

ESB

Les meilleures solutions
open source



The word "SMILE" is written in a bold, white, sans-serif font. The letter "I" is replaced by a red smiley face. The letter "M" has a red heart shape above it. The letter "L" has a red speech bubble above it. The word is surrounded by various red decorative elements: stars, circles, a rocket, and a heart.

I.T IS OPEN

I - PREAMBULE

I.1 SMILE

Smile est une **société d'ingénieurs experts** dans la mise en œuvre de **solutions open source** et l'intégration de systèmes appuyés sur l'open source. Smile est membre de l'**APRIL**, l'association pour la promotion et la défense du logiciel libre, du **PLOSS** – le réseau des entreprises du Logiciel Libre en Ile-de-France et du **CNLL** – le conseil national du logiciel libre.

Smile compte plus de 1200 collaborateurs dans le monde ce qui en fait le premier intégrateur français et européen de solutions open source.

Depuis 2000 environ, **Smile mène une action active de veille technologique** qui lui permet de découvrir les produits les plus prometteurs de l'open source, de les qualifier et de les évaluer, de manière à proposer à ses clients les produits les plus aboutis, les plus robustes et les plus pérennes.

Cette démarche a donné lieu à **toute une gamme de livres blancs** couvrant différents domaines d'application. La gestion de contenus (2004), les portails (2005), la business intelligence (2006), la virtualisation (2007), la gestion électronique de documents (2008), les PGIs/ERPs (2008), les VPN open source (2009), les Firewall et Contrôle de flux (2009), les Middleware orientés messages (2009), l'e-commerce et les Réseaux Sociaux d'Entreprise (2010), le Guide de l'open source et NoSQL (2011), Mobile et Recensement et audit (2012) et plus récemment Big Data et ERP open source pour l'e-commerce (2014). Chacun de **ces ouvrages présente une sélection des meilleures solutions open source** dans le domaine considéré, leurs qualités respectives, ainsi que des retours d'expérience opérationnels.

Au fur et à mesure que des solutions open source solides gagnent de nouveaux domaines, Smile sera présent pour proposer à ses clients d'en bénéficier sans risque. Smile apparaît dans le paysage informatique français comme **le prestataire intégrateur de choix pour accompagner** les plus grandes entreprises dans l'adoption des meilleures solutions open source.

Ces dernières années, Smile a également étendu la gamme des services proposés. Depuis 2005, un département consulting accompagne nos clients, tant dans les phases d'avant-projet, en recherche de solutions, qu'en accompagnement de projet. Depuis 2000, Smile dispose d'un studio graphique, devenu en 2007 Smile Digital – agence interactive, proposant outre la création graphique, une expertise e-marketing, éditoriale, et interfaces riches. Smile dispose aussi d'une agence spécialisée dans la TMA (support et l'exploitation des applications) et d'un centre de formation complet, Smile Training. **Enfin, Smile est implanté à Paris, Lille, Lyon, Grenoble, Nantes, Bordeaux, Marseille et Montpellier. Et présent également en Suisse, au Benelux, en Ukraine, en Russie, au Maroc et en Côte d'Ivoire.**

1.2 QUELQUES REFERENCES DE SMILE

SMILE est fier d'avoir contribué, au fil des années, aux plus grandes réalisations Web françaises et européennes. Vous trouverez ci-dessous quelques clients nous ayant adressé leur confiance.

Web

EMI Music, Salon de l'Agriculture, Mazars, Areva, Société Générale, Gîtes de France, Groupama, Eco-Emballage, CFnews, CEA, Prisma Pub, Veolia, JCDecaux, 01 Informatique, Spie, PSA, Boiron, Larousse, Dassault-Systèmes, Action Contre la Faim, BNP Paribas, Air Pays de Loire, Forum des Images, IFP, BHV, Gallimard, Cheval Mag, Afssaps, Bénéteau, Carrefour, AG2R La Mondiale, Groupe Bayard, Association de la Prévention Routière, Secours Catholique, Canson, Veolia, Bouygues Telecom, CNIL, Danone, Total, Crédit Agricole...

E-Business

Krys, La Halle, The North Face, Kipling, Vans, Pepe Jeans, Hackett, Minelli, Un Jour Ailleurs, Decitre, ANWB, Solaris, Gibert Joseph, De Dietrich, Macif, Figaroclassifieds, Furet du Nord, Gîtes de France, Camif Collectivité, GPdis, Projectif, ETS, Bain & Spa, Yves Rocher, Bouygues Immobilier, Nestlé, Stanhome, AVF Périmédical, CCI, Snowleader, Darjeeling, Cultura, Belambra...

Collaboratif

HEC, Bouygues Telecom, Prisma, Veolia, Arjowiggins, INA, Primagaz, Croix Rouge, Eurosport, Invivo, Faceo, Château de Versailles, Eurosport, Ipsos, VSC Technologies, Sanef, Explorimmo, Bureau Veritas, Région Centre, Dassault Systèmes, Fondation d'Auteuil, Gaz Electricité de Grenoble, Ville de Niort, Ministère de la Culture, PagesJaunes Annonces, Primagaz, UCFF, Apave, Géoservices, Renault F1 Team, INRIA, CIDJ, SNCD, CS informatique, Serimax, Veolia Propreté, Netasq, Corep, Packetis, Alstom Power Services, Mazars, COFRAC, Assemblée Nationale, DGAC, HEC...

Systèmes d'Information Métiers

Veolia Transport, Solucom, Casden Banque Populaire, La Poste, Christian Louboutin, PubAudit, Effia Transport, France 24, Publicis, Nouvelles Frontières, Jus de Fruits de Mooréa, Espace Loggia, Bureau Veritas, Skyrock, Lafarge, Cadremploi, Groupe Vinci, IEDOM, Carrefour, Corsair, Le Bon Coin, Jardiland, Trésorerie Générale du Maroc, Ville de Genève, ESCP, Faiveley Transport, INRA, Yves Rocher, ETS, Perouse Médical, Auchan ecommerce, Viapresse, Danone...

Infrastructure

Agence Nationale pour les Chèques Vacances, Pierre Audoin Consultants, Rexel, Motor Presse, OSEO, Sport24, Eco-Emballage, Institut Mutualiste Montsouris, Ionis, Osmoz, SIDEL, Atel Hotels, Cadremploi, Institut Français du Pétrole, Mutualité Française, Bouygues Telecom, Total, Ministère de l'écologie, Orange, Carrefour, Jardiland, Kantar, Coyote, France Televisions, Radio France...

Consultez nos références, en ligne, à l'adresse : <http://www.smile.fr/clients>.

II - SOMMAIRE	
I - PREAMBULE	1
I.1 SMILE	1
I.2 QUELQUES REFERENCES DE SMILE	2
II - SOMMAIRE	4
III - EN RESUME	6
IV - CE LIVRE BLANC	7
V - ARCHITECTURE	8
V.1 EVOLUTION DES SYSTEMES D'INFORMATION	8
V.1.a Mainframe => monolithique, peu évolutif, coût élevé	8
V.1.b Client lourd /serveur => silo applicatif, redondance, utilisation complexe	8
V.1.c Clients web / serveurs => fragmentation	9
V.1.d Cloud computing => non maîtrise de son SI	9
V.2 URBANISATION DES SYSTEMES D'INFORMATION	10
V.3 SOA: SERVICE ORIENTED ARCHITECTURE	11
V.3.a Service	12
V.3.b Contrat de service	13
VI - L'INTEGRATION	14
VI.1 LES MIDDLEWARES	14
VI.2 ENTERPRISE INTEGRATION PATTERNS OU EIP	15
VII - ESB : ENTERPRISE SERVICE BUS	17
VII.1 ORIGINES	17
VII.2 EAI - ENTERPRISE APPLICATION INTEGRATION	17
VII.3 LA GENESE DES ESB	19
VII.4 DEFINITION	19
VII.5 CARACTERISTIQUES PRINCIPALES	20
VII.6 BASES TECHNIQUES	21
VII.7 LES ESB DE LA FONDATION APACHE	21
VII.8 LES NOUVEAUX COURANTS	22
VII.8.a ESB à la demande (iPaaS)	23
VII.8.b API Management	25
VIII - CAS D'UTILISATION	27
VIII.1 E-COMMERCE	27
VIII.2 EXPOSITION D'APIS	28
VIII.3 COMMUNICATION AVEC DES SERVICES EXTERNES	29
VIII.4 GERER LE LEGACY ET PALIER LES SILOS	29
IX - LES ESB OPEN SOURCE	31
IX.1 LES ESB SELECTIONNES	31

IX.2 MULE ESB	32
IX.2.a Présentation	32
IX.2.b Spécificité du produit	33
IX.2.c Détail sur le projet	34
IX.3 REDHAT JBOSS FUSE	39
IX.3.a Présentation	39
IX.3.b Spécificité du produit	40
IX.3.c Détail sur le projet	41
IX.4 TALEND ESB	46
IX.4.a Présentation	46
IX.4.b Spécificité du produit	47
IX.4.c Détails sur le projet	48
IX.5 WSO2 ESB	54
IX.5.a Présentation	54
IX.5.b Spécificités du produit	55
IX.5.c Détail sur le projet	55
IX.6 AUTRES ESB OPEN SOURCE	61
IX.6.a Open ESB	61
IX.6.b Petals ESB	61
IX.6.c RedHat Jboss ESB	62
IX.6.d Ultra ESB	62
X - SYNTHESE	63
XI - REFERENCES	64
XII - GLOSSAIRE	65
XIII - REMERCIEMENTS	66

III - EN RESUME

Le présent livre blanc a pour but premier de décrire l'état de l'art concernant les offres ESB open source disponibles sur le marché. A l'aide de notre expérience sur des projets pour nos clients et de nos expérimentations internes dans les divers domaines de l'intégration, nous avons construit ce livre blanc pour fournir un descriptif des solutions ESB les plus abouties.

Pour ce faire, nous avons construit ce document de manière didactique en reconstruisant d'abord les grandes étapes historiques de l'architecture informatique qui ont permis d'aboutir aux grands concepts d'intégration et de middleware.

Nous avons donc pris le parti de développer ce que nous pouvons appeler une longue introduction dans le but de tous nous mettre d'accord sur les termes et donc de commencer sereinement la description du concept - ou plutôt de l'ensemble des concepts - entourant l'ESB. Nous avons d'ailleurs, pour simplifier la compréhension, ajouté quelques cas d'utilisation classiques.

Au final, cela vous permettra de bénéficier d'une vision claire de l'offre ESB open source avec l'approche orientée « produits et solutions » que nous affectionnons particulièrement et qui est, rappelons le, l'objectif principal de ce livre blanc.

Nous ne visons pas l'exhaustivité mais souhaitons tout de même vous apporter la vision la plus complète à date car gardons-le à l'esprit : le marché de l'intégration évolue rapidement.

IV - CE LIVRE BLANC

Le temps c'est de l'argent ! C'est par cette simple expression que l'on pourrait commencer ce livre blanc. Depuis la création du premier réseau de transmission optique de message en 1794 par Claude Chappe, le monde n'a cessé de vouloir télécommuniquer toujours plus vite et de transmettre toujours plus de messages, toujours plus volumineux, toujours plus complexes.

Les entreprises ont suivi le mouvement de cette modernisation. Pour répondre de plus en plus rapidement à leurs clients, ils ont du s'adapter et investir dans leurs outils de communications internes.

Aujourd'hui ne déroge pas à la règle. Avec l'avènement du commerce en ligne et de la numérisation de l'ensemble des activités des entreprises, celles-ci doivent faire face à de nouveaux défis. En plus de la rapidité, les systèmes informatiques se sont complexifiés et certains peuvent compter plusieurs centaines voire milliers d'applications. Fonctionnant à l'origine indépendamment les unes des autres, elles doivent maintenant s'échanger des données ; malheureusement, bien souvent, leur conception ne leur permet pas. Elles utilisent des protocoles, des formats et ont des SLA bien différents.

Dans un premier temps les solutions ETL (Extract Transform and Load) se sont imposées. Fonctionnant par batch, elles permettent de synchroniser deux systèmes par l'export des données de l'un puis la transformation et l'import dans l'autre. Cette solution, de prime abord très intéressante, a trouvé ses limites dans le besoin d'avoir une donnée presque immédiatement disponible dans une ou plusieurs autres applications. Les systèmes d'Enterprise Application Integration (EAI) puis d'Enterprise Service Bus (ESB) sont nés au début des années 2000 pour y répondre.

Comme les autres livres blancs publiés par Smile, cet ouvrage vous présente :

- Les concepts et enjeux autour de cette technologie
- Des cas d'utilisation classiques
- Un tour d'horizon des solutions open source existantes sur le marché avec leurs atouts et leurs faiblesses

V - ARCHITECTURE

V.1 EVOLUTION DES SYSTEMES D'INFORMATION

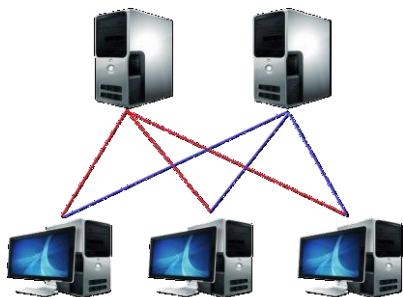
V.1.a Mainframe => monolithique, peu évolutif, coût élevé



Au début de l'informatique existait le mainframe, une boîte noire qui contenait toute l'intelligence, la mémoire et la puissance de calcul d'une entreprise. Apparus dans les années 50 et démocratisés dans les années 60-70 les mainframes sont souvent associés à IBM qui tenait près de 90 % du marché. Cette boîte était considérée à l'époque comme le saint Graal d'une entreprise moderne.

Les besoins et les capacités informatiques de l'époque étaient limités et le mainframe apportait une solution clé en main contenue dans un serveur unique. On retrouvait en son sein une application monolithique qui traitait les données, gérait leur persistance et les présentait. Qui dit serveur unique, dit défi technique, auquel il a fallu répondre dans leur conception pour assurer une haute disponibilité et une intégrité des données. Cela a eu pour conséquence des coûts de développement et donc d'acquisition très élevés pour les entreprises. En contre partie, le système avait l'avantage d'assurer une cohérence et une fiabilité importante au sein du SI.

V.1.b Client lourd /serveur => silo applicatif, redondance, utilisation complexe



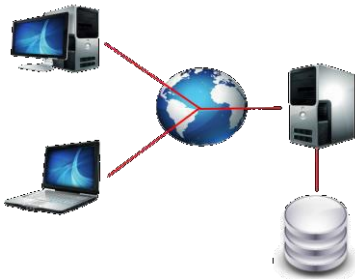
Dans les années 80 les ordinateurs individuels font leur apparition. Petit à petit les mainframes disparaissent pour laisser place à un modèle client-serveur. La NASA a ainsi éteint son dernier mainframe, IBM System Z9, en 2012. Cette nouvelle architecture est basée sur le principe que l'on découple la couche de présentation, gérée directement sur un poste client, et un serveur, conservant les traitements et la gestion de la persistance des données.

Ce découpage a permis de simplifier la conception du côté serveur et à donc eu pour conséquence une baisse des prix et donc un engouement des entreprises pour son utilisation. L'informatique, jusqu'à présent cantonnée à de très grandes entreprises ou institutions, commence à être utilisée plus largement dans les grandes et moyennes entreprises. Cet engouement provoque des complications non prévues à l'époque.

Une entreprise étant en général constituée de plusieurs métiers ayant des besoins propres, il a fallu multiplier les serveurs et les applications clientes.

L'architecture des systèmes d'information des entreprises s'est brutalement complexifiée. Des silos applicatifs et de la redondance fonctionnelle entre application sont apparus faute de gouvernance. Les couches de présentation sur les postes clients se sont multipliées et, faute d'une réflexion globale, sont extrêmement hétérogènes à l'utilisation. Cela les rend difficilement maîtrisables par les utilisateurs finaux. De nouveaux concepts et métiers sont alors apparus pour éviter l'anarchie. La gouvernance des Systèmes d'information apparaît avec notamment la notion d'urbanisation du SI, ou d'Enterprise Architecture, dans les pays anglo-saxons.

