

Arquitectura e implementación de Microservicios con Spring Cloud Netflix OSS

ISC. Ivan Venor García Baños
Instructor





+

Agenda

1. Presentación
2. Objetivos
3. Contenido
4. Despedida



Microservices



Esta página fue intencionalmente dejada en blanco.



Microservices



+

NETFLIX
OSS

3. Contenido

- i. Arquitectura de sistemas monolíticos
- ii. Introducción a la Arquitectura Orientada a Servicios
- iii. Fundamentos Spring Boot 2.x
- iv. Arquitectura de Microservicios
- v. Microservicios con Spring Cloud y Spring Cloud Netflix OSS



3. Contenido

- i. Arquitectura de sistemas monolíticos
- ii. Introducción a la Arquitectura Orientada a Servicios
- iii. Fundamentos Spring Boot 2.x
- iv. Arquitectura de Microservicios
- v. Microservicios con Spring Cloud y Spring Cloud Netflix OSS



+

ii. Introducción a la Arquitectura Orientada a Servicios

Microservices



+

ii. Introducción a la Arquitectura Orientada a Servicios

ii.i ¿Qué es SOA?

ii.ii Principios de diseño de orientación a servicios



Microservices



ii.i ¿Qué es SOA?





Objetivos de la lección

ii.i ¿Qué es SOA?

- Analizar lo que es la Arquitectura Orientada a Servicios o Service Oriented Architecture.
- Comprender SOA a nivel macro y analizar sus posibles implementaciones.



ii.i ¿Qué es SOA? (a)

- SOA
- La Arquitectura Orientada a Servicios (SOA) es un framework conceptual que permite a las organizaciones unir los objetivos de negocio con la infraestructura de TI integrando los datos y la lógica de negocio de sus sistemas separados.
- SOA es un enfoque de desarrollo de aplicaciones de software empresarial, en el cual los procesos del software se descomponen en servicios, que después se hacen disponibles y visibles en una red.



ii.i ¿Qué es SOA? (b)

- SOA es una representación de una arquitectura abierta, extensible y federada basada en composición, que promueve la orientación a los servicios interoperables e independientes de los proveedores, los cuales pueden ser identificados en catálogos con gran potencial de reutilización e implementados como servicios Web.
- Cada servicio expuesto provee funcionalidades para poder ser adecuado a las necesidades de la empresa, mientras esconde los detalles inherentes de implementación.



ii.i ¿Qué es SOA? (c)

- SOA aborda la complejidad, inflexibilidad y debilidades de los enfoques tradicionales para la interoperabilidad de aplicaciones separadas, tales como compartición de archivos, compartición de base de datos, conversión y adaptación de estructuras de I/O para la intercomunicación de sistemas.



ii.i ¿Qué es SOA? (d)

- SOA provee la infraestructura de tecnologías de la información que permite a diferentes aplicaciones intercambiar datos y participar en los procesos de negocio, independientemente del sistema operativo o de los lenguajes de programación con los cuales los servicios y los sistemas interconectados han sido desarrollados.



ii.i ¿Qué es SOA? (e)

- Existen diversas definiciones de SOA que incluyen el término “servicios web” o “web-services” sin embargo, es necesario hacer la distinción de estos conceptos y aclarar que SOA no es lo mismo que Servicios Web.
- La Arquitectura Orientadas a Servicios, a diferencia de los Servicios Web, define y trata un paradigma de orientación a servicios, en tanto que los Servicios Web son sólo una forma posible de implementar la infraestructura SOA utilizando una estrategia de implementación tecnológica específica.



+

ii.i ¿Qué es SOA? (f)

