CLOUD NATIVE
COMPUTING FOUNDATION

Migrating Legacy Monoliths to Cloud Native Microservices Architectures on Kubernetes

Dan Kohn, Executive Director

This presentation is available at:
https://github.com/cncf/presentations

Cloud Native Computing Foundation

- Non-profit, part of the Linux Foundation
  - Founded December 2015
- Current projects:

Orchestration | Monitoring | Distributed Tracing API | Logging | Service Mesh | Remote Procedure Call | Service Discovery | Container Runtime | Container Runtime | Networking API | Service Mesh | Distributed Tracing

- Platinum members:

Today the Linux Foundation is much more than Linux

Security

We are helping global privacy and security through a program to encrypt the entire internet.

Networking

We are creating ecosystems around networking to improve agility in the evolving software-defined datacenter.

Cloud

We are creating a portability layer for the cloud, driving de facto standards and developing the orchestration layer for all clouds.

Automotive

We are creating the platform for infotainment in the auto industry that can be expanded into instrument clusters and telematics systems.

Blockchain

We are creating a permanent, secure distributed ledger that makes it easier to create cost-efficient, decentralized business networks.

Web

We are providing the application development framework for next generation web, mobile, serverless, and IoT applications.

Let's Encrypt

ONAP
OPEN NETWORK AUTOMATION PLATFORM

CLOUD NATIVE COMPUTING FOUNDATION

AUTOMOTIVE GRADE LINUX

HYPERLEDGER

node JS

We are regularly adding projects - for the most up-to-date listing of all projects visit – http://tlfprojects.org

Non-Virtualized Servers: Sun (2000)

- Launching a new application? Buy a new server; or a rack of them!
- Building block of your application is physical servers

Virtualization: VMWare (2001)

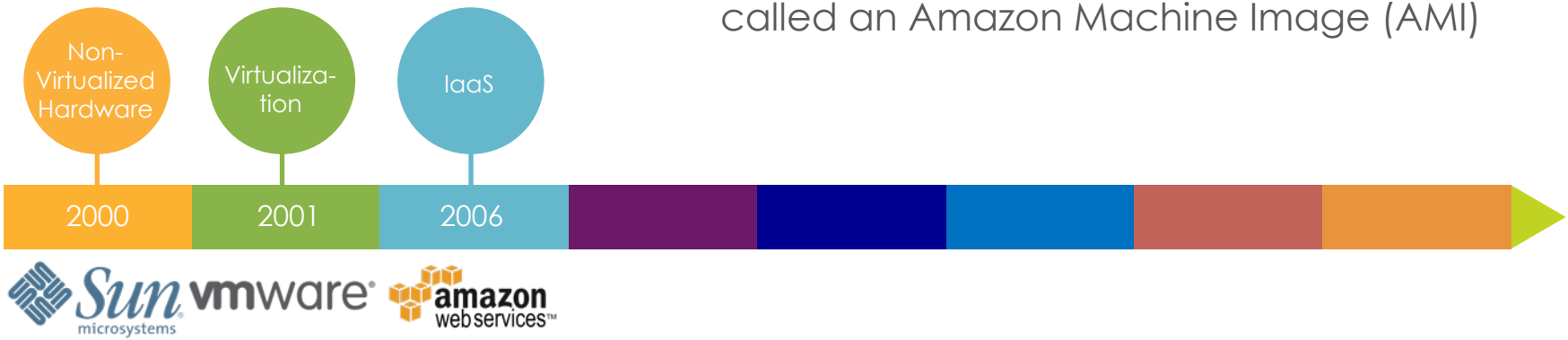
vmware®

- Releases for server market in 2001
- Popularizes virtual machines (VMs)
- Run many VMs on one physical machine, meaning you can buy fewer servers!
- Architectural building block becomes a VM

Sun microsystems vmware

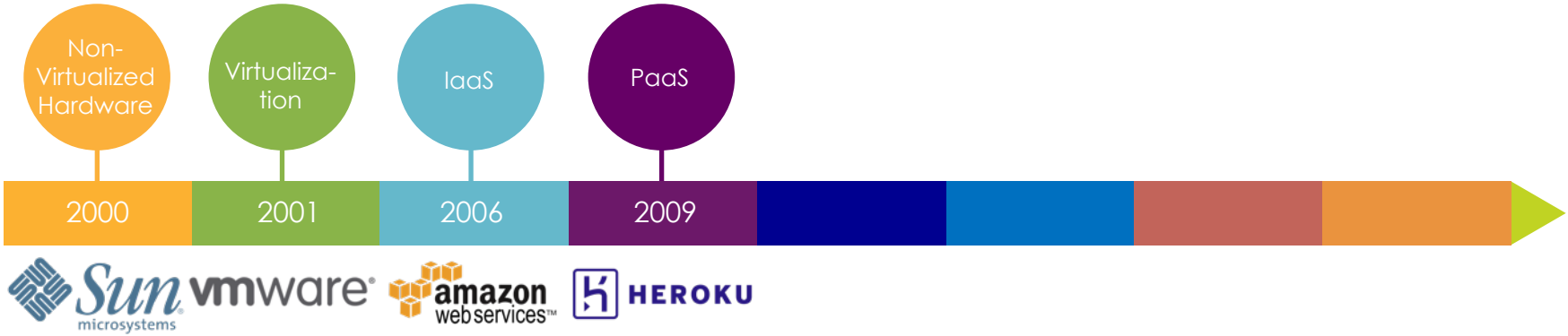
IaaS: AWS (2006)

- Amazon Web Services (AWS) creates the Infrastructure-as-a-Service market by launching Elastic Compute Cloud (EC2) in 2006
- Rent servers by the hour
- Convert CapEx to OpEx
- Architectural building block is also a VM, called an Amazon Machine Image (AMI)



