Tecnologías avanzadas para el desarrollo de Web Services

Parte 2

Agenda

- WS-Addressing
- MTOM
- Consistencia de datos
  - WS-Transaction
  - WS-Business Activity
  - Consistencia eventual

Mensajería

WS-Addressing, MTOM

SOAP

- Provee una forma estándar de estructurar mensajes utilizando XML
- Neutralidad en protocolo de transporte
- Especifica un modelo de procesamiento que indica cómo se deben procesar los mensajes
- Una forma de adjuntar datos no-XML a los mensajes.

SOAP

- Provee una forma estándar de estructurar mensajes utilizando XML
- Neutralidad en protocolo de transporte
- Especifica un modelo de procesamiento que indica cómo se deben procesar los mensajes
- Una forma de adjuntar datos no-XML a los mensajes.

Neutralidad en transporte

- SOAP no impone el uso de un determinado protocolo para el intercambio de mensajes

- A través del concepto de "*binding*" SOAP permite especificar:
  - cómo los mensajes SOAP se encapsulan en un protocolo de transporte
  - cómo los mensajes SOAP deben ser tratados con las primitivas del protocolo

Sin embargo…

- Existen dependencias en protocolos de transporte

Sin embargo…

- Existen dependencias en los protocolos de transporte
  - HTTP URI
    - Determinar dirección del Web Service
  - Cabezales http: SoapAction
    - Determinar el método del Web Service
    - Creador de problemas de interoperabilidad
    - Obsoleto a partir de SOAP 1.2!

Debilidades

- Necesidad de colocar información en el protocolo de forma estándar
  - Dirección del servicio
  - Operación a consumir
  - Otra información
- Dificultad al enviar mensajes a través de intermediarios utilizando diferentes protocolos
- Mecanismos de seguridad “atados” al protocolo de transporte
- Solución:
  - colocar información en cabezales SOAP
  - Mantener neutralidad y eliminar dependencias en protocolo de transporte

WS-Addressing

WS-Addressing

- Estándar de la W3C
  - Actualmente en versión 1.0

- Propone un mecanismo estándar e independiente del transporte para:
  - Referenciar Web Services
  - Direccionar mensajes

Endpoint Reference

- Es un “puntero” a un Web Service.
- Especifica:
  - Address
  - Reference Parameters
  - Metadata.

```
<wsa:EndpointReference xmlns:wsa="..." xmlns:example="...">
    <wsa:Address>http://example.com/weather</wsa:Address>
    <wsa:ReferenceProperties>
        <example:ServiceLevel>Basic</example:ServiceLevel>
    </wsa:ReferenceProperties>
    <wsa:ReferenceParameters>
        <example:CityCode>NYC</example:CityCode>
    </wsa:ReferenceParameters>
</wsa:EndpointReference>
```

Message Addressing Props

- Definen el conjunto de cabezales SOAP con información para el direccionamiento de mensajes.

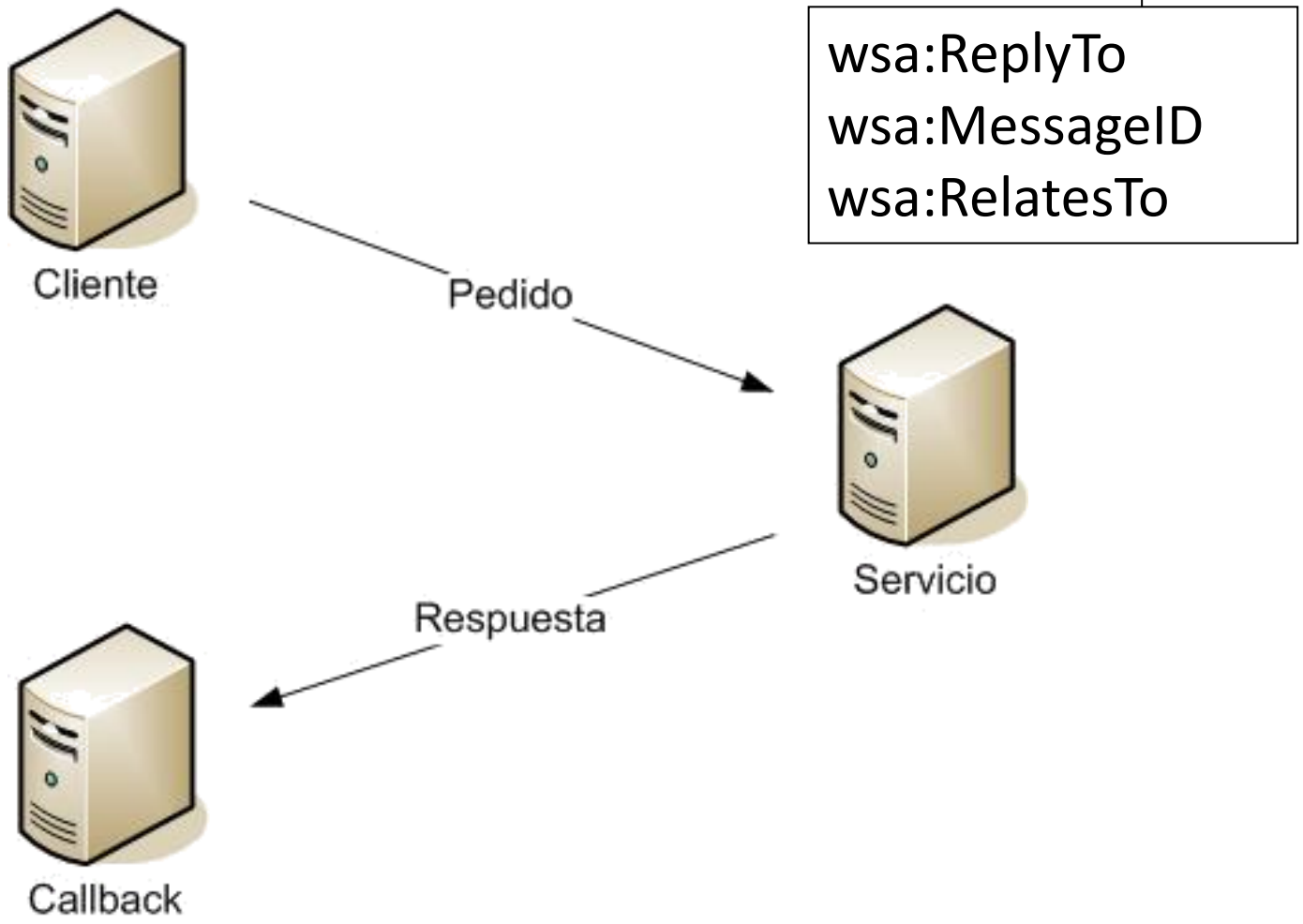
- Elementos requeridos: *To* y *Action*
  - **To**: Indica a donde se envía el pedido.
  - **Action**: Indica la acción a realizar por el receptor del mensaje.

```
<wsa:To>http://example.com/fabrikam/Purchasing</wsaTo>
<wsa:Action>http://example.com/fabrikam/SubmitPO</wsa:Action>
```

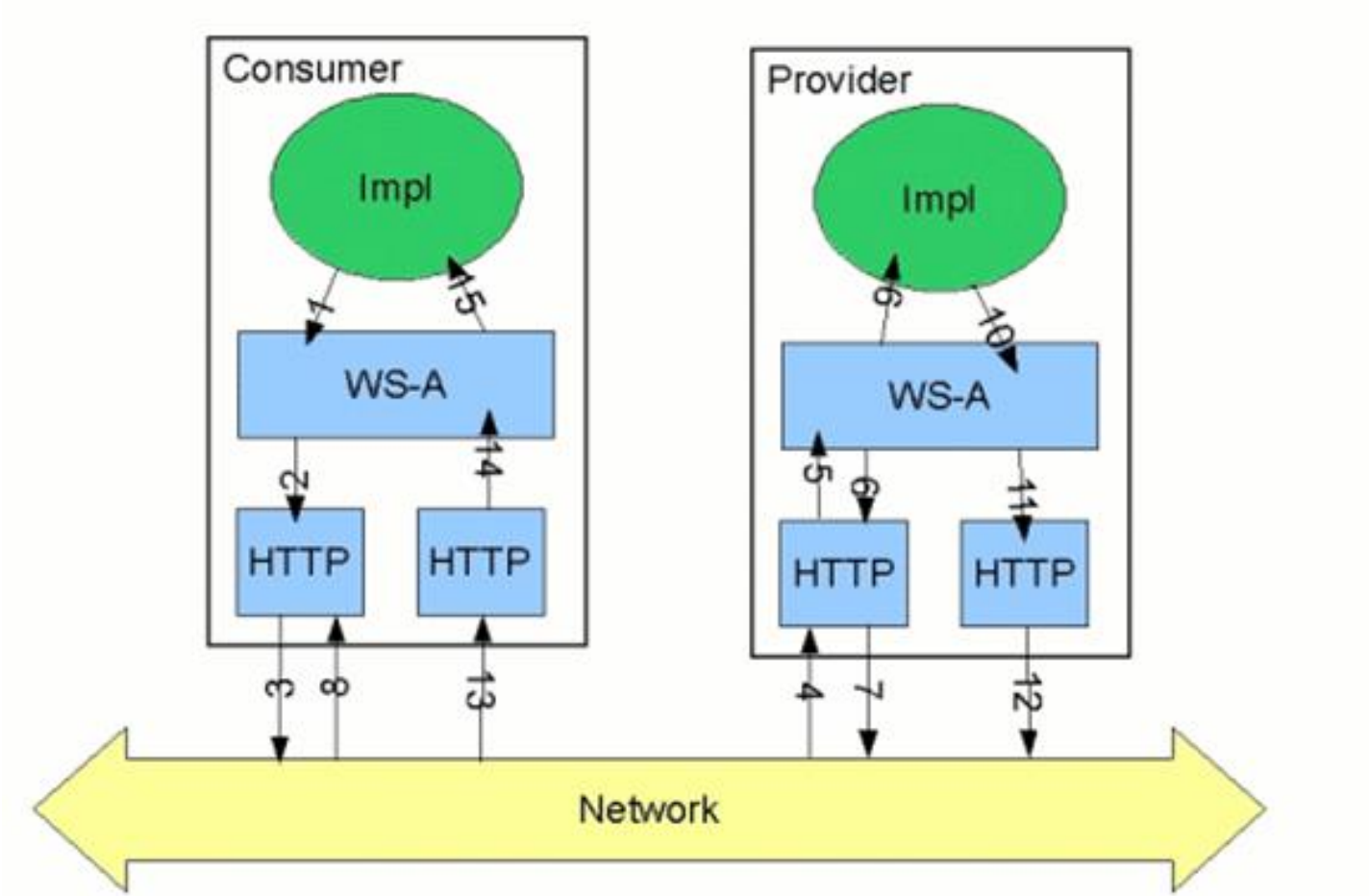
Message Addressing Props

- **ReplyTo**
  - Especifica la dirección de un EPR al que se debe enviar la respuesta a una solicitud.
- **FaultTo**
  - Especifica la dirección de un EPR al que se debe invocar cuando ocurra una situación anómala y no se pueda terminar de procesar la solicitud
- **MessageID**
  - String que identifica de forma univoca el mensaje.
- **RelatesTo**
  - String que permite correlacionar mensajes
  - Complementa ReplyTo para dar solución al uso de respuestas desacopladas entre WS.