V.1.c Clients web / serveurs => fragmentation

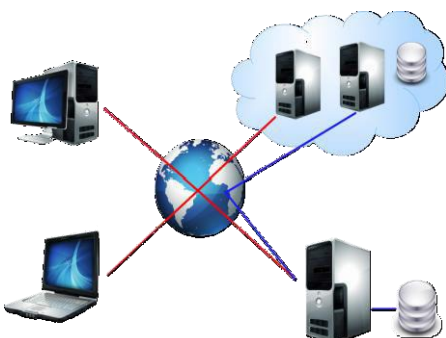


A la fin des années 90, avec le développement des réseaux d'entreprise et de l'internet, apparaît une nouvelle famille applicative : les applications web, plus légères que leurs consœurs car basées sur un outil commun, le navigateur web. Celles-ci sont en général uniquement composées de fichiers de paramètres de présentation et d'interaction avec les données (HTML, CSS, JavaScript).

C'est dans un premier temps un petit retour en arrière car le traitement est de nouveau entièrement supporté par des serveurs d'applications. Cependant, ce retour apporte des gains en termes de flexibilité, de simplicité et réduit donc les coûts de la partie client.

Accompagnant cette révolution, les architectures n-tiers font leur apparition. Elles décomposent le système en modules dédiés à une tâche particulière. La plus utilisée est probablement l'architecture 3-tiers : présentation, métier, accès/stockage des données. Il est ainsi plus facile de maintenir et faire évoluer les différentes briques applicatives mais ce type d'architecture entraîne un effet pervers en augmentant la fragmentation du Système d'information pour lequel une gouvernance devient obligatoire.

V.1.d Cloud computing => non maîtrise de son SI



Durant les années 2000, avec l'essor de l'internet, de l'augmentation de la capacité et de la fiabilité des réseaux, un nouveau concept apparaît, le cloud computing. L'idée étant de sous-traiter à des tiers une partie du SI ou des applications spécialisées et de ne payer que leur réelle utilisation. Ces spécialistes mettant leur plate-forme à disposition

de plusieurs entreprises clientes, il en résulte une mutualisation des coûts de structure qui font baisser le coût global des applications.

Cette démarche, pose actuellement des questions sur la sécurisation des données, leur protection juridique et leur difficulté d'intégration avec le SI interne des entreprises. Encore une fois, cette évolution augmente la fragmentation et complexifie la gouvernance des SI par les entreprises qui de plus deviennent dépendantes de SI tiers.

V.2 URBANISATION DES SYSTEMES D'INFORMATION

Dans les années 80, l'urbanisation, jusqu'à présent cantonnée à l'organisation du territoire, est reprise par le secteur bancaire qui se retrouve confronté à des besoins de simplification et de rationalisation de leur SI comportant plusieurs milliers d'applications. Ces entreprises n'ont plus le contrôle de leur SI car il est devenu trop complexe. Elles ont cherché des approches dans d'autres secteurs d'activités qui auraient des problématiques similaires. Ce portage de l'urbanisation du territoire au système d'information est réalisé en parallèle en Europe et aux États-Unis où cette science sera appelée Enterprise Architecture.

Par extension, l'urbanisation d'un système d'information est une démarche qui consiste à anticiper et à faire évoluer le système d'information d'une entreprise dans son ensemble en fonction de :

- sa stratégie
- son organisation
- ses contraintes internes et externes (délais, coûts, législation, etc.).

Afin de garantir :

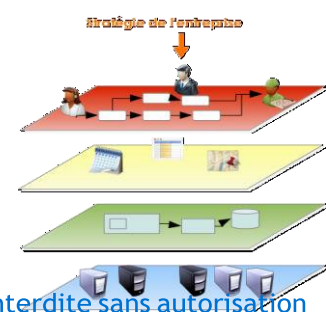
- ses objectifs métiers
- sa simplicité
- sa cohérence.

La démarche d'urbanisation va permettre de :

- se ré-appropriier son SI
- faire évoluer son SI en fonction des besoins
- harmoniser ses composants
- maîtriser l'entropie.
- réaliser des économies

Cette démarche est en général mise en œuvre dans les grandes entreprises au sein d'un cabinet d'urbanisme rattaché à la DSI ou directement à la direction de l'entreprise.

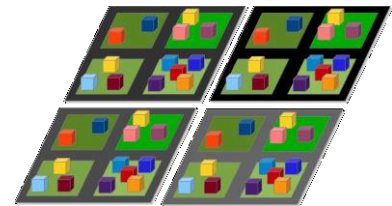
L'urbanisation est basée sur une référence à 4 niveaux de vision :



- Métier : Qui fait quoi et pourquoi ?
- Fonctionnelle : De quoi a-t-on besoin pour réaliser les processus métiers ?
- Applicative : Que va-t-on mettre à disposition ? Qui implémentera les fonctionnalités demandées ?
- Physique : Où se trouvent ces applications ?

Ces niveaux vont permettre de réaliser une cartographie précise du système d'information en utilisant les concepts de l'urbanisation des villes. On découpera ainsi le SI en :

- Zone : métier de l'entreprise
- Quartier : contenu dans une zone et ayant un type commun d'informations traitées
- Îlot : contenu dans un quartier et étant un ensemble de traitement et de données homogènes



Chacun de ces blocs ainsi défini évolue indépendamment et interagit avec les autres via des interfaces publiques et des zones d'échanges, les routes.

V.3 SOA: SERVICE ORIENTED ARCHITECTURE

Le concept de Service Oriented Architecture, ou SOA, a été inventé par Yefim V. Natis, un analyste de Gartner, qui a publié le 12 avril 1996 une note de recherche¹ sur les orientations architecturales des SI portant ce nom. Il définissait le SOA comme :

« une architecture logicielle qui commence par la définition des interfaces et l'élaboration d'une application entièrement basée sur leurs implémentations et les appels à celles-ci. »

Cette publication est restée longtemps inaperçue et ce n'est qu'en 2001 que l'on entend de nouveau parler avec l'essor des web-services. Dans les années suivantes l'OASIS group et l'Open Group reprennent le concept et s'attaquent en parallèle à sa définition concrète. Ce n'est qu'en octobre 2006 qu'une définition claire de ce qu'est le modèle SOA est officiellement publiée par l'OASIS Group². En avril 2009 L'Open Group sort sa propre définition au sein de sa publication SOA Source Book³.

SOA n'est pas une architecture mais un concept philosophique qui repose sur l'urbanisation de SI et concrétise sa vue fonctionnelle en une vision de services.

Deux éléments majeurs dans toutes les définitions qui lui sont données sont à respecter :

- le SI doit être entièrement basé sur des services (web ou non)
- chaque service doit être autonome et doit avoir un couplage faible avec ses clients

Ces objectifs sont de :

- rendre flexible le SI (ou agile) pour le faire évoluer rapidement
- qu'une fonctionnalité ait une unique implémentation qui pourra être réutilisée plusieurs fois
- réduire les coûts d'intégration en se basant sur des standards de communication et des interfaces publiques
- pouvoir créer des applications composites qui puissent utiliser plusieurs services.

Attention, SOA n'étant qu'un concept philosophique, il n'impose pas une architecture précise ou les types d'implémentations utilisables. Cela dit, si l'on suit la définition faite par certains groupes ou entreprises, certains protocoles ou patrons architecturaux sont recommandés. (SOAP, WS-*).

V.3.a Service

Un service est un composant du SI qui met à disposition un accès vers une ou plusieurs fonctions métiers.

Un service se doit d'être simple d'emploi, réutilisable et interopérable. C'est le lien entre des consommateurs se trouvant dans la couche métier et des fournisseurs qui implémentent une ou plusieurs fonctionnalités. Un service expose ces fonctionnalités en exposant des opérations via une interface publique et respecte un contrat de service.

Un service est sans état, il ne conservera aucune mémoire sur le contexte de son utilisation.

Un service doit respecter au minimum l'un de ces critères :

- mutualisation : il est utilisé par au minimum 2 clients
- facilitateur : il permet l'interopérabilité avec un autre SI
- nécessaire : il est utilisé dans une composition.

V.3.b Contrat de service

Un contrat de service est clairement défini pour chaque service mis à disposition.

Il contient :

- son ou ses adresses
- la signature des opérations (nom, structure et format du message de requête, structure et format du message de réponse)
- les protocoles disponibles
- ses garanties via un contrat de Service Level Agreement (SLA)
 - performance
 - disponibilité
 - sécurité
 - temps de rétablissement
 - pénalités de non respect
 - support mis à disposition
 - Etc.
- les directives à respecter par le client
 - nombre d'erreurs acceptables
 - sécurisation des données d'authentification
 - etc.
- des outils de mesures mis en œuvre pour vérifier les garanties et les directives.

VI - L'INTEGRATION

VI.1 LES MIDDLEWARES

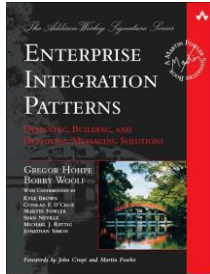
Un middleware, ou intergiciel en français, est par définition un outil qui se place entre plusieurs applications informatiques pour leur permettre de communiquer entre elles. Un middleware va ainsi permettre d'intégrer des composants dans un système plus global. Il existe 3 niveaux d'intégration implémentés par les middlewares :

- l'intégration de données, qui consiste à transmettre des données entre des composants informatiques stockant des données brutes (base de données, datawarehouse, etc)
- l'intégration applicative, qui consiste à transmettre des messages entre applications (CRM, ERP, WMS, e-commerce, etc.)
- l'intégration de processus, qui consiste à définir les flux opérationnels (processus métiers).

Ces 3 niveaux d'intégration ont donné naissance à plusieurs patrons et familles d'outils les implémentant :

- l'intégration de données :
 - Extract-Transform-Load (ETL) : comme son nom l'indique, il extrait des données d'un côté, les transforme dans un format compatible avec la deuxième source et les charge dans cette dernière.
- l'intégration applicative :
 - Message Oriented Middleware (MOM) : permet la transmission asynchrone de messages entre applications via des files d'attente
 - Enterprise Application Integration (EAI) : permet la transmission synchrone ou asynchrone de messages entre applications
 - Enterprise Application Bus (ESB) : évolution des EAIs d'une structure en hub vers une structure en bus.
- l'intégration de processus :
 - Business Process Management (BPM) : gère les processus et les relations entre les différents acteurs d'un SI qu'ils soient humains ou machines.

VI.2 ENTERPRISE INTEGRATION PATTERNS OU EIP



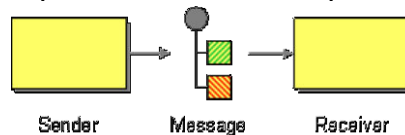
Les Enterprise Integration Patterns⁴ sont des patrons de conception rassemblés et conceptualisés par Gregor Hohpe et Bobby Woolf dans un livre publié en octobre 2003. Cette bible décrit, nomme et normalise 65 patrons d'intégration de messages.

En plus d'en fournir une liste, ce livre apporte une syntaxe et un vocabulaire propres, utilisés depuis sa création comme référence lors de l'établissement de spécifications de projets d'intégration applicative.

Les EIP modélisent tout échange d'information comme un transfert de message entre deux Endpoints par l'intermédiaire d'un channel.

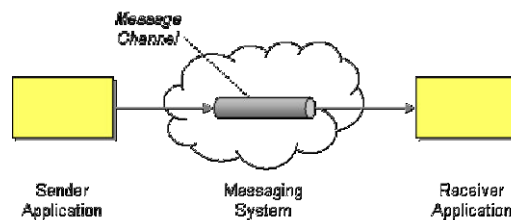
Message

Un message est composé d'une entête (header), d'un contenu, le message que l'on veut transmettre (payload) et optionnellement d'une pièce jointe.



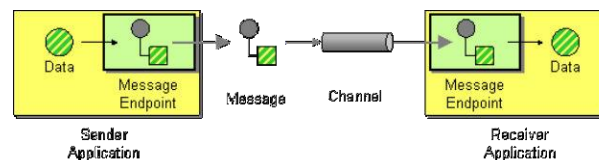
Channel

Un channel est un canal de communication entre applications dans lequel transitent des messages.



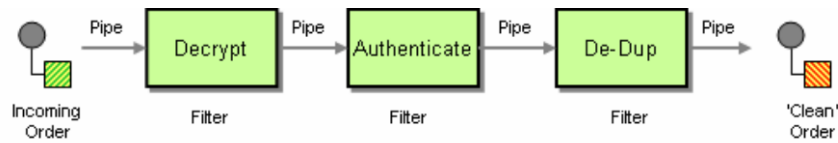
Endpoint

Un endpoint est un composant chargé de la connexion entre une application et un channel.



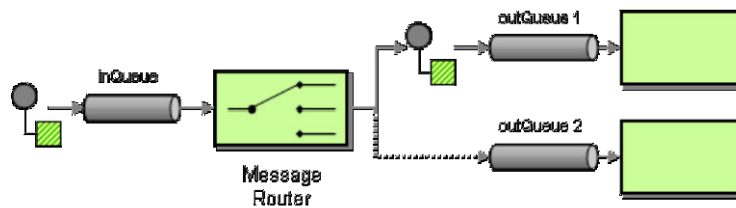
Pipes and Filters

Un filter est un processeur qui réalise une tâche unique et indépendante. Un pipe est un canal de transmission entre des filters.



Router

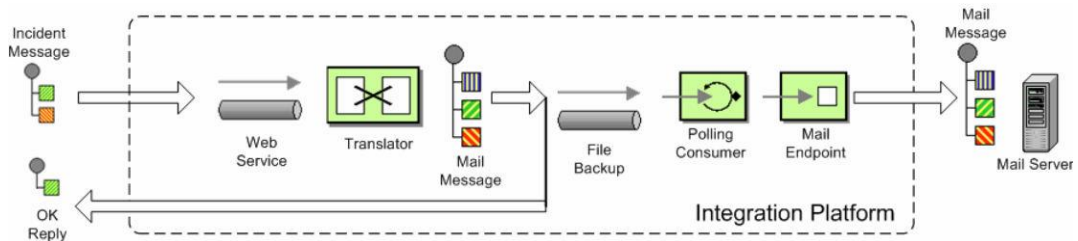
Un router est un filtre spécifique qui oriente un message en fonction d'une condition.



A l'aide de ces composants de base, il est possible de décrire exactement tous les flux d'intégration applicative.

Exemple

Une application envoie des tickets d'incident à un web-service exposé par la plateforme d'intégration qui les transforme et les transmet par mail à un administrateur.



VII - ESB : ENTERPRISE SERVICE BUS

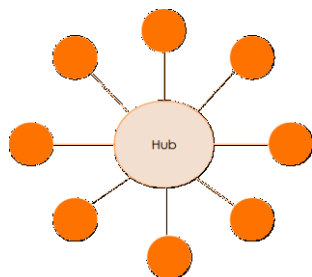
Nous avons vu dans les chapitres précédents toutes les pièces qui ont permis l'émergence des ESB. Nous allons maintenant les assembler.

VII.1 ORIGINES

Toutes les architectures successives évoquées précédemment, mainframe, client /serveur, n-tiers, cloud, ont révélé plusieurs défauts :

- Application monolithique : une application réalise l'ensemble des fonctionnalités, stocke l'ensemble des données afférentes et met à disposition une UI pour interagir avec celles-ci. Cette architecture est en général difficilement maintenable et évolutive.
- Couplage fort : un client est lié à son serveur via un protocole, un format propre, des adresses potentiellement fixées en dur. Toute modification de l'un des composants peut nécessiter une nouvelle livraison des autres pour que le système puisse continuer à fonctionner. On appelle cela souvent le « syndrome du plat de spaghetti. »
- Silo applicatif : Une ou plusieurs applications sont réalisées pour un métier donné. Elles sont capables uniquement de communiquer entre elles. Cependant, la structure d'une entreprise vit et donc évolue. Le SI sous-jacent doit évoluer également pour répondre aux attentes des utilisateurs. Les silos empêchent cette évolution et ne permettent pas de partager des informations et des fonctionnalités entre les différents métiers. Cela a pour conséquence des duplications, des problèmes de communication, des évolutions disparates et finalement une augmentation des coûts.

VII.2 EAI - ENTERPRISE APPLICATION INTEGRATION



C'est en arrivant sur ce constat simple que le concept architecturale d'Entreprise Application Integration ou EAI est apparu au début des années 2000. Ce terme a fait un énorme buzz marketing à son époque mais la fièvre est retombée rapidement... En cause, les solutions proposées par les différents éditeurs (Oracle, SAP, Tibco, WebMethods, IBM, etc.) et la jeunesse de la vision architecturale.

