

¡Bienvenidos!

ORACLE OTN Tour LA 2016 Guatemala

SOA y Microservices
Diferencias y Aplicaciones

Sandra Flores

SOA Architect



@sandyFloresMX



desarrolloconsoa.blogspot.mx



Oracle Technology Network LA

Comunidad mundial más grande de Desarrolladores, Administradores
y Arquitectos que utilizan productos Oracle

Y

OR AUGGT

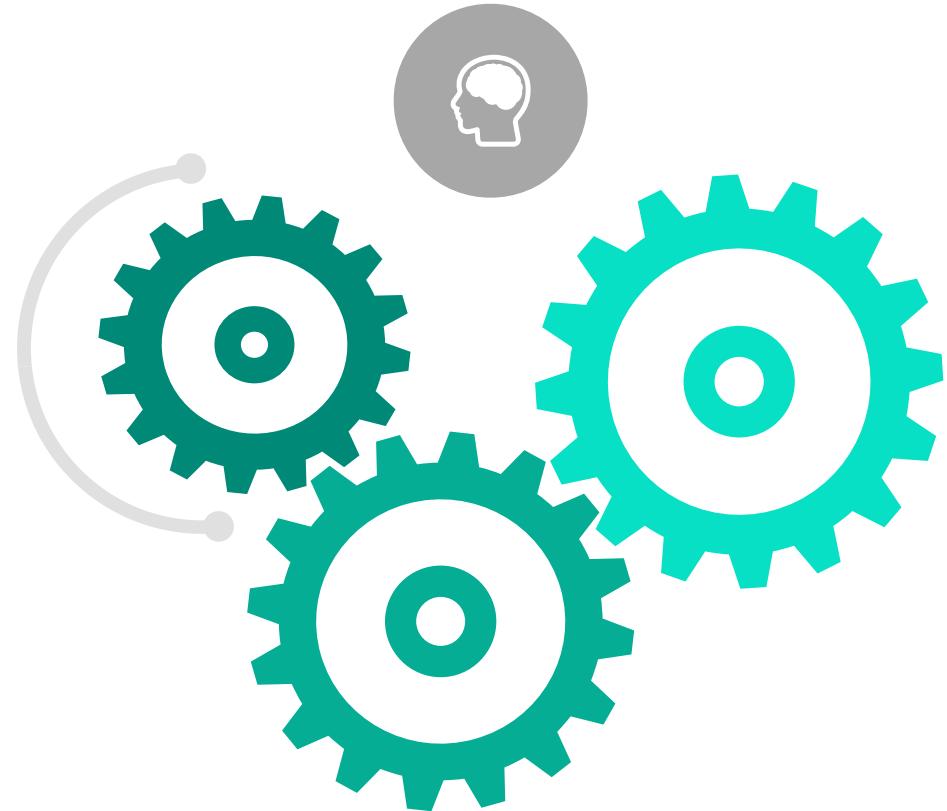
ORACLE USERS GROUP - GUATEMALA

Comunidad de tecnólogos Oracle en Guatemala que se reúnen frecuentemente virtual o
físicamente a intercambiar conocimiento sobre Oracle Tech.

Introducción

SOA y Microservices

En los últimos años el término Microservices ha atraído la atención de muchos de nosotros. Gran cantidad de información se genera al respecto y pareciera que cada vez más gente se quiere unir a éste movimiento. Sin embargo, aún existe mucha confusión, constantemente escuchamos preguntas como: ¿Qué son los Microservices? ¿Dónde y cómo implemento una solución con Microservices? ¿Acaso los Microservices no son más que Servicios SOA bien diseñados? ¿Si yo ya tengo mi solución SOA, debería pensar en cambiar a Microservices? ¿Si yo ya implemento servicios REST, significa que estoy haciendo Microservices?



Contenido

Temas **relevantes** de la sesión



SOA

Conceptos generales de SOA



Microservices

¿Qué son los Microservices?



Diferencias

¿En qué se diferencian SOA y Microservices?



