Temas de Hoy

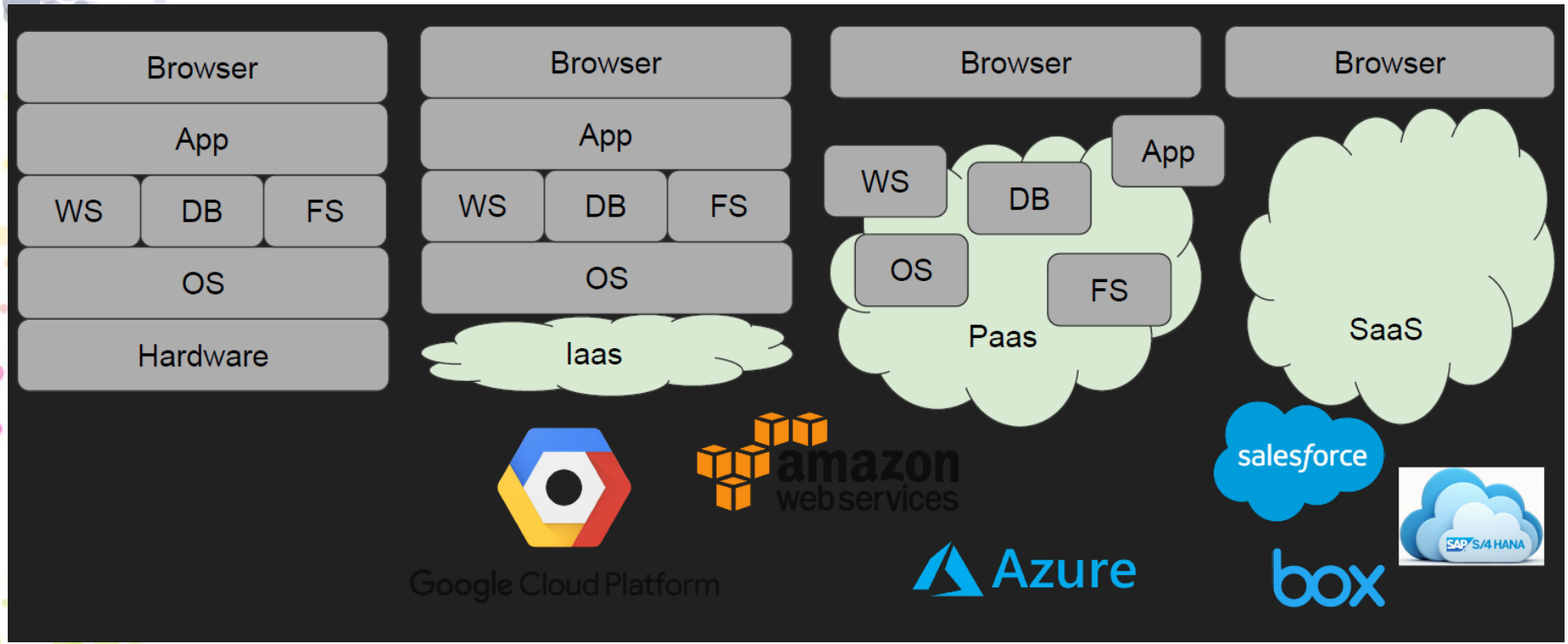
Microservicios

Desarrollos Móviles

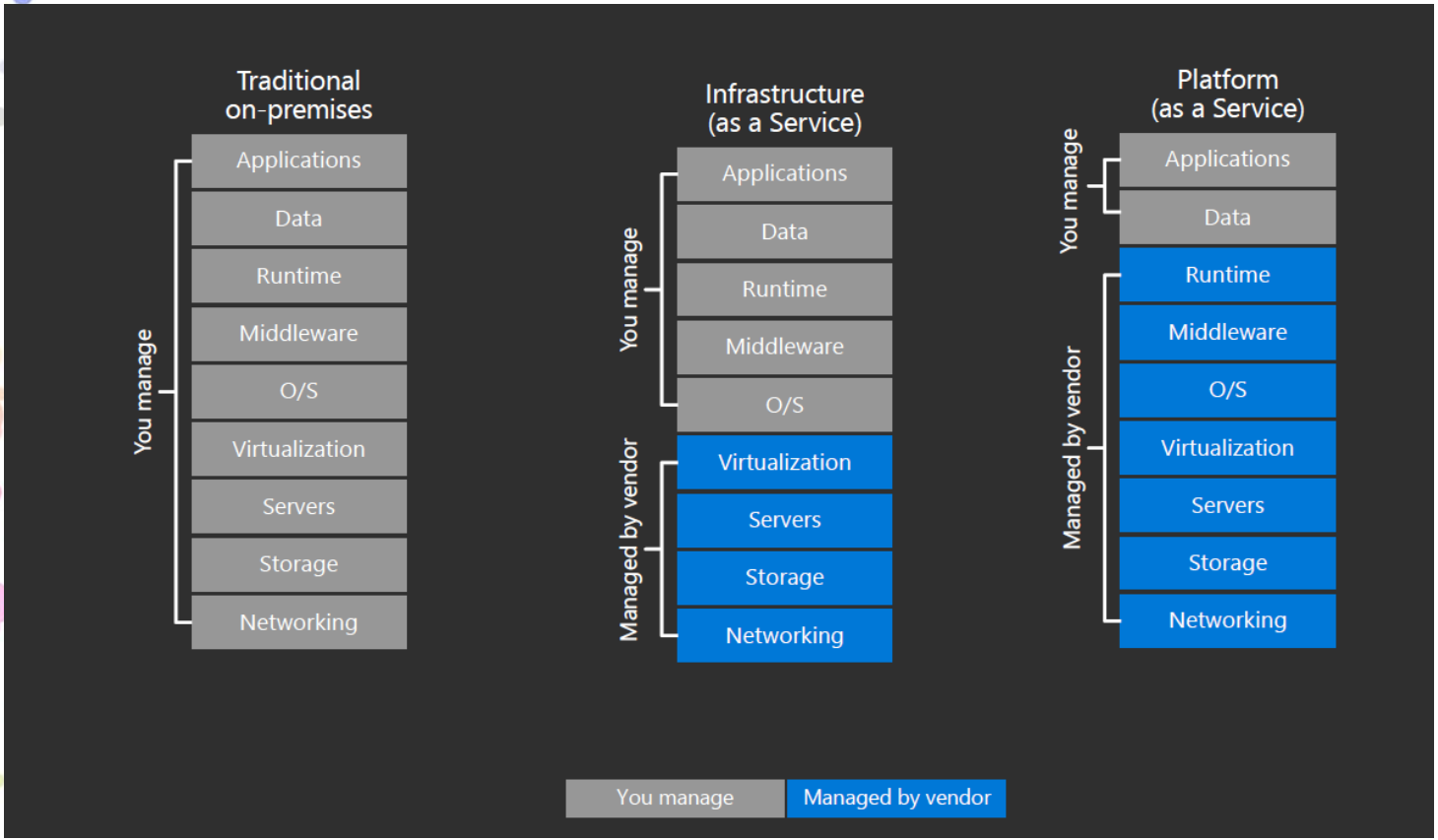
IaaS, PaaS, SaaS

Frontend & Backend

IT → IaaS, PaaS, SaaS



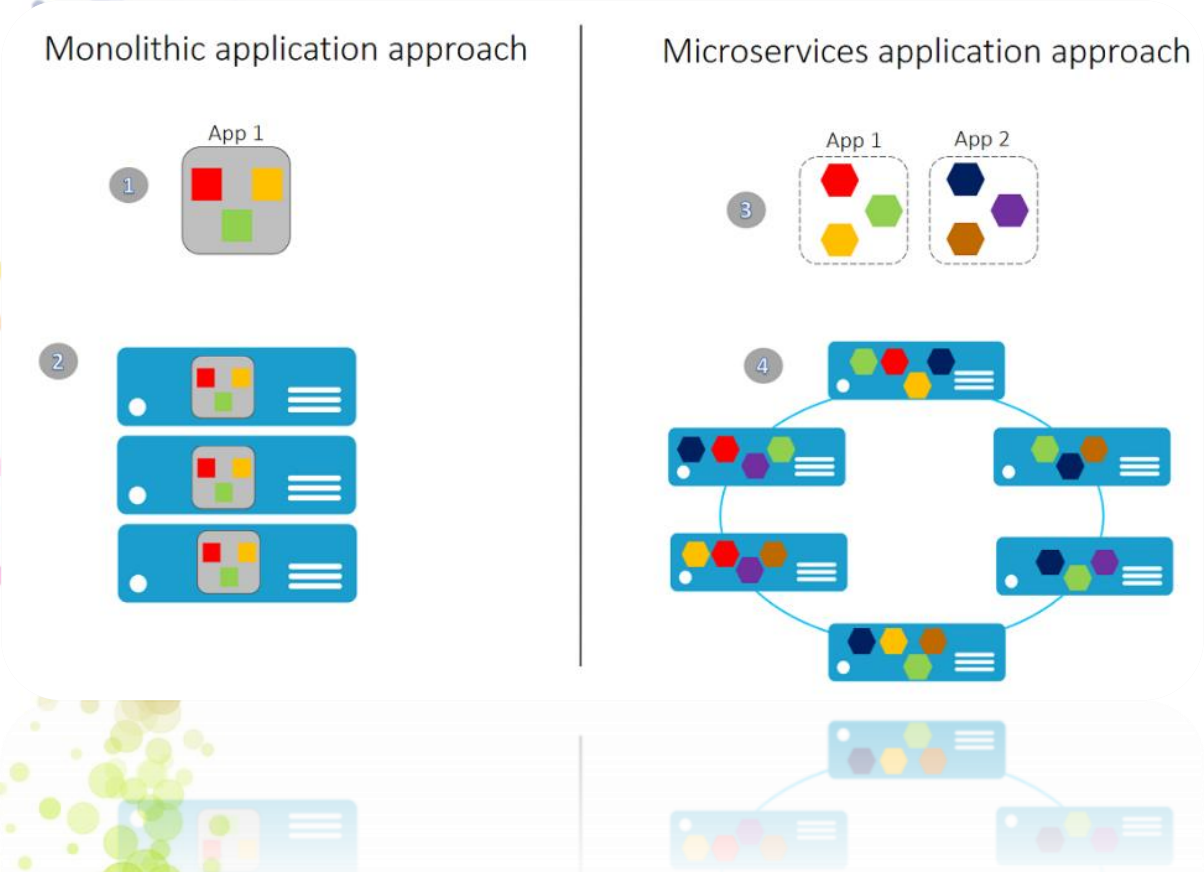
On premises vs Cloud



Microservicios: ¿Qué son?

Las aplicaciones de microservicios se componen de servicios pequeños centrados en el cliente, escalables y con control de versiones independiente que se comunican entre sí mediante protocolos estándar con interfaces bien definidas.

Por dónde empezamos…



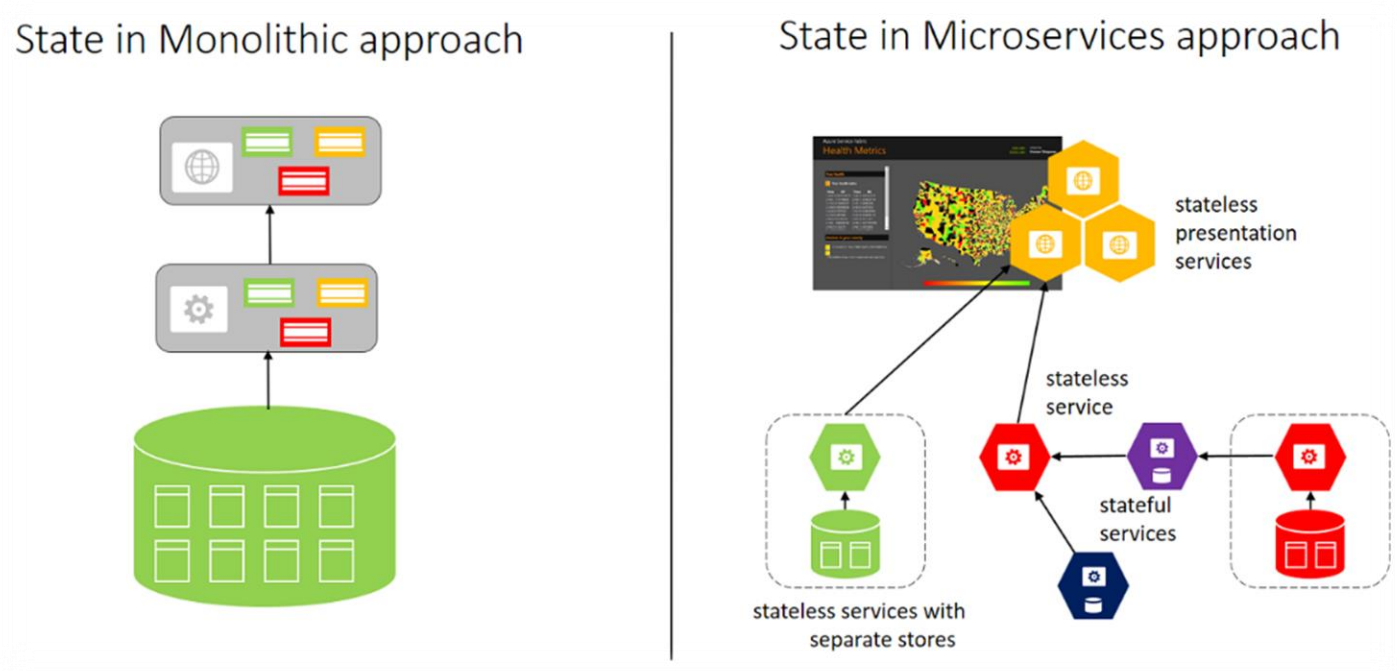
1) Las aplicaciones monolíticas contienen funciones específicas del dominio y normal mente se dividen en capas funcionales, como Web, negocios y datos.

2) Para escalar aplicaciones monolíticas, es preciso clonarlas en varios servidores, máquinas virtuales o contenedores.

3) Las aplicaciones de microservicios separan las funciones en servicios más pequeños independientes.

4) Este enfoque de microservicios se escala horizontalmente mediante la implementación de cada servicio de manera independiente, con la creación de instancias de estos servicios en servidores , máquinas virtuales y contenedores.

Estado



El enfoque monolítico de la izquierda una base de datos única y niveles de tecnologías específicas.

El enfoque de microservicios tiene un gráfico de microservicios interconectados donde el estado normalmente tiene como ámbito el microservicio y se usan diversas tecnologías.

