

ORCHESTRATE YOUR SERVICES ON OPENSHIFT USING SPRING CLOUD KUBERNETES

Capgemini Craftsmanship Community #83
Wrocław, 2nd of June 2017

Krzysztof Sobkowiak (@ksobkowiak)

The Apache Software Foundation Member
Senior Solution Architect at Capgemini

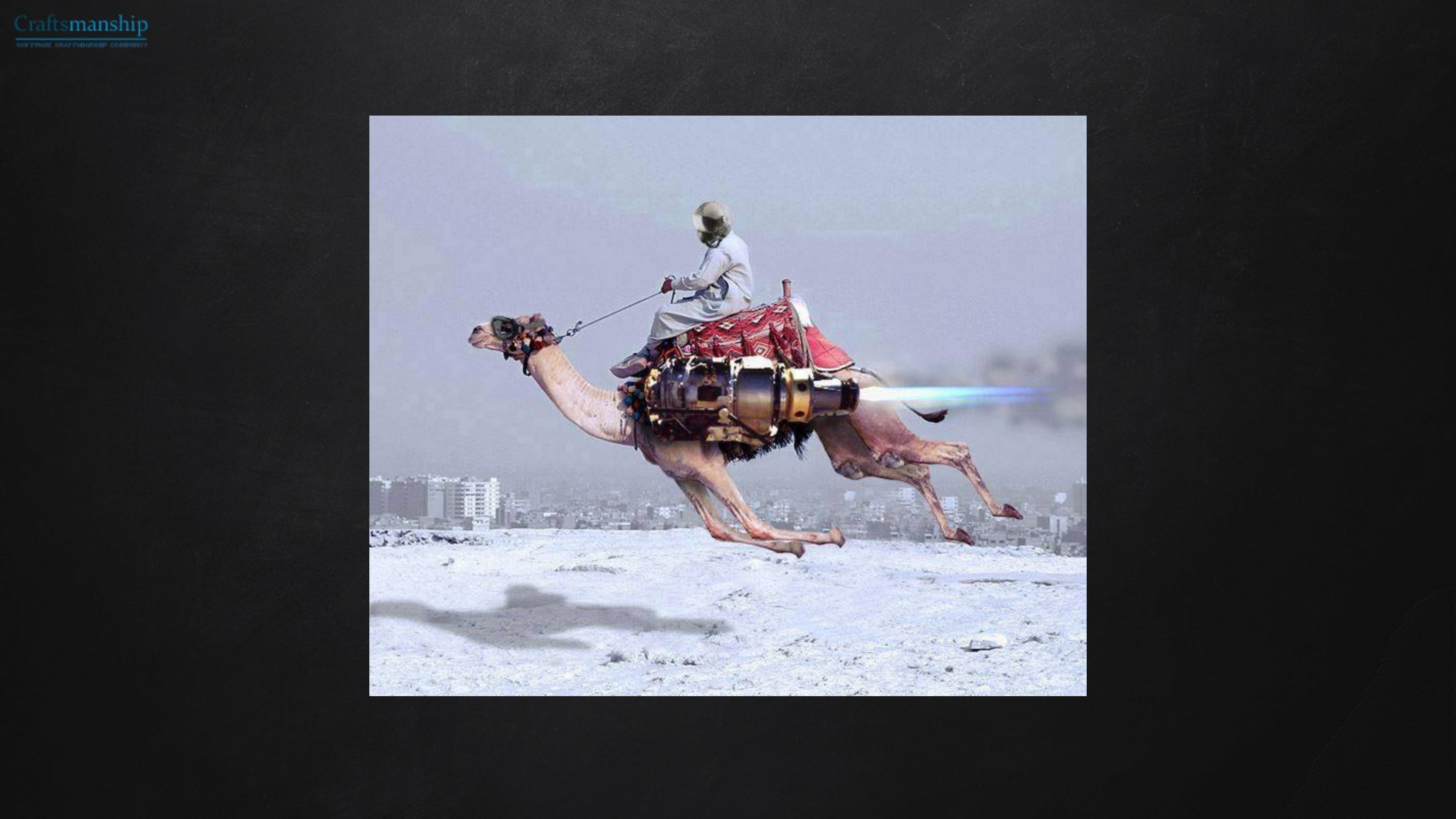
YOUR SPEAKER

- FOSS enthusiast, evangelist & architect
- Senior Solution Architect at Capgemini
- The Apache Software Foundation
 - Member
 - Apache ServiceMix committer & PMC chair (V.P. Apache ServiceMix)
 - active at Apache Karaf, CXF, Camel, ActiveMQ
- Member/developer at OASP, OPS4J





Views in this presentation are my personal views and do not necessarily reflect the views of Capgemini.



