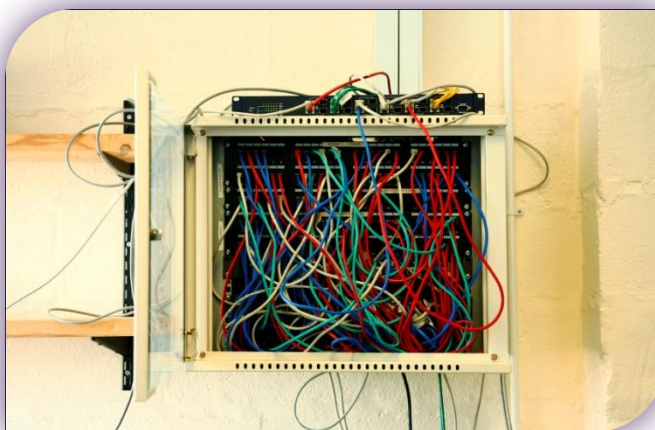


Livre blanc

Bus de Service -ESB- Nouvelle technologie pour l'intégration

Adrien Louis - Architecte - Petals Link

Novembre 2008



Version 1.4 – Novembre 2008
Pour plus d'information :
www.petalslink.com
Tél : +33 (0)5 62 73 43 80
Mail : contact@petalslink.com

Préambule

Pourquoi ce livre blanc?

L'objectif de ce livre blanc, destiné à un public large possédant une petite culture de l'intégration ou de l'architecture, est d'aborder et comprendre les technologies SOA, et particulièrement, les technologies de bus de service (ESB).

A partir d'exemple simples, vous comprendrez l'évolution des technologies d'intégrations, tel que l'EAI, ainsi que les possibilités d'architecture qui en découlent.

Petals Link

Petals Link (marque commerciale de EBM Websourcing) est éditeur de logiciels open source, spécialiste en SOA. Petals Link fournit du service autour de ses logiciels, dont le principal, Petals ESB, est un **bus de service (ESB) open-source**, qui résout les problématiques d'exploitation d'architectures décentralisées, à grande échelle. Il est complété par une offre open source de **gouvernance SOA** et de **supervision métier (BAM)**.

Quelques références

Agence Centrale des Organismes de la Sécurité Sociale (ACOSS), Direction Générale de la Modernisation de l'Etat (DGME), Orange, Alcatel-Lucent, Akerys, Académie de Toulouse, Conseil général de la Gironde, Région Limousin, Ville de Lyon, Cegedim Activ, Almerys/Orange Business Services, Thales, EADS, Service d'infrastructure de Défense (SID), Armée de l'air, Direction Générale de l'Armement (DGA)

Table des matières

Préambule.....	2
Pourquoi ce livre blanc?	2
Petals Link	2
Quelques références.....	2
Table des matières	3
Introduction	5
De l'intégration « ad hoc » aux ESB	6
L'architecture point à point.....	6
L'EAI : Une rationalisation de l'intégration sans les standards	7
Transport des données (Couche transport).....	8
Connexion entre applications (Couche broker).....	9
Transformation des données (Couche broker)	9
Orchestration des processus métiers (Couche BPM)	9
Autres services techniques	9
L'approche orientée services	10
Couplage lâche.....	11
Description d'un service	11
Granularité des services	12
Utilisation des standards	13
L'émergence des ESB.....	13
Les Bus de service, ou ESB	15
Définition.....	15
Architecture et caractéristiques.....	15
Distribution	15
Fiabilité	16
Interopérabilité et ouverture du système d'information	16
Connectivité.....	17
Services techniques.....	17
Normes et Standards	17

Etude de la mise en place d'un ESB	18
Contexte de l'entreprise.....	18
Processus de l'entreprise.....	19
Processus 1 : Réservation par téléphone à une agence.....	19
Processus 2 : Réservation par Internet	19
Processus 4 : Mise à jour des données de l'application centrale	20
Limitations.....	20
Latence du système	20
Manque de coordination entre les agences.....	21
Mise en place de l'ESB.....	21
Insertion du bus et routage des données.....	21
Transformation des données et routage.....	24
Structuration des données : ouverture et réutilisation.	25
Passage au temps réel.....	25
Bénéfice de la mise en place de l'ESB.....	27
Améliorations du SI.....	28
Organisation du SI en « services ».....	28
Annuaire des services de l'entreprise.....	30
Routage des messages basé sur le contenu.....	32
Ouverture du SI à des partenaires	33
Conclusion	35
Bibliographie.....	36
Livres	36
Articles Internet	36
Petals Link.....	37
Sites web et projets	37
Contacts	37

Introduction

Le Système d'Information (SI) d'une entreprise est généralement basé sur des logiciels et des sources de données hétérogènes, résultant de l'utilisation successive de technologies diverses, ou de l'acquisition d'autres sociétés.

Cette hétérogénéité a amené les directions informatiques à considérer la problématique d'intégration comme un enjeu majeur dès lors que les entreprises ont cherché à faire communiquer différents départements ou services entre eux pour optimiser leurs processus métiers.

Depuis le début des années 90, les solutions d'intégration se sont succédées, avec l'objectif de permettre une intégration de plus en plus fluide et de plus en plus simple. Aujourd'hui, les produits d'EAI¹, qui sont apparus à la fin des années 90, sont supplantés par les ESB² qui proposent sur une nouvelle approche basées sur les normes et les standards.

L'ESB offre une vision étendue de l'intégration des applications d'un SI, voire de toute une chaîne de production. Chaque application se connecte au bus ESB de façon indépendante avec le reste du système, et les différentes informations échangées entre les applications circulent sur ce bus.

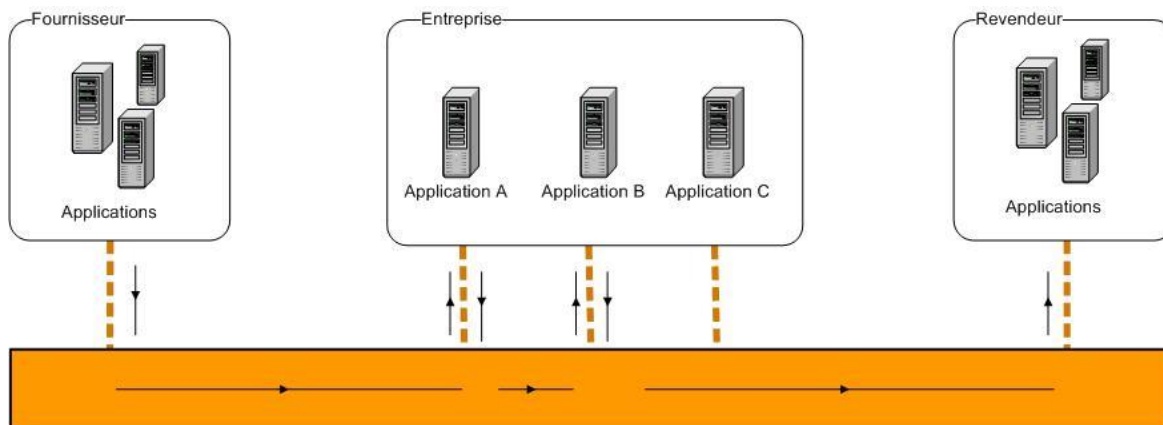


Figure 1: Echange d'informations entre applications via un bus ESB

Dans un premier temps, ce document présente l'évolution de l'intégration des Système d'Information. Ensuite, il propose une définition d'un ESB et une description des différents services qu'il doit proposer. Enfin, il expose une étude de cas de l'intégration d'un ESB dans une société.

¹ EAI : Enterprise Application Integration. Outils d'intégration s'appuyant sur des technologies propriétaires.

² ESB : Enterprise Service Bus. Outils d'intégration s'appuyant sur les standards des services Web.

De l'intégration « ad hoc » aux ESB

Du début des années 90 à aujourd'hui, on peut identifier trois approches d'architecture d'intégration successives :

l'intégration « ad hoc » mettant en œuvre de middlewares plus ou moins propriétaire sans approche méthodologique spécifique,

l'approche EAI, apparue à la fin des années 90, basée sur une approche de rationalisation des flux d'information dans l'entreprise a permis l'émergence de l'approche orientée services, mais reposait sur des technologies propriétaires de chaque éditeur qui limitaient l'interopérabilité,

L'approche ESB, qui reprend les principes de l'approche EAI, en se basant sur des standards.

L'architecture point à point

Au début des années 1990, les connecteurs et les protocoles de transport n'étaient pas basés sur des standards : ils étaient spécifiques à l'interaction entre deux applications et nécessitaient souvent des développements spécifiques. La plupart des échanges entre applications se faisaient avec des fichiers non structurés, avec tous les risques d'erreurs que cette solution basique peut engendrer.

Ce type d'intégration a donné lieu à des architectures dites « accidentelles », résultant d'un amalgame de connexions propriétaires hétérogènes, comparable à un plat de spaghettis, construites au fur et à mesure de l'intégration de nouvelles applications dans le SI.

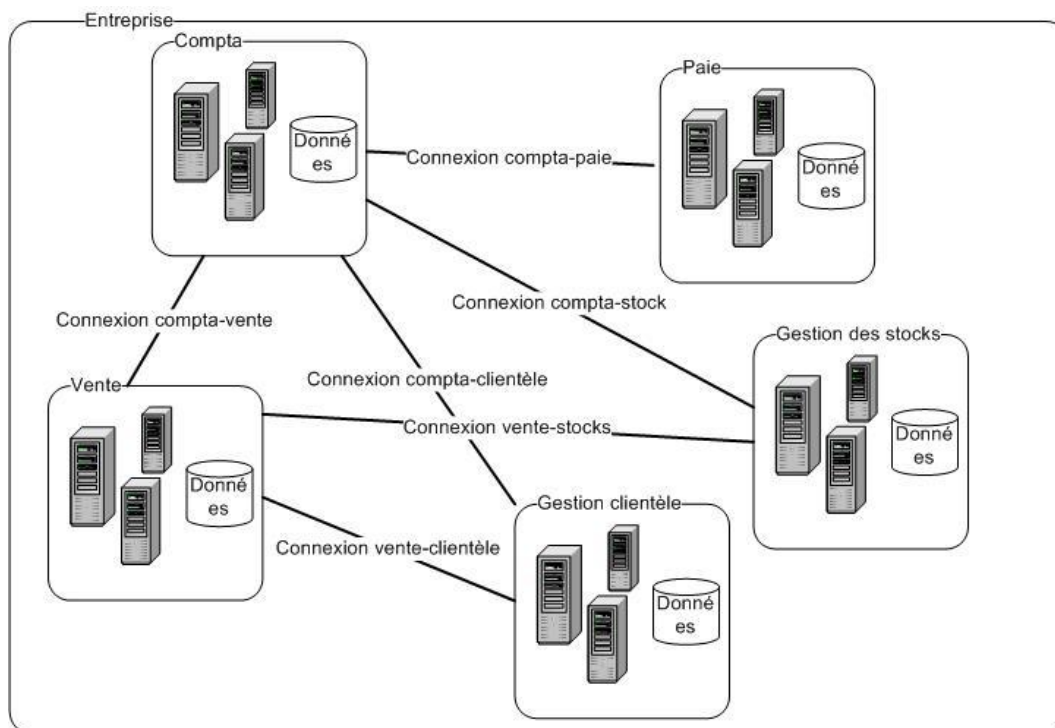


Figure 2: Architecture point à point (dite architecture accidentelle)

L'évolution d'une telle architecture est évidemment très coûteuse. En effet, l'intégration d'une nouvelle application dans ce contexte d'architecture, nécessite la réalisation d'une connexion spécifique pour chaque application interagissant avec cette nouvelle application.