Aplicaciones

Dudas generales de Microservices



01

02

03

04

SOA

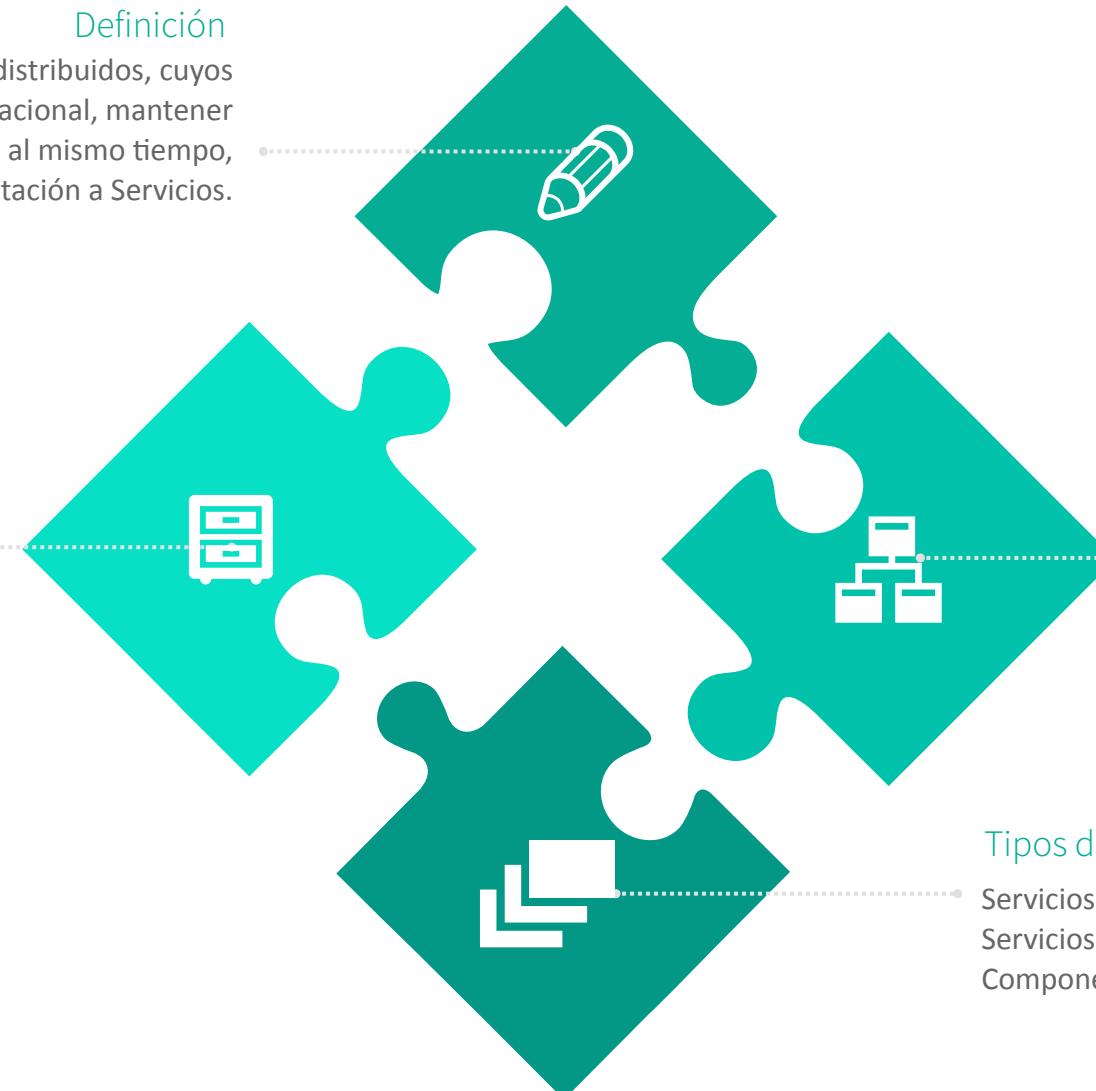
Service Oriented Architecture

Definición

Modelo arquitectónico para sistemas distribuidos, cuyos fines son mejorar la agilidad organizacional, mantener una alineación entre negocio y TI, y al mismo tiempo, implementar la Orientación a Servicios.

Bases

Se basa en Servicios como unidades fundamentales que engloban la lógica de negocio. Así, los procesos de negocio son compuestos de dichas piezas.



Foco

Sigue una serie de principios de la Orientación a Servicios y patrones de diseño. Se enfoca en la composición y reutilización de Servicios y promueve que éstos sean agnósticos en su mayoría, y orquestaciones para procesos completos.

Tipos de Servicios

Servicios Web SOAP
Servicios REST
Componentes

Microservices

Martin Fowler



1

2

3

1

Enfoque para desarrollo de una sola aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso, y en constante comunicación usando mecanismos ligeros, a menudo un API HTTP.

2

Están construidos alrededor de las capacidades de negocio y tienen independencia de despliegue, gracias a la maquinaria totalmente automatizada que los soporta.

3

Son altamente escalables y cada uno posee una firma asociada limitada de un módulo. Diferentes servicios pueden ser escritos en diferentes lenguajes de programación y también pueden ser administrados por diferentes equipos de trabajo.

<http://martinfowler.com/articles/microservices.html>

Microservices

SOA Patterns ORG



<http://soapatterns.org>

- ¿Cómo puede un servicio desplegarse de forma independiente para evitar las limitaciones impuestas por una implementación monolítica?
- Cada servicio es tratado como un producto independiente, por ende se desarrolla, se empaqueta y se despliega de forma independiente. Dicho servicio puede ser calificado como un "Microservice".
- Los Microservices están diseñados como servicios altamente autónomos que dependen principalmente de la comunicación asíncrona entre los distintos servicios.
- El uso de contenedores de software (patrón Containerized Service Deployment Pattern) produce una variación del modelo. La tecnología de los contenedores se utiliza a menudo para empaquetar una aplicación completa junto con el servidor de aplicaciones y otra infraestructura necesaria para generar un único despliegue. Utilizando la tecnología de contenedores, cada Microservice puede ser "contenerizado" de forma independiente como si se tratara de una aplicación independiente.

