Aplicación de los principios de SOA en las redes de telecomunicaciones, un estudio de FOKUS OpenSOATelco Playground

Por:

Daniel Alberto Ortiz Yepes

Asesor de Tesis:

Yezid Donoso

Tesis presentada como requisito de obtener el título de:

Ingeniero de Sistemas y computación

Universidad de los Andes

Facultad de Ingeniería

Departamento de Ingeniería de sistemas y computación

Bogotá D.C., Colombia

Junio de 2010

Aplicación de los principios de SOA en las redes de telecomunicaciones, un estudio de FOKUS OpenSOATelco Playground

Daniel Alberto Ortiz Yepes

Universidad de los Andes

Facultad de Ingeniería, 2010

Asesor de Tesis: Yezid Donoso

Resumen

Para la empresas de comunicaciones (telcos) la adopción de una arquitectura orientada a servicios (SOA) le puede representar beneficios que se manifiestan en la reducción de costos y el aumento en los ingresos. La reducción de costos se presenta al integrar sus sistemas de manera organizada y extensible lo cual le permite agilizar sus procesos operativos y de negocio y eliminar los sistemas redundantes. El aumento en los ingresos se presenta debido a que SOA favorece un modelo en donde la telco permite que terceras partes incluyan dentro de sus aplicaciones las capacidades de la red a cambio de un pago por uso. El éxito de SOA se manifestó inicialmente en el área de las tecnologías de información y ahora se busca transladar al área de las comunicaciones. En este documento se hace un estudio de OpenSOA telco playground, una plataforma que representa una implementación SOA con enfoque en el sector de las telecomunicaciones la cual busca entregar las ventajas mencionadas.

Índice general

1. Introducción **5**

2. Descripción general **8**
2.1. Objetivos 8
2.2. Antecedentes 8
2.3. Identificación del problema y su importancia 10

3. Diseño y especificación **12**
3.1. Definición del problema 12
3.2. Especificaciones 13
3.2.1. Requerimientos no funcionales 13
3.2.2. Requerimientos funcionales 13
3.3. Restricciones 14
3.4. Arquitecturas orientadas a servicios 14
3.4.1. Ventajas de SOA 16
3.4.2. Principios de SOA 16
3.4.3. Los servicios web 17
3.4.4. El bus de servicios empresariales 17
3.4.5. La composición de servicios y el lenguaje BPEL 18

4. Desarrollo del diseño **21**
4.1. Recolección de información 21
4.1.1. OMA Service environment 21
4.1.2. Prestación de servicios a terceros según OSE 22
4.1.3. Los estándares TMF NGOSS y TMF eTom y su relación con los sistemas de soporte de operación 23
4.2. Alternativas de diseño 24

5. Implementación **26**
5.1. Vista de niveles 26
5.2. Arquitectura funcional de OpenSOATelcoPlayground 27
5.2.1. Aplicaciones de terceros 28
5.2.2. Policy Evaluation Enforcement and management 28

5.2.3. Broker de servicios 29
5.2.4. Registro de servicio 30
5.2.5. OSS/BSS 31
5.2.6. Service creation environment 32
5.2.7. Web 2.0 telco enabler 33
5.2.8. Seguimiento de la solicitud de un servicio en OSTP 34

6. Validación **35**
6.1. Aplicación de OSTP en la IPTV 35
6.1.1. Bloques básicos de funcionalidad 35
6.1.2. Ejemplo de un evento de televisión con servicios premium y regular 38
6.1.3. Otros escenarios de IPTV sobre SOA 40

7. Conclusiones **41**

Capítulo 1

Introducción

En la actualidad el mundo de las tecnologías de la información se encuentra experimentando un auge de las tecnologías relacionadas con SOA. Por el lado de las telecomunicaciones desde los años ochenta se ha visualizado un panorama en donde las telco puedan permitirle a terceras partes hacer uso de las capacidades de sus redes con el fin de poder ofrecer un variado ecosistema de servicios que le representen gran valor al usuario. A pesar de que se han introducido tecnologías que permiten concretar esa visión con el paso de los años su adopción ha sido pobre debido a que la manera en que se deben implementar los servicios dista bastante con respecto a lo que la corriente mayoritaria de programadores está acostumbrado y también porque muchas telcos no se consideraban lo suficientemente listas para exponer sus capacidades.

A medida que pasa el tiempo la infraestructura de las telco se vuelve más compleja y se presentan sistemas duplicados que implementan la misma funcionalidad para diferentes servicios lo cual implica que el cliente tiene una vista inconsistente de los servicios que le presta la telco cada vez que hace uso de roaming, configura un servicio o intenta acceder a la red usando un dispositivo distinto. Para resolver estos problemas la industria ha dictado nuevas visiones de lo que debe ser el panorama del futuro. La visión de la *Open Mobile Alliance* dicta:

Sin importar que dispositivo tenga, Sin importar que servicio yo quiera, Sin importar que portador o red me encuentre usando, Yo puedo COMUNICARME, ACCEDER e INTERCAMBIAR información [1].

Por otro lado el TeleManagent Forum, la agrupación de las empresas prestadoras de servicios de telecomunicaciones, ha especificado los procesos de negocio más esenciales para una telco y cuales características debe tener el software que soporta el negocio y las operaciones de la organización, lo anterior tiene incidencia en la prestación de servicios en que se han identificado procesos de negocio claves para poder prestar un servicio. Para una telco no solo es importante prestar un servicio si no también la velocidad en que puede hacer que los elementos necesarios de su red estén debidamente configura-

dos para que se pueda entregar el servicio, adicionalmente a la prestación del servicio la telco debe asegurarse de que el servicio se está prestando bajo las condiciones pactadas de seguridad, calidad de servicio, etc.

La división FOKUS del instituto Fraunhoufer ha desarrollado la plataforma *OpenSOATelco Playground* (OSTP) la cual busca brindar una visión de cómo se puede incorporar SOA a la infraestructura de una telco, especialmente para aquellas que hacen uso de la arquitectura IMS. El primer reto que se debe superar a la hora de ofrecer funcionalidades para que sean explotadas por terceros es tener una amplia base de programadores que escriban aplicaciones y evitar que estos tengan que manejar los variados detalles de señalización con los que cuentan las redes de telecomunicaciones como el protocolo SIP en las redes IMS o SS7 en las redes de telefonía conmutada. Con el fin de buscar este objetivo las redes telco deben abstraer sus funcionalidades básicas en bloques conocidos como habilitadores (enablers) los cuales exponen interfaces basadas en WebServices los cuales son de amplio uso en la tecnología SOA, los Web Services a su vez pueden ser invocados desde páginas en JavaScript/AJAX la cual es una tecnología muy popular en la construcción de páginas web con funcionalidad enriquecida.

OSTP ofrece uno de los mecanismos más útiles de SOA: la orquestación mediante la cual se pueden combinar varios servicios expresados como web services y crear funcionalidades más complejas, todo esto sin la necesidad de escribir código al hacer uso del lenguaje BPEL. El ambiente de ejecución de servicios de OSTP se encuentra guiado por políticas las cuales permiten determinar las condiciones bajo las cuales se presta un servicio, estas condiciones se refieren a aspectos tales como seguridad, restricciones horarias, calidad de servicio y determina un mecanismo específico con el cual se deben tomar todas las solicitudes entrantes y aplicarles todas las políticas pertinentes.

Mediante el modelo de exposición de capacidades se favorece el aumento de ingresos debido a que la telco puede cobrar por el uso de sus servicios y usar esta funcionalidad como un factor diferenciador al crear un ecosistema de servicios que atraiga más usuarios. Al aprovechar la integración de los sistemas de soporte operacional con la infraestructura de comunicaciones la telco también se disminuye la complejidad en el despliegue y creación de servicio, disminuyendo el *time to market*.
Este documento está organizado de la siguiente manera:

En el capítulo 2 se describe la próblemática que le da origen a OSTP.

En el capítulo 3 se muestran las especificaciones de OSTP y los conceptos básicos de una arquitectura orientada a servicios.

En el capítulo 4 se comentan los estándares que influencian el diseño de OSTP y se menciona un diseño alternativo.

En el capítulo 5 se muestra la arquitectura de OSTP.

En el capítulo 6 se aborda un ejemplo que valida la idea de implementar SOA dentro de una telco.

Capítulo 2

Descripción general

En los últimos años las arquitecturas orientadas a servicio (SOA) se han convertido en una alternativa popular para la integración de aplicaciones al establecer un ambiente que garantiza atributos de seguridad, alto desacoplamiento, interoperabilidad y escalabilidad entre otros. Estas características le facilitan a las empresas de telecomunicaciones (también conocidas como telcos) integrar fácilmente sus redes de transporte, sistemas de prestación de servicio y sistemas de soporte de operación y negocio y a su vez permitir que aplicaciones de terceros hagan uso de sus funcionalidades de manera controlada. Como resultado la telco obtiene beneficios dentro de los cuales se encuentra el obtener una fuente adicional de ingresos, reducir sus gastos y simplificar varios procesos de negocio.

2.1. Objetivos

- Hacer una presentación de los conceptos generales de las arquitecturas orientadas a servicios (SOA).
- Hacer un estudio de la arquitectura de OpenSOATelco Playground (OSTP) y de las ventajas que presenta su adopción.
- Hacer un recuento de los estándares de la industria de comunicaciones que guían el diseño de OSTP.
- Presentar escenarios que ilustren los servicios de comunicaciones compuestos.

2.2. Antecedentes

Se considera que históricamente las industrias de las telecomunicaciones y de tecnologías de información recorrieron caminos diferentes debido a que la primera se enfocaba en la entrega de servicios de voz tales a través de la telefonía fija y la móvil y la

última se enfocaba en el desarrollo de aplicaciones. Esta diferencia histórica llevó a que cada dominio desarrollara sus tecnologías y prácticas específicas: por el lado de las telecomunicaciones se tienen tecnologías como el sistema de telefonía fija, SIP y el protocolo SS7 y por el otro se manejan tecnologías basadas en Java, C++, WebServices, RPC, Servlets, etc. La corriente mayoritaria de desarrolladores de aplicaciones está enfocada en las tecnologías de información y para ellos el desarrollar una solución dirigida a tecnologías de telecomunicaciones resultaba complicado debido a que los modelos de programación eran distintos a los que dominaban.

La idea de que un operador de servicios de comunicaciones le permita a terceros acceder a sus funcionalidades se remonta a la red inteligente de telefonía en los años 80 y se han presentado varias propuestas que a través del tiempo han tenido un nivel de aceptación bajo. A continuación se hace un recuento de estas principales propuestas las cuales están descritas con mayor detalle en [18].

