



Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROYECTO FINAL DE CARRERA

ESTUDIO DE ARQUITECTURAS DE REDES ORIENTADAS A SERVICIO

(Study of Service Oriented Network
Architectures)

Estudios: Ingeniería de Telecomunicaciones

Autor: María González Quiroga

Director/a: Jaume Comellas Colomé

Año: 2011

Índice

Agradecimientos	5
Resumen	6
Resum	7
Abstract.....	8
1. Acrónimos.....	9
2. Introducción	14
2.1. Breve historia de la arquitectura de software	14
2.2. Objetivos.....	20
3. SOA: Arquitectura Orientada a Servicio	21
3.1. La aparición de SOA	21
3.2. Servicio.....	23
3.2.1. Orientación a Objetos y Desarrollo Basado en Componentes	24
3.2.2. Arquitectura de Servicios	25
3.3. Definición de SOA.....	29
3.3.1. Taxonomías de SOA.....	30
3.3.2. Requerimientos	32
3.3.3. Enfoques para crear una SOA	45
3.4. SOA y los Web Services.....	46
3.5. Beneficios de desplegar SOA.....	48
3.6. Implementando SOA	51
3.6.1. Componentes de SOA	51
3.6.2. Arquitectura SOA y tecnologías utilizadas	51
3.6.3. SOA en redes ópticas	78
3.7. Ventajas e inconvenientes de implementar SOA.....	98

3.7.1. Ventajas	98
3.7.2. Inconvenientes.....	99
4. Ejemplos de aplicación de SOA.....	100
4.1. EWA.....	100
4.2. Diseño de ASMF (Autonomic Service Management Framework)	105
4.2.1. ASMF	105
4.2.2. Red de Servicios Web.....	106
4.2.3. Estructura de la aplicación	108
4.2.4. Servicio distribuido de localización y composición	110
4.3. Arquitectura BT.....	115
4.3.1. Principios de la arquitectura.....	116
4.3.2. Vista del dominio de arquitectura	117
4.3.3. La red	119
4.4. ESNET	135
4.4.1. La misión de ESnet.....	135
4.4.2. ESNET	135
4.4.3. Localización de la ESnet en EE.UU. y la Ciencia Internacional	136
4.4.4. Requisitos de la red de datos y características de colaboración.....	137
4.4.5. Resumen de los requisitos de la red.....	140
4.4.6. Permitir la ciencia futura: Evolución de ESnet durante los 10 años próximos.....	141
4.4.7. Desarrollo y despliegue de los servicios de comunicación orientados a servicio.....	147
4.4.8. El papel crítico de la monitorización y reporting.....	149
4.4.9. Nuevos servicios de monitorización	152
5. Conclusiones	161
6. Referencias.....	165

Agradecimientos

A mi tutor de proyecto Jaume Comellas por la paciencia, la comprensión y la ayuda que me ha brindado durante este tiempo.

A mis amigos de toda la vida, tanto los de Barcelona (sobre todo a ti Vicky) como los de San Clodio, por estar ahí cuando os he necesitado y por haberme dado tantos días felices a lo largo de mi vida.

Agradecer a mis compañeros de carrera todos los momentos de alegría, de tristeza, de risas y de nervios que hemos compartido juntos, en especial a Ana que ha sido mi gran apoyo en todos los aspectos durante estos años.

A Carmen, Carlos, Paco y Carmen por ser tan buenos conmigo, por estar ahí para nosotros en todo momento, y por la alegría y felicidad que aportan a nuestras vidas.

A toda mi familia, especialmente a mis primos (Víctor y Jeni), a mis abuelos (David y Arturo), seguro que os haría mucha ilusión poder verlo...y sobre todo a mi abuela Clara por todo el cariño que siempre me ha demostrado.

A mi novio, Carlos, he de dar gracias a aquella clase de Fonaments en la que nos conocimos...Gracias por darme tu apoyo y comprensión, gracias por tus palabras de ánimo cuando las he necesitado y por todos los momentos de felicidad que has aportado y aportas a mi vida. Gracias también por el amor y el cariño que me demuestras cada día. Doy gracias por la suerte de tenerte a mi lado, 631!

Y especialmente a mis padres, Arturo y Carmen, nunca podré agradecerlos todo lo que habéis hecho por mí. Soy lo que soy gracias a vuestros esfuerzos y sacrificios. Gracias por vuestra confianza en mí, por el cariño y el amor con el que he podido crecer y por hacerme la hija más afortunada del mundo.

Resumen

El presente proyecto es un estudio sobre las arquitecturas de redes orientadas a servicio. Este documento pretende explicar cuándo aparecen las SOA, el porqué del éxito de su implantación en las empresas y cuáles son sus beneficios, así como el concepto de servicio, el cual es imprescindible para llevar a cabo un correcto diseño de este tipo de arquitecturas. Para ello, también se explica cómo se estructuran las SOA, los componentes que las forman y cuáles son las principales tecnologías que intervienen en su diseño. Web Services y BPM son dos de las tecnologías clave para poder desplegar SOA y las más utilizadas y aceptadas por la industria.

Finalmente, en este documento se analizan, también, cuatro ejemplos de arquitecturas que se basan en la orientación a servicio: EWA de IBM, ASMF, la arquitectura BT y ESnet.

Resum

El present projecte és un estudi sobre les arquitectures de xarxes orientades a servei. Aquest document pretén explicar quan apareixen les SOA, el perquè de l'èxit de la seva implantació en les empreses i quins són els seus beneficis; així com el concepte de servei, el qual és imprescindible per dur a terme un correcte disseny d'aquest tipus d'arquitectures. Per això, també s'explica com s'estructuren les SOA, els components que les formen i quines són les principals tecnologies que intervenen en el seu disseny. Web Services i BPM són dues de les tecnologies clau per poder desplegar SOA, i les més utilitzades i acceptades per la indústria.

Finalment, en aquest document s'analitzen, també, quatre exemples d'arquitectures que es basen en l'orientació a servei: EWA d'IBM, ASMF, l'arquitectura i ESnet.

Abstract

This project studies the service oriented network architectures. This document aims to explain when SOA's appear, the reason for the success of their implementation in companies and what are their benefits as well as the concept of service, which is essential for the proper design of such architectures. It also explains how SOA is structured, the components that form it and what are the main technologies involved in its design. Web Services and BPM are two of the key technologies for SOA deployment and the most used and accepted by the industry.

Finally, this document also analyzes four examples of architectures that are based on service-orientation: EWA IBM, ASMF, BT and ESnet architecture.

1. Acrónimos

3GPP	3rd Generation Partnership Project
AAAS	Authentication, Authorization, and Auditing Subsystem
AAEF	Autonomic Application Enabling Fabric
ABC	Autonomic Business Component
AC	Admission Control
AM	Autonomic Manager
AON	Application Oriented Network
AS	Advertising Service
AuS	Authentication Service
ASB	Autonomic Service Broker
ASMF	Autonomic Service Management Framework
ATM	Asynchronous Transfer Mode
BAM	Business Activity Monitoring
BGP	Border Gateway Protocol
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BPMS	Business Process Management Suite
C/S	Cliente/Servidor
CANARIE	Canada's Advanced Research and Innovation Network
CBDI	Component Based Development and Integration
CCAMP	Common Control And Measurement Plane
CE	Customer Edge
CLI	Calling Line Identity
CNLS	Connectionless Network Service
COCS	Connection Oriented Circuit Service
COPS	Connection Oriented Packet Service
CORBA	Common Object Request Broker Architecture
CPI	Customer Port Identifier
CSCF	Call Session Control Function
CSPF	Constrained Shortest Path First
DCE	Distributed Computation Environment
DCOM	Distributed Component Object Model
DIFFSERV	Differentiated Services
DOE	Department Of Energy
DWDM	Dense Wave Division Multiplexing
E2E	End to End

E2ECS	End to End Connection Service
E2ECU	End to End Coordination Unit
E2ENS	End to End Negotiation Service
EAI	Enterprise Application Integration
ENNI	External Network to Network Interface
ESB	Enterprise Service Bus
ESNET	Energy Sciencies Network
ETL	Extract, Transform and Load
EWA	Enterprise Workframe Architecture
FM	Fault Management
FR	Frame Relay
FTP	File Transfer Protocol
GMPLS	General Multiprotocol Label Switching
GVPN	GMPLS VPN
HSS	Home Subscriber Subsystem
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISBN	International Standard Book Number
ISP	Internet Service Provider
IT	Information Technology
ITER	International Thermonuclear Experimental Reactor
J2EE	Java 2 Platform Enterprise Edition
JAVA RMI	Java Remote Method Invocation
JMS	Java Message Service
KPI	Key Performance Indicator
L1VPN	Layer 1 VPN
LAN	Local Area Network
LHC	Long Hadron Collider
LS	Look up Service
LSP	Label Switched Path
MA	Measurament Archieve
MAN	Metropolitan Area Network
MM	Memebership Manager
MOWS	Management Of Web Services
MP	Measurament Point
MPLS	Multiprotocol Label Switching

MR	Managed Resource
MSAN	Multiservice Access Node
MUWS	Management Using Web Services
NATO	North Atlantic Treaty Organization
NLR	National Lambda Rail
NNI	Network to Network Interface
NOC	Network Operations Center
NTE	Network Termination Equipment
OAM&P	Operations, Administration, Maintenance and Provisioning
OASIS	Organization for Advancement of Structured Information Standards
OGF	Open Grid Forum
OMA	Open Mobile Alliance
OMG	Object Management Group
OPN	Optical Private Network
OS	Operating System
OSCARS	On-demand Secure Circuits and Advance Reservation System
OSF	Open Software Foundation
OSPF	Open Shortest Path First
OSS	Operations Support System
O-VPN	Optical Virtual Private Network
P2P	Peer to Peer
PAN	Personal Area Network
PB	Petabyte
PBX	Private Branch Exchange
PCE	Path Computation Element
PE	Provider Edge
PIT	Port Information Table
PM	Policy Manager
PPI	Provider Port Identifier
PSS	Path Setup Subsystem
PSTN	Public Switched Telephone Network
PTT	Push to Talk
QoS	Quality Of Service
R&E	Research and Education
RFC	Request For comments
RM	Resource Management
RMI	Remote Method Invocation
RON	Regional Optical Network
RP	Resource Protector
SC	Science Collaboration

SCA	Service Component Architecture
SCF	Systems Capability Framework
SDH	Synchronous Digital Hierarchy
SDN	Science Data Network
SDO	Service Data Object
SEF	Service Execution Framework
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SLS	Standard Service Level Specification
SMTP	Simple Message Transfer Protocol
SNMP	Simple Network Management Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOE	Service Oriented Enterprise
SP	Service Provider
SQA	Software Quality Assurance
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Traffic Engineering
TMN	Telecommunication Management Network
TrS	Transformation Service
ToS	Topology Service
TS	Trading Service
UDDI	Universal Description, Discover and Integration
UNI	User Network Interface
URI	Uniform Resource Identifier
VCa	Virtual Carrier
VC	Virtual Circuit
VM	Virtual Machine
VoIP	Voice Over IP
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WAN	Wide Area Network
WBUI	Web-Based User Interface
WDM	Wavelength Division Multiplexing
WS	Web Service
WSDL	Web Service Description Language
WSDM	Web Service Distributed Management
WS-I	Web Service Interoperability
WSN	Web Service Network

WSRF Web Services Resource Framework
XML Extensible Markup Language

2. Introducción

2.1. Breve historia de la arquitectura de software

Cada vez que se narra la historia de la arquitectura de software o de la ingeniería de software, se reconoce que en un principio, hacia 1968, Edsger Dijkstra, propuso que se estableciera una estructuración correcta de los sistemas de software antes de lanzarse a programar, escribiendo código de cualquier manera. Dijkstra, quien sostenía que las ciencias de la computación eran una rama aplicada de las matemáticas y sugería seguir pasos formales para descomponer problemas mayores, fue uno de los introductores de la noción de sistemas operativos organizados en capas que se comunican sólo con las capas adyacentes y que se superponen “como capas de cebolla”. Inventó o ayudó a precisar además docenas de conceptos: el algoritmo de camino más corto, los stacks, los vectores, los semáforos, los abrazos mortales. De sus ensayos arranca la tradición de hacer referencia a “niveles de abstracción” que ha sido tan común en la arquitectura subsiguiente [1].

Aunque Dijkstra no utiliza el término arquitectura para describir el diseño conceptual del software, sus conceptos sientan las bases para lo que luego expresarían en 1971 Niklaus Wirth como *stepwise refinement* y DeRemer y Kron en 1976 como *programming-in-the large* o programación en grande, ideas que poco a poco irían decantando entre los ingenieros primero y los arquitectos después [1].

En la conferencia de la OTAN de 1969, un año después de la sesión en que se fundara la ingeniería de software, P. I. Sharp formuló estas sorprendentes apreciaciones comentando las ideas de Dijkstra: “Pienso que tenemos algo, a parte de la ingeniería de software, algo de lo que hemos hablado muy poco pero que deberíamos poner sobre el tapete y concentrar la atención en ello, es la cuestión de la arquitectura de software. La arquitectura es diferente de la ingeniería, ejemplo de lo que quiero decir, echemos una mirada a OS/360. Partes de OS, si vamos al detalle, han utilizado técnicas que hemos acordado constituyen buena práctica de programación. La razón de que OS sea un amontonamiento amorfo de programas es que no tuvo arquitecto. Su diseño fue delegado a series de grupos de ingenieros, cada uno de los cuales inventó su propia arquitectura. Y cuando esos pedazos se clavaron todos juntos no produjeron una tersa y bella pieza de software” [1].

Una novedad importante en la década de 1970 fue el advenimiento del diseño estructurado y de los primeros modelos explícitos de desarrollo de software. Estos modelos comenzaron a basarse en una estrategia más orgánica, evolutiva, cíclica, dejando atrás las metáforas del desarrollo en cascada que se inspiraban más bien en la línea de montaje de la ingeniería del hardware y la manufactura. Poco a poco el diseño se fue independizando de la implementación, y se forjaron herramientas, técnicas y lenguajes de modelado específicos.

En la misma época, otro precursor importante, David Parnas, demostró que los criterios seleccionados en la descomposición de un sistema impactan en la estructura de los programas y propuso diversos principios de diseño que debían seguirse a fin de obtener una estructura adecuada. Parnas desarrolló temas tales como módulos con ocultamiento de información, estructuras de software y familias de programas, enfatizando siempre la búsqueda de calidad del software. En 1972, Parnas publicó un ensayo en el que discutía la forma en que la modularidad en el diseño de sistemas podía mejorar la flexibilidad y el control conceptual del sistema, acortando los tiempos de desarrollo. Introdujo entonces el concepto de ocultamiento de información (information hiding), uno de los principios de diseño fundamentales en diseño de software aún en la actualidad. En la segunda de las descomposiciones que propone Parnas comienza a utilizarse el ocultamiento de información como criterio. Cada módulo deviene entonces una caja negra para los demás módulos del sistema, los cuales podrán acceder a aquél a través de interfaces bien definidas, en gran medida invariables [1].

En 1975, Brooks, diseñador del sistema operativo OS/360 y Premio Turing 2000, utilizaba el concepto de arquitectura del sistema para designar “la especificación completa y detallada de la interfaz de usuario” y consideraba que el arquitecto es un agente del usuario. También distinguía entre arquitectura e implementación; mientras aquella decía qué hacer, la implementación se ocupa de cómo.

Como escriben Clements y Northrop en 1996 en todo el desarrollo ulterior de la disciplina permanecería en primer plano esta misma idea: la estructura es primordial, y la elección de la estructura correcta ha de ser crítica para el éxito del desarrollo de una solución. “La elección de la estructura correcta sintetiza, como ninguna otra expresión, el programa y la razón de ser de la arquitectura de software”.

En la década de 1980, fue surgiendo nuevo paradigma, el de la programación orientada a objetos. Paralelamente, hacia fines de la década de 1980 y comienzos de la siguiente, la expresión arquitectura de software comienza a aparecer nuevamente en la literatura para hacer referencia a la configuración morfológica de una aplicación [1].

Los científicos comprendieron los principios de SOA a mediados de 1980 cuando la computación distribuida y las llamadas a procedimientos remotos llegaron al mercado. Sin embargo, su uso durante los años 1980 y 1990 se limitó a proyectos de vanguardia, proyectos donde arquitectos tuvieron la visión, la disciplina y el dinero para invertir en las fases iniciales de la aplicación en desarrollo, sabiendo que las recompensas de SOA estarían entre la escalabilidad, agilidad y reutilización. No hubo middleware estándar para la infraestructura, ni un nivel de programación de aplicaciones de interfaz y protocolo para las interfaces SOA, a pesar de los intentos como el DCE lanzado por Open Software Foundation (OSF), y CORBA definido y controlado por OMG, para proporcionar dichas normas [2]

El primer estudio en que aparece la expresión “arquitectura de software” en el sentido en que hoy lo conocemos es sin duda el de Perry y Wolf; ocurrió en 1992, aunque el trabajo se fue gestando desde 1989. En él, los autores proponen concebir la arquitectura de software por analogía con la arquitectura de edificios, una analogía de la que luego algunos abusaron [3], otros encontraron útil y para unos pocos ha devenido inaceptable [4].

“La década de 1990, fue la década de la “arquitectura de software”, dando cumplimiento a las profecías de Perry y Wolf, fue sin duda la década de consolidación y diseminación de la arquitectura de software en una escala sin precedentes. Las contribuciones más importantes surgieron en torno del instituto de ingeniería de la información de la Universidad Carnegie Mellon (CMU SEI). En la misma década, demasiado pródiga en acontecimientos, surge también la programación basada en componentes, que en su momento de mayor impacto impulsó a algunos arquitectos mayores, como Paul Clements [5], a afirmar que la arquitectura de software promovía un modelo que debía ser más de integración de componentes pre-programados que de programación [1].

La incorporación de la arquitectura SOA finalmente entra en las empresas hacia 2003 principalmente por [6]:

- La incesante presión de los negocios para la agilidad. Cuando una empresa quiere modificar sus procesos, productos o servicios, no puede permitirse el lujo de esperar por mucho tiempo. Debe ser posible cambiar la forma de aplicación de los sistemas de trabajo simplemente modificando los componentes que ya están en uso, en lugar de comprar o codificar nuevos componentes o sistemas enteros desde cero.
- La flexibilidad de la arquitectura SOA basada en Servicios Web de apoyo a múltiples aplicaciones.
- La aceptación unánime de proveedores de los estándares de Servicios Web, especialmente de Simple Object Access Protocol (SOAP) y Web Service Description Language (WSDL) [6].

La figura 1 sintetiza la evolución cronológica de las arquitecturas desde sus fases iniciales de arquitecturas monolíticas hasta hoy en día [7].

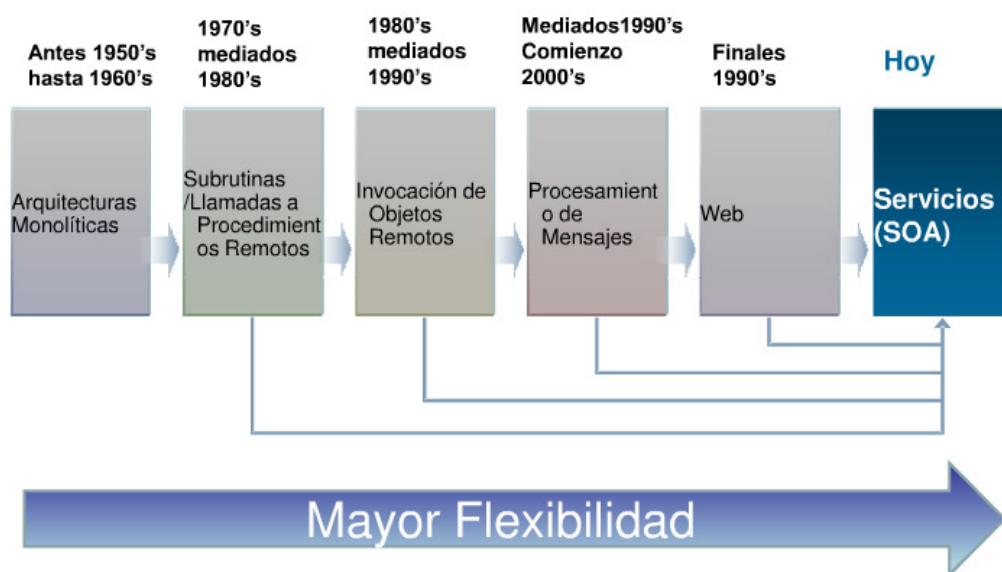


Figura 1: Evolución de las arquitecturas en el tiempo.

La figura 2 muestra la evolución de las arquitecturas en base al desarrollo y aparición de nuevos conceptos y nuevas tecnologías. En los inicios encontramos las primeras arquitecturas basadas en protocolos monolíticos, entonces sólo una capa del proceso de transferencia era controlada. El siguiente paso fue la aparición de los protocolos estructurados, en los cuales diversas capas se coordinan para realizar la comunicación. Como respuesta a la evolución del

hardware surgió la arquitectura cliente/servidor, un modelo versátil, modular, basado en mensajes. Este tipo de arquitectura dio paso a las arquitecturas cliente/servidor 3-Tier y multi-Tier o N-Tier, a los objetos distribuidos (cabe destacar las tres tecnologías más importantes RMI, DCOM y CORBA), los componentes (los Java Beans constituyen la arquitectura de componentes de Java independientes de la plataforma y su uso ha sido de gran valor en el desarrollo de aplicaciones). Sin duda el concepto de los Web Services fue crucial para llegar a la arquitectura orientada a servicio [8].



Figura 2: Evolución y antecedentes.

La figura 3 resume las propiedades de cada tipo de arquitectura [8]:

	Programación Estructurada	Objetos	Componentes	Servicios
Granularidad	Muy fina	Fina	Intermedia	Gruesa
Contrato	Definido	Privado/Público	Público	Publicado
Reusabilidad	Baja	Baja	Intermedia	Alta
Acoplamiento	Fuerte	Fuerte	Débil	Muy débil
Dependencias	Tiempo de Compilación	Tiempo de Compilación	Tiempo de Compilación	Run-Time
Ámbito de Comunicación	Intra-Aplicación	Intra-Aplicación	Inter-Aplicaciones	Inter-Empresas

Figura 3: Propiedades de las diferentes arquitecturas.

La figura 4 analiza el impacto de SOA en la evolución de las tecnologías de la información desde el punto de vista del desarrollo de las aplicaciones [9]. A medida que las

arquitecturas han ido adoptando las diferentes formas de desarrollo, el desacoplamiento, la interoperabilidad y la estandarización han ido creciendo.

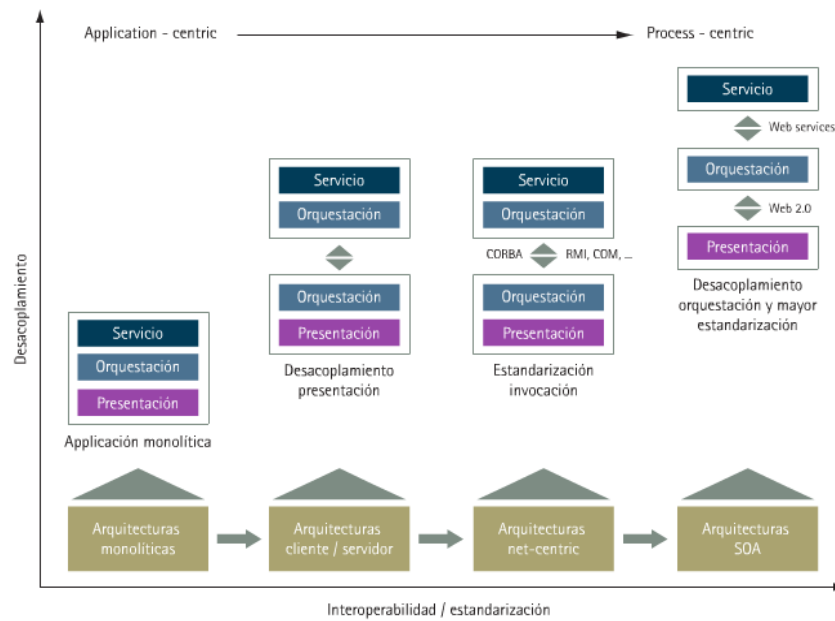


Figura 4: Impacto SOA en la evolución de las TI.

2.2. Objetivos

El presente proyecto final de carrera pretende dar un conocimiento, tanto a nivel de concepto como de tecnologías utilizadas, de las arquitecturas orientadas a servicio. La investigación de estudios realizados y la recopilación de datos tienen como fin llegar a alcanzar los siguientes objetivos:

1. Conocer el porqué y las necesidades de la arquitectura orientada a servicio.
2. Entender el significado y conceptos básicos de SOA.
3. Explicar la relación y las diferencias entre SOA y los Web Services.
4. Identificar los elementos que forman parte de la arquitectura SOA y cómo llevar a cabo su implementación.

3. SOA: Arquitectura Orientada a Servicio

3.1. La aparición de SOA

Los sistemas IT han evolucionado exponencialmente, lo cual se ha transmitido en un crecimiento de la complejidad del software de las empresas. Las tecnologías tradicionales han alcanzado su límite de capacidad y en las empresas cada vez se pide una respuesta más rápida a los requerimientos de negocio, reducciones de costes y a la integración de nuevos patrones y de nuevos clientes. Las organizaciones de IT han buscado diferentes implementaciones antes de llegar a desarrollar SOA y los problemas básicos con lo que se han encontrado y con los que hemos de contar cuando queramos hacer implementaciones de SOA son [10]:

-Complejidad: Los entornos son complejos. Las obligaciones de cumplir los presupuestos y la eficiencia de operación hacen necesario una reutilización de los sistemas, más que un reemplazo de éstos. El fácil y barato acceso a Internet ha hecho posible la constante creación de nuevos modelos de negocio que se han de evaluar para seguir el ritmo de la competencia. EL crecimiento por fusión y adquisición están a la orden del día y por ello es necesario que las organizaciones IT, aplicaciones e infraestructuras se integren y absorban. A esta complejidad se le suma las soluciones punto a punto que lo hacen todavía más complicado, con lo que es necesario desarrollar un entorno heterogéneo en el que se utilicen todo tipo de hardware, sistemas operativos, lenguajes, tipos de datos...

-Programación redundante y no reutilizable: las aplicaciones igual que las empresas se van desarrollando a partir de fusiones y adquisiciones, lo cual hace que se tenga que trabajar con aplicaciones redundantes o aplicaciones con funciones que no son fácilmente reutilizables. Si cada unidad de negocio se desarrolla independientemente, a la hora de actualizar o de introducir los nuevos productos y servicios, la redundancia hace que aumenten los costes y el tiempo necesario para introducir los cambios en cada parte afectada.

-Interfaces múltiples: Si tenemos que interconectar N sistemas existentes necesitamos $N(N-1)$ interfaces para hacer posible esta realización, con lo que si queremos introducir un nuevo sistema deberá crear documentación, testear y mantener $2N$ nuevos interfaces.

Hasta ahora se han desarrollado diferentes modelos de programación que han ido intentando perfeccionar las barreras comentadas. Al aparecer la tecnología Java se introdujo una

programación de plataforma neutral. XML describió sus propios datos neutrales. En la actualidad, Web Services han roto toda barrera para interconectar aplicaciones de una manera basada en un modelo orientado a objeto neutral. Los Web Services son otro de los pilares de SOA y se definen como un conjunto de protocolos que se usan para que los servicios sean publicados, descubiertos y usados en una tecnología neutra y de una forma estándar.

Podemos concluir que los modelos estándares de SOA se basan en J2EE y Web Services. SOA es un modelo de programación y de arquitectura a la vez. Nos permite diseñar sistemas software que proveen de servicios a otras aplicaciones a través de interfaces publicados y descubiertos. Los servicios pueden ser invocados sobre una red. Si se utilizan Web Services para implementar una SOA, se crea un camino para construir aplicaciones dentro de un modelo de programación más robusto y flexible. Se reduce el desarrollo, los costes y el riesgo de implementación.

3.2. Servicio

El acrónimo SOA significa Arquitectura Orientada a Servicio. El concepto más importante de SOA es el de SERVICIO. Según el CBDI (Component Based Development and Integration) Forum su definición sería: “Vehículo a partir del cual las necesidades de los clientes son satisfechas de acuerdo con el contrato negociado (implícito o explícito), el cual incluye el acuerdo de servicio (Service Agreement), la función ofrecida etc”. Otra definición posible que podemos encontrar, es la que nos da el W3C (World Wide Web Consortium): “Componente capaz de realizar una tarea” [11].

Se puede definir un servicio como un paquete funcional de software al cual se puede acceder a través de una infraestructura de red. Los servicios son autónomos, autocontenidos y uno no puede tener control, ni autoridad sobre ellos. Un ejemplo de servicio puede ser, dentro de un entorno de negocio, una transacción simple (como obtener la cuota de stock, conseguir las direcciones de los clientes...), una transacción de negocio más compleja (calendario de entregas, cobertura de ventas...) o servicios de sistema (autenticación de usuario, mensaje de conexión...). Las funciones de negocio desde el punto de vista de aplicación no son funciones de sistema, y son atómicas (aunque estén formadas por funciones más pequeñas, se considera como un paquete indivisible). Las transacciones de negocio tienen que implementarse como una composición de funciones de bajo nivel transparentes al que las llama. Por último los servicios de sistema son funciones genéricas que pueden ser abstraídas fuera de la plataforma particular, como Microsoft Windows o Linux.

La riqueza funcional de las aplicaciones nos la da el nivel de granularidad. La granularidad es la capacidad de descomponer las aplicaciones de negocio en servicios y tiene implicaciones prácticas, no es sólo un proceso de abstracción. Los servicios pueden ser de baja granularidad o funciones complejas de alta granularidad (fine-grained o coarse-grained respectivamente) [10]. Generalmente, los servicios son funciones coarse-grained ya que suelen ser el resultado de ejecutar varias operaciones fine-grained; por ejemplo, abrir una cuenta está compuesto por identificar al usuario y crear una cuenta nueva etc. Esta es la manera de desarrollar un entorno de aplicaciones basadas en componentes, donde los servicios se definen como un conjunto de componentes reusables que pueden ser usados para la construcción de nuevas aplicaciones, o para integrar los recursos de software ya existentes.

3.2.1. Orientación a Objetos y Desarrollo Basado en Componentes

Se pueden encontrar paralelismos entre la Orientación a Servicios (OS), la orientación a objetos (OO) y el desarrollo basado en componentes (CBD) [11]:

- Los servicios también representan bloques de construcción naturales que nos permiten organizar capacidades de forma que nos sean familiares.
- Como los objetos y los componentes, los servicios son un bloque fundamental que combina información y comportamiento, esconde el trabajo interno para que esté fuera de intrusos y presenta un interfaz simple para el resto del organismo.
- Los objetos usan tipos de datos abstractos y datos abstractos. Los servicios tienen un nivel similar de adaptabilidad a través del aspecto o del contexto de orientación.
- Se puede organizar también en clases y jerarquías de servicios heredando los comportamientos y pudiendo ser utilizados después de forma única o como jerarquías o colaboraciones.

Por otro lado, mientras los componentes son la mejor forma de implementar servicios, se debe entender que una aplicación correctamente basada en componentes, no necesariamente es una aplicación correctamente orientada a servicios. La clave para comprender esta diferencia radica en ver como una arquitectura orientada a servicios (SOA) implica una capa adicional de arquitectura (una nueva abstracción) implementada con una granularidad más alta y ubicada más cerca del consumidor de la aplicación (figura 5) [12].

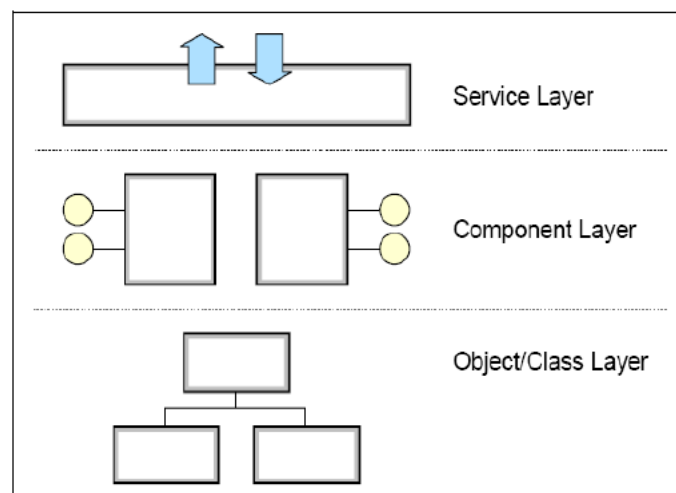


Figura 5: Capas de aplicación.

Un servicio es una forma de exponer una visión externa de un sistema, con reutilización interna y una composición tradicional basada en el diseño de componentes. En una SOA, un servicio mapea una función identificada durante un proceso de análisis de negocio, dependiendo de la función de negocio de que se trate, la granularidad del mismo puede ser más o menos baja o alta. Los servicios no se diseñan en base a las entidades de negocio; cada servicio es una unidad que maneja operaciones a través de un conjunto de entidades de negocio.

Como ya hemos destacado, un servicio es una unidad de procesamiento de granularidad gruesa, que consume y produce un conjunto de objetos pasados por valor, implementada sobre una colección de componentes que trabajan en colaboración para entregar la funcionalidad del negocio que el mismo representa; los componentes son de una granularidad más baja que la de los servicios. Mientras un servicio mapea una funcionalidad del negocio, un componente típicamente mapea las entidades del negocio y las reglas que las operan.

3.2.2. Arquitectura de Servicios

El siguiente diagrama de la figura 6 muestra la interrelación entre las arquitecturas de Aplicación, Servicios y Componentes, y la implementación de procesos de negocio mediante orquestación de servicios [11].

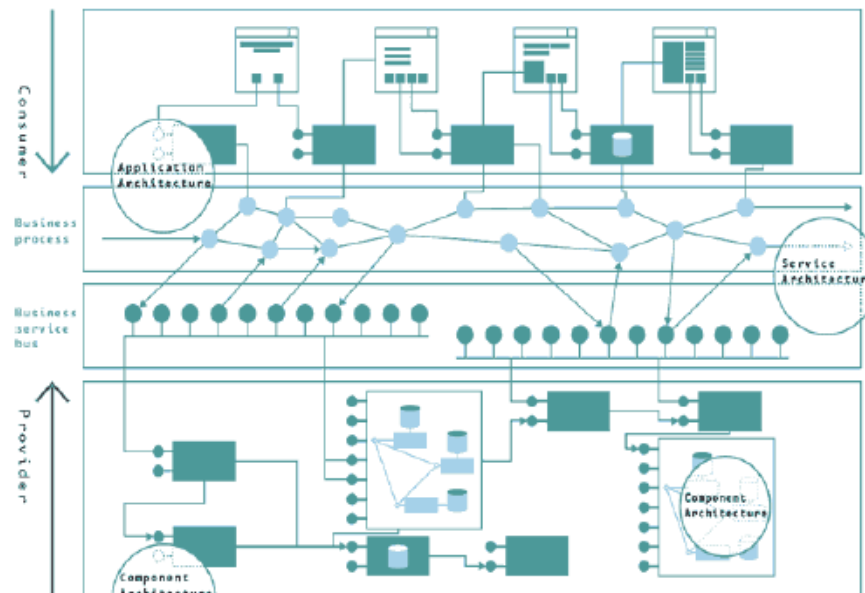


Figura 6: Tres perspectivas de arquitectura.

1. Estructura y características de los Servicios

Los servicios son una forma de encapsular componentes/programas reusables (building blocks) para proveer funcionalidad a otros usuarios y a otros servicios. Cuando un servicio provee servicios a otro, al servicio que invoca lo llamaremos consumidor, para distinguirlo del usuario. Con los servicios se interactúa mediante el intercambio de mensajes. Un servicio consiste de 3 elementos [12]:

- **Contrato:** El uso de la funcionalidad que provee un servicio es gobernada por un contrato. Especifica el propósito, la funcionalidad, las restricciones y el modo de uso del servicio. Es definido por el negocio, en términos del negocio.
- **Implementación:** La funcionalidad en sí misma que provee el servicio, puede ser realizada utilizando cualquier tecnología.
- **Interfaces:** Para acceder a la funcionalidad el consumidor necesita “interfacear” con el servicio. Proveen la forma de acceder a la funcionalidad de acuerdo al contrato. Un servicio puede ofrecer múltiples interfaces para permitir su consumo de diferentes maneras.

Las características funcionales de los servicios son:

- Invocación: sincrónica o asincrónica
- Intercambio: uni-direccional, bi-direccional
- Complejidad: referido a la granularidad

Las características no-funcionales:

- Requerimientos de Volúmenes
- Calidad del Servicio
- Tiempo de ejecución del Servicio

2. Categorización de los Componentes de Servicio

Según la función que cumplen pueden identificarse en primera instancia las siguientes categorías:

- Administración de datos
- Lógica de negocios básica
- Lógica de negocios compuesta
- Interacción con el usuario
- Componentes utilitarios comunes

3. Principios Comunes de la Orientación a Servicios

- Comparten un contrato formal
- Bajamente acoplados
- Abstraen la lógica que existen debajo (autocontenidos y modulares)
- Interoperables
- Componibles
- Reusables
- Autónomos
- Sin estado
- Descubribles (transparentes a la ubicación)

4. Pasos a seguir para definir los Servicios

1. Definir el propósito del servicio (orientado al negocio).

2. Determinar la información que debe de manejar el servicio (metadata y schemas).



3. Identificar los potenciales consumidores.
4. Definir los aspectos de niveles de servicio, seguridad y performance que brindará el servicio.
5. Determinar las funciones (métodos) encapsuladas dentro del servicio, es decir el comportamiento interno.
6. Definir las interfaces, los parámetros y el mapeo con las funciones o métodos internos.
7. Definir como deberá ser testeado el servicio (test information, service invocation, validez de los resultados, etc).
8. Definir la documentación a incorporar.

3.3. Definición de SOA

La definición de SOA por el W3C es: “Conjunto de componentes, los cuales pueden ser invocados y cuyas descripciones de interfaces pueden ser invocadas y descubiertas”.

Según el CBDI sería: “Estilo resultante de políticas, prácticas y frameworks que permiten que la funcionalidad de una aplicación se pueda proveer y consumir como conjuntos de servicios, con una granularidad relevante para el consumidor...”. Los servicios pueden ser invocados, publicados y descubiertos, y extraídos de la implementación usando interfaces simples y estándares.

OASIS (Organization for Advancement of Structured Information Standards) definió SOA como: “SOA es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de diferentes dominios. Provee una manera uniforme de ofrecer, descubrir, interactuar con ellos y sus capacidades de uso para producir el efecto deseado consistente en precondiciones y expectativas medibles”.

Otras definiciones que podemos encontrar:

BEA: “Es una estrategia de IT que organiza las funciones discretas contenidas en las aplicaciones empresariales en servicios estandarizados, interoperables, de forma que puedan ser combinados y reutilizados fácil y rápidamente para adaptarse a los requerimientos de negocio”.

IBM: “SOA representa una forma de construir sistemas distribuidos que permite ofrecer las funcionalidades de una aplicación como servicios tanto para aplicaciones de usuario final como para otros servicios”. IBM se refiere a SOA como “la plataforma que alinea el Negocio con Tecnología”

Gartner: “SOA es una arquitectura de software que comienza con una definición de interfaz y construye toda la topología de la aplicación como una topología de interfaces, implementaciones y llamadas a interfaces. Sería más adecuado llamarla “arquitectura orientada a interfaces”. SOA es una relación de servicios y consumidores de servicios, ambos suficientemente amplios para representar una función de negocios completa”.

SUN: “Una arquitectura orientada a servicios es una estrategia donde las aplicaciones hacen uso (o se basan) en servicios disponibles en una red. Siendo una manera de compartir funciones (típicamente de negocios) de una manera flexible y extendida.”[12]

Oracle indica “SOA se ha movido de la sobreexpectativa (hype) a una completa aceptación como estrategia de Tecnología (TI) para entregar soluciones de Negocio” [13].

SOA no es sólo una arquitectura de servicios visto desde una perspectiva de la tecnología, si no las políticas, prácticas y frameworks con los que se garantiza la forma correcta de que los servicios sean provistos y consumidos [11]. Es importante que si un servicio no va a ser usado por múltiples consumidores, la especificación sea generalizada; los servicios necesitan ser extraídos de la implementación y los desarrolladores de las aplicaciones del consumidor no deberían necesitar saber nada sobre los modelos de bajo nivel y sus reglas.

SOA es una tecnología madura, y respaldada por los principales expertos y empresas, por ejemplo Gartner dentro de su HypeCycle (diagrama de madurez) ya en el 2009 lo consideró a nivel de plena productividad (plateau of productivity). Además Gartner la tiene dentro de las tecnologías Top Ten en la industria de Seguros [14].

3.3.1. Taxonomías de SOA

El siguiente diagrama de la figura 8 muestra una vista tecnológica de SOA con una fuerte orientación a su implementación [12].

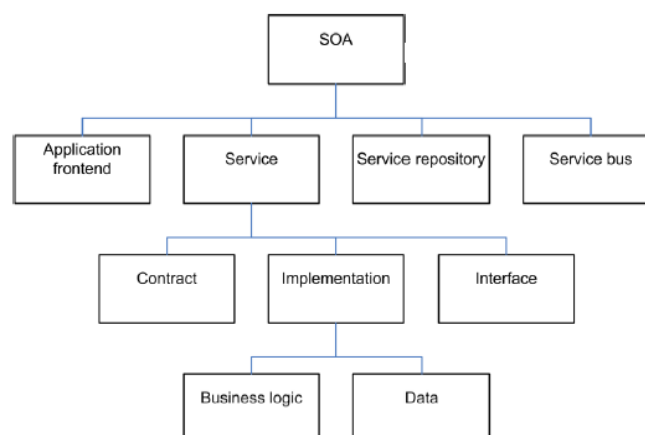


Figura 8: Diagrama de componentes de una arquitectura SOA.

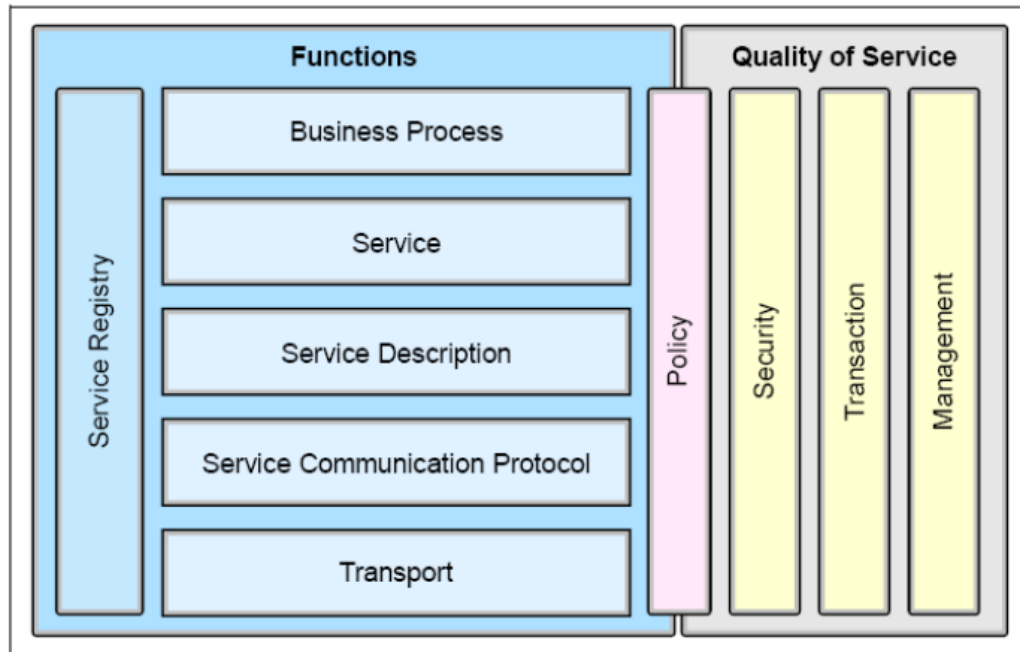


Figura 9: Componentes SOA.