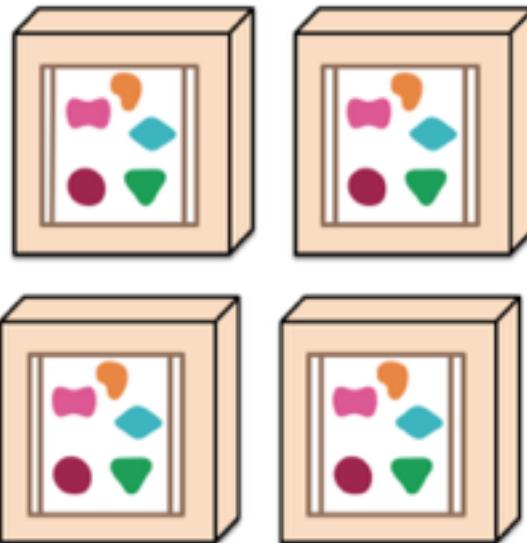
De Monolito a Microservices

Según Martin Fowler

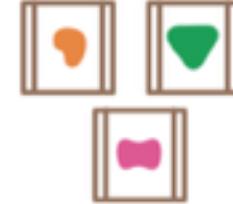
A monolithic application puts all its functionality into a single process...



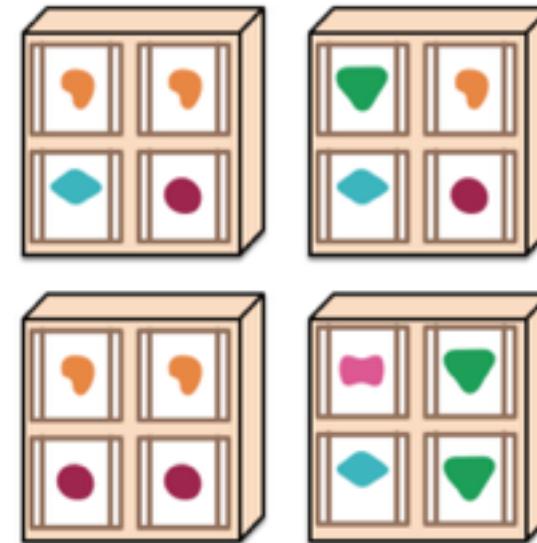
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...

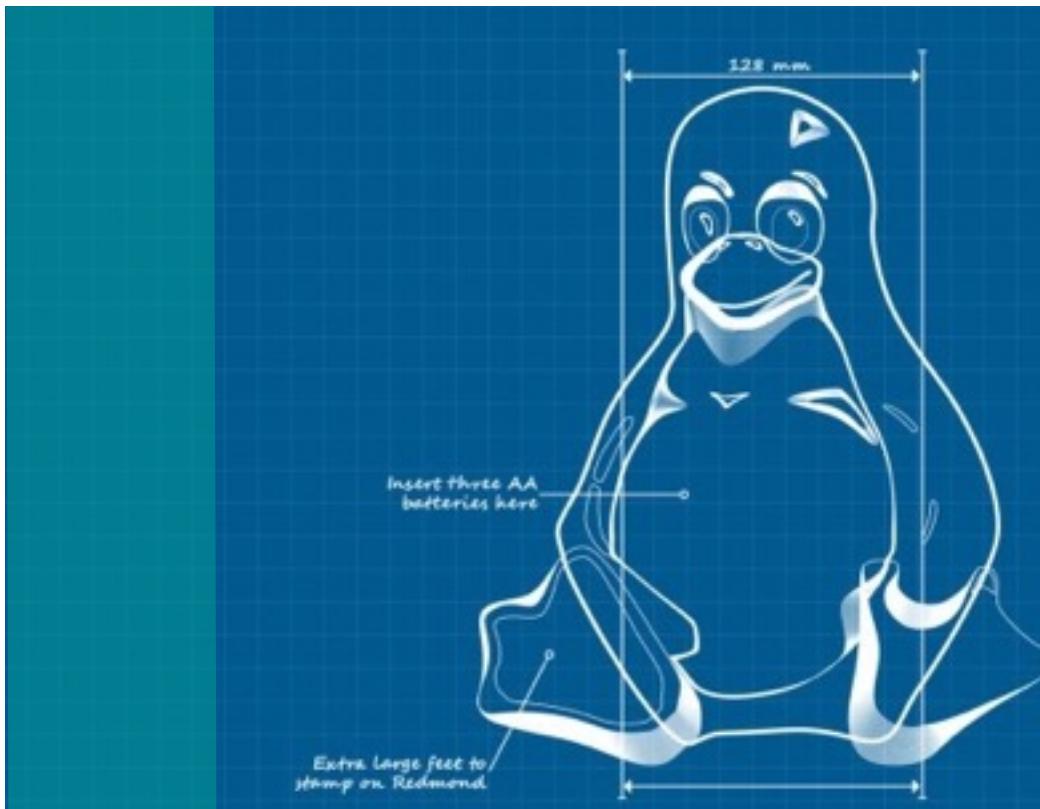


... and scales by distributing these services across servers, replicating as needed.



Filosofía UNIX

Creada en 1978, simplificada en 1994



Crear programas que hagan solo una cosa
y que la hagan bien.



Crear programas que trabajen juntos.



Crear programas que usen flujos de texto,
porque es una interface universal.

Características

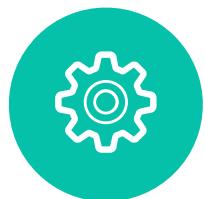
Los **Microservices**:



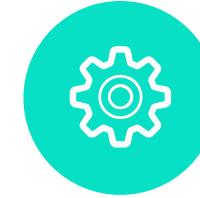
Son totalmente autónomos y abstractos.



