

# Servicios



Sistemas de Información Orientados a Servicios

**RODRIGO SANTAMARÍA**

Definiciones

Atributos

Clasificaciones

# Servicios

# Definiciones

3

**¿QUÉ ES UN SERVICIO?  
INTERFAZ  
CONTRATO**

# ¿Qué es un servicio?

4

- “Realización de un trabajo para otra persona”
  - Bastante vago
- “Cualquier tarea de negocio que el cliente pueda entender”
  - Acotación para el mundo empresarial

# Interfaz

5

- **Firma** (*signature*): descripción de parámetros de entrada/salida y posibles excepciones
  - Es la definición de interfaz más común
- La **interfaz** equivale al servicio para el cliente
  - Implica más aspectos para el desarrollador (implementación, backends, frontends, etc.)
- Conocer la firma/interfaz no es suficiente para usar el servicio
  - Como consumidor/cliente, debemos conocer el comportamiento completo y la semántica del servicio
  - Es decir, la interfaz debe estar acompañada de un **contrato**

# Contrato

6

- Especificación completa de un servicio entre un proveedor y un consumidor determinados
- Procedimiento típico
  1. Se describe un servicio como una interfaz bien definida
  2. Se define un contrato específico con un consumidor basado en dicha interfaz, lo que implica
    - Calidad del servicio (**QoS**)
    - Acuerdos particulares (Service Level Agreements - **SLAs**)

# Contrato (ejemplo)

7

- **Sea un servicio de establecimiento de llamadas**
  - Un cliente (central de llamadas) lo contrata para uso diario
    - El SLA garantiza 100.000 llamadas al día entre las 8h y las 14h
    - La llamada al servicio debe ser síncrona y la calidad alta (QoS)
  - Otro cliente contrata el servicio para su uso en segundo plano
    - Su SLA garantiza 50.000 llamadas entre las 0h y las 2h
    - La llamada vale con que sea asíncrona y de calidad media
  - Otro posible SLA podría ser el tiempo de respuesta, etc.

# Atributos

8

**INDEPENDENCIA  
GRANULARIDAD  
OTROS ATRIBUTOS**



# Atributos

9

- Atributo = característica del servicio
- No hay consenso sobre qué atributos son obligatorios, recomendables u opcionales
- Para cada implantación concreta, se deberá definir el grado de importancia de cada atributo
- En general, probablemente el único atributo obligatorio es que el servicio sea **independiente**

# Independencia

10

- Sinónimos: auto-contención, autonomía, autarquía
- Aunque todos los servicios deberían ser independientes, hay ciertas dependencias que casi siempre existirán (por ejemplo, `string`)
  - Deberíamos minimizar las dependencias para que SOA sea aplicable a distintos propietarios

¿Por qué digo que `string` es una dependencia que existirá siempre?

# Granularidad gruesa

11

- Granularidad: tamaño o entidad del servicio
- No hay métrica exacta de qué se considera ‘grueso’
  - La granularidad es relativa al problema a resolver
- Una granularidad gruesa ayuda a separar la estructura interna de la firma o interfaz externa.
- Evaluación de las posibles ventajas y desventajas de mantener un determinado grado de granularidad

¿Se os ocurre algún ejemplo en el que mantener una granularidad gruesa pueda no ser lo mejor?

# Otros atributos

12

- **Visibilidad:** normalmente hay un lugar público donde se puede buscar el servicio o su descripción
- **Sin estado:** generalmente no queremos que el estado del sistema o del servicio se mantenga
  - Habría que definir la duración del estado y si es una cuestión técnica o de negocio
- **Idempotencia:** si un servicio modifica algo y no tenemos respuesta de confirmación, deberíamos asegurar que el servicio es idempotente

# Otros atributos (II)

13

- **Reusabilidad:** idealmente, cada funcionalidad debería implementarse sólo una vez
  - Cuidado, reusabilidad/rendimiento deben estar en equilibrio
- **Composibilidad:** un servicio puede llamar a otros servicios
  - Aspecto tratado en el modelado de procesos de negocio (BPM)
- **Técnico:** podemos tener servicios técnicos independientes que no atiendan a una funcionalidad de negocio

