

KOSTYANTYN MYROSHNYCHENKO

**MATURIDADE DE PLATAFORMAS SOA OPEN
SOURCE VS PROPRIETÁRIOS**

Orientador: Professor Rui Ribeiro

Universidade Lusófona de Humanidades e Tecnologias

Departamento de Engenharia Informática e Sistemas de Informação

**Lisboa
2013**

KOSTYANTYN MYROSHNYCHENKO

**MATURIDADE DE PLATAFORMAS SOA OPEN
SOURCE VS PROPRIETÁRIOS**

Dissertação apresentada para a obtenção do Grau de Mestre em Gestão de Sistemas e Tecnologias de Informação, no curso de Mestrado em GSTI, conferido pela Universidade Lusófona de Humanidades e Tecnologias.

Orientador: Professor Rui Ribeiro

Universidade Lusófona de Humanidades e Tecnologias

Departamento de Engenharia Informática e Sistemas de Informação

**Lisboa
2013**

Agradecimentos:

Quero agradecer a minha família pelas oportunidades que me deram e pelos inúmeros sacrifícios que cometeram para garantir o meu conforto, segurança e educação ao longo da minha vida. O apoio constante da minha mãe, meu pai e as minhas avós, contribuiu de forma incalculável para a elaboração deste trabalho, que lhes é dedicado.

Também queria agradecer o meu orientador, o professor Rui Ribeiro, que sugeriu este tema e acompanhou o meu progresso ao longo da elaboração do trabalho. Os seus conselhos e a sua orientação permitiram a este trabalho de tomar a direcção correcta e ultrapassar os diversos obstáculos encontrados durante a pesquisa, estruturação e redacção.

Finalmente, agradeço aos meus amigos Viktoriya, Renato e Ricardo, que me deram imenso apoio moral e incentivo para finalizar este trabalho.

Abstract

O bom funcionamento de uma empresa passa pela coordenação dos seus vários elementos, pela fluidez das suas operações diárias, pelo desempenho dos seus recursos, tanto humanos como materiais, e da interacção dos vários sistemas que a compõem. As tecnologias empresariais sentiram um desenvolvimento contínuo após a sua aparição, desde o processo básico, para gestão de processos de negócios (BPM), para plataformas de recursos empresariais (ERP) modernos como o sistema proprietário SAP ou Oracle, para conceitos mais gerais como SOA e cloud, baseados em standards abertos. As novas tecnologias apresentam novos canais de trânsito de informação mais rápidos e eficientes, formas de automatizar e acompanhar processos de negócio e vários tipos de infra-estruturas que podem ser utilizadas de forma a tornar a empresa mais produtiva e flexível. As soluções comerciais existentes permitem realizar estes objectivos mas os seus custos de aquisição podem revelar-se demasiado elevados para algumas empresas ou organizações, que arriscam de não se adaptar às mudanças do negócio. Ao mesmo tempo, software livre está a ganhar popularidade mas existem sempre alguns preconceitos sobre a qualidade e maturidade deste tipo de software.

O objectivo deste trabalho é apresentar SOA, os principais produtos SOA comerciais e open source e realizar uma comparação entre as duas categorias para verificar o nível de maturidade do SOA open source em relação às soluções SOA proprietárias.

A company's performance depends on the coordination of its different elements, the fluidity of its daily operations, the performance of its resources, both human and material, and on the interaction of the different systems that comprise the company. Enterprise technology has experienced a constant development effort, since the first basic process gave rise to BPM technologies, through modern proprietary ERP technologies (such as Oracle or SAP), then evolving into more general concepts such as SOA and the cloud, based on open standards. New technologies offer faster and more efficient communications channels, ways to automate and monitor different sorts of processes, and infrastructures that enable more productivity and flexibility. Existing commercial solutions allow for the realization of these goals but their costs can be prohibitive for some companies, which in this case risk failing to adapt to the changes in business needs. At the same time, open source software is gaining popularity, but there are yet some fears about it's quality and maturity.

The object of this paper is to present what the concept of SOA is, introduce some of the main commercial SOA offerings and make a comparison between both categories to determine the maturity of open source offerings relative to their commercial alternatives.

Key words: enterprise integration, SOA, open source, web services

Abreviaturas

ABAP: Advanced Business Application Programming

API: Application Programming Interface

BAM: Business Activity Monitoring

BPM: Business Process Management

CEP: Complex Event Processing

EAI: Enterprise Application Integration

EJB: Enterprise Java Beans

ESB: Enterprise Service Bus

Java EE: Java Enterprise Edition

JB: Java Business Integration

JSP: Java Server Pages

JSF: Java Server Faces

JMS: Java Message Service

KPI: Key Performance Indicator

MVC: Model View Controller

POJO: Plain Old Java Object

SOA: Service-Oriented Architecture

SOAP: Simple Object Access Protocol

WSDL: Web Service Description Language

XML: eXtensible Markup Language

ÍNDICE DE CONTEÚDOS

AGRADECIMENTOS:	1
ABSTRACT	2
ABREVIATURAS	4
INTRODUÇÃO	9
1. REVISÃO BIBLIOGRÁFICA	10
1.1 Origens do SOA	10
1.2 Benefícios conceptuais do SOA	11
1.3 SOA inicial e moderno	13
1.4 Organização SOA	14
1.5 Sobre Open Source Software – visão resumida	15
1.6 SOA Proprietário e SOA Open Source	16
2. FUNDAMENTOS SOA	18
2.1 Tecnologias-chave em SOA	18
2.2 Termos	20
2.3 Conceito de Serviços	21
2.4 Conceito de Processos de negócio	24
2.5 Stack empresarial	25
2.5.1 Camada de sistemas operacionais	26
2.5.2 Camada de componentes de serviços	27
2.5.3 Camada de Serviços	27
2.5.4 Camada de Processos de Negócios	27
2.5.5 Camada de Consumidor	28
2.5.6 Camada de Integração	28
2.5.7 Camada de Qualidade de Serviço	28
2.5.8 Camada de Arquitectura de Informação e Business Intelligence	29
2.5.9 Camada de Governance	29
2.6 Componentes da stack empresarial	29
2.6.1 Servidor de Aplicação	30
2.6.2 ESB	31
2.6.3 Registo e Repositório de Serviços, Governância	32
2.6.4 CEP, BAM	33
2.6.5 Enterprise Decision Management, Business Rules	34
2.6.6 Ferramentas de desenvolvimento e modelação	35
3. APRESENTAÇÃO DO SOFTWARE SOA	36
3.1 Suites SOA Comerciais	36
3.1.1 IBM WebSphere	36
3.1.2 Oracle Weblogic	37

3.1.3 Tibco ActiveMatrix Service Grid.....	39
3.1.4 Microsoft Biztalk.....	40
3.1.5 SAP Netweaver.....	41
3.2 Suites SOA open source.....	43
3.2.1 WSO2.....	43
3.2.2 JBoss.....	44
3.2.3 Apache ServiceMix.....	45
3.2.4 Mule.....	46
3.2.5 Petals.....	48
4. COMPARAÇÃO DE COMPLETUDE DA STACK E STANDARDS	49
4.1 Suites comerciais.....	49
4.2 Open source.....	51
4.3 Análise.....	54
5. CONCLUSÕES.....	60
BIBLIOGRAFIA.....	61
APÊNDICE.....	65

Índice de figuras

Fig 1. Serviços e processos na organização (Davis 2009).....	14
Fig. 2.1. SOAP, WSDL e UDDI (Erl 2005).....	19
Fig. 2.2. Estrutura do documento XML com SOAP	22
Fig 2.3. Ciclo de vida de um serviço (IBM).....	23
Fig. 2.4. Processos e serviços (Erl 2009).....	24
Fig 2.5. Modelo abstracto de stack SOA (IBM).....	26
Fig. 2.6. Stack SOA típica (Davis 2009).....	30
Fig.2.7. ESB Fiorano (Fiorano 2007).....	31
Fig. 2.8. CEP simplificado da Tibco (Tibco 2009).....	34
Fig. 3.1. SOA IBM (IBM).....	36
Fig 3.2. SOA Oracle.....	38
Fig 3.3. Tibco ActiveMatrix.....	39
Fig 3.4. Exemplo de SOA com Biztalk (Microsoft).....	40
Fig. 3.5. SOA com Netweaver.....	41
Fig. 3.6. WSO2 Carbon.....	43
Fig. 3.7. JBoss SOA Platform	44
Fig. 3.8. Apache ServiceMix.....	45
Fig. 3.9. Mule.....	47
Fig. 3.10. Petals	48
Fig. 4. Quadrante de comparação	56

Índice de tabelas

Componentes IBM.....	37
Componentes Oracle.....	38
Componentes Tibco.....	40
Componentes Microsoft.....	41
Componentes SAP.....	42
Componentes WSO2.....	44
Componentes JBoss.....	45
Componentes ServiceMix.....	46
Componentes Mule.....	47
Componentes Petals.....	48
Standards Comerciais.....	49
Standards Open Source.....	51
Classificação de Completude da Stack.....	55
Comparação de Suporte a Standards.....	55
Avaliação Final.....	56
Tabela de Standards em Apendice.....	65

Introdução

Este trabalho vai estudar um conceito de tecnologia de suporte empresarial para conhecer as suas possibilidades e os benefícios que pode trazer à organização. Também será apresentada a situação da tecnologia actual e as funcionalidades dos produtos. Finalmente será contemplado o panorama open-source e as suas vantagens, e verificar a sua viabilidade para competir com a oferta comercial.

Este trabalho é realizado para clarificar o posicionamento SOA como conceito arquitectural e organizacional, as tecnologias disponíveis para realizar SOA na organização, e estudar a possibilidade de utilizar software open source para atingir o mesmo objectivo.

Para realizar o estudo foi feita uma consulta de livros da autoria do reconhecido fundador do conceito Thomas Erl, tal como livros sobre a criação de SOA por profissionais na área (James Bean, Jeff Davis). Para a escolha dos produtos foram consultados estudos realizados pelos institutos de análise Forrester, Gartner, Wintergreen e Bloor. Também foi consultada a documentação técnica dos diferentes fornecedores de tecnologias e organizações de standards, estudos de caso comparativos entre fornecedores comerciais, fóruns das comunidades dos vários produtos e blogs de profissionais na área de tecnologia empresarial. Foram recolhidos dados sobre os aspectos mais relevantes para uma introdução da tecnologia e de como ela complementa as outras tecnologias mais utilizadas, dados sobre as actualizações e diferenciações dos vários produtos e das potencialidades, e agregados numa análise qualitativa.

O presente tema foi escolhido para complementar o conhecimento sobre as tecnologias empresariais e a sua evolução, e examinar a viabilidade de implementação realística de SOA na empresa, através de software open source.

No primeiro capítulo serão introduzidos o conceito da arquitectura orientada a serviços, os seus benefícios e implicações para a empresa. O segundo capítulo descreve as tecnologias complementares ao SOA, a descrição técnica de serviços e processos, e os componentes convencionais de suites SOA. No terceiro capítulo serão apresentadas algumas suites open source e proprietárias actualmente usadas. No quarto capítulo estão agregados os dados sobre as funcionalidades das suites descritas no capítulo anterior, e é realizada uma comparação das funcionalidades e aderência aos standards. O quinto capítulo reúne as conclusões e as considerações para futuros estudos nesta área.

1. Revisão Bibliográfica

1.1. Origens do SOA

Na década de 90, a computação distribuída, a Internet, e E-commerce passaram a ser mais disponíveis para o ambiente empresarial. As tecnologias diminuíram de escala e passaram a representar um menor investimento com plataformas distribuídas e soluções de infraestruturas partilhadas. O hardware viu a sua performance aumentar enquanto os custos estabilizaram, ou em certos casos baixaram. Este ambiente facilitou o desenvolvimento e implementação de aplicações ponto a ponto, uma empresa podia rapidamente participar em e-commerce e ter maior presença na Web.

As empresas sentiram o efeito das pressões económicas, competitivas e de mercado: através de maior disponibilidade tecnológica, os produtos e serviços podiam ser introduzidos mais facilmente. Pequenas startups e empreendedores inovadores podiam entrar no mercado por um custo modesto, e não tinham de gerir grandes organizações ou processos. Para manter ou melhorar a posição de mercado, uma empresa tinha de ser mais dinâmica e adaptável, por esta razão as organizações de tecnologias tornaram-se parceiros iguais das organizações de negócios.

Para lidar com as pressões, era necessário redefinir o foco: foi dada atenção à gestão do desenvolvimento, infraestrutura e custos operacionais. As tecnologias responderam com um maior foco de excelência tecnológica, desenvolvimento rápido de aplicações, a introdução de consolidação e virtualização para a gestão de custos, e integração para resolver inconsistências entre as aplicações ponto a ponto. No entanto, este ambiente contribuiu à proliferação de aplicações autónomas, aumento de custos operacionais e de suporte, e um aumento do grau de dificuldade na conexão dos vários sistemas e partilha de informação. Uma das óbvias necessidades de integração manifestou-se na noção “360 graus” do cliente, em que a empresa foca na sua relação total com o cliente (actual e potencial) em vez de apenas na encomenda. O número de aplicações cliente e maneiras de interagir com o negócio levaram à informação inconsistente, e para resolver este problema as tecnologias de integração EAI e EII viram a sua popularidade aumentar. (Bean, 2009)

Tecnologias como IBM CICS, CORBA, e Microsoft DCOM foram das primeiras utilizadas para realizar a integração de tecnologias em rede. Estas tecnologias são precursores dos serviços Web e partilham com estes a possibilidade de descoberta de serviços de conexões dinâmicas, mas num ambiente tipicamente interno. Antes de SOA existiam também linguagens de modelação de processos de negócios, por exemplo PML, e motores de execução de processos, como a ProcessWise da ICL.

A principal diferença entre estas tecnologias e o movimento SOA com base em serviços Web é de serem completamente proprietários, enquanto SOA é baseado em standards e interfaces abertos.

SOA apareceu como uma combinação de arquitectura e facilitador tecnológico para a empresa. Do ponto de vista conceptual, SOA herda vários princípios dos conceitos já existentes de orientação à objectos, computação distribuída, gestão de processos de negócios e envolve uma mudança de paradigma, de invocação de métodos de objectos para a troca de informação entre serviços. A Arquitectura Orientada a Serviços é mais do que uma plataforma ou tecnologia, é um conceito organizacional para a empresa e as suas interações com os seus clientes, fornecedores e parceiros. SOA procura agilizar o negócio da empresa, aproveitar ao máximo os recursos tecnológicos e as suas funcionalidades, facilitar a integração dos diferentes recursos e a comunicação entre os componentes da empresa.

SOA ganhou ímpeto com a emergência dos serviços Web, padrão inicialmente desenvolvido com apoio da Microsoft, que foi levado a um maior público em 2000. Grandes empresas como a Oracle, SAP, IBM, HP e Sun rapidamente juntaram-se ao movimento por causa do interesse aumentado na integração pelas empresas dos seus negócios com outros sistemas, departamentos e empresas. (Davis, 2009)

1.2. Benefícios conceptuais do SOA

As infraestruturas em maior estado de fragmentação e complexidade estão a limitar a capacidade das tecnologias de informação à reagir às necessidades de negócios. Muitas organizações possuem um conjunto de aplicações desconectadas pré-fabricadas que na sua maioria não eram previstas para interoperabilidade, integração e reutilização da informação.

Tradicionalmente, os sistemas de negócios eram desenhados com uma orientação funcional, resultando em silos de serviços de informação. O problema fundamental é que os processos de negócios, que neste caso devem abranger o silo, e não são adaptáveis às

mudanças nas necessidades de negócios, pois estão embebidos no sistema. Os EAI e outros middlewares permitem aos sistemas de comunicar entre si, mas não resolvem o problema por completo. Estas soluções têm uma capacidade limitada para combinar processos entre as aplicações, permitem pouca adaptabilidade aos processos e tem grande custo.

Adicionalmente, estas soluções utilizam tecnologias proprietárias, requerendo competências específicas da parte dos programadores e vinculação aos produtos daquela empresa. Os sistemas estão fortemente acoplados, o que implica que se um interface é mudado, todos os sistemas têm de ser ajustados. Estes factores dificultam a adaptação e um aumento total do orçamento dedicado às tecnologias de informação.

SOA procura responder a estes obstáculos e obter os seguintes benefícios:

- Maior interoperabilidade. SOA promove a utilização de standards da industria, permitindo às aplicações em silo existentes interagir mais facilmente do que com soluções EAI.
- Reutilização. As funcionalidades das aplicações existentes na empresa podem ser encapsuladas e estes serviços reutilizados no desenvolvimento e expostos para consumo externo. Os processos de negócios podem ser construídos sob forma de orquestração de serviços, aumentando as oportunidades de reutilização.
- Maior agilidade nos processos. Com SOA existe menor separação entre o modelo e a implementação do processo: desta maneira é possível alterar processos que já estão implementados.
- Melhor visibilidade. SOA permite ter uma melhor visão do funcionamento dos processos através da exposição das funcionalidades de negócios como serviços e da integração de processos automatizados com BPM nos portais empresariais para ajudar no suporte à decisão.
- Menores custos de manutenção. SOA promove a eliminação de serviços redundantes e a consolidação das funcionalidades de negócio existentes num pequeno número de serviços partilhados. A planificação SOA também possibilita a remoção de sistemas e aplicações antigos minimizando o impacto no resto do sistema.
- Agnóstico à tecnologia: sistemas SOA podem ser implementados independentemente de plataformas ou tecnologias específicas (tais como Java ou .NET). Serviços em C# que correm em .NET e serviços Java em Java EE ambos podem ser consumidos por

uma única aplicação ou cliente. Sistemas antigos podem ser encapsulados e expostos como serviços, permitindo o seu consumo independentemente da linguagem ou plataforma original.