- La red inteligente de telefonía introducida en los años 80 planteó la idea de establecer una capa de servicios encima de la red física con el objetivo de extraer la inteligencia del servicio de los conmutadores telefónicos y desplegar la funcionalidad de esta capa en puntos de servicio de control. La idea tuvo un alcance limitado debido a que el rango de servicios que se podían desarrollar era bastante reducido.
- En los Estados Unidos se introdujo la red inteligente avanzada de telefonía aplicable a fijos y móviles la cual incorporaba el modelo *Customized Aplications for Mobile Enhanced Logic* más conocido como CAMEL el cual se aplica hoy en día en varias redes de telecomunicaciones a nivel mundial y permitió la introducción de varios servicios. No obstante no se logró llegar al mercado abierto de servicios que en un principio se esperaba debido a lo extraño que le resultaba el modelo de programación a la mayoría de los desarrolladores.
- En los años noventa se introdujeron tecnologías de sistemas distribuidos tales como *CORBA* y *RMI* los cuales impulsaron la creación de APIs tales como *Parlay*, *3GPP Open Service architecture* (OSA) y *Java APIs for integrated networks* (JAIN) las cuales buscaban facilitar la implementación de servicios de telecomunicaciones al abstraer los detalles de señalización de las redes subyacentes tales como SIP o el *ISDN user protocol*. Mediante estas APIs se podía invocar el uso de servicios de control de llamada, conferencia, mensajería y presencia entre otros. La adopción de estos modelos siguió siendo baja debido a que las APIs tenían un grado de complejidad alto y en la época las telco no se encontraban lo suficientemente listas para abrir sus redes.
- A principios de los años 2000 *the Parlay Group* lanzó una versión simplificada de *Parlay/OSA* basada en la popular tecnología de Web Services llamada ParlayX la cual reconocía el Internet como el facilitador de un mercado de servicios abierto y tomaba provecho de la familiaridad que la mayoría de los programadores tenía con la tecnología de Web Services.

- El auge de la Web 2.0 ha introducido un modelo en el cual muchos sitios de Internet permiten que sus funcionalidades sean empleadas por terceras partes tal como sucede con Google o Facebook los cuales exponen sus funcionalidades mediante APIs que son invocadas por páginas de terceros. Estas páginas se conocen con el nombre de *mash-ups* o aplicaciones híbridas en donde se integra el contenido de diferentes fuentes, adicionalmente la Web 2.0 ha fomentado la creación de servicios hacía las comunidades de usuarios en donde el uso de dispositivos de comunicación juega un papel fundamental.

2.3. Identificación del problema y su importancia

Las grandes empresas se caracterizan por poseer un gran número de aplicaciones que corren sobre distintas plataformas, están escritas sobre múltiples lenguajes de programación y emplean diversos mecanismos de comunicación. Esta situación hace que la integración de dos o más aplicaciones sea una tarea compleja que a su vez facilita el establecimiento de funcionalidades redundantes. El rápido ritmo de cambio y crecimiento de las organizaciones conduce a este estado lo cual dificulta el establecimiento de una infraestructura de tecnología bien planeada y organizada, las telco no han sido ajenas a este fenómeno.

Por otro lado como consecuencia de la saturación de los ingresos por servicios de voz, la exposición de servicios para que sean empleados por terceros dentro de sus aplicaciones se plantea como una alternativa atractiva para que las telcos generen nuevas fuentes de ingresos. Esta apertura le da lugar a un modelo en donde ganan las telcos al obtener dinero cada vez que se usa su servicio, los desarrolladores de aplicaciones al crear un mercado para sus aplicaciones y el público usuario de servicios al disponer de una mayor variedad de servicios que aprovechen los dispositivos de comunicación que a diario usa.

La idea de introducir extensibilidad dentro de una plataforma de comunicación no es del todo nueva ya que por ejemplo IMS contempla el uso de servidores de aplicaciones SIP que implementen servicios de valor agregado. Mediante la introducción de SOA se busca enriquecer este modelo de extensibilidad al ofrecer numerosas características que hagan viable el modelo de apertura para una telco:

- Un eficiente y barato soporte del ciclo de vida de la prestación de un servicio que contempla las etapas de diseño, creación, aprovisionamiento, despliegue y administración con lo cual se quiere asemejar la oferta de servicios de una telco con un sistema de bloques lego los cuales sean fáciles de agregar, remover y cambiar de lugar.
- El soporte de un modelo de programación con el cual los desarrolladores de aplicaciones estén familiarizados como lo es el de Web Services y JavaScript y que no

requiere de un extenso tiempo de aprendizaje para comenzar a desarrollar aplicaciones que usen los servicios expuestos.

- Soporte para la aplicación de políticas que permitan aplicar directivas de autenticación, autorización, acuerdos de nivel de servicio entre otros mediante las cuales la telco puede establecer las condiciones bajo las cuales se presta el servicio.
- Integración entre la red de comunicación y los sistemas de soporte de negocio y operativos (OSS/BSS) lo cual lleva a una rápida detección en las fallas del servicio y agiliza el proceso de activación de un servicio.
- Conformidad con el estándar OSE el cual dicta la interoperabilidad entre servicios de comunicación y es de amplia aceptación en la industria. Con esta conformidad se busca que los servicios prestados no presenten barreras geográficas, de dispositivo o de operador.

Capítulo 3

Diseño y especificación

Existen dos factores que ejercen una gran influencia sobre el diseño de OSTP. El primero son las arquitecturas orientadas a servicios las cuales dictan una filosofía de diseño para los componentes del sistema y la segunda son los estándares del sector de las telecomunicaciones que especifican componentes y procesos comunes a todo el sector. A lo largo de esta sección se plantea el problema que busca resolver OSTP, las especificaciones de la plataforma, las principales restricciones a las que se encuentra sometido y se hace una introducción a las arquitecturas orientadas a servicios .

3.1. Definición del problema

En muchos países los tiempos de un mercado regulado de empresas de telecomunicaciones han quedado atrás. Hoy en día muchos de los mercados presentan condiciones de libre competencia o un nivel muy cercano a esta. Los clientes han desarrollado una demanda por calidad y para ellos es muy fácil desertar de un operador y cambiarse a otro si estos ven que sus expectativas no se cumplen.

Dentro del mercado uno de los factores determinantes para el éxito es la capacidad de ofrecerle al cliente un amplio ecosistema de servicios de valor agregado lo cual es consecuencia de un entorno de desarrollo de servicios que permite que estos se implementen rápidamente. Muchos servicios hoy en día están disponibles a través de internet y los clientes ya no están dispuestos a pagar para hacer uso de estos como por ejemplo Google Maps donde anteriormente los operadores cobraban por servicios similares [4]. Esta situación podría llevar a pensar que un operador no tiene como competir frente a la gratuidad de los servicios de internet pero esto no es cierto debido a que una telco cuenta con factores competitivos como la localización del usuario, la información de presencia y una QoS de alto grado.

Para una telco no solo es suficiente prestar un servicio si no que también debe prestarse de manera apropiada (En términos de disponibilidad, QoS, etc). Anteriormente el sec-

tor de las comunicaciones creía que el éxito de una telco se encontraba en la gestión de su red pero hoy en día se sabe que el éxito se halla en la gestión de los servicios lo cual fomenta una actitud proactiva donde no se espera hasta que el cliente se queje si no que se identifican en tiempo real los clientes que están siendo afectados [2].

3.2. Especificaciones

3.2.1. Requerimientos no funcionales

- **Seguridad:** La solución debe proveer mecanismos de verificar la autenticación y autorización de los usuarios que invocan un servicio con el fin de evitar que personas hagan uso de un servicio por el cual no han pagado o que se modifiquen las reglas de cobro de este.
- **Interoperabilidad:** La solución debe interactuar con los distintos protocolos y estándares que se manejan en el sector de las comunicaciones como ParlayX, SIP o SMPP y a su vez ofrecerle a los usuarios externos una abstracción de estos mediante web services y JavaScript con los cuales los programadores de aplicaciones se encuentran bastante familiarizados.
- **Flexibilidad:** La solución debe soportar un ambiente de rápida creación y despliegue de servicios mediante la cual se pueda reaccionar rápidamente a las necesidades del cliente, cambios en el mercado y nuevas oportunidades.
- **Reutilización:** La solución debe mantener una colección de bloques funcionales básicos los cuales puedan ser usados por varias aplicaciones finales y que no deban ser implementados de nuevo cada vez que se quiera componer un nuevo servicio.
- **Usabilidad:** La solución debe presentarle al usuario una visión unificada de todos los servicios lo cual significa que se deben asegurar características como un manejo de grupos común, un único perfil de acceso para múltiples servicios y la continuidad del servicio ante la movilidad del usuario o cambio de dispositivo .

3.2.2. Requerimientos funcionales

El principal requerimiento funcional de OSTP es el de permitir la ejecución de un servicio basado en políticas las cuales se encuentran separadas de la implementación funcional. Una política se puede entender como un mecanismo para describir cómo manejar recursos [17]. Mediante el uso de políticas se pueden controlar aspectos como:

- El direccionamiento de las solicitudes dependiendo el lugar de destino u origen de esta, por ejemplo el tratamiento de una solicitud destinada a un usuario en

Colombia podría tratarse de una manera distinta a aquella destinada a un usuario en los Estados Unidos.

- Verificar y asegurar el cumplimiento de los acuerdos de nivel de servicio.
- Verificar si el usuario solicitante se encuentra autorizado para hacer uso del servicio.
- Controlar la prestación del servicio dependiendo del día o la hora.
- Realizar diferenciación de usuarios de acuerdo a segmentos (gold, platino, normal).

3.3. Restricciones

Teniendo en cuenta que OSTP es un producto destinado a los telcos se deben tener en cuenta en la especificación los principales estándares que guían el sector con el fin de presentar un producto que se alinee con los requerimientos que expresan estos estándares. Los estándares con los que OSTP busca conformidad son:

- **TMF eTom (Enhanced telecommunications operations map):** Esquema de procesos de negocio requeridos dentro de una empresa de telecomunicaciones. Existen algunos procesos de negocio que son afectados por la operación de OSTP [8].
- **TMF NGOSS (Next generation operation support systems):** Sistemas de soporte de negocio y operaciones de próxima generación, este favorece la implementación de sistemas que poseen interfaces orientadas a servicio las cuales facilitan la integración de estas aplicaciones con las del resto de la empresa [9].
- **OSE (OMA Service environment)**: Principios arquitecturales que favorecen la interoperabilidad entre servicios de comunicaciones, adicionalmente estructura la manera en que estos servicios se exponen [10].

3.4. Arquitecturas orientadas a servicios

Dentro de los sistemas información en una organización de gran tamaño es frecuente encontrar que:

- Se posee una gran variedad de sistemas que corren sobre plataformas distintas, middleware distinto y están escritas en una gran variedad de lenguajes de programación. Adicionalmente muchos sistemas poseen diferentes dueños dentro de la compañía que están poco dispuestos a ceder el control sobre estos .

- Existe una interconexión con sistemas de otras organizaciones, las cuales pueden encontrarse a grandes distancias. Estos sistemas están sujetos a diferentes regulaciones y políticas de administración .
- Muchos de los sistemas presentes han estado en producción por mucho tiempo con lo cual es difícil modificarlos debido a que no se tienen los diseños de estos y es muy difícil encontrar a las personas que los desarrollaron.

