estudios de postgrado en computación

Bases de datos avanzadas

Universidad de Los Andes

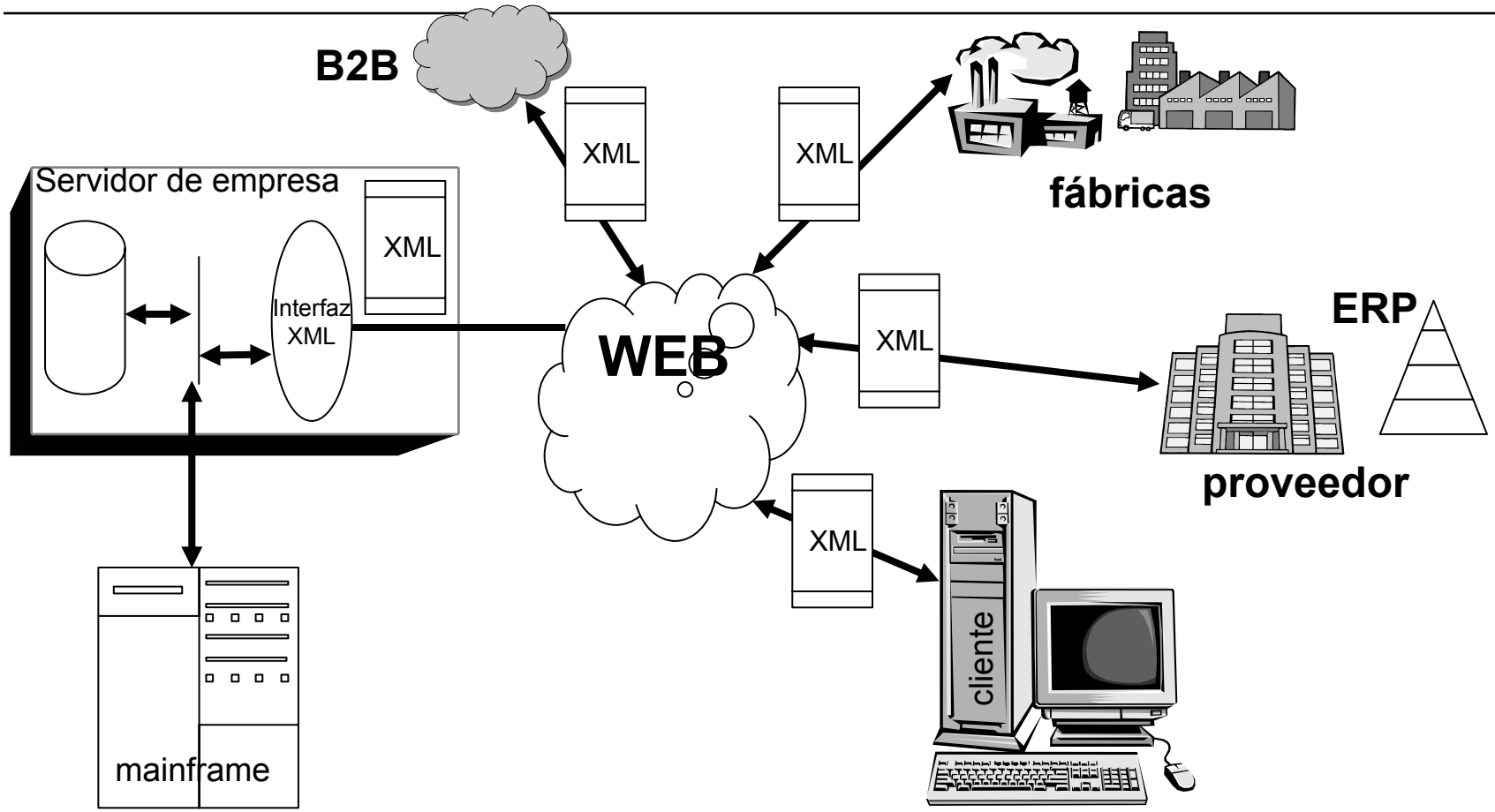
Postgrado en Computación

Prof. Isabel M. Besembel Carrera

Unidad II. Sesión 15. Servicios Web .

UNIVERSIDAD DE LOS ANDES

Introducción



UNIVERSIDAD DE LOS ANDES

Arquitectura de los servicios web

- Servicio web: módulo aplicación web que ofrece una interfaz programable accesible por XML
- Descripción del servicio: definición de las interfaces de servicio en términos de operaciones y parámetros en WSDL (Web Service Definition Language)
- Protocolo SOAP (Single Object Application Protocol): introducido en 1999 por Don Box de DevelopMentor, Dave Winer de Userland y un equipo de Microsoft COM.
  - ✓ Protocolo de invocación de servicios y respuesta en XML usado arriba de http
  - ✓ Componentes:
    - Sobre contentivo del mensaje y cómo debe ser tratado
    - Especificación de enlace a un protocolo transporte para intercambio de mensajes
    - Conjunto de reglas de codificación para representar los datos de la aplicación
    - Representación de las llamadas y respuestas

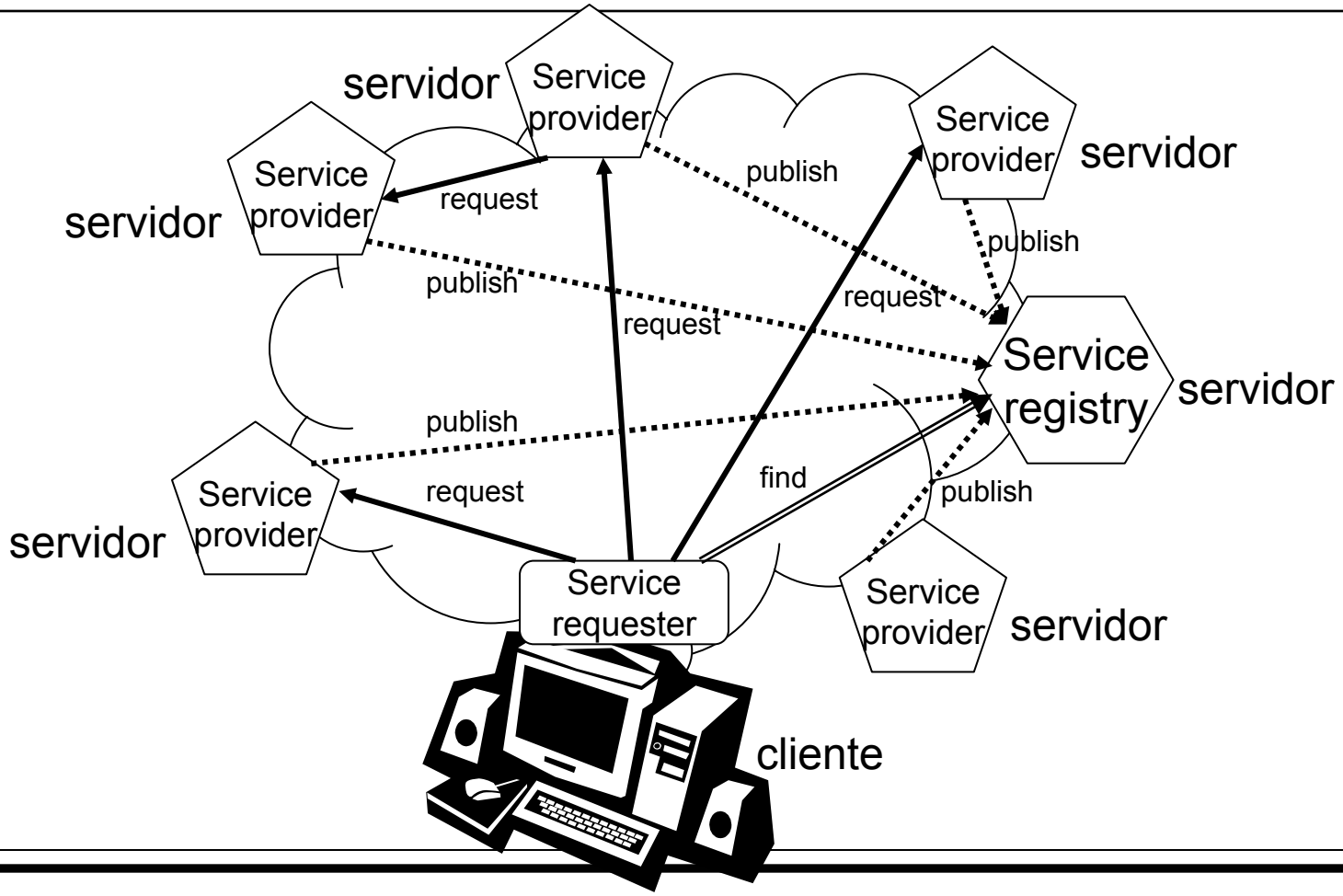
UNIVERSIDAD DE LOS ANDES

Componentes de la arquitectura

- Basado en la arquitectura CORBA para objetos distribuidos, pero usa XML para la descripción e invocación de los servicios
- Proveedor de servicio (service provider): aplicación que se ejecuta en el servidor que contiene un módulo accesible por el cliente web vía el protocolo SOAP
- Registro de servicio (service registry): catálogo de los proveedores de servicio gestionado por un servidor a nivel aplicación
- Solicitador de servicio (service requester): aplicación cliente ligada a un servicio que invoca las funciones por mensajes XML intercambiados vía protocolo SOAP
- Arquitectura recursiva: una aplicación solicitadora de servicio puede a su vez ser una aplicación proveedora de servicio

UNIVERSIDAD DE LOS ANDES

Arquitectura de servicios web



UNIVERSIDAD DE LOS ANDES

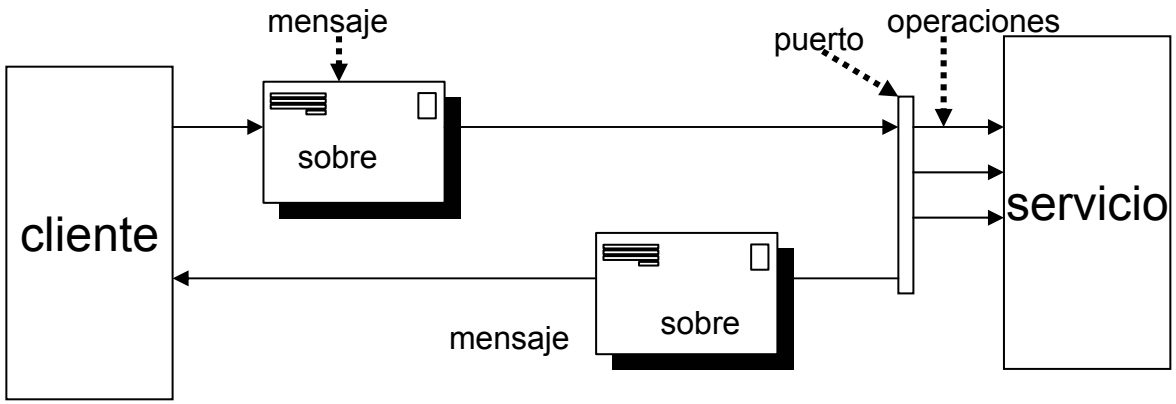
Servicios con WSDL …

- WSDL: lenguaje con sintaxis XML para describir los servicios web
- Cliente: el compilador WSDL genera el código de invocación de servicio
- Servidor: genera el código de interfaz entre el protocolo y el servicio
- Modelo de intercambio inspirado en la navegación marítima: mensajes contenidos en sobres intercambiados de puerto en puerto
- Puerto WSDL: nombre que identifica un grupo de operaciones que permiten el acceso al servicio web vía un protocolo específico (SOAP, HTTP, SMTP, etc.)
- Enlace WSDL: especificación del tipo de acceso, del protocolo de transporte y de la codificación de los parámetros en el mensaje para un tipo de puerto dado