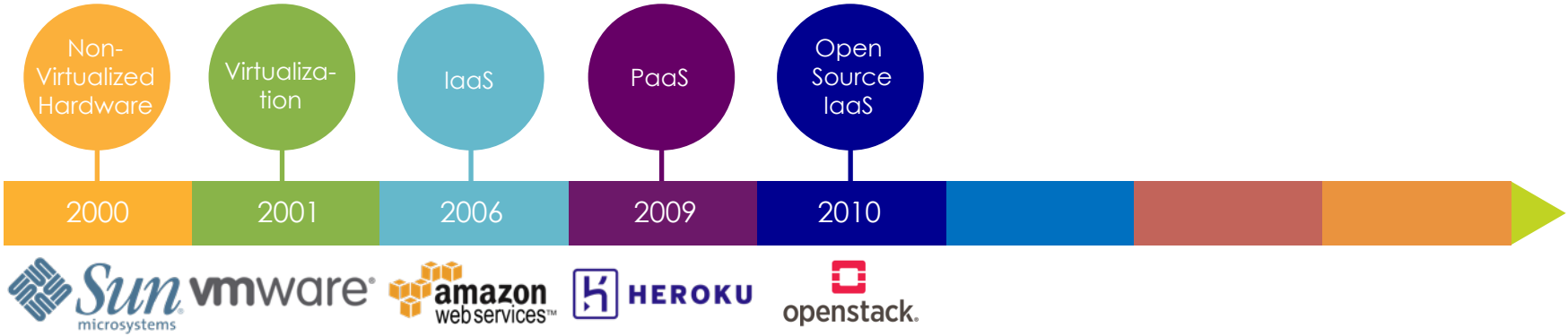
PaaS: Heroku (2009)

- Heroku popularizes Platform-as-a-Service (PaaS) with their launch in 2009
- Building block is a buildpack, which enables containerized 12-factor applications
  - The process for building the container is opaque, but:
  - Deploying new version of an app is just: `git push heroku`



Open Source IaaS: OpenStack (2010)

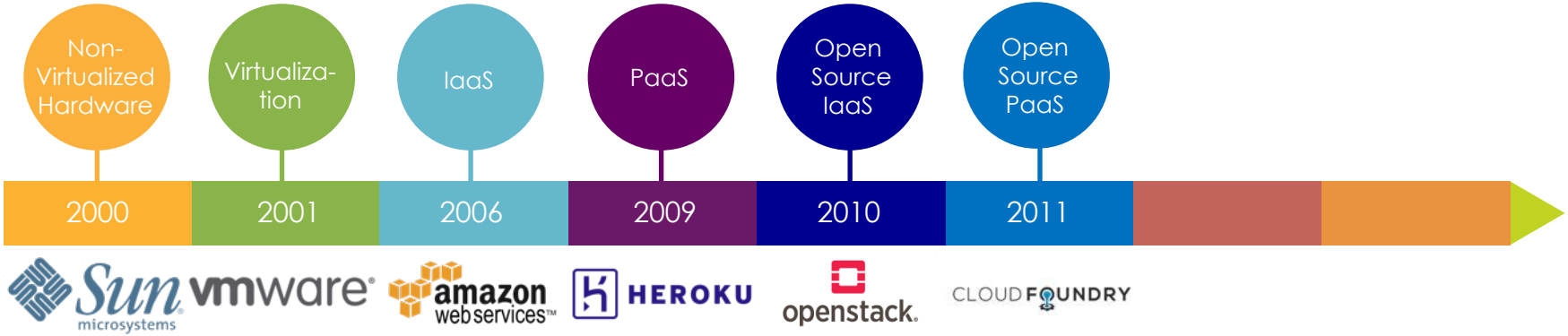
- OpenStack brings together an extraordinarily diverse group of vendors to create an open source Infrastructure-as-a-Service (IaaS)
- Competes with AWS and VMWare
- Building block remains a VM



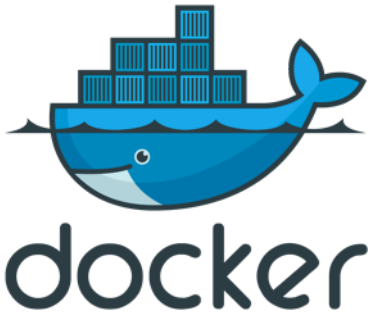
Open Source PaaS: Cloud Foundry (2011)

CLOUD FOUNDRY

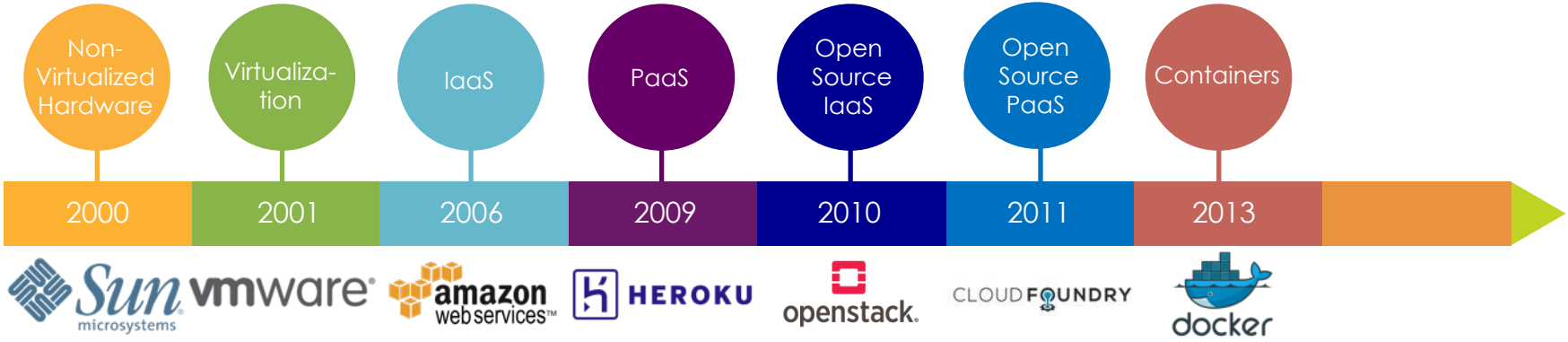
- Pivotal builds an open source alternative to Heroku's PaaS and launches the Cloud Foundry Foundation in late 2014
- Building block is Garden containers, which can hold Heroku buildpacks, Docker containers and even non-Linux OSes



Containers: Docker (2013)



- Docker combines LXC, Union File System and cgroups to create a containerization standard adopted by millions of developers around the world
- Fastest uptake of a developer technology ever
- Enables isolation, reuse and immutability

