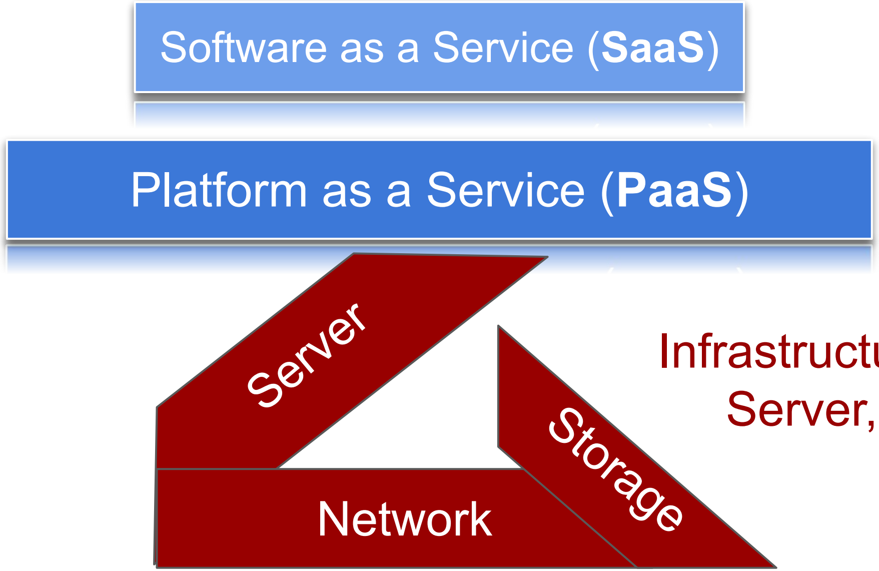
Cloud Computing
An Evolution or Revolution
云计算的前世今生

Kai Mike Zhang
Solution Architect @ IBM
mkz100@gmail.com

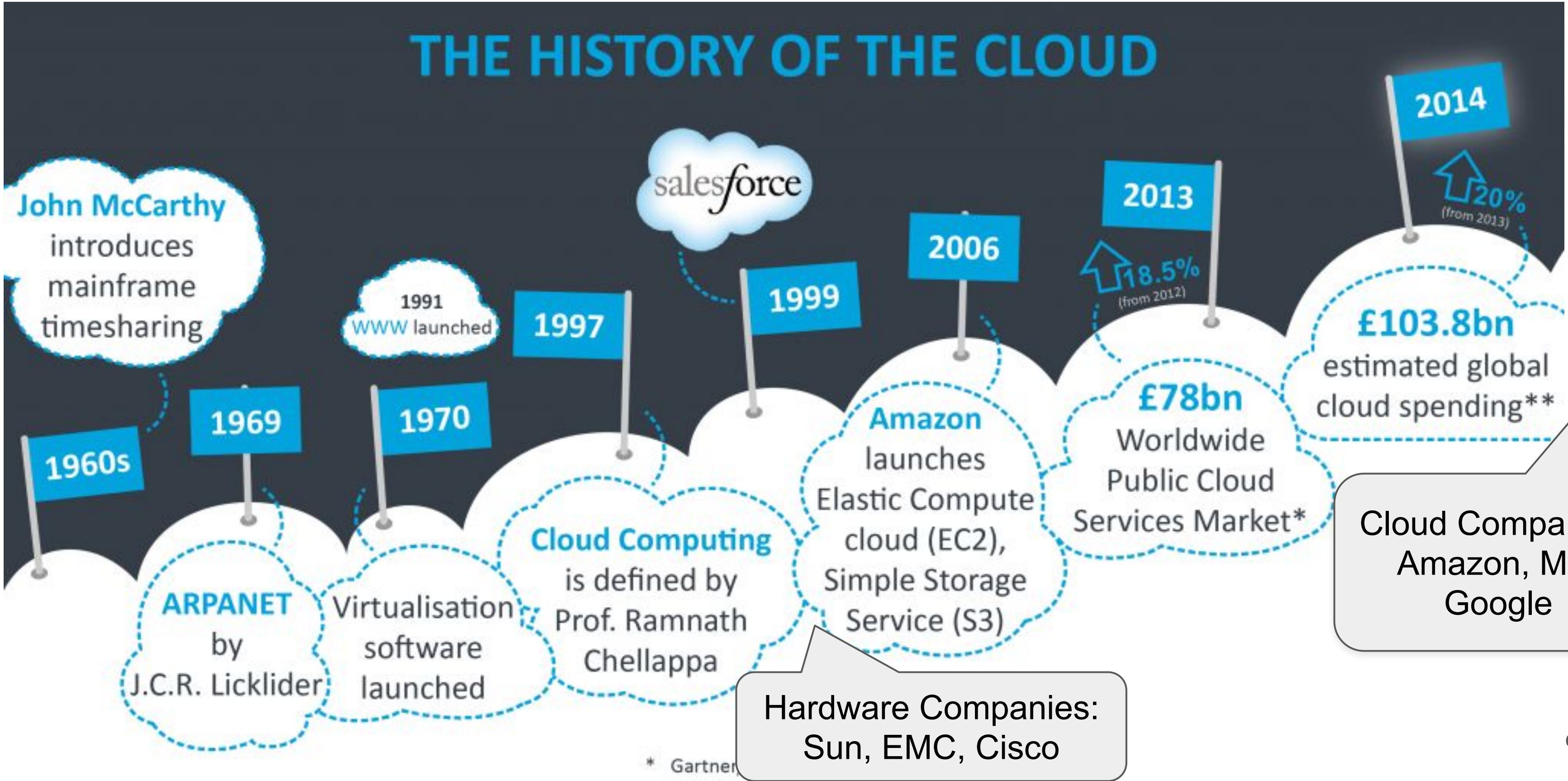
Cloud Computing Concepts
Embodied in Chinese Character “云” (Cloud)


Software as a Service (**SaaS**)
Platform as a Service (**PaaS**)
Server
Storage
Network

Infrastructure as a Service (**IaaS**)
Server, Network & Storage

Kai Zhang, mkz100@gmail.com

Cloud Computing Timeline



THE HISTORY OF THE CLOUD

1960s: John McCarthy introduces mainframe timesharing

1969: ARPANET by J.C.R. Licklider

1970: Virtualisation software launched

1991 WWW launched

1997: Cloud Computing is defined by Prof. Ramnath Chellappa

1999: salesforce

2006: Amazon launches Elastic Compute cloud (EC2), Simple Storage Service (S3)

2013: 18.5% (from 2012) £78bn Worldwide Public Cloud Services Market*

2014: 20% (from 2013) £103.8bn estimated global cloud spending**

* Gartner

Hardware Companies: Sun, EMC, Cisco

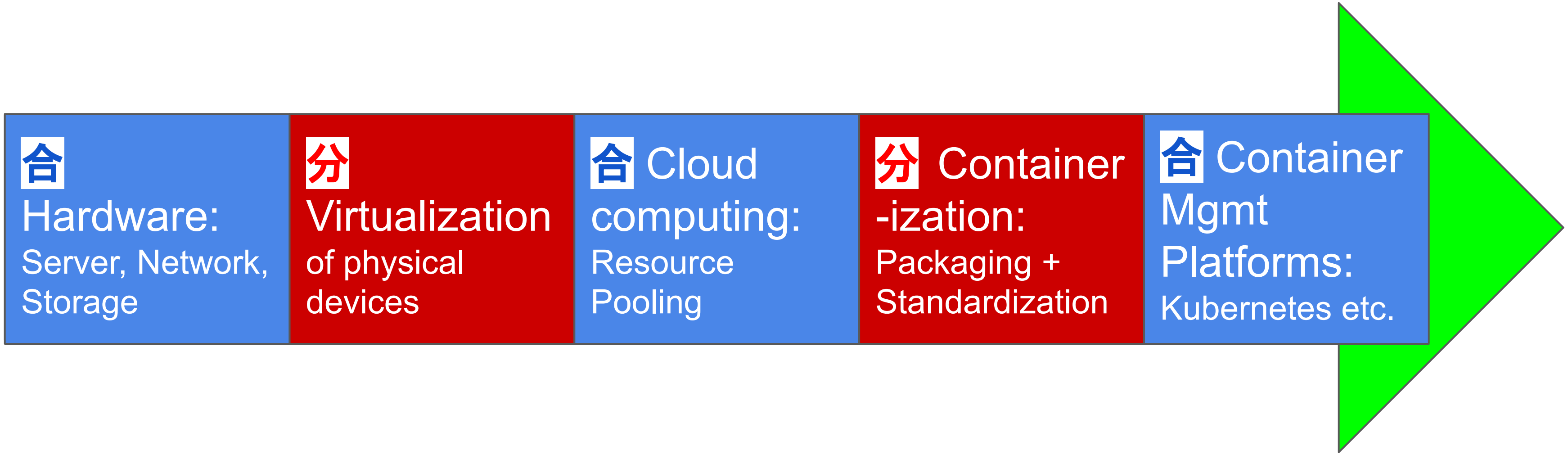
Cloud Companies: Amazon, MS, Google

com

Cloud Computing in Chinese Philosophy

“分久必合，合久必分”

A long-lasting separation is always followed by a reunion; vice versa!



合 Hardware: Server, Network, Storage

分 Virtualization of physical devices

合 Cloud computing: Resource Pooling

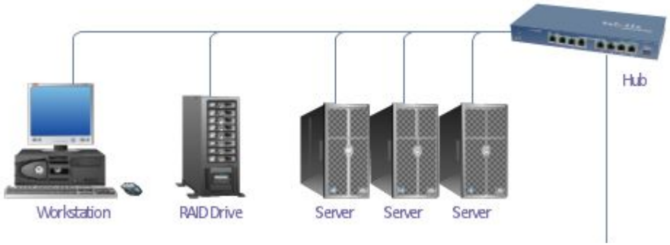
分 Container -ization: Packaging + Standardization

合 Container Mgmt Platforms: Kubernetes etc.