Creating business value through software is about speed, safety, iteration, and continuous improvement



- Simple configuration
- Curated dependencies and transitive dependencies
- Built in metrics, monitoring
- Slim profile for deployment (...micro even?)

#microprofile

Writing a single service is nice...

...but no microservice is an island

CHALLENGES OF DISTRIBUTED SYSTEMS

- Configuration management
- Service registration & discovery
- Routing & balancing
- Fault tolerance (Circuit Breakers!)
- Monitoring

NETFLIX | OSS

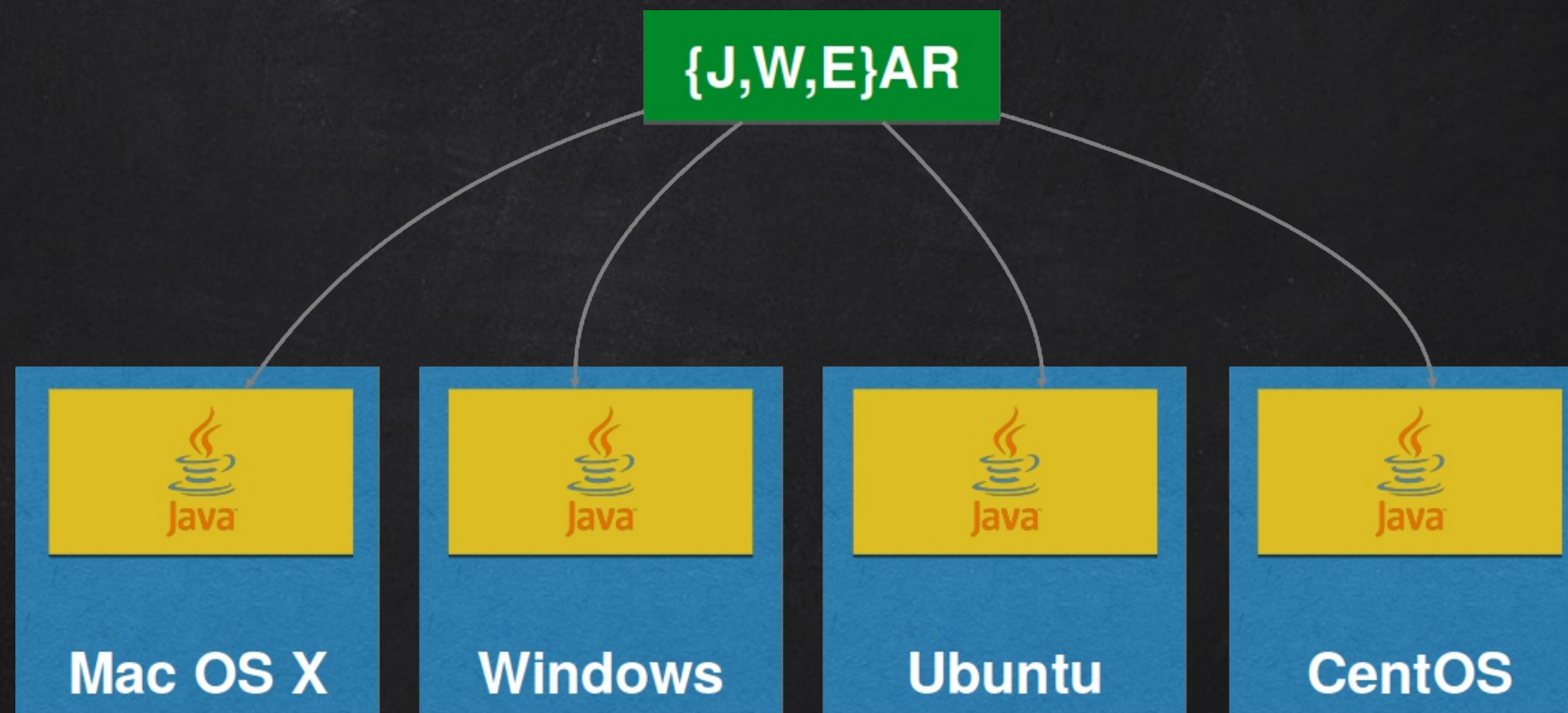


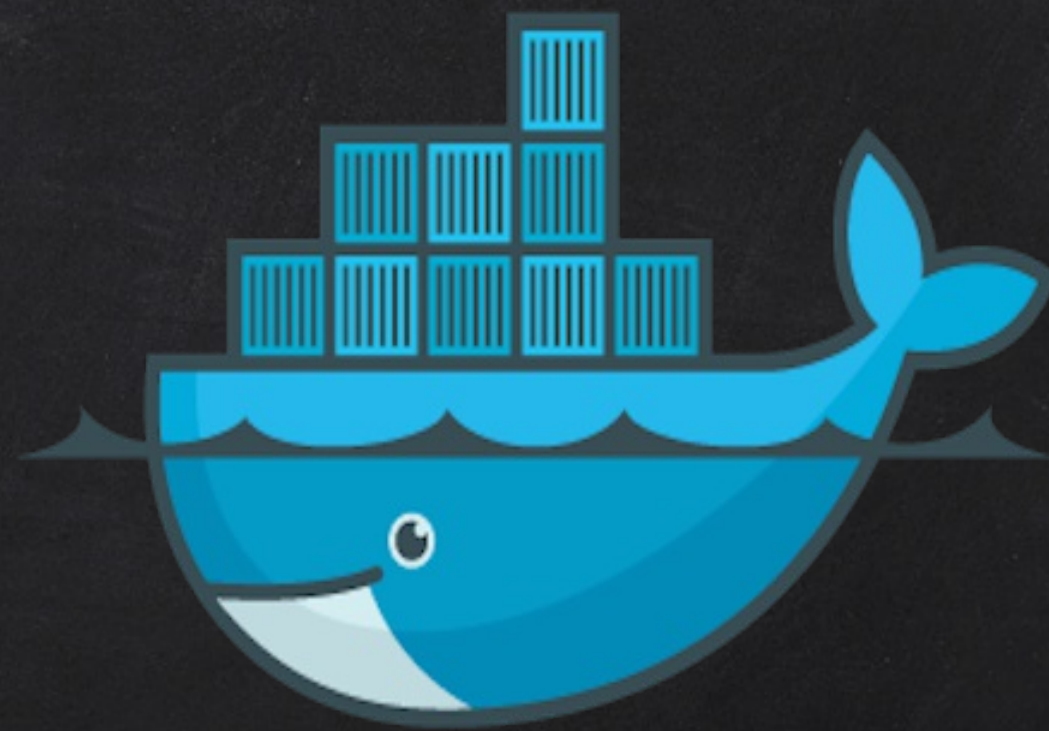
- Distributed configuration
- Service Discovery
- Loadbalancing
- Bulkheading
- Circuit Breaker
- Fallback
- Versioning/Routing
- Based on AWS