Finalmente, é preciso se lembrar que SOA não é uma tecnologia ou uma plataforma, mas sim um conjunto de práticas, standards e conceitos reunidos para resolver os problemas existentes e proporcionar as vantagens descritas. SOA é agnóstico à implementação, ou seja, não necessita de uma tecnologia específica para ser implementado. SOA segue uma série de standards e tem organizações que apoiam o movimento, como a W3C ou a OASIS, que mantêm e desenvolvem standards, especificações e extensões para SOA. (Erl, 2005)

1.3. SOA inicial e moderno

Pode-se distinguir duas fases na evolução do conceito SOA, a fase inicial onde as diferentes organizações tomavam conhecimento dos princípios gerais, e uma segunda fase que viu o movimento crescer, evoluir e aumentar o seu leque de funcionalidades.

Uma das mais frequentes definições de SOA na fase inicial requeria a possibilidade de:

- particionar a lógica de automatização de negócios em unidades separadas para representar serviços
- ter independência entre as unidades para poder compo-las de várias formas
- ter comunicação entre as unidades de lógica de forma a preservar a sua independência

Estas características fundamentais de encapsulamento, acoplamento frouxo (loose coupling), e comunicação por mensagens, realizadas através dos princípios da orientação a serviços e do conjunto tecnológico de serviços Web, permitem a implementação de um SOA “primitivo”.

As tendências e desenvolvimentos influenciaram o movimento SOA. As grandes empresas e grupos estão constantemente a desenvolver novas especificações de serviços Web e implementar melhor suporte a XML e serviços Web nas plataformas tecnológicas contemporâneas. SOA contemporâneo é construído sobre os princípios do SOA primitivo, aplicando os avanços tecnológicos e da indústria. As características, incluindo as do SOA

primitivo, encontradas no SOA moderno são: a qualidade de serviço, autonomia, baseado em standards abertos, suporta a diversidade de fornecedores, promove a descoberta, federação, composabilidade e agilidade organizacional.(Erl, 2005).

1.4. Organização SOA

O intuito da arquitectura é de combinar as possibilidades tecnológicas, estandardizar os processos de criação e gestão de processos de negócio, partilhar os recursos, ligar vários sistemas já existentes entre eles de forma a produzir um ambiente de trabalho produtivo e simples o suficiente para os consumidores. SOA aposta principalmente (mas não exclusivamente) nos Serviços Web, uma forma de as várias componentes comunicarem entre elas. Estes serviços permitem implementar a lógica e os processos de negócio, e ter uma comunicação contínua através da totalidade do sistema. Uma analogia pode ser feita com peças LEGO: as funcionalidades individuais são extraídas, encapsuladas e disponibilizadas como recursos para construir novas funcionalidades.(Bloomberg, 2006)

SOA também ajuda a preservar os investimentos TI em via de extinção (encapsulando-os / service-enabling) e a utilização de serviços provenientes do exterior. SOA permite também exportar os seus serviços para fora do sistema, tornando assim outras entidades consumidores dos seus serviços. SOA contemporâneo suporta a diversidade de fornecedores. A variedade de soluções middleware e arquitecturas disponíveis permite encontrar combinações mais adequadas ao tamanho e complexidade da empresa sem se dedicar a um único fornecedor, e o advento do software open source expande ainda mais as possibilidades. (Davis, 2009)

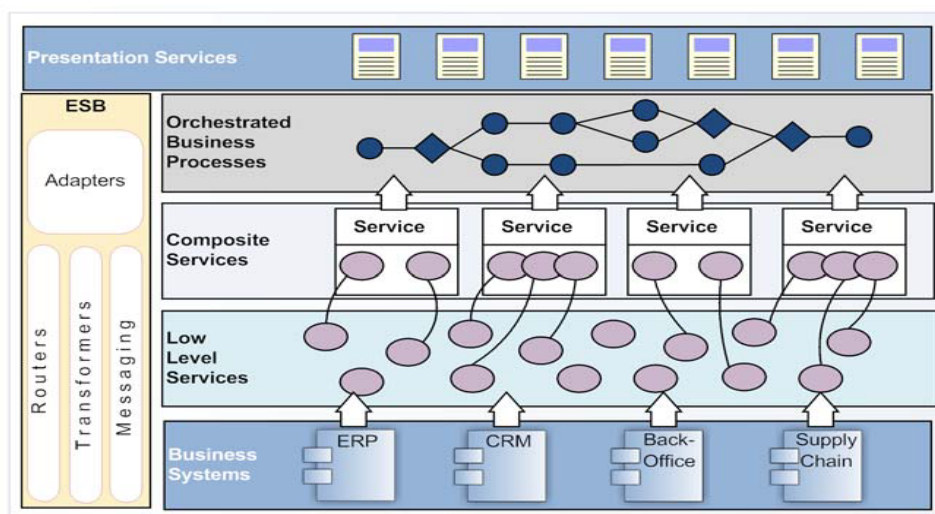


Fig 1. Serviços e processos na organização (Davis 2009)

A figura 1.1 ilustra a organização dos serviços num SOA e a sua relação com as outras aplicações e os processos de negócios. Os serviços de baixo nível são compostos e orquestrados em processos de negócios utilizando um ESB para realizar a comunicação entre os diferentes sistemas.

1.5. Sobre Open Source Software – visão resumida

Open source é a designação geralmente dada ao software que é distribuído livremente e com acesso ao código fonte. Na realidade, existe uma definição oficial elaborada pela Open Software Initiative (OSI) contendo uma série de condições para a distribuição de software open source de modo a preservar os objectivos e princípios por detrás do movimento open source, tais como: promover a facilidade de evolução do software, permitir modificações e melhoramentos, manter a integridade do código original, etc.

OSS (Open Source Software) nasceu com Richard Stallman que criou em 1989 a primeira General Public Licence para o seu projecto GNU, que era uma colecção de licenças para aplicações individuais do GNU que eram incompatíveis entre elas. Stallman queria elaborar uma única licença que seria usada para qualquer projecto, o que permitiria a vários projectos de partilhar o código. Hoje em dia, aplicações open source tem boa presença no mercado pela vantagens competitivas que apresenta. (“Free software movement - Wikipedia, the free encyclopedia”)

Os proponentes do open source apresentam diversas vantagens que este oferece face aos produtos comerciais. Estas vantagens não se limitam ao SOA:

- produto sem custos de aquisição, os custos vem dos potenciais contratos de suporte à implementação, integração e formação do utilizador. Uma confusão frequente é sobre a palavra “free”: “gratis” vs “livre”. No caso do OSS, o significado é “livre”, ou seja, livre distribuição.
- longevidade do produto e vendor lock-in (vinculação ao fornecedor): as empresas comerciais podem se dissolver, mergir com outras ou serem adquiridas, o seu produto pode ser descontinuado ou integrado em outra solução alterando as suas funcionalidades, em contraste o desenvolvimento de um produto OSS é feito pela comunidade e é continuado mesmo se o autor original deixar o projecto. Usar

software open source permite evitar a vinculação a um único fornecedor e ter mais flexibilidade na escolha.

- suporte pela comunidade: a comunidade à volta dum projecto OSS popular é activa e participa no desenvolvimento. Uma pergunta posta nos fóruns do projecto evoca rapidamente uma resposta ou um comentário, os potenciais problemas com a aplicação podem ser resolvidos rapidamente e uma nova patch pode ser emitida no próprio dia, enquanto o suporte oficial das empresas comerciais pode deixar a resolução de bugs para uma próxima versão, que as vezes demora meses a ser disponível. Uma outra vantagem é de não haver prioritização dos clientes pelo tamanho dos seus contratos, todos os utilizadores são igualmente importantes.
- resposta aos pedidos: o desenvolvimento de um projecto OSS é na maioria das vezes completamente transparente, as alterações ao código são documentadas e os desenvolvedores estão atentos aos pedidos dos seus utilizadores, o que leva à potencial e rápida inclusão de funcionalidades desejadas.
- acesso ao código fonte: as aplicações existentes podem ser modificadas, personalizadas e reconfiguradas conforme às necessidades. A natureza aberta do modelo permite uma rápida propagação dos melhoramentos e best practices. Standards abertos permitem estabelecer padrões de conformidade para permitir a comunicação e a segurança entre as várias tecnologias.

(Williams, Clegg, & Dulaney, 2005), (Wilson, 2007)

1.6. SOA Proprietário e SOA Open Source

A questão da utilização do open source para SOA foi bastante discutida durante toda a existência do conceito SOA.

As diferenças entre os produtos SOA proprietários e OSS manifestam-se de várias formas e cada opção apresenta vantagens e críticas. Plataformas proprietárias tem a vantagem de terem equipas dedicadas ao suporte disponibilizadas pelo fornecedor, a presença no mercado assegura a interoperabilidade com os parceiros de negócio, e o utilizador terá a garantia da maturidade do produto. No entanto, estas vantagens são acompanhadas por um custo de entrada mais elevado quando comparado ao open source.

Open source é considerado um “seguidor”, ou seja que estes imitam um produto

proprietário já existente. Este entanto vários produtos open source são inovadores e criadores de standards. Uma das razões do baixo nível de desenvolvimento quando comparado a um produto proprietário é a falta de recursos, os projectos open source são frequentemente mantidos por pequenas equipas que têm responsabilidades à tempo inteiro em outros projectos.

As empresas comerciais têm suites completas de produtos que cobrem todas as funcionalidades dos SOA moderno, que já estão preparados para integrar outras soluções da mesma empresa e soluções dos outros fornecedores principais ou seus parceiros. Os projectos open-source são mais fragmentados e com menor âmbito, embora haja alguns como a JBoss e a WSO2 Carbon que também abragem todas as funcionalidades, incluindo BPM.

No entanto os projectos open source apresentam maior flexibilidade, permitem uma implementação mais rápida para colmatar um defeito no sistema ou efectuar uma funcionalidade específica e geralmente a integração é mais facilmente realizada em projectos de pouca complexidade. A vasta maioria de software SOA open source tem opções de suporte à instalação, manutenção e formação por empresas comerciais, e várias empresas optam por incluir open source no seu SOA, seja para implementar um SOA completo open source de raiz, substituir um ou vários componentes proprietários ou integrar os seus vários ERPs para expor o negócio ao exterior. (Schmelzer, 2010)

2. Fundamentos SOA

2.1. Tecnologias-chave em SOA

Extensible Markup Language (XML) foi criada pela organização W3C. Tal como a HTML, a linguagem XML é derivada da Standard Generalized Markup Language do final dos anos 60, através da qual se pode definir linguagens de marcação para documentos. XML tem sido usada numa variedade de aplicações, incluindo as notáveis XHTML, RSS, XML-RPC e SOAP.

XML ganhou popularidade na era do movimento eBusiness no final da década de 90, com o aumento da viabilidade da Internet como plataforma de negócios através de linguagens de scripting do lado do servidor. Através de XML, os desenvolvedores podiam juntar contexto e sentido a qualquer informação enviada através de protocolos Internet. XML, além de permitir a representação de dados de uma maneira estandardizada, também foi um fundamento para o desenvolvimento de especificações adicionais: a linguagem XML Schema Definition Language (XSD) e o Extensible Stylesheet Language Transformations (XSLT), que fazem parte do conjunto tecnológico XML.

A arquitectura de representação de dados XML estabelece o formato e a estrutura das mensagens que são enviadas entre os serviços, os esquemas XSD preservam a integridade e validade dos dados da mensagem, e XSLT é utilizado para converter as diferentes representações dos dados através o mapeamento de esquemas.

A especificação Simple Object Access Protocol (SOAP) foi submetida à W3C em 2000, e foi desenhada para unificar ou substituir as formas de comunicações RPC proprietárias. O objectivo era de serializar os dados à transmitir em XML antes do transporte e deserializá-lo para o seu formato original no destino.

O potencial do framework aberto de comunicações pela Internet para eBusiness foi rapidamente reconhecido pelas empresas e fabricantes de software, o que levou à criação de uma tecnologia baseada na Web para acabar com a disparidade existente do software e facilitar as comunicações entre empresas e internamente: os serviços Web.

A especificação para a Web Services Description Language (WSDL) foi submetida à W3C em 2001. Esta linguagem é usada para especificar o interface público de um serviço Web, a informação que lhe atribui a sua identidade e possibilita a sua invocação, um WSDL é um documento XML que especifica a localização de um serviço e as operações que este

expõe.

A primeira geração de standards de serviços Web é completada pela especificação UDDI (Universal Description, Discovery and Integration). Originalmente desenvolvida pela UDDI.org, esta especificação foi submetida à OASIS, que continuou o desenvolvimento em colaboração com a organização original. UDDI permite a criação de registos de descrição de serviços dentro e fora da organização, centralizar os registos de serviços numa única localização para os consumidores de serviços. UDDI não é universalmente adoptado em todos os projectos SOA, embora usufrua da participação da Microsoft, Oracle, Sun, IBM e mais de 220 outras empresas na iniciativa.

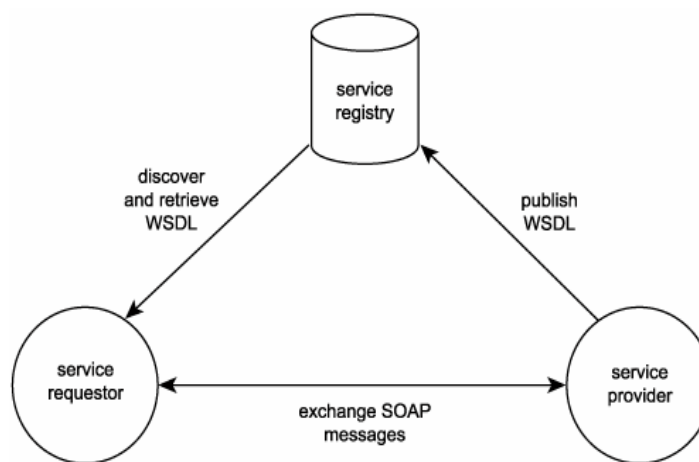


Fig. 2.1. SOAP, WSDL e UDDI (Erl 2005)

A figura 2.2. representa as interações de fornecedores (provider) e requisitores (requestor) de serviços: WSDL descreve os serviços, SOAP estabelece um formato universal de mensagem para transporte de serviços e UDDI estabelece um formato de registo de serviços. O fornecedor de serviços publica os seus serviços com uma descrição WSDL, o requisitor descobre o WSDL no registo UDDI e requisita o serviço ao fornecedor.

SOAP pode ser utilizado em conjunção com qualquer protocolo de transporte (HTTP, SMTP, TCP ou JMS), e tipicamente utiliza-se SOAP/JMS dentro da empresa, e SOAP/HTTP para fora. (Davis, 2009)

Em alternativa ao SOAP com WSDL, que pode ser demasiado “pesado” para algumas implementações, SOA pode ser implementado através de REST (Representational State Transfer) sobre HTTP. Desenvolvido em 2000 por Roy Fielding, um dos criadores de HTTP, REST é um estilo arquitectural que considera páginas Web como recursos, e apresenta uma forma universal e leve de expor recursos por HTTP e XML. No entanto, REST torna difícil integrar algumas características de SOA de segunda geração (“SOA moderno”), em contraste ao SOAP, que suporta políticas WS-*. As plataformas de Amazon, Yahoo e Google, entre

outros, são baseadas em REST (chamadas RESTful), pelas vantagens de escalabilidade e menor peso, como também os feeds RSS. O consenso geral é que um sistema SOA empresarial que executa serviços críticos ao negócio (por ex. transacções financeiras) será mais bem servido por SOAP, enquanto aplicações de publicação de informação e projectos de pequena escala onde não há penalidades na repetição de pedidos aproveitarão melhor REST. Os dois modelos não são mutualmente exclusivos, é possível utilizar os dois modelos no mesmo sistema para aproveitar dos benefícios de cada um no ambiente apropriado.

2.2. Termos

“Standards”, “Especificações”, “Extensões”

Estes termos são frequentemente encontrados quando se fala de SOA, e as vezes podem ser utilizados para definir o mesmo conceito. Normalmente, uma especificação é um documento que propõe um standard, que por sua vez se torna um standard da indústria depois da especificação ser submetida, aceite e publicada por uma organização reconhecida de standards. Uma “extensão” é um termo que designa uma especificação WS-* ou uma das suas funcionalidades.

Seguem algumas organizações mais importantes que contribuem e mantêm standards para SOA:

– **World Wide Web Consortium (W3C)**

A W3C foi originalmente fundada em 1994 e foi importante na propagação da Web como meio global de partilha de informação. O projecto inicial foi desenvolvimento do HTML, uma das linguagens mais populares da indústria TI. A W3C também desenvolveu, com a popularização dos negócios na Web, standards-chave baseados em XML (como XML Schema e XSLT), e standards importantes para serviços Web: SOAP e WSDL. A W3C é conhecida pelo seu rigor e formalidade no desenvolvimento de standards, com vários passos de revisão e publicação, mas este rigor custa tempo: o processo de desenvolvimento de um standard pode demorar na W3C dois a três anos.

– **Organization for the Advancement of Structured Information Standards (OASIS)**

A OASIS foi originalmente estabelecida em 1993 sob o nome SGML Open, e alterou o nome cinco anos mais tarde para representar uma mudança do SGML para standards abertos. Entre os standards mantidos e desenvolvidos pela OASIS contam WS-BPEL, contribuições

para UDDI, e o framework WS-Security.

A OASIS foca na utilização de standards estabelecidos pela W3C para desenvolver especificações adicionais específicas às indústrias, e os seus processos de desenvolvimento são mais curtos que os da W3C. A OASIS conta mais de 600 organizações entre os seus membros.

– **Web Services Interoperability Organization (WS-I)**

A WS-I é conhecida pelo desenvolvimento do Basic Profile, documento de recomendações que estabelece como os standards disponíveis devem ser usados em conjunto, para assegurar a maior interoperabilidade do SOA da organização, internamente e com outros SOAs. O Basic Profile define as configurações para versões específicas de WSDL, SOAP, UDDI, XML e XML schema, assegurando que todos os aderentes ao Basic Profile possam interagir com facilidade. A WS-I também desenvolveu o Basic Security Profile, um documento semelhante que reúne as tecnologias de segurança mais modernas para serviços Web e XML.