Durante muchos años al problema de integración de aplicaciones dentro de las organizaciones le han surgido diferentes alternativas de solución. Es posible proponer que una de las posibles soluciones a este problema es la armonización, es decir tomar todas las aplicaciones y reescribirlas desde cero sobre una plataforma común lo cual no se considera apropiado debido a las implicaciones de riesgo y costo que tal emprendimiento implicaría.

Teniendo en cuenta lo anterior muchos vendedores de software intentaron colocar en el mercado tecnologías que integren aplicaciones heterogéneas con el problema de que estas soluciones eran propietarias y presentan muchos problemas a la hora de cruzar fronteras organizacionales en gran parte debido a que el intercambio de mensajes es bloqueado por los firewall ya que este se realiza a través de puertos no convencionales.

El siguiente paso después de las soluciones propietarias es el de las arquitecturas orientas a servicios las cuales consideran a una empresa como un sistema distribuido a gran escala que cuenta con una gran variedad de sistemas heterogéneos. Como tal SOA no es una tecnología específica si no una manera de estructurar los sistemas de información de una empresa con el fin de favorecer la integración entre estos. Dentro de los elementos de una solución SOA se presentan tres elementos fundamentales [7]: Los **servicios** representan las unidades de funcionalidad sobre las cuales se apoyan los procesos de negocio, la **infraestructura** representa los elementos que se encargan de implementar las funciones de interconexión aunque muchas veces también se aplican otras funciones como seguridad, aplicación de políticas, gestión de la calidad del servicio (QoS), etc. y **las políticas y procesos** representan los mecanismos que permiten darle gestión a este sistema heterogéneo con múltiples dueños.

Existen muchas tecnologías que pueden usarse para implementar SOA, de hecho SOA no favorece ninguna de manera explícita. No obstante los servicios web han tomado mucha fuerza debido a que estos representan un estándar mínimo de interoperabilidad acordado por los grandes fabricantes de software como Oracle, Microsoft, HP, SAP, etc. y se ejecutan sobre tecnologías de comunicación conocidas como http o el correo electrónico debido a que en la mayoría de los firewalls los puertos que estas emplean se encuentran abiertos.

3.4.1. Ventajas de SOA

Al incorporar SOA a una organización se espera obtener ciertos beneficios que ayuden a superar las fallas mencionadas anteriormente, los principales atributos de calidad que se ven mejorados son:

- **Interoperabilidad:** Facilita la invocación de servicios desde sistemas heterogéneos.
- **Flexibilidad:** Facilita el rápido cambio de un servicio con el fin de que se puedan seguir los requerimientos del negocio.
- **Bajo acoplamiento:** Al distribuir la funcionalidad de un servicio sobre varios subsistemas se debe procurar minimizar las dependencias entre los componentes del sistema con lo cual se reducen los efectos de los cambios y las fallas.

3.4.2. Principios de SOA

En [6] se citan 4 principios que deben seguirse para poder implementar una arquitectura SOA exitosa:

1. Las fronteras son explícitas: Muchas veces una llamada a un servicio puede implicar que el mensaje cruce distintos dominios administrativos lo cual indica que se deben considerar mecanismos de autenticación que permitan que las operaciones se ejecuten de manera segura.

2. Los servicios son entidades autónomas: Un servicio no se encuentra incrustado dentro de una aplicación cliente o un componente, este debe considerarse como una entidad autónoma que posee su propio ciclo de vida y contrato de funcionamiento.

3. Los contratos y los esquemas se comparten, más no las implementaciones: Para que una llamada de servicio funcione lo único que debe conocer el consumidor es el formato y la secuencia en que se envían los mensajes mediante la cual se favorece la flexibilidad al poder cambiar fácilmente la implementación de un servicio sin que el consumidor se de cuenta del cambio.

4. La compatibilidad del servicio está basada en las políticas: Se favorece la existencia de políticas dentro de una solución SOA donde una política se entiende como un conjunto de declaraciones comprensibles por máquinas que permiten describir los requerimientos no funcionales de un servicio. Estas políticas se anexan con el contrato del servicio. Al separar la política de la implementación de un servicio esta se puede colocar en un repositorio dedicado y ser alcanzada dinámicamente en tiempo de ejecución, adicionalmente le permite al consumidor de un servicio adaptarse de manera más fácil a los requerimientos de un proveedor.

3.4.3. Los servicios web

Los WebServices son un conjunto de estándares [21] que representan el mínimo común denominador de funcionalidad que debe tener un servicio para lograr interoperabilidad. Estos estándares han sido respaldados por los mayores vendedores de software los cuales han comprometido seguirlos. Los tres estándares principales son SOAP, WSDL y UDDI aunque existen diferentes estándares que cubren aspectos como seguridad, confiabilidad, transacciones y metadatos entre otros.
SOAP Soporta la invocación de servicios remotos apoyándose en el uso de http, udp o smtp entre otros como mecanismo de transporte, un mensaje SOAP consta de las siguientes partes:

- Sobre: Delimita el mensaje a transmitir, es obligatorio.
- Encabezado: Es opcional y tiene información acerca de la prestación del servicio relacionada con la transaccionalidad y seguridad.
- Cuerpo: Contiene la información a ser transmitida.

WSDL Describe un servicio web, muchas veces este es generado de manera automática por las herramientas de desarrollo. El mensaje tiene las siguientes partes:

- Tipos de datos que maneja el servicio.
- Tipos de mensaje que intercambia el servicio (HTTP, JMS, MQ, etc).
- Operaciones disponibles del servicio.
- Tipos de puerto.
- Localización de servicios.
- Descripción de un servicio.

UDDI es un servicio de directorio que sirve para hacer localización dinámica de WebServices, adicionalmente permite publicar y descubrir información sobre WebServices.

3.4.4. El bus de servicios empresariales

El bus de servicios empresariales (ESB) es un bloque de infraestructura que goza de amplia aceptación en diseños SOA. Su misión principal es la de proveer una infraestructura de transporte que permita a enlazar al consumidor de un servicio con su proveedor. Un ESB es uno de los principales elementos que aseguran la interoperabilidad debido a que permite interconectar aplicaciones heterogéneas [19].

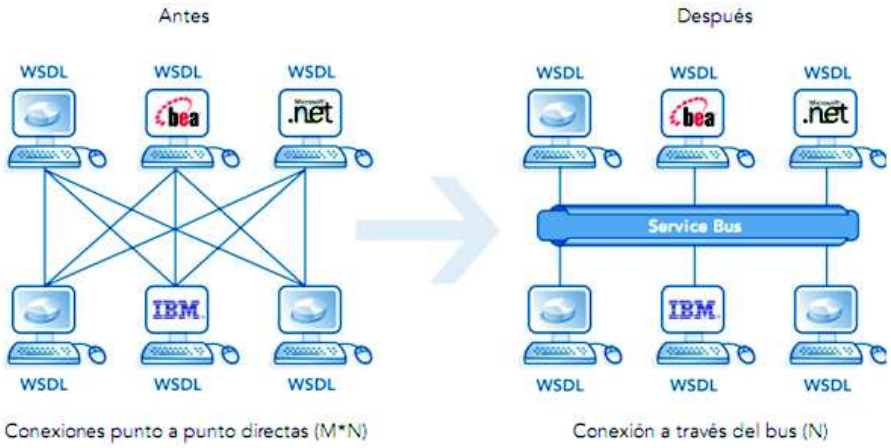

Figura 3.1: Evolución de la integración uno a uno a la integración por medio de un bus de servicios, imagen [19].

Un ESB es una alternativa de integración bastante atractiva frente a la solución punto a punto debido a que este provee flexibilidad y simplicidad ya que para responder a un cambio en un servicio solo se debe modificar la configuración del ESB mas no se debe incurrir en una redacción adicional de código.

Generalmente el formato de transporte más popular es el XML mediante el cual se pueden realizar transformaciones de datos de manera instantánea. Un ESB tambien puede implementar funciones adicionales como la seguridad y la composición de procesos. Una funcionalidad de mayor importancia es la de mediación con la cual un consumidor de servicio no tiene que dar la dirección del proveedor si no que con solo saber si el servicio existe le pasa la solicitud al bus y este encuentra el nodo que implementa la funcionalidad del servicio, esta última funcionalidad permite implementar el balanceo de carga y la tolerancia a fallos. Adicionalmente un ESB se caracteriza por implementar varios patrones de mensajería tales como publish-suscribe, asíncrono, síncrono y solicitud-respuesta entre otros.

3.4.5. La composición de servicios y el lenguaje BPEL

Uno de los objetivos de SOA es el de eliminar las implementaciones redundantes de una misma funcionalidad ya que esto aumenta la complejidad de los sistemas en la organización y dificulta la mantenibilidad por que a la hora de hacer un cambio en una funcionalidad esta se debe hacer en varios lugares. Para ayudar a eliminar la redundancia SOA promueve extensamente la reutilización de componentes lo cual busca que se mantengan servicios que representen una funcionalidad fundamental y posteriormente estos se combinen para obtener servicios de menor granularidad que representen funcionalidades de negocio más complejas.

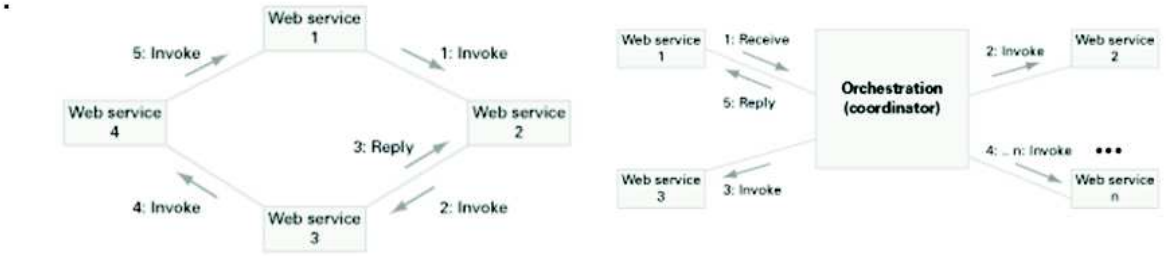

Figura 3.2: Composición de servicios web. A la izquierda se muestra la coreografía y a la derecha la orquestación, imagen [11].

Existen dos patrones de composición de servicios: el de orquestación y el de coreografía. La orquestación se basa en un servicio que invoca las funcionalidades de otros servicios de manera centralizada, en la coreografía cada servicio gestiona la interacción con otros servicios y conoce cuando ejecutar las operaciones [11].

