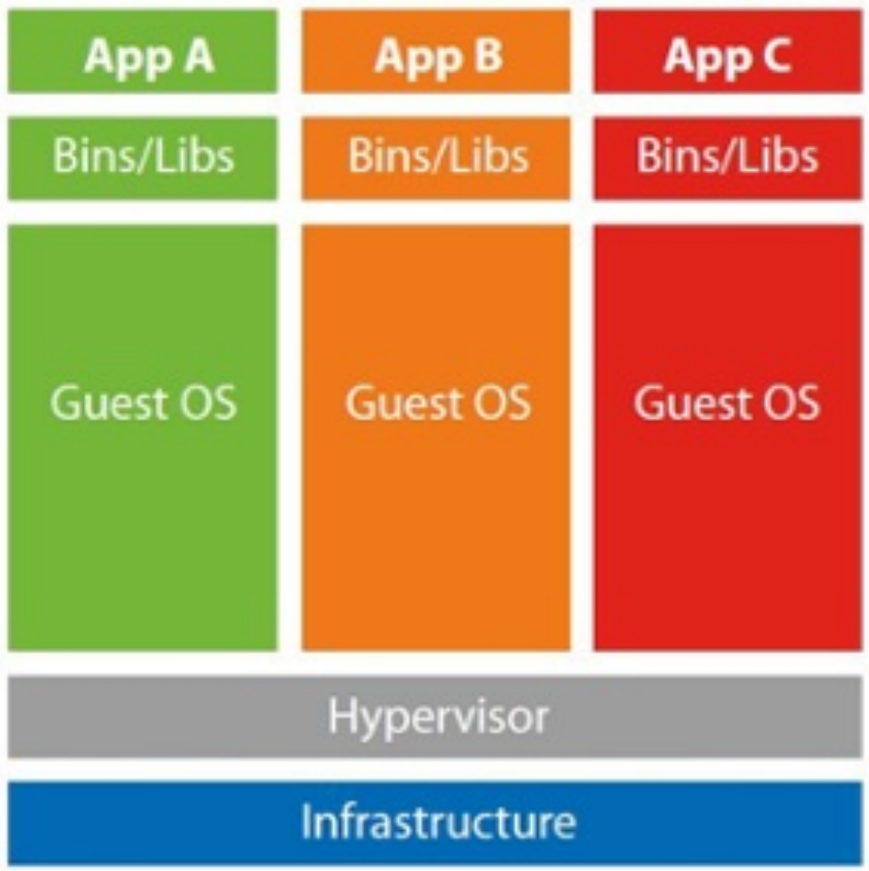
What is Container?

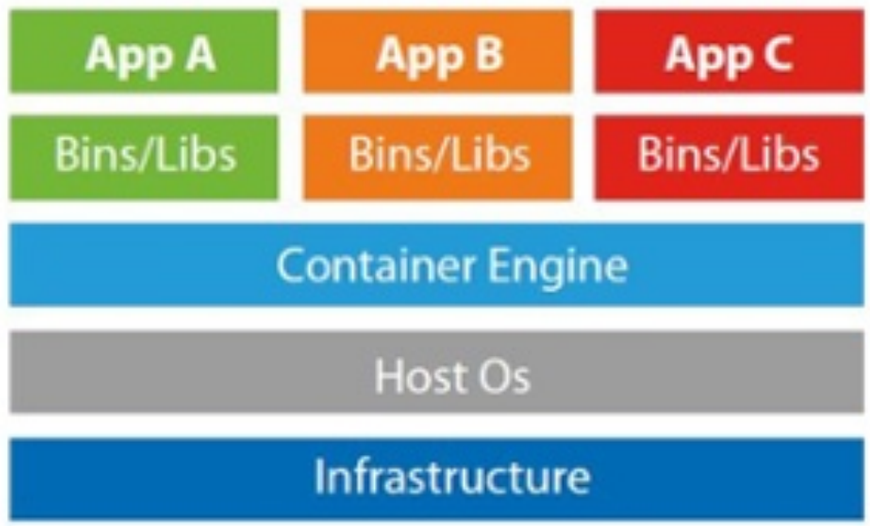


Comparing Containers and Virtual Machines

- Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware.

- Containers are more portable and efficient.



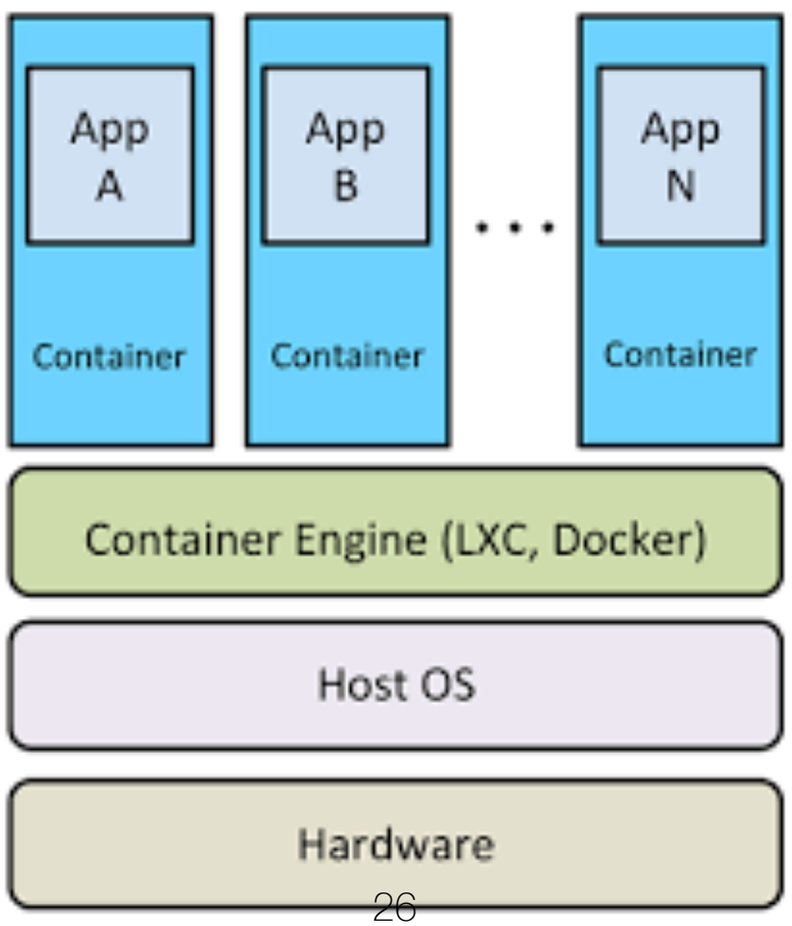


Containers

- Containers are an abstraction at the app layer that packages code and dependencies together.