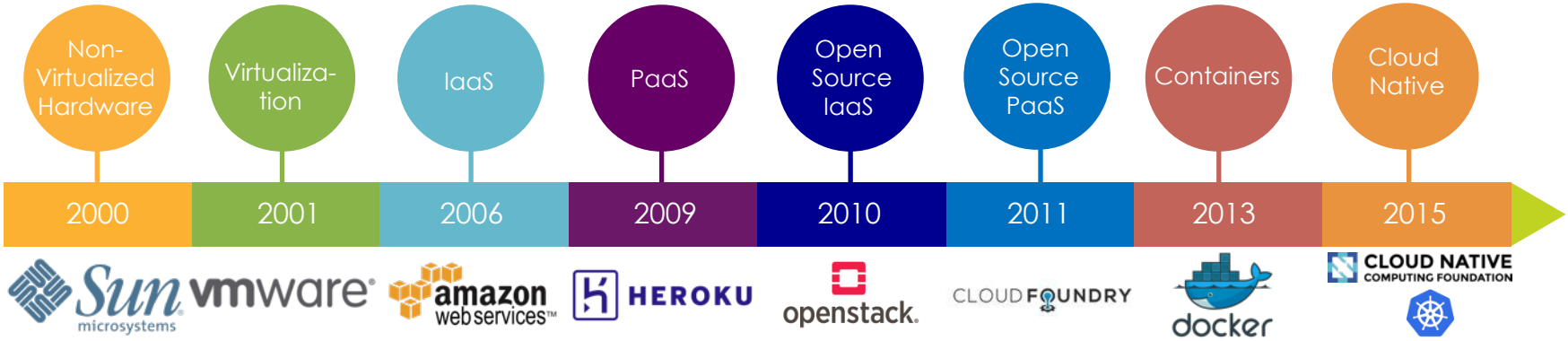


Cloud Native: CNCF (2015)

CLOUD NATIVE COMPUTING FOUNDATION

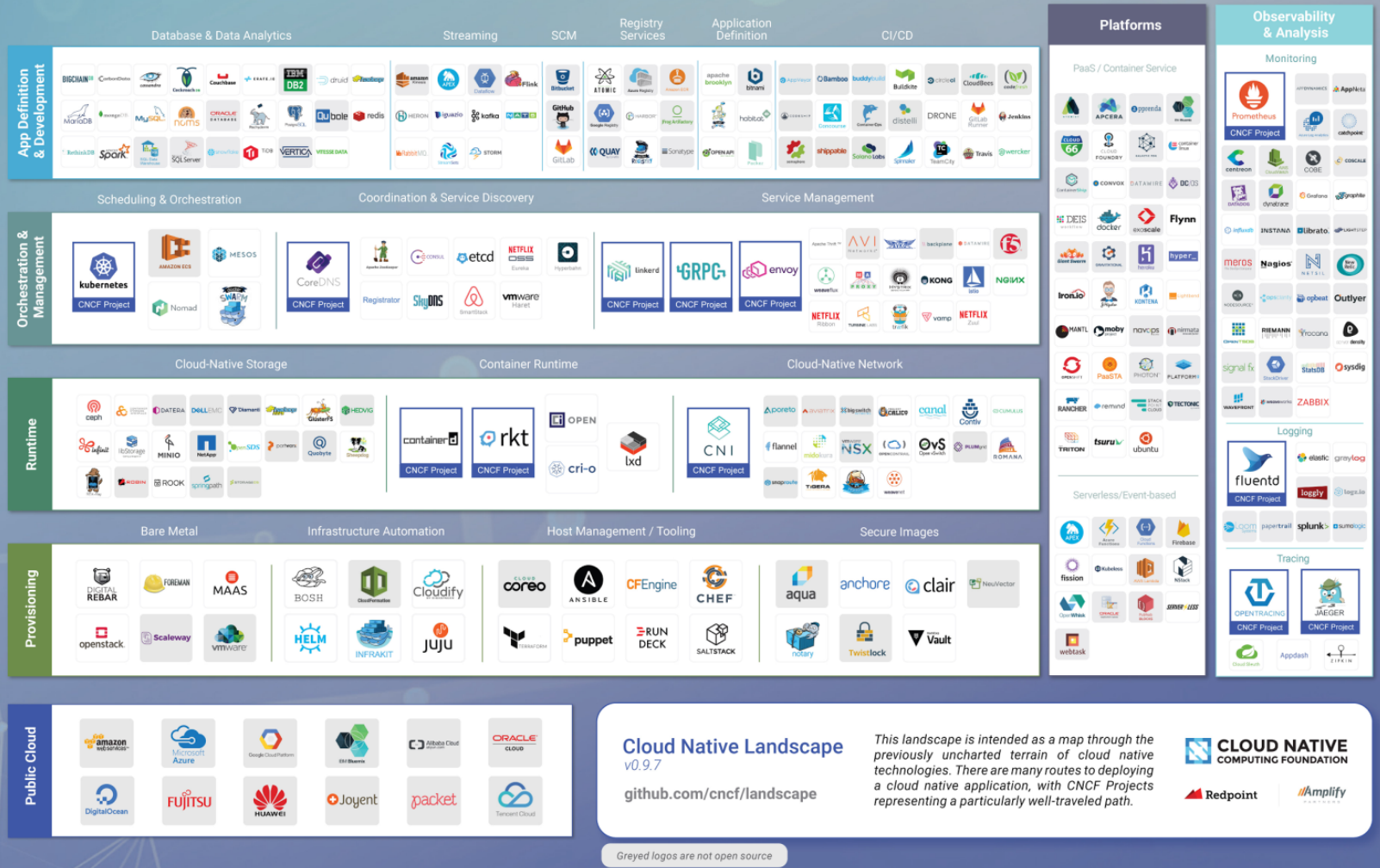
kubernetes

- Cloud native computing uses an open source software stack to:
  - segment applications into *microservices*,
  - packaging each part into its own *container*
  - and dynamically *orchestrating* those containers to optimize resource utilization



What Have We Learned?

- Core Building Block:
  - Servers ➡ Virtual Machines ➡ Buildpacks ➡ Containers
- Isolation Units
  - From heavier to lighter weight, in spin-up time and size
- Immutability
  - From pets to cattle
- Provider
  - From closed source, single vendor to open source, cross-vendor


App Definition & Development
Database & Data Analytics
Streaming
SCM
Registry Services
Application Definition
CI/CD
Orchestration & Management
Scheduling & Orchestration
Coordination & Service Discovery
Service Management
kubernetes
CNCF Project
CoreDNS
CNCF Project
linkerd
CNCF Project
GRPC
CNCF Project
envoy
CNCF Project
Runtime
Cloud-Native Storage
Container Runtime
Cloud-Native Network
container
CNCF Project
rkt
CNCF Project
CNI
CNCF Project
Provisioning
Bare Metal
Infrastructure Automation
Host Management / Tooling
Secure Images
Public Cloud
Platforms
PaaS / Container Service
Serverless/Event-based
Observability & Analysis
Monitoring
Prometheus
CNCF Project
Logging
fluentd
CNCF Project
Tracing
OPENTRACING
CNCF Project
JAEGER
CNCF Project
Cloud Native Landscape
v0.9.7
github.com/cncf/landscape
This landscape is intended as a map through the previously uncharted terrain of cloud native technologies. There are many routes to deploying a cloud native application, with CNCF Projects representing a particularly well-traveled path.
CLOUD NATIVE COMPUTING FOUNDATION
Redpoint
Amplify
Greyed logos are not open source

Avoid Vendor Lock-in

Open source software stack enables deployment on any public, private cloud or hybrid cloud

Enable Unlimited Scalability

Scales from several nodes on your laptop to tens of thousands of self-healing multi-tenant nodes

Increase Agility and Maintainability

By splitting applications into microservices
with explicitly described dependencies

Improve Efficiency and Resource Utilization

Via a central orchestrating process that dynamically manages and schedules microservices

Achieve Resiliency

To failures of individual containers, machines, and even data centers and to varying levels of demand

Cloud Native Enables High Performance

High-performing teams deploy more frequently and have much faster lead times.