Estabelecida em 2002, a WS-I integrou a OASIS em abril de 2012, devido à “convergência entre as actividades das organizações” e para “salvaguardar as suas realizações e continuar a avançar a missão”.

2.3. Conceito de Serviços

Os serviços e serviços compostos representam os elementos-base para a composição de uma plataforma SOA. Um serviço pode ser visto como uma função de negócio inteligente que combina dados e lógica para formar uma interacção abstracta com um serviço de negócio subjacente.

Um serviço tem de suportar dois requerimentos essenciais: uma interface correctamente definida e uma ligação (binding). O interface é o contrato que define a especificação do serviço e é representado como um documento WSDL para web services baseados em SOAP. A ligação é o protocolo de comunicação que define como o cliente irá comunicar com o serviço. Exemplos de tais protocolos: SOAP sobre HTTP; Java Messaging Service, RMI e EJB. Através do uso de uma combinação destes requerimentos, será possível desenvolver um cliente para interagir com o serviço. O desenho do interface irá determinar o nível de utilidade do serviço.

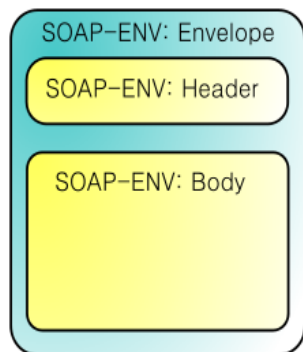


Fig. 2.2. Estrutura do documento XML com SOAP

Um tipo primitivo (ou atômico) pode ser usado para tarefas simples como filtragem de conteúdos ou routing e normalmente envolve dois ou três componentes individuais. Uma composição complexa pode ser um processo BPEL que contém múltiplos nós ou passos. Uma maneira de resolver os problemas de integração que possam surgir na implementação desses serviços é através de tecnologias de mediação de web services.

Um serviço composto, como o nome indica, é criado combinando a funcionalidade de um ou mais componentes. Serviços compostos podem servir para abstrair a funcionalidade e são considerados serviços de granularidade [grossa] (por exemplo um serviço para criar um novo cliente). Um serviço composto pode depois ser combinado com outros serviços para formar serviços compostos de nível mais alto ainda. Os serviços compostos partilham os mesmos requerimentos que os componentes: um interface e uma ligação. Estes compostos podem ser primitivos (para processos simples) ou complexos (baseado em BPEL com vários segmentos e nós).

- Serviços Web: especificações

Existe uma grande variedade de especificações associadas com serviços web, colectivamente referidas por “WS-*”. Estas especificações formam um framework básico de serviços web estabelecidos por standards de primeira geração representados por WSDL, UDDI e SOAP. “WS-” é um prefixo usado para indicar a associação com serviços web, e existem vários standards WS- incluindo WS-Addressing, WS-Discover, WS-Federation, WS-Policy, WS-Security e WS-Trust, que servem para definir padrões de interacção entre serviços Web.

- Serviços: composição

Para estandardizar o desenvolvimento de aplicações seguindo os princípios SOA,

foram desenvolvidos frameworks de forma a permitir uma melhor integração e unificação dos componentes e modelos:

O framework OSGi (Open Services Gateway initiative) foi o primeiro a ser desenvolvido (v 1.0 em maio de 2000) para standardizar a composição de serviços e definir uma plataforma unificada, implementando um modelo de componentes completo e dinâmico para Java. É actualmente mantido pela OSGi Alliance que publica actualizações e documentação do projeto. Pouco tempo depois surgiu o SCA (Service Component Architecture), uma especificação originalmente criada por fornecedores incluindo Oracle e IBM, que também disponibiliza um modelo para a composição de serviços e criação de aplicações segundo os princípios SOA. SCA é actualmente mantida pela OASIS, e tem bastantes aderentes entre os fornecedores: IBM, Oracle, SAP AG, Oracle, Sun, Redhat, TIBCO, Intel e outros. A maior parte das arquitecturas conforma-se com um dos frameworks mencionados mas há sempre a possibilidade de integrar composições de outros frameworks. Entre os frameworks open source frequentemente utilizados actualmente existem: Apache Axis2, Apache CXF, Spring WS, JBossWS e Metro da Sun, com suporte variado a standards e especificações.

- Gestão do ciclo de vida dos serviços:

As ferramentas de governance focam na gestão da expansão do negócio, na gestão do reconhecimento e consumo dos serviços, enquanto as ferramentas de gestão destinam-se ao departamento informático para assegurar os níveis de qualidade, segurança e disponibilidade através da monitorização e SLAs. Muitos ESBs incorporam funcionalidades de gestão.

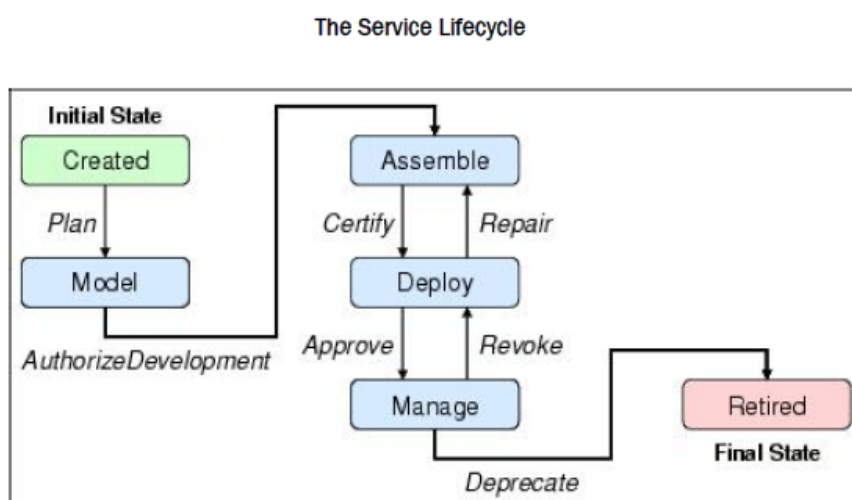


Fig 2.3. Ciclo de vida de um serviço (IBM)

Existem vários modelos de gestão e governance de serviços, para objectivos de apresentação consideraremos o modelo IBM (Fig 2.3):

O ciclo de vida de um serviço pode ser decomposto em seguintes passos:

- Modelação: levantamento de requisitos, modelação e simulação, desenho
- Construção: construção e testes
- Produção (Deploy): integração de pessoas, processos e informação
- Gestão (Manage): gestão de aplicações e serviços, gestão de identidade, gestão de aderência (compliance), métricas de negócio

2.4. Conceito de Processos de negócio

Os processos de negócio são escritos na linguagem BPEL (*Web Services Business Process Execution Language - WS-BPEL*), uma linguagem de orquestração para especificar interacções com Web Services e permitir o suporte de operações de negócios. O facto de ser uma linguagem de orquestração implica um controlo central e automatização de todas as transacções pelo servidor em contraste com uma linguagem de coreografia, onde os processos seguem um protocolo de acção (a coreografia). Inicialmente, BPEL apenas permitia realizar interacções entre serviços web, sem poder incluir agentes humanos no processo. Em 2007, um conjunto de developers (SAP, Oracle, Adobe, IBM, BEA e Active Endpoints) publicaram as especificações BPEL4PEOPLE que permitem a modelação das interacções humanas em BPEL, possibilitando desta forma a modelação de processos ainda mais completos. Um processo de negócio em SOA pode ser um serviço ou uma composição de serviços.

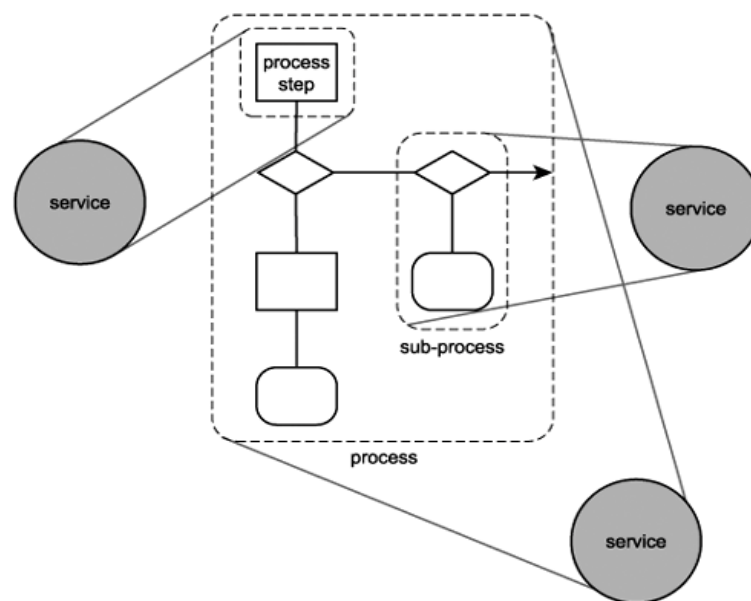


Fig. 2.4. Processos e serviços (Erl 2005)

BPEL permite também a integração de código Java, na sua versão BPELJ, onde blocos de código Java podem ser executados durante a actividade do processo, sem ter de recorrer a webservices (por exemplo efectuar um cálculo do lado do cliente ou construir um documento XML).

Os processos de negócio necessitam de um servidor apropriado, mas existe uma multitude de servidores BPM open source ou freeware que podem ser utilizados/adaptados pela empresa: IBM WebSphere Business Integration Server Foundation, Oracle BPEL, Active Endpoints ActiveWebflow Server (para JEE) ou OpenStorm Service Orchestrator Microsoft para .NET.

A norma JSR 208 (também conhecida por JBI, Java Business Integration) estende a plataforma JEE para um meio de desenvolvimento com BPEL e WSCI (web services choreography interface – descreve a funcionamento de protocolos e serviços web, e pode ser considerado o complementar do BPEL, na medida em que descreve o funcionamento técnico enquanto BPEL descreve o funcionamento lógico dos processos).

Os processos de negócio são modelados com a linguagem BPMN (Business Process Modeling Notation), com a possibilidade de o fazer também em UML. Existem várias formas de transformar um modelo BPMN/UML em código BPEL, um deles sendo o plugin bpmn2bpel para Eclipse ou a ferramenta Visual Paradigm. É possível modelar directamente em BPEL (por exemplo no Oracle BPEL Process Designer), mas a legibilidade de BPMN é útil para apresentações a pessoas não-TI. Infelizmente, é difícil obter código BPEL legível por um humano a partir de um modelo BPMN, mas as ferramentas de modelação oferecem uma visão compreensível dos modelos. (“Introduction to BPEL,” Techtarget)

2.5. Stack empresarial

A “stack” SOA é uma abstracção de SOA em camadas lógicas, onde a distribuição em camadas representa a separação de interesses. A realização dos serviços é efectuada pelos componentes da arquitectura, que também são responsáveis pelas suas funcionalidades e sua qualidade de serviço. Os processos de negócios são suportados por uma coreografia dos serviços expostos e combinados em aplicações compostas. Uma arquitectura de integração deve suportar o roteamento, mediação e tradução dos serviços (e suas mensagens) através de um ESB (ou produto similar de comunicação como o Websphere MQ). Os serviços realizados

devem ser monitorizados e geridos para assegurar a qualidade de serviço e aderência aos requerimentos não-funcionais. O diagrama na fig 2.5 representa SOA como um conjunto de nove camadas lógicas. Uma camada não depende necessariamente da camada inferior, por exemplo, um SOA pode ter a camada de consumidor (ou camada de apresentação) a interagir directamente com a camada de serviços. Uma organização pode escolher preencher as camadas do modelo conforme as suas necessidades de integração ou requisitos operacionais.

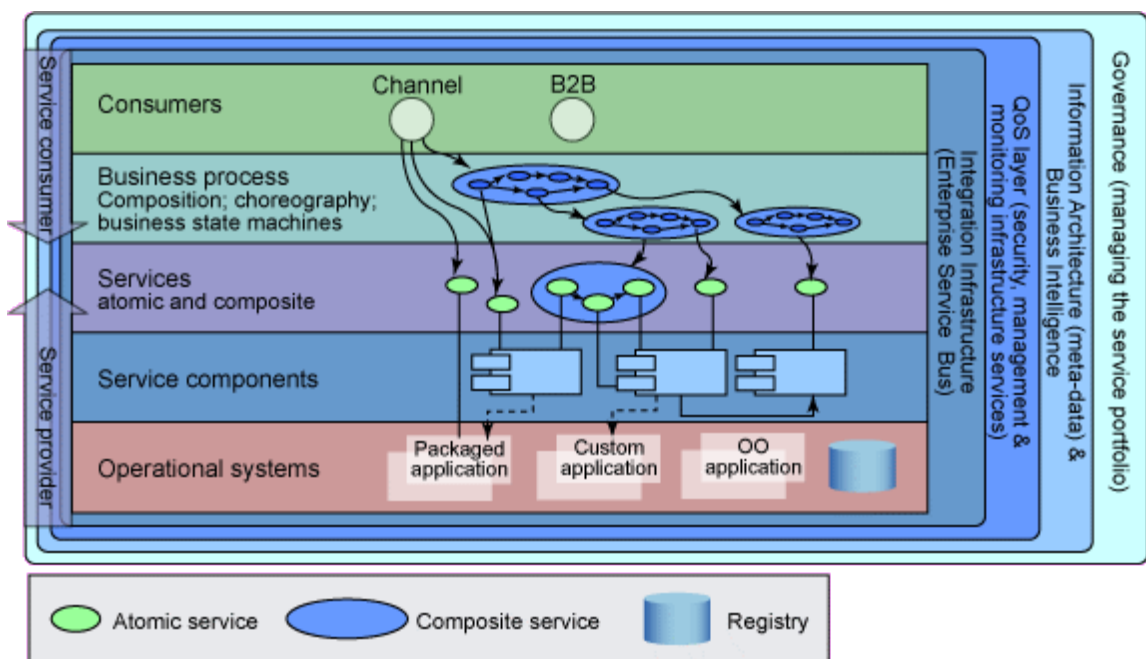


Fig 2.5. Modelo abstracto de stack SOA (IBM)

Neste modelo, a mesma entidade pode ser o consumidor e o provedor dum serviço. A separação é feita para preservar a relação lógica de negócios do ponto de vista do serviço. Uma organização pode ter diferentes departamentos que utilizam este padrão arquitectural (um departamento sendo o fornecedor de serviço e outro o seu consumidor) alterando-o segundo os seus requisitos.

Vamos examinar as camadas em mais detalhe:

2.5.1. Camada de sistemas operacionais.

Esta camada é composta de aplicações-legacy, sistemas existentes como CRM e ERP, e outras implementações mais antigas orientadas a objectos. Estes sistemas incluem:

- Aplicações monolíticas JEE ou .NET;

- Aplicações e sistemas legacy;
- Sistemas existentes de processamento de transacções
- Aplicações e soluções ERP e CRM (por ex. SAP ou Oracle)

2.5.2. *Camada de componentes de serviços:*

Esta camada contém componentes de software contendo a informação relativamente a implementação, realização ou operação de um serviço. Os componentes de serviços refletem a definição de serviços, incluindo as funcionalidades e a qualidade de serviço. Estes componentes podem aderir às definições SCA ou SDO.

A camada de componentes de serviços garante o alinhamento da implementação TI através da conformidade aos contratos definidos na camada de serviços (camada 3).

2.5.3. *Camada de Serviços*

Esta camada inclui todos os serviços definidos dentro da arquitectura. As especificações abstratas contém informação suficiente para permitir aos consumidores a invocação de funções de negócios expostas pelo provedor de serviço. A especificação pode ser (mas não obrigatoriamente) escrita em WSDL, e pode incluir um documento de políticas, descrições para a gestão SOA e informação sobre as dependências de serviços.

Os serviços expostos que residem nesta camada podem ser descobertos e invocados pelos consumidores de serviços, e podem ser utilizados numa coreografia para criar um serviço composto.

As funcionalidades das componentes empresariais e de negócios também podem ser expostos através da exportação dos seus interfaces como descrições de serviços nesta camada.

2.5.4. *Camada de Processos de Negócios*

Esta camada contém as composições e coreografias dos serviços expostos na camada 3. Os serviços são juntados em processos através de uma orquestração ou coreografia e agem como uma aplicação única. Estes processos podem ser desenhados com aplicações de modelação como o IBM WebSphere Business Integration Modeler. Esta camada também cobre a gestão do ciclo de vida de processos e inclui um motor de processos de negócios (por exemplo um motor WS4BPEL).

A camada de processos de negócios comunica com a camada de apresentação para comunicar os inputs e resultados dos vários utilizadores do sistema através de portais ou

aplicações B2B. As mensagens dos processos são transmitidas e transformadas pela camada de integração (camada 6), a estrutura das mensagens é definida dentro da camada de arquitectura de informação (camada 8), os KPIs para cada tarefa ou processo são definidas na camada de qualidade de serviço (camada 7), o desenho das agregações é apoiado pela camada de governance (camada 9), e todos os serviços devem ser representados e descritos na camada de serviços (camada 3).

2.5.5. *Camada de Consumidor*

A camada de consumidor (também conhecida como camada de apresentação) contém todas as funcionalidades necessárias para apresentar as funcionalidades TI e dados ao utilizador final. Esta camada também pode conter um interface para a comunicação entre aplicações.

Esta camada contém as possibilidades para criar o front-end de processos e aplicações compostas, e responder às mudanças através de canais, portais, aplicações-cliente e outros mecanismos. A adopção de padrões de acesso comprovados (por exemplo portais) e standards abertos (por exemplo Web Services for Remote Portlets) pode diminuir o tempo dos ciclos de desenvolvimento e produção através da reutilização de blocos pré-fabricados e verificados.