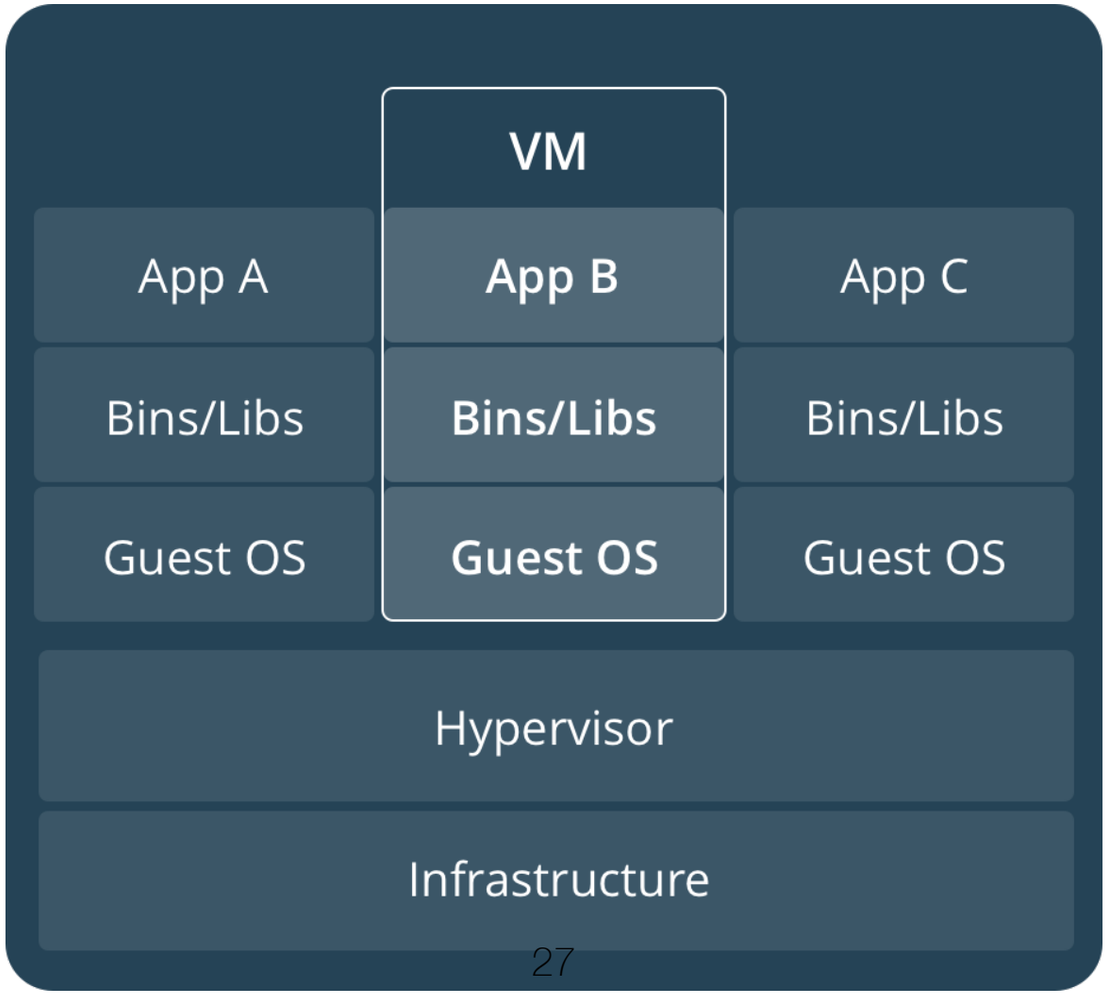
- Multiple containers can run on the same machine and share the OS kernel with other containers, each running as isolated processes in user space.

- Containers take up less space than VMs (container images are typically tens of MBs in size), and start almost instantly.



Virtual Machines

- Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.
- The hypervisor allows multiple VMs to run on a single machine.
- Each VM includes a full copy of an operating system, one or more apps, necessary binaries and libraries - taking up tens of GBs. VMs can also be slow to boot.



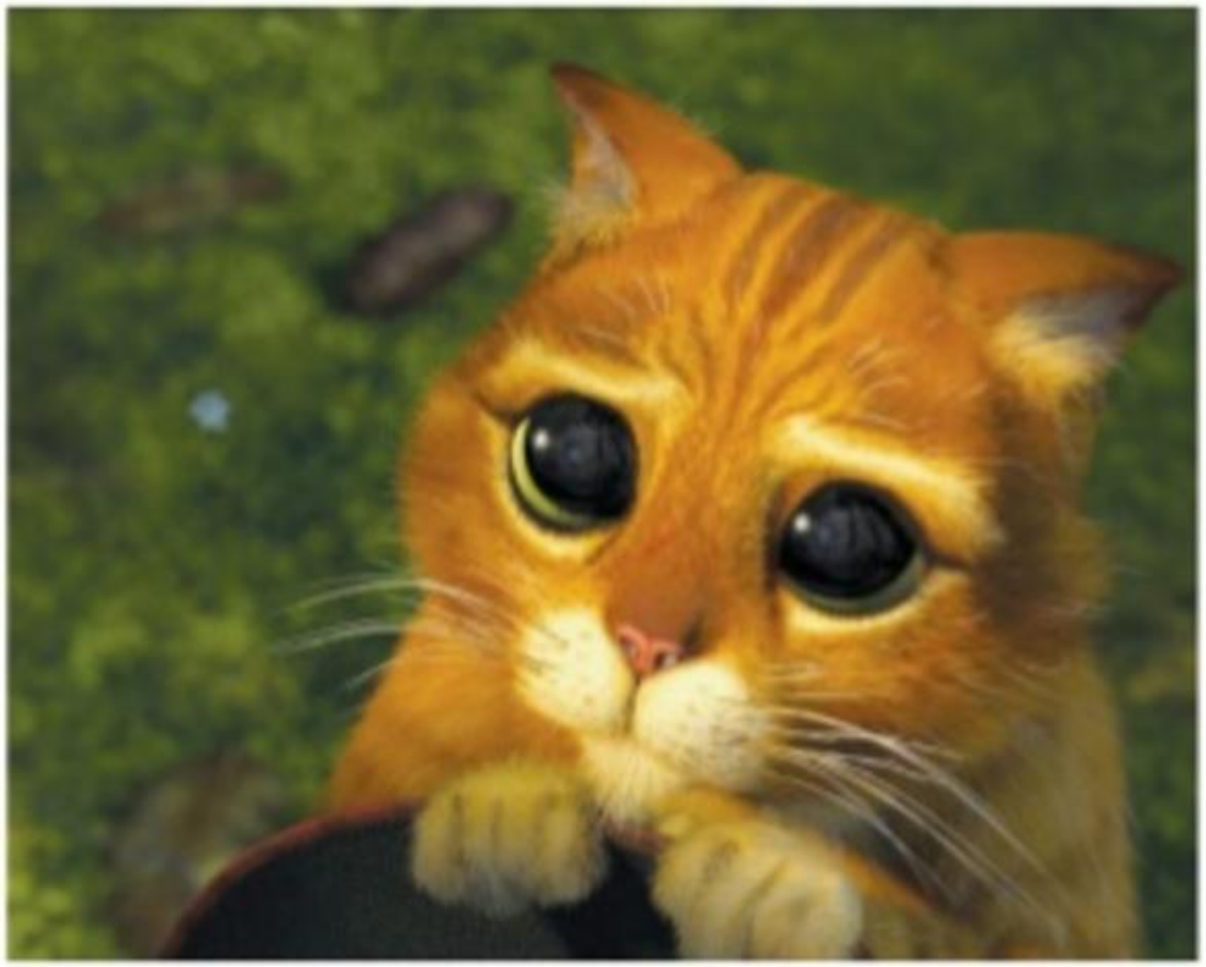
Digitally Transform Your Business

Reduce IT Costs
Optimize infrastructure costs for your existing applications and datacenters while enabling cloud migration and streamlining operations for total cost savings of 50% or more.