They make changes with fewer failures, and recover faster from failures.

200x more frequent deployments

2,555x shorter lead times

3x lower change failure rate

24x faster recovery from failures

2016 State of DevOps Report Infographic from Puppet

Greenfield Applications

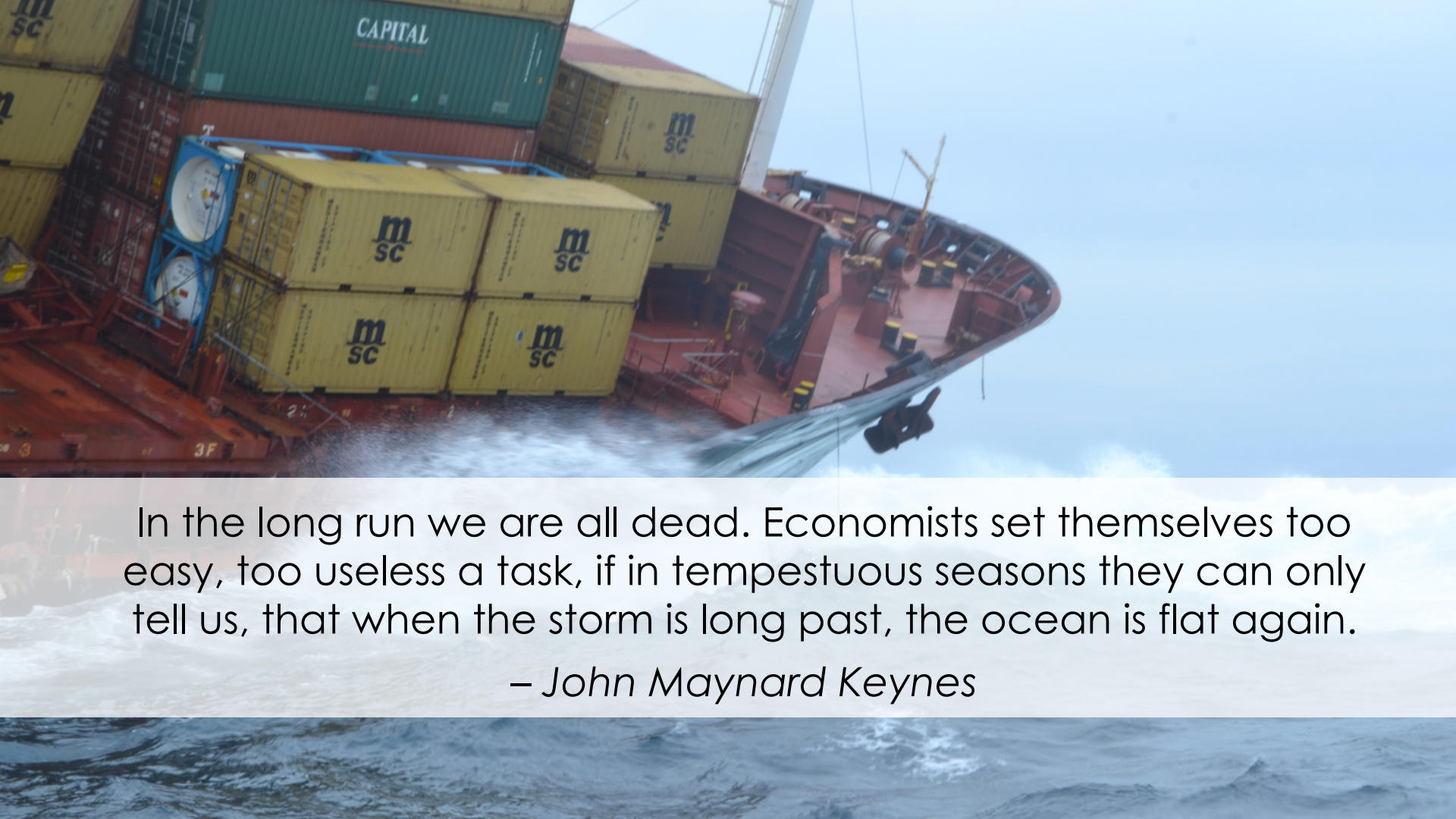
Cloud Native application architectures are the default choice for greenfield applications

Greenfield Application Design

- The leading choice for cloud native orchestration is:

kubernetes

  - Selected by top enterprises and startups like Bloomberg, Box, Reddit and the New York Times
  - One of the highest development velocity projects in the history of open source
  - Amazing group of technology giants and startups cooperating to rapidly improve