UNIVERSIDAD DE LOS ANDES

Servicios con WSDL

- Mensaje WSDL: grupo de datos codificados en XML que permiten la invocación de una operación asignada a un puerto cliente o de responder (servidor)
- Elementos:
  - Type: define el conjunto de tipos de los parámetros, estándar XML
  - Message: describe los parámetros de entrada y salida por cada operación
  - Port: define un grupo de operaciones bajo un puerto
  - Binding: especifica un enlace entre el tipo de puerto y el protocolo de acceso
  - Service: especifica la colección de puertos y sus URLs de acceso



UNIVERSIDAD DE LOS ANDES

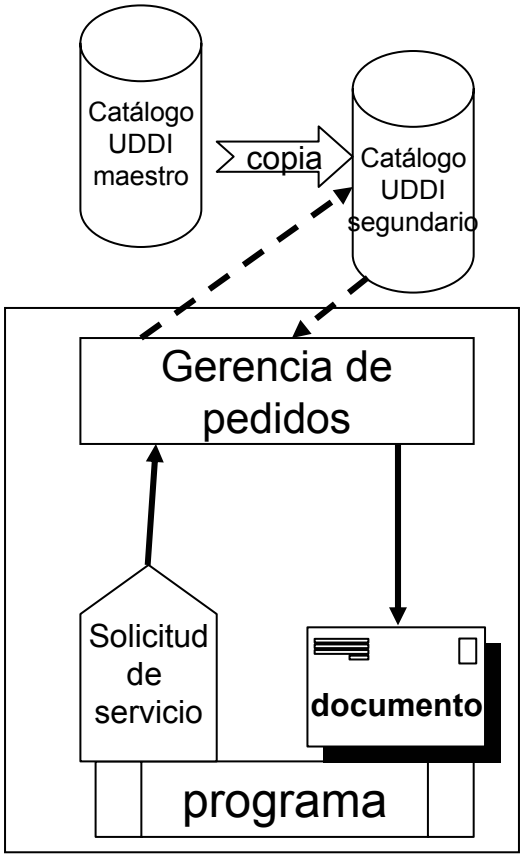
Ejemplo WSDL

```
<definitions name="Hotel" targetNamespace="http://ejemplo.com/hotel.wsdl"
 xmlns:wns="http://ejemplo.com/hotel.wsdl" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">
    <types>
        <!--    definición de los tipos de datos    -->
    </types>
    <message>
        <!--    declaración de los mensajes entrada y salida por separado    -->
    </message>
    <portType>
        <!--    declaración de las operaciones  asociación con los mensajes    -->
    </portType>
    <binding>
        <!--    definición del enlace wsdl – soap acciones y codificaciones    -->
    </binding>
    <service name="Hotel">
        <!-- declaración de los puertos (grupo de operaciones y protocolos de acceso)    -->
    </service>
</definitions>
```

UNIVERSIDAD DE LOS ANDES

Catálogo de servicios UDDI

- Catálogo UDDI (Universal Description, Discovery, and Integration Registry): catálogo de servicio accesible en web que describe las empresas y sus servicios, tanto a nivel funcional como técnico, que permite a los usuarios descubrir los servicios disponibles
- Funciones:
  - Registrar su empresa dando su identidad, descripción, lista de categorías y tipos de servicios ofertados
  - Registro de los servicios que ofrece la empresa con sus nombres. Descripción y categorías
  - Registro de las operaciones para cada uno de los servicios definidos con sus datos técnicos correspondientes



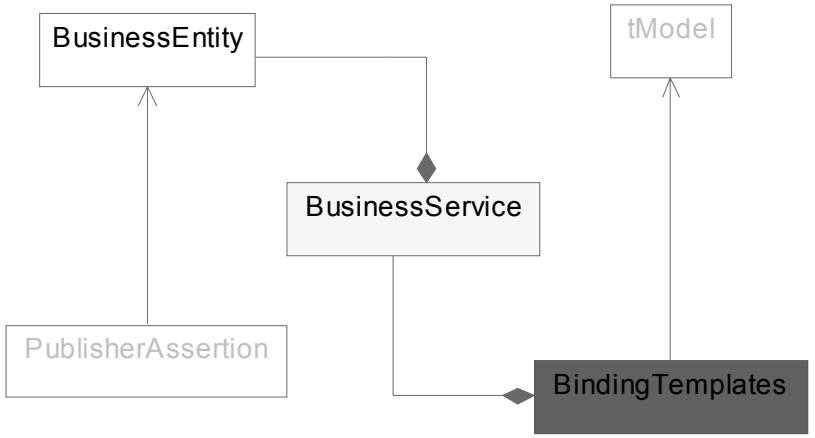
UNIVERSIDAD DE LOS ANDES

Contenido del catálogo

- Páginas blancas
  - ✓ BusinessKey, AuthorizedName, Operator, Name, Description, CategoryBag, BusinessServices
- Páginas amarillas
  - ✓ ServiceKey, BusinessKey, Name, Description, CategoryBag, BindingTemplates
- Páginas verdes
  - ✓ BindingKey, ServiceKey, Description, AccessPoint, TModelInstances

El descubrimiento de los servicios se hace con el comando **find** emitido por la gerencia de pedidos

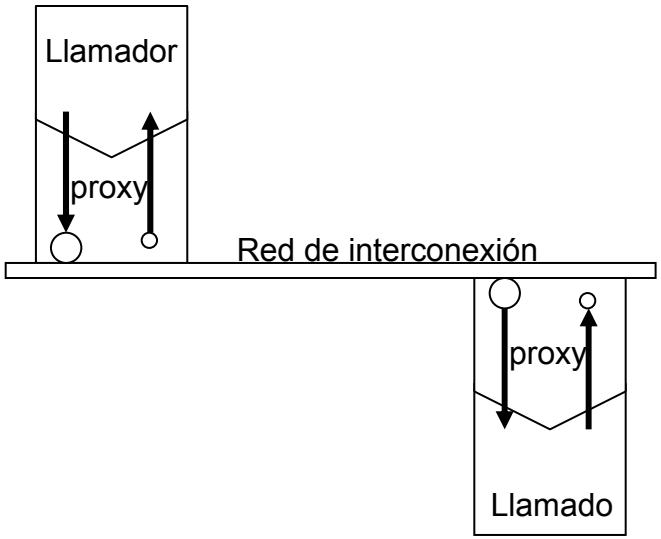
www.uddi.org



UNIVERSIDAD DE LOS ANDES

RPC

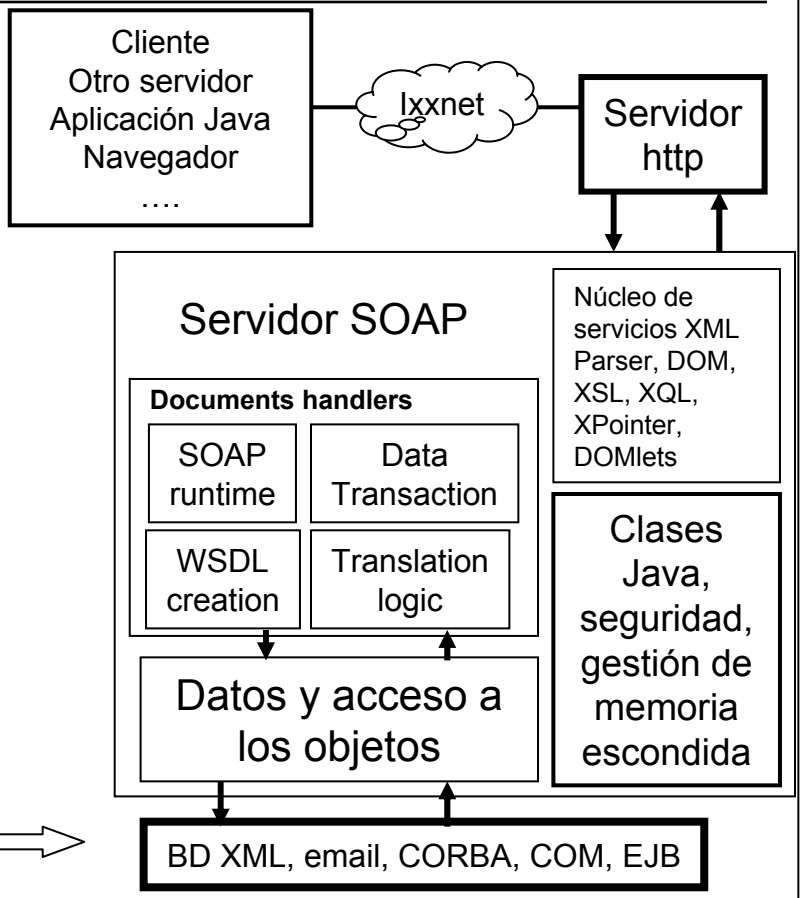
- Técnica introducida en la arquitectura distribuida DCE en 1980
- Ofrece las funciones de llamador y de llamado en el site del llamador
- El origen de procuración (proxy) transforma la invocación al procedimiento en el envío de un mensaje desde el sitio llamador al sitio llamado, donde lo recibe un transformador simétrico que hace la invocación al llamado



UNIVERSIDAD DE LOS ANDES

XML - RPC

- David Winer, Userland, 1998 introdujo dos tipos de documentos <methodCall> y <methodResponse> sobre http en formato MIME text/xml
- Extendido y estandarizado en 2001 por W3C como la versión 2.1 de SOAP
- Objetivo:
  - ✓ Permitir la invocación sobre los puertos de conexión de servicios web que ofrecen operaciones dándoles la ilusión de ser locales
- Arquitectura de un servidor SOAP ⇨



UNIVERSIDAD DE LOS ANDES

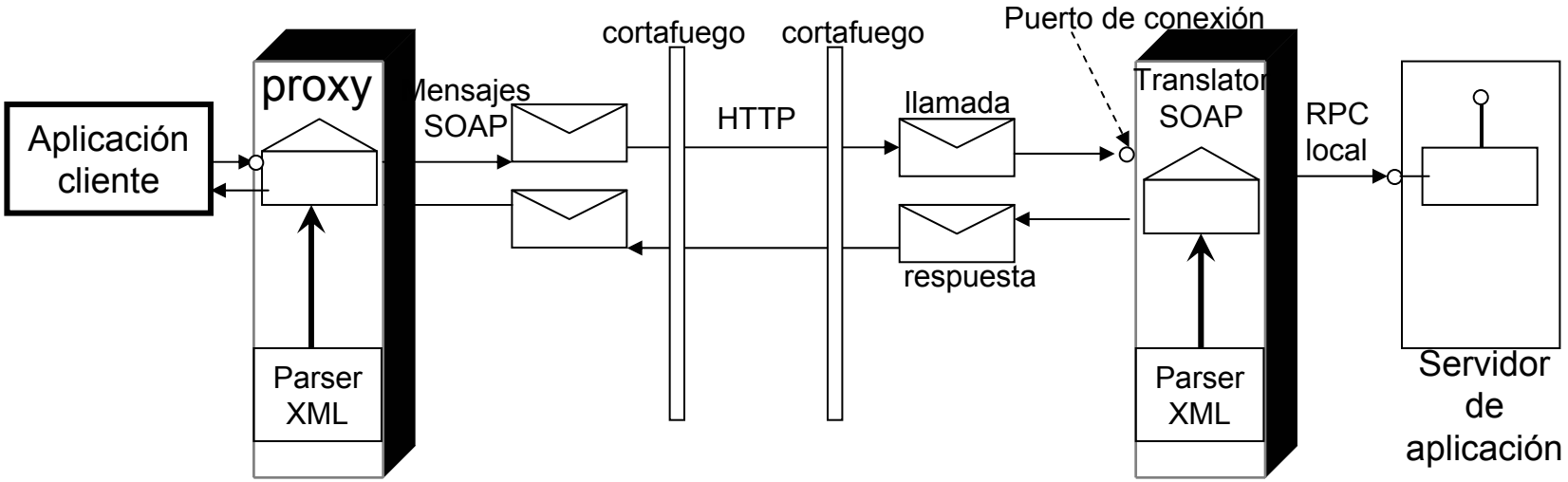
SOAP…

- Codificación universal de llamadas y respuestas de procedimientos de tipo RPC
- Protocolo independiente de plataforma y de lenguaje de programación
- Partes:
  - ✓ Construcción del sobre “envelope”: define el contenido del mensaje, quien debe tratarlo y si es opcional u obligatorio
  - ✓ Enlace: define cómo intercambiar los sobres entre los participantes usando un protocolo de transporte (http, smtp, etc.)
  - ✓ Reglas de codificación: definen el mecanismo de serialización que puede ser utilizado para intercambiar instancias de los tipos de datos de la aplicación
  - ✓ Representación RPC: define una convención para representar el llamdo a procedimientos distantes en SOAP

UNIVERSIDAD DE LOS ANDES

SOAP…

- Mensaje SOAP: documento XML cuya raíz es <envelope> que contiene un elemento <Heading> y uno <Body> que permite invocar una operación distante o transmitir la respuesta de la operación
- Sobre SOAP: raíz del documento XML que contiene el mensaje SOAP definiendo el espacio de nombres, el estilo de codificación, etc.