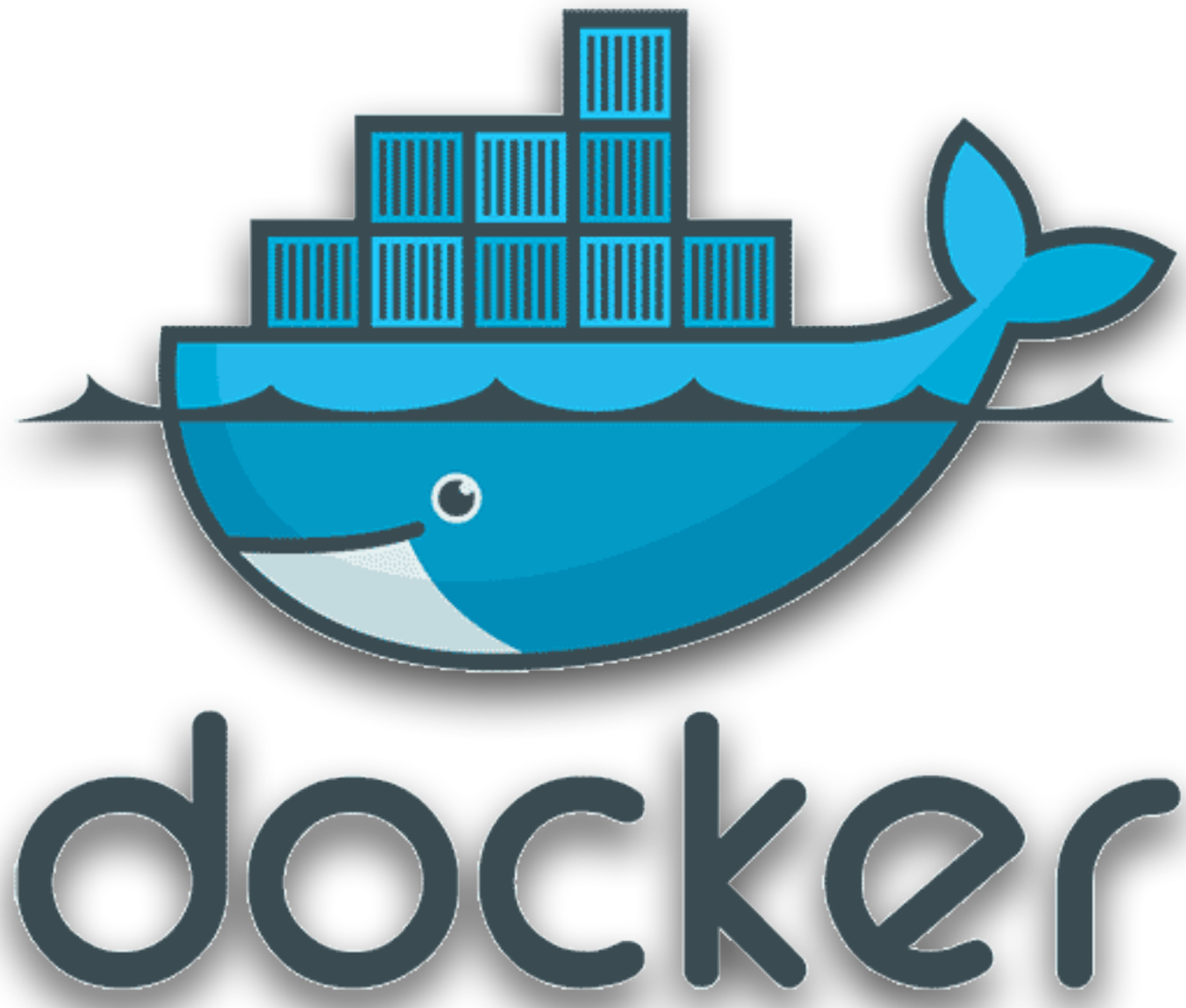
Accelerate Innovation
Unlock the creativity of software developers to deliver new apps and services to customers faster. Gain the freedom to build the right app with the right components and accelerate digital strategies.

Agile IT, Agile Budget
Docker makes business faster by enabling you to respond quickly to changing market and customer needs. Speed to delivery, scale and response allows companies to focus driving value for their customers.

Pets vs Cattle Analogy



https://rhelblog.redhat.com/2016/02/08/container-tidbits-does-the-pets-vs-cattle-analogy-still-apply/


docker

What is Docker?

- Docker is a light-weight, reliable and fast containerization technology.
- Docker is an open container platform for developers and sysadmins to build, ship, and run distributed applications.
- Docker allows you to build, test, and deploy applications quickly.
- Docker helps developers build and ship higher-quality applications, faster.
- Docker helps sysadmins deploy and run any app on any infrastructure, quickly and reliably.

Why to use Docker?

- **Agility**
  Container technology provides the solution for the applications to be wrapped up quickly from its components such binaries, libraries, configuration files, as a single package and be deployed on different platforms/environments without any consideration of compatibility issues.

- **Less Overhead**
  Containers utilize the advantage of sharing the Guest OS and its resources, which makes them light-weight, fast and superior to Virtual machines.

- **Version Control**
  Like any version control tool, containers provides the capability to access different versions of containers and not worry about the environment specific customizations.

When to use Docker?