Respuestas desacopladas



Respuestas desacopladas



Ejemplo de solicitud

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:To>http://example.com/fabrikam</wsa:To>
    <wsa:Action>http://example.com/fabrikam/Delete</wsa:Action>
    <wsa:MessageID>http://example.com/someuniquestring</wsa:MessageID>
    <wsa:ReplyTo>
          <wsa:Address>http://example.com/business/client1</wsa:Address>
    </wsa:ReplyTo>
  </S:Header>
  <S:Body>
    <f:Delete xmlns:f="http://example.com/fabrikam">
      <maxCount>42</maxCount>
    </f:Delete>
  </S:Body>
</S:Envelope>
```

Ejemplo de respuesta

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
    xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <S:Header>
    <wsa:To>http://example.com/business/client1</wsa:To>
    <wsa:Action>http://example.com/fabrikam/DeleteAck</wsa:Action>
     <wsa:MessageID>http://example.com/otheruniquestring</wsa:MessageID>
    <wsa:RelatesTo>http://example.com/someuniquestring</wsa:RelatesTo>
</S:Header>
  <S:Body>
    <f:DeleteAck xmlns:f="http://example.com/fabrikam"/>
  </S:Body>
</S:Envelope>
```

Aspectos destacables

- Identificación de mensajes

- Respuestas desacopladas
  - Respuestas correctas
  - SOAP Faults
  - Correlación de mensajes

MTOM

Message Transmission Optimization Mechanism

Motivación

- SOAP
  - Provee una forma estándar de estructurar mensajes utilizando XML
  - Define mecanismos para utilizar distintos protocolos de transporte para el envío de mensajes
  - Especifica un modelo de procesamiento que indica cómo se deben procesar los mensajes
  - Una forma de adjuntar datos no-XML a los mensajes.

Cómo lo hace?

- SOAP utiliza la codificación a Base64 para convertir datos binarios a XML

- XMLSchema soporta el uso de este tipo de serialización
  - tipo de datos *xsd:base64binary*

Codificación base64

- Base 64 toma los datos binarios y los convierte en una serie de caracteres ASCII
- Toma 3 octetos de bits y los convierte en 4 caracteres del estándar ASCII
  - Se usan solo 64 caracteres del estándar
  - a-z, A-Z, 0-9, + y /
- Ejemplo
  - Mary had a little lamb...
  - TWFyeSBoYWQgYSBsaXR0bGUgbGFtYi4uLiA=

Codificación base64

Texto ASCII	Mary had
Representación hexadecimal	4D 61 72 79 20 68 61 64
Representación binaria agrupada en bytes	01001101 01100001 01110010 01111001 00100000 01101000 01100001 01100100
Representación binaria agrupada de a 6 bits	010011 010110 000101 110010 011110 010010 000001 101000 011000 010110 010000=
Representación decimal de los bloques de 6 bits	19 22 05 50 30 18 01 40 24 22 16=
Codificación base64	TWFyeSBoYWQ=

Sin embargo…

- Tamaño del texto un %25 más grande
- En base64 6 bits son codificados con 8 bits
  - Base64(010011) = T (8 bits)
- Gran impacto en datos binarios de gran tamaño!

MTOM (Message Transmission Optimization Mechanism)

- MTOM es un estándar de la W3C con el propósito de optimizar la transferencia de archivos binarios entre Web Services.

- MIME gran protagonista!

MTOM

- Compuesto por:
  - Abstract SOAP Transmission Optimization Feature
    - mecanismo de optimización abstracto para mensajes SOAP
  - Optimized MIME multipart/Related Serialization of SOAP messages
    - Implementación basada en XOP e indep. del medio de transporte
  - HTTP SOAP Transmission Optimization Feature
    - extensión para ser usada con HTTP

Ejemplo: Serialización original

```
HTTP/1.1 200 OK
Content-Length: 4542508
Content-Type: text/xml; charset=utf-8

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/ ">
  <s:Body>
    <GetDataResponse xmlns="http://tempuri.org/ ">
      <GetDataResult>
   UesDBAoAAAAAAGRoaTsAAAAAAAAA..../Cj8KBxAEAG9xMgAHAFBBQ0syMDA=
      </GetDataResult>
    </GetDataResponse>
  </s:Body>
</s:Envelope>
```

Ejemplo: Serialización MTOM

```
HTTP/1.1 200 OK
Content-Length: 3407478
Content-Type:          multipart/related;          type="application/xop+
start="<http://tempuri.org/0>";          boundary="uuid:774aed72-9fbd-43ba-
24a5a025899e+id=2"; start-info="text/xml"
Server: Microsoft-HTTPAPI/1.0
MIME-Version: 1.0
Date: Fri, 03 Sep 2010 18:33:41 GMT

--uuid:774aed72-9fbd-43ba-a09a-24a5a025899e+id=2
Content-ID: <http://tempuri.org/0>
Content-Transfer-Encoding: 8bit
Content-Type: application/xop+xml;charset=utf-8;type="text/xml"

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDataResponse xmlns="http://tempuri.org/">
      <GetDataResult>
        <xop:Include
          href="cid:http%3A%2F%2Ftempuri.org%2F1%2F634191248212968750"
            xmlns:xop="http://www.w3.org/2004/08/xop/include"/>
      </GetDataResult>
    </GetDataResponse>
  </s:Body>
</s:Envelope>

--uuid:774aed72-9fbd-43ba-a09a-24a5a025899e+id=2
Content-ID: <http://tempuri.org/1/634191248212968750>
Content-Transfer-Encoding: binary
Content-Type: application/octet-stream

PK##
□□□□dhi;□□□□□□□□□□□□      □□□META-INF/PK##
```

Ejemplo: Serialización MTOM

- Content-Type:
  - Deja de ser text/xml para utilizar multipart/related
  - Se agregan a su vez tres elementos:
    - Tipo de adjuntos: application/soap+xml
    - Boundary: frontera de cada fragmento del mensaje multiparte
    - Boundary incial: indica dónde comienza el mensaje

Ejemplo: Serialización MTOM

- Cada Boundary contiene la siguiente información:
  - Content-type: tipo de mensaje contenido en el boundary
  - Content-ID: identificador del boundary

- En el mensaje soap se sustituye el archivo binario por una referencia a una boundary

Transacciones

WS-Coordination

WS-AtomicTransaction

WS-BusinessActivity

WS-Transactions

- Compuesto por 3 estándares de la OASIS
  - WS – Coordination
  - WS – AtomicTransaction
  - WS – BusinessActivity

- Actualmente, se encuentran en la versión 1.2

WS-AtomicTransaction

Transacciones de corta duración

Transacciones

- Son un conjunto de tareas que cumplen las propiedades ACID

Transacciones

- Son un conjunto de tareas que cumplen las propiedades ACID
  - Atomicidad (Atomicity)
    - Se realizan todas las tareas o ninguna

Transacciones

- Son un conjunto de tareas que cumplen las propiedades ACID
  - Atomicidad (Atomicity)
    - Se realizan todas las tareas o ninguna
  - Consistencia (Consistency)
    - Las tareas realizadas no violan ninguna de las restricciones de integridad

Transacciones

- Son un conjunto de tareas que cumplen las propiedades ACID
  - Atomicidad (Atomicity)
    - Se realizan todas las tareas o ninguna
  - Consistencia (Consistency)
    - Las tareas realizadas no violan ninguna de las restricciones de integridad
  - Aislamiento (Isolation)
    - Las tareas no pueden acceder o ver datos que se encuentren en estados intermedios.

Transacciones

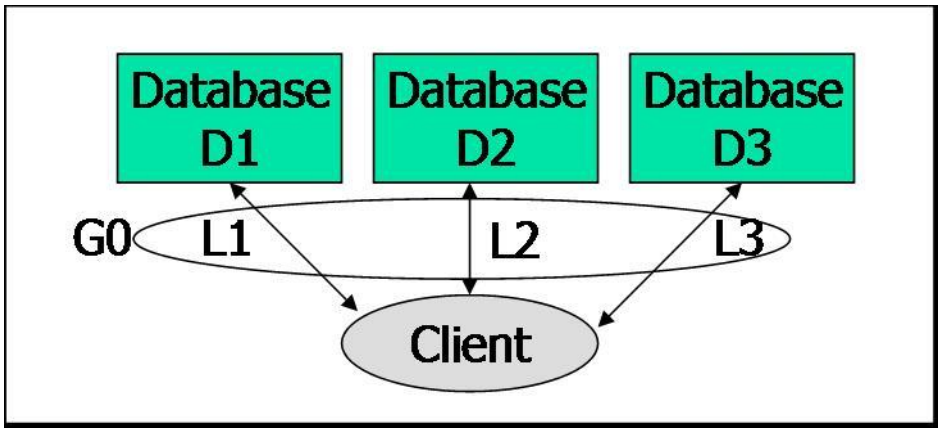
- Son un conjunto de tareas que cumplen las propiedades ACID
  - Atomicidad (Atomicity)
    - Se realizan todas las tareas o ninguna
  - Consistencia (Consistency)
    - Las tareas realizadas no violan ninguna de las restricciones de integridad
  - Aislamiento (Isolation)
    - Las tareas no pueden acceder o ver datos que se encuentren en estados intermedios.
  - Durabilidad (Durability)
    - Una vez realizada la transacción, se garantiza que esta persistirá a pesar de fallas en el sistema

Transacciones distribuidas

- Bases de datos independientes con sus propias transacciones locales (Li)

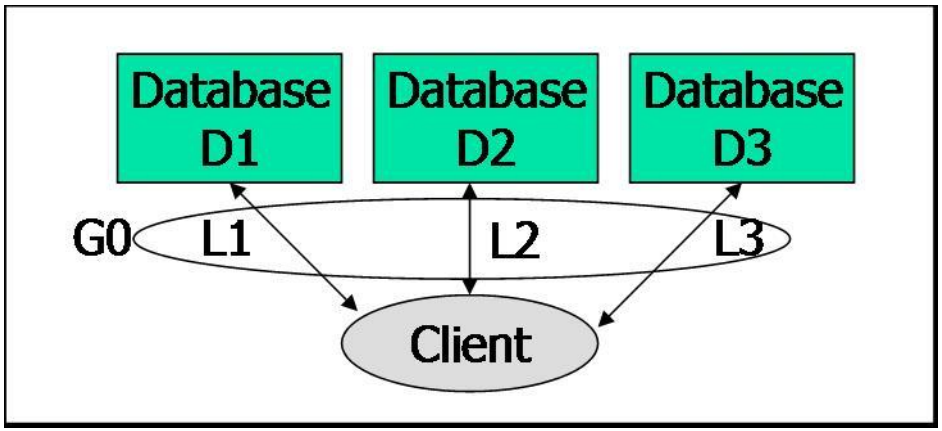
Transacciones distribuidas

- Bases de datos independientes con sus propias transacciones locales (Li)
- Una única transacción global (G0)



Transacciones distribuidas

- Bases de datos independientes con sus propias transacciones locales (Li)
- Una única transacción global (G0)



- Deseable mantener propiedades ACID!