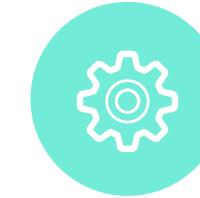
Tienen un alcance funcional pequeño y limitado.



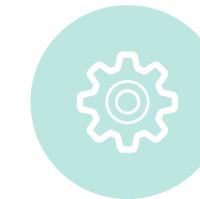
No requieren participar en transacciones distribuidas.



Están organizados entorno a las capacidades del negocio, por equipos de trabajo especializados.



No dependen de invocaciones síncronas de otros servicios.



No deben ser parte de composiciones complejas de servicios.

Características Cont.

Los **Microservices**:



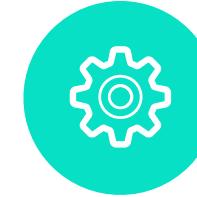
Soportan interoperabilidad por medio de mecanismos de comunicación basados en mensajes.



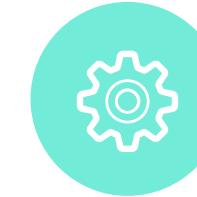
Usan tecnología de Componentización.



Usan infraestructura automatizada.



Se enfocan en el planteamiento Smart Endpoints & Dumb Pipes



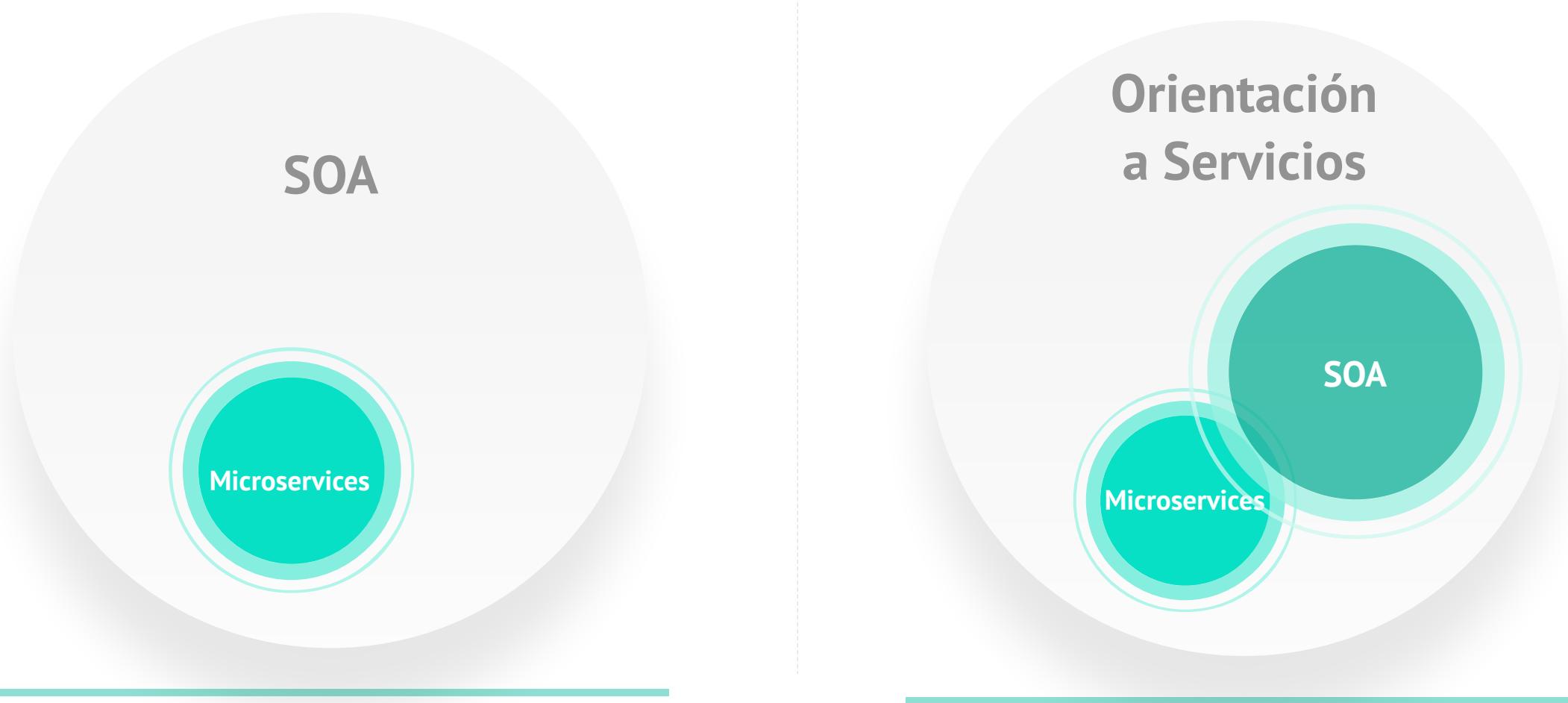
Poseen un gobierno descentralizado, sin ataduras a lenguajes o plataformas.