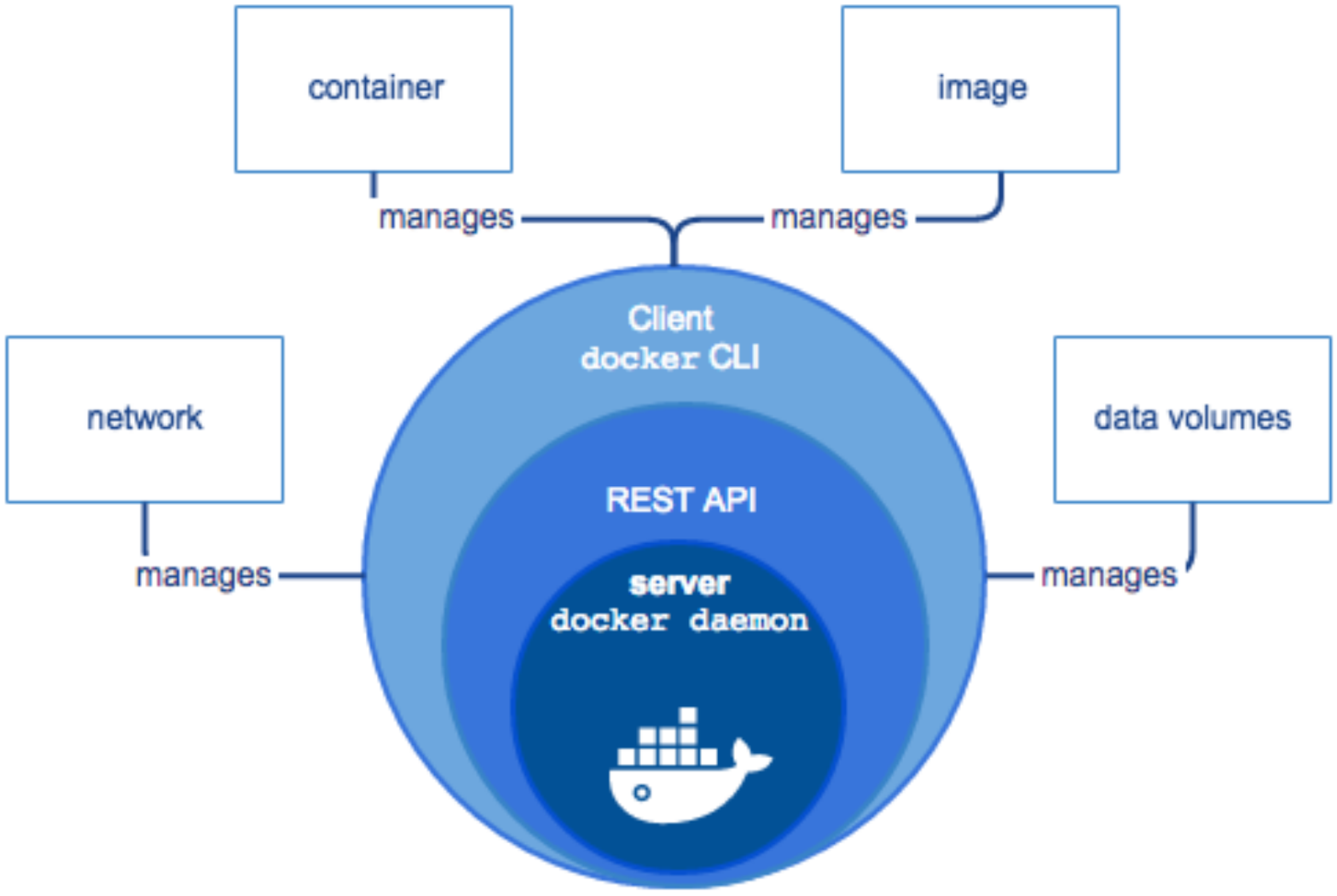
- **Microservices**
  Build and scale distributed application architectures by taking advantage of standardized code deployments using Docker containers.

- **Continuous Integration and Delivery**
  Accelerate application delivery by standardizing environments and removing conflicts between language stacks and versions.

- **Data Processing**
  Provide big data processing as a service. Package data and analytics packages into portable containers that can be executed by non-technical users.

- **Containers as a Service**
  Build and ship distributed applications with content and infrastructure that is IT-managed and secured.

Docker Engine



Docker Engine Components

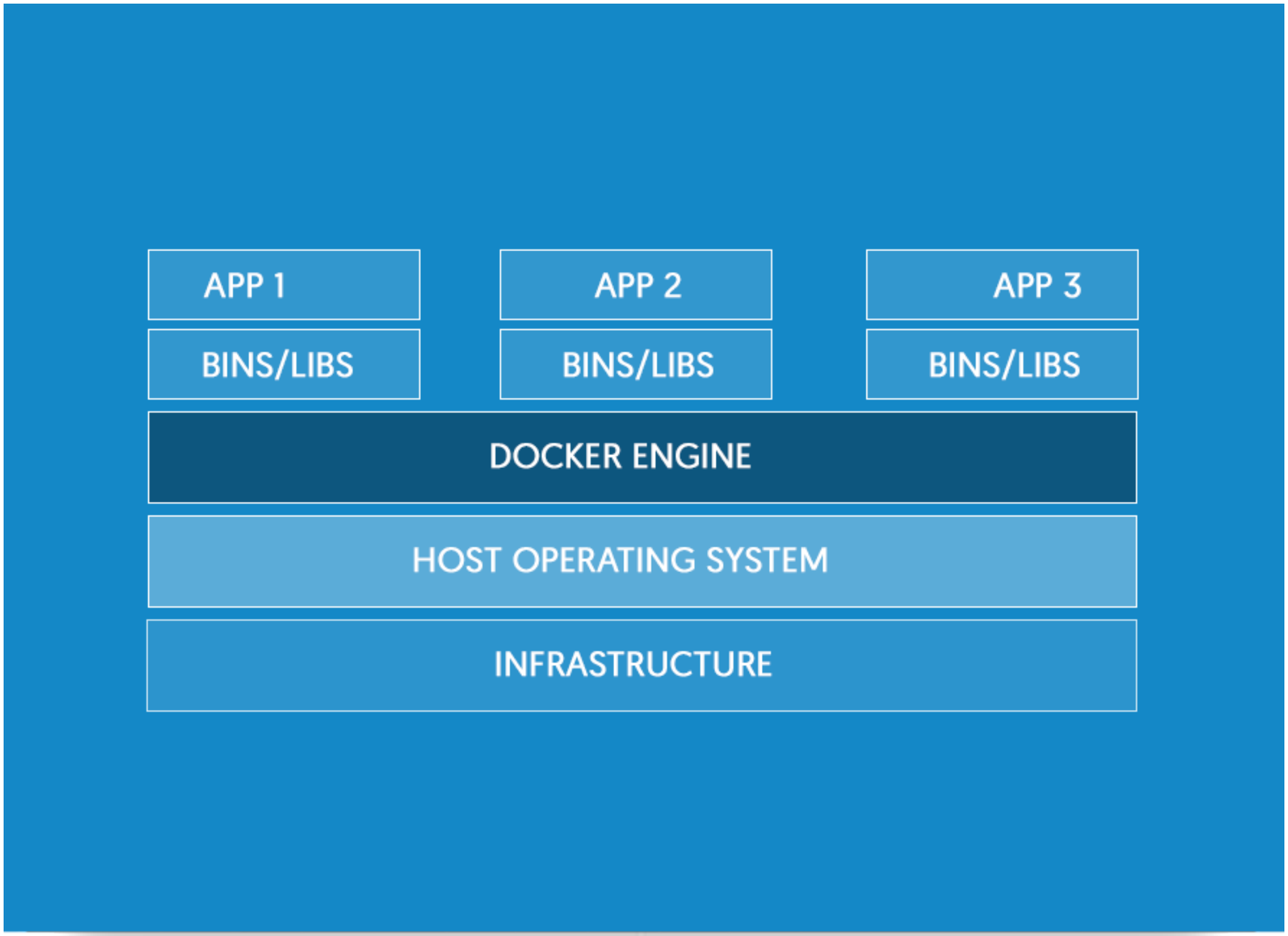
Docker Engine is a client-server application with these major components:

- **A server which is a type of long-running program called a daemon process (the dockerd command)**