Transacciones distribuidas

- Verificar propiedades ACID:
  - ACID(Li) => ACID(G0)

Transacciones distribuidas

- Verificar propiedades ACID:
  - ACID(Li) => ACID(G0)
    - *C(L1),C(L2),C(L3) => C(G0)*
      - *Si cada base de datos mantiene la consistencia, la transacción global es consistente*

Transacciones distribuidas

- Verificar propiedades ACID:
  - ACID(Li) => ACID(G0)
    - *C(L1),C(L2),C(L3) => C(G0)*
    - *I(L1),I(L2),I(L3) => I(G0)*
      - *Si cada base de datos mantiene los datos no visibles a otros actores, la transacción global cumple con la propiedad de aislación*

Transacciones distribuidas

- Verificar propiedades ACID:
  - ACID(Li) => ACID(G0)
    - *C(L1),C(L2),C(L3) => C(G0)*
    - *I(L1),I(L2),I(L3) => I(G0)*
    - *D(L1),D(L2),D(L3) => D(G0)*
      - *Si cada transacción persiste los datos y los mantiene a pesar de fallas en los sistemas, también lo hará la transacción global.*

Transacciones distribuidas

- Verificar propiedades ACID:
  - ACID(Li) => ACID(G0)
    - *C(L1),C(L2),C(L3) => C(G0)*
    - *I(L1),I(L2),I(L3) => I(G0)*
    - *D(L1),D(L2),D(L3) => D(G0)*

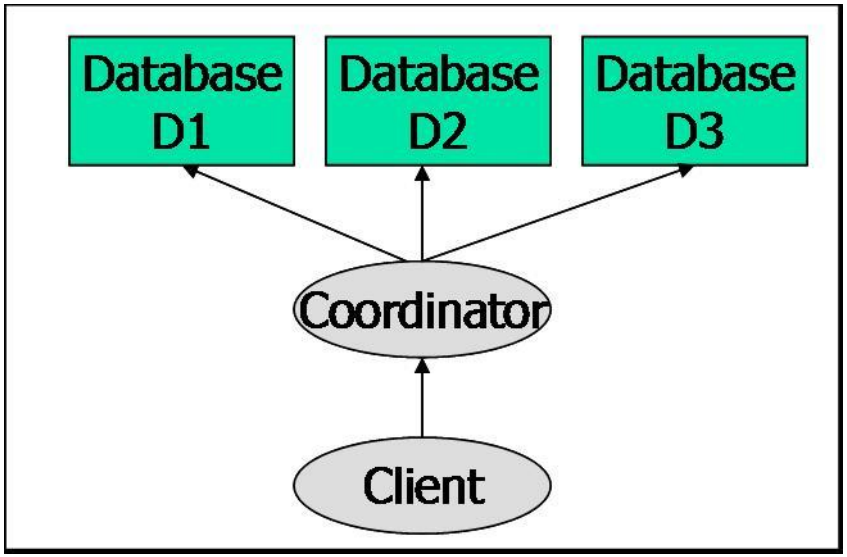
- La atomicidad es un problema!
  - *Abort(L1),Commit(L2),Commit(L3)*
    - *Localmente cada transacción mantiene la atomicidad pero la combinación no mantiene la atomicidad global*

Transacciones distribuidas

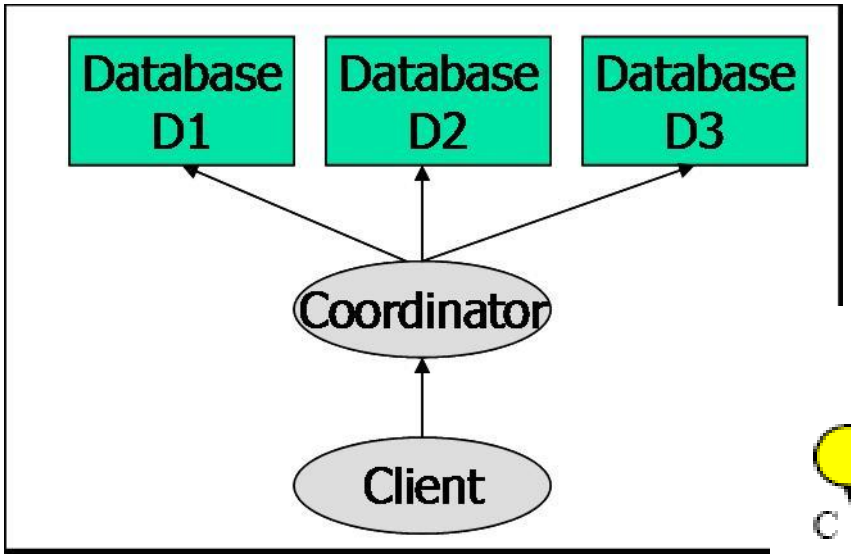
- Solución: agregar una operación *preparar* y un coordinador de la transacción global

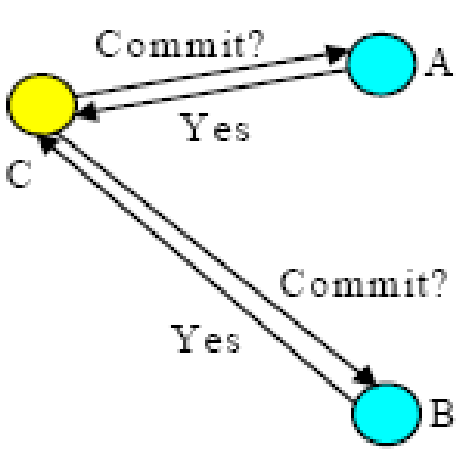
- El coordinador pregunta a cada participante
  - *Preparado para hacer commit?*
  - *Commit o Abort*
- Si alguno de los participantes aborta, la transacción global aborta
- A este protocolo se lo conoce como Two-Phase Commit

Two-Phase Commit (2PC)

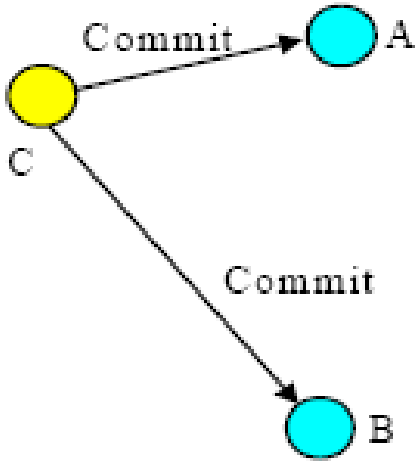


Two-Phase Commit (2PC)





Phase 1



Phase 2

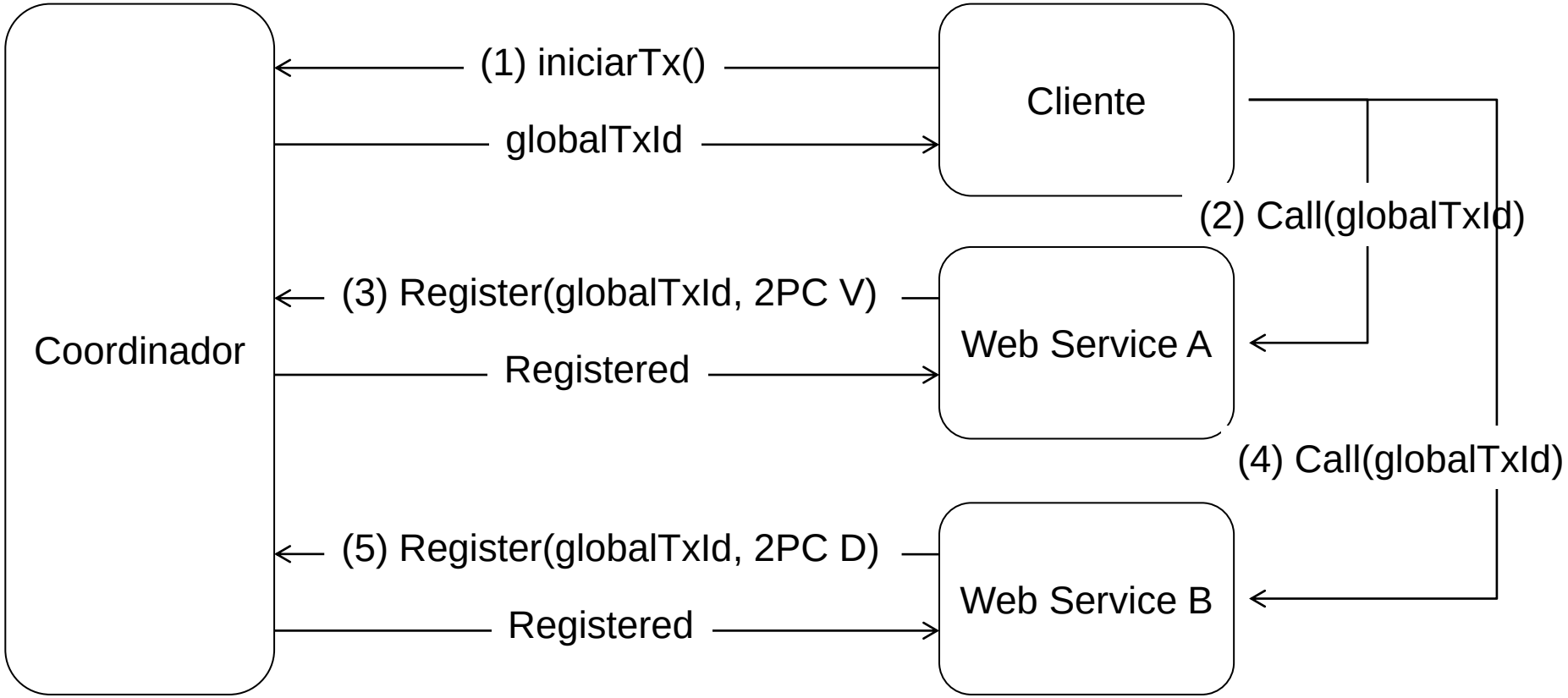
WS-AtomicTransaction

- WS-AT es un estándar de la OASIS con el propósito de proveer propiedades ACID a los Web Services.
- Protocolos
  - Durable Two-Phase Commit
  - Volatile Two-Phase Commit
  - Completition
- Actualmente, se encuentra en la versión 1.2

