Kubernetes Meetup Sofia

Title:

How We Injected the Kubernetes Virus Into Our Corporation Without Activating the Immune System! :-J

Abstract:

With a brief overview of SAP’s highly successful client/server architecture, we outline the evolution of the SAP’s core architecture driven by enterprise customer needs and the ensuing innovators dilemma that SAP - as a decidedly mature business with on-premise focus - faced with the “cloud”. We discuss with the factors that led to the adoption of Kubernetes as gateway technology to a globally distributed system. We will discover an architectural design pattern for scale-out hidden in plain sight (!) that is employed also in Project Gardener. Lastly, we conclude how under the patronage of cloud-native technologies, Gardener is making inroads into commercial, productive services at SAP.

How we injected the Kubernetes virus into our corporation without activating the immune system!

:-J

Vasu Chandrasekhara
Chief Architect, SAP
CNCF GB Rep for SAP

PUBLIC

CLOUD NATIVE COMPUTING FOUNDATION
Platform Engineering
Meetup
November

SAP

➔ The pun in the title insinuates a couple of humorous narratives.

- Isn’t SAP famous for its monolothic architecture?

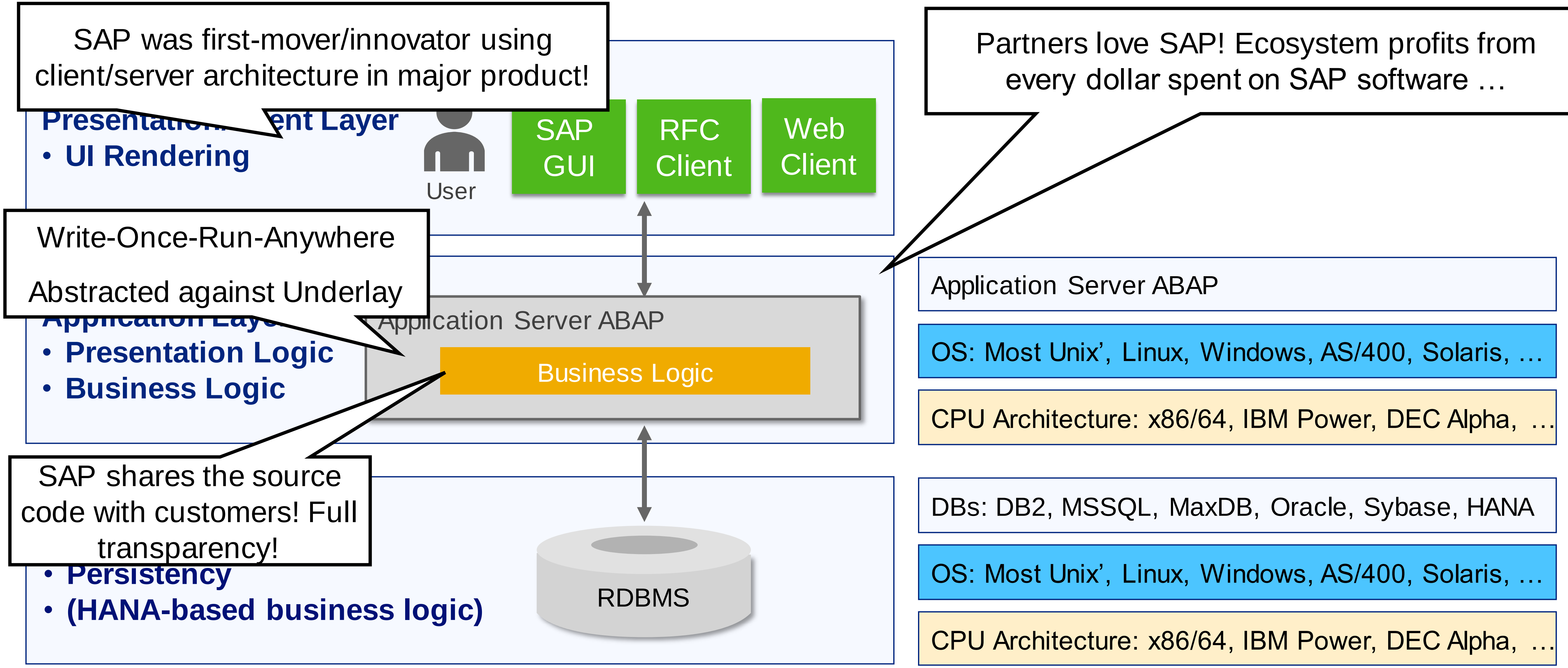
➔ What was there before? And what made the “old” so incredibly successful?

- Why corpororate immune system?
- Innovators Dilemma?
- Why?
- What were the difficulties?

SAP’s Classic R/3 Architecture

PUBLIC

R/3 Three-Tier Client/Server Architecture = ABAP Platform



SAP was first-mover/innovator using client/server architecture in major product!

Presentatio ent Layer

- UI Rendering

Partners love SAP! Ecosystem profits from every dollar spent on SAP software …

Write-Once-Run-Anywhere

Abstracted against Underlay

Applicatio Laye

- Presentation Logic
- Business Logic

Application Server ABAP

OS: Most Unix’, Linux, Windows, AS/400, Solaris, …

CPU Architecture: x86/64, IBM Power, DEC Alpha, …

SAP shares the source code with customers! Full transparency!

- Persistency
- (HANA-based business logic)

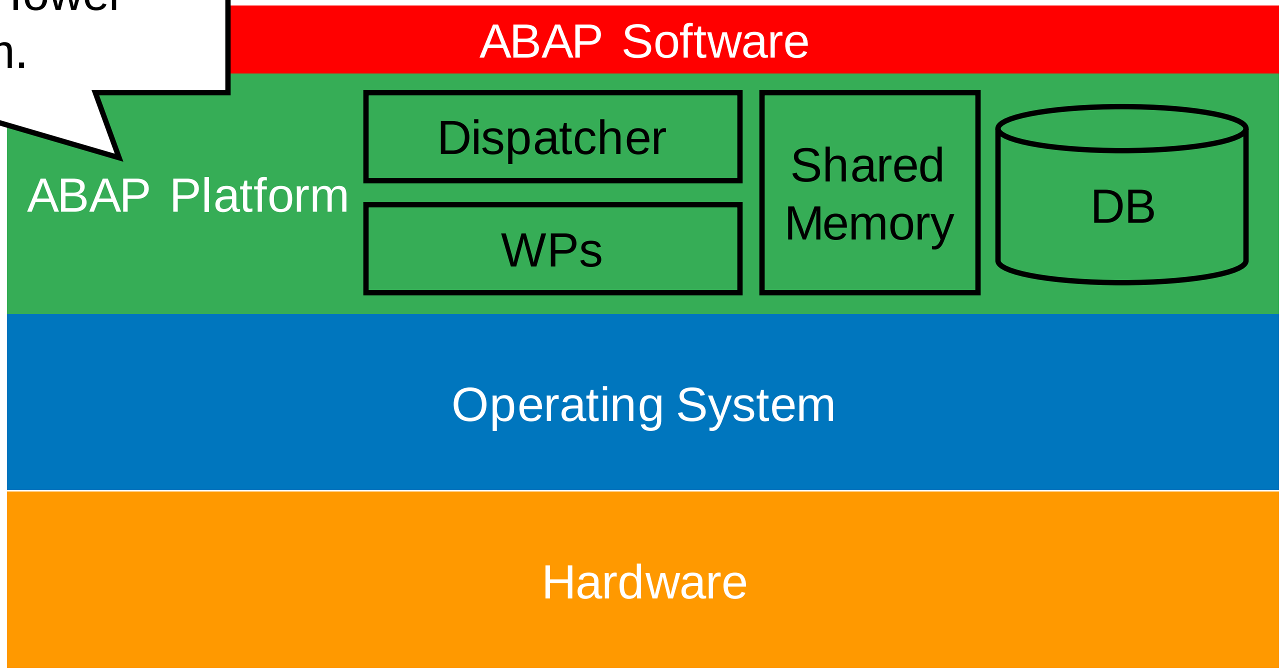
DBs: DB2, MSSQL, MaxDB, Oracle, Sybase, HANA

OS: Most Unix’, Linux, Windows, AS/400, Solaris, …

CPU Architecture: x86/64, IBM Power, DEC Alpha, …

The ABAP Platform “Monolith”

ABAP Platform “Monolith” is an Operating System in its own right, adjusted to run on the lower Operating System.



An Operating System is the body of software that

- abstracts the hardware platform,
- protects software principals from each other,
- multiplexes machine’s hardware resources.

The ABAP Platform is core to the value stream.

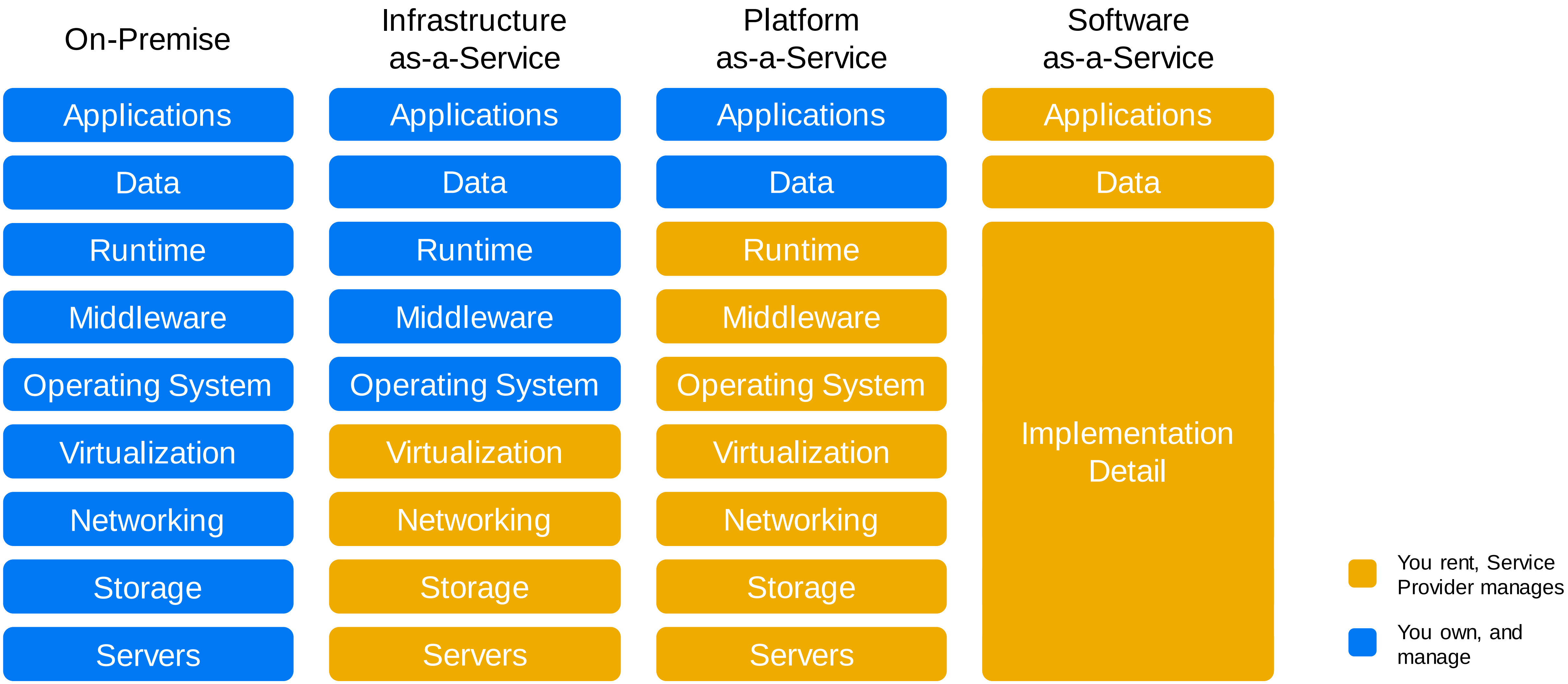
The “corporate immune system” is optimized to protect this value stream!

Want to introduce something completely / fundamentally new? No breaking change!

“Change is the law of life. And those who look only to the past or present are certain to miss the future.”

John F. Kennedy

From On-Premise to the Cloud

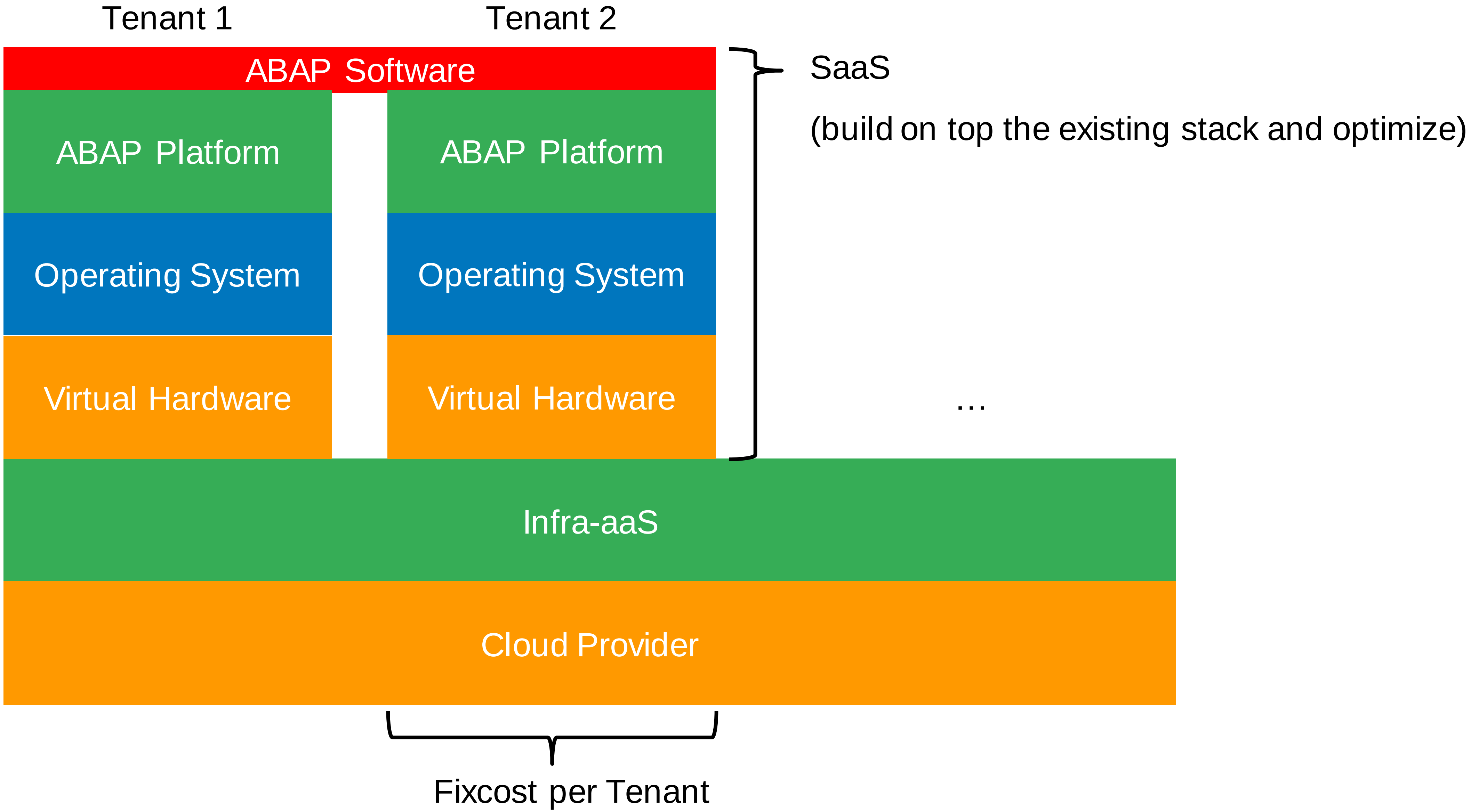


On-Premise	Infrastructure as-a-Service	Platform as-a-Service	Software as-a-Service
🟦 Applications	🟦 Applications	🟦 Applications	🟨 Applications
🟦 Data	🟦 Data	🟦 Data	🟨 Data
🟦 Runtime	🟦 Runtime	🟨 Runtime	🟨 Implementation Detail
🟦 Middleware	🟦 Middleware	🟨 Middleware	
🟦 Operating System	🟦 Operating System	🟨 Operating System	
🟦 Virtualization	🟨 Virtualization	🟨 Virtualization	
🟦 Networking	🟨 Networking	🟨 Networking	
🟦 Storage	🟨 Storage	🟨 Storage	
🟦 Servers	🟨 Servers	🟨 Servers	

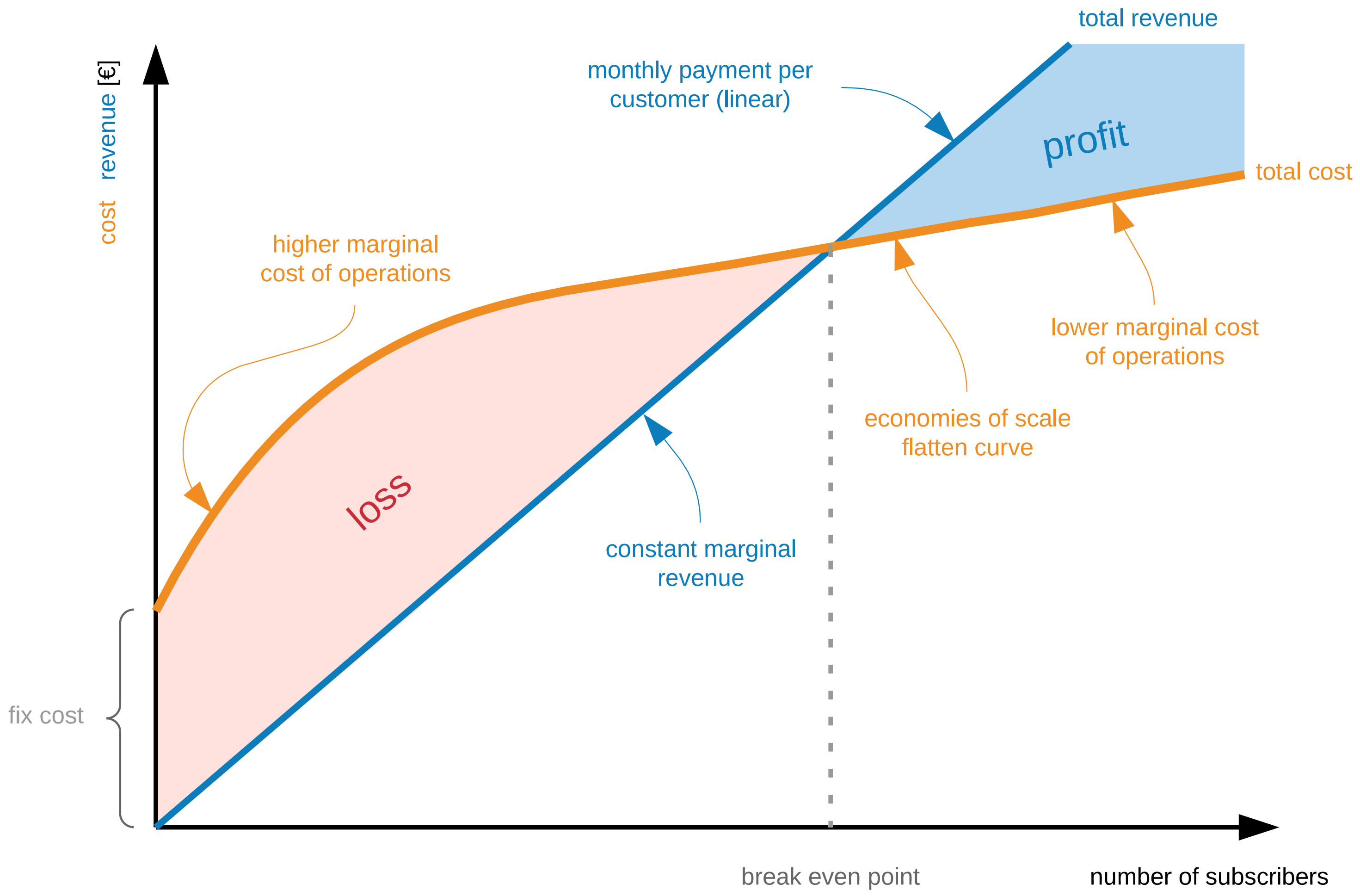
🟨 You rent, Service Provider manages

🟦 You own, and manage

Step 1: Move to Cloud SaaS (Business Model Progression)



