11. SISTEMAS DE INTEGRACIÓN

Colaboración entre sistemas heterogéneos (y no tanto)

Autor: Esp. Damián Barry

11.1 SISTEMAS ORGANIZACIONALES E INTERACCIONES.

11.1.1 Actores, roles y necesidades

Dentro de los distintos sistemas organizacionales, ya sean privados, públicos o de gobierno, nos encontramos con una gran cantidad de actores que necesitan intercambiar servicios e información. Estas operaciones están sumamente atomizadas, requiriendo la intervención simultánea de los actores mencionados.

A dichas operaciones se agregan además intereses contrapuestos y necesidades de control y contralor que lo hacen más complejo aún.

Esta complejidad genera una gran cantidad de dialectos de comunicación entre las partes, provocando el clásico “teléfono descompuesto”. Debido esto, normalmente, a reglas de interpretación poco claras. Donde el principal damnificado es el hombre, donde normalmente es sometido a largas esperas y en algunos casos re-hacer los trámites y operaciones por dicha falta de interpretación.

Es por ello que la alta interoperabilidad de dichos servicios e información requiere de una interpretación precisa de lo que se solicita como servicio y de lo que se pretende informar o recibir como información.

De aquí que la mayor necesidad de las organizaciones y actores intervinientes sea la definición de estándares de servicios y comunicaciones que garanticen o por lo menos minimicen la posibilidad de confusiones y malas interpretaciones en el intercambio de información.

Esta situación no es menor si tenemos en cuenta que muchos de los servicios propuestos tienen una criticidad muy alta, como puede ser por ejemplo la correcta identificación de una persona como paciente en un hospital y el acceso a su historia clínica centralizada. Fallar en esta situación podría en algunos casos provocar la muerte del paciente por falta o mala interpretación de la información requerida.

11.1.2 Sistemas e interoperabilidad

La sociedad y su sistema de organización social, requiere que sus organizaciones, ya sean privadas, semipúblicas o públicas interaccionan tanto con los ciudadanos (en cualquiera de sus roles) como con otras organizaciones.