# Otros atributos (III)

14

- **Capacidad QoS/SLA:** atributos no funcionales relacionados con el rendimiento, fiabilidad, etc.
  - Inicialmente, podemos despreocuparnos de estos aspectos
- **Pre/Post-condiciones:** ayudan a identificar la semántica del servicio
  - Requisitos previos que debe satisfacer el consumidor
  - Salida/propiedades garantizadas tras la ejecución del servicio

# Clasificaciones

15

**SERVICIOS BÁSICOS**  
**SERVICIOS COMPUESTOS**  
**SERVICIOS DE PROCESO**  
  
**OTRAS CLASIFICACIONES**

# Clasificaciones

16

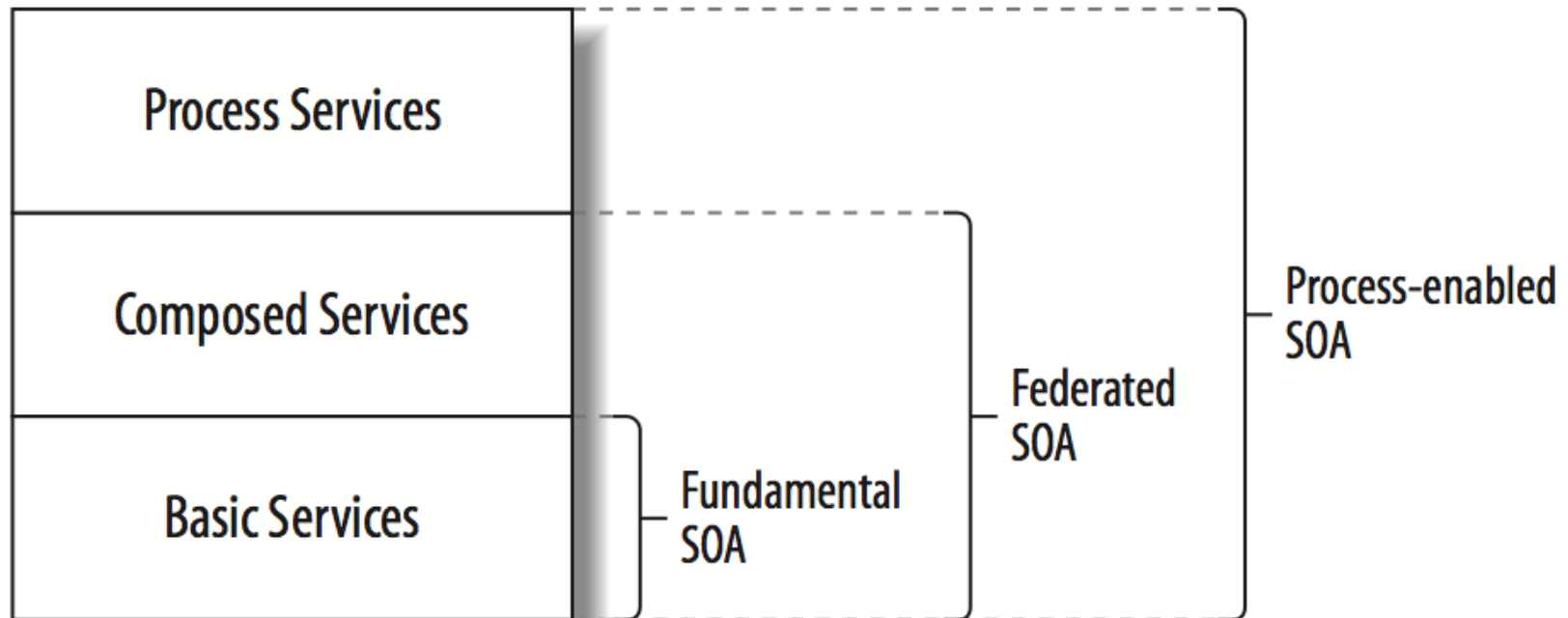
- Hay muchas clasificaciones de los servicios.
- Cada clasificación atiende necesidades diferentes
- Para una determinada clasificación, los tipos de servicios que defina generalmente definirán los atributos de los mismos.