Estas práticas promovem uma visão de apresentação unificada e um ponto único de acesso às aplicações e acessos suportados. O ponto único de acesso integra-se com outros serviços críticos como segurança e confiança e melhora a usabilidade dos processos e aplicações.

2.5.6. *Camada de Integração*

A camada de integração contém as funcionalidades de mediação, roteamento e transporte dos pedidos de serviço do consumidor ao provedor correcto. As funcionalidades da camada de integração podem ser implementadas por um ESB ou um serviço de mediação e transporte como o Websphere MQ. O objectivo desta camada é de permitir a integração de serviços através de vários mecanismos.

2.5.7. *Camada de Qualidade de Serviço*

A camada de qualidade de serviço oferece as funcionalidades para realizar os requerimentos não-funcionais de um SOA. É uma camada de observação das outras camadas e pode emitir alertas ou eventos quando são detectados (ou, preferivelmente, antecipados) problemas de aderência aos contratos. Basicamente é a camada que assegura os requerimentos de fidelidade, disponibilidade, escalabilidade e segurança. Também permite valorizar as

funcionalidades de negócios do SOA através da monitorização de processos de negócios.

2.5.8. *Camada de Arquitectura de Informação e Business Intelligence*

A oitava camada assegura a inclusão de considerações importantes relativamente à arquitectura de dados e às arquitecturas de informação que podem ser usados para a criação de business intelligence através de data marts e data warehouses. Esta camada é especialmente aplicada à soluções SOA para indústrias específicas, e inclui estruturas de dados para indústrias, XML schemas, e protocolos de comunicação de dados de negócios.

2.5.9. *Camada de Governance*

A camada de governance cobre todos os aspectos da gestão operacional de ciclo de vida em SOA. Esta camada ajuda a fazer decisões e gerir todos os aspectos da solução SOA, incluindo performance, segurança e monitorização. A camada de governance pode ser aplicada a todas as outras camadas na stack SOA. Esta camada oferece um framework de governance que inclui SLAs baseados em QoS e KPIs, políticas de planificação e gestão para soluções SOA, e linhas condutoras para efeitos de segurança de aplicação.

Nota-se que as regras e políticas de negócios não são reunidas em uma camada, porque as regras de negócio abrangem todas as camadas. Por exemplo, as regras e políticas para processos de negócio são definidas pela intersecção das camadas de governance e da camada de processos de negócios. As regras de validação na camada de consumidor e as transformações de input e output provenientes e destinadas a esta camada são definidas na intersecção da camada de consumidor e da camada governance.

2.6. Componentes da stack empresarial:

Existem vários padrões na integração middleware para cobrir um máximo de funções essenciais e formar uma base conforme com os princípios fundamentais SOA e ao mesmo tempo realizar as necessidades de negócios da organização. Uma stack empresarial normalmente inclui:

- Servidor de Aplicação
- ESB ou software de mediação de mensagens
- Registo e/ou Repositório, software para Governance

- Motor de processamento de eventos complexos e monitorização de negócio (CEP e BAM)
- Software de Gestão de decisões empresariais (Enterprise Decision Management)
- Ferramentas de modelação e desenvolvimento
- B2B e outros adaptadores diversos

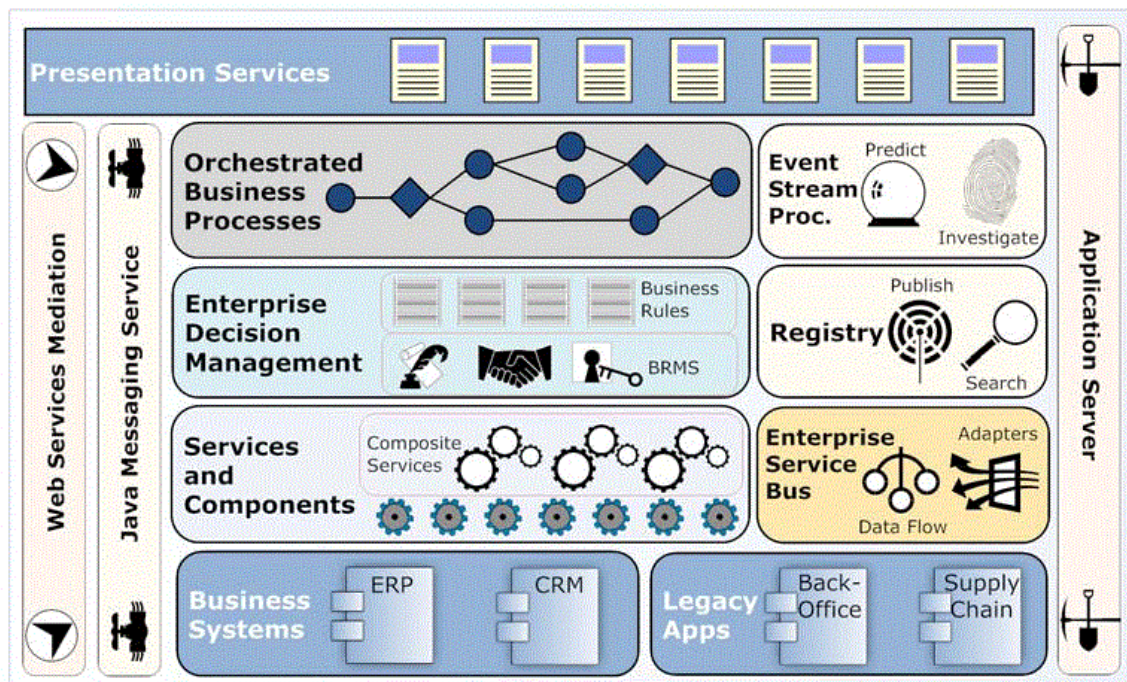


Fig. 2.6. Stack SOA típica (Davis 2009)

2.6.1. Servidor de Aplicação

O servidor de aplicação é um conjunto de software que oferece um ambiente integrado para executar aplicações dos clientes. Também chamados de “containers”, este middleware inclui uma variedade de linguagens, conectores e adaptadores, bibliotecas de recursos, e código administrativo para executar, configurar e conectar as aplicações. Num ambiente empresarial, normalmente funciona em conjunto com um servidor Web e uma/várias bases de dados, ou pode executar as funções de ambos o servidor Web e servidor de aplicação. Em contraste a outros modelos cliente/servidor, o servidor de aplicação executa a maior parte da lógica de negócios, evitando a instalação de software cliente no ponto de destino, oferecendo desta forma maior flexibilidade e eficiência na organização e com parceiros, fornecedores ou clientes. Isto faz do servidor uma peça chave num ambiente SOA.

Um servidor de aplicação expõe a lógica de negócio às aplicações através de vários protocolos, com ajuda de APIs como o modelo EJB (Enterprise JavaBeans), e o cliente pode invocar esta lógica como se invocasse um método a partir de um objecto. As outras funcionalidades do servidor de aplicação incluem balanceamento de carga, tratamento de excepções, disponibilidade, segurança das aplicações e gestão de sistemas distribuídos.

As seguintes 3 plataformas para servidores de aplicação são as que se encontram mais frequentemente: J2EE (Java Enterprise Edition), .NET (Microsoft) e LAMP (Linux, Apache, MySQL, PHP). As suites SOA comerciais incluem por norma um servidor J2EE ou .NET proprietário, adaptado e integrado com os outros componentes da suite, mas outros servidores podem ser integrados através de adaptadores.

2.6.2. ESB

Um enterprise service bus (ESB) é um componente central da arquitectura SOA. O termo apareceu primeiro em 2002, seguido de uma vasta quantidade de produtos que se auto-proclamavam de ESBs. Efectivamente há sempre discussão sobre o que compõe realmente um ESB. No mínimo um ESB suporta um sistema de comunicação empresarial baseado em XML. As mensagens podem ser inteligentemente roteadas e transformadas através de uma arquitectura descentralizada. Transformação de dados, routing, mediação de serviços, conversão de protocolos, Webservices e SLAs todos integraram o ESB, que se estabeleceu como o apogeu da computação empresarial.

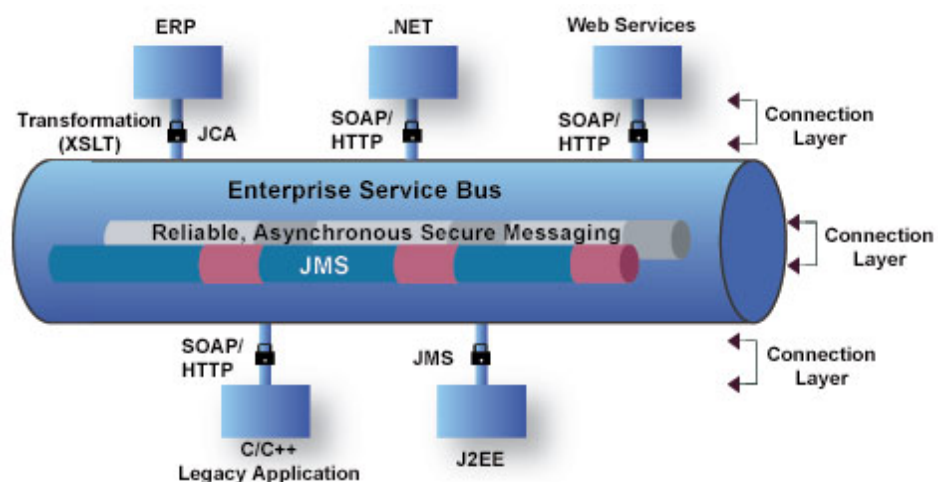


Fig.2.7. ESB Fiorano (Fiorano 2007)

A emergência dos ESBs coincidiu com a crescente standardização de protocolos de Web Services e de mensagens tal como o Java Messaging System e SOAP. Os primeiros produtos ESB de facto apresentavam um middleware baseado em JMS, (Sonic software e Fiorano). Pouco depois apareceram soluções open-source como o Mule e ServiceMix. A seguir a estes, surgiram outros ESB open-source: Apache Synapse, JBoss ESB, OpenESB da Sun.

Num ambiente Java, o bus é apostado num backbone JMS. As mensagens são depositadas no bus, com regras de roteio e transformações que são aplicadas para permitir a navegação da mensagem até o seu serviço de destino. Os ESBs incluem adaptadores para sistemas proprietários, o caminho preferencial para trabalhar com parceiros é de usar XML sobre HTTP (SOAP, XML ou REST). Isso elimina a necessidade de instalar software próprio no destino e permite uma integração mais rápida, simples e flexível.

2.6.3. Registo e Repositório de Serviços, Governância

A promessa de reutilização, flexibilidade, integração e governância em SOA é realizada pela separação das descrições dos serviços da sua implementação, e usar esta metadata através do ciclo de vida do serviço. Os detalhes técnicos são documentados em metadata artifacts/deliverables tais como WSDL, esquema XML ou documentação SCA (Service Component Architecture) e envolvem o que o serviço faz, como pode ser invocado ou como interage com outros serviços. Estes artefactos podem ser associados a outras anotações e metadata para informar os potenciais consumidores do serviço sobre as suas funcionalidades e objectivos.

A metadata é utilizada durante as várias etapas do ciclo de vida para tarefas como reutilização, configuração, aplicação das políticas requeridas pelos SLA e dar uma visão mais completa do ambiente de serviços.

A reusabilidade dos serviços depende fortemente da capacidade de descrever e publicar as funcionalidades dos serviços para os potenciais consumidores. Um registo permite organizar a informação sobre os seus serviços e fornece funcionalidades para a sua publicação e descoberta. Os standards usados actualmente para estas funcionalidades é o UDDI (Universal Description, Discovery and Integration) e WSDL (Web Services Description Language), juntamente com o SOAP, são standards para descrever serviços e seus provedores, e como os serviços são consumidos.

A governância num sistema organizacional:

- estabelece constrangimentos sobre as decisões
- determina quem toma responsabilidade e quem tem autoridade para tomar decisões
- estabelece constrangimentos e parâmetros que controlam, guiam e influenciam as decisões
- prescreve consequências de não-adesão às regras

Do ponto de vista técnico, Governance SOA é realizada através de aplicações de gestão dos recursos no registo e repositório, estabelecimento de perfis e permissões, gestão de contratos dos serviços. Os principais fornecedores tem o seu próprio software de Governance que é integrado com o respectivo registo/repositório, mas prevê a ligação de diferentes sistemas.

2.6.4. CEP, BAM

As empresas modernas recebem um grande número de mensagens e eventos, e precisam de os analisar e reagir em tempo real para manter a competitividade e agilidade. Software de processamento complexo de eventos (anteriormente existente como Processamento de Stream de Eventos) permite processar um grande volume de eventos em tempo real e identificar padrões ou anomalias que possam afectar o negócio ou as operações, e elicitar uma resposta conforme as regras previamente estabelecidas.

Um motor de processamento de eventos pode, além de participar no aspecto de negócios da organização, também participar no aspecto operacional e processar eventos de sistema e tornar o sistema de informação mais útil ainda, permitindo automatizar respostas operacionais e detectar padrões de falhas ou identificar oportunidades de optimização.

BAM (Business Activity Monitoring) é designação dada ao software que permite realizar a monitorização da actividade de negócio. Uma actividade de negócio pode ser um processo de negócio orquestrado por software BPM, ou um processo de negócio que inclui uma série de actividades abrangendo múltiplos sistemas e aplicações. BAM destina-se aos gestores e efectua as funcionalidades de processamento de eventos e mostrar informação em tempo real através de dashboards com KPIs (Key Performance Indicators) sobre as actividades de negócio. BAM distingue-se de Business Intelligence pelo processamento de eventos e apresentação dos resultados em tempo real enquanto os dashboards BI são actualizados em intervalos pré-determinados.

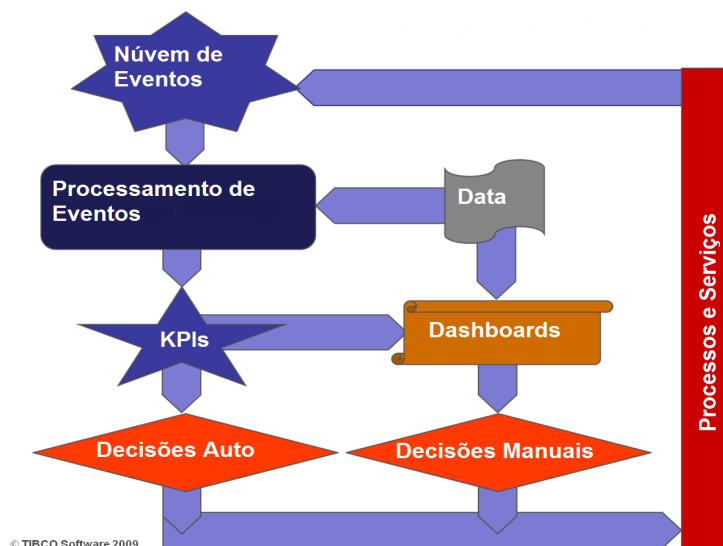


Fig. 2.8. CEP simplificado da Tibco (Tibco 2009)

Todas as soluções BAM processam eventos. As soluções BAM da prévia geração necessitavam um investimento em BPM antes de poderem ser implementadas, mas a geração moderna de BAM são baseadas em motores CEP, retirando a dependência BPM.

2.6.5. Enterprise Decision Management, Business Rules

Com o crescimento do volume de informações recebidas na operação dos negócios, torna-se necessário automatizar e agilizar o processo de tomada de decisões. A metodologia EDM consiste na utilização de um motor de regras de negócio (BRE) para executar regras, acoplado com um sistema de gestão de regras de negócios (BRMS). As regras de negócio explicam procedimentos inerentes ao negócio escritos de forma percebível a quem está envolvido no negócio, por exemplo políticas de extensão de crédito se o cliente tiver mais de X anos na empresa. Estas regras, se forem definidas no código das aplicações, são difíceis de mudar em caso de alteração dos requisitos de negócio, por isso um BRMS procura separar as regras de negócios do código das aplicações. Esta metodologia permite também transformar as regras de negócio em recursos da organização, e publicá-las para consumo como serviços de decisão que podem ser invocados por outros serviços e aplicações. EDM permite assim centralizar a gestão das regras e lógica de negócio.

O EDM pode ser complementado de CEP ou BAM para juntar as funcionalidades de identificação de padrões do motor de detecção com a publicação da lógica de negócios de forma a automatizar e simplificar as operações. É um componente essencial de apoio aos

processos de negócios da empresa, permitindo um melhor alinhamento entre os serviços e processos e oferecendo uma infraestrutura explícita para a tomada de decisões inteligentes.

2.6.6. *Ferramentas de desenvolvimento e modelação*

Uma organização com SOA necessita de ferramentas para todas as fases do ciclo de vida de SOA: ferramentas de desenho, modelação e planeamento dos processos de negócio e serviços, ambientes de desenvolvimento integrados para a implementação técnica de serviços e do software em geral. Este software desempenha um papel importante para todos os utilizadores do sistema TI, especialmente no que envolve o factor humano e o talento da empresa. Todos os principais fornecedores oferecem um conjunto de software que oferece todas ou parte destas funções na sua stack, será importante o software de modelação de processos e serviços e software de desenvolvimento de aplicações.

No ambiente Java, o IDE predominante de desenvolvimento de aplicações é o Eclipse, enquanto a plataforma .NET utiliza o Visual Studio, mas existe uma variedade de IDEs proprietários que oferecem funcionalidades específicas para todas as fases do ciclo de vida e serão descritas em mais detalhe na apresentação dos provedores de SOA.

Outros componentes de uma stack SOA incluem (a completar):

B2B – Possibilidade de realizar transações business to business. Estas transacções são frequentemente encontradas nas interacções com fornecedores.

Adaptadores e conectores – Pacotes de código preparados para facilitar interacções com outros sistemas, criar APIs personalizadas, permitir acesso às bases de dados, etc.