Evolución

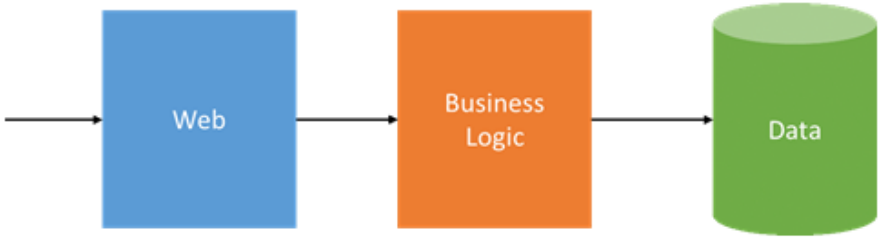


Figure 1. Three-Tier Monolithic Application

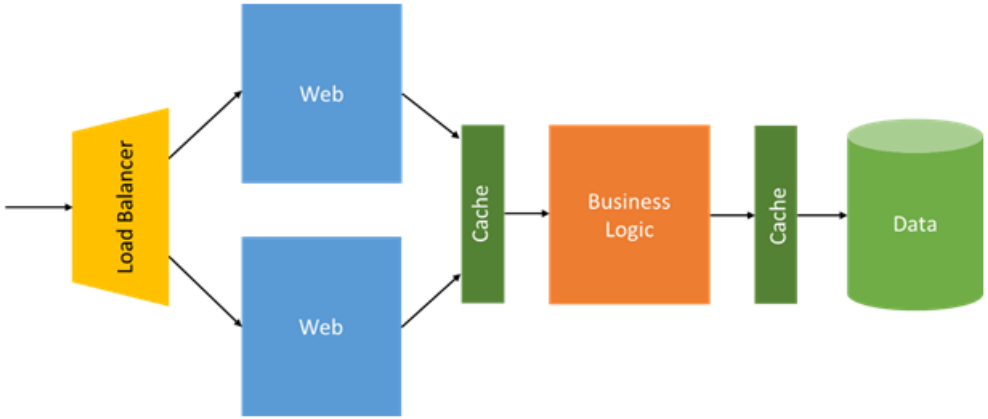


Figure 2. Three-Tier Monolithic Application with Caches

Hacia los microservicios…

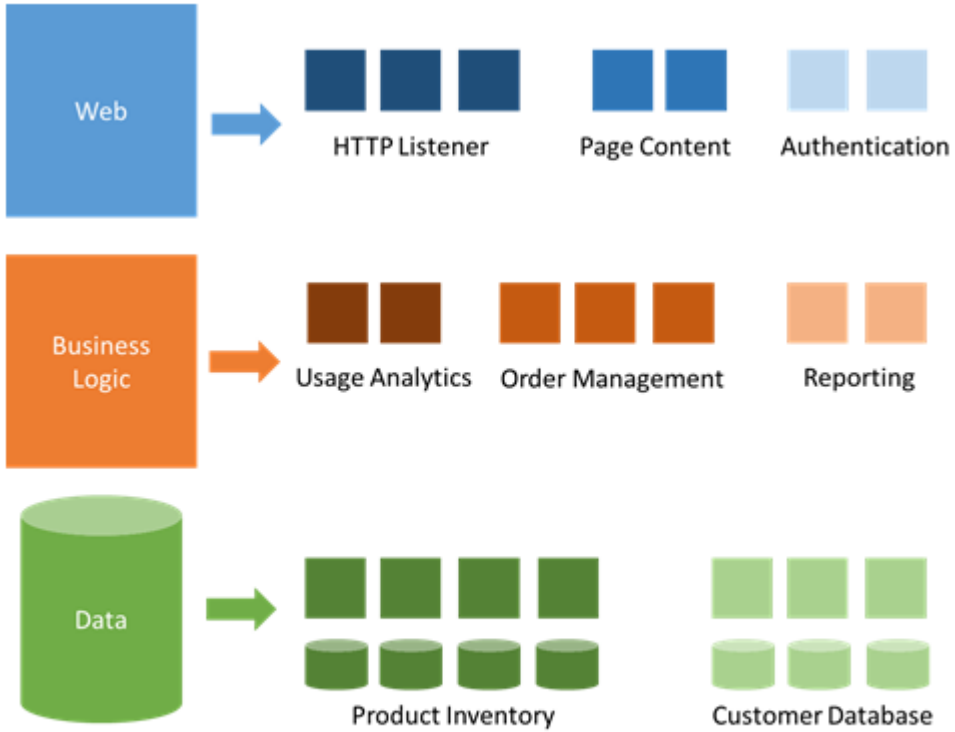


Figure 3. Breaking the Monolith into Microservices

Deploy de Microservicios

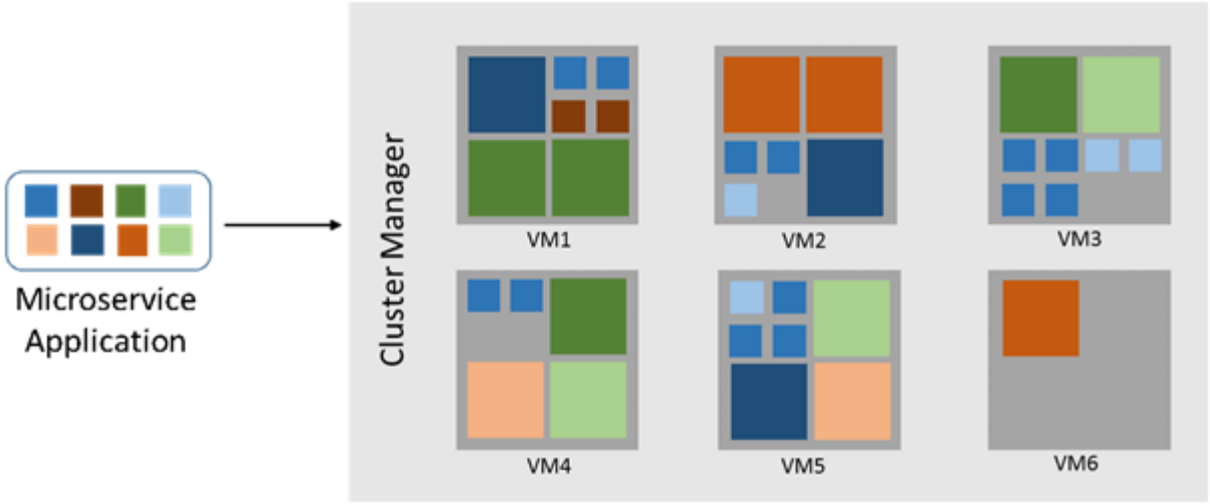


Figure 4. Cluster of Servers with Deployed Microservices

Plataforma de Microservicios

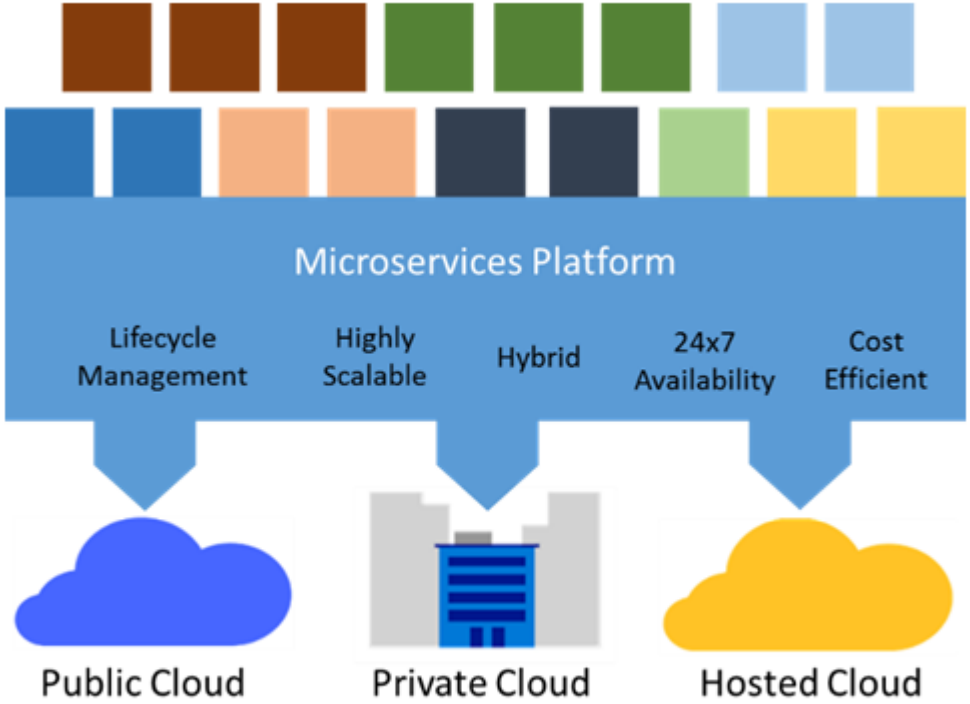


Figure 5. Microservices Platform

Microservicios y SOA

Integrar diferentes aplicaciones como Un conjunto de servicios

SOA

Diseñar una aplicación como un conjunto de servicios

MICROSERVICIOS

SOA…

Classic SOA

- **Integrates** different applications as a set of services

Typical implementation solution

- Heavy-weight
- Orchestration
- ESB
- WS*/SOAP
- License-driven
- Intelligent Communication Layer

Target Problem

- Integrate (Legacy) Software

Microservicios

MICROSERVICE

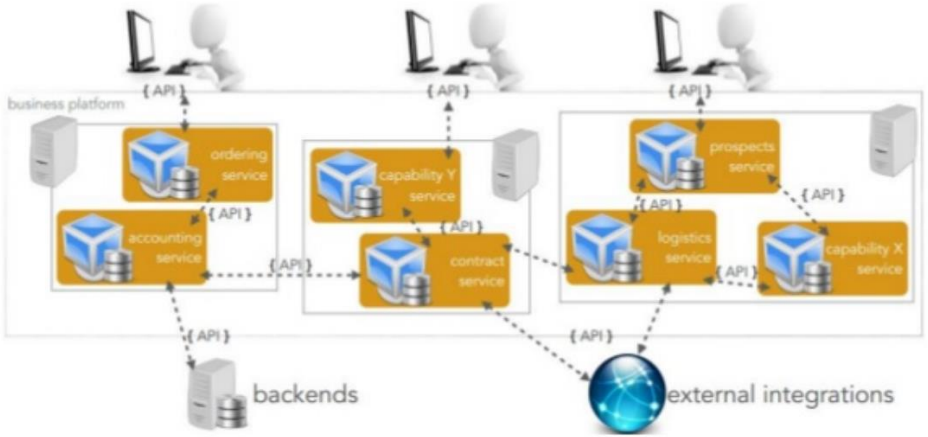
- **Architect** a single application as a set of services