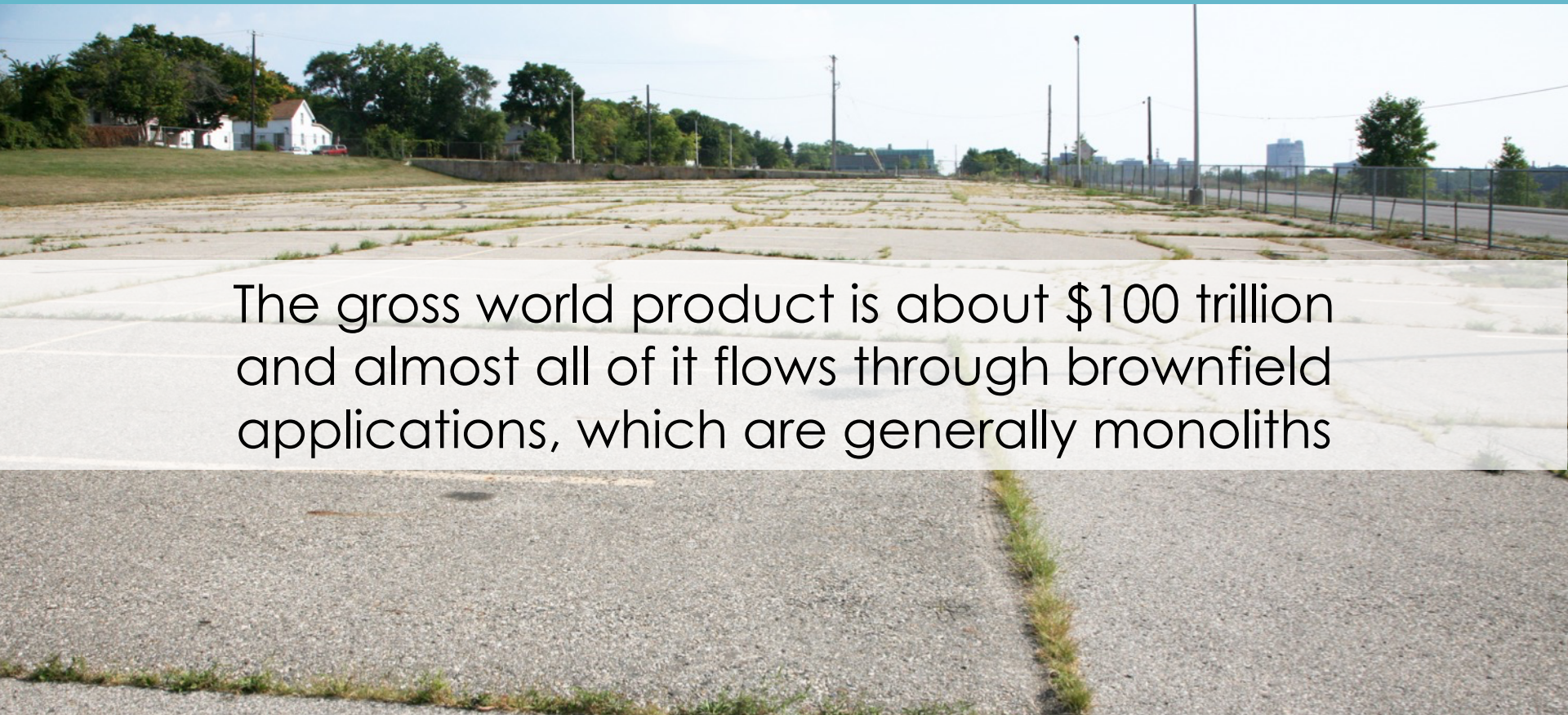


In the long run we are all dead. Economists set themselves too easy, too useless a task, if in tempestuous seasons they can only tell us, that when the storm is long past, the ocean is flat again.