Funciones:

1. Transporte: Mecanismo utilizado para trasladar las peticiones desde el cliente, hasta el proveedor del servicio, y viceversa.
2. Protocolo de comunicación: Es el sistema de comunicación entre el cliente y el proveedor de servicios.
3. Descripción del servicio: Es un esquema utilizado para describir qué servicio es, cómo se le puede invocar, y cuáles son los datos necesarios para realizar su invocación.
4. Servicio: Es la implementación del servicio.
5. Proceso de negocio: Es una colección de servicios, invocados en una determinada secuencia, con un conjunto particular de reglas para satisfacer un requisito de negocio.

6. Registro de servicios: Es un repositorio de servicios y datos, usado por los proveedores de servicio, para publicar los servicios para que los clientes tengan donde buscarlos.

Calidad del servicio:

1. Políticas: Son un conjunto de reglas bajo las cuales, un proveedor de servicio hace que el servicio esté disponible para los clientes.
2. Seguridad: Son un conjunto de reglas que podrían ser aplicadas en la identificación, autorización y control de acceso a los servicios, por parte del cliente.
3. Transacción: Conjunto de atributos que podrían ser aplicados sobre un grupo de servicios para devolver un conjunto de datos consistentes.
4. Gestión: Conjunto de atributos que podrían ser aplicados para gestionar los servicios proporcionados.

3.3.2. Requerimientos

Los requerimientos que se han de tener en cuenta a la hora de desplegar SOA son [10]:

1. Recursos existentes: Es el requerimiento más importante. Los recursos existentes, como datos de vital importancia, aplicaciones etc. deben ser integrados en los nuevos sistemas.
2. Soportar todo tipo de integración requerida: Incluye la interacción de los usuarios, la conectividad de la aplicación, integración de procesos, de información, y construir para integrar.
3. Permitir el crecimiento de las implementaciones y la migración de los recursos: Es el aspecto más crítico de las implementaciones de esta arquitectura. Incontables implementaciones anteriores han fracasado por este motivo, su complejidad, los costes y los esquemas de implementación con los que se hace difícil trabajar.
4. Construir en un marco de componentes estándar: De esta forma puedes conseguir una mayor reutilización de los módulos y sistemas.

5. Permitir implementaciones de nuevos modelos de cómputo: Ejemplos específicos de este requerimiento son nuevos modelos de clientes basados en portales, grid computing, on demand computing y cloud computing.

A través de los años se han propuesto y adoptado diferentes paradigmas de cómputo gracias al descubrimiento de nuevos avances tecnológicos como procesadores multinúcleo (multicore) y entornos de red de cómputo. Entre estos paradigmas se hallan el clúster computing, el grid computing, on demand computing, P2P computing, service computing, market-oriented computing y el más reciente el Cloud computing. Los servicios de cómputo necesitan ser altamente fiables, escalables y autónomos para soportar accesos múltiples, descubrimientos dinámicos y la composición de los mismos. En particular, cada consumidor puede determinar el nivel de servicio que requiere a través de los parámetros de QoS y los SLAs (Service Level Agreements). Entre todos los paradigmas citados los que más futuro parecen tener son el grid computing y el cloud computing.

1. Grid Computing

Un grid permite particiones, selecciones y agregaciones de una amplia variedad de recursos geográficamente distribuidos incluyendo supercomputadores, sistemas de almacenamiento, fuentes de datos y dispositivos especializados pertenecientes a diferentes organizaciones para resolver los problemas de recursos [15].



Figura 10: Campos de aplicación del Grid Computing.

Utilizando grids se pueden unir capacidades tecnológicas dispares para crear un único e integrado sistema. Esto permite manejar, compartir y acceder virtualmente a dispositivos a lo largo y ancho de la empresa.

Grid Computing se puede definir como la aplicación de recursos de muchos computadores en una red al mismo tiempo para la solución de un problema, usualmente un problema que requiere un gran número de ciclos de procesamiento o acceso a mucha cantidad de datos [16].

El grid computing no es sólo una aplicación con un gran número de instrucciones por segundo (MIPS) para dar una solución de cálculo para un problema complejo, sino que te permite dividir los recursos en múltiples entornos de ejecución aplicando algunos conceptos como la partición de hardware o de software, el time-sharing, la simulación de máquina o máquina virtual, emulación y calidad de servicio. Esta virtualización, o desarrollo bajo demanda de los recursos permite usarlos como y cuando sean necesarios dentro del grid. La virtualización es una forma simple de manejar los recursos, el espacio, las aplicaciones, los servicios. Es por ello que SOA maximiza la utilización de los recursos en un entorno de trabajo con grids, con lo que responde de una manera eficiente a los cambios en el entorno de negocio tanto externo como interno [10].

2. On Demand Computing

On Demand (OD) Computing es un modelo de empresa cada vez más popular en la que los recursos informáticos se ponen a disposición del usuario, según sea necesario. Los recursos se pueden mantener dentro de la empresa del usuario, o pueden ser puestos a disposición por un proveedor de servicios. El modelo On-Demand se desarrolló para superar el reto común de ser capaz de satisfacer las demandas fluctuantes de manera eficiente. Debido a que la demanda por parte de las empresas, de recursos de cómputo, puede variar drásticamente de un momento a otro, el mantenimiento de los recursos suficientes para cumplir con los requisitos de pico puede ser costoso. Por el contrario, si la empresa reduce los costes por mantener sólo un mínimo de recursos de computación, no habrá recursos suficientes para satisfacer las necesidades de pico.

On demand se podría clasificar como un servicio que siempre está disponible pero las empresas sólo pagan cuando lo utilizan [16].

SOA puede ser un prerequisite esencial para “on demand computing” o cálculo bajo demanda. SOA es una arquitectura que permite aplicaciones bajo demanda. Como un

subconjunto propio de “on demand computing”, Web Services bajo demanda son simplemente servicios de negocio expuestos usando estándares de Web Services. “On demand computing” puede cubrir un espectro ancho. Una finalidad de este espectro es focalizar el entorno de aplicación. Otra finalidad se fija en el entorno de operación, el cual incluye partes como la infraestructura y el cálculo autónomo. En el centro de tu negocio “on demand” habrá un servicio de negocio también “on demand” donde los servicios de aplicación de nivel pueden ser descubiertos, ensamblados, reconfigurados y repartidos bajo demanda con capacidades de integración just-in-time [10].

3. Cloud Computing

Hoy en día el paradigma que está emergiendo es el Cloud computing, el cual promete servicios fiables a través de la siguiente generación de centros de datos basado en tecnologías de virtualización de cálculo y de almacenamiento. Los clientes acceden a los datos y a las aplicaciones de una “Cloud” en cualquier sitio del mundo “on demand”, la “Cloud” parece ser un único punto de acceso para satisfacer todas las necesidades de los consumidores. La infraestructura de “Cloud” es para los clientes, robusta y estará disponible en cualquier momento.

Una posible definición de Cloud sería: “Es un tipo de sistema paralelo y distribuido que se compone de un conjunto de equipos interconectados y virtualizados que son provistos y presentados dinámicamente como uno o más recursos de cómputo unificado basado en acuerdos de nivel de servicio establecidos a través de negociaciones entre el proveedor de servicio y los consumidores” [15].

Cloud Computing se trata de un estilo de computación donde las tecnologías de la información se prestan cómo un servicio, estos servicios están localizados en los centros de datos (Cloud o nube), permitiendo a los usuarios el acceso a los servicios sin necesidad de que el usuario tenga el conocimiento, la experiencia y la infraestructura para poder mantener este servicio.

Enfoca un concepto general que incorpora al software como un servicio, por ejemplo muchos servicios Web 2.0 recientes, usan esta tendencia, donde lo único que necesitan los usuarios para poder acceder al servicio es una conexión a Internet. Un ejemplo muy claro es Google Apps que

proporciona aplicaciones de negocios en línea al que se accede desde un navegador web, mientras que el software y los datos se almacenan en los servidores de Google.

Como muestra el gráfico de la figura 11, los recursos son distribuidos de forma inteligente en una serie de servidores que conforman la nube del Cloud Computing, permitiendo que ante el uso por parte del usuario final, estos sean “tomados” de acuerdo a la disponibilidad existente, haciendo correr las aplicaciones más rápido y sin peligro de colapso por sobreutilización, ya que siempre existen recursos disponibles al sumar los existentes entre todos los equipos que participan equilibradamente en esta nube [17].



Figura 11: Distribución y uso de recursos On-Demand.

Podría parecer que una Cloud es una combinación de clústers y Grids, pero no es así. Las Clouds son claramente la siguiente generación de centros de datos virtualizados a través de tecnologías “hypervisor” como VM’s, proporcionados dinámicamente bajo demanda como una colección de recursos personalizada para encontrar un acuerdo específico de nivel de servicio, el cual se establece a través de una negociación y es accesible como un servicio compuesto, vía Web 2.0.

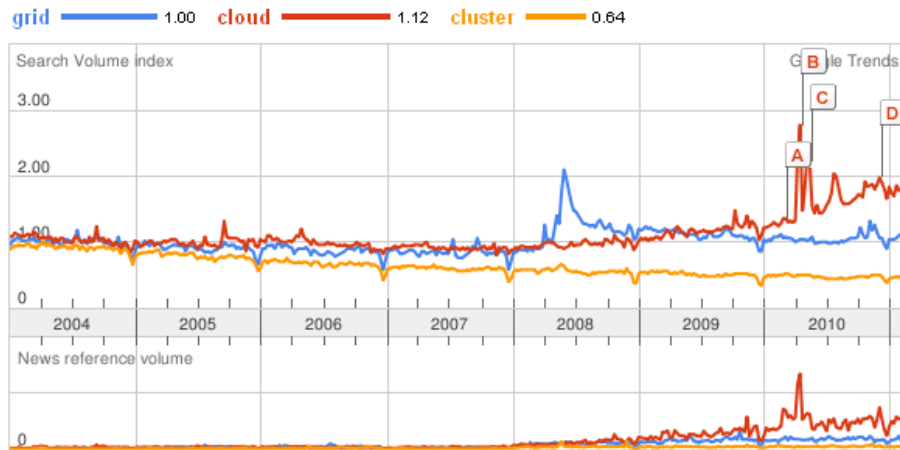


Figura 12: Tendencias de las búsquedas por Google en los últimos años.

La popularidad de los diferentes paradigmas varía con el tiempo. En la figura 12 podemos observar la popularidad de las búsquedas por Web. Se muestran las tendencias de la búsqueda en Google durante los últimos años, de los términos “Clúster computing”, “Grid computing” y “Cloud computing”. Podemos ver que desde 2009 el término Cloud computing empezó a ganar más popularidad [18].

El uso del Cloud Computing no tiene límites. Actualmente se está utilizando para:

- Servicios Internet (hospedaje de sitios y aplicaciones)
- Software de gestión empresarial (las compañías alojan en su cuenta Cloud Computing sus aplicaciones y los usuarios se conectan a ellas directamente para su desempeño diario)
- Software comercial

Las principales ventajas de utilizar Cloud Computing en lugar de los sistemas tradicionales son:

1. Bajo coste. No requiere implementar, actualizar, reparar infraestructuras tecnológicas (datacenters, servidores locales, redes, etc). Todo es entregado por el proveedor de Cloud Computing.

2. Escalabilidad. Al trabajar con el concepto On-Demand, se solicita en cualquier momento un aumento o disminución de recursos asignados a la cuenta, de acuerdo a las necesidades reales.
3. Seguridad. Cada plan Cloud Computing cuenta con firewalls lógicos y físicos de alta fiabilidad, entregados por el proveedor con la finalidad de que no se puedan vulnerar los sistemas.
4. Velocidad. El acceso de alta velocidad hará del uso del Cloud Computing para sus usuarios una grata experiencia.
5. Estabilidad. El Uptime es un aspecto importante para sus sitios Web y aplicaciones. Con el sistema de distribución de recursos este aumenta significativamente, pues si una máquina presenta algún fallo las otras continuarán sirviendo las aplicaciones y archivos.
6. Soporte Real. No se depende más de la disponibilidad de tiempo o intenciones de personal de soporte interno o externo. Se proporciona el servicio en forma dedicada y permanente, las 24 horas del día y se ocupa del correcto funcionamiento de redes y hardware sin necesidad de intervención por parte del cliente [17].

Arquitectura de la Cloud

Los consumidores confían en los proveedores de la Cloud para el suministro de todas sus necesidades de cómputo, requieren un mantenimiento de la QoS específica para conseguir sus objetivos y para sostener sus operaciones. Los proveedores de la Cloud necesitan considerar y encontrar diferentes parámetros de QoS para cada consumidor individual tal y como se especifica en los SLA's. Para llevar a cabo esto, los proveedores de la Cloud no pueden continuar desplegando arquitecturas con sistemas de gestión de recursos céntricos, ya que no les sale a cuenta compartir sus recursos y encima considerar todos los requerimientos con igual importancia. En vez de gestión de recursos orientada a mercado, es necesario regular la oferta y la demanda de los recursos de la Cloud en un equilibrio de mercado, proveer feedback en términos de incentivos económicos tanto para consumidores como para proveedores, y promocionar mecanismos de asignación de recursos basados en QoS que diferencie los requerimientos de servicio basados en su utilidad [19].

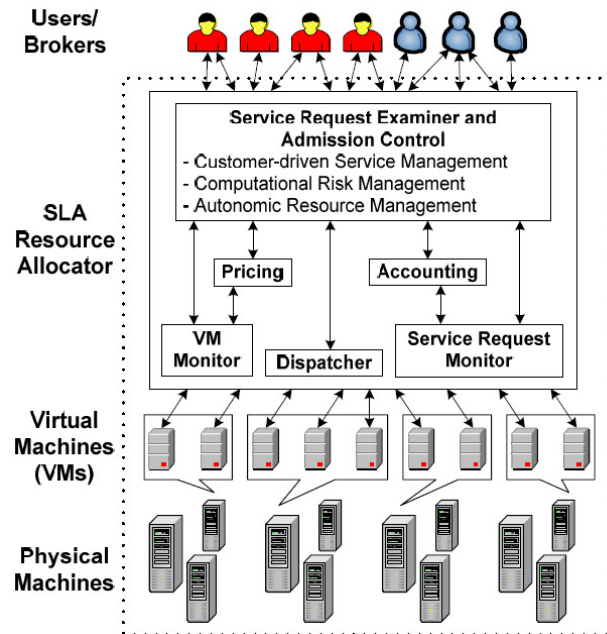


Figura 13: Nivel alto de la arquitectura cloud orientada a mercado.

La figura 13 esquematiza la arquitectura de nivel alto de los Centros de Datos y Clouds, para soportar la asignación de recursos orientada a mercado. Podemos hablar de cuatro partes involucradas [15]:

- **Users/Brokers:** Usuarios o agentes actuando en su propio nombre, requiriendo servicios desde cualquier sitio del mundo, a los Centros de Datos y a la Cloud para ser procesados.
- **SLA Resource Allocator:** Actúa como interfaz entre el proveedor de servicio del Centro de Datos/Cloud y los usuarios/agentes externos. Requiere de la interacción de los siguientes mecanismos para soportar la gestión de recursos orientada a SLA.
 1. **Service Request Examiner and Admission Control:** Cuando una solicitud de servicio se presenta por primera vez, el mecanismo del Service Request Examiner and Admission Control interpreta la solicitud proporcionada como solicitud de QoS antes de determinar si acepta o rechaza la solicitud. Por lo tanto, asegura que no haya sobrecarga de los recursos y que muchas solicitudes no puedan ser cumplidas con éxito debido a la limitada disponibilidad de recursos. Por ello, también necesita de información del último estado respecto a la disponibilidad del recurso (desde el mecanismo VM Monitor) y el

trabajo de procesamiento (desde el mecanismo Service Request Monitor) para poder tomar decisiones de asignación de recursos con eficacia. Después asigna solicitudes a los VM's y determina los derechos de recurso para los VM's asignados.

2. Pricing: El mecanismo de Pricing decide como se cargan las solicitudes de servicio. Por ejemplo, pueden ser cargadas basándose en el tiempo de sumisión (peak/off-peak), las tasas de fijación de precios (fixed/changing) o la disponibilidad de los recursos (supply/demand). Pricing sirve de base para el manejo de la oferta y demanda de los recursos de cómputo dentro de los Centros de Datos y facilita la priorización de recursos en las asignaciones de manera efectiva.
 3. Accounting: Este mecanismo mantiene el uso actual de los recursos por solicitudes, así el coste final puede ser computado y cargado a los usuarios. Además, la información mantenida en el histórico puede ser utilizada por el Service Request Examiner and Admission Control para tomar decisiones de asignación de recursos.
 4. VM Monitor: Este mecanismo hace seguimientos de la disponibilidad de VM's y sus derechos de recursos.
 5. Dispatcher: Es el encargado de empezar la ejecución de las solicitudes de servicio aceptadas en los VM's asignados.
 6. Service Request Monitor: Hace el seguimiento del progreso de la ejecución del servicio solicitado.
- VM's: Múltiples VM's pueden arrancarse o pararse dinámicamente en una única máquina física para encontrar las solicitudes de servicio aceptadas, por lo tanto, proporcionando flexibilidad máxima para configurar varias particiones de los recursos en la misma máquina física para requerimientos específicos diferentes, de solicitudes de servicio. Además, múltiples VM's pueden ejecutar aplicaciones concurrentemente basadas en diferentes entornos de sistemas operativos en una única máquina física desde que cada VM está completamente aislado de los otros en la misma máquina física.

- **Physical Machines:** El Centro de Datos comprende múltiples servidores de cómputo que proporcionan recursos para encontrar las demandas de servicio. En el caso de una Cloud, hay parámetros de QoS críticos para considerar en la solicitud de servicio, como el tiempo, el coste, la fiabilidad y la seguridad. Los requerimientos de QoS no pueden ser estáticos y necesitan actualizarse dinámicamente todo el tiempo debido a los continuos cambios en las operaciones de negocio y los entornos operativos. En pocas palabras, hay que darle más importancia a los clientes desde que pagan por acceder a los servicios de la Cloud. Se han desarrollado mecanismos de negociación basados en protocolos de ofertas alternas para el establecimiento de SLA's [20]. Esto tiene gran potencial para sus adopciones en los sistemas de cómputo en la Cloud, contruidos usando VM's.

Los ofrecimientos comerciales de las Clouds orientadas a mercado deben proporcionar [15]:

- Gestión de servicio de soporte al cliente basado en los perfiles del consumidor y en los requerimientos de servicio.
- Definición de las tácticas de gestión del riesgo computacional para identificar, evaluar y gestionar los riesgos involucrados en la ejecución de aplicaciones con respecto a las necesidades de servicio y las necesidades de los clientes.
- Obtención de estrategias apropiadas de gestión de recursos basados en el mercado que abarquen tanto la gestión de servicio de cliente como la gestión del riesgo computacional para sostener la asignación de recursos orientada a SLA.
- Incorporación de modelos de gestión de recursos autónomicos que manejan ellos mismos, de manera efectiva, los cambios en los requerimientos de servicio para satisfacer tanto la demanda de nuevos servicios como las obligaciones de los servicios existentes.
- Aprovechamiento de la tecnología VM para una asignación dinámica de las cuotas de recursos de acuerdo con los requerimientos de servicio.

Plataformas de Cloud

En la siguiente tabla podemos ver una comparativa de las plataformas de Cloud [15]:



System Property	<u>Amazon</u> Elastic Compute Cloud (EC2)	<u>Google</u> App Engine	<u>Microsoft</u> Live Mesh	<u>Sun</u> Network.com (Sun Grid)	<u>GRIDS Lab</u> Aneka
Focus	Infrastructure	Platform	Infrastructure	Infrastructure	Software Platform for enterprise Clouds
Service Type	Compute, Storage (Amazon S3)	Web application	Storage	Compute	Compute
Virtualisation	OS Level running on a Xen hypervisor	Application container	OS level	Job management system (Sun Grid Engine)	Resource Manager and Scheduler
Dynamic Negotiation of QoS Parameters	None	None	None	None	SLA-based Resource Reservation on Aneka side.
User Access Interface	Amazon EC2 Command-line Tools	Web-based Administration Console	Web-based Live Desktop and any devices with Live Mesh installed	Job submission scripts, Sun Grid Web portal	Workbench, Web-based portal
Web APIs	Yes	Yes	Unknown	Yes	Yes
Value-added Service Providers	Yes	No	No	Yes	No
Programming Framework	Customizable Linux-based Amazon Machine Image (AMI)	Python	Not applicable	Solaris OS, Java, C, C++, FORTRAN	APIs supporting different programming models in C# and other .Net supported languages

Figura 14: Tabla comparativa de algunas de las plataformas Cloud más representativas.

Cloud global de intercambio y los mercados

Las empresas actualmente emplean servicios de Cloud para mejorar la escalabilidad de sus servicios y para hacer frente a las demandas de recursos con ráfagas. En el presente, los proveedores de servicio tienen precios inflexibles, generalmente limitados a tipos fijos o tarifas basadas en el uso de umbrales, y los consumidores están restringidos a ofertas de un único proveedor a la vez. También, muchos proveedores tienen interfaces en propiedad para sus servicios restringiendo así, la habilidad de los consumidores de intercambiar un proveedor por otro. Para que el Cloud computing mature es necesario que los servicios sigan interfaces estándar. Esto permite la forma básica de los servicios y por lo tanto allana el camino para la creación de la infraestructura de mercado para el comercio de servicios.

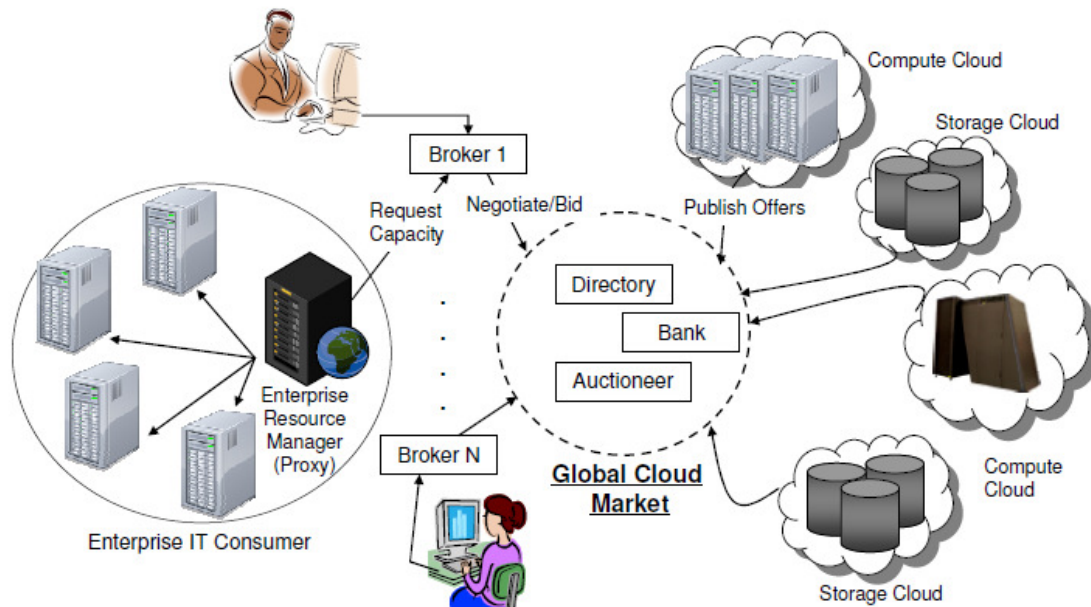


Figura 15: Intercambio global en la nube e infraestructura de mercado para servicios de compraventa.

En la figura 15 vemos un ejemplo del sistema de mercado modelado en intercambios del mundo real. El directorio de mercado permite a los participantes localizar un proveedor o un consumidor con las ofertas correctas. Los subastadores periódicamente sacan pujas y piden las respuestas de los participantes en el mercado. Los sistemas bancarios garantizan que las transacciones financieras pertenecientes a acuerdos entre los participantes, sean llevadas a cabo.

Los agentes realizan la misma función que la que hacen en el mercado en el mundo real: median entre los consumidores y los proveedores a través de la compra de capacidad al proveedor y la sub-alquilan a los consumidores. Un agente puede aceptar solicitudes de muchos usuarios, los cuales tienen elección de presentar sus requerimientos a diferentes agentes. Consumidores, agentes y proveedores están atados a sus requerimientos y las compensaciones relacionadas a través de los SLA's. Un SLA especifica los detalles del servicio para proporcionarlo en términos de las medidas acordadas por todas las partes y penaliza por las sesiones y la violación de las expectativas.

Estos mercados pueden unir Clouds dispares permitiendo a los consumidores elegir un proveedor que se acerque más a sus requisitos ya sea por ejecución de los SLA's por adelantado o por la compra de capacidades sobre la marcha.

Los proveedores pueden usar los mercados para realizar planes de capacidad con efectividad. Un proveedor está equipado con mecanismos de establecimiento de precios el cual es un conjunto de precios actuales para los recursos basado en condiciones de mercado, en la demanda de usuario y en el actual nivel de utilización de los recursos. La tarificación puede ser fija o variable dependiendo de las condiciones del mercado. El mecanismo de control de admisión, en el proveedor final, selecciona las acciones para participar en la subasta o para negociar, basado en una estimación inicial de la utilización. El proceso de negociación continúa hasta que un SLA se forma o los participantes deciden dejarlo. Este interfaz de mecanismos con sistemas de gestión de recursos del proveedor para garantizar la asignación, siendo ofrecido o negociado, puede ser reclamado, así que las violaciones del SLA no ocurren. El sistema de gestión de recursos también proporciona funcionalidades como la reserva avanzada, que permite garantizar el aprovisionamiento de las capacidades de recursos. Los agentes ganan su público a través de las diferencias entre el precio pagado por los consumidores para ganar recursos compartidos y el pago para los proveedores por alquilar sus recursos. Por consiguiente, un agente tiene que elegir usuarios cuyas aplicaciones puedan proporcionarle utilidad máxima. Un agente interactúa con los proveedores de recursos y con otros agentes para ganar o comerciar recursos compartidos. Un agente está equipado con un módulo de negociación que está formado por las condiciones actuales de los recursos y la demanda actual para tomar decisiones. Los consumidores tienen sus propias funciones, que cubren factores como límites, fidelidad de los resultados y el cambio del tiempo de las aplicaciones. Ellos también están supeditados por la cantidad de recursos que ellos pueden solicitar a cualquier hora, normalmente por la limitación de presupuestos. Los consumidores pueden también tener sus propias infraestructuras IT que generalmente no están completamente expuestas en Internet. Además, un consumidor participa en una utilidad de mercado a través de la gestión de recursos apoderados, que selecciona un conjunto de agentes basados en sus ofertas. Él entonces forma SLA's con los agentes que atan a éste a proporcionar los recursos garantizados. La empresa consumidora entonces despliega su propio entorno en los recursos alquilados o usa los interfaces de proveedor para escalar sus aplicaciones.

La idea de mercados utilitarios para recursos de cómputo ha estado siempre. Recientemente, muchos proyectos de investigación como SHARP [21], Tycoon [22], Bellagio [23] y Shirako [24] han sacado estructuras de mercado para el comercio en recursos asignados.

3.3.3. Enfoques para crear una SOA

En la tabla de la figura 16 se definen seis posibles enfoques a la hora de desplegar una arquitectura orientada a servicio [12].

Enfoque	Caracterización del Proyecto	Clasificación
Orientado a Procesos de Negocio	Los procesos de negocio necesitan explotar los recursos disponibles, y cada actividad requiere invocar una funcionalidad de IT. Para ello, cada funcionalidad debe estar disponible en una manera flexible.	TOP-DOWN
MDA (Model-Driven Architecture, basada en herramientas)	Se modela el negocio y luego las herramientas generan el detalle	TOP-DOWN
Empaquetado de Sistemas Legados	Se ha realizado una inversión importante en los sistemas existentes, pero éstos no son flexibles: no se les puede agregar funcionalidades en forma rápida, son sistemas estancos, con funciones "cautivas".	BOTTOM-UP
Composición de Sistemas Legados	Descomponer los sistemas legados monolíticos en módulos (manual o automático)	BOTTOM-UP
Orientado a Datos	Proveer acceso a los datos usando servicios, sin exponer esquemas o consideraciones de implementación	DATA-FOCUSED
Message-Driven	Se necesita integrar sistemas, comunicándolos mediante protocolos estándar no propietarios.	SERVICE-ORIENTED INTEGRATION of APPLICATIONS AND SYSTEMS.

Figura 16: Enfoques para la implementación de SOA.

3.4. SOA y los Web Services

Es importante no confundir Web Services con SOA. Web Services es una colección de tecnologías, incluyendo XML, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description, Discover and Integration (UDDI); los cuales permiten construir soluciones de programación para mensajes específicos y para problemas de integración de aplicaciones [10].

SOA es una arquitectura, es más que un conjunto particular de tecnologías como los Web Services. SOA debe ser una arquitectura de aplicación dentro de la cual todas las funciones están definidas como servicios independientes con interfaces invocables bien definidos, los cuales pueden ser llamados en secuencias bien definidas para formar los procesos de negocio. Que todas las funciones se definan como servicios incluye funciones de negocio, transacciones de negocio compuestas de funciones de bajo nivel y funciones de servicio de sistema. Que todos los servicios sean independientes quiere decir que ellos actúan como cajas negras, las funciones no saben cómo realizan la tarea pero devuelven el resultado esperado. Que los interfaces sean invocables quiere decir que a nivel de arquitectura es irrelevante si son locales o remotos, tanto da que sistema de conexión o que protocolo se utilice para la invocación, o que estructura de componentes son requeridos para realizar la conexión.

En SOA la clave está en el interfaz, éste define los parámetros requeridos y la naturaleza del resultado. Esto significa que define la naturaleza del servicio y no la tecnología utilizada. Esta función permite realizar dos de los puntos críticos: los servicios son realmente independientes y pueden ser manejados.

WS es el estándar apoyado por la industria (Microsoft, IBM, BEA, Oracle, Sun y otros), por empresas de distintos rubros, no tecnológicas (Ford, United Airlines, KPMG, Daimler-hrysler), agrupadas en un comité conocido como Web Services Interoperability, o WS-I. Este organismo tiene por principal objetivo asegurar que los grupos de trabajo que definen las especificaciones sobre WS utilizan estándares adecuados, a la vez que monitoriza el avance de sus trabajos; no define ni desarrolla estándares [12].

WS es un estándar construido a su vez en estándares como:

- WSDL (para describir contratos entre el consumidor y el proveedor de un servicio),
- UDDI (para descubrir servicios)



- SOAP (para invocar servicios)
- XML / HTTP (para la capa de transporte)

Y madurando bajo un esquema conocido como WS-*, que persigue:

- proveer seguridad a las conversaciones dentro de esta tecnología (en estado avanzado tanto en definición como implementación). Se sostiene con tres estándares:
 - WS-Security otorga elementos para firmar mensajes o encriptar (parte o todo).
 - WS-Secure Conversation contempla la posibilidad de acordar claves específicas entre las partes para conversaciones más largas.
 - WS-Trust para delegar relaciones de confianza a servicios distribuidos (mediante la creación, intercambio y validación de tokens de seguridad)
- mensajería confiable (un estado menor de avance)
- coordinación transaccional (un estado menor aún de avance)

3.5. Beneficios de desplegar SOA

Las dos ventajas más destacables de las tecnologías SOA son: por un lado, la reutilización del software que nos aporta beneficios sobre la calidad, los costes y facilidad para el despliegue. Por otro lado, la independencia de la plataforma tecnológica y las aplicaciones de la infraestructura.

SOA nos proporciona facilidad para evolucionar a modelos de negocio basados en tercerización, facilidad para abordar modelos de negocio basados en colaboración con otros entes (socios, proveedores), poder para reemplazar elementos de la capa aplicativa SOA sin disrupción en los procesos de negocio.

SOA nos ofrece la posibilidad de proporcionar un acceso consistente a toda la información relevante para la empresa, como un servicio de modo que pueda ser utilizado por cualquier aplicación, el problema es que a la hora de la implementación nos encontramos con barreras que dificultan la creación de un servicio abstracto sobre las múltiples fuentes de información de la empresa; se necesitan nuevas herramientas para poder definir una verdadera arquitectura de información de la empresa, que facilite el acceso en tiempo real a todo tipo de fuentes heterogéneas, tanto internas como externas.

Las organizaciones que tienen como enfoque el desarrollo alrededor de la creación de servicios a partir de las tecnologías existentes, combinados con enfoques basados en componentes para el desarrollo del software, obtendrán grandes beneficios como [10]:

1. Aprovechamiento máximo de los recursos existentes. Se pueden construir servicios de negocio a partir de componentes existentes, usando un entorno SOA adecuado y disponible para la empresa. Para el uso de estos servicios sólo es necesario conocer el interfaz y su nombre. Las implementaciones de servicios específicos o los componentes discretos funcionales son transparentes para el que los llama. El desconocimiento de estos componentes permite organizaciones que utilizan al máximo los nuevos descubrimientos, construyendo servicios compuestos por componentes contruidos a través de diferentes máquinas, funcionando en sistemas operativos diferentes, desarrollados con diferentes lenguajes de programación. Usando como interfaz Web Services, podemos encapsular y

acceder a los antiguos sistemas y éstos pueden ser transformados añadiendo valor, así como su funcionalidad se transforma en un servicio.

2. Infraestructura como comodidad. El desarrollo y despliegue de la infraestructura será más consistente a través de las diferentes aplicaciones empresariales. Los componentes existentes, los componentes nuevamente desarrollados y los comprados de diferentes vendedores pueden ser consolidados dentro de un entorno SOA bien definido, con los que la infraestructura se convierte en una comodidad.
3. Tiempo de implantación en el mercado rápido. Las librerías organizacionales de los Web Services reducen el tiempo de implantación en el mercado; como nuevas iniciativas, reutilizan los servicios existentes reduciendo el tiempo de diseño, desarrollo, testeo y despliegue en el proceso.
4. Reducción de costes. El coste para incrementar y crear nuevos servicios adaptando SOA y librerías de servicio tanto para aplicaciones existentes como nuevas, se reduce considerablemente.
5. Riesgo de migración. Al utilizar componentes existentes se reduce el riesgo de introducir fallos en el proceso de incrementar o crear nuevos servicios. También se reduce el riesgo de mantenimiento y manejo de la infraestructura soportando estos servicios.
6. Mejoras continuas en los procesos de negocio. SOA permite una representación clara de los procesos de flujo identificados por el orden de los componentes usados en un servicio de negocios particular y proporciona a los usuarios un entorno ideal para monitorizar operaciones de negocio. Los procesos de modelaje son reflejados en el servicio. Los procesos de manipulación son añadidos reorganizando las piezas en el proceso padre. Esta función permite cambiar los flujos de proceso mientras se monitorizan los efectos para facilitar continuas mejoras.
7. Arquitectura de proceso céntrico. El proceso es el que desarrolla la aplicación y éste se descompone en una serie de pasos los cuales representan un

servicio. Cada servicio o función componente es una subaplicación. Las subaplicaciones son encadenadas para crear un proceso de flujo capaz de satisfacer las necesidades. Esta granularidad permite a los procesos aprovechar al máximo y reutilizar subaplicaciones a través de la organización.

3.6. Implementando SOA

3.6.1. Componentes de SOA

Los elementos básicos necesarios para construir una arquitectura SOA son los siguientes:

- Operación: Es la unidad de trabajo o procesamiento en una arquitectura SOA.
- Servicio: Es un contenedor de lógica. Estará compuesto por un conjunto de operaciones, las cuales son ofrecidas a los usuarios. Los servicios son autocontenidos e independientes entre sí, con una tarea claramente definida cada uno.
- Mensaje: Para que un servicio pueda ejecutar una determinada operación, es necesario un conjunto de datos de entrada. A su vez, una vez ejecutada la operación, esta devolverá un resultado. Los mensajes son los encargados de encapsular esos datos de entrada y de salida que se envían los servicios unos a otros.
- Proceso de negocio: Son un conjunto de operaciones ejecutadas en una determinada secuencia (intercambiando mensajes entre ellas) con el objetivo de realizar una determinada tarea [25].

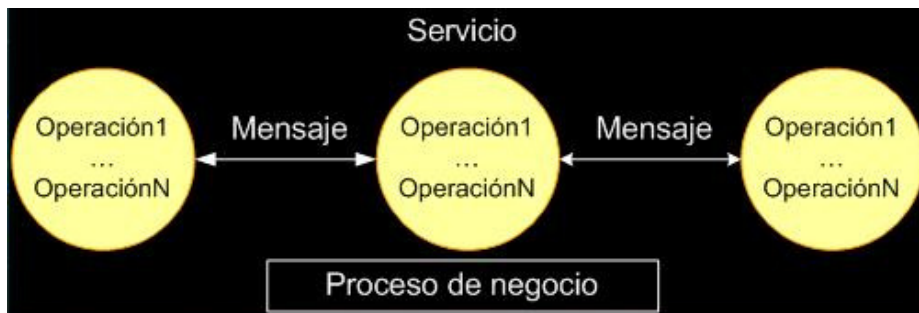


Figura 17: Elementos básicos de la arquitectura SOA.

3.6.2. Arquitectura SOA y tecnologías utilizadas

Una arquitectura SOA puede ser implementada usando cualquier tipo de infraestructura cliente/servidor. Las infraestructuras cliente/servidor son las más adecuadas debido a que nos proporcionan ciertas ventajas interesantes a la hora de implementar SOA como:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la compatibilidad del interfaz, y la facilidad de uso.

Las arquitecturas SOA están estructuradas en diferentes capas de software:

- Aplicaciones básicas - Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.
- De exposición de funcionalidades - Donde las funcionalidades de la capa aplicativa son expuestas en forma de servicios (generalmente como servicios web).
- De integración de servicios - Facilitan el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.
- De composición de procesos - Que define el proceso en términos del negocio y sus necesidades, y que varía en función del negocio.
- De entrega - donde los servicios son desplegados a los usuarios finales.

Para simplificarlo, se podrían resumir en tres:

1. El núcleo de la aplicación está formado por los Servicios Web que se comunican entre sí para llevar a cabo sus respectivas tareas y que conforman los procesos internos de negocio.

Los procesos de negocio se llevan a cabo ejecutando un conjunto de servicios en un determinado orden.

2. Para la orquestación o composición de estos servicios se puede emplear BPEL o SCA, que veremos más adelante.

3. Los interfaces de usuario se implementan como servicios a parte, ya que distintas empresas pueden usar los mismos Servicios Web, pero el interfaz que se le muestra al usuario final debe ser propia de cada empresa (interfaz corporativa), puede ser JSP, PHP, JSF, entre otros.

1. Web Services

La ruta que se ha definido en Arquitectura de Sistemas para el desarrollo de soluciones o sistemas informáticos es SOA (Arquitectura Orientada a Servicios), la cual está basada en la implementación de servicios de negocio, esto es; bloques funcionales de negocio que se pueden integrar, y compartir en distintas aplicaciones. La principal tecnología para implementar servicios SOA es Web Services. Muchas veces se confunde una Arquitectura Orientada a Servicios con los Web Services. SOA es una visión de arquitectura fruto de la evolución tecnológica y de las necesidades de negocio. Mientras que los Web Services son elementos, entre otros, que permiten su implementación y es lo que se recomienda, se promueve y exige, pero hay que tener cuidado, porque fácilmente se puede cometer errores, sino se asumen los siguientes principios [14]:

- Un Web Service no necesariamente es un servicio SOA.
- Un servicio SOA no necesariamente tiene que ser implementado como Web Service.

Para que un Web Service se considere un servicio SOA, debe cumplir con los estándares SOA, a grandes rasgos: debe corresponder a una funcionalidad del negocio bien acotada (self contained), independiente de la implementación tecnológica e independiente de los sistemas operacionales que lo soportan (loose coupling), y además debe poder ser compartido y reutilizado por otros procesos o sistemas (reuse). Por otro lado, un servicio SOA podría ser implementado con otras tecnologías, como HttpService, Mensajería, Ajax (DWR), etc, pero nuevamente para ser considerado como tal, debe cumplir los estándares SOA.

Según el W3C (World Wide Web Consortium) los Servicios Web son aplicaciones de software identificadas por un URI (Uniform Resource Identifier), cuyos interfaces y vínculos tienen la capacidad de estar bien definidos, descritos y descubiertos como objetos XML. WS soporta interacciones directas con otros agentes de software usando mensajes de intercambio basados en XML vía protocolos basados en Internet. Se puede perfeccionar esta definición pidiendo que la descripción se haga a través de un documento WSDL (Web Services Description Language) y el protocolo utilizado sea SOAP [26].

¿Y porque se debe implementar SOA usando Web Services?, en términos simples, porque es el estándar de facto, Web Services es una tecnología que opera en .Net y Java, que la mayoría de los motores de servicios contempla (bases de datos, ESB, ETL, etc), que las herramientas de diseño, desarrollo, y testing soportan, y que las grandes aplicaciones también manejan (CRM, ERP, SAP, Salesforce, etc.). Gartner de hecho tiene los Web Services situados en “plena productividad” para Arquitectura de Aplicaciones.

Los servicios SOA sirven para integrar sistemas, para construir procesos de negocio (orquestración de servicios), pero principalmente sirven para implementar aplicaciones Web. IBM lo ha promovido en el desarrollo de aplicaciones Web, en combinación con frameworks como Struts, Spring, lo ha promovido para el desarrollo de aplicaciones web “rich” (“Build rich Java Web applications with Apache Wink and Ajax” – Febrero 2010), y para el desarrollo de aplicaciones Web dinámicas (“Build RESTful Web services and dynamic Web applications with the multi-tier architecture” –Junio 2009). Oracle lo contempla en su Arquitectura de aplicaciones; Oracle ADF [14].

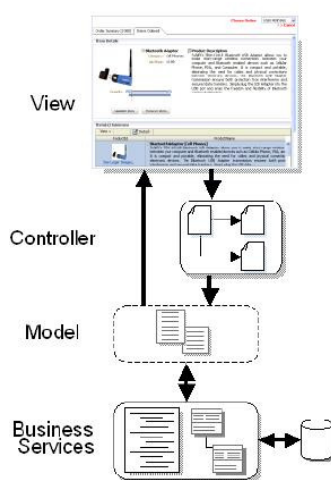


Figura 18: Arquitectura ADF de Oracle.

Pero como toda tecnología, los Web Services pueden ser mal implementados, si no se siguen los estándares y buenas prácticas, que existen para el desarrollo de Servicios. Uno de los problemas más comunes es pretender que toda funcionalidad de un sistema se puede convertir en un servicio (Web Service), y la verdad es que el enfoque es otro, las funcionalidades de Negocio, que pueden involucrar más de un sistema, son las que se deben convertir en Servicios.

Se han mencionado buenas prácticas y antipatrones para el desarrollo de servicios, las buenas prácticas apuntan a como se deben hacer las cosas, y los antipatrones apuntan a mostrar en que problemas no debemos caer, ejemplos:

- Servicio se nombra para maximizar consumo.

Erróneo: insertarRegistroCliente()

Correcto: crearNuevoCliente()

- Servicio tienen parámetros abultados (coarse grained).

Erróneo: crearNuevoCliente (rut, nombre, apellidos, email, fono, direccion)

Correcto: crearNuevoCliente (objetoCliente)

- Servicio encapsula detalles de implementación.

Erróneo : crearNuevoClientePsoft (schemaOracle, registroTablaCliente)

Correcto: crearNuevoCliente (objetoCliente)

- AntiPatrón, Servicio ParlanChines (Chatty Services).

Erróneo: consultaUF()

Correcto: (NO implementar ese tipo funciones como servicios)

¿Pero sólo basta con conocer los estándares y buenas prácticas para no equivocarse el camino en la implementación de servicios y para lograr los beneficios que promete SOA? La

verdad es que NO, se necesita más que eso, se pueden saber los patrones de diseño SOA, pero aún no interiorizarlos, o no poder lograrlo con tecnologías específicas como Java, porque sólo la experiencia real, y la supervisión continua en la primeras etapas logra que el área de TI adopte de forma adecuada SOA, y aquí es donde entra a jugar otro enfoque; “SOA Governance”, que corresponde a implementar el soporte en procedimientos, metodologías, y herramientas, para lograr un SOA exitoso:

- “SOA Governance ya no es una opción, es un imperativo, sin esta administración (governance) el retorno de la inversión es mucho menor, y todo proyecto SOA estará en riesgo”.
- “De hecho sin él (SOA Governance), muchas iniciativas SOA fallan”.