UNIVERSIDAD DE LOS ANDES

SOAP…

- Sobre de mensajes: elementos que definen la estructura de los mensajes SOAP que deben estar asociados al espacio de nombres www.w3.org/2001/09/soap-envelope
- Atributo encodingStyle en www.w3.org/2001/06/soap-encoding
- Ejemplo:

```
<env:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
        <env:Header> …..        </env:Header>
        <env:Body> ….           </env:Body>
</env:Envelope>

<env:Header>
        <t:Transaction xmlns:t="http://ejemplo.org/tx" env:mustUnderstand="1">
                <t:trid>7</t:trid>
                <t:action>commit</t:action>
        </t:Transaction>
</env:Header>
```

Solicitud de paso de la transacción 7 Tomar en cuenta obligatoriamente

UNIVERSIDAD DE LOS ANDES

SOAP…

- Cuerpo del mensaje: puede contener 0 o más elementos hijos llamados bloques
- Bloque de manejo de errores:
  - ✓ Obligatorio
    - o <faultcode> valores (VersionMismatch, Client, Server, MustUnderstand, DataEncodingUnknown)
    - o <faultstring> documentación del error
  - ✓ Opcionales
    - o <faultfactor> precisa el autor del error
    - o <detail> transmisión de un código de error de la aplicación

```
<env:Body>
        <hotel:CualHotel xmlns:hotel="http://ejemplo.com/hotel">
                <CodeAction>56</CodeAction>
        </hotel:CualHotel>
        <env:Fault>
                <faultcode>env:Client</faultcode>
                <faultstring>Parámetro incorrecto</faultstring>
                <detail="233" />
        </env:Fault>
</env:Body>
```

UNIVERSIDAD DE LOS ANDES

SOAP

- Reglas de serialización de los datos en los mensajes están definidas en www.w3.org/2001/09/soap-encoding que debe ser utilizada como valor del atributo encodingStyle
- Llamada a un procedimiento de RPC:
  - URI del nodo destino
  - Nombre del procedimiento
  - Firma del procedimiento
  - Parámetros del procedimiento
- SOAP integra un estándar extensible para RPC
- Más general

UNIVERSIDAD DE LOS ANDES

Ejemplo SOAP

```
POST /hotel HTTP/1.1
Host: http://www.ejemplo.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: http://www.ejemplo.com/Hotel#Reservar
<env:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
    <env:Body>
        <h:llamadaRes xmlns:h="www.ejemplo.com/Hotel">
            <idHotel>212</idHotel>
            <fechaRes>2003-12-19</fechaRes>
            <duracion>7</duracion>
        </h:llamadaRes>
    </env:Body>
</env:Envelope>
```

Ejemplo de reservación de un hotel

```
HTTP/1.1 256 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
<env:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
    <env:Body>
        <h:respuestaRes xmlns:h="www.ejemplo.com/Hotel">
            <reservacion>1</reservacion>
        </h:respuestaRes>
    </env:Body>
</env:Envelope>
```

Ejemplo de respuesta a la reservación de un hotel

Seguridad de los servicios web

- Transacciones:
  - ✓ Atómicas: conjunto de operaciones que se realizan todas o no se realiza ninguna
    - Necesitan el soporte del protocolo de validación en dos fases
  - ✓ Largas: compuestas de transacciones atómicas generalmente anidadas
    - Sagas: secuencia de transacciones atómicas con compensación
    - Transacciones anidadas o cohortes

Ejemplo: Procesos de Negocios ____ lenguaje de orquestación BPEL4WS

Lenguajes para modelar los procesos de negocios bajo la forma de flujo de actividades

- …de compra que conllevan el acuse recibo de la orden, envío de cotización varios días después y luego la planilla de envío de pedido con su correspondiente factura, lo cual implica varios días
- Sistema de reservaciones con solicitud de reservación, acuse de recepción, realización de la reservación, pago de la reservación

UNIVERSIDAD DE LOS ANDES

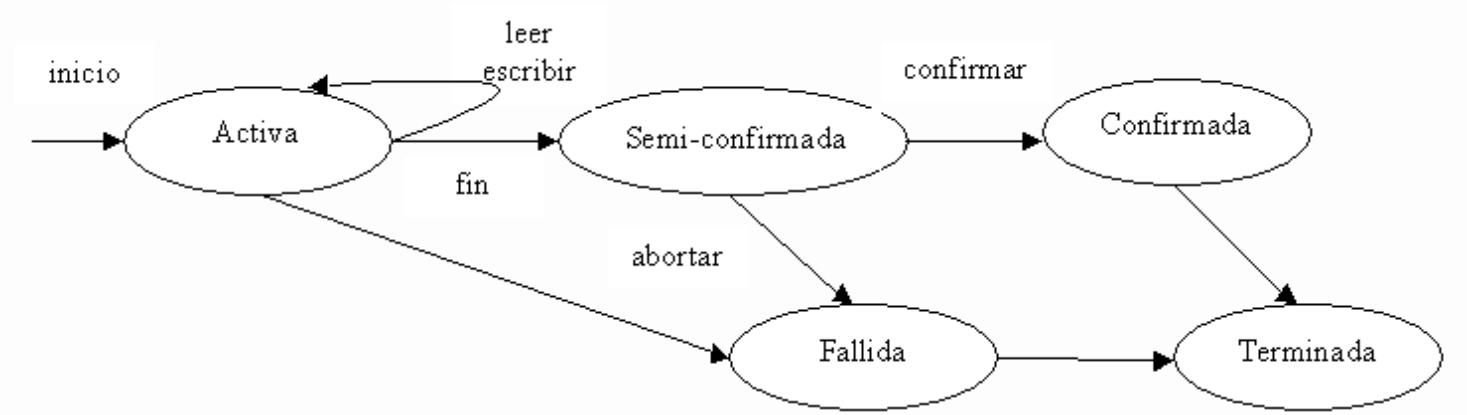
Gestión de transacciones…

- **Transacción:** unidad atómica de trabajo que se realiza por completo o bien no se efectúa en absoluto
- ***Propiedades de las transacciones (ACID)***
  - ✓ **Atomicidad**: La **T** debe realizarse completamente o no se realiza en absoluto.
  - ✓ **Consistencia**: Debe tomar una BD consistente y debe devolver una BD consistente. Una BD consistente es una BD cuyos valores almacenados cumplen con todas las restricciones de integridad definidas en el esquema y aquellas definidas por los programas.
  - ✓ **Aislamiento**: Cada **T** debe evitar que sus actualizaciones sean vistas por otras **T** mientras ella no haya sido confirmada.
  - ✓ **Durabilidad**: Las operaciones realizadas por una **T** confirmada no deben perderse

UNIVERSIDAD DE LOS ANDES

Gestión de transacciones…

- ***Estados de las transacciones (Ti) y sus operaciones***
  - ***Inicio de la transacción: T.inicio()***
  - ***Lectura de un elemento: T.leer(x)***
  - ***Escritura de un elemento: T.escribir(x)***
  - ***Fin de la transacción: T.fin()***
  - ***Confirmar la transacción: T.confirmar()***
  - ***Abortar la transacción: T.abortar()***



UNIVERSIDAD DE LOS ANDES

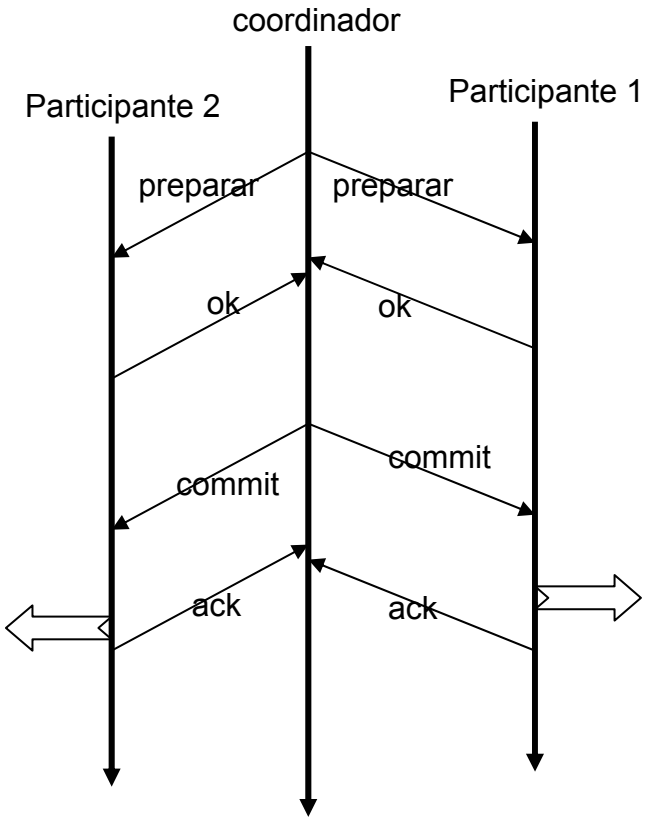
Gestión de transacciones…

- Protocolo de validación dos fases: propuesto en 1976 para Ti dos fases
- **Transacciones dos fases: Es una T que no hace bloquear luego de hacer un desbloquear. Una T dos fases debe hacer todos los bloqueos a sus granulas antes de comenzar a desbloquear.**
  - **Reglas de uso de bloquear/desbloquear: Deben ser seguidas por cualquier T en un sistema de bases de datos (SBD):**
    - **Toda T debe bloquear con los modos de operación correctos sobre g antes de ejecutar alguna operación sobre ella.**
    - **Toda T debe hacer desbloquear sobre g luego de la ejecución de la operación.**
    - **Cualquier T no puede hacer bloquear luego de haber hecho desbloquear**
- **Teorema de bloqueo en dos fases:** Si todas las transacciones de un plan de ejecución son dos fases, entonces el plan es serial.
  - **Prueba: Eswaram et al. *The notions of consistency and predicate locks in a database system.* Communications of the ACM. 19(11) 1976. Ullman. *Principles of database systems.* Computer Science Press. 1982.**

UNIVERSIDAD DE LOS ANDES

Protocolo de validación dos fases

- En un sistema distribuido el control es centralizado por un sitio llamado coordinador y los demás son los participantes
- Estandarizado por ISO y X/Open Group bajo el nombre de XA, que es un protocolo que solicita a cada administrador de recursos su votación antes de aceptar la validación. El **commit** se efectúa solamente si todos los participantes votan OK
- No es suficiente para Ti complejas distribuidas