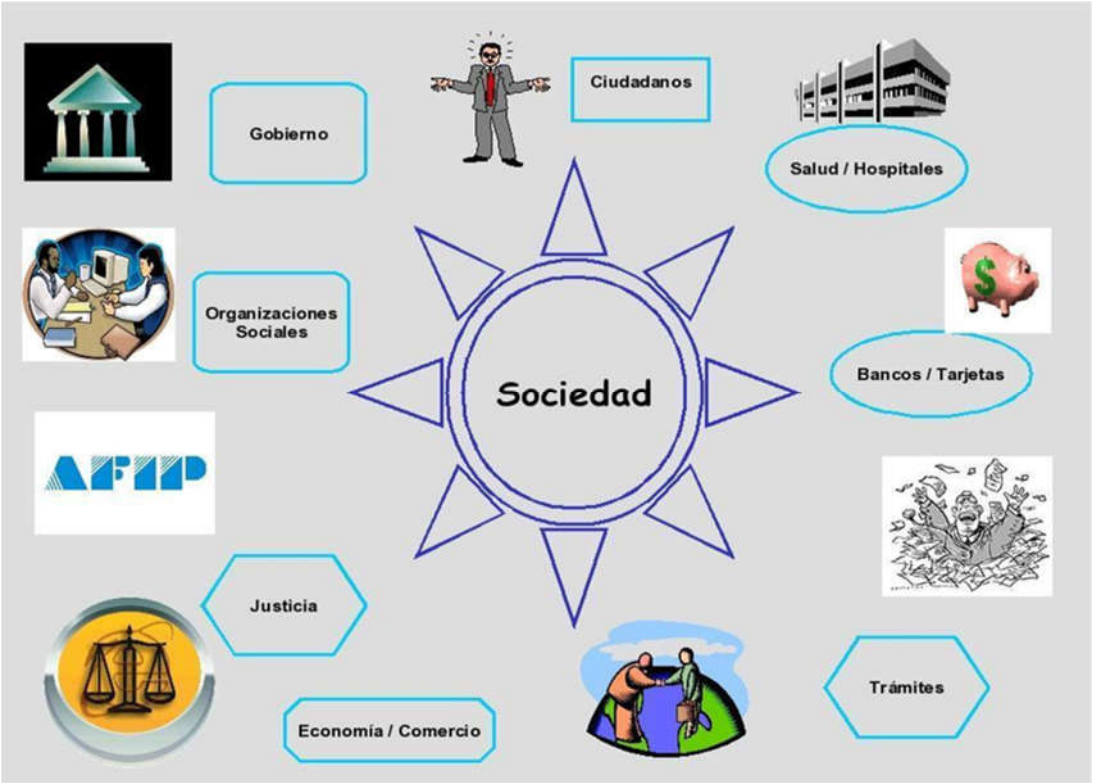

Figura 11.1

- Los ciudadanos están ligados a distintos organismos y organizaciones por diversos motivos dentro de sus obligaciones y derechos. También están vinculados a organizaciones privadas o semi-públicas por motivos de distinto grado de pertenencia de acuerdo a sus inclinaciones, actividades sociales y profesionales.
- En este sentido algunos ejemplos que podemos mencionar son:
  - Un maestro está ligado a la escuela e indirectamente al ministerio de educación.
  - Un empleado a la nómina de una empresa y al banco donde se le liquidan sus haberes.
  - Un paciente, está ligado al sistema de salud: hospitales, centros de atención, Obra social o prepaga.
  - Una Organización o empresa está vinculada a los gobiernos municipales, provinciales y nacionales debido a sus obligaciones.
  - Al igual que un ciudadano está vinculado a los gobiernos municipales, provinciales y nacionales debido a sus obligaciones.
- El nexo común es la sociedad en sí misma.
- El factor impulsor es la organización de dicha sociedad con sus actividades, sociales, económicas y las obligaciones y derechos de los ciudadanos y actividad privada (empresas y organizaciones).
- El estado está obligado a supervisar y ordenar todas las actividades ejecutadas dentro de una sociedad.

- Todos los integrantes de dicha sociedad tienen un alto nivel de interdependencia.

En la figura presentada a continuación se puede apreciar un ejemplo solamente para el área de salud:

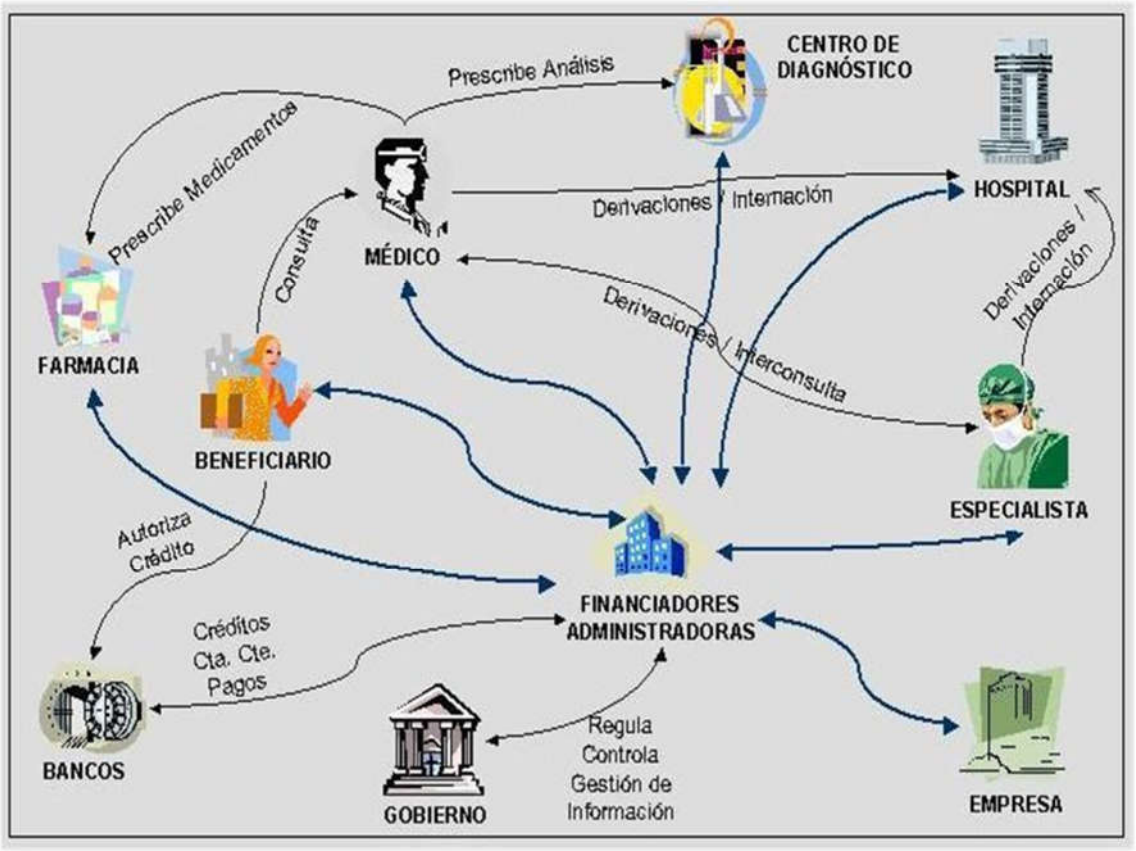

Figura 11.2

Claramente se puede observar el nivel de interoperabilidad, dependencia y atomización que nos llevan a concluir el siguiente diagnóstico:

- Sistema atomizado con muchos intermediarios.
- Industria de información intensiva.
- Gastos transaccionales y operacionales elevados con tendencia a la ineficiencia.
- Situación económica / financiera comprometida.
- Falta de políticas integrales y estándares.
- Escasa inversión en tecnología.

Es importante destacar el rol central que tienen los factores económicos dentro de esta interoperabilidad y el papel preponderante que juegan las organizaciones financiadoras y el gobierno como factores de equilibrio entre calidad y distribución de las prestaciones.