SOA Governance debe definir:

- Que Hacer: El plan global de proyecto SOA de la Empresa, define el “SOA Roadmap” (Plan de Ruta SOA).
- Quien lo Hace: La estructura organizacional (los grupos de trabajo), define la “SOA Office”.
- Como Hacerlo: Los procesos (procedimientos) de administración, las normas.
- Como Medirlo: Las métricas para medir el éxito

Si la implementación de servicios SOA falla, entonces es determinante la implementación de un “SOA Governance”, y el primer y más importante paso en Governance es establecer una área especializada; el “Centro de Excelencia SOA” (SOA COE), que permite enseñar y supervisar la adopción de SOA en una Empresa, un área clave liderada por Arquitectos de Sistemas [14]

Puesto que cada Servicio Web puede estar implementado en una tecnología heterogénea es necesario cumplir una serie de estándares para hacer posible la comunicación entre ellos. Los más utilizados son los siguientes [25]:

- Web Services Protocol Stack: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML: Es un lenguaje de marcado capaz de describir distintos tipos de datos. Es un estándar ampliamente aceptado y utilizado como medio de descripción de datos.
- SOAP: Es un protocolo de comunicación entre procesos basado en el intercambio de mensajes en formato XML dentro de una red. A su vez SOAP está basado en XML y es completamente independiente de la plataforma y del lenguaje en el que estén implementados los procesos que se comunican.
- WSDL: Es un lenguaje basado en XML que permite describir servicios web (como su nombre indica). Un documento WSDL especifica, entre otras cosas, dónde se encuentra el servicio así como las operaciones que pone accesibles a otros servicios.
- UDDI: Es un directorio, basado en XML, en el que las distintas empresas dan de alta servicios web que ponen al servicio de otra empresas.
- WS-Security (Web Service Security): Protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los “actores” y la confidencialidad de los mensajes enviados.

Los inconvenientes con los que nos encontramos son [27]:

-Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de cómputo distribuido como CORBA (Common Object Request Broker Architecture).

-Su rendimiento es bajo si se compara con otros modelos de cómputo distribuido, tales como RMI (Remote Method Invocation), CORBA, o DCOM (Distributed Component Object Model). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.

-Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en *firewall* cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

Por otro lado, las ventajas que aportan los Web Services son las siguientes:

-La interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.

-Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

-Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad de cambiar las reglas de filtrado.

-Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

-Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C, por tanto no hay secretismos por intereses particulares de fabricantes concretos y se garantiza la plena interoperabilidad entre aplicaciones.

Los beneficios de WS incluyen el desacoplo de los interfaces de servicio con las consideraciones de implementación y de las plataformas, la posibilidad de vínculos de servicios dinámicos y un incremento del cross-language y de la interoperabilidad de cross-platforms. Estos beneficios derivan del interfaz estándar XML y de las descripciones de acceso en WSDL. WSDL se utiliza para potenciar una arquitectura orientada a servicio permitiendo las preferencias como EAI (Enterprise Application Integration), integración de aplicaciones business-to-business y el grid computing [26].

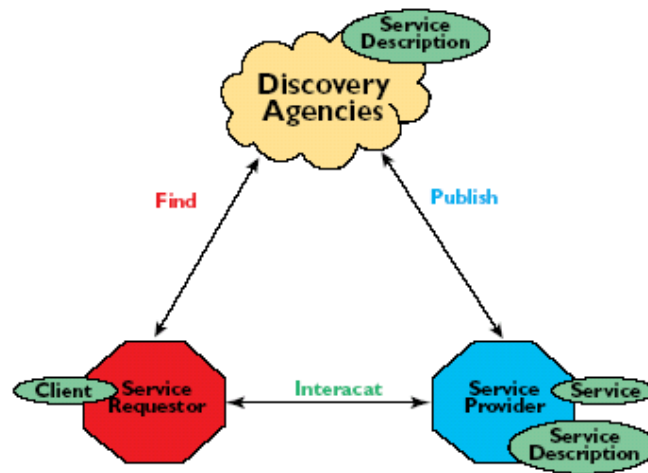


Figura 19: Diagrama de SOA.

En la figura 19 podemos ver que en una arquitectura orientada a servicio el “Service Provider” tiene los servicios diseñados para que otros los puedan utilizar. El proveedor crea una descripción de servicio (WSDL) que detalla el interfaz, esto incluye las operaciones del servicio y los mensajes de entrada y salida para cada operación. Después se crea la descripción de la implementación de los vínculos para los servicios, describiendo como se envían los mensajes por el cable en el que se encuentra localizado el servicio. Ahora WSDL ya contiene toda la información necesaria para invocar al servicio. Esta descripción es publicada por el proveedor a una o más empresas (Discovery Agencies); normalmente se realiza a través de un registro tipo UDDI que permite información adicional describiendo el hosting business y se hacen asociaciones con la taxonomía para ser publicadas junto con la descripción WSDL, así otros pueden encontrar el servicio usando una gran variedad de criterios de búsqueda incluyendo búsquedas basadas en categoría. Finalmente el “Service Requester” encuentra la descripción del servicio a través de las empresas (Discovery Agencies) y usa la descripción WSDL para desarrollar o configurar un cliente que interactuará con el servicio mediante el Service Provider.

2. Gestión de los procesos de negocio

Los trabajos sobre la composición de servicios son varios. Van desde la coreografía de servicios a la orquestación de servicios. La coreografía de servicios se centra en los mensajes entre actores (incluso los que no están identificados) que participan en los procesos de negocio y da un punto de vista abstracto de las interacciones de negocio pero no se centra en el proceso

de ejecución. La orquestación de servicios se dirige a los procesos de negocio a través de esquemas y organizaciones de las invocaciones de servicios.

Existen tecnologías dentro del ambiente SOA, y BPM, que tienen mucho en común, y por lo tanto muchas veces no se logra visualizar de forma clara sus diferencias, estamos hablando de los “coordinadores de tareas”, u “orquestadores” en general , herramientas que permiten organizar actividades en un flujo. Bajo este concepto podemos tomar [14]:

- ETL (Extract Transform and Load)
- WorkFlow tradicional
- ESB (Enterprise Service Bus)
- BPM o BPMS (Business Process Management Suite)

Estas herramientas en general permiten a través de asistentes gráficos crear un proceso a partir de la organización de tareas, o subprocessos.

Orquestadores en la Arquitectura de Referencia SOA

Desde el punto de vista de SOA, los orquestadores se organizan de la siguiente forma:

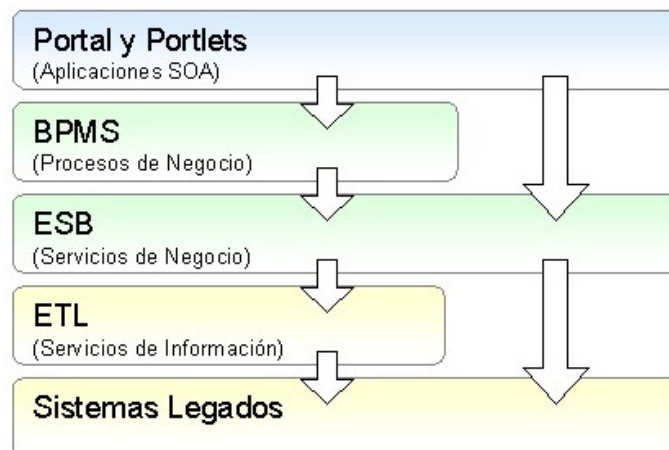


Figura 20: Esquema de los orquestadores desde el punto de vista de SOA.

Extract, Transform and Load (ETL)

Nacida para traspasar información desde un repositorio de datos origen a uno destino, esta tecnología está orientada a los procesos de datos, procesos basados en repositorios como Base de Datos y archivos.

ETL es una de las tecnologías más antiguas de integración, integración a nivel de datos, y tal como las Bases de Datos relacionales, su utilidad se ha consolidado en el tiempo. Incluso en la actualidad dentro de SOA tiene su puesto en lo que son los “Servicios de Información”, porque a pesar de que su principal uso es en procesos Batch (procesos de actualización masivos), también se puede usar para el desarrollo de funcionalidades específicas pero que involucran la integración de entidades de información dispares (distintas).

Tal como su nombre lo indica permite Extraer (Extract), Transformar (Transform), y Cargar (Load), información desde, y hacia distintas fuentes de datos.

- Extraer: permite extraer la información de distintas fuentes de datos, principalmente bases de datos, y de la mayoría de los proveedores conocidos: Oracle, Sybase, DB2, MySQL, esto lo hace directamente desde las tablas, o usando “Stored Procedures” existentes. También permite extraer datos de archivos como XML, Excel, CSV, Texto, entre otros. Y han evolucionado hasta incluso obtener información de colas de mensajes, objetos Java, y Web Services, abarcando de cierta forma el terreno de los ESB.
- Transformar: a través del manejo de Metadata (datos en un formato estándar desacoplado de la fuente) permite unir la brecha existente debido a la diferencia de los tipos de datos, por ejemplo el proceso de información “Actualizar Datos de un Cliente” que involucra actualizar en distintos lugares, puede requerir que un dato que está en entero deba pasarse a tipo float (real), o que los apellidos que están separados en una base de datos deban guardarse juntos en un solo campo en otra base de datos, o que un código de comercio que está en cierto estándar (como EAN) se deba traducir a otro estándar (como ISBN) porque así se maneja en el otro repositorio.
- Cargar: permite guardar la información procesada en la misma diversidad de entidades que maneja dentro de la extracción: Bases de Datos, y archivos de distinto tipo.



En la actualidad las herramientas de ETL permiten componer estos procesos mediante asistentes gráficos, entornos de desarrollo (IDE) que realizan una introspección de las fuentes de datos, obteniendo todos los objetos y sus campos de datos, permitiendo a través de “arrastrar y soltar” construir el flujo de proceso de datos. Además pueden incorporar reglas de negocio que permiten potenciar la funcionalidad de los procesos.

Los principales objetos que maneja un ETL son:

- Tablas de Bases de Datos
- Procedimiento Almacenados (Stored Procedures)
- Archivos XML
- Archivos de Texto, Archivos Excel.
- Transformaciones de Datos.

Otra característica que ofrecen actualmente los ETL, y que es muy importante para la arquitectura SOA, es que los procesos ETL se pueden ejecutar como Web Services, para lo cual el motor de ETL publica el proceso en formato SOAP, pudiendo ser ejecutado desde cualquier plataforma (Java, .Net), y ser integrado en tecnologías como ESB (Enterprise Service BUS).

Luego, en el contexto de SOA podríamos indicar que ETL es la herramienta que permite diseñar y construir procesos de información (servicios de información) basados en actividades (automatizadas) a nivel de datos.

WorkFlow Tradicional

Decimos WorkFlow “Tradicional”, porque la forma actual y bajo SOA de implementar un workflow es BPM, pero existen aún tecnologías WorkFlow tradicionales (no BPM), pero la mayoría se está transformando en soluciones BPM.

La verdad es que actualmente muchas herramientas WorkFlow se acercan a lo que es BPM, viéndolo sólo desde el punto de vista del flujo de los procesos de negocio, por eso aquí vamos a ver el concepto de WorkFlow Tradicional que se refiere al WorkFlow como era antes de evolucionar a BPM.

WorkFlow (Tradicional) es la tecnología que permite coordinar actividades humanas (realizadas por personas, Human Task), aquí se definen roles, actividades, reglas de negocio, en buenas cuentas un flujo de trabajo (WorkFlow), pero que a diferencia de BPM no contempla en forma natural los “servicios de negocio”, es decir no contempla las actividades de sistemas (System Task) bajo el estándar SOA.

Luego en una solución (WorkFlow tradicional) el participante (usuario) debe indicar el cumplimiento de la tarea (cambio de estado), para de esta forma continúe el flujo de trabajo, y cuando debe interactuar con algún sistema, debe levantar dicho sistema y realizar la actividad en él, por ejemplo ejecutar el sistema de stock para agregar un nuevo producto.

Esa es otra gran diferencia con BPM, en BPM se trata de proveer una sola interfaz para un participante del proceso, ocultando la interacción con los sistemas, mientras en un WorkFlow (tradicional) la persona debe interactuar con distintos ambientes o aplicaciones, dicho de otra forma: la persona debe manejar distintas aplicaciones (sistemas), y además registrar su avance en el WorkFlow.

Con WorkFlow (al igual que BPM) se le da seguimiento y control a los procesos de negocio, es decir, podemos saber el estado actual de cada proceso, en qué lugar del flujo se encuentra. Otra similitud con BPM, o dicho de otra forma; otra característica que BPM heredó de los WorkFlow, es que a través del proceso generalmente fluye información (documentos, datos), lo que se llama metadata, u objeto de negocio (BPM).

Enterprise Service Bus (ESB)

Esta tecnología permite integrar sistemas a nivel de servicios, cuando se habla de tecnología para implementar SOA en una empresa la primera herramienta que se debe incorporar es el “Service Bus”.

Esta herramienta permite componer procesos a partir de servicios SOA, los servicios SOA son componentes funcionales encapsulados fuertemente reutilizables, haciendo el “típico” símil: son piezas de lego que permiten construir un sistema uniendo bloques, y en este caso construir un proceso uniendo servicios SOA.

Al igual que BPM absorbió las funcionalidades de WorkFlow (a través de BPEL4People), también absorbió las funcionalidades del ESB (a través de BPEL), de hecho las BPMS (BPM Suite) suelen traer como parte de sus componentes el ESB.

La diferencia con BPM es que el ESB no maneja las actividades humanas (Human Task) sólo maneja actividades automatizadas (System Task).

La diferencia con ETL (tradicional) es que ETL está orientado a componentes de datos, y ESB está orientado a componentes de Servicios.

Los objetos que principalmente maneja ESB son:

- Web Services
- Conectores a Colas de Mensajes, y JMS
- Conectores a Aplicaciones de clase Mundial (PeopleSoft, SAP, JDEdwards)
- Conectores a J2EE (EJB) y .Net
- Transformaciones de Datos

Los ESB también manejan Metadata para la transformación de datos, y para lograr que los sistemas se entiendan a pesar de la diferencia en los tipos de datos que manejan.

Una característica importante es que los procesos se pueden ejecutar a través de WebServices, es decir un proceso de negocio ESB se puede publicar en protocolo SOAP, permitiendo que una gran variedad de plataformas puedan acceder a estos procesos.

Algunos ESB tienen conectores para Base de Datos, permitiendo integrar consultas a tablas (SQL Query), y procedimientos almacenados (Stored Procedures), con esto el ESB comienza a abarcar terreno de los ETL, pero a decir verdad esta característica pone en riesgo la orientación a servicios (SOA), ya que los servicios deben corresponder a la capa de negocio, y no a la de datos.

Resumiendo ESB desde el punto de vista SOA es la herramienta que permite desarrollar procesos, basados en servicios SOA.

BPM Suite

BPM es la solución que permite construir procesos de negocio basándose en coordinar tanto las actividades interactivas (human task) como los servicios (system task). BPM une lo mejor del mundo del WorkFlow Tradicional y del ESB.

En un mismo flujo se integran tareas realizadas por personas, con tareas automatizadas (sistemas), entregándole al participante de un proceso de negocio una sola interfaz, ocultando la interacción con los sistemas legados.

Las otras características de valor agregado de BPM son: que ofrece un grupo de herramientas para atender todo el ciclo de vida de un proceso de negocio, dentro de las cuales se destaca, el modelador (diseño procesos), el IDE (desarrollo de servicios), y el monitor (BAM), este último permite realizar un seguimiento y control más acabado de los procesos de negocio.

BPM o Business Process Management son las actividades que tienen como objetivo el análisis, diseño, ejecución y monitorización. La monitorización se hará a través del BAM o Business Activity Monitoring, permite el control de los procesos de negocio.

BPM es la solución a una serie de retos planteados como:

- Solucionar el problema de la distribución de procesos en múltiples aplicaciones construidas sobre diversas plataformas tecnológicas.
- Traducir la lógica de negocio de una organización definiendo sus flujos de interacciones manuales y automáticas de forma completa.
- Dotar de dinamismo, respondiendo a la demanda de los clientes y a los cambios en las condiciones de mercado.
- Soportar la larga duración, una instancia de un proceso puede permanecer activa durante meses o incluso años.

Desde una perspectiva de gestión es un enfoque estructurado que emplea métodos, políticas y métricas para gestionar y optimizar de manera continua las actividades y procesos de una organización. Desde el punto de vista tecnológico, agrupa una serie de herramientas software para el modelado, ejecución y monitorización de los procesos de negocio.

BPM estandariza los interfaces entre componentes tecnológicos, reduce los costes de integración de los sistemas, y provee de lógica de negocio a nivel de proceso interconectando servicios reutilizables.

Permite encadenar los procesos para ganar en eficiencia y asegurar la mejora continua de los mismos. También permite la modificación rápida en función de la demanda cambiante y reduce los costes de mantenimiento.

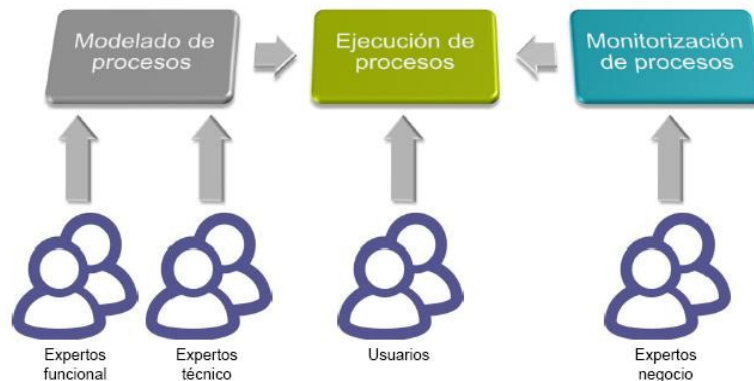


Figura 21: Esquema de integración de BPM.

Sabemos que los Web Services son la tecnología más apropiada para implementar arquitecturas SOA, las cuales representan procesos de negocio adecuadamente. BPM permite la implantación de un proceso de negocio utilizando SOA. Una suite BPM debe dar soporte al modelado, ejecución y monitorización de procesos.

Un proceso consta de diversas tareas ejecutadas en un flujo controlado:

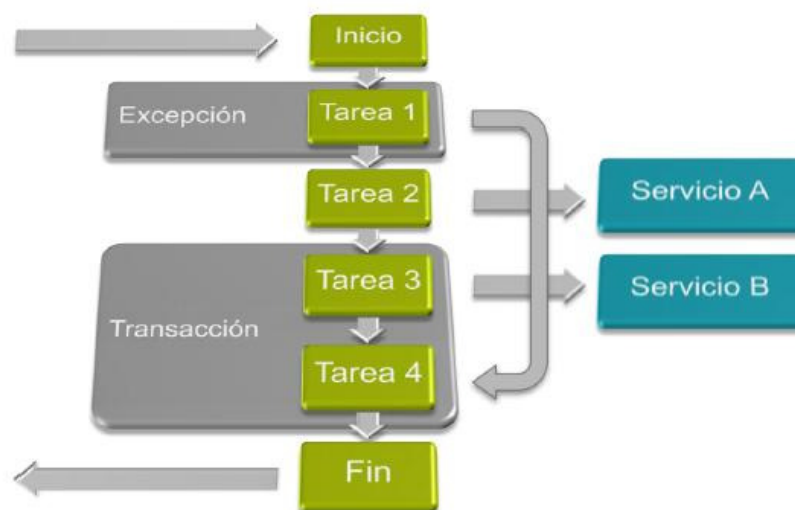


Figura 22: Flujo de un proceso.

Modelado de los procesos

Las soluciones BPM de modelado de procesos deben proveer a los expertos funcionales de [28]:

1. Capacidad para capturar procesos ya existentes de manera estructurada mediante algún tipo de notación (BPMN) que permita representar las relaciones existentes entre distintos departamentos y/o sistemas involucrados en los mismos.

2. Capacidad para definir nuevos procesos, o realizar modificaciones sobre los ya existentes utilizando la misma notación que en el caso anterior.

3. Simulación de parámetros de proceso (tiempo de ejecución, costes, throughput), en función de las variables independientes del mismo.

4. Automatización de la documentación del proceso modelado en formato fácilmente exportable.

5. Plantillas de procesos predefinidos.

Desde el punto de vista de los expertos técnicos, las soluciones BPM de modelado de procesos deben proveer por ejemplo de:

- Facilidades para agilizar la importación y exportación de modelos analíticos creados por expertos funcionales a modelos aplicativos ejecutables.
- Capacidades de control de flujo (ej. instanciación de subprocesos, toma de decisiones en base a reglas predefinidas).
- Soporte a eventos.
- Gestión de excepciones.
- Definición de alarmas.
- Adecuación a lenguajes estándares de definición de procesos como BPEL o SCA.

Ejecución de procesos

Habitualmente, un motor de ejecución de procesos:

-Ofrece diversos mecanismos de invocación de procesos de manera síncrona, por ejemplo vía Web Services (publicando el WSDL), RMI.

-Ofrece diversos mecanismos de invocación de procesos de manera asíncrona, por ejemplo vía colas (IBM MQSeries, JMS), recepción de emails.

-Permite la invocación de procesos desde otros procesos (subprocesos).

-Hace posible la definición de transacciones (definición de nodos o conjuntos de nodos de un proceso como recursos).

-Permite el versionado de procesos.

-Tiene en cuenta la escalabilidad (ej. posibilidad en crecimiento en número de usuarios y recursos), el rendimiento (ej. permite la ejecución de miles de procesos diferentes de manera concurrente) y la fiabilidad (ej. soluciones 24x7).

Monitorización de procesos

La monitorización de procesos permite obtener información de negocio (Business Activity Monitoring, BAM) y técnica, con el fin de identificar patrones de utilización, situaciones de riesgo, indicadores de desempeño, etc. Además del almacenamiento y agregación de datos a partir de key performance indicators (KPIs), las soluciones de monitorización deben tener en cuenta [28]:

- La usabilidad de la aplicación de consulta para expertos de negocio. El perfilado según tipos de usuarios. Los métodos de envío de alertas (ej. email).
- Los mecanismos de captura de información, ofreciendo la capacidad de procesar datos generados por diversas fuentes, incluidas aquellas ajenas al propio BPM (ej. otras aplicaciones no BPM).
- El rendimiento de la solución, minimizando el impacto en la escalabilidad y consumo de recursos en entornos de producción.

3. Orquestación de los servicios

BPEL

La orquestación (o composición) de servicios web se basa en un modelo centralizado en el cual las interacciones no se realizan directamente entre los servicios web sino que existe una entidad encargada de definir la lógica de interacción. El lenguaje BPEL permite definir la lógica de orquestación entre los diferentes servicios web [29].

BPEL es la unión entre negocio y tecnología, al ser un estándar usado por los fabricantes podemos elegir entre distintas plataformas, permite la interoperabilidad y fomenta la competitividad y la mejora de las plataformas. Básicamente consiste en un lenguaje basado en XML diseñado para el control centralizado de la invocación de diferentes servicios Web, con cierta lógica de negocio añadida, que ayudan a la programación en gran escala (programming in the large). El consultor de negocio usa el lenguaje BPMN (Business Process Modeling Notation) para el modelado de los procesos, y será luego traducido a lenguaje máquina (BPEL) para ser ejecutado.

Se puede resumir [30]:

- ❖ BPM: es el proceso total.
- ❖ BPMN: es la parte que se usa para representar el proceso.
- ❖ BPEL: el código ejecutable del proceso.
- ❖ BAM: la parte del BPM que permite la monitorización.
- ❖ SOA: la arquitectura que permite implementar BPM con servicios. Su diseño es responsabilidad de los arquitectos informáticos.
- ❖ Web Services: permiten que los servicios se integren en un proceso de manera estándar. Responsabilidad de los desarrolladores.

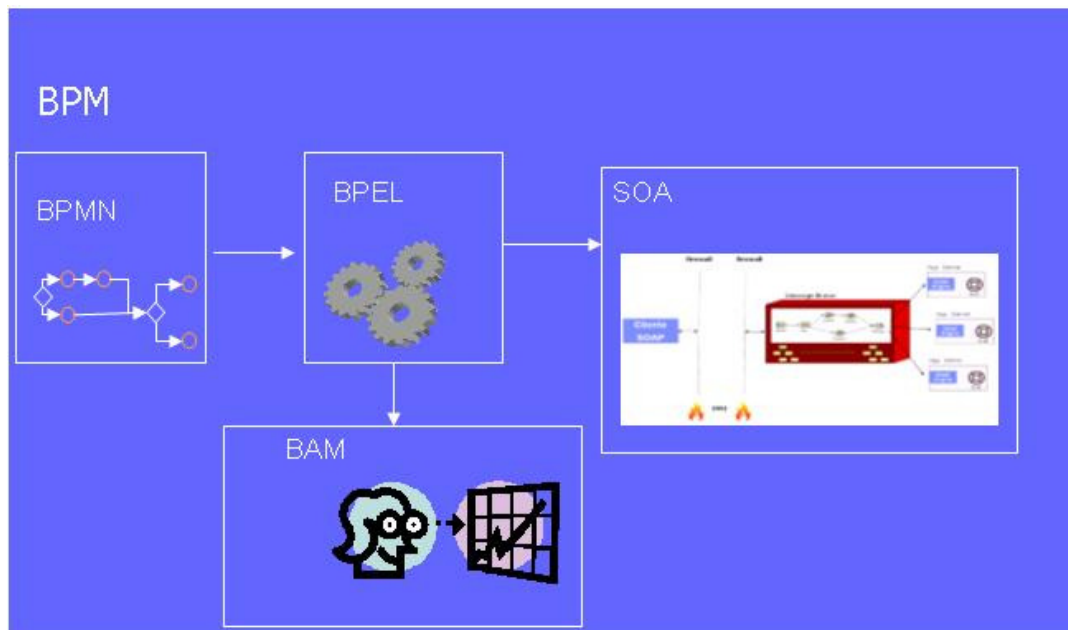


Figura 23: Esquema BPM, BPMN, BPEL, BAM y SOA.

SCA

BPEL resulta muy útil para la orquestación de servicios pero resulta insuficiente para la definición de procesos de negocios complejos debido a [31]:

- No soporta macro-flujos con intervención humana
- No incluye mecanismos para integrar a los flujos componentes que no sean servicios

SCA es un nuevo estándar creado por BEA e IBM para superar las limitaciones de BPEL. Con SCA se pueden crear procesos complejos que incluyen diversos componentes (procesos BPEL, interacción con humanos, motores de reglas, etc.). SCA utiliza SDO (Service Data Objects JSR 235) para representar los datos XML que fluyen por el proceso.

Una de las principales ventajas de adoptar SCA y BPEL es que la lógica de negocios queda separada de la programación de los componentes, esto va a permitir que poco a poco sean los usuarios finales quienes modelen la lógica de negocios. Progresivamente irán apareciendo también nuevas herramientas que permitan monitorizar el desempeño de los procesos y así se cerrará el círculo virtuoso del BPM [32].

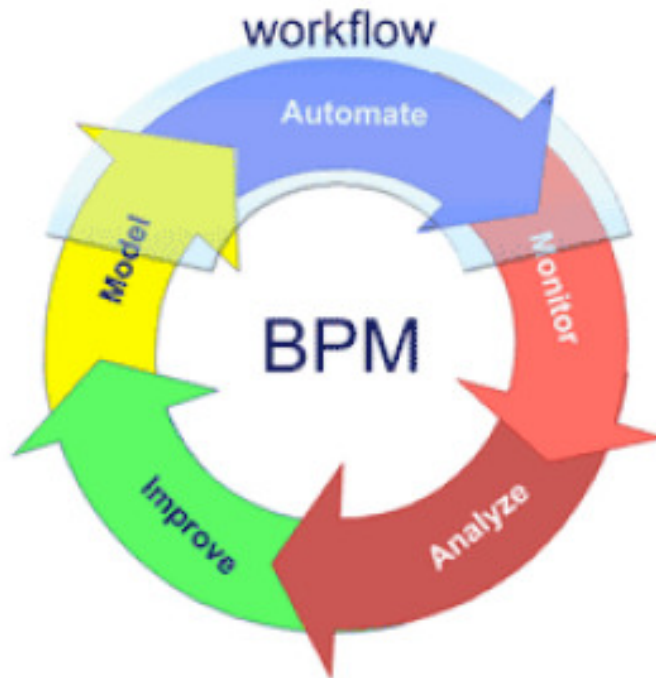


Figura 24: Flujo de BPM.

SCA no sustituye a BPEL sino que lo complementa [31].

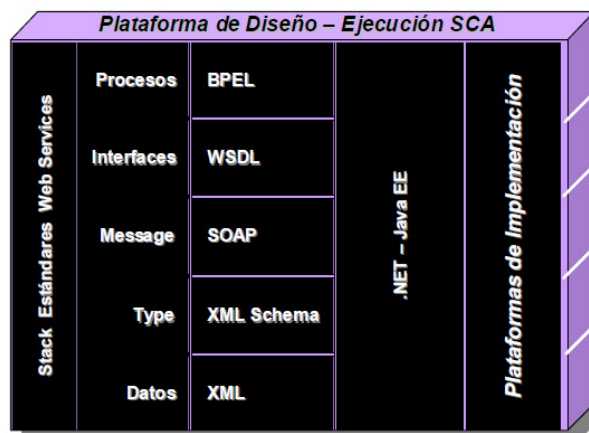


Figura 25: Plataforma de diseño. Ejecución de SCA.

Las características principales de BPM son [30]:

- Debe permitir gestionar el ciclo de vida de los servicios.
- Simular procesos de negocio.
- Monitorizar KPIs (key performance indicators) de los procesos de negocio.

-Agilidad en el cambio de los procesos.

4. Arquitectura de referencia SOA

Uno de los aspectos relevantes en SOA es definir la Arquitectura de Referencia para la Empresa, esta definición permite tener un marco de referencia en donde ubicar los nuevos desarrollos.

La Arquitectura de Referencia SOA plasma los distintos componentes de una solución SOA, principalmente Procesos de Negocio y Servicios, además muestra cómo interactúan estos componentes con los usuarios de negocio, y con los sistemas existentes en la Empresa (sistemas legados) [14].

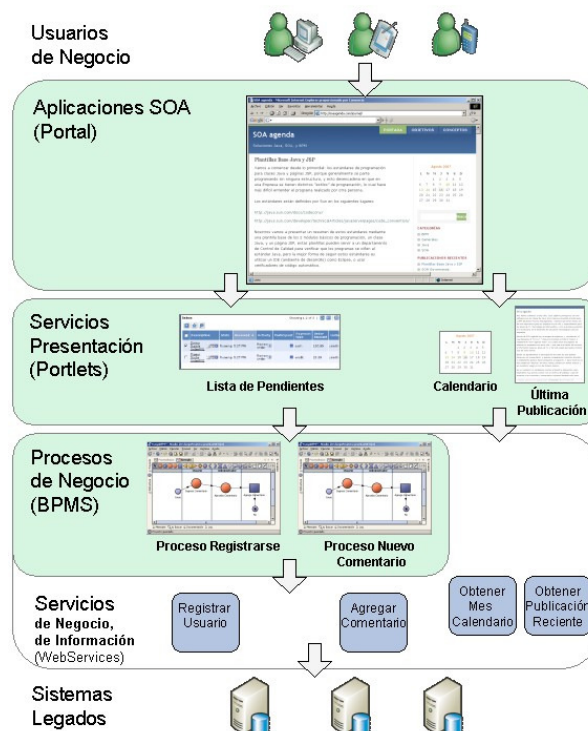


Figura 26: Arquitectura de referencia SOA.

Esta Arquitectura debe ser complementada con los componentes específicos de cada Empresa. Además cada proveedor de soluciones (IBM, Oracle, BEA, etc.) tiene su propia Arquitectura SOA de Referencia, que incorpora sus herramientas específicas, pero toda Arquitectura de Referencia por lo menos contempla lo siguiente:

- Usuarios de Negocio: son los usuarios de las aplicaciones, pero en SOA son también los participantes de los procesos de negocio, estos pueden utilizar distintas tecnologías para acceder a la aplicación (o proceso de negocio): Desktop, Notebooks, PDAs, móviles.
- Aplicación SOA y Portal: Las aplicaciones (aplicaciones SOA, o aplicaciones compuestas), están implementadas usando componentes reutilizables (Portlets, y Servicios), para lo cual se utiliza la tecnología de Portales. Una aplicación de este tipo incorpora todas las funcionalidades de un proceso bajo un entorno común. La ventaja principal de las soluciones de Portal es que una aplicación desarrollada para un dispositivo se puede ajustar a otro con muy poco esfuerzo, es decir, una aplicación que funciona en un Desktop se puede adaptar para que se vea en una PDA, ajustando los portlets, y su distribución para cada dispositivo.
- Servicios de Presentación (Portlets): son los componentes de presentación reutilizables, que en la práctica corresponden a secciones reutilizables de las páginas Web. Ejemplos: un portlet de “Calendario”, un portlet para mostrar las “Publicaciones Recientes” de un blog. En el caso de los “Procesos de Negocio” (BPMS) generalmente ofrecen un portlet para ejecutar los procesos, al que llamaremos portlet “Lista de Pendientes”.
- Procesos de Negocio: son la implementación BPM de los procesos, son procesos que incorporan tareas interactivas (interacción participante), con actividades automatizadas (servicios). Ejemplo: el proceso de “publicar un comentario en un Blog”, que dentro de sus tareas interactivas está el “ingresar el comentario”, y “aprobar el comentario para su publicación”, y una actividad automatizada es el servicio de “ingresar el comentario en el sistema de Blog”.
- Servicios de Negocio: son componentes funcionales del negocio que se pueden reutilizar en los distintos procesos, y distintas aplicaciones, generalmente son servicios compuestos (por otros servicios). Ejemplo “ingresarComentarioBlog”.

- Servicios de Información: son los servicios atómicos que pueden ser parte de servicios de más alto nivel. Su principal característica es que acceden directamente a los recursos, o sistemas legados, encapsulan las funcionalidades específicas de los sistemas existentes, dándole así una interfaz que permita integrarlos al estándar SOA.
- Sistemas Legados: son los sistemas existentes en la Empresa, que no están integrados (sistemas silo o isla). Son los que soportan actualmente la operación de negocio, y que no están bajo el nuevo esquema de “orientación a servicios”.

5. SOA, AON y AUTONOMIC COMPUTING

La red de nueva generación permitirá disponer de diversas aplicaciones multimedia, con diferentes requisitos de QoS, sobre una infraestructura común de transporte basada en IP. Sin embargo, junto con el servicio de consolidación sobre IP y el rápido aumento de la escalabilidad de la red, la tarea de servicio y gestión de la red se ha vuelto extremadamente compleja y costosa. Se están realizando esfuerzos en diversas áreas, por ejemplo, SOA [33], redes orientadas a aplicación [34], y autonomic computing [35] para lograr un entorno de servicio de entrega escalable, seguro, y autogestionable para acortar el tiempo de mercado de nuevas aplicaciones de Internet, así como reducir los gastos de gestión de los proveedores de servicios. Sin embargo, estas tecnologías no son desarrollables de forma coordinada para la óptima escalabilidad, utilización de recursos, y el desempeño de QoS. En un ASMF (entorno de servicio de gestión autónomo), un negocio o aplicación de gestión con requisitos de QoS end-to-end puede ser compuesto automáticamente de un servicio estándar de componentes de manera distribuida y gestionado autónómicamente según acuerdos de servicios [36].

Internet está experimentando una fase de dramática evolución impulsada por los nuevos modelos de computación distribuida como grid computing, redes peer-to-peer (P2P), y búsqueda Web. Junto con la tendencia de consolidación de servicio sobre IP (Protocolo Internet), Internet se está convirtiendo en un sistema extremadamente complejo y la gestión de la red se ha convertido en uno de los problemas más importantes. Se están realizando esfuerzos en diversas áreas, por ejemplo, service-oriented architecture (SOA), application oriented networking (AON), y autonomic computing, con objetivos de reducir la complejidad de desarrollo, reducir el coste de la gestión, y acortar el tiempo de mercado de las nuevas aplicaciones de Internet. Aunque SOA,

AON, y autonomic computing pueden facilitar cada creación de servicio, entrega, o gestión, desde diferentes perspectivas, las tecnologías no son desarrolladas de forma coordinada para una escalabilidad, una utilización de recursos, y un rendimiento de calidad del servicio (QoS) óptimos. ASMF se construye sobre la base de la idea de que todo es un servicio. Actualmente, los Internet Service Providers (ISP) tradicionales están ampliando paulatinamente sus negocios para dar servicio a aplicaciones de capas superiores, y, por tanto, sus infraestructuras están evolucionando más allá de la conectividad a abarcar contenido, procesamiento y almacenamiento. En un amplio sentido de la red ISP, los recursos y los servicios convergen en la misma comprensión genérica, lo que denota que cualquier capacidad que puede ser compartida y explotada en un entorno de red, incluidos tanto los tradicionales recursos de computing/networking como los servicios [37]. En consecuencia, los servicios de administración y gestión de recursos han significado lo mismo. La perspectiva genérica de los servicios es compatible también con los esfuerzos de la gestión utilizando servicios Web (MUWS) y la gestión de los servicios Web (MOWS), dirigido por el comité de Gestión de Servicios Web Distribuidos (WSDM) OASIS. La capa de núcleo funcional de ASMF es AAEF (Autonomic Application-Enabling Fabric), que se formó como una red P2P superpuesta, autogestionable constituida por ASB (Autonomic Service Brokers). La superposición de ASB contiene una base de datos distribuida de registro de servicio y permite la construcción automática de aplicaciones como una QoS end-to-end garantizada de acuerdo con el principio de la SOA de una manera distribuida. Además, la AAEF no sólo soporta acuerdos de servicios (SLA) basados en una gestión autónoma sino que también integra la tecnología AON para optimizar el diseño de la superposición de ASB y mejorar la entrega de mensajes en el sistema.

SOA

En SOA, como ya hemos comentado, diversos programas de software, dispositivos informáticos, y recursos de redes están encapsulados vía interfaces normalizados comunes como componentes del servicio loosely-coupled; cada componente de servicio publica su ubicación y descripción del servicio a un registro (también llamado un service broker). El servicio en SOA es provisionado de acuerdo con el lema “find, bind and execute” como muestra la figura 27, y los servicios complejos o aplicaciones pueden ser ensamblados bajo demanda consultando el registro y localizando y combinando los componentes de servicio requeridos. En SOA, un

componente de servicio puede ser fácilmente reutilizado/cedido para diferentes funciones y convenientemente acondicionados en una aplicación más avanzada [38].

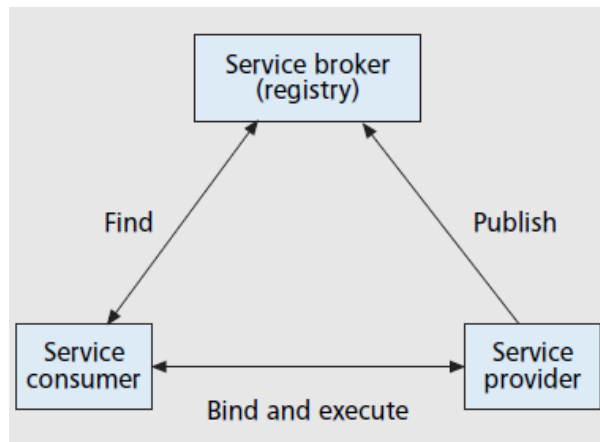


Figura 27: Paradigma SOA, Find, Bind and Execute.

UDDI es el actual estándar de la industria para implementar el registro de servicios en SOA. UDDI se propuso como una entidad central, pero la preocupación por la escalabilidad ha motivado el estudio del diseño distribuido conectando múltiples registros localizados o brokers con las tecnologías de superposición peer-to-peer [39]. La aplicación permite también estar formada como una red superpuesta de broker, donde se ha de debatir las importantes cuestiones abiertas como cómo organizar la topología de superposición para facilitar respuesta al servicio de consulta, cómo buscar un conjunto de servicios correlados que satisfagan un requisito de QoS end-to-end, y cómo negociar SLA's y gestionar los recursos en un enfoque distribuido. El éxito del despliegue de SOA requiere un mecanismo confiable y capaz de realizar composición de servicios de manera automática. La mayoría de las soluciones actuales de composición están basadas en plantillas [33] y en enfoques basados en semántica [40], pero, por lo que sabemos, la mayoría de las soluciones actuales de composición requieren una aplicación centralizada, lo cual limita considerablemente la escalabilidad. En particular, uno de los estos enfoques, es BPEL, el cual está convirtiéndose en el estándar de la industria. El proceso BPEL está centralizado en un solo documento BPEL que invoca a los componentes de servicio requeridos en función de un flujo de trabajo de composiciones predefinido [33]. En la propuesta ASMF, la composición de servicios y la invocación del mismo han de ser manejados por la red superpuesta de brokers de una manera distribuida. SOA es un modelo de referencia, ejecutado por diferentes técnicas. Actualmente, la aplicación SOA basada en Web services se está convirtiendo en popular y se está normalizando en las comunidades de grid computing y Web computing, donde el servicio definición es descrito en eXtensible Markup Language (XML) y

presentado en forma WSDL. En SOA, los mensajes intercambiados entre servicios son generalmente en formato XML y comunicados por el protocolo SOAP.

AON

La implementación eficiente de SOA requiere una potente mensajería troncal, que es una de las principales motivaciones para integrar la tecnología AON iniciada por Cisco. Una red basada en AON puede interceptar de manera transparente el contenido y el contexto de mensajes de aplicación, realizar operaciones en esos mensajes según las políticas y normas, y hacer la inspección y procesamiento de la información. El coste y la complejidad asociada con el despliegue SOA disminuirán significativamente al incorporar la aplicación de servicios de mensajería dentro de la red, en lugar de middleware stacks [34]. Aunque AON actualmente no tiene una solución específica, en este ejemplo, se coge una interpretación orientada a la aplicación redes: los dispositivos IP pueden interceptar y tramitar la solicitud de mensajes.

En la actualidad, no está claro cómo aprovechar la capacidad AON para mejorar la red y la gestión del servicio. La mensajería existente de middleware SOA, corriendo sobre una mejor infraestructura de redes, debe implicar vigilancia de los recursos, QoS routing, localización del problema y planes de recuperación para asegurar la adecuada prestación de servicios. Con AON, el sistema de mensajería está incorporado en el backbone, donde el control de la disponibilidad de recursos y QoS de la información son mucho más fáciles de acceder que en la capa de aplicación, lo que se espera es que el encaminamiento de mensajes en AON pueda ser simplificado considerablemente en comparación con las soluciones de middleware.

AUTONOMIC COMPUTING

La característica fundamental del autonomic computing es la gestión automatizada de recursos de cómputo, que abarca las características de auto-configuración, auto-optimización, auto-saneamiento, y autoprotección. Un sistema autónomo es una colección de elementos autónomos. Cada elemento autónomo consta de un gestor autónomo (AM) y del recurso manejado (MR) (Figura 28).

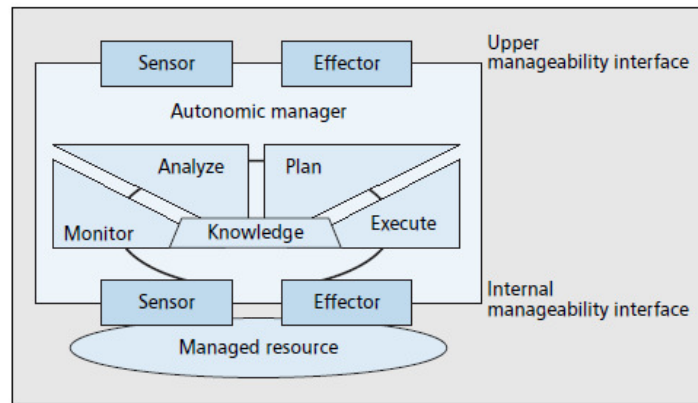


Figura 28: Arquitectura para un elemento autónomo.