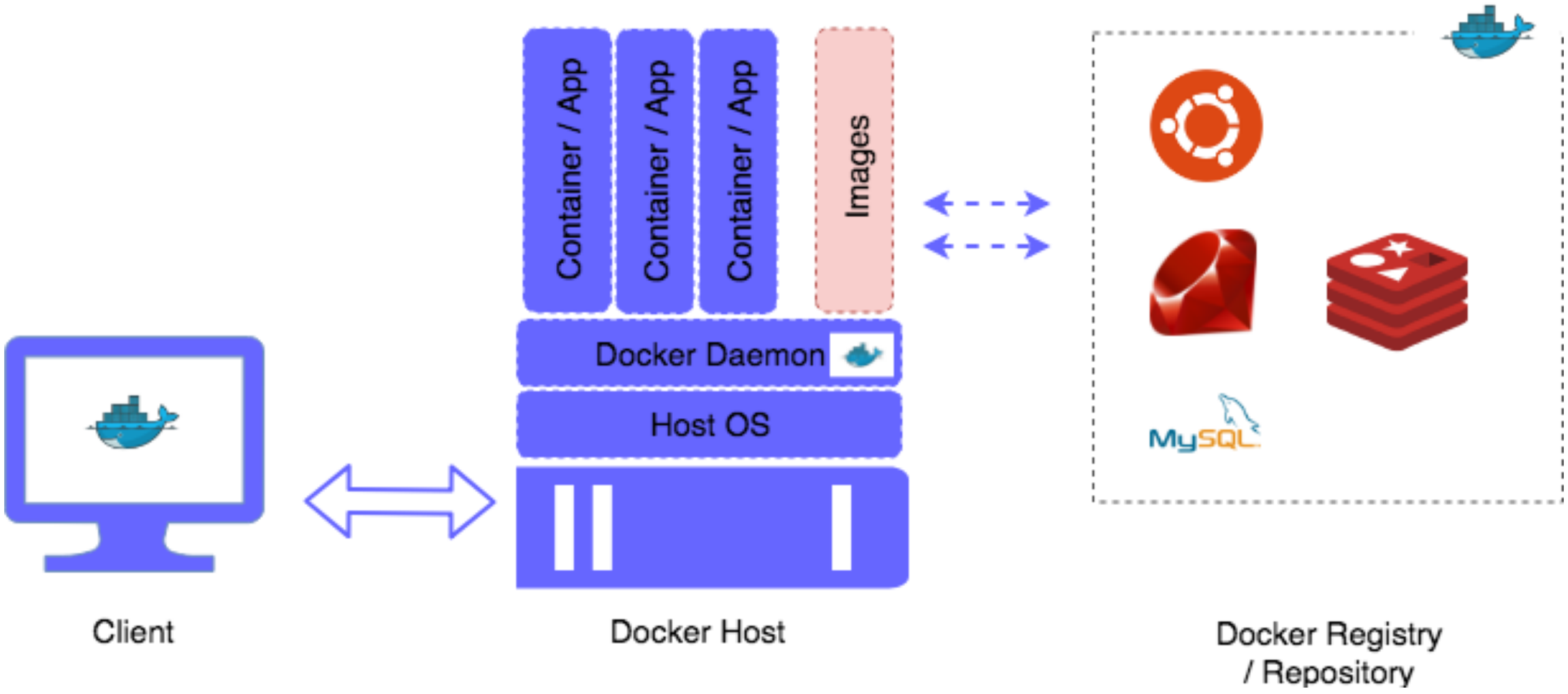
- **A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.**

- **A command line interface (CLI) client (the docker command).**

Architecture



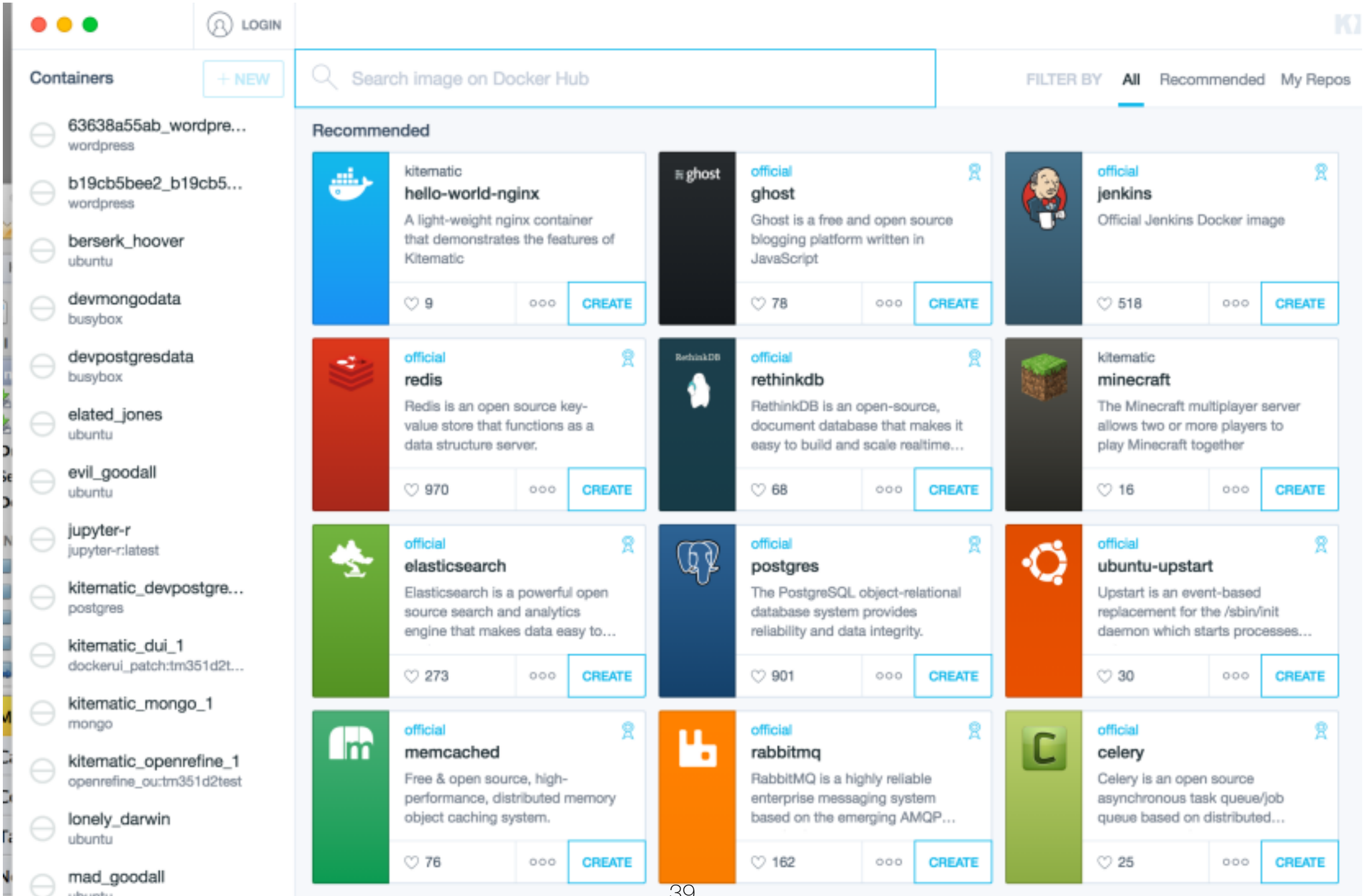
Docker Overview Architecture with Registry / Repository



DOCKER HUB - REGISTRY

- Docker Hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker Cloud so you can deploy images to your hosts.
- It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