La evolución de los sistemas sociales hace pensar que estamos frente a una problemática de demanda infinita y recursos limitados y es aquí donde estas organizaciones tienen la responsabilidad de administrar y distribuir en toda la población el derecho y acceso a la información y un equilibrio entre las actividades desarrolladas en una sociedad.

11.2 ORGANIZACIONES DEDICADAS A LA INTEROPERABILIDAD

11.2.1 OASIS

OASIS, acrónimo de (Organization for the Advancement of Structured Information Standards) es un consorcio internacional sin fines de lucro que se orienta al desarrollo, consenso y adopción de estándares para el intercambio de información entre aplicaciones (e-Business, e-Government).

Los miembros del consorcio deciden cómo y qué trabajos se deben emprender a través de un proceso abierto y democrático.

Los trabajos técnicos abarcan algunas de las siguientes categorías entre otras: Adoption Services, Computing Management, Document-Centric Applications, e-Commerce, Law & Government, SOA, Standards Adoption, Web Services, XML Processing.

11.2.1.1 Historia

OASIS fue fundada inicialmente en 1993 como una asociación comercial, denominada “SGML Open”, para promover la adopción del estándar SGML. Principalmente mediante el desarrollo de actividades educativas, incluyendo también actividades técnicas como el intercambio de especificaciones en organizaciones administrativas.

En 1998, con el apogeo de XML en TI, SGML cambio su enfoque de SGML a XML, y cambio su nombre a OASIS. El área de interés del consorcio abarca desde promociones de adopción de estándares hasta el desarrollo de especificaciones técnicas de los mismos. En julio del 2000 se creó un nuevo comité técnico. En la actualidad hay cerca de 70 comités.

Durante 1999 OASIS se reunió con la UN/CEFACT, el comité de las Naciones Unidas para negociar los estándares de negocios, para que conjuntamente se desarrollen nuevas normas y estándares para el intercambio de información electrónica. La junta inicial se la llamo “ebXML” y se reunieron por primera vez en noviembre de 1999, fue establecida por un periodo de 3 años. En la reunión final bajo los primeros estatutos, en Viena, UN/CEFACT y OASIS coincidieron en dividir el

trabajo en las dos organizaciones y coordinar la finalización de los trabajos en comités específicos. En 2004 OASIS propuso la especificación de ebXML a la ISO TC154 donde fue aprobada como la ISO 15000.

Algunos de los estándares desarrollados por OASIS:

- **DITA** (OASIS Darwin Information Typing Architecture) es un módulo portable y extensible basado en el lenguaje XML para temas básicos como ayuda on-line, documentación de proyectos, asesoramientos y capacitación.
- **OpenDocument** (OASIS Open Document Format for Office Applications) es un formato de documento abierto para el respaldo de documentos de oficina tales como documentos de texto, planillas de cálculo, presentaciones, memos, y gráficos.
- **SAML** (Security Assertion Markup Language) es un estándar basado en XML para la seguridad en el intercambio, autenticación y autorización de información.
- **XRI** (eXtensible Resource Identifier) es un esquema compatible con URI y un protocolo de resoluciones usado por identificadores abstractos para la identificación y intercambio de recursos a través de dominios y aplicaciones.
- **XDI** (XRI Data Interchange) es un estándar para el intercambio, interconexión y sincronización de datos ("dataweb") mediante el uso de múltiples dominios y aplicaciones usando documentos XML (XRIs, eXtensible Resource Identifiers) y nuevos métodos distribuidos de control de datos llamados "linkContract".
- **LegalXML** es un estándar para el intercambio de información legal y jurídica entre todas las partes del sistema judicial.
- Advanced Message Queueing Protocol (**AMQP**) es el estándar para el intercambio de mensajes sincrónicos y asincrónicos mediante utilización de colas. Incluye ademas todos los estándares relacionados
- Biometric Identity Assurance Services (**BIAS**) SOAP Profile es el estandar que permite especificar un perfil SOAP que implementa las operaciones abstractas de Servicios que garantizan identidad biométrica (BIAS) especificadas en INCITS 442 como mensajes SOAP.
- Toda la definición de estándares del ecosistema para la implementación de **Web Services** incluyendo entre otros: Web Services Basic Reliable and Secure Profiles (WS-BRSP), Web Services Calendar (WS-Calendar), Web Services Secure Exchange (WS-SX), Web Services Security (WSS), Web Services Transaction (WS-TX), WS-BPEL Extension for People (BPEL4People), etc.

El consorcio es de carácter internacional y por lo tanto crea las normas para un mercado internacional. Con la oficina principal en Estado Unidos de América, el consorcio tiene la representación corporativa en Europa y Asia, y participación activa de miembros en los cinco continentes.

Los miembros incluyen:

- Usuarios que buscan asegurar sus requisitos comerciales.

- Agencias gubernamentales que quieren minimizar la superposición de normas y reducir el riesgo de la nueva tecnología.
- Los proveedores de software, impulsando la cooperación de la industria a través de normas y estándares.

Además, ofrece un rango de niveles de miembros para los usuarios y vendedores, gobiernos y universidades, grupos de comercio y proveedores de servicios.