TYPICAL PROBLEMS DEVELOPING MICROSERVICES

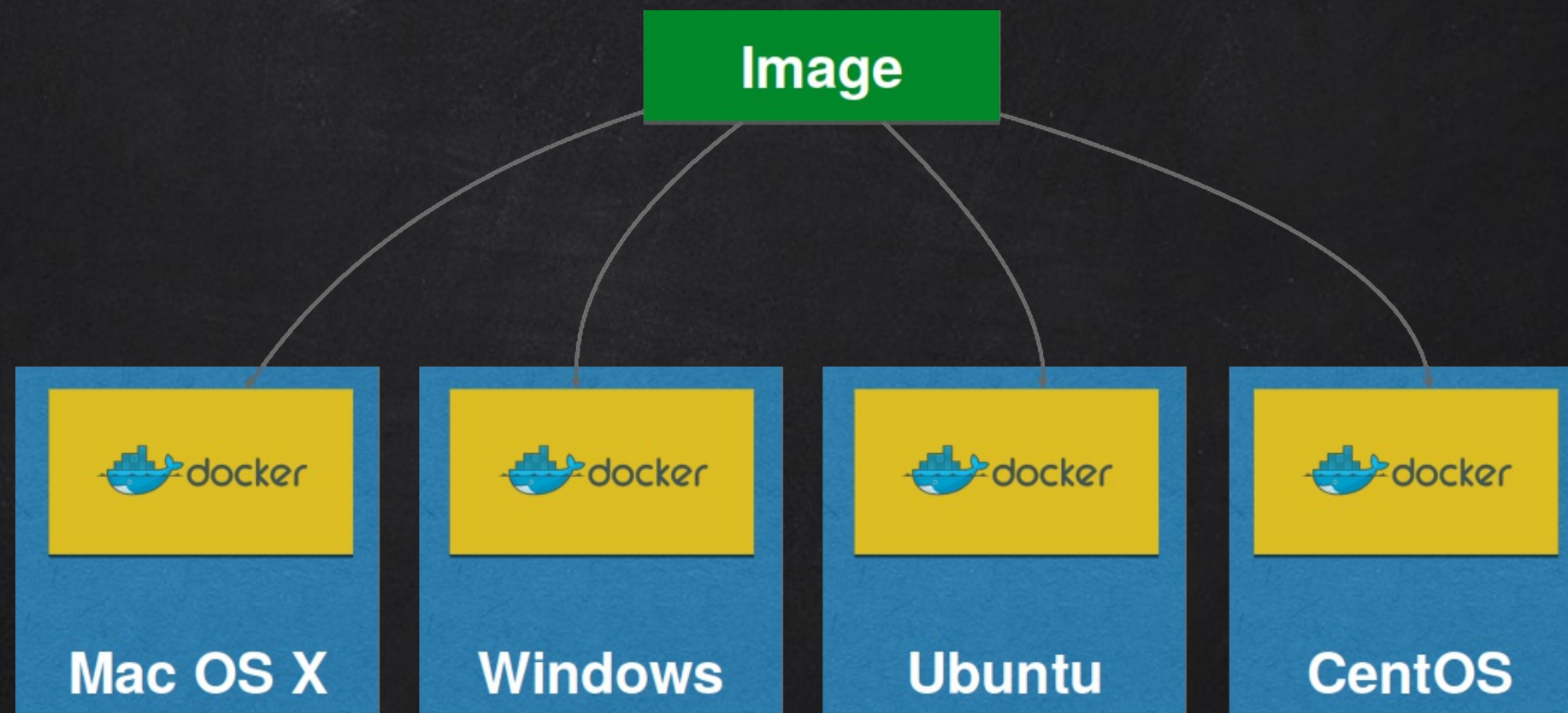
- How to run them all locally?
- How to package them (dependency management)
- How to test?
- Vagrant? VirtualBox? VMs?
- Specify configuration
- Process isolation
- Service discovery
- Multiple versions?

WORA = WRITE ONCE RUN ANYWHERE





PODA = PACKAGE ONCE DEPLOY ANYWHERE




The Docker Mission



Any Application

Anywhere

 **Local**

 **Cloud**

 **Data Center**



DEVOPS CHALLENGES FOR MULTIPLE CONTAINERS

- How to scale?
- How to avoid port conflicts?
- How to manage them in multiple hosts?
- What happens if a host has a trouble?
- How to keep them running?
- How to update them?
- Where are my containers?

WHAT ABOUT NON-JAVA?



WHAT IF YOU COULD DO ALL OF THIS RIGHT NOW WITH AN OPEN-SOURCE PLATFORM?

- 100% open source, ASL 2.0
- Technology agnostic (java, nodejs, python, golang, etc)
- Built upon decades of industry practices
- 1-click automation
- Cloud native (on premise, public cloud, hybrid)
- Complex build/deploy pipelines (human workflows, approvals, chatops, etc)
- Comprehensive integration inside/outside the platform

MEET KUBERNETES

Greek for Helmsman; also the root of the word Governor (from latin: gubernator)

- Container orchestration platform
- Supports multiple cloud and bare-metal environments
- Inspired by Google's experience with containers
 - Rewrite of Google's internal framework Borg
- Open source, written in Go
- Manage applications, not machines



KUBERNETES



- Distributed configuration
- Service Discovery
- Load-balancing
- Versioning/Routing
- Deployments
- Scaling/Autoscaling
- Liveness/Health checking
- Self healing

KUBERNETES CONCEPTS

Pods

colocated group of containers that share an IP, namespace, storage volume, resources, lifecycle

Replica Set

manages the lifecycle of pods and ensures specified number are running (next gen Replication Controller)