UNIVERSIDAD DE LOS ANDES

Transacciones distribuidas complejas

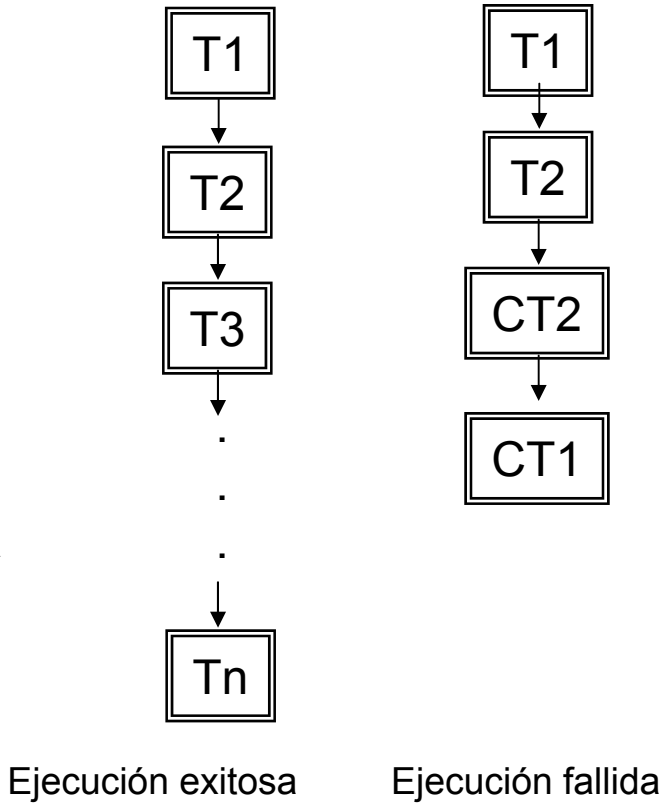
- ***Transacciones anidadas*:** Las Ti forman una jerarquía donde aquellas que tienen la misma madre deben ser serializables. Una Ti pasa solamente cuando su madre pasa y por ello, las modificaciones se hacen en la BD cuando pasa la raíz.
- **Modelo Cohorte:** transacciones anidadas cooperantes, donde unas pueden validarse en dos fases y otras por una, y donde la coherencia del conjunto está asegurada por la transacción raíz
  - ✓ Árbol de transacciones cooperativas disparadas por la transacción madre
- **Modelo de sagas:** transacciones anidadas donde las subtransacciones tienen una compensación asociada que permite deshacer su efecto anulando su propiedad de aislamiento (ACID) de sus actualizaciones
  - ✓ Propuesto por García-Molina en 1991
  - ✓ Objetivo: gestionar correctamente las transacciones largas sin aislar los datos modificados por las subtransacciones participante

UNIVERSIDAD DE LOS ANDES

Sagas

- **Saga:** secuencia de Ti ACID {T1, T2, …, Tn} que se deben ejecutar en ese orden asociadas a una secuencias de compensaciones {CT1, CT2, …, CTn}, donde el abandono de una Ti implica el abandono de la saga completa y la ejecución de la secuencia de compensaciones en orden inverso

- Es una Ti larga compuesta de subTi
- Mediante sagas se puede relajar el principio de aislamiento de las Ti ACID
- Cada subTi libera sus bloqueos al terminar
- Otra saga puede ver los resultados parciales, la anulación de una saga puede provocar la anulación de la otra
- Es el modelo que se adapta mejor al comercio electrónico



Ejecución exitosa Ejecución fallida

UNIVERSIDAD DE LOS ANDES

Gestión de Ti-XAML

- Integración de la gestión de Ti simples a los servicios web: integración de XA y SOAP
- Servicio web visto como un manejador de recursos (RM) para XA
- XAML (XA Markup Language): Objetivos
  - ✓ Ofrecer una especificación de interfaz estándar para el modelo de interacción y coordinación de Ti en XML
  - ✓ Ser compatible con XA, los monitores transaccionales y los estándares industriales
  - ✓ Especificar las interfaces y los protocolos que pueden ser anexados a los servicios web, en específico SOAP
  - ✓ Especificar los modelos de interacción que integran las Ti y los modelos más generales de servicios web
- Grupo de trabajo XAML no logró dar una especificación coherente

UNIVERSIDAD DE LOS ANDES

Gestión de Ti-BTP

- BTP (Business Transaction Protocol): de Oasis para la gestión de Ti distribuidas. Dialecto XML.
  - ✓ Puede ser integrado en SOAP a nivel de los encabezados de mensajes
  - ✓ Gestiona la cohorte de Ti distribuidas
  - ✓ Define un conjunto de mensajes en pares solicitud-respuesta, necesarios para la coordinación de Ti mediante esquemas XML
    - Begin y begun: inicio de Ti
    - Context y context-reply: intercambio de contextos transaccionales
    - Request-status y status: intercambio del estado de una Ti
    - Enrol y enrolled: incorporar un servicio en una Ti
    - Prepare y prepared: preparar la validación
    - Confirm y confirmed: commit la Ti
    - Cancel y cancelled: abortar la Ti

UNIVERSIDAD DE LOS ANDES

Ejemplo BTP

<btp:request-status id?>

<btp:target-identifier>…URI… </btp:target-identifier>

<btp:qualifiers>? … calificadores … </btp:qualifiers>

<btp:target-additional-information>? … direcciones de información adicional …

</btp:target-additional-information>

<btp:reply-address>? … direcciones … </btp:reply-address>

</btp:request-status>

<btp:status id?>

<btp:responders-identifiers> … URI … </btp:responders-identifiers>

<btp:status-value> created | enrolling | active | resigning | resigned | preparing | prepared | confirming | confirmed | cancelling | cancelled | cancel-contradiction | confirm-contradiction | hazard | contradicted | unknown | inaccessible </btp:status-value>

<btp:qualifiers>? … calificadores … </btp:qualifiers>

<btp:target-additional-information>? … direcciones de información adicional …

</btp:target-additional-information>

</btp:satus>

UNIVERSIDAD DE LOS ANDES

Gestión de Ti-WS-Transaction

- Propuesta de Microsoft, IBM y BEA (julio 2002) como dialecto XML para coordinar los servicios web y soportar Ti simples (ACID) y largas (Ti de negocios)
  - ✓ WS-Coordination: coordinación de actividades
    - Servicio de activación con la creación de una instancia de coordinación caracterizada por un contexto, que contiene los datos necesarios para la relación coordinador-participante (identificador y comportamiento de la actividad y los protocolos usados)
    - Conjunto de protocolos de coordinación
    - Servicio de registro de uno o varios protocolos de coordinación disponibles
    - Integración en el encabezado de los mensajes SOAP
    - Coordinador genérico extensible: anexar nuevos protocolos o extensión de los protocolos existentes

UNIVERSIDAD DE LOS ANDES

Gestión de Ti-WS-Transaction

- ✓ WS-Transaction: soporte de Ti simples y largas
  - Para Ti atómicas
  - Servicio de coordinación usado para crear una Ti atómica asociada a un coordinador
  - Anexar coordinadores intemedios
  - Propagar el contexto en los mensajes SOAP
  - Registrar los participantes
  - Protocolos soportados: validación 2 fases (2PC), los que fuerzan la anulación o la aceptación de un eventual commit (completion y completionWithAck), los de notificación al comienzo de la validación (PhaseZero) o (OutcomeNotification)
  - Posibilidad de extensión de los protocolos existentes
- ✓ Mejora del 2PC: ReadOnly para evitar rehacer una Ti que sólo lee y Replay para rehacer una Ti

UNIVERSIDAD DE LOS ANDES

Gestión de Ti-WS-Transaction

- Ejemplo de servicios para participantes

```
<wsdl:portType name="2PCParticipantPortType">
   <wsdl:operation name="Prepare">
         <wsdl:input message="wstx:Prepare"/>
   </wsdl:operation>
</wsdl:portType>
```

OnePhaseCommit
Commit
Rollback
Unknown
Error

- Ejemplo de servicios del coordinador

```
<wsdl:portType name="2PCCoordinatorPortType">
    <wsdl:operation name="Prepare">
         <wsdl:input message="wstx:Prepare"/>
   </wsdl:operation>
</wsdl:portType>
```

Aborted
ReadOnly
Committed
Replay
Unknown
Error

UNIVERSIDAD DE LOS ANDES

Gestión de Ti-WS-Transaction

- Familia de protocolos para las actividades de negocios
  - ✓ Ti simples tienen una duración breve, la confianza de los participantes y mantienen sus bloqueos hasta terminar
  - ✓ Actividades de negocios son más largas y necesitan compartir sus recursos antes de terminar por ello el modelo de Ti propuesto es un modelo intermedio entre cohortes y sagas
  - ✓ Protocolos:
    - BusinessAgreement:
      - coordinador envía mensajes: Close (cierra subTi), Cancel (aborta subTi), Compensate (compensa subTi) y Forget (fin seguido de una falla de subTi)
      - Participantes envían mensajes de: Completed (fin con compensación o cierre), Faulted (falla de subTi), Closed (fin exitoso de subTi), Canceled (aborto efectuado) y Exited (fin definitivo)
    - BusinessAgreementWithComplete:
      - Igual al anterior pero el coordinador puede incluir el mensaje Complete

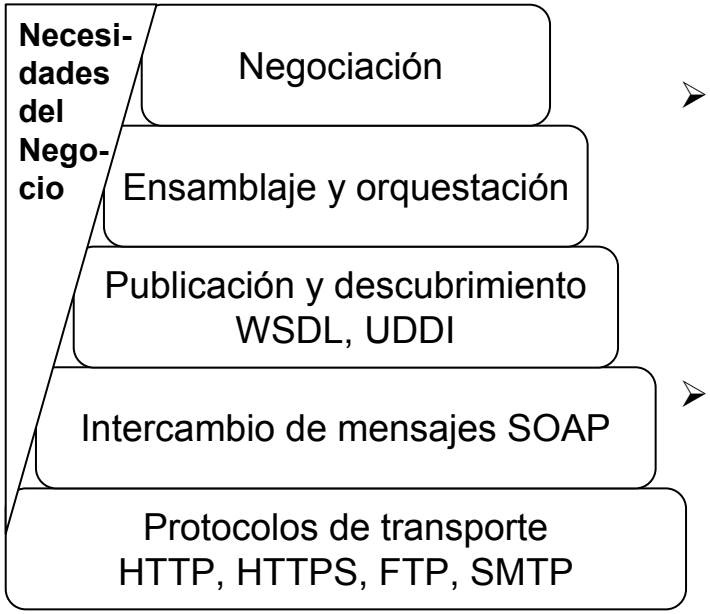
UNIVERSIDAD DE LOS ANDES

Gestión de procesos de negocios

- Procesos de Negocios: modelado (BMP-Business Process Modelling)
  - ✓ Módulo funcional que realiza el encadenamiento de operaciones distribuidas ejecutadas por los actores que intercambian mensajes, donde cada operación local tiene sus objetos y reglas de negocios
  - ✓ Actores o participantes pueden ser humanos o aplicaciones
  - ✓ Encadenamiento de operaciones: secuencias, ejecuciones paralelas o escogencias
  - ✓ Argot XML: WSFL (IBM), XLang (Microsoft), BPML (BMI.org), BPSS (ebXML) y BPEL4WS (IBM, Microsoft y BEA). Aún no hay estándar.
  - ✓ OMG propone el estándar para modelado de negocios EDOC (Extended Distributed Object Computing) basado en UML y con una arquitectura de colaboración, se puede considerar complementario a los argot XML

UNIVERSIDAD DE LOS ANDES

Arquitectura de integración de servicios web



- Orquestación de actividad: mecanismo de invocación, control y coordinación de las actividades que participan en un proceso del negocio
- Composición de servicios: técnicas que permiten ensamblar los servicios web para realizar los procesos del negocio mediante las primitivas de control (lazos, condiciones, excepciones, etc.) e intercambio (envío y recepción de mensajes)
- Funciones:
  - ✓ Modelado de los procesos de colaboración: participantes representados por roles
  - ✓ Composición de los servicios web: servicios simples o compuestos
  - ✓ Codificación y ejecución de los flujos de trabajo (workflows)