Los archivos de los Comités Técnicos de OASIS, documentos y los correos electrónicos son públicamente accesibles incluso para los no-miembros y pueden ser supervisados y mejorados abiertamente.

OASIS se caracteriza por su gobierno y su forma de operación. Los mismos miembros fijan la agenda técnica, usando un proceso sencillo diseñado para promover el consenso en la industria y para unir esfuerzos dispares. Los trabajos completados son debatidos mediante el voto abierto (Ballot). Los funcionarios de la Junta Directiva de OASIS y la Junta Asesora Técnica son escogidos por elección democrática para servir por el término de dos años. La Dirección del consorcio esta basada en el mérito individual y no en su contribución financiera, situación corporativa o cita especial.

OASIS mantiene participación y relaciones con algunas de siguientes organizaciones:

- W3C
- ISO
- ISO/IEC JTC1
- ITU
- UNECE
- RosettaNet
- Y muchas otraS.

11.2.2 W3C

El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones miembros, el staff de tiempo completo y el público interesado, trabajan juntos para desarrollar estándares web.

La misión del W3C es llevar la World Wide Web a su máximo potencial mediante el desarrollo de protocolos y pautas que aseguren el crecimiento a largo plazo de la Web.

10.2.1 Principios de estándares abiertos

El 29 de agosto de 2012, cinco organizaciones mundiales líderes firmaron conjuntamente un acuerdo para afirmar y adherirse a un conjunto de Principios en apoyo del Paradigma de Estándares Moderno; un modelo abierto y de empoderamiento colectivo que permite mejorar radicalmente la forma en que las personas de todo el mundo desarrollan nuevas tecnologías e innovan para la humanidad.

10.2.2 Estándares

Los estándares del **W3C** definen una plataforma web abierta que permite el desarrollo de aplicaciones, contando con un potencial para permitir a los desarrolladores crear experiencias interactivas enriquecidas, impulsadas por grandes fuentes de datos disponibles en cualquier dispositivo. Aunque el alcance de la plataforma continúa evolucionando en permanente expansión, los líderes de la industria hablan casi al unísono sobre cómo HTML5 será la piedra angular de esta plataforma. Pero la fuerza total de la plataforma se basa en muchas más tecnologías que el W3C está creando, incluyendo CSS, SVG, WOFF, la pila de Web Semántica, XML y una variedad de API's.

W3C desarrolla estas especificaciones técnicas mediante un proceso diseñado para maximizar el consenso sobre el contenido de un informe técnico, para garantizar una alta calidad técnica y de difusión de los contenidos que le permiten obtener el respaldo de la comunidad en general.

Entre los principales estándares podemos mencionar:

- Estándares para el Diseño Web de aplicaciones, implican estándares para la creación y la representación de páginas web, incluidos HTML, CSS, SVG, Ajax y otras tecnologías para aplicaciones web ("aplicaciones web"). Esta sección también incluye información sobre cómo hacer que las páginas sean accesibles para personas con discapacidad (WCAG), internacionalizarse y hacer que funcionen en dispositivos móviles.
- Dispositivos Web: W3C se centra en tecnologías para permitir el acceso a la Web en cualquier lugar, en cualquier momento y con cualquier dispositivo (ubicuidad). Esto incluye el acceso a la Web desde teléfonos móviles y otros dispositivos móviles, así como el uso de tecnología Web en productos electrónicos de consumo, impresoras, televisión interactiva e incluso automóviles.
- La arquitectura web se centra en las tecnologías y los principios fundamentales que sustentan la web, incluidos los URI y HTTP.
- Web Semántica: Además de la clásica "Web de documentos", el W3C contiene una pila de tecnologías para admitir una "Web de datos", el tipo de datos que

se encuentran en las bases de datos. El objetivo final de la Web de datos es permitir que las computadoras realicen un trabajo más útil y desarrollen sistemas que puedan soportar interacciones confiables a través de la red. El término "Web Semántica" se refiere a la visión de W3C referidos a los datos enlazados. Las tecnologías de Web Semántica permiten a las personas crear repositorios de datos en la Web, construir vocabularios y escribir reglas para manejar datos. Los datos enlazados están potenciados por tecnologías como RDF, SPARQL, OWL y SKOS.

La visión del **W3C** para la Web implica la participación, el intercambio de conocimientos y, por lo tanto, la creación de confianza a escala mundial.

La Web se inventó como una herramienta de comunicación destinada a permitir que cualquier persona, en cualquier lugar, pudiera compartir información. Durante muchos años, la Web fue una herramienta de "solo lectura" para muchos. Los blogs y wikis trajeron más autores a la Web, y las redes sociales surgieron del floreciente mercado de contenido y experiencias Web personalizadas. Los estándares **W3C** han respaldado esta evolución gracias a una arquitectura sólida y principios de diseño.

La Web ha transformado la forma en que nos comunicamos entre nosotros. Al hacerlo, también ha modificado la naturaleza de nuestras relaciones sociales. Las personas ahora "se encuentran en la Web" y mantienen relaciones comerciales y personales, en algunos casos sin siquiera conocerse en persona. **W3C** reconoce que la confianza es un fenómeno social, pero el diseño de tecnología puede fomentar la confianza y la seguridad. A medida que haya más actividad en línea, será aún más importante respaldar interacciones complejas entre las partes de todo el mundo.