Service

Single, stable name for a set of pods, also acts as LB

Label

used to organize and select group of objects

KUBERNETES CONCEPTS

Node

Machine (physical or VM) in the cluster

Master

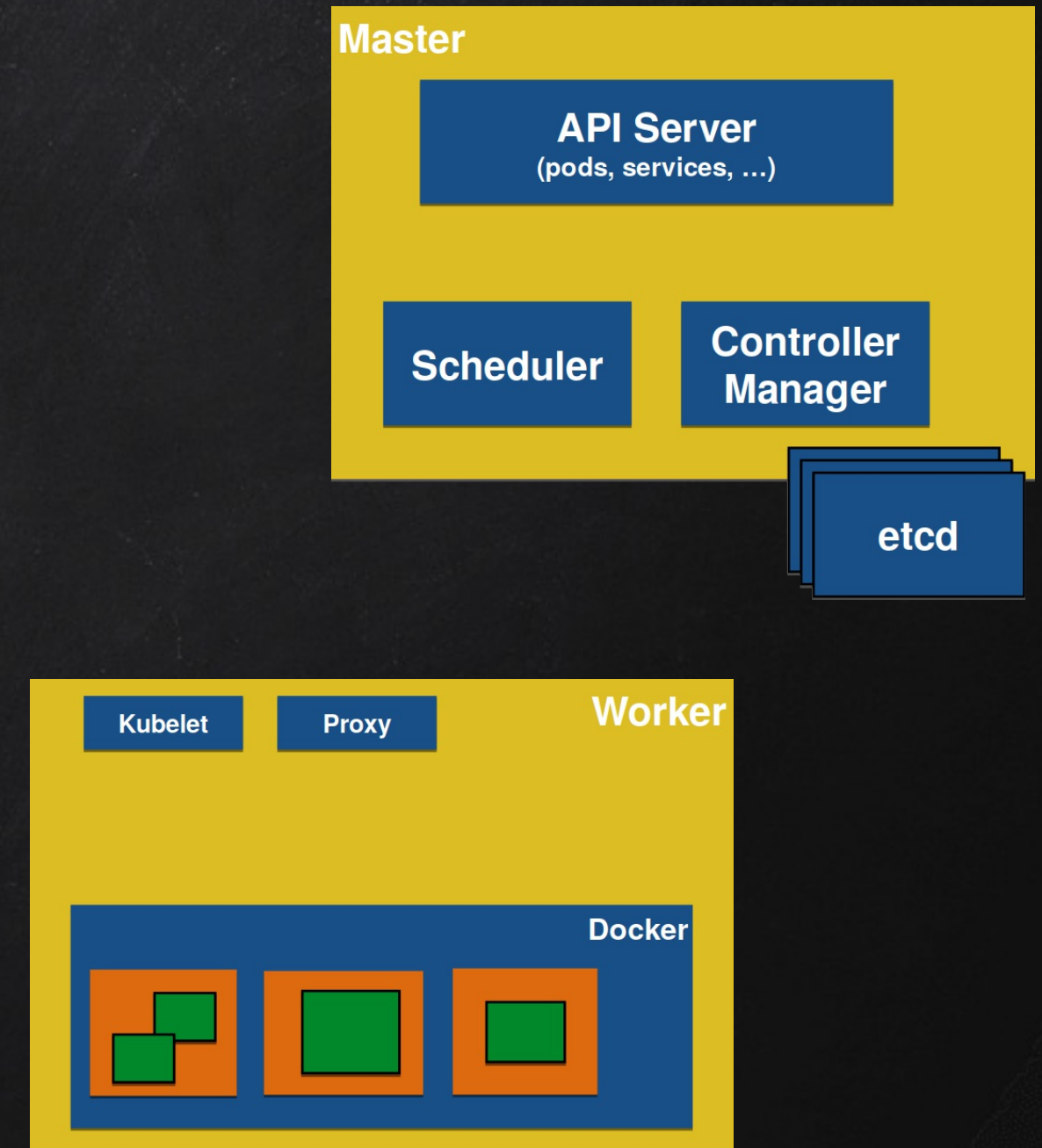
Central control plane, provides unified view of the cluster