# Clasificación fundamental

17

- Tres tipos de servicio, que definen tres capas SOA



[Josuttis07]

# Servicios Básicos

18

- Composibilidad: **atómicos**, no se dividen en otros
- **Sin estado**, generalmente
- Proveen funcionalidad básica, de tiempo corto de ejecución, usualmente invocados de manera síncrona
- Involucran a un único backend\*
- Dos tipos
  - Básicos de datos
  - Básicos de lógica

# Servicios básicos de datos

19

- **Servicios básicos de lectura/escritura**
  - De datos a un nivel suficientemente alto para tener sentido de negocio
  - Ejemplos
    - Crear un nuevo cliente
    - Cambiar o recuperar la dirección de un cliente
    - Devolver una lista de clientes según un criterio de búsqueda

# Servicios básicos de datos

20

- Estos servicios deben encapsular los detalles técnicos necesarios para evitar inconsistencias en los backends sobre los que operan
- Para ello, deben cumplir con las propiedades ACID
  - Atómico: la llamada se realiza con éxito o no se realiza
  - Consistente: tras la llamada, el backend queda en un estado consistente
  - Aislado: el servicio que está siendo procesado no está influenciado por otras llamadas en ejecución en el mismo backend al mismo tiempo
  - Duradero: si el servicio tiene éxito, su efecto es persistente.
    - Un fallo del sistema no desharrá el resultado de la llamada