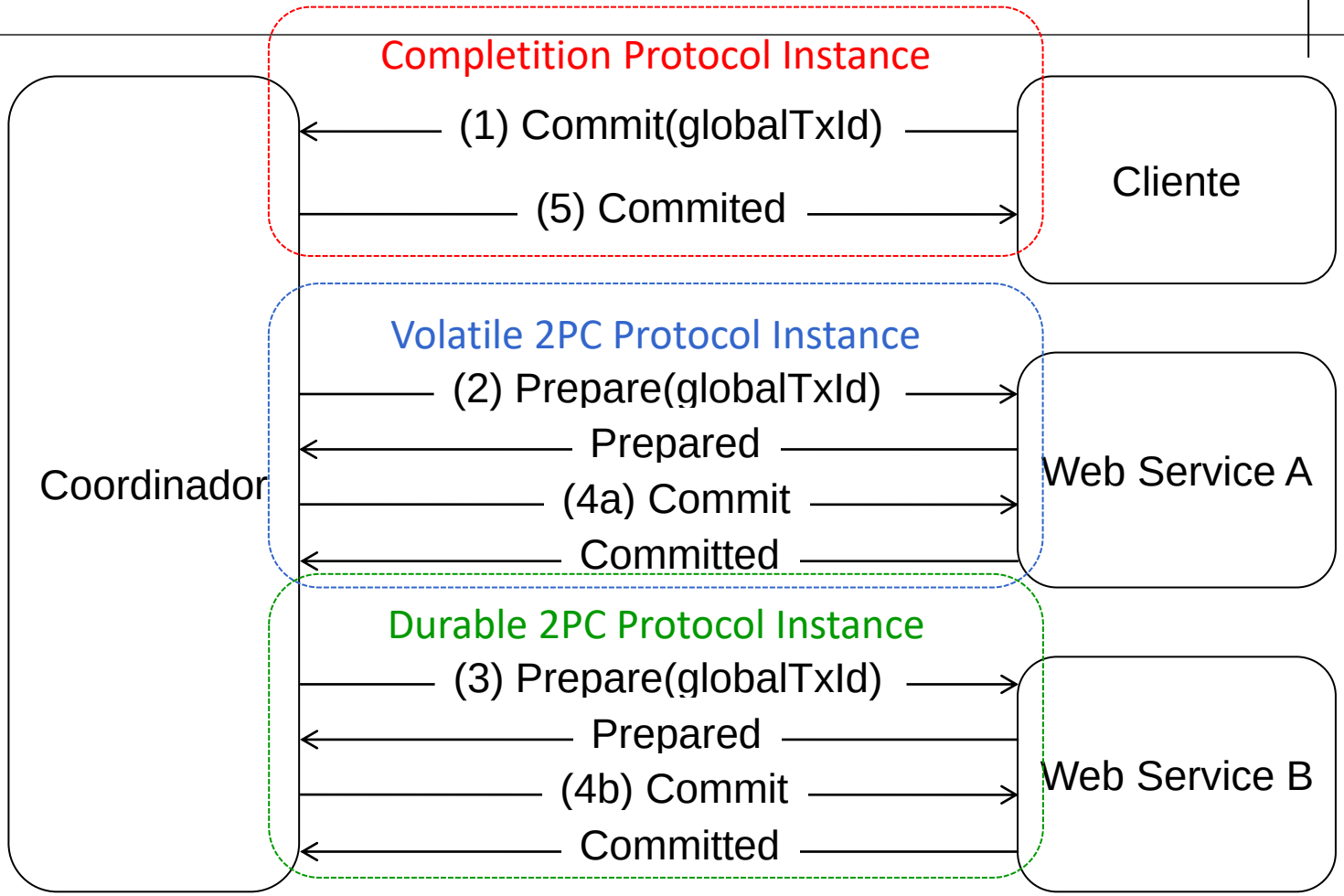
Protocolos

- Completition
  - Este protocolo es utilizado por los clientes con el fin de intentar confirmar o abortar la transacción
  - El resultado es la confirmación o cancelación de la transacción
- Volatile Two-Phase Commit
  - Aplica el protocolo 2PC a recursos volátiles (p. ej: cache, colas de mensajes en memoria)
- Durable Two-Phase Commit
  - Aplica el protocolo 2PC a recursos durables en el tiempo (p. ej: bases de datos, colas de mensajes persistentes)