The Old Days Before Cloud Computing

合
Hardware:
Server, Network, Storage

- Buy & Manage your own Hardware and put them together -
  - Your own servers
  - Your own networking
  - Your own storages
- For a web site hosting Case:
  - Buy your own hardware
  - Buy & install your own software
  - Hire IT staff and software Engineers
  - Manage your own IT env.
- Problems :
  - On-site operation and management: very expensive and inefficient
  - Huge waste of under-utilizing resources
  - Poor isolation for resource sharing

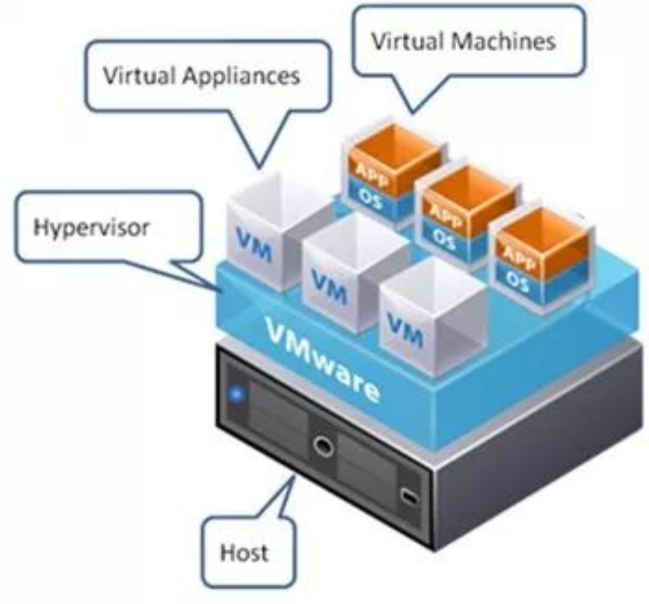


Kai Zhang, mkz100@gmail.com

Virtualization - Virtual Machines (VMs)

分
Virtualization
of physical
devices

- Virtualization help separate the physical hardware into virtual devices:
  - Virtual machines with virtual CPUs, memories, storages, networks …
- The benefits of virtual machines:
  - Allow remote and easy management;
    Recreate a VM in mins
  - Better resource sharing:
    physical resources can be divided into many VMs.
  - Better isolation:
    VMs don't interfere each other

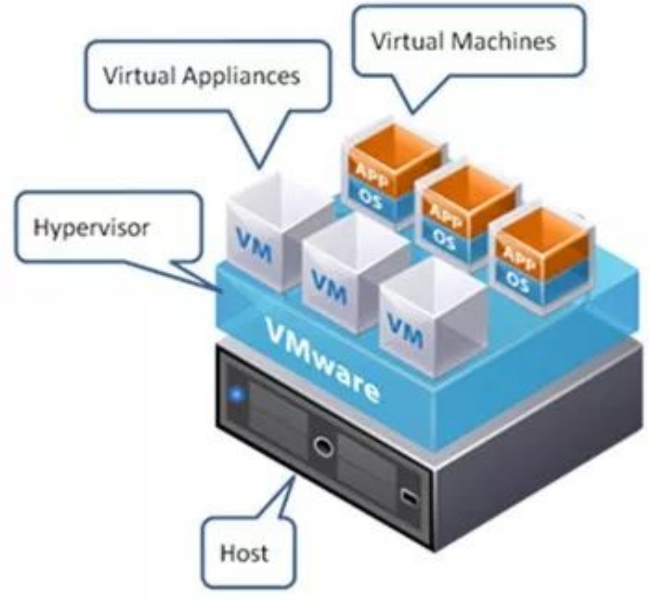


Kai Zhang, mkz100@gmail.com

Virtualization - Virtual Machines (VMs)

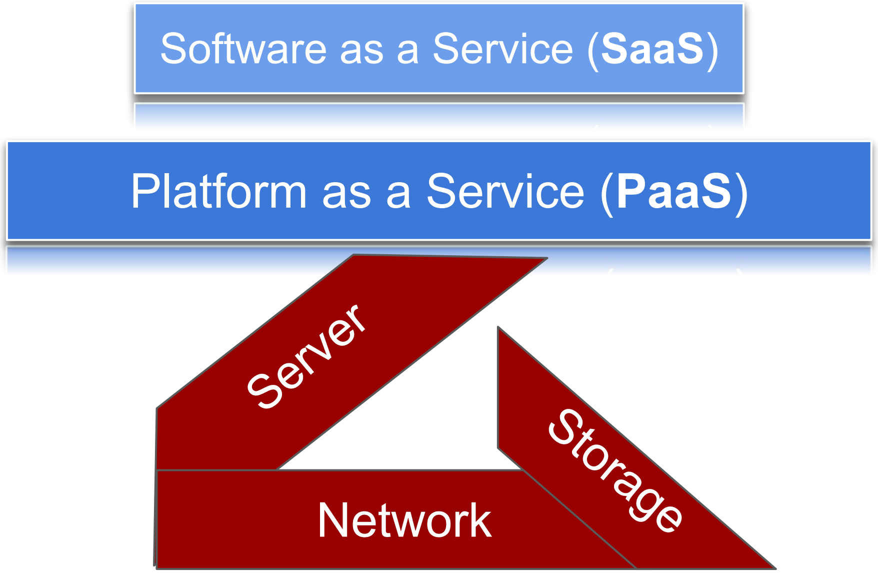
分
Virtualization
of physical
devices

- For a web site hosting Case:
    - Buy less hardware
    - Buy & install less software
    - Hire less IT staff and software Engineers
    - Manage your own IT env more efficiently
- Still Problems :
    - Still have to buy hardware
    - Still have to buy & install software
    - Still have to hire IT staff and software Engineers
    - Still have to manage your own IT env.



Cloud Computing

合 Cloud computing: Resource Pooling



Software as a Service (**SaaS**)

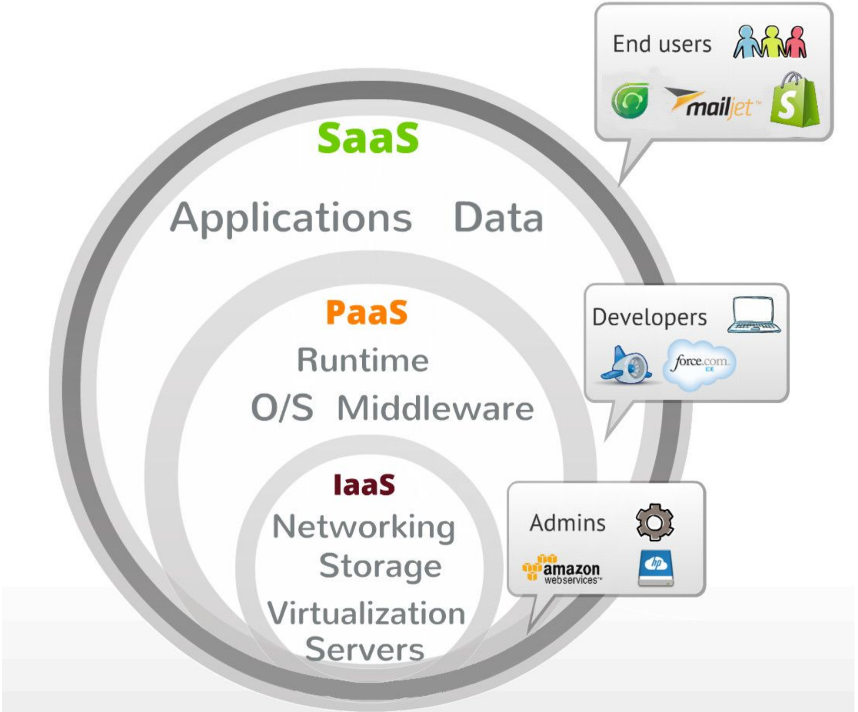
Platform as a Service (**PaaS**)

Server

Storage

Network

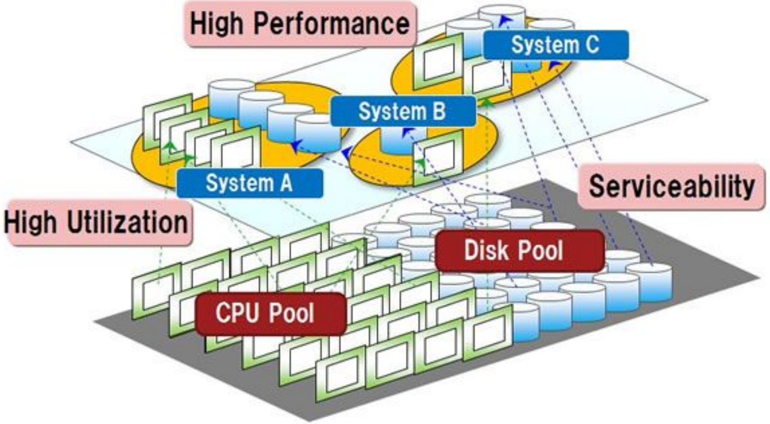
Infrastructure as a Service (**IaaS**)
Server, Network & Storage



Cloud Computing - IaaS

合 Cloud computing: Resource Pooling

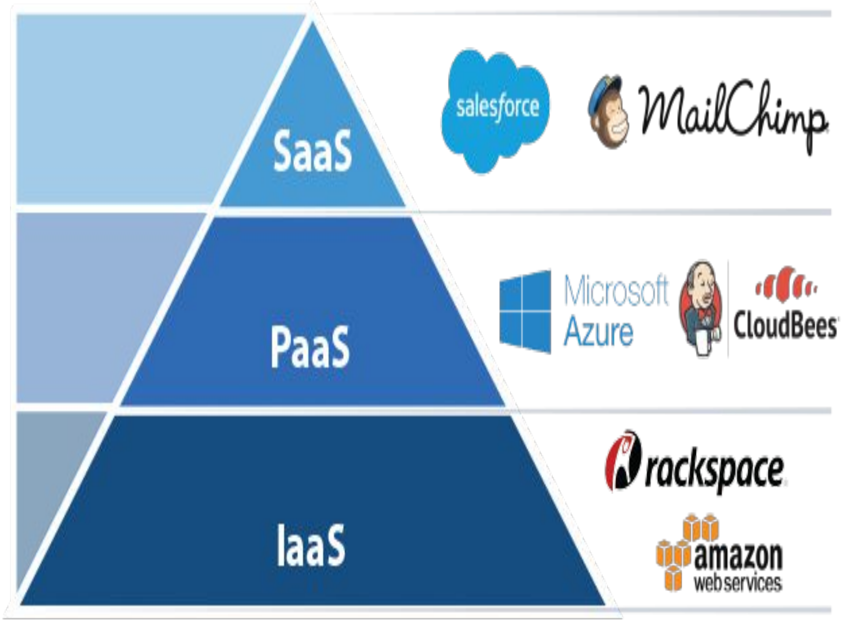
- Cloud Companies (IaaS) pool resources for sharing
  - VMs, storages, network resources are pooled together
- The benefits :
  - Much better sharing of resources
  - Dynamic provisioning & scale (Elastic Computing)
  - Much easier & more efficient to operation & maintain
  - Manage your own IT env.



🏠 Cloud computing: Resource Pooling

Cloud Computing - IaaS

- For a web site hosting Case:
  - Rent hardware instead of buy
  - Buy & install less software
  - Hire less IT staff and software Engineers
  - Don't need to manage your own hardware env
- Still Problems :
  - Still have to buy & install software
  - Still have to hire IT staff and software Engineers



Kai Zhang, mkz100@gmail.com

合 Cloud computing: Resource Pooling

Cloud Computing - **PaaS**

- The problem with IaaS :
    - Still have to buy, install and manage the software
- Take a web site hosting Case as an example:
    - A website created with PHP is usually hosted on LAMP stack
        - L: Linux OS
        - A: Apache web server
        - M: MySQL database
        - P: PHP runtime
    - With IaaS, you still has to install all those software yourselves.
    - With PaaS, those tasks are taken care for you.
- For a web site hosting Case:
        - Rent hardware instead of buy
        - No need for buy, install and manage the software
        - No need to hire IT staff because no IT env to manage
        - Hire software Engineers to create the web site

合 Cloud computing: Resource Pooling

Cloud Computing - **SaaS**

- The Problem with PaaS
  - Still have to hire software Engineers to create & maintain the website
- A SaaS provider can help tremendously :
  - It will provide the pre-built applications that can be easily customized for the customers
  - It will manage all hardware and software for the customers
  - It can provide dynamic provisioning & scale (Elastic Computing) instantly upon requests
- Benefits for the customers:
  - No need to hire a single IT and software staff
  - No need to invest in hardware and software
  - The website design and creation is done quickly and efficiently by the professionals
  - Leverage the powerful and proven pre-built applications / modules.