De plus, il est très difficile d'avoir une vision, pour un processus métier donné, des flux d'informations qui circulent dans l'entreprise.

L'EAI : Une rationalisation de l'intégration sans les standards

Une architecture d'intégration efficace est une architecture où chaque application est connectée au « système » de façon indépendante et unique, sans avoir de connaissances a priori de la topologie du système d'information global. Une telle application obéit aux ordres du « système », en traitant des tâches et en retournant des informations.

Avec ce type d'architecture, le système d'information de l'entreprise devient agile dans le sens où il peut être modulé de façon simple, les ajouts, modifications ou suppressions d'applications n'ayant pas d'impact sur le reste du domaine.

C'était la promesse des produits d'EAI apparus au milieu des années 90. Un EAI peut être vu comme un cœur, auquel chaque application se connecte de façon indépendante.

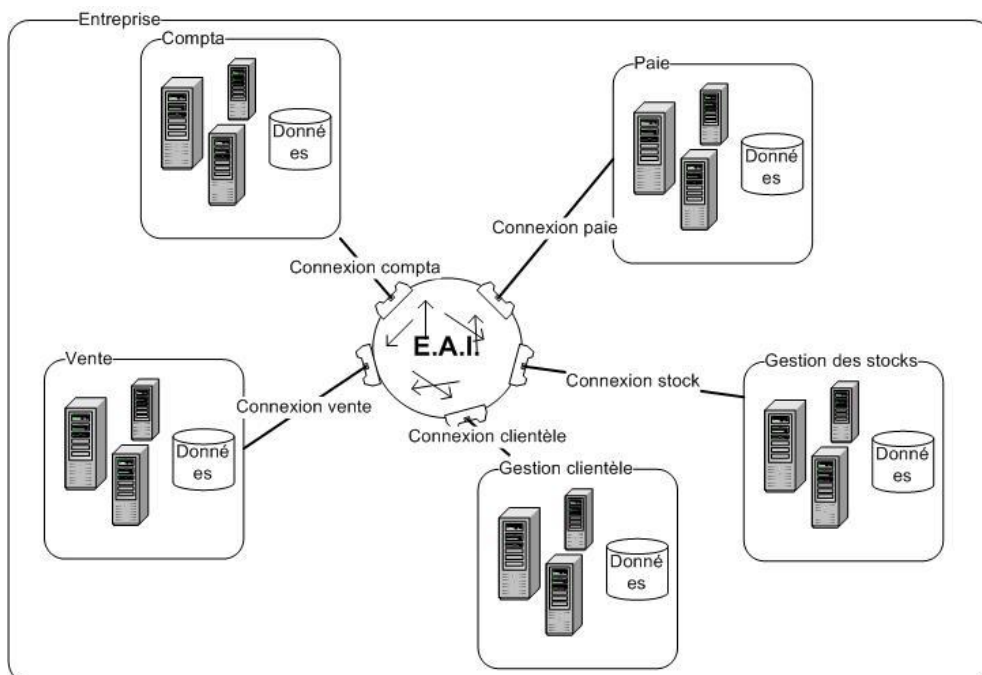


Figure 3: Architecture basée sur un outil d'EAI

Voici un schéma de principe de l'architecture d'un EAI :

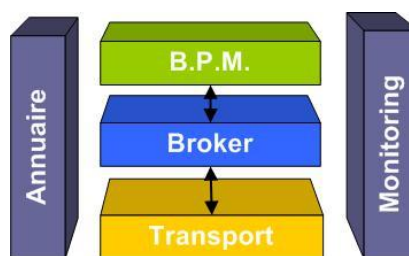


Figure 4: Agencement des différentes couches d'un EAI

L'EAI propose les fonctionnalités suivantes :

Transport des données (Couche transport)

L'EAI fournit une couche de transport homogène sur laquelle vont pouvoir s'appuyer les applications. Cette couche de transport est généralement un MOM³ (Message Oriented Middleware) propriétaire.

³ MOM : Middleware Orienté Message. Système permettant l'échange de messages asynchrones entre applications ; comparable à une messagerie.

Connexion entre applications (Couche broker)

Chaque application se connecte à l'EAI de façon indépendante, au travers d'un connecteur. L'EAI met en relation les connecteurs de deux applications données de façon interne et transparente.

Un connecteur contient toute la logique technique de connexion à l'application. Des connecteurs vers les principaux progiciels du marché sont en général disponibles auprès des éditeurs d'EAI, la réalisation d'une connexion avec une application propriétaire étant à développer spécifiquement. Le connecteur gère généralement l'authentification, les transactions, les droits d'accès, etc.

Ainsi, une application n'utilise plus qu'une seule connexion unique avec l'EAI pour communiquer avec le reste des applications.

Transformation des données (Couche broker)

Les données transmises d'une application à une autre ne sont pas comprises de la même façon, et n'ont que rarement la même structure. L'adresse d'un client sera par exemple représentée sur un seul champ dans une application, mais sera séparée en plusieurs champs dans une autre (rue, ville, code postal, ...).

Orchestration des processus métiers (Couche BPM)

Les outils d'EAI sont généralement couplés à des outils de BPM ⁴(Business Process Management) qui automatisent les processus de l'entreprise. On définit alors les échanges entre les différents départements de l'entreprise. Par exemple, la prise d'une commande d'un client entraîne une communication entre le service de commande, de gestion des stocks, de facturation, etc.

Autres services techniques

L'EAI fournit aussi d'autres services transversaux comme le monitoring des données, un référentiel des applications connectées et des informations manipulées par ces applications.

L'EAI simplifie l'architecture du SI d'une entreprise, et fluidifie les échanges d'informations.

Cependant, le principal reproche que l'on peut cependant faire aux outils d'EAI est leur aspect propriétaire.

La logique d'intégration d'un EAI étant propriétaire, les éléments de l'outil (connecteurs, transformateurs de données, orchestration des processus) ne sont

⁴ BPM : Business Process Management. Analyse et modélisation logicielle des procédures mises en place par l'entreprise pour réaliser ses activités

pas standardisés, ce qui lie l'entreprise à l'éditeur qu'elle aura choisi, avec les conséquences que l'on connaît (coût du conseil et des interventions, pérennité de la solution, etc.)

Chaque EAI ayant sa propre plateforme d'intégration, la communication entre deux entreprises utilisant des EAI différents repose sur le support de standards d'interopérabilité. De plus en plus, les éditeurs d'EAI offrent des connecteurs standardisés, en utilisant principalement des technologies comme SOAP⁵ (Simple Object Access Protocol) ou les services web⁶.

Bien souvent, les produits d'EAI sont mis en œuvre dans des architectures de type hub and spoke⁷, en centralisant toute la logique d'intégration. Ceci introduit un « single point of failure » dans le système d'information, dans le sens où si la plateforme de l'EAI s'arrête, plus aucune application ne peut communiquer, et tous les processus métiers sont stoppés. Ceci oblige à concevoir des solutions EAI hautement disponibles, avec un coût d'autant plus important.

L'approche orientée services

La notion de SOA (Service Oriented Architecture) définit un style d'architecture reposant sur l'assemblage de services proposés par les applications. Dans ce style d'architecture, les différents composants logiciels sont connectés par un couplage lâche.

Un « service », au sens de la SOA, est une connexion à une application, offrant l'accès à certaines de ses fonctionnalités. Les fonctions proposées par un service peuvent être des traitements, des recherches d'informations, etc. Par exemple, une application de gestion de clientèle peut par exemple offrir un service retournant les coordonnées (adresse, tél, ...) d'un client.

Dans une architecture de services, chaque élément applicatif doit fournir tout ou partie de ses fonctionnalités sous formes de services appelables par d'autres applications. Pour fournir ses services, un élément applicatif peut utiliser des services proposés par d'autres applications, pouvant être issues de technologies hétérogènes.

Le développement d'une nouvelle application peut se focaliser sur la réalisation de nouvelles fonctionnalités pour le système d'information, et réutiliser

⁵ SOAP: Simple Object Access Protocol. Protocole fondé sur XML pour l'échange d'informations en environnement décentralisé.

⁶ Web Services. Services (ensemble d'opérations) accessible via le web, reposant sur le protocole standard http.

⁷ Hub and Spoke : concentrateur et rayon. Architecture de réseau mettant en œuvre un point de connexion central. A partir duquel on peut atteindre chacune des terminaisons situées à la périphérie.

des fonctionnalités déjà proposées par d'autres applications.

Dans une SOA, la notion de référentiel de services est fondamentale, car pour pouvoir réutiliser les services, il faut connaître à la fois leur existence et leurs caractéristiques.

Une approche SOA permet de réaliser facilement l'orchestration d'un processus métier par l'assemblage de différents services, en utilisant un outil de BPM. L'homogénéité des interfaces des services assurent encore ici une intégration simple avec un outil de BPM et la réalisation rapide de nouveaux processus.

Globalement, « l'approche SOA » est l'union de :

- Une méthodologie pour identifier et concevoir des applications comme des assemblages de services,
- Un ensemble d'outils et d'infrastructures pour faciliter la création de ces services et leur utilisation,
- Des patterns de construction de services.

Couplage lâche

La SOA vise l'indépendance maximale des services. Un service doit être au minimum lié à la structure d'un autre et doit imposer le minimum de contraintes à un service l'utilisant également.

Il est alors possible de définir rapidement et facilement de nouveaux processus métiers, par assemblage de services existants, et permettre ainsi à l'entreprise de réagir rapidement aux évolutions de leur contexte de marché.

Pour réaliser ce couplage faible, l'effort de conception de l'architecture doit être porté sur la normalisation des interfaces des services, définissant une manière de communiquer la plus souple et la plus riche possible. Cette normalisation entraîne également une indépendance entre cette couche de communication et la couche de transport effective.

Description d'un service

Un service propose une interface de distribution basée sur un ensemble de composants métier, eux-mêmes constitués d'objets implémentant des méthodes de bas niveau.

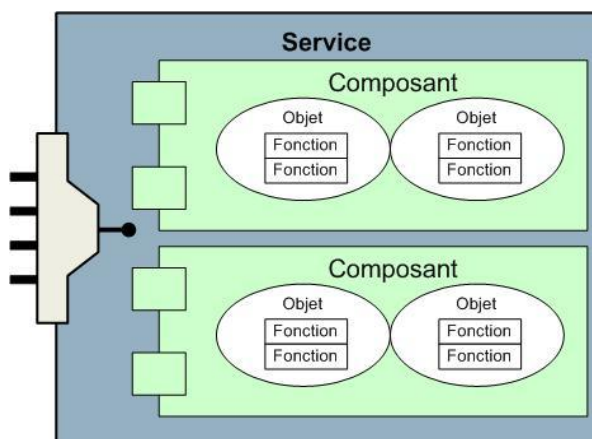


Figure 5 : Composition d'un service

Voici les différentes responsabilités associées à chaque niveau :

- Le niveau objet ou classe

Le **niveau Object/Class** représente la réalisation proprement dite des traitements à réaliser par le service.