L'EAI, à la base un pattern architectural décrivant les principes d'intégration applicative, est devenu une famille de produits middleware appliquant ce pattern. Ces produits sont basés sur la topologie d'hub and Spokes. L'EAI, en son centre, ou hub, interagit avec l'ensemble des applications avec un connecteur propriétaire, le spoke. L'EAI est un Single Point of Failure de l'ensemble du SI.

En 2003, plusieurs études sur les projets de mise en œuvre d'un EAI révèlent que la majorité, on parle de plus de 70 %, des projets a échoué. Les principales causes observées⁵ :

- Un projet en perpétuelle évolution : un EAI est par nature une application dynamique qui doit évoluer en permanence. De nombreuses entreprises l'ont considéré comme une application comme une autre sans prendre en compte, et budgéter, une maintenance importante dans le temps.
- Expertise en EAI rare : un EAI est un composant applicatif complexe qui interagit avec de nombreux sous-systèmes et est généralement basé sur des principes et des technologies propres à chaque éditeur. Les compétences dans le domaine sont donc extrêmement restreintes.
- Des standards concurrents : A l'origine basé sur des protocoles communs, les éditeurs de solutions d'intégrations ont petit à petit ajouté leurs propres normes et spécifications pour gérer des cas particuliers. cela a rendu les différents systèmes non interopérables entre eux.
- Utiliser un EAI comme un outil : un EAI n'est pas un outil qui vit seul dans son monde. C'est un composant au cœur d'un système d'information et ce système doit être pensé et organisé pour l'utiliser de manière efficace. Une vision réductrice a des conséquences graves sur l'ensemble du SI.
- Créer des interfaces est un art : Concevoir simplement techniquement une interface de communication ne suffit pas ; il faut avoir la vision top-down de l'intégration et faire discuter ensemble les différents interlocuteurs, fournisseurs de données, consommateurs de données, utilisateurs finaux pour embrasser l'ensemble du sujet et concevoir une interface simple et surtout utile pour tous.
- Le manque de détail : Une information négligeable au départ peu se révéler extrêmement coûteuse par la suite si elle n'est pas connue et prise en compte. Un référentiel clair des interactions, de leur mode de fonctionnement et de leurs objectifs doit être défini et surtout maintenu au cours du temps.
- La responsabilité budgétaire mal définie : un EAI, étant au cœur du SI, est employé par de nombreux services et métiers de l'entreprise qui ont chacun leur propre budget. La question de la responsabilité et des coûts induits se pose donc.

VII.3 LA GENESE DES ESB

Fin 2002, Sonic, une filiale de Progress Software Corporation, propose le premier logiciel sous la terminologie Enterprise Service Bus (ESB), SonicMX. Tous les éditeurs concurrents lui emboîtent le pas les années suivantes. Vendus comme une révolution par rapport à la lourdeur et la complexité des EAI, les ESB ne sont en réalité bien souvent qu'un simple re-branding de ceux-ci. Ce n'est à partir de 2005-2006 que le terme ESB prend pleinement consistance et que les solutions éditées se démarquent ouvertement de leurs prédécesseurs, notamment poussées par des solutions émergent dans le monde Open Source.

VII.4 DEFINITION

L'ESB est comme la SOA ou les EAI une notion qui possède plusieurs définitions plus ou moins complètes selon les organismes et éditeurs de solution.

Roy Schulte de Gartner en donne l'une des premières définitions en 2003 :

« Un Enterprise Service Bus (ESB) est une nouvelle architecture qui exploite les services web, les systèmes orientés messages, le routage intelligent et la transformation. L'ESB agit comme une colonne vertébrale légère et omniprésente de l'intégration à travers laquelle les services logiciels et les composants applicatifs dialoguent »

En 2007 Anne Thome Manes de Burton Group définit l'ESB de cette manière :

« Un Enterprise Service Bus (ESB) est une solution middleware qui permet l'interopérabilité d'environnements hétérogènes utilisant un modèle orienté service (SOA). Bien que fréquemment associé avec des concepts comme l'intégration et la médiation un ESB fournit également une plate-forme de service comparable à un serveur d'application. En réalité les ESB représentent l'unification du monde de l'intégration et des serveurs d'application. La vraie avancée des ESB est sa capacité à virtualiser des services. Un conteneur de services ESB abstrait un service et l'isole de son protocole de communication, de ses méthodes d'invocations, de ses patrons d'échange de message, de ses exigences en termes de qualité de service et d'autres besoins d'infrastructure. »



Si l'on essaye de faire la synthèse de ces définitions et des produits existants sur le marché, on peut dire que les ESB sont des composants middlewares basés sur le

principe d'un bus de messages permettant la communication entre applications hétérogènes, facilitant l'implémentation des concepts de la SOA en favorisant l'utilisation de web services et de standards et en mettant en œuvre les EIP.

ESB légers et ESB lourds

Dans la littérature concernant les ESB, vous trouverez souvent la notion d'ESB légers et d'ESB lourds pour caractériser les différentes solutions existantes sur le marché. Cette différenciation très obscure est due à ce problème de définition d'un ESB. En général, on parlera d'un ESB lourd pour un ESB descendant directement des solutions d'EAI et ayant une couverture fonctionnelle très importante, mais, revers de la médaille, qui seront complexes et lourds à mettre en place. Les ESB légers, quant à eux, ont en général fait le choix de réduire leur périmètre fonctionnel et de se baser sur des architectures logicielles agiles et simples et donc, par définition, plus légères à mettre en place.

On remarquera que les solutions propriétaires sont presque exclusivement des ESB lourds et les solutions open source presque exclusivement des ESB légers.

VII.5 CARACTERISTIQUES PRINCIPALES

Les solutions ESB ont toutes plusieurs caractéristiques fondamentales qu'elles se doivent d'implémenter :

- **Connectivité et communication** : Un ESB doit permettre de communiquer avec de multiples protocoles et formats basés sur des standards reconnus par tous (SOAP, REST, XML, JSON, EDI, JMS, CORBA, etc.) et ce de manière synchrone ou asynchrone (le deuxième étant à privilégier). Pour ce faire, il doit notamment être capable d'interagir avec un Middleware Orienté Message (MOM) embarqué ou non au sein de l'ESB.
- **Routing** : Un ESB doit permettre de découpler fournisseurs et consommateurs de données. Il doit ainsi mettre à disposition un mécanisme permettant de distribuer les messages depuis un fournisseur vers un ou des consommateurs de manière transparente pour l'un et l'autre. On retrouvera ici un certain nombre d'implémentations de patrons EIP.
- **Transformation** : un fournisseur et un consommateur ne dialoguant pas forcément dans un même langage, un ESB doit mettre à disposition des moyens de transformation des messages. (cf EIP)
- **Orchestration** : l'orchestration peut être de deux ordres :
 - **Technique** : Les ESB légers n'appliquent que la partie technique en mettant en œuvre des pipes, des filtres et une forme basique de gestion de processus réduite à l'instant présent.
 - **Fonctionnelle** : Les ESB lourds incorporeront en plus l'orchestration fonctionnelle capable de gérer une longue traîne que l'on retrouvera également dans les applications de BPM.

- Outil de Surveillance :
 - Technique : Les ESB doivent permettre de surveiller l'acheminement des messages et de fournir un ensemble de métriques techniques afférentes aux flux et à la plate-forme. (Service Activity Monitoring – SAM)
 - Métier : En sus certains ESB vont permettre de réaliser un contrôle fonctionnel et métier des messages qui transitent (Business Activity Monitoring – BAM)

VII.6 BASES TECHNIQUES

Techniquement un Enterprise Service Bus est composé de :

- Un framework d'intégration, qui réalise les opérations de médiation sur les flux de messages. Il en existe plusieurs sur le marché :
 - Spring
 - Apache Camel
 - Apache Synapse...
- Un framework de services, qui permet de concevoir les web services qui pourront être utilisés par les applications du SI. Il en existe plusieurs sur le marché :
 - Apache CXF
 - Apache Axis2...
- Un système de gestion de composants, qui permet de gérer dynamiquement l'installation, le démarrage, l'arrêt, la désinstallation à chaud de composants techniques. Il en existe plusieurs sur le marché :
 - OSGi
 - SCA...
- Un serveur d'application, qui est un conteneur de contexte d'exécution pour application qui va notamment gérer toutes les connexions. Il en existe plusieurs sur le marché :
 - Apache Tomcat
 - Jboss...
- Un Middleware Orienté Message, qui permet de gérer des files d'attentes de messages dans le cadre d'échanges asynchrones. Il en existe plusieurs sur le marché :
 - Apache ActiveMQ
 - RabbitMQ
 - Apache Qpid...
- Un registre, qui permet de stocker et gérer des informations contextuelles nécessaires aux composants pour fonctionner (par exemple l'adresse d'un service)

VII.7 LES ESB DE LA FONDATION APACHE

La fondation Apache possède 2 ESB à son catalogue :

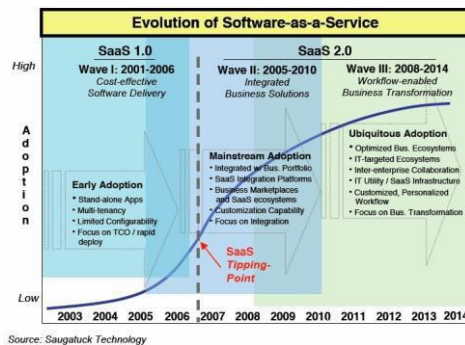
- Apache ServiceMix (2005 – LogicBlase/FuseSource)
 - framework d'intégration : Apache Camel
 - framework de service : Apache CXF
 - système de gestion de composant : OSGi → Apache Karaf
 - MOM : Apache ActiveMQ
- Apache Synapse (2006 – WSO2)
 - framework d'intégration : Apache Synapse
 - framework de service : Apache Axis2

Ces deux projets sont de très bonne qualité mais sont livrés bruts. En général on préférera donc utiliser une version packagée avec un set d'outils de développement et d'administration comme ceux que l'on retrouvera dans les solutions RedHat Jboss Fuse ESB, Talend ESB ou WSO2 ESB.

VII.8 LES NOUVEAUX COURANTS

La vision ESB est une tendance de fond qui accompagne notre société vers son souhait de plus de rapidité, plus d'interaction, plus de dématérialisation. L'ESB est devenu la base de nouvelles offres commerciales ces derniers mois pour répondre à ces besoins.

VII.8.a ESB à la demande (iPaaS)



Avec l'avènement du cloud, le besoin de toujours plus de connectivité et le souhait des entreprises d'externaliser leur infrastructure se sont démocratisés. Il existait jusqu'à présent un vide pour lier toutes ces nouvelles applications et services dans un environnement déporté. Les Infrastructures Plateform as a Service, ou iPaaS, sont là pour y combler ce vide.

Gartner en donne la définition suivante :

« Integration Platform as a Service (iPaaS) est une suite de services cloud permettant le développement, l'exécution et la gouvernance de flux d'intégration rendant possible la connexion de n'importe quelle combinaison de processus, services, applications et données locales ou dans le cloud au sein d'une organisme ou entre plusieurs organismes. »

Cloudhub



Éditeur : MuleSoft

Site internet de la solution : <http://cloudhub.io>

Date de création : 2010

En février 2010, MuleSoft est la première société à entrer sur ce marché avec sa plate-forme Cloudhub. En 2012, Cloudhub est pleinement intégré dans le cycle de release de Mule Studio et Mule ESB. Aujourd'hui, MuleSoft axe clairement son développement sur l'enrichissement de cette plate-forme.

Cloudhub met à disposition sa plate-forme au travers de 3 offres et donne la possibilité de tester ses possibilités durant 30 jours.

Pour utiliser pleinement les capacités de cette plate-forme il est recommandé de lui adjoindre les services d'Amazon SQS pour la partie MOM.

OpenShift



Éditeur : Red Hat

Site internet de la solution : <https://www.openshift.com/>

Site de la version Open Source : <http://openshift.github.io/>

Date de création : 2011

OpenShift est la plate-forme PaaS de RedHat et est pleinement opérationnelle depuis juin 2013 avec :

- OpenShift Origin : projet open source
- OpenShift Online : Public Cloud
- OpenShift Enterprise : Cloud privé

Cette plate-forme repose en version publique sur Amazon AWS avec RHEL et SeLinux. La version open source et la version Entreprise peuvent également reposer sur l'iaaS OpenStack. La spécificité de ce PaaS est qu'il n'est pas centré sur une technologie ou un langage. Vous pouvez aussi bien y déployer du Java, du Ruby, du PHP ou du NodeJS.

OpenShift met à disposition des Cartridges contenant une technologie ou un framework installable par un simple clic. L'ensemble des produits de Jboss, et donc également de Fuse, y sont disponibles.

Le Cartridge Fuse ESB existe depuis début 2014. Il est uniquement distribué en version Alpha et évolue régulièrement. Malgré son jeune âge, le produit est déjà extrêmement complet et stable.

WSO2 Cloud/Stratos



Éditeur : WSO2

Site internet de la solution : <http://wso2.com/cloud/>

Site de la version Open Source :

<http://wso2.com/cloud/private-paas/>

Date de création : 2011

Après avoir publié une première version de Stratos en décembre 2010, WSO2 sort en juillet 2011 sa plate-forme SaaS StratosLive qui permet d'exploiter dans le cloud l'ensemble des produits de la stack WSO2. En juin 2013, WSO2 reverse à la fondation Apache Stratos, renomme son service en WSO2 Cloud et revoit toute sa stratégie dans ce domaine.

WSO2 se base sur OpenStack pour fournir ses services et met à disposition une AMI Amazon EC2. WSO2 Cloud est en réalité 3 offres :

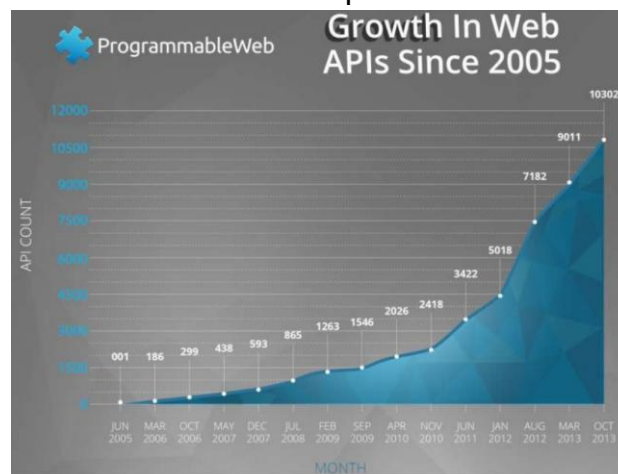
- WSO2 Public Cloud : une infrastructure cloud partagée entre utilisateurs. Actuellement seul deux produits sont disponibles mais cette offre devrait s'étendre dans le futur avec App Cloud, API Cloud

- WSO2 Managed Cloud : une infrastructure dédiée mais gérée par WSO2
- WO2 Private Cloud : une plate-forme cloud pouvant être installée et gérée entièrement par le client.

Cette plate-forme n'est pas encore considérée comme « production ready ».

VII.8.b API Management

Le 7 février 2000 Salesforce lance la première API publique et très rapidement d'autres acteurs lui emboîtent le pas comme Ebay. Depuis 2010, nous assistons à une véritable explosion du nombre de services exposés sur le web.

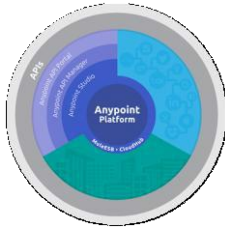


Depuis peu, une véritable monétisation des APIs prend forme et ce développement nécessite de nouveaux outils pour répondre aux défis de ce marché. Ainsi, en novembre 2006, Mashery lance le premier service de gestion d'API. Depuis, de nombreux concurrents sont entrés sur le marché comme 3scale, Apigee, Layer 7 ou Vordel (Axway). Le monde de l'Open source a suivi le courant et s'attaque aux marchés depuis plusieurs mois.

Un API Manager est concrètement une solution permettant :

- d'exposer des APIs (en REST et/ou en SOAP)
- des transformations légères et d'appliquer des politiques
- de gérer le cycle de vie des services
- de présenter, documenter et/ou vendre des APIs
- de surveiller les usages des APIs

Anypoint Platform for API



Éditeur : MuleSoft

Site internet de la solution : <http://cloudhub.io>

Date de création : 2013

MuleSoft a lancé sa Plateforme Anypoint for APIs courant 2013. Disponible, pour le moment, uniquement sur sa plate-forme cloud, cette offre, utilisant pour cœur son ESB, s'est enrichie d'un véritable gestionnaire d'API, d'outils complétant son studio et d'un générateur de portail.