Typical implementation solution

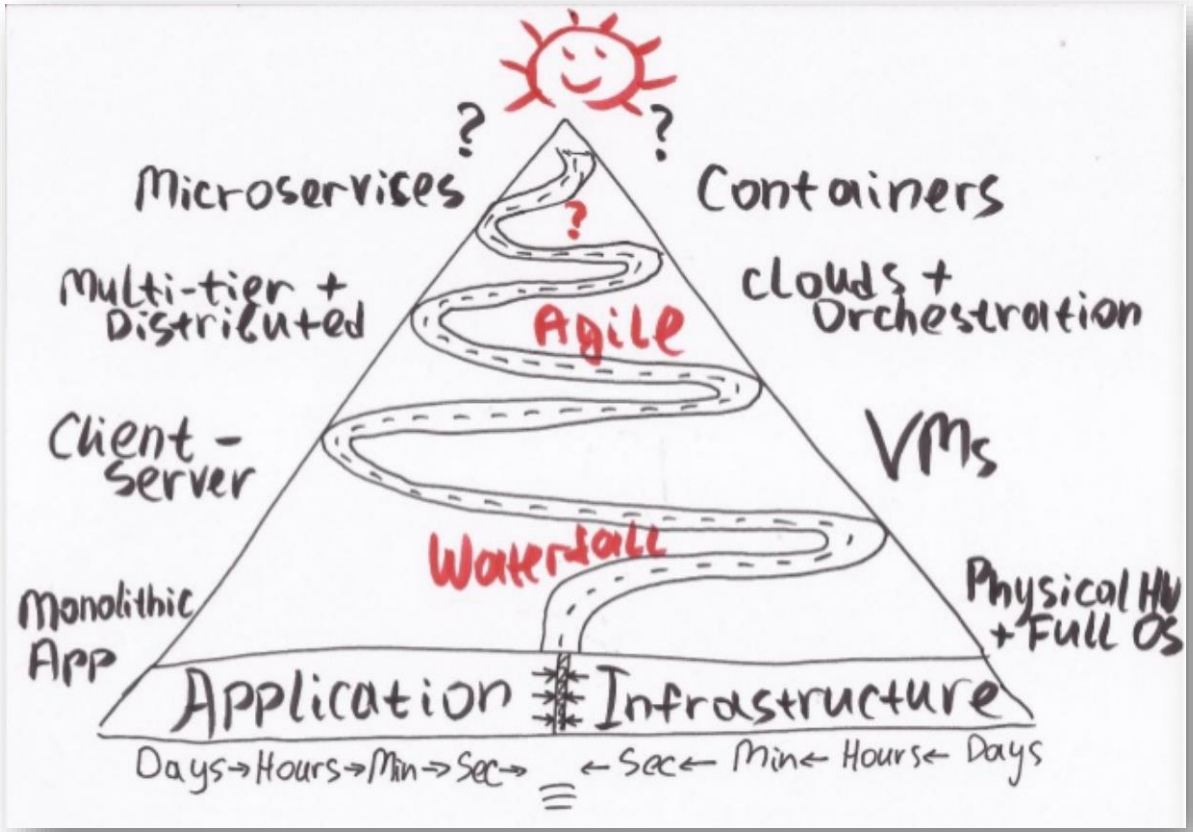
- Light-weight
- Choreography
- HTTP/REST/JSON
- Intelligent Services
- Dumb Communication Layer

Target Problem

- Architect new Business Platform



¿Por qué?


Microservices
Containers
Multi-tier + Distributed
clouds + Orchestration
Agile
Client - Server
VMs
Waterfall
Monolithic App
Physical HW + Full OS
Application
Infrastructure
Days→Hours→Min→Sec→
←Sec←Min←Hours←Days

Escalamiento

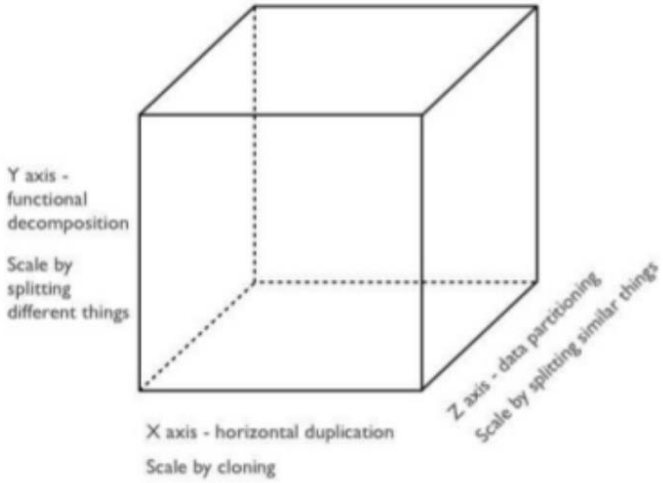
- The Scale Cube

X-AXIS SCALING

Network name: Horizontal scaling, scale out

Y-AXIS SCALING

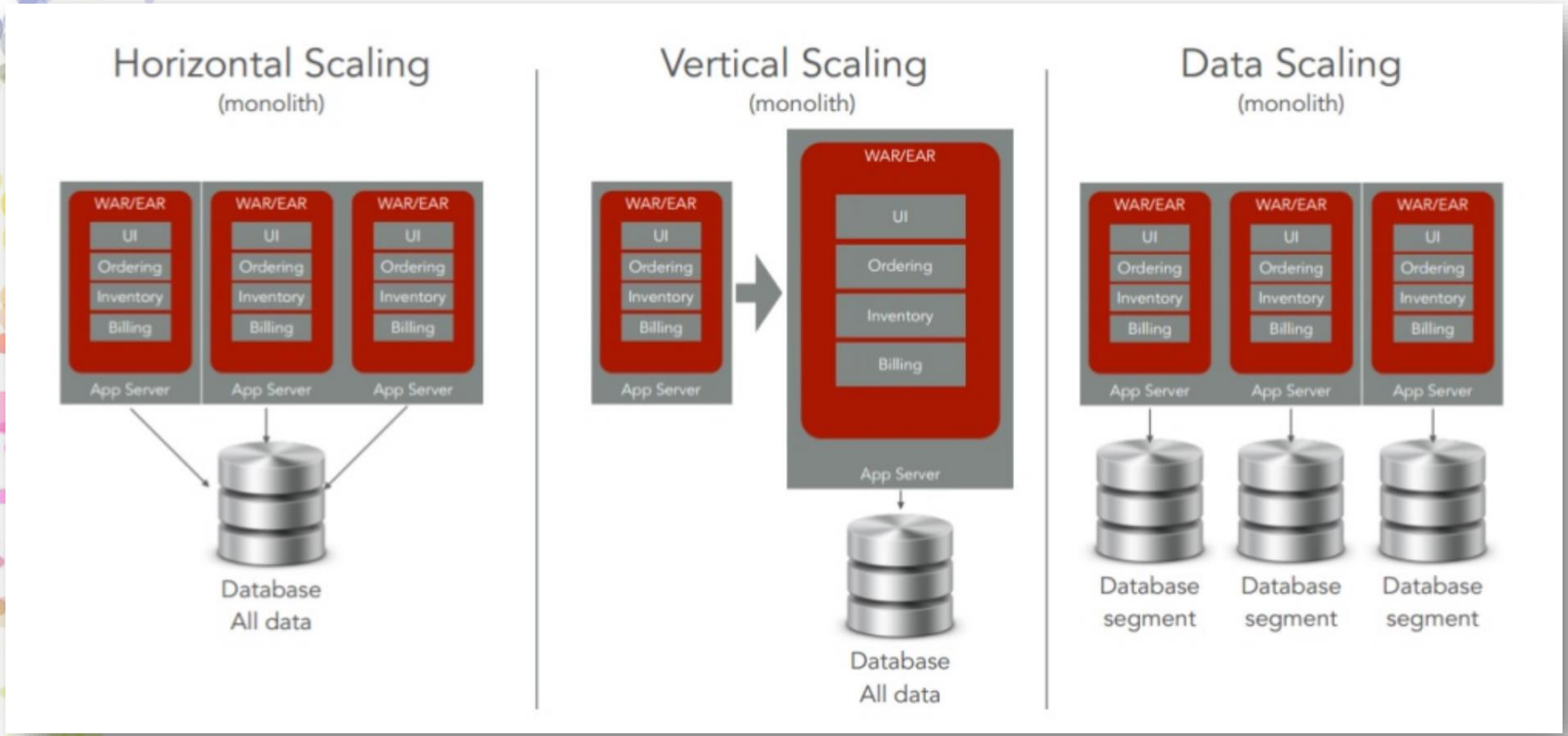
Network name: Layer 7 Load Balancing, Content switching, HTTP Message Steering