- Le niveau composant

Le **niveau « Composant »** correspond au modèle de composants proposé par la plateforme utilisée pour le développement. Par exemple, dans le monde J2EE, il s'agit du modèle EJBs qui est le modèle de composant du langage Java. Le rôle d'un tel modèle de composants est de proposer des facilités de packaging et des services techniques comme la connexion aux bases de données, la gestion des transactions ou de la sécurité.

Publication de services : Distribution, et accès distant aux services

Le **niveau « Service »** correspond à la problématique de mise à disposition de services sur le réseau d'entreprise, c'est-à-dire à l'utilisation d'un protocole de communication pour accéder aux composants depuis des applications situées sur d'autres machines. Ce niveau correspond principalement à des préoccupations techniques (protocole utilisé, génération automatique des proxies, ...) et organisationnelles (annuaires des services, contrats de services ...). Techniquement, ce niveau peut être réduit au minimum si on utilise les mécanismes de distribution fournis avec le modèle de composants (par exemple RMI pour les EJBs). Cependant, il peut nécessiter un certain nombre de traitements, dont le développement peut être automatisé dans certains cas, si on utilise une autre technologie pour la distribution (Web Services ou MOM).

Granularité des services

Un enjeu de la SOA est de trouver la bonne granularité des services proposés par une application. En effet, un service à granularité trop fine n'offre que peu d'intérêt au niveau métier. Des services renvoyant uniquement le nom d'un client, ou juste son adresse n'ont guère de valeur ajoutée.

Des services à granularité plus forte, créés à partir de plusieurs composants structurés, offrent plus d'intérêt dans la réalisation d'un processus global. Cependant, un service à granularité trop forte et réalisant un processus métier complet sera peu intéressant dans la mesure où il sera difficilement réutilisable pour la construction d'autres processus.

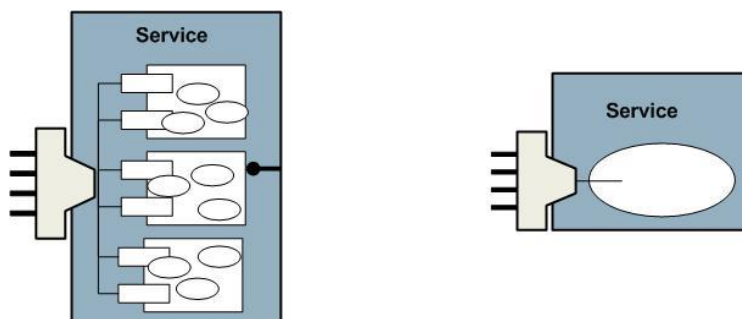


Figure 6 : Service à forte granularité versus service à faible granularité

Utilisation des standards

Les problématiques de couplage lâche et d'homogénéité entre les services, aspects inhérents à la SOA, amènent à définir des standards (internes à l'entreprise ou supportés par un organisme de normalisation). L'utilisation de normes n'est pas imposée par la SOA, mais une entreprise ayant en perspective de s'ouvrir à l'extérieur ou d'intégrer de nouvelles applications diminuera considérablement ses coûts d'intégrations en utilisant des standards reconnus du marché et supportés par une majorité d'outils.

Des exemples de tels standards sont le protocole SOAP, basé lui-même sur le langage standardisé XML et le protocole de transport http.

Ainsi, les WebServices sont de bons candidats à la représentation technique des services de l'entreprise. Basés sur SOAP, ils correspondent également à la notion de services métiers auxquels fait référence la SOA, et sont également standardisés.

L'autre atout des services web est qu'ils s'accompagnent d'une multitude de standards répondant aux exigences de sécurité, de transactions, d'authentification, etc.

Ces normes, rassemblées sous l'appellation générique WS-*, ne sont toutefois pas toutes adoptées définitivement, certaines problématiques donnant parfois lieu à l'émergence de plusieurs normes concurrentes (comme WS-Reliability et WS-Reliable Messaging, pour standardiser le protocole fiable d'échange de messages, par exemple).

L'émergence des ESB

Les standards J2EE et Web Services ont profondément modifié le paysage de l'intégration. Ainsi, de nouveaux produits basés sur ces standards émergent depuis

deux ans sous le nom d'Enterprise Service Bus.

Ces produits peuvent être vus comme des supports à une implémentation concrète d'une SOA, et sont basés principalement sur des standards comme XML et les WebServices.

Les ESB reprennent les grands principes de l'EAI, mais l'utilisation poussée de standards rend leur coût de licence beaucoup plus abordable.

Comme avec les EAI, l'intégration des applications du SI d'une entreprise au bus ESB peut se faire de façon incrémentale.

La réalisation de la communication avec les partenaires de l'entreprise en mode B2B⁸ (Business to Business) se trouve simplifiée grâce à la mise en œuvre de standards reconnus par tous.

La section suivante détaille les caractéristiques de ces ESB.

⁸ B2B : Business To Business. Echanges ou transactions commerciales effectuées d'entreprise à entreprise.

Les Bus de service, ou ESB

Définition

Un « Enterprise Service Bus » est une solution d'intégration implémentant une architecture totalement distribuée, et fournissant des services comme la transformation des données ou le routage basé sur le contenu (CBR), ainsi qu'une interopérabilité accrue par l'utilisation systématique des standards comme XML, les Web Services et les normes WS-*⁹.

L'ESB est une solution packagée qui permet de mettre en œuvre la SOA.

Architecture et caractéristiques

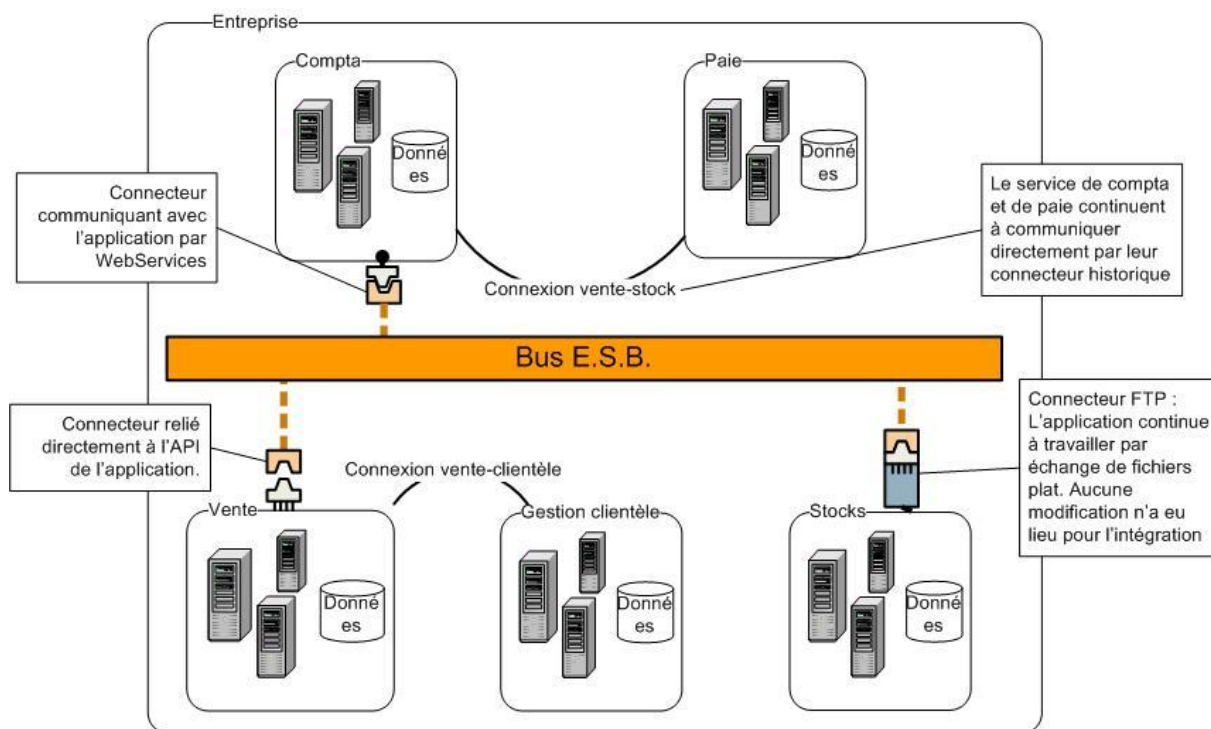


Figure 7 : Architecture du SI d'une entreprise bâtie autour d'un ESB

Distribution

La notion de distribution est centrale pour un ESB. En effet, par essence les applications à intégrer sont réparties sur différentes machines ou systèmes d'informations.

⁹ WS-* : Ensemble de normes associées aux Web Services et traitant par exemple la sécurité ou l'orchestration de services.

Par la mise en œuvre de ce principe de distribution, le « bus » de l'ESB peut devenir virtuel, les données de configuration et d'administration étant alors distribuées sur les extrémités de l'ESB, c'est-à-dire au plus près des applications à intégrer. Ceci permet de contourner les problèmes l'architecture « hub and spoke » proposée classiquement par les solutions EAI, et amène à des architectures sans SPOF¹⁰.

Fiabilité

Comme toute solution d'entreprise, un ESB doit apporter des garanties de fiabilités. Dans cette optique, la plupart des ESBs sont construits sur des MOM, et tous permettent de d'utiliser un MOM comme moyen de transport. L'utilisation d'un MOM permet de garantir dans certaines configurations que les messages sont bien transmis une et une seule fois.

L'autre point important concernant la fiabilité est la possibilité de construire des architectures sans SPOF comme décrit ci-dessus. Ainsi, quand un serveur tombe, le reste du système peut continuer à fonctionner.

Interopérabilité et ouverture du système d'information

Les ESB sont basés sur des standards reconnus, ce qui facilite leur interopérabilité et l'interconnexion des SI de deux entreprises partenaires utilisant des ESB, même différents.

Les messages circulant dans un bus peuvent être transmis au bus du partenaire. A terme, les aspects techniques de transaction, sécurité, etc. seront supportés complètement par la mise en œuvre des normes WS-* spécifiques à chaque problématique.

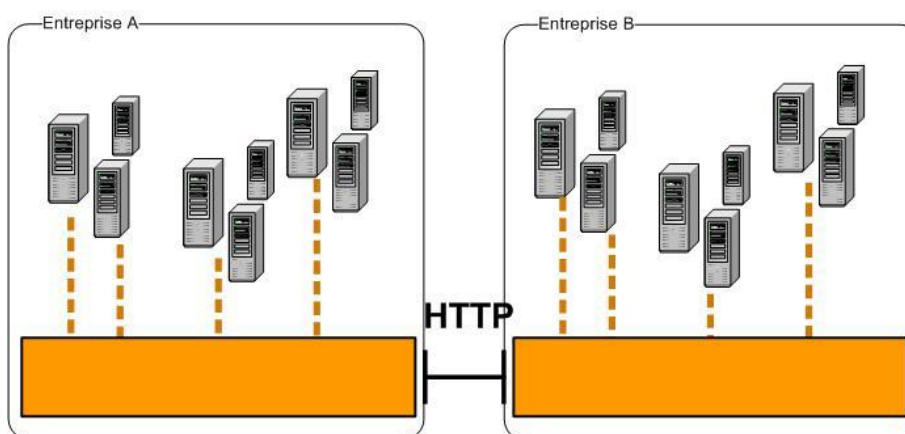


Figure 8 : Deux entreprises reliées par leurs bus ESB

¹⁰ SPOF ; Single Point Of Failure : point unique par lequel passent tous les traitements et qui paralysent l système en cas de panne.