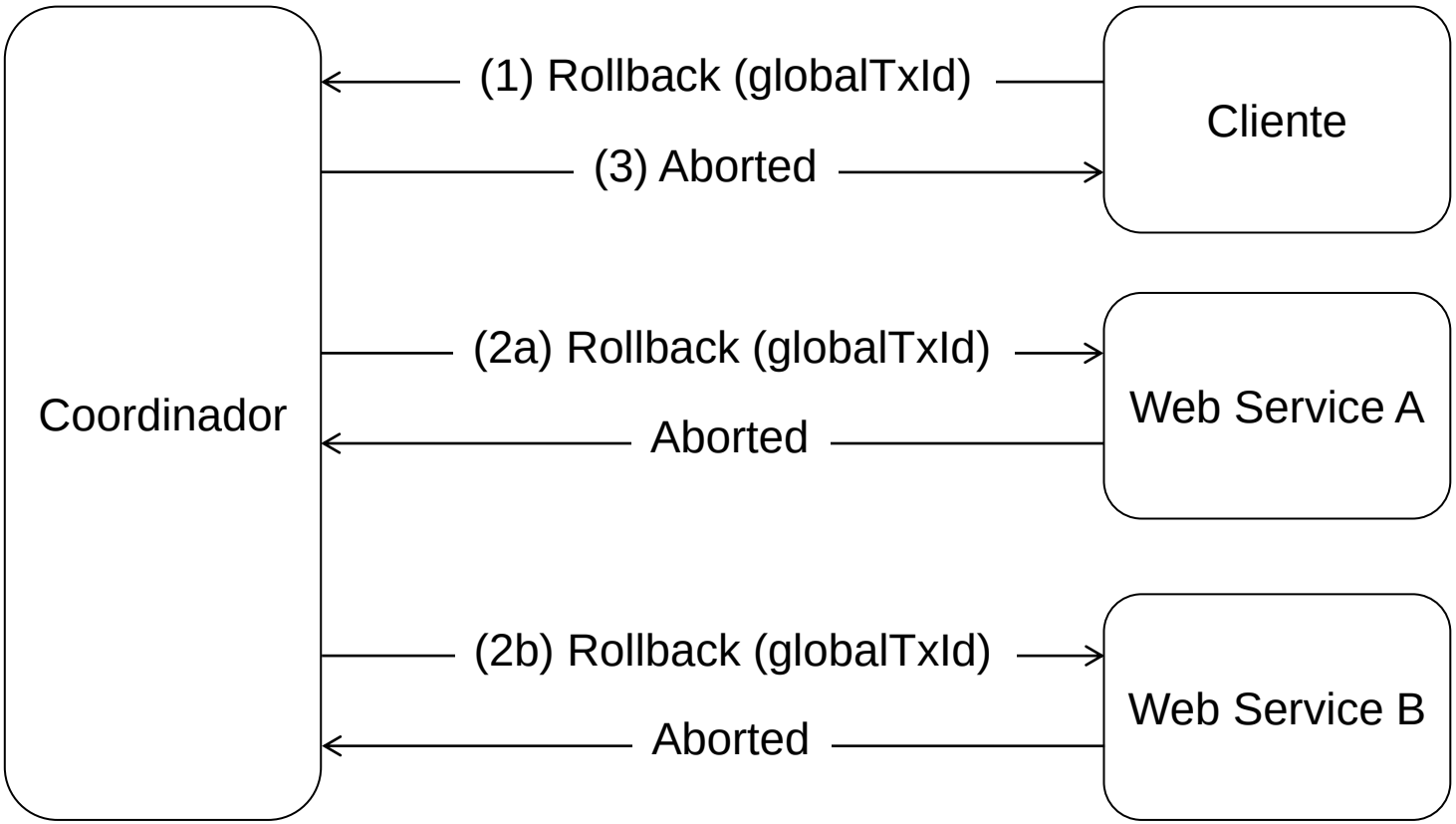
Atomic Transaction



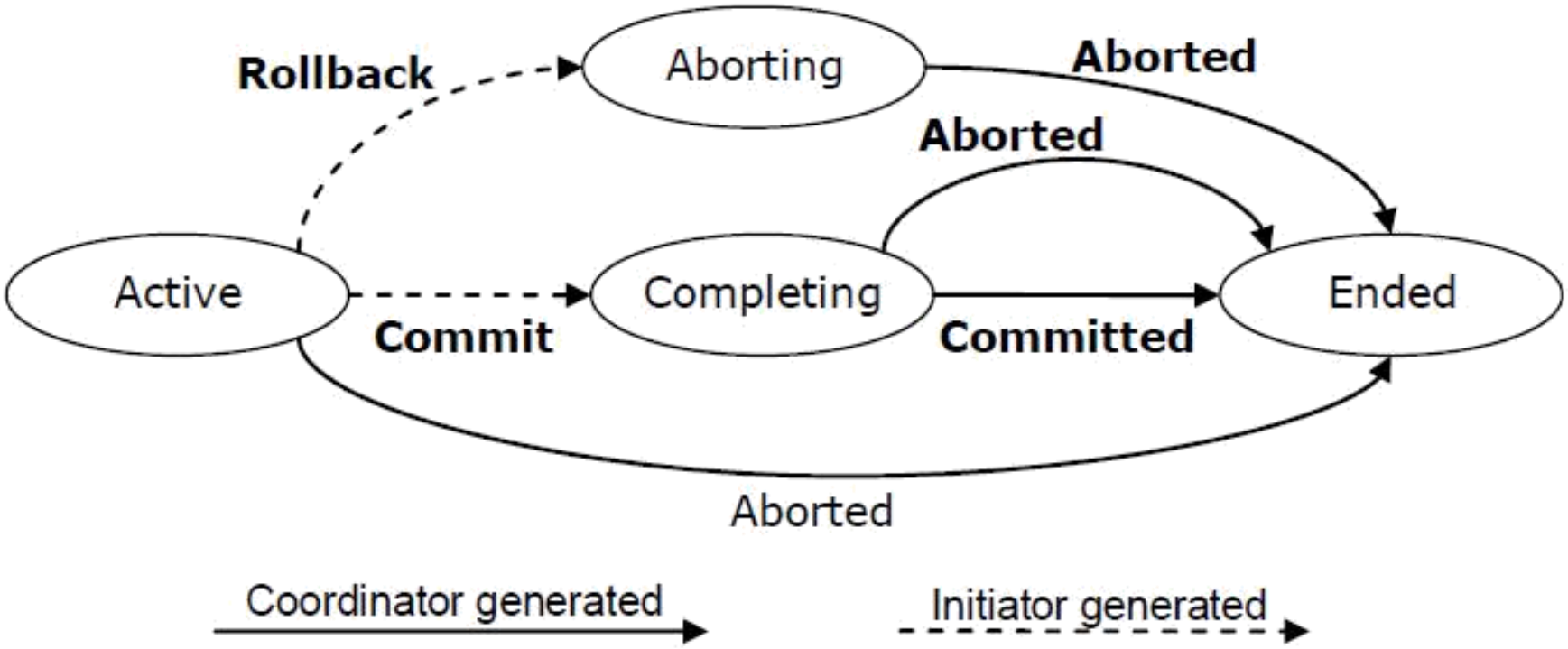
Atomic Transaction Commit



Atomic Transaction Abort

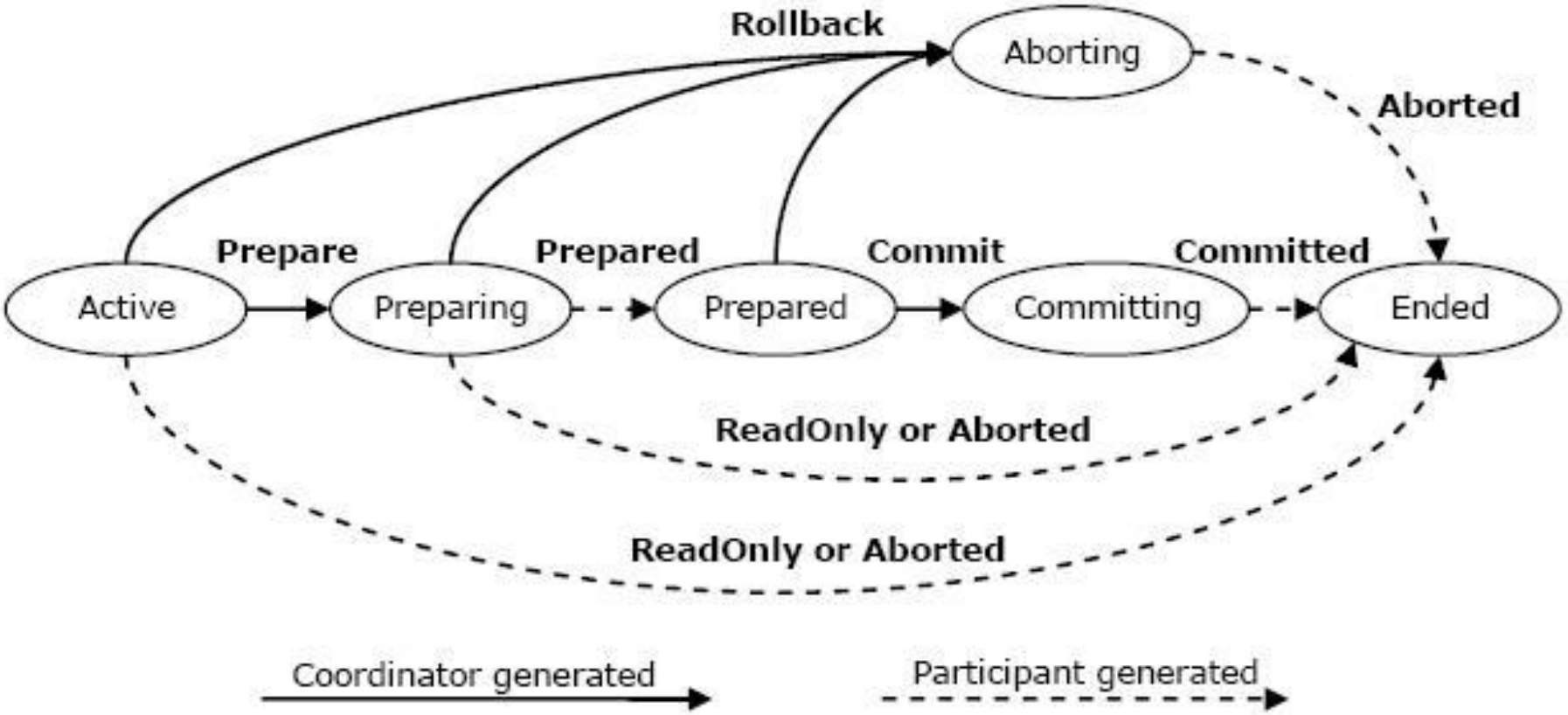


Protocolo Completition



- Estados y transiciones entre cliente y coordinador

Protocolo 2PC



- Estados y transiciones entre coordinador y participantes

Ejemplos SOAP

```
<env:Envelope xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
 <env:Header>
 <wscoor:CoordinationContext
      env:mustUnderstand='1'
      xmlns:wscoor='http://docs.oasis-open.org/ws-tx/wscoor/2006/06'
      xmlns:ns3='http://www.w3.org/2005/08/addressing'>
 <wscoor:Identifier>urn:-3f570b7e:d61:4ad386e0:73</wscoor:Identifier>
 <wscoor:CoordinationType>http://docs.oasis-open.org/ws-tx/wsat/2006/06
 </wscoor:CoordinationType>
 <wscoor:RegistrationService>
 <ns3:Address>http://vmxp.localdomain:8080/ws-c11/RegistrationService
 </ns3:Address>
 <ns3:ReferenceParameters>
 <wsarj:InstanceIdentifier
 xmlns:wsarj='http://schemas.arjuna.com/ws/2005/10/wsarj'>-
3f570b7e:d61:4ad386e0:73</wsarj:InstanceIdentifier>
 </ns3:ReferenceParameters>
 <ns3:Metadata />
 </wscoor:RegistrationService>
 </wscoor:CoordinationContext>
  </env:Header>
 <env:Body>
   <ns1:depositar xmlns:ns1="http://services.lins.fing.edu
      <idCuenta>1</idCuenta>
      <monto>5000.0</monto>
   </ns1:depositar>
 </env:Body>
</env:Envelope>
```

globalTxId

Dirección del Coordinador

Comentarios

Comentarios

- WS-AT no es muy deseable para integración B2B
  - Alto acoplamiento entre sistemas
  - Locks en BDs

- WS-AT más aplicable a EAI

WS-BusinessActivity

Transacciones de larga duración con Web Services

Transacciones

- Son un conjunto de tareas que cumplen las propiedades ACID
  - Atomicidad (Atomicity)
    - Se realizan todas las tareas o ninguna
  - Consistencia (Consistency)
    - Las tareas realizadas no violan ninguna de las restricciones de integridad
  - Aislamiento (Isolation)
    - Las tareas no pueden acceder o ver datos que se encuentren en estados intermedios.
  - Durabilidad (Durability)
    - Una vez realizada la transacción, se garantiza que esta persistirá a pesar de fallas en el sistema

Transacciones largas o Actividades de negocio

- Son un conjunto de tareas que cumplen las propiedades ACID
  - Atomicidad (Atomicity)
    - Se realizan todas las tareas o ninguna
  - Consistencia (Consistency)
    - Las tareas realizadas no violan ninguna de las restricciones de integridad
  - Aislamiento (Isolation)
    - Las tareas no pueden acceder o ver datos que se encuentren en estados intermedios.
  - Durabilidad (Durability)
    - Una vez realizada la transacción, se garantiza que esta persistirá a pesar de fallas en el sistema

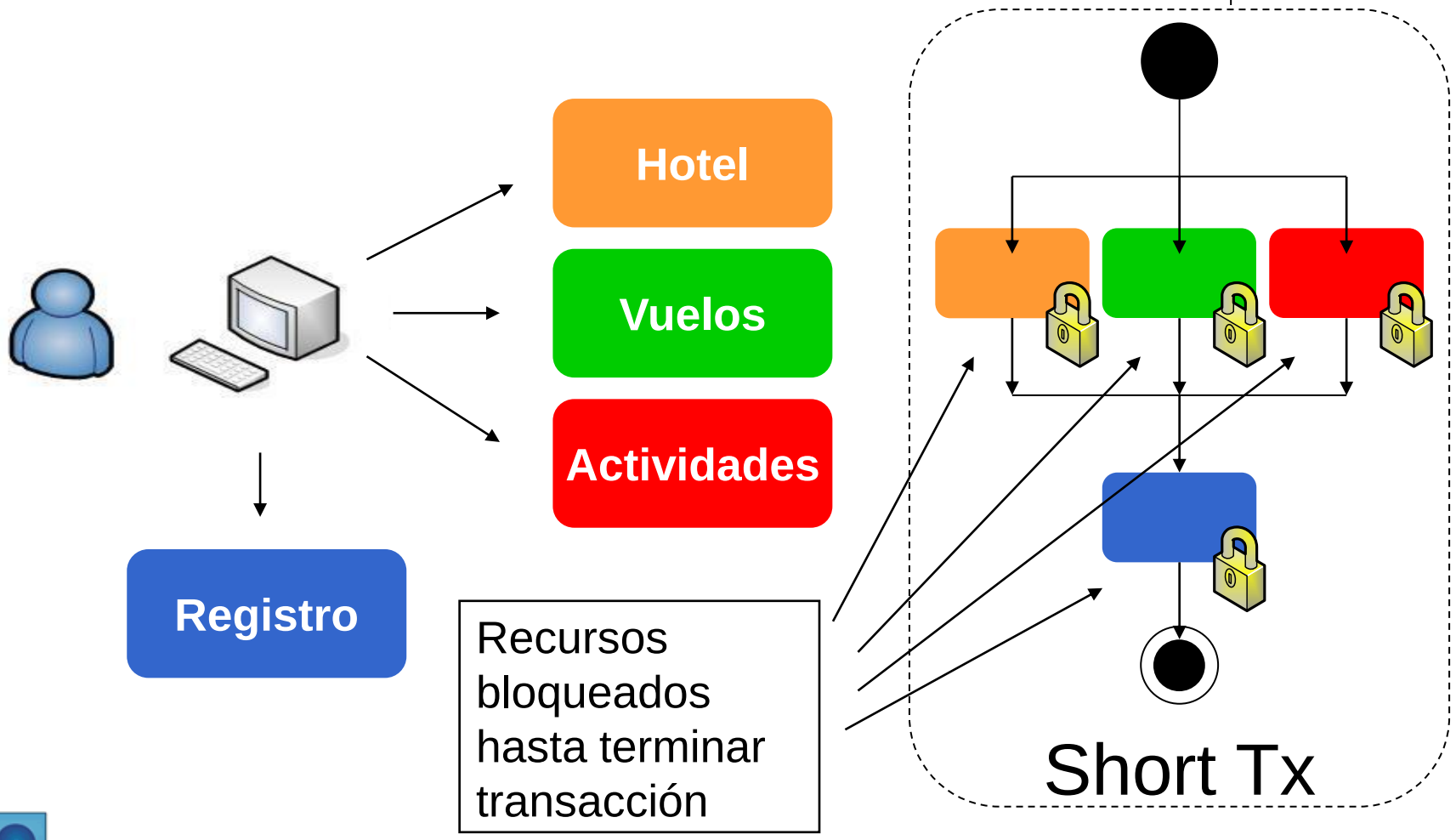
Actividades de negocio

- Pueden llegar a involucrar un gran número de transacciones atómicas independientes entre sí.
- Pueden llegar a consumir una gran cantidad de recursos y por un largo período de tiempo.
- Pueden llegar a demorar horas, días o incluso meses!
  - Aprobaciones, ensablado, manufactura o entregas pueden ser necesarias antes de dar una respuesta
- Las acciones intermedias son permanentes y pueden impactar por fuera del sistema.