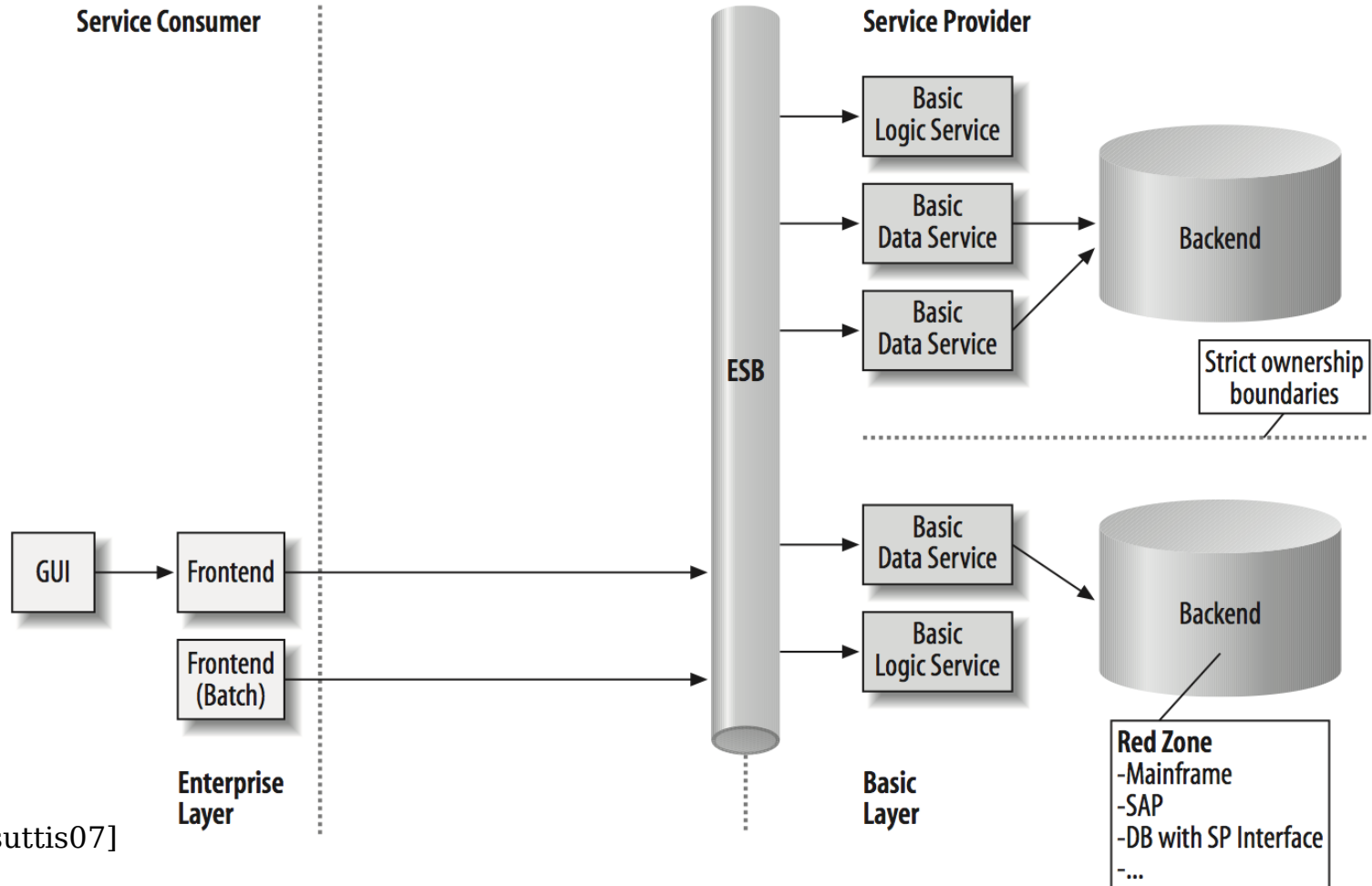
# Servicios básicos de lógica

21

- Definen reglas fundamentales de negocio
- Ejemplos
  - Definir catálogos de producto y listas de precios
  - Decidir si un determinado año es bisiesto
  - Decidir si una fecha es fiesta o no
- Suelen ser minoría, y generalmente ya están definidos en el legado del sistema
  - Por ello a menudo son simplemente envoltorios

# SOA fundamental

22



# Backend

23

- Cualquier entidad responsable de un grupo específico de datos y funcionalidad, por ejemplo
  - Una base de datos
  - Un servidor web
  - Un servidor de altas prestaciones
  - Un motor de reglas
  - Un grupo de servidores J2EE
  - Una conexión remota a otra compañía
- Representan el núcleo esencial de un sistema
  - Nunca debe corromperse o volverse inconsistente