Connectivité

Pour connecter les différentes ressources applicatives à intégrer, l'ESB propose un ensemble de connecteurs basés sur la norme J2CA.

Ainsi, tous les ESB proposent des connecteurs techniques vers la plupart des formats techniques d'échange : fichier CSV, formats XML, RMI, RPC, ...

Certaines offres ESB incluent des connecteurs « métiers », permettant d'intégrer des progiciels du marché. Dans tous les cas, l'utilisation de J2CA permet d'utiliser des connecteurs fournis par les éditeurs de logiciels eux-mêmes selon ce standard.

Services techniques

Une offre ESB doit offrir des services techniques comme la transformation des messages, le routage basé sur le contenu, et éventuellement l'orchestration des services.

Ces services techniques proposés par les ESB permettent de mettre en œuvre des intégrations par couplage faible, c'est-à-dire qu'une application n'a pas à s'adapter aux formats ou aux spécificités des applications qu'elle intègre via l'ESB : les adaptations éventuelles sont traitées au niveau de l'ESB.

Normes et Standards

Les normes pertinentes dans le cadre des ESB sont dans trois domaines :

Standards W3C : Il s'agit des standards relatifs aux Web Services comme XML, SOAP, WSDL. On peut aussi citer la note « Web Service Architecture » qui explicite nombre de concepts mis en œuvre dans les ESB.

Standards OASIS : Normes WS-* comme WS-Security, WS Addressing, ... ainsi que les normes BPEL et UDDI

JSR Java : J2CA et JBI

Etude de la mise en place d'un ESB

Contexte de l'entreprise

Ce chapitre traite un cas d'étude de mise en place d'un ESB au sein d'une société de location de véhicules.

Cette société est constituée d'un siège et d'agences réparties géographiquement.

Pour le cas d'étude, seule la problématique de réservation de véhicule est abordée. La gestion des classes de véhicules et les sur classements ne sont pas gérés. Une agence gère un nombre de véhicules restreint.

Voici quelques caractéristiques de ce cas d'école :

Autonomie des agences

Historiquement, le siège ne gérait que les services « Corporate » (facturation, paie, contrats entreprise, politique commerciale, communication, ...).

Les réservations s'effectuaient uniquement dans les agences qui géraient leur parc de véhicules de façon indépendante.

Site de réservation en ligne

Suite à l'avènement d'Internet, la société s'est dotée d'un site permettant la réservation en ligne. Ce site est hébergé au siège, et utilise une base de données regroupant l'état du parc de véhicules de toutes les agences. Ceci a nécessité la mise en place d'un protocole de communication entre les agences et le siège.

Ainsi, à la fin d'une journée, chaque responsable d'agence lance un batch qui extrait les données de son système locale, les placent dans un fichier plat, et envoie ce fichier au siège par une connexion FTP. Une application située au siège lit tous les fichiers reçus en fin de journée, et met à jour la base de données centrale.

Lorsqu'une réservation est effectuée par Internet, le système de réservation en ligne envoie automatiquement un e-mail à l'agence concernée avec un récapitulatif de la réservation, afin que l'agence puisse effectivement bloquer le véhicule pour la période de réservation spécifiée.

Processus de l'entreprise

Le SI de l'entreprise peut se représenter comme suit :

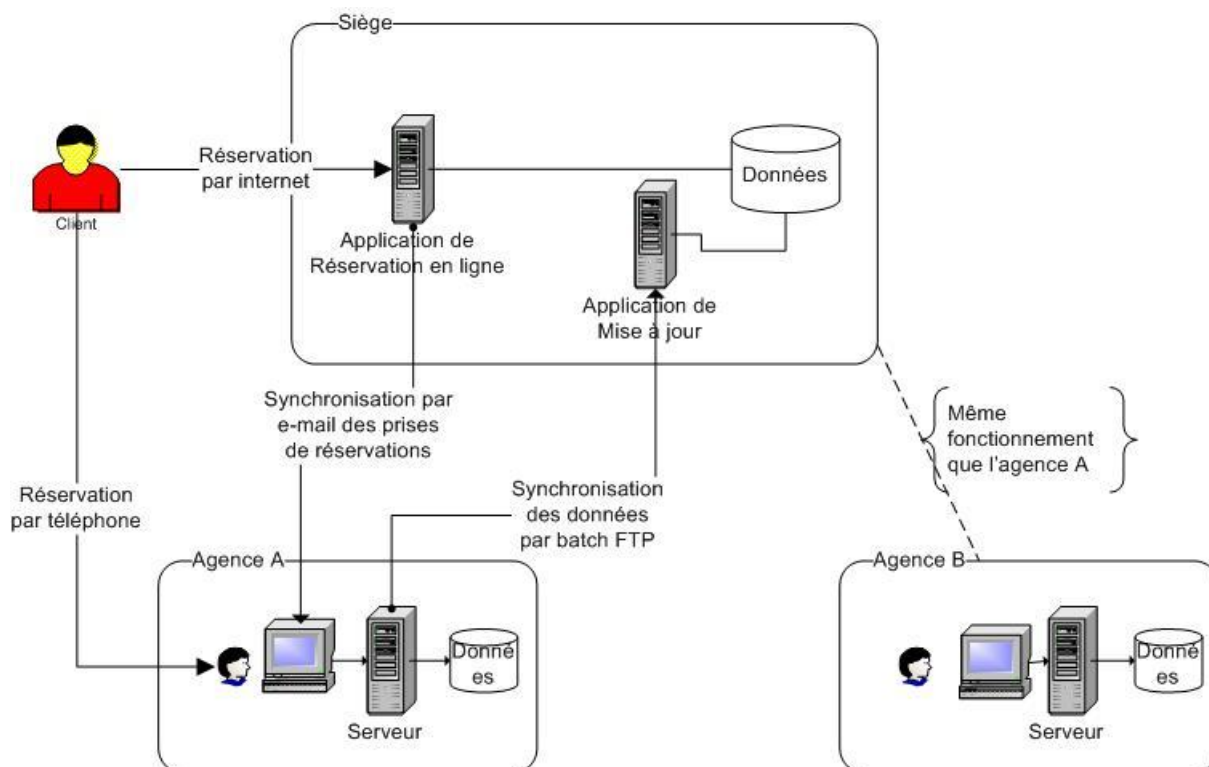


Figure 9 : Système d'Information de la société de location

Les principaux processus représentatifs pour le cas d'étude sont les suivants :

Processus 1 : Réservation par téléphone à une agence

Un client contacte par téléphone une agence et demande la réservation d'un véhicule pour une période donnée.

L'agent de réservation vérifie grâce à son application locale qu'un véhicule est disponible et enregistre la réservation dans ce cas.

Processus 2 : Réservation par Internet

Un client se connecte au site pour effectuer une réservation pour une période donnée dans une ville donnée.

L'application interroge la base de données centrale pour vérifier s'il existe un véhicule disponible pour la période dans les agences de la ville mentionnée.

Si la réservation est possible, le système envoie automatiquement un e-mail au responsable de l'agence concernée en précisant la période de réservation et les coordonnées du client. Le responsable d'agence enregistre cette réservation dans son application.

Processus 3 : Enlèvement et restitution d'un véhicule

Un client loue un véhicule dans une agence et règle le montant de la location.

Il doit rendre le véhicule dans cette même agence, qui est la seule à détenir le dossier de réservation. Le client paie éventuellement des frais supplémentaires (dépassement du kilométrage ou autre).

Processus 4 : Mise à jour des données de l'application centrale

A la fin d'une journée, le responsable d'agence lance un batch d'export des données de son application.

Un outil extrait de la base de données toutes les informations de l'agence et les placent dans un fichier plat (informations de réservations et de paiement de location).

Le fichier est ensuite envoyé à l'application centrale via une connexion FTP. L'application centrale lit tous les fichiers reçus en fin de journée, et met à jour la base de données centrale.

Limitations

Latence du système

Les mises à jour de l'état du parc se font par batch journalier, en fin de journée.

Au niveau du siège, un processus nocturne lit tous les fichiers reçus en fin de journée, et les remonte dans la base de données centrale.

Cette technique d'intégration est proche d'un ETL¹¹ (Extract, Transform and Load).

Avec ce mode de fonctionnement, la maison mère connaît l'état du parc de véhicules avec un jour de délai.

Deux cas de figures se présentent lorsqu'un client souhaite réserver par Internet un véhicule dans une agence du groupe:

L'application de réservation par Internet accepte la réservation, en se basant sur les données de la veille, et un e-mail est automatiquement envoyé à l'agence concernée. Dans le même temps, des réservations se font directement à l'agence concernée et plus aucun véhicule n'est disponible.

¹¹ ETL : Extract, Transform and Load. Outil chargé d'extraire les données dans différentes sources, de les transformer et de les charger dans un entrepôt de données

L'agence devra alors soit trouver un véhicule de remplacement dans une autre agence, soit annuler la réservation du client, cette deuxième solution ayant pour effet une perte de clientèle à terme, et donc de chiffre d'affaire.

L'application de réservation par Internet refuse la réservation par manque de disponibilité de véhicule dans cette agence donnée. Dans le même temps, un client restitue par anticipation son véhicule à l'agence, ou annule sa réservation directement à l'agence. La compagnie perd alors une réservation, car un véhicule est disponible mais l'application de réservation par Internet n'en a pas connaissance. Ceci entraîne également une perte de chiffre d'affaire.

Manque de coordination entre les agences

Un client se présente dans une agence pour louer un véhicule mais l'agence n'a plus de véhicule disponible. Cependant, il existe peut être une agence proche qui pourrait prêter un véhicule.

L'agence ne connaît pas d'autres agences, le client repart sans avoir pu réserver son véhicule, insatisfait. Le chiffre d'affaire et l'image de la société en sont affectés.

Le responsable de l'agence connaît une autre agence proche, et essaie de la contacter par téléphone afin de demander si un véhicule est disponible. L'agence n'est pas disponible à ce moment là, le client repart sans avoir pu effectuer sa réservation, insatisfait. Le chiffre d'affaire et l'image de la société en sont affectés.

Un autre problème lié est qu'un client doit ramener son véhicule dans l'agence où il l'a emprunté, car les dossiers de réservations ne sont physiquement pas accessibles d'une agence à l'autre.

Mise en place de l'ESB

La mise en place du bus ESB se fait progressivement dans le groupe, de manière incrémentale. Il n'est ainsi pas nécessaire de bloquer tout le SI pendant la période de transition.

Insertion du bus et routage des données

Le remplacement du transfert de fichiers par FTP est la première étape de la mise en place du bus.