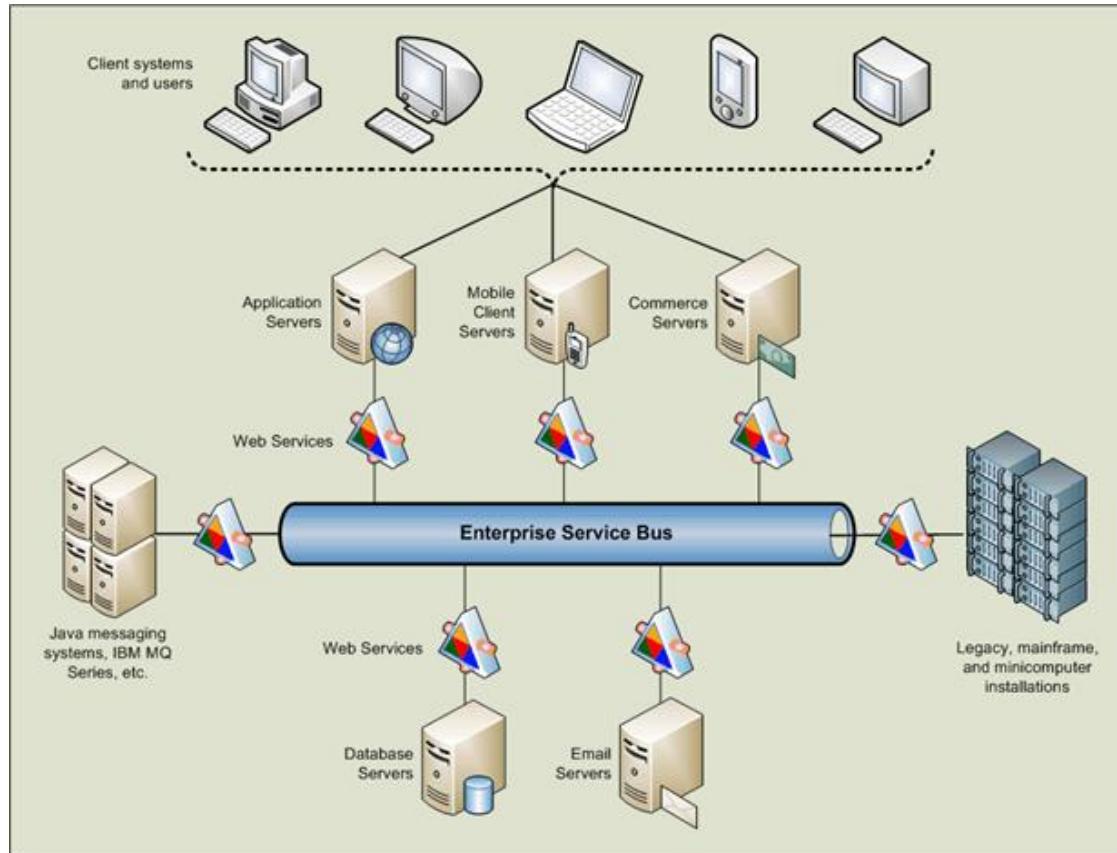
- Podemos resumir que SOA es un paradigma arquitectónico que permite el tratamiento de procesos de negocio distribuidos de sistemas heterogéneos, que se encuentran bajo el control o responsabilidad de diferentes propietarios donde sus conceptos clave son:
 - Disponibilizar un activo de negocio como un servicio.
 - La interoperabilidad entre diversos lenguajes y aplicaciones, y
 - El bajo acoplamiento entre los componentes.
- Los principales actores en una SOA son la infraestructura, la arquitectura y los procesos.



+

NETFLIX
OSS

ii.i ¿Qué es SOA? (g)





Resumen de la lección

ii.i ¿Qué es SOA?

- Analizamos a grandes rasgos la Arquitectura Orientada a Servicios.
- Comprendemos de que trata las Arquitecturas SOA.
- Debatimos referente a las diversas implementaciones SOA en la industria.
- Analizamos el marco de trabajo SOA como un conjunto de prácticas que convierten los activos funcionales de una empresa a servicios para que sean reutilizados a través de sus sistemas separados.



Esta página fue intencionalmente dejada en blanco.



Microservices



+

NETFLIX
OSS

ii. Introducción a la Arquitectura Orientada a Servicios

ii.i ¿Qué es SOA?

ii.ii Principios de diseño de orientación a servicios





+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios

Microservices



Objetivos de la lección

ii.ii Principios de diseño de orientación a servicios

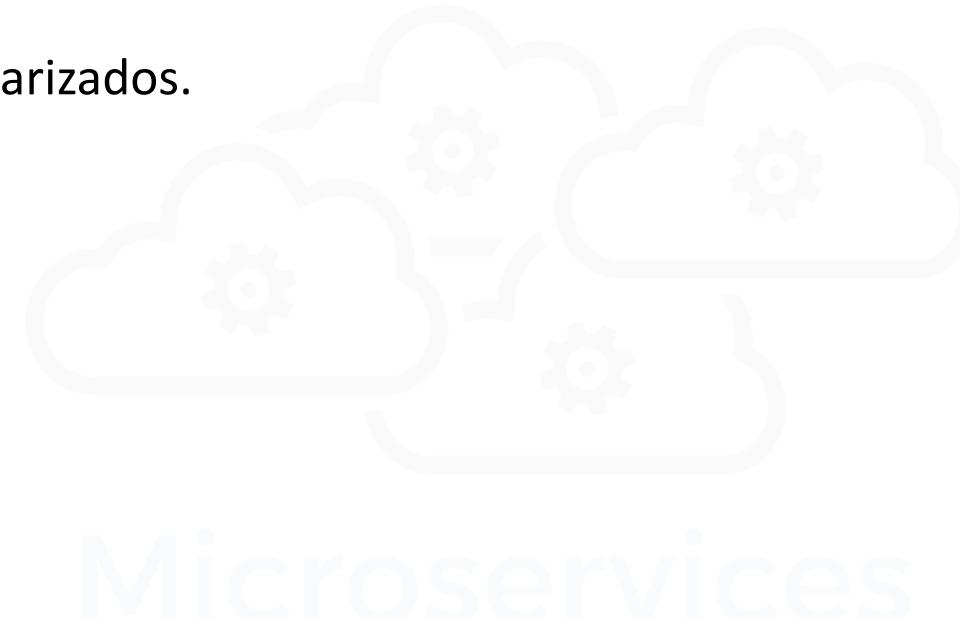
- Verificar los principios de diseño de orientación a servicios.
- Analizar los diferentes principios de diseño de orientación a servicios para tener en cuenta al momento de especificar servicios-web ya sea basados en arquitectura SOA o en arquitectura basada en microservicios.
- Distinguir las diferencias entre los distintos principios de diseño de orientación a servicios.