# Servicios compuestos

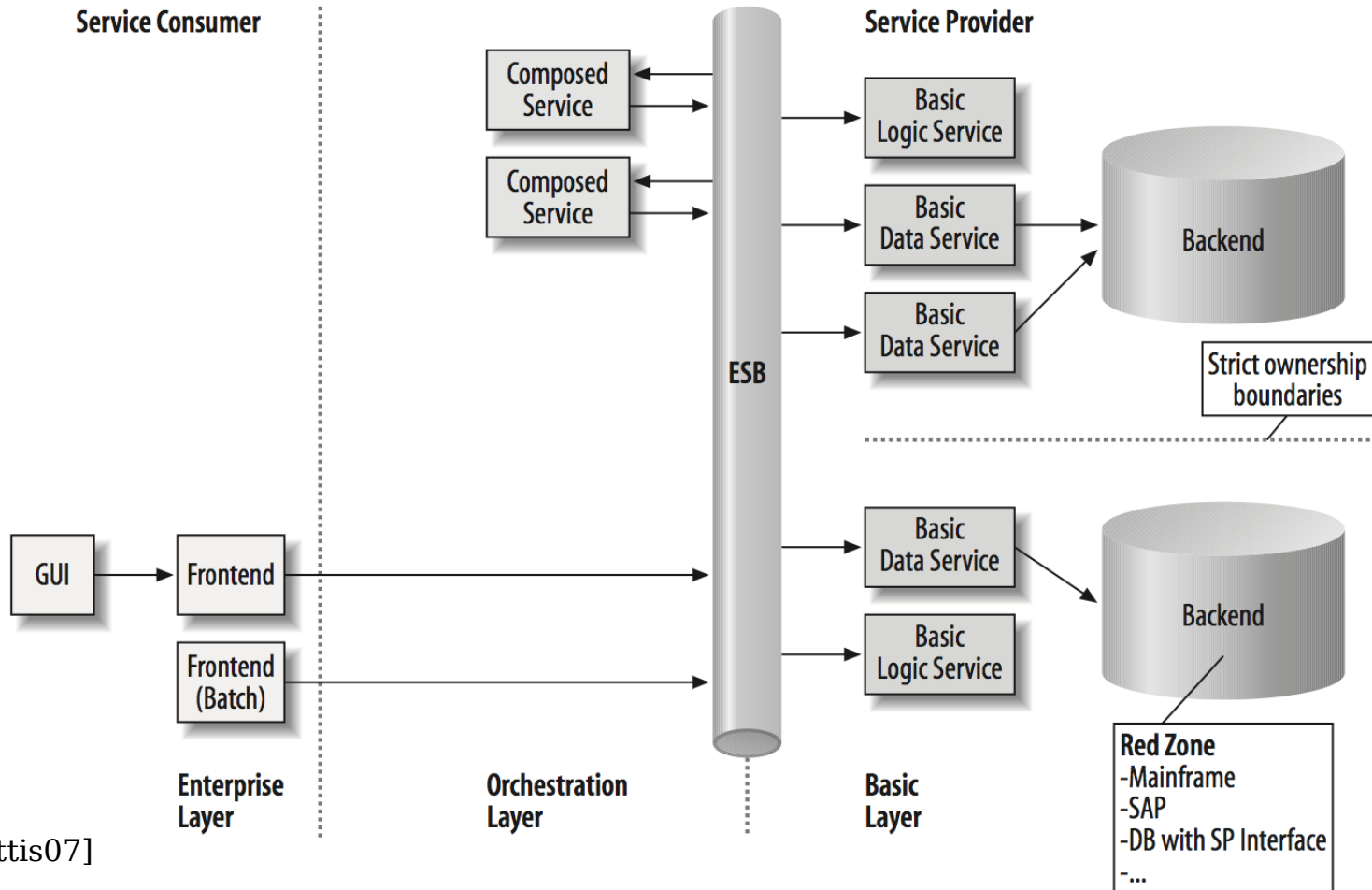
24

- **Orquestación:** composición de nuevos servicios a partir de servicios existentes
- Ejecución corta y sin estado
- Generalmente asociados a múltiples backends
- Deben cumplir con las propiedades ACID
- Ejemplos
  - Modificar la dirección de un cliente en todos los backends
  - Transferir dinero de un backend a otro



# SOA federada

25



[Josuttis07]

# Servicios compuestos y múltiples backends

26

- **Un servicio compuesto necesitará**
  1. Conocer qué backends tiene que acceder