Le mode d'envoi des données des agences vers le siège via une connexion FTP est le suivant :

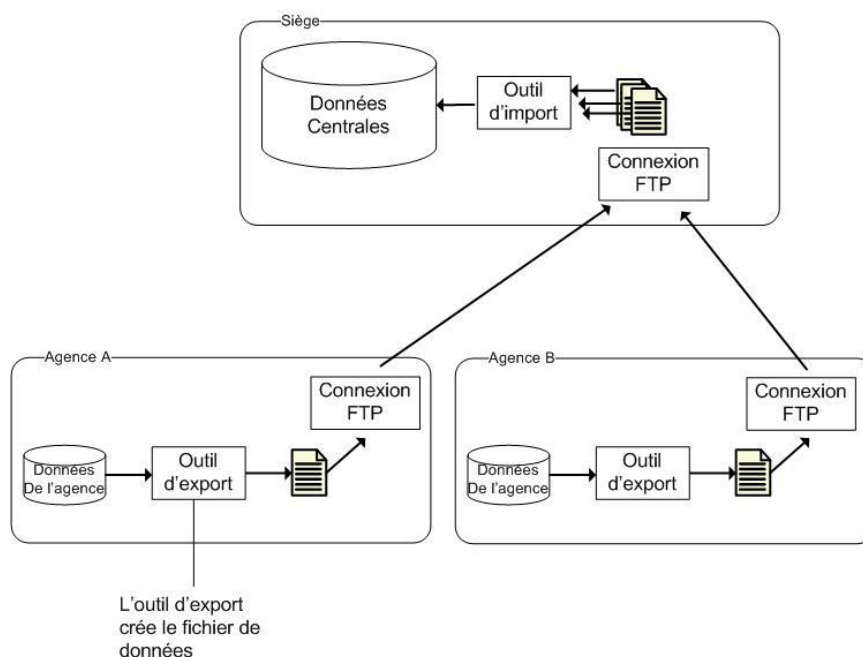


Figure 10 : Envoi des données par batch journalier via FTP

Le bus va remplacer la connexion FTP et devenir le nouveau moyen de transport des fichiers de données envoyés depuis les agences vers le siège de la société.

Un service ESB standard de transfert de fichier est mis en place dans chaque agence.

Ce service va lire le fichier produit par l'outil d'export des données. Le fichier est empaqueté dans un message qui va être envoyé au destinataire, le siège de la société.

Au niveau du siège de la société, un service ESB standard reçoit les messages envoyés par les agences, en extrait les fichiers, et copie ces derniers dans le répertoire habituel où le serveur FTP copiait les fichiers.

L'application d'import de ces fichiers continue de fonctionner normalement, en analysant toutes les nuits le contenu du répertoire et en remontant dans la base centrale les données issues des nouveaux fichiers.

La Figure 11 illustre ce nouveau fonctionnement.

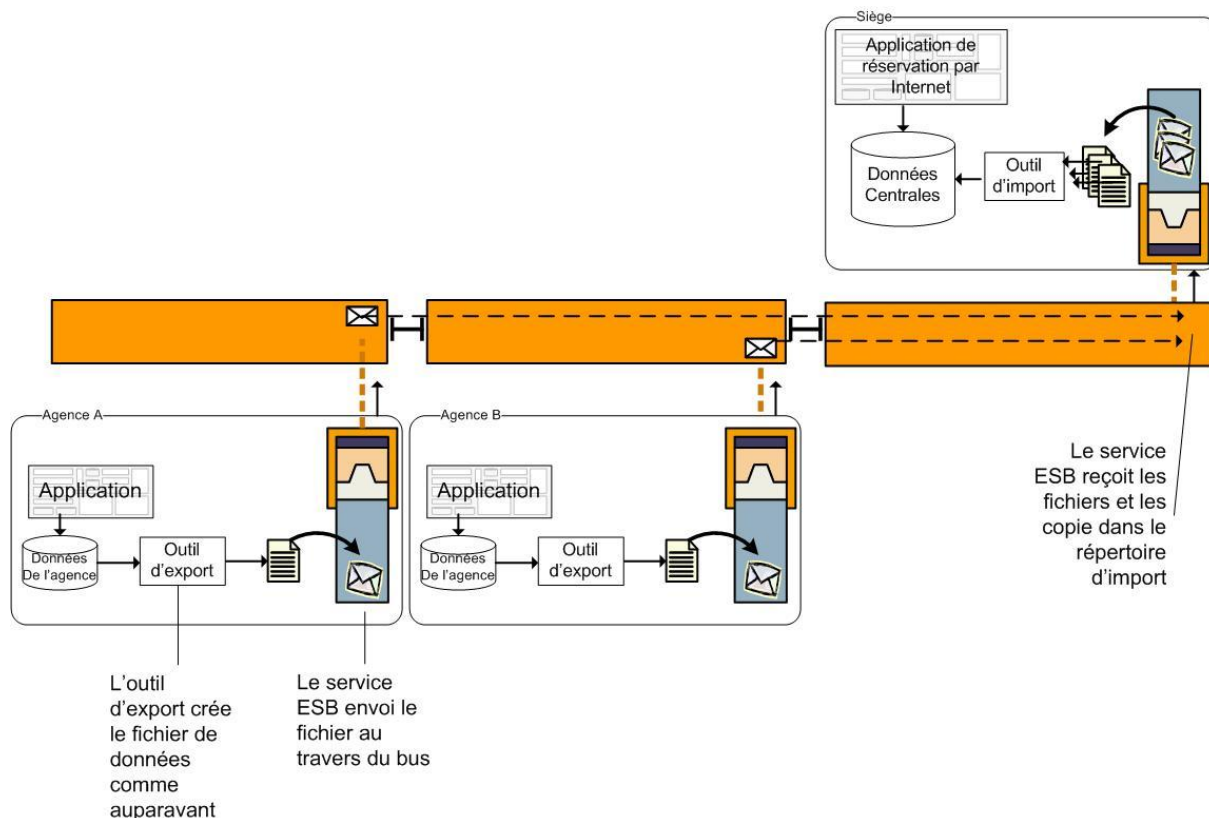


Figure 12 : Envoi des données par batch journalier en utilisant le bus

Pour les applications des agences ou de l'application de mise à jour des données du siège, il n'y a aucun changement ; l'insertion du bus s'est faite de façon transparente. Cependant, ceci constitue déjà une amélioration de la communication :

Garantie de délivrance des données.

Une des qualités de services qu'offre un ESB est la garantie de délivrance des données envoyées (qualité de service inhérente à l'utilisation d'un MOM).

Si un problème technique nécessite le redémarrage du bus, ou si un destinataire (ici, le siège de la société) n'est pas joignable momentanément (suite à un arrêt de machine par exemple), les messages non délivrés seront envoyés de nouveau (un système de sauvegarde étant intégré au bus).

Une connexion FTP interrompue pendant le transfert de données auraient nécessité un nouvel envoi des données et donc un système de surveillance du transfert de fichier.

Sécurité du transport des données.

Lors des envois effectués d'une agence au siège, les données transitent par Internet. La connexion FTP initiale est vulnérable, les informations circulent en clair et peuvent être interceptées.

Le bus garantit un fort niveau sécurité pour les données, encryptant ces dernières et les rendant illisibles.

Transformation des données et routage

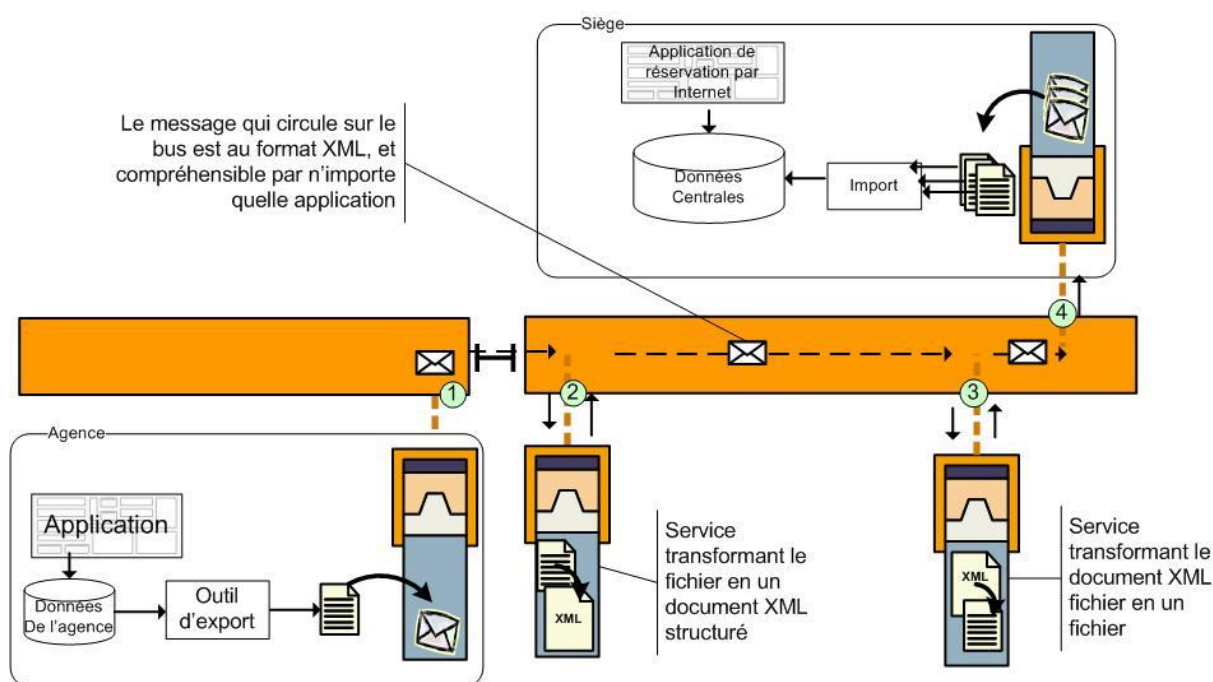
Les données traversant le bus sont toujours représentées par le fichier généré par l'outil d'export. La transformation de la représentation de ces données sous une forme XML, plus structurée et plus lisible, va permettre de pouvoir partager ces informations sur le bus.

La structure du fichier généré par l'outil d'export n'est effectivement compréhensible que par l'outil d'import, situé au siège de la société.

Dans l'optique de rendre ces informations utilisables par d'autres applications (notamment les autres agences), il faut les présenter sous une forme standard et compréhensible.

Encore une fois, il n'est pas nécessaire de modifier les applications, la transformation des données va se faire dans le bus. Un service de transformation s'intercale dans le processus d'envoi des messages au siège. Ce service est configuré pour reconnaître la structure du fichier généré par l'outil d'export et générer un autre fichier mieux structuré.

Les données transitant dans le bus ne sont donc plus envoyées directement à leur destinataire, mais vont suivre un circuit, transitant par différents points de l'ESB, jusqu'à atteindre finalement le siège. Ce circuit est défini directement dans le message contenant les données.