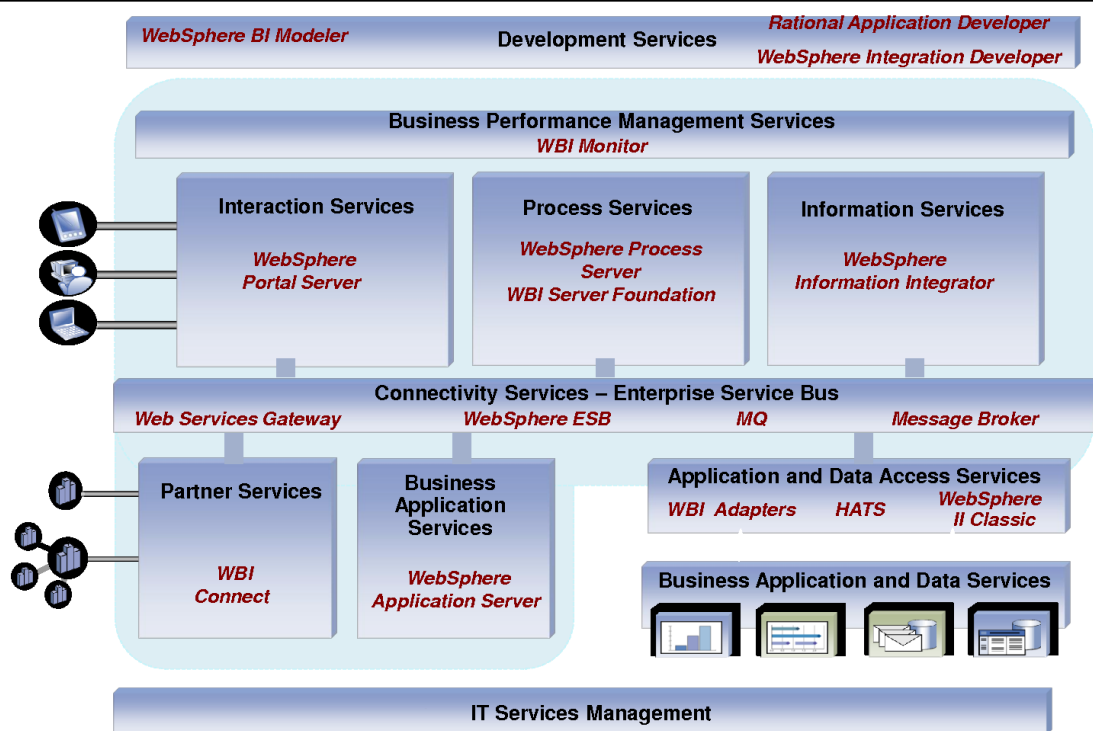
3. Apresentação do software SOA

3.1. Suites SOA Comerciais

Como referência de comparação comercial, utilizaremos algumas das suites estudadas pelos institutos de análise Forrester e Gartner: IBM SOA Foundation baseado em Websphere, Oracle Weblogic, Tibco ActiveMatrix, Microsoft Biztalk e SAP Netweaver.

3.1.1. IBM WebSphere

IBM WebSphere Reference Architecture



■ IBM Software Offerings

Fig. 3.1. SOA IBM (IBM)

IBM é um dos líderes na indústria software e hardware TI e marca consistentemente valores altos nos estudos de mercado Forrester, Gartner e Wintergreen. O ano 2011 marcou o seu 11º ano consecutivo de liderança no mercado de integração empresarial, com 32% de cota de mercado. (“IBM News room - 2012-04-02 Report: IBM Named Marketshare Leader in Middleware Software - United States,”)

A IBM acompanha o movimento SOA e tem uma gama de produtos para responder aos desafios de integração e orientação a serviços. Além de desenvolver soluções proprietárias, a IBM participa no open source através de contribuições de código ou aplicações (por ex Eclipse doado ao open source em 2001, e contribuições continuada à Eclipse Foundation), pela participação no desenvolvimento de standards aberto (IBM colabora com a W3C, OASIS, Opengroup, WS-I, Open Management Group e outras), e, recentemente pela oferta de algumas aplicações em versão comunidade (grátis mas limitado e sem suporte).

A stack SOA IBM Websphere apresenta uma gama variada de produtos que responde às necessidades SOA e BPM da empresa. Esta variedade é tal que introduz redundância entre as funcionalidades dos diferentes produtos. A IBM promete que a suite é totalmente interoperável e modular, suporta standards actualizados e integra-se facilmente aos produtos dos outros fornecedores. A suite suporta também integração com o sistema operativo de mainframe z/OS.

	IBM
Application server	WebSphere Application Server v 8.5
ESB	WebSphere ESB, MQ, Message Broker
Registry/Repository	IBM Websphere Registry and Repository
Governance	Tivoli Composite Application Manager
BRE, BRMS	IBM Operational Decision Management / ILOG Rules for BRMS
CEP, BAM	Websphere Business Events
BPM	Websphere Business Process Manager
Desenvolvimento e modelação	Rational Developer IBM Eclipse Business Modeler
Adaptadores	SAP, Siebel, SQL, Oracle, JD Edwards, e vários específicos às indústrias
Funcionalidades	B2B
Sistemas operativos	Windows 7, 8, Server 2003, 2008, 2012, Vista, XP; Solaris 10; SUSE 10, 11; RHEL 5, 6; Asianux 3; IBM i 6.1, 7.1; HP-UX 11v2, 11v3, AIX 6.1, 7.1

3.1.2. Oracle Weblogic

A Oracle é um outro grande actor no mercado de tecnologias de informação. Notório pelos seus sistemas de gestão de bases de dados e ERPs, a Oracle participa no desenvolvimento de integração com a oferta Fusion Middleware. Baseada no servidor de aplicação Weblogic, esta solução possui todos os componentes para realizar SOA e BPM, tal

como uma gama de adaptadores pré-feitos e modelos específicos à indústrias, e integra facilmente com outros produtos Oracle, como as bases de dados ou ERP.

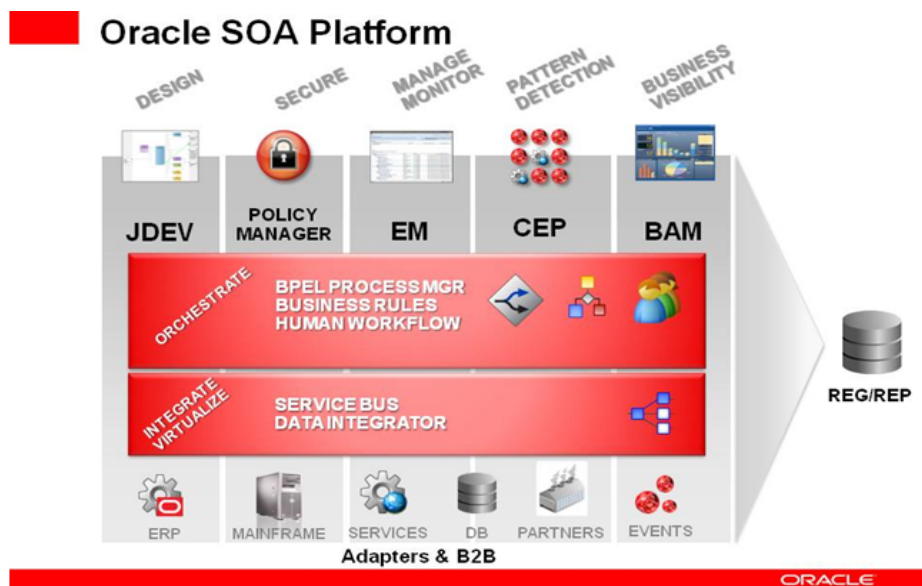


Fig 3.2. SOA Oracle

A Oracle é um outro grande actor no mercado de tecnologias de informação. Notório pelos seus sistemas de gestão de bases de dados e ERPs, a Oracle participa no desenvolvimento de integração com a oferta Fusion Middleware. Baseada no servidor de aplicação Weblogic, esta solução possui todos os componentes para realizar SOA e BPM, tal como uma gama de adaptadores pré-feitos e modelos específicos à indústrias, e integra facilmente com outros produtos Oracle, como as bases de dados ou ERP.

Tal como a IBM, a Oracle suporta open source e permite a utilização de aplicações open source juntamente com o seu software. Além disso, contribuiu e constinua a desenvolver vários projectos como o Glassfish Application Server, Berkeley DB e InnoDB. Engenheiros da Oracle integram várias comunidades open source e reconhecem a importância da utilização de standards abertos. (“Oracle’s Support for Open Source and Open Standards,” Oracle.) A Oracle detinha 43% do mercado de servidores de aplicação em 2011 com o servidor WebLogic, segundo a Gartner.

	Oracle
Application server	Oracle Weblogic Server 12
ESB	Oracle Service Bus, Oracle Mediator
Registry/Repository	Oracle Service Registry, Oracle Enterprise Repository
Governance	Oracle Web Service Management
BRE, BRMS	Oracle Rules

CEP, BAM	Oracle CEP, Oracle BAM
BPM	BPEL PM
Desenvolvimento e modelação	Jdeveloper, Weblogic Workshop
Adaptadores	SAP, Siebel, SQL, Oracle, JD Edwards, e vários específicos às indústrias
Funcionalidades	B2B
Sistemas operativos	Windows 7, Vista, XP, Server 2008 R2; Mac OS Snow Leopard (10.3.6); Solaris 10; SUSE 10, 11; RHEL 5, 6; Oracle Linux 5, 6; AIX 6.1, 7.1

3.1.3. *Tibco ActiveMatrix Service Grid*

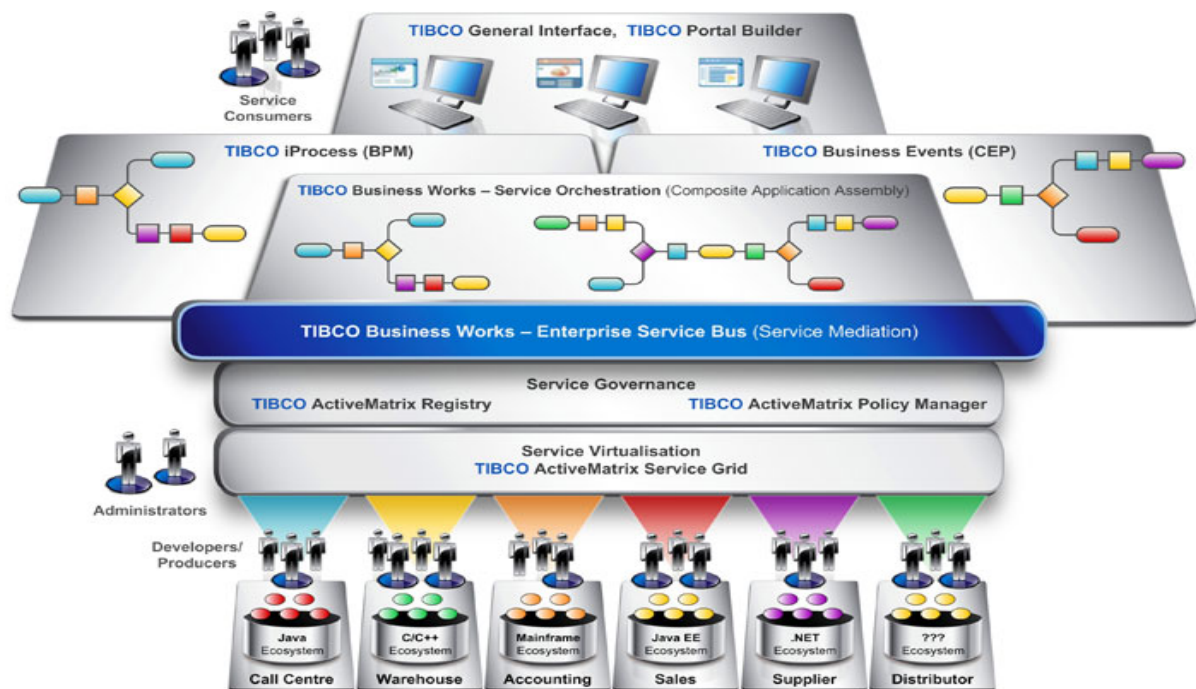


Fig 3.3. *Tibco ActiveMatrix*

Na década de 80, a empresa Tenekron Software digitalizou Wall Street com o seu “information bus”, permitindo negócios automatizados em tempo real. Em 1997 o fundador da Tenekron fundou a TIBCO (The Information Bus Company). A sua tecnologia foi usada por empresas como a SAP, IBM e Oracle. Mais tarde, tornou-se parceiro da Microsoft em tecnologias “push” (entrega de conteúdos internet através de browsers).

Actualmente o software da Tibco foca-se em middleware para comunicação B2B e B2C, integração de sistemas e feedback em tempo real. Um exemplo do seu software é o sistema de recomendações personalizadas da Amazon. A Tibco revela bom posicionamento nas análises de Governance Integrada e Enterprise Service Bus da Forrester nos últimos anos.

A solução Activematrix baseia-se na plataforma Service Grid para uma realizar uma infraestrutura SOA distribuída escalável e extensível, com funcionalidades BPM, CEP e

Governance. Também conta com uma quantidade impressionante de adaptadores para integração com outros sistemas, incluindo mainframes.

	Tibco
Application server	Tibco ActiveMatrix Service Grid
ESB	ActiveMatrix Service Bus, Businessworks
Registry/Repository	ActiveMatrix Registry
Governance	ActiveMatrix Policy Director, AMX Service Lifecycle Governance
BRE, BRMS	Tibco Business Events
CEP, BAM	Tibco Business Events, Tibco Spitfire
BPM	Tibco iProcess
Desenvolvimento e modelação	Tibco Application Development Platform, Tibco General Interface
Adaptadores	SAP, Siebel, SQL, Oracle, JD Edwards, IBM, e vários específicos à indústrias
Funcionalidades	B2B
Sistemas operativos	AIX 5.3, HP-UX 11i v3, Solaris 9, 10, RHEL 4, 5, OpenSuse 10, Windows XP Pro, Server 2003, 2008

3.1.4. Microsoft Biztalk

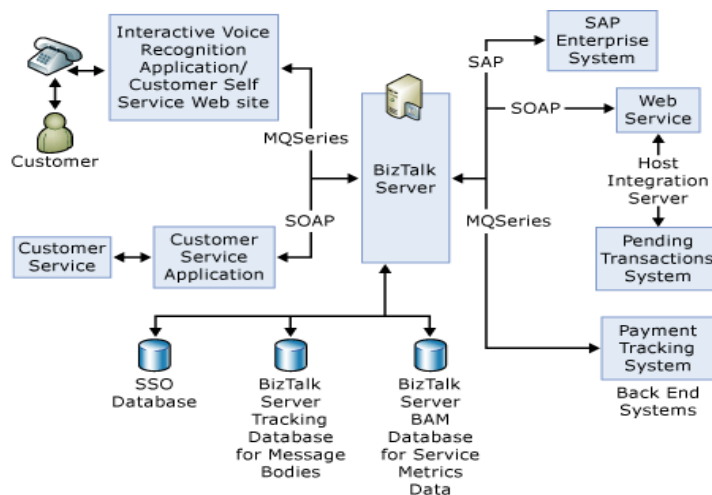


Fig 3.4. Exemplo de SOA com Biztalk (Microsoft)

A Microsoft participa também na arena SOA, com o servidor Biztalk baseado na plataforma .NET e framework WCF (Windows Communication Foundation). A suite permite efectuar todas as funcionalidades de um SOA exceptuando a componente de Governance. Este componente Governance pode ser implementada com software de outras empresas, por exemplo com o produto SOA Governance for Biztalk da SOA Software.

BizTalk Server pode ser estendido com o ESB Toolkit para disponibilizar funcionalidades ESB, ou pode ser utilizado um outro ESB ou sistema de mensagens

Microsoft, por exemplo MQ, WCF é uma alternativa ao SCA do mundo Java e suporta interoperabilidade com aplicações WCF numa máquina Windows ou serviços Web desenvolvidos noutra plataforma a correr em Windows ou outros sistemas operativos.