+

ii.ii Principios de diseño de orientación a servicios (a)

- Principios de diseño de orientación a servicios:
 - Contratos de servicios estandarizados.
 - Bajo acoplamiento.
 - Abstracción.
 - Reusabilidad.
 - Autonomía.
 - Sin estado.
 - Descubrimiento.
 - Composición.
 - Interoperabilidad.





ii.ii Principios de diseño de orientación a servicios (b)

- Contrato de servicios estandarizados (a).
- Cuando un servicio se implementa como un Servicio Web, se debe adherir un contrato o interfaz de comunicaciones explícitamente declarado y definido, colectivamente, por uno o más descriptores del servicio, en el cual debe figurar especificaciones como:
 - Nombre del Servicio.
 - Forma de acceso.
 - Funcionalidades que ofrece.
 - Descripción de los datos de entrada de cada funcionalidad que ofrece.
 - Descripción de los datos de salida de cada funcionalidad que ofrece.



ii.ii Principios de diseño de orientación a servicios (c)

- Contrato de servicios estandarizados (b).
- Mediante contratos de servicios estandarizados, todo consumidor de servicios accederá a éste mediante su contrato definido, logrando la independencia entre el consumidor y la implementación del servicio.
- De esta forma, se evita el manejo incorrecto de los datos, se evita trabajo innecesario al momento de la invocación del servicio y también se pone de manifiesto de la existencia de un modelo de datos a nivel de servicio, siendo esta una de las primeras necesidades que se debe tener en cuenta al momento de definir servicios.



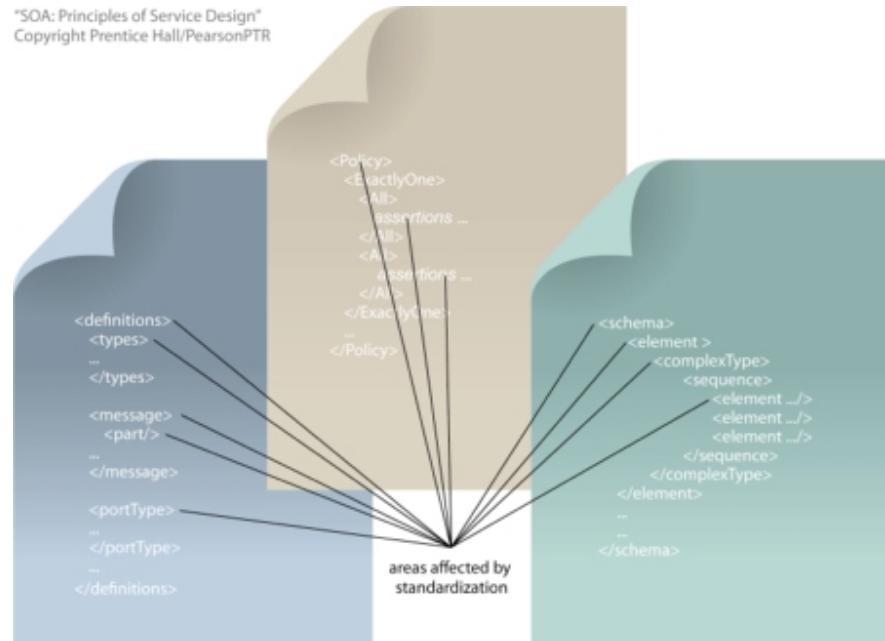
+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (d)

- Contrato de servicios estandarizados (c).

"SOA: Principles of Service Design"
Copyright Prentice Hall/PearsonPTR



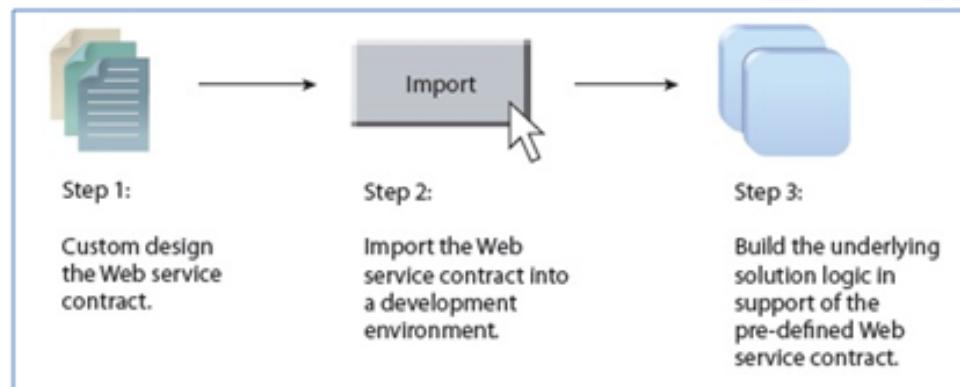


+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (e)

- Contrato de servicios estandarizados (c).





ii.ii Principios de diseño de orientación a servicios (f)

- Bajo acoplamiento (a).
- Este principio hace referencia a que los servicios tienen que ser independientes los unos de los otros.
- Para lograr el bajo acoplamiento entre los servicios, se define que el único medio de acceso a los servicios es el contrato o interfaz, logrando así la independencia entre el servicio que se va a ejecutar y el que lo llama.
- El bajo acoplamiento permite que los servicios sean totalmente reutilizables.