Containerization

分 Container -ization: Packaging + Standardization

- The Problems with Cloud deployment and portability
  - DevOps requires quick and reliable deployment from Dev to Test to Production
  - Customers might want to migrate their app from one vendor to another (e.g. Amazon to Google)
- Are the applications easy to move from one cloud to another?
  - Not quite!! Because
    - Applications have to be run on the exact same runtime envs AND
    - The runtime envs usually differs on different clouds.
  - A real example :
    - When we try to duplicate a PaaS web hosting env on the Aliyun Silicon Valley datacenter. We find it is very hard to make it identical to our Beijing datacenter. e.g.
      - The Apache is not available on Silicon Valley datacenter, instead of we have to use NGinx web server
      - The database version is different too.
      - There are other configuration issues.

Containerization

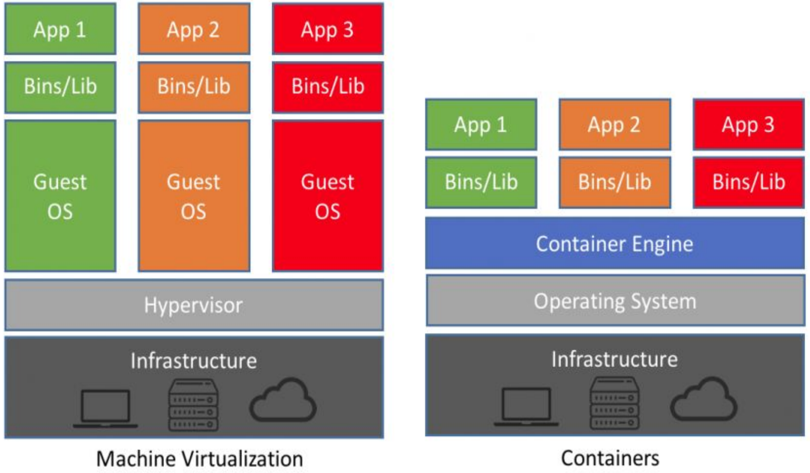
分 Container
-ization:
Packaging +
Standardization

- In short, moving applications from one cloud to another is HARD
- Containerization is born for addressing this issue
- Think of what a real container does when moving goods from China to US
  - In old days without containers, you have to load the goods piece by piece on a truck and unload and reload piece by piece to a freight train and do the same for a ship … and all over again at the other end.
  - With containers, just load your goods in a container and it will be shipped fast and with fewer problems on the way.

Containerization

分 Container -ization: Packaging + Standardization

- How containers helps :
  - Packaging : everything is packed together. No need to move and reassemble piece by piece.
  - Standardization : the container is designed in standard size and weight etc. It can be easily ported from trucks to trains to ships without any problems.
- Containers vs Virtual Machines -
  - Containers provide a way to virtualize an OS so that multiple workloads can run on a single OS instance.
  - With VMs, the hardware is being virtualized to run multiple OS instances.
  - Containers are lightweight, fast vs VMs.



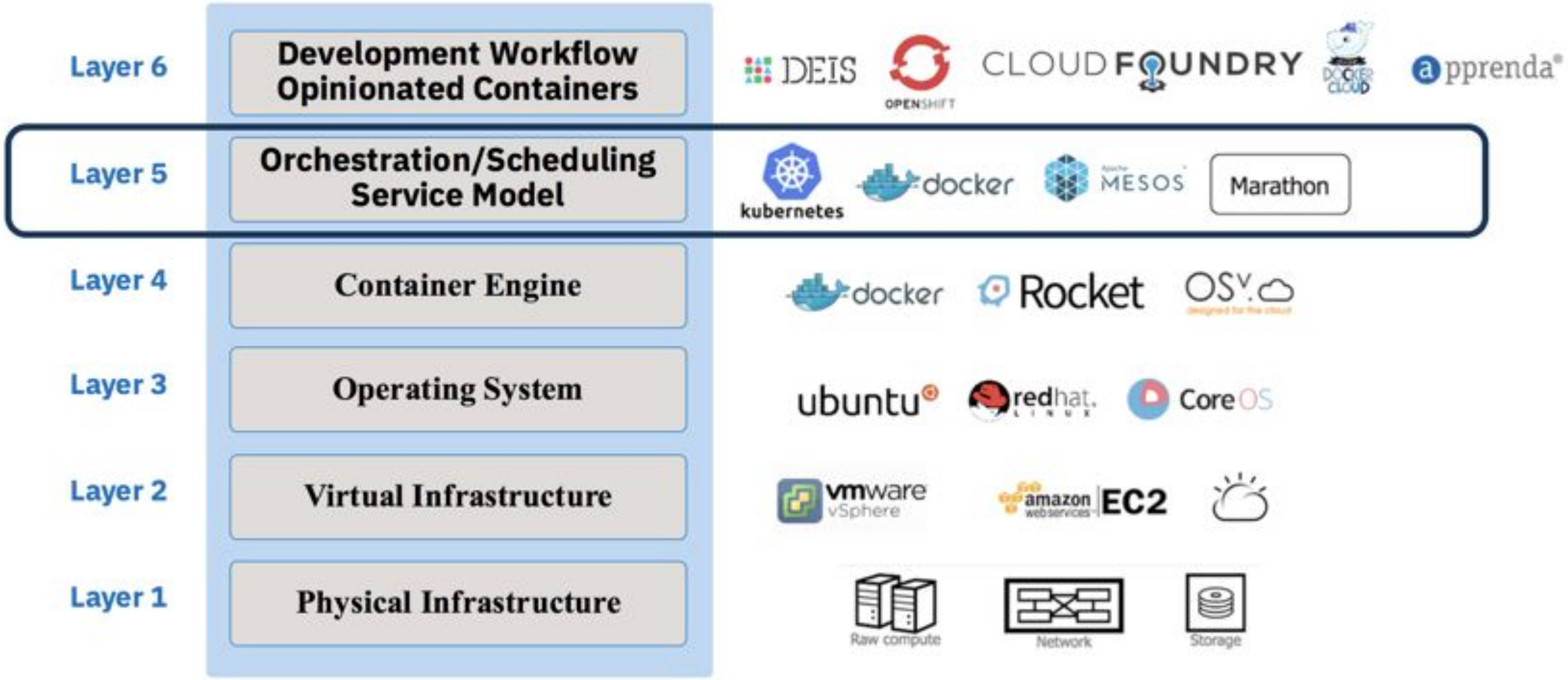
Kai Zhang, mkz100@gmail.com

Container Management & Orchestration

合 Container Mgmt Platforms:
Docker etc.

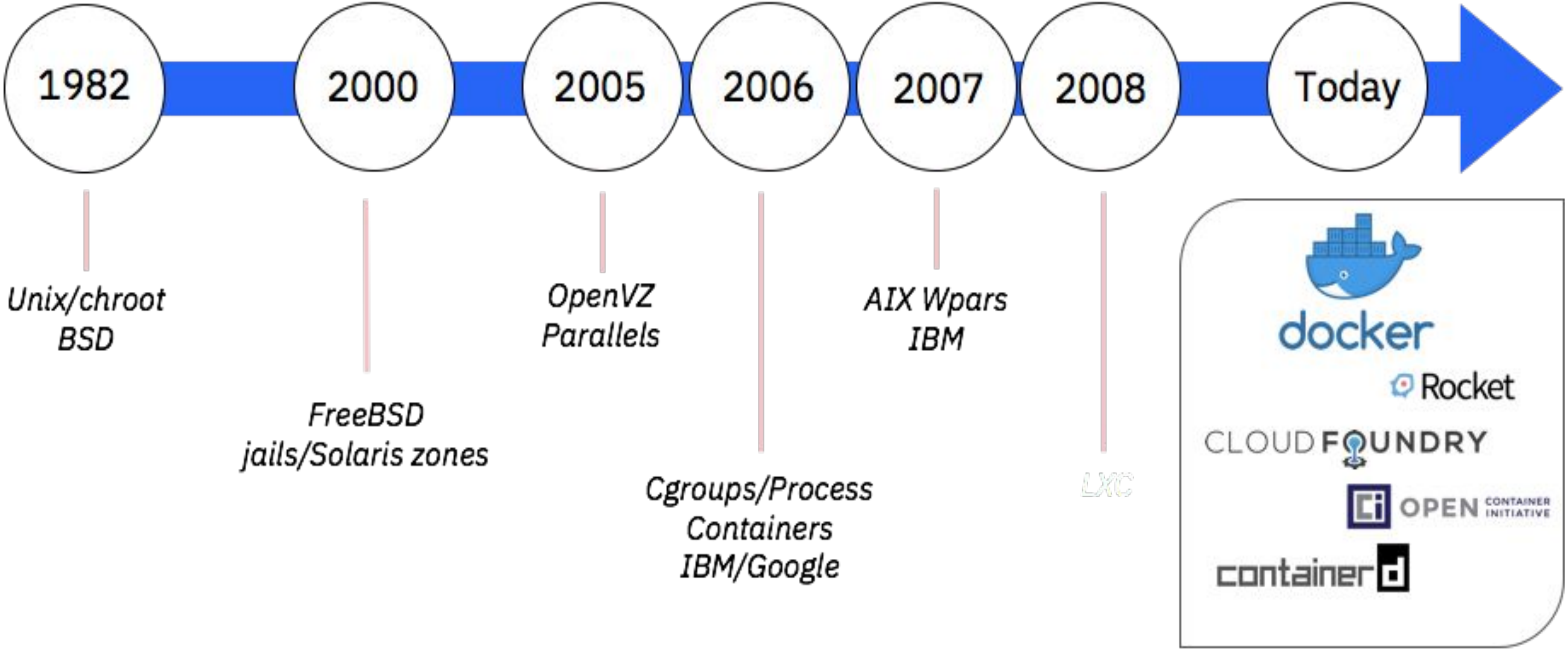
- Challenge :
  - 100 servers;
  - 10 VMs on each server;
  - 10 containers on each VM
  - 10,000 containers total to manage!
- Container Management Platform Automation :
  - Self-discovery
  - Self-repair
  - Auto-scaling

Container Management Layer



Layer 6	Development Workflow Opinionated Containers
Layer 5	Orchestration/Scheduling Service Model
Layer 4	Container Engine
Layer 3	Operating System
Layer 2	Virtual Infrastructure
Layer 1	Physical Infrastructure