11.3 TECNOLOGÍAS NECESARIAS

11.3.1 SOA

SOA es la sigla que en inglés representa a “Service Oriented Architecture” (Arquitectura Orientada a los servicios).

Básicamente la arquitectura propuesta es un nuevo paradigma dentro de las ciencias informáticas que permite organizar y utilizar distintas capacidades distribuidas (servicios) ofrecidos por distintos actores (personas y/u organizaciones) a lo largo de Internet.

Esta arquitectura toma más fuerza gracias al surgimiento del concepto de Web Services (servicio Web) que se explicará más adelante.

Hasta ahora, la mayoría de las discusiones sobre SOA giraban en torno a temas sobre la integración de sistemas dispares, el aprovechamiento de las Bases de Datos existentes en una organización o la creación de una arquitectura robusta. Si bien todos estos temas son relevantes para SOA, existen otros temas significativos e

interesantes relacionados con SOA que merecen atención. Para lograr su objetivo de conectar sistemas heterogéneos y autónomos, SOA se adhiere a varios principios de diseño esenciales. Uno de los principios mantiene que la existencia de servicios independientes implica la presencia de partes de código y datos interconectados mediante mensajería.

Es más, los servicios están íntimamente vinculados a la mensajería ya que el único camino de entrada y salida de un servicio es a través de mensajes. No obstante, los servicios seguirán funcionando de forma independiente de los demás.

Por lo tanto, SOA proporciona una metodología y un entorno de trabajo para documentar capacidades de negocio y puede dar soporte a las actividades de integración y consolidación entre organizaciones.

En un ambiente SOA, los nodos de la red hacen disponibles sus recursos a otros participantes en la red como servicios independientes a los que tienen acceso de un modo estandarizado. La mayoría de las definiciones SOA identifican la utilización de Servicios Web (empleando SOAP y WSDL) en su implementación, no obstante, se puede implementar SOA utilizando cualquier tecnología basada en servicios.

Al contrario de las arquitecturas orientadas a objetos, SOAs está formada por servicios de aplicación débilmente acoplados y altamente interoperables. Para comunicarse entre sí, estos servicios se basan en una definición formal independiente de la plataforma subyacente y del lenguaje de programación (por ejemplo, WSDL. Ver Web Services más adelante). La definición de la interfaz encapsula (oculta) las particularidades de una implementación, lo que la hace independiente del fabricante, del lenguaje de programación o de la tecnología de desarrollo (como Java o .NET). Con esta arquitectura, se pretende que los componentes de software desarrollados sean muy reusables, ya que la interfaz se define siguiendo un estándar; así, un servicio C Sharp podría ser usado por una aplicación Java.

Dos ejemplos de soluciones comerciales SOA serían "Java Enterprise System" de Sun Microsystems y "Connected Services Framework" (BizTalk® Server + SQL Server + Windows Server + .NET Framework) de Microsoft.

Los lenguajes de alto nivel como BPEL (ver más adelante) llevan el concepto de servicio un paso adelante al proporcionar métodos de definición y soporte para flujos de trabajo y procesos de negocio.

11.3.2 Web Services

Un servicio Web es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar Web Services para intercambiar mensajes (servicios) en

redes como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los Web Services.

Los Web Services, garantizando el uso de estándares en la tecnología subyacente utilizada, permiten el gran crecimiento del concepto SOA en la actualidad. Ya que los Web Services permiten independizarse de la plataforma y de los lenguajes de programación a ser utilizados.

Los Web Services, no son por lo tanto aplicaciones con una interfaz gráfica con la que las personas puedan interactuar, sino que son un conjunto de especificaciones de software accesible en Internet (o en redes privadas que usen tecnologías Internet) por otras aplicaciones. De esta forma podemos desarrollar aplicaciones que hagan uso de otras aplicaciones que estén disponibles en Internet.

Un típico ejemplo podría ser un Web Service al que se le pudiese preguntar por una empresa y que nos responda en tiempo real el valor que están cotizando las acciones de dicha compañía.

De esta forma cualquier aplicación (ya sea Web o de escritorio) que quiera mostrar esta información sólo tendría que solicitarla a través de Internet al Web Service cuando la necesitase.

Otro ejemplo podría ser uno que, al pasarle el nombre de una ciudad, nos devolviese la temperatura, humedad, y otras condiciones de clima.

11.3.2.1 Estándares empleados

- **Web Services Protocol Stack**: Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- **XML** (eXtensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.
- **XSchema**: XML Schema Definition Language
- **XSL/XSLT/XPath**: Extensible Stylesheet Language
- **JSON**: JavaScript Object Notation
- **SOAP** (Simple Object Access Protocol) o XML-RPC: Protocolos sobre los que se establece el intercambio.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como **HTTP**, **FTP**, o **SMTP**.
- **WSDL** (Web Service Definition Language): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- **UDDI** (Universal Description, Discovery, and Integration): Protocolo para publicar la información de los servicios Web. Permite a las aplicaciones comprobar qué servicios web están disponibles.