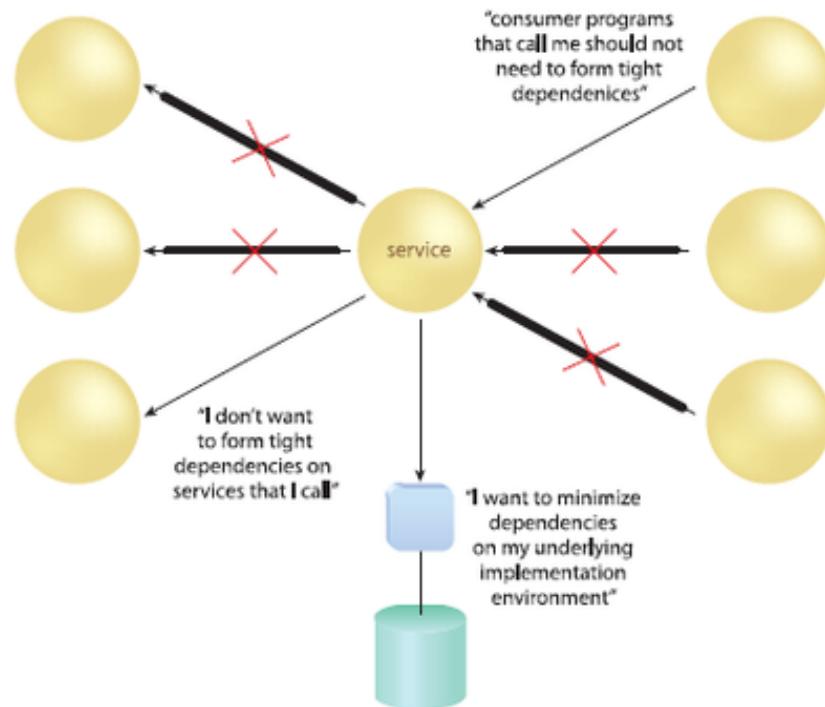


+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (g)

- Bajo acoplamiento (b).





ii.ii Principios de diseño de orientación a servicios (h)

- Abstracción (a):
- A nivel fundamental, la abstracción permite ocultar los detalles de implementación de un servicio, tanto como sea posible.
- El principio de abstracción permite encapsular el servicio como una “caja negra”, del cuál no se saben los detalles y únicamente está definido por su contrato, habilitando así el bajo acoplamiento, que a su vez, la definición y especificación del contrato es el mínimo acoplamiento posible entre el servicio y un consumidor.



ii.ii Principios de diseño de orientación a servicios (i)

- Abstracción (b):
- El uso de estándares permite definir interfaces uniformes que esconden la lógica del servicio respecto del entorno que le rodea (sistemas proveedores y consumidores).
- Este nivel de abstracción permite centrarse exclusivamente en la especificación del servicio, sin incluir información tecnológica ni de ninguna otra naturaleza, más allá de la propia especificación estándar del servicio, desde el punto de vista del negocio al que sirve, la cual queda definida en su contrato.

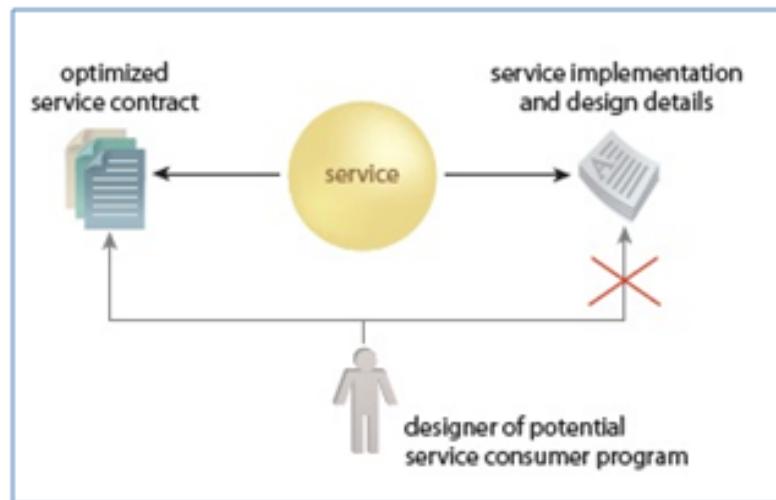


+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (j)

- Abstracción (c):





ii.ii Principios de diseño de orientación a servicios (k)

- Reusabilidad (a):
- La reusabilidad (reutilización) es el principal principio de la arquitectura SOA, cada servicio debe ser analizado, diseñado y construido de manera que su uso pueda ser explotado al máximo por otros sistemas consumidores del servicio.
- Los servicios deben de ser definidos de tal forma que puedan utilizarse en diferentes contextos y satisfacer distintos objetivos de negocio, únicamente centrandose en las capacidades del negocio y no en alguna tecnología específica.



ii.ii Principios de diseño de orientación a servicios (I)