UNIVERSIDAD DE LOS ANDES

Gestión de procesos de negocios

Funcionalidad BMP	Funcionalidad clásica	Responsable
Modelado de procesos	Programación	desarrollador
Composición de servicios	Composición de módulos	desarrollador
Codificación de los flujos de trabajo	Generación de código ejecutable	Compilador y preprocesador
Gestión de Ti	Reinicio en caliente	Manejador de Ti
Almacenamien-to de mensajes	Bitácoras y reinicio en frío	Administrador
Seguridad de los mensajes	confidencialidad	Administrador/ Desarrollador

- Un flujo de trabajo describe como los participantes intercambian los datos y las condiciones de encadenamiento de actividades
- Modelo con redes de Petri con predicados: actividades = transiciones, mensajes = tokens y condiciones = predicados

- ✓ Gestión de transacciones y excepciones: Ti tipo sagas o cohortes
- ✓ Almacenamiento de mensajes: guardar la bitácora de mensajes intercambiados y sus operaciones, para robustez en caso de fallas
- ✓ Seguridad de mensajes: autenticación de usuarios, cifrado y firma de mensajes

UNIVERSIDAD DE LOS ANDES

Servicios web con WSFL

- WSFL de IBM basado en XML para flujo de control
  - ✓ Modelo de flujo: describe como los mensajes de los servicios agregados se transforman en mensajes a las actividades
  - ✓ Procesos de negocios compuestos se describen con un documento de raíz flowModel que especifica el nombre del modelo y el tipo de proveedor
  - ✓ Proveedor de servicio: entidad que gestiona un servicio web e implementa sus operaciones, colocada en un registro UDDI (businessEntity) que ofrece un conjunto de portType e identificado con locator de type static o de búsqueda (uddi)

```
<flowModel name="reservacionVacaciones" serviceProviderType="flujoReservacionVacaciones">
        .....
</flowModel>
<serviceProvider name="AgenciaX" type="AgenciaDeViajes">
        <locator type="static" service="AgenciaX.ve/av.wsdl" />
</serviceProvider>
```

```
<locator type="uddi" bindTime="startup"
selectionPolicy="first">
<uddi-api:find_service
businessKey="uddi_key" generic="1.0"
xmlns:uddi-api="urn:uddi-org:api">
```

UNIVERSIDAD DE LOS ANDES

WSFL

- Cuando una actividad A se termina, el control pasa a las actividades descendientes si la condición de la transición es cierta disparandose las actividades
- Actividad con más de un enlace de salida (fork). En la ejecución, el fork es efectivo si varios arcos desdendientes tienen la condición cierta y las actividades se ejecutan en paralelo
- Sincronización de actividades paralelas con nodos de acoplamiento (join). Una actividad resultado de un join se dispara solamente cuando todas las actividades entrantes al nodo join se terminan
- Enlaces de control permiten todas las estructuras de control clásicas de la programación estructurada

UNIVERSIDAD DE LOS ANDES

Ejemplo WSFL

```
<serviceProvider name="AgenciaX" type="AgenciaDeViajes">
        <locator type="static" service="AgenciaX.ve/av.wsdl" />
</serviceProvider>
<serviceProvider name="AirLines" type="AirCompany">
        <locator type="uddi" bindTime="startup" selectionPolicy="first">
          <uddi-api:find_service businessKey="uddi_key" generic="1.0" xmlns:uddi-api="urn:uddi-org:api">
                <tModelBag>
                        <tModelKey>UDDI:C1AIR26D-6534-0923-6H74-54D876E566</tModelKey>
                </tModelBag>
          </uddi-api:find_service>
        </locator>
</serviceProvider>
<activity name="reservarVacaciones">
        <performedBy serviceProvider="AgenciaX">
        <implement>
                <export>
                        <target portType="reservacionPT" operation="reservarVacaciones" />
                </export>
        </implement>
</activity>
<activity name="anularVacaciones" exitCondition="anulacionAceptada=True" />
```

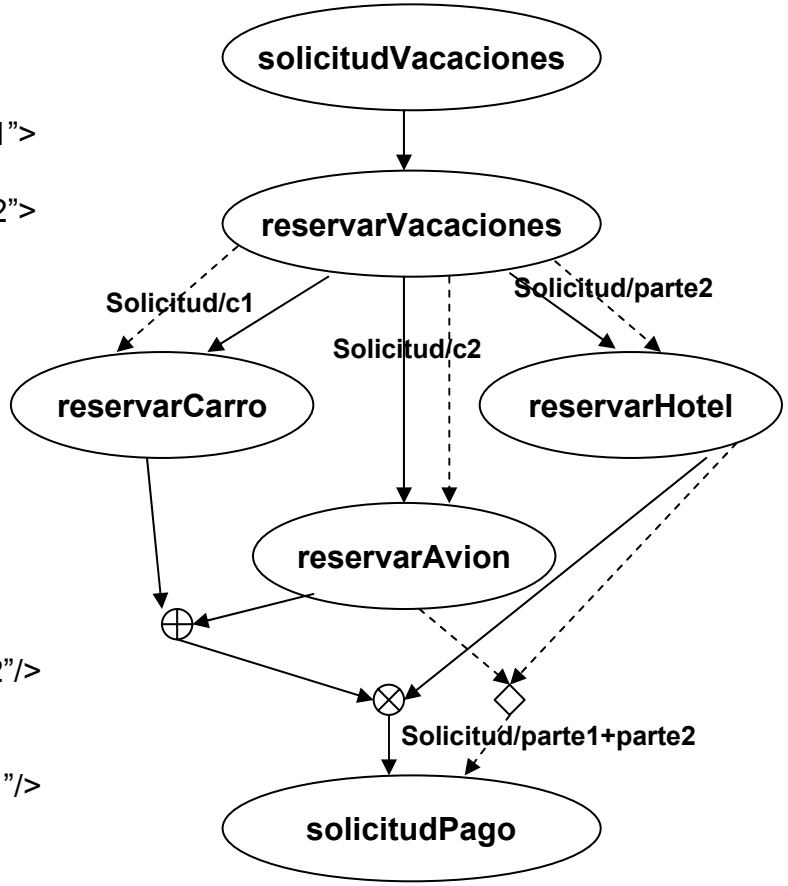
UNIVERSIDAD DE LOS ANDES

WSFL

Enlaces de control

```
<controlLink source="solicitudVacaciones"
             target="reservarVacaciones" />
<controlLink source="reservarVacaciones"
             target="reservarCarro" transitionCondition="Solicitud/c1">
<controlLink source="reservarVacaciones"
             target="reservarAvion" transitionCondition="Solicitud/c2">
<controlLink source="reservarVacaciones"
             target="reservarHotel" />
<controlLink source="reservarAvion"
             target="solicitudPago" />
<controlLink source="reservarHotel"
             target="solicitudPago" />
<activity name="solicitudPago">
  <join condition="(reservarCarro OR reservarAvion") AND
                   reservarHotel" when="deferred">
</activity>
<dataLink source="reservarVacaciones" target="reservarHotel">
  <map sourceMessage="Solicitud" targetMessage="Solicitud/parte2"/>
</dataLink>
<dataLink source="reservarVacaciones" target="reservarAvion">
  <map sourceMessage="Solicitud" targetMessage="Solicitud/parte1"/>
</dataLink>
….. Etc ….
```

Enlaces de datos



UNIVERSIDAD DE LOS ANDES

WSFL

- Modelo complementario:
  - ✓ Representación de servicios a partir de otros servicios, modelo global:
  - ✓ Grafo jerárquico de llamadas a las operaciones entre servicios web vistos como colecciones de operaciones WSDL
  - ✓ Un arco enlaza un servicio A a un servicio B si una operación de A invoca una operación en B
  - ✓ Mensaje de salida de A es el mensaje de entrada de B

Característica	Técnica de construcción utilizada
Modelado de procesos	Modelo de flujo que describe las interacciones entre las actividades que constituyen el servicio, con enlaces de control y de datos
Composición de servicios	Agregación de actividades asociadas a los proveedores de servicio. Servicio compuesto es un servicio
Codificación de los flujos de trabajo	Definición de transiciones condicionales entre actividades, las que son servidas por servicios simples o compuestos
Gestión de Ti	No soportado, incluido en otra especificación WSEL (Web Service Endpoint Language)
Almacenamien-to de mensajes	No soportado, incluido en otra especificación WSEL
Seguridad de los mensajes	No soportado, incluido en otra especificación

UNIVERSIDAD DE LOS ANDES

XLang

- Composición de servicios web en un contexto transaccional, complementario de SOAP y WSDL
- Lenguaje de orquestación de servicios web que integra un modelo de transacciones flexible
- Objetivo: permitir una especificación formal de los procesos de negocio como interacciones largas con estados (sagas)
- Contrato: especifica como un servicio invoca a otro servicio conectando sus operaciones de salida a las operaciones de entrada del otro
- Acciones compuestas con constructores: <sequence>, <switch>, <pick>, <all>, <while>
- Soporta definiciones de contexto y dentro de ellas de Ti, atómicas o no, con acciones de compensación en el modelo saga

UNIVERSIDAD DE LOS ANDES

XLang

```
<context>
        <sequence> ….. </sequence>
        <transaction atomic="false" name="Pago">
          <compensation>
                <sequence> …. Acciones de compensación  …. </sequence>
          </compensation>
        </transaction>
</context>
```

- Puede ser visto como una herramienta de modelado de procesos de negocios por organigramas. Constructores:
  - ✓ Begin: inicio del proceso
  - ✓ Action: proceso que recibe un mensaje de un puerto y envía un mensaje a un puerto, puede ser sincrona o no
  - ✓ Decision: evalúa reglas en la secuencia en que aparecen escritas
  - ✓ While: contiene una regla y representa un proceso repetitivo según una condición
  - ✓ Abort: anula la transacción y ejecuta eventualmente una compensación

UNIVERSIDAD DE LOS ANDES

Usado en Biztalk Server (EAI de Microsoft), abril 2002 WS-Routing para completar XLang para el flujo de datos

XLang

- Fork: abanico de hasta 64 procesos, todas las secuencias de procesos de negocio desde fork deben conectarse a un join
- Join: sincroniza los flujos simultáneos (and, or) en un proceso único
- Transaction: representa un conjunto de acciones atómicas. Ti anidadas con sagas
- End: fin del flujo de procesos

Característica	Técnica de construcción utilizada
Modelado de procesos	Procesos como interacciones entre servicios web con ayuda de secuencias, escogencias, iteraciones, condiciones, etc.
Composición de servicios	Como WSDL, servicio compuesto como WSDL que puede ser usado en una nueva composición
Codificación de los flujos de trabajo	Flujo de control según bloques estructurados en acciones secuenciales o paralelas. No se especifica flujo de datos
Gestión de Ti	Modelo saga flexible de Ti anidadas definidas en bloques contexto y bloques compensación
Almacenamien-to de mensajes	No soportado, incluido en otra especificación
Seguridad de los mensajes	No soportado, incluido en otra especificación

UNIVERSIDAD DE LOS ANDES

BPML

- Propuesto por BPMI.org en agosto 2000 cuya versión 0.4 aparece em 2001
- Mensaje: unidad de interacción entre participantes, esquemas XML usado para definir el contenido de los mismos
- Participante: entidades como sistemas, aplicaciones, usuarios, etc. Estáticas (conocidos de antemano) o dinámicas (encontrados durante el desarrollo del proceso)
- Actividad: unidades de ejecución que intercambia mensajes con los participantes. Simples o complejas (compuestas de simples)
  - ✓ Simple: producen y consumen mensajes y pueden hacer un tratamiento funcional
    - Acepta y consume el mensaje: consume
    - Produce y envía un mensaje: produce
  - ✓ Compleja: sequence, all, choice, switch, for each, repeat, join, etc.
- Reglas: permiten implementar la lógica del negocio, afectan la selección de una actividad y el consumo de un mensaje