El AM opera de acuerdo con un bucle de control “monitorizar, analizar, planificar y ejecutar”, con conocimiento del entorno de cómputo y las políticas de gestión. La comunicación entre la AM y el MR se hace a través del interfaz de gestión del MR, que se compone de un “sensor” y un “efector”. El sensor es utilizado para obtener datos del MR, y el efector se utiliza para realizar operaciones sobre el MR. El administrador autónomo también proporciona interfaces de sensor y efector para otros gestores autónomos, de modo que cada elemento autónomo a sí mismo puede ser tratado como un recurso gestionable para el recurso o la composición de servicios. Para comodidad podemos diferenciar las interfaces como interfaz de gestión superior e interfaz de gestión interna. Para formar un auto-sistema, todos los elementos autónomos son orquestados juntos bajo políticas de nivel alto [35].

La arquitectura basada en autonomic computing y SOA, son ambos modelos de referencia establecidos basados en componentes. Por naturaleza, las dos arquitecturas pueden integrarse perfectamente: cada elemento autónomo puede encapsularse en un interfaz de servicio Web estándar para formar un componente de servicio autónomo, y el principio de SOA “find, bind & execute” y los esquemas de mensajería asociados pueden utilizarse para orquestar los componentes de servicio autónomos en un auto-sistema. IBM que fue el iniciador de la arquitectura basada en autonomic computing, ya propuso un interfaz manejable de un elemento autónomo de acuerdo con el estándar WSDM de OASIS. Sin embargo, para nuestro conocimiento, los estudios del sistema de integración existente no han considerado ni la cuestión de la composición distribuida de servicios ni la integración con AON.

3.6.3. SOA en redes ópticas

Internet nos introduce varias limitaciones a la hora de introducir nuevos servicios, o para dar evoluciones de los ya existentes. Por ejemplo, un dominio único no nos proporciona un

establecimiento de conexiones end-to-end a través de diferentes sistemas autónomos (Ases-Autonomous Systems). La colaboración entre dominios es esencial para poder soportar servicios más sofisticados que demandan conexiones de largo recorrido (long-haul) [41].

Uno de los problemas más importantes cuando consideramos conexiones interdominio es como satisfacemos el TE (Traffic Engineering) y otras obligaciones locales. Los clientes que necesitan conexiones end-to-end a través de múltiples dominios deben tener mecanismos para elegir la ruta de interdominios que tiene en cuenta los parámetros de TE como el ancho de banda y el retardo (pueden ser usados para calcular el coste de las conexiones). Actualmente, los protocolos de enrutamiento de interdominios como BGP (Border Gateway Protocol) no llevan ningún tipo de información de TE. La única tecnología existente hoy en día que proporciona mecanismos que soporta la provisión de conexiones en redes ópticas es GMPLS (Generalized Multiprotocol Label Switching). Mientras GMPLS es una solución predominante ya que está muy estandarizada y es soportada en las industrias, su desarrollo cuando consideramos un escenario de interdominios, resulta estrechamente ligado al plano de control. Esto quiere decir que los planos de control de diferentes interdominios deben interactuar para hacer posibles las conexiones. Esta interacción es necesaria a efectos de compartir información de la topología y la asignación de recursos. Para soportar la provisión de servicios end-to-end de una manera completa y automática es necesario que cada dominio administrativo desarrolle GMPLS. Éste es un requerimiento muy grande ya que el dominio debe elegir el mecanismo para establecer las conexiones internas que mejor se adapte a las necesidades. Además, un dominio puede establecer conexiones a través de funciones de gestión sin la necesidad de un plano de control.

Una propuesta para el aprovisionamiento de servicios a través de multidominios es en vez de tener un plano de control GMPLS, tener una capa de servicio para las interacciones interdominio. El plano de control dentro de cada dominio, ofrece un conjunto de servicios para instalar o borrar las conexiones de red. La capa de servicio es la encargada de controlar todas las funciones relacionadas con la provisión de servicios interdominio end-to-end (enrutamiento, señalización, control de admisión, política de aplicación y negociación de interdominios). La capa de servicio de interdominio nos da un dominio para definir su propio conjunto de interfaces de servicio para interactuar con otros dominios. Sólo un conjunto de funciones comunes soportadas por los interfaces deben ser estandarizadas, ya que es más fácil estandarizar interfaces que estandarizar protocolos. Las dos principales funciones necesarias para proporcionar servicios interdominio son las funciones de enrutamiento y señalización. El

enrutamiento consiste en la publicación de topologías virtuales entre dominios, mientras que la señalización consiste en la negociación end-to-end de interdominios. El conjunto de servicios y sus interfaces resultan una solución con pérdida de acoplamiento al contrario que el plano de control, con lo que los costes para soportar la provisión de interdominios se reducen cuando lo comparamos con una solución GMPLS. Desde que la provisión de servicios depende básicamente de la definición de los interfaces en la capa de servicio, no es necesario mejorar la infraestructura de redes (hardware y software) para soportar los protocolos GMPLS. El plano de servicio abstrae los detalles que hay por debajo de como es creada la provisión de conexiones por cada proveedor de red. Esta capa se implementa usando Web Services, tecnologías más adecuadas para las arquitecturas orientadas a servicio (SOA) [42]. EL objetivo principal de SOA es ayudar a las organizaciones a conducir sus negocios hacia un SOE (Service-Oriented Enterprise). En SOA cada entidad lógica es vista como un servicio. Los servicios pueden definirse como primitivos (self-sufficient) o como compuestos (dependiente de otros servicios). Una red de servicio puede ser creada a través de la composición de un conjunto de servicios primitivos, un ejemplo de composición será el interdominio VPN óptico (O-VPN). La principal idea es facilitar la interacción entre las redes de cliente (ej. Redes IP/MPLS) y las redes de proveedor ópticas, así como entre proveedores de redes ópticas. Adoptando Extensible Markup Language (XML) para el intercambio de datos y Web Services Description Language (WSDL) para la descripción de los interfaces de servicio, hace que la integración de las diferentes aplicaciones y las interacciones entre los diferentes dominios administrativos sean fácilmente añadidas sin necesidad de seguir especificaciones complejas.

Cada dominio puede definir sus servicios y publicar sus interfaces WSDL en el registro de servicios Web. La propuesta también está dirigida a dar soporte a un nuevo tipo de modelo de negocio, los VCa's (Virtual Carriers). Los VCa's se pueden ver como portadores (carriers) intermediarios virtuales (carrier's carriers) que ofrecen servicios especializados para clientes de baja inversión. Este descubrimiento permite a los portadores indicar una red óptica en una subred de VCa específica, dando a cada VCa diferentes niveles de gestión de control, incluyendo operaciones, administración, mantenimiento y aprovisionamiento (OAM&P). Se cree que esta propuesta puede contribuir a la extensión de servicios de mercado de longitudes de onda ópticas, facilitando la venta, compra y gestión de longitudes de onda, y lo más probable es que actúe como un catalizador de oportunidades de mercado.

➤ Web Services y la gestión de redes ópticas

Una de las primeras arquitecturas en las que se aplicaron Web Services para la gestión de las redes ópticas [43], consideró que los usuarios podían contratar y gestionar sus propios recursos (lightpaths). Mediante los sistemas de gestión, los usuarios pueden concatenar, dividir, publicar/contratar, y establecer end-to-end lightpaths. Esta propuesta no considera las obligaciones impuestas por cada dominio desde el establecimiento del camino end-to-end, no tiene en cuenta las políticas del dominio.

Se han propuesto soluciones a este problema, para asegurar que las reglas de cada dominio son respetadas [44]. Para ello se utiliza una arquitectura basada en Web Services para establecer el camino end-to-end considerando las políticas de utilización en la admisión y en el control de la reserva de recursos.

Otra opción es la que reúne condiciones de las dos anteriores [41], ya que utiliza Web Services, pero teniendo en cuenta el aprovisionamiento de los servicios del interdominio vía composición de servicios y negociación entre dominios. Además, el control y la gestión de los caminos no son dados a los clientes. El proveedor tendrá el control de los recursos.

Otra propuesta es el servicio GMPLS orientado a cliente [45], el objetivo es proveer una arquitectura de gestión para la diferenciación de capacidades basada en el modelo de referencia Telecommunication Management Network (TMN). Esta propuesta no considera gestiones de multidominio.

➤ Propuestas para el aprovisionamiento de Servicios de interdominio

Aunque el aprovisionamiento de servicios de interdominio es un tema de gran interés hoy en día, no hay reglas comunes de cómo satisfacer los requerimientos de dichos servicios. Una de las razones es la carencia de modelos de negocio que dirigen la facturación de interdominios, la asignación de recursos, y los SLA's (Service Level Agreements).

MESCAL propone una arquitectura para soportar QoS de interdominios. Esta idea depende de las extensiones del protocolo BGP, que puede ser un proceso a largo plazo sin la garantía de que se convierta en un estándar [46].

En otras referencias los autores discuten cómo soportar servicios IP/MPLS interproveedor [47, 48]. Al mismo tiempo, las escrituras del IETF sobre el Common Control and Measurement Plane (CCAMP) definen un marco para el establecimiento y el control de Label Switch Paths (LSP's) MPLS y GMPLS en redes multidominio [49].

CCAMP tiene en cuenta la arquitectura PCE (Path Computation Element) como una caja negra para anunciar y calcular caminos entre interdominios [48]. El grupo de trabajo PCE es el encargado de definir la arquitectura para el cálculo de caminos para MPLS y GMPLS Label Switch Paths (LSP's). De todos modos, hay muchos puntos que discutir y normalizar como la elección del mejor camino entre interdominios, el protocolo de comunicación entre PCE's y cómo soportará BGP la información de QoS.

También encontramos referencias, que presentan el concepto de L1VPN (Layer 1 VPN), no obstante, los autores no consideran el aprovisionamiento de servicios L1VPN en escenarios multidominio [50].

Todos estos trabajos, no consideran la idea de tener topologías virtuales siendo anunciadas entre dominios ópticos.

Otros nos introducen el mecanismo para aprovisionar los recursos con redes de servicios virtuales [51] y también se presentan algunos esquemas para GMPLS Virtual Private Networks (GVPN) [52]. El servicio GVPN describe servicios VPN que cuentan con BGP para auto-discovery de VPN y con GMPLS para la señalización [53].

➤ Aprovisionamiento de servicios interdominio

Una de las arquitecturas propuestas para el aprovisionamiento de servicios interdominio está en la línea del plan de trabajo de CANARIE donde la red es tratada como un conjunto de servicios [54]. Esto ofrece más funcionalidad y flexibilidad para los ISP's (Internet Service Providers). Los protocolos de asignación de ruta OSPF (Open Shortest Path First) y BGP pueden ser expuestos como servicios. Los servicios pueden ser contruidos desde cero o desde sistemas heredados usando envolturas [55]. WSDL traza las funcionalidades de cada servicio dentro del interfaz que puede ser libremente definido en base a cada proveedor. Esta propuesta no necesita seguir el proceso estrecho y a largo plazo de estandarización. Los servicios son definidos por los proveedores y después registrados en un registro público o privado. Entonces, los servicios pueden ser descubiertos, vinculados y ejecutados dinámicamente.

Hay autores que discuten el futuro de Internet y sugieren que es necesario rediseñar Internet completamente [56]. Como alternativa, los autores consideran que el número de ideas que ya existen pueden juntarse para resolver los cuellos de botella actuales. Por ejemplo, usando soluciones actuales como los Web Services, es posible proveer de servicios más complejos y sofisticados sobre la infraestructura de red ya existente. La red ya no puede ser vista como una simple infraestructura física, pero sin fisuras como parte de una aplicación distribuida [54]. Desde que las soluciones que se encuentran en la extensión de BGP para ofrecer QoS no han evolucionado, podemos centrarnos en topologías virtuales para soportar QoS de interdominio. A través de la publicidad de las topologías virtuales, cada dominio tendrá un gráfico entero de la red y entonces podrá aplicar algoritmos de cálculo de caminos para encontrar el camino óptimo para un par de interdominios fuente-destino. Al mismo tiempo esta solución puede tener en cuenta las políticas de BGP para satisfacer los contratos de interconexión de cada dominio. Los lighthpaths son establecidos dentro de cada dominio óptico, es una decisión local. Un dominio puede usar un protocolo específico (ej. GMPLS) y seguir políticas locales para abrir y cerrar las conexiones. La arquitectura se puede ver como un conjunto de servicios primitivos y compuestos. Esta es una alternativa para GMPLS con una clave diferente, el establecimiento de servicios e2e (end-to-end) entre dominios se construyen a partir de Web Services, no por señalización GMPLS. Esta propuesta libera a los proveedores de la espera de la estandarización desde que las especificaciones de GMPLS para extensiones de protocolo para permitir alcances de dominios y anuncios de TE, han sido convertidos en RFC [49]. Por último, el enfoque presentado se centra en un escenario regional. Este escenario regional está formado por "condominios de dominios" por los que un grupo de dominios están de acuerdo en anunciar topologías virtuales a los otros. Esta publicación sigue un modelo de interconexión donde todos los dominios que forman parte del mismo condominio obtienen topologías virtuales de otros dominios. Estos condominios de dominios podrían definir reglas de negocio para crear nuevas relaciones que permitan interacciones orientadas más a cliente. Dar más poder de decisión a los clientes fomentará competencia entre ISP's imponiendo diferentes disciplinas económicas y ofreciendo mejores servicios a bajo coste. Está consensuado que el monopolio no sea conducido por el cliente si no por el proveedor. Si los usuarios finales pueden elegir la ruta de dominio observando precios y la calidad de servicio, los ISP's tendrán que hacer frente a una presión competitiva para impulsar el despliegue de buenos y nuevos servicios con el fin de atraer y mantener clientes.

➤ Identificación de los servicios

Existen diversas tecnologías para el cálculo distribuido como CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Model), y Java RMI (Remote Method Invocation), estas tecnologías adoptan un modelo de comunicación fuertemente acoplado y síncrono (petición/respuesta) y muestran un bajo nivel de interoperabilidad. Por otro lado, usando estándares XML, protocolos de Internet como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), y SMTP (Simple Message Transfer Protocol), y Web Services ofrecen un modelo de comunicaciones de acoplamiento débil y asíncrono con un alto grado de interoperabilidad.

La figura 29 muestra el paradigma de ejecución de SOA (ya visto anteriormente), en el cual los proveedores registran sus servicios en un registro público o privado y los consumidores consultan este registro para encontrar servicios que tengan en cuenta ciertos criterios. Después de obtener la dirección de los servicios los clientes pueden enlazar e invocar los servicios.

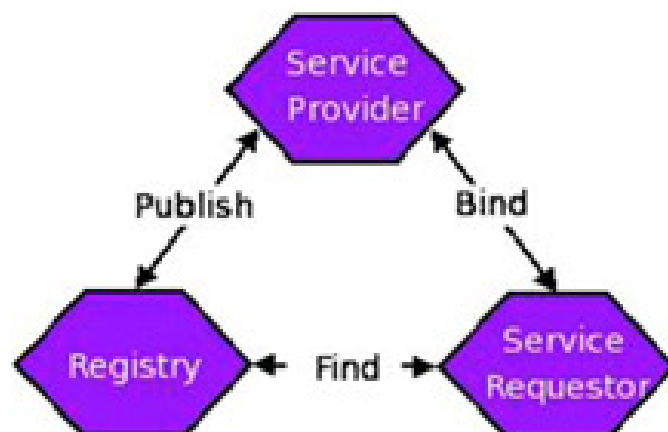


Figura 29: Paradigma SOA.

La estrategia a seguir es identificar los servicios primarios que son necesarios para soportar servicios más sofisticados (figura 30).

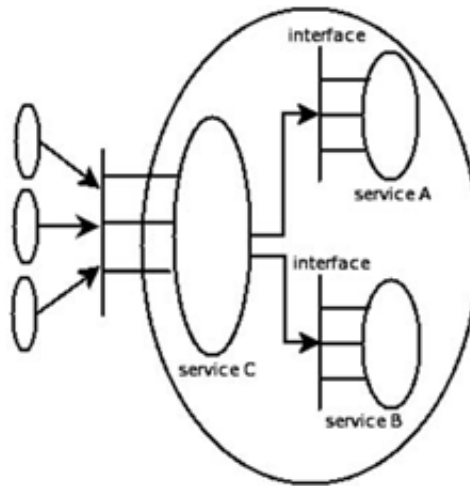


Figura 30: Composición de servicios.

Los servicios A y B son servicios primitivos de nivel bajo. El servicio C se construye a partir de la composición o agregación de dos servicios primitivos. Si un servicio compuesto es un servicio por sí mismo, este mecanismo puede ser aplicado de forma recursiva para crear otros servicios compuestos. La composición de servicios es la clave para encontrar nuevos caminos para diseñar, desarrollar y gestionar servicios de red [55].

Los servicios definidos en la secuencia representan las tareas que son necesarias para soportar interacciones de interdominios a fin de establecer conexiones e2e y O-VPN's. Hasta cierto punto, cada servicio incorpora las funcionalidades proporcionadas por la señalización GMPLS y los protocolos de enrutamiento. AS (Advertising Service) es un servicio de red primario para publicar la topología de red de cada dominio a los otros dominios (función de enrutamiento). El segundo servicio de red primario es el servicio de negociación de interdominios e2e (E2ENS). Este es el responsable de realizar las negociaciones entre dominios con el fin de establecer las conexiones entre interdominios e2e (función de señalización). El servicio de conexión entre interdominios e2e (E2ECS) ofrece un interfaz desde el cual los clientes piden las conexiones interdominio, es un servicio de red compuesto ya que depende de E2ENS para realizar sus tareas. Finalmente, dos servicios más son definidos para soportar servicios de interdominio O-VPN, el TS (Trading Services) responsable de reservar recursos entre los dominios y el servicio O-VPN responsable de activar/desactivar y monitorizar el O-VPN. Estos dos servicios están contruidos a partir de la composición de los descritos anteriormente, por ejemplo, TS usa E2ENS para reservar los recursos para los O-VPN's. También se define un servicio responsable de encontrar rutas de interdominios (función de enrutamiento), el servicio

PCE (Path Computation Element) que sigue las especificaciones definidas por el IETF para este elemento [48].

La arquitectura también comprende módulos internos para gestionar peticiones externas y cumple con las políticas locales. Estos módulos junto con los módulos de Web Services reúnen todas las funciones necesarias para proporcionar conexiones inter e intradominio [57].

- Arquitectura para provisión de servicios interdominio basada en SOA.

El plano de gestión de la arquitectura está dividido en dos partes lógicas (figura 31). La primera parte, o capa de servicio, agrupa los Web Services que son responsables de ofrecer a las entidades externas (clientes y otros dominios) un interfaz para acceder a las funcionalidades del sistema de gestión [41].

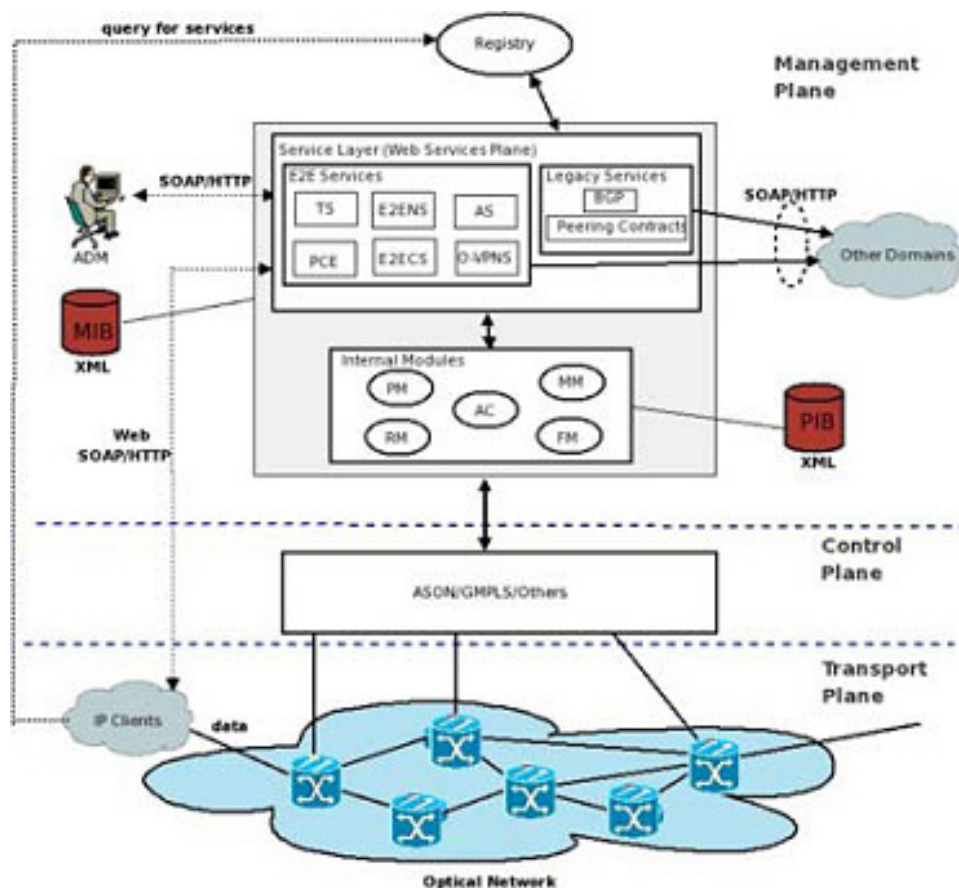


Figura 31: Prototipo de arquitectura.

Los Web Services forman la arquitectura SOA y abstraen los detalles de cómo se implementan las tareas internas (figura 32). De esta forma el dominio óptico se ve como un conjunto de Web Services primarios o compuestos, siendo cada uno responsable de dar sus

funcionalidades específicas. La capa de servicio se divide en dos partes, los servicios e2e y los servicios heredados. Los servicios e2e se construyen desde el principio y realizan interacciones de interdominios e2e, mientras que los servicios heredados exponen sistemas heredados como los sistemas de enrutamiento y de facturación, como Web Services.

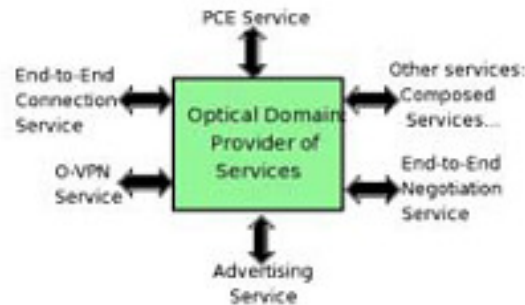


Figura 32: Dominio óptico visto como un grupo de Web Services.

La segunda parte lógica del plano de gestión está relacionada con los módulos internos que son necesarios para gestionar el control y la admisión de las peticiones así como la aplicación de las políticas locales.

La capa de servicio

La capa de servicio proporciona un interfaz que es invocado por otras aplicaciones (ej. El navegador Web u otros servicios). El administrador del dominio modela los servicios, define su comportamiento y publica el interfaz en un registro, así los servicios pueden ser consultados y accedidos por entidades externas. Los Web Services en la capa de servicio son descritos en la secuencia.

1. AS

El servicio de publicación es el responsable de publicar la topología virtual a otros dominios. Esta topología virtual representa los caminos (lightpaths) que cruzan el dominio óptico. Los detalles de cómo se construyen los lightpaths están escondidos para los clientes. Adoptando esta propuesta, la QoS de interdominio proporcionada a los servicios es posible desde que cada dominio tiene una topología virtual de todos los otros dominios y un algoritmo CSPF (Constraint Shortest Path First) suficiente para encontrar la ruta más apropiada para cumplir las obligaciones e2e. Para publicar la topología virtual el administrador del dominio tiene que definirla. Un dominio

dato puede tener diferentes topologías virtuales que son publicadas siguiendo unas reglas específicas como la variación de la cantidad de tráfico durante el día, los servicios que son ofrecidos, disponibilidad de los recursos, y la presencia de fallos o cuellos de botella. Por ejemplo, el administrador del dominio puede definir un estándar de topología virtual para ser usado bajo condiciones normales y otras topologías virtuales son publicadas cuando se detectan unas condiciones específicas. Usando esta propuesta el administrador de dominio podrá definir reglas teniendo en cuenta estrategias de negocio. La topología virtual puede tener un tiempo de expiración, forzando su actualización continua. Los clientes pueden contratar el AS definiendo reglas para cuando la topología virtual sea publicada y que nivel de información se va a publicar junto a la topología virtual.

En la figura 33 podemos ver dos dominios y sus topologías virtuales y la publicación entre ellos. Sólo se muestra un solo coste para simplificar. La selección del camino entre dominios se calcula considerando la topología virtual publicada por los dominios ópticos.

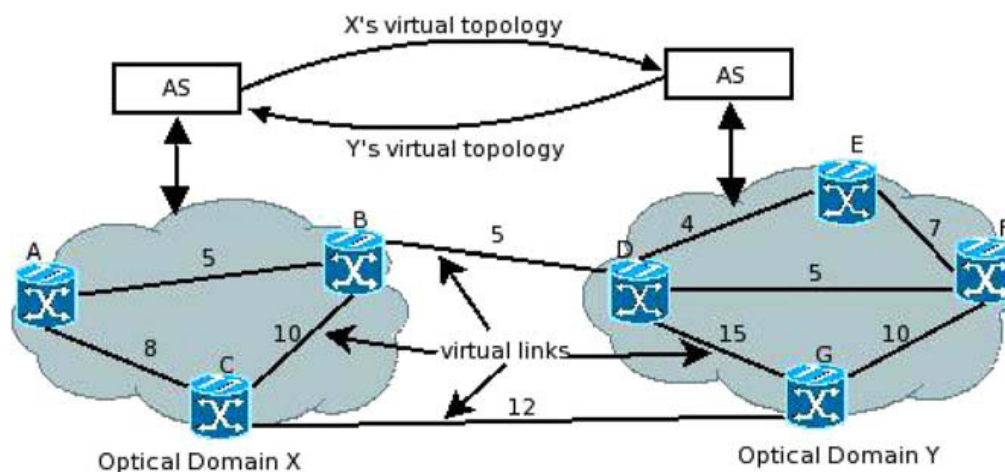


Figura 33: Publicación de topologías virtuales.

Cada conexión virtual representa un conjunto de recursos (lightpaths) que pueden ser usados para establecer conexiones e2e. La cantidad de lightpaths debajo de cada conexión virtual es una decisión del dominio local y depende de los recursos físicos disponibles para cada dominio óptico. El coste de cada conexión virtual no cambia al ritmo que son consumidos o liberados los lightpaths. Este coste refleja el coste para usar un único lightpath en la conexión virtual. Cuando no hay más lightpaths en una conexión virtual dada, se pueden crear más entre

los dos nodos. Cómo y cuándo crear los lightpaths pertenece al administrador del dominio local. El AS y el PCE juntos realizan el enrutamiento en el plano de servicio.

2. E2ENS

Este servicio es el responsable de negociar las conexiones interdominio e2e con otros dominios. Se puede adoptar un modelo basado en dos fases, por el cual el dominio “head-end” (dominio donde se ha recibido la solicitud de servicio) negocia con los otros dominios buscando un recurso disponible (lightpath). La primera fase consulta y reserva el recurso. Durante la primera fase, los parámetros de tráfico son analizados en cada dominio con la finalidad de verificar si la conexión puede ser aceptada. La segunda fase confirma la reserva en el caso de que todos los dominios envueltos en la negociación tengan el recurso requerido. Si uno de los dominios no tiene el recurso requerido, la reserva tiene que liberarse. Actualmente usamos el ancho de banda, el nivel de protección y el tipo de tráfico (High Priority (HP) and Low Priority (LP)) como parámetros de tráfico de QoS para ser negociados. Cada dominio es responsable de encontrar un recurso teniendo en cuenta las limitaciones locales. E2ENS no toma ninguna decisión de cómo se conduce la selección de recursos en cada dominio óptico. El servicio sólo puede llevar los parámetros pasados por el cliente en la solicitud de conexión. La topología virtual da un punto de vista general sobre el enrutamiento entre dominios. La diferencia entre la publicación de la topología virtual y la cantidad actual de recursos en cada dominio óptico es resuelta por E2ENS. El servicio de negociación es usado no solamente para llevar los parámetros del tráfico real requeridos por el cliente, sino también para confirmar que todavía existen recursos debajo de la conexión virtual y para reservar los recursos en cada dominio óptico. E2ENS realiza la función de señalización en el plano de servicio.

3. E2ECS

Este servicio compuesto es invocado por los clientes con el fin de preguntar por una única conexión interdominio e2e. Primero, el servicio necesita encontrar una ruta interdominio e2e invocando el servicio PCE. Hay que destacar que para encontrar una ruta el servicio PCE depende del AS. La ruta interdominio es calculada a través de la topología virtual de cada dominio. Finalmente, E2ECS invoca a E2ENS para negociar los recursos con los dominios implicados. Después de realizar estas fases y considerando que la conexión e2e se establece a través de los dominios, el cliente puede empezar a enviar datos a través de la conexión.

4. EL SERVICIO PCE

El servicio PCE es invocado por los módulos internos de su dominio. Es el responsable de encontrar una ruta interdominio de modo que la conexión interdominio e2e pueda establecerse. Como la topología virtual de cada dominio es publicada por AS del dominio, encontrar una ruta de interdominio es sólo aplicar el algoritmo de enrutamiento. El PCE no es sólo el responsable de encontrar una ruta si no de obtener la topología virtual de los otros dominios [48,58]. El mecanismo publicado puede ofrecer funcionalidades específicas para ciertos tipos de servicios como O-VPN, y esta modularización permite desarrollar cada módulo independientemente. Aunque el servicio PCE está siendo actualmente invocado sólo por módulos locales, éste puede representar una región o área dada. Otros dominios que no tienen dicho servicio pueden usar el servicio PCE de un tercer proveedor. Este es un típico escenario SOA, donde los servicios son ofrecidos, solicitados y usados.

Finalmente para tener un escenario SOA completo es necesario tener un mecanismo donde los servicios sean publicados y encontrados. El Registry es el módulo donde el end-point de cada servicio es almacenado. Esta arquitectura no utiliza el repositorio de UDDI (Universal Description, Discovery and Integration) ya que UDDI tiene más funcionalidades de las necesarias [59]. El registro actúa como un directorio privado donde los interfaces de servicio son registrados y proporciona un mecanismo para solicitar estos servicios.

La figura 34 muestra como se realiza el cálculo orientado a servicio. El dominio publica su servicio en el registro a través de un interfaz Web y recopila información sobre el servicio (nombre, descripción, interfaz end-point y el servicio end-point). Un usuario dado solicita la disponibilidad del servicio en la base de datos Registry y recupera un interfaz WSDL y el endpoint del servicio que mejor se adapta a la solicitud. Como un interfaz WSDL es self-describing, esto hace posible generar en tiempo de ejecución la infraestructura necesaria para invocar al servicio. Finalmente, la aplicación del cliente invoca al servicio a través de la aplicación de cliente usando SOAP (Simple Object Access Protocol) sobre HTTP.

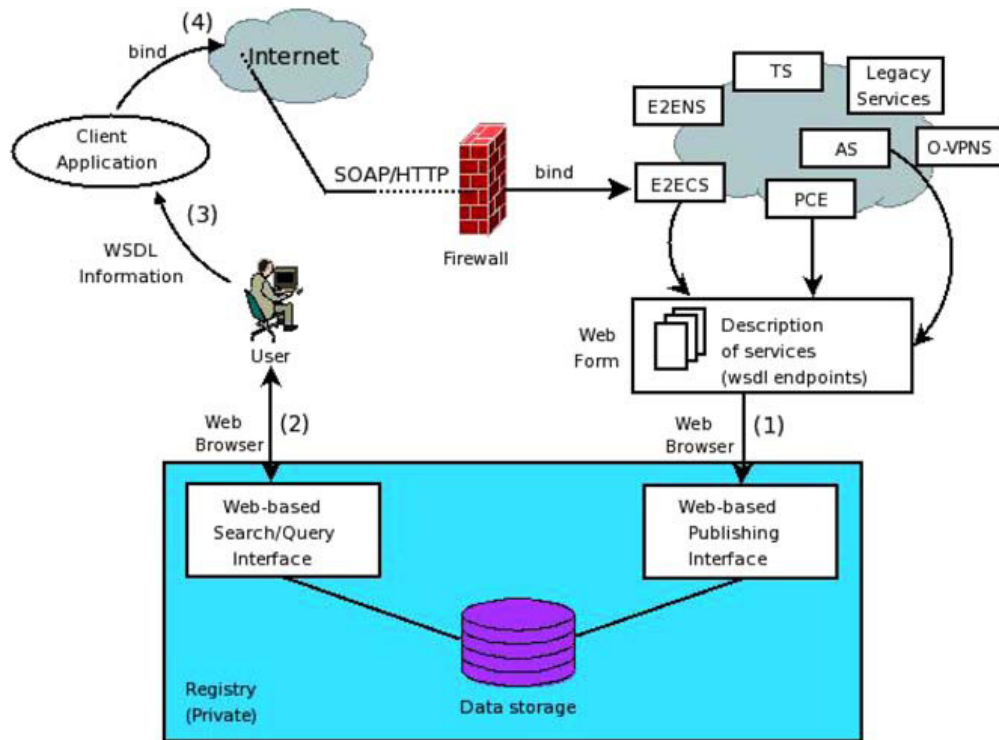


Figura 34: Escenario de arquitectura SOA.

Módulos Internos

Los módulos internos realizan tareas de los dominios locales para poder proporcionar los servicios. El módulo control de admisión (AC) recibe peticiones de conexión y verifica los SLA's predefinidos. Los SLA's son definidos por el contrato entre cliente y proveedor y especifica qué tipo de conexión o servicio puede solicitar un cliente dado. El módulo PM (Policy Manager) es el responsable de aplicar las políticas definidas por el dominio. Las políticas de arreglo y gestión de fallos son definidas [60,61]. El módulo que gestiona los recursos (RM-Resource Manager) maneja toda la información relacionada con el uso de la topología virtual y los recursos ópticos. El módulo que gestiona los fallos (FM-Fault Manager) recibe eventos vinculados a fallos generados por los equipos ópticos de red y prepara los lightpaths contenidos en la fibra que falla.

En esta propuesta sólo se consideran los fallos de fibra aunque existen fallos de lightpath y fallos de dispositivo óptico. Los lightpaths contenidos en la fibra que han fallado son agrupados de acuerdo con su nivel de protección y el módulo FM envía a cada grupo de lightpaths al gestor de políticas que le toca [60]. Finalmente, el módulo MM (Membership Manager) es el responsable de la gestión de la información de cada miembro de cada O-VPN.

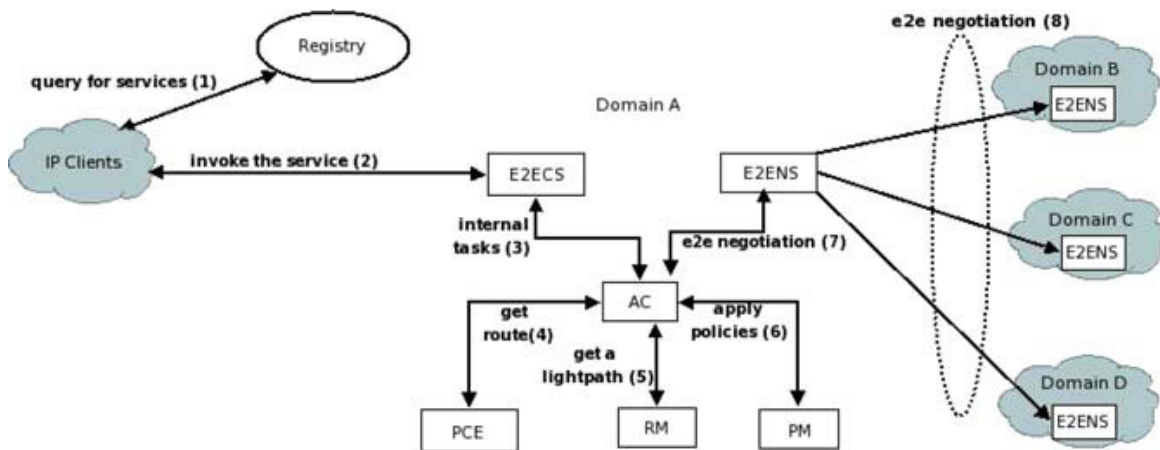


Figura 35: Conexión e2e interdominio.

La figura 35 muestra como los módulos internos interactúan los unos con los otros para establecer una conexión entre interdominios e2e. Durante el primer paso (1), el cliente consulta el registro buscando el servicio de conexión de interdominios e2e. El end-point con la localización del servicio se retorna al cliente (2). Una vez obtenido el interfaz y el end-point del servicio, los clientes ya están preparados para invocar este servicio. La invocación debe incluir todos los parámetros de tráfico requeridos por el cliente. E2ECS recibe la petición y la pasa al módulo AC (3), el cual la valida y pide al PCE una ruta (4). Después se reserva un recurso en el dominio local (5), siguiendo las políticas de admisión de una nueva conexión (6). La negociación realizada por el servicio de negociación de interdominios e2e se realiza en los pasos (7) y (8). Simplificando el paso (8) representa las fases de reserva y confirmación. Destacar que en cada dominio se siguen los pasos (3) (realizado por el E2ENS), el (5) y el (6).

La ruta multidominio e2e se calcula en dos pasos. En el primer paso, el PCE tiene en cuenta toda la topología formada por cada topología de dominio óptico virtual. Esta ruta representa el camino más corto con el coste más pequeño. De todos modos cada ruta no sigue el enrutamiento BGP, ni tampoco los contratos de interconexión entre dominios. Después del cálculo de la ruta basado en la topología virtual, los servicios heredados (ej. BGP y contratos de interconexión) deben ser accedidos con la finalidad de verificar si las políticas de interconexión se respetan. Si los contratos de interconexión aceptan la ruta calculada por el servicio PCE, entonces esa ruta es la utilizada para establecer la conexión interdominio. Si no es aceptada la ruta se usará sin QoS. Además, los acuerdos finales pueden ser hechos a través del E2ENS. La negociación entre dominios se realiza en paralelo. El E2ENS head-end invoca al E2ENS de los dominios pertenecientes a la ruta interdominio e2e. Las limitaciones locales en cada dominio son

aplicadas con el fin de aceptar o no la conexión. El E2ENS de cada dominio retorna la confirmación o la negación de la petición.

Actualmente, el protocolo de negociación no soporta proposiciones contrarias de los flujos de bajada (downstream). Si el dominio invocado no tiene el recurso requerido durante la fase de reserva, la conexión e2e no será aceptada. De todos modos, se podrá encontrar otra ruta interdominio e2e y se negociará. El número de intentos de negociación pueden ser definidos por el cliente.

➤ El servicio O-VPN

L1VPN Background

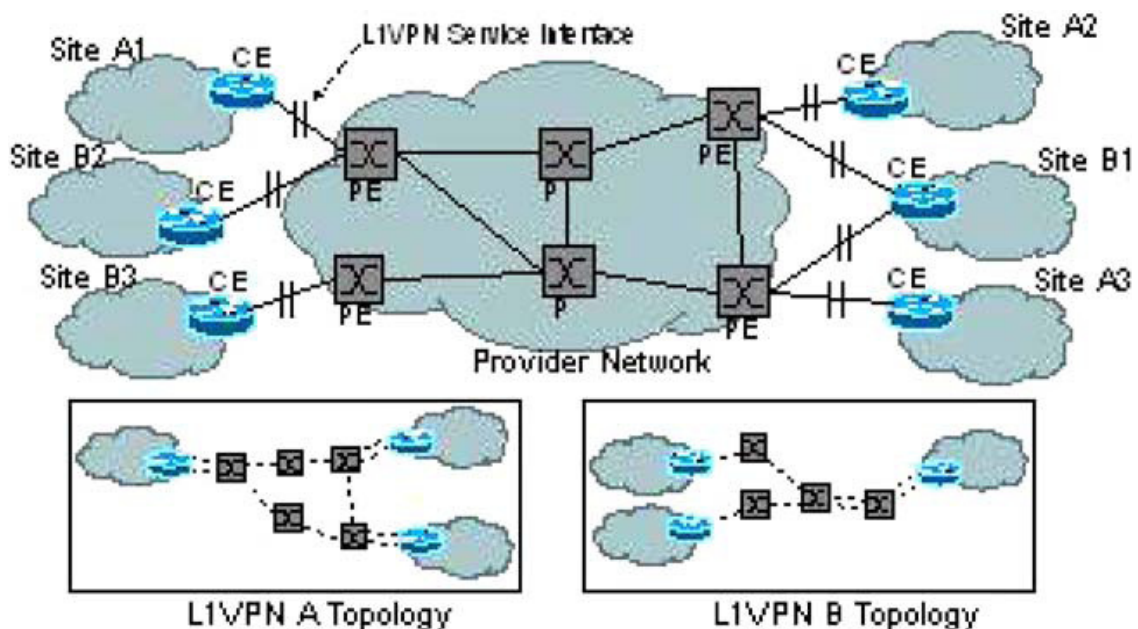


Figura 36: Servicios Ópticos VPN A y B.

La figura 36 muestra dos VPN's ópticas con sus topologías virtuales. Los dispositivos CE (customer edge) son nodos de cliente que reciben el servicio por parte del proveedor. Los dispositivos PE (Provider Edge) son nodos de proveedor que están conectados al menos a un CE. Los dispositivos de proveedor (P) son el núcleo de los nodos de proveedor que no están conectados a los nodos de cliente.

Hay dos modelos de asignación para L1VPN. En el modelo distribuido los recursos de red son usados por múltiples VPN's de una manera time-sharing. Es decir, un recurso que está siendo usado, liberado por una VPN puede ser usado por otra VPN.

En el modelo dedicado, los recursos son reservados para una VPN específica para su tiempo de vida y esos recursos no pueden ser usados por otra VPN. El modelo funcional L1VPN incluye funciones para el mantenimiento de las funciones de los miembros, cálculo de ruta y mantenimiento de la información de ruta, control de conexión y gestión de servicio. La L1VPN sólo especifica las funcionalidades relacionadas con el mantenimiento de la información de los miembros. Las otras tres funciones son comunes con las conexiones individuales [62].

Tenemos tres tipos de arquitectura para proporcionar el servicio L1VPN [63]. En la arquitectura centralizada no hay enrutamiento distribuido ni funciones de señalización. En la arquitectura distribuida, las funciones de servicio de L1VPN usan señalización distribuida y enrutamiento dentro de la red del proveedor. PE y CE se comunican a través de un plano de control común. También hay una arquitectura híbrida en la cual algunas funciones son centralizadas y otras distribuidas. En este caso, las funciones específicas de la VPN son centralizadas y las funciones comunes como proporcionar la conexión dentro del dominio son distribuidas usando, por ejemplo, GMPLS.

En esta arquitectura se soportan ambos, el modelo dedicado y el distribuido y se basa en el modelo de gestión de servicio por el cual los clientes acceden al sistema de gestión de proveedor para pedir las conexiones. Además, la distribución de funcionalidades sigue un modelo híbrido donde algunas tareas son realizadas por el plano de control (distribuido) y otras son realizadas por el sistema de gestión (centralizado). El aprovisionamiento de la conexión dentro del dominio óptico es realizado por el plano de control mientras el mantenimiento de los miembros y la conexión entre dominios son realizados por el sistema de gestión [62].

Los detalles físicos de la red óptica son abstraídos usando el concepto de topología virtual. La O-VPN se establece sobre cada topología virtual reservando las conexiones e2e entre puertos CE de múltiples interdominios [64]. Al mismo tiempo, el modelo de gestión de servicios está bien indicado para el servicio O-VPN ya que este facilita la negociación de los recursos e2e en cada dominio.

Establecimiento de O-VPN interdominio

Para soportar O-VPN interdominio se definen dos servicios, el TS y el propio servicio O-VPN [38].

1. Trading Service (TS)

Es el servicio responsable de reservar y liberar los recursos ópticos para una O-VPN dada. Una vez se reserva el recurso por una O-VPN, ninguna otra O-VPN puede usar el recurso. Este servicio permite a los clientes reservar los recursos necesarios con la finalidad de garantizar que cuando se instale la O-VPN los recursos reservados estarán disponibles. Cuando TS recibe una petición para reservar recursos, se encamina cada petición al módulo AC para validar la petición.

Durante la fase de reserva, el cliente envía información relacionada al establecimiento de la O-VPN (tipo de tráfico, nivel de protección...). Para conectar los puertos CE de la O-VPN también son los clientes los que informan y la combinación de puertos es verificada por el módulo de gestión de los miembros.