Manejan sus propio almacenamiento de datos (Persistencia Políglota)

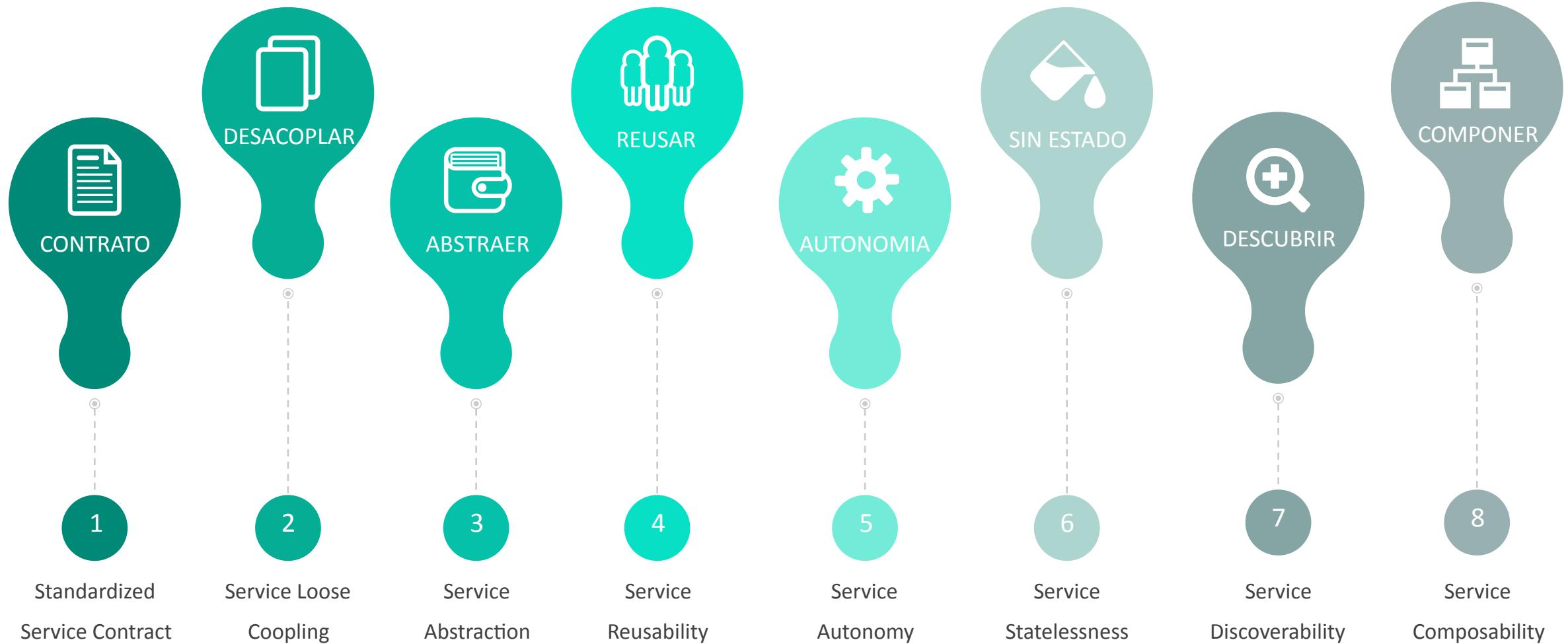
SOA vs Microservices

Relación entre SOA, Microservices y la Orientación a Servicios



Orientación a Servicios

Principios Básicos



SOA vs Microservices

¿Realmente están peleados?