Step 2: Optimize the Margin Costs


total revenue
monthly payment per customer (linear)
profit
total cost
cost revenue [€]
higher marginal cost of operations
lower marginal cost of operations
economies of scale flatten curve
loss
constant marginal revenue
fix cost
break even point
number of subscribers

Marginal cost of operations = opportunity cost for adding the next subscriber

(subscription fee must be higher)

PUBLIC

Public

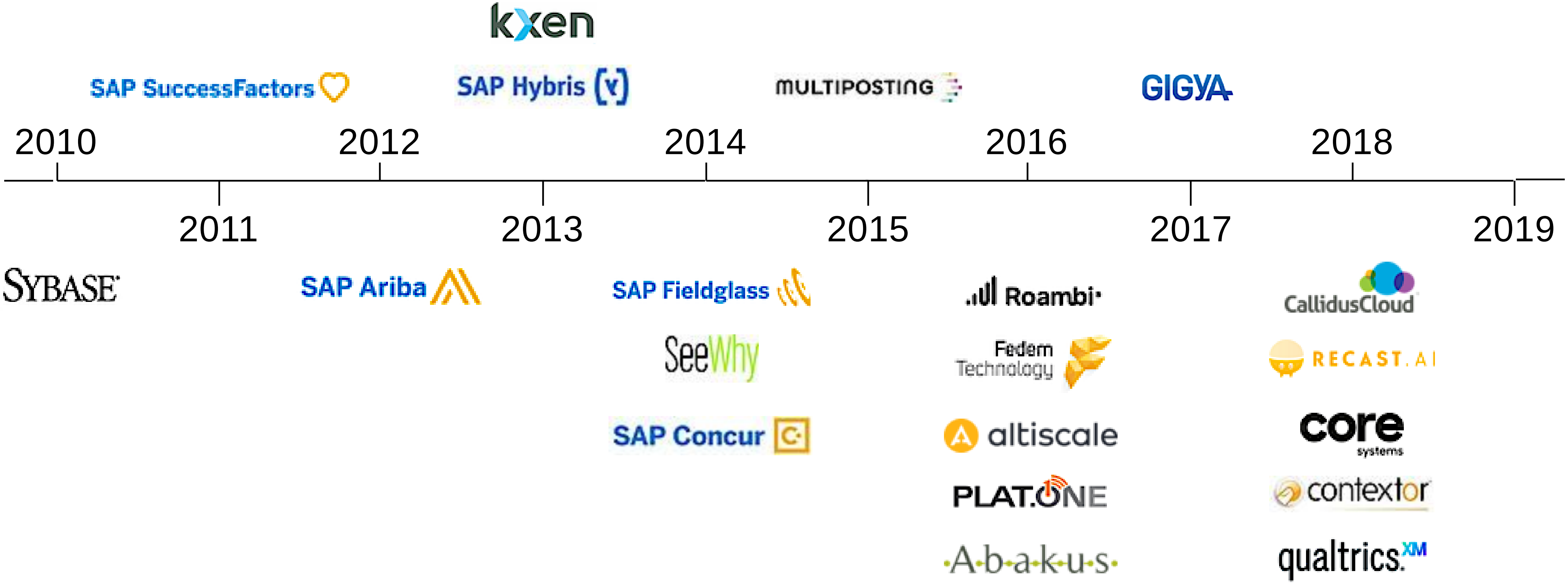
The subscription fee must be competitive.

Cost of operation is tied to the technology stack.

SaaS run on a technology stack that is designed for clouds is more competitive.

Also, there was always more than ABAP ...

SAP Acquisitions



kxen

SAP SuccessFactors

SAP Hybris

MULTIPOSTING

GIGYA

2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

SYBASE

SAP Ariba

SAP Fieldglass

SeeWhy

SAP Concur

Roambi

Fedem Technology

altiscale

PLAT.ONE

A-b-a-k-u-s

CallidusCloud

RECAST.AI

core systems

contextor

qualtrics XM

This is an Innovators Dilemma.

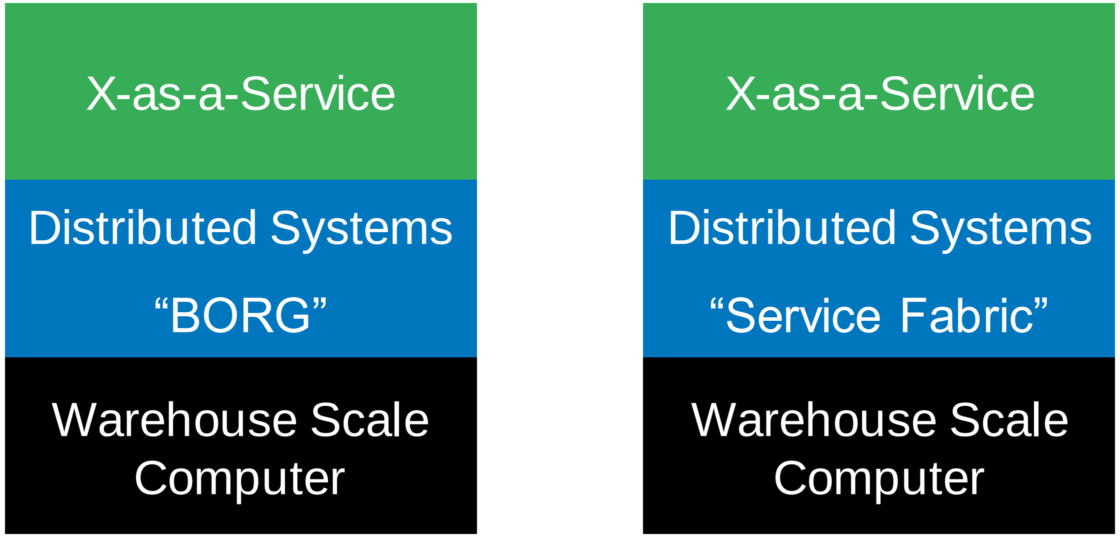
And we were thinking:

Is there possibly a common denominator?

Clouds and competitive, modern SaaS are inherently powered by “Distributed Systems”

(IaaS/PaaS are important implementation details)

Hyperscalers:



X-as-a-Service

Distributed Systems

“BORG”

Warehouse Scale Computer

X-as-a-Service

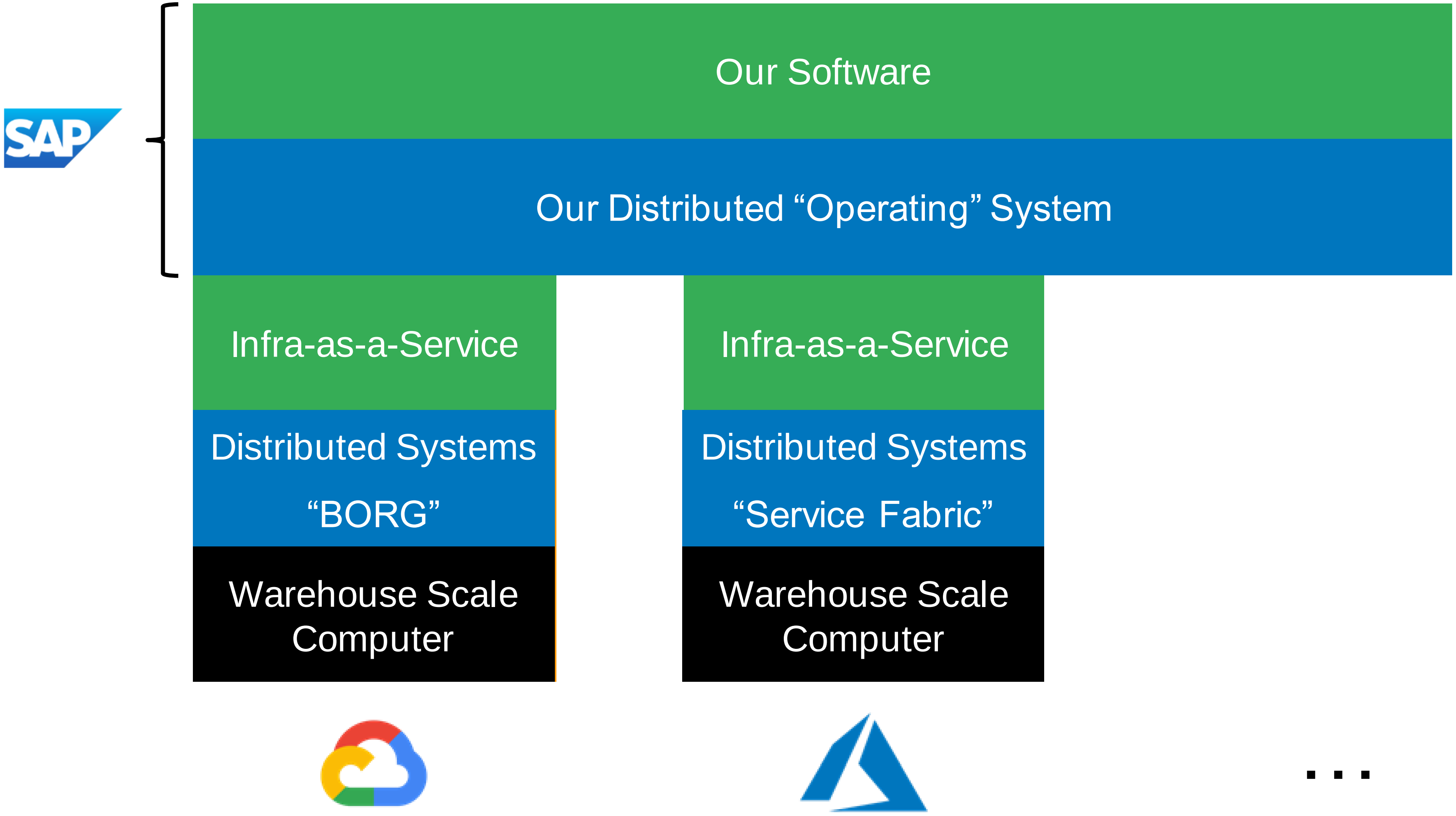
Distributed Systems

“Service Fabric”

Warehouse Scale Computer

. . .

Idea:



Our Software

Our Distributed “Operating” System

Infra-as-a-Service

Distributed Systems

“BORG”

Warehouse Scale Computer

Infra-as-a-Service

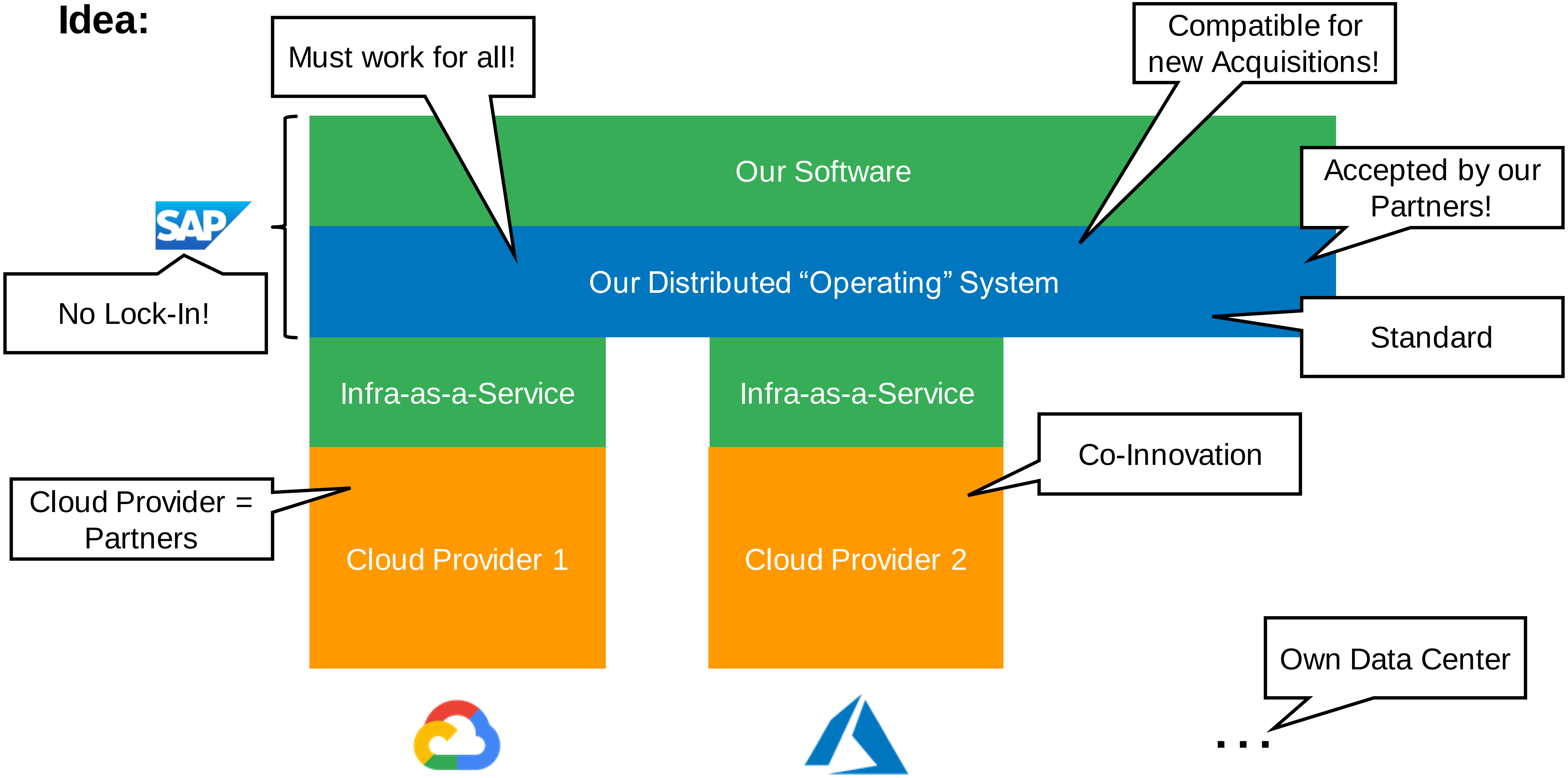
Distributed Systems

“Service Fabric”

Warehouse Scale Computer

...

Idea:



Must work for all!

Compatible for new Acquisitions!

Accepted by our Partners!

Standard

No Lock-In!

Our Software

Our Distributed “Operating” System

Infra-as-a-Service

Infra-as-a-Service

Co-Innovation

Cloud Provider = Partners

Cloud Provider 1

Cloud Provider 2

Own Data Center

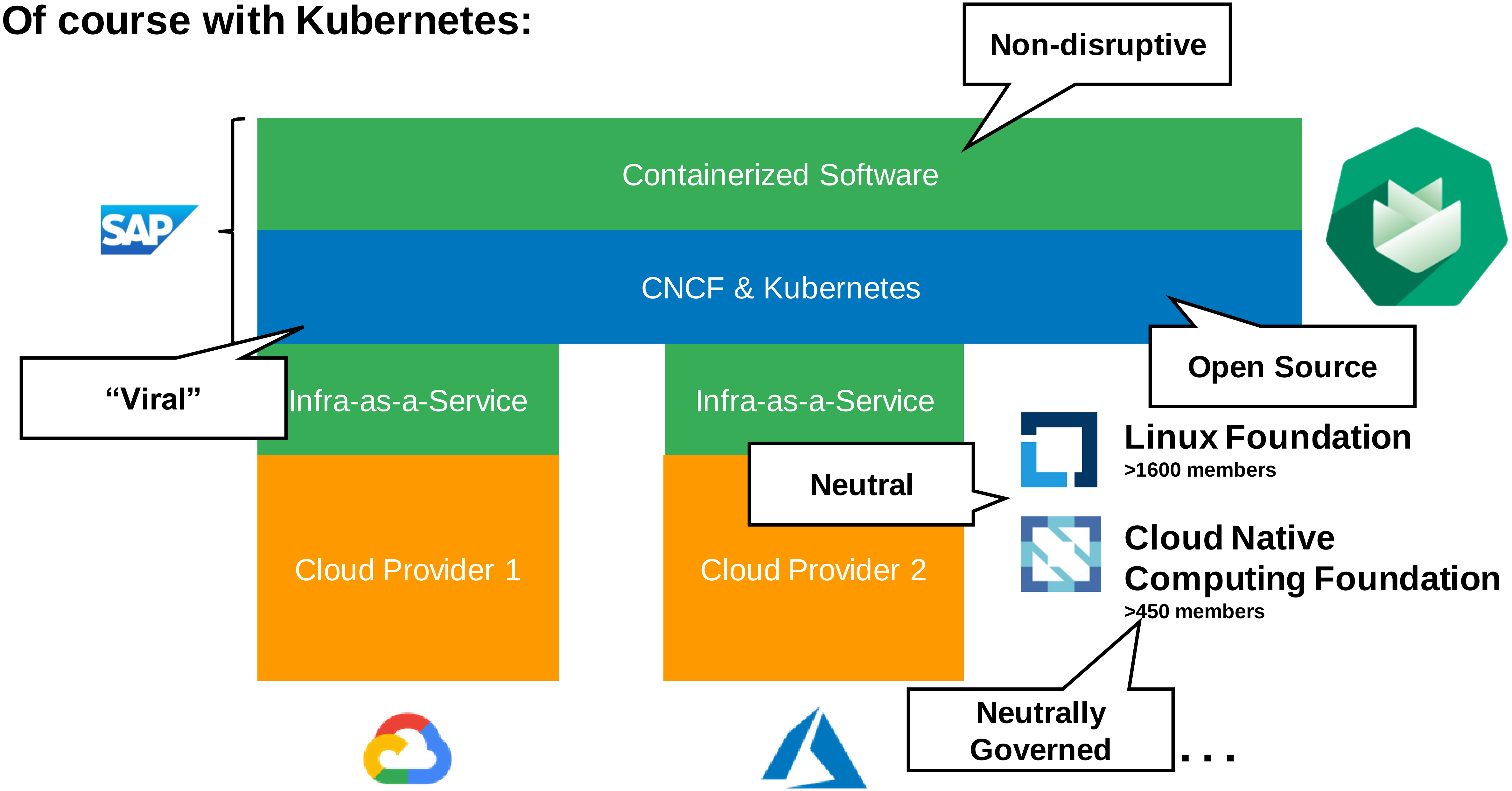
...

How do we bring a “Distributed Operating System” into the corporate value stream?

(Answer: without disruption! We slip it underneath as underlay. The product-teams need to adopt it themselves :-)

And how did we get started?