A Brief Container History



1982 – Unix/chroot BSD

2000 – FreeBSD jails/Solaris zones

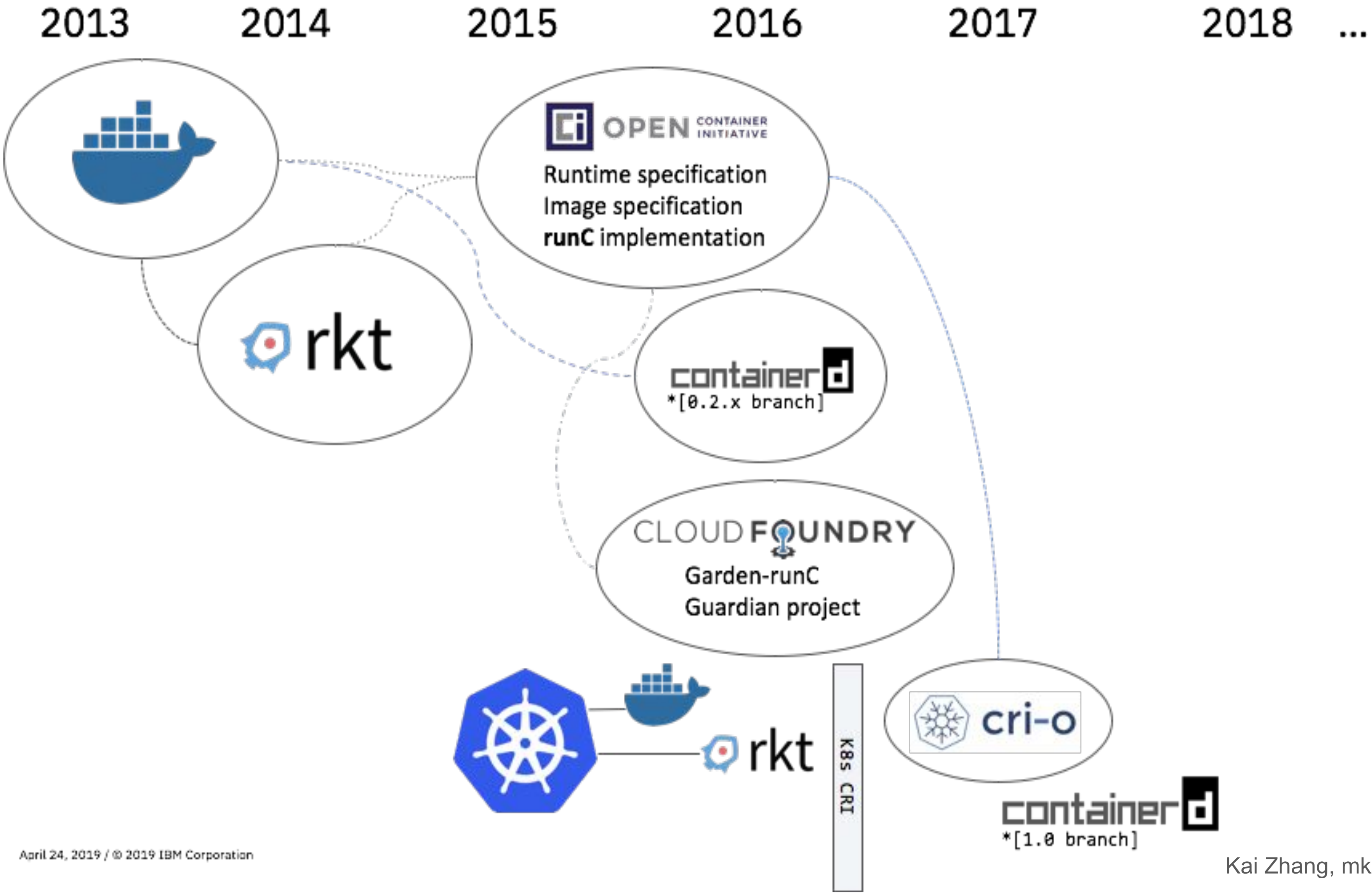
2005 – OpenVZ Parallels

2006 – Cgroups/Process Containers IBM/Google

2007 – AIX Wpars IBM

2008 – LXC

Today – docker, Rocket, CLOUD FOUNDRY, OPEN CONTAINER INITIATIVE, containerd



2013 2014 2015 2016 2017 2018 ...

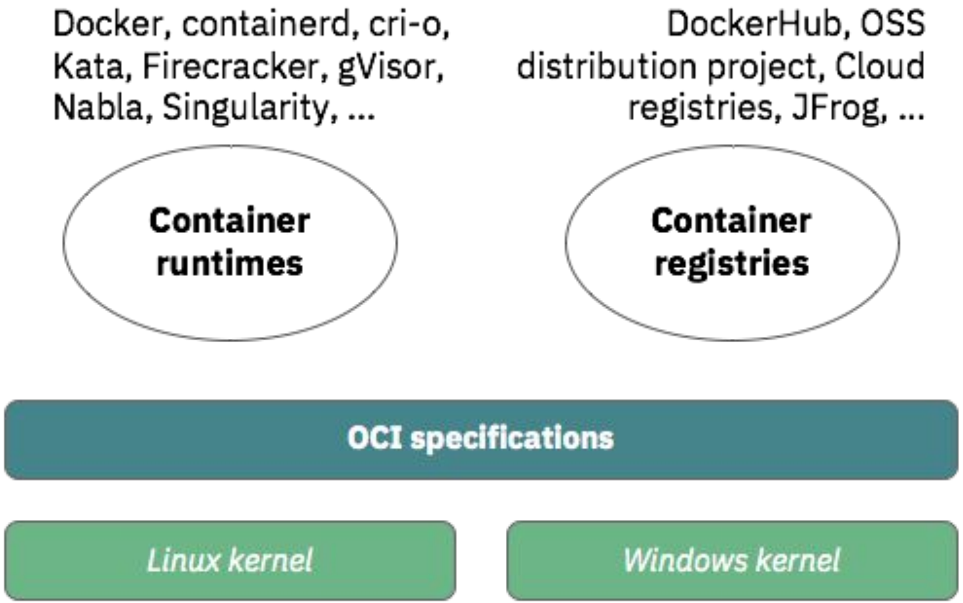
OPEN CONTAINER INITIATIVE

Runtime specification
Image specification
runC implementation

Garden-runC
Guardian project

Kai Zhang, mkz100@gmail.com

A Standard Container Substrate



Docker, containerd, cri-o, Kata, Firecracker, gVisor, Nabla, Singularity, …

DockerHub, OSS distribution project, Cloud registries, JFrog, …

Container runtimes

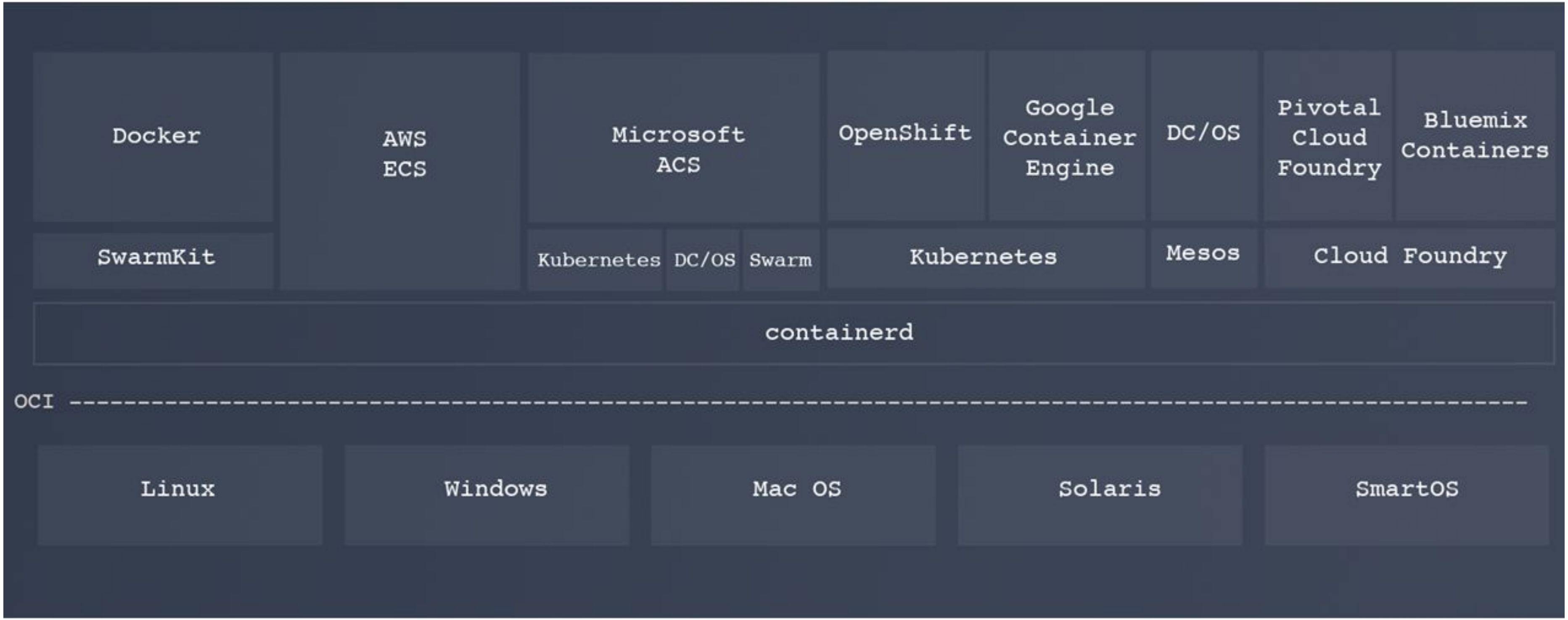
Container registries

OCI specifications

Linux kernel

Windows kernel

containerd’s Role in Container Ecosystem



Docker

SwarmKit

AWS ECS

Microsoft ACS

Kubernetes

DC/OS

Swarm

OpenShift

Google Container Engine

Kubernetes

DC/OS

Mesos

Pivotal Cloud Foundry

Bluemix Containers

Cloud Foundry

containerd

OCI

Linux

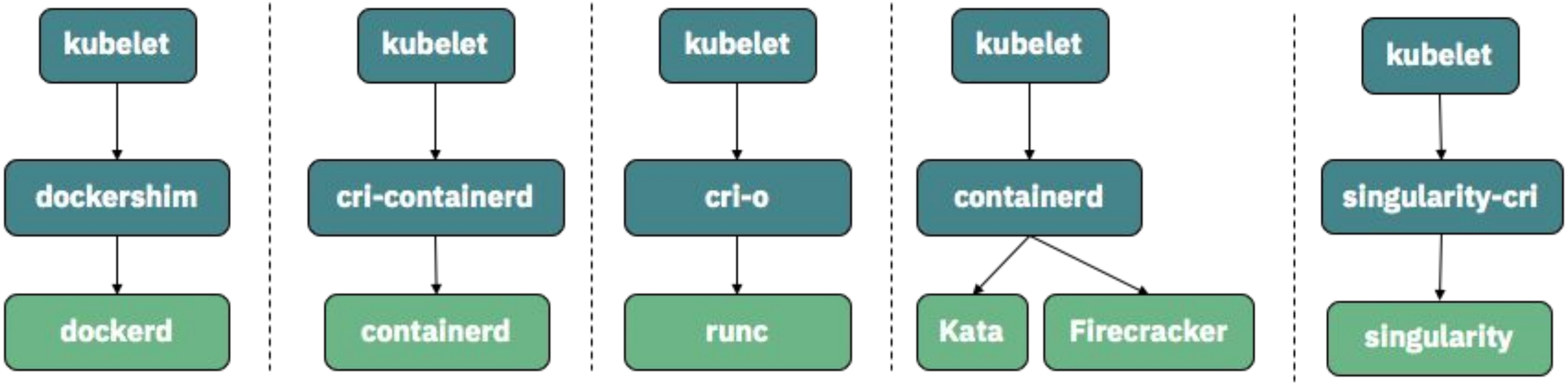
Windows

Mac OS

Solaris

SmartOS

What CRI Runtimes Exist?



```
kubelet --container-runtime {string}
        --container-runtime-endpoint {string}
```