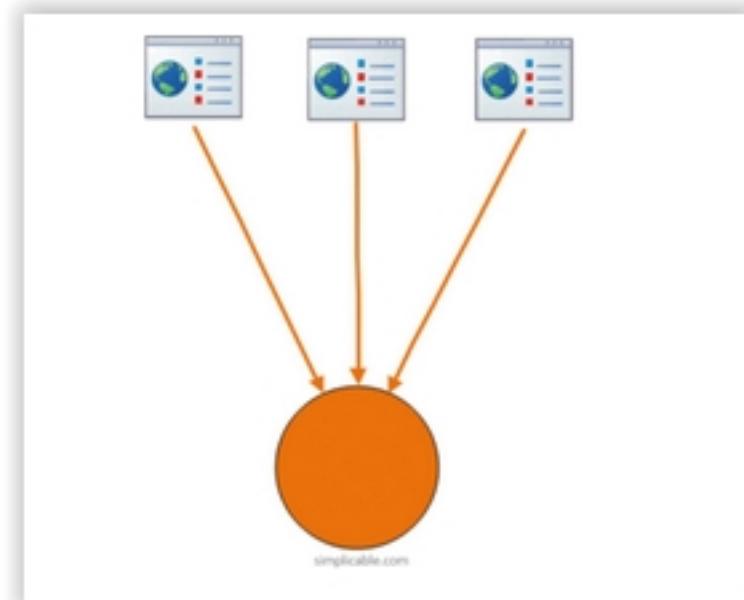
- Reusabilidad (b):
- De esta manera, los servicios pueden ser reusables dentro de la misma aplicación, dentro del dominio de aplicaciones de la empresa o incluso dentro del dominio público.
- Por otro lado, estos servicios reutilizables deben estar diseñados de manera tal que su solución lógica sea independiente de cualquier proceso de negocio o tecnología en particular.



+

ii.ii Principios de diseño de orientación a servicios (m)

- Reusabilidad (c):
- Con este principio se busca reducir las posibilidades de duplicación de lógica.

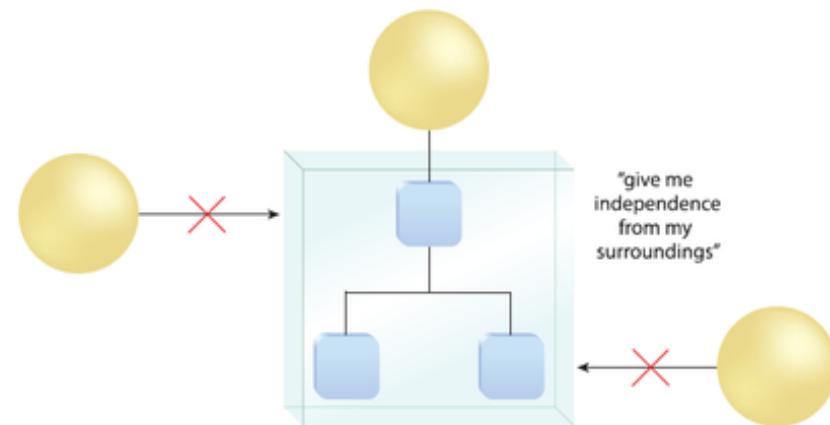




+

ii.ii Principios de diseño de orientación a servicios (n)

- Autonomía (a):
- Este principio refiere a que todo servicio debe tener su propio entorno de ejecución, logrando que dicho servicio sea totalmente independiente.
- De esta forma se asegura la reusabilidad del servicio desde el punto de vista de la plataforma de ejecución.



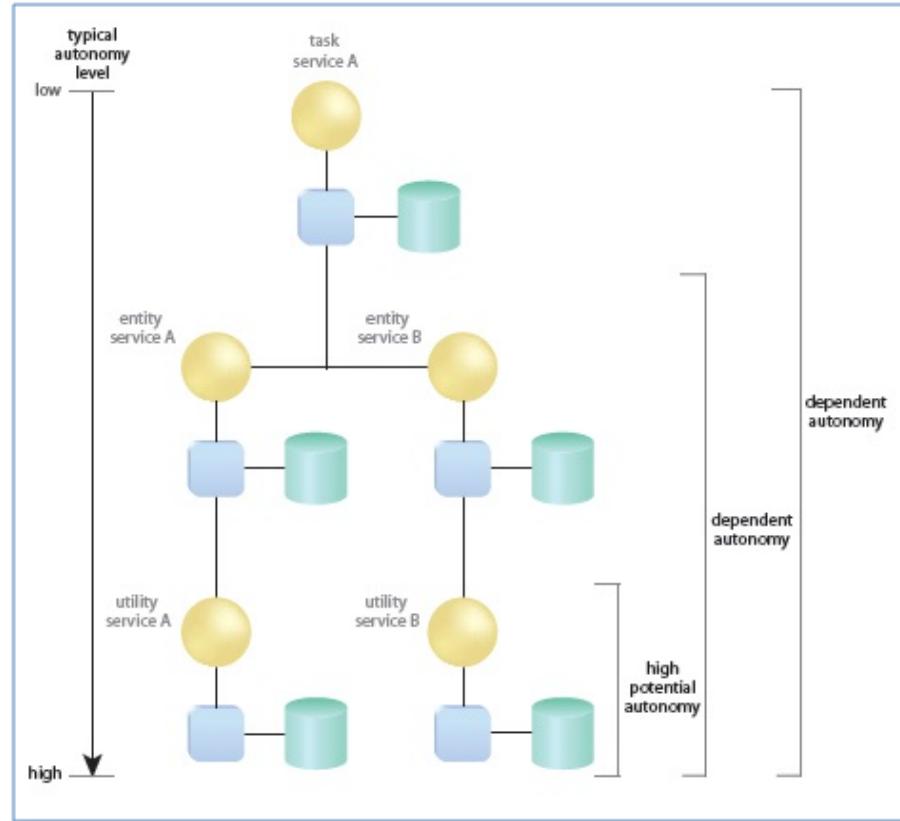


+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (n)