UNIVERSIDAD DE LOS ANDES

BPML

- Transacción: coordinadas (simples) 2PC y extendidas (~sagas)
- Abstracciones de procesos: describen las interacciones entre un proceso y sus participantes
  - ✓ Proceso abstracto: definición de datos y mensajes
  - ✓ Ejecución de procesos: implementación concreta de un proceso abstracto

Proceso abstracto

```
<abstract name>
        <annotation />*
        <meta/>?
        <import />*
        <schema />?
        <message />*
        <participant name />*
        <ruleSet />*
        <activity name />*
        (simpleActivity | complexActivity )
</abstract>
```

Proceso concreto

```
<process name>
  <supports abstract />*
  <annotation />*
  <meta />?
  <import />*
  <schema />?
  <message />*
  <participant name />*
  <ruleSet />*
  <activity name />*
    (simpleActivity | complexActivity )
</process>
```

```
<abstract name="Reservacion">
  <message name="EntradaRes" type="request">
  <xsd:element name="service" type="xsd:string"/>
  </message>
  <message name="SalidaRes" type="response">
  <xsd:element name="rec" type="recibido" />
  </message>
  <operation name="reservar">
    <input message="EntradaRes" />
    <output message="SalidaRes" />
  </operation>
</abstract>
```

Ejemplo de proceso abstracto

UNIVERSIDAD DE LOS ANDES

BPML

- Objetivos:
  - ✓ Coordinar procesos de negocios colaborativos entre casas comerciales
  - ✓ Integrar aplicaciones existentes como procesos componentes y as{i ensamblar dichos componentes para fabricar nuevas aplicaciones
  - ✓ Autorizar el modelado de procesos independientes de los sistemas de oficina y los protocolos B2B
  - ✓ Tratar los problemas de fiabilidad gestionando las excepciones, retrazos e intervenciones humanas garantizando un estado consistente
  - ✓ Permitir múltiples implementaciones de procesos de negocio sobre varios sistemas integrando los cambios en el tiempo
  - ✓ Facilitar el intercambio de modelos entre sistemas y la compartición de un modelo de referencia
  - ✓ Soportar un formato de datos con estructuras, relaciones y tipos complejos
  - ✓ Integrar las diferentes actividades de la empresa, incluyendo la búsqueda de información, actualizaciones y difusión

UNIVERSIDAD DE LOS ANDES

BPML

```
<produce>
        <participant />
         <output />*
        extensión
</produce>
<operation>
        <participant />?
        <input />
        <output />
        <exception code />*
        extensión
</operation>
<join select />
<all | choice | sequence>
        activity
        activity+
</all | choice | sequence>
<switch exclusive?>
        <case />+
        <otherwise />?
</switch>
<spawn ref />
```

Permiten producir actividades nuevas

Característica	Técnica de construcción utilizada
Modelado de procesos	Procesos como mensajes intercambiados entre participantes y actividades simples o complejas, construidas con secuencias, escogencias, iteración, etc.
Composición de servicios	No está acoplado directamente a servicios web. La especificación del modelo aparte de la implementación de los procesos abstractos que es el servicio web
Codificación de los flujos de trabajo	Flujo de control según actividades complejas que soportan acciones secuenciales o paralelas. Flujo de datos especificados por los mensajes
Gestión de Ti	Modelo simple y saga flexible de Ti anidadas definidas por elementos transacción y elementos compensación
Almacenamien-to de mensajes	No soportado, incluido en otra especificación
Seguridad de los mensajes	No soportado, incluido en otra especificación

UNIVERSIDAD DE LOS ANDES

Ejemplo BPML

```
<produce name="notificarCliente">
        <participant select="EntradaRes/cliente" />
        <output message="notificacionCliente" />
</produce>
```

```
<operation>
        <input message="EntDepositarCheque" />
        <output message="SalDepositarCheque" />
        <exception code="cuentaInexistente" />
</operation>
```

```
<activity name="reservarVacaciones">
  <all>
    <produce>
        <participant select="hotel" />
        <output message="reservarHotel" />
    </produce>
    <produce>
        <participant select="AirLines" />
        <output message="reservarAvion" />
    </produce>
  </all>
</activity>
```

Dispara dos actividades en paralelo

Crea una nueva instancia del proceso actual: <spawn>

Crea 2 procesos y espera que terminen

```
<spawn reservarVacaciones>
<spawn reservarVacaciones>
<join select="reservarVacaciones">
```

```
<transaction type="nested" model="extended">
  <produce name="facturarCliente">
    <participant name="servicioFacturacion" />
    <output message="facturaRecibida" />
        <assign select="recibido" />
        <assign select="EntRecFactura/cliente" />
    </output>
  </produce>
  <compensate>
    <participant name="servicioFacturacion" />
    <output message="reembolsar" />
        <assign select="recibido" />
        <assign select="EntRecFactura/cliente" />
  </compensate>
</transaction>
```

Ti extendida con compensación (reembolso)

UNIVERSIDAD DE LOS ANDES

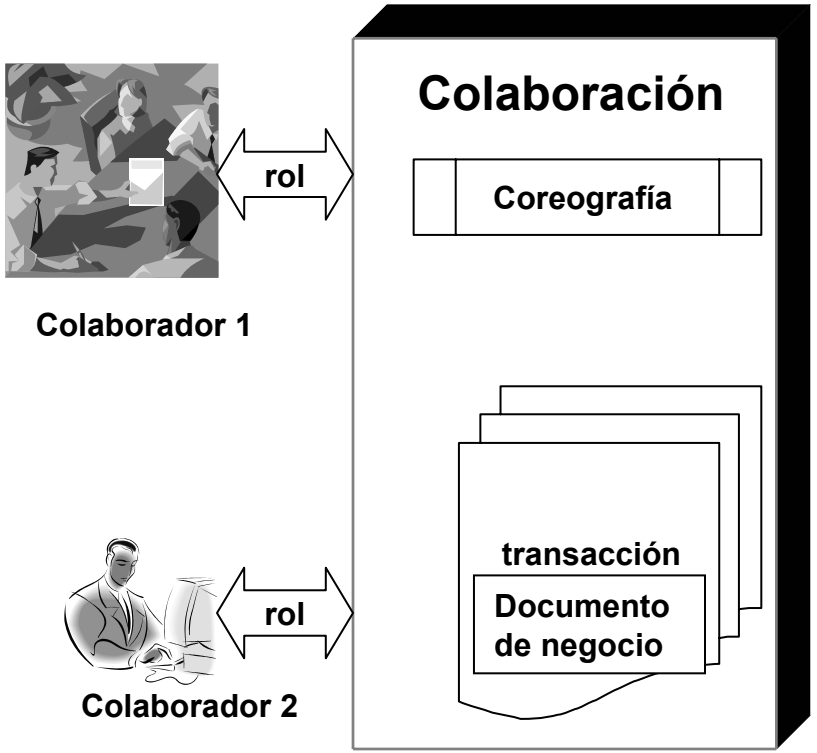
BPSS

- Grupo normativo de más de 700 empresas para modelos de comercio electrónico e intercambio comercial (ebXML), ebBPSS es un lenguaje XML de modelado de procesos de negocios
- Colaboración de negocio: compuesta por un conjunto de transacciones de negocio entre colaboradores
- Cada colaborador juega uno o varios roles en la colaboración
- Colaboraciones multi-partes se descomponen en binarias
- Colaboración binaria: conjunto de actividades donde cada una de ellas puede ser una transacción u otra colaboración binaria
- Transacción: unidad atómica de colaboración
- Documento de negocio: documento para realizar una transacción

UNIVERSIDAD DE LOS ANDES

BPSS

- Coreografía: describe el orden de ejecución y las transiciones entre actividades que componen una colaboración (~workflow), definidas con:
  - ✓ Start, complete, activity, synchronization, transition, fork, join
- Especificación completa en www.ebxml.org/specs/ebBPSS.pdf



UNIVERSIDAD DE LOS ANDES

BPSS

Ejemplo de transacción de negocio que consume un documento y envía otro

```
<BusinessTransaction
                        name="reservarHotel">
  <RequestingBusinessActivity
    name="reservar"
    isNotRepudiationRequired="true"
    timeToAcknowledgeReceipt="P2D"
    timeToAcknowledgeAcceptance="P3D">
   <DocumentEnvelope
    BusinessDocument="solicitudRes" />
  </RequestingBusinessActivity>
  <RespondingBusinessActivity
    name="reservar"
    timeToAcknowledgeReceipt="P5D">
     <DocumentEnvelope
      isPositiveresponse="true"
      BusinessDocument="aceptacionRes"/>
    </DocumentEnvelope>
  </RespondingBusinessActivity>
</BusinessTransaction>
```

Característica	Técnica de construcción utilizada
Modelado de procesos	Procesos como colaboraciones binarias recursivas entre roles que representan los colaboradores.
Composición de servicios	No está acoplado directamente a servicios web. La especificación del modelo aparte de la implementación. Servicios web permiten implementar Ti de negocios
Codificación de los flujos de trabajo	Flujo de control según UML, con transiciones y condiciones. Elementos de control son la coreografía de colaboraciones, actividades paralelas, secuenciales y condiciona.
Gestión de Ti	Ti consume un documento de negocio de entrada y produce uno de salida opcional. Ti simples y sujetas a restricciones temporales
Almacenamien-to de mensajes	Por cada documento de negocio (mensaje) el emisor puede guardarlo. Diversas opciones de seguridad.
Seguridad de los mensajes	Identidad del emisor puede ser verificada (autorización). Documentos de negocios cifrados y firmados (opcional)

UNIVERSIDAD DE LOS ANDES

BPEL2WS

- Especificación inicial que integra XLang y WSFL de IBM, BEA y Microsoft (julio 2002)
- BPEL4WS (Business Process Execution Language for Web Services)
- Partes: integradas en la section <process>
  - ✓ Especificación de los colaboradores del proceso <partners>
  - ✓ Contenedores de datos accedidos por los procesos <containers> descritos como tipos de mensajes y que soportan el estado de los procesos de negocio
  - ✓ Definición de las actividades a seguir en respuesta a los errores <faultHandlers>
  - ✓ Encadenamiento de las actividades

UNIVERSIDAD DE LOS ANDES

BPEL2WS

- Actividades para componer los procesos:
  - ✓ Esperar la recepción de un mensaje <receive>
  - ✓ Enviar un mensaje luego de recepción <reply>
  - ✓ Invocar una operación del servicio web <invoke>
  - ✓ Actualizar los valores de un contenedor <assign>
  - ✓ Generar una excepción <throw>
  - ✓ Terminar un proceso <terminate>
  - ✓ Esperar un periodo de tiempo o una fecha dada <wait>
  - ✓ Ejecutar una operación vacía <empty> (sincronización)
  - ✓ Ejecutar en secuencia una colección de actividades <sequence>
  - ✓ Seleccionar un caso de un conjunto <switch>
  - ✓ Esperar un mensaje o expiración de un retrazo <pick>
  - ✓ Lanzar varias actividades en paralelo <flow>
  - ✓ Definir una actividad anidada dentro de otra <scope>
  - ✓ Invocar una actividad de compensación <compensate>

Posibilidad de integrar otras actividades por un mecanismo de extensión declarado en un espacio de nombres

Hasta los momentos parece que este será el estándar de W3C

UNIVERSIDAD DE LOS ANDES

BPEL2WS

Característica	Técnica de construcción utilizada
Modelado de procesos	Procesos como actividades básicas de invocación de los servicios o de actualización de los contenedores. Actividades estructuradas con los constructores: secuencia, escogencia, iteración, etc.
Composición de servicios	Procesos ejecutables o abstractos que invocan los servicios. Proceso conectado a un servicio por un enlace . Procesos con una interfaz de servicio que invoca los servicios de la interfaz de otro .
Codificación de los flujos de trabajo	Flujo de control según actividades complejas que soportan acciones secuenciales o paralelas. Flujo de datos especificados en los envíos de mensajes y en las actualizaciones de los contenedores
Gestión de Ti	Dos modelos de Ti WS-Transaction, simple y clásico (Ti corta) y flexible con Ti anidadas largas y acciones de compensación (cohorte y sagas)
Almacenamiento de mensajes	Posible por la escritura de mensajes en los contenedores
Seguridad de los mensajes	Soportado por WS-Security