Mulesoft ne s'est pas contenté d'apporter des outils mais a souhaité réellement mettre sa touche personnelle dans le domaine. Mulesoft a ainsi racheté en 2013 le site ProgrammableWeb qui est la référence de l'information sur le sujet, et s'est investi dans la création d'un nouveau langage de description des APIs REST, le RAML.

Pour le moment, la plate-forme ne gère officiellement que des APIs REST.

Il est à noter qu'en plus d'investir dans le domaine, MuleSoft a su s'attirer l'attention de grands noms du domaine comme Salesforce qui y a investi plusieurs millions de dollars en 2013.

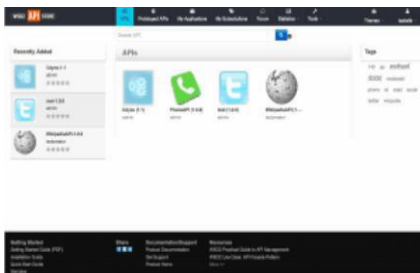
WSO2 API Manager



Éditeur : WSO2

Site internet de la solution : <http://wso2.com/>

Date de création : 2012



WSO2 API Manager, sorti en 2012, a été la première solution open source sur le marché. Basée sur son ESB, utilisant son service d'identification WSO2 Identity Server et son service de monitoring WSO2 Business Activity Monitor, WSO2 API Manager est une solution complète et en phase avec le marché. Cette solution est disponible à la fois en « on premise » et dans le cloud.

WSO2 API Manager a été reconnu en 2013 par Gartner comme une solution innovante sur le marché. Comme tous les produits de WSO2, celui-ci est sous licence Apache 2.0.

VIII - CAS D'UTILISATION

Nous avons vu précédemment les différents concepts liés aux problématiques d'échanges de données au sein du SI et ce qu'était un ESB. Mais concrètement à quoi ça sert un ESB ? Dans quels cadres les utilise-t-on ?

VIII.1 E-COMMERCE

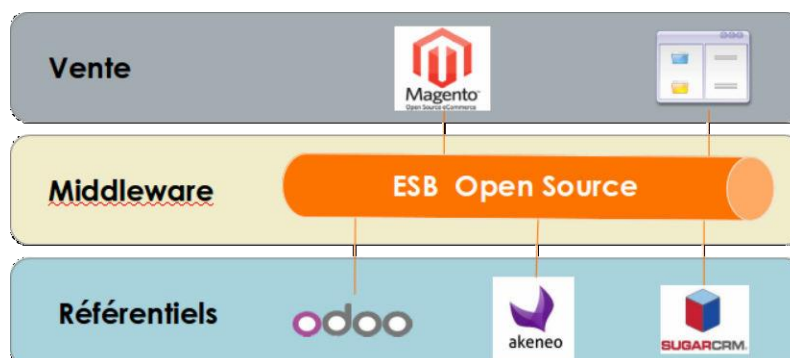
Imaginons une entreprise spécialisée dans le chocolat qui souhaite se développer dans le commerce porteur de machine à faire du chocolat chaud et de la vente de dosettes.

L'entreprise a choisi de focaliser essentiellement son effort commercial sur la mise en place d'une plate-forme de commerce électronique et l'ouverture de magasins étendards dans les capitales européennes. Le client final pourra ainsi passer sa commande à la fois depuis le site e-commerce de la marque ou acheter directement ses dosettes dans un magasin.

Les principaux enjeux de la plate-forme seront :

- le partage de référentiels communs
- la rapidité des échanges
- la capacité du SI à tenir la charge

Le prestataire chargé de la conception de cette architecture, par exemple Smile, propose à l'entreprise de mettre en place l'architecture technique suivante :



Pour la vente le prestataire recommande de créer un site e-commerce basé sur Magento et un client lourd en java pour les magasins.

Pour la gestion des référentiels le prestataire recommande l'utilisation de :

- la PIM Akeneo pour gérer le référentiel produit
- SugarCRM pour gérer le référentiel client
- Odoo pour gérer la facturation.

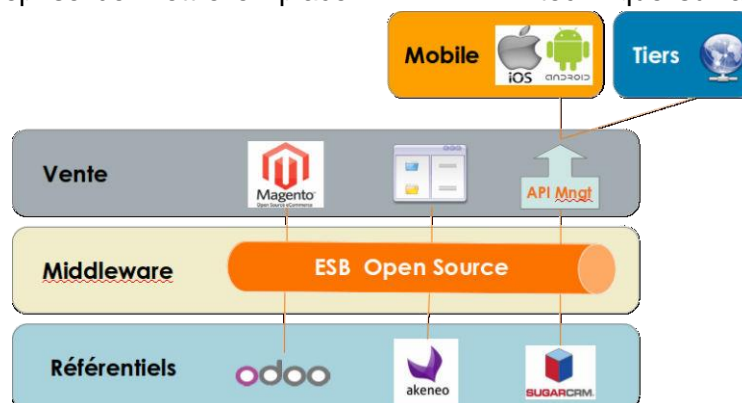
Entre ces deux couches, pour fiabiliser, découpler et organiser les échanges, le prestataire recommande la mise en place d'un ESB.

VIII.2 EXPOSITION D'APIS

Notre entreprise spécialisée en chocolat souhaite développer ses ventes en se dotant de canaux supplémentaires :

- une application mobile dédiée à la marque pour Smartphone et tablette
- en confiant à des sociétés tierces la possibilité de vendre ses produits

Le prestataire chargé de la conception de cette architecture, par exemple Smile, propose à l'entreprise de mettre en place l'architecture technique suivante :

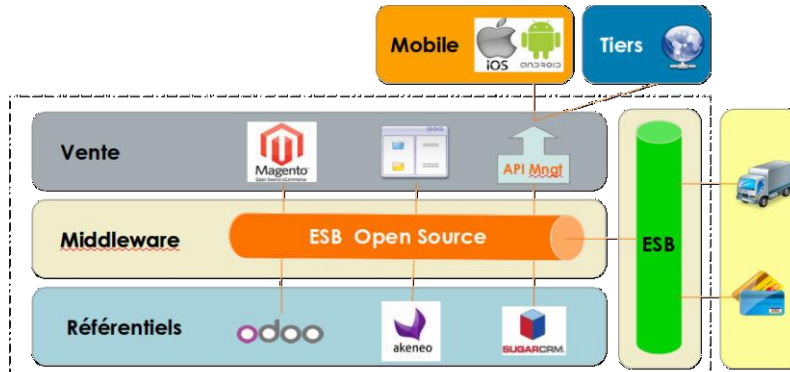


Le prestataire recommande d'ajouter à l'architecture existante un gestionnaire d'API, basé sur un ESB, pour exposer vers l'extérieur un ensemble de service REST qui pourront être utilisés aussi bien par les applications mobiles de l'entreprise que par des prestataires externes. Le gestionnaire d'API s'occupera de la sécurisation des flux avec l'extérieur, de leur retransmission vers le SI interne et de la documentation des services exposées.

VIII.3 COMMUNICATION AVEC DES SERVICES EXTERNES

Notre entreprise fait appel à des entreprises externes pour la logistique et les moyens de paiements. Elle souhaite donc que l'architecture proposée intègre la possibilité de communiquer avec des services extérieurs.

Le prestataire chargé de la conception de cette architecture, propose à l'entreprise de mettre en place l'architecture technique suivante :

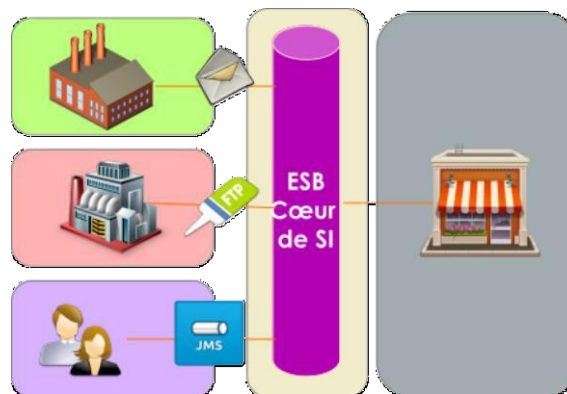


Le prestataire recommande l'ajout d'un deuxième ESB dédié aux échanges vers l'extérieur du SI. Celui-ci a pour but d'isoler dans un composant unique tous les échanges à risque pour le SI.

VIII.4 GERER LE LEGACY ET PALIER LES SILOS

L'entreprise chocolatière existe depuis 60 ans et possède déjà un SI mais celui-ci souffre du temps. Son SI a été construit au fur et à mesure de sa croissance organique et de ses acquisitions. Ainsi, toute la partie fabrication a été rachetée à deux autres entreprises ayant leur propre SI. De même, les commerciaux s'adressant aux grands comptes utilisent leurs propres applications, elles-mêmes détachées du reste du SI. Le Système d'information global de l'entreprise est ainsi actuellement constitué de silos qu'il va falloir « casser » pour la bonne marche de l'entreprise.

Reprendre une intégration de zéro serait trop dangereux et avec un ROI faible pour l'entreprise sur quelques années. Le prestataire propose une architecture souple qui permettra de connecter ces SI entre eux et qu'ils puissent évoluer en relative indépendance :



Le prestataire propose d'ajouter un ESB dont le but sera de fédérer les différents silos pour qu'ils puissent communiquer ensemble. Le prestataire aurait pu faire le choix de réutiliser l'ESB mis en place pour la nouvelle activité de l'entreprise mais afin de respecter les bonnes pratiques de l'urbanisation et afin de découpler les responsabilités, il a préféré en ajouter un nouveau. Ce nouvel ESB « cœur de SI » communiquera avec :

- la nouvelle activité via des services SOAP et REST
- la première usine via des e-mails
- la seconde usine via une dépose de fichiers sur son FTP
- les applications dédiées aux commerciaux via des files d'attentes JMS

IX - LES ESB OPEN SOURCE

IX.1 LES ESB SELECTIONNES

Nous avons sélectionné les 4 solutions qui nous semblent les plus crédibles, les plus solides et les plus pérennes aujourd'hui, celles sur lesquelles on peut envisager sans risque de construire une architecture critique pour une entreprise.

Les outils sélectionnés ne se différencient pas tant par la liste des fonctionnalités, qui pour l'essentiel découle des EIP. Ils se distinguent en revanche par les possibilités d'interfaçage, par des fonctionnalités avancées ou leur possibilité d'utilisation dans le cloud. Ils se distinguent aussi par leur dynamique de développement et leur modèle économique.

Les produits, sélectionnés par ordre alphabétique, sont les suivants:

- Mule ESB
- RedHat Jboss Fuse
- Talend ESB
- WSO2 ESB

IX.2 MULE ESB



Éditeur : MuleSoft
Site internet de la solution :
<http://www.mulesoft.com/>
Date de création : 2003
Licence : CPAL / commerciale
Version étudiée : 3.5.0

IX.2.a Présentation

La société

MuleSoft a été fondée par Ross Mason et Dave Rosenberg en 2006 pour développer et commercialiser leur produit phare Mule ESB.

La société est basée aux États-Unis et possède des bureaux aux Royaume-Uni, en Argentine et en Australie. Elle dispose également d'une structure commerciale en France.

MuleSoft a vu son actionnariat fortement évoluer ces dernières années. Les plus grands fonds technologiques y ont investi et dernièrement, SAP, Salesforce et Cisco ont souhaité participer directement au projet.

La société qui avait pour credo la facilité d'intégration applicative depuis son lancement a choisi de se focaliser et d'investir depuis peu dans le cloud grâce à sa plate-forme étendard AnyPoint. Elle a renforcé son pôle en achetant ProgrammableWeb en avril 2013.

Modèle économique

MuleSoft nous propose la base de son ESB en Open Source et vend, sous licence commerciale, une version améliorée. La tarification de celle-ci est basée sur le nombre de cores utilisés et le niveau de support.

MuleSoft a une stratégie de rémunération basée sur l'abonnement à ses services et en particulier ses services clouds.

La société se rémunère également en fournissant des prestations d'expertises techniques et de formation autour de ses produits.

Les centres de support de MuleSoft sont basés à Buenos-Aires (Argentine) et San-Francisco (États-Unis). Un nouveau centre devrait voir le jour à Londres dans les prochains mois.

Le produit

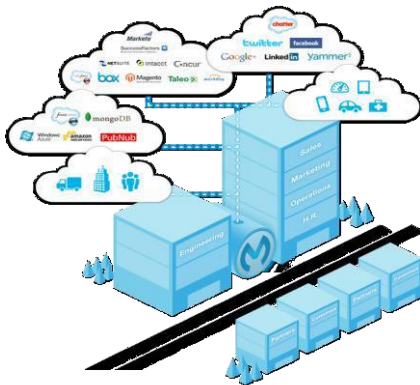
Ross Mason a commencé à travailler sur le projet Mule ESB en 2003. À l'époque, il n'existait pas d'outil simple d'intégration et encore moins de produits open source. La vue des produits propriétaires sur le marché, lourds et complexes, lui donna l'idée de repartir sur des bases saines où le développeur pourrait intégrer des applications sans devoir réinventer la roue.

Ross Masson et son équipe ont créé un outil spécifique et très éloigné techniquement de la concurrence.

Mule ESB est devenue au fil des années l'une des solutions les plus utilisées dans le domaine et peut s'enorgueillir de posséder plus de 3500 clients de part le monde dont 35 % du Global 500.

IX.2.b Spécificité du produit

Anypoint Platform



Ross Mason a annoncé en avril 2013 la création de la plate-forme Anypoint⁶. Cette plate-forme a été calibrée pour répondre à tous les besoins de l'intégration de systèmes legacy mais également avec les nouveaux enjeux du domaine. En particulier, Ross souhaite mettre l'accent sur ce qu'il appelle la « nouvelle entreprise » basée sur un système d'information très fragmenté qui emploie de manière importante toutes les possibilités des offres SaaS ou du Cloud au sens général.

MuleSoft dans son souci d'être toujours simple et léger n'a pas souhaité mettre en place une plate-forme lourde avec une suite logicielle complètement intégrée basée sur un socle commun. Elle est constituée de l'ensemble des solutions de MuleSoft, Mule ESB, CloudHub, Anypoint API Manager, Anypoint Registry, etc. Chaque produit pouvant être utilisé en standalone ou combiné avec les autres.

IX.2.c Détail sur le projet

Architecture du produit

« OSGi ? No Thanks » par cette simple phrase que Ross Mason a prononcée en 2007 on comprend que Mule a emprunté un chemin contraire à tous ses concurrents. Mule, pour sa version 3, a revu l'intégralité de l'architecture de sa solution. Ses développeurs et architectes se sont penchés sur OSGi, l'ont testé et l'on trouvé bien conceptuellement pour un vendeur de solution mais malheureusement beaucoup trop complexe pour un développeur souhaitant intégrer une solution basée dessus.

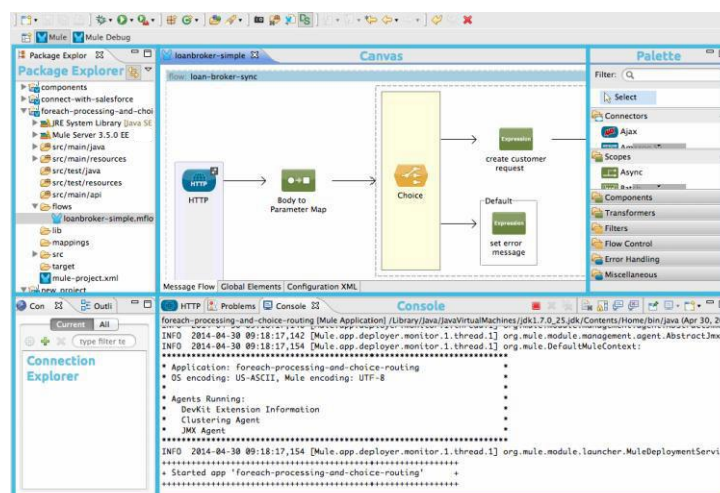
Mule a donc créé son propre système de gestion dynamique de module en évitant les écueils du modèle de classloader JEE. Cela lui permet de simplifier ses développements et l'entropie de son application. Mais pour le futur on-t-il fait le bon choix en ne partant pas sur OSGi ?