SOA



Standardized
Service Contract



Service
Reusability



Service
Composability

Microservices



Service
Autonomy



Service
Abstraction



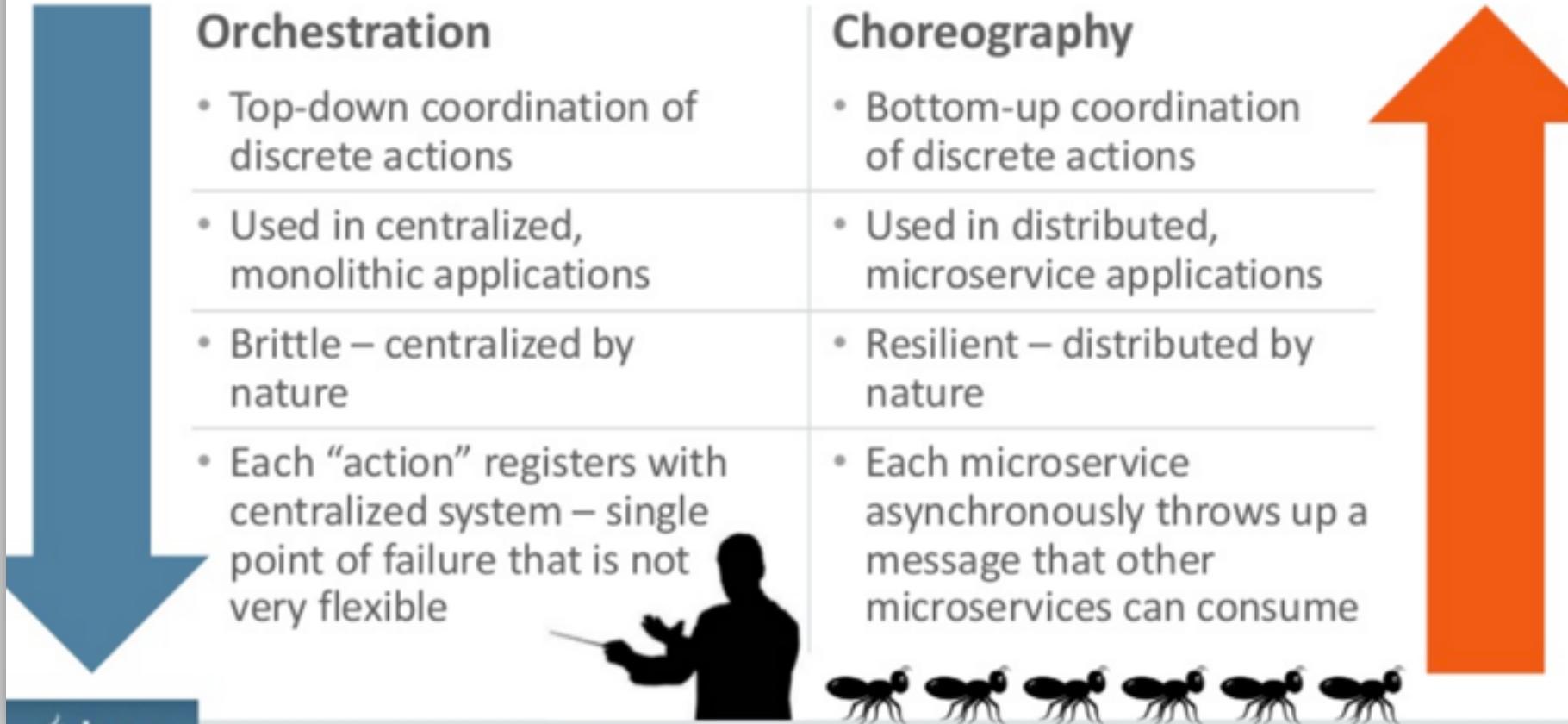
Service
Loose Coupling

SOA vs Microservices

Coreografía **VS** Orquestación

Choreography Tends to Be Better Than Orchestration

Orchestration	Choreography
<ul style="list-style-type: none">• Top-down coordination of discrete actions	<ul style="list-style-type: none">• Bottom-up coordination of discrete actions
<ul style="list-style-type: none">• Used in centralized, monolithic applications	<ul style="list-style-type: none">• Used in distributed, microservice applications
<ul style="list-style-type: none">• Brittle – centralized by nature	<ul style="list-style-type: none">• Resilient – distributed by nature
<ul style="list-style-type: none">• Each “action” registers with centralized system – single point of failure that is not very flexible	<ul style="list-style-type: none">• Each microservice asynchronously throws up a message that other microservices can consume



SOA vs Microservices

Comparación

	SOA	Microservices
Monolith pattern (http://bit.ly/1Gjr2Y0)	Yes	No
Polyglot Programming & Persistence (http://bit.ly/18BvDIj & http://bit.ly/1XYiak2)	Not traditionally (use of Suites)	Yes
API gateway pattern (http://bit.ly/1WTyNLJ)	Yes	Yes
Orchestration (http://bit.ly/1U0SWil)	Yes	No
Choreography (http://bit.ly/1ssALZQ)	No	Yes
Event Collaboration (http://bit.ly/25Dk7oE)	Yes	Yes
Canonical Schema (http://bit.ly/1r6KkfK)	Very common	No
Schema centralization (http://bit.ly/1sVlqkc)	Very common	No
Decouple Contract (http://bit.ly/1O8mVpm)	Yes	Could be...
Bounded Context (http://bit.ly/1o7AK8B)	Some times	Yes
Ubiquitous Language (http://bit.ly/1c8nXQe)	Some times	Yes
Bulkhead (http://bit.ly/1c8nXQe)	Not really...	Yes
Tolerant Reader (http://bit.ly/1aa4mr9)	Some times	Yes
Client-side Service Discovery (http://bit.ly/1OunUyq)	Initially only (service registry)	Recommended
Server-side Service Discovery (http://bit.ly/1X3RmzA)	Yes	Yes



Oferta de Oracle

Para implementación de **Microservices**

Oracle's Microservices Strategy

Oracle's Microservices Roadmap		Today With Oracle Products
State	Planned - Oracle Cloud State Service	Oracle Coherence or Oracle WebLogic
Configuration	Planned - Oracle Cloud Config Service	Oracle Coherence
Runtime	Planned - Runtime - Jersey + Grizzly	Oracle WebLogic, Node, Java SE, etc
Eventing	Planned - Oracle Cloud Eventing Service	Oracle Coherence
Messaging	Oracle Messaging Cloud Service	Oracle Messaging Cloud Service
Management/Logging/Alerting	Oracle Management Cloud Service	Oracle Management Cloud Service
Datastore	Oracle Database or NoSQL Cloud Service	Oracle Database or NoSQL Cloud Service
Central Source of Truth	Planned - Oracle Microservices Platform	Oracle Coherence
Service Discovery	Planned - Oracle Microservices Platform	Oracle App Container Cloud Service
API Gateway/Load Balancer	Planned - Oracle Microservices Platform	Oracle App Container Cloud Service
Container Orchestration	Planned - Oracle Microservices Platform	Oracle App Container Cloud Service
Infrastructure	Oracle Cloud	Oracle Cloud
Build/Deploy	Oracle Developer Cloud Service	Oracle Developer Cloud Service



SOA vs Microservices

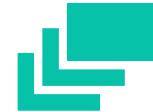
Conceptos Erróneos



¿Si yo ya implemento servicios REST, significa que estoy haciendo Microservices?