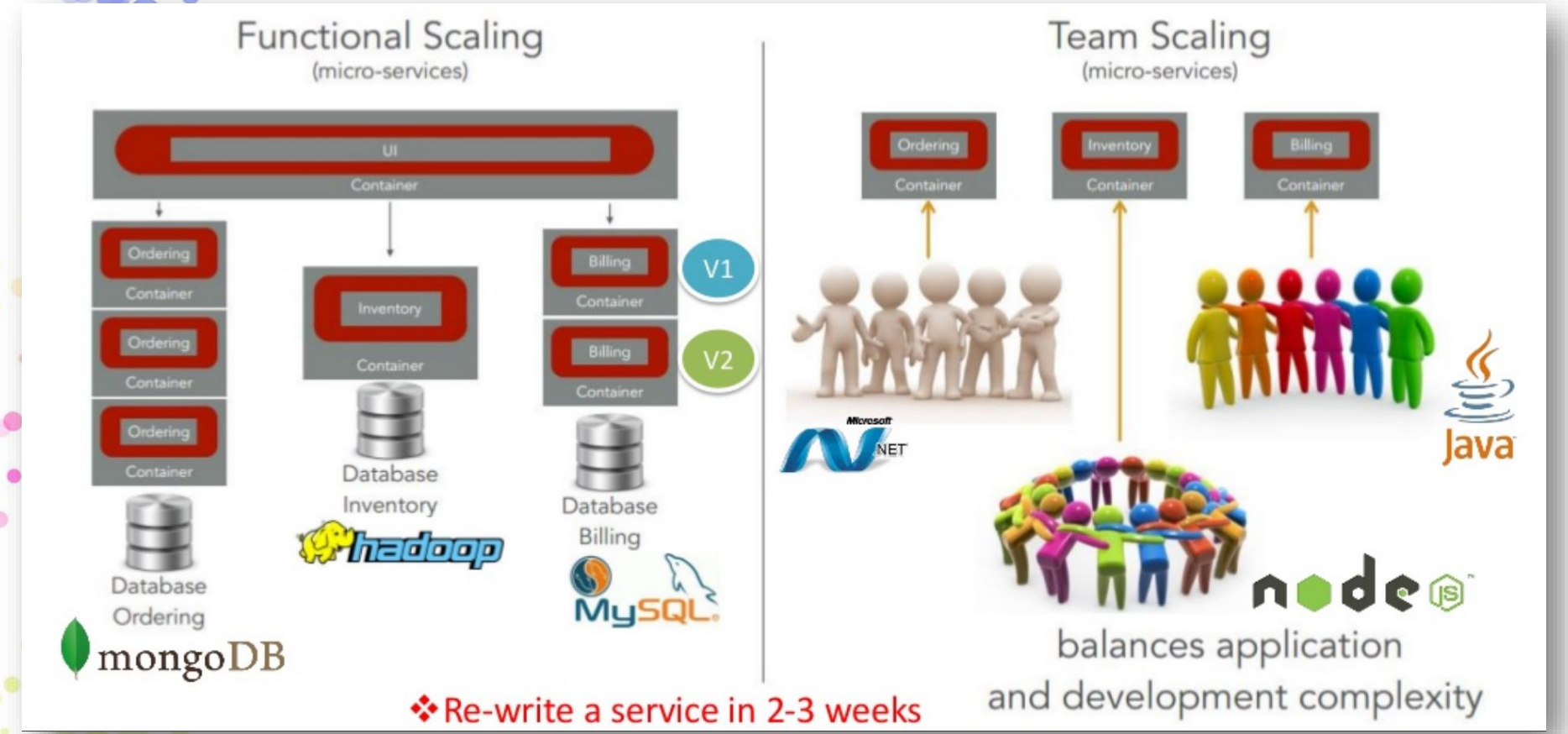
Z-AXIS SCALING

Network name: Layer 7 Load Balancing, Content switching, HTTP Message Steering

Escalamiento



Escalamiento



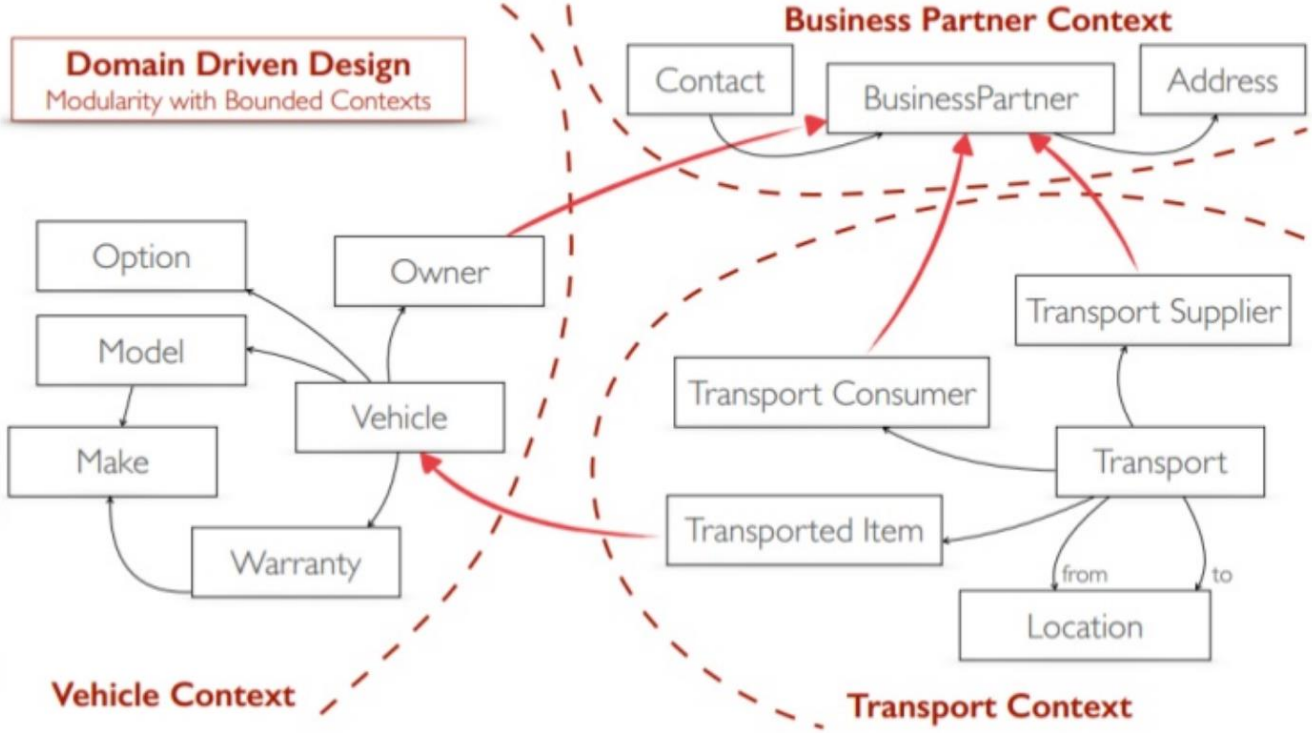
Microservicios: ¿Cómo?

Principios de los Microservicios

1. Modelled Around Business Domain

Benefits

- ✓ Domain modules can **migrate independently** to next version!
- ✓ **Database schema** is **internal** to the domain bundle, i.e. **NOT part of the API!**
- ✓ Persistence and/or database **technology can differ between modules** in one system!