- Autonomía (b):





ii.ii Principios de diseño de orientación a servicios (ñ)

- Sin estado (a):
- Los servicios no deben de almacenar algún tipo de información con referencia a los consumidores del mismo.
- La gestión de exesiva carga de información minimiza y deteriora la escalabilidad del servicio lo cual afecta gravemente la disponibilidad del servicio.
- De forma ideal, todos los datos que necesita un servicio para su ejecución deben provenir directamente de los parámetros de entrada que provee el consumidor del servicio.

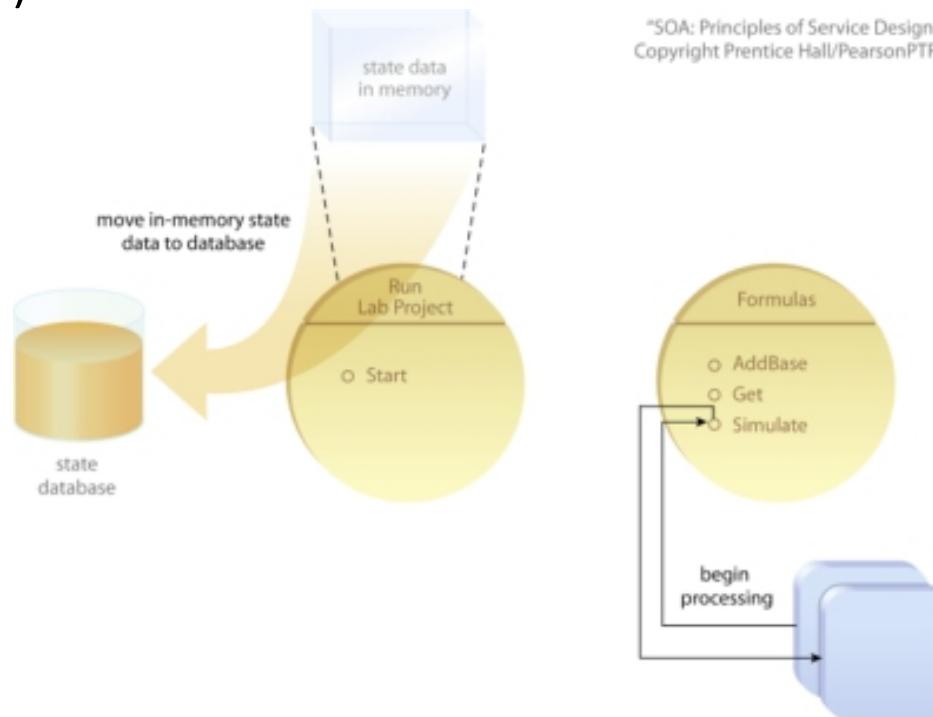


+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (o)

- Sin estado (b):





ii.ii Principios de diseño de orientación a servicios (p)

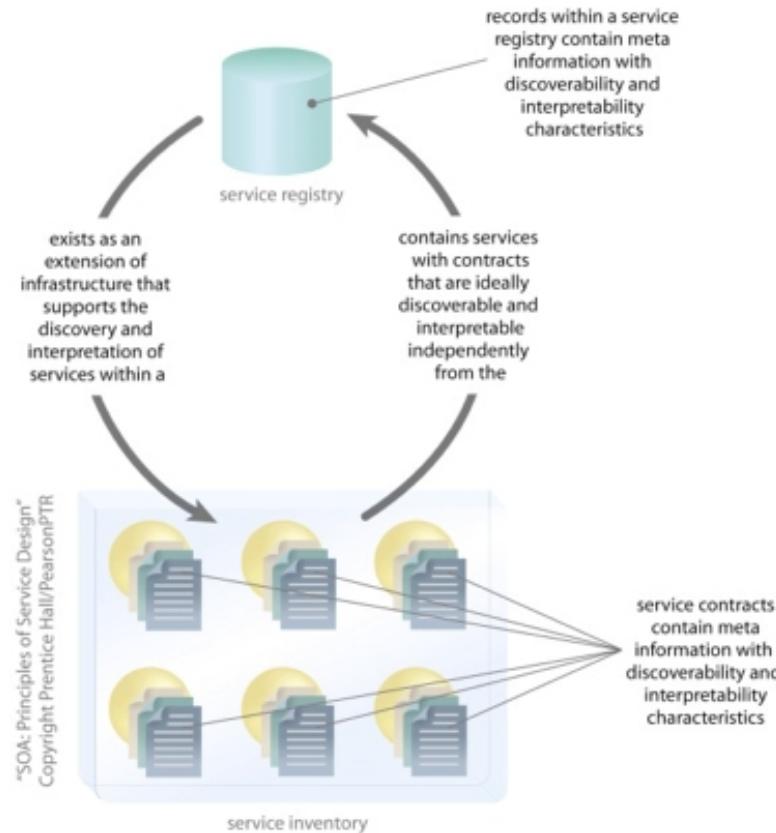
- Descubrimiento (a):
- Todo servicio expuesto debe poder ser descubierto de alguna forma para que pueda ser utilizado.
- El descubrimiento de servicios evita la ducplicidad accidental de servicios que proporcionen una misma funcionalidad.
- Para el caso de aplicar descubrimiento de servicios en una arquitectura SOA basada en Servicios Web, se deberá contar con la publicación de los servicio a traves de un registro UDDI (Universal Description, Discovery and Integration).