Of course with Kubernetes:



Non-disruptive

Containerized Software

CNCF & Kubernetes

“Viral”

Open Source

Infra-as-a-Service

Infra-as-a-Service

Neutral

Cloud Provider 1

Cloud Provider 2

Linux Foundation
>1600 members

Cloud Native Computing Foundation
>450 members

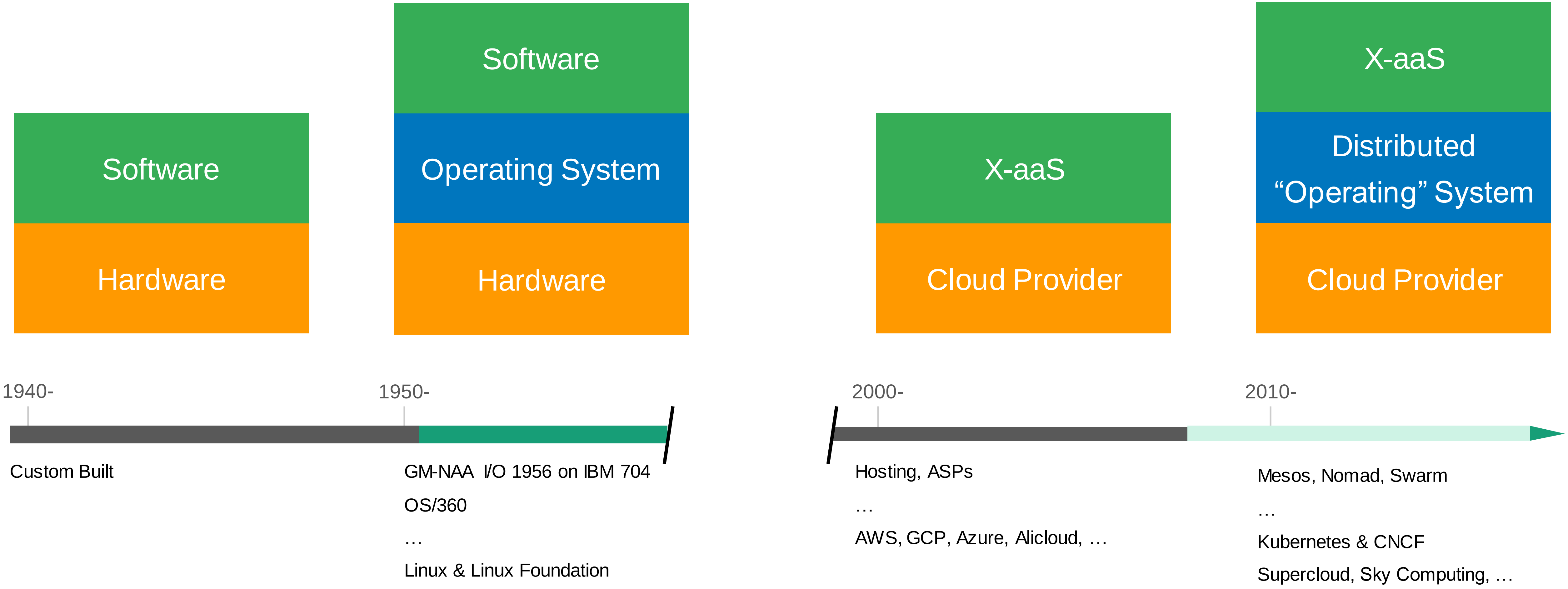
Neutrally Governed

. . .

Paradigm Shift: Computing is on the path of industrial Standardization

Cambrian Explosion of Innovations leveraging Operating System

> *"Open source has been a key enabler for distributed systems, allowing developers to build on top of existing software and collaborate with others to solve complex problems."*
>
> **– Tim O'Reilly, Entrepreneur and Publisher**



Software

Hardware

1940-

Custom Built

Software

Operating System

Hardware

1950-

GM-NAA I/O 1956 on IBM 704

OS/360

…

Linux & Linux Foundation

X-aaS

Cloud Provider

2000-

Hosting, ASPs

…

AWS, GCP, Azure, Alicloud, …

X-aaS

Distributed "Operating" System

Cloud Provider

2010-

Mesos, Nomad, Swarm

…

Kubernetes & CNCF

Supercloud, Sky Computing, …

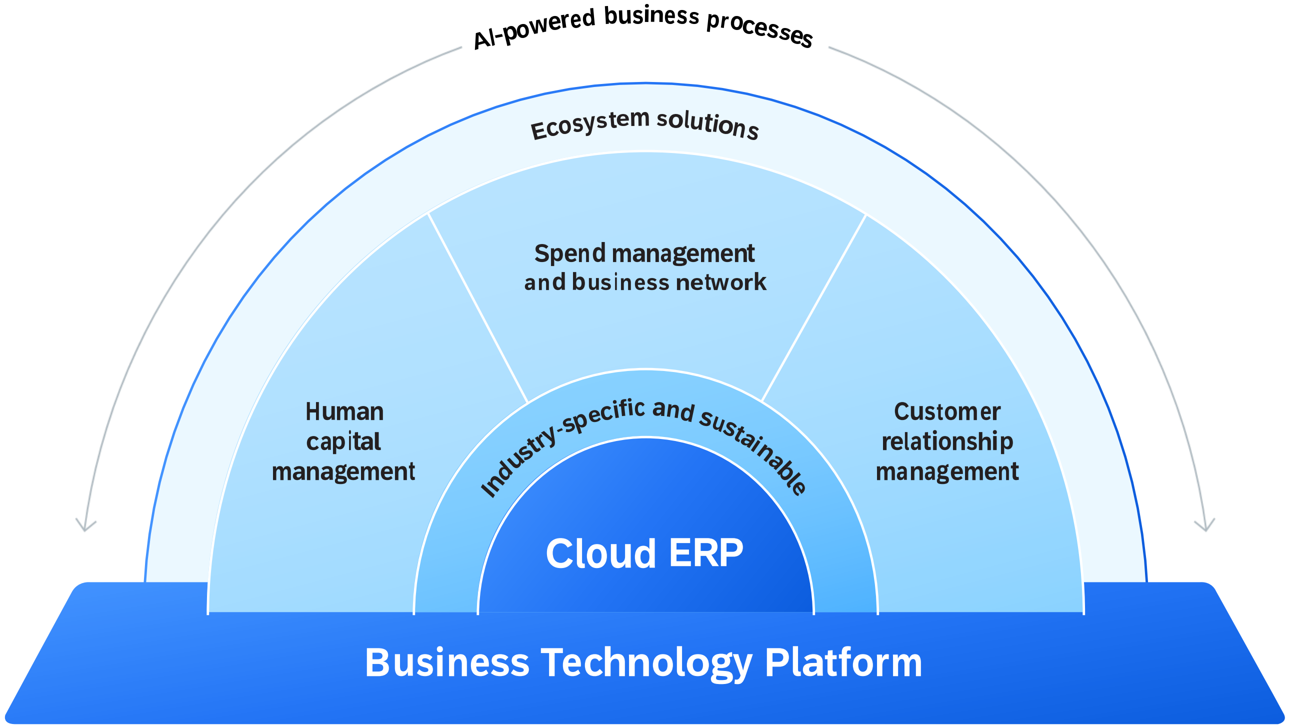
… skipping a number of Corporate Cloud Transformation, Open-Source, and OSPO topics …

“More than 85% of organizations will embrace a cloud-first principle by 2025 and will not be able to fully execute on their digital strategies without the use of cloud-native architectures and technologies.”

- Gartner

PUBLIC

High-Level for SAP’s commercial Services:

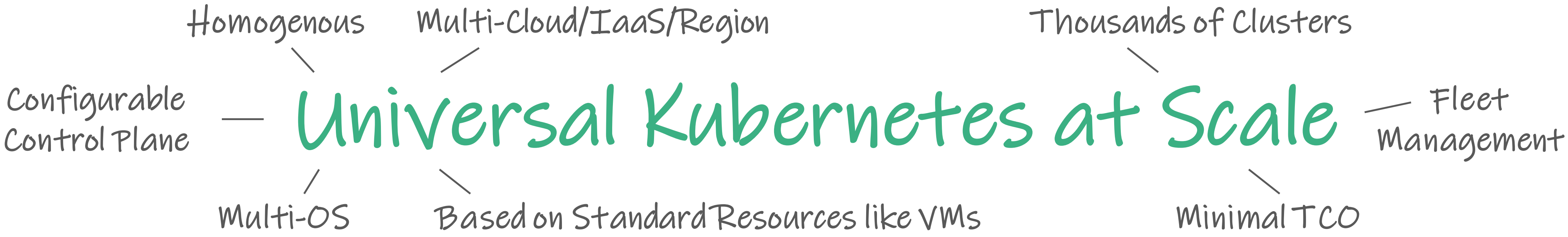

AI-powered business processes
Ecosystem solutions
Spend management and business network
Human capital management
Customer relationship management
Industry-specific and sustainable
Cloud ERP
Business Technology Platform

Cloud-first

Cloud-native architectures

Gotta start somewhere …

The **Gardener** Mission Statement



Homogenous

Multi-Cloud/IaaS/Region

Thousands of Clusters

Configurable Control Plane

Universal Kubernetes at Scale

Fleet Management

Multi-OS

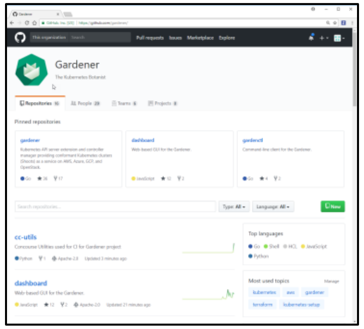
Based on Standard Resources like VMs

Minimal TCO

https://gardener.cloud

Multicluster Ops UI

Any Infrastructure.
Public. Private.
Proprietary.
Bare Metal.



gardener
Kubernetes API server extension and controller manager providing conformant Kubernetes clusters (Shoots) as a service on AWS, Azure, GCP, and OpenStack.

dashboard
Web-based GUI for the Gardener.

gardenctl
Command-line client for the Gardener.

cc-utils
Concourse Utilities used for CI for Gardener project

dashboard
Web-based GUI for the Gardener.

Open Source

(part of the Distributed "Operating" System Story)

Lingua Franca

Infra-as-a-Service

Cloud Providers

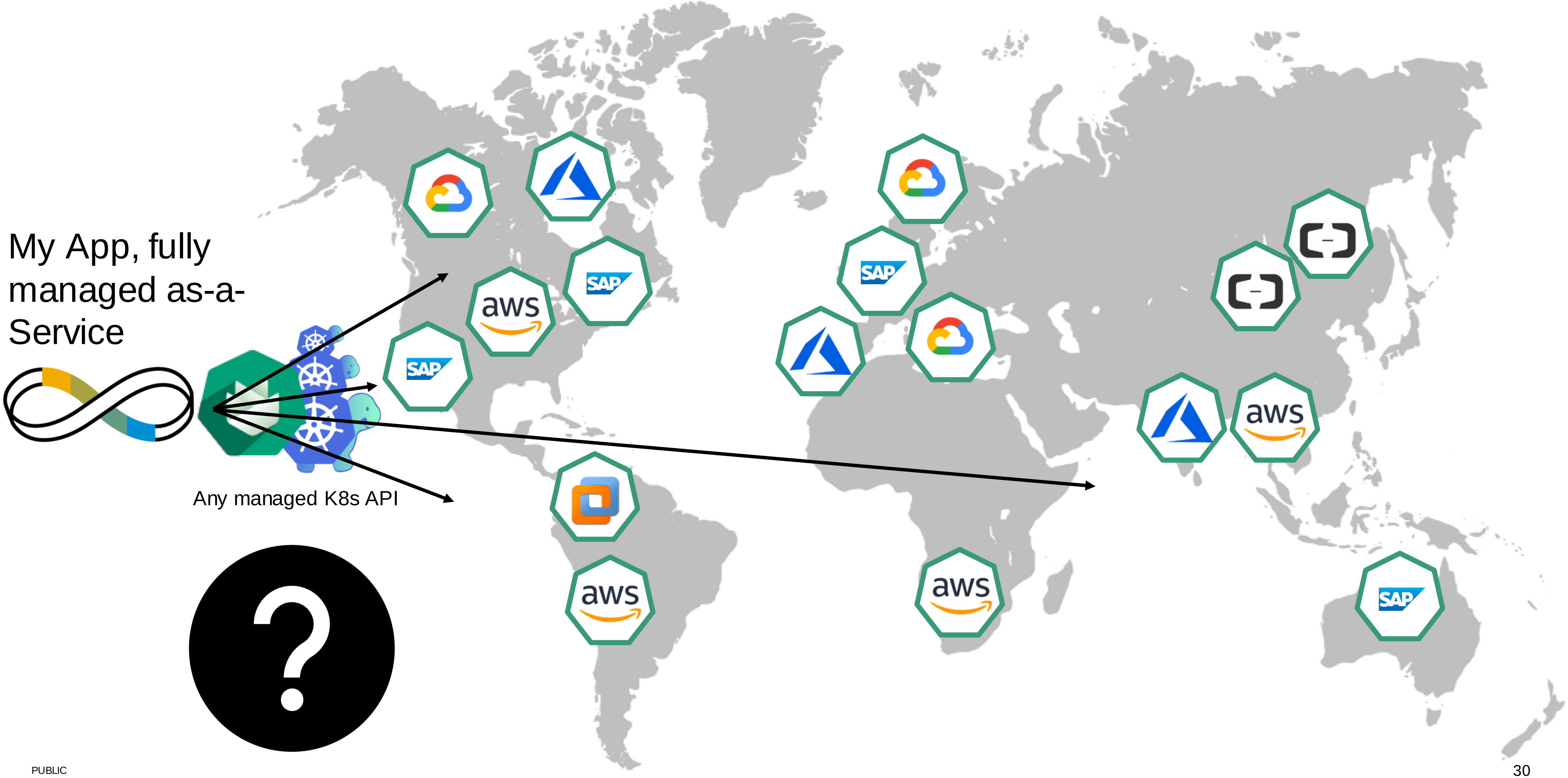
. . .

Kubernetes+Containers help with Portability, with Abstraction of IaaS Providers.
CNCF overall helps with Open Standards.

But how does this help me to create my globally distributed SaaS?

Challenge: Deliver SaaS in Clouds and at Global Scale (with Kubernetes)

My App, fully managed as-a-Service

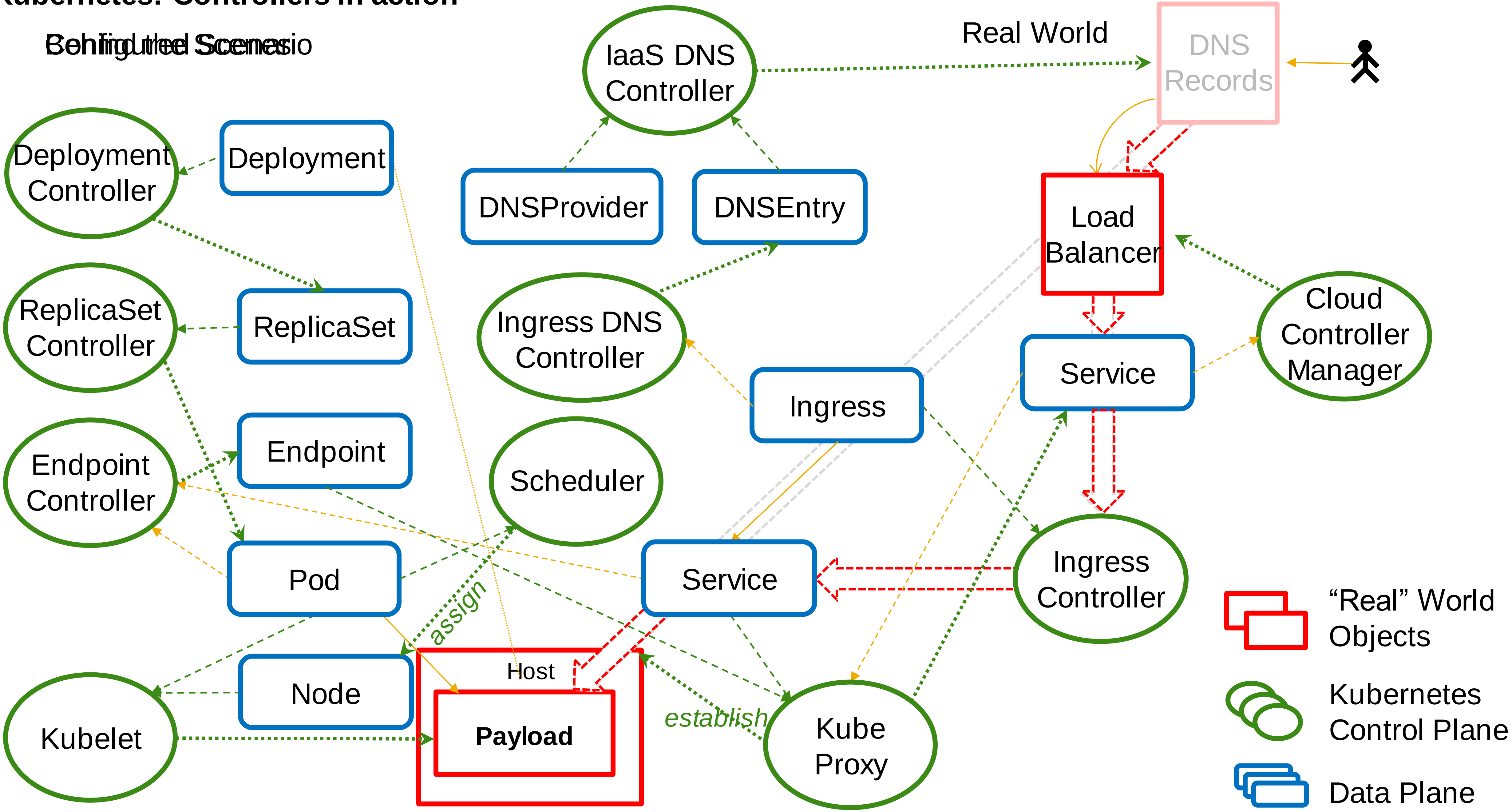


Kubernetes Controller in Action

Kubernetes: Controllers in action

Behind the Scenes

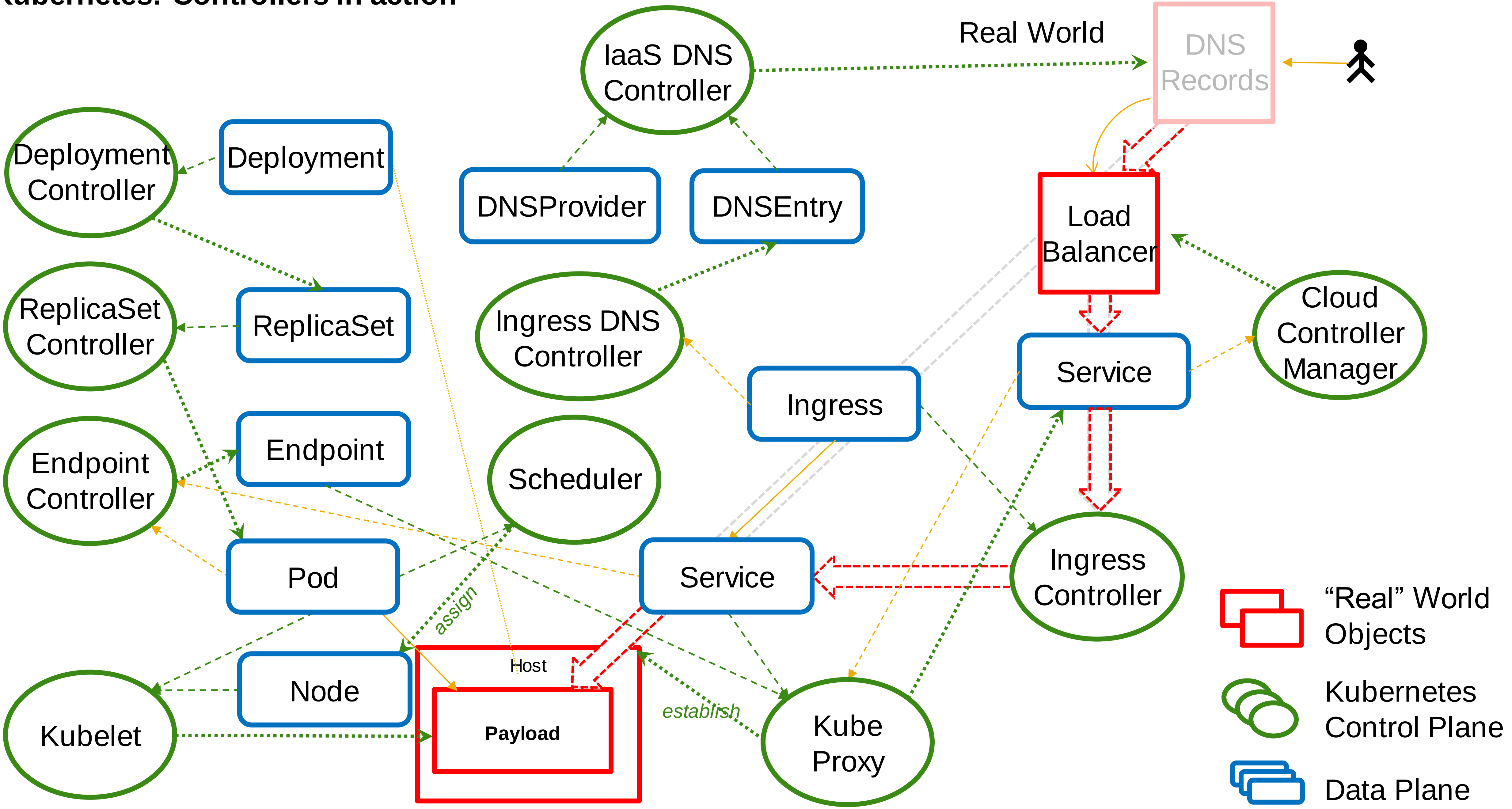
Configured Scenario



Four Lessons

1. Would a Waterfall Architecture Council approve this?
2. Control through Choreography, or “Adam Smith’s Invisible Hand”.
3. With honest participants, this architecture is extremely robust.
4. ➔ …

Kubernetes: Controllers in action



There is a pattern

hidden

in front of your eyes!