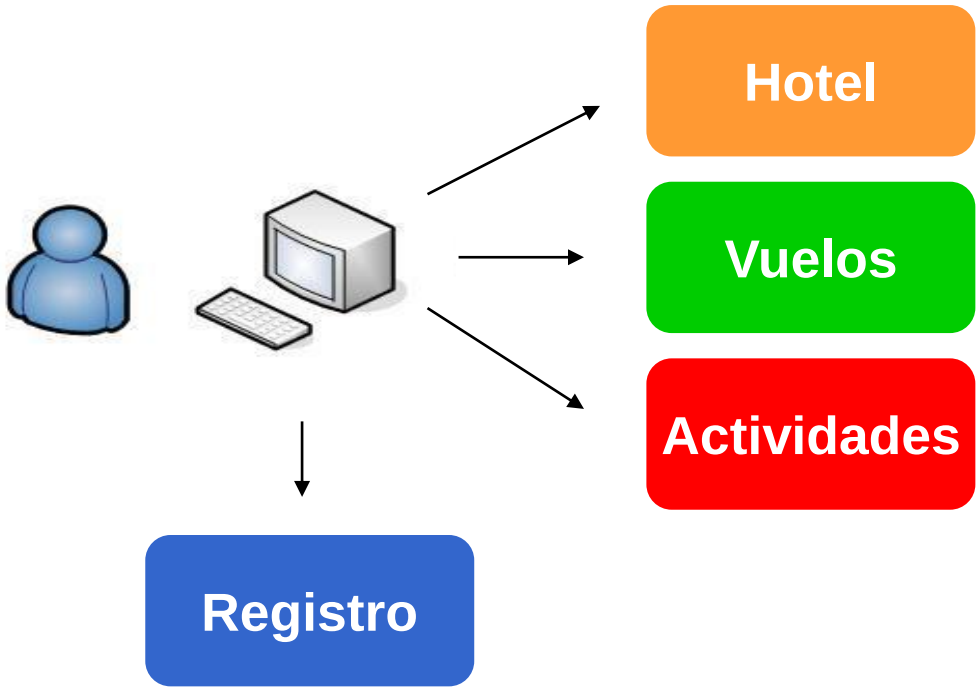
Actividades de negocio

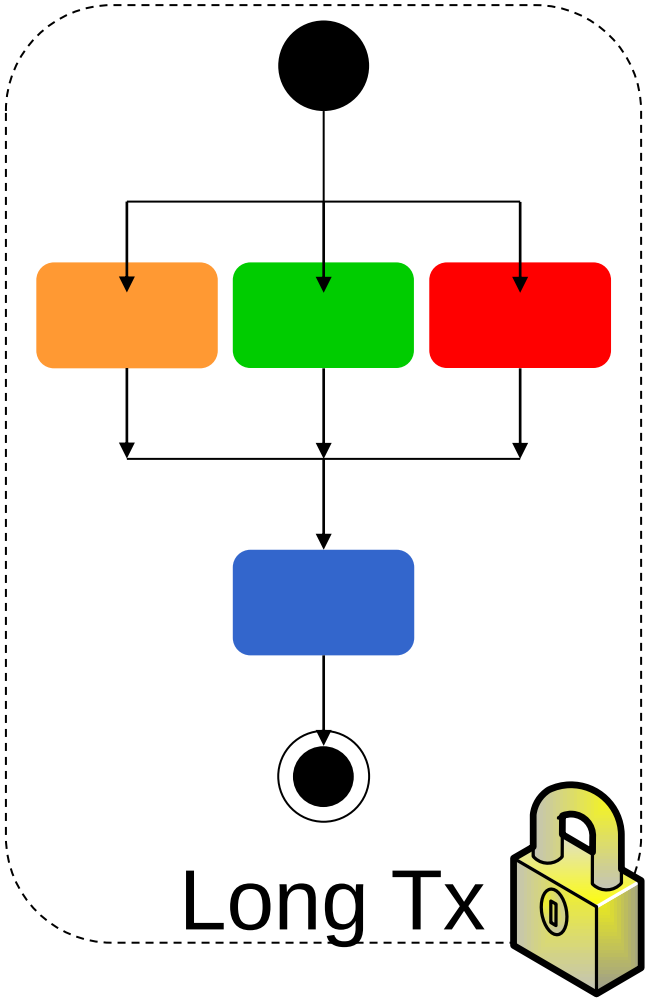
- No hay rollback como en las Transacciones tradicionales.
  - Abortar la transacción normalmente no es suficiente
- En caso de errores, se requiere de una actividad que deshaga las acciones tomadas.
  - El tratamiento de errores puede requerir cierta lógica de negocio
    - Tareas de compensación para revertir los efectos previamente confirmados por transacciones (atómicas) ya finalizadas.
- Sus participantes pueden pertenecer a diferentes dominios de confianza
  - Toda relación de confianza debe ser explicitamente definida.

Escenario


Hotel
Vuelos
Actividades
Registro
Recursos bloqueados hasta terminar transacción
Short Tx

Escenario





Escenario con transacciones de corta duración

- El usuario crea un paquete de viajes eligiendo un vuelo, un hotel y una serie de actividades.
  - El usuario puede realizar la selección en cualquier momento y tomarse el tiempo que desee.
  - Cuando finaliza, se registra el paquete en el sistema y se termina la transacción.

- Todas las tareas quedan enmarcadas en una única transacción corta, quedando todos los recursos bloqueados hasta que termine la transacción.
  - Un usuario bloquea a todos los demás!
  - Transacciones cortas no sirven debido a la naturaleza del problema.
  - Probemos otra cosa… Transacciones de larga duración!

Tx Largas+WS = WS-BA

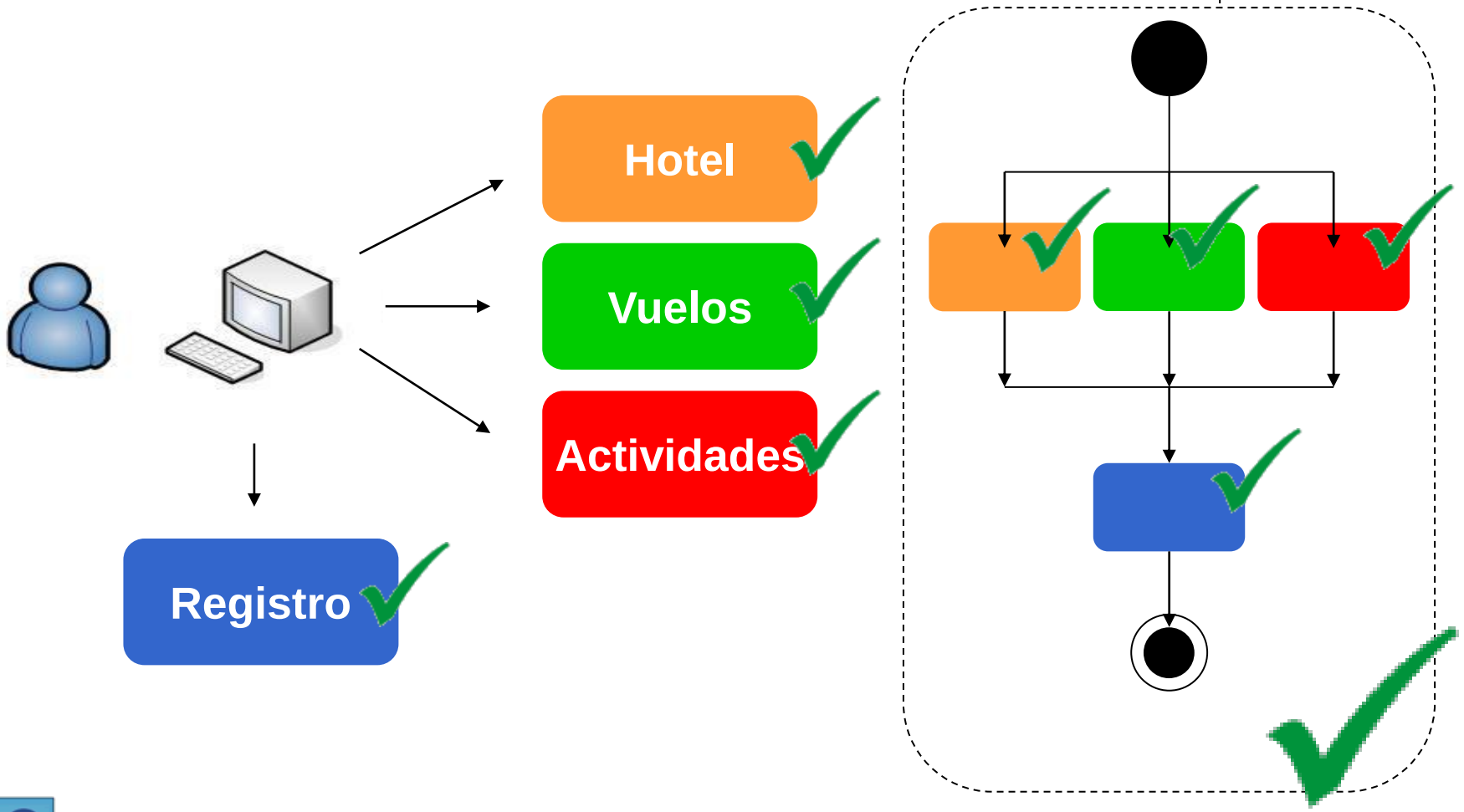
- WS-BA es un estándar de la OASIS con el propósito de definir mecanismos para la coordinación de actividades que involucren transacciones de larga duración y el tratamiento de errores en Web Services.

- Actualmente, en la versión 1.2

WS-BA: Contexto de ejecución

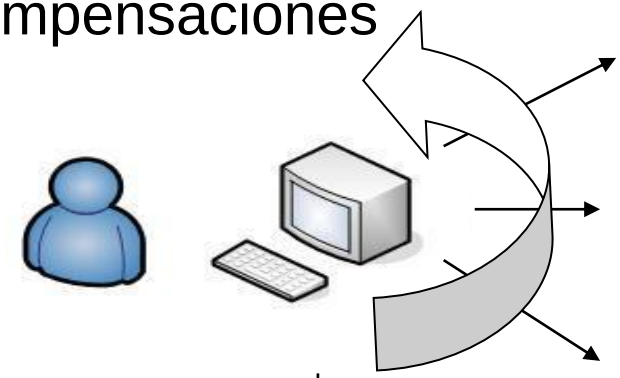
- Define dos contextos de ejecución
  - AtomicOutCome
    - Todos los participantes de la transacción llegan a un mismo y único resultado
    - Todos finalizan ok o todos compensan.
  - MixedOutCome
    - Permite finalizar una transacción donde los resultados de cada participantes sean independientes entre sí
      - Un participante: ok
      - Un participante: compensa
      - Transacción larga: ok
    - Opcional su implementación