Kai Zhang, mkz100@gmail.com

CRI Implementations

containerd

- A stable, core, performant core container runtime for the cloud
- Has a CRI implementation, and is a CNCF graduated project

cri-o

- “all the runtime Kubernetes needs and nothing more”; RH created
- CRI implementation over runc and 2 open libraries; K8s incubator

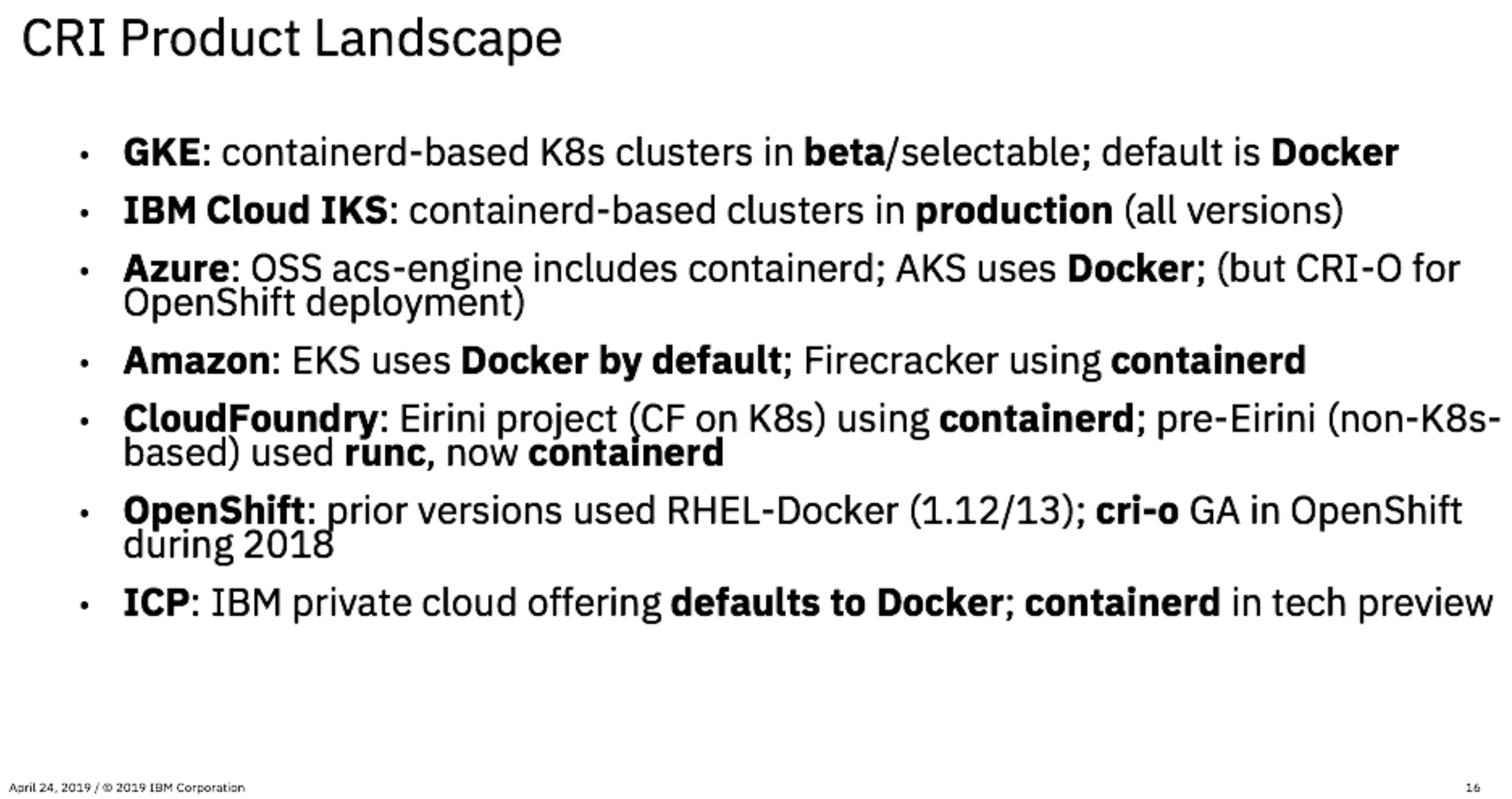
katacontainers

- Intel Clear Containers + Hyper.sh combined project
- Lightweight virtualization (KVM/qemu) under cri-o and containerd

Firecracker

- Amazon open source project announced Nov 2018; lightweight virt.
- Uses Rust-based VMM instead of qemu; plugs into containerd

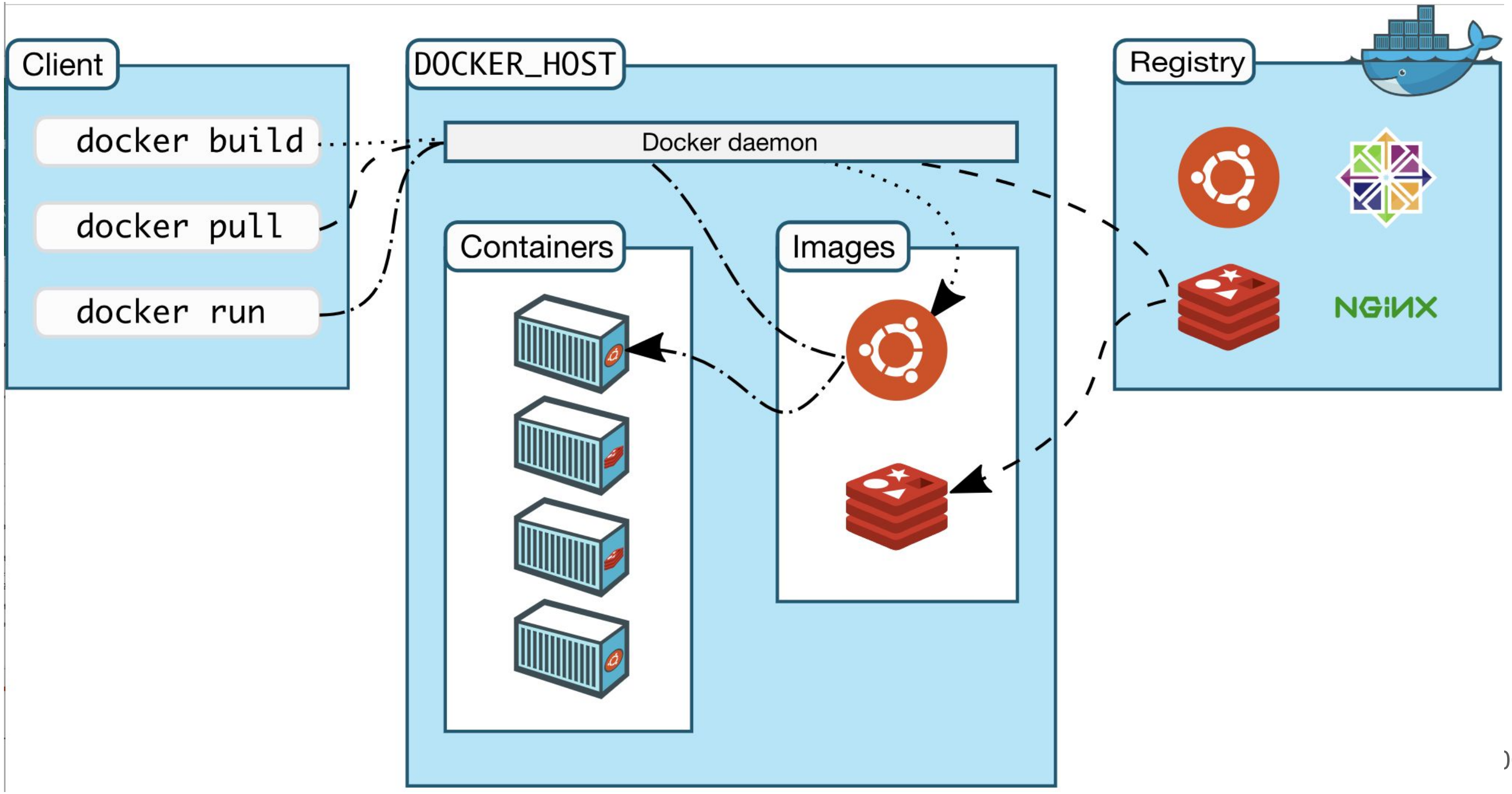
- CRI implementation over Sylabs Singularity runtime project
- Userbase traditionally from academia/HPC use cases