+

ii.ii Principios de diseño de orientación a servicios (q)

- Descubrimiento (b):





+

ii.ii Principios de diseño de orientación a servicios (r)

- Composición (a):
- El principio de composición asegura la habilidad efectiva de un servicio para ser utilizado para componer otros servicios más complejos.
- La composición es uno de los requisitos más críticos para lograr una arquitectura orientada a servicios.
- El diseño de composición de servicios más complejos, basados en servicios de menor nivel (más atómicos) deben de ser visualizados con anticipación para evitar un doble esfuerzo.



+

ii.ii Principios de diseño de orientación a servicios (s)

- Composición (b):
- El concepto de desarrollo de software a partir de componentes existentes independientes, fomenta el concepto de composición.
- Dada la composición de servicios es que es posible automatizar un proceso de negocio debido a que, dicho proceso, se ejecuta mediante la combinación y composición de múltiples servicios.
- SOA dispone de su agilidad de implementación de servicios mediante la composición de servicios.

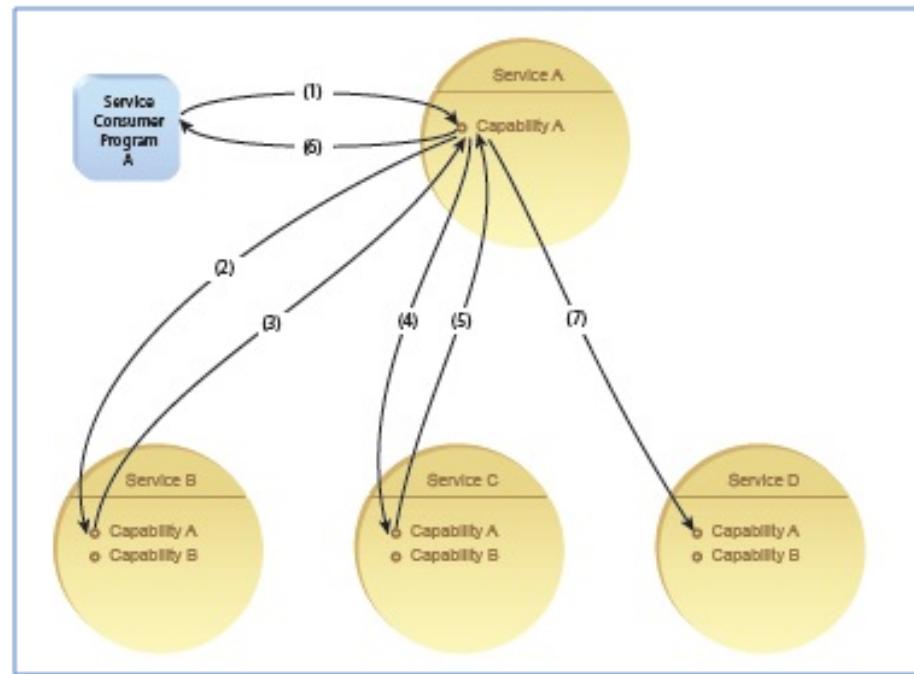


+

NETFLIX
OSS

ii.ii Principios de diseño de orientación a servicios (t)

- Composición (c):
- Dada la implementación de Servicios Web como, pilar fundamental en una arquitectura SOA, WS-BPEL y WS-CDL permiten la composición de servicios mediante orquestación y coreografía.