- **WS-Security**: Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.
- **RESTFul**: Protocolo de facto alternativa sobre HTTP para la implementación de servicios. Se consideran una opción más liviana para implementar Web Services.

11.3.3 Microservice-SOA

El término "Arquitectura de microservicio" ha surgido en los últimos años para describir una forma particular de diseñar aplicaciones de software como conjuntos de servicios desplegables de forma independiente. Si bien no existe una definición precisa de este estilo arquitectónico, existen ciertas características comunes en torno a la organización en torno a la capacidad empresarial, la implementación automatizada, la inteligencia en los puntos finales y el control descentralizado de idiomas y datos.

En resumen, el estilo arquitectónico de Microservicios es un enfoque para desarrollar una sola aplicación como un conjunto de pequeños servicios, cada uno ejecutándose en su propio proceso y comunicándose con mecanismos ligeros, a menudo una API de recursos HTTP. Estos servicios se basan en las capacidades comerciales y se implementan de forma independiente mediante una maquinaria de implementación totalmente automatizada. Existe un mínimo de administración centralizada de estos servicios, que pueden estar escritos en diferentes lenguajes de programación y utilizar diferentes tecnologías de almacenamiento de datos.

El estilo arquitectónico de “Microservicios” es considerado parte de la arquitectura SOA, con ciertas particularidades que requieren ser respetadas.

11.3.3.1 Características

No podemos decir que existe una definición formal del estilo arquitectónico de microservicios, pero podemos intentar describir lo que vemos como características comunes para las arquitecturas que se ajustan a la etiqueta. Al igual que con cualquier definición que describa características comunes, no todas las arquitecturas de microservicios tienen todas las características, pero esperamos que la mayoría de las arquitecturas de microservicios exhiban la mayoría de las características. Si bien los autores hemos sido miembros activos de esta comunidad bastante flexible, nuestra intención es intentar una descripción de lo que vemos en nuestro propio trabajo y en esfuerzos similares de los equipos que conocemos. En particular, no estamos estableciendo una definición a la que conformarnos.

11.3.3.2 Componentization via Services

Durante el tiempo que hemos estado involucrados en la industria del software, ha habido un deseo de construir sistemas conectando componentes, de manera muy similar a como vemos que las cosas se hacen en el mundo físico. Durante las últimas dos décadas, hemos visto un progreso considerable con grandes compendios de bibliotecas comunes que forman parte de la mayoría de las plataformas de idiomas.

Cuando se habla de componentes, nos encontramos con la difícil definición de lo que hace un componente. Nuestra definición es que un componente es una unidad de software que se puede reemplazar y actualizar de forma independiente.

La arquitectura de microservicios utiliza bibliotecas, pero su forma principal de componente de su propio software es dividiéndolas en servicios. Definimos bibliotecas como componentes que están vinculados a un programa y se llaman mediante llamadas a funciones en memoria, mientras que los servicios son componentes fuera de proceso que se comunican con un mecanismo como una solicitud de servicio web o una llamada a procedimiento remoto. (Este es un concepto diferente al de un objeto de servicio en muchos programas OO).

Una razón principal para utilizar los servicios como componentes (en lugar de bibliotecas) es que los servicios se pueden implementar de forma independiente. Si se tiene una aplicación que consta de varias bibliotecas en un solo proceso, un cambio en un solo componente implica tener que volver a implementar toda la aplicación. Pero si esa aplicación se descompone en varios servicios, se puede esperar que muchos cambios de un solo servicio solo requieran que ese servicio se vuelva a implementar. El objetivo de una buena arquitectura de microservicio es minimizarlos a través de límites de servicio cohesivos y mecanismos de evolución en los contratos de servicio.

El enfoque de microservicios para la división es diferente y se divide en servicios organizados en torno a la capacidad del negocio o problema a resolver. Dichos servicios requieren una implementación de software de pila amplia para esa área de negocio, incluida la interfaz de usuario, el almacenamiento persistente y cualquier colaboración externa. En consecuencia, los equipos son multifuncionales, incluida la gama completa de habilidades necesarias para el desarrollo: experiencia del usuario, base de datos y gestión de proyectos.

11.3.3.3 Puntos finales inteligentes y tuberías tontas (Smart endpoints and dumb pipes)

Al construir estructuras de comunicación entre diferentes procesos, hemos visto muchos productos y enfoques que hacen hincapié en poner mucha inteligencia en el mecanismo de comunicación en sí. Un buen ejemplo de esto es Enterprise

Service Bus (ESB), donde los productos ESB a menudo incluyen instalaciones sofisticadas para enrutamiento de mensajes, coreografía, transformación y aplicación de reglas comerciales.

La comunidad de microservicios favorece un enfoque alternativo: terminales inteligentes y tuberías tontas. Las aplicaciones creadas a partir de microservicios tienen como objetivo ser lo más desacopladas y cohesivas posibles: poseen su propia lógica de dominio y actúan más como filtros en el sentido clásico de Unix: reciben una solicitud, aplican la lógica según corresponda y producen una respuesta. Estos están coreografiados usando protocolos RESTish simples en lugar de protocolos complejos como WS-Choreography o BPEL u orquestados por una herramienta central.