Outil de développement



MuleSoft met à disposition depuis 2012 un studio de développement, AnyPoint Studio, basé sur Eclipse. Ce studio a pour particularité de faciliter le développement et le déploiement sur Mule ESB et Cloudhub en utilisant une interface unique.

Mule a souhaité mettre l'accent sur la simplicité d'utilisation et une interface claire et très graphique. L'usage du studio est assez intuitif et une aide contextuelle est disponible pour comprendre l'usage et la configuration de chaque composant.

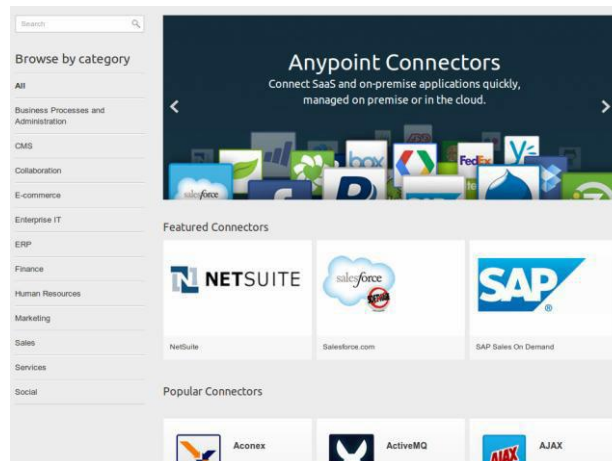


Anypoint Studio est livré avec un runtime intégré pour pouvoir tester ses développements facilement et, par un simple clic, permet de déployer vers un mule ESB en standalone et sur Cloudhub ou d'exporter dans une archive.

MuleSoft nous offre donc un studio bien conçu : simple, efficace et ergonomique.

Web-services et connecteurs

C'est l'une des grandes forces de Mule ESB. Mule fournit aux développeurs un ensemble de connecteurs spécialisés pour des services aussi variés que Twitter, Facebook ou Microsoft Dynamics. Certains de ces connecteurs ont également la particularité d'être certifiés par le service appelé (comme SAP) ou issus de partenariat (comme pour Salesforce).



MuleSoft n'oublie pas les autres services et permet de créer son propre connecteur ou d'utiliser tous les protocoles ou formats standards, SOAP, REST, FTP, POP3, JMS, etc.

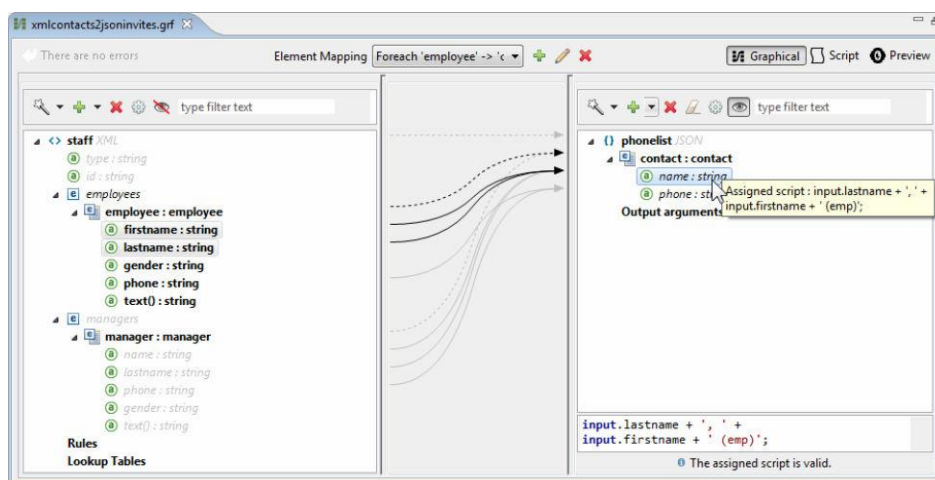
Mule ESB apporte une bibliothèque de connecteurs vers les services les plus courants qui facilitent la vie des développeurs. Mais attention une partie de la bibliothèque fait partie de l'offre commerciale du produit.

Médiation et routage

Mule ESB possède un ensemble de composants lui permettant de réaliser l'essentiel des EIP existant.

Mule ESB s'appuie sur un DSL en XML complété par un langage d'expression, le Mule Expression Language (MEL) utilisable dans tous les composants de médiation.

L'une des grandes forces, là encore pour aider le développeur, c'est son composant Data Mapper qui permet de réaliser la transformation d'un message et donc le mapping d'une structure d'un message d'origine vers la structure d'un message cible, de manière graphique par un simple glisser-déposer.



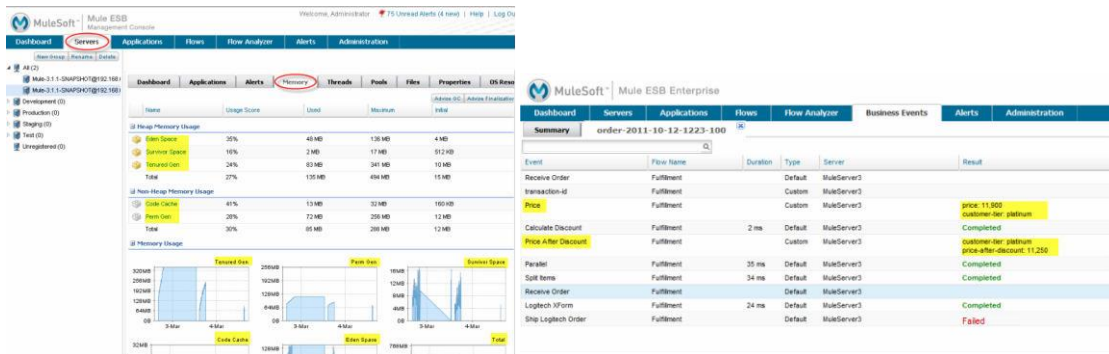
MuleSoft nous offre tous les outils classiques pour faire la médiation et le routage des messages, avec en petit plus son Data Mapper, gadget à l'origine, mais qui depuis la dernière version est devenu performant et réellement utilisable en production.

Sécurité

Mule offre un certain nombre d'outils de base permettant de sécuriser les flux d'informations : Spring Security, WS-Security, SAML, etc.

Mule ESB dans sa version commerciale propose des outils supplémentaires comme le support de l'OAuth2, les coffres d'accréditations, le cryptage ou le filtrage selon un élément du message.

Monitoring



Mule ESB met à disposition la classique interface JMX qui permettra de suivre les différentes métriques depuis les outils de monitoring du marché.

En plus du JMX, MuleSoft fournit à travers son application Mule Management Console (MMC), dans sa version commerciale, une interface web complète permettant de gérer les serveurs. Tout a été pensé pour avoir à portée de main tous les moyens de suivre l'activité, de gérer des alertes et d'analyser en temps réel les flux d'informations.

En plus de ces moyens, MMC permet de suivre l'activité business avec la visualisation des différents événements, des erreurs et des durées de traitements.

MuleSoft apporte donc à travers son MMC un outil simple d'emploi qui apporte toutes les informations nécessaires à un administrateur pour surveiller et agir pro-activement sur les serveurs. A noter que cette offre n'existe que pour la version commerciale de la solution.

Installation et configuration

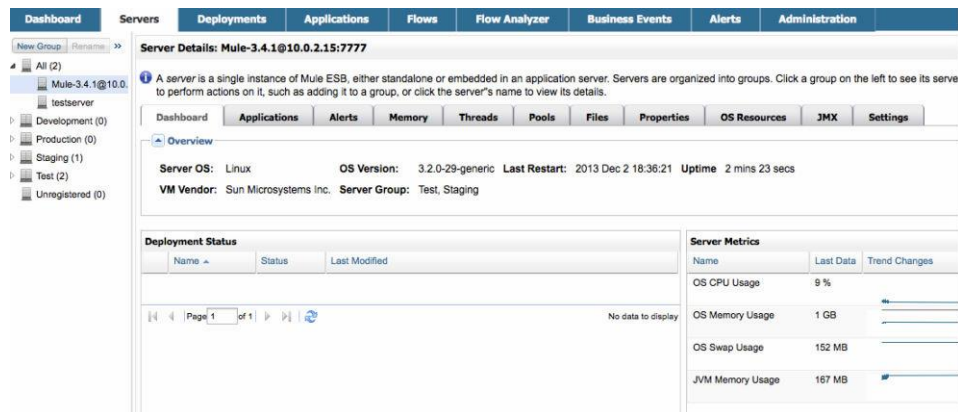
Mule est, comme les autres ESB présentés dans ce livre blanc, un ESB léger. En cela, il bénéficie d'un processus d'installation extrêmement simple. On télécharge et on lance le runtime soit par un .exe sous Windows ou soit par un script sh sous Linux et Mac.

En plus de ce déploiement en « stand-alone », Mule offre la possibilité de déployer Mule ESB dans un serveur d'application (Tomcat, Jboss, WebLogic, WebSphere, etc.) sous forme de war. Et si cela ne convient toujours pas, vous pouvez également embarquer Mule directement dans votre application. Mais attention aux limitations !

Les possibilités offertes au niveau de la configuration sont limitées toujours par cette volonté d'être le plus simple et le plus accessible possible.

Clustering

Mule ESB peut être déployé dans un cluster composé de 2 à 8 nœuds. La découverte entre chacun de ces nœuds est réalisée en multicast.



The screenshot displays the Mule Management Console (MMC) interface. The top navigation bar includes tabs for Dashboard, Servers, Deployments, Applications, Flows, Flow Analyzer, Business Events, Alerts, and Administration. The left sidebar shows a tree view of server groups: All (2), Mule-3.4.1@10.0.0.1, testserver, Development (0), Production (0), Staging (1), Test (2), and Unregistered (0). The main content area is titled 'Server Details: Mule-3.4.1@10.0.0.1:7777'. It includes a sub-navigation bar with tabs for Dashboard, Applications, Alerts, Memory, Threads, Pools, Files, Properties, OS Resources, JMX, and Settings. The 'Overview' section shows server information: Server OS: Linux, OS Version: 3.2.0-29-generic, Last Restart: 2013 Dec 2 18:36:21, Uptime: 2 mins 23 secs, VM Vendor: Sun Microsystems Inc., and Server Group: Test, Staging. Below this is a 'Deployment Status' table with columns for Name, Status, and Last Modified, which is currently empty. To the right, the 'Server Metrics' section displays a table with columns for Name, Last Data, and Trend Changes, showing OS CPU Usage (9%), OS Memory Usage (1 GB), OS Swap Usage (152 MB), and JVM Memory Usage (167 MB).

MuleSoft fournit un outil pour faciliter la gestion de ce mode, Mule Management Console (MMC). Cet outil va permettre de déployer les développements réalisés sur chacun des nœuds et de monitorer la plate-forme. Cet outil est utilisable cross environnements.

Mule ESB délègue la responsabilité du load balancing et du MOM à un outil tiers à la charge de l'utilisateur de la solution.

iPaaS



MuleSoft met à disposition son ESB dans le cloud grâce à sa plate-forme iPaaS Cloudhub.io. Mule s'appuie sur l'infrastructure d'Amazon AWS et recommande son usage pour la partie MOM, par

exemple avec Amazon SQS.

Cloudhub est le centre névralgique de la stratégie de MuleSoft et bénéficie donc d'un support avancé de ses équipes.

Depuis peu, la plate-forme Cloudhub, en plus de réaliser de l'intégration, peut être utilisée pour faire de l'API Management en exposant des services REST, en gérant les politiques de sécurité afférentes et en fournissant un portail pour les documenter.

MuleSoft peut être considéré comme ayant pour le moment le projet le plus abouti en termes fonctionnels et de maturité du produit. Il est à noter que des développements sont en cours chez l'éditeur pour proposer du cloud privé dans les prochains mois.

IX.3 REDHAT JBOSS FUSE



Éditeur : RedHat

Site internet de la solution :

<http://www.jboss.org/products/fuse/>

Date de création : 2007

Licence : Apache 2.0

Version étudiée : 6.1

IX.3.a Présentation

La société

RedHat est une entreprise dont la présentation n'est plus à faire. Société américaine à l'origine mais ayant des bureaux dans le monde entier, RedHat est un spécialiste de grande renommée dans le monde de l'open source. En 1993 Bob Young fonde AAC, en 1994 Mac Ewing crée sa version de linux Red Hat Linux, en 1995 Bob rachète la société de Mac et la fusionne avec la sienne pour fonder officiellement la société Red Hat Software.

En 2006, Red Hat achète la société Jboss, fondée en 1999 par Marc Fleury, spécialisé dans les intergiciels. Jboss deviendra par la suite une division à part entière de la société.

En 2012, Red Hat met la main sur la société Fuse Source, extrêmement connue pour son ESB léger qui vient compléter les produits de la gamme de Jboss.

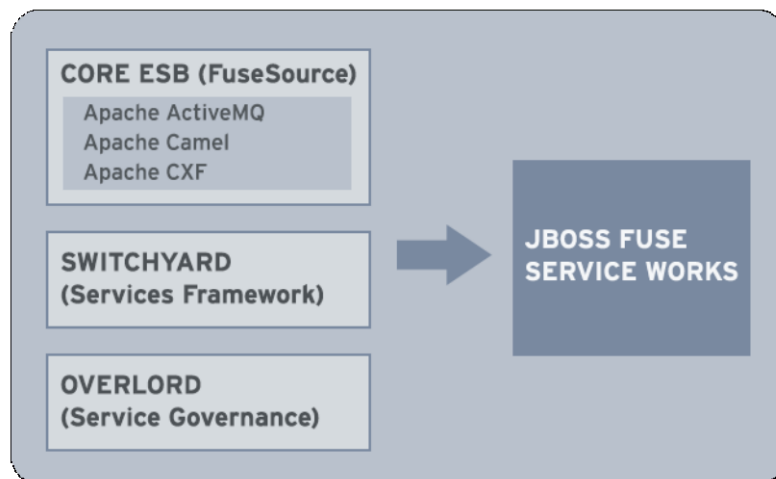
Modèle économique

RedHat est une société où le terme Open Source est écrit dans ses gènes. Tous ses produits sont librement disponibles.

Red Hat se finance sur la formation, l'expertise, l'accompagnement de projet et surtout sur le support et les garanties qu'il apporte à ses outils.

Le produit

RedHat Jboss Fuse est un vieux projet bien connu mais... pas sous ce nom. Fuse n'est autre que la version « officielle » de l'ESB léger Apache ServiceMix. Rob Davis, James Strachan, Hiram Chirino et Guillaume Nodet tous employés de RedHat ont créé successivement Apache ServiceMix, Apache ActiveMQ et Apache Camel entre 2003 et 2007 et continuent à contribuer activement à ces différents projets.



Aujourd'hui Fuse commence à se démarquer du ServiceMix de base en s'intégrant petit à petit dans la plate-forme Jboss et en apportant des évolutions importantes comme l'usage de fabric8 pour la gestion centralisée du déploiement de composants ou Hawtio pour l'interface d'administration.

RedHat Jboss Fuse est un projet en perpétuel évolution, innovant et supporté par une équipe solide de passionnés.

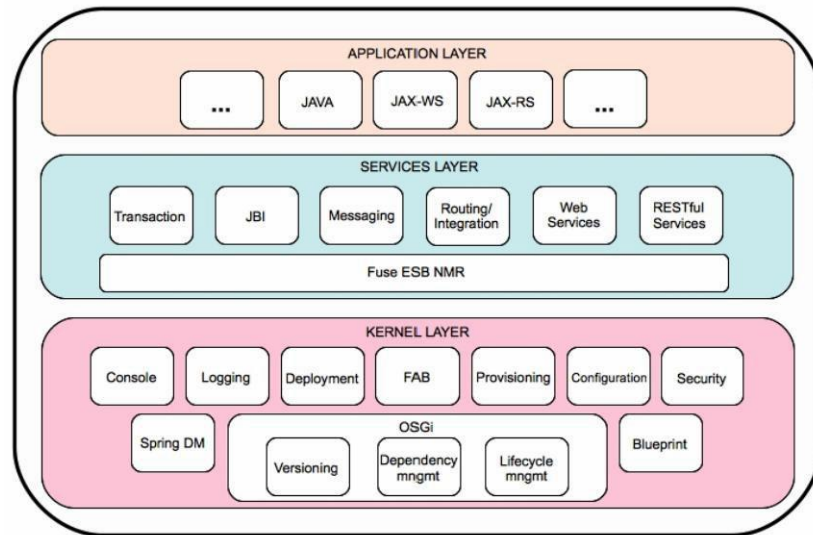
IX.3.b Spécificité du produit

Intégration avec l'écosystème Jboss

Depuis le rachat par Red Hat de Fuse Source, les produits Fuse sont intégrés petit à petit à l'ensemble de l'écosystème Red Hat. En plus du studio, on pourra constater ce travail dans Jboss Fuse Service Work, une plate-forme SOA qui combine l'ESB de Fuse, le framework de service Switchyard et l'outil de gouvernance Overlord de Red Hat.