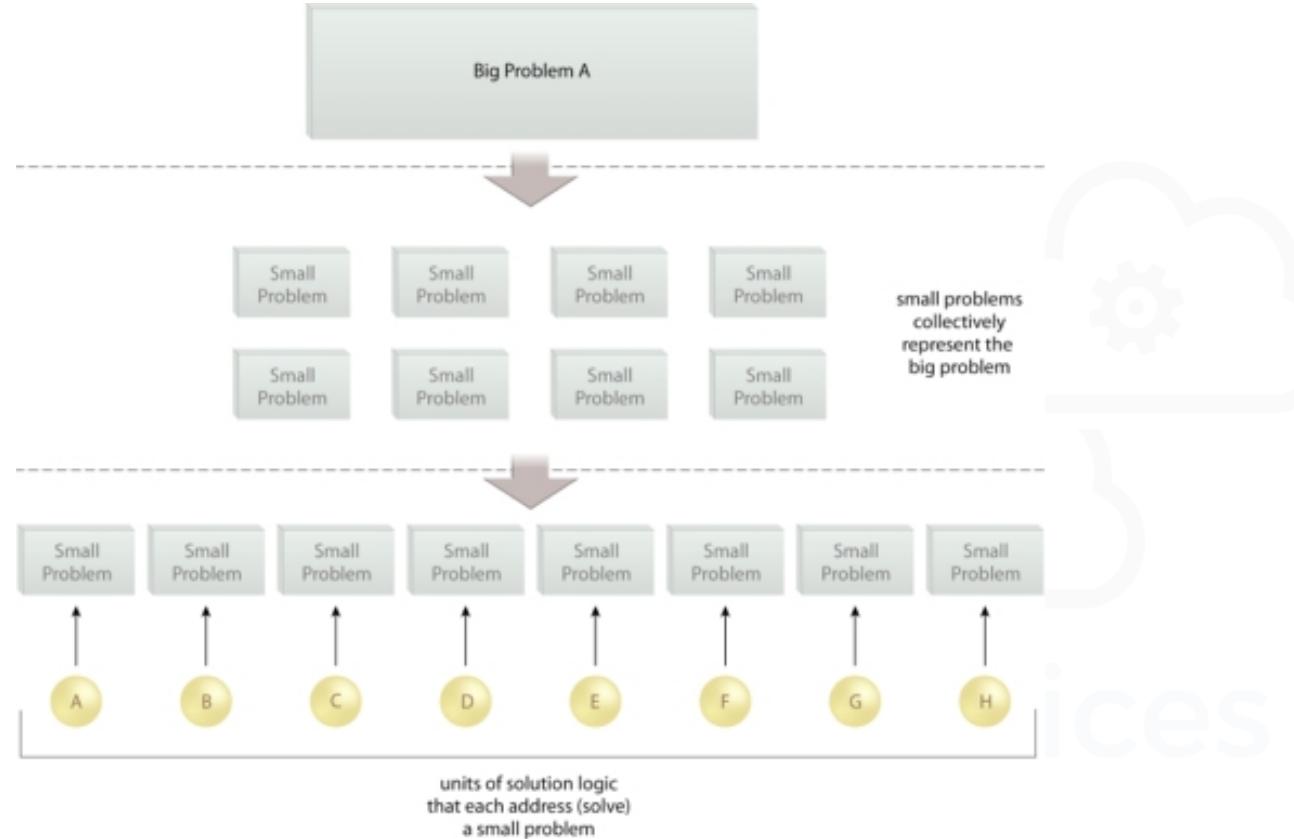




+

ii.ii Principios de diseño de orientación a servicios (u)

- Composición (d):





ii.ii Principios de diseño de orientación a servicios (v)

- Interoperabilidad (a):
- La interoperabilidad se refiere a que los servicios deben poder interactuar, entre ellos, a pesar de estar implementados en tecnologías diferentes.
- Para lograr interoperabilidad deben aplicarse los principios de diseño de orientación a servicios, de esta forma, es posible visualizar a la interoperabilidad como una característica de los servicios.
- La interoperabilidad de servicios se logra mediante la implementación de estándares que permitan tener diferentes tipos de consumidores.



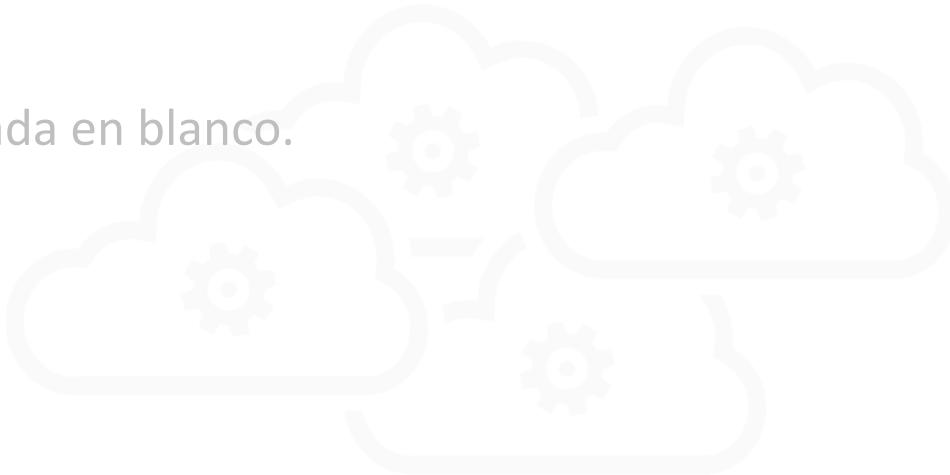
Resumen de la lección

ii.ii Principios de diseño de orientación a servicios

- Analizamos a grandes rasgos la Arquitectura Orientada a Servicios.
- Comprendemos de que trata las Arquitecturas SOA.
- Debatimos referente a las diversas implementaciones SOA en la industria.
- Analizamos el marco de trabajo SOA como un conjunto de prácticas que convierten los activos funcionales de una empresa a servicios para que sean reutilizados a través de sus sistemas separados.



Esta página fue intencionalmente dejada en blanco.



Microservices