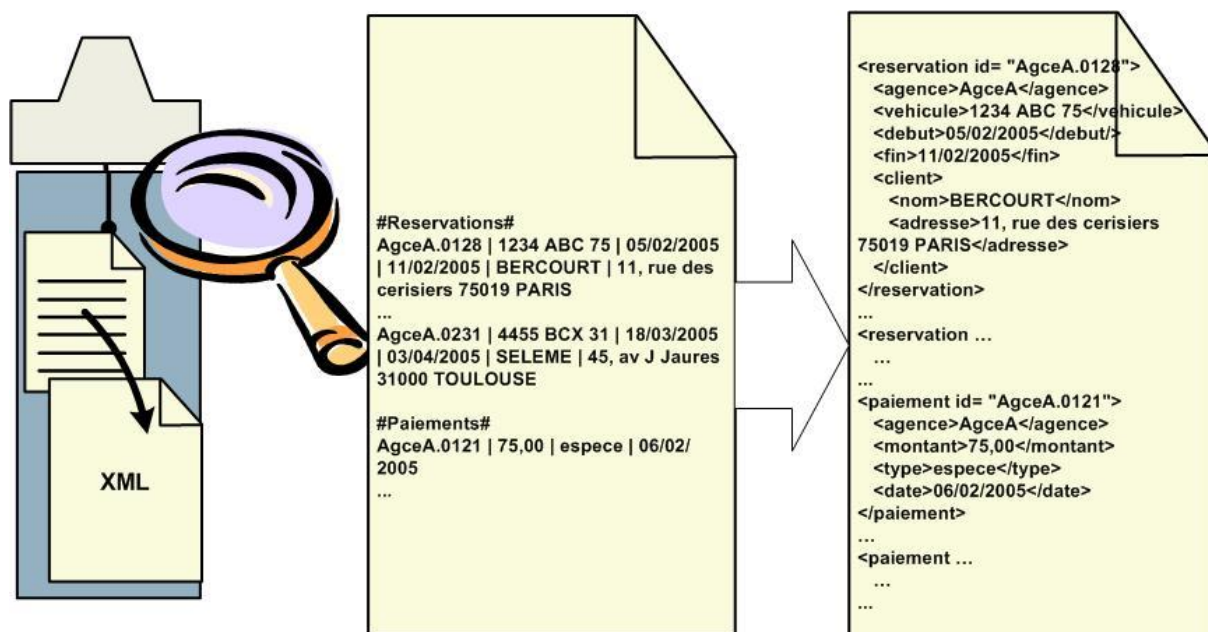


Figure 14 : Transformation des données en XML.

Structuration des données : ouverture et réutilisation.

Comme le montre le schéma ci-dessus, chaque information de réservation ou de paiement est maintenant empaquetée dans un élément (élément de réservation et élément de paiement), et présentée sous une forme compréhensible et élégante.

L'organisation des données dans un document XML structuré peut permettre, maintenant, de traiter chaque information séparément.

Dans l'optique de séparer le traitement des informations de réservations et de paiement, il est possible d'imaginer un autre service, situé après le service de transformation du fichier en document XML, qui enverrait chaque élément XML de réservation vers une première application, et chaque élément XML de paiement vers une autre application.

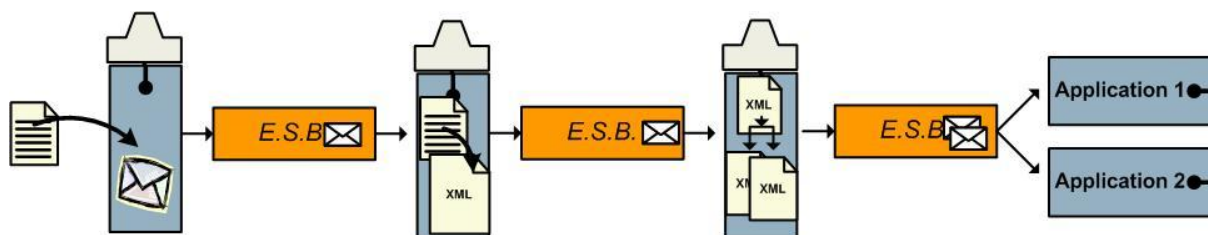


Figure 15 : Transformation et routage de données suivant la nature des informations.

Passage au temps réel

Le mode de transfert d'information est pour le moment un transfert en bloc.

L'état complet du système de chaque agence est envoyé chaque jour au siège.

Un système dont les informations sont partagées en temps réel adopte une philosophie différente. Chaque nouvelle information est instantanément diffusée, au travers de services.

L'application utilisée dans les agences et l'application de mise à jour des données du siège doivent maintenant être modifiées, dans un cas pour émettre un message dès qu'une réservation ou un règlement est effectué, dans l'autre pour mettre à jour la base de données du siège dès qu'un message arrive.

La connexion entre le système de l'agence et le bus est déplacée. Le bus ne se connecte plus derrière l'outil d'export, mais directement sur l'application de l'agence.

Une adaptation doit être ici réalisée afin de pouvoir envoyer un message sur le bus dès qu'une réservation ou un paiement est enregistré dans le système de l'agence.

Un adaptateur de type JCA¹² est utilisé pour réaliser la connexion entre l'application et l'ESB.

Cet adaptateur permet l'envoi des informations de réservations ou de paiement sur le bus ESB, ces informations seront directement des éléments XML de type réservation ou paiement, comme vu précédemment.

Pour l'application de mise à jour des données du siège, un adaptateur sera également réalisé. Les messages reçus contenant des informations au format XML seront interprétés et transmis à l'application de mise à jour via cet adaptateur ; chaque message reçu générant alors instantanément une mise à jour des données du siège.

Le déploiement de ces adaptateurs sur la globalité du SI de la société peut se faire de manière progressive. Certaines agences peuvent continuer à travailler par batch journalier, le siège recevant alors en une seule fois la totalité des informations de ces agences.

Pour les autres agences où l'adaptateur a été installé, les informations sont remontées immédiatement, permettant à l'application de réservation par Internet d'être au plus juste de la réalité du parc de véhicules.

¹² Java 2 Connector Architecture

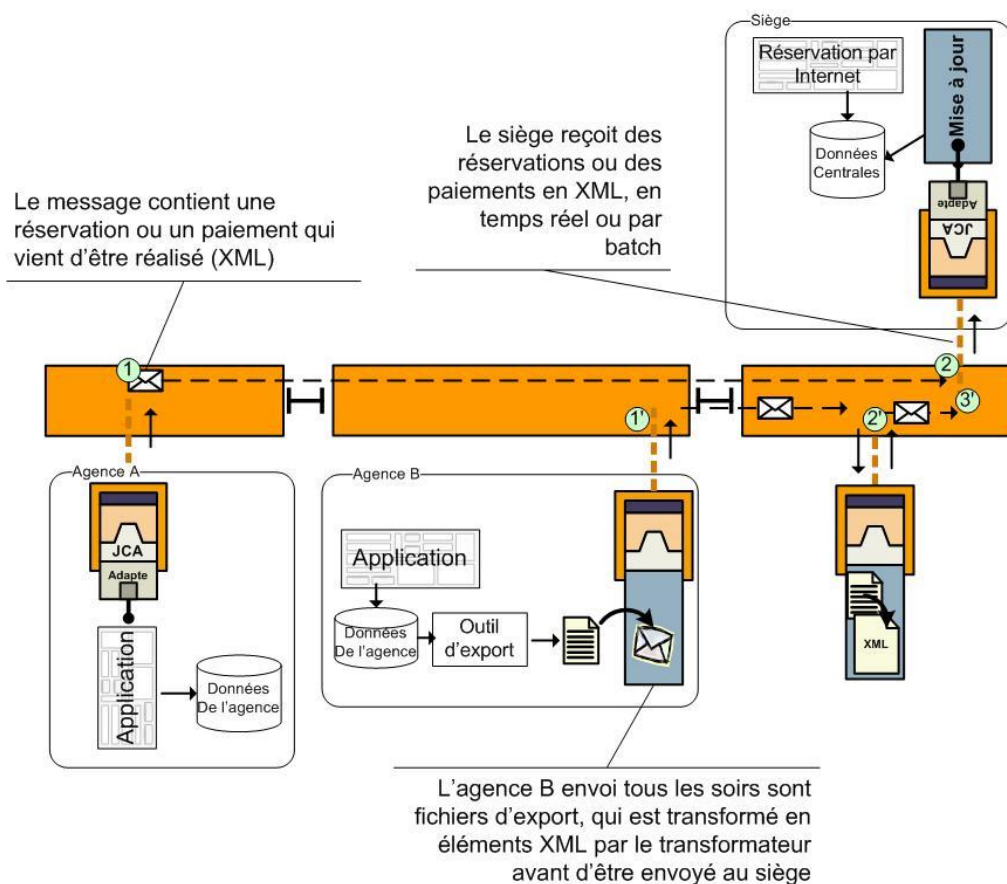


Figure 16 : Passage progressif au temps réel

Accès en temps réel aux informations.

La base de données centrale de la société est maintenant mise à jour dès qu'une réservation est effectuée dans une agence. Ceci permet à l'application de réservation par Internet d'être plus fiable quant aux disponibilités des véhicules proposés sur le site.

Les erreurs dues à la latence de l'ancien système (réservations acceptées alors qu'il n'y a plus de véhicule disponible, réservations refusées alors que des véhicules sont disponibles) ont disparues.

Bénéfice de la mise en place de l'ESB

A ce stade de la mutation de l'architecture du SI de la société, les échanges d'informations entre les agences et la base de données centrale sont instantanés et sûrs. Les données sont également sécurisées par le bus.

La garantie de délivrance des informations est assurée par l'ESB, si une panne intervient par exemple au niveau du siège, les informations envoyées par les agences restent en transit sur le bus, et seront délivrées à la base de données celle-ci sera à nouveau disponible.

Ces informations sont délivrées en temps réel, ce qui permet d'offrir au site de réservation par Internet des données réelles.

Les informations sont représentées sous une forme standardisée, et peuvent être compréhensible par n'importe quelle application de l'entreprise.

Le SI est prêt maintenant à évoluer et à s'ouvrir, grâce au socle robuste et ouvert qu'est l'ESB, permettant ainsi à la société de gagner en qualité sur les services qu'elle propose et d'élargir son offre à moindre coût. Ces aspects sont présentés dans la section suivante.

Améliorations du SI

L'utilisation d'un ESB comme cœur du système d'information de la société ouvre à cette dernière de nouvelles perspectives d'organisation et d'évolution.

Organisation du SI en « services »

Dans une optique de développement des processus de la société de manière organisée, il est judicieux de passer à une approche SOA (orientée services). Le bus ESB étant déjà en place, il suffit d'offrir au travers d'interfaces de services les différentes fonctionnalités des applications déjà existantes.

Chaque application publie alors sur le bus ce qu'elle est capable de réaliser, sous forme de « services ».

Ces services peuvent alors être utilisés afin d'automatiser les processus existants et de créer de nouveaux processus ou de nouvelles applications plus rapidement, en évitant la réalisation de fonctionnalités déjà implémentées.

L'application des agences peut maintenant être découpée en trois parties :

- une partie dite de « présentation », avec laquelle l'agent travaille,
- une partie « réservation », enregistrant les demandent de réservation, et notifiant la base centralisée des réservations effectuées,
- une partie « consultation », donnant les informations concernant l'état du parc de véhicules.

Ces trois parties sont un découpage logique de l'application utilisée par les agences, et ne nécessitent pas forcément un déploiement sur des machines séparées.