IX.3.c Détail sur le projet

Architecture du produit



La base du Kernel de Fuse ESB est développée sur Apache Karaf, un runtime et un conteneur léger de composants OSGi. Ce kernel contient tous les éléments de base nécessaires au produit : provisioning, gestion des logs, injection de dépendances, gestion de la sécurité, console d'administration.

Au dessus du kernel se trouve la couche de services. Celle-ci va contenir toutes les interfaces et les implémentations de services qui permettront d'interagir avec les applications clientes. On y retrouvera notamment le framework de web-services Apache CXF, la librairie de médiation Camel et le routeur de message normalisé qui transmettra les messages entre les différents composants OSGi.

La partie application est constitué des développements des utilisateurs de Fuse qui interagiront avec des applications ou services tiers et des services internes à Fuse ESB.

Cette architecture est en réalité une machinerie complexe mais bien huilée qui s'appuie sur de nombreux composants open source reconnus comme faisant partie de ce qui se fait de mieux actuellement.

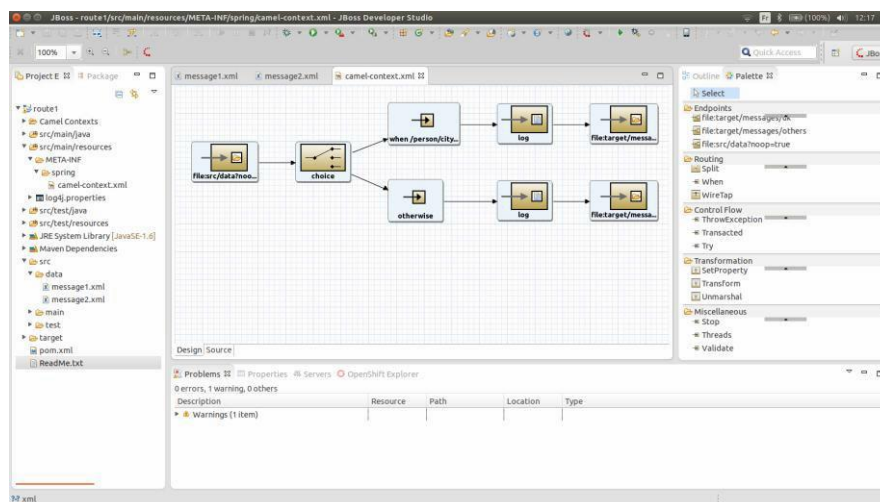
Outil de développement



par défaut).

Les développements sur Fuse ESB sont réalisables depuis quelques mois depuis le JBoss Developer Studio basé sur Eclipse. Il sera nécessaire d'installer sur celui-ci le JBoss Developer Studio Integration Stack disponible sur le Central de Red Hat pour pouvoir développer (il n'est pas mis à disposition

Le studio n'est en réalité qu'un outil de développement qui permet d'initialiser par clic un projet Fuse en utilisant des archétypes Maven.



Le studio offre la possibilité de créer et d'éditer ses flux Camel de manière graphique comme ses concurrents et de les débbugger.

Depuis peu il est également possible d'éditer les flux Camel directement depuis la console d'administration web sous Hawtio de manière graphique ou en éditant les sources.

```

1 <route xmlns="http://camel.apache.org/schema/spring" customId="true" id="foo">
2   <from uri="activemq:personnel.records" customId="true" id="personnelRecords"/>
3   <choice id="choice1">
4     <when customId="true" id="isCityLondon">
5       <xpath>/person/city = 'London'</xpath>
6       <to uri="file:target/messages/uk" customId="true" id="messagesUK"/>
7     </when>
8     <otherwise id="otherwise1">
9       <to uri="file:target/messages/others" customId="true" id="messagesOthers"/>
10    </otherwise>
11  </choice>
12 </route>
  
```

Fuse nous met à disposition un studio de développement très orienté développeur Java, limité sur sa partie graphique et qui aurait notamment besoin d'ajout d'aide contextuelle pour les nouveaux arrivants.

Web-services et connecteurs

Fuse ESB s'appuie sur CXF comme Framework de web-service qui est la combinaison

Apache CXF de Celtix (IONA Technologies/Progress Software/Fuse Source) et XFire (CodeHaus).

Il n'y a pas de connecteurs à proprement parler mais des médiateurs spécifiques fournis par la librairie de médiation Camel pour permettre d'exploiter plus facilement de nombreux services comme Twitter, Salesforce, SAP, etc. Ce sont des composants performants et utilisés dans de nombreux projets.

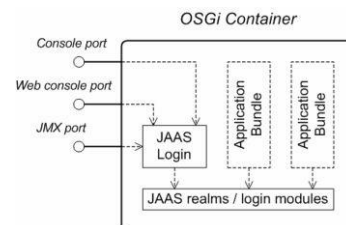
Médiation et routage



Fuse ESB s'en remet à Apache Camel pour toute sa partie médiation et routage. Ce projet issu de Service Mix est devenu l'une des librairies d'intégration de référence sur le marché. Forte d'une grande communauté, elle met à disposition de nombreux médiateurs et implémente l'ensemble des EIP.

Sécurité

Fuse ESB se base sur la spécification JAAS pour sécuriser l'ensemble des composants de son produit et met à disposition l'ensemble des outils classiques à travers CXF et Camel comme SSL, gestion de certificats, WS-Security, OAuth 2, etc.

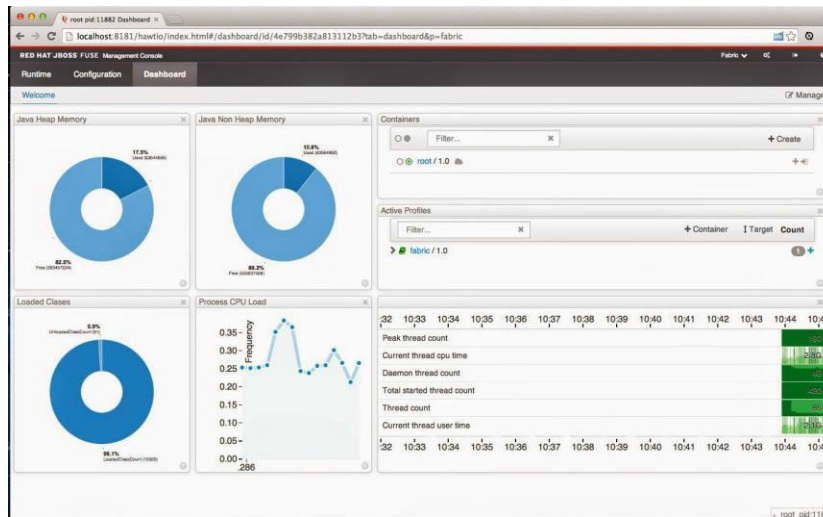


Monitoring

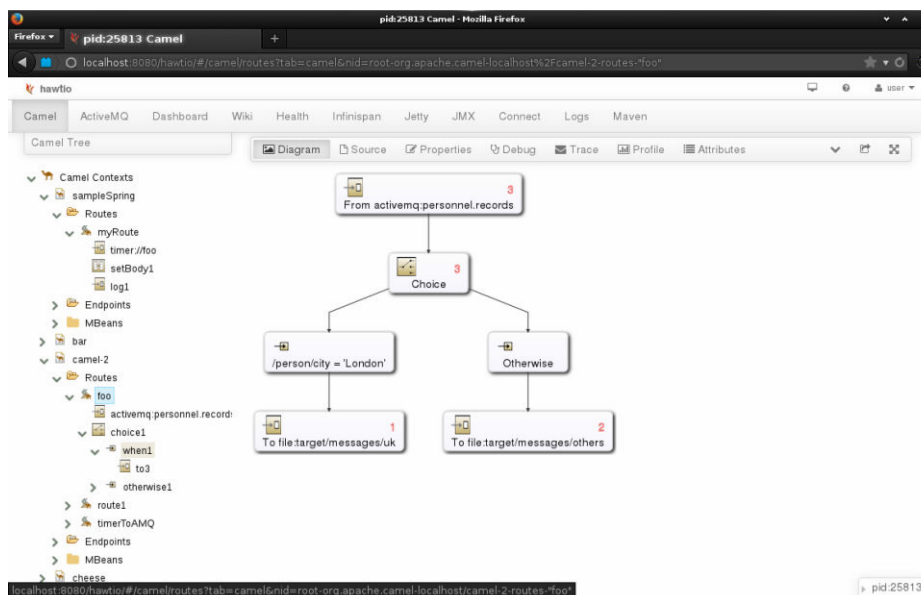


Red Hat a sorti début 2013 un nouvel outil, Hawtio, qui est une console web utilisant AngularJS pour sa partie front. Depuis ce produit est intégré à de nombreux outils comme A-MQ/ActiveMQ et ServiceMix/Fuse ESB. De nombreux plugins spécifiques par exemple à Camel ou Maven sont mis à disposition.

Hawtio va ainsi fournir des tableaux de bord pour suivre en temps réel l'activité de Fuse ESB et permettre d'en créer de nouveaux personnalisés qui exploiteront les données fournies par une interface JMX.



Bien pratique la console d'administration va ainsi permettre par exemple de voir en temps réel le passage de messages dans des flux Camel.



Clustering

Fuse ESB se base sur Apache Karaf pour le clustering et comme runtime de composants OSGi. Depuis quelques mois, l'équipe de développement a mis l'accent sur son système d'approvisionnement et a créé le projet Fabric8 qui reprend une partie du code spécifique à Fuse par rapport à ServiceMix. Fabric8 est un système d'approvisionnement et de gestion multi-conteneur qui sait gérer notamment du Karaf, Tomcat, Wildfly et Vertx. Les prochaines versions de Fuse ESB changeront donc en profondeur avec ce nouveau système.

iPaaS

RedHat met à disposition depuis peu FuseESB dans le cloud sur sa plate-forme OpenShift. Le service est en version alpha et évolue rapidement.

L'environnement est configuré et pleinement fonctionnel en quelques clics. Les développements pourront être réalisés dans le Developer Studio de Jboss et déployés directement via son interface. Si l'on souhaite avoir un contrôle plus avancé que celui mis à disposition par la console d'administration web, OpenShift met à disposition un outil en ligne de commande écrit en Ruby pour interagir finement avec la plate-forme. Les développements doivent être versionnés sous Git (OpenShift en fourni un si nécessaire) et grâce à un hook, chaque commit sera directement déployé sur la plate-forme.

OpenShift est une plate-forme plus large qu'un simple iPaaS, Red Hat préférant même parler de xPaaS pour la qualifier. Cette plate-forme est éprouvée par des milliers de projets qui l'utilisent tous les jours.

IX.4 TALEND ESB



Éditeur : Talend

Site internet de la solution : <http://www.talend.com/>

Date de création : 2011

Licence : Apache 2.0 / commerciale

Version étudiée : 5.5.0

IX.4.a Présentation

La société

Talend est une société française fondée en 2005 par Bertrand Diard et Fabrice Bonan, spécialisée dans l'intégration de données et disposant de bureaux aux 4 coins du monde.

Talend sort sa première solution, Talend Studio for Data Integration, en octobre 2006. Son logiciel phare a été téléchargé plus de 20 millions de fois et la société compte plus de 3 500 clients de part le monde.

Talend est une entreprise très impliquée dans le monde de l'Open Source et fait notamment partie du conseil d'administration du consortium OW2 et contribue à de nombreux projets Apache.

La société a fait l'acquisition de la société allemande Sopera fin 2010, un membre de la fondation Eclipse et grand spécialiste de l'intégration de données outre-Rhin qui avait pour clients Deutsche Post, Zurich Insurance ou encore Landesbank Berlin.

Modèle économique

Talend publie sous licence open source et gratuitement le cœur de tous ses produits.

La société se rémunère en vendant sous licence commerciale des versions améliorées facilitant la vie des développeurs et des administrateurs. Les prix des licences sont calculés selon de nombreux paramètres, nombre de runtime, nombre de développeurs, niveau de support, etc. Talend propose également un support en développement et en production, des séances de formation et de l'expertise technique.

Le produit

Talend ESB est un produit partiellement Open Source, commercialisé depuis 2011, à la suite du rachat de la société Sopera. La solution est issue de la fusion de l'outil Sopera Advanced Service Factory (ASF), existant depuis 2007, et de l'outil Talend Data Management.

Talend ESB est un outil de premier plan permettant de jouer à la fois sur le tableau de la Qualité de Données et de la Qualité de Service. Basées sur ServiceMix, les key features qui démarquent cette solution sont une plate-forme de médiation très bien outillée, une capacité de distribution des services, un monitoring technique centralisé et de haut niveau le tout incorporé dans la plate-forme d'intégration unifiée de Talend.

Talend ESB est disponible dans une version communautaire sous licence Apache et en version entreprise sous une licence commerciale.

IX.4.b Spécificité du produit

Talend Unified Platform

Talend fournit une plate-forme unifiée permettant la gestion complète de l'intégration au sein d'un système d'information :

- DQM : Data Quality Management
- MDM : Master Data Management
- ETL : Extract-Transform-Load
- ESB : Enterprise Service Bus
- BPM : Business Process Management



Cette plate-forme apporte en support :

- un outil de développement commun, Talend Open Studio,
- un référentiel commun

- un conteneur unifié d'exécution
- un système de déploiement
- une console de monitoring unique.

IX.4.c Détails sur le projet

Architecture du produit

Talend ESB est bâti intégralement à l'aide de standards Open Source dont les principales briques sont issues de la fondation Apache et plus particulièrement de son projet Apache ServiceMix :

- Apache Camel⁸ pour la médiation
- Apache ActiveMQ⁹ comme message broker s'appuyant sur Camel et Jetty
- Apache CXF¹⁰ comme framework de développement des Web Services
- Apache Karaf¹¹ – conteneur léger de composant OSGi

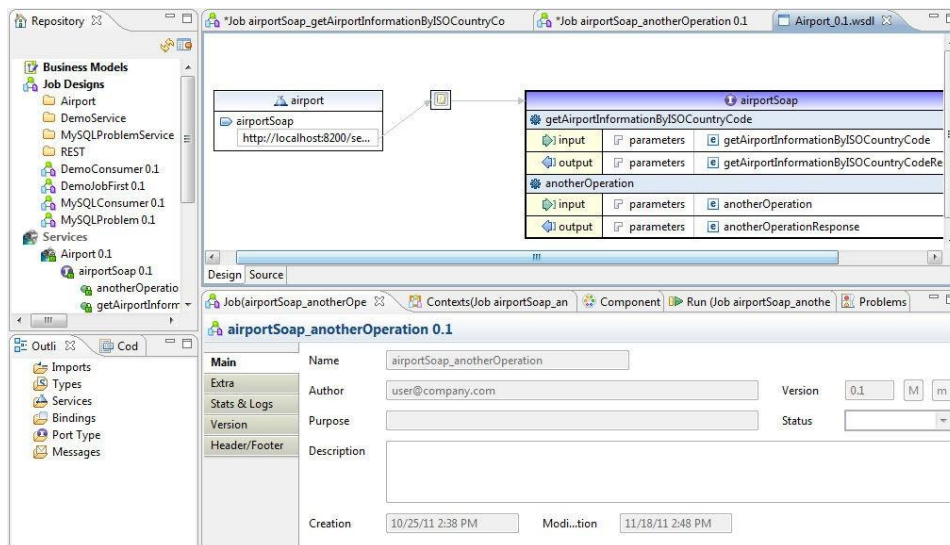


Talend les a associés à une version personnalisée d'Hyperic HQ pour le monitoring de l'ensemble.