2. Servicio O-VPN (O-VPNS)

Este servicio es el responsable de activar y desactivar las O-VPN's así como de disparar el sistema de facturación para empezar a cargar al cliente para el uso de los recursos ópticos. El O-VPNS permite a cada cliente conseguir la información del miembro de la O-VPN y monitoriza cada O-VPN sin interferir en el establecimiento de O-VPN's para otros clientes. La cantidad de información que puede ser entregada a los clientes depende del tipo de acuerdos que están preestablecidos entre los clientes y los proveedores.

El AS (Advertising Service) se propuso para publicar topologías virtuales. Al añadir el O-VPNS a la arquitectura, el AS se ha convertido también en el responsable de publicar la información de los miembros de cada O-VPN. Cada dominio define el mapeo (mapping) entre los puertos de CE y PE para cada O-VPN. La identificación de un puerto CE se conoce como Customer Port Identifier (CPI) y su equivalente en el proveedor es el Provider Port Identifier (PPI) [50]. Estos mapeos son configurados estáticamente en cada dominio óptico (formando una tabla llamada Port Information Table-PIT) y es publicada a los otros dominios que pertenecen al

interdominio O-VPN. Los dominios que no tengan al menos un puerto en el O-VPN no recibirán la información del resto de miembros. La figura 37 siguiente muestra un ejemplo.

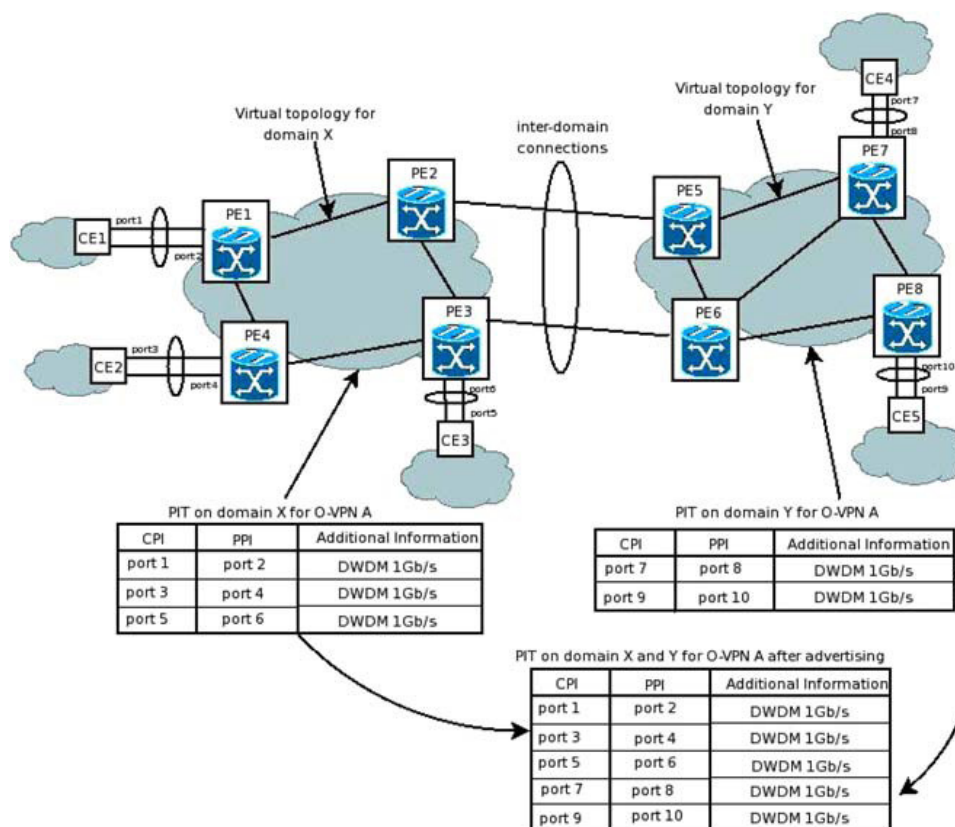


Figura 37: Ejemplo de publicación de un miembro O-VPN.

El servicio de publicación (AS) recibe las PIT's de los otros dominios y envía dichas tablas al MM (member manager) para fusionarlas con la PIT local. Cuando un cliente pide el establecimiento de una O-VPN el MM verifica si el puerto CE dado por el cliente es válido.

Después de la fase de publicación, cada dominio óptico tendrá la información completa sobre todos los miembros de las O-VPN's del interdominio. Como parte del contrato entre clientes y proveedores la PIT de cada O-VPN puede ser obtenida por los CE's para pedir el establecimiento de las O-VPN's del interdominio. Se puede ver en la siguiente figura 38.

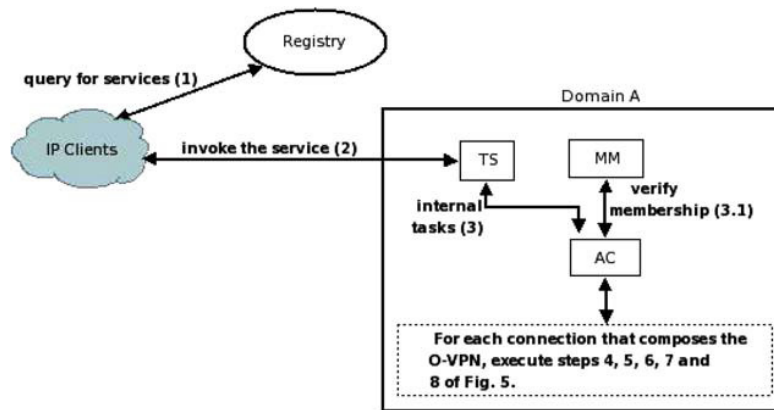


Figura 38: Establecimiento de O-VPN interdominio.

Se puede observar que gracias a la idea de la composición de servicios, el establecimiento de una O-VPN interdominio se forma usando los servicios ya definidos para el establecimiento de conexiones individuales e2e. Básicamente, sólo es necesario añadir un nuevo paso (la verificación de miembro) para proveer un servicio O-VPN. Los pasos restantes son los mismos que se necesitan para una conexión individual e2e.

La reserva de recursos para las O-VPN's se realiza a partir del TS y la activación la realiza el O-VPNS. La cantidad de tareas necesarias para ser procesada durante la fase de activación puede variar en cada dominio. Las típicas tareas incluyen la conexión de los lightpaths (crossconnection) de cada conexión e2e y la activación del sistema de facturación. Un proveedor dado puede usar los recursos reservados para el tráfico de prioridad baja, hasta que el propietario de dichos recursos no active la O-VPN. Cuando éste lo active el tráfico de prioridad baja es desviado por los lightpaths de la O-VPN. Este mecanismo es enmascarado por los clientes y no afecta a la reserva de los recursos.

El punto clave de la composición de servicios está bien ilustrado en el ejemplo del servicio O-VPN. Para proveer dicho servicio, es necesario tener otros servicios responsables de publicar las topologías de cada dominio, calcular la ruta interdominio y negociar las conexiones individuales con los otros dominios. Al ser servicios interdominio e2e primarios, el aprovisionamiento de servicios más complejos y sofisticados (como el O-VPNS) sólo se trata de usar o componer los ya definidos anteriormente. En términos de diseño e implementación dicha propuesta aumenta la creación de nuevos servicios de una forma similar a la reutilización de software.

3.7. Ventajas e inconvenientes de implementar SOA

3.7.1. Ventajas

El concepto básico de SOA es el de proveer de un conjunto básico de servicios a los que cada aplicación puede acceder para realizar su función. En esencia, la idea se refiere a escribir el código una vez para después utilizarlo en cualquier sitio. En consecuencia, el resultado es menos código, menos costes y el incremento de estandarizaciones. El SQA (Software Quality Assurance) total decrece desde que hay menos código para documentar y la documentación de empresa ya existe para los servicios SOA. Además, se deben compilar y debuggar menos líneas de código lo cual ahorra tiempo de programación. Un ejemplo de esta característica de SOA se puede ver en la figura 39 [65]:

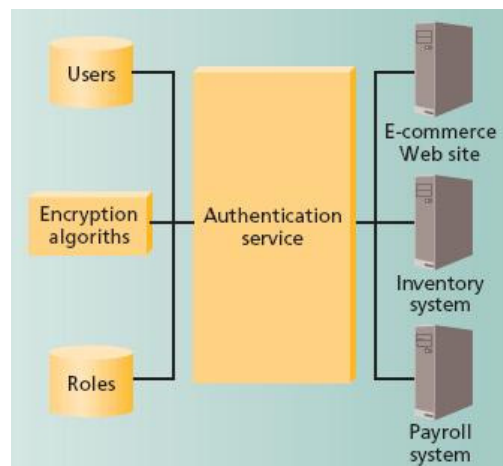


Figura 39: Tres aplicaciones que deben autenticar al usuario antes de permitir el acceso al sistema.

En el ejemplo vemos tres aplicaciones que deben autenticar a los usuarios antes de acceder al sistema. En vez de tener cada aplicación su propia autenticación, todas usan el servicio de autenticación de la empresa. Esto tiene un beneficio inmediato y es que si se detecta cualquier tipo de vulnerabilidad en el servicio de autenticación, sólo se habrá de arreglar una vez, y no tres como en el caso de tener cada aplicación su propia autenticación. Otro beneficio es la simplificación de manejo, por ejemplo en el caso de que un empleado abandone la empresa sólo ha de darse de baja una vez, ya que la base de datos es compartida para todas las aplicaciones que disponen del servicio de autenticación.

Podemos resumir las ventajas más importantes de SOA en [29]:

- Reutilización
- Interoperabilidad

- Escalabilidad
- Flexibilidad
- Eficiencia de coste, al reutilizar infraestructuras ya existentes.

3.7.2. Inconvenientes.

El primer inconveniente con el que nos encontramos al desplegar SOA es que las arquitecturas existentes no están diseñadas para soportar SOA. Es por ello que una migración hacia SOA necesita una gran planificación, reprogramación y adquisiciones. Muchos negocios no tienen presupuesto suficiente para hacer un cambio de este tipo teniendo una arquitectura funcional [65].

Otro inconveniente es que SOA no es tan escalable o tan fácil de conseguir como otras soluciones. SOA depende de XML y comunicaciones de red que requieren análisis y serialización continua, lo cual provoca que incremente el tiempo de transacción y que se haya de invertir en hardware adicional para llevar a cabo estas acciones.

Por último, las empresas son reacias a “jugárselo todo a una carta”, ya que si el servicio falla, todas aplicaciones que lo utilicen fallarán también.

4. Ejemplos de aplicación de SOA

4.1. EWA

En la actualidad tenemos diferentes marcos en los que se desarrolla SOA. Dentro de IBM hay algunos como EWA (Enterprise Workframe Architecture) [10], JADE y Strust (de Yakarta), que están siendo utilizados en escenarios de integración de clientes. En la siguiente figura podemos ver una representación del nivel alto de EWA:

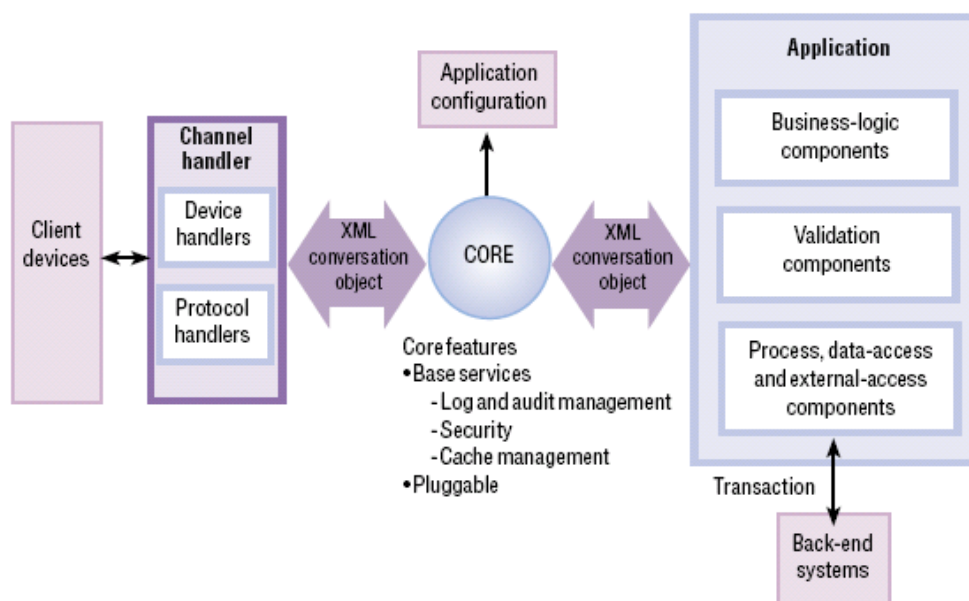


Figura 40: Estructura de EWA.

Dentro de este entorno, la configuración define una aplicación. También se describen los componentes de la aplicación, así como la secuencia y el método de su invocación. La entrada es recibida y pasada de forma neutral a la aplicación. Así, por ejemplo, la adición de una conexión a Internet a un banco, en el que existe un acceso ATM, es transparente a la aplicación lógica. Esto es posible gracias al mecanismo front-end (parte del software que interactúa con los usuarios) y a los encargados de manejar los protocolos. Los servicios a nivel de sistema son proporcionados por las “core features” y los componentes de acceso de propósito específico permiten la conexión al back-end (parte que procesa la entrada desde el front-end) de las aplicaciones de empresa, con lo que pueden seguir utilizándose con normalidad o pueden migrar

con el tiempo. Si bien EWA es compatible con la tecnología J2EE, puede conectar a sistemas exteriores basados en componentes DCOM o CORBA. Hoy en día, EWA dispone de más de 1500 componentes tanto generales como de propósito específico. Lo cual reduce considerablemente la cantidad de código que se ha de escribir a la hora de desarrollar una aplicación.

Uno de los problemas con los que nos encontramos a la hora de implementar arquitecturas es cómo encontrar un esquema que minimice el número de interfaces requeridos. La figura que sigue muestra una idea un poco simplista, pero ilustra el punto de partida para un entorno como EWA. Se introduce el concepto de bus de servicio, que es el representado por una línea central gruesa, y un servicio o “flow manager” para conectar los servicios y proporcionar una ruta de acceso para las solicitudes de servicio. El “flow manager” procesa una secuencia de ejecución definida o “service flow”, que invocará los servicios requeridos en la secuencia apropiada para obtener el resultado final.

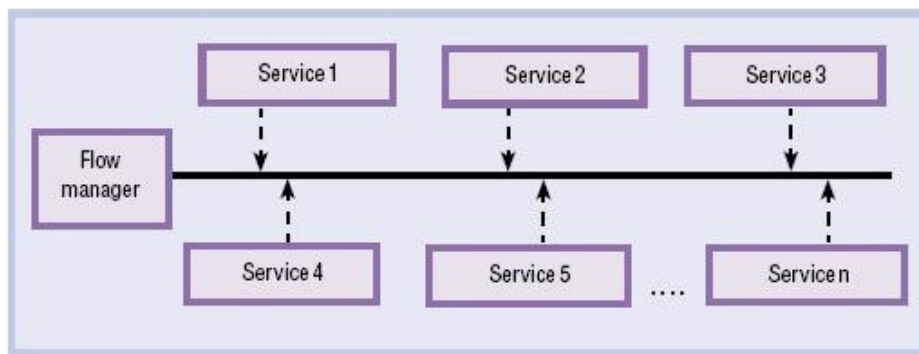


Figura 41: Integración de servicio.

BPEL (Business Process Execution Language), es un ejemplo de este tipo de tecnología para definir un proceso como un conjunto de invocaciones de servicios. BPEL es el lenguaje máquina que permite la implementación del BPM (Business Process Management), es un estándar soportado por la mayoría de fabricantes y físicamente es un fichero XML.

El paso siguiente es implementar cómo llamar a los servicios, con lo que debemos añadir la configuración de aplicación. A continuación, virtualizar las entradas y salidas. Por último, se proporciona conectividad a los procesos back-end. EL resultado final es un marco amplio que permite la ejecución de los procesos tal y como están y además están preparados

para migraciones futuras. En la siguiente figura podemos ver la imagen del nivel alto, al menos estructuralmente completa.

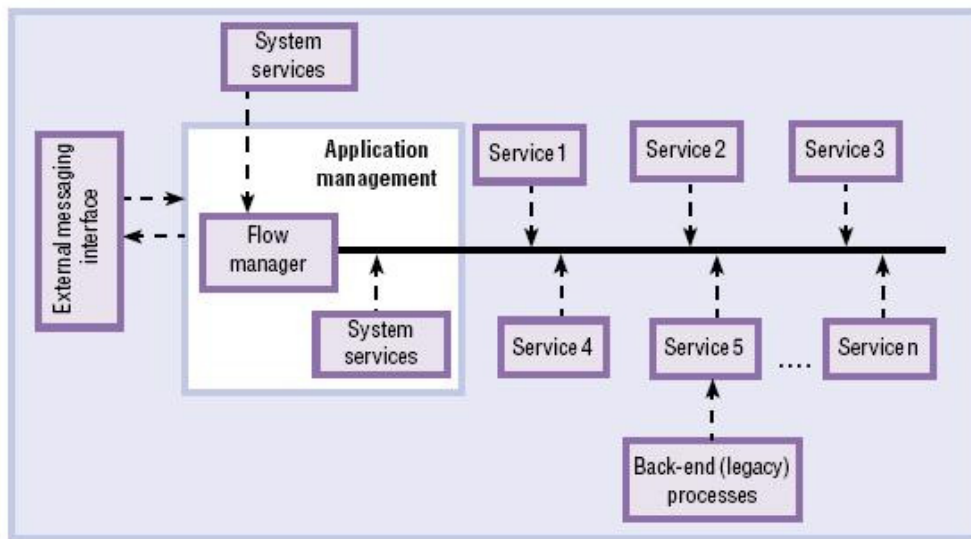


Figura 42: Estructura completa.

El esquema es semejante al diagrama de bloques del EWA, y es que en el nivel más alto cualquier marco de aplicación robusto debe proporcionar estas funciones. Lo más difícil es la creación de los componentes (1500 en el caso de EWA). El proceso de descomposición de las aplicaciones existentes en componentes es bastante laborioso, sin tener que reinventar todos los componentes de propósitos generales y de sistema que sabemos que son necesarios. Por ello muchos arquitectos de IT implementan dentro de un marco ya existente.

Requerimientos de integración dentro de la arquitectura

Al evaluar las necesidades de una arquitectura, se han de considerar varios tipos de integración, no sólo la integración de aplicaciones, sino también la integración del interfaz del usuario final, la aplicación de conectividad, integración de procesos, integración de la información y el modelo de desarrollo de construcción a integración.

La integración en el interfaz del usuario final se refiere a la forma en que el conjunto completo de aplicaciones y servicios a los que un determinado usuario accede, están integrados para proporcionar un interfaz útil, eficiente y coherente. Es un tema en evolución, y los nuevos desarrollos estarán dominados por los avances en la utilización de servidores de portal. Si bien ya los portlets pueden invocar componentes de servicio local a través de servicios Web, las nuevas tecnologías, tales como servicios Web para Portlets Remotos, permitirán a los

proveedores de contenidos y de aplicaciones crear servicios interactivos que conecten con portales a través de Internet y, por tanto, abrir muchas nuevas posibilidades de integración.

La aplicación de conectividad es un estilo de integración con todos los tipos de conectividad que deben ser soportados por la arquitectura. Por un lado, significa que las cuestiones como comunicación síncrona o asíncrona, enrutamiento, transformación, distribución de datos a alta velocidad, gateways y convertidores de protocolo. Por otro lado, también se refiere a la virtualización de entrada y salida, las fuentes, como en el canal y los “protocols handler”. El problema es la manera en que se mueven los datos en la entrada y la salida, y dentro el marco que implementa la arquitectura.

La integración de procesos tiene que ver con el desarrollo de procesos de cómputo que trazan y proporcionan soluciones para procesos de negocio, la integración de los procesos de aplicación y la integración de procesos con otros procesos. El primer requisito es que la arquitectura debe permitir un entorno en el que los problemas básicos de negocio puedan ser modelados. Sin embargo, la falta de análisis en este nivel puede presentar retos importantes para cualquier aplicación de la arquitectura, independientemente de su sofisticación técnica. La integración de aplicaciones como parte de los procesos puede incluir aplicaciones dentro de la empresa, o puede suponer la invocación de aplicaciones o servicios en sistemas remotos, tal vez las de un socio de negocio. De igual modo, a nivel de proceso de integración puede envolver la integración de los procesos en su conjunto, no sólo los servicios individuales, desde fuentes externas, tales como la gestión de la cadena de suministro (SCM) o los servicios financieros que abarcan múltiples instituciones. Para cada aplicación y para cada una de las necesidades del proceso de integración, puede utilizar tecnologías tales como BPEL para servicios Web (BPEL4WS). O el marco de la aplicación puede utilizar un programa de configuración de sistema, como el visto en EWA. El nivel alto de configuración de sistema puede ser construido utilizando BPEL4WS a un nivel inferior y, a continuación, impulsado por un motor que ofrece más funciones que sólo la gestión de los flujos. Antes que ninguna de estas construcciones, se han de entender primero los requisitos de arquitectura, para que pueda construir la infraestructura adecuada.

La integración de la información es el proceso que proporciona acceso coherente a todos los datos de la empresa, a todas las aplicaciones que lo necesiten, en cualquier forma en que lo necesiten, sin restricciones del formato, fuente o ubicación de los datos. Este requisito, cuando se aplique, puede suponer sólo el adaptador de software y un motor de transformación,



sin embargo, normalmente, el proceso es más complejo. A menudo, el concepto clave es la virtualización de los datos, que puede implicar el desarrollo de un bus de datos a partir del cual los datos se pueden solicitar utilizando interfaces o servicios por todas las aplicaciones dentro de la empresa. Por lo tanto, los datos se pueden presentar a la aplicación independientemente de si se trata de una hoja de cálculo, un archivo nativo, una estructura de lenguaje de consultas (SQL) o de otro tipo de bases de datos, o en una memoria de almacenamiento de datos. El formato de los datos en su almacén permanente también puede ser desconocido para la aplicación. La aplicación no tiene conocimiento de que sistema operativo gestiona los datos, de modo que en un sistema IBM AIX o Linux se accede a los archivos nativos de la misma manera que se accedería en Windows, IBM z / OS o casi en cualquier otro sistema. La ubicación de los datos también es transparente, porque es proporcionada por un servicio común, esta es la responsabilidad del servicio de acceso, no de la aplicación, para recuperar los datos, localmente o de forma remota y, a continuación, presentarlos en el formato solicitado. Por último, uno de los requisitos para el desarrollo del entorno de la aplicación es que ha de tener en cuenta todos los estilos y niveles de integración que se pueden aplicar dentro de la empresa, y prever su desarrollo y despliegue. Para ser verdaderamente sólida, el entorno de desarrollo debe incluir (y aplicar) una metodología que establezca claramente cómo los servicios y componentes se han diseñado y construido, a fin de facilitar la reutilización, eliminar la redundancia, y simplificar los ensayos, el despliegue y mantenimiento. Todos los estilos de integración antes mencionados podrán tener una implementación dentro de cualquier empresa, aunque en algunos casos pueden ser simplificados o no claramente definidos, por lo tanto, todos los estilos deben tenerse en cuenta al embarcarse en un nuevo marco arquitectónico. Su entorno de IT puede tener sólo un pequeño número de tipos de fuente de datos, a fin de que la integración de la información pueda ser sencilla, o el ámbito de aplicación de conectividad puede ser limitada. Aun así, la integración de funciones en el marco debe ser proporcionada por los servicios, en lugar de ser realizado adhoc por las aplicaciones, siempre y cuando queramos tener un marco que soporte con éxito el crecimiento y los cambios que vengan dados por las diferentes experiencias empresariales en el tiempo.

4.2. Diseño de ASMF (Autonomic Service Management Framework)

4.2.1. ASMF

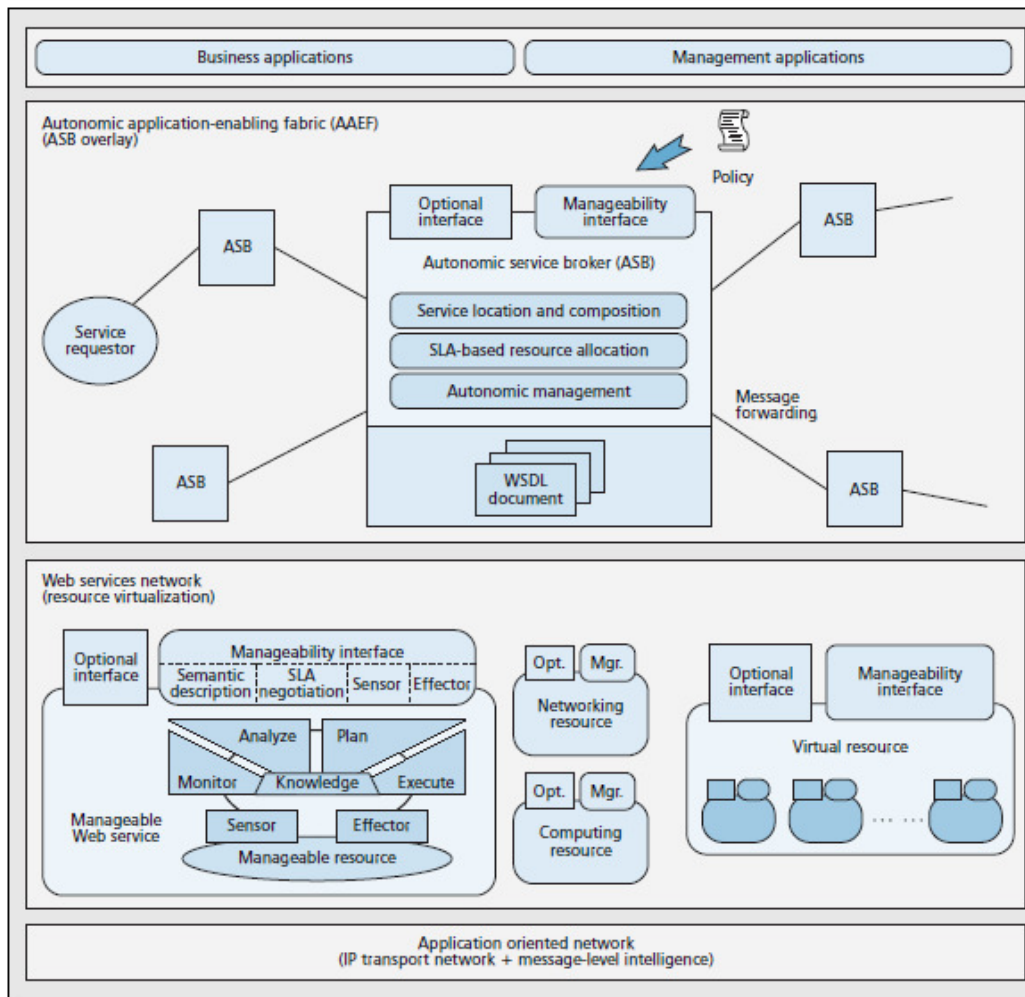


Figura 43: ASMF

La propuesta de un entorno de gestión de servicios autonómicos es la presentada en la figura 43 [36], donde las aplicaciones se crean como una composición de componentes de servicios/recursos manejables. Un componente de servicio puede ser directamente provisto por los recursos físicos, o virtualmente provisto por la combinación de otros servicios. Todos los recursos son manejables. Cada recurso manejable y el gestor autonómico asociado están envueltos como un componente de Servicio Web estándar y además conectado para formar una red de Web Services. Los componentes del servicio interactúan entre sí para formar una relación de servicio y se comportan como proveedores de servicios o clientes, correlativamente. Un componente de servicio puede actuar como proveedor de servicios y como cliente al mismo

tiempo, lo que permite construir aplicaciones de más de una jerarquía de componentes de servicio. A fin de garantizar QoS y la eficiente utilización de recursos, generalmente se contrata un acuerdo de nivel de servicio entre el cliente y el proveedor de servicios. La piedra angular del ASMF es un servicio broker basado en una estructura de aplicación, el AAEF. En AAEF, cada servicio broker está concebido como un elemento autónomo, calificado como un servicio autónomo broker, y múltiples ASB's que forman una estructura autogestionable P2P que permite la composición automática de servicios y la gestión de recursos basada en los SLA's. Insistimos en que cada ASB también ofrece un interfaz manejable, pero principalmente para operadores humanos para instalar niveles altos de políticas, por ejemplo, protocolos P2P que sustentan la red superpuesta, las políticas de seguridad, y la plantillas de SLA. Para simplificar, vamos a llamar a los servicios construidos a través AAEF como aplicaciones o procesos. Otra pieza importante del ASMF es la infraestructura AON de transporte, donde la aplicación inteligente de la red se utiliza para facilitar la comunicación mediante mensajes SOA, mejorar el rendimiento de la capa ASB superpuesta, y aplicar políticas de seguridad.

4.2.2. Red de Servicios Web

En el ASMF, se sigue las especificaciones MUWS y MOWS de OASIS para considerar Web Services manejables, que publican interfaces opcionales y manejables en documentos WSDL para facilitar la composición de las aplicaciones empresariales y de las aplicaciones de gestión, respectivamente. Todos los servicios Web manejables colectivamente forman una Web Service Network (WSN), los cuales pueden ser vistos como una capa de virtualización de recursos sobre la infraestructura física de transporte. La capacidad de virtualización para desacoplar utilidad de recursos físicos específicos puede facilitar enormemente la compartición dinámica de recursos y la optimización de la infraestructura [37]. Las especificaciones MUWS y MOWS no definen los esquemas de funcionamiento interno para la gestión de recursos. Es un prometedor diseño para aplicar una gestión autónoma para gestionar los recursos, en que se aplicará una necesaria monitorización, asignación de recursos, control de admisión, y técnicas de compartición de recursos para garantizar el cumplimiento de SLA's, así como la alta utilización de recursos y de desencadenar la renegociación de SLA cuando el volumen de trabajo o otras dinámicas superen la capacidad de adaptación del sistema de gestión interno. Destacar que, para formar una WSN, el interfaz gestionable superior de un componente de Servicio Web es necesario para seguir la WSDM estándar, considerando que el interfaz gestionable interior podría ser de componentes específicos. Según las especificaciones de WSDM de OASIS, las

capacidades gestionables ofrecidas a través del interfaz gestionable incluyen la identificación, configuración, métrica, estado, operaciones, y eventos generados por entidades manejables para fines de gestión. En ASMF, se considera también el servicio de ubicación/composición distribuido y la asignación de recursos basada en los SLA's como capacidades gestionables importantes. Así, el interfaz gestionable de cada servicio Web es también categorizado en interfaces para la descripción semántica, la negociación del SLA y la gestión autónoma (sensor y efector).

Descripción Semántica: un componente de servicio puede ser definido básicamente especificando las operaciones que el componente realiza y las propiedades del componente que se resumen en el interfaz facultativo o funcional para un Servicio Web manejable. Dentro del interfaz opcional, una operación específica se modela como un par de entradas y salidas desde el componente, y cada entrada y salida se modela como un par de nombre y tipo de datos. La descripción semántica apunta a añadir información interpretable por la máquina para el interfaz de Servicio Web opcional para permitir el descubrimiento automático de servicio y la composición. Para modelar la semántica de un componente de Servicio Web, las entidades de conceptos pueden definirse para representar ideas abstractas que luego son utilizadas para anotar la semántica de las operaciones, inputs, outputs, y las propiedades del componente [40]. Un concepto también puede ser utilizado para especificar la relación entre dos conceptos. Con el concepto anotación, la descripción entendible para la máquina de un Servicio Web, puede ser en el formato de un gráfico semántico que consta de nodos y enlaces etiquetados. Los nodos en el gráfico semántico representan operaciones, inputs, outputs, y las propiedades de un componente, así como sus tipos de datos y conceptos. Los enlaces etiquetados en el gráfico representan la relación entre los nodos [40]. Es de destacar que las descripciones semánticas y las descripciones opcionales están en realidad estrechamente unidas en un gráfico semántico; se abstrae la capacidad semántica en un interfaz separado por conveniencia en el debate de la cuestión de gestión. Utilizando las descripciones semánticas, el proceso de descubrimiento puede basarse en el encuentro entre una descripción declarativa del servicio que se buscó y una descripción del servicio que se ofrece; el proceso de composición puede basarse en integrar los gráficos semánticos de los diferentes componentes del servicio para cumplir el gráfico semántico del servicio solicitado.

Negociación del SLA: En el ASMF, la relación de servicio entre los elementos que intervienen en la prestación de un servicio está regulado por el SLA. Los SLA's son contratos

entre SP's y clientes; definiendo la métrica del rendimiento del servicio, niveles de QoS, autenticación, autorización, y normas de contabilidad relacionadas, y ciclo de vida del servicio. Un procedimiento de negociación de SLA normalmente se realiza mediante el interfaz de negociación de SLA antes de que el SLA es formalmente contratado. A fin de facilitar la negociación automática del SLA, el interfaz debe definir una plantilla estándar de SLA para presentar el contrato de servicios en un formato máquina comprensible [66]. La asignación de recursos basado en SLA es fundamental para la aplicación de la capa de virtualización de recursos; Los recursos gestionables asociados con un Servicio Web pueden ser recursos virtuales, que son, de hecho, algún tipo de capacidad de servir suministrados por otros componentes de Servicio Web a través de SLA's determinados.

Sensor y Efector: El sensor y efector son interfaces manejables para hacer posible la gestión autonómica del componente de servicio Web en el papel de un recurso de servicio Web. El gestor autonómico puede recuperar el estado de la utilización de recursos del interfaz del sensor, mediante el cual el componente de servicio también puede activamente informar al gestor. El interfaz efector se utilizará para entregar la configuración o controlar la información desde el gestor al componente de servicio. Un ejemplo típico para ilustrar las funciones de los interfaces sensor y efector es el control de admisión. Cuando el gestor recibe una nueva petición de servicio y localiza con éxito un componente SP, el gestor próximo debe comprobar la disponibilidad de recursos mediante el interfaz del sensor. Si el SP tiene suficiente capacidad sobrante para cumplir la nueva solicitud de servicio, el gestor puede entonces establecer la asignación de recursos mediante la negociación del SLA. Después de una negociación exitosa, el gestor emite los correspondientes parámetros de configuración de los recursos mediante el interfaz efector, al componente de servicio para hacer cumplir la asignación de recursos soportando el nuevo servicio. Según las especificaciones del WSRF de OASIS, los interfaces del sensor y efector de un recurso WS pueden ser aplicados mediante la definición de un documento de propiedades del recurso y un conjunto estándar de intercambios de mensajes para efectuar consultas o actualizar los valores de las propiedades de recurso de WS [67].

4.2.3. Estructura de la aplicación

La estructura de aplicación autonómica es la capa del núcleo funcional del ASMF, que es el principal responsable de crear servicios o solicitudes de acuerdo con la metodología SOA, y también proporciona gestores autonómicos a aquellos clientes que no tienen su propio sistema

de gestión. En AAEF, el servicio broker, que es el ente crítico en SOA, se ha mejorado a un elemento autónomo definido como el servicio broker autónomo. Múltiples ASB forman una red superpuesta aplicando composición de servicios y gestión de recursos en un enfoque distribuido. La conectividad de la red superpuesta puede ser mantenida por una técnica eficiente P2P; las funciones de nivel de servicio son independientes de las implementaciones P2P. La estructura interna de un ASB se ilustra en la figura 44.

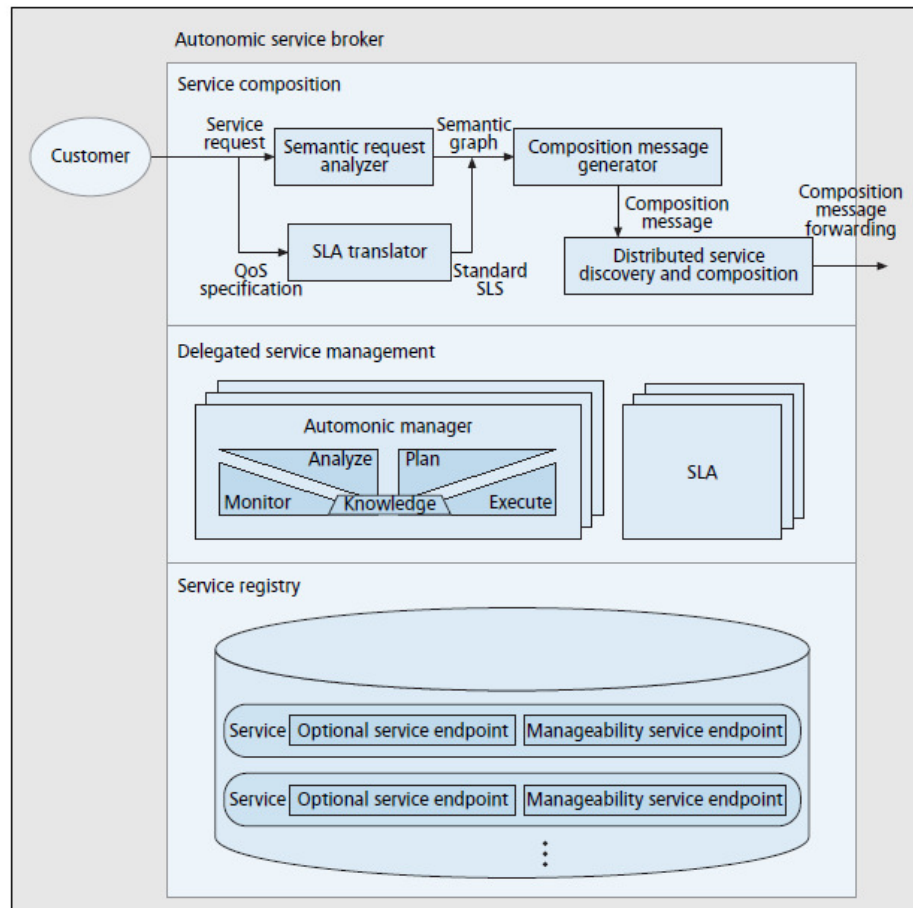


Figura 44: Diseño funcional de un ASB.

Una de las funcionalidades básicas del ASB es actuar como un registro de servicio para permitir aplicaciones basadas en SOA. Cada servicio Web manejable en una WSN publicará dos extremos de servicios: extremo de servicio opcional de la unión a la interfaz opcional y extremo de servicio de administración vinculante a la interfaz de administración. El ASB superpuesto servirá como una base de datos distribuida para almacenar todas las descripciones de servicios publicadas. Al mismo tiempo, el ASB superpuesto también se encargará de la publicación automática del servicio y la composición consultando la base de datos de registros distribuidos

de servicios. Como en cada ASB en la superposición es probable una conexión con un cliente directamente, se utiliza un analizador de solicitud de semántica para analizar la petición de servicios, que podría ser en un lenguaje natural o en un gráfico de semántica; y un traductor de SLA es utilizado para traducir las especificaciones de QoS en especificaciones de nivel de servicio estándar (SLS) sobre la base de una plantilla de SLA. El gráfico de semántica y la información de las SLS se encapsularán en un mensaje de composición, para ser entregados al ASB superpuesto para el descubrimiento y composición de los servicios. En el ASMF, algunos componentes del servicio pueden no tener suficiente almacenamiento, capacidades hardware o software para ejecutar la negociación del SLA, almacenar el SLA contraído, o aplicar una gestión autónoma. Estos elementos de servicio que intervienen en la entrega de la solicitud se pueden delegar al AAEF para aplicar las funciones de gestión del servicio. Para delegar la administración autónoma, los administradores deben ser creados sobre la marcha por los servicios compuestos, gracias a los actuales esfuerzos de normalización este proceso será más sencillo, incluido el diseño modular de un gestor autónomo, elaboración de interfaces gestionables y composición de aplicaciones de gestión.

4.2.4. Servicio distribuido de localización y composición

En AAEF, cuando una solicitud de servicio viene del ASB superpuesto debe entregar el mensaje de petición de servicio para localizar los componentes del servicio apropiados que pueden combinarse para cumplir el requisito de servicio. Ya existen diversos enfoques de ubicación del contenido en una red P2P superpuesta. Sin embargo, una cuestión importante es que el ASB superpuesto generalmente debe encontrar un conjunto de correlación de servicios para ensamblar una aplicación. Por ejemplo, cuando un cliente pide observar una película en línea, el ASB superpuesto intenta localizar un proveedor de contenidos para esa película, un proveedor de red para proporcionar un camino de entrega entre el proveedor de contenidos y el cliente, y un agente de autenticación/facturación asociado con el proveedor de contenidos. Hay una correlación temporal entre los componentes del servicio: el proveedor de contenidos puede empezar a entregar el contenido sólo después de que el cliente pasa la autenticación de identidad y una tarifa apropiada le es imputada. También existe correlación espacial: combinaciones de diferentes proveedores de contenido y proveedores de red pueden proporcionar la película en diferente calidad con diferentes cargos, y el cliente de costumbre deseará que el ASB superpuesto le permita la aplicación de la manera más eficiente. Si bien el actual enfoque BPEL no puede satisfacer los requisitos de localización y composición de

servicios de manera automática, distribuida, y permitiendo una QoS, la composición dinámica de servicios basada en semántica es una posible solución. Según el diseño del ASB de la figura 44, un cliente puede enviar una solicitud de servicio de una forma intuitiva, que será analizado en un gráfico de semántica por el primer ASB, denominado como iniciador de la composición. El gráfico y las asociadas SLS para el servicio entonces se encapsularán en un mensaje de composición para pasar al ASB superpuesto. Cuando el mensaje de composición alcanza un cierto ASB en la superposición, las descripciones semánticas de los servicios almacenados en el ASB son comparados con el gráfico de servicio semántico. Si se encuentran componentes del servicio comunes, son seleccionados para formar un gráfico de semántica parcial. Los puntos de acceso de unión a los componentes seleccionados se adjuntan al mensaje de composición y el mensaje se entrega al siguiente salto de ASB. Como el mensaje de composición viaja a lo largo, el gráfico de servicio semántico se va cumplimentando gradualmente. Cuando el mensaje de composición retorna al ASB iniciador de la composición, el gráfico está completado con todos los componentes para constituir el servicio. Circulando el mensaje de composición alrededor del ASB P2P superpuesto puede conducir a grandes gastos generales al componer un servicio. En su lugar, el iniciador de la composición podrá establecer un árbol, explotando la capacidad de comunicación vía mensajes de AON para difundir el mensaje de composición a todos los demás junto al árbol, tal como se ilustra en la figura 45.

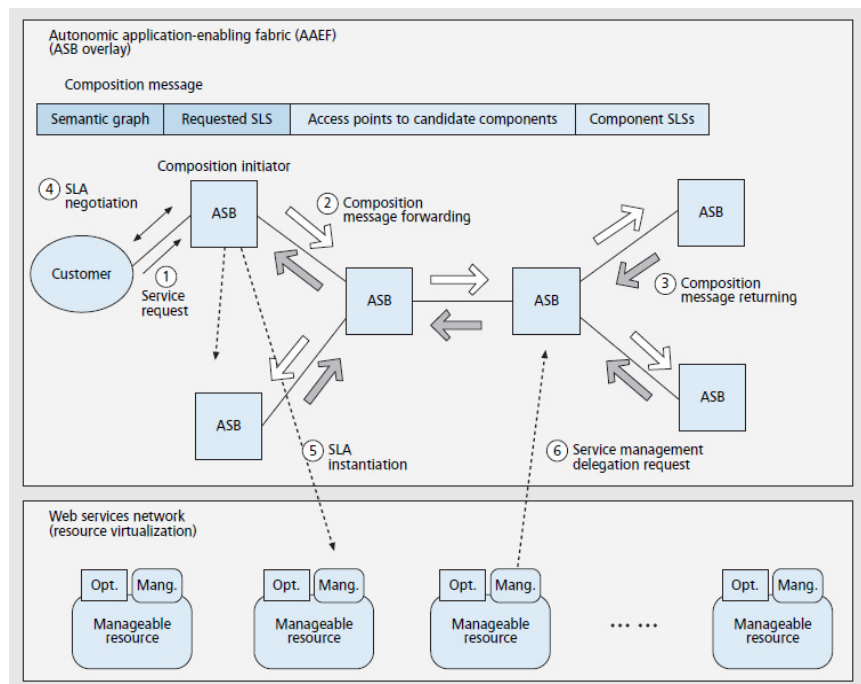


Figura 45: Servicio distribuido de localización y composición gestión basada en SLA.