Para la orquestación de servicios, la cual se emplea en OSTP ha tomado gran popularidad el lenguaje *Business Process Execution Language for web services*, más conocido como WS-BPEL. BPEL es un lenguaje que describe la interacción de los web services en XML con lo cual se apoya del WSDL que provee cada web service para conocer su interfaz, dentro de las principales operaciones se consideran:

- La invocación de otro servicio.
- La espera de la invocación del servicio presente por parte de otro servicio.
- Respuesta síncrona.
- Manipulación de variables.
- Gestión de excepciones.
- La espera de tiempo.
- La terminación del proceso.
- Invocación de actividades en paralelo.
- El case switch para la implementación de condicionales.
- El while para implementar ciclos.

En la figura 3.3 se muestra un ejemplo de un servicio orquestado. A pesar de que la sintaxis gráfica de BPEL no se encuentra estandarizada esta se emplea para proveer más claridad. La implementación de BPEL que OSTP emplea es la de activeBPEL [3].

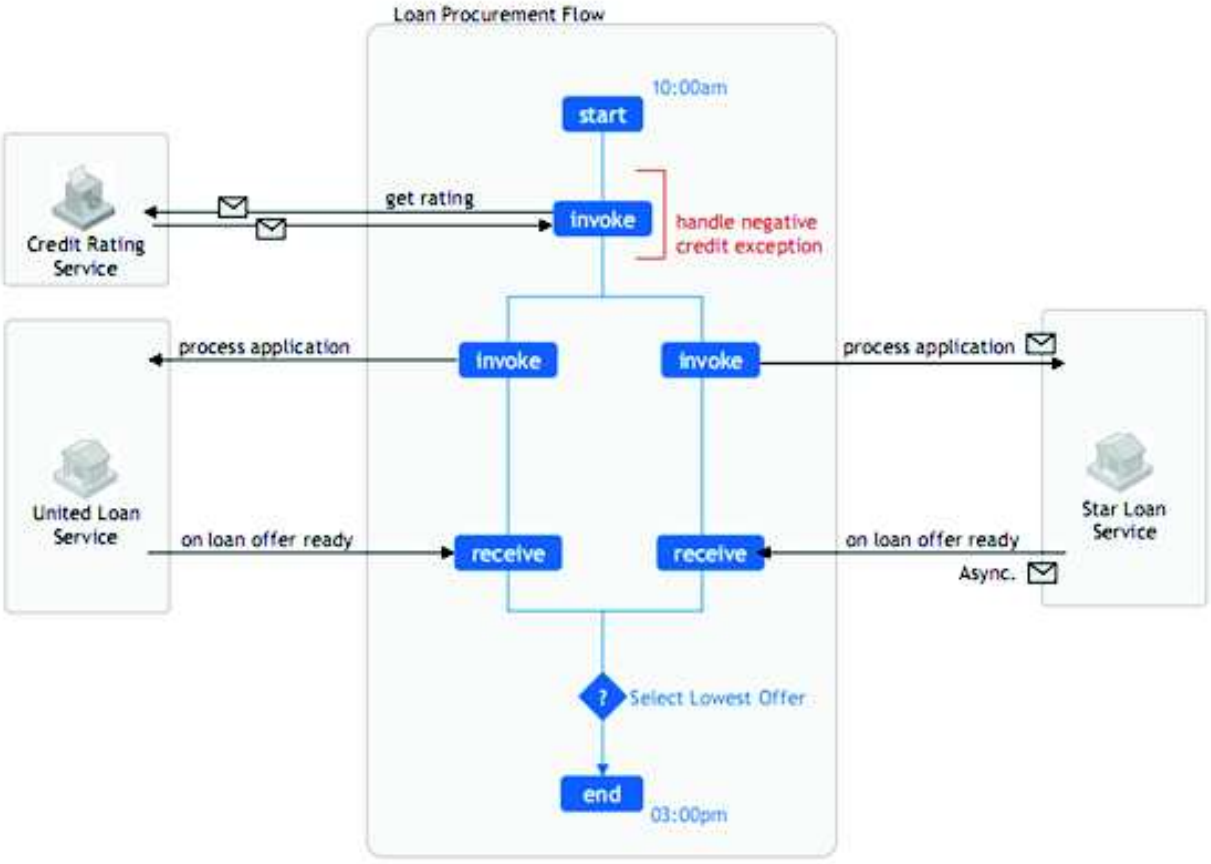

Figura 3.3: Ejemplo de orquestación de servicios web usando BPEL, imagen [11].

Capítulo 4

Desarrollo del diseño

4.1. Recolección de información

4.1.1. OMA Service environment

La *Open Mobile Alliance* (OMA) es un consorcio fundado en el 2002 que se propone concertar estándares que dicten la manera en que los servicios de datos deben diseñarse con el fin de que estos sean interoperables a través de múltiples dominios.

Una de las entidades que especifica OMA es el *enabler* (habilitador) el cual representa la abstracción de un servicio de comunicación. Un servicio de comunicación se puede entender como una capacidad específica de una red telco tal como la mensajería de texto, conferencias, presencia, etc. La definición formal de un habilitador es:

> *Un enabler es una tecnología que se emplea en el desarrollo, despliegue u operación de un servicio definida en una especificación o grupo de especificaciones publicadas como un paquete por la OMA.*

Cada habilitador especificado por OMA ha pasado un programa de pruebas de interoperabilidad para asegurarse que este se adhiera a las especificaciones impulsadas por esta organización [1]. Como resultado de esta visión OMA ha divulgado el estándar *open service environment* (OSE) [10] el cual especifica una arquitectura para el desarrollo y despliegue de habilitadores OMA y la interacción entre estos. Uno de los aspectos claves de OSE es que sigue los principios de SOA como por ejemplo el acoplamiento débil.

Dentro de OSTP OSE juega un papel muy importante debido a que los diseñadores han elegido conformarse a este estándar lo cual incide en la manera en que las terceras partes se conectan a la infraestructura de OSTP y en el reconocimiento de los habilitadores como el bloque de diseño más simple con el que puede contarse a la hora de diseñar un servicio.

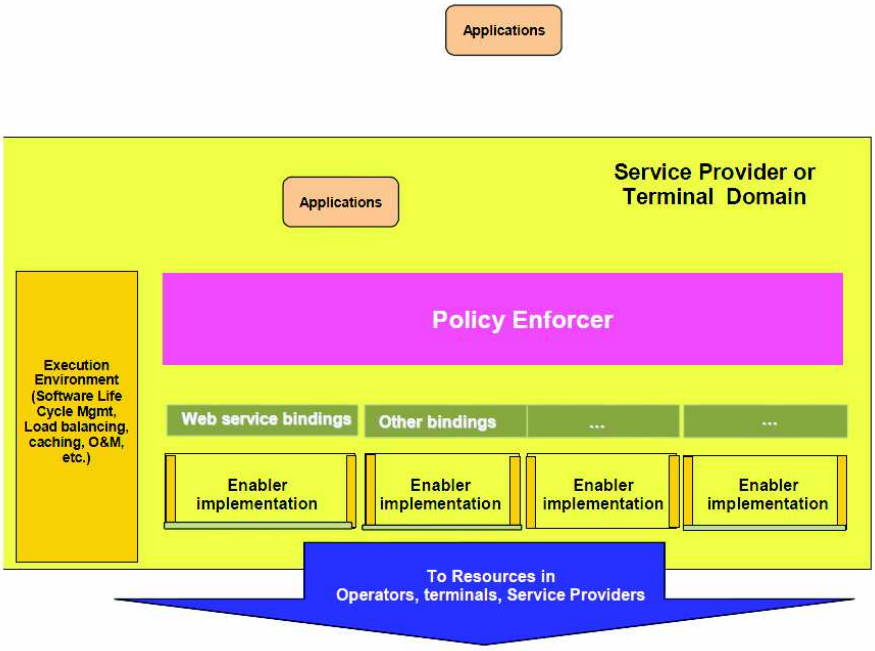

Figura 4.1: Vista de niveles del modelo arquitectural OSE, imagen [10].

Dentro de los principales objetivos del OSE se encuentran [10]:

- Facilitar la integración y manejo de recursos al favorecer la migración de arquitecturas tipo silo lo cual ayuda a reducir las complejidades de integración y desarrollo y resalta la importancia de la seguridad, la usabilidad y la privacidad.
- Fomentar la exposición controlada de recursos a desarrolladores de aplicaciones de terceros para que ellos creen y sostengan nuevos servicios a la vez que se protegen los activos del proveedor de servicios.
- Desarrollar un camino de evolución para un ambiente de servicio integrado y unificado.

4.1.2. Prestación de servicios a terceros según OSE

El modelo arquitectural OSE brinda una visión multinivel ilustrada en la figura 4.1 que facilita que el prestador de servicios le pueda exponer sus funcionalidades a terceros. OSE incorpora interfaces de servicios web las cuales traducen los servicios directamente a la lógica del habilitador o a un protocolo específico del habilitador.

- El nivel de aplicaciones representa los *mash ups* que ofrece un tercero o incluso la misma empresa, estos a su vez hacen uso de varios habilitadores.

- El nivel de habilitadores representa la abstracción de las capacidades de las redes de comunicaciones. Esta capa muchas veces realiza labores de traducción manejando WebServices hacia las capas de arriba y manejando el lenguaje específico de señalización de las redes de comunicaciones hacia abajo.

- El nivel de Cumplimiento de políticas representa el punto de autenticación y aquel por el cual el operador determina si se presta el servicio o no y bajo qué condiciones se presta. Adicionalmente se aplica la facturación (charging) , el registro de eventos (logging) y la aplicación de las preferencias del usuario.

4.1.3. Los estándares TMF NGOSS y TMF eTom y su relación con los sistemas de soporte de operación

Los sistemas de soporte operativo (OSS) son un conjunto de programas diseñados específicamente para manejar una infraestructura de red de gran tamaño y soportar procesos como inventario, planeación, facturación entre otros para las redes telco [9].

Se han identificado una gran variedad de problemas dentro de las telco debido a que con el paso de los años se han establecido silos de información representados por sistemas unidos mediante distintos protocolos como CORBA y SNMP, cada uno con diferentes repositorios de servicio y modelos de información lo cual se traduce en costos de operación elevados que se vuelven cada vez más notorios a medida que la complejidad de las infraestructuras aumenta.

Con base a estos problemas se ha identificado la necesidad de avanzar hacia sistemas que abarquen toda la empresa (Enterprise Wide). El primer paso hacia este tipo de sistemas lo ha dado el *Telecomunication Management Forum* (TMF) el cual ha puesto en circulación el estándar *eTom* donde se describen todas las actividades y procesos que un proveedor puede usar dentro de un marco SOA. Uno de los aspectos importantes de este estándar es que va más allá de los aspectos operacionales e incluye los aspectos de la calidad del servicio ofrecido al cliente [2].

Dentro de la filosofía de desarrollo de un OSS se encuentran diferentes aspectos como la unificación de los procesos de creación de servicios, procesos de negocio y procesos de gestión haciendo uso de mecanismos basados en flujos de trabajo [15].