Talend a choisi de s'appuyer sur des briques open source solides et éprouvées. L'outil bénéficie de fait pleinement des dernières avancées et du support des différentes communautés.

Outil de développement

Talend met à disposition un IDE, Talend Open Studio basé sur Eclipse, commun à tous les produits de sa plate-forme. Ce studio permet donc de créer des routes Camel, des jobs classiques d'intégration de données et des services.



Talend a séparé en deux perspectives la réalisation des flux ESB :

- une perspective orientée service qui va permettre de configurer toutes les interactions externes
- une perspective de médiation qui va permettre de transformer, filtrer et orienter les messages qui transitent.

Le studio incorpore les principaux médiateurs fourni par Apache Camel et fournit un composant configurable pour ajouter les autres.

Les avantages du studio :

- une documentation intégrée : un simple clic vous permet d'accéder à la documentation d'un composant et de voir des exemples d'utilisation.
- un runtime intégré qui permet de tester directement ses développements.
- une visualisation directement dans son IDE de la transmission des messages entre les composants en phase de test.

Talend offre un studio bien conçu qui facilite la vie et améliore la productivité des développeurs. La différenciation de perspective entre service et intégration ne dépaysera pas les connaisseurs de Talend DI.

Web-services et connecteurs

Talend ESB s'appuie sur la puissance d'Apache CXF pour développer et exposer de manière transparente des web-services basés sur des implémentations d'API tels que JAX-RS ou JAX-WS. Cette utilisation lui permet de supporter de nombreux protocoles et formats tels que SOAP, HTTP/REST, CORBA ou JMS.

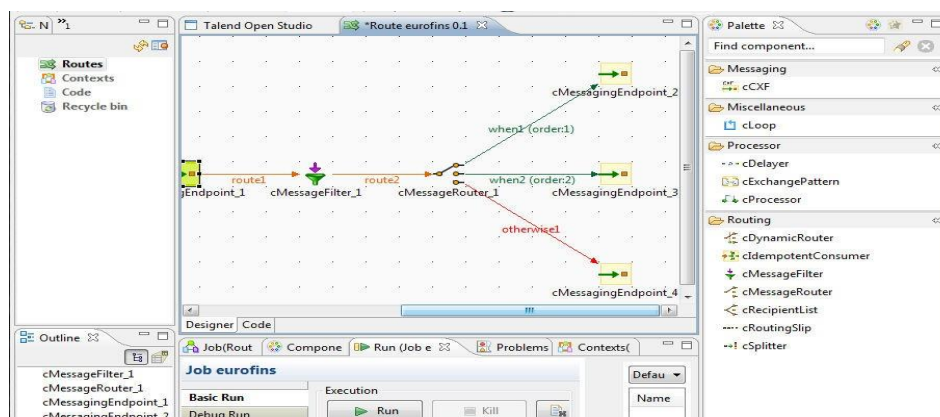
L'outil allie à Apache CXF la richesse de la bibliothèque Camel qui met à disposition la gestion de endpoints spécifiques à des services d'entreprise, ou grand public, comme Salesforce, SAP, Solr ou Twitter.

Talend bénéficie de toute la puissance et la richesse de la partie ETL (Transformation, consolidation et nettoyage de données, mapping avancé, très riche bibliothèque de connecteurs aux logiciels du marché).

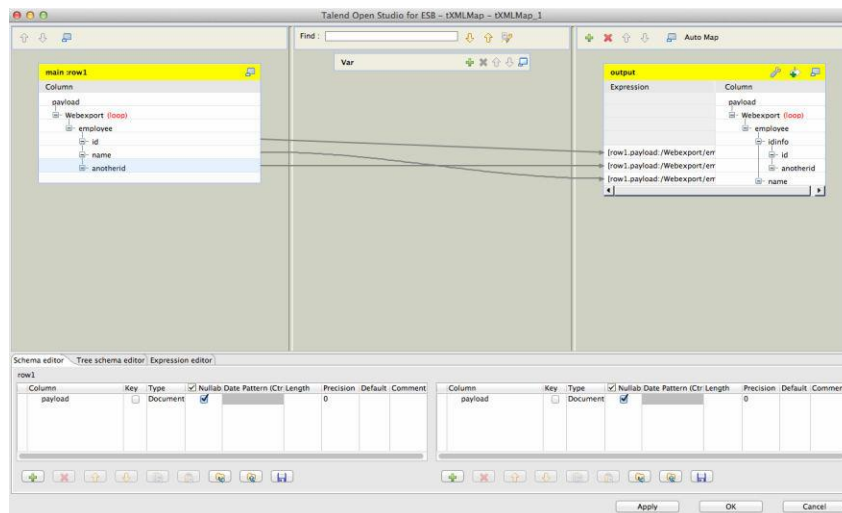
Médiation et routage

Talend, dans son esprit d'unification d'expérience, a intégré la perspective de développement des routes de médiation dans le même studio, en gardant la même logique ergonomique que celle de l'ETL. L'intégration entre les routes de médiation et les jobs talend (ETL) est native.

Talend ESB s'appuie sur Apache Camel et sa bibliothèque forte de plus de 150 composants¹² et de Camel Extra qui apporte une dizaine de composants supplémentaires non hébergés directement sur le site de la fondation Apache pour des problèmes d'incompatibilités de licences (Hibernate, Neo4j, ZeroMQ, etc). Apache Camel est une librairie extensible de composants et il est facile de créer son propre composant qui pourra ensuite être partagé avec la communauté. Chaque composant est documenté en détail en tant que consommateur et/ou producteur avec l'ensemble de ces options et illustré par de nombreux exemples.



Talend a récemment introduit des outils de mapping graphiques avancés (cMap et Talend Data Mapper) qui simplifient et enrichissent ses capacités de mapping de données.



Sécurité

Sécuriser l'accès aux services déployés est un élément clé lors de la mise en place d'un ESB. La solution ESB de Talend est conforme aux spécifications WS-Security et WS-Policy. Elle fournit un module Secure Token Service (STS) pour générer des jetons de sécurité. Talend supporte de nombreux autres standards de sécurités comme SAML et Oauth.



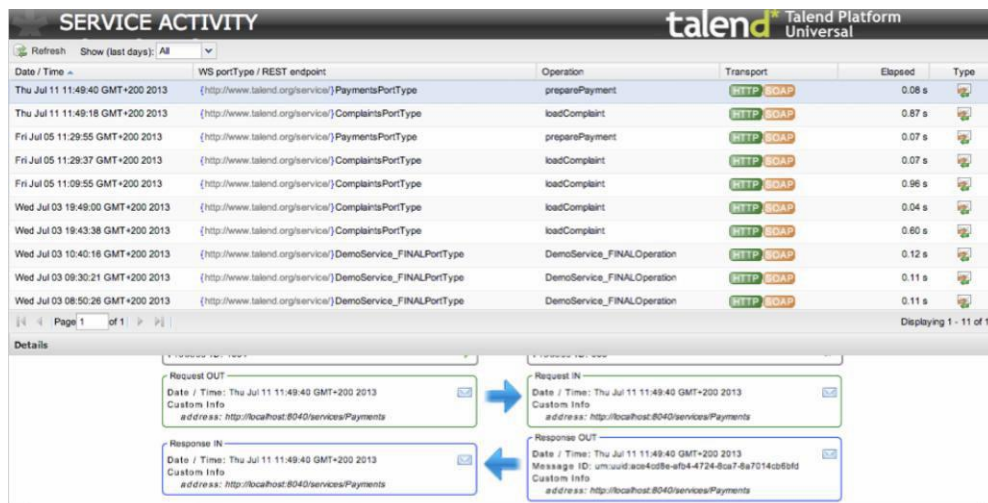
Dans sa version entreprise Talend fournit des outils supplémentaires liés à la sécurité, comme XML Key Management Specification (XKMS) et Talend Security Service. Ce dernier est basé sur le gestionnaire d'identité Apache Syncope et permet d'implémenter ses propres politiques de sécurité via XACML.

Talend fournit tout ce que l'on peut attendre d'un ESB et même plus dans sa version professionnelle avec l'ajout d'un gestionnaire d'identités.

Monitoring

Talend ESB met à disposition une interface JMX au niveau de chacune des briques de sasolution.

Une interface Service Activity Monitoring (SAM) est mise à disposition pour visualiser les messages échangés et suivre les métriques afférentes aux services déployés. Cette interface est disponible au sein de l'outil d'administration de la suite Talend : Talend Administration Center (TAC).



Date / Time	WS portType / REST endpoint	Operation	Transport	Elapsed	Type
Thu Jul 11 11:49:40 GMT+200 2013	{http://www.talend.org/service/} PaymentsPortType	preparePayment	HTTP SOAP	0.08 s	
Thu Jul 11 11:49:18 GMT+200 2013	{http://www.talend.org/service/} ComplaintsPortType	loadComplaint	HTTP SOAP	0.87 s	
Fri Jul 05 11:29:55 GMT+200 2013	{http://www.talend.org/service/} PaymentsPortType	preparePayment	HTTP SOAP	0.07 s	
Fri Jul 05 11:29:37 GMT+200 2013	{http://www.talend.org/service/} ComplaintsPortType	loadComplaint	HTTP SOAP	0.07 s	
Fri Jul 05 11:09:55 GMT+200 2013	{http://www.talend.org/service/} ComplaintsPortType	loadComplaint	HTTP SOAP	0.96 s	
Wed Jul 03 19:49:00 GMT+200 2013	{http://www.talend.org/service/} ComplaintsPortType	loadComplaint	HTTP SOAP	0.04 s	
Wed Jul 03 19:43:38 GMT+200 2013	{http://www.talend.org/service/} ComplaintsPortType	loadComplaint	HTTP SOAP	0.60 s	
Wed Jul 03 10:40:18 GMT+200 2013	{http://www.talend.org/service/} DemoService_FINALPortType	DemoService_FINALOperation	HTTP SOAP	0.12 s	
Wed Jul 03 09:30:21 GMT+200 2013	{http://www.talend.org/service/} DemoService_FINALPortType	DemoService_FINALOperation	HTTP SOAP	0.11 s	
Wed Jul 03 08:50:28 GMT+200 2013	{http://www.talend.org/service/} DemoService_FINALPortType	DemoService_FINALOperation	HTTP SOAP	0.11 s	

Details

```

graph LR
    subgraph "Request OUT"
        R1[Request OUT  
Date / Time: Thu Jul 11 11:49:40 GMT+200 2013  
Custom Info  
address: http://localhost:8040/services/Payments]
    end
    subgraph "Response IN"
        R2[Response IN  
Date / Time: Thu Jul 11 11:49:40 GMT+200 2013  
Custom Info  
address: http://localhost:8040/services/Payments]
    end
    subgraph "Request IN"
        R3[Request IN  
Date / Time: Thu Jul 11 11:49:40 GMT+200 2013  
Custom Info  
address: http://localhost:8040/services/Payments]
    end
    subgraph "Response OUT"
        R4[Response OUT  
Date / Time: Thu Jul 11 11:49:40 GMT+200 2013  
Message ID: um:uuid:ace408e-efb4-4724-8ca7-8a7014cb6fd  
Custom Info  
address: http://localhost:8040/services/Payments]
    end
    R1 --> R3
    R3 --> R4
    R4 --> R2
  
```

Pour collecter et exploiter les logs, il est possible d'utiliser Talend Log Server qui est fourni dans la version Entreprise de la solution. Talend Log Server s'appuie sur le trio : Logstash, Elasticsearch et Kibana.

Pour innover et enrichir ses fonctionnalités de monitoring, Talend a aussi a intégré Hyperic HQ. Ces choix sont certes judicieux mais ils impliquent la nécessité d'avoir de nouvelles connaissances pour pleinement les exploiter.

Installation et configuration

Dans sa version commerciale, la solution Talend fournit depuis la version 5 un assistant d'installation graphique ou en ligne de commande pour l'ensemble de la suite dont notamment l'application d'administration, le référentiel partagé, la base de donnée, le moteur de compilation, les serveurs d'exécution. Cet assistant ne se contente pas d'installer les produits de Talend mais permet également d'installer les produits tiers nécessaires comme SVN, Tomcat ou MySQL ou de s'intégrer avec ceux déjà existants.

Dans la version communautaire toute la procédure d'installation sera à réaliser manuellement. Les modules supplémentaires comme STS, SAM, Zookeeper et ActiveMQ seront à configurer entièrement.

Clustering

Talend ESB runtime s'appuie sur le conteneur OSGI Apache Karaf. C'est la puissance de celui-ci et en particulier de son composant Apache Cellar qui lui confère la capacité de mise à l'échelle. Apache Cellar est construit sur Hazelcast, un système de cache distribué en mémoire.

La distribution des événements et la gestion de la disponibilité des services sont nativement gérées par Cellar. La phase de découverte des nœuds dans le cluster se fait soit par multicast soit par WKA.

iPaaS

Talend ne dispose pas d'offre iPaaS au moment de la réalisation de ce document mais commercialise une offre dédiée « platform for Hybrid Cloud », préparée pour simplifier son intégration dans un cloud existant.

Par ailleurs, Talend a commercialisé par le passé la partie ETL en SaaS, et compte tenu de leurs communications et l'enjeu du cloud dans le marché, une offre en iPaaS verra certainement le jour dans les mois à venir.

IX.5 WSO2 ESB



Éditeur : WSO2

Site internet de la solution :

<http://www.wso2.com/>

Date de création : 2005

Licence : Apache 2.0

Version étudiée : 4.8.1

IX.5.a Présentation

La société

WSO2 est une société américaine dont le siège social se situe à Palo Alto en Californie. La société édite une suite de solutions dédiées à la création d'architecture SOA et en particulier WSO2 ESB.

La société a été fondée par le Dr. Sanjiva Weerawarana et Paul Fremantle, anciens de chez IBM, créateurs notamment de Apache SOAP et WSIF (Web Services Invocation Framework).

WSO2 a aujourd'hui des bureaux aux Etats-Unis (Palo Alto et Bloomington), au Royaume-Uni (Londres) et au Sri Lanka (Colombo). L'éditeur n'a pas de bureau en France.

Modèle économique

WSO2 ESB est basé sur des composants 100% Open Source sous licence Apache 2. L'outil ne possède pas de version premium payante. Son code source est librement accessible sur un dépôt SVN.

La société se rémunère :

- Sur le support en développement et en production dont la tarification est basée sur le nombre d'instances de production.
- En proposant des séances de formation et un programme « quick starter » pour commencer à développer avec un coach.
- Avec son offre de services professionnels : consultants SOA ou experts techniques.

WSO2 propose également des certifications sur ces différentes solutions.

Le produit

WSO2 ESB est un projet totalement open source développé depuis 2005. Une nouvelle version du produit est publiée tous les 6 mois environ.

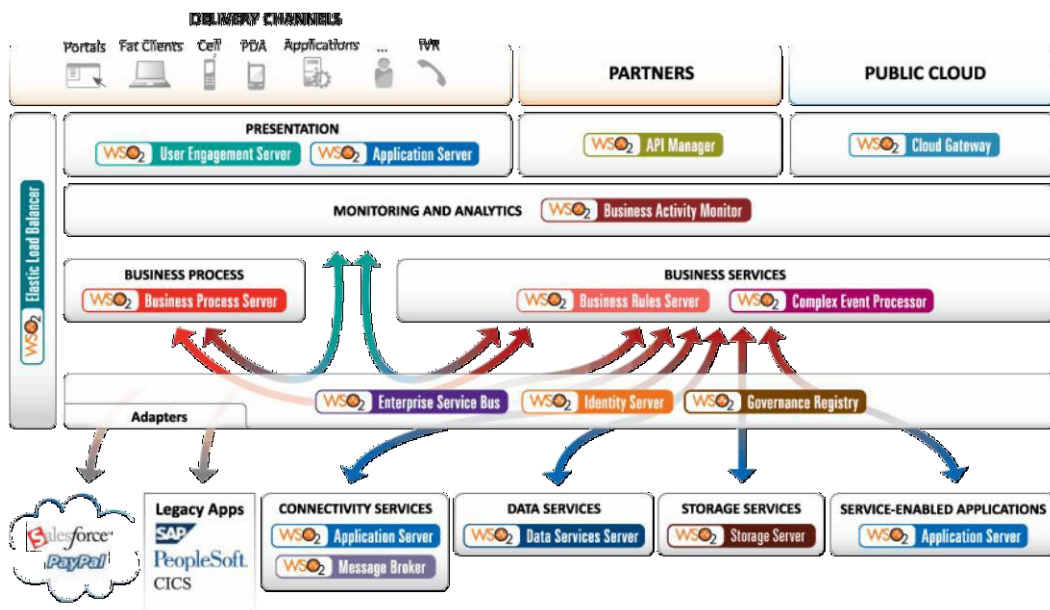
WSO2 ESB repose sur la plateforme Carbon, implémentant les spécifications OSGi, commune à tous les produits de WSO2, modulaire, extensible et pouvant nativement être mis à l'échelle. Il utilise les projets Apache Synapse, pour la composante de médiation, et Apache Axis2, pour les web services, dont l'éditeur est le créateur et le principal contributeur.