Cada ASB todavía intenta llenar el gráfico de semántica después de recibir el mensaje y luego continúa la expedición a través del árbol. Cuando cada nodo del árbol termina su transformación, devuelve el mensaje de composición al iniciador. El iniciador recibe múltiples mensajes de composición de retorno, cada uno de los cuales porta un gráfico de semántica parcialmente o totalmente lleno junto a determinadas ramas de árbol. El iniciador de la composición lo completará combinando los múltiples gráficos parciales.

Negociación e instanciación del SLA

El procedimiento de composición del servicio puede incorporarse con la funcionalidad de negociación del SLA. Con la gestión de recursos basada en el SLA, un cliente presenta una solicitud de servicio asociado con rango de QoS preferente y un rango de costes aceptables, según figura 44. En consecuencia, cada componente de servicio también incluye en su descripción del servicio el conjunto de niveles de provisión de QoS y los correspondientes cargos, expresados en un formato estándar SLS. En AAEF, mientras que el mensaje de composición se entrega en el ASB superpuesto, la información SLS de todos los componentes candidatos se recoge. Cuando el gráfico de semántica del servicio se cumple en el iniciador de la composición después del procedimiento de la composición, puede ser anotado por la QoS y cargar información desde las SLS's del componente candidato. El iniciador ASB de la composición puede avanzar el gráfico del servicio anotado para el cliente. El cliente puede calcular la solución de QoS /coste desde el gráfico basado en un determinado modelo y luego, contactar directamente con los componentes de servicio seleccionados para contratar el SLA. Sin embargo, el cliente podrá delegar al ASB iniciador para encontrar la solución óptima desde el gráfico de servicio. En caso de que haya una solución válida que reúna los requerimientos del cliente de QoS/coste y no estuviese disponible, el ASB puede hacer una recomendación de ajuste al cliente y que reinicie una nueva composición y procedimiento de negociación bajo una nueva solicitud de servicio. Después de que una solución de servicio se logre, el ASB envía mensajes de instanciación de servicio a cada componente de servicio seleccionado para contratar los compromisos de recursos de SLA, instanciar los servicios y activar los gestores autónomos asociados. Si un componente de servicio debe delegar al AAEF para almacenar el SLA contraído y aplicar el gestor autónomo asociado, envía un mensaje de petición de delegación de la administración del servicio al ASB donde su descripción del servicio es publicado; de esta manera, el ASB puede rastrear el uso de recursos de los componentes del servicio bajo su gestión. El procedimiento de negociación/instanciación del SLA también se pone

de manifiesto en la figura 45, además del procedimiento de la localización/composición del servicio.

Red de transporte orientada a aplicación

En ASMF, consideramos una infraestructura de transporte AON, que integra la mensajería del backbone orientada a servicio dentro de la red. Según la información de Cisco [34], la estructura de la AON pueden soportar esquemas de seguridad de la capa de red hasta la capa de aplicación; esto también trae la conveniencia de hacer compatibles las políticas de seguridad en servicios y aplicaciones. Aquí, en particular, discutimos cómo explotar la tecnología AON para mejorar el rendimiento de la red ASB superpuesta. En AAEF, las descripciones de servicio publicadas por los componentes de servicio son almacenados de manera distribuida en la red ASB superpuesta. La tecnología Peer-to-peer es atractiva para construir la ASB superpuesta debido a sus propiedades de auto-organización y equilibrio de carga.

Cuando la red ASB superpuesta está construida sobre AON, con cada ASB conectado a un router AON, tanto la capacidad AON para acceder a la información de la capa de aplicación como a la de red, puede ser utilizada para establecer una topología de superposición a conciencia y fortalecer el rendimiento de la superposición mediante la QoS de la capa de red o técnicas de ingeniería de tráfico. Mediante routers AON, el camino más corto en la tabla de enrutamiento de la capa de red estará disponible para el procedimiento de construcción de la superposición para explotar la proximidad de la red. Al mismo tiempo, consciente de su contenido, los routers AON que sustentan la ASB superpuesta pueden solicitar esquemas de DiffServ (Differentiated Services) a la capa de red para acelerar la entrega de los mensajes más importantes. Además, cuando un enlace lógico en la superposición se encuentra interrumpido debido a diversas razones, el re-enrutamiento de la capa de red puede ser recurrido para primero restablecer el enlace de la subcapa antes de reconstruir la topología de la superposición. Como una capacidad de fijación de una capa inferior puede mejorar la estabilidad de la red superpuesta considerablemente, especialmente cuando es utilizada una topología simple de superposición (por ejemplo, un anillo). En una red orientada a las aplicaciones, una superposición P2P puede no ser el mejor enfoque para organizar la red ASB. La superposición P2P fue desarrollada para distribuir contenidos (de una forma en equilibrio de carga) en servidores que residen en el extremo de la red. Con AON, cada router (en el extremo o en el núcleo) puede interpretar y procesar tanto los mensajes de la capa de aplicación como los

paquetes IP, así que el enrutamiento de la capa de aplicación y de la capa de red puede ser perfectamente integrado. Por ejemplo, el multicasting orientado a aplicación puede ser convenientemente logrado en la capa de red en AON. Concretamente, la red superpuesta puede reunir la lista de la capa de aplicación de identificaciones o direcciones de nodos destino en un mensaje multienvío. El enrutador AON interceptará el mensaje y recuperará las correspondientes direcciones IP de los destinos, por lo que los mensajes entonces pueden entregarse fácilmente mediante un multicast IP. De acuerdo con nuestros debates anteriores, un eficiente sistema multicast facilitará considerablemente la localización y composición de servicios distribuidos.

4.3. Arquitectura BT

La nueva arquitectura BT proporciona un entorno para una infraestructura de servicios radical no sólo para dar una experiencia de cliente revolucionaria, productos novedosos y tiempo rápido de mercado para los nuevos servicios, sino que también transforma el coste base para la empresa. La meta es proporcionar servicios de comunicaciones ‘simples y completos’ a los clientes, sin importar el tiempo ni el lugar. Los servicios deben ser fáciles de adquirir y de usar. Los clientes deben gestionar sus propios servicios cuando sea necesario para llegar a un acuerdo con los servicios de cliente, esto debería ser añadido a través de un único punto de contacto [68].

El usuario debe poder acceder a los servicios por cualquier vía de acceso a la red, es decir cobre, fibra, 2G, 3G, WLAN o WiMAX. Los servicios deben estar disponibles para todo dispositivo desde teléfono fijo a móvil, a PDA o PC. La arquitectura es la clave para definir este tipo de escenario. Describamos ahora las principales características de la arquitectura mostradas en la figura 46:

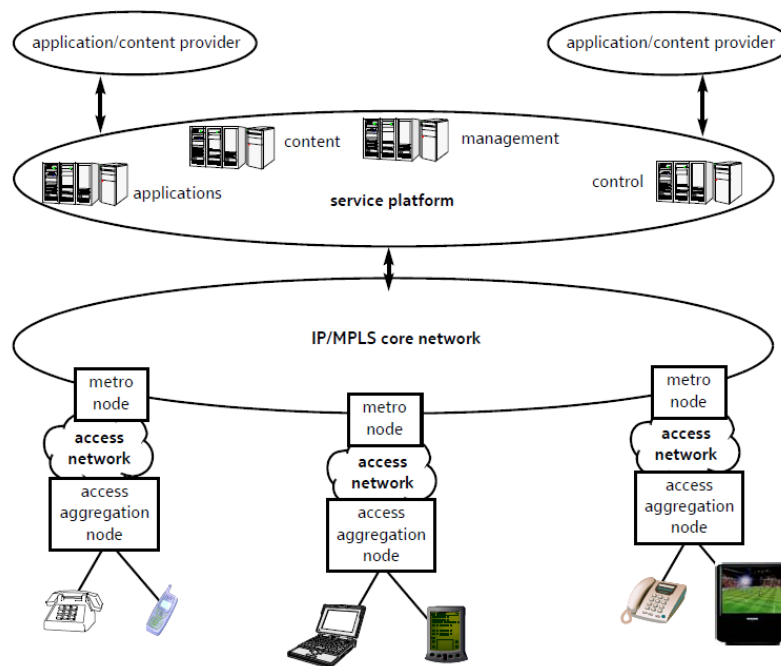


Figura 46: La nueva arquitectura.

1. Network: La core network es una parte de transporte SDH/óptico que soporta un núcleo MPLS. El servicio de red lo da IP. El acceso a la red converge en una plataforma multiservicio. Racionalizando las redes y la tecnología empleada se reducen los costes.
2. La plataforma de servicio: Soporta servicios de convergencia móvil-fijo y servicios multimedia. Se basa en la aplicación del Generation Partnership Programme, la arquitectura IP Multimedia Subsystem (3GPP IMS) [69] para redes fijas.
3. Entorno de aplicaciones: Se ha diseñado para soportar tanto servicios multimedia como móviles [70].
4. Para permitir una visión simple y completa, OSS será transformado en un sistema el cual es parte del servicio, más que la parte ejecutora del servicio. De manera progresiva, estos sistemas convergerán con los componentes inteligentes de la red.

4.3.1. Principios de la arquitectura

1. Multi-networks: Todos los servicios son proporcionados en una red convergente con un core común y un conjunto de tecnologías de acceso comunes. Nuevos servicios serán dados por IP en el nivel de servicio, pero la red convergente acarreará los servicios heredados como el frame relay, ATM y PSTN.
2. Movilidad: La actual PSTN asume que los usuarios están en una ubicación fija. La nueva arquitectura asume que tanto usuarios como dispositivos son móviles.
3. Nuevos servicios: Las plataformas de aplicaciones y contenidos dadas proporcionan, a un grande rango de proveedores, el poder crear y ofrecer servicios en la infraestructura. Esta plataforma expone un rango de capacidades que permitirá el desarrollo rápido de los servicios, su testeo y despliegue.
4. Service execution: Un entorno de ejecución del servicio basado en 3GPP IMS [68], un estándar de industria móvil que proporcionarán el núcleo de los servicios futuros. La capa de ejecución del servicio proporciona una capa de control de sesión del servicio independiente y reutilizable. Usando SOA, tendremos una red de servicios reutilizable dentro de un entorno de aplicación.
5. Rápido aprovisionamiento del servicio: los dominios de ejecución del sistema y de los servicios convergen para proporcionar una mejora radical en la experiencia del usuario, con la que los usuarios pueden proporcionar y gestionar sus propios servicios de forma sencilla. Esta automatización elimina la latencia entre un usuario pidiendo un servicio y

la disponibilidad de éste. Los sistemas necesitan basarse en el usuario, no en el servicio o en el tipo de acceso usado.

6. Infraestructura TI común: Todos los componentes de ejecución de sistemas y servicios se ejecutarán en una infraestructura TI común. Las plataformas estándares serán las utilizadas y las no estándares se eliminarán.
7. Estándares: La arquitectura usa estándares industriales para definir los componentes, eliminando la necesidad de un desarrollo hecho a medida. El objetivo es reducir costes y crear plataformas flexibles.

4.3.2. Vista del dominio de arquitectura

En la siguiente figura tenemos una visión general del dominio de arquitectura:

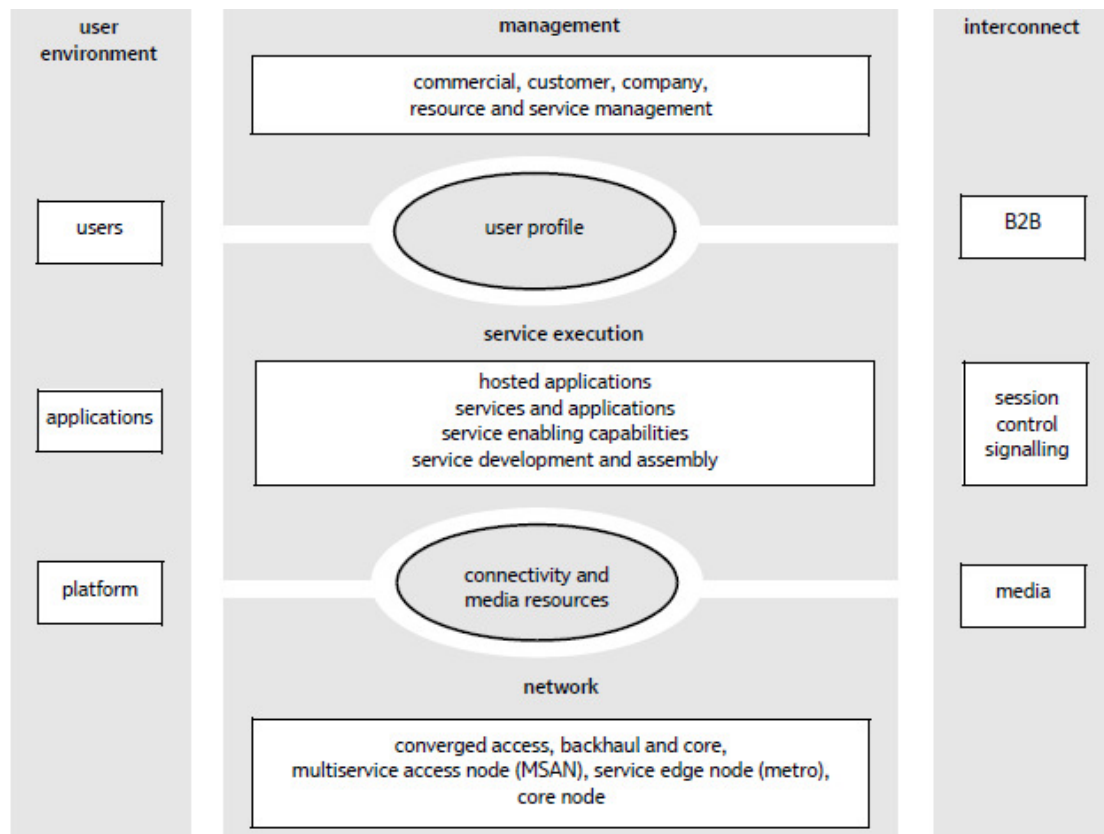


Figura 47: Dominio de la arquitectura.

Las entidades en un entorno de capacidad unificada son agrupadas principalmente en dominios:

-Red: Los cuatro nodos que representan la arquitectura son: el nodo de entorno de usuario, el nodo de acceso a la red multiservicio (MSAN), el nodo metro y el nodo núcleo, éstos definen una política de aplicación de funcionalidades IP, datos, transporte, switching, núcleo y acceso. El nodo de entorno de usuario contiene todas las premisas de usuario incluyendo los gateways de la red de usuario y otros equipos necesarios para el usuario, redes de área local y equipos host de aplicaciones cliente y servidor, así como instalaciones asociadas al mantenimiento. Los gateways de usuario terminan en los servicios WAN de BT. Los nodos de acceso a la red multiservicio (MSAN's) terminan en la infraestructura de acceso física (cobre y fibra) y permiten que el acceso se conecte a los nodos de metro de manera eficiente. De forma alternativa el acceso puede ser proporcionado por el operador de red. Los nodos metro permiten que las combinaciones de servicios lleguen a las redes de acceso. El nodo metro es el dispositivo terminal de la red de servicio. Toda red de servicio específica funciona y está interconectada con las redes de operador y se implementa en los nodos metro. Los nodos del núcleo proporcionan interconexiones de alta velocidad entre los nodos metro, incluyendo switching en el dominio óptico y en MPLS. La arquitectura especifica la prestación de servicios IP. Para soportarlo se requieren mecanismos para gestionar el acceso de usuario a la red y para dar la QoS apropiada. La aplicación de esta política la lleva el dominio de la red. Las funciones de decisión y gestión de la política son llevadas a cabo por el dominio de ejecución de servicio. Estas funciones convergentes están incluidas en el subdominio de conectividad y de recursos del medio.

-Ejecución del servicio: El dominio de ejecución de servicio es el responsable de soportar y entregar los servicios, aplicaciones y los contenidos de usuario. Esto proporciona la experiencia de usuario a través de los interfaces de usuario y tiene el papel de coordinar varios aspectos del servicio y subrayar los recursos de red. El dominio de ejecución de servicio es la aplicación de la arquitectura 3GPP IMS para redes fijas. Las características clave de esta arquitectura orientada a servicio incluye la definición de un conjunto de capacidades de servicios disponibles expuestas en un entorno de aplicación. El entorno de aplicaciones incluye las herramientas necesarias para crear y presentar aplicaciones. El acceso del usuario a los servicios y la configuración de éstos se conduce a través de un profile de usuario. Este profile es gestionado vía el dominio de gestión por ejemplo como parte de un servicio ordenado. También es gestionado a través del dominio de ejecución del servicio para cambios no contratados, por ejemplo el desvío de una llamada. La arquitectura es conducida por el profile y es usada tanto por la ejecución del servicio como por el dominio de gestión.

-Gestión: El dominio de gestión abarca la funcionalidad para ejecutar los negocios de BT, las redes y los servicios. Esto incluye la funcionalidad para el cliente y la gestión comercial, la gestión de la compañía, la gestión de los recursos y la gestión de servicios. Esto también incluye la abstracción del OSS (operations support system) en capacidades reutilizables las cuales serán expuestas en el entorno de aplicaciones de ejecución del servicio. Esto complementa las capacidades de servicio disponibles para permitir a los desarrolladores de aplicaciones integrar aplicaciones con funciones OSS.

-Interconexión: El dominio de interconexión contiene interfaces para partners, proveedores etc. Incluye la interconexión con otros dominios administrativos a través de la señalización, control de sesión y los interfaces del medio.

-Entorno de usuario: Incluye tanto al servicio del consumidor como el de negocio. Incluye un rango completo de dispositivos, aplicaciones y usuarios requeridos para acceder a los servicios de BT.

4.3.3. La red

La reducción sustancial de los costes en la vida de una red es una de las claves en las nuevas arquitecturas.

La red objetivo permite añadirlo a partir de la relación de la infraestructura actual con el acceso a multiservicio y a las redes de core. Esta red objetivo especifica el mínimo rango de tecnologías requeridas por las nuevas redes. Esto lleva a la reducción de costes y de los tiempos de lanzamiento de servicio reduciendo la complejidad comparada con las redes actuales. Con esto lograremos que tanto el mantenimiento de la red como la entrega de servicios sean más simples.

1. Características clave.

La arquitectura física de la red se representa en términos de cuatro tipos de nodos:

- Entorno de cliente
- Nodo de acceso multiservicio.

- Nodo metro.
- Nodo núcleo.

Podemos ver esta relación en la figura:

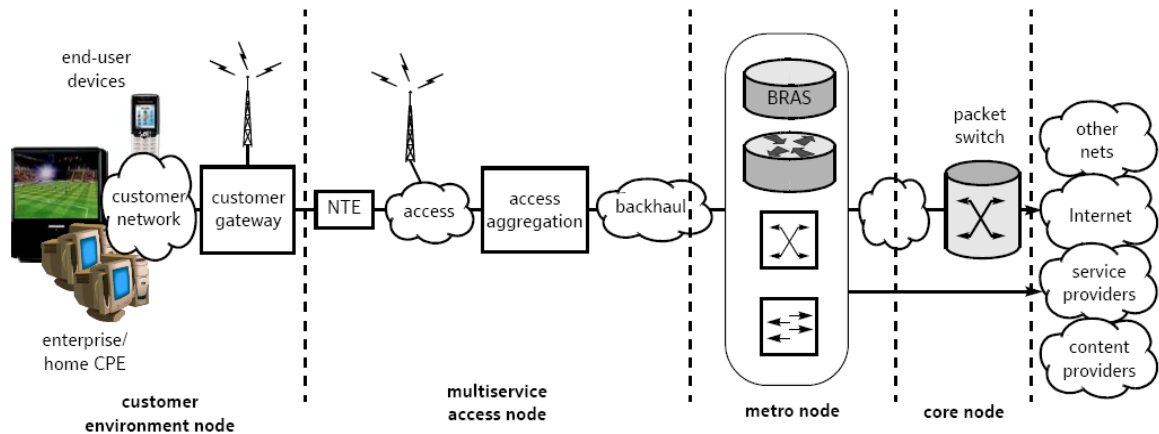


Figura 48: Arquitectura de la red física.

Los nodos se pueden ubicar en puntos de presencia física. Algunas funciones del nodo metro también pueden ser implementadas en MSAN's y gateways de cliente.

1.1. Nodo de entorno de cliente: abarca todos los dispositivos de red de la parte de cliente del punto de terminación de la red, como:

Gateway de cliente, reside en los equipos de terminación de red (NTE) en la parte del cliente y actúa como interfaz entre diversos servicios de la red de acceso y la red de cliente, proporcionando conectividad al cliente y la presentación del servicio.

Tecnologías de red local conectadas al Gateway de cliente, tanto las tecnologías de red en casas de usuarios como en redes de negocios/empresas.

Los dispositivos conectados a las redes de usuario, por ejemplo PC, teléfono, impresora, PBX, etc.

La función clave del Gateway de cliente es actuar como un servicio de control y gestión del punto final y proporciona servicio para la red de usuario. Proporciona un interfaz para la red de acceso y distribuye la voz y los servicios broadband IP a las redes de usuario y dispositivos finales. Proporcionará una gran variedad de servicios para ofrecerlos a los negocios y mercados residenciales.

El Gateway de usuario tendrá las funciones de:

- Controlar la interacción de servicios dentro del entorno de cliente.
- Simplificar la conexión de otros dispositivos de usuario para todo el conjunto de servicio.
- Interactuar con otros elementos, como los de ejecución de servicio, para invocar otras características y servicios.
- Simplificar el soporte de cliente ofreciendo un rango de test remoto, gestión y características de configuración, vía interfaces de dominio de sistemas.

1.2. Nodo de acceso multiservicio

MSAN termina la infraestructura física de acceso (cobre y fibra), y permite el acceso eficiente a los nodos metro. MSAN puede introducir funcionalidades NTE para servicios específicos.

1.3. Nodo Metro

Es el interfaz entre varios servicios de acceso agregados en un área metropolitano y una plataforma núcleo MPLS. Fuera de UK el nodo metro puede actuar como interfaz entre acceso nacional e internacional y las utilidades de la red núcleo contratadas a los proveedores. La principal función del nodo metro es actuar como una agregación y punto de switching, enrutando tráfico entre las redes de acceso y de núcleo. También contiene los interfaces requeridos para conectar redes de otros operadores. Proporciona funciones 'high-touch' como la inspección profunda de paquetes y la señalización de control de llamada/ transcodificación de los medios de comunicaciones de VoIP. También proporciona funciones 'low-touch' como enrutamiento sin conexión y agregación orientada a conexión y switching.

1.4. Nodo núcleo o core

Proporciona conmutación en el núcleo del tráfico entre los nodos metro. Soporta interfaces troncales de alta capacidad y no termina ninguna conexión de usuario. Los nodos núcleo deben estar interconectados densamente con los nodos metro, éstos por lo menos deben estar conectados a dos nodos núcleo. Bajo un punto de vista de arquitectura el núcleo transporta tráfico WDM, TDM y MPLS y todas las formas de tráfico adaptadas a MPLS en los nodos metro finales.

2. Servicios de red

La arquitectura de red multiservicio está diseñada para soportar todos los servicios necesarios. Esta arquitectura se puede describir con los siguientes puntos:

- Los servicios síncronos de ancho de banda son los servicios más contratados y con retraso y tasa de error menores.
- Los servicios de paquetes de ancho de banda proporcionan conectividad point-to-point presentada a los clientes sobre interfaces de paquetes de red (UNI).
- Las VPNs de Ethernet son servicios sin conexión (any-to-any).
- Las VPN IP son similares a los servicios sin conexión VPN IP, aunque elementos de orientación a conexión son soportados por los servicios de gestión de acceso al ancho de banda y por los servicios a tiempo real (VoIP).
- El servicio de Internet proporciona acceso a Internet también a través de otros proveedores de servicios de Internet.
- La capacidad de itinerancia permite a los usuarios acceder a los servicios desde una gran variedad de redes fijas y wireless.
- Los servicios de voz incluyen servicios domésticos e internacionales PSTN y de voz VPN.
- Ancho de banda 'on demand' es una capacidad genérica para que los recursos de red sean asignados en tiempo real en respuesta a las peticiones formuladas como parte de los nuevos servicios a tiempo real, como el aprovisionamiento automático de las peticiones de cliente vía Web e-Service portal o control de llamada multimedia.

En la arquitectura presentada todos los servicios de red son representados, con o sin prestación de servicios de ancho de banda bajo demanda, ya sea como:

- Servicios orientados a conexión (disponibilidad de ancho de banda y conjunto de SLA's entre pares de end-points).
- Servicios sin conexión (disponibilidad de ancho de banda y SLA's presentados como un servicio 'any-to-any').

3. Consolidación de la red

Existen requerimientos para una red física simple consolidada que soporte múltiples modos de operación de transferencia de red:

- Orientado a conexión –circuit switched mode (COCS)
- Orientado a conexión –packet switched mode (COPS)
- Modo sin conexión (CNLS)

Con el fin de racionalizar la duplicación de tecnologías en la red existente, se pretende consolidar múltiples tecnologías de red en un número mínimo de implementaciones de cada modo de red.

En la figura 49 se muestra el soporte a estos modos de red para cada tipo de nodo. Múltiples VPN's separados son implementadas dentro de cada modo/capa lógica de red.

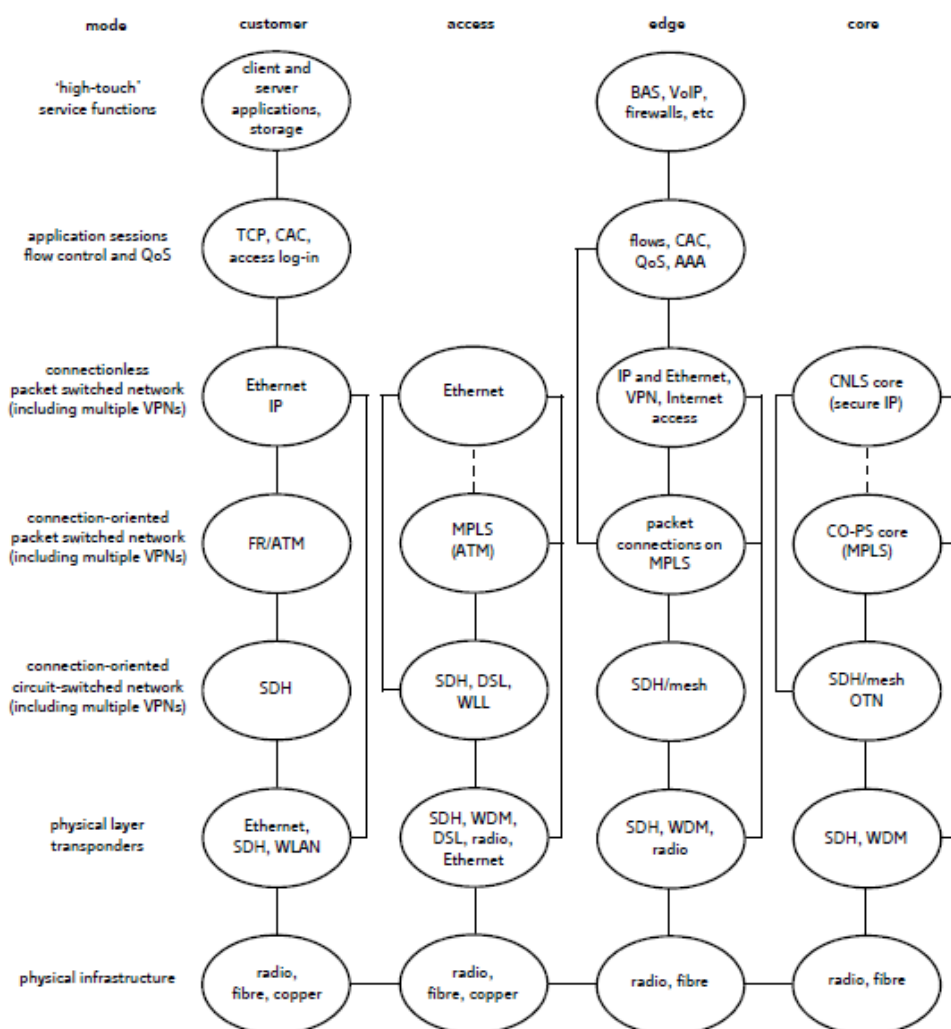


Figura 49: Relación entre los modos y la capa de red.

La arquitectura de red enfocada soporta todos los servicios usando una red simple convergente que implementa los tres modos de transferencia requeridos:

- Internet, MPLS IP VPN y los servicios Ethernet VPN son soportados por una única red IP CNLS lógica implementada por una red MPLS multiservicio.
- Frame Relay, ATM y los servicios de paquetes de ancho de banda son soportados por una única red MPLS CO-PS lógica, implementada por esta red MPLS multiservicio.
- SDH, WDM y servicio de alta velocidad de ancho de banda y redes CO-PS y CNLS son soportadas por una única red lógica mallada CO-CS implementada por redes SDH y DWDM.

4. Interfaces

El dominio del servicio de ejecución dispara la configuración a tiempo real de la seguridad de la red y la política de QoS donde es requerida para implementar los servicios a tiempo real. El dominio de los sistemas automatiza el aprovisionamiento y la gestión de la red vía interfaces con los gestores de los elementos de red y los elementos de red. La arquitectura soporta el uso de redes de proveedor para proveer de servicio a los clientes en caso de no ser económicamente viable para BT operar en la infraestructura física de la red. También puede ser necesario el soporte de la externalización del suministro del servicio a otros operadores existentes y nuevas redes CO-CS, CO-PS y CNLS. Los interfaces seguros con redes de portador de usuario permiten la contratación mutua de las capacidades de la red. La arquitectura descrita soporta interconexiones con redes ya instaladas que se reemplazará para facilitar la migración de servicio de las redes existentes a la arquitectura mencionada.

Ejecución del servicio

La arquitectura SEF (Service Execution Framework) proporciona una estructura para servicios nuevos (ver figura 50).

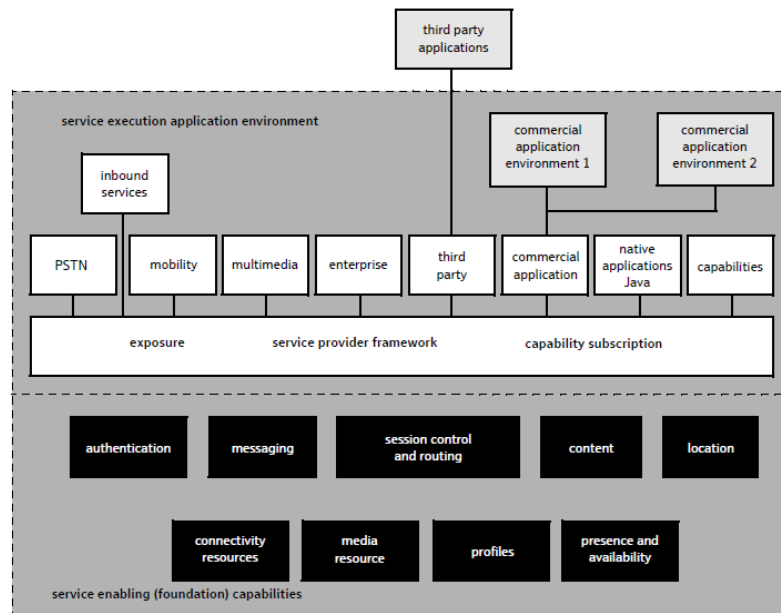


Figura 50: SEF.

Para llevar a cabo esta tarea:

-Enfoque de usuario centralizado: los usuarios tienen acceso a los servicios donde, cuando y como ellos quieran. Esto es posible gracias a su derecho y a sus preferencias contenidas en un fichero el cual es utilizado para llevar a cabo la entrega del servicio.

-Tiempos de lanzamiento de productos reducidos: la ejecución del servicio es una arquitectura orientada a servicio. Define un número de servicios comunes que son expuestos en Servicios Web dentro de un entorno de servicio ensamblado. Estos servicios facilitadores son funciones como el control de sesión, autenticación, presencia, localización, profile. El entorno de ensamblado del servicio proporcionará las herramientas necesarias y el soporte para facilitar que los subyacentes sean construidos en nuevos servicios.

-Proporcionar una plataforma para la convergencia fijo-móvil y los servicios multimedia. Para proporcionar la entrega de productos multimedia en una red de entrega de servicios IP, se requiere un mecanismo de control de sesión independiente del servicio. La arquitectura de ejecución del servicio se especifica para ser SIP (Session Initiation Protocol). Un requerimiento clave es que soporta servicios fijos, móvil convergentes (FMC). Para ello se adopta 3GPP IMS basado en SIP.

1. Entorno de aplicación.

La arquitectura SEF especifica la separación de las aplicaciones de las capacidades de servicio que proporcionan. Estas capacidades se muestran en las aplicaciones a través de interfaces controlados. Las aplicaciones pueden ir desde la PSTN o servidor de llamadas a través de servicios basados en Java. El entorno de aplicación contiene las siguientes áreas:

- Arquitectura: Especifica que el acceso a los interfaces debe ser a través de estándares abiertos, preferiblemente Web Services. Controla el acceso a las aplicaciones y podría controlar el acceso a las capacidades tanto de aplicaciones internas como de aplicaciones de terceros. Esto proporciona control y refuerza el SLA entre aplicación y capacidad.
- Marco de proveedor de servicio: proporciona la habilidad para encontrar y usar una capacidad subyacente. También permite crear el SLA.
- Entorno de aplicación: Especifica una función de subscripción al servicio que permite a los usuarios registrarse para usar una aplicación. La función de subscripción al servicio también aplica la política para tratar con los requerimientos de coordinación e interacción del servicio.

El concepto de entorno de aplicación está incorporado en la arquitectura de aplicación OMA [64]. El foco de los interfaces estándar para las capacidades más comunes es el Parlay Group, especialmente Parlay Web Services y Parlay X [71].

Las aplicaciones construidas sobre SOA utilizan y proporcionan Web Services. Esto permite el desarrollo de un ecosistema donde crece y se desarrolla un rango amplio de servicios.

Para la explotación del entorno de aplicaciones es fundamental la infraestructura y el proceso requerido para la creación de los servicios. A largo plazo, el objetivo para los clientes es que sean capaces de construir sus propios servicios de una manera sencilla a partir de herramientas on-line. Hoy en día muchos desarrolladores están fuera del mercado de las telecomunicaciones debido a las barreras técnicas que experimentan para poder entrar. Barreras como interfaces propietarios y complejos y herramientas de desarrollo industriales específicas.

2. Capacidades de servicio.

SEF especifica un conjunto de capacidades del core que son presentadas en entornos de aplicación a través de interfaces basados en estándares. Aunque estos interfaces estándar expuestos dan servicios independientes en una SOA, debajo del nivel de exposición las

funciones son dependientes y construidas conjuntamente para proporcionar una plataforma para la conexión futura de servicios a la red.

2.1. Control de sesión y enrutamiento.

BT se encuentra al frente de aplicar una arquitectura 3GPP IMS para dar servicio en una red fija, móvil convergente. La arquitectura de ejecución del servicio especifica SIP como mecanismo de control de sesión para los servicios, si estos son llamadas de voz, telefonía multimedia o video-streaming. 3GPP IMS es una arquitectura de control de sesión que permite movilidad.

El motor que permite movilidad es la función IMS de registro y autorización. Gracias a ello, los usuarios pueden encontrar su proxy SIP y registrarse para tener acceso a sus servicios. En la arquitectura 3GPP el servidor SIP es llamado CSCF (Call Sesión Control Function).

El subsistema de subscripción (HSS-Home Subscriber Subsystem) actúa como almacén para los perfiles de servicio de usuario y los datos de servicio de usuario. Este perfil contiene las identificaciones públicas y privadas de los usuarios con los que pueden acceder. Los datos de servicio incluyen datos activos como los balances de pre-pago. HSS actúa como almacén central el cual comparte sus datos con las aplicaciones.

3GPP IMS especifica la separación entre el control de sesión y la lógica de aplicación. Todas las características de servicios residen en el servicio de aplicación. La habilidad de los usuarios para acceder a estas características recae en el CSCF basado en el perfil de usuario recogido del HSS.

El control de sesión es también responsable de la señalización de la red para permitir el acceso de los usuarios y para aplicar las políticas de la red. Las funciones de control de aplicación de estas políticas son definidas como componentes de conectividad de los recursos en la arquitectura SEF.

2.2. Perfil

El perfil de usuario contiene las identificaciones de usuario públicas y privadas, credenciales para la autorización y la información del servicio de subscripción. Estos datos son almacenados y manejados en el perfil y usados por el IMS y otras aplicaciones. Los datos son creados y modificados por el proceso de cumplimiento de servicio y por la configuración del servicio

incluyendo el propio manejo de usuario. Por razones de desempeño, los datos del perfil pueden ser almacenados fuera del componente del perfil.

2.3. Presencia y localización.

Los servicios del futuro se darán en base al contexto de usuario, por ejemplo su localización, dispositivo, disponibilidad y preferencias. Las preferencias son manejadas a través del perfil y usadas por el control de sesión y las aplicaciones. La localización será tanto de red como geográfica. La disponibilidad de la información de contexto de usuario da oportunidad al desarrollo de servicios de usuario centralizados.

2.4. Autenticación

El componente de autenticación necesita tratar con el portal, la red y el dispositivo de autenticación.

Los métodos de identificación incluyen identidad de la línea de llamada (CLI: Calling Line Identity), ID de usuario y contraseña, tarjeta SIM (Subscriber Identity Module), smartcard, y certificados digitales. En el futuro también habrá reconocimiento de voz y escáner de retina.

2.5. Mensajería.

La habilidad de enviar y recibir mensajes será fundamental para los servicios de comunicaciones futuras. El componente de mensaje deberá tratar con mensajes a tiempo real y mensajes que no sean a tiempo real. Los mensajes a tiempo real comprenden IM (Instant Messaging) y PTT (Push to Talk). Mensajes que no sean a tiempo real serían e-mail, SMS, MMS, voice mail.

La arquitectura de mensajes SEF utiliza una aproximación modular que permite estos mecanismos clave, decodificación de mensajes any-to-any, notificación y enrutamiento de mensajes basados en contexto y control de preferencias de usuario.

Los tres componentes funcionales principales son:

- Gateway de mensaje: Proporciona para la adaptación de la tecnología, tipos de mensaje en y desde formatos de mensaje más comunes usados por el componente de mensaje. Pasa los mensajes a la función de procesamiento de núcleo y de enrutamiento.
- Función de procesamiento de núcleo y enrutamiento: Proporciona un conjunto de utilidades comunes de procesamiento de mensajes. Incluye filtros de spam. Detección de virus, generación CDR (Charging Detail Record), mensajes de enrutamiento.
- Almacén de mensajes: Proporciona una utilidad de almacenamiento y envío, almacén de usuario personal y cualquier archivo necesario con fines de auditoría.

2.6. Recursos de Conectividad

La arquitectura de los recursos de conectividad especifica los mecanismos para la aplicación de la política de transporte, el control dinámico de cortafuegos y la traducción de direcciones de red, ancho de banda y QoS.

La premisa principal detrás de este componente es que se tiene una solicitud de un servicio garantizado de conexión desde la aplicación / capa de control de sesión y lleva a cabo los cambios necesarios en los componentes apropiados en el dominio de transporte. Proporcionará una API de red que oculta la complejidad de la red y la simplifica, los requisitos relativos a las capas superiores que se necesitan para comprender una gama genérica de servicios orientados a conexión y servicios sin conexión.

2.7. Recursos del medio

La arquitectura SEF especifica las capacidades comunes que permiten a las aplicaciones controlar los servidores del medio dentro de la red. La arquitectura de los recursos del medio tiene dos funciones principales:

- Primero, la recogida y envío de contenidos para y desde el cliente, esto incluye la recogida de tonos y la entrega de anuncios.

- Segundo, la manipulación del medio para permitir a los clientes interconectarse por ejemplo la transcodificación de codecs.

2.8. Contenido

La arquitectura de contenido añade un componente clave para los servicios multimedia, esta es la función necesaria para gestionar y distribuir los contenidos a través de la red para la entrega para los usuarios finales.

SEF especifica una arquitectura de contenidos genérica basada en componentes que facilita tiempos rápidos para comercializar los nuevos servicios de contenidos, ofrece apoyo a una diversidad de grupos de proveedores de contenidos y proveedores de servicios, y da la agilidad necesaria para soportar los cambios de tecnología. La arquitectura está fuertemente determinada bajo el punto de vista de que un ecosistema de las diferentes organizaciones participará en el contenido del ciclo de vida de entrega.

Las siguientes etapas del ciclo de vida del contenido desde el proveedor de contenido hasta el usuario final están soportadas por la arquitectura:



- interfaz con los proveedores de contenido: debe soportar una amplia gama de contenido desde vídeo hasta las aplicaciones, así como material pre-codificado y metadatos relacionados con el contenido,
- la contribución a través de la red física y de la red de transferencia - también puede incluir un centro de estudio para facilitar la ingestión y facilidades de edición,
- ingestión: codificación y almacenamiento de contenido y metadatos, así como la capacidad de verificar la ingestión para fines de auditoría y de calidad,
- edición y producción: Edición de contenido, ensamblaje, la transcodificación para diferentes dispositivos y los medios de comunicación, la personalización y la intermediación,
- distribución: a los proveedores de servicios, a los recursos del medio (incluida la red de distribución de contenido, set-top box), o a los componentes de mensajería, vía satélite, redes públicas y privadas, y medios físicos,
- Gestión de derechos digitales - para hacer cumplir la decodificación del contenido usando las claves de licencia.

2.9. La contabilidad y la medición

La contabilidad SEF y la mediación de arquitectura especifica los componentes para reunir y mediar en los registros de datos que permiten a las aplicaciones cargar los usuarios finales en tiempo real o los arreglos para la generación de los pagos. La mediación es vista como la transformación de los acontecimientos en hechos imponibles. La contabilidad es la gestión en tiempo real de la cuenta del usuario final.

Una de las funciones clave del componente de mediación es para gestionar la recogida y almacenamiento seguros de la de información, para dar cuenta de los usuarios y el proveedor de servicios de transacciones. Esta información está sujeta al cobro de registro de datos que son sinónimo de registros detallados de llamadas, actualmente en manos de eventos PSTN.

Sistemas

La arquitectura de sistemas tiene el doble objetivo de mejorar el servicio al cliente y la reducción de costes.

Para proporcionar a los clientes de un servicio sencillo y completo, los sistemas ya no pueden ser pensados como únicamente un retorno OSS de negocio de procesos de la



información (BPI), sino que se integran como parte del servicio de entrega. Por ejemplo, un usuario final haciendo una función self-service ha de usar el mismo sistema que un agente de servicio de cliente. La única diferencia entre estos dos usuarios serán los derechos de acceso. Se definirán y gestionarán a través de un perfil individual.

El acceso de usuario es posible gracias a los mecanismos de autenticación y autorización. Este proceso de políticas es el mismo que el requerido por el dominio de ejecución del servicio para permitir el acceso a los usuarios a los distintos servicios. Esto lleva a una convergencia entre los sistemas y el mundo de la inteligencia de red a través de datos compartidos y funciones.

1. Características clave del dominio de sistemas

La arquitectura de sistemas se basa en las características:

- Diseño desde el exterior: Las soluciones serán diseñadas desde la perspectiva del cliente y la experiencia del usuario.
- Información de valor, no los sistemas de información. El valor del core en la arquitectura nueva es la preeminencia dada al mantenimiento de la información de alta calidad, la cual fluye a una alta velocidad a cualquier lugar que se necesite.
- Capacidades reutilizables: la arquitectura nueva será orientada a servicio con capacidades reutilizables transformando los procesos de entrega.
- Soporta componentes imprecisos de negocio. Las arquitecturas orientadas a servicios tienen grandes implicaciones para el proceso.
- El enfoque propuesto para procesos será conducido por una nueva arquitectura de componentes de negocio atómicos que permite la pérdida de acoplamiento entre áreas de negocio funcionales.
- No divisiones artificiales: Las divisiones entre la gestión de la infraestructura IT interna y la gestión de dominios de la infraestructura de servicio desaparecerá.
- COTS: La arquitectura se basa en el empleo comercial de la plataforma de paquetes integrados (COTS) a través de hubs de integración de aplicaciones empresariales (EAI), para construir soluciones orientadas a servicio.