El siguiente paso en cuanto a los sistemas OSS se refiere está dictado por el marco de referencia *Next Generation Operations System and Software* (NGOSS) el cual también es gestionado por TMF y es de gran aceptación en la industria. OSTP busca incorporar la compatibilidad con los sistemas NGOSS en dos procesos clave:

- **Aprovisionamiento del servicio:** Tiene como objetivo entregarles el servicio a los clientes a tiempo y de manera correcta lo cual involucra que todos los elementos

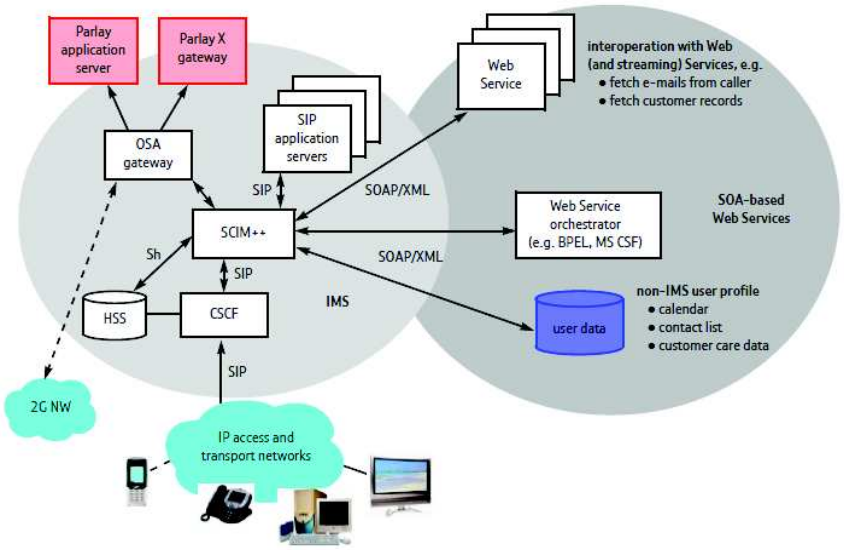

Figura 4.2: Diagrama de bloques de la orquestación de servicios haciendo uso del SCIM, imagen [20].

de infraestructura necesarios para la prestación del servicio deben configurarse de la manera apropiada.

- **Aseguramiento del servicio:** Tiene como objetivo asegurarse de que la prestación del servicio se dé en las condiciones adecuadas cubriendo parámetros como la calidad del servicio (QoS) e involucra actividades como el monitoreo del estatus y desempeño de la red.

4.2. Alternativas de diseño

Existe otro enfoque alternativo en cuanto a la prestación de servicios que se basa en realizar la orquestación directamente dentro de la red IMS en vez de realizarla sobre una capa de abstracción . Este enfoque hace uso del componente SCIM para hacer la orquestación de servicios, tal componente se encuentra entre la sesión de control de la IMS y los servidores de aplicación. Una de las empresas que ha explorado esta solución es *AlcatelLucent* [20], la arquitectura de esta solución se muestra en la figura 4.2 la cual coloca el SCIM en contacto con los servidores de aplicación SIP y el S-CSCF. Al llegar un mensaje SIP proveniente del CSCF el SCIM lo interpreta y activa los servidores de aplicación que considere necesarios, los diseñadores proveen una API en java para desarrollar la lógica de la aplicación.

A pesar de que las especificaciones de IMS indican el SCIM es el elemento encargado de la orquestación se reconoce que su descripción funcional es bastante limitada por lo

que los diseñadores de OSTP decidieron implementar las capacidades de orquestación sobre un bus de servicios cuya funcionalidad es bastante conocida y evolucionada. Adicionalmente este enfoque tiene la desventaja de que el desarrollador de aplicaciones debe tener conocimientos del funcionamiento de la mensajería SIP y de Java para desarrollar aplicaciones lo cual va contra los objetivos de OSTP.

Capítulo 5

Implementación

En este capítulo se decribe la arquitectura de OSTP. La descripción se divide en dos partes: La primera es la visión por niveles donde se sitúa OSTP y la segunda es la discusión de la arquitectura funcional y sus elementos.

5.1. Vista de niveles

La figura 5.1 muestra la ubicación de OSTP dentro de una infraestructura telco, se pueden apreciar 4 niveles con responsabilidades distintas:

- El nivel de equipamiento del cliente hace referencia a aquellos dispositivos por medio de los cuales los usuarios finales acceden a las redes de comunicaciones.
- El nivel de control y conectividad hace referencia a aquella infraestructura con la cual el operador de comunicaciones le provee la conectividad al usuario final, dentro de este se encuentran los flujos de datos del usuario (voz, video, sms, etc) y de señalización donde la ultima se emplea para coordinar y controlar el primero. Dentro de este nivel se hacen dos distinciones: las tecnologías NGN e IMS la cuales se distinguen por llevar el tráfico mediante IP y las tecnologías legado las cuales son previas a las primeras y comprenden diferentes tecnologías como la PSTN, las redes móviles tales como UMTS, GSM, etc.
- El nivel de los Habilitadores (Enablers) los cuales se encargan de abstraer la funcionalidades de las redes de comunicación que emplean sus protocolos particulares (tales como SS7 para la PSTN y SIP para las NGN/IMS) y traducen estos protocolos a lenguajes compatibles con la tecnología de WebServices tales como WSDL/SOAP los cuales pueden usarse desde varias plataformas de programación.
- El nivel de lógica de servicio consta de las aplicaciones finales que hacen uso de las capacidades de las redes de las telco, estas aplicaciones pueden ser ofrecidas por la misma telco o por terceros.

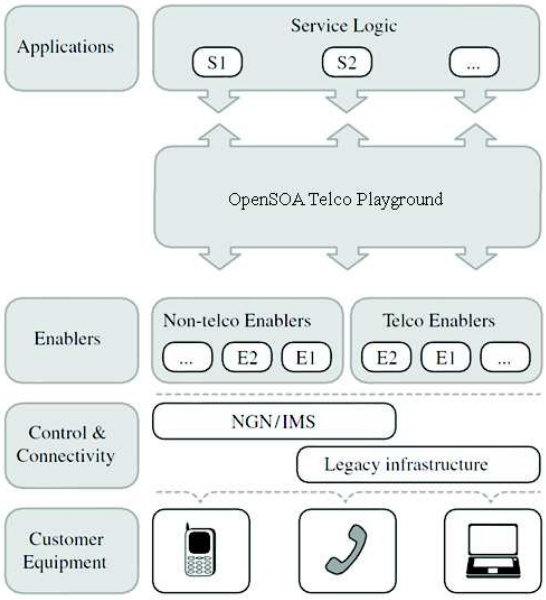

Figura 5.1: Vista de niveles la cual ubica OSTP dentro de la infraestructura de una telco, imagen adaptada de [17]

Teniendo en cuenta esta distribución de niveles, OSTP se sitúa entre los niveles de lógica de servicio y habilitadores lo cual significa que este busca enlazar a los consumidores de un servicio con los proveedores pero hay que recordar que por el curso de este enlace se aplican varios procesos asociados a la prestación del servicio tales como la autenticación, la autorización y la composición.

5.2. Arquitectura funcional de OpenSOATelcoPlayground

La arquitectura funcional de OSTP es mostrada en la figura 5.2. OSTP consta de un ESB como elemento principal de integración. Vale la pena recordar que OSTP busca ser una arquitectura compatible con el estándar OSE, donde los niveles estudiados en la sección 4.1.2 se mapean directamente a un componente de esta visión funcional. Los componentes de la izquierda son los referidos por OSE y los de la derecha son los implementados en OSTP:

- Policy enforcer - PEEM
- Applications - 3rd Party service providers / Web 2.0 mashups
- Enabler implementation - Enabler exposure API

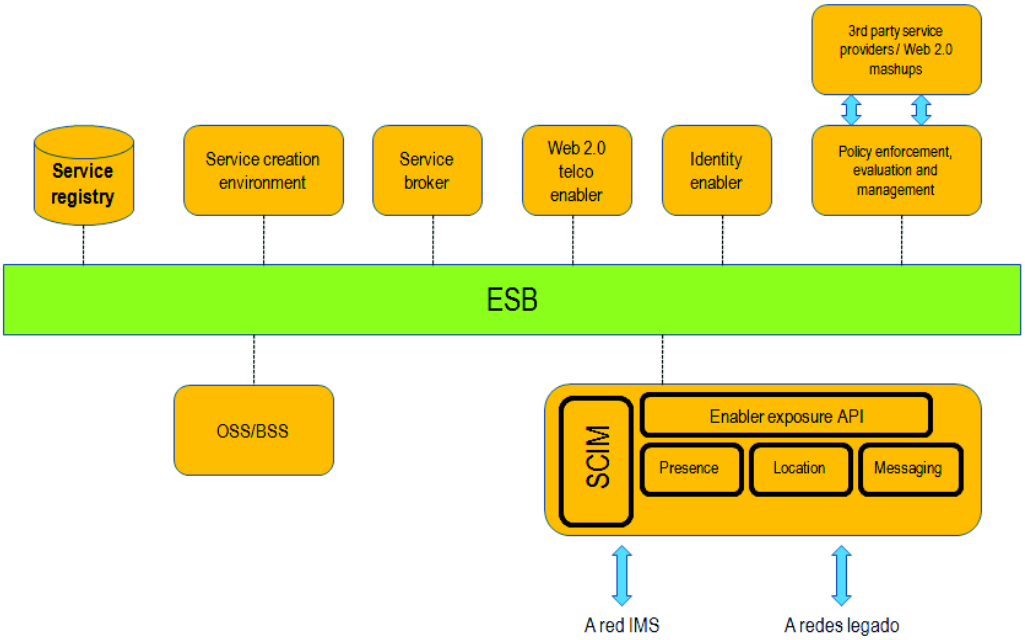

Figura 5.2: Descomposición de componentes de OSTP, imagen basada en [15].

5.2.1. Aplicaciones de terceros

Estos componentes corresponden a los consumidores finales de las aplicaciones desarrolladas usando los enablers provistos. Cabe anotar que dichos componentes pueden encontrarse en una ubicación física diferente a la de la infraestructura de la telco.

5.2.2. Policy Evaluation Enforcement and management

Este componente corresponde al nivel *Policy enforcer* especificado en el OSE y se puede considerar como el punto de entrada de todas las solicitudes de servicio donde a estas se les evalúan diferentes parámetros siendo los principales la autenticación y la autorización. Tambien el servicio se ajusta a las preferencias del usuario como por ejemplo los aspectos de privacidad y se aplican las reglas de cobro. OSTP cuenta con una implementación del policy enforcer de OSE conocida como OpenPEEM [12].

El modo de funcionamiento del PEEM se ilustra en la figura 5.3 y procede de la siguiente manera:

- Interceptar una solicitud entrante.
- Aplicarle una transformación al formato de datos de la solicitud a un formato que el motor de políticas maneje en caso de que sea necesario.