UNIVERSIDAD DE LOS ANDES

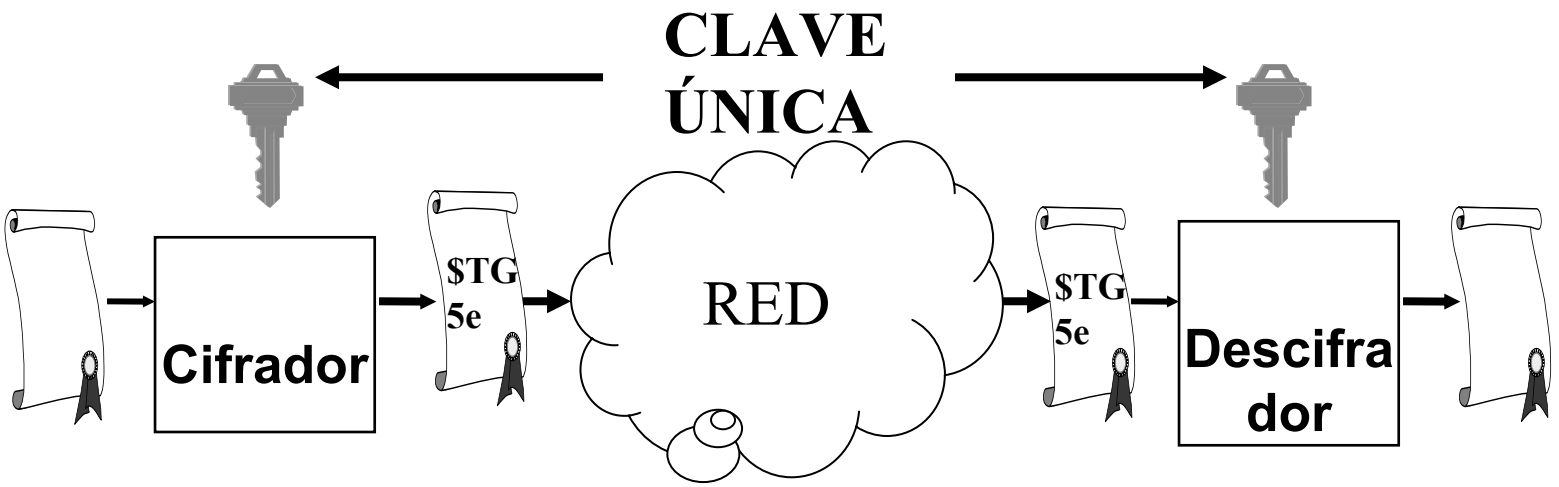
Cifrado de mensajes XML

- XML Encryption: estándar XML para intercambio de elementos XML cifrados, claves públicas de cifrado y las referencias a los algoritmos necesarios
- Gestión de claves de cifrado: XKMS (XML Key Management Specification)
  - ✓ Cifrado con claves asimétricas e intercambio de la clave pública
  - ✓ Cifrado con claves simétricas e intercambio de clave secreta cifrada
- Definición del tipo de dato cifrado <EncryptedData> (documento, elemento, contenido y especio de nombres usado para datos cifrados (xmlenc)
- XML-Encryption y XML-Signature de W3C

UNIVERSIDAD DE LOS ANDES

Algoritmos de cifrado

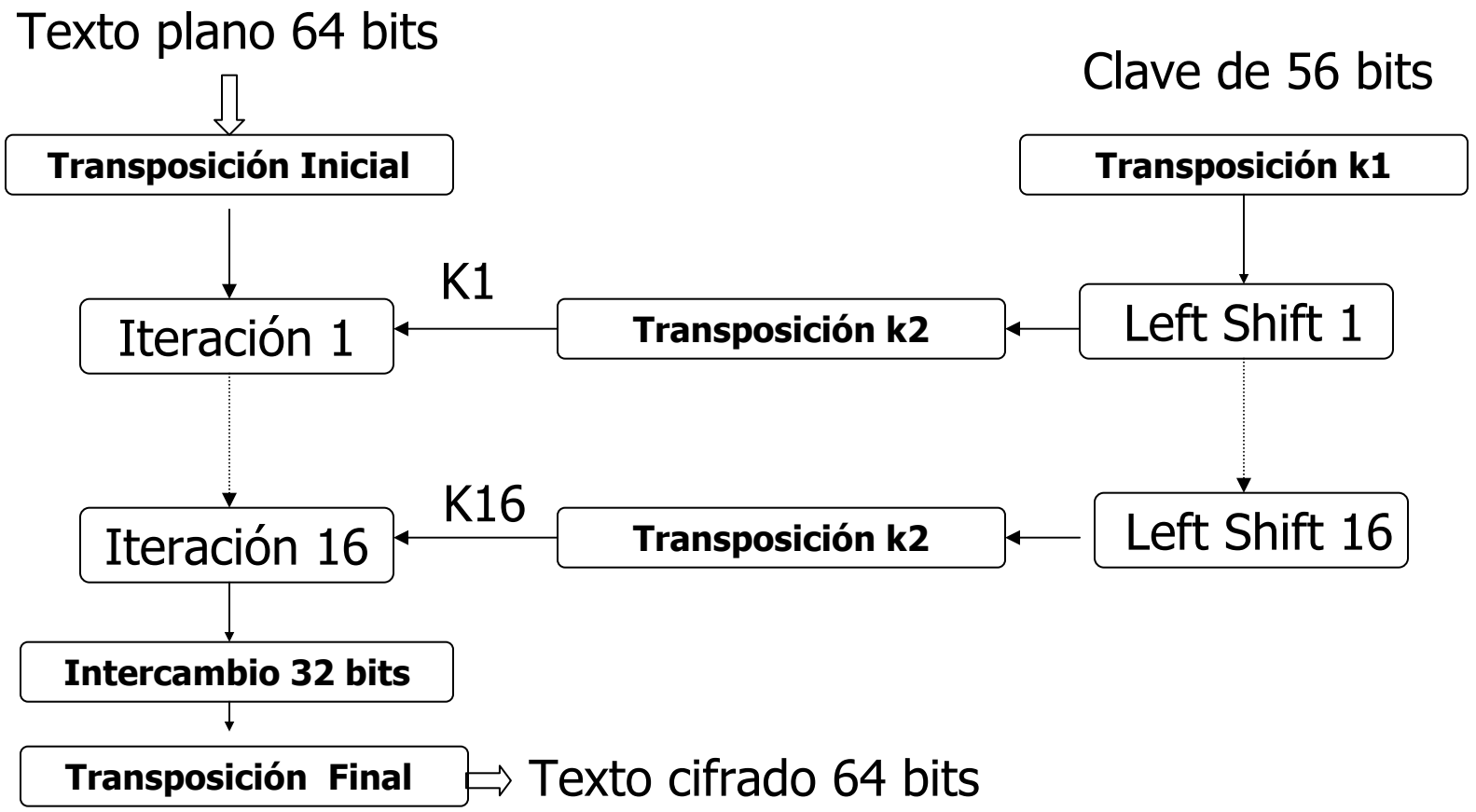
- DES y AES para datos
- RSA para claves



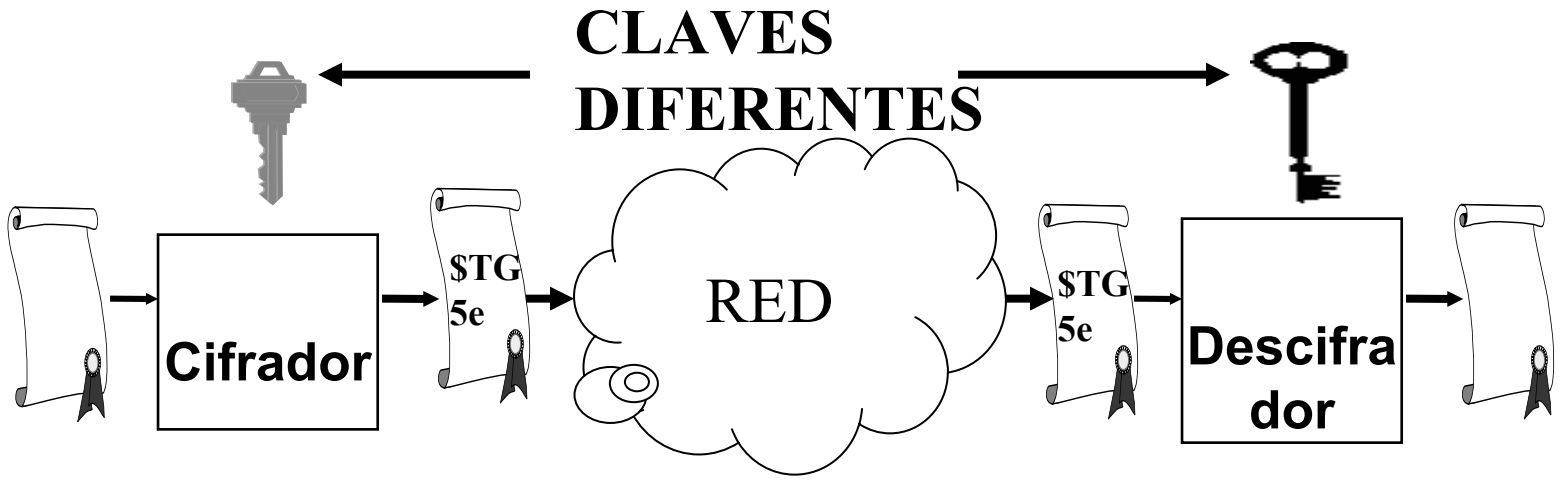
Pero: ¿Cómo se transfiere la clave?

UNIVERSIDAD DE LOS ANDES

Cifrado simétrico -DES - (Data Encription Standard)



Cifrado asimétrico (clave pública, PKI)



UNIVERSIDAD DE LOS ANDES

Algoritmo RSA

1. Seleccionar dos números aleatorios primos y grandes (más de 100 dígitos) p y q
2. N=pq
3. Escoger un entero pequeño E que sea relativamente primo a (p-1)x(q-1), es decir a Φ (n).
4. Calcular D tal que DE=1mod Φ (n)