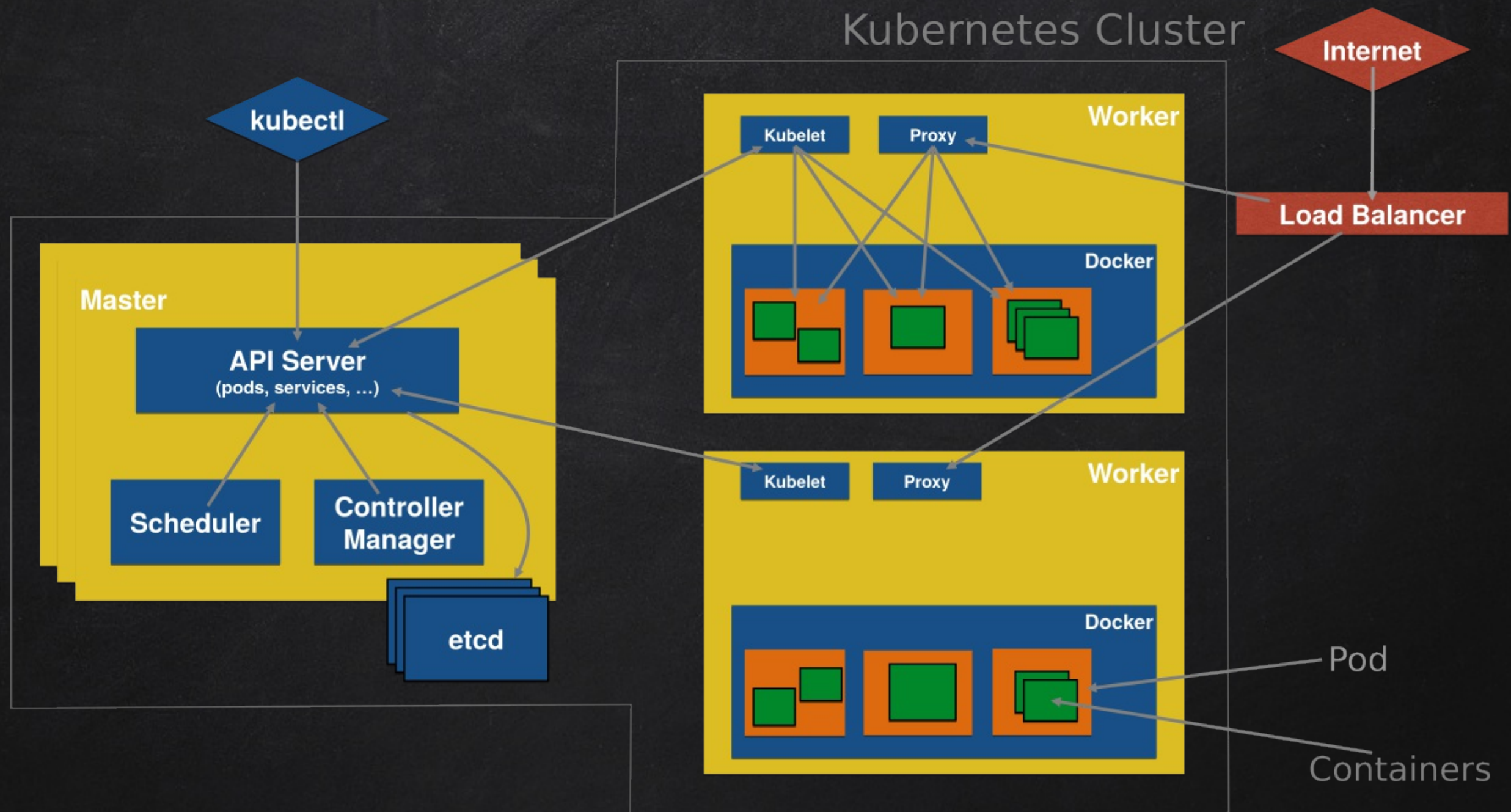
- etcd: distributed key-value store used to persist Kubernetes system state

Worker

Docker host running kubelet (node agent) and proxy services

- Runs pods and containers





OPENSIFT IS KUBERNETES

- Team self service application deployment
- Developer focused workflow
- Enterprise ready (LDAP, RBAC, Oauth, etc)
- Higher level abstraction above containers for delivering technology and business value
- Integrated Docker registry
- Jenkins Pipeline out of the box
- Build/deployment triggers
- Software Defined Networking (SDN)
- Docker native format/packaging
- CLI/IDE/Web based tooling



OPENSIFT



WHAT ABOUT CLIENT-SIDE LOAD BALANCING?

Spring Cloud Kubernetes:

- DiscoveryClient
- Ribbon integration
- Actuator/Health integrations
- Hystrix/Turbine Dashboard integrations (kubeflix)
- Zipkin Tracking
- Configuration via ConfigMaps



fabric8

FABRIC8 ALL THE THINGS!

- Built on top of Kubernetes
- Wizards to create microservices
- Package as immutable containers
- Rolling upgrade across environments
- 1-Click install of fully configured CI/CD (Jenkins Pipeline, Nexus, Git)
- Feedback loops
- Lots of developer tooling
- ChatOps
- iPaaS/Integration
- Chaos Monkey





THANKS!

Any questions?

You can find me at
@ksobkowiak
krzysztof.sobkowiak@capgemini.com
<http://krzysztof-sobkowiak.net>



This work is licensed under a Creative Commons Attribution 4.0 International License.