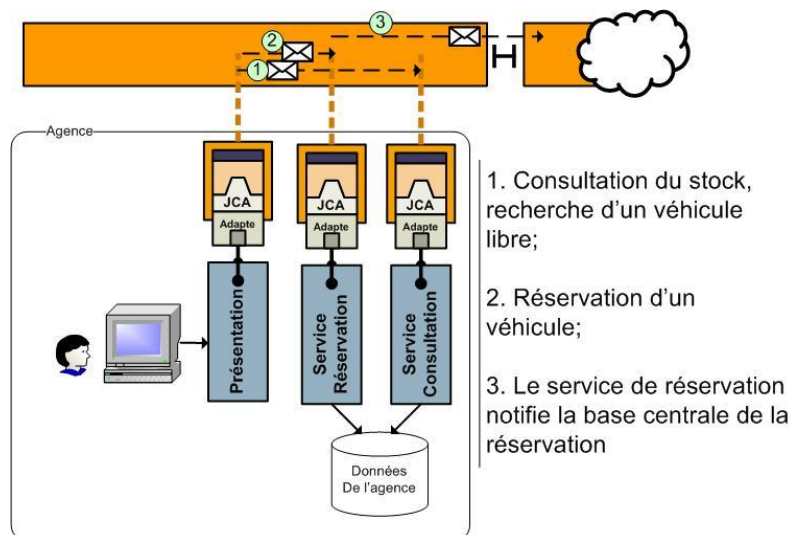


Figure 17 : Architecture orientée services dans les agences

En ce qui concerne le siège, un service de consultation de la base centrale est mis en place, ce qui découple l'application de réservation par Internet de la base de données centrale.

L'application de réservation en ligne utilise ce service via le bus ESB pour consulter les informations.

Un service de mise à jour des données est également offert, c'est ce service qui est utilisé par les agences pour mettre à jour les données de leur parc sur la base centrale.

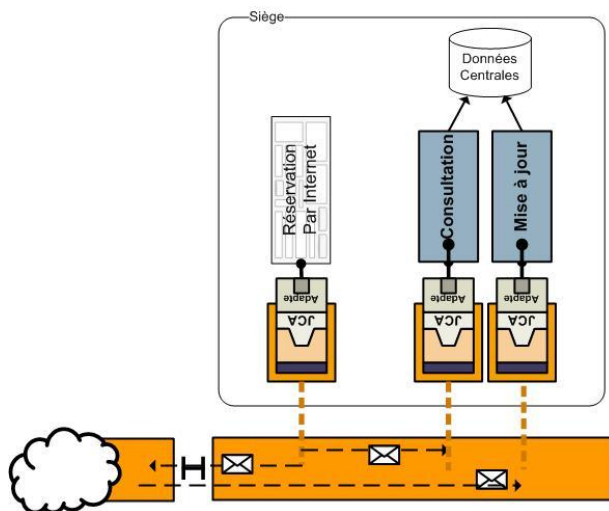


Figure 18 : Architecture orientée services au sein du siège de la société

Automatisation des réservations par Internet.

Cette architecture de services permet maintenant à la société d'automatiser le processus de réservation en ligne. En effet, le processus initial reposait sur l'envoi automatique par l'application Internet d'un email au responsable de l'agence concernée par la réservation. Ce dernier effectuait alors la réservation. La base centrale était ensuite mise à jour.

Il est possible d'orchestrer ce processus, en définissant l'enchaînement des services intervenant lors de la réservation.

Un service d'orchestration est ajouté à l'ESB, et gère l'ordre d'appel des services lors de la réalisation de ce processus.

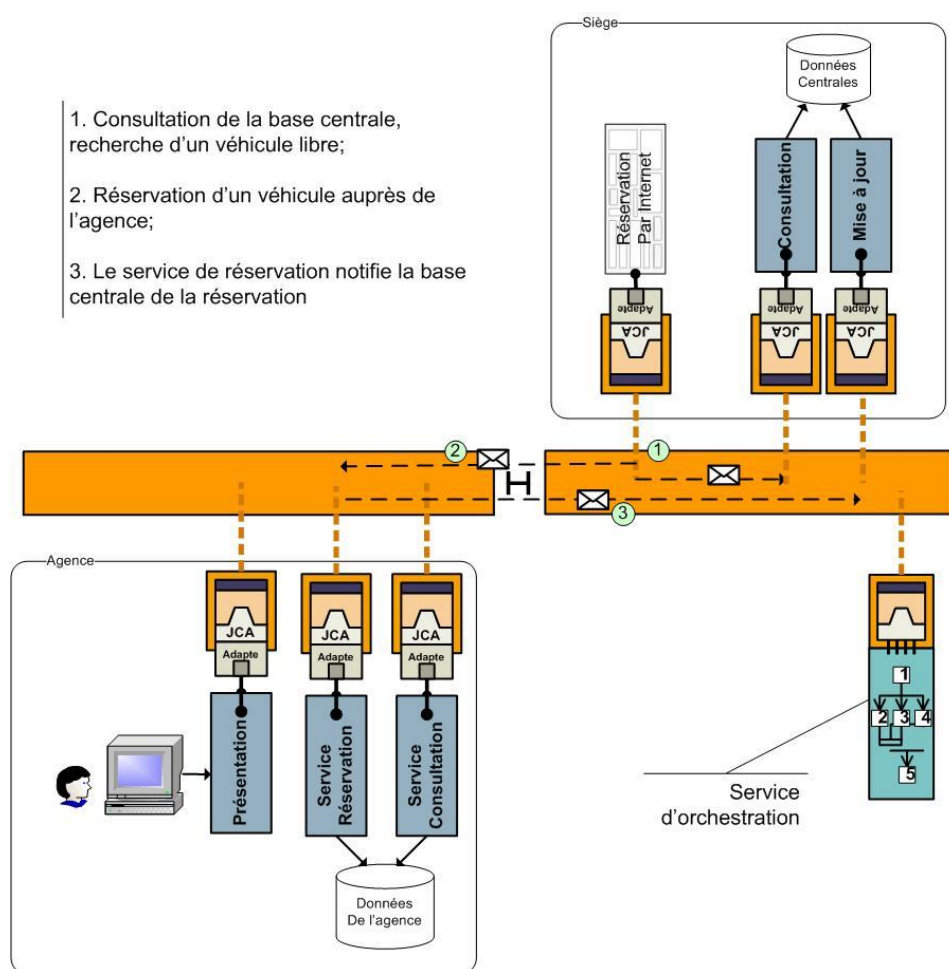


Figure 19 : Processus de réservation en ligne orchestré

Annuaire des services de l'entreprise

Un service d'annuaire peut être installé sur le bus. Cet annuaire va recenser tous les services existants connectés à l'ESB ainsi que leurs caractéristiques.

L'application de chaque agence va enregistrer automatiquement dans cet annuaire ses services, sa localisation géographique, etc.

Il suffit maintenant pour une application de consulter cet annuaire pour connaître automatiquement la liste des agences existantes, leur localisation géographique, les services qu'elle propose.

Une nouvelle agence qui se connecte au bus enregistre automatiquement ses informations dans l'annuaire, et pourra travailler en collaboration avec le reste du SI.

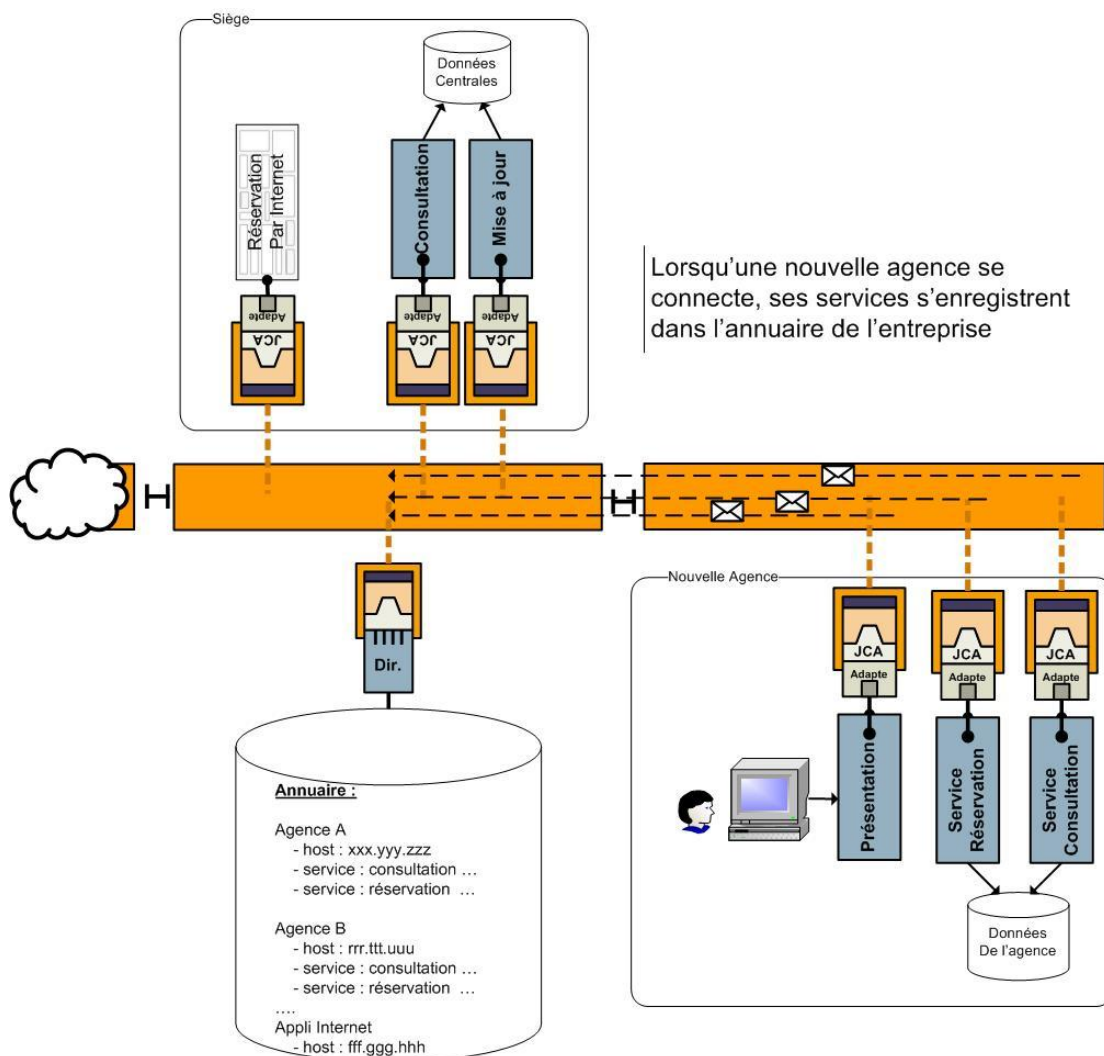


Figure 20 : Utilisation d'un annuaire de services

Partage des informations entre les agences.

Les agences peuvent maintenant utiliser les services d'autres agences librement. Ce partage d'informations va permettre l'amélioration du processus de location de véhicule. Jusqu'à présent, un client devait restituer son véhicule dans l'agence d'origine, car seule cette dernière possédait le dossier de location correspondant.

Maintenant, un client pourra restituer son véhicule dans n'importe quelle agence, qui utilisera les services de l'agence initiale pour clore le dossier de

location. Cette amélioration est un atout supplémentaire pour la société vis-à-vis de sa clientèle.