	Microsoft
Application server	BizTalk2010, AppFabric
ESB	Biztalk ESB Toolkit 2.1
Registry/Repository	Microsoft UDDI Services
Governance	
BRE, BRMS	Microsoft Business Rules Framework
CEP, BAM	Biztalk Server Activity Monitoring, Microsoft StreamInsight
BPM	Sharepoint
Desenvolvimento e modelação	Visual Studio, Web Service Software Factory
Adaptadores	SAP, Siebel, SQL, Oracle, JD Edwards, PeopleSoft, TIBCO, IBM
Funcionalidades	B2B
Sistemas operativos	Windows Server 2008 R2, Windows Server 2008 with Service Pack 2, Windows 7, Windows Vista with Service Pack 2

3.1.5. SAP Netweaver

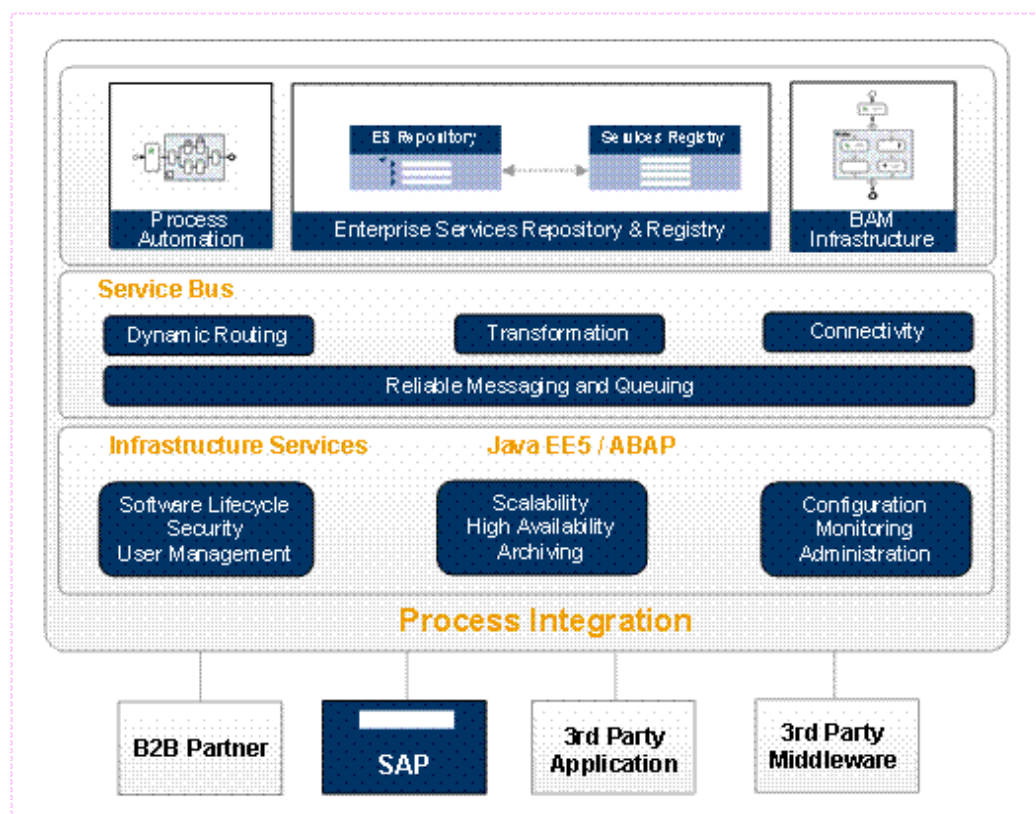


Fig. 3.5. SOA com Netweaver

A SAP é conhecida pelas suas soluções ERP, CRM e SCM e é um actor importante no mercado empresarial e industrial. A solução de integração Netweaver permite integrar aplicações SAP e não-SAP e é baseado em ABAP, linguagem de programação para negócios com sintaxe semelhante ao COBOL utilizada em todas as aplicações SAP. Netweaver é também a fundação tecnológica para várias aplicações SAP que tomou a iniciativa estratégica de juntar a infraestrutura e as aplicações numa “apliestrutura”: o Netweaver Application Server é o run-time onde são executadas as outras aplicações (CRM, SCM, ERP, SRM, PLM) SAP.

SAP Netweaver, apesar de ser baseado em ABAP, utiliza também C, C++ e Java EE, e a plataforma é baseada em open standards para seguir a visão estratégica SAP. As contribuições open source da SAP são a MaxDB, uma base de dados open source desenvolvida com o rigor SAP e SAP Hana, uma plataforma de analíticas.

	SAP
Application server	SAP Netweaver Application Server 7.3
ESB	SAP Netweaver Process Integration (PI)
Registry/Repository	SAP Enterprise Service Registry and Repository
Governance	Decision Service Management
BRE, BRMS	NetWeaver Process Orchestration
CEP, BAM	SAP Sybase Event Stream Processor, EventInsight (BAM Limitado)
BPM	Netweaver BPM, NetWeaver Process Orchestration
Desenvolvimento e modelação	Sybase PowerDesigner, SAP Netweaver Visual Composer, Web Dynpro
Adaptadores	SAP, Siebel, SQL, Oracle, JD Edwards, e vários específicos às indústrias
Funcionalidades	B2B
Sistemas operativos	Windows 2000 Professional, XP Professional, Server 2003; Solaris 9, 10; SUSE SLES 9; RHEL 4; HP-UX 11.11, 11.23; AIX 5.2, 5.3

3.2. Suites SOA open source

3.2.1. WSO2

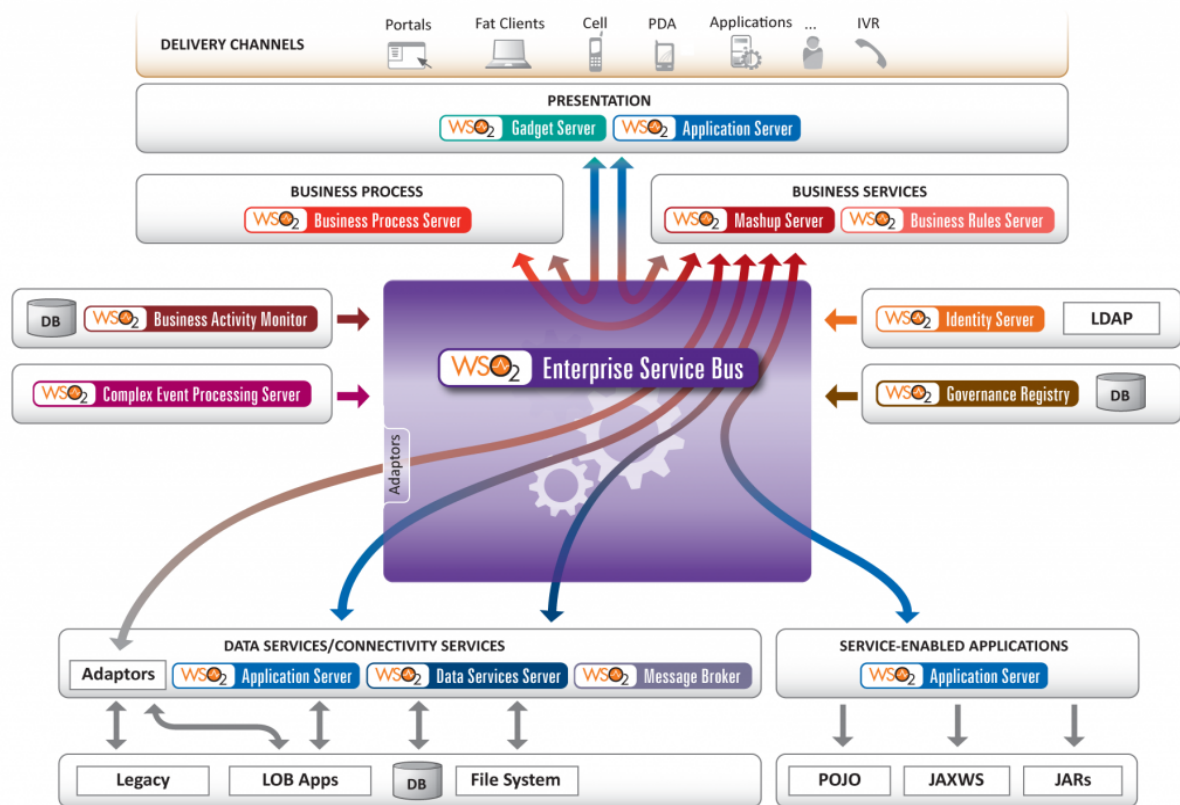


Fig. 3.6. WSO2 Carbon

A suite WSO2 apresenta uma solução completa para realizar BPM e SOA de raiz ou complementar um ERP e/ou SOA existentes. Denominada Carbon, a plataforma utiliza um framework próprio desenvolvido com base no servidor Apache tomcat. A suite é modular e integrada, e o estúdio de desenvolvimento baseado em Eclipse está integrado com todos os componentes. O leque de standards suportados é considerável: a quasi-totalidade de transportes, formatos, protocolos e especificações está referenciada.

A WSO2 é um contribuinte-chave na comunidade Apache, em projectos como Apache Axis2, Apache Synapse, Apache Axiom e outros. Os produtos WSO2 são distribuídos sob licença Apache versão 2, e existem opções de subscrição para suporte. Os seus clientes incluem empresas de áreas diversas: ebay, Volvo, British Airways, Lockheed Martin utilizam aplicações WSO2 para áreas específicas do seu negócio. (“Case Studies | WSO2”)

	WSO2
Application server	WSO2 Application Server
ESB	WSO2 ESB
Registry/Repository	WSO2 Governance Registry
Governance	WSO2 Governance Registry
BRE, BRMS	WSO2 Business Rules Server
CEP, BAM	WSO CEP e BAM
BPM	WSO2 Business Process Server
Desenvolvimento e modelação	WSO2 Developer Studio
Adaptadores	WSO2
Funcionalidades	B2B
Sistemas operativos	Windows, Linux, Solaris, Ubuntu, Fedora, Mac OS X, Gentoo, SUSE, Debian com um runtime JDK 1.6.x

3.2.2. JBoss

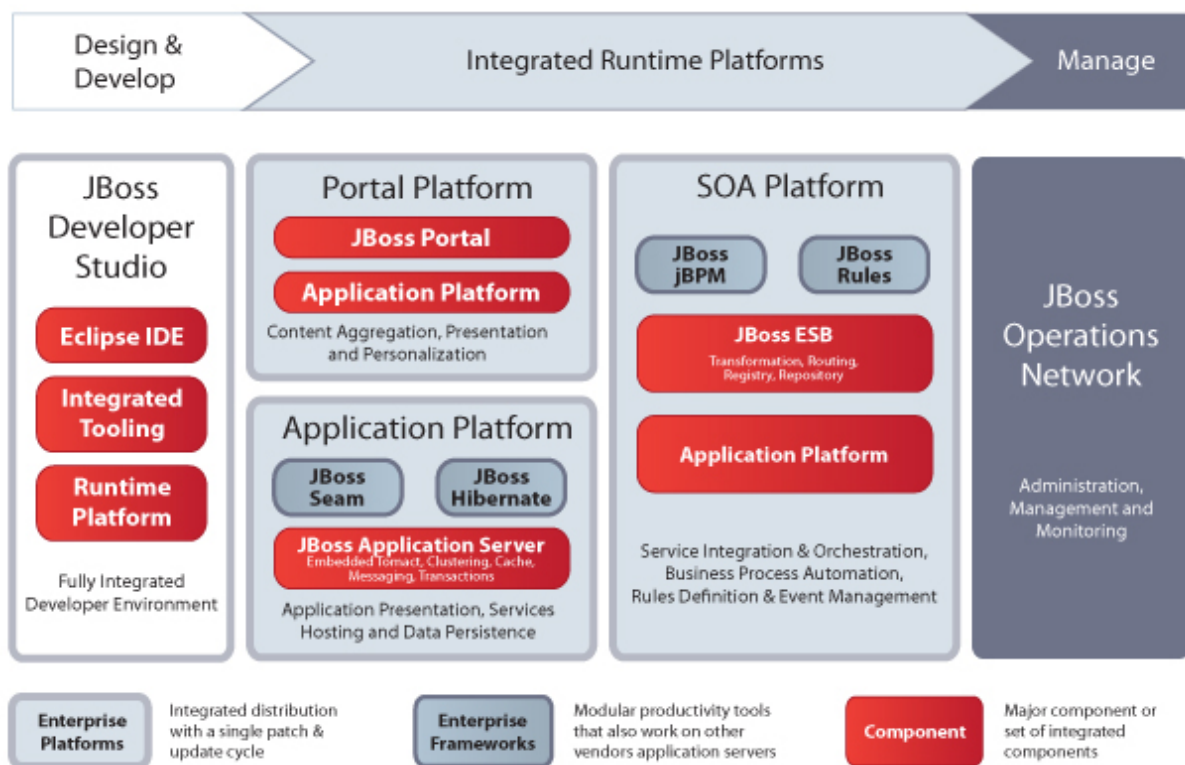


Fig. 3.7. JBoss SOA Platform

JBoss SOA Platform é um projecto open source adquirido pela Red Hat, é baseada no servidor de aplicação JBoss. A suite combina vários projectos com maturidade e integra-os para permitir a realização de um SOA e BPM. Integra um estúdio próprio de desenvolvimento, mas também é possível utilizar o Eclipse para o desenvolvimento de aplicações. A componente de governance pode ser a versão empresarial Operations Network

ou a iniciativa open source JBoss Overlord. Os componentes da suite tem maturidade, e suporte pode ser obtido através da Redhat, empresas comerciais ou a comunidade JBoss.

A plataforma de aplicação JBoss é utilizada em muitas empresas, em vários casos para substituir uma plataforma Oracle existente. (“Red Hat | Stater Further Increases Stability and Performance of Mission-Critical Applications with JBoss Enterprise Appl”), (“Red Hat | Ampersand Creates Robust Customer Loyalty Program with Red Hat Solutions”)

	JBoss
Application server	JBoss Application Server 7.1
ESB	JBoss ESB
Registry/Repository	Maven Repository
Governance	JBoss Operations Network, JBoss Overlord
BRE, BRMS	Drools, JBoss Enterprise BRMS
CEP, BAM	Drools Fusion, Overlord BAM
BPM	JBoss JBPM
Desenvolvimento e modelação	JBoss Developer Studio, Eclipse
Adaptadores	Comunidade JBoss
Funcionalidades	B2B possível
Sistemas operativos	A maioria dos sistemas capazes de correr uma JVM incluindo Linux, UNIX, Windows e Mac OS.

3.2.3. *Apache ServiceMix*



Fig. 3.8. *Apache ServiceMix*

Apache ServiceMix é um projecto ESB distribuído open source da Apache Software Foundation que permite realizar SOA numa empresa. Inicialmente construído com base em JBI, com o declínio do standard passou a se basear em OSGi através da integração de Apache

Karaf que adiciona funcionalidades de container de aplicação. ServiceMix é facilmente embebível e pode ser corrido numa infraestrutura cliente/servidor Java, como ESB standalone ou como serviço dentro de outro ESB. ServiceMix vem integrado com outros projectos Apache: Apache ActiveMQ, Apache Camel e Apache CXF para funcionalidades de infraestrutura e comunicação e Apache ODE para BPM. Eventualmente poderá ser integrado com outros projectos open source como o repositório Maven e o CEP Esper para completar o leque SOA e Apache Tuscany para SCA e SDO.

A comunidade Apache é muito activa e competente, existe uma grande quantidade de plugins e adaptadores para os projectos Apache e o desenvolvimento de aplicações pode ser realizado em Eclipse. A fraqueza do ServiceMix revela-se na falta de um componente governance integrado, que pode ser implementado com software de outras empresas (por exemplo o produto WSO2).

	Apache ServiceMix
Application server	ServiceMix
ESB	ServiceMix NMR, ActiveMQ, Camel
Registry/Repository	(Maven)
Governance	
BRE, BRMS	Apache ODE
CEP, BAM	(Esper)
BPM	Apache ODE
Desenvolvimento e modelação	Eclipse
Adaptadores	comunidade
Funcionalidades	Múltiplos plugins
Sistemas operativos	Linux, UNIX, Mac OS, Windows

3.2.4. *Mule*

Mule é um ESB distribuído que contém containers para aplicações Java e serviços Web, e conectores para todo tipo de protocolos, bases de dados, bindings e portais. É um ESB com maturidade e possibilidades de integração com servidores de aplicação e bases de dados comerciais e open source. A Mulesoft é também provedora de CloudHub, uma PaaS de integração. Os produtos Mulesoft são utilizados em várias grandes empresas de diversas áreas, tais como Facebook, Zynga, Popcap, Forbes e Nestlé.

A versão empresarial do Mule inclui, integrado ao ESB, o Business Event Analyzer para BAM e o Anypoint Service Registry para realizar governance. Para realizar um SOA

completo, o Mule deve ser complementado de componentes BRE e CEP, como por exemplo o CEP Esper e a suite BPM Activiti, ambos open source.

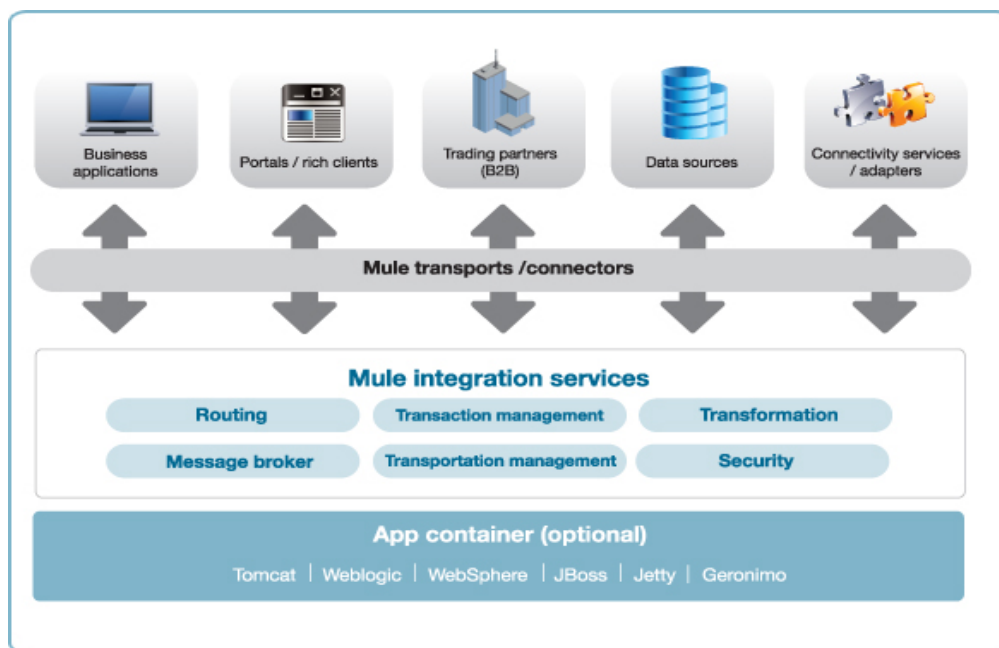


Fig. 3.9. Mule

A TiVo escolheu o Mule ESB que permitiu implementar uma infraestrutura SOA robusta e escalável para expandir o negócio e reduzir o tempo de desenvolvimento de novos serviços em 75%. (“Case Study: TiVo | MuleSoft”) No caso da Nespresso, o Mule ESB e o servidor JBoss foram utilizados para realizar a integração entre os vários componentes ERP, gestão de armazéns e canais de distribuição e formar uma infraestrutura SOA capaz de suportar as necessidades do negócio e abrir novos canais de distribuição. (“Case Study: Nespresso | MuleSoft”)

	Mule
Application server	Mule server
ESB	Mule ESB
Registry/Repository	Anypoint Service Registry
Governance	Anypoint Service Registry
BRE, BRMS	(Activiti)
CEP, BAM	(Esper – CEP), Business Event Analyzer - BAM
BPM	(Activiti)
Desenvolvimento e modelação	Mule Studio, Eclipse
Adaptadores	Mule
Funcionalidades	Múltiplos plugins, B2B
Sistemas operativos	Sistemas operativos capazes de correr JVM

3.2.5. *Petals*



Fig. 3.10. Petals

Petals é mantida pela comunidade open source OW2. A plataforma utiliza um modelo de arquitectura distribuída P2P similar ao do Mule, onde o contentor/servidor pode ser instalado em várias máquinas Windows, Mac OS ou Linux, cada instalação com os seus próprios serviços e um registo partilhado ao invés da topologia centralizada ou em clusters.