2. Entorno de capacidad de sistemas

SCF (Systems Capability Framework) divide la funcionalidad en el sistema de dominio en un número de subsistemas derivados de capacidades de ingeniería reutilizables.

Las capacidades de ingeniería reutilizables se definen como servicios exigibles relacionados con los sistemas de implementación de una operación de negocio de alto nivel, con la capacidad de ser desarrollada y reutilizada a través de una amplia gama de productos y canales de negocio para facilitar el rápido despliegue de soluciones de sistema integrados.

Cada capacidad de ingeniería reutilizable definirá un conjunto de servicios exigibles y datos asociados, encapsulando implementaciones de sistema de transacciones de negocio de alto nivel. El objetivo es reutilizar las capacidades a través de un rango de productos facilitando el rápido despliegue de soluciones de sistemas integrados.

3. Modelo de información.

Permitir el modelo simple y completo de servicio de cliente y reducir el coste de integración de sistemas, depende de la información subyacente (común a través de los dominios), la cual trata con el servicio de usuario.

La estrategia para implementar funciones de sistema usando paquetes COTS requiere de un modelo de información para ser conducido por la necesidad de este método orientado a solución.

Sin embargo, hay una reducción dramática en el coste de integración usando paquetes COTS, los cuales se ajustan a los estándares de la industria para modelos de información y proceso.

El que mejor aplica a la infraestructura de la empresa BT sería el modelo de compartición de información (SIM-Shared Information Model) de TMF (Telemanagement Forum's), asociado a la iniciativa de modelo de proceso de mejora de eTOM (enhanced Telecommunications Operations Map) [66].

4. Infraestructura de sistemas- el concepto de grid.

La arquitectura física objetivo introduce la idea de integración de 'grid'. El grid tiene cuatro capas:

- La tecnología hub principal basada en EAI, presenta los servicios de hub, la implementación física propone una topología de tres hubs (para la gestión de cliente, servicio y la gestión de recursos y ejecución del servicio), la propuesta de cada una es presentar los servicios soportados por las aplicaciones de sistemas.
- El protocolo de integración ligero.
- Capa de extracción de datos resistente, la cual puede mover una cantidad enorme de datos a través de la inteligencia de negocio o intenciones de sincronización de database, el objetivo es eliminar soluciones basadas en lotes.
- Capa de gestión de Servicios Web, la cual hace de interfaz con la capa de conmutación, permitiendo la exportación de servicios de conmutación como un Servicio Web y permitiendo a los sistemas de conexión con hub utilizar Servicios Web.

La razón más importante para utilizar grids es que permite la gestión coherente de operación, lo cual es importante en un entorno distribuido.

5. Arquitectura de componente de negocio

La arquitectura grid define como los componentes físicos que dan soporte a las capacidades de sistemas de ingeniería son instanciados. Con objeto de fomentar la reutilización, la arquitectura de sistema está desarrollando el concepto de componentes de negocio.

ABC es una colección de gente, habilidades, procesos internos y capacidades de sistema que se ilustran en la figura 51. Esta contrata sus servicios a otros ABC's, y presenta todos sus servicios vía interfaces de tipo XML electrónico orientados a servicio.

El número de ABC's mostrados en la figura 51, contiene cada uno diversas capacidades de sistemas de ingeniería. La gestión de la infraestructura interna de ABC será instanciada por funcionalidades inherentes contenidas en otros componentes. Si se requiere, este desarrollo será hecho como parte del componente subyacente. Los ABC's han definido interfaces y por ello es posible empezar a interactuar con socios externos usando los mismos métodos o incluso subcontratando componentes enteros.

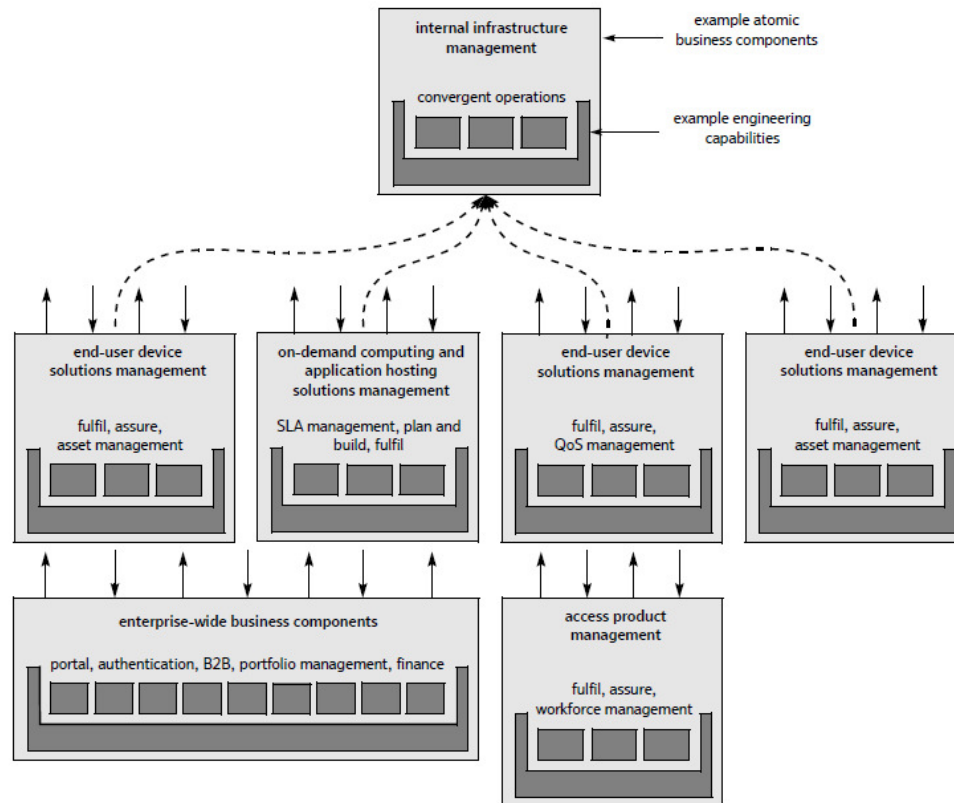


Figura 51: Componentes autónomos de negocio.

Otro punto importante es que esta estructura puede ser usada para la empresa y las infraestructuras de servicio. Como por ejemplo en la figura 51, la gestión de la infraestructura interna ABC puede usar los mismos componentes que un ABC que es utilizado para entregar la gestión ICT a las corporaciones.

4.4. ESNET

4.4.1. La misión de ESnet

La misión de las ESnet's es ofrecer una infraestructura de la red de comunicaciones interoperable, eficaz, fiable y de alto rendimiento, junto con servicios de Grid y de colaboración para dar soporte a la ciencia de colaboración (SC) a gran escala, ciencia que es parte integrante de la misión de la Oficina DOE de la Ciencia [73].

ESnet debe prestar servicios que hagan posible los programas de la SC que dependen de:

- Compartición de gran cantidad de datos.
- Miles de colaboradores en todo el mundo.
- Procesamiento distribuido de datos.
- Gestión distribuida de datos.
- Simulación distribuida, visualización, y la gestión computacional.
- Colaboración con los Estados Unidos y con la Comunidad de Investigación y Educación Internacional.

Con este fin, Esnet proporciona servicios de red y de colaboración a los laboratorios del DOE. ESnet sirve también programas en la mayor parte de DOE.

4.4.2. ESNET

Las principales definiciones y funciones de ESnet se pueden resumir en:

- Una gran red IP construida sobre una infraestructura de circuito nacional con conexiones de alta velocidad a todas las principales redes de US y redes de investigación y educación internacionales (R&E).
- Una organización de 30 profesionales estructurados para el servicio.

- Nivel 1 proporcionando ISP de forma igualitaria a todas las principales redes comerciales, del gobierno, y de investigación y educación (R&E)
- La red primaria del DOE proporciona servicios de Internet a casi todos los laboratorios de DOE y a la mayoría de los sitios del DOE. Esto se traduce en que la ESnet proporciona un valor estimado de 50.000 - 100.000 usuarios del DOE y más de 18.000 investigadores (no DOE) de las universidades, de otras agencias gubernamentales, y de la industria privada que usan las instalaciones de SC con acceso global a Internet.

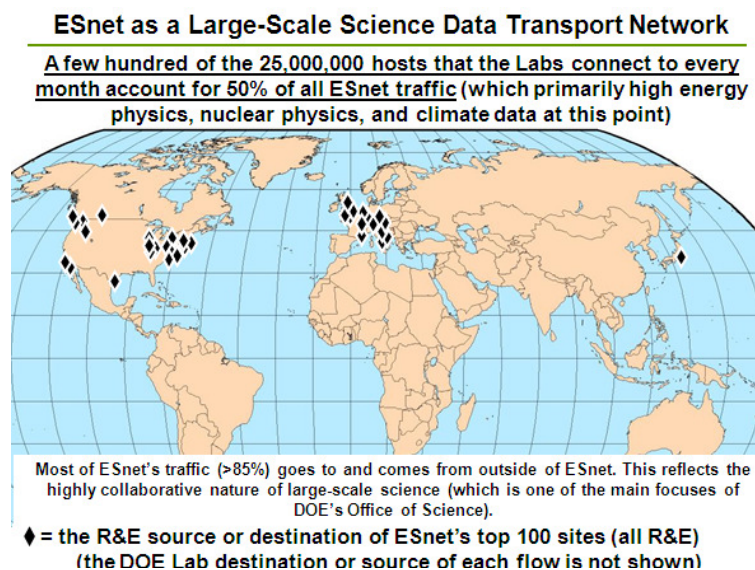


Figura 52: Alcance de la ESnet.

4.4.3. Localización de la ESnet en EE.UU. y la Ciencia Internacional

Una gran parte de todo el tráfico de datos nacional de apoyo a la ciencia de E.E.U.U es llevado por tres redes – ESnet, Internet2, y National Lambda Rail.

Estas tres entidades representan bastante bien el ámbito de aplicación de la arquitectura de red orientada a la ciencia. ESnet es una red en el sentido tradicional de la palabra. Conecta al usuario final a otras redes. Internet2 es principalmente una red troncal. Conecta redes regionales de EE.UU. con las internacionales y entre ellas. NLR es una colección de lightpaths o canales lambda que se utilizan para construir redes especializadas de R&E. ESNet sirve a una comunidad del campus conectado directamente - la Oficina de Laboratorios de Ciencias. En esencia ESNet interconecta las redes de área local de todos los laboratorios con el mundo exterior. ESNet también proporciona la interconexión y enrutamiento necesarios para que los

laboratorios tengan acceso a la Internet global. Internet2 sirve a una comunidad de redes regionales que conectan los campus universitarios. Estas redes regionales - NYSERNet (noreste de EE.UU.), SURAnet (sudeste de EE.UU.), CENIC (California), etc, - tienen puntos de agregación regional llamados GigaPoPs e Internet2 interconecta estos GigaPoPs. Internet2 es una red de tránsito, en su mayoría, de las universidades y/o de las redes regionales y proporciona la interconexión y las rutas para fines de acceso a Internet del usuario. Esto es muy similar a la situación en Europa, donde está GÉANT que es como Internet2. Las interconexiones de las Redes de Investigación de la Unión Europea Nacional y las redes de Educación (NREN), que a su vez, se conectan a la LAN de las instituciones de la ciencia europea y de la educación. (Las redes nacionales son como las redes regionales de EE.UU., pero organizadas en torno a las naciones europeas). Las redes de máximo nivel - ESnet, Internet2, GÉANT, etc - colaboran estrechamente para garantizar que cuenten con una conexión adecuada con los demás de manera que todas las instituciones conectadas dispongan de alta velocidad con la conectividad de extremo para dar soporte a sus misiones de la ciencia y la educación. ESnet e Internet2 han mantenido reuniones conjuntas de ingeniería desde hace varios años (Joint Techs) y ESnet, Internet2, GÉANT, y CANARIE (Canadá) también han formado una organización internacional, un equipo de ingeniería que se reúne varias veces al año. Un objetivo de la ESnet es que la conectividad del DOE de EE.UU. y de los laboratorios de R&E de las instituciones debe ser tan bueno como de laboratorio a laboratorio y de universidad a universidad. La clave para asegurarse de esto es la ingeniería, operaciones, y constante supervisión. ESnet ha trabajado con Internet2 y con la comunidad internacional de R&E para establecer un conjunto de monitores que pueden ser utilizados para comprobar continuamente una malla completa de los caminos a través de todos los puntos de interconexión más importantes.

4.4.4. Requisitos de la red de datos y características de colaboración

Hay unos 20 instrumentos e instalaciones importantes funcionando o que están siendo desarrollados actualmente por la SC, más el LHC (CERN, Suiza) e ITER (Francia). Hasta la fecha, ESnet ha caracterizado 14 de éstos para sus requisitos futuros. Las instalaciones tales como aceleradores grandes del DOE (RHIC en Brookhaven, SNS en la Oak Ridge) y centros de supercomputación (NERSC en Lorenzo Berkeley, NLCF en la Oak Ridge, y ALCF en Argonne), así como el LHC en el CERN, son típicos de la infraestructura de hardware de la ciencia



soportados por la oficina de la ciencia. Estas instalaciones generan cuatro tipos de requisitos de la red: ancho de banda, conectividad y presencia geográfica, confiabilidad, y servicios de red.

Para determinar los requisitos de la SC basándose en cómo el proceso de investigación científica cambiará, un conjunto de casos de estudio fueron desarrollados, en los cuales pidieron a las comunidades científicas que describiesen cómo esperaban tener que hacer su ciencia en cinco-diez años para hacer un progreso significativo. Los informáticos entonces trabajaron con los científicos para traducir los nuevos procesos a los requisitos de la red - particularmente éstos relacionados con la colaboración, la compartición de los datos y el análisis remoto, control instrumental remoto, y las simulaciones a grande escala junto con los demás y/o con fuentes externas de los datos [74].

Las necesidades del ancho de banda son determinadas por la cantidad de datos producidos y la necesidad de mover los datos para el análisis remoto. La conectividad y la presencia geográfica son determinadas por la localización de los instrumentos y las instalaciones, y las localizaciones de la comunidad de colaboración asociada, incluyendo el cómputo remoto y/o distribuido y el almacenamiento usado en los sistemas del análisis.

Estas localizaciones también establecen los requisitos para la conectividad a la infraestructura de la red que soporta a colaboradores (ej. Conectividad ESnet a Internet2 y redes de US regionales R&E, y las GÉANT y las redes nacionales Europeas R&E-NREN's).

Los requisitos de la confiabilidad son conducidos por cómo se relacionan la instalación y los recursos remotos. Por ejemplo, el análisis de datos off-line - donde un experimento funciona y genera datos y se analizan los datos después - puede ser tolerante a un cierto nivel de interrupciones de la red. Por otra parte, cuando la operación remota o el análisis debe ocurrir dentro de la duración de ciclo de funcionamiento de un experimento (análisis "en línea", ej. en experimentos de la fusión por confinamiento magnético), o cuando otros componentes críticos dependen de la conexión (ej. un sistema de ficheros distribuido entre los centros de supercomputación), entonces el tiempo muerto muy pequeño de la red es aceptable.

La cuestión de la confiabilidad es crítica y conduce mucho el diseño de la red. Muchas instalaciones científicas en las cuales el DOE ha invertido desde centenares de millones a los mil millones de dólares, junto con sus grandes comunidades asociadas de la ciencia, dependen en gran medida de la creación de la red. No es de extrañar, cuando los experimentos de estas

instalaciones dependen de la red, entonces estas instalaciones y los científicos exigen que la red proporcione una disponibilidad muy alta (99.99+%), además de un ancho de banda muy grande.

El cuarto requisito está en el área de tipos de servicio. En el pasado, las redes proporcionaron típicamente un solo servicio de red - entrega de paquetes de datos "Best Effort"- en los cuáles se construyen todas las aplicaciones de alto nivel de hoy (ftp, email, Web, librerías de sockets para la comunicación de aplicación a aplicación, etc.), y el Best Effort multicast IP (donde solo un paquete saliente, a veces no fiable, es entregado a receptores múltiples). En la consideración de las aplicaciones futuras de la red de la comunidad científica, se han identificado otros varios servicios de red como requisito, incluyendo la garantía del ancho de banda, aislamiento de tráfico, y multicast confiable. La garantía del ancho de banda es típicamente necesaria para el análisis en línea, que implica siempre apremios del tiempo. Otro tipo de uso que requiere garantías de ancho de banda son los sistemas distribuidos del flujo de trabajo tales como los usados por análisis de datos de la alta física de la energía. La inhabilidad de un elemento (ordenador) en el sistema del flujo de trabajo para comunicar adecuadamente datos a otro, se extenderá a través de todo el entorno de flujo de trabajo, retrasando otros sistemas que participan mientras esperan los resultados intermedios requeridos, reduciendo así la eficacia total del sistema global.

Se requiere aislamiento de tráfico porque el mecanismo primario de hoy de transporte - TCP - no es ideal para transportar cantidades grandes de datos a través de distancias grandes (ej. intercontinentales). Hay protocolos que satisfacen mejor esta tarea, pero estos protocolos no son compatibles con el reparto equitativo del transporte TCP en una red Best Effort, y son penalizados típicamente por la red de manera que reducen su eficacia. Para solucionar este problema, es necesario un servicio que pueda aislar protocolos de transporte masivo de datos de tráfico Best-Effort. El multicast confiable es un servicio que, aunque no es del todo nuevo, se debe realzar para aumentar su eficacia. El multicast prevé entregar una sola secuencia de datos a destinos múltiples sin tener que replegar toda la secuencia en el origen, como es el caso, por ejemplo, al usar una conexión basada en TCP con el origen separado de cada receptor. Esto es importante cuando los datos que se entregan a los múltiples sitios son demasiado voluminosos para ser replegados en el origen y enviados a cada sitio de recepción individualmente. Hoy, el multicast IP proporciona esta capacidad de una manera frágil y limitada (el multicast IP no

proporciona entrega confiable como lo hace el transporte basado en TCP). Se requiere de nuevos servicios para dar soporte al multicast confiable y robusto.

En los casos de estudio que se han hecho [74], una de las instalaciones más importantes de la SC han identificado un requisito para cada una de estas capacidades de la red. Los casos de estudio de “High Performance Network Planning Workshop”, “DOE Science Networking Roadmap Meeting” y “Science Requirements for ESnet Networking”, fueron escogidos para conseguir una buena sección representativa de la ciencia de la SC y para proporcionar las predicciones realistas basadas en cambios altamente probables en el proceso científico del futuro. Los casos de estudio se llevaron a cabo durante varios años e incluyeron los siguientes programas de la oficina científica y recursos asociados: Energía de la fusión por confinamiento magnético, NERSC, ACLF, NLCF, física nuclear (RHIC), fuente de neutrón del Spallation, fuente de luz avanzada, Bioinformática, química/combustión, ciencia del clima, y alta física de energía (LHC).

4.4.5. Resumen de los requisitos de la red

La combinación de los casos de estudio y de las tendencias del patrón de tráfico agrega aspectos cuantitativos a los requisitos generales que fueron identificados prematuramente. La capacidad agregada de la red debe alcanzar 100-200 Gb/s en los próximos años. La confiabilidad de la red debe aumentar de los 99.9% históricos a 99.99% y la disponibilidad 99.99% a 99.999% en el extremo.

Los “peerings” – interconexiones de red externas entre las redes nacionales de R&E, las redes internacionales R&E y ESnet – deben aumentar ambos en ancho de banda y confiabilidad de una manera similar. Además, varios servicios nuevos de red específicos relacionados con la garantía del ancho de banda se deben introducir en la red de producción.

Un requisito general es que debe haber flexibilidad en el aprovisionamiento de la capacidad de la red. El conocimiento de la mayor necesidad de ancho de banda dentro de la red cambiará en un cierto plazo, y los recursos presupuestarios disponibles para la red pueden también cambiar. Debe ser posible agregar y mover la capacidad de centro a centro según sea necesario y desplegar nuevas capacidades en un horario determinado por las necesidades de la ciencia y la disponibilidad de fondos.

4.4.6. Permitir la ciencia futura: Evolución de ESnet durante los 10 años próximos

Basándose en las proyecciones de los programas científicos y los cambios en el tráfico y los patrones observados de la red durante los últimos años, está claro que la red debe desarrollarse substancialmente para resolver las necesidades de la oficina científica del DOE. La tendencia actual de los patrones de tráfico continuará desarrollándose; la ciencia a grande escala tiene en proyecto dar lugar a los 100 flujos de datos superiores que representan cerca de la mitad de todo el tráfico de la red. A medida que los experimentos del LHC aumentaron en 2006-07, los datos de los centros Tier-1 (FNAL y BNL) aumentaron de 200-2000 veces. Una cantidad comparable de datos fluirá de los centros Tier-1 a los Tier-2 (universidades de E.E.U.U.) para el análisis de datos.

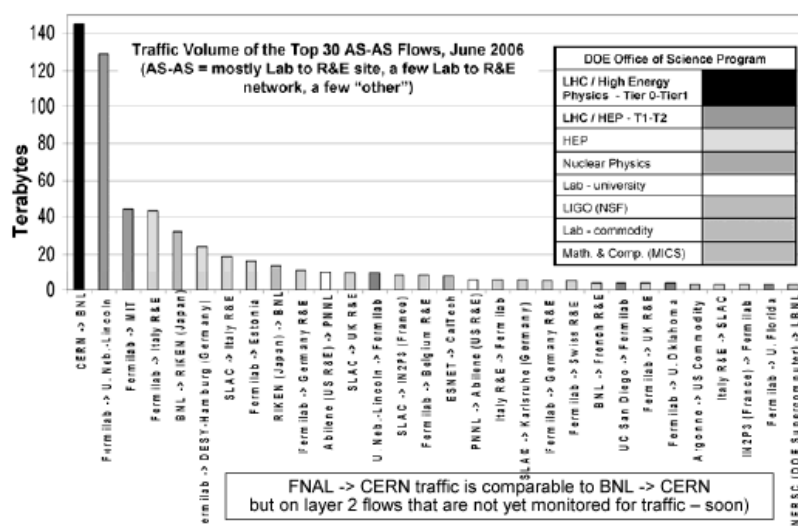


Figura 53: Patrones de tráfico debidos a las nuevas aplicaciones de la red por el LHC.

El supercomputador nacional del DOE National Leadership Class Facility en ORNL anticipó un nuevo modelo de cómputo en el que las tareas de simulación se distribuyen entre los recursos centrales y una colección de estaciones finales remotas que generan tráfico substancial de la red. Como los modelos de clima alcanzarán la sofisticación y la exactitud anticipadas en los próximos años, la cantidad de datos del clima, los cuales se moverán dentro y fuera del centro de NERSC se incrementarán dramáticamente (están ya en los 100 workflows superiores). Del mismo modo, las instalaciones experimento en las nuevas instalaciones Spallation Neutron Source y las instalaciones de Energía de Fusión por Confinamiento Magnético comenzarán a usar la red de modo que requieren un ancho de banda bastante elevado con la calidad garantizada del servicio. Esta evolución en patrones y volumen de tráfico dará lugar a los 100 -

1000 flujos que explican una fracción muy grande de todo el tráfico en la red, así como el volumen de tráfico total de ESnet crecerá: Los flujos de datos a gran escala de la ciencia abrumarán todo en la red.

En el futuro, los pocos gigabits/sec de tráfico medio en el backbone aumentarán a 40 Gb/s (tráfico de LHC) y después aumentarán a probablemente el doble de esa cantidad. Esto conseguirá un tráfico de backbone a 100 Gb/s en 2010-2012 según lo predicho por el análisis de requisitos científicos. La antigua arquitectura de ESnet (de 2004) no reúne los nuevos requisitos que se demandan. El anillo actual del core no se puede escalar para manejar los flujos de datos grandes previstos. El Point-to-Point, la cola de los circuitos comerciales de telecomunicación no es ni confiable ni escalable al ancho de banda requerido.

1. ESnet4: Una nueva arquitectura para satisfacer los requisitos de la ciencia

Para acomodar este crecimiento, y el cambio en los tipos de tráfico, la arquitectura de la red debe cambiar. Los requisitos generales para la nueva arquitectura son que proporcione:

- Establecimiento de una red IP de producción de alta velocidad, escalable, y confiable, conectividad para la universidad y la colaboración internacional, conectividad altamente confiable del lugar para dar soporte a las operaciones del laboratorio así como a la ciencia, y conectividad global del Internet.
- Soporte para los altos anchos de banda de flujos de datos científicos incluyendo conectividad escalable, confiable, y alta velocidad de la red a los laboratorios del DOE.
- Provisión dinámica, los circuitos virtuales con la calidad garantizada del servicio (ej. para el ancho de banda dedicado y para el aislamiento de tráfico).

Para resolver estos requisitos, la capacidad y la conectividad de la red deben aumentar para incluir la conectividad completamente redundante para cada sitio, el acceso de alta velocidad al core para cada sitio (por lo menos 20 Gb/s, generalmente, y a 40-100 Gb/s para algunos sitios) y anchos de banda nacionales de la core/backbone de 100 Gb/s en 2009/2010 en backbones independientes. La estrategia para la ESnet de nueva generación se basa en un sistema de principios arquitectónicos que conducen a cuatro elementos importantes de la red y a un nuevo servicio de red para manejar flujos de datos grandes. Los principios arquitectónicos son:

- Utilizar las topologías del anillo para la redundancia de la trayectoria en cada parte de la red - no tan solo en el core de la WAN.
- Proporcionar conexiones múltiples e independientes por todas partes para proteger los fallos del hardware y de la fibra.
- Disponer de una red de Core - la red IP - especializada para manejar un número grande de flujos de datos pequeños (de centenares a los millares de octetos cada uno) del tráfico general IP.
- Disposición de una segunda red core - la red de datos científicos (SDN) - especializada para un número relativamente pequeño (centenares a los millares) de los flujos de datos masivos (gigabytes al Terabyte cada uno) de la ciencia a gran escala (que por el volumen cuentan con el 50% de todo el tráfico de ESnet y lo dominará ya totalmente en un futuro próximo).

Estos principios de la arquitectura conducen a cuatro elementos importantes para construir la nueva red:

- Una red core IP de alta confiabilidad basada en velocidades elevadas, routers IP de alta capacidad:
 - El acceso a Internet, para la ciencia y el tráfico operacional del laboratorio, y una cierta reserva para los datos de la ciencia llevados por SDN.
 - Los servicios de colaboración de la ciencia.
 - Peering con todas las redes necesitadas para el acceso confiable a Internet global.
- La red core de SDN (Science Data Network) basada en switches Ethernet los cuales soportan MPLS y/switches de capa 1 (ópticos):
 - Circuitos múltiples de 10 Gb/s con una topología rica en ancho de banda total elevado para soportar tráfico científico a gran escala y la redundancia necesaria para una alta fiabilidad.
 - Aprovisionamiento dinámico de los circuitos de ancho de banda garantizado para gestionar flujos de datos científicos grandes, de alta velocidad.



- Compartición dinámica de algunos caminos ópticos con la Comunidad de R&E para gestionar situaciones de picos de tráfico y para proporcionar servicios especializados como todos los caminos ópticos end-to-end para usos que aún no disponen de interfaces de encapsulación (como Infiniband).

- Una trayectoria alternativa para el tráfico de producción IP.

- Anillos de Metropolitan Area Network (MAN) que conectan los laboratorios a los cores para proveer:

- Ancho de banda fiable y elevado (múltiples circuitos de 10 Gb/s) para la conectividad con el core.

- Soporte para el tráfico de producción IP y el científico a grande escala.

- Conexiones múltiples entre el core de la SDN, el core IP y los diferentes lugares.

- Los bucles de los anillos principales para proporcionar las conexiones duales para lugares remotos de la MAN no son prácticos.

Estos elementos están estructurados para proporcionar una red con los caminos completamente redundados para todos los laboratorios de la SC. Los cores IP y SDN son independientes el uno del otro y los dos están estructurados en anillo para la resistencia. Estos dos núcleos nacionales están interconectados en varios lugares con redes de área metropolitana de estructura en anillo que también incorpora a los laboratorios del DOE. Esto eliminará todos los puntos simples de fallo excepto cuando las fibras múltiples puedan estar en el mismo conducto (como ocurre con frecuencia entre los puntos de presencia de áreas metropolitanas y los sitios físicos). En los lugares donde los anillos metropolitanos no son prácticos (como los laboratorios separados geográficamente) la resistencia se obtiene con conexiones duales a uno de los anillos de core (figura 54).

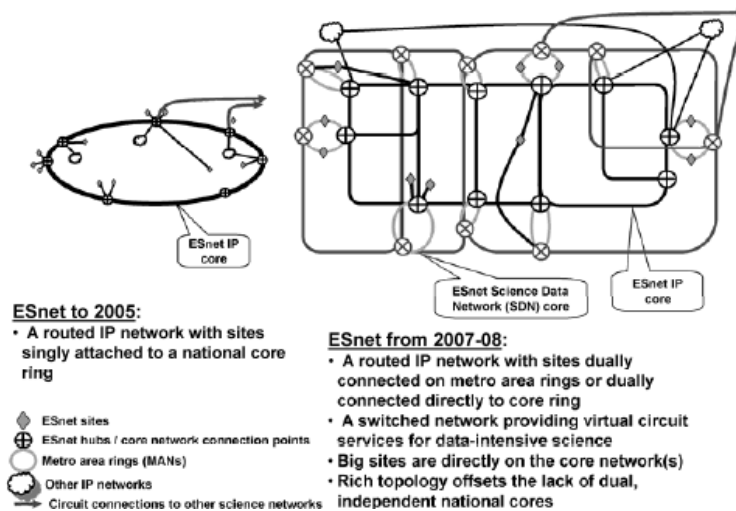


Figura 54: La Evolución de la arquitectura de la ESnet.

Las ventajas teóricas de esta arquitectura están claras pero también ha de ser práctica la implementación. Es decir, ¿cómo ESnet va a conseguir 100Gb/s en el backbone y conectividad redundante en los distintos lugares de 20-40 Gb/s que es lo necesario para la comunidad de la SC en 3-5 años?

2. Implementando ESnet4

Internet2 - la red que sirve a la comunidad de R&E de E.E.U.U. tiene alianzas con Level 3 Comunicaciones Co. e Infinera Corp. para implementar una infraestructura de fibra óptica dedicada con presencia nacional y una topología rica - "la red Internet2." La fibra será provisionada con el equipo de multiplexación Infinera Dense Wave Division Multiplexing que utiliza un diseño óptico-eléctrico avanzado e integrado. Level 3 mantendrá la fibra y el equipo de DWDM como parte de su red comercial, una consideración importante para la confiabilidad. El equipo de DWDM estará inicialmente preparado para proveer 10 circuitos ópticos (lambdas o ondas) a través de la fibra (40 ondas son la capacidad actual de la configuración, 80 la máximas).

ESnet tienen una alianza con Internet2 para:

- Compartir la infraestructura óptica.
- Desarrollar los nuevos servicios de red orientados a circuito.

- Explorar los mecanismos que se podrían utilizar para el Network Operations Center de ESnet (NOC) y el NOC de la universidad de Internet2/Indiana para tener un back up la una de la otra para la recuperación en caso de caída de una.

ESnet construirá su red IP de nueva generación y su nueva red SDN orientada a circuito principalmente en los circuitos ópticos de Internet2 dedicados a ESnet, junto con algunos del National Lambda Rail y otros.

ESnet proveerá y pondrá en funcionamiento sus propios equipos (routers y switches) y que están instalados en varios centros comerciales de telecomunicación alrededor del país, como ha hecho los últimos 20 años. Las relaciones de interconexión entre ESnet e Internet comercial, varias redes de investigación y educación de E.E.U.U. y numerosas redes internacionales continuarán y se desarrollarán como lo han hecho los últimos 20 años.

ESnet4 también supondrá una ampliación de los anillos múltiples de área metropolitana a 10Gb/s en la Bahía de San Francisco, Chicago, Long Island, Newport News (VA / Washington, DC), y Atlanta para proporcionar conexiones múltiples, independientes de los distintos lugares a la red core de ESnet. La implementación de la redes de área metropolitana que conectan los laboratorios al core ESnet es una combinación de redes de R&E, de las redes de fibra oscura, y los circuitos comerciales lambda gestionados. De hecho, en la nueva arquitectura todos los grandes laboratorios SC quedan efectivamente vinculados directamente tanto a la red IP y como a las redes centrales SDN.

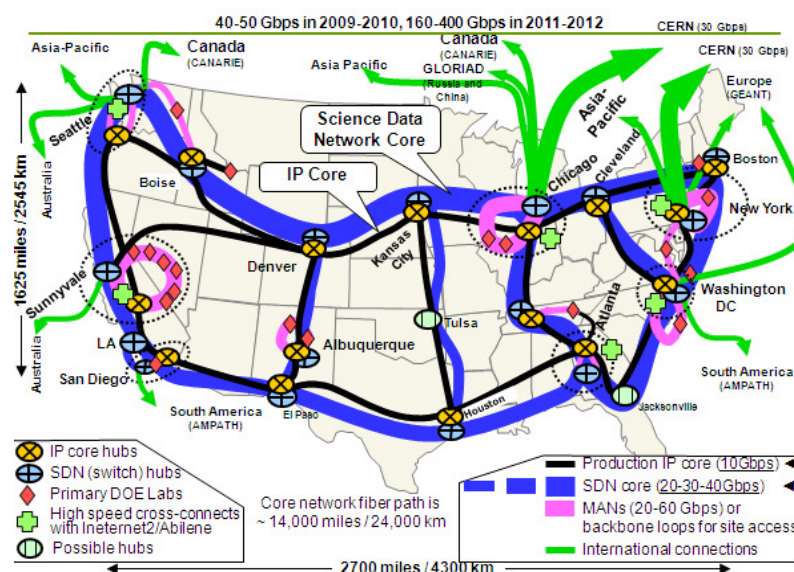


Figura 55: Planificación de la configuración de la ESnet4.

3. Nuevos servicios de red

Los nuevos servicios de red son también críticos para que ESnet resuelva las necesidades de la ciencia a grande escala [74, 75, 76, 77]. El aprovisionamiento dinámico de los circuitos virtuales que proporcionan el aislamiento de tráfico es necesario para permitir el uso de los mecanismos no estándar de transporte que no pueden coexistir con transporte basado en TCP y para proporcionar un ancho de banda garantizado.

El ancho de banda garantizado fue identificado como muy importante en tres situaciones específicas. La primera situación es que es la única manera que tenemos para tratar la fecha límite programada - ej. donde las cantidades fijas de datos tienen que alcanzar sitios en un horario fijo para que el proceso no se detenga y pueda ser alcanzado. Esto es muy importante para el análisis de datos de ciertos experimentos. La segunda situación es donde los elementos que computan remotamente están implicados en el control de experimentos en tiempo real [74]. La tercera situación es cuando los sistemas de análisis basados en grids consisten en centenares de grids en las docenas de universidades que deben funcionar bajo control de un encargado del workflow que coreografía complejos workflows. Esto requiere la calidad del servicio de red para asegurar un flujo constante de datos y de resultados intermedios entre los sistemas. Sin esto, los sistemas con muchas interdependencias podrían parar y comenzar, causando las interrupciones que se propagarían a través de todos los sistemas. Esto crearía un ambiente inestable e ineficaz de la producción que reduciría el rendimiento de procesamiento total necesario para continuar con la generación constante de datos. Esto es una gran preocupación debido a la cantidad enorme de datos que salen de los experimentos de LHC.

Además de los circuitos virtuales, hay otro servicio de red nuevo que es esencial para la monitorización end-to-end del servicio.

4.4.7. Desarrollo y despliegue de los servicios de comunicación orientados a servicio

La SC del DOE ha financiado el proyecto de OSCARS (On-demand Secure Circuits and Advance Reservation System) para convertirse y desplegar varias tecnologías que proporcionan dinámicamente el aprovisionamiento de los circuitos y la QoS que se pueden integrar en un entorno de red de la producción. Tales “circuitos” se llaman los “circuitos virtuales” (VC’s) porque

se definen en software y son mutables (en comparación con los circuitos establecidos hardware). Los elementos de este sistema (ilustrado en la figura 56) son:

- Web-Based User Interface (WBUI) que pedirá al usuario un username/contraseña y lo remitirá al AAAS (Authentication, Authorization, and Auditing Subsystem).

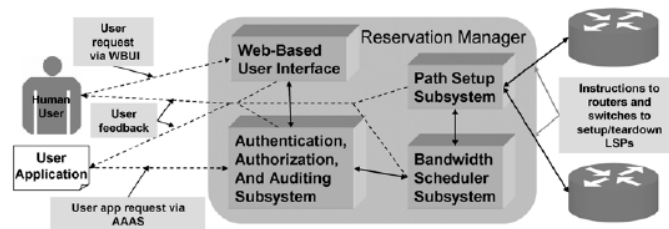


Figura 56: Arquitectura del sistema de gestión de circuito virtual de OSCARS.

- AAAS que autenticará a usuarios, gestionará la autorización de acceso, hará cumplir la política, y generará los expedientes del uso.
- Path Setup Subsystem (PSS) que seguirá la reserva y liberación de los caminos (presente VC's)

El aprovisionamiento extremo a extremo de VC's inicialmente se llevarán a cabo a través de la administración de switches Ethernet de los circuitos ópticos de canal en las MAN, y las VLAN's de Ethernet se gestionarán como paths MPLS (Multi-Protocol Label Switching and Label Switched Paths - LSPs) en el core de la SDN así como los VC's MPLS en la red IP.

Hay dos campos en los cuales OSCARS debe funcionar: 1) intra-dominio, para establecer un servicio planificable, de circuito de ancho de banda garantizado dentro del límite de la red ESnet; 2) inter-dominio, para proporcionar QoS end-to-end entre los laboratorios del DOE y los E.E.U.U. y las universidades europeas.

La configuración de circuitos de ancho de banda garantizado en el inter-dominio no es una tarea trivial. Implica típicamente el circuito virtual que se extiende a través de cinco a siete redes autónomas: la red del laboratorio/del campus en cada extremo, el laboratorio/abastecedor de servicio del campus (por ejemplo ESnet, un RON (red óptica regional) de los E.E.U.U., o NREN europeo) y la red de tráfico nacional o pan-europeo de los E.E.U.U. (por ejemplo ESnet, Internet2, GÉANT) o SInet (Japón). Diferencias en infraestructura de la red, en el hardware, en la capacidad de enlace, etc., se deben tratar en el límite del interdominio para proporcionar características constantes del servicio (ancho de banda, retardo, y jitter) a través de dominios,

igual que las aplicaciones de diversas políticas, tales como las políticas aceptables del uso (AUPs), service level agreement (SLAs) y requisitos de la seguridad. Sin embargo, los circuitos del interdominio son esenciales, especialmente entre ESnet, Internet2, y GÉANT. (Hay que tener en cuenta que OSCARS no aborda la importante cuestión de las políticas de intermediación entre dominios. La aplicación de tales políticas, sin embargo, es crítica al despliegue de OSCARS como servicio de producción. El trabajo de colaboración se está haciendo con GÉANT, proyecto común de la investigación para asegurar un marco compatible de autenticación/autorización).

En ausencia de los estándares acordados para el interfaz de inter-dominio (llamado interfaz externo de red-red, "ENNI") la comunidad está asegurando la interoperabilidad a través de la colaboración en el desarrollo del software.

4.4.8. El papel crítico de la monitorización y reporting

Para construir los sistemas a gran escala, extensamente distribuidos que funcionan confiablemente para realizar el análisis de datos complejo (el caso de estudio de LHC) o las tareas de cómputo de simulación, las aplicaciones distribuidas y el middleware deben poder aprender, en tiempo real, los cambios inesperados en el estado de la comunicación entre todos sus componentes. Sin esta capacidad los usuarios o los operadores de sistema dejan de intuir qué ha ido mal. Un problema que parece venir a partir de un componente puede realmente ser un problema de las comunicaciones no denunciado de una parte muy diversa del sistema. Una red confiable que supervisa el servicio que describe el estado actual de la aplicación de las comunicaciones permite que las aplicaciones adapten su comportamiento a las circunstancias que cambian, o por lo menos que fallen y que anuncien exactamente por qué está fallando. Un cambio esencial en los servicios de red durante los cinco años próximos será proporcionar la información confiable, comprensiva, oportuna, e interpretable sobre el estado de todos los componentes de las redes en la trayectoria end-to-end de una forma que se puede interpretar y utilizar significativamente por aplicaciones de nivel de usuario. Esta capacidad se debe acompañar por una capacidad correspondiente en las aplicaciones y el middleware para aceptar los servicios de la comunicación que supervisan resultados y para hacer algo inteligente con esos resultados. Esto puede incluir la adaptación del funcionamiento del sistema al cambio / disminución de la capacidad de servicio de comunicación, cierre correcto del sistema, notificar al

usuario lo que está pasando (en términos que son útiles para los usuarios implicados), y así sucesivamente.

Los resultados del monitor se deben presentar de una manera que sea significativa a la opinión del usuario de la red. Junto con las nuevas capacidades proporcionadas por los circuitos virtuales, supervisando los servicios que pueden divulgar los problemas directamente a las aplicaciones de red y los usuarios moverán las comunicaciones de la red hacia un modelo gestionado del servicio.

Background

Todas las redes hacen la monitorización extensa en tiempo real que se utiliza para una variedad de aplicaciones. La supervisión a corto plazo (en la orden de minutos) se utiliza para identificar y eliminar errores de problemas en cada elemento de la red - circuitos, interfaces, equipos de switching y routing, estado de routing (conectividad lógica), y así sucesivamente. La monitorización se utiliza principalmente para detectar fallos o inicio de fallos en el funcionamiento a través de la degradación del rendimiento o de algunos indicadores de aspectos principales de la red. ESnet, por ejemplo, supervisa casi 5.000 características de elementos de la red en tiempo real en su red nacional. Un sistema de monitorización comercial de la red (espectro) se utiliza para manejar esta información, genera alarmas de operador, y así sucesivamente.

La monitorización a medio plazo (de horas a días) del tráfico del interfaz, se hace para la gestión de la capacidad: Pueden desarrollarse zonas de conflicto en la red debido a los cambios en la demanda o la capacidad del usuario, los cambios en la capacidad de la red (aumento, o interrupciones), o los cambios del encaminamiento. Puede ser posible tratar estas zonas a través de los cambios de la configuración (encaminamiento), pues las redes están creciendo densamente malladas internamente y más ricamente conectadas entre sí. Esta clase de información puede también accionar la reconfiguración física de las partes de la red - típicamente aumentando ancho de banda del interfaz cuando es posible.

La monitorización a largo plazo (meses a los años) del tráfico, da soporte a la planificación de las configuraciones de red futuras: Las tendencias de tráfico que se presentan durante meses o años (ej. figuras 6, 7, 8 y 9) son esenciales en la planificación de cambios en el futuro de la arquitectura y en las grandes mejoras que se producirán en años futuros. Éstas son una de las varias métricas que conducen el diseño de la generación siguiente de la red.

Típicamente, el uso detallado del interfaz de la red (granularidad a nivel de minuto) está disponible en línea alrededor de un mes y después se archiva para referencias futuras. La información recopilada (datos del monitor resumidos en cada hora, a diario, o la granularidad semanal) está disponible en línea varios años. ESnet, por ejemplo, casi supervisa mil interfaces lógicas de la red en 64 routers y switches, y recoge y archiva cerca de 325 GBy/mes de datos monitorizados.