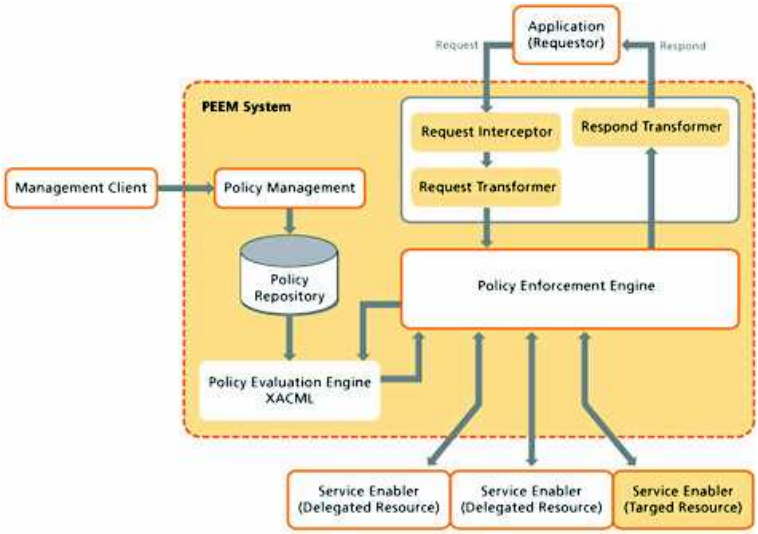

Figura 5.3: Flujo de datos dentro de el Policy enforcement and management, imagen [12].

- Se obtienen las políticas pertinentes del repositorio de políticas.
- Se evalúa la solicitud de acuerdo a las políticas obtenidas. Los resultados más sencillos de esta acción son que la solicitud se niege o se apruebe, aunque pueden existir resultados más complejos.
- El PEEM le pasa la solicitud al broker(No mostrado) el cual como resultado final invoca el habilitador del servicio.

El motor de evaluación de políticas se considera el elemento central del PEEM e implementa operaciones tales como la adición, modificación y eliminación de políticas. Este motor de políticas emplea como protocolo subyacente el protocolo *XCAP XML configuration access protocol* y usa el modelo PEP/PDP de la IETF [12].

5.2.3. Broker de servicios

El broker de servicios se encarga de proveer la orquestación de componentes en SOA y de gestionar el ciclo de vida de un servicio. Su función principal es interactuar con todos los componentes conectados al ESB y tiene las siguientes características:

- Soporta el descubrimiento de servicios mediante el protocolo UDDI.
- Orquesta los habilitadores que representan la abstracción de las capacidades de la red como parte de la ejecución de un servicio completo.
- Los servicios son ofrecidos a través de interfaces que están conformes a la especificación OMA.

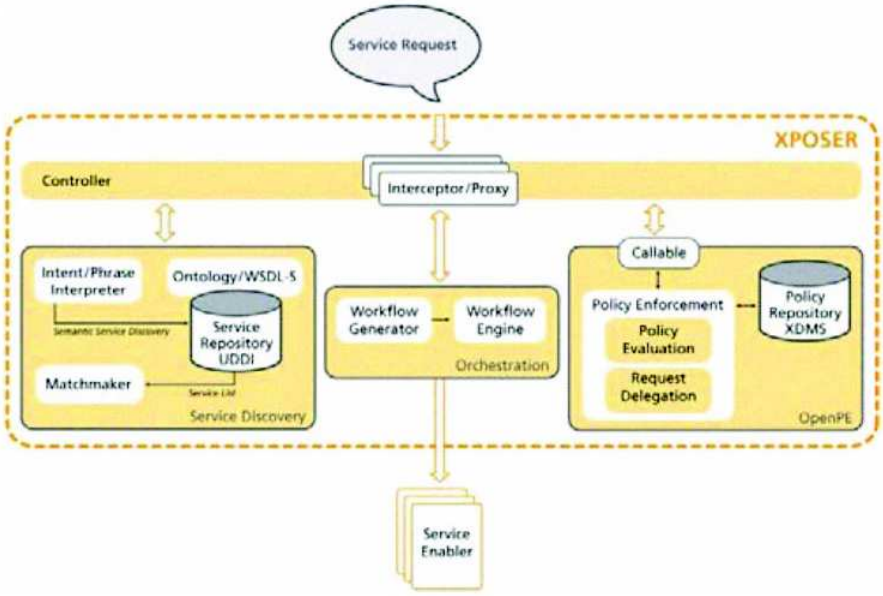

Figura 5.4: Arquitectura del broker de servicios, imagen [16].

Dentro de OSTP se incluye una implementación del service broker conocida como el *XPOSER: Extended policy based semantically enabled service broker* el cual consta de los siguientes componentes [16]:

- OpenPE: Ejecuta tareas de evaluación y aplicación de políticas.
- Un servicio de registro semánticamente enriquecido para descubrimiento de servicios mejorado, basado en el intercambio de WSDLs.
- Un motor de gestión de procesos de negocio basado en activeBPEL para realizar composición de servicios.

5.2.4. Registro de servicio

El registro de servicios es el componente que posee la información de los servicios tal como sus ubicaciones e interfaces. En caso de que el broker necesite localizar un determinado servicio le hace una consulta al registro de servicios empleando el protocolo UDDI y con base al resultado de la consulta procede a invocarlo. Dentro de la sección anterior se comentó que este registro de servicio es semánticamente enriquecido debido a que se busca que los servicios puedan ser compuestos y descubiertos de la manera más automatizada posible. Debido a que WDSL no provee la sufiente información para poder lograr este fin los diseñadores de OSTP emplean mecanismos de ontologías para poder facilitar el descubrimiento de los servicios [16].

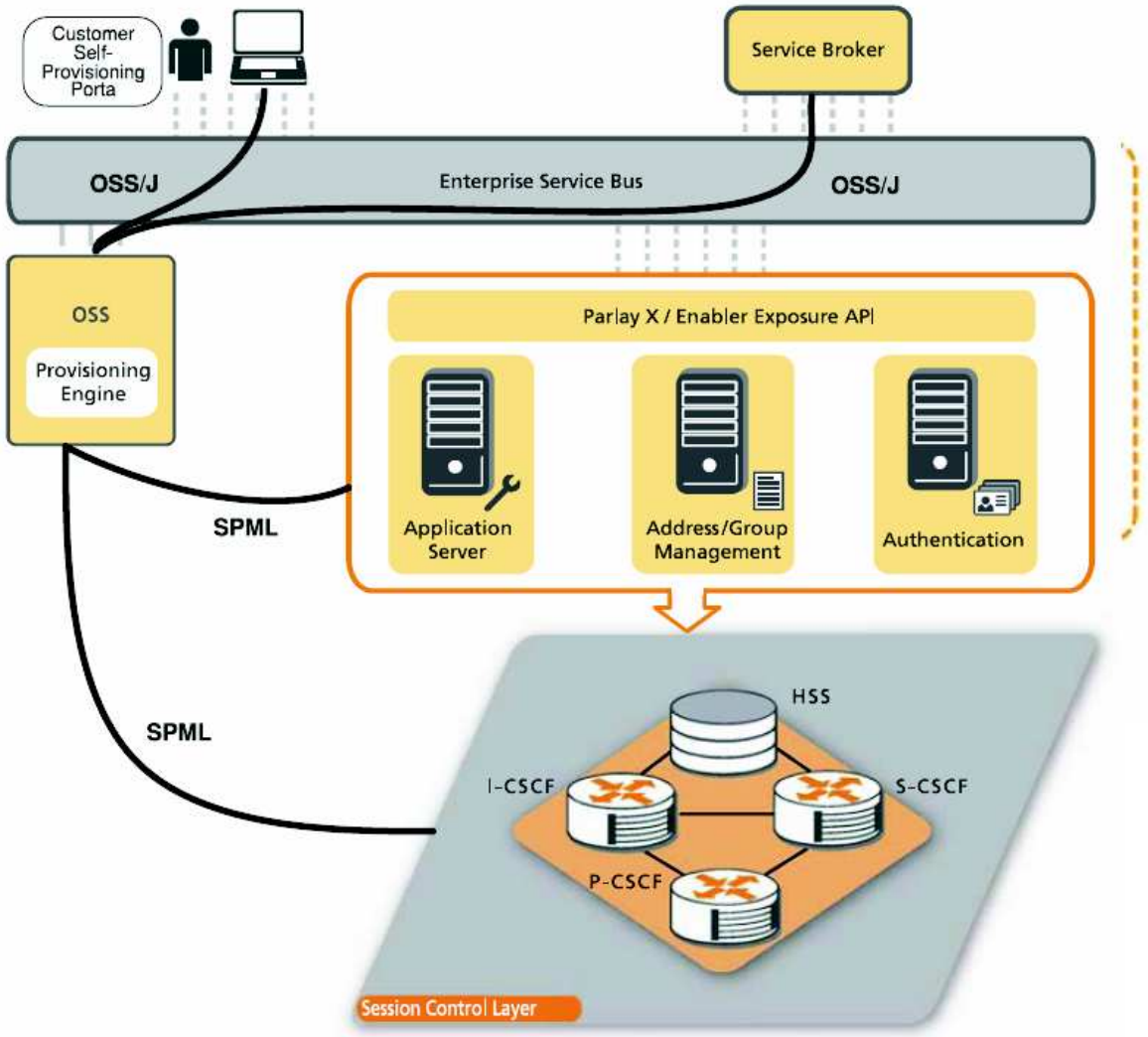

Figura 5.5: Ejemplo de autoservicio haciéndo uso de un portal web, imagen [15].

5.2.5. OSS/BSS

La introducción de tecnologías que favorecen la convergencia como IMS llevan a que la complejidad de sus redes aumente lo cual dispara los costos de gestión si estos no son controlados, además aumenta la base de dispositivos finales de usuario que deben ser soportados. Una de las formas de fomentar la reducción en los costos es siguiendo el principio de SOA de eliminar las islas de servicios y procurar que todos los sistemas puedan comunicarse entre sí.

Los sistemas BSS y OSS (BOSS) están estrechamente relacionados con los servicios de telecomunicaciones debido a que estos se apoyan sobre los procesos de aprovisionamiento y aseguramiento del servicio. Una telco puede tener varias decenas de sistemas

BOSS y la idea de SOA es de colocarles interfaces orientadas a servicios con el fin de poder implantar mejoras en ambos procesos y de esta manera lograr la formación de la infraestructura BOSS *plug and play* basada en servicios web.

Una de las tecnologías conformes al estándar de NGOSS es *OSS through Java initiative* (OSS/J) [15] el cual es un conjunto de APIs basadas en WebServices y sobre la cual se basa la implementación del NGOSS de OSTP. OSS/J se considera el vehículo de harmonización de los BOSS con el cual se realiza la integración.

El aprovechamiento de las capacidades de orquestación e intermediación (brokerage) de OSTP resulta de gran utilidad debido a que permite integrar los procesos de servicio, procesos de gestión y procesos de negocio, adicionalmente la orquestación permite que los servicios ofrecidos por los BOSS sean tenidas en cuenta desde el momento en que se crea un servicio.