DOCKER HUB - REGISTRY



MY PUBLIC DOCKER REGISTRY

https://hub.docker.com/u/phyominhtun/

phyominhtun/whalesay
public

0 STARS | 72 PULLS | DETAILS

phyominhtun/myweb
public

0 STARS | 16 PULLS | DETAILS

phyominhtun/elasticops
public

0 STARS | 3 PULLS | DETAILS

#docker pull phyominhtun/imagename

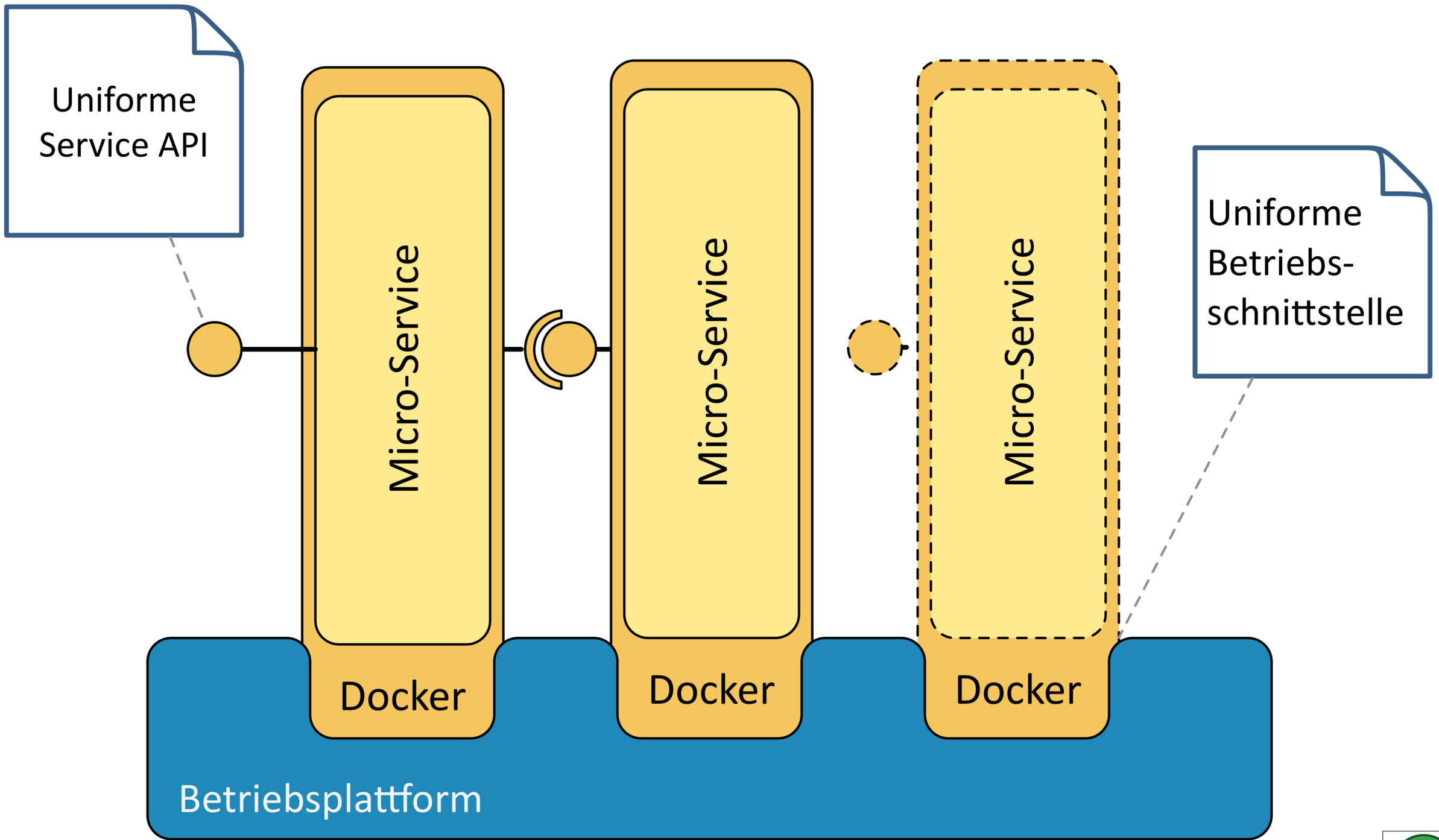
POWERING MICROSERVICES WITH DOCKER

- Docker containers are lightweight by design and ideal for enabling microservices application development.

- Accelerate development, deployment and rollback of tens or hundreds of containers composed as a single application.

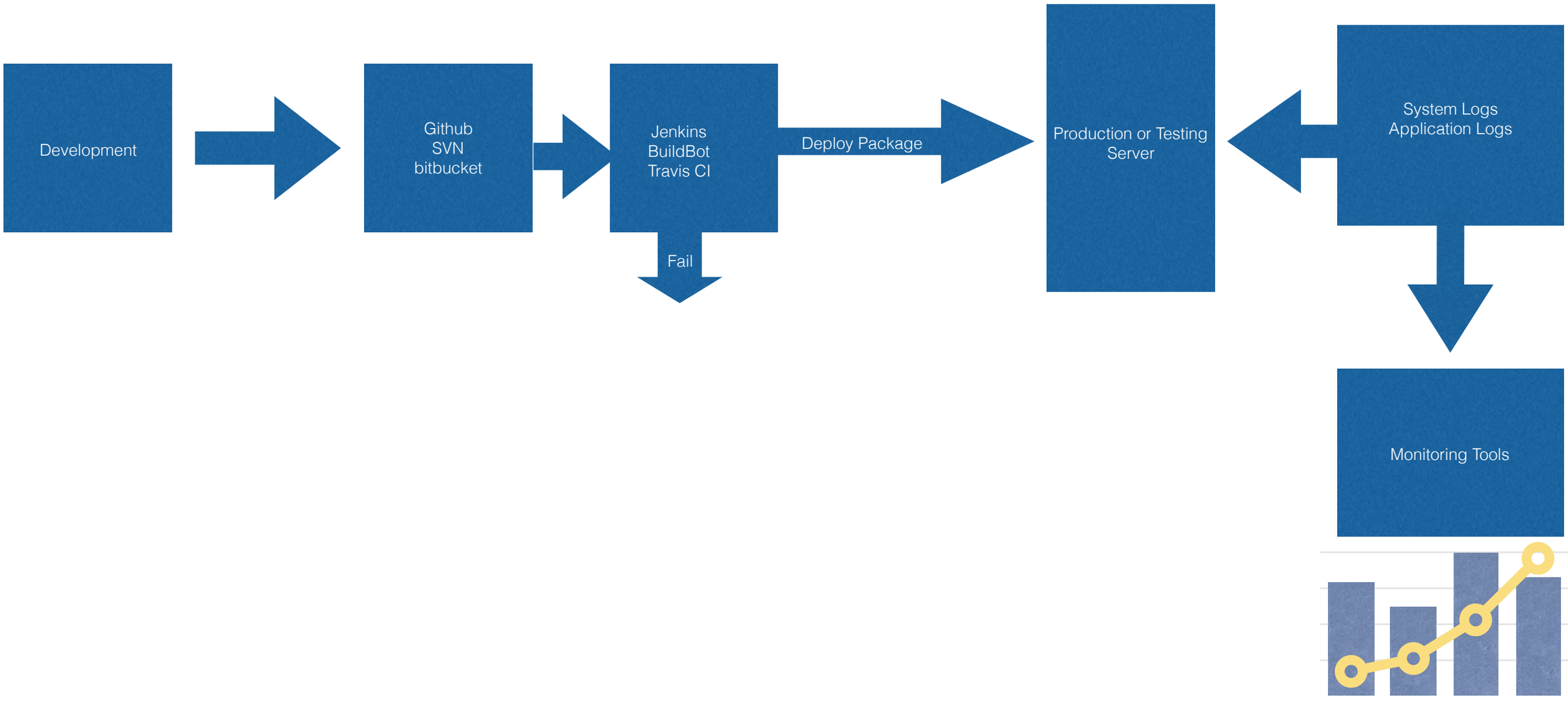
- Whether building new microservices or transitioning monoliths to smaller services, simple to use tools make it easy to compose, deploy and maintain complex applications.