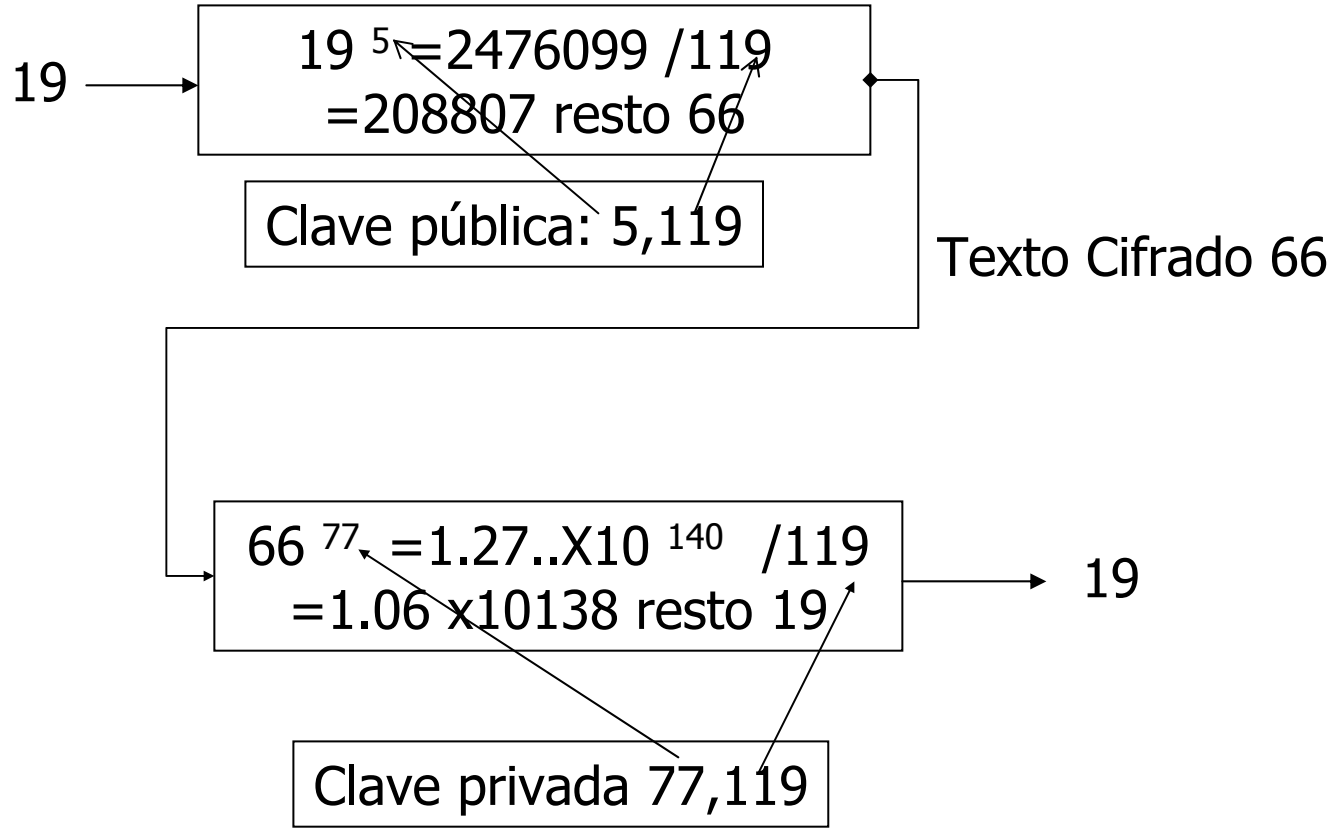
Entonces:

E y N constituyen la clave pública.

D y N constituyen la clave privada.

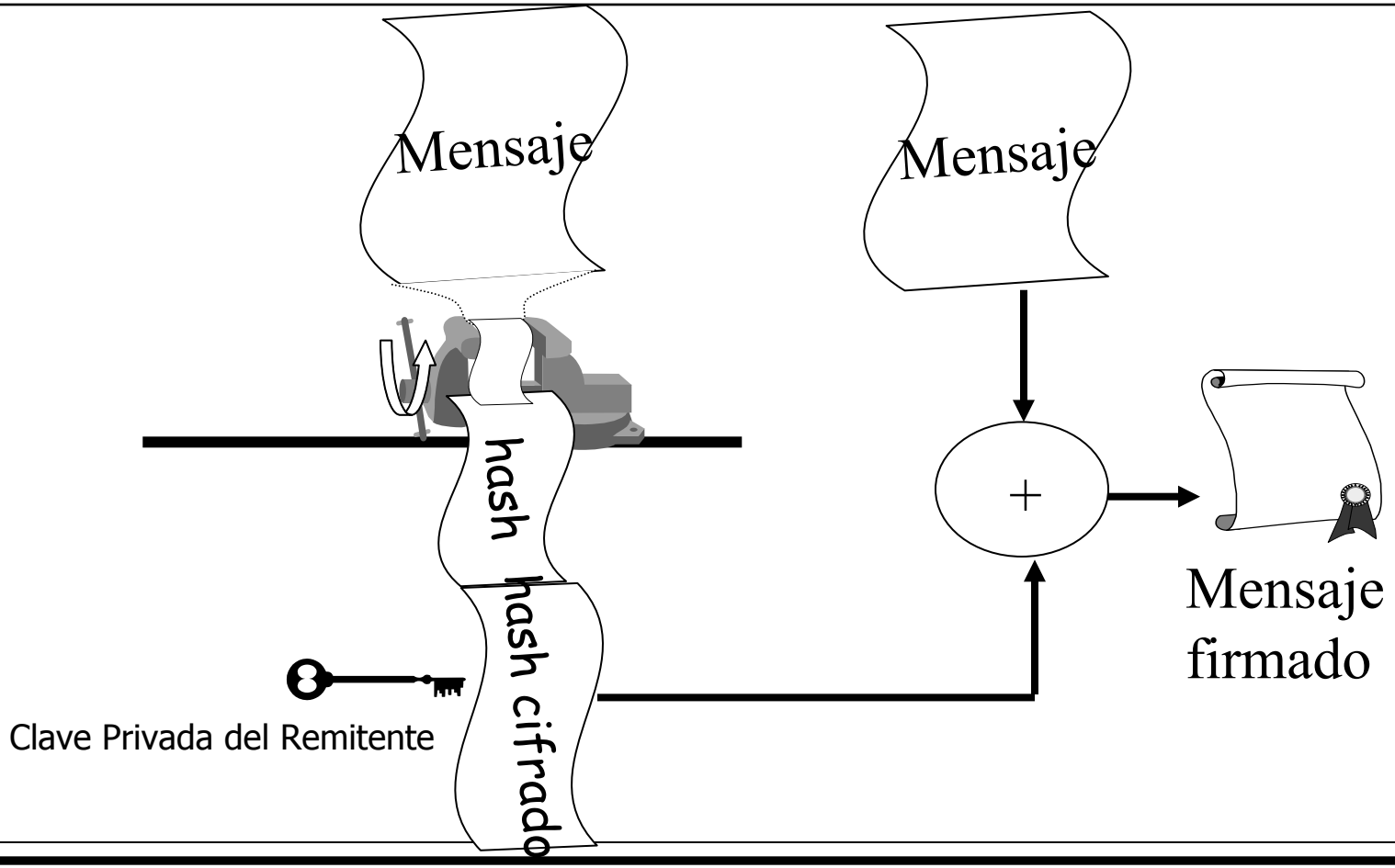
UNIVERSIDAD DE LOS ANDES

Ejemplo



UNIVERSIDAD DE LOS ANDES

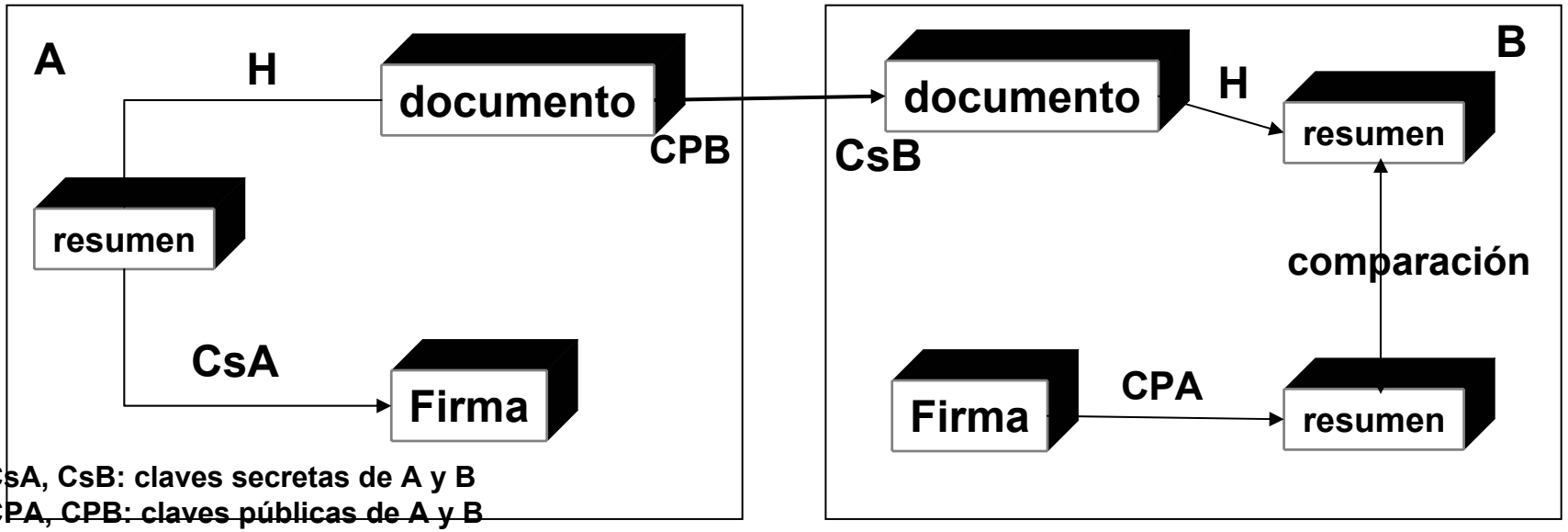
Firma digital



UNIVERSIDAD DE LOS ANDES

Firma XML

- Parte del mensaje XML que permite identificar de forma segura el emisor del elemento XML, combinación del elemento y la clave secreta, la descripción de la información firmada y las claves necesarias en la validación



UNIVERSIDAD DE LOS ANDES

Estructura de los mensajes

- Elemento raíz <Signature> con el identificador y el espacio de nombres
- Validación en dos etapas:
  - ✓ Calculada sobre el contenido de <SignedInfo>
  - ✓ De cada resumen de objetos firmados

```
<Signature>
  <SignedInfo> (método de canonización) (método de firmado)
    <Reference (URI)? >
                (Transformación) ?
                (método de validación de resúmenes-digest) ?
                (valor del digest)
    </Reference>
  </SignedInfo>
                (valor de la firma)
                (KeyInfo) ?
                ( Objeto ) *
</Signature>
```

UNIVERSIDAD DE LOS ANDES

XKMS

- Especificación de W3C y ETF para integrar la gestión de claves y certificados a las aplicaciones XML
- Partes:
  - ✓ X-KRSS (XML Key Registration Service Specification): gestión de claves
  - ✓ X-KISS (XML Key Information Service Specification): controlador de firmas
- Delegar el tratamiento de la seguridad a un servicio especializado en Internet que ofrece los servicios de gestión de claves públicas, certificados y firmas accesible en SOAP

UNIVERSIDAD DE LOS ANDES

WS-Security

- Propuesta por IBM, Microsoft y Verisign en abril 2002
- Extensión de SOAP para seguridad de servicios web
- Múltiples tokens de seguridad, dominios, formatos de firmas y técnicas de cifrado
- Mecanismos:
  - ✓ Propagación del token de seguridad
  - ✓ Verificación de la integridad del mensaje por su firma
  - ✓ Confiabilidad del mensaje por el cifrado de todo o parte del mismo
- Conceptos:
  - ✓ Solicitud: información a verificar realizada por un cliente
  - ✓ Token de seguridad: colección de solicitudes
  - ✓ Token de seguridad firmado: certificado X.509 o ticket Kerberos

UNIVERSIDAD DE LOS ANDES

WS-Security

- ✓ Prueba de posesión: datos usados por un proceso que prueban el conocimiento del emisor (sólo el emisor lo conoce)
- ✓ Integridad: procedimiento para verificar que el mensaje no fue modificado en la transmisión
- ✓ Confidencialidad: procedimiento para ver los datos protegidos

- Define como colocar los tokens, las firmas y las claves de cifrado en el encabezado de los mensajes SOAP
- Integración de XML-Signature y XML-Encryption en los mensajes SOAP

```
<S:Envelope xmlns:S="http://www.w3c.org/2001/12/soap-envelope" xmlns:ds="http://www.w3c.org/2000/09/xmldsig#">
  <S:Header>
        <wsse:Security xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext">
        <wsse:UserNameToken Id="ident">
                <wsse:UserName>NombreDelEmisor </wsse:UserName>
        </wsse:UserNameToken> ……..
```