Principios de los Microservicios

2. Culture Of Automation

2 Microservices

3 Months

10 Microservices

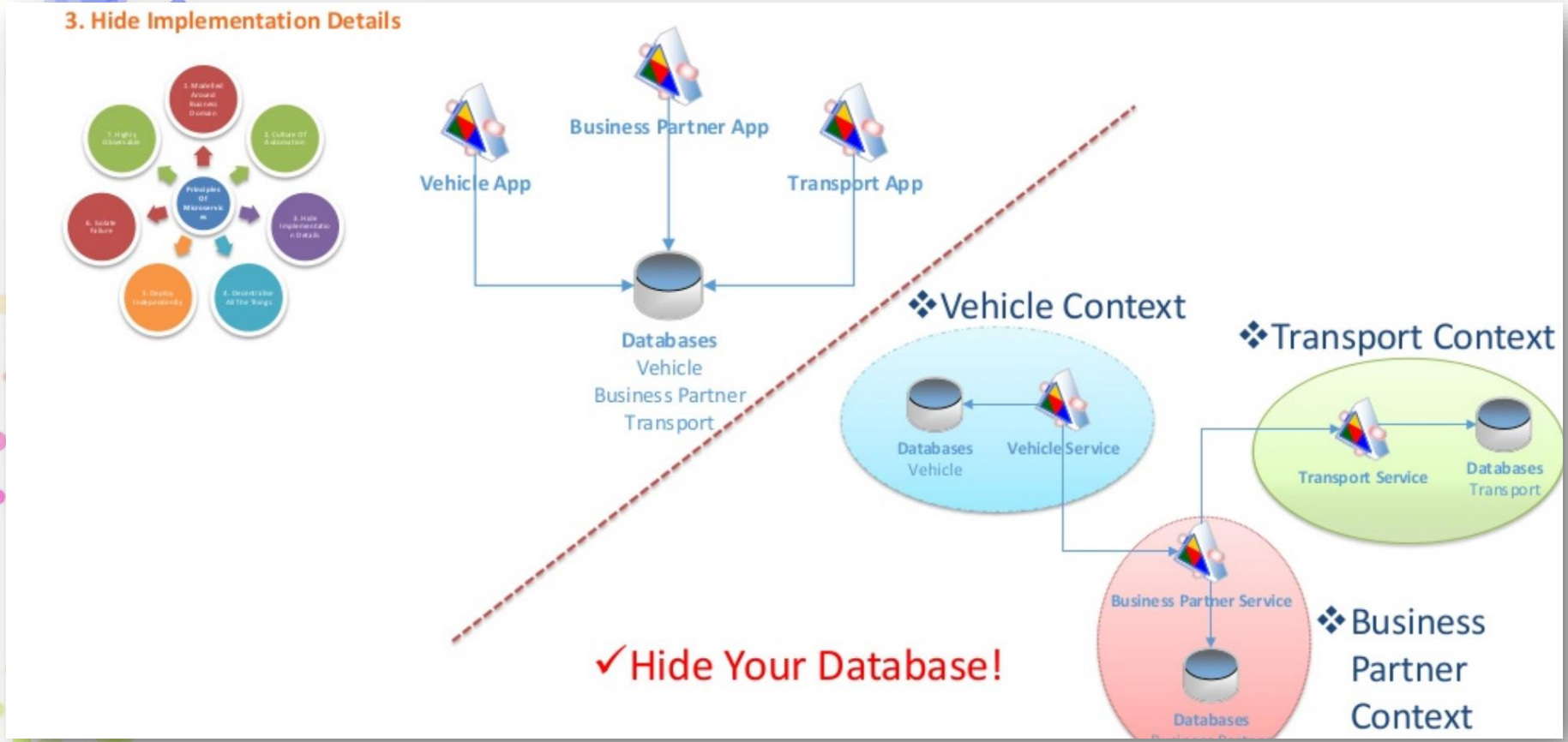
12 Months

60 Microservices

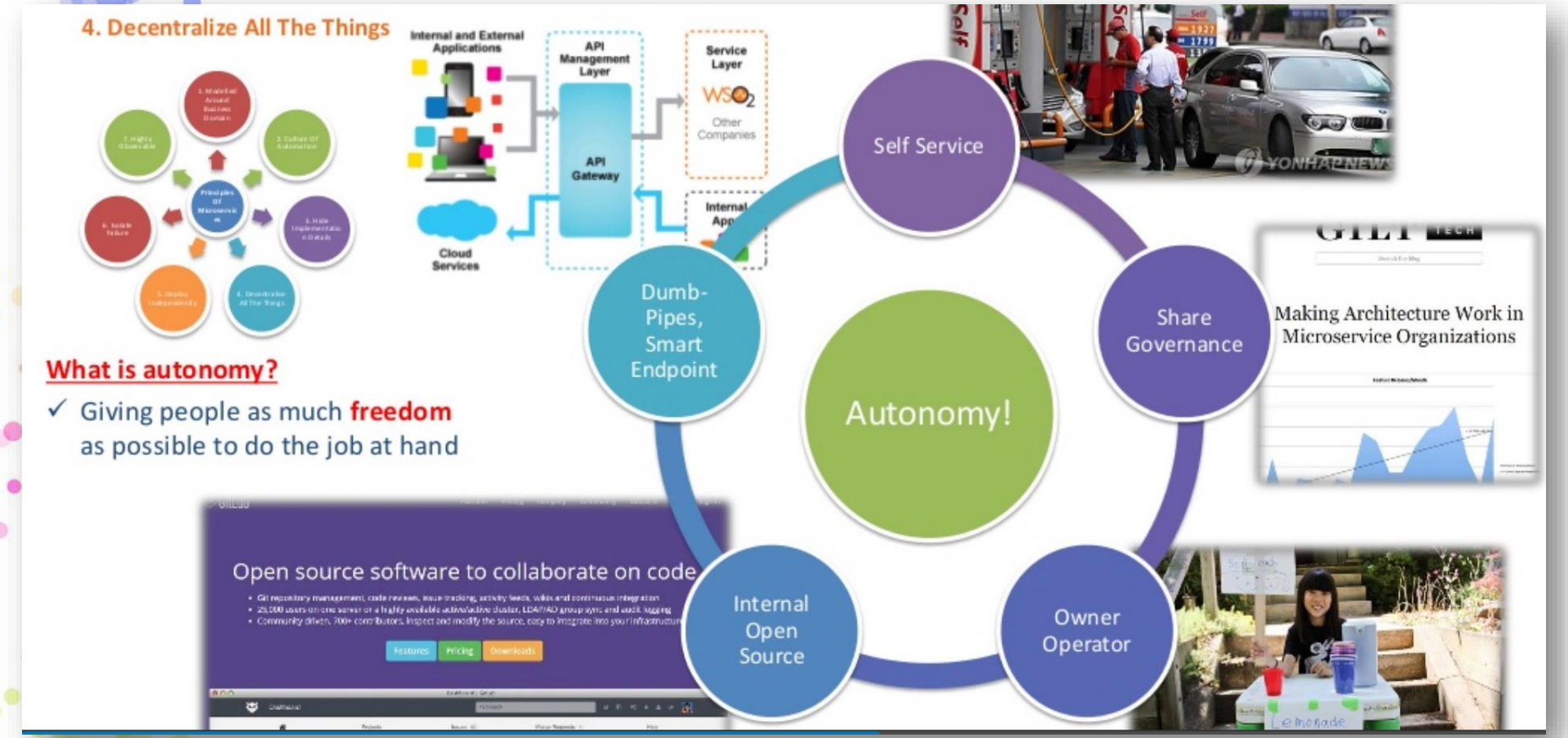
18 Months

- Infrastructure Automation
- Automated Testing
- Continuous Delivery!