Scalable Service Management

with a

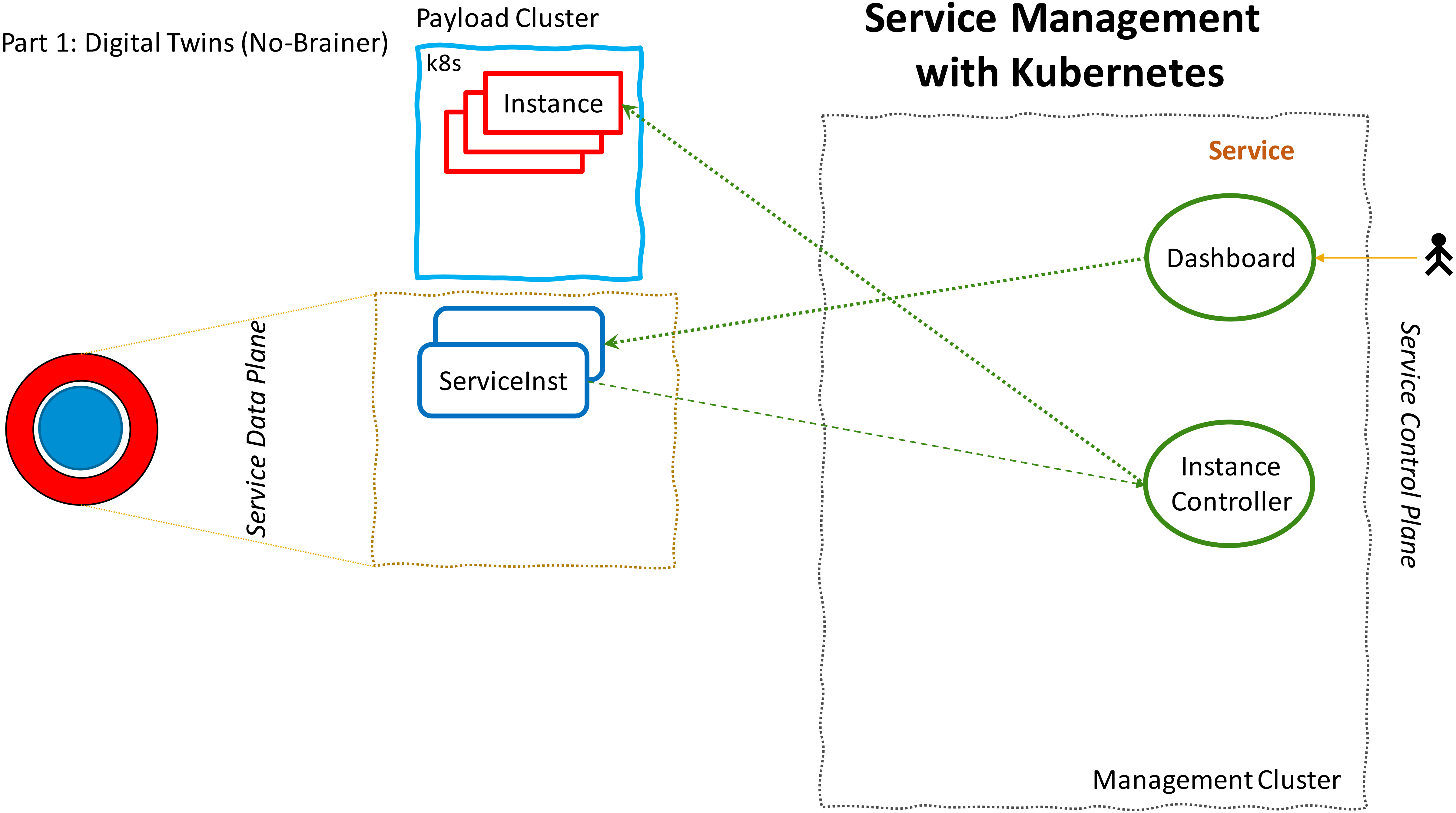
Cloud-Native Service Pattern

Service Management with Kubernetes

Adapting Kubernetes Design Patterns

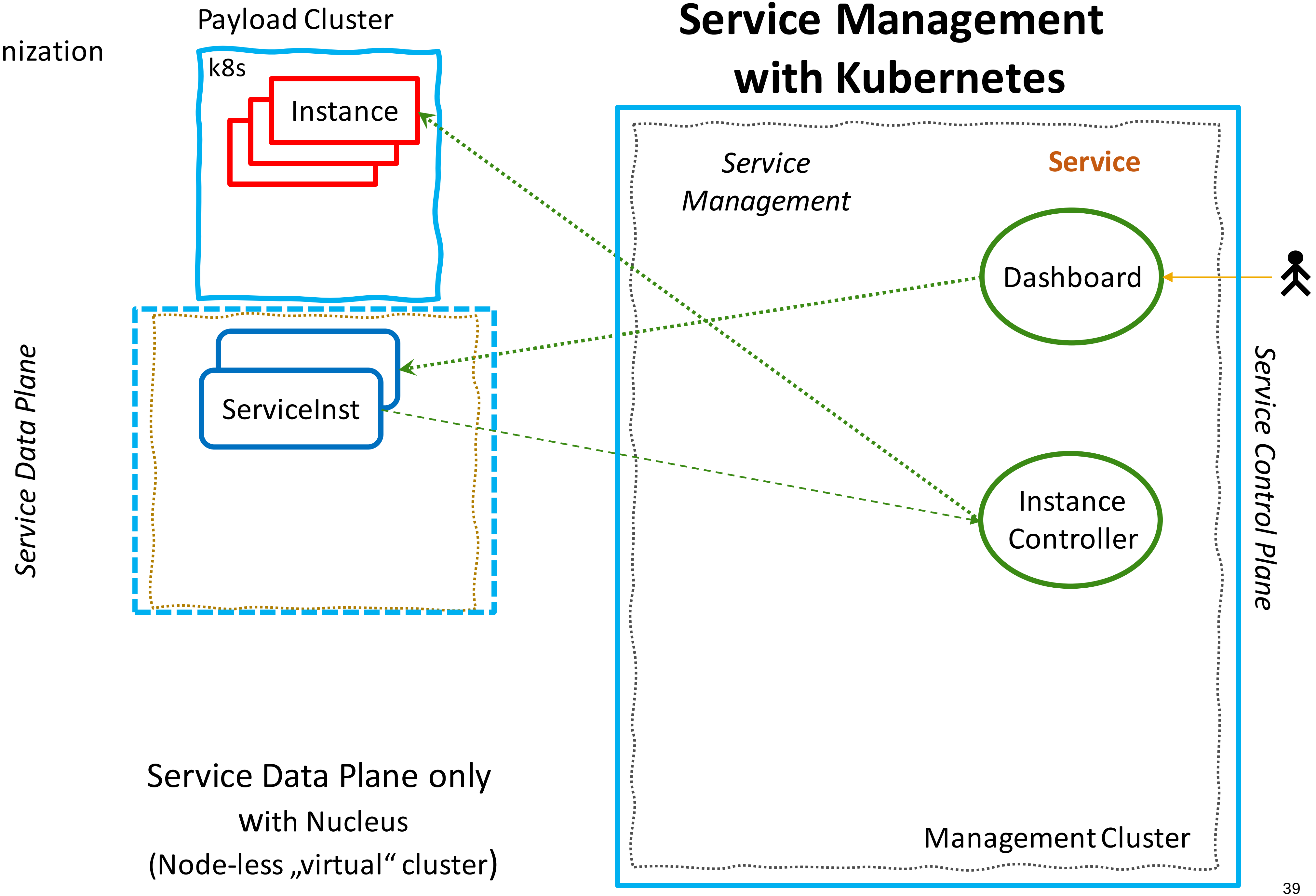
Service Management with Kubernetes

Part 1: Digital Twins (No-Brainer)



Part 2: Cluster Organization

Service Management with Kubernetes



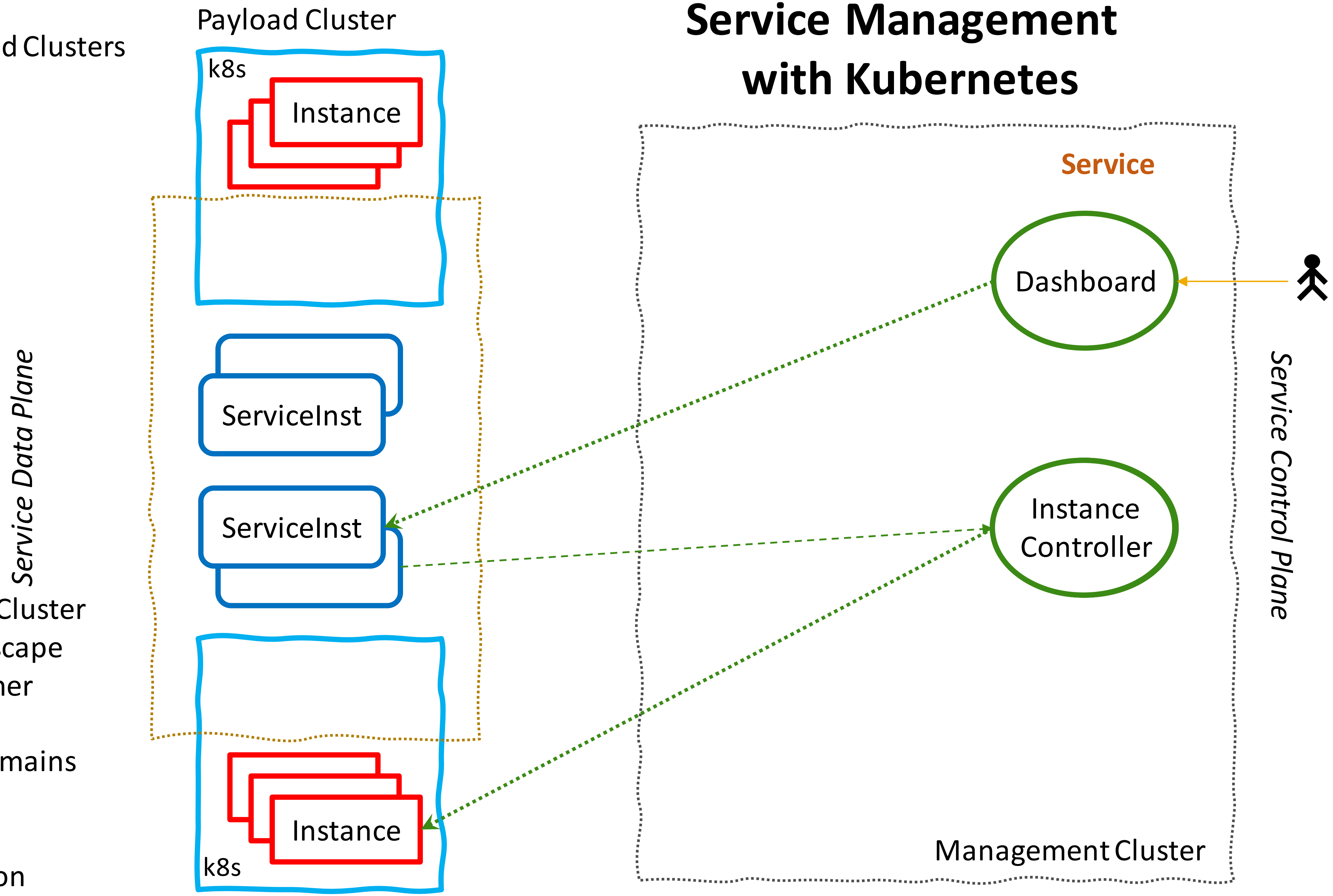
Service Data Plane only

with Nucleus

(Node-less „virtual“ cluster)

Service Management with Kubernetes

Part 3: Multi-Payload Clusters

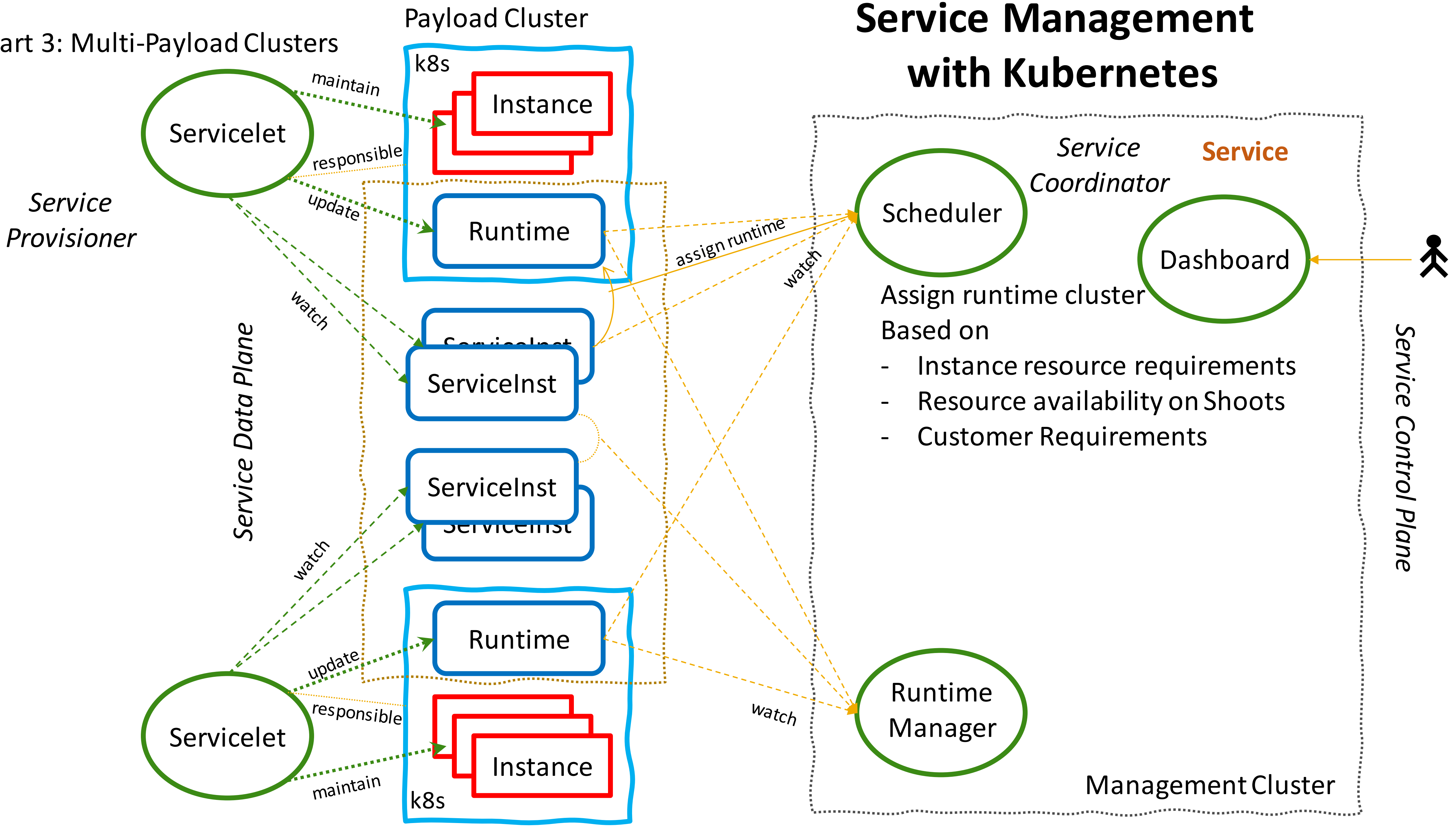


WHY?