CRI Product Landscape

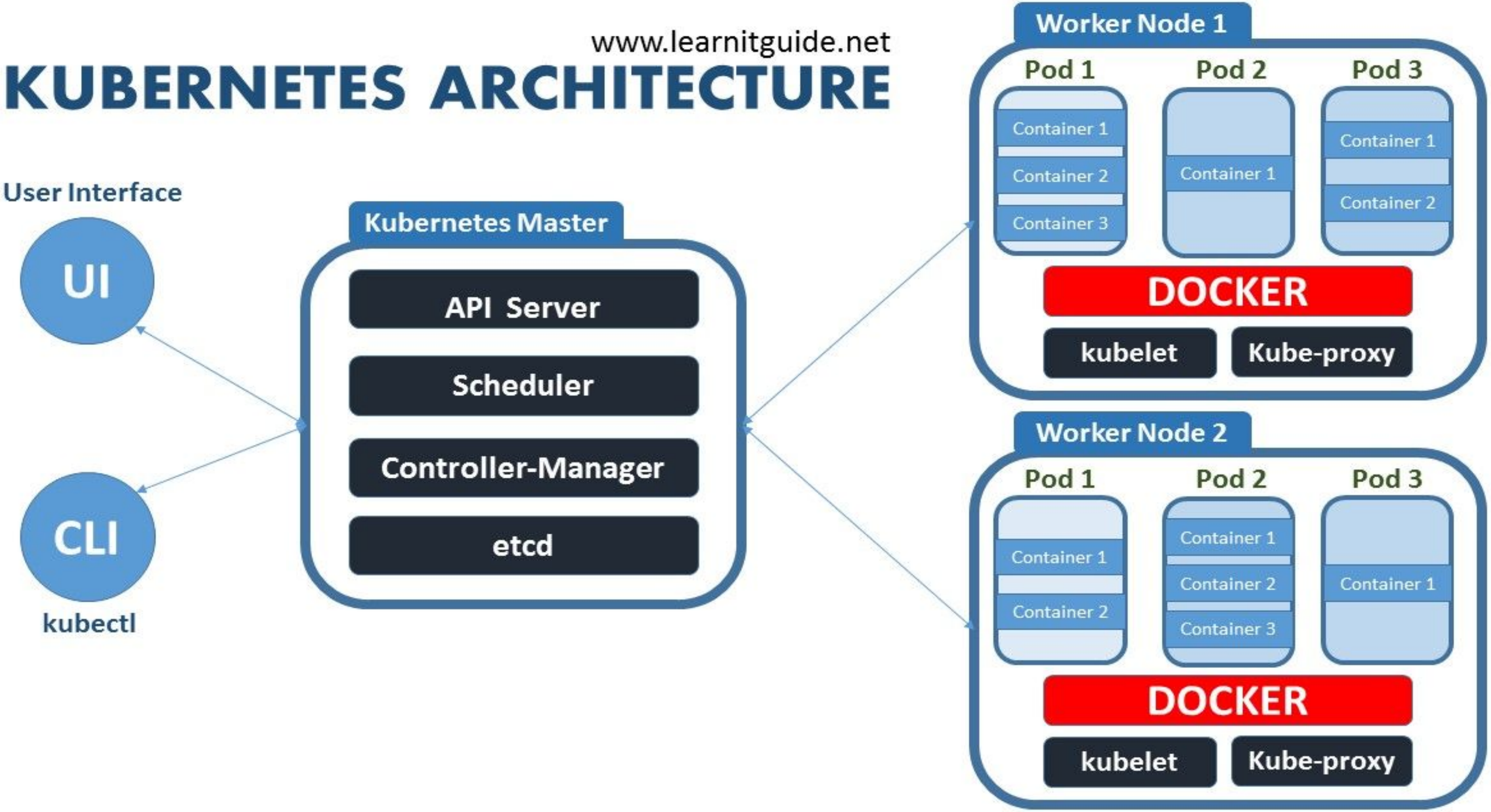
- **GKE**: containerd-based K8s clusters in **beta**/selectable; default is **Docker**
- **IBM Cloud IKS**: containerd-based clusters in **production** (all versions)
- **Azure**: OSS acs-engine includes containerd; AKS uses **Docker**; (but CRI-O for OpenShift deployment)
- **Amazon**: EKS uses **Docker by default**; Firecracker using **containerd**
- **CloudFoundry**: Eirini project (CF on K8s) using **containerd**; pre-Eirini (non-K8s-based) used **runc**, now **containerd**
- **OpenShift**: prior versions used RHEL-Docker (1.12/13); **cri-o** GA in OpenShift during 2018
- **ICP**: IBM private cloud offering **defaults to Docker**; **containerd** in tech preview