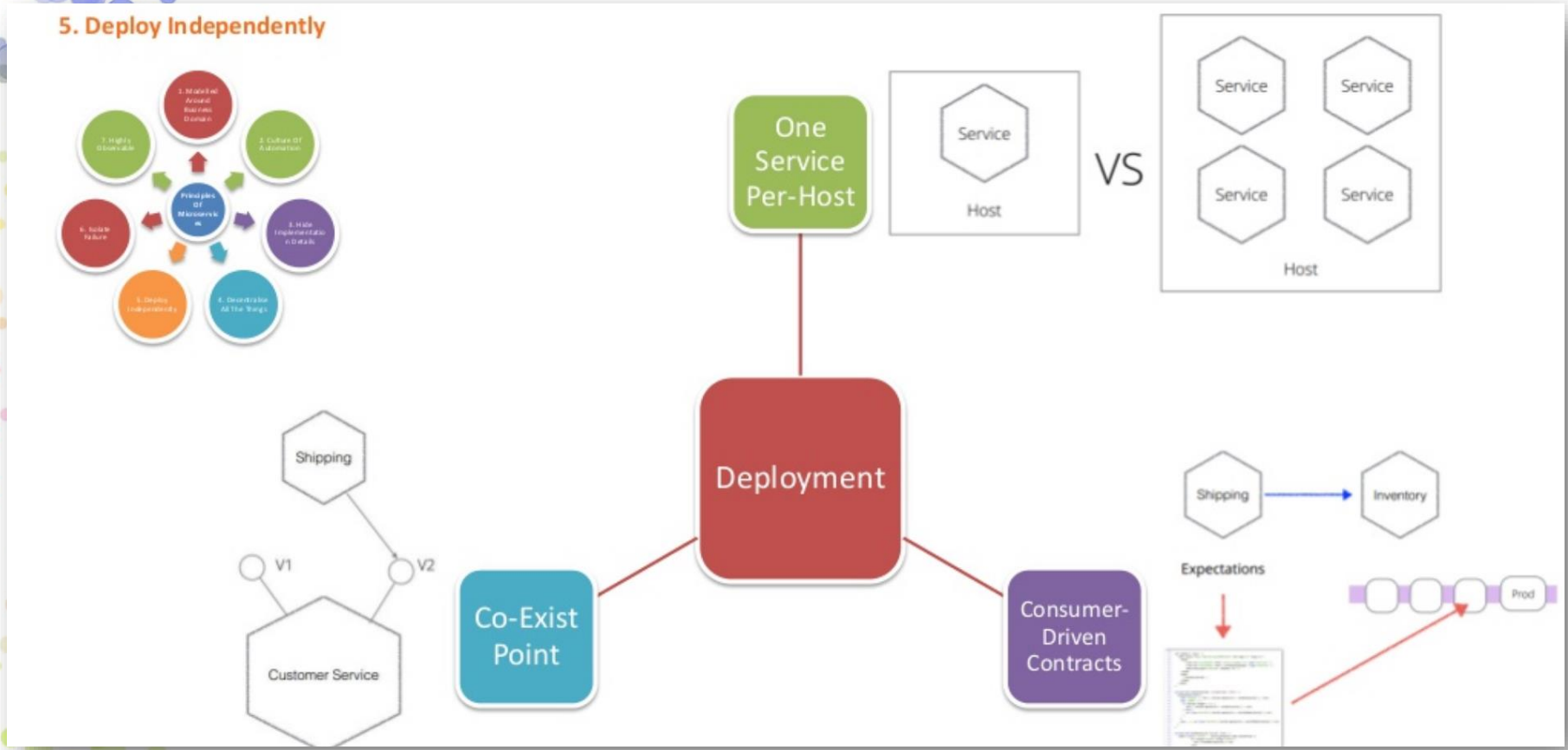
Principios de los Microservicios



Principios de los Microservicios

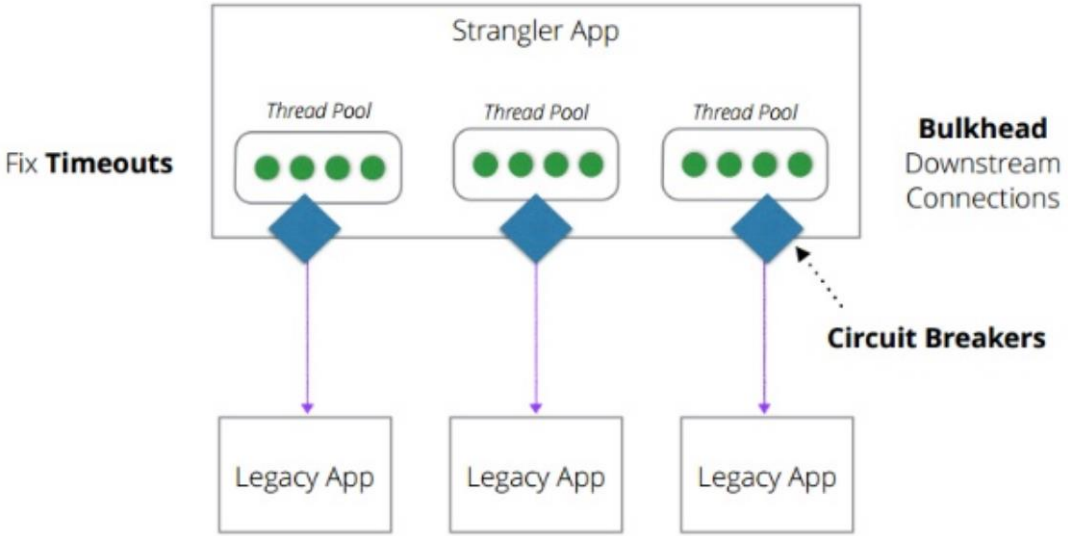


Principios de los Microservicios



Principios de los Microservicios

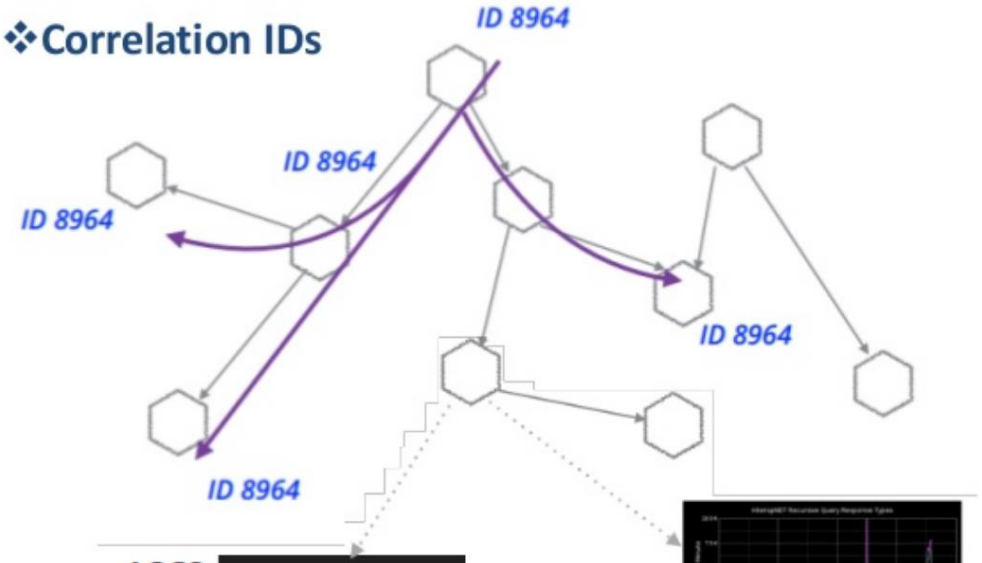
6. Isolate Failure



Principios de los Microservicios

7. Highly Observable

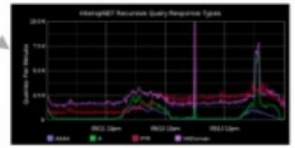
❖Correlation IDs



LOGS

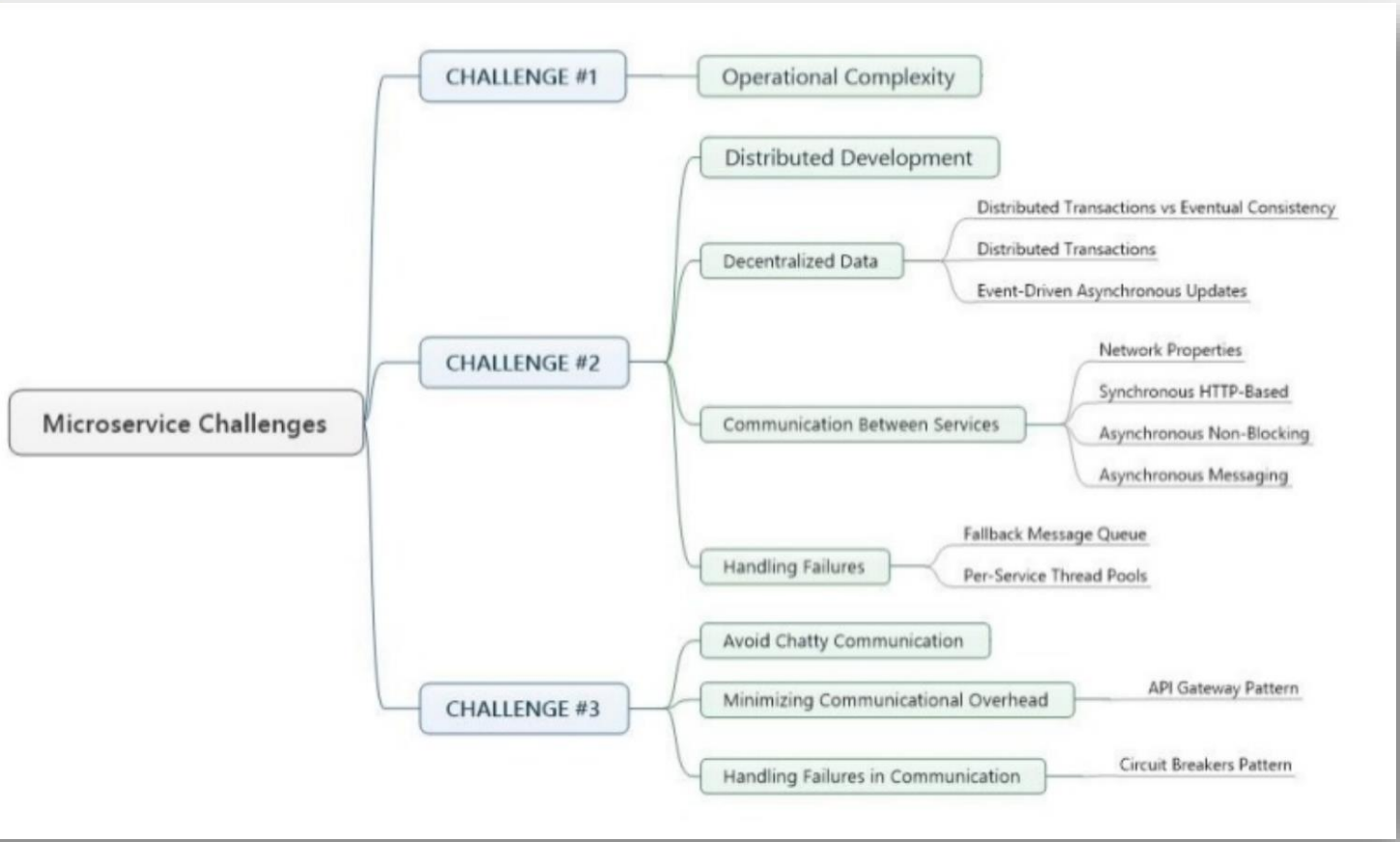
kibana

STATS



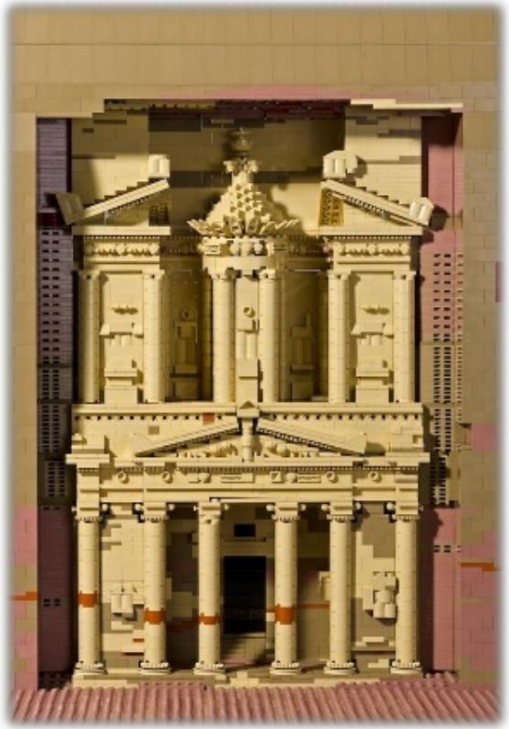
❖Aggregation

Desafíos de los Microservicios

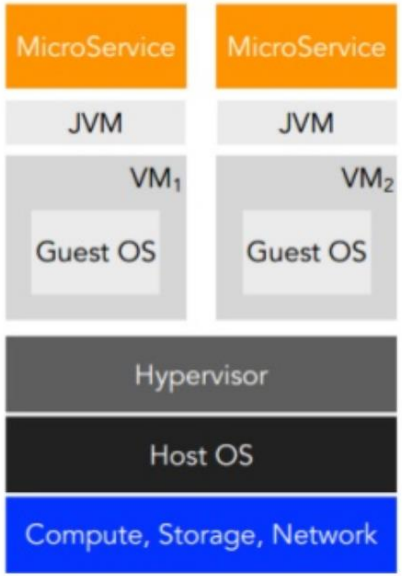


Deploy

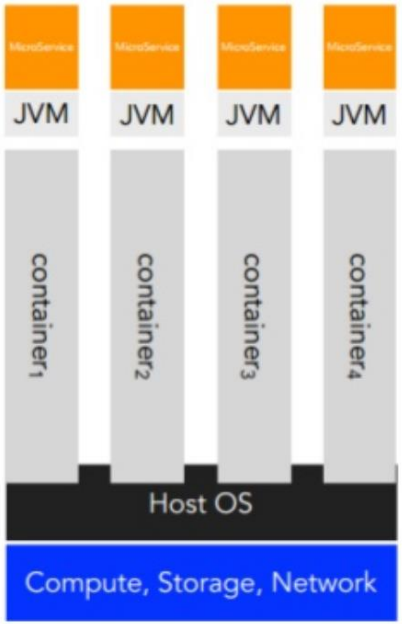
Deployment Options



VM's abstract underlying hardware, but limit resource utilisation