- Size Limit of single Cluster
- Multi-Region Landscape
- Deploy into Customer Environments
- Multiple Failure Domains
- Latency
- Security
- Customer Separation

Service Management with Kubernetes

Part 3: Multi-Payload Clusters

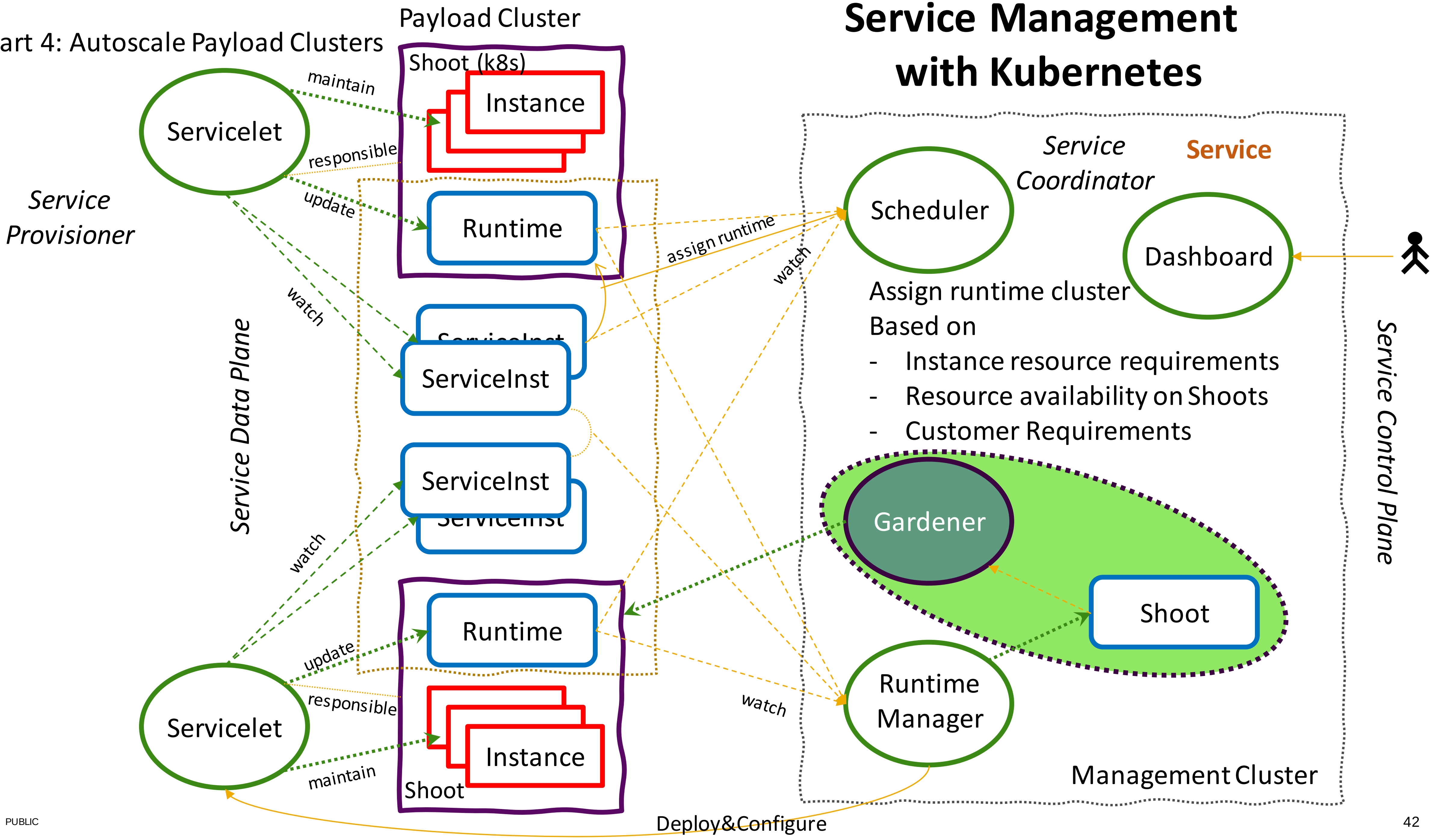


Assign runtime cluster
Based on

- Instance resource requirements
- Resource availability on Shoots
- Customer Requirements

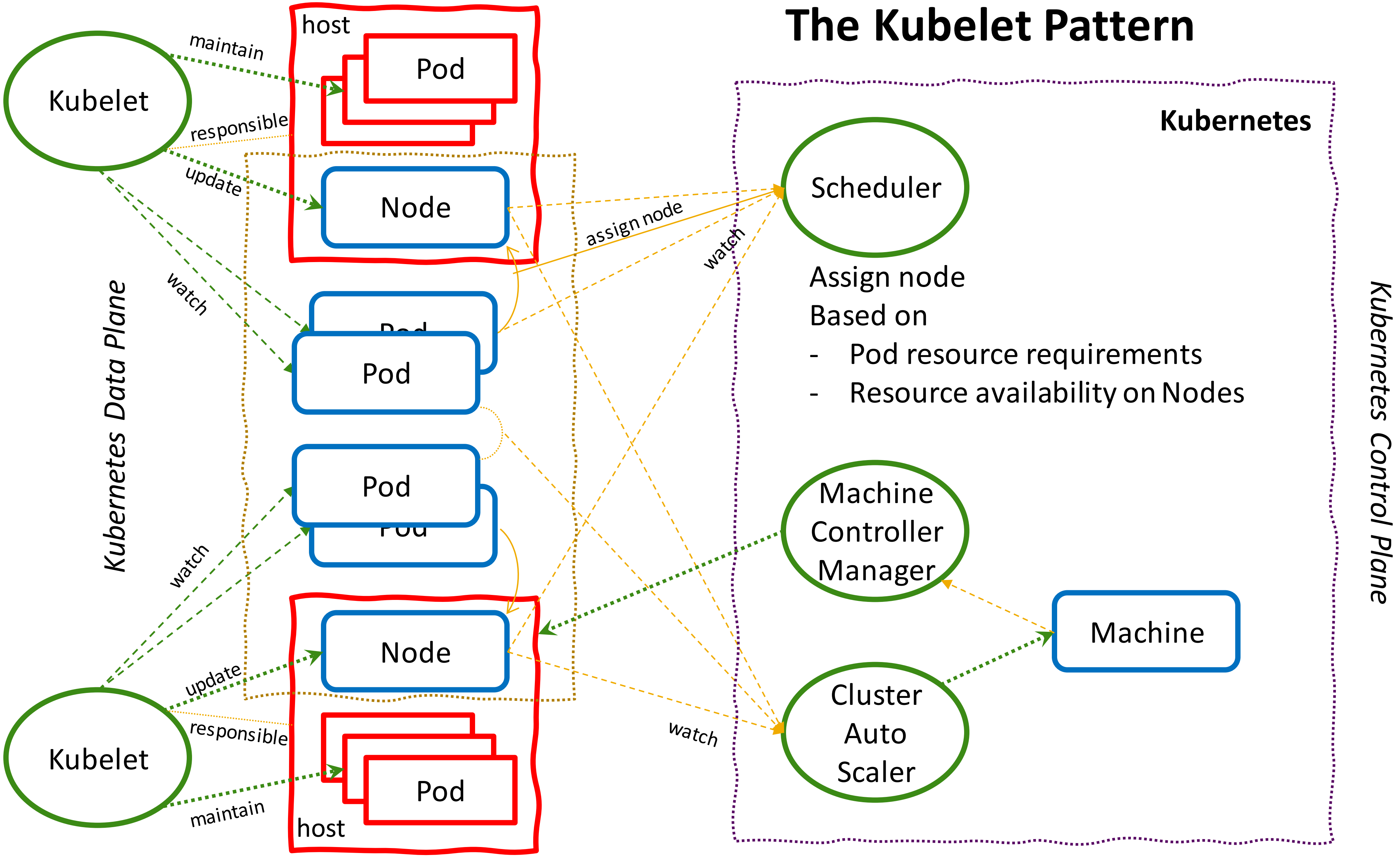
Service Management with Kubernetes

Part 4: Autoscale Payload Clusters



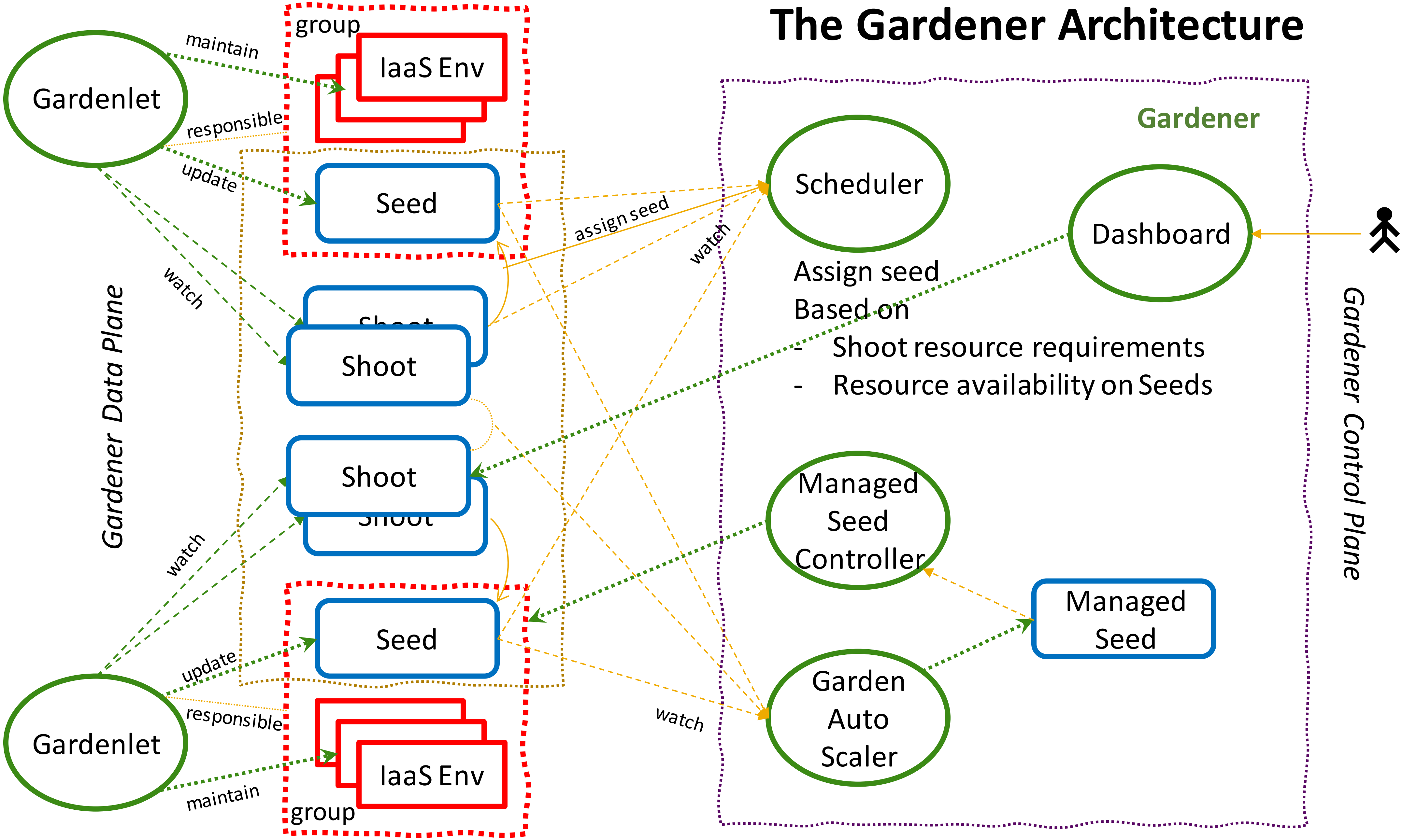
Assign runtime cluster
Based on
- Instance resource requirements
- Resource availability on Shoots
- Customer Requirements

The Kubelet Pattern



Assign node
Based on
- Pod resource requirements
- Resource availability on Nodes

The Gardener Architecture



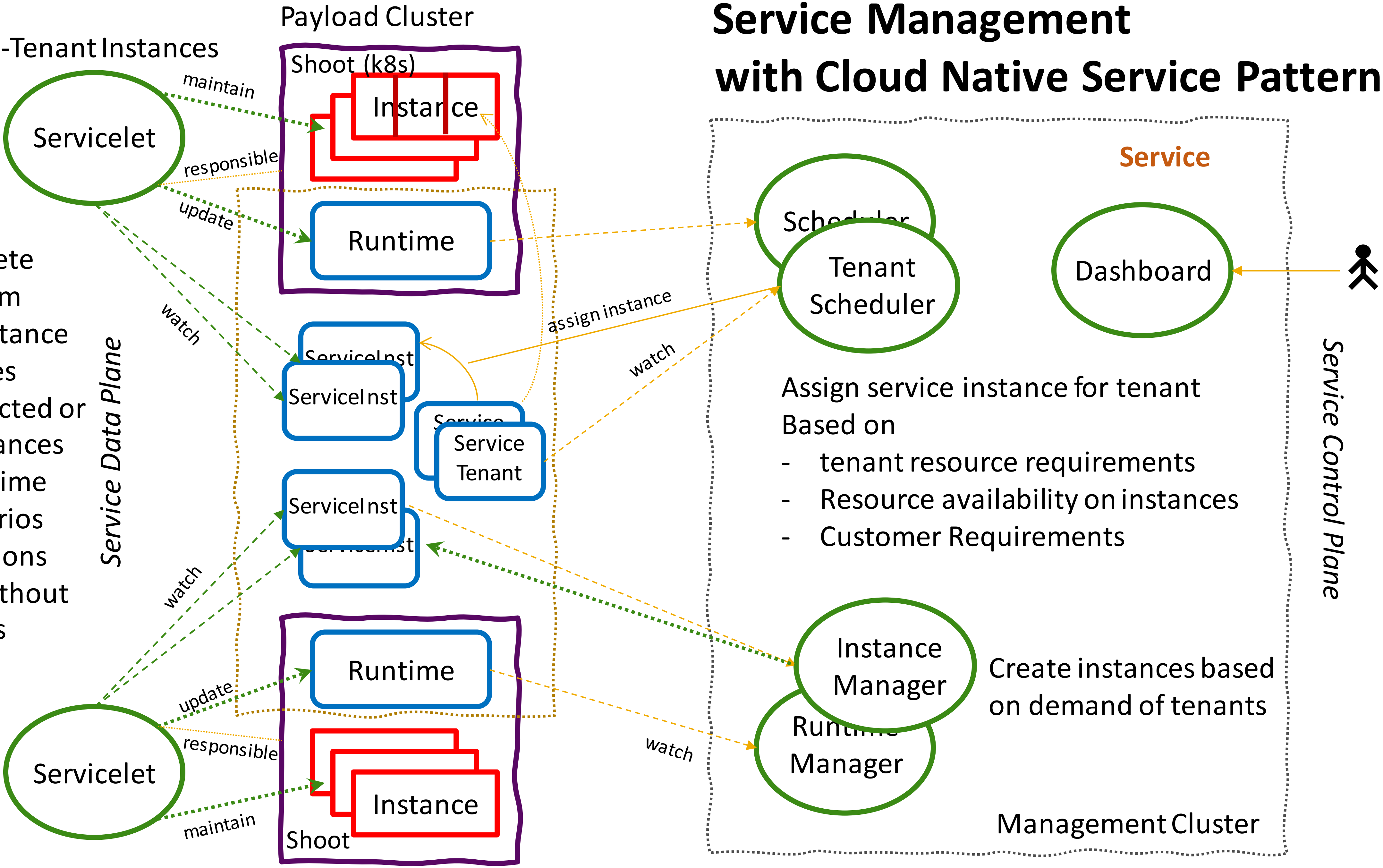
Assign seed
Based on

- Shoot resource requirements
- Resource availability on Seeds

Service Management with Cloud Native Service Pattern

Part 5: Multi-Tenant Instances

- Cover complete spectrum from central to instance based services
- Support dedicted or Sharded instances at the same time
- Hybrid scenarios
- All combinations selectable without code changes

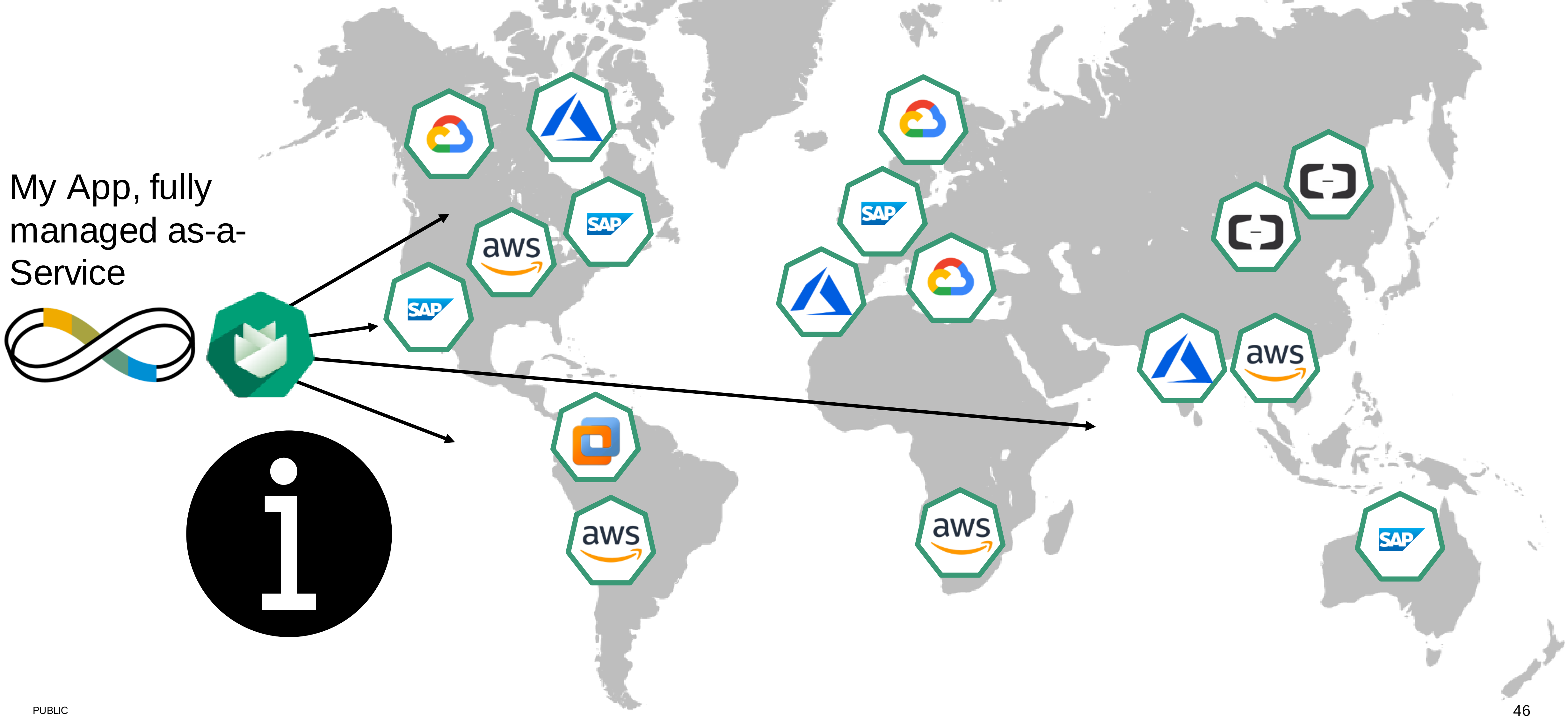


Assign service instance for tenant
Based on
- tenant resource requirements
- Resource availability on instances
- Customer Requirements

Create instances based on demand of tenants

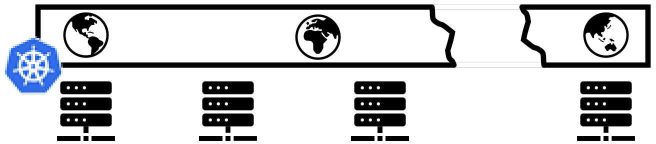
Conclusion: Deliver SaaS in Multi-Clouds and at Global Scale with Kubernetes and Kubernetes inspired Cloud-Native Service Pattern

My App, fully managed as-a-Service



Start with a “Distributed System”

Top Down



➔ Gardener API

Autoscaling (Seeds, Pivoting Control Planes, …)

Localized Business Logic in “Gardenlet”

Gardener Extension (GCP, AWS, OpenStack, …)