MICROSERVICES WITH DOCKER

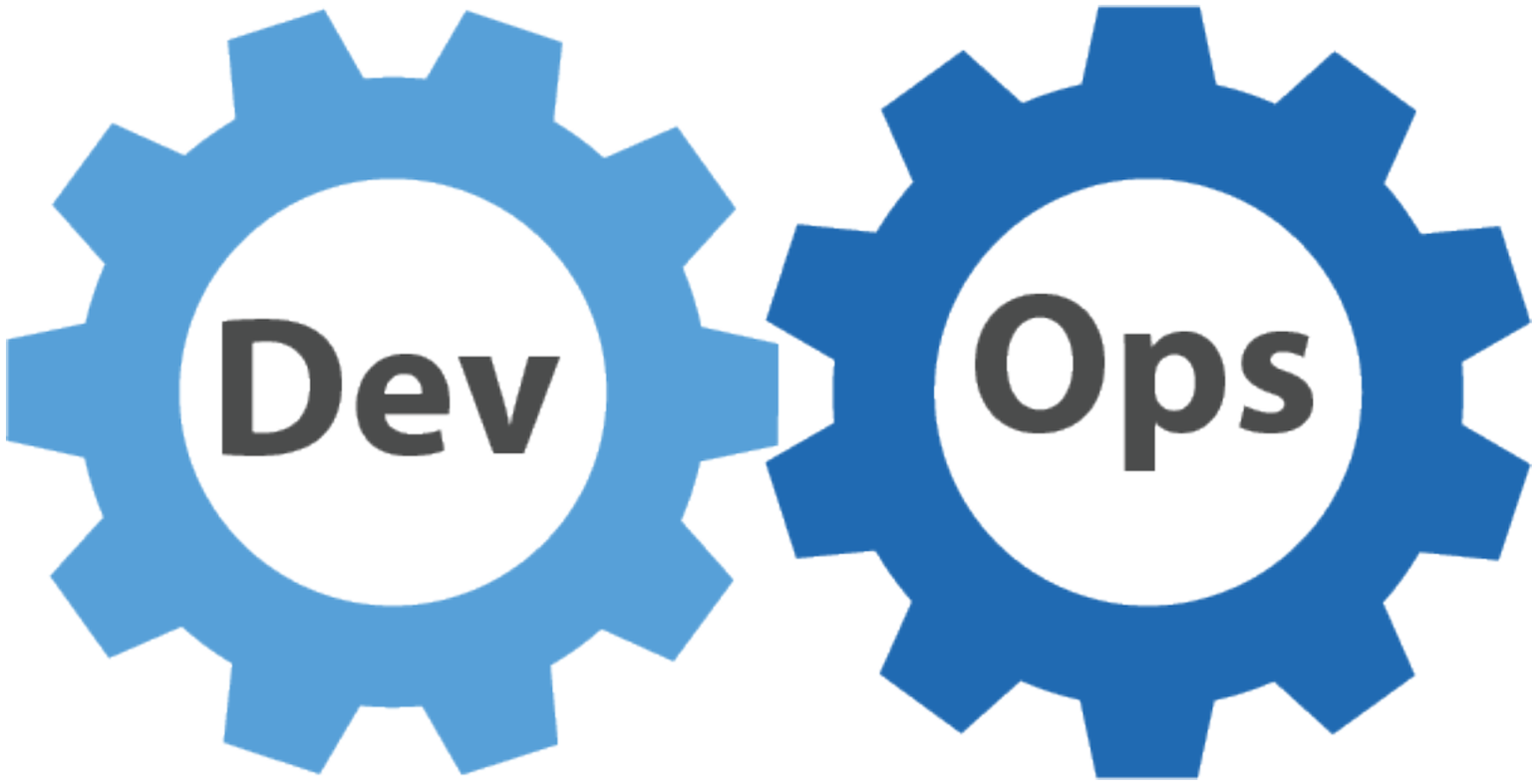


Docker Fits into the DevOps (Development + Operations)

DevOps Pipeline (CI/CD)



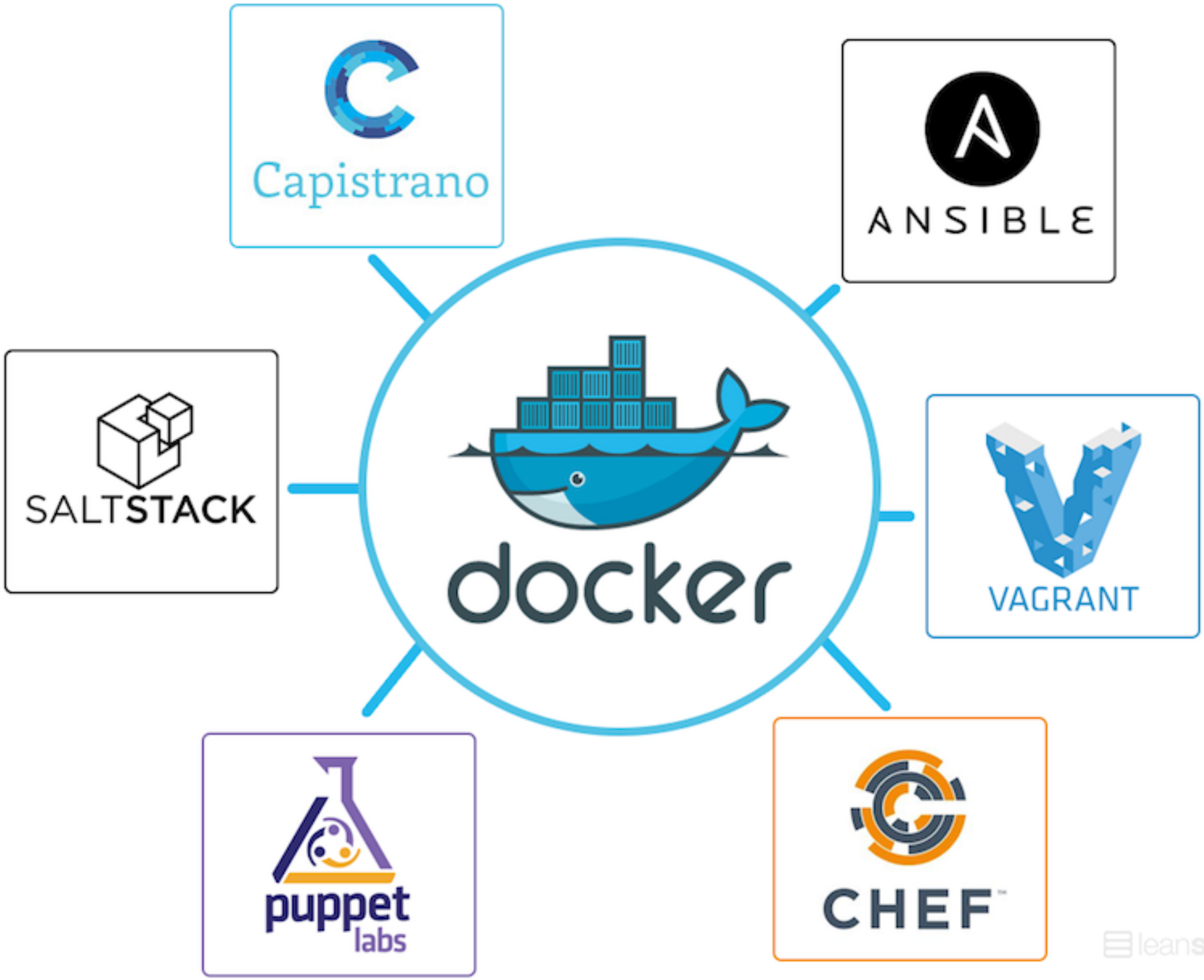
Docker Fits into the DevOps (Development + Operations)