– *John Maynard Keynes*

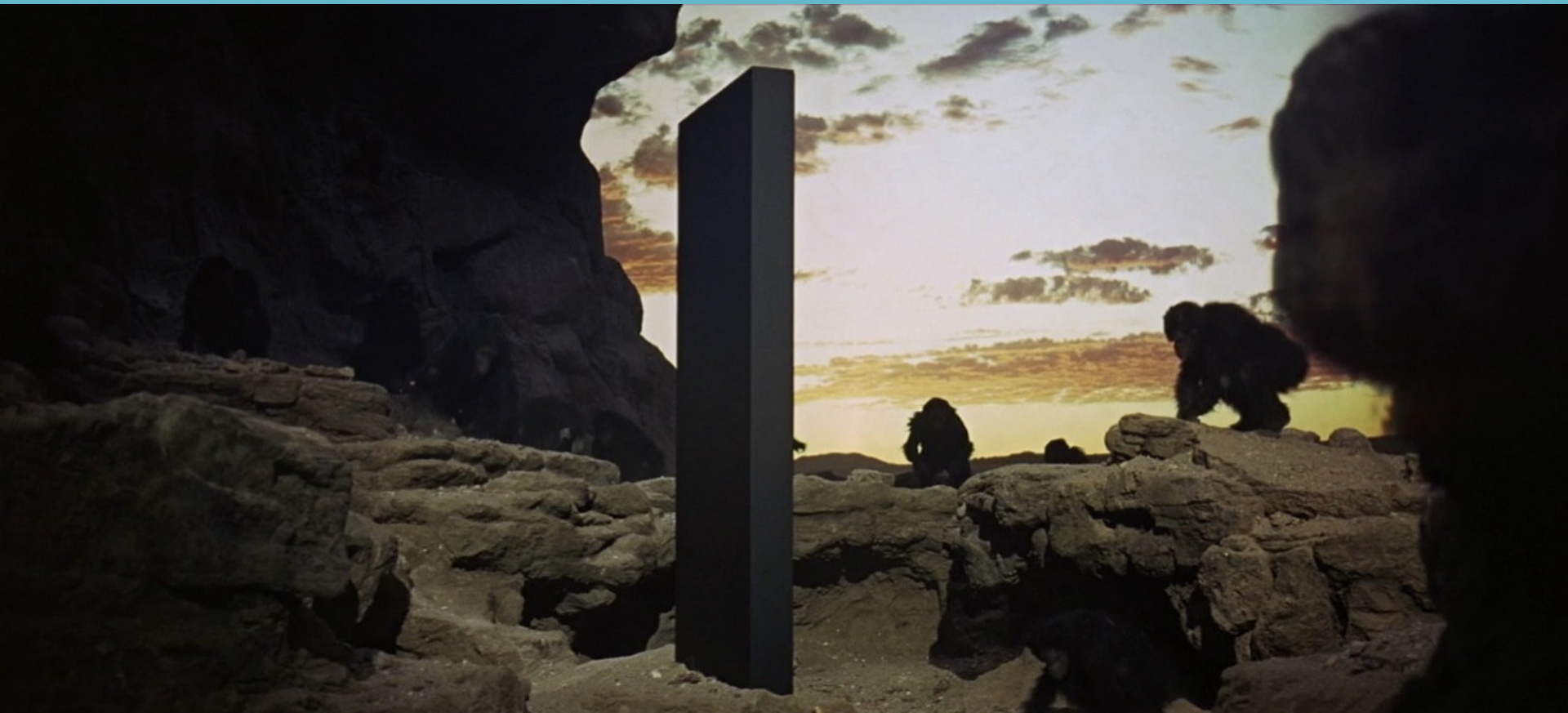
…too easy, too useless a task…

The Real World Consists of Brownfield Applications



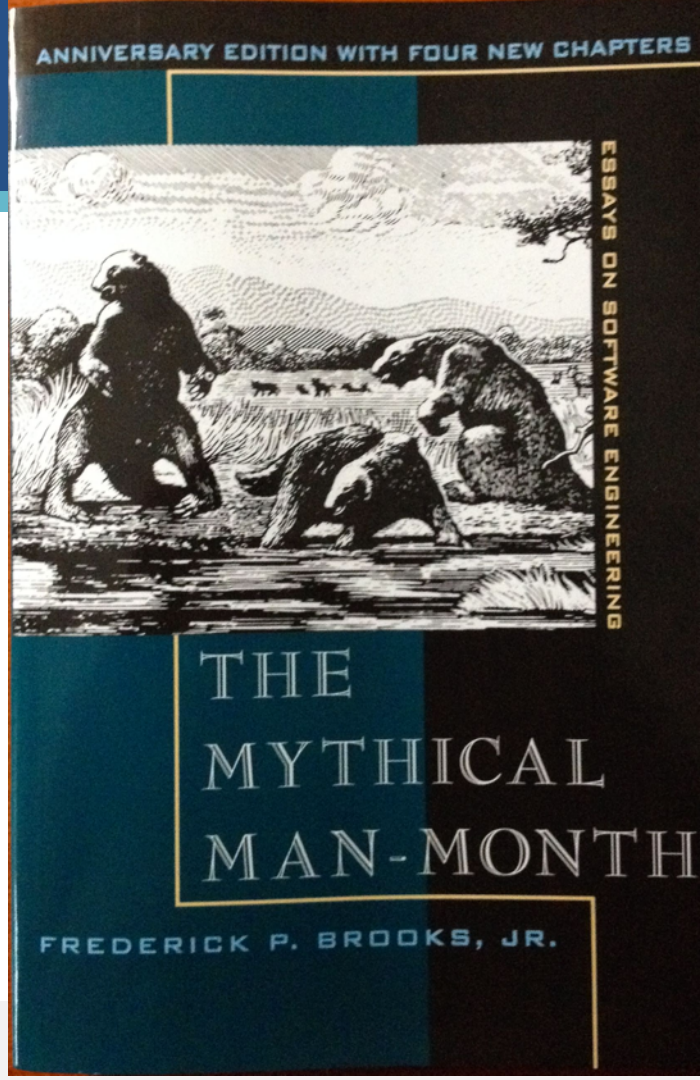
The gross world product is about $100 trillion and almost all of it flows through brownfield applications, which are generally monoliths

Nearly All Production Applications Are Monoliths



How About a Rewrite?

- The Second System Syndrome says that many rewrites end in failure
- The existing system evolves faster than the second one can catch up