Platform Rollout/Policy/Config of Available Options

Machine
Machine Set
Machine Deployment
Machine Class

Not Included: Embedd into a Platform w/Policy

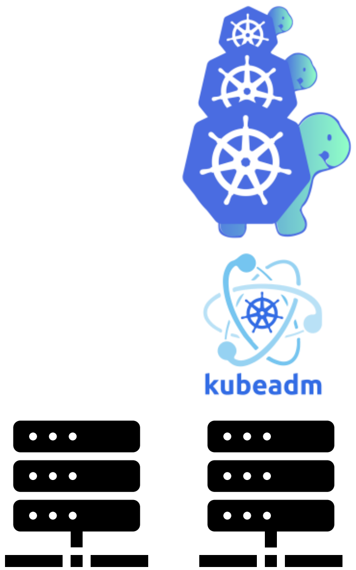
Cluster API Provider (GKE, AKS, kubeadm, …)

➔ Cluster API

kubeadm, k3s, kind, …

Start with a “(Virtual) Machine”

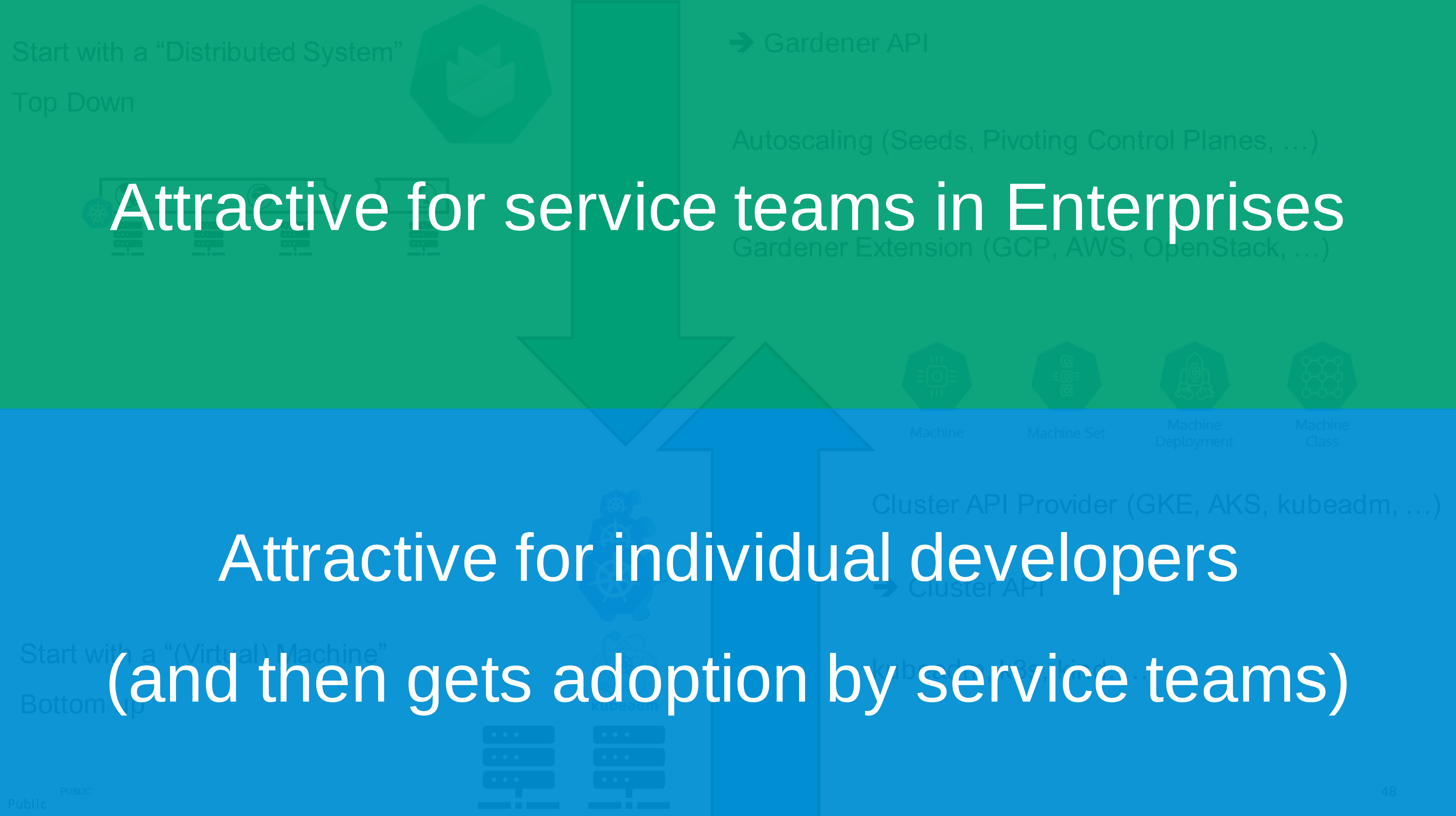
Bottom up



Attractive for service teams in Enterprises

Attractive for individual developers

(and then gets adoption by service teams)



Start with a “Distributed System”

Top Down

➔ Gardener API

Autoscaling (Seeds, Pivoting Control Planes, …)

Gardener Extension (GCP, AWS, OpenStack, …)

Cluster API Provider (GKE, AKS, kubeadm, …)

➔ Cluster API

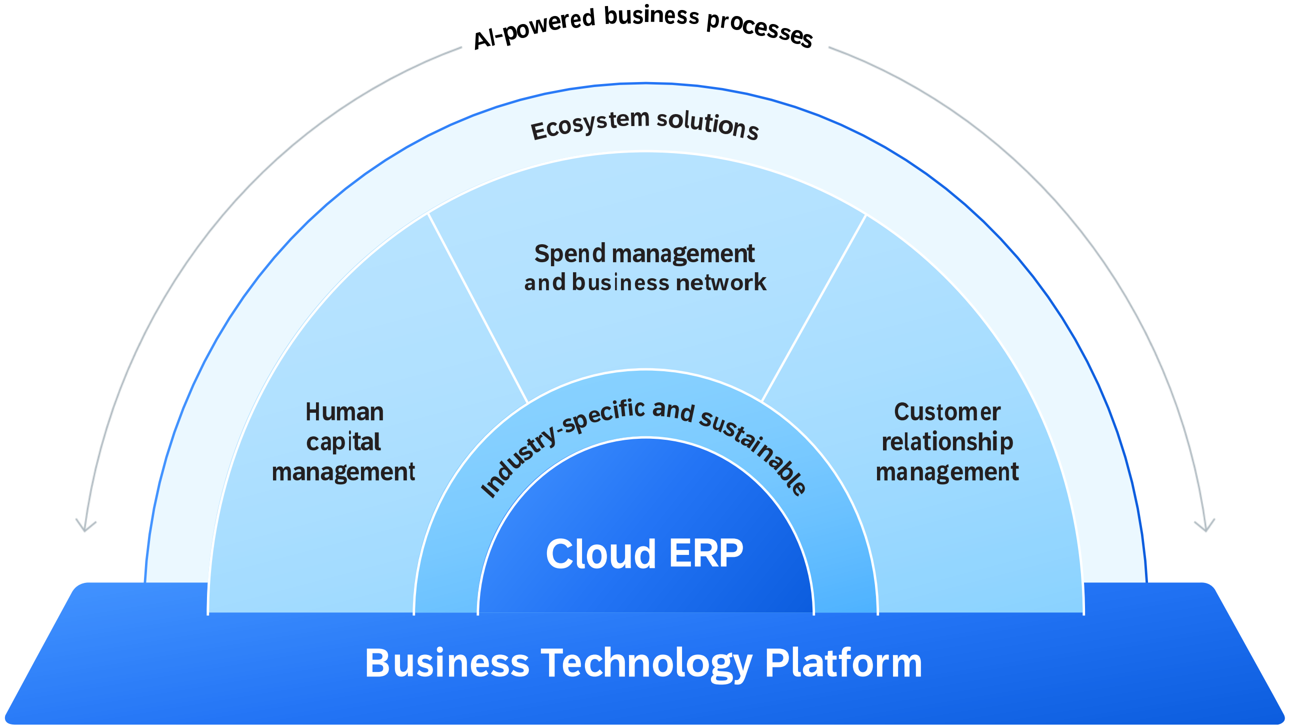
kubeadm, k3s, kind, …

Start with a “(Virtual) Machine”

Bottom Up

Have we been successful?

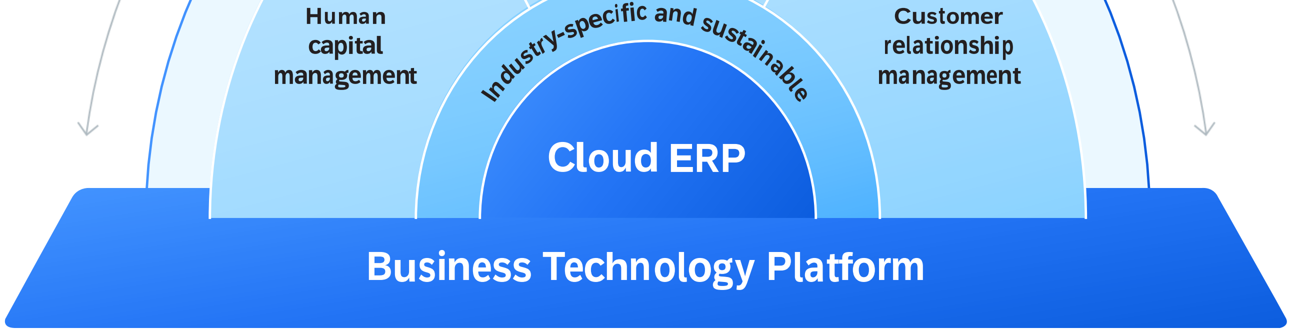
Gardener is powering the Business Technology Platform (BTP)

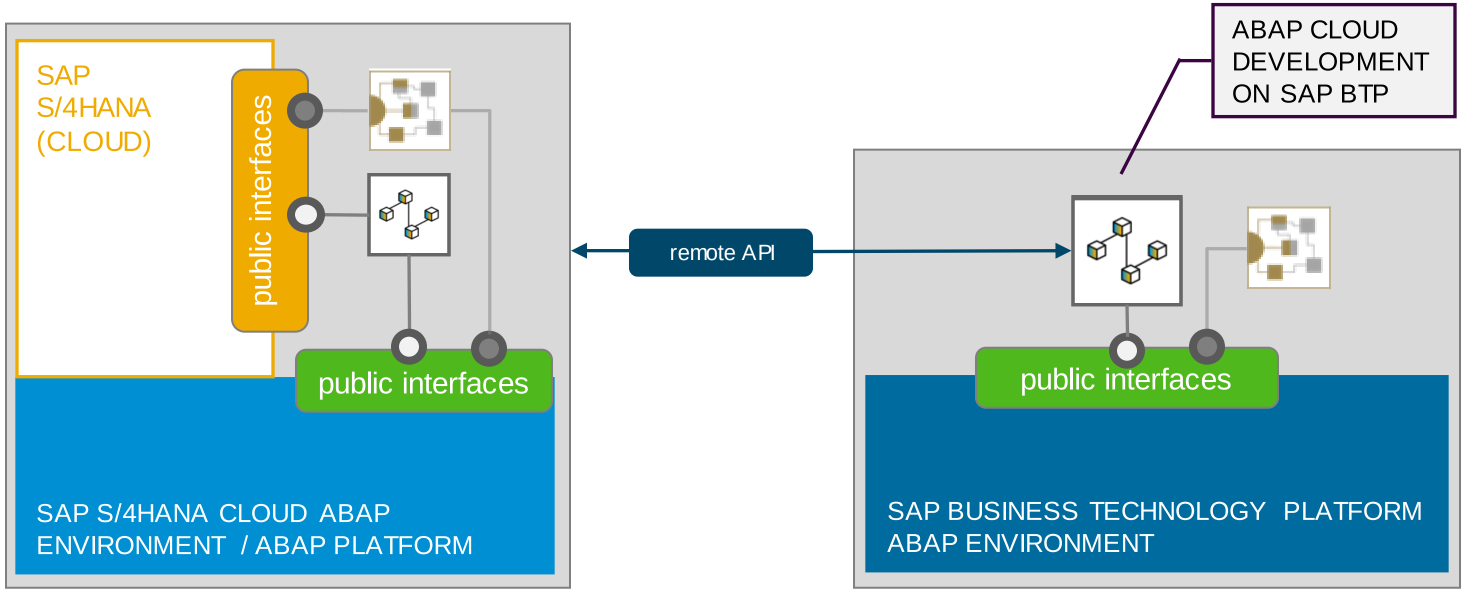


AI-powered business processes
Ecosystem solutions
Spend management and business network
Human capital management
Customer relationship management
Industry-specific and sustainable
Cloud ERP
Business Technology Platform

Internally available for all products to use

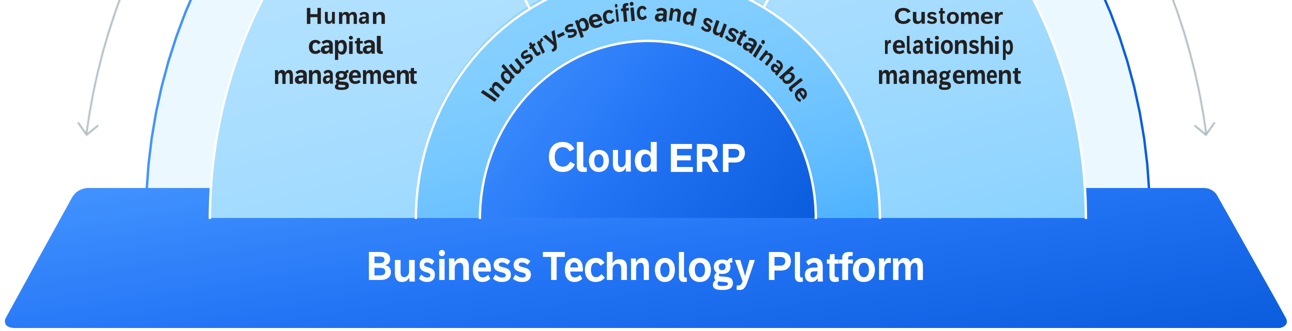
ABAP Cloud


Human capital management
Industry-specific and sustainable
Customer relationship management
Cloud ERP
Business Technology Platform

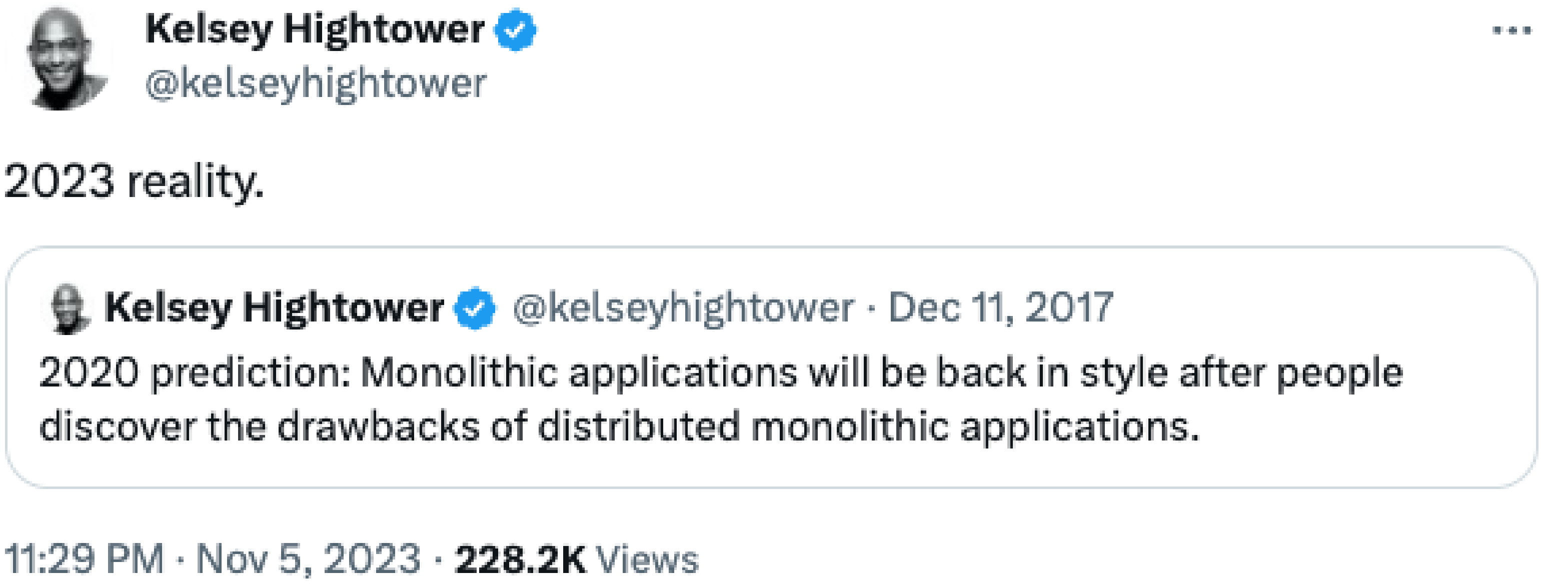

SAP S/4HANA (CLOUD)
public interfaces
public interfaces
SAP S/4HANA CLOUD ABAP ENVIRONMENT / ABAP PLATFORM
remote API
ABAP CLOUD DEVELOPMENT ON SAP BTP
public interfaces
SAP BUSINESS TECHNOLOGY PLATFORM ABAP ENVIRONMENT

ABAP

ABAP Cloud


Human capital management
Industry-specific and sustainable
Customer relationship management
Cloud ERP
Business Technology Platform

STEAMFUNK



Kelsey Hightower
@kelseyhightower

2023 reality.