O ESB pode ser complementado com outros componentes Petals para disponibilizar todas as funcionalidades de um SOA, e pode ser integrado com a maioria de servidores de aplicação open source ou proprietários. Petals é baseado em JBI, mas as versões mais recentes integram o framework Petals Fractals para disponibilizar SCA e OSGi, e existe a possibilidade de integrar o framework Axis2.

	Petals
Application server	Petals ESB
ESB	Petals ESB
Registry/Repository	Petals Master
Governance	Petals Master
BRE, BRMS	Petals BPEL Engine
CEP, BAM	Petals View
BPM	Petals BPM
Desenvolvimento e modelação	Petal Studio
Adaptadores	comunidade
Funcionalidades	Múltiplos plugins e componentes
Sistemas operativos	Windows, Linux, MacOS

4. Comparação de completude da stack e standards

Para realizar a comparação foram reunidos vários standards e especificações actualmente utilizados no desenvolvimento de serviços web e o suporte de cada suite a esses standards. Na lista de factores de comparação figuram especificações, protocolos de comunicação, frameworks de composição de serviços web e o suporte às diferentes tecnologias Java actualmente utilizadas como JSP, EJB e JPA. No caso da Microsoft, existem tecnologias .Net para realizar funcionalidades equivalentes, como Windows Forms e XBAP. A tabela no apêndice 1 contém uma explicação mais detalhada de cada elemento.

Na recolha dos seguntes dados, existem casos onde a versão específica do standard em questão não foi especificado na documentação do produto, apenas consta a designação geral do standard ou especificação, e uma pesquisa mais aprofundada nos fóruns de suporte não trouxe mais esclarecimentos. Nestes casos consta a palavra “sim” e o standard em questão é considerado como plenamente suportado.

4.1. Suites comerciais

Suites:	IBM	Oracle	Tibco	Microsoft	SAP
Standards Serviços Web:					
JSR 109/921 (Implementing)	1.3	1.3	-	-	1.2
JSR 181 (metadata)	2.0	2.0	2.0	-	2.0
APIs para serviços Web:					
JAX-WS	2.2	2.2	2.0	-	2.0
JAX-RS	1.0, 1.1	1.1	1.0	-	1.0
Suporte	Ajax, WS-	Ajax, WS-	Ajax, WS-	Ajax, WS-	Ajax, WS-
Frameworks de composição:					
OSGI	sim	sim	sim	-	sim
SCA	Com Feature Pack 7 ou Spring	sim	sim	-	Com plugins
Outros	Spring	Spring, CAPS, JBI	Spring, JBI	Spring, ASMX, WCF	Spring, ABAP 4
Standards XML:					
XSLT	2.0	2.0	2.0	2.0	2.0
XQuery	1.0	1.0	1.0	1.0	1.0

XPath	2.0	2.0	2.0	2.0	2.0
Protocolos de transporte:					
HTTP/HTTPS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS
JMS	sim	sim	sim	Com JNBridge	sim
Outros	Suporte RESTful, POP, IMAP, SMTP, AMQP, FIX, TCP, UDP, FTPS, SFTP, CIFS, MLLP	Suporte RESTful, POP, IMAP, SMTP, AMQP, FIX, TCP, UDP, FTPS, SFTP, CIFS, MLLP	Suporte RESTful, POP, IMAP, SMTP, AMQP, FIX, TCP, UDP, FTPS, SFTP, CIFS, MLLP	Suporte RESTful via WCF, POP, IMAP, SMTP, AMQP, FIX, TCP, UDP, FTPS, SFTP, CIFS, MLLP	Suporte RESTful, POP, IMAP, SMTP, AMQP, FIX, TCP, UDP, FTPS, SFTP, CIFS, MLLP
Messaging:					
SOAP	1.1, 1.2	1.1, 1.2	1.1, 1.2	1.1, 1.2	1.1, 1.2
SOAP w/ attachments/SAAJ	1.2/1.3	1.3	1.3	1.3	1.3
SOAP MTOM	sim	sim	sim	sim	sim
Description:	IBM	Oracle	Tibco	Microsoft	SAP
UDDI	3.0	3.0	3.0	3.0	3.0
WSDL	1.1	1.1	1.1	1.1	1.1
JAXR	sim	sim	sim	sim	sim
Outros	Múltiplas BD	Múltiplas BD	Múltiplas BD	Múltiplas BD	Múltiplas BD
Interoperability:					
Basic Profile	1.1, 1.2, 2.0	1.1, 1.2, 2.0	1.1	1.1, 1.2, 2.0	1.1
Basic Security Profile	1.0	1.0	parcial	parcial	1.0
Security/Reliable communication:					
WS-Security	1.1	1.1, 1.0	1.0	1.0	1.0
WS-SecurityPolicy	1.2	1.2	1.2	1.2	1.2
WS-Policy	1.5	1.5, 1.2	1.2	1.2	1.2
WS-Addressing	1.0	1.0	1.0	1.0	1.0
WS-ReliableMessaging	1.1	1.2, 1.1	1.2, 1.1	1.1	1.2, 1.1
WS-Trust	1.3,1.2	1.4,1.3	1.3	1.3	1.3
Data Bindings:					
JAXB	2.0, 2.1, 2.2	2.1	2.0, 2.1, 2.2	2.0	2.0
SDO	2.1.1	2.1.1	-	-	2.1
JMS	sim	sim	sim	sim	sim
XMLBeans	2.0	2.0	-	-	2.0

Transacções atómicas:					
WS-AtomicTransactions	1.0, 1.1, 1.2	1.0, 1.1, 1.2	1.0, 1.1, 1.2	1.0, 1.1, 1.2	1.0, 1.1, 1.2
WS-Coordination	1.0, 1.1, 1.2	1.0, 1.1, 1.2	1.0, 1.1, 1.2	1.0, 1.1, 1.2	1.0, 1.1, 1.2
Tecnologias Java:					
JEE	JEE6, JSE7	JEE6, JSE7	JEE6, JSE6	.NET 4.0	JEE5, JSE6
Servlet	3.0	3.0	2.5	-	2.5
JSP	2.2	2.2	sim	- Windows forms, XBAP	2.1
JSF	2.0	2.1	não	-	1.2
EJB	3.1	3.1	3.0	-	3.0
JDBC	4.1	4.0	4.0	-	4.0
JPA	2.0	2.0	2.0	- NPA	2.0

Informação recolhida em:

IBM: (“WebSphere Application Server Version 8.5 Information Center,” IBM)

Oracle: (“Oracle Fusion Middleware 11g (11.1.1.6) Documentation Library,”),
 (“Features and Standards Supported by WebLogic Web Services - 11g Release 1 (10.3.6),”)

Tibco: (“Tibco Activematrix 3.2.0 Release Notes,” Tibco.)

Microsoft: (“Microsoft BizTalk Server Product Overview,” Microsoft), (“Windows Communication Foundation,” Microsoft)

SAP Netweaver: (“Technology Standards | SCN,” SAP), (“SAP NetWeaver 7.3 – SAP Help Portal Page,” SAP)

4.2. Open source

Suites:	WSO2 Carbon	JBoss	Apache ServiceMix	Mule	Petals
Standards Serviços Web:					
JSR 109/921 (Implementing)	1.3	1.3	-	-	-
JSR 181 (metadata)	2.0	2.0	2.0	-	2.0
APIs para serviços Web:					
JAX-WS	2.2	2.2	2.0	2.2	2.1
JAX-RS	1.1	1.1	1.1	1.1	-

Suporte	Ajax, WS-	Ajax, WS-	Ajax, WS-	Ajax, WS-	Ajax, WS-
Frameworks de composição:					
OSGI	sim	sim	sim	sim	sim
SCA	-	com Tuscany	com Tuscany	-	sim
Outros	Spring, WSF, Axis2	Spring, CXF, JCA	Spring, CXF, Axis2, JBI	CXF, Spring, JCA	JB1, Fractals, Axis2(com Presto)
Standards XML:					
XSLT	2.0	1.0 (2.0 com saxon)	2.0	2.0	2.0
XQuery	1.0	1.0	1.0	1.0	-
XPath	2.0	1.0	2.0	2.0	-
Protocolos de transporte:					
HTTP/HTTPS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS	HTTP 1.0/1.1, TLS, SSL, SOAP/HTTP, REST/HTTP, XML/HTTP, XML/JMS
JMS	1.1	1.1	1.1	1.1	1.1
Outros	Suporte RESTful, POP, IMAP, SMTP, AMQP, TCP, UDP, FTPS	Suporte RESTful, FTP, UDP, POP, IMAP, AMQP, FTPS, TCP, UDP	Suporte RESTful, FTP, TCP, UDP, POP, IMAP, SMTP	Suporte RESTful POP, IMAP, SMTP, SMTPS, RMI, TCP, UDP, XMPP	Suporte RESTful FTP/SFTP, SMTP, POP, IMAP, SSL
Messaging:	WSO2 Carbon	JBoss	Apache ServiceMix	Mule	Petals
SOAP	1.1, 1.2	1.1, 1.2	1.1, 1.2	1.1, 1.2	1,1, 1.2
SOAP w/ attachments/SAAJ	1.3	1.3	sim	sim	-
SOAP MTOM	sim	sim	sim	sim	sim
Description:					
UDDI	3.0	2.0, 3.0	-	2.0, 3.0	-
WSDL	1.1, 2.0	1.1	1.1	2.0	1.1, 2.0
JAXR	1.0	1.0	-	1.0	-
Outros	Múltiplas BD	Multiplas BD	Multiplas BD	Multiplas BD	Multiplas BD
Interoperability:					
Basic Profile	1.1	1.1	1.1	1.1	1.1
Basic Security Profile	1.1	1.1	-	1.0	-
Security/Reliable communication:					
WS-Security	1.0, 1.1	1.1	1.1	1.1	1.0
WS-SecurityPolicy	1.2	1.3	1.2	1.2	1.2 (Axis2)

WS-Policy	1.5	1.5	1.5	-	1.5 (Axis2)
WS-Addressing	1.0	1.0	1.0	-	1.0 (Axis2)
WS-ReliableMessaging	1.1 (Sandesha2)	1.1	1.1	1.1	1.1 (Sandesha2)
WS-Trust	1.3	1.3	1.3	1.3	1.3 (Axis2)
Data Bindings:					
JAXB	2.1	2.2	2.2	sim	sim
SDO	(com tuscany)	(com tuscany)	(com tuscany)	(com tuscany)	sim
JMS	sim	sim	sim	sim	sim
XMLBeans	2.3	Com plugins	Com plugins	Com plugins	sim
Transacções:					
WS-AtomicTransactions	(com Kandula)	1.1	-	-	-
WS-Coordination	(com Kandula)	1.1	-	-	-
Tecnologias Java:					
JEE	JEE 6, JSE 7	JEE 6, JSE 7	JEE 6, JSE 7	JEE 6, JSE 7	JEE 6, JSE 7
Servlet	3.0	3.0	2.5	3.0	2.5
JSP	- (custom ui)	2.2	1.1.2	sim	-
JSF	-	2.0	-	sim	-
EJB	3.0	3.1, 3.0	3.0, 3.1	3.1	3.0, 3.1
JDBC	4.0	4.0	4.0	sim	sim
JPA	-	2.0	-	sim	-

Dados recolhidos em:

WSO2: (“WSO2 Carbon Documentation - WSO2 Carbon 4.0.3 - WSO2 Documentation”)

JBoss: (“JBoss Enterprise SOA Platform Supported Configurations - Red Hat Customer Portal.”), (“Documentation - JBoss AS 7.1 - Project Documentation Editor.” 2012)

Apache ServiceMix: (“Apache ServiceMix Documentation” Apache.), (“Apache CXF -- Index,” Apache.)

Mule: (“Platforms and Technologies Compatible with Mule ESB - Mule User Guide - Mulesoft.org Documentation.”), (“Supported Web Service Standards - Mule User Guide - Mulesoft.org Documentation.”)

Petals: (“Petals Enterprise Service Bus - Petals ESB 4.1 - Petalslink Documentation.”), (“Components User Guide - Petals Components - Petalslink Documentation.”)

4.3. Análise

A análise será realizada do ponto de vista de completude da stack e do suporte aos standards abertos, à partir dos dados recolhidos nos capítulos anteriores. Esta análise permitirá verificar o estado de maturidade do software SOA comercial e open source e verificar se existe alguma vantagem decisiva para um dos lados.

Em primeiro lugar determinaremos se uma stack permite implementar um SOA completo sem adição de componentes de partes terceiras, e atribuiremos a cada stack uma pontuação numa escala de 0 a 5, onde 0 indica nenhuma funcionalidade SOA, e 5 indica um SOA completo, com todos os componentes do capítulo 2.6 presentes na solução. Serão descontados valores por componentes ausentes ou com funcionalidade limitada (um servidor embebido no ESB vale 0.5 valores para os objectivos desta avaliação).

A segunda comparação é realizada entre os standards suportados pelas várias suites, segundo os dados nas tabelas das partes 4.1 e 4.2, e será avaliada seguindo uma escala de 0 a 12 valores. Cada categoria da tabela de standards onde a suite suporta todos os standards mais actualizados confere 1 valor, se a suite só tem suporte parcial é-lhe dado 0.25, 0.5 ou 0.75 valores, e se a stack não adere a nenhum dos standards presentes na categoria é-lhe atribuído zero valores pela categoria em questão. Apenas é considerado o suporte aos standards sem a adição de componentes, bibliotecas ou plugins adicionais. Os valores obtidos serão agrupados para cada suite de forma a dar uma pontuação total, e as notas de cada suite serão representadas num gráfico de maneira a formar uma representação visual do estado de maturidade relativo do SOA comercial e open source.

Note que existe uma grande variedade de standards actualmente em utilização, e que apenas foram recolhidos alguns dos mais frequentemente encontrados. O caso da Microsoft também merece menção particular, pois a plataforma baseia-se em .NET em vez de Java, pelo que a sua categoria das tecnologias Java contém os componentes .NET análogos aos componentes Java que realizam funções semelhantes.

Classificação da completude da stack:

Suite	Nota (Máx = 5)	Observações
IBM Websphere	5	Alguns componentes com funcionalidades redundantes
Oracle Weblogic	5	
Tibco ActiveMatrix	5	
Microsoft Biztalk	4	Ausência de componente governance, funcionalidades ESB adicionadas através de toolkit adicional
SAP Netweaver	5	
WSO2 Carbon	5	
JBoss SOA Platform	5	
Apache ServiceMix	3	Servidor contido no ESB, Ausência de repositório, governance, CEP, mas possível com instalação de outros projectos Apache ou open source
Mule	3.5	Ausência de CEP e BPM, servidor contido no ESB
Petals ESB	3.5	A comunidade OW2 classifica o Petals Master como projecto em incubação pelo que é deduzido 0.5 valores, servidor contido no ESB, PetalsView relativamente limitado quando comparado à outras soluções BPM

Comparação de standards:

Suite/Categoria	SW	API	Fwk	XML	TSP	Msg	Dsc*	Int	SC	DBng	TA	TJ
IBM Websphere	1	1	0.75	1	1	1	1	1	1	1	1	1
Oracle Weblogic	1	1	1	1	1	1	0.75	1	1	1	1	1
ActiveMatrix	0.5	0.75	1	1	1	1	1	0.5	0.75	0.5	1	0.75
Microsoft Biztalk	-	0.5	0.75	1	1	1	0.75	0.75	0.75	0.5	1	0.75
SAP Netweaver	0.75	0.75	0.75	1	1	1	0.75	0.75	0.75	0.75	1	0.5
WSO2 Carbon	1	1	0.75	1	1	1	1	0.75	0.75	0.75	0	0.75
JBoss SOA	1	1	1	0.75	1	1	1	0.75	1	0.5	1	1
ServiceMix	0.5	1	0.75	1	1	1	0.5	0.5	1	0.5	0	0.5
Mule	0	1	0.75	1	1	1	1	0.75	0.75	0.5	0	1
Petals ESB	0.5	0.5	1	0.25	1	0.5	0.5	0.5	0.25	0.75	0	0.5

*Description: WSDL 2.0 ainda tem pouca utilização e não conta para a avaliação.