Docker Architecture



)0@gmail.com

KUBERNETES ARCHITECTURE



www.learnitguide.net

User Interface

UI

CLI

kubectl

Kubernetes Master

API Server

Scheduler

Controller-Manager

etcd

Worker Node 1

Pod 1: Container 1, Container 2, Container 3

Pod 2: Container 1

Pod 3: Container 1, Container 2

DOCKER

kubelet

Kube-proxy

Worker Node 2

Pod 1: Container 1, Container 2

Pod 2: Container 1, Container 2, Container 3

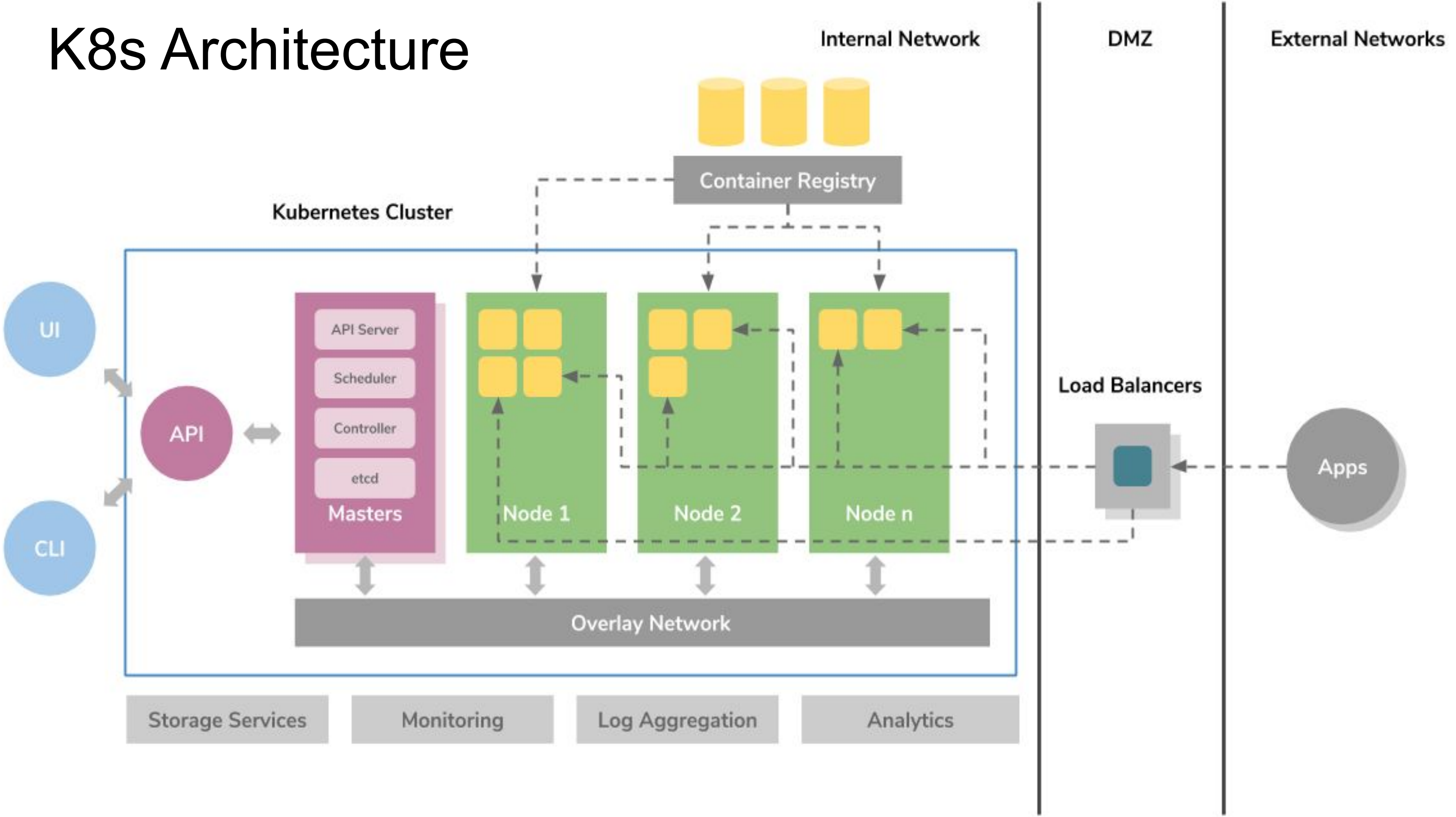
Pod 3: Container 1

DOCKER

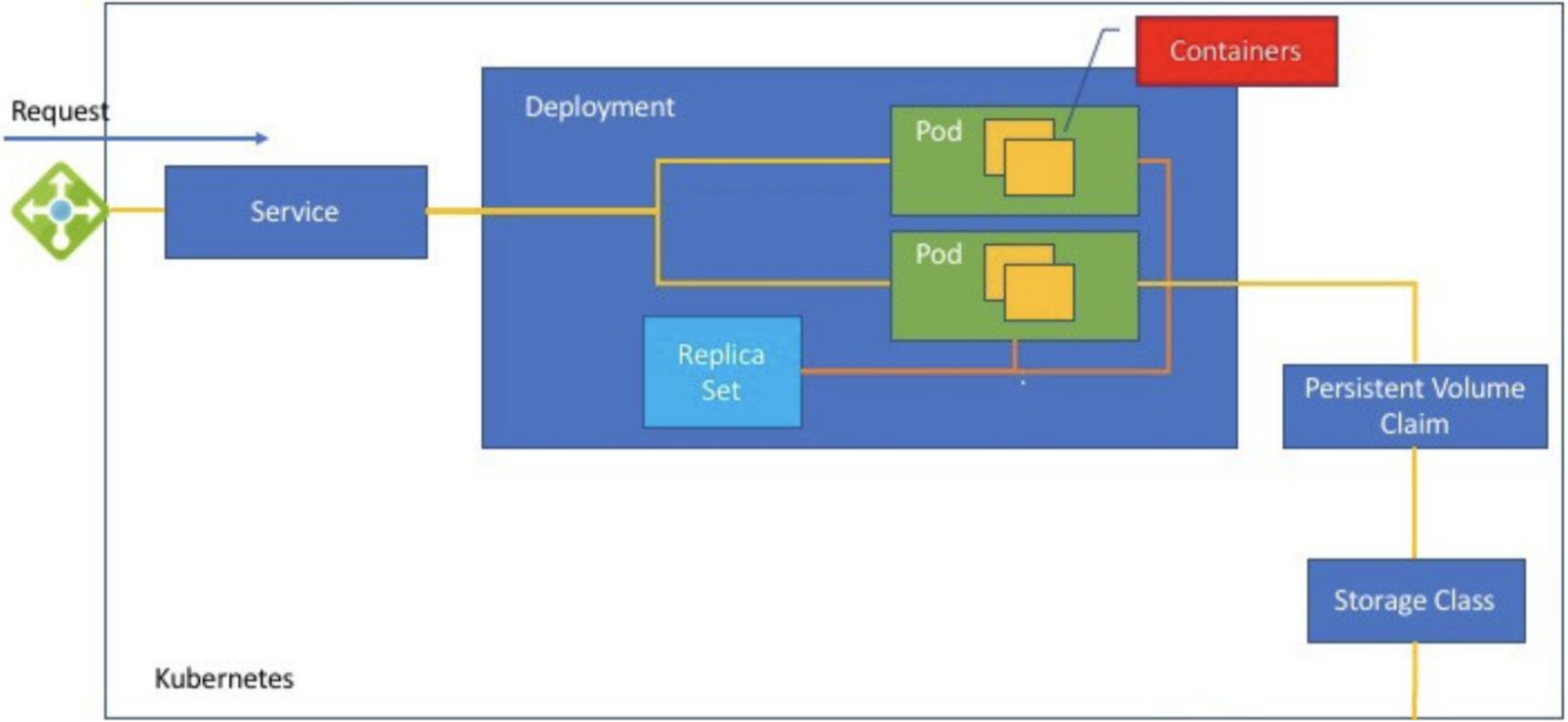
kubelet

Kube-proxy

K8s Architecture

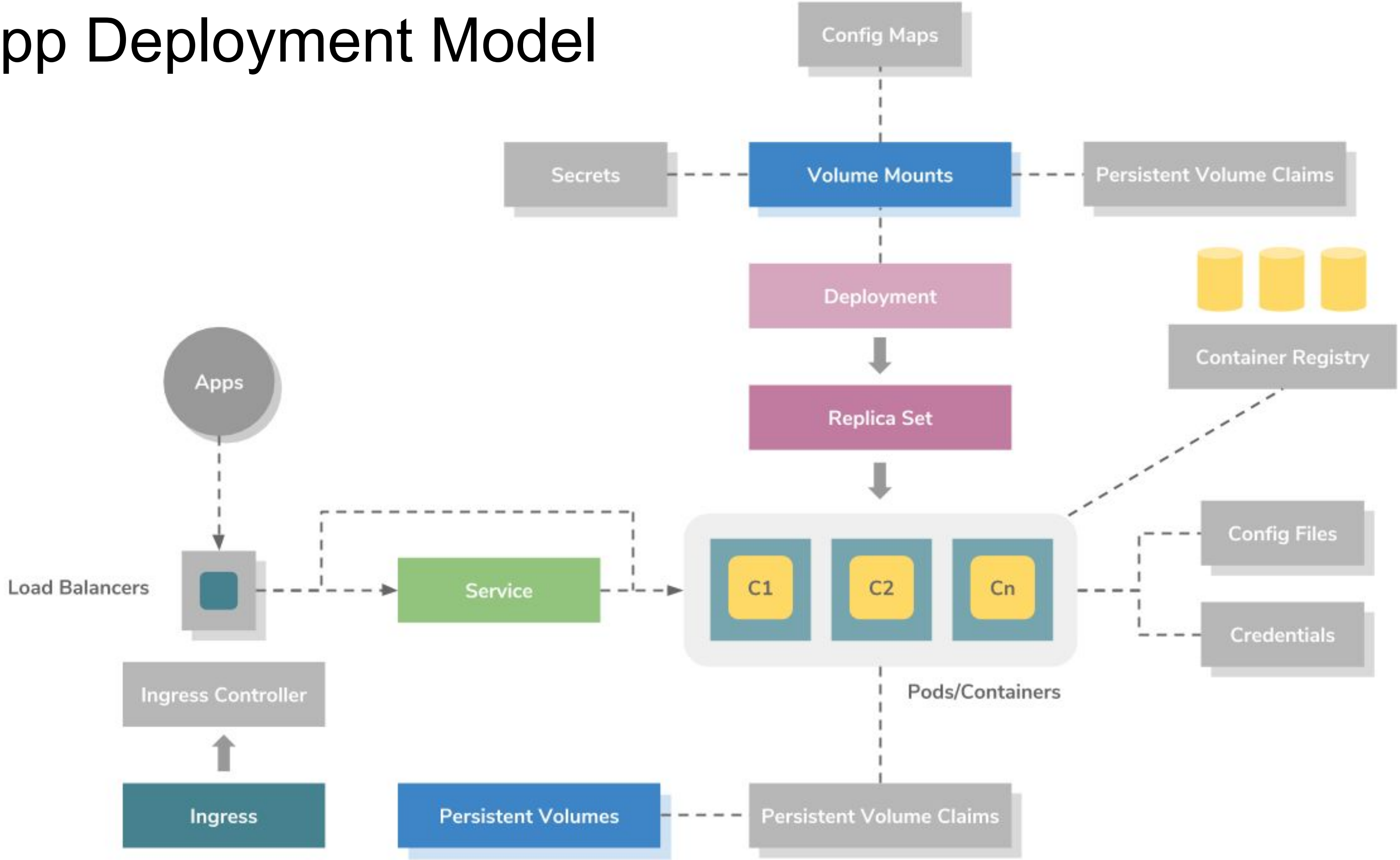


Kubernetes Objects

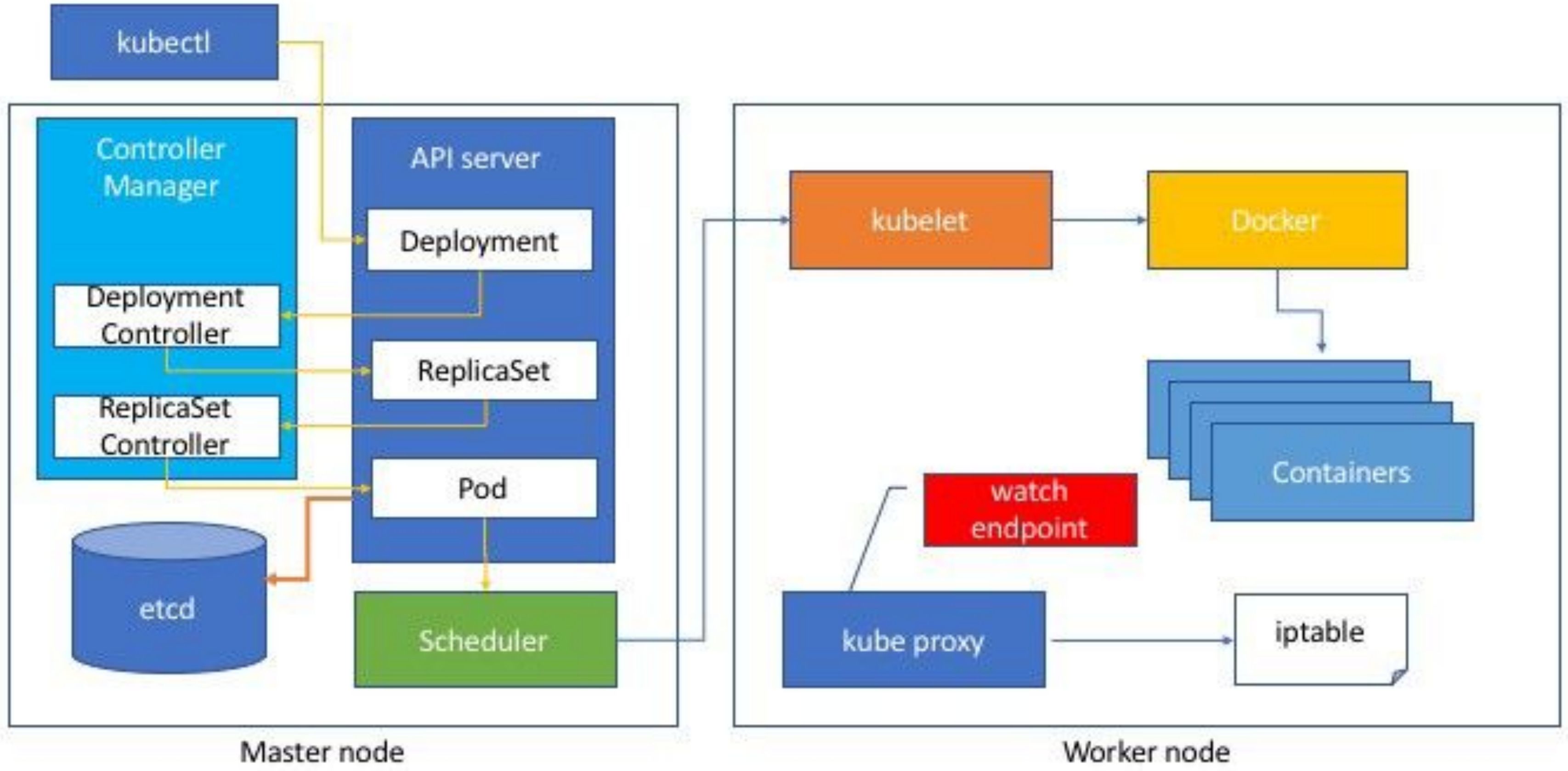


Kai Zhang, mkz100@gmail.com

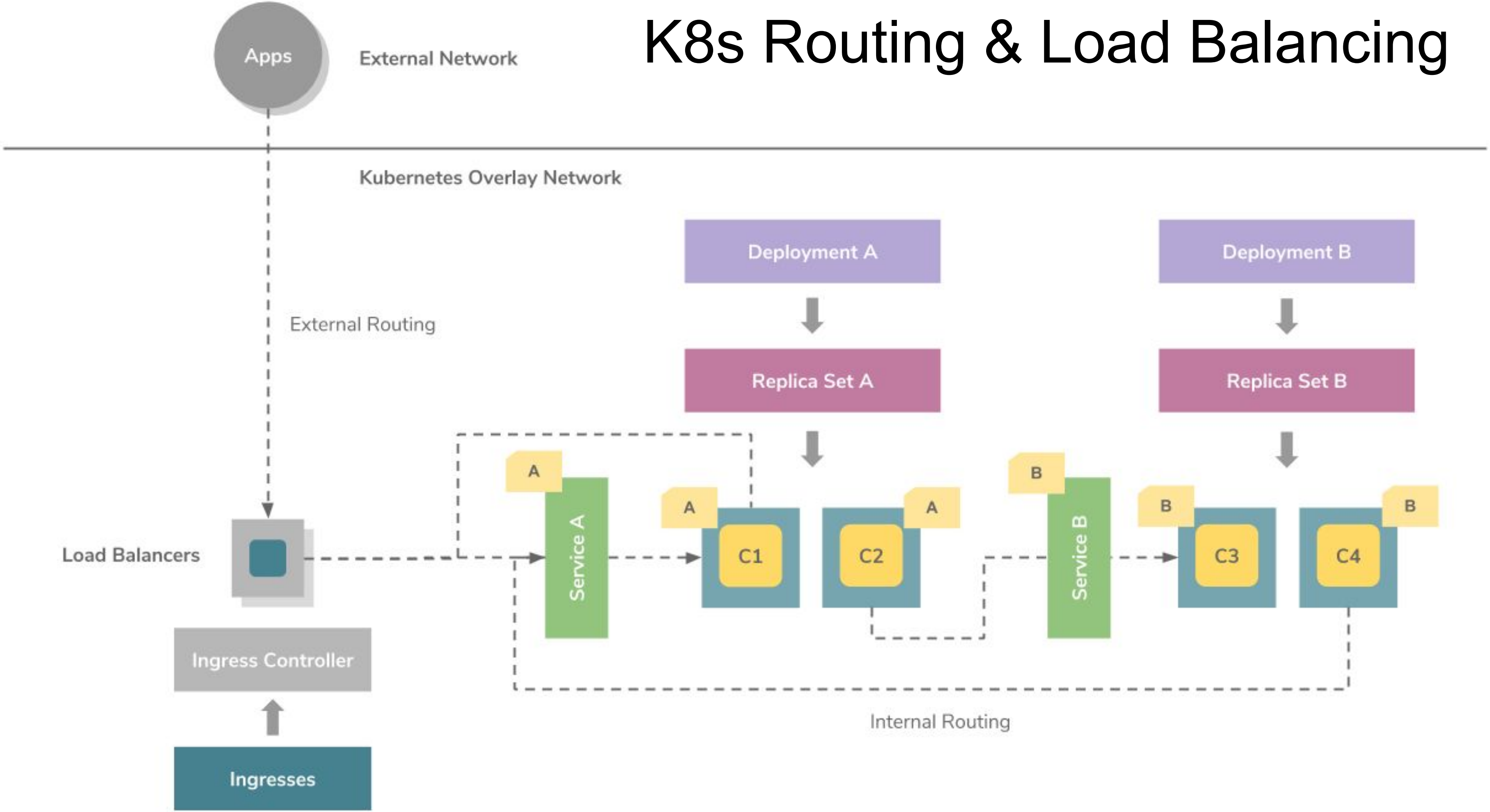
App Deployment Model



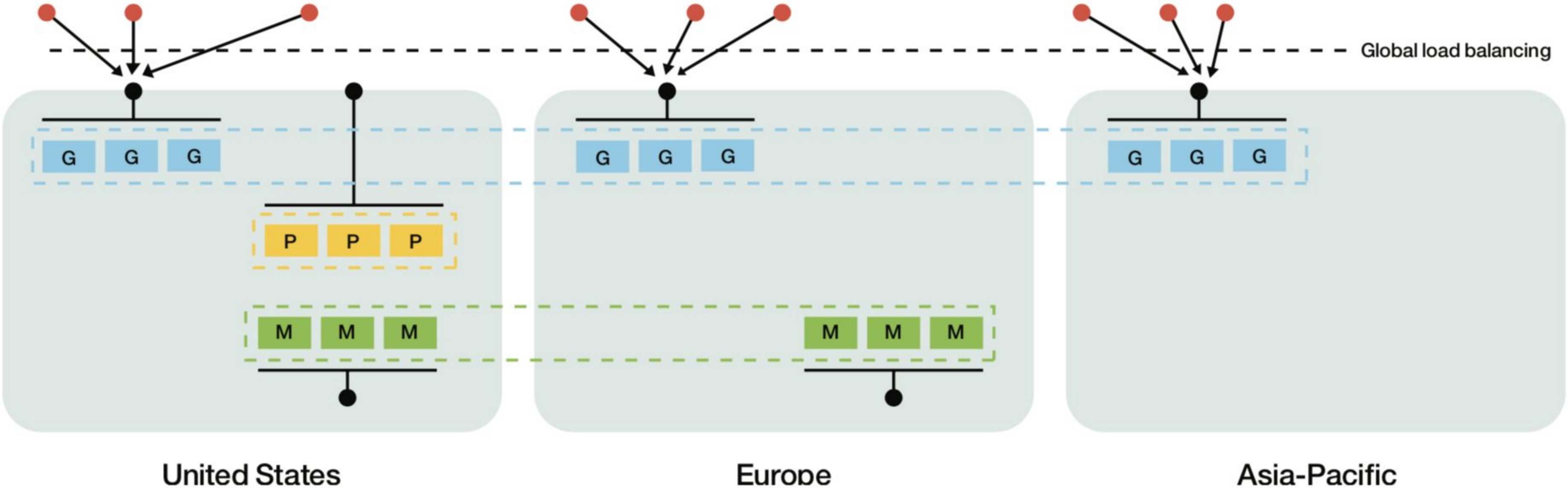
Kubernetes App Deployment Flow



K8s Routing & Load Balancing

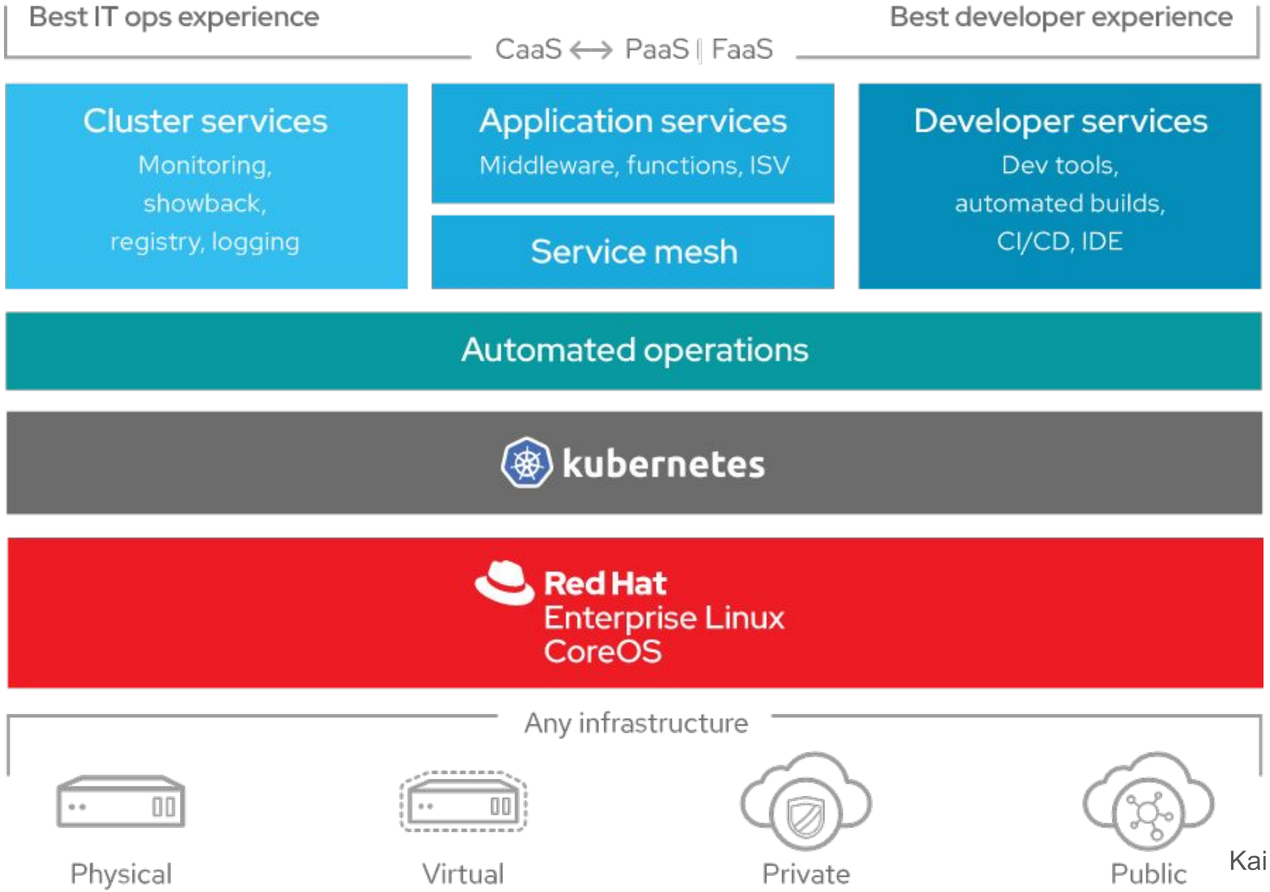


Kubernetes High Availability Solution Architecture



Kai Zhang, mkz100@gmail.com

OpenShift a hybrid cloud, enterprise Kubernetes application platform



Best IT ops experience

Best developer experience

CaaS ⟷ PaaS | FaaS

Cluster services

Monitoring, showback, registry, logging

Application services

Middleware, functions, ISV

Service mesh

Developer services

Dev tools, automated builds, CI/CD, IDE

Automated operations

kubernetes

Red Hat Enterprise Linux CoreOS

Any infrastructure

Physical

Virtual

Private

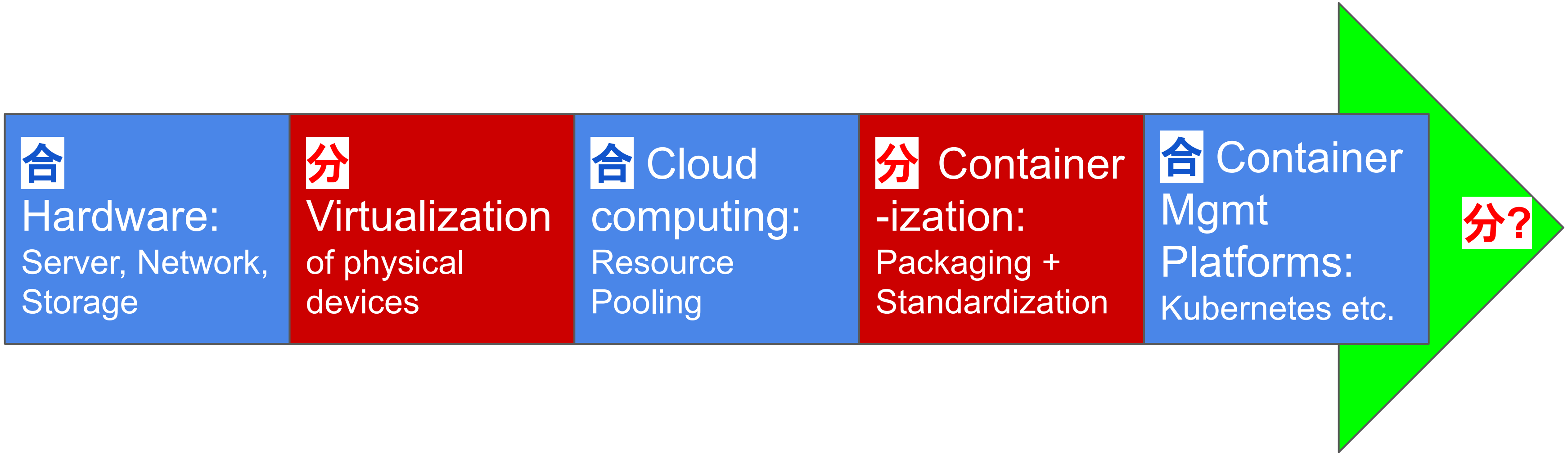