  2. Mapear los datos de entrada a los atributos específicos de los servicios básicos de los distintos backends
  3. Reaccionar apropiadamente si uno o más backends fallan
- **Este último aspecto (consistencia) puede tratarse**
  - Usando *commit de dos fases* (2PC)
  - Usando algún tipo de *compensación*

# Consistencia: 2PC

27

- Algoritmo distribuido para transacciones atómicas

1. Votación

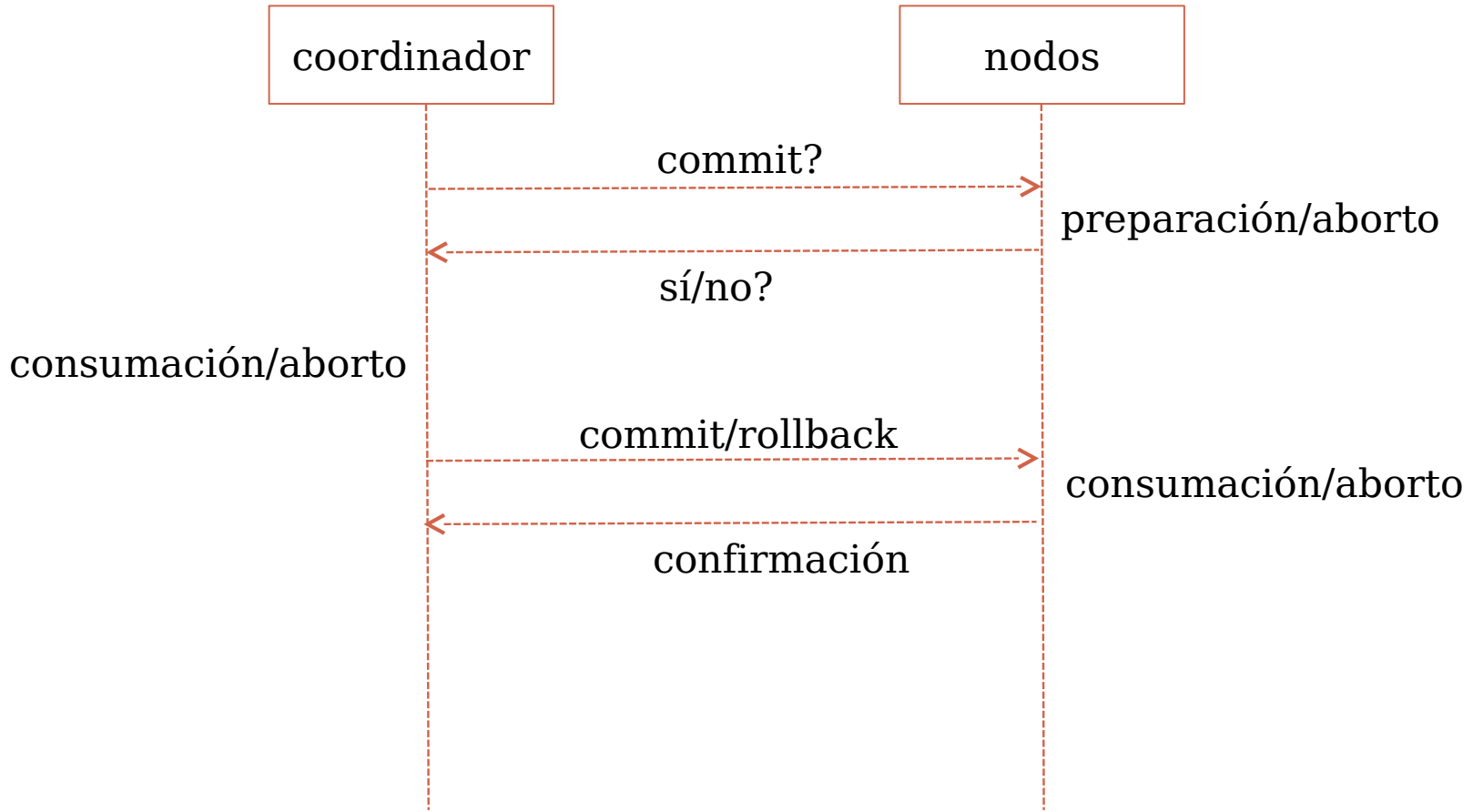
- El coordinador manda a los nodos implicados una petición de consumación
- Cada nodo ejecuta la transacción hasta el punto de consumación
- Cada nodo responde afirmativa o negativamente