Avaliação final:

Suite/Nota	Completeness	Compliance
IBM Websphere	5	11.5
Oracle Weblogic	5	11.75
Tibco ActiveMatrix	5	9.75
Microsoft Biztalk	4	8.75
SAP Netweaver	5	9.75
WSO2 Carbon	5	9.75
JBoss SOA	5	11
ServiceMix	3	8.5
Mule	3.5	8.75
Petals ESB	3.5	6.25

Os resultados obtidos nesta tabela estão representados graficamente na figura 4:

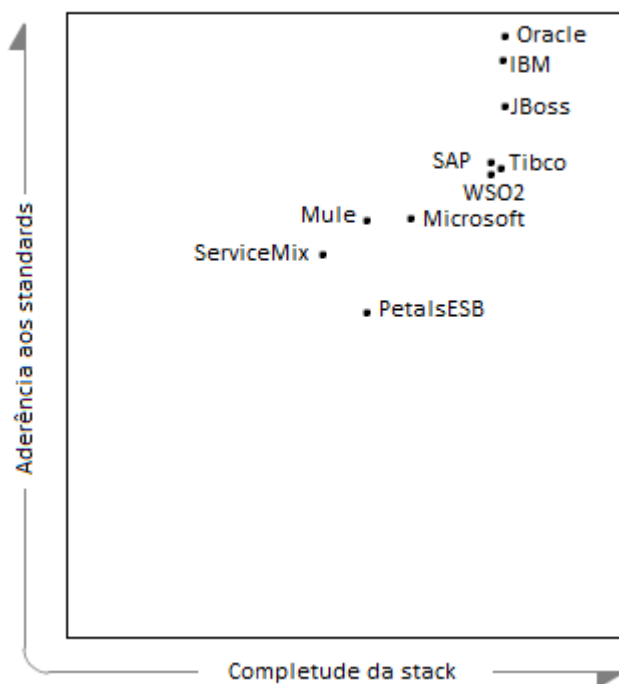


Fig. 4. Quadrante de comparação

Esta figura representa o estado de maturidade das suites SOA comerciais e open source “out-of-the-box”, ou seja sem a instalação de componentes ou bibliotecas adicionais. A JBoss e a WSO2 ambas apresentam suites completas e rivalizam com as suites comerciais presentes. ServiceMix, Mule e Petals são apresentados primariamente como ESBs que podem ser completados de outros componentes para realizar um SOA completo, o que explica a sua

posição mais baixa em termos de completude, e o facto de não serem considerados componentes adicionais prejudica a sua posição em termos de suporte standards visto que estes projectos são previstos para serem integrados com outros componentes e bibliotecas que lhes permitem ter suporte a mais standards e especificações. Por exemplo, o framework Axis2 adiciona, entre outras funcionalidades, o suporte às especificações WS-Coordination, WS-AtomicTransaction, WS-Policy e WS-Addressing.

Uma das razões pela completude das stacks comerciais é a série de aquisições de outras empresas com software que complementa as suas soluções ou melhora o seu posicionamento face aos outros participantes no mercado: por exemplo a IBM com a compra da Datapower, Ilog e Aptsoft ou a Oracle com a BEA e Amberpoint. Esta prática permite integrar um produto com maturidade para a solução proprietária, substituir um produto pouco performante, e ao mesmo tempo adicionar os clientes da empresa adquirida. No entanto, esta prática também suscita problemas para os existentes utilizadores da solução: incertezas sobre se terão de instalar e configurar outro produto, ou se o software que será substituído ainda terá suporte nos próximos anos, se terão de reavaliar os contratos ou adquirir outros produtos e aprender a utilizá-los.

Os fornecedores comerciais viram a popularidade de standards abertos crescer e tiveram de adaptar os seus modelos comerciais. O objectivo do software comercial é oferecer uma solução completa, pronta a funcionar com um mínimo de configuração para permitir realizar SOA e BPM na empresa, e, após a análise efectuada, verifica-se que o seu leque de funcionalidades está de facto completo e maduro. No entanto, como já foi referido, SOA é um conjunto de best-practices, uma forma de organização que deverá atravessar todos os actores e sistemas da empresa, e não se implementa simplesmente através da compra de um software mas sim através do estabelecimento de um plano de implementação à longo prazo, desenvolvido a partir de uma devida análise das necessidades de cada interveniente, dos parceiros de negócio e dos processos de negócio existentes, e será bem sucedido só através da aderência de todos os participantes às políticas previamente estabelecidas. O software é uma ferramenta de apoio a estas práticas, mas não será útil se a organização não aderir ao plano, e existe sempre uma percentagem de empresas que tentam adoptar SOA mas não vêem nenhum retorno.

Nem todas as empresas necessitam a instalação de todos os componentes das suites comerciais (na maioria dos casos estas suites são modulares, ou seja é possível adquirir

apenas o que é necessário), mas sem ter experiência prévia em SOA, podem acabar por adquirir componentes desnecessários ou redundantes.

Os fornecedores comerciais oferecem versões de avaliação com duração limitada do seu software para dar a oportunidade de experimentar os seus produtos (Netweaver, Oracle), ou versões comunidade com funcionalidades limitadas (Websphere Liberty Profile), e uma organização pode servir-se destas versões para testar o software no seu sistema, mas estas versões não se destinam para o ambiente de produção e são apenas utilizadas para referência de desempenho.

Finalmente, notam-se algumas curiosidades relativamente à aderência aos standards da parte das suites comerciais, por exemplo o facto de a SAP Netweaver não suportar JEE 6 ou as versões Servlet e JSF mais recentes: este atraso pode explicar-se pelo tamanho da plataforma, quantidade de componentes a actualizar e por uma priorização em outras áreas. Pela mesma razão, o versionamento de software comercial é por norma relativamente lento quando comparado com open source.

As soluções open source são mais frequentemente projectos de âmbito mais reduzido, concebidos para resolver um problema específico ou dar uma alternativa a um produto comercial existente (JBoss, ServiceMix), e no caso do SOA, as duas filosofias alinham-se através da utilização de standards abertos e independência de plataforma. Com open source, uma organização tem mais flexibilidade nas suas escolhas, e pode avaliar o software na sua versão completa. Como foi verificado na análise, software SOA open source tem maturidade adequada e aderência aos standards actuais, possui interoperabilidade com as principais suites comerciais e é possível integrar componentes open source para realizar todas as funcionalidades necessárias para SOA e BPM.

Para uma organização sem experiência prévia em SOA, o investimento inicial em esforço de desenho, implementação e formação será semelhante independentemente do software envolvido, pelo que software open source parece ser a opção de menor risco. Uma iniciativa SOA é um projecto de longo termo, os profissionais recomendam uma implementação faseada e sistemática, pelo que pode se revelar arriscado cometer-se a um único fornecedor (sobretudo se este cobrar custos de licença). Com software open source, uma organização tem mais liberdade na escolha do seu software, mais facilidade na substituição de produtos se estes se revelarem inadequados e corre menos riscos no caso de uma empresa ir à falência ou ser comprada (o software continua disponível na comunidade).

Software open source também tem a vantagem de ser mais rápido a ser actualizado, e responder mais especificamente ao objectivo para o qual foi criado. As comunidades dos projectos open source contam com profissionais na área que participam no desenvolvimento ou suporte, e podem promover mudanças para corresponder com as necessidades no seu negócio ou resolver bugs, que são implementadas mais rapidamente devido ao tamanho mais reduzido em comparação com o análogo comercial.

Em termos de suporte ao open source, existem várias possibilidades, como empresas especializadas em open source que oferecem serviços de consultoria, formação e suporte, ou a organização que mantém o projecto (Red Hat para JBoss, WSO2), ou até os fóruns do projecto onde os outros utilizadores partilham código e técnicas e podem ajudar a resolver problemas.

5. Conclusões

A partir da análise do capítulo anterior, SOA open source revela-se como uma alternativa viável ao SOA comercial, tanto para uma organização sem SOA, como para uma empresa que já utiliza SOA com software comercial. O sucesso do SOA depende em maior medida do desenho e da competência dos agentes humanos do que do próprio software. Antes de adoptar SOA, é necessário estudar as suas necessidades de integração, os diferentes modelos de implementação e até pode ser útil contratar os serviços de uma consultora para ajudar no desenho do plano e na escolha de software. Cada SOA é unicamente organizado e qualquer software terá de ser configurado especificamente para a organização, que deve decidir se os custos de licença do software comercial são justificados face à existência de software com as mesmas funcionalidades sem custos de entrada, mas que requerem mais esforço de implementação (e possivelmente formação).

Os diferentes casos de sucesso referidos anteriormente testemunham à viabilidade de open source para missões críticas e às vantagens da implementação do SOA no negócio. Open source dá a possibilidade a novas empresas de usufruir de um leque de ferramentas considerável para competir com outras empresas, sem ter de investir uma quantia significativa, e a natureza aberta é condutiva à inovação, adaptação e igual oportunidade. Os preconceitos existentes sobre software open source não se verificaram na análise do software SOA, que revelou-se de qualidade e confiança, pelo menos para os projectos referenciados neste estudo.

Para aprofundar o estudo sobre SOA open source, será interessante realizar a instalação destas plataformas num negócio fictício e avaliar a implementação de cada uma em termos de tempo de instalação e configuração, acessibilidade para o utilizador e clareza da documentação disponível, e testar o seu desempenho em situações de alto volume de operações.

Bibliografia

Apache CXF – Index. Data de acesso: 5 de janeiro 2013, em <http://cxf.apache.org/>

Apache ServiceMix Documentation -. (n.d.). Data de acesso: 4 de janeiro 2013, em <http://servicemix.apache.org/docs/4.4.x/user/what-is-smx4.html>

Bean, J. (2009). *SOA and Web services interface design principles, techniques, and standards*. San Francisco, Calif.; Oxford: Morgan Kaufmann.

Bloomberg, J. (2006, December 11). The Lego® Model of SOA | ZapThink. Data de acesso: 11 de janeiro 2013, em <http://www.zapthink.com/2006/12/11/the-legoreg-model-of-soa/>

Case Studies | WSO2. (n.d.). Data de acesso: 15 de janeiro 2013, em <http://wso2.com/casestudies/>

Case Study: Nespresso | MuleSoft. (n.d.). Data de acesso: 16 de janeiro 2013, em <http://www.mulesoft.com/case-study-nespresso>

Case Study: TiVo | MuleSoft. (n.d.). Data de acesso: 16 de janeiro 2013, em <http://www.mulesoft.com/case-study-tivo>

Components User Guide - Petals Components - Petalslink Documentation. (n.d.). Data de acesso: 15 de janeiro 2013, em <https://doc.petalslink.com/display/petalscomponents/Components+User+Guide>

Davis, J. (2009). *Open source SOA*. Greenwich, Conn.: Manning.

Documentation - JBoss AS 7.1 - Project Documentation Editor. (14 de dezembro 2012). Data de acesso: 5 de janeiro 2013, em <https://docs.jboss.org/author/display/AS71/Documentation>

Erl, T. (2005). *Service-oriented architecture : concepts, technology, and design*. Upper Saddle River, NJ: Prentice Hall Professional Technical Reference.

Features and Standards Supported by WebLogic Web Services - 11g Release 1 (10.3.6). (n.d.). Data de acesso: 15 de janeiro 2013, em http://docs.oracle.com/cd/E23943_01/web.1111/e13759/standards.htm#WSOVR137

Free software movement - Wikipedia, the free encyclopedia. (n.d.). Data de acesso: 7 de janeiro 2013, em http://en.wikipedia.org/wiki/Free_software_movement

IBM News room - 2012-04-02 Report: IBM Named Marketshare Leader in Middleware Software - United States. (n.d.). Data de acesso: 19 de janeiro 2013, em <http://www-03.ibm.com/press/us/en/pressrelease/37376.wss>

Introduction to BPEL. (n.d.). Data de acesso: 9 de janeiro 2013, em <http://searchoracle.techtarget.com/tip/Introduction-to-BPEL>

JBoss Enterprise SOA Platform Supported Configurations - Red Hat Customer Portal. (n.d.). Data de acesso: 10 de janeiro 2013, em <https://access.redhat.com/knowledge/articles/111863>

Microsoft BizTalk Server Product Overview. (n.d.). Data de acesso: 5 de janeiro 2013, em <http://www.microsoft.com/biztalk/en/us/overview.aspx>

Oracle Fusion Middleware 11g (11.1.1.6) Documentation Library. (n.d.). Data de acesso: 5 de janeiro 2013, em http://docs.oracle.com/cd/E23943_01/index.htm

Oracle's Support for Open Source and Open Standards. (n.d.). Data de acesso: 6 de janeiro 2013, em <http://www.oracle.com/us/technologies/open-source/index.htm>

Petals Enterprise Service Bus - Petals ESB 4.1 - Petalslink Documentation. (n.d.). Data de acesso: 7 de janeiro 2013, em <https://doc.petalslink.com/display/petalsesb41/Petals+Enterprise+Service+Bus#PetalsEnterpriseServiceBus-Softwarecomponents>

Platforms and Technologies Compatible with Mule ESB - Mule User Guide - Mulesoft.org Documentation. (n.d.). Data de acesso: 7 de janeiro 2013, em <http://www.mulesoft.org/documentation/display/current/Platforms+and+Technologies+Compatible+with+Mule+ESB>

Red Hat | Ampersand Creates Robust Customer Loyalty Program with Red Hat Solutions. (n.d.). Data de acesso: 11 de fevereiro 2013, em <http://www.redhat.com/resourcelibrary/case-studies/ampersand-creates-robust-customer-loyalty-program-with-red-hat-solutions>

Red Hat | Stater Further Increases Stability and Performance of Mission-Critical Applications

with JBoss Enterprise Appl. (n.d.). Data de acesso: 11 de fevereiro 2013, em

<http://www.redhat.com/resourcelibrary/case-studies/stater-further-increases-stability-with-jboss>

SAP NetWeaver 7.3 – SAP Help Portal Page. (n.d.). Data de acesso: 19 de janeiro 2013, em

<http://help.sap.com/nw73>

Schmelzer, R. (2010, February 25). Open Source Solutions for SOA: Check Your Bias at the Door | ZapThink. Data de acesso: 10 de janeiro 2013, em

<http://www.zapthink.com/2010/02/25/open-source-soa/>

Supported Web Service Standards - Mule User Guide - Mulesoft.org Documentation. (n.d.).

Data de acesso: 15 de janeiro 2013, em

<http://www.mulesoft.org/documentation/display/current/Supported+Web+Service+Standards>

Technology Standards | SCN. (n.d.). Data de acesso: 12 de janeiro 2013, em

<http://scn.sap.com/docs/DOC-8631>

Tibco Activematrix 3.2.0 Release Notes. (n.d.). Data de acesso: 4 de janeiro 2013, em

WebSphere Application Server Version 8.5 Information Center. (n.d.). Data de acesso: 15 de

janeiro 2013, em http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.wlp.doc%2Fae%2Fwlp_prog_model_support.html

Williams, J., Clegg, P., & Dulaney, E. (2005, May 6). The Advantages of Adopting Open

Source Software | Open Source Advantages | InformIT. Data de acesso: 17 de janeiro 2013, em <http://www.informit.com/articles/article.aspx?p=376255>

Wilson, J. A. J. (20 de fevereiro 2007). Benefits of open source code. Data de acesso: 3 de

janeiro 2013, em <http://www.oss-watch.ac.uk/resources/whoneedssource>

Windows Communication Foundation. (n.d.). Data de acesso: 15 de janeiro 2013, em

<http://msdn.microsoft.com/en-us/library/dd456779.aspx>

WSO2 Carbon Documentation - WSO2 Carbon 4.0.3 - WSO2 Documentation. (n.d.). Data de

acesso: 9 de janeiro 2013, em

<http://docs.wso2.org/wiki/display/Carbon403/WSO2+Carbon+Documentation>

Apêndice

Tabela de comparação de standards SOA:

Standards:	Descrição
Standards Serviços Web:	
JSR 109/921 (Implementing)	Modelo de programação para implementação de serviços web
JSR 181 (metadata)	Modelo de programação para metadata de serviços web
APIs para serviços Web:	
JAX-WS	API para serviços web SOAP que estende a fundação estabelecida pelo modelo JAX-RPC
JAX-RS	API para serviços REST
Frameworks de composição:	
OSGI	Framework de composição
SCA	Framework de composição
Outros frameworks prop/oss	Axis/Axis2, CXF, Metro, Spring, JBI , WCF, Fractals, JBossWS
LStandards XML:	
XSLT	Linguagem de transformação XML
XQuery	Linguagem de consulta XML
XPath	Linguagem de consulta XML
Protocolos de transporte:	
HTTP/HTTPS	Protocolos de transporte
JMS	Standard de comunicação para componentes baseados em Java EE.
Outros ()	SSL, TLS, HTTPS – protocolos seguros
Messaging:	
SOAP	Transporte tradicional de serviços web
SOAP w/ attachments/SAAJ	Extensibilidade com attachments
SOAP MTOM	Otimização (Message transmission optimization mechanism)
Description:	
UDDI	Universal description, discovery and integration
WSDL	Web service description language
JAXR	API Java para acesso de registos XML como UDDI e ebXML
Interoperability:	
Basic Profile	Perfil de interoperabilidade
Basic Security Profile	Perfil de segurança
Security/Reliable communication:	
WS-Security	Especificações relevantes à segurança
WS-SecurityPolicy	Especificações relevantes à segurança
WS-Policy	Especificações relevantes à publicação de políticas e requisitos

WS-Addressing	Especificações relevantes ao endereçamento
WS-ReliableMessaging	Especificações relevantes à comunicação
WS-Trust	Especificações relevantes à segurança
Data Bindings:	
JAXB	Especificação para manipular esquemas XML para JAX-WS
SDO	Especificação de Service Data Objects
JMS	API Java de mensagens
XMLBeans	Análogo ao JAXB para JAX-RS
Transacções atômicas:	
WS-AtomicTransactions	Conjunto de protocolos para execução de serviços atômicos, funciona dentro do framework WS-Coordination
WS-Coordination	Framework extensível para protocolos de coordenação
Tecnologias Java:	
JEE	Plataforma de programação em Java para servidores de aplicação. (EE designa enterprise edition, SE – standard edition)
Servlet	Classe Java em Java EE que se conforma às especificações da API Servlet, usada em interações com os servidores.
JSP	Tecnologia utilizada no desenvolvimento de aplicações para Web semelhante às tecnologias ASP ou PHP. Uma página criada com a tecnologia JSP, após instalada em um servidor de aplicação compatível com Java EE, é transformada em um Servlet Java
JSF	Framework MVC de aplicações Web baseado em Java que se destina a simplificar o desenvolvimento de interfaces de usuário baseadas em web. 2.0
EJB	Arquitetura de componentes para construção modular de aplicações empresariais. A especificação EJB é dos vários APIs na especificação Java EE. Encapsula a lógica de negócios de uma aplicação.
JDBC	Conjunto de classes e APIs para instruções SQL a bases de dados
JPA	Padrão de persistência Java