Un posible escenario que favorece la integración con los sistemas BOSS es un portal de autoservicio donde el usuario puede seleccionar nuevos servicios para usar. Al suscribirse a un nuevo servicio mediante el portal conectado al ESB, el broker inicia el motor de aprovisionamiento con lo cual se configuran todos los elementos de la red sobre los cuales se basa el servicio como por ejemplo el HSS de la IMS. Todo lo anterior se realiza sin intervención humana, la figura 5.5 ilustra este proceso.

Por el lado del aseguramiento del servicio este debe hacerse tanto en la capa SDP (los componentes SOA) como en la capa de control de sesión de la IMS. La integración de los componentes BOSS favorece la existencia de sistemas de configuración y resolución de fallas automatizados en donde se aplican estrategias de auto reparación mediante los cuales se localizan los componentes que producen la falla y se reemplazan por otros que cumplan la misma función o el componente fallido es reiniciado.

5.2.6. Service creation environment

OSTP provee las facilidades necesarias para crear un servicio y ponerlo en producción, todo en el marco de orientación al cliente y rápido *time to market*[17]. El proceso de desarrollo busca enlazar el pensamiento del personal técnico con las perspectivas de la gerencia y el usuario. Se considera que entre más facil sea de crear un servicio aumenta el número de servicios disponibles y más grande será la aceptación de la plataforma.

El ambiente de creación de servicios sigue el paradigma de desarrollo orientado por modelos y hace uso de la técnica *Xtreme model driven design* (XMDD) que a su vez se guía por el proceso *One thing approach*. En esta técnica los clientes diseñan, animan, validan y controlan el desarrollo a traves del ciclo de vida del proyecto

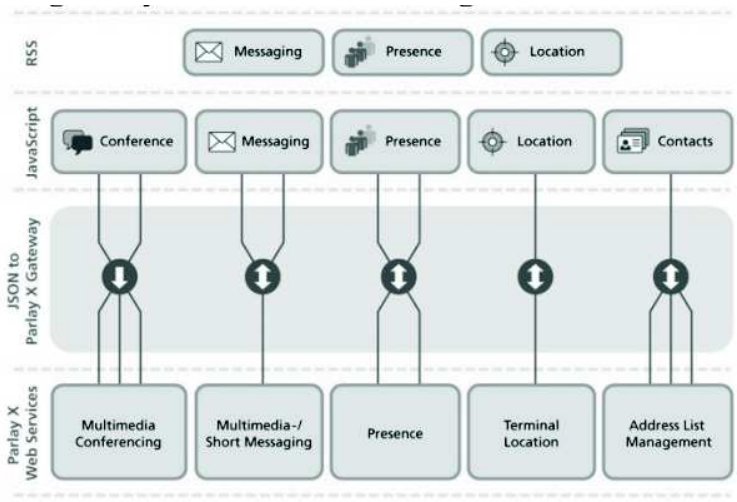

Figura 5.6: Comunicación entre cliente y servidor haciendo uso de JSON-RPC, imagen [14].

OSTP hace uso de la implementación de XMDD conocida como jABC en donde cada bloque funcional de la red telco como un bloque ParlayX por ejemplo se modela como un *Service Independent Building Block*, estos bloques se unen en grafo de nivel de servicio para lograr el servicio compuesto y posteriormente se genera el código de manera automática. Una vez diseñado el servicio se publica su WSDL en el registro de servicio. Las condiciones de prestación del servicio se manejan por aparte y se expresan como políticas del PEEM durante el tiempo de creación y abarcan aspectos como los mecanismos de exposición y las políticas de acceso.

5.2.7. Web 2.0 telco enabler

También conocido como Anubis Gate [5], este componente se basa en una API en JavaScript/AJAX que le permite a los desarrolladores invocar las funcionalidades de telecomunicaciones que el *mash-up* requiera desde un browser. Estas funcionalidades incluyen aquellas especificadas en ParlayX y otras adicionales. La razón de la existencia de esta API es el reconocimiento de JavaScript/AJAX como uno de los lenguajes más difundidos para la construcción de aplicaciones Web 2.0 el cual permite controlar la presentación e interactividad de una página web.

Debido a que la mayoría de los browsers modernos solo permiten que se invoquen servicios web locales, las solicitudes a servicios web remotos deben dirigirse a través de un *proxy*. Para implementar la comunicación cliente-servidor se hace uso de la tecnología JSON-RPC la cual se comunica con el extremo del servidor el cual implementa ParlayX [14], esta comunicación se detalla en la figura 5.6.

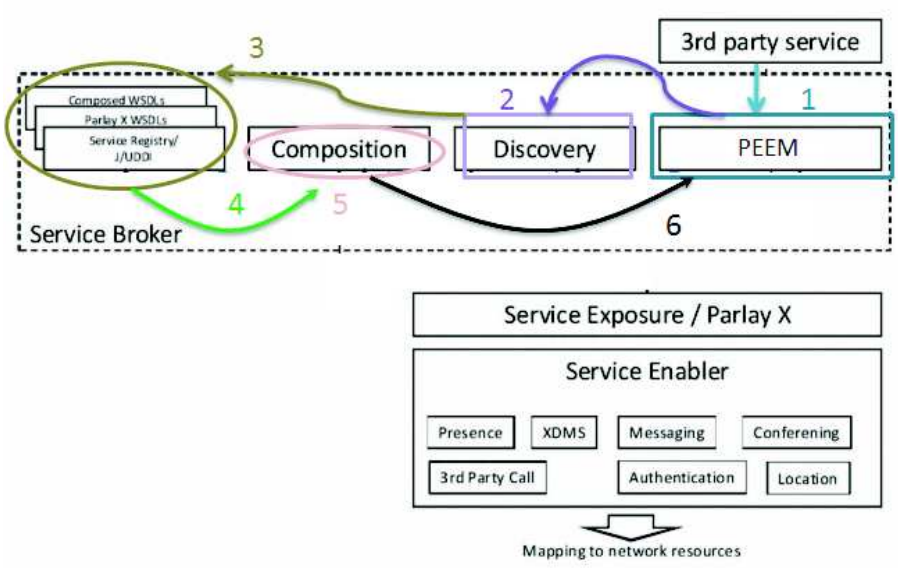

Figura 5.7: Seguimiento de la ejecución de la solicitud de un servicio, imagen basada en [14] .

5.2.8. Seguimiento de la solicitud de un servicio en OSTP

Hasta el momento se ha mostrado la funcionalidad de los distintos bloques funcionales de OSTP en esta sección se quiere mostrar cómo estos funcionan en conjunto para atender la solicitud de un servicio, el flujo se muestra en la figura 5.7 [16]:

1. Se hace la autenticación.
2. Se le delega la solicitud al service discovery.
3. Se descompone la solicitud dependiendo de los enablers diponibles según lo indique el WSDL de cada implementación de enabler .
4. Se le transfiere la solicitud al compositor de servicios.
5. Se compone el workflow del servicio.
6. Se ejecuta el workflow del servicio.

Capítulo 6

Validación

6.1. Aplicación de OSTP en la IPTV

La tecnología de televisión sobre IP se considera un competidor interesante frente a otras tecnologías de presentación de televisión como el satélite, el cable y la televisión terrestre debido a que esta ofrece la posibilidad de un canal de retorno que fluye en dirección contraria a la prestación del servicio es decir el televidente le transmite información al distribuidor de contenido, los escenarios mostrados en este capítulo hacen referencia a los mostrados en el artículo [13].

La introducción del canal de retorno implica la introducción de nuevos servicios que mejoran la experiencia del usuario los cuales se caracterizan principalmente por ser personalizados e interactivos. La personalización se traduce en poder estudiar el contenido que consume el televidente y con base a este se le pueden formular recomendaciones de contenido y con la interactividad se pueden introducir diferentes funcionalidades tales como los cuestionarios interactivos.Los autores de OSTP al ser fuertes defensores de IMS proponen que la entrega de la IPTV se realice sobre esta misma plataforma a diferencia de otras alternativas como la entrega de los contenidos IPTV directamente sobre internet abierta.

Teniendo en cuenta los principios básicos de OSTP en el campo de IPTV se buscan componentes básicos de funcionalidad que puedan representarse como habilitadores que a su vez puedan ser orquestados con el fin de crear aplicaciones de mayor complejidad.

6.1.1. Bloques básicos de funcionalidad

Dentro del diseño de una solución IPTV el principal componente es el flujo (stream) el cual tiene un origen que lo genera y un consumidor el cual representa el destino final.

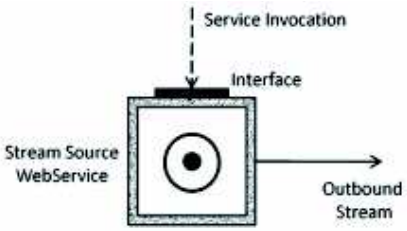

Figura 6.1: Convención empleada para expresar un habilitador con una envolvente de webService, imagen [13].

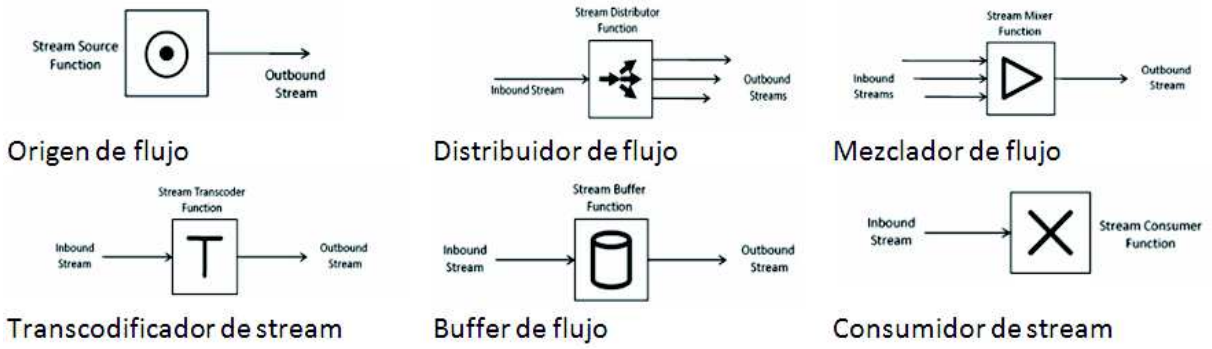

Figura 6.2: Símbolos de representación de los habilitadores relacionados con IPTV, imagen basada en [13].

Un flujo puede contener varios subflujos que pueden mostrarse a voluntad del usuario final como por ejemplo los subtítulos o un lenguaje alternativo. También es posible mezclar varios subfujos e incorporarlos dentro de un solo flujo y no darle la alternativa al usuario de desactivarlos, tal es el caso de la publicidad que financia un programa.