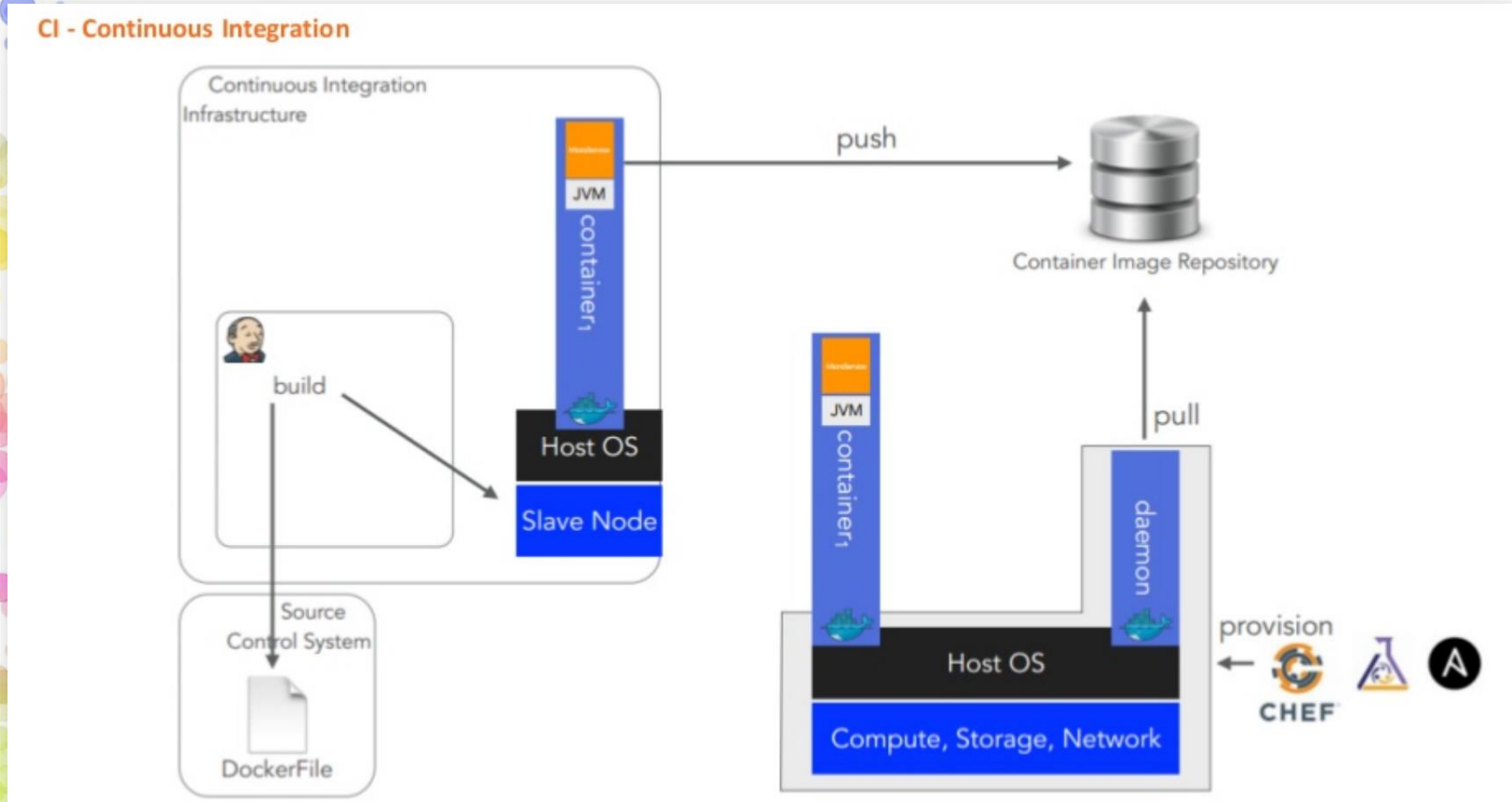


Containers have own isolated resources



Contenedores

Delivery Continuo



Ventajas y Desventajas

- Cada MS es pequeño y enfocado en una funcionalidad especifica.
- Cada MS puede ser desarrollado por un equipo.
- Cada MS tiene bajo acoplamiendo con el resto de la solución.
- Cada MS es independiente del lenguaje.
- Cada MS sólo tiene negocio (no presentación)
- Cada MS puede escalar bajo demanda.

- Overhead
- Requiere skills de DevOps
- Tiene los problemas cualquier Sistema Distribuido: difíciles de manejar, de verlos como un todo, trazabilidad punta a punta, etc.
- Hasta una cantidad son manejables… y despues?

¿Preguntas?

¡Hasta la próxima Semana!