Public

Kai Zhang, mkz100@gmail.com

Thank YOU! And time to separate ...

“分久必合，合久必分”

A long-lasting separation is always followed by a reunion; vice versa!



合 Hardware: Server, Network, Storage

分 Virtualization of physical devices

合 Cloud computing: Resource Pooling

分 Container -ization: Packaging + Standardization

合 Container Mgmt Platforms: Kubernetes etc.

分?

Resources

- Docker sites:
    - https://www.docker.com/ Official site
    - https://hub.docker.com Registry for container images
    - Docker Essentials: A Developer Introduction https://cognitiveclass.ai/courses/docker-essentials
    - A simple, interactive and fun **playground to learn Docker** https://labs.play-with-docker.com/
- Kubernetes Sites:
    - https://kubernetes.io/ Official site, kubectl cmd reference doc
    - A Beginner’s Guide to Kubernetes
    - The Illustrated Children's Guide to Kubernetes https://www.youtube.com/watch?v=4ht22ReBjno
    - http://kubernetesbyexample.com/ A hands-on introduction to Kubernetes by examples
    - The Kubernetes Cheat Sheet
    - A simple, interactive and fun **playground to learn Kubernetes** https://labs.play-with-k8s.com/
    - Learn Kubernetes w/ hands labs and scenarios https://www.katacoda.com/courses/kubernetes
- Cloud Native Computing Foundation https://www.cncf.io/projects/
    - **CNCF Trail Map** https://github.com/cncf/landscape/blob/master/README.md#trail-map
    - **Cloud Native Interactive Landscape** https://landscape.cncf.io/