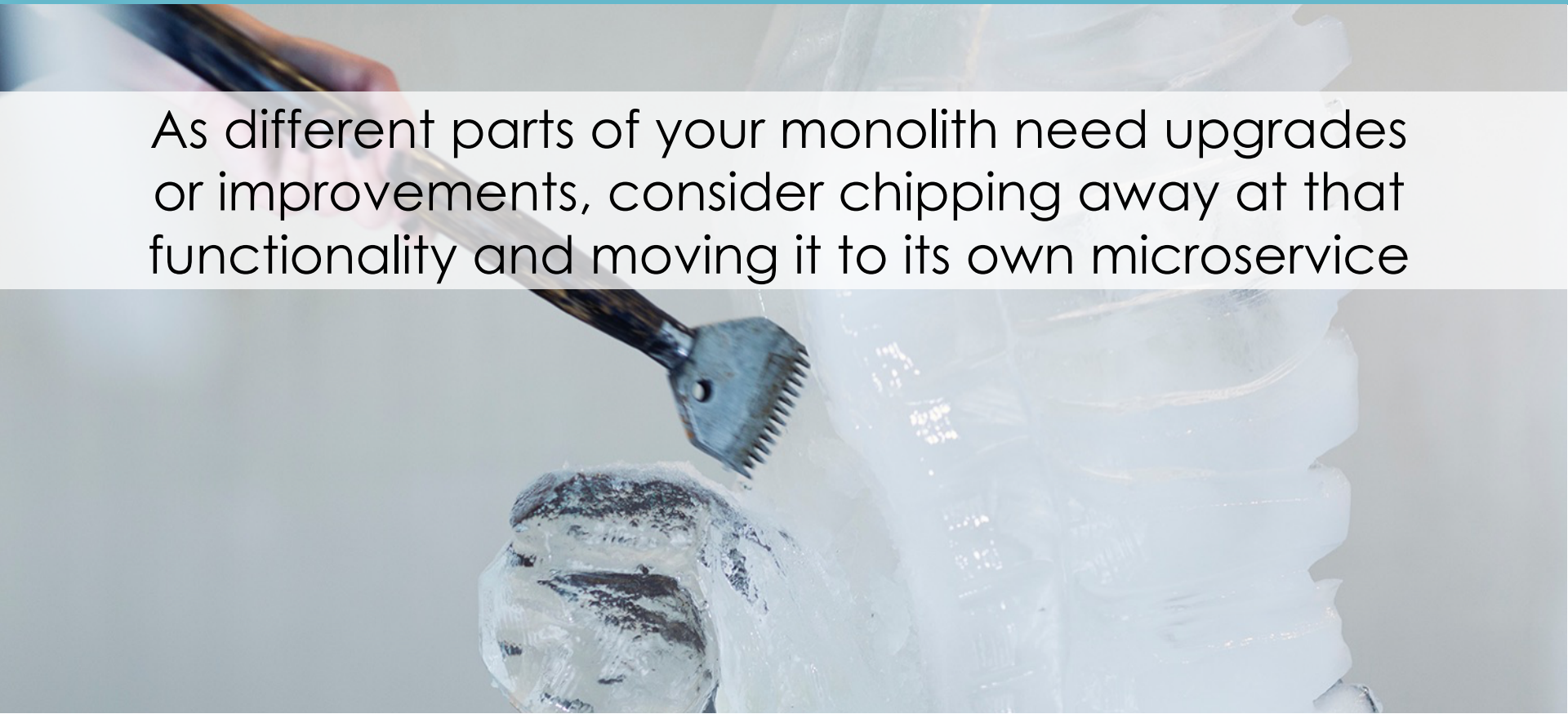


Step One Is to Stop Digging

Step Two is Lift-and-Shift Your Monolith

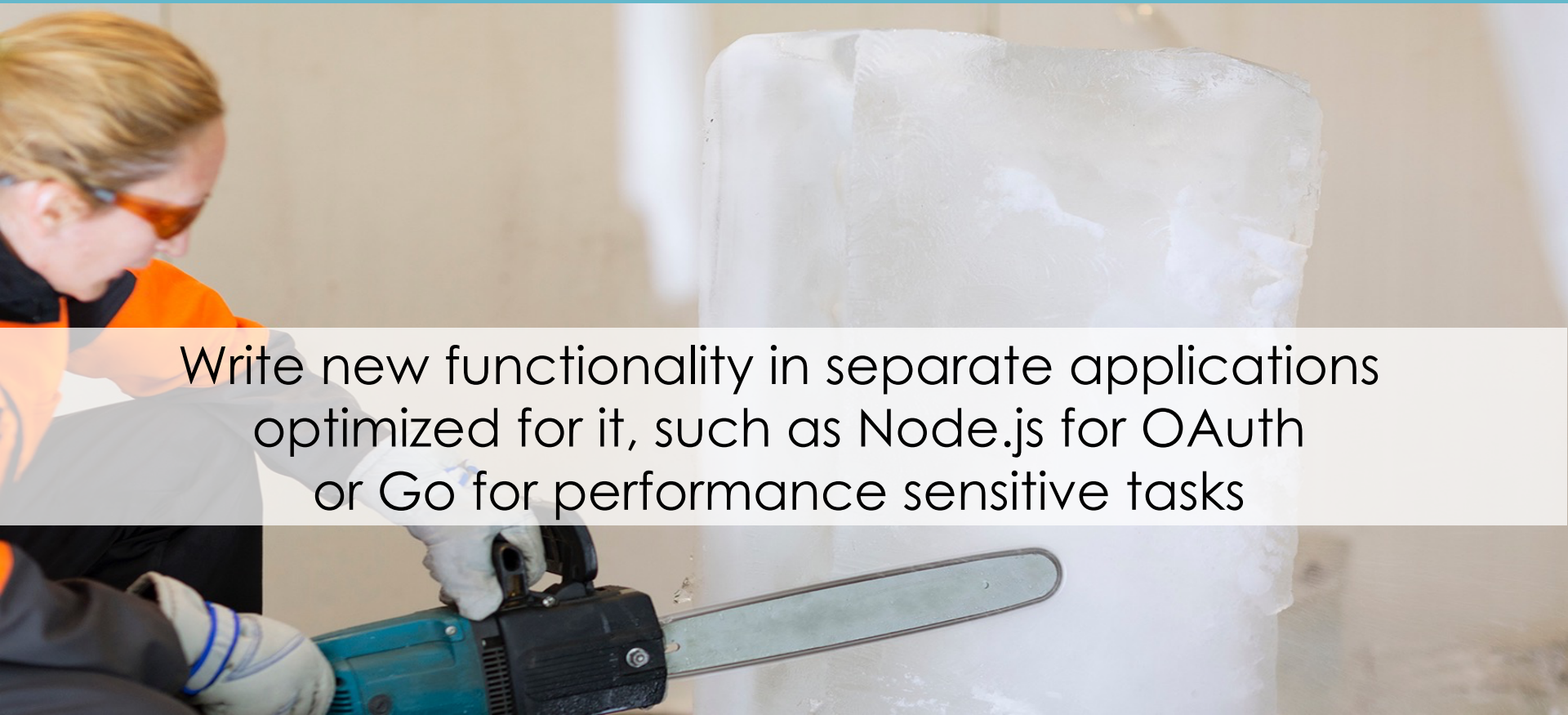
StatefulSets are helpful for providing stable, persistent storage and unique network identifiers to your monolith

Chip Away at the Monolith



As different parts of your monolith need upgrades or improvements, consider chipping away at that functionality and moving it to its own microservice

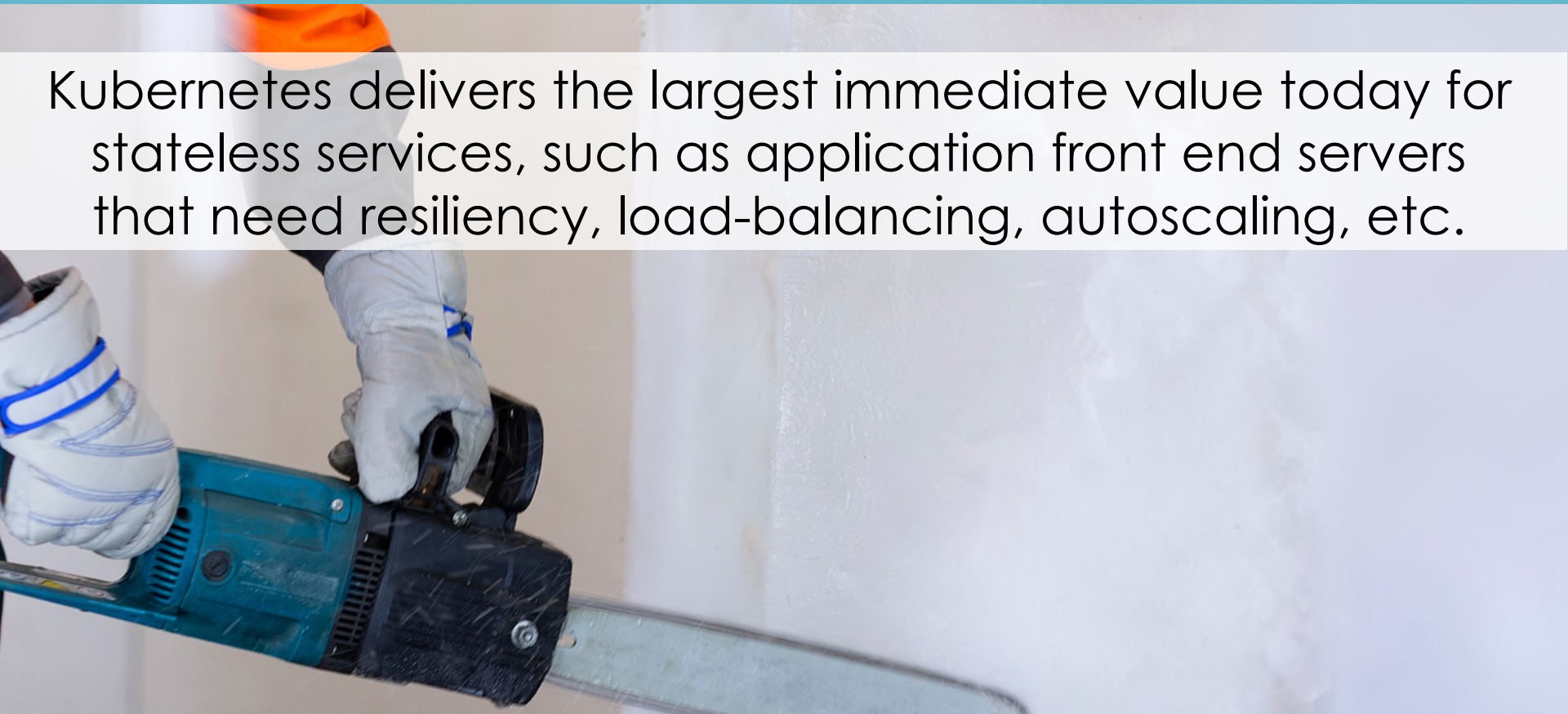
New Functionality Can Go in New Apps



Write new functionality in separate applications optimized for it, such as Node.js for OAuth or Go for performance sensitive tasks