Metas del diseño del sistema de monitorización de la red

El estado detallado del enlace de red a tiempo real y el rendimiento de los datos se recoge y se archiva rutinariamente en casi todas las redes de producción. Sin embargo, lo que interesa a los operadores de red es el comportamiento de los interfaces específicos de los routers o switches y del enlace que los conecta. Por lo tanto los datos se recogen y los archivos de los datos se organizan y se ponen en un índice de este modo. Además, la forma de estos datos es típicamente peculiar en cada red, haciendo la información casi inútil al usuario que intenta ver el comportamiento end-to-end. Para ser útil al usuario para la supervisión end-to-end, la información debe satisfacer un sistema adicional de requisitos. Debe haber herramientas para mapear la vista del usuario (según lo representado, por ejemplo, por un “traceroute” del path de aplicación a aplicación) a la vista de la red y entonces recoger y mapear los correspondientes datos de monitorización de la red para mostrarlos al usuario. Es decir, las herramientas deben convertir la vista del usuario a la representación física del path - los sistemas de los interfaces y de los enlaces que abarcan los paths en el nivel físico; extraer los datos relacionados del archivo; mapear de vuelta al usuario; y devolver los resultados al usuario en un formato que sea estándar a través de todas las redes.

Además, la trayectoria end-to-end entera se debe incluir en la monitorización. En un entorno R&E típico tales trayectorias implican de cinco a seis dominios de la red: la LAN, la red regional o nacional, una segunda red nacional o PAN (Personal Area Network) nacional, nuevamente dentro de una red regional o nacional, y en la LAN del sitio en el otro extremo de la trayectoria. Cada uno de estos dominios debe proporcionar los datos para los segmentos de la trayectoria del usuario que son parte de ese dominio. Esta clase de monitorización del dominio cruzado es crítica para las aplicaciones de alto rendimiento que dependen de componentes extensamente distribuidos y para los operadores de red que requieren cada vez más manejar las trayectorias end-to-end. Un usuario debe poder recibir notificaciones de las interrupciones del

servicio suscribiéndose a las alarmas para una trayectoria dada. Además, el informe debe proporcionar la información sobre la fuente de la interrupción - es debido a la congestión (al cuál puede contribuir el usuario), o los errores del enlace que es un problema de la red, o cualquier otro problema. Actualmente las interrupciones previstas se registran por los operadores de red en un sistema de calendario y este sistema debe también divulgar las interrupciones futuras al usuario. Una vez más este problema desde el punto de vista del usuario es que las descripciones están dadas en términos de topología física de la red. Para ser útil al usuario, la topología física se debe mapear en descripciones de la trayectoria del usuario y los fallos del punto se deben divulgar en términos de su impacto en la trayectoria del usuario.

4.4.9. Nuevos servicios de monitorización

PerfSONAR es un primer paso significativo en la monitorización del cross-domain por los operadores y los usuarios de red. PerfSONAR tiene tres contextos: 1)perfSONAR es primero un consorcio de organizaciones que intentan construir el rendimiento del middleware de la red que es interoperable a través de redes múltiples y útil para el análisis intra e inter- red. Una de las metas principales es hacerlo más fácil para solucionar problemas de rendimiento end-to-end en las trayectorias que cruzan varias redes. 2) el perfSONAR es un protocolo. Asume un sistema (varios tipos de servicios), define el estándar de protocolo (sintaxis y semántica) por el que se comunican, y permite que cualquier persona escriba un servicio que desempeñe uno de esos papeles. El protocolo se basa en mensajes SOAP XML y en Open Grid Forum (OGF) Network Measurement Working Group (NM-WG). 3) el perfSONAR es, finalmente, un sistema de ejemplo de código (puesta en práctica de servicios) procura dar un marco interoperable del rendimiento del middleware en ejecución. Esos sistemas de código son desarrollados por diversos socios. Algunos trozos del código son “más importantes” que otros porque su meta es asegurar la interoperabilidad entre los dominios (por ejemplo el servicio de operaciones de búsqueda y el servicio de autenticación). Diversos subconjuntos de código son importantes para cada socio, con un alto grado de superposición. Los servicios desarrollados actúan como capa intermedia, entre las herramientas de la medida del rendimiento y las aplicaciones de diagnóstico o de visualización.

Funcionalidad: Para satisfacer las necesidades de las diversas comunidades de usuarios de los datos de la red - los operadores y los ingenieros de red, el personal de soporte de la red en las instituciones de los usuarios finales, y los usuarios finales ambos en el proceso de depurar

el rendimiento de las aplicaciones distribuidas o como parte de un servicio que reporta problemas de la red al gestor del recurso de la aplicación - hay varios aspectos de la supervisión de la red que deben ser tratados.

Hay tres categorías generales de la medida del rendimiento de los datos - medidas activas, medidas pasivas, y variables del estado de la red (variables del SNMP) que pueden ser vistos como productores de los datos. Desde el punto de vista del usuario de la red de datos estos datos deben estar disponibles en diversas formas y deben tener varios servicios asociados a ellos para homogeneizar la información de las diversas redes y para presentar los datos de manera útil. Los datos se deben proporcionar como flujos de datos o vía interrogación. Las herramientas de análisis, las alarmas umbrales, y las herramientas de monitorización son los consumidores de datos que, alternadamente, necesitan que los datos se muestren de varias maneras. Por lo tanto, entre los productores de datos y los consumidores de los datos puede haber un conducto de agregadores, de correladores, de filtros, y de los servicios del almacenamiento intermedio que se pueden ver como los transformadores de datos y archivos de datos. Además, los servicios - los productores, los consumidores, los transformadores, y los archivos de datos - son todos los recursos que necesitan ser descubiertos y utilizados casi ciertamente dentro de un marco de la autenticación y de la autorización que mantenga la política prescrita por los operadores de red que poseen los datos de medición.

Arquitectura: Una arquitectura orientada a servicio (SOA) ha sido adoptada por la comunidad la cual consiste en tres capas y una colección de funciones definidas de servicio (figura 57).

- Measurement Point Layer: es la capa más baja de la arquitectura. Recoge medidas de la red, transforma los resultados en un formato estándar, y publica la información en un archivo a medida, o en otro servicio.
- Service Layer: La capa de servicio incluye la gestión de datos, manipulación, y los servicios de transformación y una colección de los servicios que proporcionan la autenticación y autorización estándar, descubrimiento del servicio, etc. La capa de servicio no es una simple capa in-and-out, sino que contiene el conducto o los servicios compuestos como el archivo de medida que es un servicio y un consumidor de servicios a la vez.

- Interface Layer: La capa de interfaz proporciona a los clientes representaciones útiles de la aplicación.

Los servicios: Los servicios actualmente existentes tienen siete categorías:

- Measurement Point (MP) Service: Crea y/o publica información de la monitorización relacionada con las medidas activas y pasivas
- Measurement Archive (MA) Service: Almacena y publica información de la monitorización.
- Look up Service (LS): Coloca todos los servicios que participan y sus capacidades.
- Topology Service (ToS): proporciona la información de la topología de la red.
- Authentication Service (AuS): Gestiona el acceso del nivel de dominio a los servicios.
- Transformation Service (TrS): realiza la manipulación (agregación, estadística) en conjuntos de datos disponibles.
- Resource Protector (RP) Service: arbitra el uso de los recursos limitados de medida basados en la política del uso del dueño del recurso.

Uso del sistema: Los servicios MP en la capa más baja crean o recogen datos de medida de la red. Los operadores de red mantienen con frecuencia el acceso exclusivo de la gestión a sus dispositivos de red por razones operacionales y de seguridad. Los operadores de red pueden utilizar el marco del perfSONAR desarrollando servicios MP que preguntan a sus dispositivos de red para obtener información del estado y entregan esta información a los servicios MA. Esto proporciona una funcionalidad importante de la abstracción de los datos aislando el método usado para obtener los datos, de la representación estandarizada de la publicación de los datos del perfSONAR. Esto permite a la capa media de servicios del perfSONAR procesar y analizar datos de diversas fuentes dentro de un dominio, o de fuentes a través de dominios múltiples, usando un solo interfaz estandarizado.

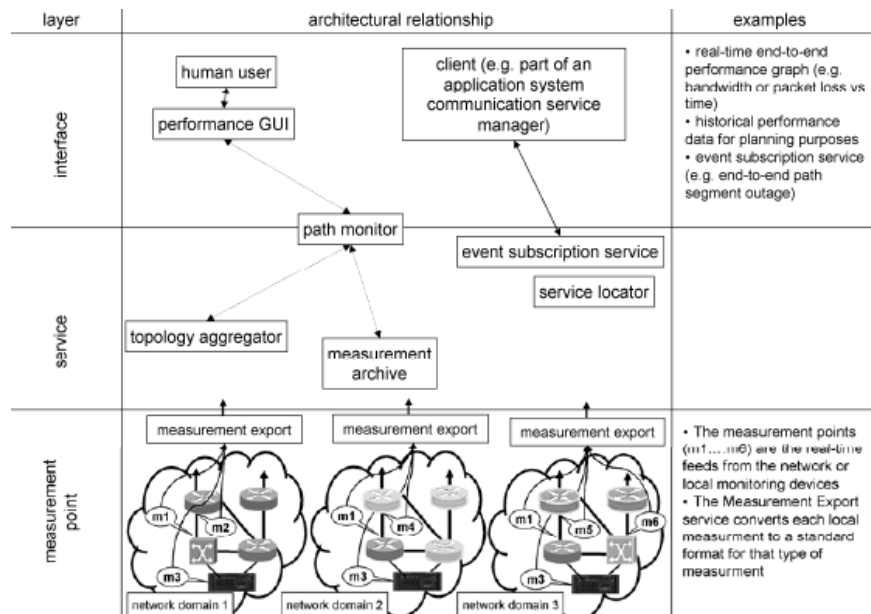


Figura 57: Arquitectura de PerfSONAR.

Esta arquitectura proporciona una separación clara entre las políticas acerca de cómo ha de acceder el control local MP a la infraestructura de la red, y las políticas que gobiernan cómo los servicios perfSONAR internos y externos acceden a los datos resultantes en los servicios MA. Tiene otras ventajas tales como permitir que los consumidores múltiples compartan los mismos datos de tal modo que reducen la carga de medición en el sistema subyacente. La capa media de perfSONAR contiene un sistema de servicios cooperantes, incluyendo el MA, el LS, el ToS, el TrS, y el AuS. Estos servicios pueden ser utilizados individualmente, o en conjunto para proporcionar el acceso uniforme a las mediciones de la red a través de dominios múltiples. Todos los servicios colocan su presencia y capacidades con el LS de su dominio local. Los LS cooperan para funcionar como un registro global a través de todos los dominios. Esto permite que los servicios se encuentren dentro de un dominio, y permite que las aplicaciones encuentren servicios a través de dominios múltiples. El LS permite que los MP localicen el MA que puede almacenar sus resultados. Permite que las aplicaciones del usuario localicen el MA que contiene datos del interés.

Los servicios ToS soportan el análisis automatizado de la red identificando la estructura subyacente en las redes y proporcionando la información sobre cómo se interconectan los dominios múltiples de la red. Esta capacidad será esencial en los entornos futuros del establecimiento de una red donde los servicios del circuito alterarán dinámicamente la infraestructura subyacente de la red usada por aplicaciones en tiempo real.

El MA se puede configurar para aceptar y para almacenar peticiones de instalación así como solicitudes de publicación. La petición de publicación incluye la suscripción, y los resultados se envían directamente al cliente (o indirectamente vía un TrS). Como cliente, el MA registra su propia presencia con un LS, se suscribe en un MP, al otro MA, o a los ToS, y publica la medición de los datos a los suscriptores.

El MA puede enviar peticiones de la disponibilidad y de la autorización del recurso al RP.

Monitorización del Multi-Dominio: El primer despliegue de la producción del entorno del perfSONAR es la monitorización multi-dominio de la red privada óptica de la red de LHC (LHCOPN o OPN) (figura 58). LHCOPN es la red que transfiere datos de la instalación Tier-0 del LHC a los Centros de Datos Tier-1 del CERN en varios países.

En este caso el perfSONAR proporciona un conjunto de convenios para representar datos de la red en un formato común, junto con el enfoque de SOA que permite que los diversos servicios componentes del perfSONAR se utilicen para ensamblar aplicaciones de monitorización para diversos propósitos.

Oper. State: Up
Admin. State: Normal Oper.

Domain	CERN			USLHCNET			
Link Structure	EP	←-----	-----→	DP	↔	DP	←-----
Type	EndPoint	ID Part Info	ID Part Info	Demarc	Domain Link	Demarc	ID Part Info
Local Name	CERN-T0	S513-C-BE1	CERN-FERMI-LHCOPN-001-GVA-CERN	USLHCNET-GEN	CERN-FERMI-LHCOPN-001-GVA-CHI	USLHCNET-CHI	CERN-FERMI-LHCOPN-001-CHI-ESNET
State Oper.	-	Up	Up	-	Up	-	Up
State Admin.	-	Normal Oper.	Normal Oper.	-	Normal Oper.	-	Normal Oper.
Timestamp	-	2007-04-08 T05:04:08+02:00	2007-04-08 T05:04:11+02:00	-	2007-04-08 T05:04:53+02:00	-	2007-04-08 T05:03:59+02:00

Page generated

ESNET				FERMI				
-----→	DP	↔	DP	←-----	DP	↔	EP	
ID Part Info	Demarc	Domain Link	Demarc	ID Part Info	ID Part Info	Demarc	Domain Link	EndPoint
CERN-FERMI-LHCOPN-001-STARLIGHT-Tail	ESNET-STARLIGHT	CERN-FERMI-LHCOPN-001-FERMI-STARLIGHT	ESNET-FERMI	CERN-FERMI-LHCOPN-001-Site-Tail	md8	FERMI-ESNET	md2	FERMI-T1
Up	-	Up	-	Up	Up	-	Up	-
Normal Oper.	-	Normal Oper.	-	Normal Oper.	Normal Oper.	-	Normal Oper.	-
2007-04-08 T01:40:37.0	-	2007-04-08T01:40:37.0	-	2007-04-08 T01:40:37.0	2007-04-08 T01:40:01.0-6:00	-	2007-04-08 T01:40:01.0-6:00	-

Figura 58: Vista de E2Emon generada de los datos para un link OPN.

Los servicios MP del perfSONAR se desarrollan dentro de cada dominio de la red para monitorizar los enlaces relacionados con cada dominio OPN. Algunos dominios están proporcionando la información de estado en tiempo real directamente de su MP. Otros dominios hacen que el MP almacene los datos en un MA, que publica la información actual e histórica.

El MP en cada dominio consiste en dos componentes. El componente específico del dominio en las diversas redes típicamente interfaces con el sistema de monitorización operacional de la red para obtener los datos del estado del enlace para el tramo de la trayectoria end-to-end dentro de esa red particular. Virtualmente cada red hace la monitorización interna de manera diferente, forma en la que se ha desarrollado históricamente junto con la red. El componente del perfSONAR de cada MP toma los datos que resultan, genera un archivo estándar de XML, y lo publica vía el interfaz del servicio del MP, o lo envía a un MA para archivar y publicar. Esta información es utilizada por una aplicación llamada E2Emon (aplicación de monitorización que utiliza protocolos del perfSONAR).

E2Emon utiliza protocolos del perfSONAR para recuperar el estado del circuito actual cada minuto, más o menos, de los MA's y MP's en todos los dominios que soportan los circuitos. E2Emon es en sí mismo un servicio que produce la muestra basada en Web, en tiempo real del estado total de la red, y genera alarmas cuando uno de los informes del MP o del MA desencadena problemas. El interfaz web para un solo enlace se muestra en la figura 58, y la visión amplia de OPN se muestra en la figura 59.

[Start page](#)

E2ECU view

All E2E Links

Problem Links

Domain view

CERN

DEN

ESNET

FERMI

GARR

GRANT2

IN2P3

RENATER

SURFNET

SWITCH

USLHCNET

Project view

IGTMD

LHCOPN

Availability Statistics

Current Month

Last Month

All E2E Links

E2E Link ID	State	Oper	State Admin	Additional Info
CERN-FERM-LHCOPN-001	Up		Normal Oper.	
FERMI-IN2P3-IGTMD-002	Down		Normal Oper.	Error: E2E Link is not contiguous (End Point missing or gap found) Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
GRIDKA-SARA-LHCOPN-001	Up		Normal Oper.	Error: E2E Link is not contiguous (End Point missing or gap found) Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
FERMI-IN2P3-IGTMD-001	Down		Trouble Shooting	Error: E2E Link is not contiguous (End Point missing or gap found) Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
GRIDKA-IN2P3-LHCOPN-001	Down		Normal Oper.	Error: E2E Link is not contiguous (End Point missing or gap found) Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
CERN-BNL-LHCOPN-001	Up		Normal Oper.	
CERN-SARA-LHCOPN-001	Up		Normal Oper.	
CERN-GRIDKA-LHCOPN-001	Up		Normal Oper.	
CERN-FERM-LHCOPN-002	Up		Normal Oper.	Error: E2E Link is not contiguous (End Point missing or gap found) Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
CERN-SARA-LHCOPN-002	Up		Normal Oper.	Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
CERN-ONAF-LHCOPN-001	Up		Normal Oper.	
CERN-SARA-LHCOPN-002	Up		Normal Oper.	Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
CERN-RAL-LHCOPN-001	Up		Normal Oper.	Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
CERN-BNL-LHCOPN-002	Up		Normal Oper.	Error: E2E Link is not contiguous (End Point missing or gap found) Warning: Operational state is known not for all involved links Warning: Administrative state is known not for all involved links
CERN-SARA-LHCOPN-003	Up		Normal Oper.	Warning: Operational state is known not for all involved links

Figura 59: Vista de E2Emon generado de los datos de todos los links OPN que muestran el estado operacional y el administrativo de cada link.

Estas herramientas están siendo utilizadas por el E2ECU (End-to-End Coordination Unit), que es una función del Network Operations Center de GÉANT que proporciona la gestión total de los circuitos de OPN.

Otro uso importante del multi-dominio del perfSONAR es para la monitorización del rendimiento del path. Esto presenta no sólo el estado operacional de la trayectoria como en el ejemplo anterior, sino que también proporciona el rendimiento en tiempo real tal como la utilización de la trayectoria y/o el descarte de paquetes.

Las herramientas de monitorización del rendimiento de paths múltiples están en desarrollo. Un ejemplo - Traceroute Visualizer - se ha desplegado aproximadamente en 10 redes de R&E en los E.E.U.U. y en Europa tiene por lo menos algunos de los servicios requeridos por el MA para soportar la herramienta.

La entrada del usuario a la herramienta es un “traceroute” entre los elementos de una aplicación distribuida que define la trayectoria a través de la red IP. La herramienta analiza la trayectoria y la información de la topología que se recupera de servicios del perfSONAR; entonces solicita los servicios del MA en las redes que intervienen. El MA retorna la información solicitada de la utilización, que se pasa a una herramienta de representación gráfica. A modo de ejemplo, la trayectoria entre el laboratorio nacional de Lorenzo Berkeley y la Poznań, centro del supercomputador de Polonia implica el cruzar cinco límites del dominio y se demuestra en la figura 60 y en la 61.

path	domain
traceroute to www6.infn.it (193.206.84.223), 64 hops max, 40 byte packets	
1 ln1000gw (131.243.2.1) 0.340 ms 0.306 ms 0.271 ms	LBNI LAN
2 er1kgw (131.243.128.5) 2.325 ms 2.551 ms 1.885 ms	
3 lbl2-ge-lbn1.es.net (198.129.224.2) 198.665 ms 1.261 ms 1.447 ms	
4 slacmr1-lblmr1.es.net (134.55.219.10) 1.598 ms 1.409 ms 1.451 ms	ESnet (SF Bay MAN)
5 snv2mr1-slacmr1.es.net (134.55.217.2) 1.886 ms 1.795 ms 1.739 ms	
6 snv2sdn1-snv2mr1.es.net (134.55.207.37) 1.740 ms 1.896 ms 1.742 ms	
7 denvr1-snv2sdn1.es.net (134.55.220.49) 46.050 ms 29.245 ms 28.937 ms	ESnet WAN nnp
8 chlcdr1-denvr1.es.net (134.55.209.46) 52.483 ms 52.485 ms 52.484 ms	
9 chlcdr1-a-chlcdr1.es.net (134.55.218.101) 52.462 ms 52.496 ms 52.486 ms	
10 washdr1-chlcdr1.es.net (134.55.218.96) 69.152 ms 69.164 ms 69.154 ms	GEANT WAN core
11 washcr1-sdn2-washdr1.es.net (134.55.220.53) 69.155 ms 69.016 ms 69.066 ms	
12 esnet-washrt1-fra.de.geant2.net (62.40.126.77) 161.564 ms 161.518 ms 161.588 ms	
13 so-6-2-0-rt1.gen.ch.geant2.net (62.40.112.21) 169.615 ms 169.584 ms 169.669 ms	GARR (Italian R&E net) core
14 so-2-0-0-rt1.mil.it.geant2.net (62.40.112.34) 177.056 ms 177.070 ms 176.924 ms	
15 gar1-gwrt1.mil.it.geant2.net (62.40.124.130) 177.070 ms 176.960 ms 176.927 ms	
16 rt1-m1-rt-m2.m2.garr.net (193.206.134.190) 177.363 ms 177.199 ms 177.251 ms	INFN Frascati
17 rt-m2-rt-m2.m2.garr.net (193.206.134.230) 186.649 ms 189.231 ms 186.571 ms	
18 rt-m2-rc-fra.fra.garr.net (193.206.134.214) 187.135 ms 187.042 ms 187.159 ms	
19 rc-fra-nu-inf.fra.garr.net (193.206.138.206) 187.161 ms 187.138 ms 187.168 ms	
20 ***	
21 www6.infn.it (193.206.84.223) 187.324 ms 187.123 ms 187.162 ms	

Figura 60: Vista de la aplicación del path end-to-end.

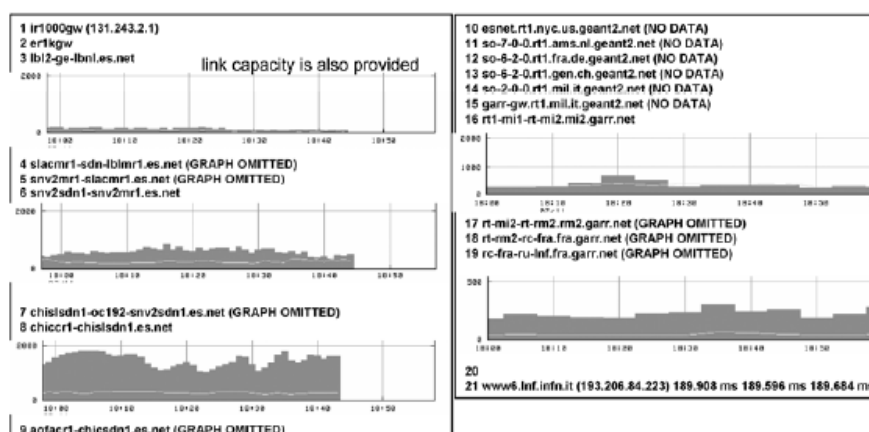


Figura 61: Aplicación del path de ida del tráfico mostrado en barras en aquellos interfaces de los dispositivos de red que tienen asociados servicios MP.

Estado: el perfSONAR se está desarrollando con la colaboración entre organizaciones de 25 redes de los E.E.U.U. y Europa. El marco básico es completo y se están documentando los protocolos. Se están desarrollando y se están desplegando los nuevos servicios [78, 79].

Calculadora de la presencia de fallo en la red: es importante solucionar el problema de determinar el impacto causado por el fallo de un elemento particular de la red, y proporcionar esta información a la aplicación. ESnet ha estado experimentando con un enfoque automatizado para solucionar este problema. El enfoque implica dos cuestiones: 1) determinar exactamente la topología dinámica de la red y 2) usar la topología para determinar el estado actual de la red total.

Mapeo de la Topología: Para monitorizar exactamente la red se debe modelar exactamente la red. Esto se logra supervisando cada interfaz de la red y derivando un modelo exacto de la conectividad de la capa IP de la red sobre cada hora. Los cambios diarios de la conectividad de la capa IP que ocurren en el transcurso de operaciones regulares se capturan cada noche y están archivados para poder contestar preguntas retrospectivas sobre la conectividad.

Calculadora de la presencia de fallo en la red: La calculadora computa los dispositivos (routers, interfaces) que serán aislados de la red, dada una lista de routers e interfaces fuera de servicio. La topología actual de la red se utiliza para crear una lista de “vertices” y de “edges”. Un gráfico de conexiones de la red es derivado examinando cada “edge” en el modelo de la topología de la red y ensamblando los sistemas de routers en cada extremo de cada “edge”. En

condiciones normales cuando la red tiene el 100% de disponibilidad, el proceso de todos los edges da lugar a un solo sistema de dispositivos que representan la red ESnet completamente conectada según lo representado por todos los “vértices” que se muestran en un solo sistema de “vértices” conectados. Para computar el efecto de quitar un sistema de routers o de enlaces, cada “edge” conectado, con los argumentos dados, se quita de la lista del “edge” antes de ejecutar el algoritmo de la conexión.

Los dispositivos afectados que resultan, terminan en sistemas de vértices que están separados unos de otros y por lo tanto inalcanzables. Cada sistema es una parte desconectada de la red, esto se ilustra en la figura 62. Esta clase de representación se puede combinar con una descripción de la trayectoria más o menos de la misma manera que se hace para el servicio de monitorización de la trayectoria del perfSONAR, para proporcionar la información de la aplicación sobre el impacto de fallos previstos en la red.

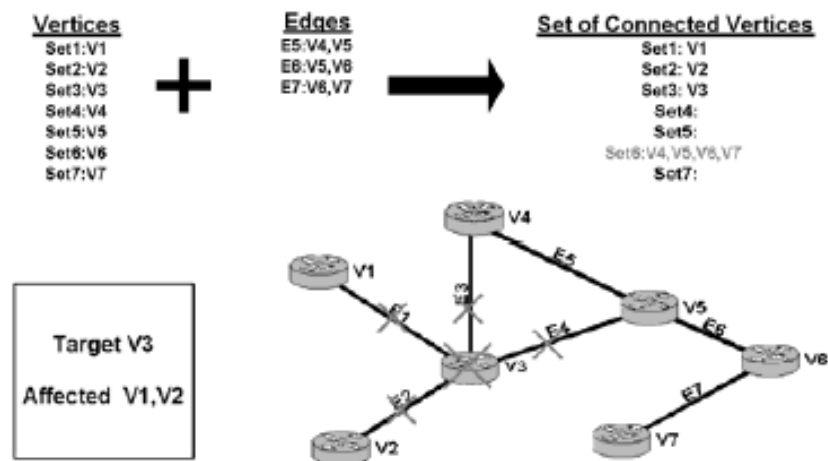


Figura 62: Ejemplo de cálculo de la presencia de fallo resultante de un fallo de un solo elemento de red.

5. Conclusiones

SOA ha entrado con fuerza en las empresas a nivel mundial. Son muchas las definiciones y las pautas que se han publicado para describir como llevar a cabo la compleja tarea de migración. SOA no sólo se presenta como una arquitectura, sino como un paradigma, un conjunto de mejores prácticas, una estrategia TI, una metodología de diseño, etc. que da como resultado una arquitectura de red mejorada, actualizada y preparada para cambios y futuras migraciones.

Todo documento que habla de SOA hace mención a ventajas como reutilización, flexibilidad, escalabilidad, interoperabilidad, eficiencia y reducción de costes.

La constante evolución de las tecnologías y dispositivos hace necesaria una transformación de la arquitectura de red, tanto a nivel software como hardware, para lograr máxima eficiencia, menor coste y pérdidas en tiempo de integración mínimas. Las redes actuales han de proporcionar diversas aplicaciones multimedia, con diferentes requisitos de QoS, sobre una infraestructura común de transporte basada en IP. Además, Internet con los nuevos modelos de computación distribuida como grid computing, redes peer-to-peer (P2P), el cloud computing, etc. está continuamente renovándose y los servicios actuales ven necesarias las conexiones de largo recorrido, lo cual exige la interoperabilidad entre dominios. El cloud computing está a la orden del día y es gracias a la funcionalidad que es capaz de ofrecer. Los clientes acceden a los datos y a las aplicaciones en cualquier momento y en cualquier lugar del mundo a través de la nube. Son muchos los que dicen que SOA y cloud computing se complementan uno al otro, es decir el cloud computing necesita de un enfoque SOA para que pueda desarrollarse y funcionar de manera eficiente. En estas conexiones la infraestructura a larga distancia se ha de basar en protocolos que permitan a los clientes poder escoger sus parámetros de TE como el ancho de banda, el retardo, etc. parámetros que se pueden controlar en conexiones ópticas a través, por ejemplo, del protocolo GMPLS en el plano de control o a partir de la introducción de una capa de servicio específica para las interacciones interdominio, solución que permite reducción de costes comparado con GMPLS.

SOA, AON y autonomic computing están hoy presentes en la mayoría de empresas para obtener un entorno escalable, seguro, y autogestionable para poder reducir el tiempo de

mercado de nuevas aplicaciones de Internet, minimizar los gastos de gestión de los proveedores de servicios, reduciendo la complejidad de desarrollo.

El despliegue de SOA es único para cada empresa, por ello, el departamento de TI ha de tener un amplio conocimiento de toda la empresa para poder ofrecer una arquitectura capaz de integrar los sistemas a los procesos de negocio (ha de soportar cualquier tipo de integración requerida) y obtener así una estrategia la cual derive en una ventaja global para la empresa. Para llegar a implementar SOA hay que cambiar el concepto organizativo, lo cual requiere de un SOA Governance, que dota a las organizaciones de la capacidad de aplicar las pautas SOA, estándares, etc. para poder llegar a su finalidad, con éxito, de alinear las necesidades de negocio, aplicaciones y servicios con las TI (ha de definir qué hacer, quién lo hace, cómo hacerlo y cómo medirlo).

SOA es un escenario del que toda empresa obtiene básicamente dos ventajas: la reutilización del software, que hace posible migraciones sin la necesidad de empezar de cero y está preparada para el crecimiento futuro de las aplicaciones; y la independencia de la plataforma tecnológica así como de las aplicaciones, ya que el interfaz se define de forma neutral para poder utilizar cualquier tecnología y/o aplicación de una manera sencilla y ágil, lo que hace posible una mayor reutilización de los módulos y sistemas.

La base para la interconexión de las aplicaciones de una manera neutral la forma los Web Services. SOA está basado en estándares abiertos y cabe mencionar que el más utilizado actualmente en las organizaciones son los Web Services. Gracias a este conjunto de protocolos se hace posible la publicación, el descubrimiento y la utilización de los servicios con una tecnología neutra y de forma estándar para que puedan ser utilizados fácilmente incluso por el resto de empresas (a través del registro de los servicios). Los Web Services son el estándar apoyado por la industria y las empresas sobretodo porque dota de la ventaja de la interoperabilidad, pero no necesariamente SOA se ha de implementar a partir de esta tecnología.

Es interesante destacar, como el propio nombre de SOA indica, la orientación a servicio. La arquitectura orientada a servicio aumenta una capa y el nivel de granularidad, es decir, un servicio representa una funcionalidad de negocio, mientras un componente representa las entidades de negocio y las reglas que las operan. Estos servicios son independientes y al tener sus interfaces bien definidos y publicados, la ubicación de éstos ya no es importante.

Gracias a las herramientas de gestión de procesos de negocio, básicamente BPM, SOA consigue un aumento de productividad en el desarrollo de sistemas, ya que la codificación de la lógica de negocio que se hacía de manera manual, lenta y era propensa a errores, ahora se consigue de una manera rápida, segura y ganando las empresas en agilidad y flexibilidad. Con SOA y BPM se hace sencillo poder invocar Web Services remotos, de cualquier parte del mundo, y ejecutarlos siguiendo las reglas de negocio definidas.

La orquestación es otra de las claves para SOA, ya que es necesaria una herramienta para poder realizar la composición de los distintos servicios definidos. En su mayoría, se apuesta por el lenguaje BPEL, aunque es cierto que se empieza a hablar de SCA creado por IBM y BEA para mejorar aquellos puntos en que BPEL no resulta útil, ya que flojea en cuanto la complejidad de la definición de procesos de negocio aumenta.

Tras el estudio de diferentes enfoques, podemos destacar la arquitectura EWA, entorno en el que IBM ha desarrollado SOA y el cual está siendo utilizado en sus clientes. El concepto que introduce IBM para realizar la migración es el de bus de servicio, al cual están conectados todos los servicios descritos y a partir del cual se puede acceder a cualquiera de dichos servicios. Este concepto es uno de los más importantes y aplicados por todas las empresas para desplegar SOA. El ESB enruta mensajes entre los servicios, convierte los protocolos de transporte entre servicios, transforma los formatos de los mensajes entre servicios y distribuye eventos de negocio entre fuentes heterogéneas.

El artículo sobre ASMF nos describe cómo diseñar un entorno que integre SOA, AON y autonomic computing. El ASB es el motor en este entorno y se desarrolla como un elemento autónomo que a su vez con el resto de ASB's forman una estructura P2P autogestionable que permite la composición de los servicios y la gestión de los recursos. La implementación de esta arquitectura tiene su punto más importante en la encapsulación de los recursos en Web Services manejables.

De acuerdo con BT la red de nueva generación ha de centrar su desarrollo en el dominio de ejecución de servicio, la cual cosa confían en desarrollarlo implementando SOA. El punto fuerte por el que SOA es el entorno elegido, es su idoneidad para dotar de la reutilización de capacidades a los usuarios.



ESNET siendo una red de alta velocidad que da servicio a los científicos del DOE de Estados Unidos y a colaboradores de todo el mundo, y que necesariamente ha de ser una red flexible, robusta, de gran seguridad y con gran ancho de banda para poder manejar grandes cantidades de datos, ha apostado por SOA como arquitectura de red. Tras varios casos de estudio se entiende que la red ha de evolucionar hacia nuevas aplicaciones y nuevas capacidades orientadas a servicio.

6. Referencias

- [1] Carlos Billy Reynoso: "Introducción a la arquitectura de software", marzo 2004.
- [2] http://ingenieria.ucaldas.edu.co/auditoria/index.php/SOA_service_oriented_architectures
- [3] WWISA (Worldwide Institute of Software Architects): "Philosophy", 1999.
- [4] Karl Reed and Jason Baragly: "Why we need a different view of software architecture", The Working IEEE/IFIP Conference on Software Architecture (WICSA), Amsterdam 2001.
- [5] P. Clements: "Coming attractions in Software Architecture", technical report 1996.
- [6] Gartner: "Predicts 2003: SOA is changing Software", December 2002.
- [7] Javier Rosado Carrizo: "Arquitectura Orientada a Servicio".
- [8] Billy Reynoso: "Arquitectura Orientada a Servicios".
- [9] Centro de Alto Rendimiento Accenture: "Arquitectura Orientada a Servicios (SOA)", 2008.
- [10] Kishore Channabasavaiah and Kerrie Holley, IBM Global Services, and Edward M. Tuggle, Jr., IBM Software Group, "On demand operating environment solutions: Migrating to a service-oriented architecture", white paper, April 2004.
- [11] David Sprott and Lawrence Wilkes CBI Forum, "The architecture Journal: Understanding Service-Oriented Architecture", January 2004.
- [12] ARQ-RFC "Pautas y recomendaciones para SOA v.091", July 2006.
- [13] Oracle SOA Governance Solution, <http://www.oracle.com/technologies/soa/docs/soa-governance-datasheet.pdf>, 2009.
- [14] <http://www.soaagenda.com>
- [15] Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities, 2008.
- [16] María de los Ángeles Ibarra, "Tecnología On Demand", 2006.
- [17] <http://www.areahospedaje.com>
- [18] <http://www.google.es/trends>
- [19] R. Buyya, D. Abramson and S. Venugopal: "The Grid Economy". Proceedings of the IEEE, 93(3): 698-714, IEEE Press, USA, March 2005.
- [20] S. Venugopal, X. Chu, and R. Buyya: "A Negotiation Mechanism for Advance Resource Reservation using the Alternate Offers Protocol". In Proceedings of the 16th International Workshop on Quality of Service (IWQoS 2008), Twente, The Netherlands, June 2008.
- [21] Y. Fu, J. Chase, B. Chun, S. Schwab and A. Vahdat. SHARP: "An architecture for secure resource peering". ACM SIGOPS Operating Systems Review, 37(5):133-148, December 2003.
- [22] K. Lai, L. Rasmusson, E. Adar, L. Zhang, and B. A. Huberman. Tycoon: "An implementation of a distributed, market-based resource allocation system". Multiagent and Grid Systems, 1(3):169-182, 2005.
- [23] A. AuYoung, B. Chun, A. Snoeren and A. Vahdat: "Resource allocation in federated distributed computing infrastructures". In Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT Infrastructure (OASIS 2004), Boston, USA, October 2004.
- [24] D. E. Irwin, J. S. Chase, L. E. Grit, A. R. Yumerefendi, D. Becker and K. Yocum: "Sharing networked resources with brokered leases". In Proceedings of the 2006 USENIX Annual Technical Conference (USENIX 2006), Boston, USA, June 2006.

- [25] Elena Frau: "Arquitectura Orientada a Servicios".
<http://www.slidefinder.net/S/SOA/894646>
- [26] Christopher Ferris: "What Are Web Service?", 2003.
- [27] http://es.wikipedia.org/wiki/Servicio_web
- [28] David Pereira: "Arquitecturas Empresariales. Orientación a servicios (SOA) y gestión de Procesos de Negocio (BPM)", 2009.
- [29] Juan Ramón Pérez: "Servicios Web: orquestación y Coreografías", marzo 2007.
- [30] S.P.Bravo: "SOA y la integración con el modelo de negocio de la empresa".
- [31] Especificación de la arquitectura del BPS: "Arquitectura de Referencia SOA", octubre 2009.
- [32] Huijbart Aalbers: "Introducción a SOA (II)".
- [33] J. Pasley, "How BPEL and SOA are Changing Web Services Development," IEEE Internet Comp., vol. 9, May–June 2005,
- [34] Cisco Systems, "Cisco Application-Oriented Networking," white paper, 2005;
- [35] IBM, "An Architectural Blueprint for Autonomic Computing (4th ed.)," white paper, June 2006.
- [36] Yu Cheng, Alberto León-García y Ian Foster: "Toward an Autonomic Service Management Framework: A Holistic, Vision of SOA, AON, and Autonomic Computing", May 2008.
- [37] K. Czajkowski, I. Foster, and C. Kesselman, "Agreement- Based Resource Management," Proc. IEEE, vol. 93, March 2005.
- [38] ITU-T Rec M.3060/Y.2401, "Principles for the Management of the Next Generation Networks," March 2006.
- [39] G. Koloniari and E. Pitoura, "Peer-to-Peer Management of XML Data: Issues and Research Challenges," SIGMOD Record, vol. 34, June 2005.
- [40] K. Fujii and T. Suda, "Semantics-Based Dynamic Service Composition," IEEE JSAC, vol. 23, December 2005.
- [41] Fabio L. Verdi, Mauricio F. Magalhães, Eleri Cardozo,Edmundo R. M. Madeira and Annikki Welin: "A Service Oriented Architecture-based Approach for Interdomain Optical Network Services", Journal of Network and Systems Management, Vol. 15, No. 2, June 2007.
- [42] M. P. Papazoglou and D. Georgakopoulos, Service Oriented Computing, Communications of the ACM, Vol. 46, No. 10, pp. 25–28, October 2003.
- [43] R. Boutaba, W. Golab, Y. Iraqui, and B. St. Arnaud, Lightapths on Demand: A Web-Services-Based Management System. IEEE Communications Magazine, Vol. 42, No. 7, pp. 2–9, July 2004.
- [44] D. L. Truong, O. Cherkaoui, H. Elbiaze, N. Rico, and M. Aboulhamid, A Policy-based approach for user controlled Lightpath Provisioning. IFIP/IEEE NOMS, pp. 859–872, April 2004.
- [45] M. Brunner, G. Nunzi, T. Dietz and I. Kazuhiko: "Customer-Oriented GMPLS Service Management and Resilience Differentiation. eTransactions on Network and Service Management", pp. 2–12, Fourth Quarter 2004.
- [46] M. P. Howarth, P. Flegkas, G. Pavlou, N. Wang, P. Trimintzios, D. Griffin, J. Griem, M. Boucadair, P. Morand, A. Asgari, and P. Georgatsos, Provisioning for Interdomain Quality of Service: the MESCAL Approach. IEEE Communications Magazine, Vol. 43, No. 6, pp. 129–137, June 2005.
- [47] L. Fang, N. Bitá, J. Roux, and J. Miles, Interprovider IP-MPLS Services: Requirements, Implementations, and Challenges. IEEE Communications Magazine, Vol. 43, No. 6, pp. 119–128, June 2005.

- [48] A. Farrel, J.-F. Vasseur, and J. Ash, Path Computation Element (PCE)-based Architecture. IETF RFC 4655, August 2006.
- [49] A. Farrel, J.-F. Vasseur, and A. Ayyangar, A Framework for Inter-Domain MPLS Traffic Engineering. IETF RFC 4726, November 2006.
- [50] T. Takeda, D. Brungard, D. Papadimitriou, and H. Ould-Brahim, Layer 1 Virtual Private Networks: Driving Forces and Realization by GMPLS. IEEE Communications Magazine, Vol. 43, No. 7, pp. 60–67, 2005.
- [51] S. Tomic, Issues of Resource Management in Two-Layer GMPLS Networks with Virtual Network Service. IEEE Globecom, pp. 1803–1807, 2004.
- [52] R. S. Ravindran, P. Ashwood-Smith, H. Zhang, and G.-Q. Wang, Multiple Abstraction Schemes for Generalized Virtual Private Switched Networks. IEEE Canadian Conference on Electrical and Computer Engineering – CCECE, pp. 0519–0522, May 2004.
- [53] H. Ould-Brahim and Y. Rekhter, GVPN Services: Generalized VPN Services using BGP and GMPLS Toolkit. IETF draft, work in progress, February 2005.
- [54] B. Arnaud, UCLP Roadmap: Web Services Workflow for Connecting Research Instruments and Sensors to Networks. Draft, December 2004.
- [55] V. de Souza and E. Cardozo, A Service Oriented Architecture for Deploying and Managing Network Services. Proceedings of the 3rd International Conference on Service Oriented Computing (ICSOC '05), LNCS-Springer-Verlag, Vol. 3826, pp. 465–477, December 2005.
- [56] D. Clark, C. Partridge, R. T. Braden, B. Davie, S. Floyd, V. Jacobson, D. Katabi, G. Minshall, K. K. Ramakrishnan, T. Roscoe, I. Stoica, J. Wroclawski, and L. Zhang, Making the world (of communications) a different place. Report of a working session of the End-to-End Research Group – Internet Research Task Force, January 2005.
- [57] F. L. Verdi, R. Duarte, F. de Lacerda, E. Madeira, E. Cardozo, and M. Magalhães, Web Services based Provisioning of Connections in GMPLS Optical Networks. The Brazilian Symposium on Computer Networks (SBRC). Fortaleza, Brazil, May 2005.
- [58] F. Ricciato, U. Monaco, and D. Ali, Distributed Schemes for Diverse Path Computation in Multidomain MPLS Networks, IEEE Communications Magazine, Vol. 43, No. 6, pp. 38–146, June 2005.
- [59] OASIS UDDI Specification: <http://www.uddi.org/>.
- [60] C. Carvalho, F. L. Verdi, E. Madeira, and M. Magalhães, Policy-based Fault Management for Integrating IP over Optical Networks. The 5th IEEE International Workshop on IP Operations & Management (IPOM '05), LNCS-Springer-Verlag, Vol. 3751, pp. 88–97, October 2005.
- [61] F. L. Verdi, C. Carvalho, E. Madeira, and M. Magalhães, Policy-based Grooming in Optical Networks. 4th IEEE Latin American Network Operations and Management Symposium (LANOMS), pp. 125–136, August 2005.
- [62] T. Takeda, I. Inoue, R. Aubin, and M. Carugi, Layer 1 Virtual Private Networks: Service Concepts, Architecture Requirements, and Related Advances in Standardization. Communications Magazine, IEEE, Vol. 42, No. 6, pp. 132–138, 2004.
- [63] T. Takeda, H. Kojima, and I. Inoue, Layer 1 VPN architecture and its evaluation, 5th International Symposium on Multi-Dimensional Mobile Communications, Vol. 2, pp. 612–616, September 2004.
- [64] T. Takeda, H. Kojima, and I. Inoue, Optical VPN architecture and mechanisms. The 9th Asia-Pacific Conference, (APCC), pp. 751–755, 2003.
- [65] Jared T. Howerton: “Service Oriented Architecture and Web 2.0”, May-June 2007.