Cada bloque básico de funcionalidad posee entradas y salidas para uno o varios flujos y también posee una interfaz de control mediante la cual se modifica el comportamiento del bloque (figura 6.2). La funcionalidad de la interfaz del bloque está representada por un contrato de tipo WSDL que describe las operaciones soportadas por cada bloque y sobre las cuales se puede realizar una orquestación. Los bloques empleados en una transmisión IPTV se muestran en la figura 6.2 y se describen en la tabla 6.1.

Componente	**Descripción**	**Operaciones**
Origen de flujo	Punto de partida de la transmisión	▪ Seleccionar contenido a transmitir. ▪ Seleccionar codificación del contenido. ▪ Seleccionar eaquema de cifrado y llaves .
Distribuidor de flujo	Determina la forma en que se entrega el contenido (unicast, multicast IP o multicast a nivel de aplicación)	▪ Adicionar receptor. ▪ Eliminar receptor.
Mezclador de flujo	Se componen una o más streams a partir de varias streams de origen	▪ Seleccionar substreams a mezclar. ▪ Seleccionar modo de mezcla (Reemplazar, adicionar, re -codificar, etc).

Transcodificador de stream	Modifica el formato de un stream (Aplicar cifrado o modificar tasa de muestreo) entre otras aplicaciones posibles	▪ Seleccionar stream. ▪ Seleccionar codec de salida o método de cifrado.
Buffer de flujo	Se controla la salida de un flujo pero el contenido de este no se modifica	▪ Detener: No se muestra nada en la salida pero el flujo que llega se almacena. ▪ Continuar: Se reproduce en la salida desde la parte más antigua que llegó al buffer. ▪ Adelantar: Se reproduce en la salida un momento determinado del buffer sin ser el más antiguo.
Consumidor de stream	Punto de terminación del flujo	No tiene operaciones asociadas

Tabla 6.1: Habilitadores empleados en la entrega de un servicio IPTV

6.1.2. Ejemplo de un evento de televisión con servicios premium y regular

Con base a los bloques de funcionalidad presentada arriba un operador de televisión busca ofrecer una transmisión de televisión de dos tipos:

- Premium: El usuario tiene la opción de saltarse los comerciales debido a esto paga un importe mayor al que paga el usuario regular, adicionalmente tiene prioridad a la hora de comprar artículos relacionados con el programa.

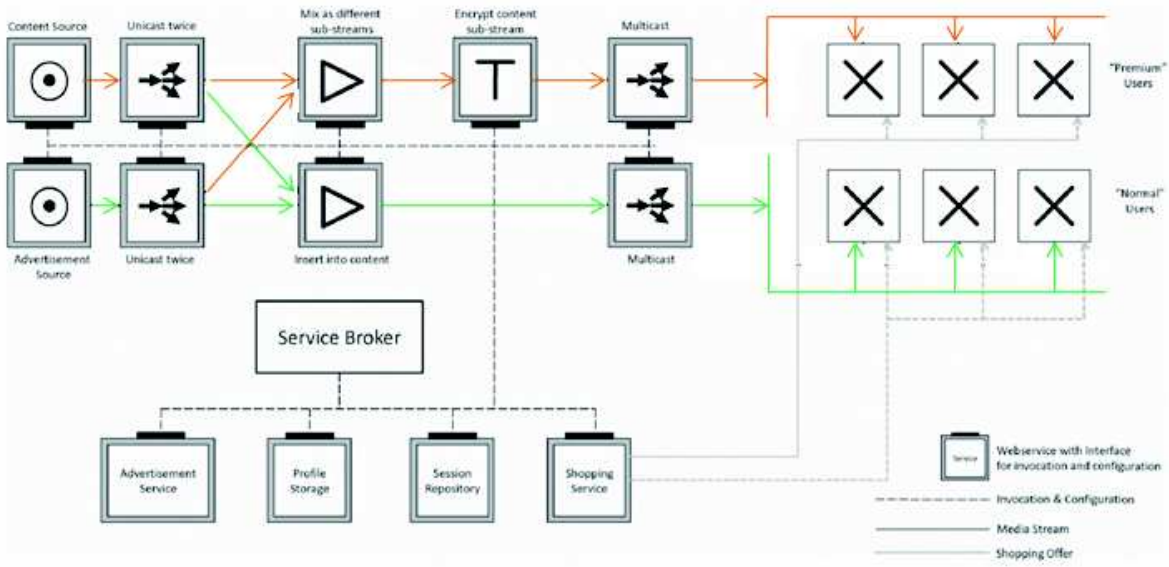

Figura 6.3: Ejemplo de prestación de un servicio IPTV haciendo uso de OSTP, en verde se muestra el flujo destinado a los usuarios regulares y en naranja el flujo destinado a los usuarios premium, imagen basada en [13].

- Regular: El usuario paga menos en comparación que la suscripción premium pero debe tener que ver los comerciales a cambio.

La composición del servicio se muestra en la figura 6.3 en donde las fuentes existentes son el evento televisado y los comerciales. Para el usuario regular los comerciales se mezclan formando un stream único debido a la obligatoriedad de estos, en cambio para el usuario premium estos se incluyen como un substream lo cual le permite desactivarlos a su voluntad. El flujo de los usuarios premium se cifra con el fin de evitar intentos de sabotaje por parte de usuarios no autorizados.

En la parte de abajo de la figura se muestran los bloques que no son de flujo pero que soportan la entrega de servicios: el Broker de servicios debe coordinar a todos los bloques funcionales, el almacenamiento del perfil almacena el tipo de suscripción del usuario, el servicio de publicidad busca soportar la entrega de publicidad personalizada y el servicio de sesión busca almacenar los usuarios que han almacenado cada servicio.

6.1.3. Otros escenarios de IPTV sobre SOA

- Compartir contenido: Un usuario puede tomar contenido de su autoría y compartirlo con su lista de amigos y crear una sala de conversación con base a este contenido con lo cual se requieren servicios básicos que manejen la subida del contenido, la búsqueda de la lista de amigos y la creación de la sala de conversación.
- Cuestionarios interactivos: Se le muestran al usuario preguntas relacionadas con algún tema (series de televisión, películas, etc) y el usuario las responde y con base a esta información se puede desplegar publicidad o conocer más del usuario.

Capítulo 7

Conclusiones

Las arquitecturas orientadas a servicios son un enfoque de sistemas distribuidos el cual tiene como principal objetivo integrar sistemas heterogéneos ofreciendo beneficios de flexibilidad, bajo acoplamiento e interoperabilidad. Por otro lado muchas empresas de comunicación se enfrentan a una situación de saturación de ingresos por cuenta de los servicios de voz y una de las maneras de aumentar los ingresos es el fomentar un ambiente que ofrezca servicios de valor agregado donde estos servicios muchas veces son desarrollados por terceras partes e incluyen funcionalidades propias de la telco.

Para poder materializar la visión del *jardín abierto* donde muchos desarrolladores incluyen dentro de sus aplicaciones servicios de las telco debe ser posible que esta exponga sus servicios de manera controlada lo cual implica que se requiere la aplicación de políticas. Otro requerimiento clave es que desarrolladores no tienen que conocer los detalles de señalización de las redes de comunicaciones si no que se les entrega una abstracción de las capacidades de la red basada en Web services y JavaScript/AJAX.

La arquitectura funcional de OSTP se centra en un bus de servicios empresariales el cual integra componentes débilmente acoplados. Esta arquitectura adicionalmente busca ser conforme con los estándares OSE, TMF eTom y TMF NGOSS.

Bibliografía

[1] Open Mobile Alliance. The oma perspective on soa in telecoms. http://www.openmobilealliance.org/document/OpenStandardsForum2008_Unmehopa_OMA_final.pdf.

[2] Y. D. Wong, C. Ting. Network management to service management - a challenge to telecom service providers. In *Conference on Innovative Computing, Information and Control*, number 1, pages 280–280, 2007.

[3] Active endpoints. Activebpel. http://www.activevos.com/community-open-source.php.

[4] Chakraborty et al. On the changing face of service composition in telecom. In *Proceedings of the 2007 Workshop on Middleware for next-generation converged networks and applications*, number 1, 2007.

[5] FOKUS. Anubis gate. http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_soa_telco_playground/software/java_script_telco_api/index.html.

[6] Ian Gorton. *Essential software architecture*. Springer, 2006.

[7] Nicolai Josuttis. *SOA in practice, the art of distributed system design*. O'Reilly Media, 2007.

[8] Telecommunication management forum. Business process framework (etom). http://www.tmforum.org/BusinessProcessFramework/1647/home.html.

[9] Telecommunication management forum. Solution frameworks (ngoss). http://www.tmforum.org/BestPracticesStandards/SolutionFrameworks/8428/Home.html.

[10] Open mobile alliance. Open service environment. http://www.openmobilealliance.org/Technical/release_program/ose_v1_0.aspx, October 2009.

[11] Oracle. A hands on introduction to bpel. http://www.oracle.com/technology/pub/articles/matjaz_bpel1.html.

[12] Y. Gerlat S. Dutkowski, Magedanz. The openpeem as core for service orchestration within the open ims playground at fokus. In *International Conference on Intelligence in Networks (ICIN 2007)–Emerging Web and Telecom Services: Collision or Coopetition?*, number 1, 2007.

[13] C. Riede T. Magedanz, J. Fiedler. Converged media and communication services, sdp for soa-based rich media service mashups. In *First conference on IT revlolutions*, number 1, pages 1–7, 2008.

[14] F. Schreiner T. Magedanz, N. Blum. Services, enablers and architectures: Definition of a connected web 2.0 / telco service broker to enable new flexible service exposure models. In *Proc. of International Conference on Intelligence in Networks (ICIN)*, number 1, 2008.

[15] F Schreiner T. Magedanz, N. Blum and S. Wahle. Form ims managenent to soa based ngn management. *Journal of network and systems management*, 40(11):33–52, 2009.

[16] F. Schreiner S. Wahle T. Magedanz, N. Blum. A researh infrastructure for soa-based service delivery frameworks, the open soa telco playground at fraunhoufer fokus. In *International conference on testbeds and research infrastructures for the development of networks and communities and workshops*, number 5, pages 1–6, 2009.

[17] I. Boldea T. Magedanz, N. Blum and Tiziana Margaria. Service-oriented access to next generation networks from service creation to execution. *Mobile Networks and Applications*.

[18] N. Blum T. Magedanz and S. Dutkowski. Evolution of soa concepts in telecommunications. *IEEE Computer Magazine*, 40(11):46–50, 2007.

[19] TIBCO. El bus de servicios empresariales. `http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.08.Articulo.El%20papel%20de%20un%20bus%20de%20servicios%20empresariales%20(ESB)%20en%20una%20SOA.pdf`.

[20] K. Kocan V. S. Abhayawardhana and J. Matthews. Service blending for converged ims services - bt alcatel-lucent collaboration. *BT technology Journal*, 25(2):211–218, 2007.

[21] W3C. Web services architecture. `http://www.w3.org/TR/ws-arch/`.