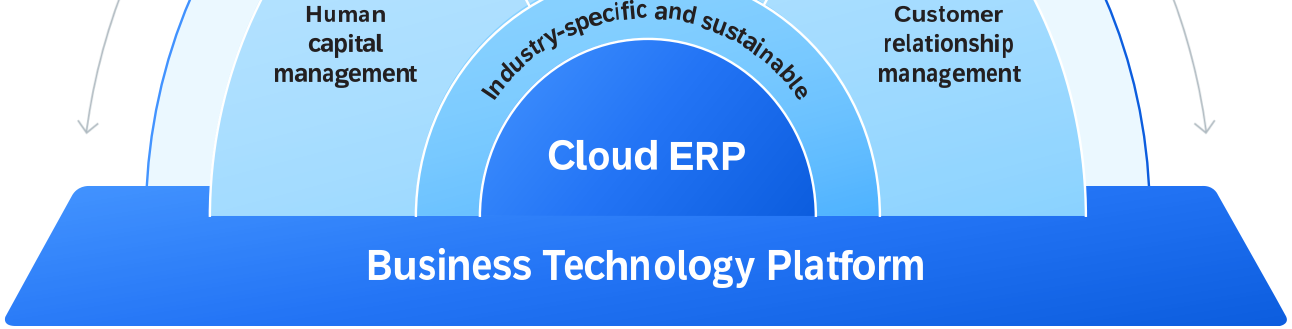
Kelsey Hightower @kelseyhightower · Dec 11, 2017

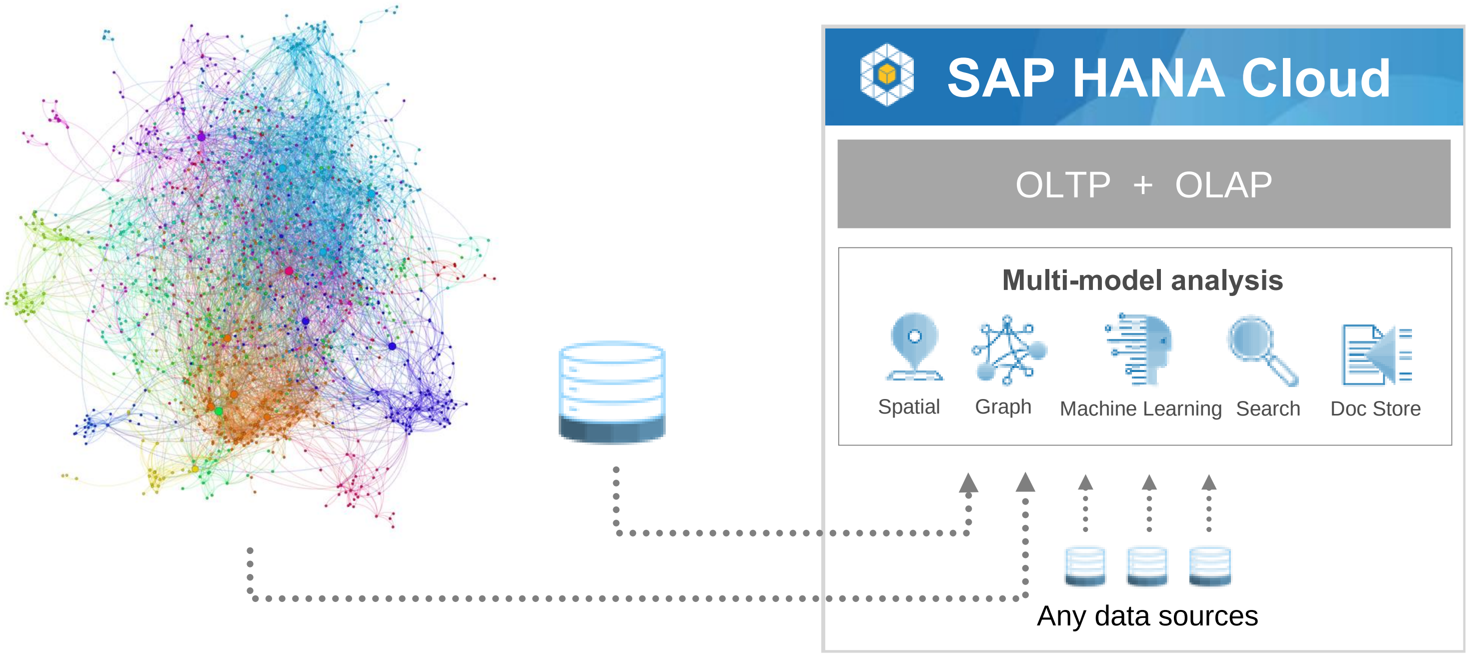
2020 prediction: Monolithic applications will be back in style after people discover the drawbacks of distributed monolithic applications.

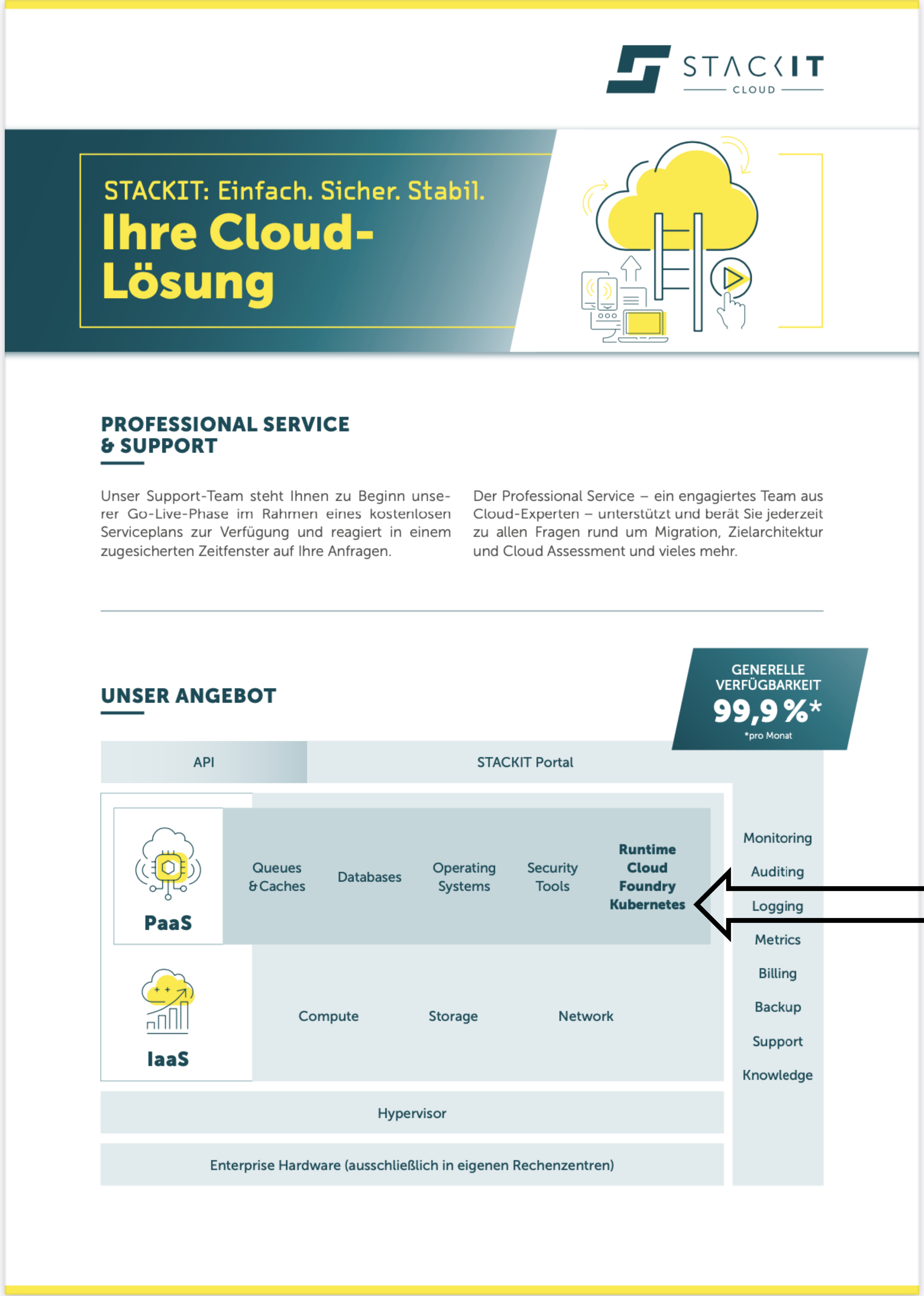
11:29 PM · Nov 5, 2023 · **228.2K** Views

https://twitter.com/kelseyhightower/status/1721293702017216639

HANA Cloud


Human capital management
Industry-specific and sustainable
Customer relationship management
Cloud ERP
Business Technology Platform


SAP HANA Cloud
OLTP + OLAP
Multi-model analysis
Spatial
Graph
Machine Learning
Search
Doc Store
Any data sources



STACKIT CLOUD

STACKIT: Einfach. Sicher. Stabil.

Ihre Cloud-Lösung

PROFESSIONAL SERVICE & SUPPORT

Unser Support-Team steht Ihnen zu Beginn unserer Go-Live-Phase im Rahmen eines kostenlosen Serviceplans zur Verfügung und reagiert in einem zugesicherten Zeitfenster auf Ihre Anfragen.

Der Professional Service – ein engagiertes Team aus Cloud-Experten – unterstützt und berät Sie jederzeit zu allen Fragen rund um Migration, Zielarchitektur und Cloud Assessment und vieles mehr.

UNSER ANGEBOT

GENERELLE VERFÜGBARKEIT 99,9 %*

*pro Monat

API

STACKIT Portal

PaaS

Queues & Caches

Databases

Operating Systems

Security Tools

Runtime Cloud Foundry Kubernetes

IaaS

Compute

Storage

Network

Monitoring

Auditing

Logging

Metrics

Billing

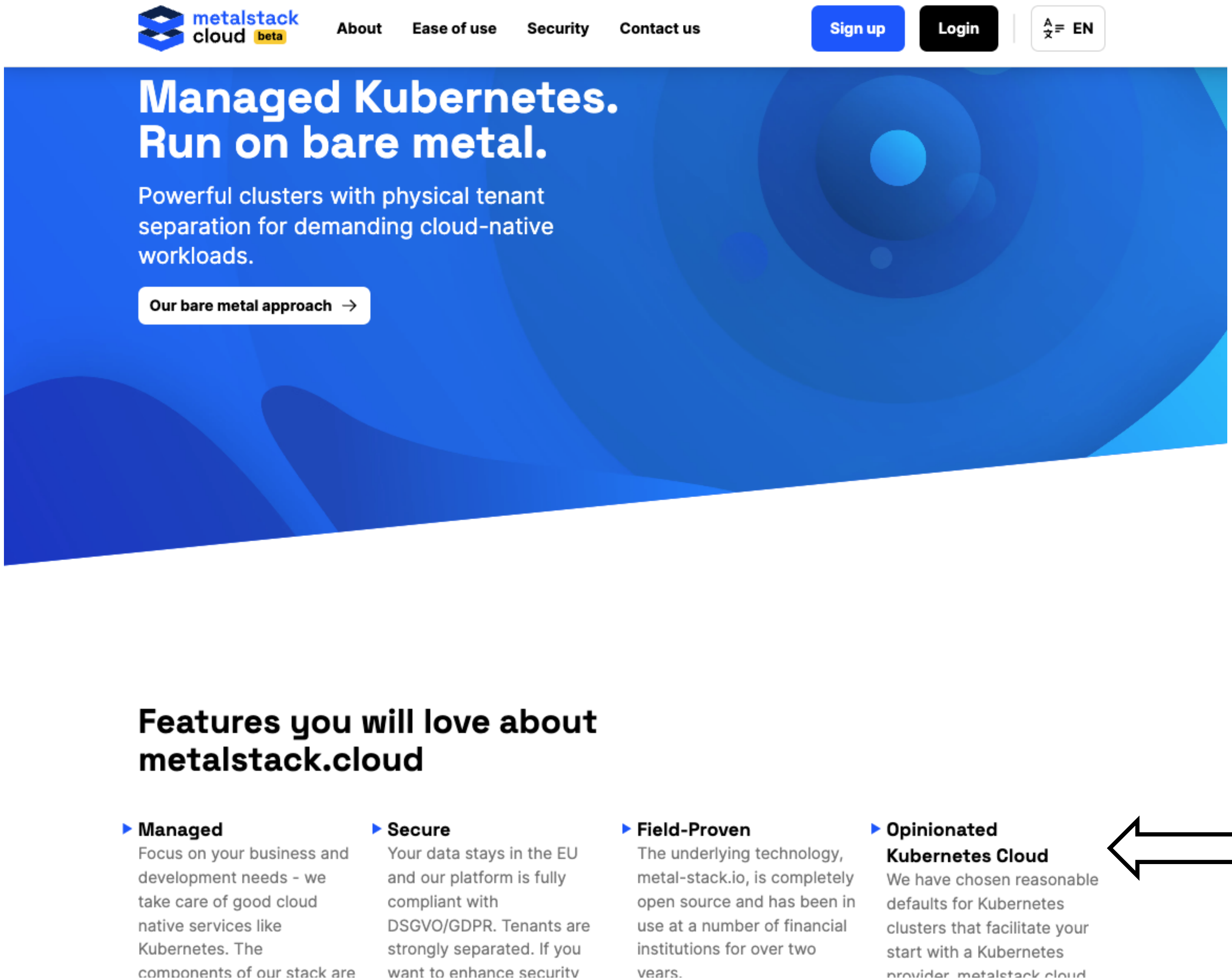
Backup

Support

Knowledge

Hypervisor

Enterprise Hardware (ausschließlich in eigenen Rechenzentren)



Managed Kubernetes. Run on bare metal.

Powerful clusters with physical tenant separation for demanding cloud-native workloads.

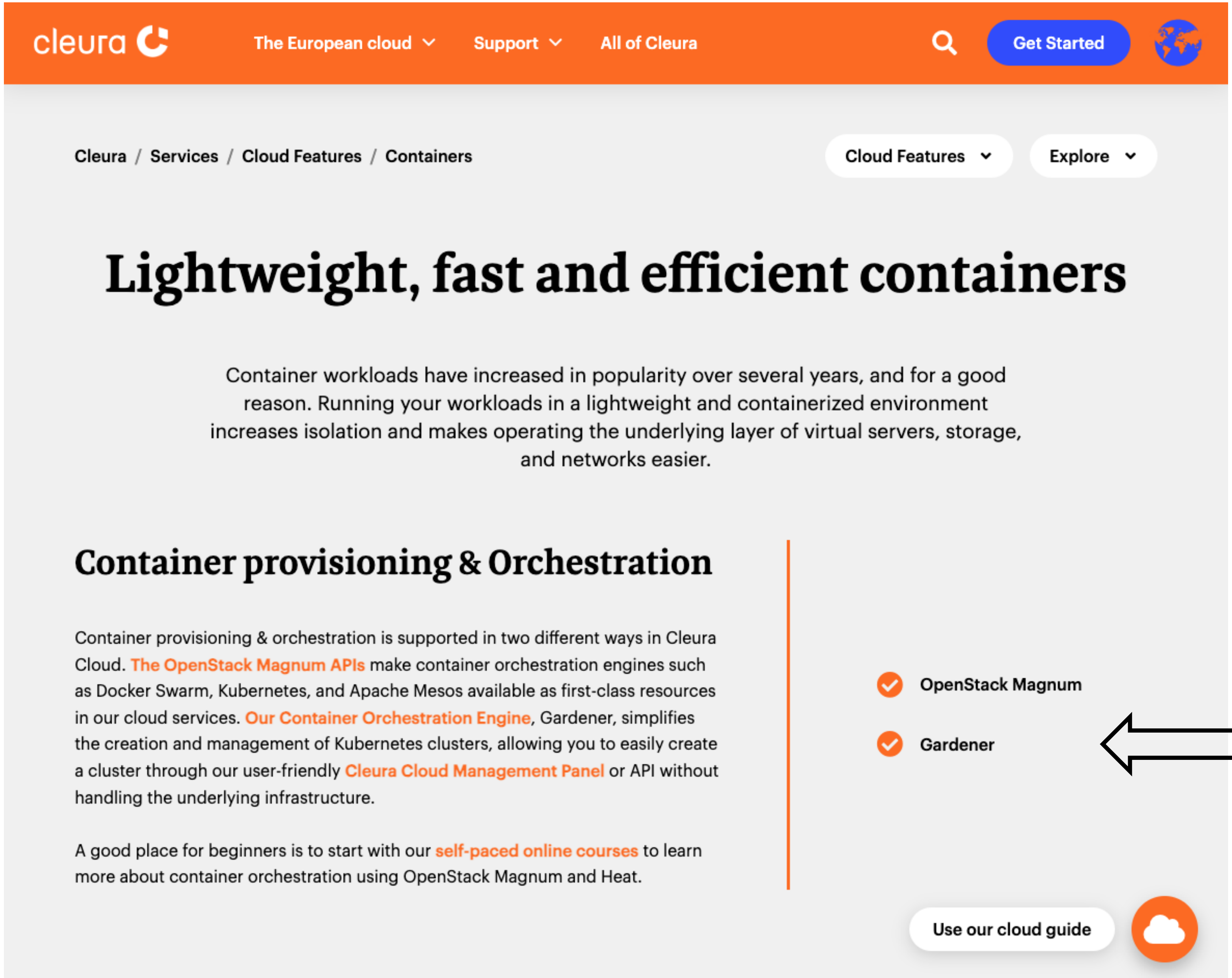
Features you will love about metalstack.cloud

- **Managed**
  Focus on your business and development needs - we take care of good cloud native services like Kubernetes. The components of our stack are

- **Secure**
  Your data stays in the EU and our platform is fully compliant with DSGVO/GDPR. Tenants are strongly separated. If you want to enhance security

- **Field-Proven**
  The underlying technology, metal-stack.io, is completely open source and has been in use at a number of financial institutions for over two years.

- **Opinionated Kubernetes Cloud**
  We have chosen reasonable defaults for Kubernetes clusters that facilitate your start with a Kubernetes provider. metalstack.cloud



Lightweight, fast and efficient containers

Container workloads have increased in popularity over several years, and for a good reason. Running your workloads in a lightweight and containerized environment increases isolation and makes operating the underlying layer of virtual servers, storage, and networks easier.

Container provisioning & Orchestration

Container provisioning & orchestration is supported in two different ways in Cleura Cloud. The OpenStack Magnum APIs make container orchestration engines such as Docker Swarm, Kubernetes, and Apache Mesos available as first-class resources in our cloud services. Our Container Orchestration Engine, Gardener, simplifies the creation and management of Kubernetes clusters, allowing you to easily create a cluster through our user-friendly Cleura Cloud Management Panel or API without handling the underlying infrastructure.

A good place for beginners is to start with our self-paced online courses to learn more about container orchestration using OpenStack Magnum and Heat.

- ✓ OpenStack Magnum
- ✓ Gardener

23 | Technologies

Home Blog Usecases 23REP 23KE 23MKS Jobs DE | EN

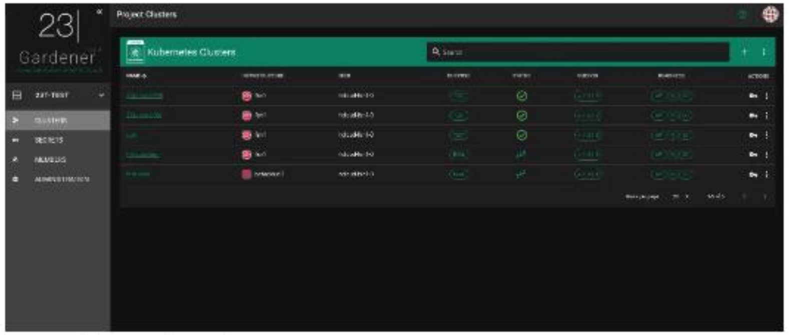
23KE

Enterprise-class Kubernetes Engine

23KE is an enterprise-class Kubernetes engine for industrial use cases as well as cloud service providers. Its strength is its focus on Kubernetes itself. It focuses on scalability, reliability, and self-healing of Kubernetes clusters. No bloatware with tons of features on top that only a few or no one needs and only exist for themselves. This enables 23KE to achieve a high production grade.

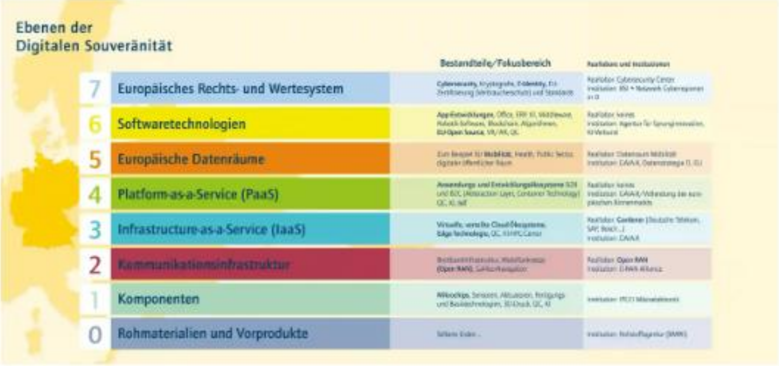
Based on the open source project Gardener it offers Kubernetes Clusters as a Service at scale. With a lot out-of-the-box functionalities for the daily operations routine of the Kubernetes clusters.

Contact us via email to get further information about 23KE.