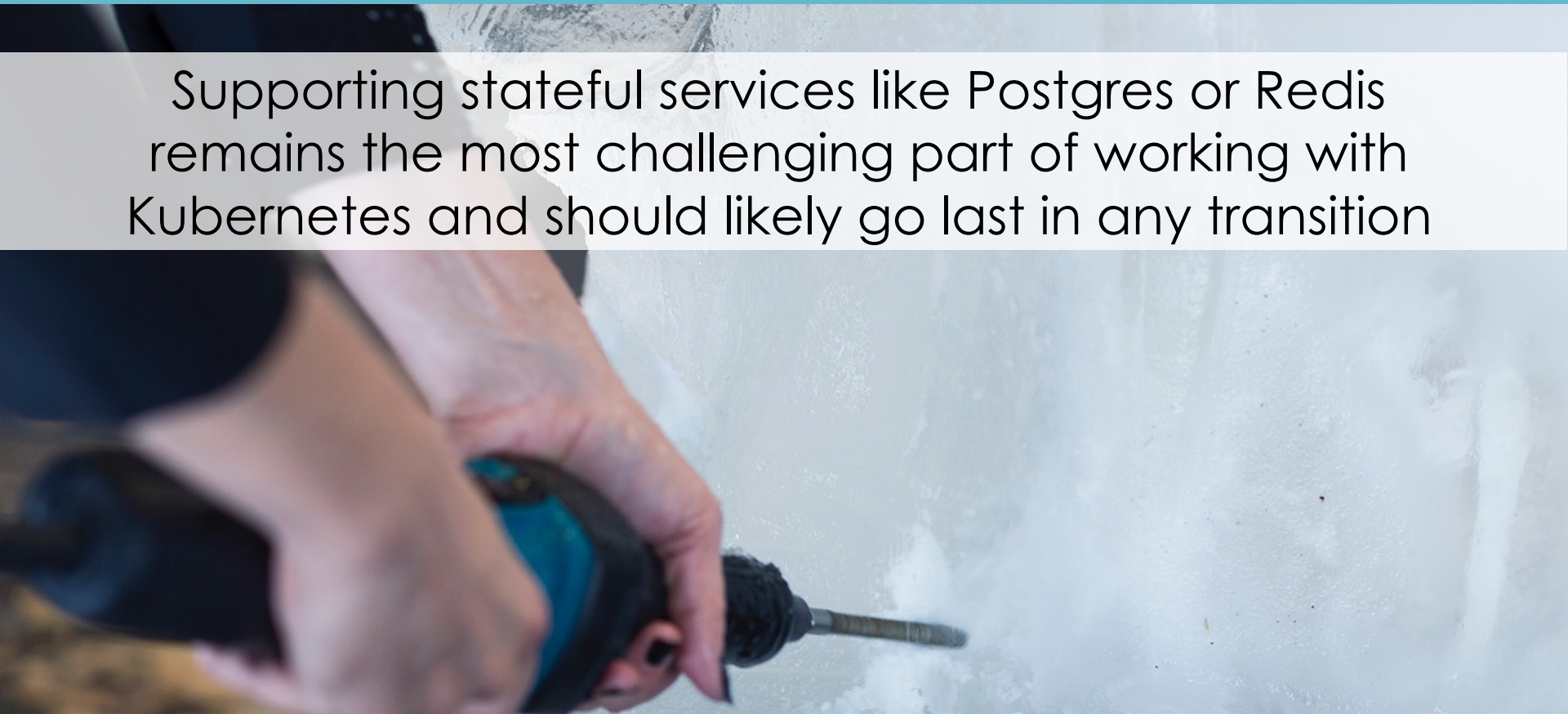
Start with Stateless Services

Kubernetes delivers the largest immediate value today for stateless services, such as application front end servers that need resiliency, load-balancing, autoscaling, etc.



Transition Your Data Stores Last

Supporting stateful services like Postgres or Redis remains the most challenging part of working with Kubernetes and should likely go last in any transition



Monoliths Are the Antithesis of Cloud Native

Inflexible, tightly-coupled, brittle

Consider Complementary Projects

Consider the constellation of CNCF projects
(and their competitors) to provide related services

Prometheus OpenTracing Fluentd Linkerd gRPC CoreDNS containerd rkt CNI Envoy Jaeger

Evolving Your Monolith into a Microservice

And Eventually You Have a Collection

Kubernetes and Greenfield

Avoid the “soft bigotry of low expectations” of thinking that you need to do a greenfield rewrite to get the benefits of cloud native

Kubernetes and Brownfield

kubernetes ❤ brownfield applications

More Info on Evolving Monoliths

- Box
  - https://architecht.io/box-co-founder-on-moving-to-microservices-and-the-promise-of-kubernetes-a49f01b1c0c0
- Key Bank
  - https://developers.redhat.com/blog/2016/10/27/the-fast-moving-monolith-how-we-sped-up-delivery-from-every-three-months-to-every-week/
- TicketMaster
  - https://www.linux.com/news/learn/kubernetes/ticketmaster-chooses-kubernetes-stay-ahead-competition
- Wikipedia
  - https://kubernetes.io/case-studies/wikimedia/
- GitHub
  - https://githubengineering.com/kubernetes-at-github/

KubeCon + CloudNativeCon

- KubeCon + CloudNativeCon North America 2017
  - Austin, Texas
  - December 6 - 8, 2017
  - Registration & Sponsorships now open
- KubeCon + CloudNativeCon Europe 2018
  - Copenhagen, Denmark
  - May 2 - 4, 2018
  - More info

Follow-Up

Dan Kohn

cncf.io • dan@linuxfoundation.org • @dankohn1

This presentation is available at:
https://github.com/cncf/presentations