Escenario: AtomicOutCome



Escenario: AtomicOutCome

Ejecución de compensaciones

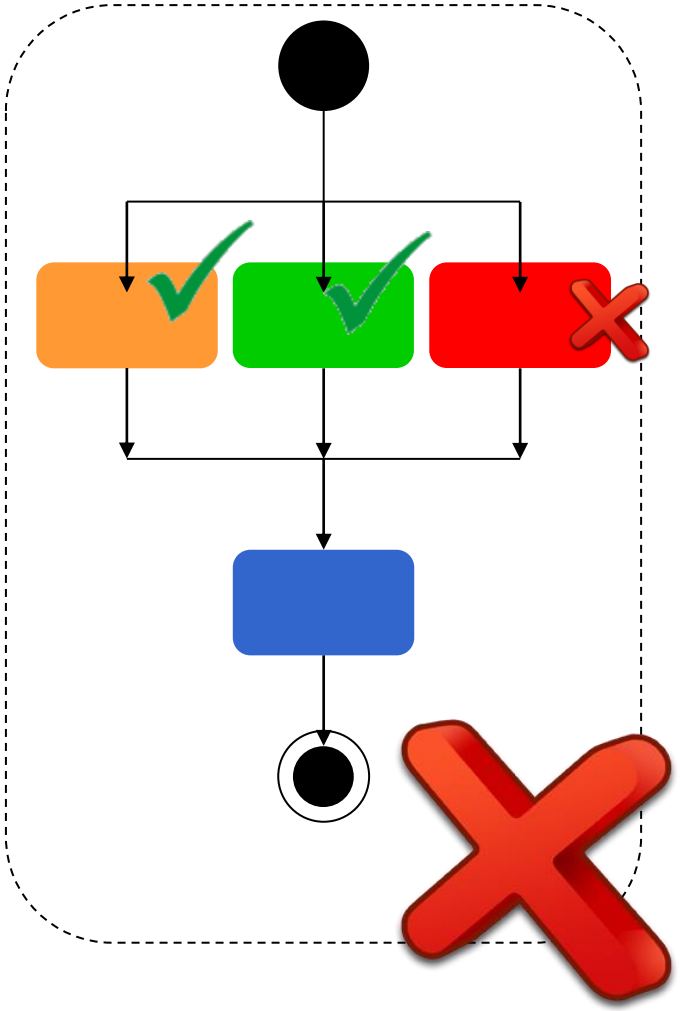


Registro

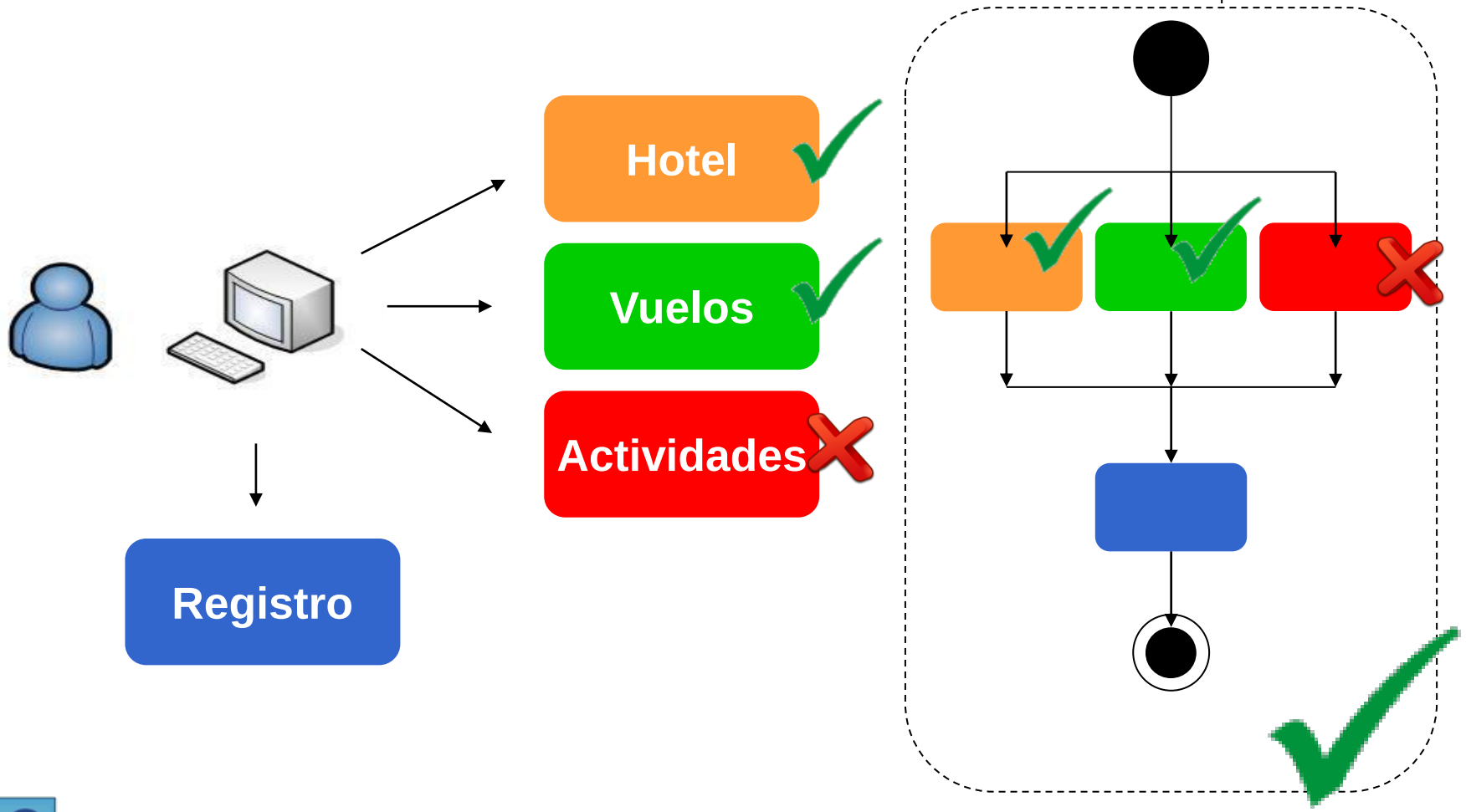
Hotel

Vuelos

Actividades



Escenario: MixedOutCome



Ejen

```
<env:Envelope xmlns:env='http://schemas.xmlsoap.org/soap/envelope/'>
 <env:Header>
  <wscoor:CoordinationContext env:mustUnderstand='1'
       xmlns:wscoor='http://schemas.xmlsoap.org/ws/2004/10/wscoor'>
   <wscoor:Identifier xmlns:wscoor='http://schemas.xmlsoap.org/ws/2004/10/wscoor'>
      urn:-3f57c379:570:4893ab85:cd
   </wscoor:Identifier>
   <wscoor:CoordinationType
       xmlns:wscoor='http://schemas.xmlsoap.org/ws/2004/10/wscoor'>
         http://schemas.xmlsoap.org/ws/2004/10/wsba/AtomicOutcome
   </wscoor:CoordinationType>
   <wscoor:RegistrationService
       xmlns:wscoor='http://schemas.xmlsoap.org/ws/2004/10/wscoor'>
       <wsa:Address
xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'>
              http://localhost:8080/xts/soap/RegistrationCoordinator
       </wsa:Address>
       <wsa:ReferenceParameters
              xmlns:wsa='http://schemas.xmlsoap.org/ws/2004/08/addressing'>
          <wsarj:InstanceIdentifier
                 xmlns:wsarj='http://schemas.arjuna.com/ws/2005/10/wsarj'>
                    -3f57c379:570:4893ab85:cd
          </wsarj:InstanceIdentifier>
       </wsa:ReferenceParameters>
   </wscoor:RegistrationService>
  </wscoor:CoordinationContext>
 </env:Header>
 <env:Body>
   <ns1:bookSeats xmlns:ns1='http://www.jboss.com/jbosstm/xts/demo/Re
      <how many>3</how many>
   </ns1:bookSeats>
 </env:Body>
</env:Envelope>
```

globalTxId

Contexto de ejecución

Dirección del Coordinador

Resumen

- WS-BA define mecanismos para la coordinación de actividades que involucren transacciones de larga duración y el tratamiento de errores.

- Define dos contextos de ejecución:
  - AtomicOutCome
  - MixedOutCome

Sin embargo…

- No tuvo suficiente apoyo de la industria
  - .NET nunca proveyó una implementación
- En la práctica se utiliza un WS-BA “casero”
  - Implementación adhoc sin estandarización
- Surgieron otros enfoques para mantener la consistencia en sistemas distribuidos
  - Consistencia eventual

Consistencia eventual

LINS

Laboratorio de Integracion de Sistemas

Consistencia eventual

- Un enfoque menos estricto que las transacciones ACID.

- Es posible continuar con el trabajo a pesar que el sistema no quede consistente inmediatamente.

Consistencia eventual

- En un futuro, el sistema eventualmente quedará consistente.
- Durante este tiempo, el resto de los sistemas podrá acceder a estos datos “inconsistentes”.
- Favorece la disponibilidad frente la consistencia.

Consistencia eventual

- Si no hay nuevas actualizaciones sobre un objeto, eventualmente todos los accesos retornarán el último valor actualizado.

- Si no hay una falla, la ventana máxima de inconsistencia estará determinada por factores como retrasos en la comunicación, carga del sistema o el número de réplicas involucradas.

Consistencia eventual

- Consecuencia importante:
  - A nivel global, el sistema no siempre se encuentra en un estado consistente.
- Existe una “ventana de inconsistencia”
  - Período en el cual el sistema no es consistente
  - Puede causar que un determinado servicio/sistema trabaje con datos obsoletos hasta que esa “ventana” se acabe
- El desarrollador debe ser consciente se está trabajando con consistencia eventual

Escenario 1:
Domain Name System (DNS)

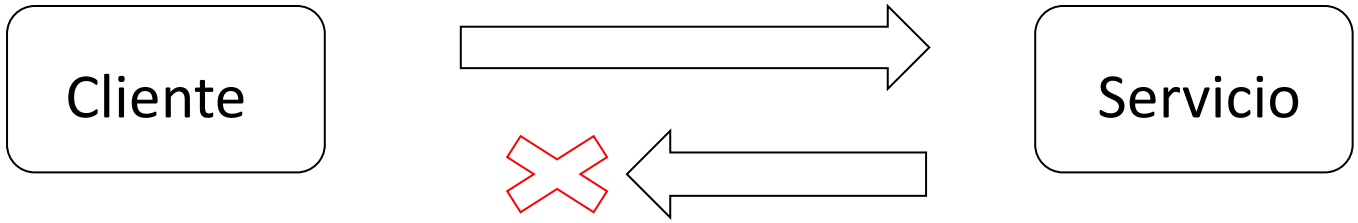
- Las actualizaciones de un nombre de host se distribuyen a lo largo de la red según un patrón de configuración y en combinación con políticas de caché

- Eventualmente todos los clientes van a acceder a los últimos datos actualizados del DNS.

Escenario 2: Pagos en línea

Laboratorio de Integracion de Sistemas

Fallos en la comunicación
p.ej: timeouts en la respuesta



Cliente no conoce resultado de la operación.
Debe cancelar solicitud

Servicio procesó correctamente el pago

Escenario 2: Pagos en línea

- Ante errores de comunicación (p.ej: timeouts) existen diferencias entre el resultado del cliente y del servicio.
  - Cliente: no se pudo procesar el pago
  - Servicio: el pago se procesó correctamente.
- El cliente debe comunicarse con el servicio para anular el pago realizado.
- Existe una ventana de tiempo donde el sistema global (cliente+servicio) es inconsistente.

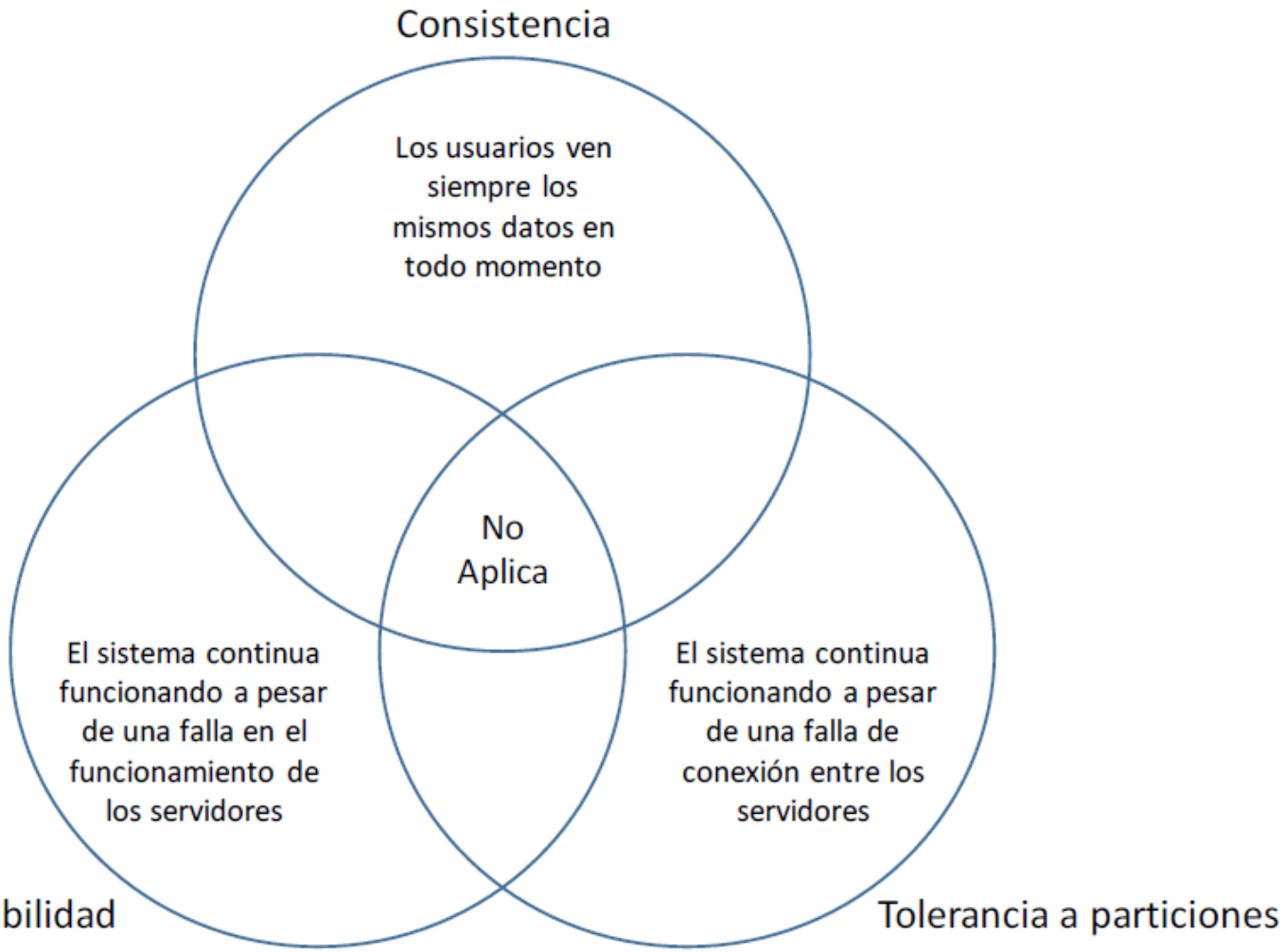
Teorema CAP

- En un sistema distribuido en el cual se comparten datos (una colección de nodos interconectados que comparten esos datos), solo se pueden obtener dos de las siguientes tres propiedades en un proceso de escritura/lectura de datos:
  - Consistencia
  - Disponibilidad
  - Tolerancia a particiones (fallos en comunicación)