El segundo enfoque de uso común es la mensajería a través de un bus de mensajes ligero. La infraestructura elegida es típicamente tonta (tonta, ya que actúa solo como un enrutador de mensajes): implementaciones simples como RabbitMQ o ZeroMQ no hacen mucho más que proporcionar una estructura asíncrona confiable: la inteligencia aún vive en los puntos finales que están produciendo y consumir mensajes; en los servicios.

11.3.3.4 Gobernanza descentralizada

Una de las consecuencias de la gobernanza centralizada es la tendencia a estandarizar en plataformas tecnológicas únicas. La experiencia demuestra que este enfoque es restrictivo: no todos los problemas son un clavo y no todas las soluciones son un martillo. Preferimos usar la herramienta adecuada para el trabajo y, aunque las aplicaciones monolíticas pueden aprovechar diferentes idiomas hasta cierto punto, no es tan común.

Al dividir los componentes monolíticos en servicios, Se cuenta con una opción al construir cada uno de ellos. Se pueden usar distintas capas de servicios o frameworks de middleware para crear una página de informes simple. Utilizar distintos lenguajes de acuerdo a la mejor adaptación al problema a resolver. Contar con distintos almacenamientos y bases de datos tanto relacionales como las conocidas como NOSQL.

11.4 3 BPM - Orquestación de Servicios

Otro punto importante a tener en cuenta dentro de la mensajería en general y el uso de XML o JSON en particular es la necesidad de construir adaptadores que nos permitan integrar dichas aplicaciones. Sin estos adaptadores no podríamos comunicar una aplicación que habla un idioma X con otra que habla un idioma Z, para ello son fundamentales lo que en la arquitectura se denominan parsers (decodificadores).

Conjuntamente con los parsers se debe contar con un proceso que nos permita coordinar esta interacción entre partes. Ya que muchas veces cuando hablamos de Web services no nos limitamos solamente al uso de un servicio sencillo sino a la ejecución distribuida de aplicaciones complejas. Como solución y componente fundamental veremos que los lenguajes de Proceso o como se denominan normalmente BPM (Business Process Modeling) permiten estandarizar y orquestar una relación compleja entre servicios y procesos.

Dentro de una arquitectura SOA usando Web services correctamente orquestados se debe pensar seriamente en intermediarios que permitan la correcta distribución de mensajes y aplicación de procesos y reglas de negocio. También a estos intermediarios se los suele denominar HUB o Switch de integración de aplicaciones. Un ejemplo de ello es lo implementado por el HIPPA (Health Insurance Portability & Accountability Act) que ha regulado la implementación de lo que ha denominado como “Clearing Houses” que tienen la responsabilidad de administrar mensajes, solicitudes y reclamaciones entre asegurados, aseguradoras y hospitales y clínicas dentro de USA. Esta situación es extensible a la gran cantidad de estándares de negocio propuestos y consensuados en OASIS.

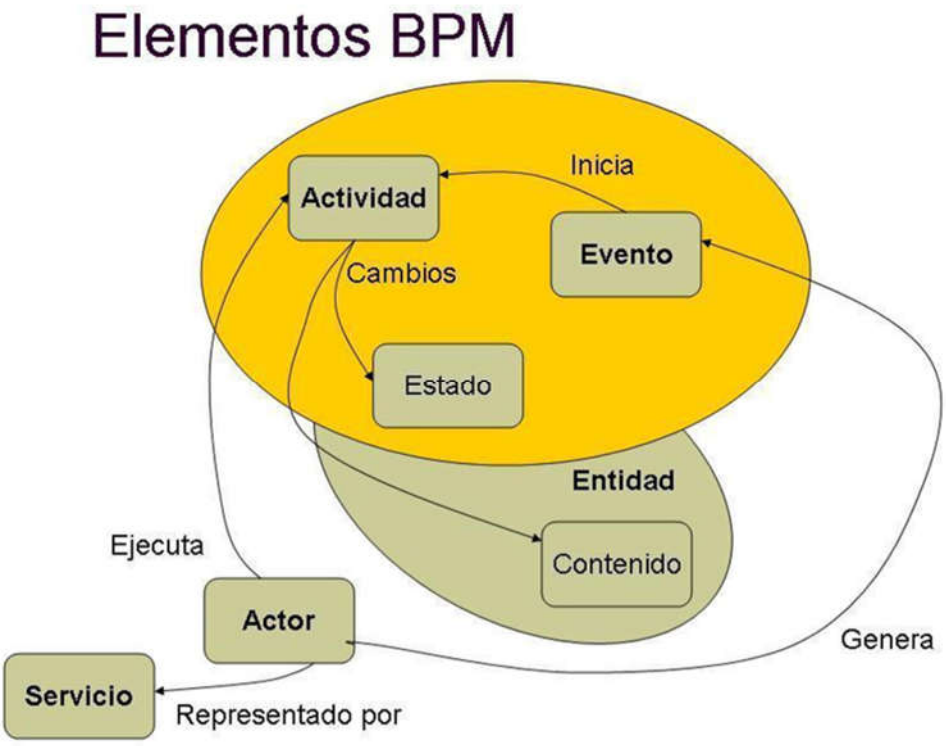

Figura 11.3

Para tener una mayor comprensión del significado de la mencionada coordinación, veremos distintas topologías de coordinación (orquestación) para transacciones y mensajes entre servicios:

11.4.3.1 Relación Binaria: El contexto y la actividad están implícitas en cada servicio, logrando de esta forma un criterio de auto-coordinación.