2. Consumación

- **Éxito:** si todos los nodos responden afirmativamente, el coordinador multidifunde un mensaje de consumación, y cada nodo realiza la operación y confirma al coordinador
- **Fallo:** en caso contrario, el coordinador multidifunde un mensaje de retroceder y cada nodo aborta y responde con una confirmación

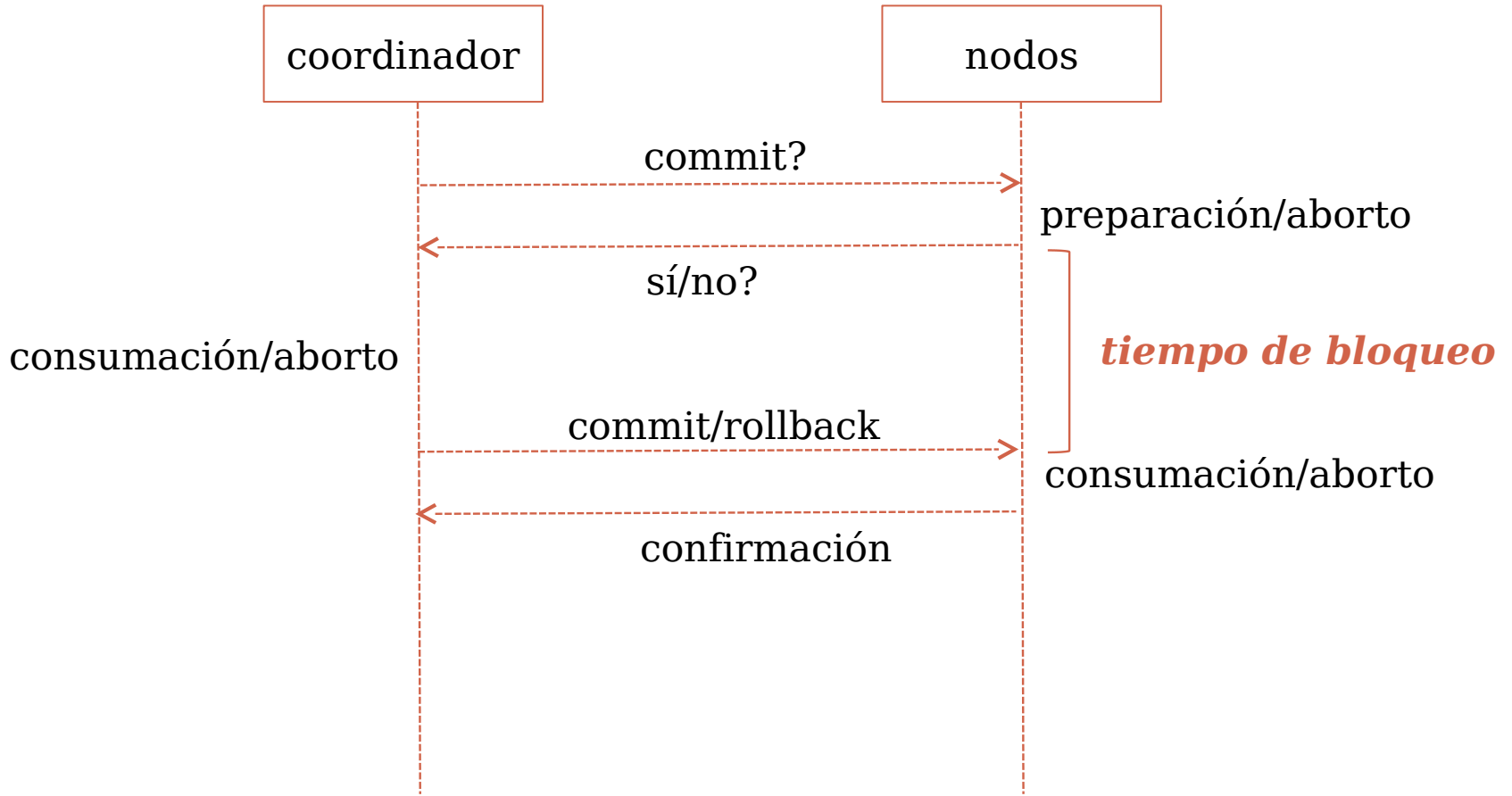
# Consistencia: 2PC

28



# Consistencia: 2PC

29



# Consistencia: compensación

30

- **Opción no bloqueante**
  - Cada nodo (backend) se modifica por separado
  - Si alguna modificación no tiene éxito, se ‘compensa’ el problema de algún modo, p. ej.:
    - Se vuelve al estado anterior en los nodos que aplicaran la modificación con éxito
    - Se envía un informe de error a un operario para que trate con él manualmente
- **Opción más común que el 2PC**
  - Sobre todo en sistemas grandes, donde garantizar 2PC en todos los backends puede llevar a grandes retardos o incluso a interbloqueos