Rapide et ayant une empreinte mémoire réduite le bus de service d'entreprise de WSO2 est hautement interopérable et supporte de nombreux transports, formats et protocoles comme POP3, JMS, AMQP, FIX, CIFS, MLLP, SMS, SOAP, REST, EDI, HL7, OAGIS, Hessian, CORBA/IIOP.

IX.5.b Spécificités du produit

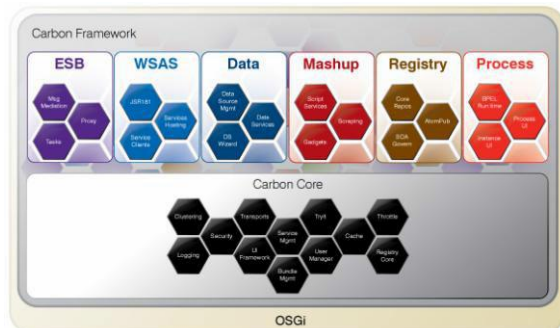
Plateforme Middleware WSO2

WSO2 ESB n'est qu'un produit parmi une plate-forme middleware complète orientée SOA. WSO2 y ayant ajouté en début d'année une série de produits dédiés au mobile pour surfer sur la vague du « Bring Your Own Device (BYOD) ». Tous les produits de cette plate-forme sont basés sur le même framework, WSO2 Carbon.



IX.5.c Détail sur le projet

Architecture du produit



WSO2 ESB est basé sur le framework Carbon qui possède plus de 250 composants OSGi. Les composants de base comme la gestion des logs, la gestion du déploiement des configurations ou l'interface d'administration sont utilisés dans tous les produits de WSO2. D'autres sont spécifiques à un produit donné ou

commun à plusieurs comme la gestion du registre.

Grâce à ce framework il est possible de créer facilement son propre produit basé sur cette librairie de composants.

WSO2 ESB est basé sur le conteneur de web-service Apache Axis 2 et l'ESB léger Apache Synapse. Ces 2 solutions sont très orientées SOAP et supportent les spécifications WS-*

La solution embarque un registre, issu de WSO2 Governance Registry, pour stocker les différentes configurations des web-services, routes et endpoints.

WSO2 fournit une architecture agile permettant de facilement faire évoluer le produit et riche d'une importante bibliothèque de composants. On regrettera par contre que celle-ci soit peu documentée.

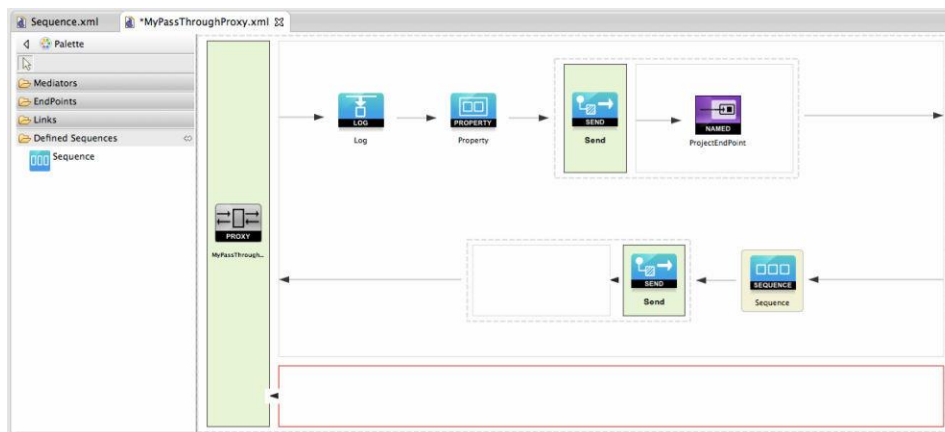
Outil de développement



WSO2 met à disposition un IDE basé sur Eclipse, commun à tous les produits de sa plate-forme. Récemment refondu, le studio permet de créer des conteneurs d'applications composites (CAR) pouvant

notamment contenir les définitions des web-services, des routes et des endpoints.

WSO2 Developer Studio, étant un outil commun à toute la plate-forme, il n'embarque pas directement les différents produits. Pour tester et déployer ses développements, il est possible de configurer dans l'outil les paramètres de connexion à son serveur qui permettront par un simple clic de déployer les archives CAR.



L'outil a été refondu fin 2013 et est depuis en actif évolution.

WSO2 ESB offre un deuxième outil de développement directement embarqué avec le serveur. L'ensemble des possibilités de l'ESB peuvent y être directement configuré. Cette outil est en voie de dépréciation et devrait être remplacé à terme entièrement par le studio.



WO2 travaille actuellement avec Codenvy pour mettre à disposition un IDE dans le cloud.

La variété des outils de développement mis à disposition est assez intéressante mais l'accompagnement utilisateur est encore à améliorer.

Web-services et connecteurs

WSO2 ESB permet d'exposer des web-services en SOAP et en REST grâce à Apache Axis 2 et supporte de nombreux protocoles et formats tels que POP3, JMS, AMQP, FIX, CIFS, MLLP, SMS, EDI, HL7, OAGIS, Hessian, CORBA/IIOP.

L'outil dispose depuis peu (4.8) de connecteurs spécifiques à certains services dont le nombre est encore très limité : Google Spreadsheet, JIRA, Lnkedin, Paypal, Salesforce , Twilio et Twitter.

WSO2 devrait accentuer ces efforts dans ce domaine dans les prochains mois pour rejoindre ces concurrents.

Médiation et routage



sur un DSL en XML.

WSO2 ESB repose entièrement sur Synapse pour la médiation et le routage. Apache Synapse met à disposition une quarantaine de composants de médiation utilisables dans les routes. La syntaxe pour les configurer est basée

Les messages transmis entre les composants et vers l'extérieur doivent être en général encapsulés dans un message SOAP car Apache Axis2 restreint le format utilisable.

En plus de la documentation d'Apache Synapse¹³, WSO2 met à disposition deux documentations : la première reprenant la description de chacun des composants au sein du guide utilisateur¹⁴ et la seconde étant une documentation ciblée sur l'implémentation d'EIPs particuliers¹⁵. La documentation fournit quelques exemples d'utilisation simples.

WSO2 met à disposition toutes les briques de base nécessaires à la médiation et au routage.

Sécurité

WSO2 ESB offre la possibilité de sécuriser ses flux en utilisant Apache Rampart pour la gestion de WS-Security et WS-Trust pour les messages SOAP. Il met à disposition des médiateurs Oauth2 et LDAP pour authentifier les messages dans un autre format.

L'outil offre également la possibilité de mettre en place des polices, par exemple pour limiter le nombre de messages envoyés.

WSO2 ESB peut être couplé avec WSO2 Identity Manager pour bénéficier d'une authentification centralisée des flux.

WSO2 met donc à disposition tous les outils de sécurité que l'on attend de la part d'un ESB.

Monitoring

WSO2 ESB propose dans son interface d'administration des outils graphiques de monitoring. Il est ainsi possible de visualiser les constantes systèmes et les statistiques concernant les différents flux et web-services configurés.



WSO2 ESB met à disposition une interface JMX et SNMP pour pouvoir être intégré avec les outils de supervision classiques existants sur le marché.

L'outil offre la possibilité de configurer les logs avec une granularité au niveau d'un web-service ou d'un flux particulier grâce aux possibilités de Log4J. L'ensemble de cette configuration peut être modifié à la volée dans l'interface d'administration de l'ESB.

En complément du Service Activity Monitor (SAM), il est possible de faire du Business Activity Monitor (BAM) en le combinant avec WSO2 BAM.

WSO2 ESB offre une solution de monitoring très poussée et son alliance avec WSO2 BAM offre un réel avantage à la solution.

Installation et configuration

L'installation de base est très simple puisqu'il suffit de décompresser l'archive et de lancer un jar via un script fourni. En quelques secondes on se retrouve devant un ESB pleinement fonctionnel.

La connexion à une base de données ou à une application particulière comme ActiveMQ nécessite de déposer manuellement sur le serveur les librairies clientes et de déclarer ces nouvelles possibilités dans un fichier de configuration.

WSO2 ESB étant un ESB léger il ne contient que le strict nécessaire. Il ne possède par exemple pas de MOM et recommande l'installation d'ActiveMQ ou de WSO2 MQ (basé sur Qpid).

WSO2 offre des outils dont la mise en route est très rapide. Cependant, le paramétrage de l'outil lors de la mise en production peut vite devenir complexe et

chronophage compte tenu de la multiplicité des paramètres et des fichiers de configuration. La documentation elle-même se retrouve vite limitée par rapport aux possibilités.

Clustering

WSO2 ESB se base sur les possibilités du framework Carbon pour la clusterisation. La solution n'intègre pas directement un load balancer et il sera nécessaire d'en configurer un devant les différents nœuds du cluster. WSO2 Load Balancer est recommandé mais les autres outils du marché répondront parfaitement au besoin.

WSO2 ESB offre la possibilité de partager son registre entre ses différents nœuds en utilisant une base de données traditionnelle et ou d'en déléguer la responsabilité à WSO2 Governance Registry.

Pour offrir de meilleures performances, le produit inclut la solution Hazelcast qui va permettre de distribuer le cache de données entre les différents nœuds du cluster.

iPaas

WSO2 ESB, comme l'intégralité de la plate-forme WSO2, peut être déployée dans le cloud. WSO2 se base sur le framework PaaS Stratos reversé en juin 2013 à la fondation Apache. La société offre 3 possibilités :

- WSO2 Public Cloud, une plate-forme cloud partagée entre les différents utilisateurs
- WSO2 Managed Cloud, une plate-forme cloud dédiée à un utilisateur et gérée par WSO2
- WSO2 Private Cloud, un outil téléchargeable permettant de déployer et gérer sa propre plate-forme cloud.

Ce service encore en Beta n'a pu être testé pour le moment mais sa description nous semble prometteuse et offre une innovation face à ses concurrents : être totalement maître de son cloud.

IX.6 AUTRES ESB OPEN SOURCE

Il existe de nombreux autres ESB open sources sur le marché et il serait trop long de tous les citer. Vous retrouverez ici quelques ESB qui restent de bonne facture mais que nous n'avons pas sélectionnés dans notre top du moment.

IX.6.a Open ESB



Éditeur : OpenESB Community

Site internet de la solution : <http://www.open-esb.net/>

Date de création : 2006

Licence : CDDL

OpenESB aussi commercialisé sous le nom de Glassfish ESB et était l'ESB de Sun Microsystems avant son rachat par Oracle. Oracle n'a pas souhaité poursuivre son développement et depuis lors une communauté d'utilisateur s'est formée pour l'assurer. Structuré autour de l'implémentation complète de JBI, cet ESB ne bénéficie plus d'un gros poids lourd derrière lui pour en assurer une réelle évolution.

IX.6.b Petals ESB



Éditeur : Linagora / OW2

Site internet de la solution : <http://petals.ow2.org/>

Date de création : 2006

Licence : LGPL

Petals ESB est édité par le consortium OW2 et Linagora qui a fait acquisition de la société PetalsLink en 2012. Nous n'avons pas d'information sur l'évolution future du produit.

IX.6.c RedHat Jboss ESB



Éditeur : Red Hat

Site internet de la solution : <http://jbossesb.jboss.org/>

Date de création : 2006

Licence : LGPL

Jboss ESB est un ESB d'ancienne génération originellement développé par une compagnie d'assurance canadienne (Aviva Canada). Cette solution est en voie de dépréciation remplacé par des solutions plus légères de l'éditeur, SwitchYard et Fuse.

IX.6.d Ultra ESB



Éditeur : AdroitLogic

Pays d'Origine : Sri Lanka

Site internet de la solution : <http://adroitlogic.org/>

Date de création : 2010

Licence : AGPL / commerciale

UltraESB est un nouvel arrivant sur le marché mais qui a en réalité beaucoup d'expérience. La société supportant le produit possède dans ses rangs l'un des plus anciens et actifs contributeurs d'Apache Synapse, qui a souhaité repartir sur une nouvelle base technique pour son nouveau projet.

AdroitLogic fait beaucoup parler de son ESB en utilisant abondamment ses benchmarks de performances qu'il publie sur <http://esbperformance.org/>.

Aujourd'hui, il a encore tout à prouver en termes de fonctionnalités, d'adhérence de la communauté et de références.

X - SYNTHÈSE

Grâce à ce livre blanc, vous avez pu apprendre ce qu'est un ESB, les principes et les enjeux adressés par cette famille de solutions.

Nous vous avons montré la diversité et la crédibilité des offres open source sur le marché.

Nous n'avons pas souhaité vous donner un classement entre ces solutions mais des informations sur leurs principales qualités ou défauts. Les entreprises, les besoins et les capacités de celles-ci étant très différentes, il ne nous est pas possible de vous fournir la solution unique qui répondra à tous vos critères. Nous espérons tout de même que ce livre blanc vous aura apporté quelques indices.

N'oubliez pas, la première question n'est pas de choisir une solution ESB, l'important est avant tout de définir une stratégie claire de votre système d'information à l'échelle de votre entreprise. Cette stratégie doit d'abord être une vision métier des besoins actuels et des besoins futurs. Le cadre technique devra simplement permettre d'atteindre ces objectifs.

L'ESB est une boîte à outils formidable mais complexe. Si dans cette vision un besoin middleware se fait sentir, étudiez bien les différentes possibilités mises à votre disposition. En matière d'architecture, il est essentiel de savoir ajuster l'ambition au problème et les solutions les plus sophistiquées ne sont pas toujours les plus appropriées.

Depuis plusieurs années, Smile a construit une expertise des middlewares au service d'architectures extensibles et performantes et nos experts seront heureux de vous conseiller et vous aider à tirer le meilleur parti d'une solution ESB open source.

XI - REFERENCES

- 1 - <https://www.gartner.com/doc/302868/service-oriented-architectures>
- 2 - <https://www.oasis-open.org/committees/soa-rm/>
- 3 - <http://www.opengroup.org/soa/source-book/intro/>
- 4 - <http://www.eaipatterns.com/>
- 5 - http://www.ebizq.net/topics/int_sbp/features/3463.html
- 6 - <http://blogs.mulesoft.org/introducing-the-anypoint-platform/>
- 7 - <http://blogs.mulesoft.org/osgi-no-thanks/>
- 8 - <http://camel.apache.org/>
- 9 - <http://activemq.apache.org/>
- 10 - <http://cxf.apache.org/>
- 11 - <http://karaf.apache.org/>
- 12 - <http://camel.apache.org/components.html>
- 13 - <https://synapse.apache.org/userguide/mediators.html>
- 14 - <https://docs.wso2.org/display/ESB481/Mediators>
- 15 - <https://docs.wso2.org/display/IntegrationPatterns/Enterprise+Integration+Patterns+with+WSO2+ESB>

XII - GLOSSAIRE

- SLA (Service Level Agreement) : Définit la qualité de service requise entre un prestataire et un client.
- REST (Representational State Transfer) : Style d'architecture de service web basé sur la manipulation de ressources via les verbes http
- SOAP (Simple Object Access Protocol) : Protocole de requête et réponses basé sur une structure XML, et pouvant être utilisé sur différents protocoles de transport comme HTTP, SMTP ou FTP
- JMS (Java Message Service) : Interface de programmation permettant l'échange de messages asynchrones entre applications.
- Runtime : Serveur d'exécution pour des traitements informatiques
- JAR (Java ARchive) : Fichier compressé utilisé pour distribuer des classes Java.

XIII - REMERCIEMENTS

Un grand remerciement à toutes les personnes ayant travaillé sur le livre blanc :

- Grégory EVE
- Boudjema LARID
- Badr CHENTOUF
- Laury MAGNE.