Source: https://github.com/gardener/dashboard

Digital sovereignty layer model



Ebenen der Digitalen Souveränität

		Bestandteile/Fokusbereich	Realisationen und Institutionen
7	Europäisches Rechts- und Wertesystem	[illegible]	[illegible]
6	Softwaretechnologien	[illegible]	[illegible]
5	Europäische Datenräume	[illegible]	[illegible]
4	Platform-as-a-Service (PaaS)	[illegible]	[illegible]
3	Infrastructure-as-a-Service (IaaS)	[illegible]	Realisation: Gardener [illegible]
2	Kommunikationsinfrastruktur	[illegible]	Realisation: Open RAN [illegible]
1	Komponenten	[illegible]	[illegible]
0	Rohmaterialien und Vorprodukte	[illegible]	[illegible]

Source: https://www.acatech.de/allgemein/digitale-souveraenitaet-acatech-impuls-entwirft-schichtenmodell-als-handlungsrahmen-fuer-die-eu/

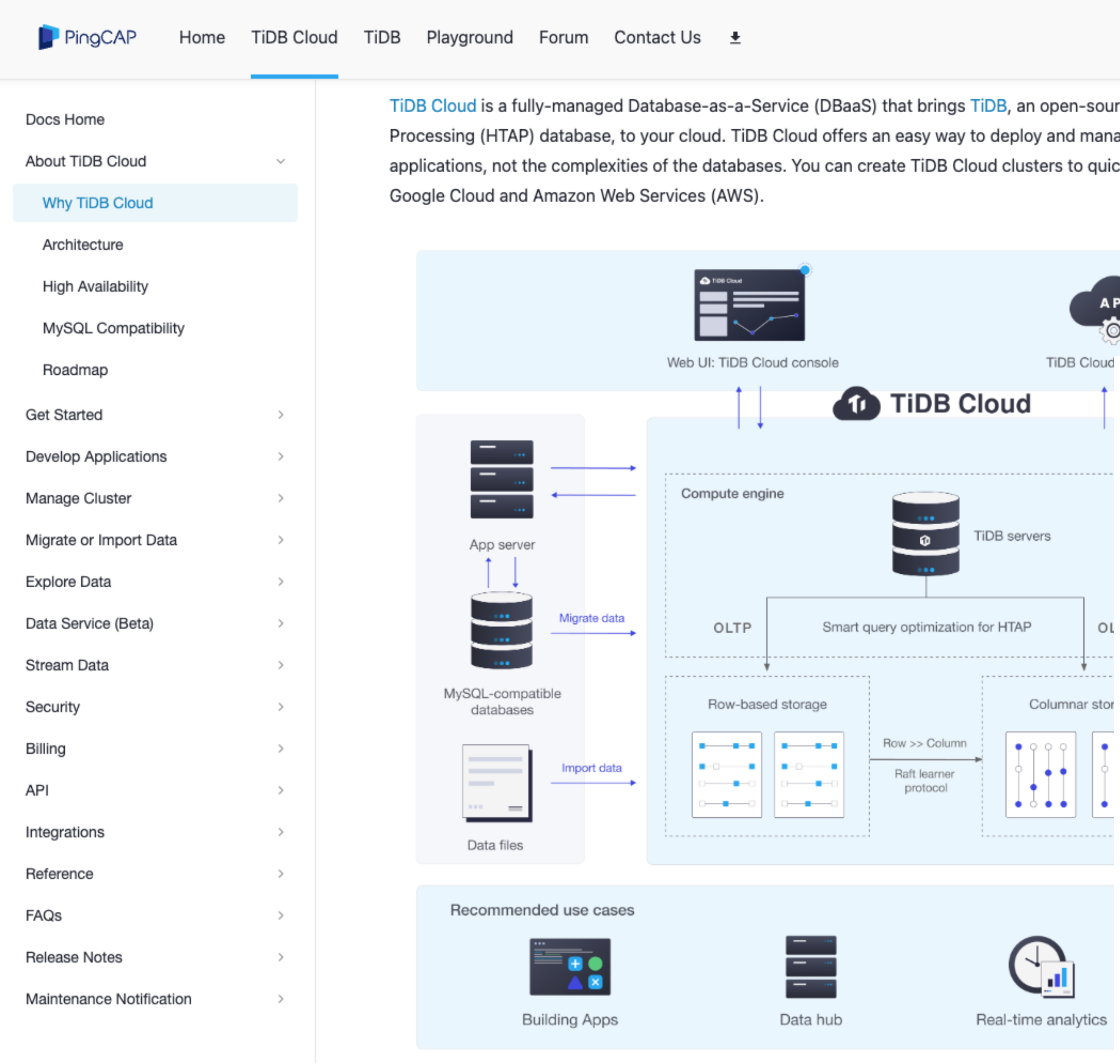
In the document Digitale Souveränität – Status quo und Handlungsfelder by the Deutschen Akademie der Technikwissenschaften (acatech) the Gardener project, the core of 23KE, is listed as a key building block in Gaia-X for the new Infrastructure as a Service layer in the digital sovereignty layer model.

Why do we use Gardener?

codesphere BETA

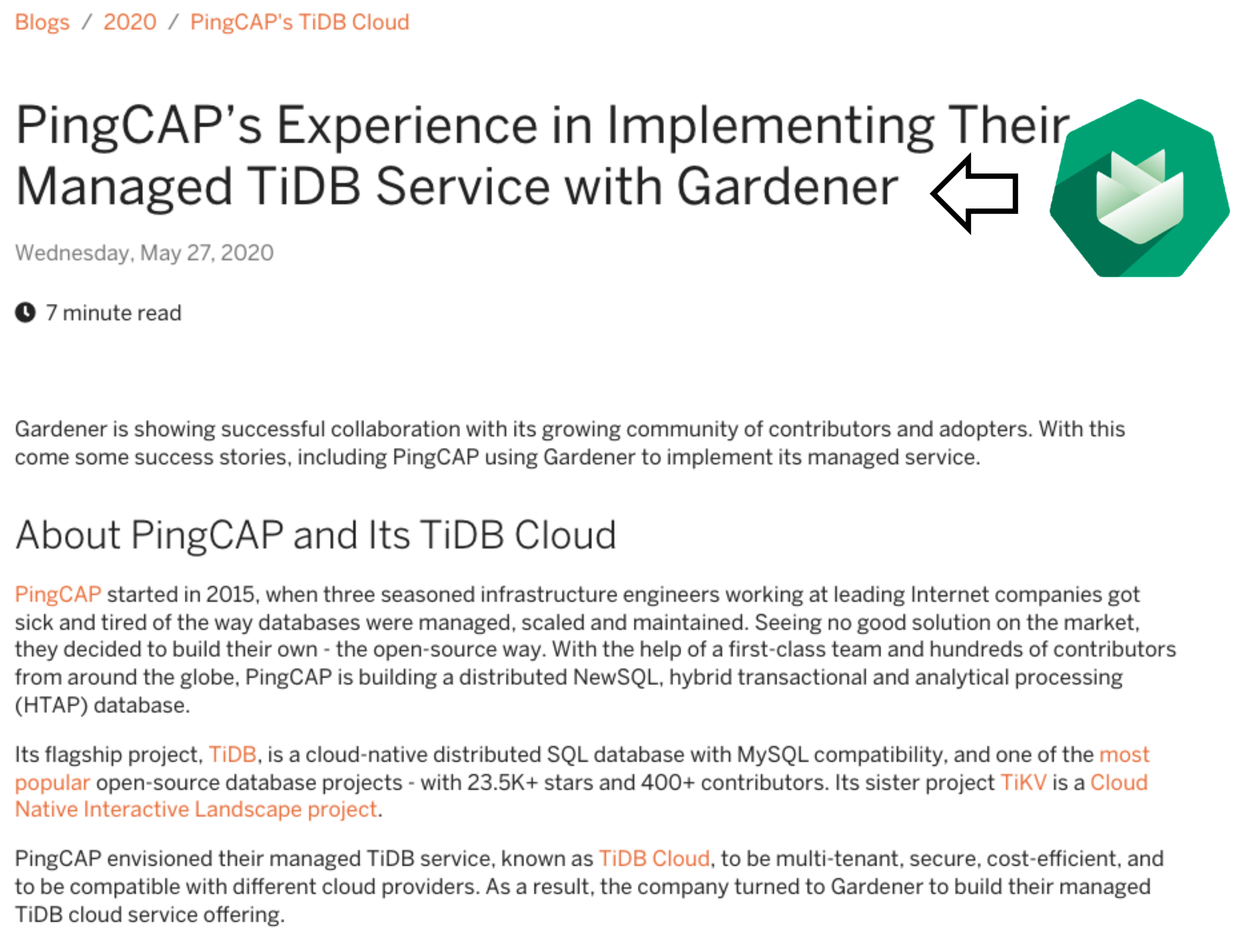
- [illegible]
  - [illegible]
  - [illegible]
  - [illegible]
  - [illegible]
- [illegible]
  - [illegible]
  - [illegible]
- [illegible]
- [illegible]

[illegible]



TiDB Cloud is a fully-managed Database-as-a-Service (DBaaS) that brings TiDB, an open-source Hybrid Transactional and Analytical Processing (HTAP) database, to your cloud. TiDB Cloud offers an easy way to deploy and manage databases to let you focus on your applications, not the complexities of the databases. You can create TiDB Cloud clusters to quickly build mission-critical applications on Google Cloud and Amazon Web Services (AWS).

TiDB Cloud
Fully Managed TiDB as a Service



Blogs / 2020 / PingCAP's TiDB Cloud

PingCAP's Experience in Implementing Their Managed TiDB Service with Gardener

Wednesday, May 27, 2020

7 minute read

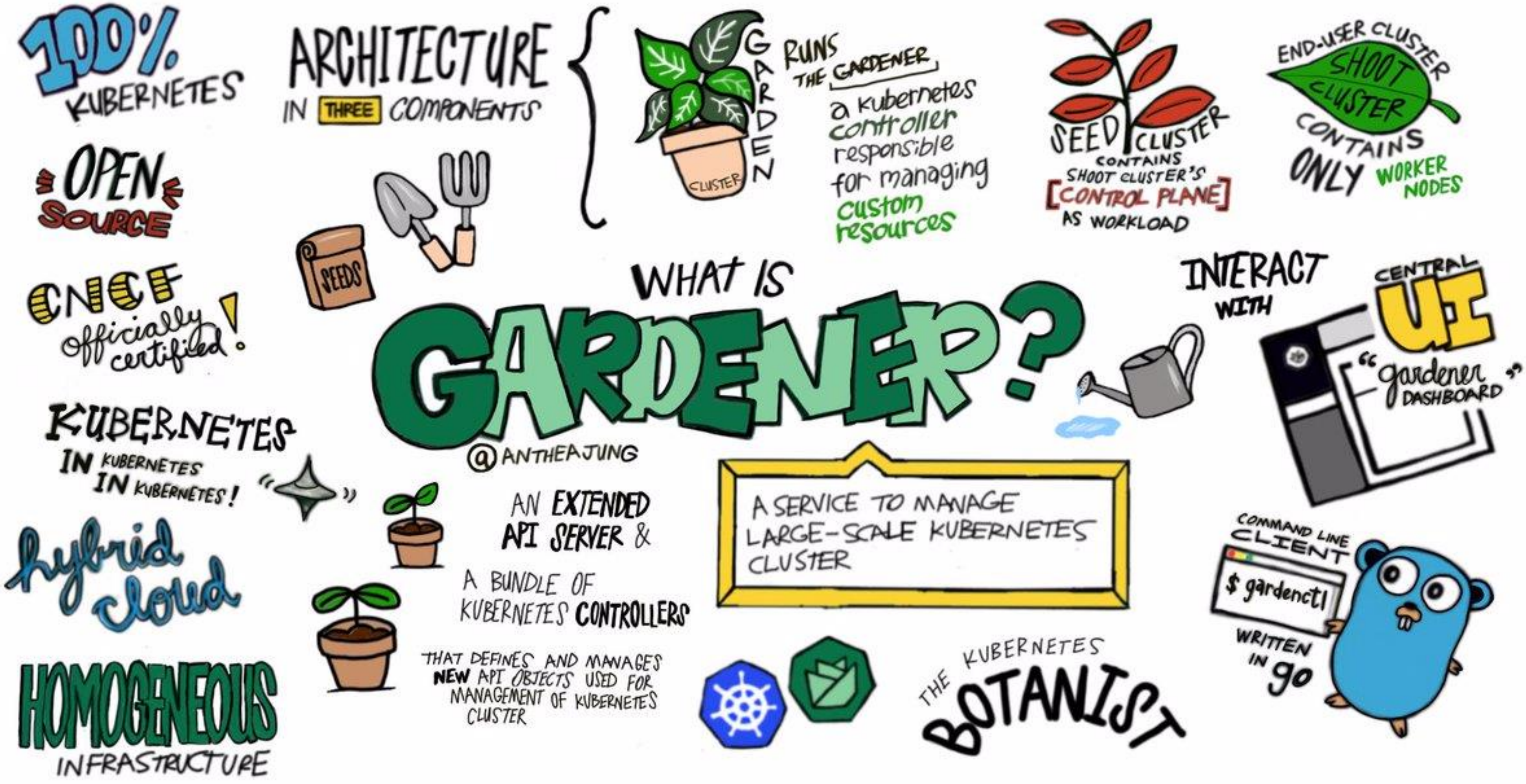
Gardener is showing successful collaboration with its growing community of contributors and adopters. With this come some success stories, including PingCAP using Gardener to implement its managed service.

About PingCAP and Its TiDB Cloud

PingCAP started in 2015, when three seasoned infrastructure engineers working at leading Internet companies got sick and tired of the way databases were managed, scaled and maintained. Seeing no good solution on the market, they decided to build their own - the open-source way. With the help of a first-class team and hundreds of contributors from around the globe, PingCAP is building a distributed NewSQL, hybrid transactional and analytical processing (HTAP) database.

Its flagship project, TiDB, is a cloud-native distributed SQL database with MySQL compatibility, and one of the most popular open-source database projects - with 23.5K+ stars and 400+ contributors. Its sister project TiKV is a Cloud Native Interactive Landscape project.

PingCAP envisioned their managed TiDB service, known as TiDB Cloud, to be multi-tenant, secure, cost-efficient, and to be compatible with different cloud providers. As a result, the company turned to Gardener to build their managed TiDB cloud service offering.



WHAT IS GARDENER?

@ANTHEAJUNG

100% KUBERNETES

OPEN SOURCE

CNCF officially certified!

KUBERNETES IN KUBERNETES IN KUBERNETES!

hybrid cloud

HOMOGENEOUS INFRASTRUCTURE

ARCHITECTURE IN THREE COMPONENTS

GARDEN CLUSTER RUNS THE GARDENER

a kubernetes controller responsible for managing custom resources

SEED CLUSTER CONTAINS SHOOT CLUSTER'S [CONTROL PLANE] AS WORKLOAD

END-USER CLUSTER SHOOT CLUSTER CONTAINS ONLY WORKER NODES

A SERVICE TO MANAGE LARGE-SCALE KUBERNETES CLUSTER

AN EXTENDED API SERVER &

A BUNDLE OF KUBERNETES CONTROLLERS

THAT DEFINES AND MANAGES NEW API OBJECTS USED FOR MANAGEMENT OF KUBERNETES CLUSTER

THE KUBERNETES BOTANIST

INTERACT WITH CENTRAL UI "gardener DASHBOARD"

COMMAND LINE CLIENT

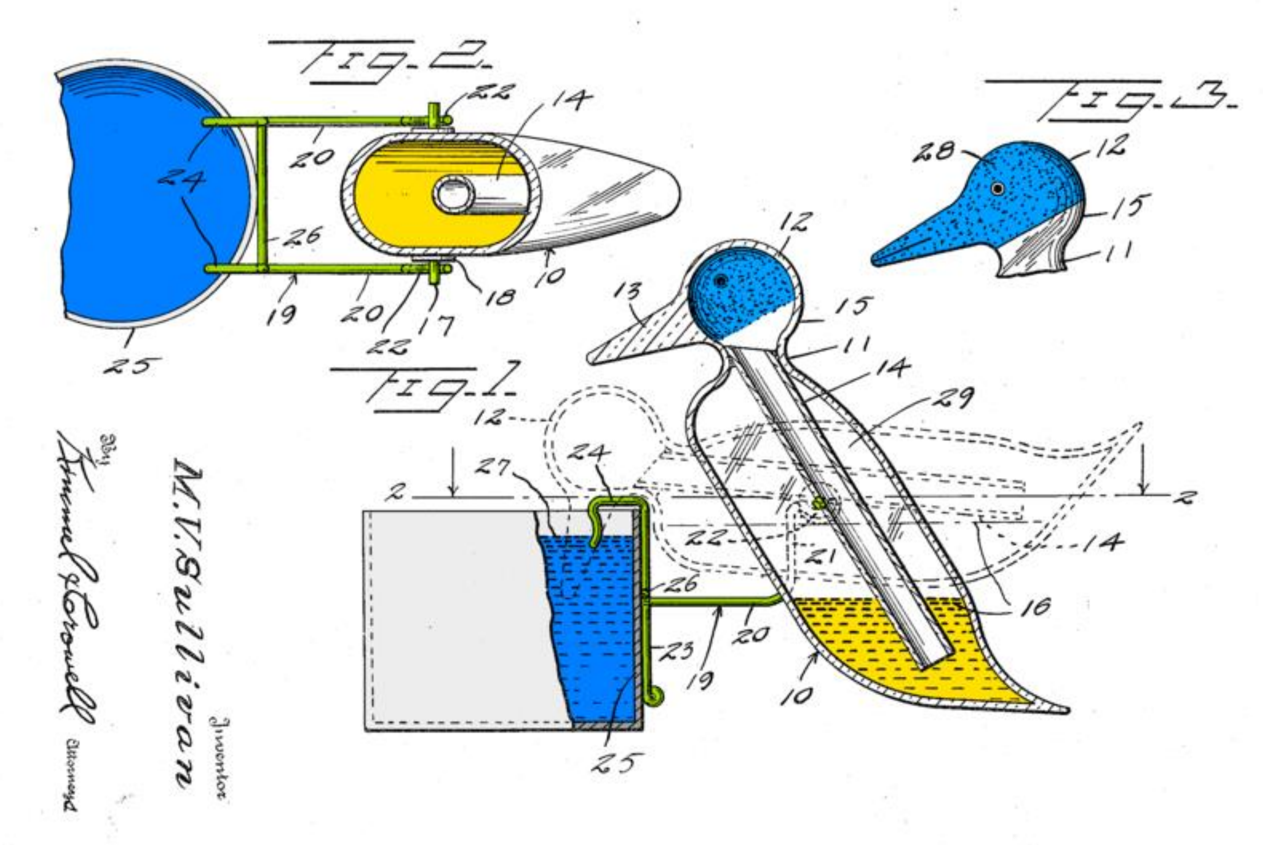
$ gardenctl

WRITTEN IN go

Gardener: An integrated solution with focus on homogeneity, re-using Kubernetes core principles to produce resilient systems at scale

How does the process of creating a Kubernetes cluster using Gardener compare to using Cluster-API?

"In conclusion, both Gardener and Cluster-API provide robust solutions for creating and managing Kubernetes clusters. The choice between the two depends on your specific needs and requirements. **Gardener provides a more integrated solution with a focus on homogeneity across different infrastructures**, while **Cluster-API** provides a more flexible and extensible solution that **can be customized to suit different infrastructure providers** and environments."



https://en.wikipedia.org/wiki/Drinking_bird