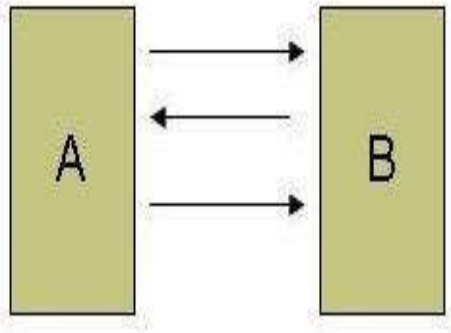

Figura 11.4

11.3.2 Hub o Spoke: Relación entre servicios coordinados por un concentrador de transacciones que tiene la responsabilidad del intercambio como así las actividades de control, validación, flujo de proceso y seguridad.

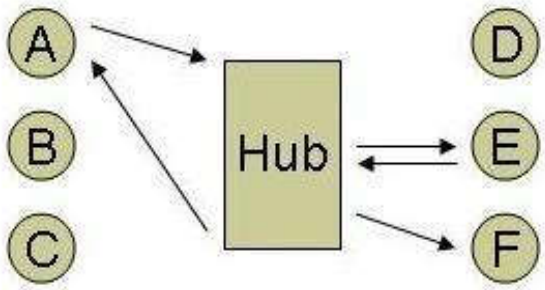

Figura 11.5

11.3.3 Punto a punto en Multigrupo: Existe un coordinador que auspicia de árbitro entre las partes, pero tanto el contexto como las actividades son explícitas pero dependientes de cada servicio. Este esquema es una solución mixta entre el esquema concentrador y el distribuido punto a punto.

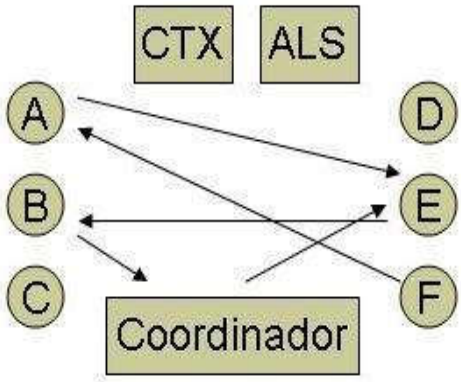

Figura 11.6

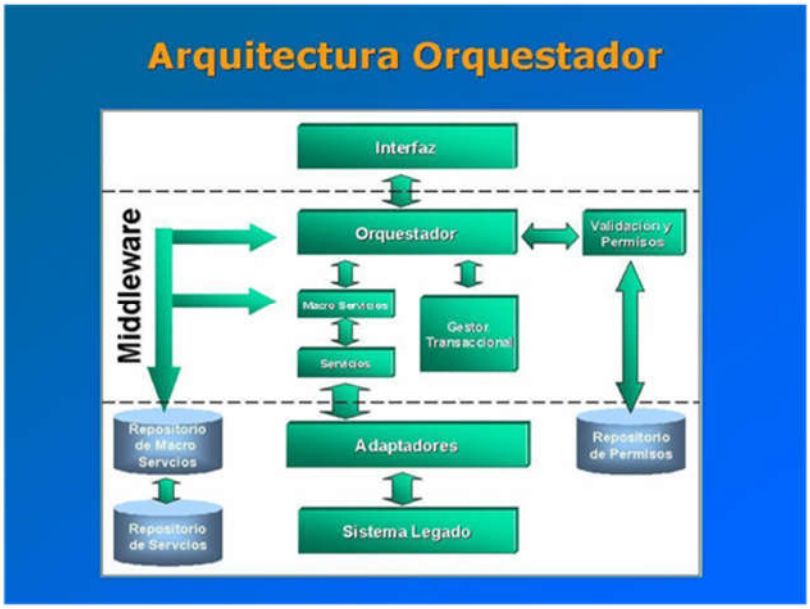

Figura 11.7

Actualmente se encuentran disponibles varios lenguajes estándar de orquestación y especificación de procesos entre los que podemos encontrar:

- XLANG - XML Business Process Language.
- BPML - Business Process Modeling Language.
- BPSS – Business Process Specification Schema.
- WSFL - Web Services Flow Language.
- WSCL – Web Services Conversation Language.
- WSCI - Web Service Choreography Interface.
- BTP - Business Transaction Protocol.
- WS-BPEL – Web Services Business Process Execution Language.
- WS-C, WS-T, WS-AT, WS-BA (Coordination, Transaction).
- BPMN - Business Process Modeling Notation.