Teorema CAP

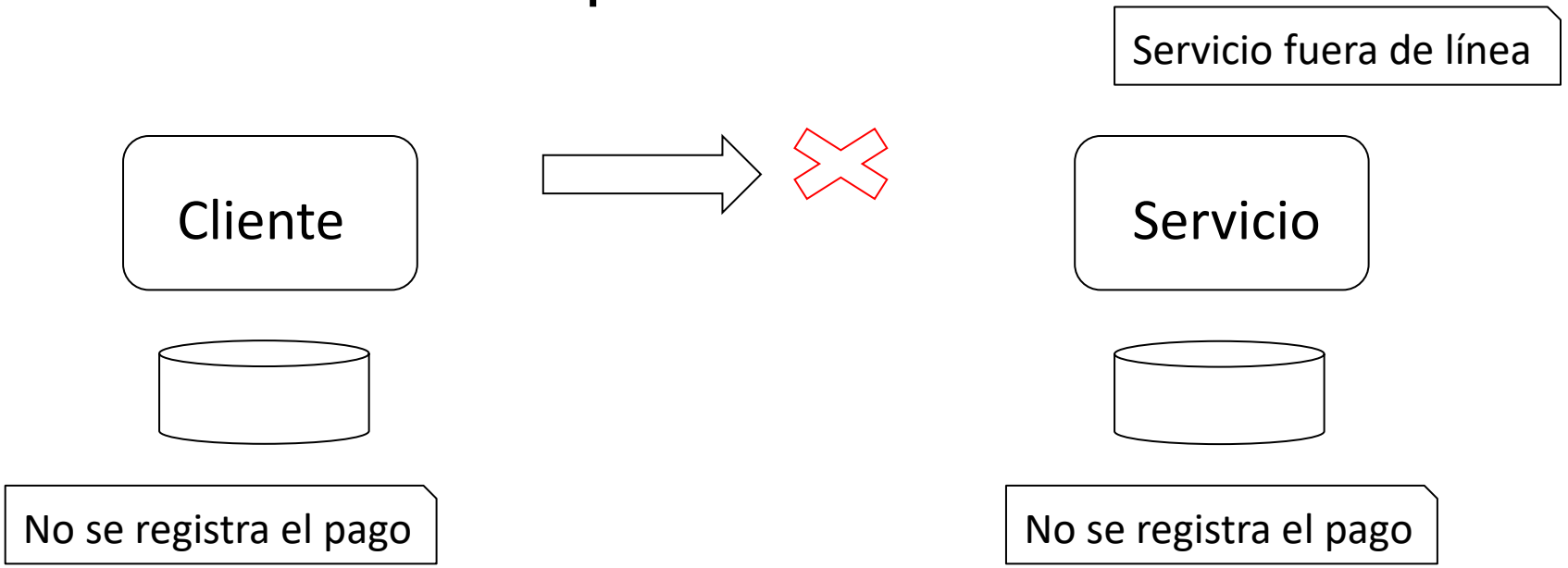
- Consistencia:
  - En todo nodo del sistema se puede observar la misma información
  - Se garantiza que una lectura de un dato en cualquier nodo, devolverá siempre el último valor escrito en el sistema global.
- Disponibilidad:
  - Cualquier solicitud a un nodo del sistema debe recibir una respuesta correcta (no error).
  - Cualquier nodo que no haya fallado, retornará una respuesta en un período de tiempo razonable
- Tolerancia a particiones (fallas en comunicaciones)
  - Es la capacidad de manejar las fallas en la comunicación entre los nodos y que dos o más nodos no se comuniquen entre sí.

Teorema CAP


Consistencia
Los usuarios ven siempre los mismos datos en todo momento
No Aplica
El sistema continua funcionando a pesar de una falla en el funcionamiento de los servidores
El sistema continua funcionando a pesar de una falla de conexión entre los servidores
Disponibilidad
Tolerancia a particiones

Escenario: Pagos en línea

- Consistencia + Tolerancia a particiones
- Sacrificamos disponibilidad



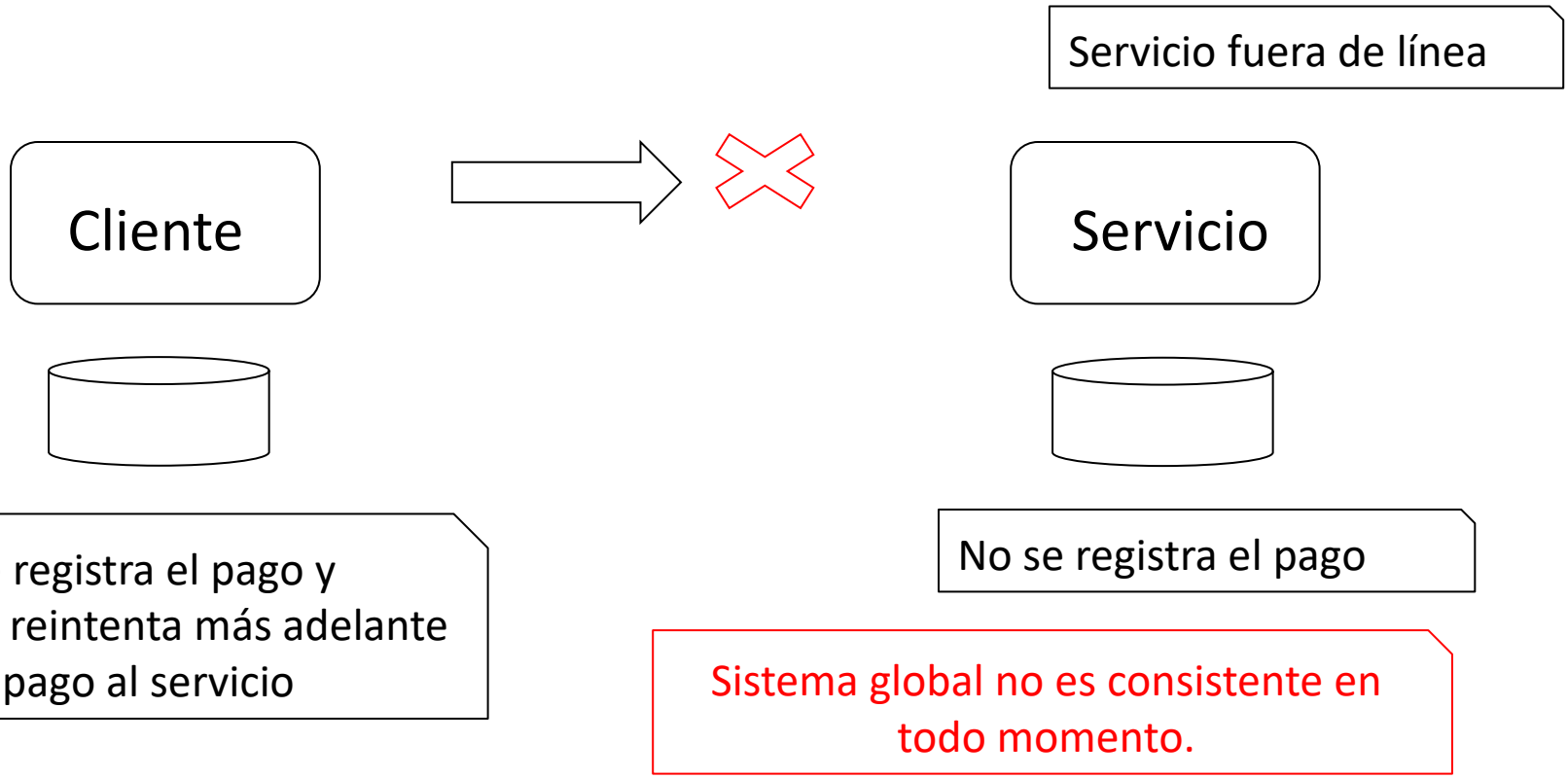
Sistema global no disponible: no es posible registra el pago

Escenario: Pagos en línea

- Consistencia + Tolerancia a particiones:
  - El servicio no está disponible
  - El cliente no puede comunicarse con el servicio para registrar el pago
  - Se prioriza la consistencia de la solución como un todo. En ambos sistemas no existe el pago.
  - Es preferible que el cliente retorne un error a dejar inconsistente el sistema
  - El cliente necesita la confirmación del servicio para responder correctamente.

Escenario: Pagos en línea

- Tolerancia a particiones + Disponibilidad
- Sacrificamos consistencia



Escenario: Pagos en línea

- Tolerancia a fallos y disponibilidad:
  - El servicio no está disponible
  - El cliente no puede comunicarse con el servicio para registrar el pago
  - Se prioriza la disponibilidad del sistema global.
  - El cliente no necesita la confirmación del servicio para responder correctamente.
  - En el cliente, el pago fue realizado. En el servicio, no.
  - Es preferible guardar el pago y reenviarlo más adelante a que el cliente lance un error.
  - Existe una ventana de tiempo donde el resultado global es inconsistente
  - Luego que el cliente se pueda comunicar con el servicio, el sistema volverá a ser consistente.

Escenario: Pagos en línea

- Consistencia + Disponibilidad
- Sacrificamos tolerancia a particiones
  - Entonces: no es posible que dos o más partes no se comuniquen entre sí.

Escenario: Pagos en línea

- Consistencia + Disponibilidad
- Sacrificamos tolerancia a particiones
  - Entonces: no es posible que dos o más partes no se comuniquen entre sí.
- Este escenario es imposible en un sistema distribuido.
  - Por su naturaleza, las comunicaciones fallan.
- Si tenemos este escenario, no hay un sistema distribuido.
  - Se utilizan las técnicas tradicionales para mantener la consistencia.

Resumen

- WS-Addressing
  - Asincronismo entre Web Services
- MTOM
  - Intercambio eficiente de datos no XML

Resumen

- WS-Atomic Transaction y WS-Business Activity
  - Dos propuestas para manejo de la consitencia que no tuvieron éxito

- Consistencia eventual como solución a problemas de consistencia entre Web Services

Fin

LINS

Laboratorio de Integracion de Sistemas