The **developers** don't have to worry about setting up the specific development environment every time.They need to focus on building good quality code.

IT operations team/Sysadmins, Docker makes it possible to set up environments that are exactly like a production server, and allow anyone to work on the same project with the exact same settings.

Docker Fits into the DevOps (Development + Operations)



Service Providers

rackspace the open cloud company · DEIS · Google Cloud Platform · amazon web services · IBM · SOFTLAYER an IBM Company · openstack · Yandex · OPENSHIFT · Microsoft Azure · DigitalOcean · CLOUD FOUNDRY · linode · Quay.io · Baidu 百度 · Baremetal Industries · clever cloud · Google Compute Engine · stackato by ActiveState · Google App Engine · domino · StackDock · Gilliam · tutum · ORCHARD · Flynn · VOXOZ

Dev Tools

codefresh · Fig · VAGRANT · heroku · Shippable · runnable beta · NODECHECKER · wercker · drone.io

Official Repositories

mongoDB · MySQL · PostgreSQL · Ruby · WordPress · Java · debian · CentOS · ubuntu · RAILS · redis · NGINX · node

docker

Operating Systems

redhat · boot2docker · CoreOS · ubuntu · debian · fedora · CentOS

Configuration Management

puppet labs · Chef · Jenkins · Capistrano · DotCi · Packer · ANSIBLE · SALTSTACK

Big Data

OpenCore · syncsort · hadoop

Service Discovery

Eureka · etcd · SkyDNS

Orchestration

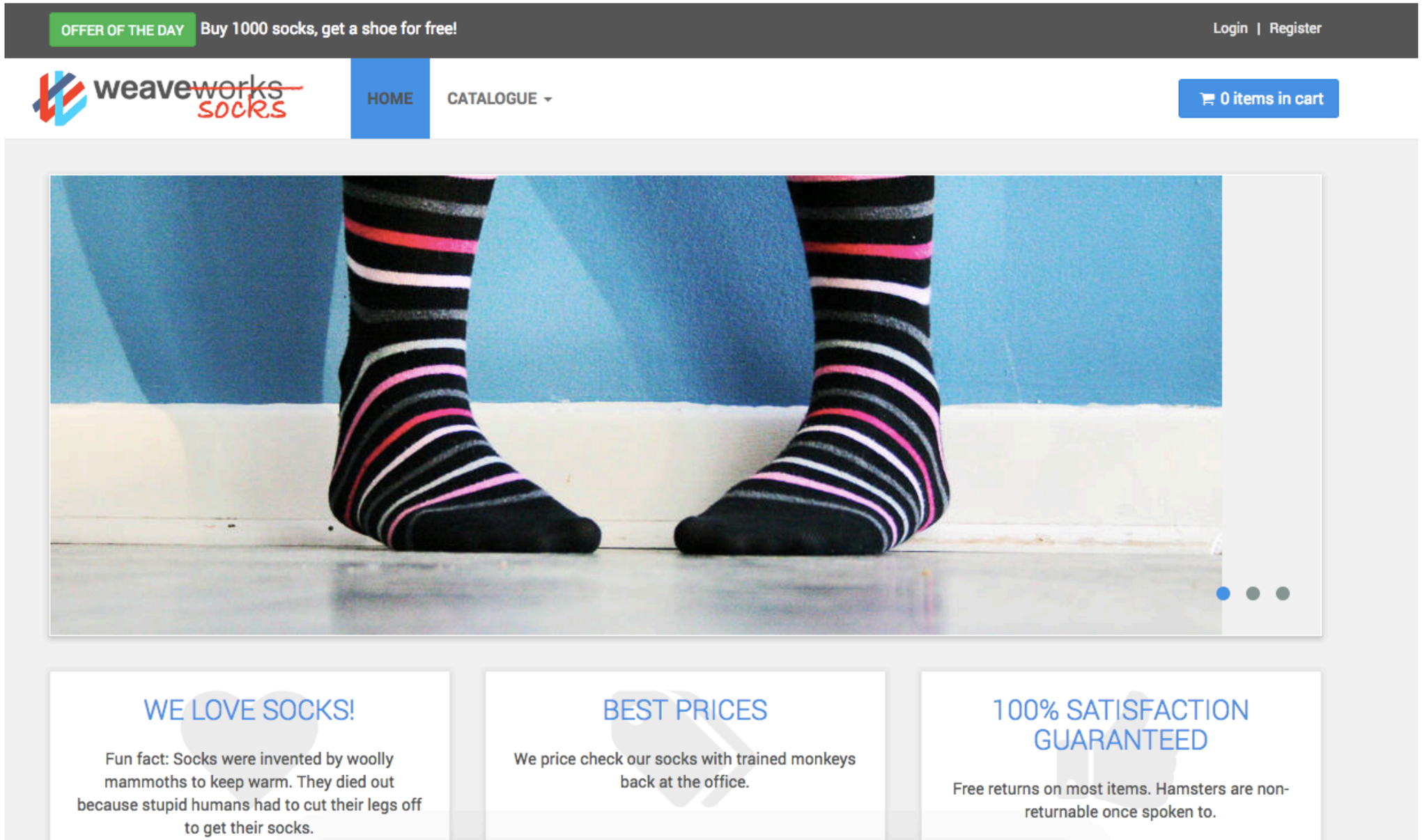
MESOS · Shipper · CONSUL · VNS3 · Marathon · fleet · ATOMIC

System Integrators

OpDemand · Wiredcraft · MomentumSI · InfoSiftr · Flux7 · SHADOW SOFT · Dev9 · relevance lab · IZURI

Demonstration - Microservices Architecture

http://128.199.177.121/



https://github.com/microservices-demo/microservices-demo

Demonstration - Microservices Architecture

Monitoring : http://128.199.177.121:4040

“With microservices you can create an architecture ecosystem that allows you to change all the time.”

- Accenture

Thank You