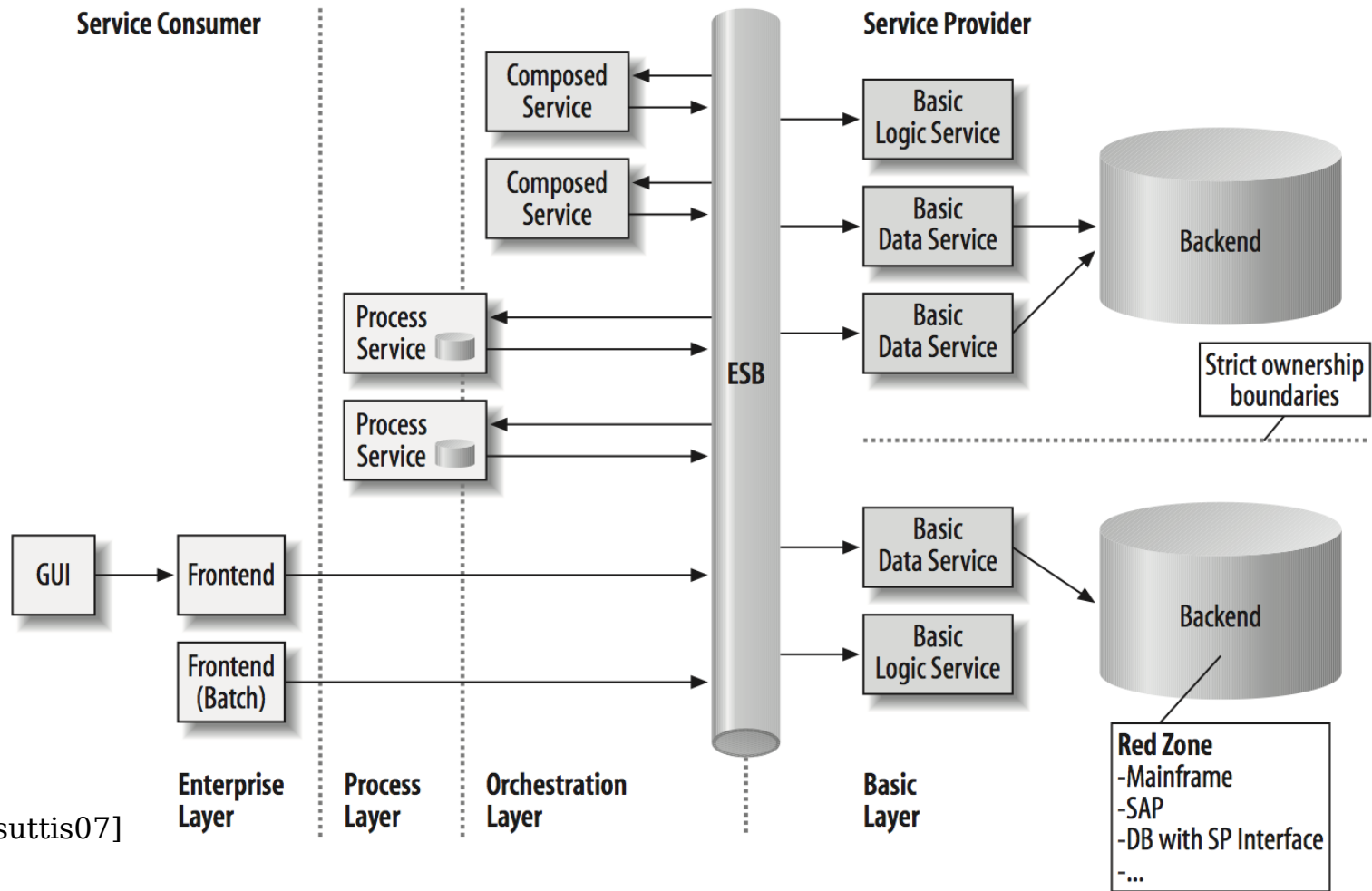
# Servicios de proceso

31

- Servicios de ejecución larga interrumpibles
- Su estado permanece durante llamadas sucesivas
- Ejemplo: compra de productos
  - El estado permanece (productos añadidos a la cesta)
  - Varios backends: portal web, sistema de transacciones, etc.
  - El servicio puede terminar por distintas causas

# SOA de procesos

32



[Josuttis07]



# Otras clasificaciones

33

*El ser humano ha dividido la naturaleza en 4 partes y cree que conoce la naturaleza, cuando lo único que ha hecho es dividirla*  
Masanobu Fukuoka

- La clasificación mostrada arriba es la propuesta por [Josuttis07], pero existen variaciones de esta misma
  - Además de otras clasificaciones totalmente distintas
    - Servicios internos/externos
    - Servicios técnicos/de negocio
    - Servicios de lectura/escritura
    - Etc.
- La clasificación de los servicios debe basarse en la comprensión del sistema y de la solución a adoptar
  - Ley de Conway (1967): *“Cualquier organización que diseña un sistema producirá un diseño cuya arquitectura es una copia de su estructura de comunicación”*

# Resumen

34

- Un **servicio** es simplemente una tarea a realizar.
- En SOA, se entiende **una interfaz bien definida** (sinónimo de servicio) como una descripción de entrada/salida, proceso interno y semántica
- La única característica común de un servicio es la **independencia**
- La relevancia y valor de otros atributos dependerá del dominio del problema, siendo algunos la **granularidad** (gruesa), el **estado** (sin estado), la composibilidad, etc.
- La **clasificación fundamental** diferencia entre servicios básicos, compuestos y de proceso
- Los servicios **básicos** no tienen estado, son pequeños, síncronos y asociados a un backend
- Los servicios **compuestos** no tienen estado, son pequeños, utilizan varios servicios básicos y backends
- Los servicios de **proceso** tienen estado, son grandes e interrumpibles
- Esta es una posible clasificación, pero hay que entender el problema y sistema al que nos enfrentamos y clasificar nuestros servicios de manera acorde

# Referencias

35

- [Josuttis07] Nicolai M. Josuttis. *SOA in practice. The Art of Distributed System Design*. O'Reilly, **2007**. Ch 3/6.
- [Gartner03] Yefim V. Natis. Service-Oriented Architecture Scenario. Gartner, **2003**.
  - <https://www.gartner.com/doc/391595>
- OASIS. *Reference model for service oriented architectures*. Committee Draft 1.0, **2007**
  - <http://xml.coverpages.org/SOA-RM-ReferenceModel200602-CD.pdf>