¿Acaso los Microservices no son más que Servicios SOA bien diseñados?



¿Si yo ya tengo mi solución SOA, debería pensar en cambiar a Microservices?



¿Porqué dicen que el ESB una “Atroz caja de espagueti”?



¿Cómo sé si los Microservices son una buena opción para mí?

KNOCK. KNOCK. WHO'S THERE?
REALITY



Casos de Exito

Soluciones con **Microservices**



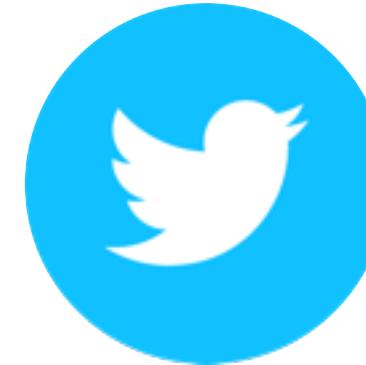
Amazon



Netflix



eBay



Twitter



Google



soundCloud



Nike



Hailo

Determinando la Factibilidad

Los **Resultados** son solo la punta del **Iceberg**



“ People try to copy Netflix, but they can only copy what they see. They copy the results, not the process. ”

Adrian Cockcroft

Former Netflix Chief Cloud Architect

Determinando la Factibilidad

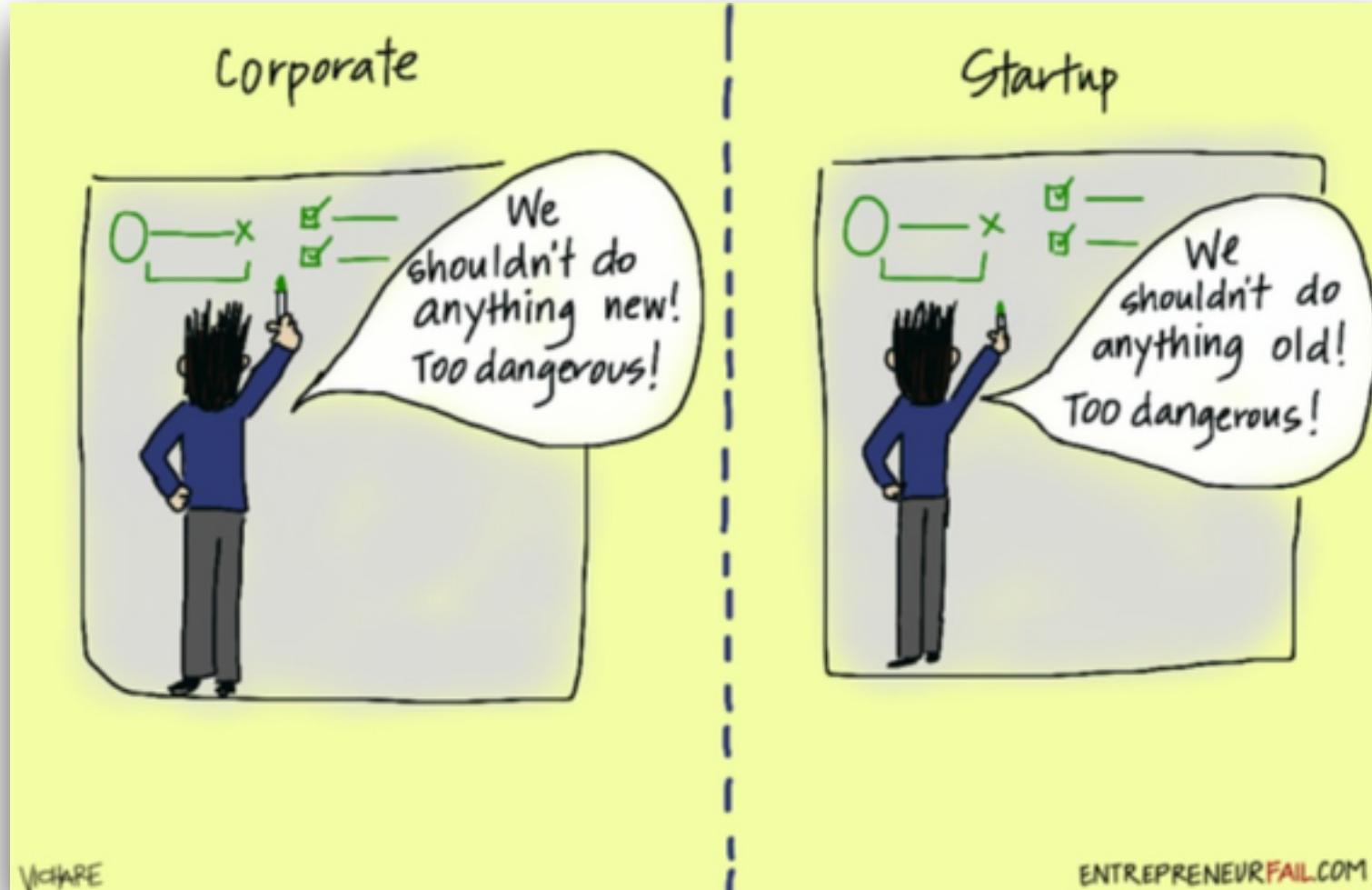
Cuestionamiento **clave** sobre **Microservices**



¿Modernización de Legados o
Implementaciones nuevas?

Corporaciones vs Startups

Aversión al riesgo



Microservices como Nuggets

ADP CTO en Dockercon 16



<https://www.youtube.com/watch?v=4ywzwCmML2Y>

Preguntas a resolver

¿Son los **Microservices** adecuados para mi implementación?

- 
1. ¿Mi negocio hace énfasis en la alta velocidad de entrega de valores a mis clientes?
 2. ¿Tengo capacidad de adaptación al cambio drástico en mi forma de trabajo?
 3. ¿Tengo experiencia definiendo, diseñando y categorizando servicios?
 4. ¿Estoy dispuesto a poner en práctica una cultura de DevOps en mis equipos de trabajo?
 5. ¿Mis datos residen en repositorios centralizados y monolíticos?
 6. ¿Estoy dispuesto a descentralizar mi repositorio de datos y usar mecanismos complejos de sincronización usando nuevas y modernas tecnologías?
 7. ¿Mi infraestructura de red es lo suficientemente robusta para soportar un fuerte incremento de comunicaciones?
 8. ¿Estoy dispuesto a cambiar la infraestructura y la base tecnológica que soporta mi operación actual?

Consideraciones Relevantes

Restricciones al implementar Microservices

Inexperiencia

Las implementaciones con Microservices aún no maduran completamente.

Oferta del Mercado

Las herramientas emergen y evolucionan con rapidez, lo cual complica las decisiones de arquitectura.

Documentación

Aún no hay mucha documentación o patrones de diseño sobre los cuales basarse para las soluciones.

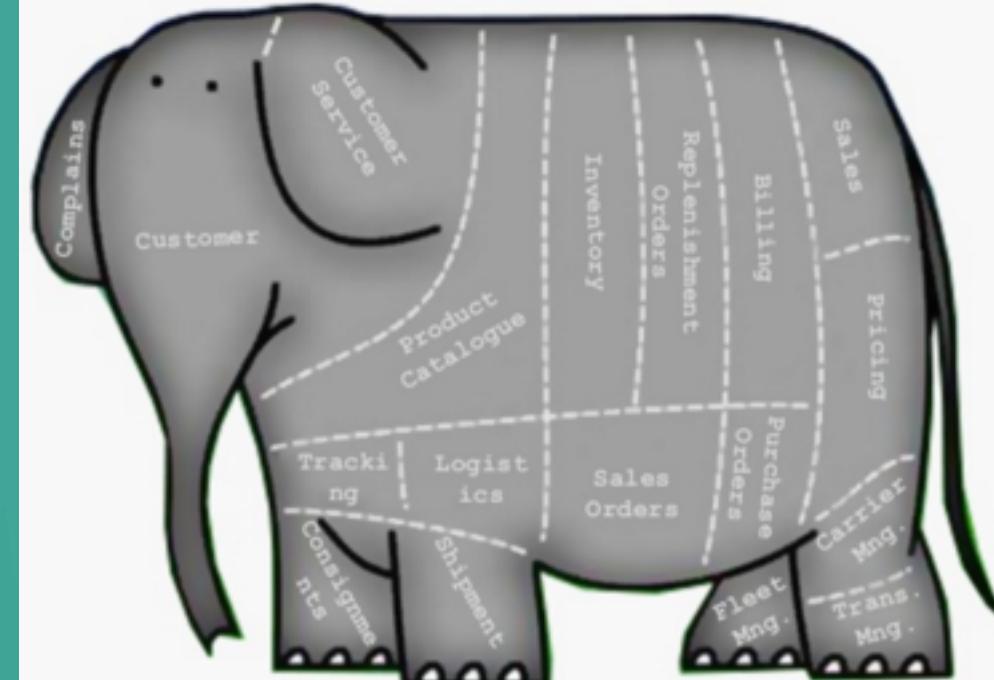
Datos

Los repositorios de datos centralizados requieren ser divididos a medida que la transición a Microservices avanza.

Estrategia

Es vital definir una estrategia de implementación, ya sea para modernización de legados o aplicaciones nuevas.

One piece at the time!



SOA

¿Cuándo usar SOA?



Integración de un stack de aplicaciones dispares



No es posible descentralizar los repositorios de datos



Existen Legados que no serán modernizados



Se requiere una plataforma bien conocida para implementar soluciones Orientadas a Servicios



Requiero centralizar y gobernar mi inventario de Servicios

¿Preguntas?

WHAT

WHY

WHERE

WHEN

WHO

HOW



¡Gracias por su tiempo!

Contacto

<http://www.oracle.com/technetwork/es/index.html>



@oracleotnla

ORAMEX

<http://oramex.com.mx>



@oramexico

Sandra Flores
SOA Architect



@sandyFloresMX

desarrolloconsoa.blogspot.mx