Routing des messages basé sur le contenu

Les agences peuvent maintenant communiquer entre elles. Lorsqu'un client arrive dans une agence pour réserver et qu'aucun véhicule n'est disponible, l'agence peut chercher dans les agences proches si des véhicules sont libres, en utilisant les services de consultation et de réservation des autres agences.

Un service de routage basé sur le contenu peut être installé sur le bus. On peut paramétrer ce service pour que le routage respecte la localisation géographique. Ce service permet, entre autres, d'envoyer un message vers les destinataires les plus proches géographiquement.

Une agence fait donc une demande de consultation sans préciser dans quelle agence elle recherche, c'est le service de routage qui va envoyer cette requête aux agences proches.

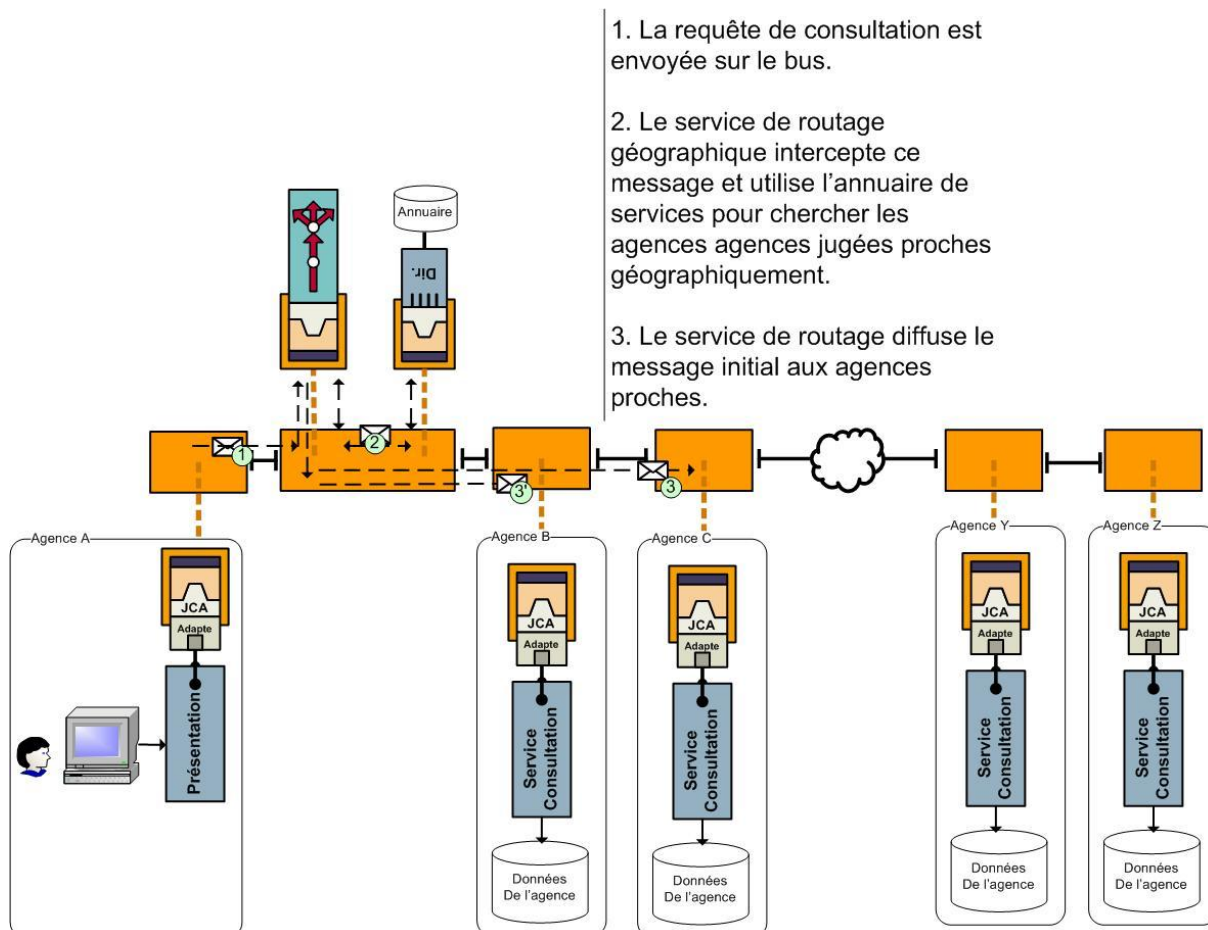


Figure 21 : Routage géographique de messages.

Ouverture du SI à des partenaires

La société de location possède maintenant un SI ouvert, configurable et adaptable. Ceci lui permet de pouvoir nouer des partenariats avec d'autres entreprises avec un minimum de développement, et surtout sans modification de la structure de son SI.

La société de location décide de se mettre en relation avec des organismes qui feront appel à elle pour des manifestations exceptionnelles nécessitant la location d'un grand nombre de véhicules (collectivités locales, grands groupes,...).

Le partenaire va alors se relier au bus ESB de la société de location, mais dans un mode restreint. Certains services de la société ne doivent en effet être visibles qu'au sein de la société elle-même.

Ainsi, la société de location expose publiquement un nouveau service qu'elle va développer, un service de Réservation pour Partenaire.

Le partenaire, de son côté, se connecte au bus, et envoie ses demandes de réservation à ce service.

Le service de la société de location se charge d'interroger les agences proches du lieu demandé par le partenaire (via leurs services de consultation).

1. Le partenaire envoie sa demande.

2. Le message du partenaire est transformé de façon à être compréhensible par le système.

3. Le service de Réservation Partenaire reçoit la demande et effectue une recherche de véhicules à proximité du lieu de la manifestation.

4. Le service de routage géographique diffuse cette demande aux agences proches du lieu.

5. Les agences concernées reçoivent la demande de consultation.

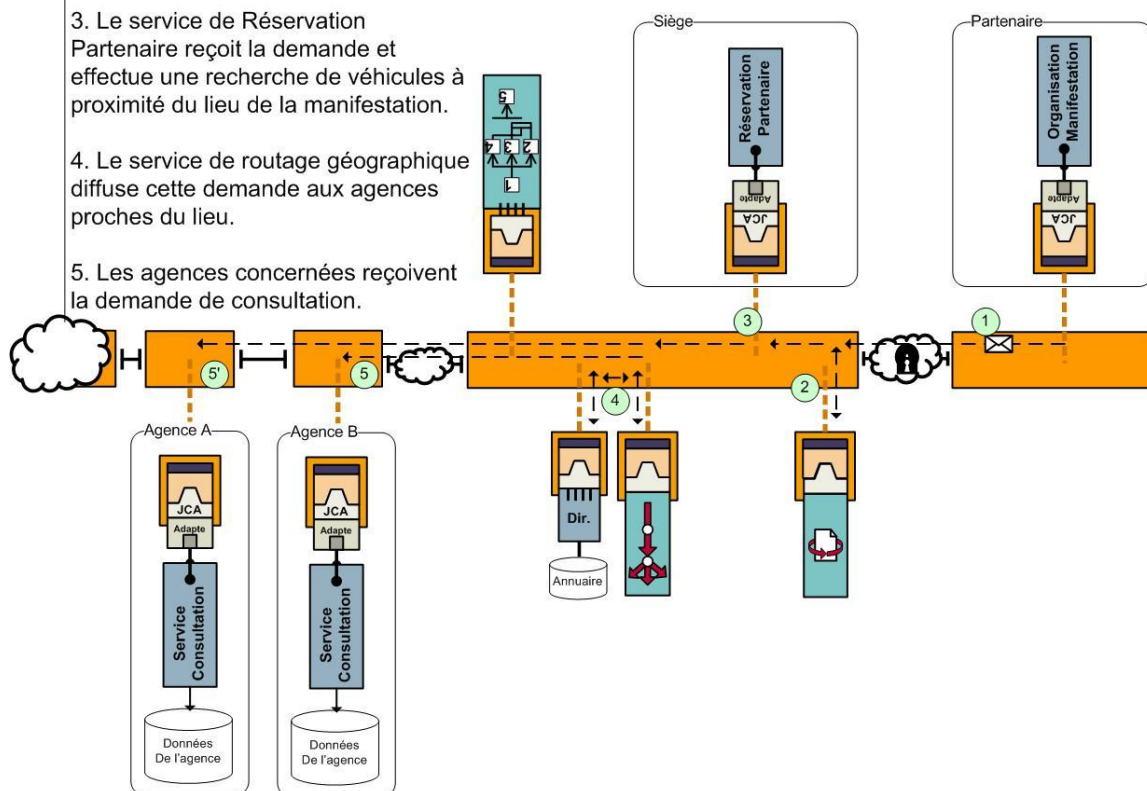


Figure 22 : Processus de réservation par un partenaire

Une fois les disponibilités des véhicules des agences proches retournées au service de Réservation Partenaire, ce dernier effectue les réservations auprès des agences (via leurs services de réservation), en choisissant par exemple de laisser au moins quelques véhicules disponibles dans chaque agence, pour que ces dernières puissent encore répondre à des demandes de réservation normales.

Les réservations effectuées, le service Relation Partenaire informe le partenaire de la réussite de la réservation.

Conclusion

Les ESB, solutions basées sur les standards tels que XML et les Web Services, prennent une place prépondérante dans ce cadre comme infrastructures robustes et ouvertes pour l'intégration des éléments constitutifs de la SOA.

De nombreuses solutions d'ESB sont apparues depuis 2005, proposées soit sous la forme d'offres commerciales comme les solutions de BEA, IBM, Sonic Software, SeeBeyond, ou encore webMethods, soit dans le cadre de communautés open source comme Apache qui propose ServiceMix et Synapse, ou OW2, communauté open source basée en Europe, qui héberge le projet **Petals ESB**.

Bibliographie

Livres

« Enterprise Service Bus, theory in practice ». David A. Chappell. O'Reilly.

Articles Internet

« ESB : tendance de fond ou nouveau buzzword ? ». Nicolas Farges, Techmetrix.net.
<http://www.zdnet.fr/techupdate/infrastructure/0,39020938,2132625,00.htm>

« Understand ESB scenarios and solutions in SOA », Rick Robinson, IBM,
<http://www.ibm.com/developerworks/library/ws-esbscen>

« Best-of-Breed ESBs », Steve Craggs, EAI Industry Consortium,
http://www.integrationconsortium.org/uploads/members/saint%20consulting%20limited/bestofbreed_esbs.pdf

Petals Link

Petals Link (marque commerciale de EBM Websourcing) est éditeur de logiciels open source, spécialiste en SOA. Petals Link fournit du service autour de ses logiciels, dont le principal, Petals ESB, est un **bus de service (ESB) open-source**, qui résout les problématiques d'exploitation d'architectures décentralisées, à grande échelle. Il est complété par une offre open source de **gouvernance SOA** et de **supervision métier (BAM)**.

Sites web et projets

- <http://petalslink.com> – Site de l'éditeur Petals Link
- <http://petals.ow2.org> – Petals ESB, l'ESB Open source à grande échelle
- <http://dragon.ow2.org> – Petals Master (ex Dragon), un outil de gouvernance intégré à PEtALS
- <http://opensuit.ow2.org> - Framework de présentation web orienté service
- <http://easywsdl.ow2.org> – Parseur WSDL 1.1 et 2.0
- <http://easybpel.petalslink.com> – Moteur d'exécution BPEL et processus

Contacts

contact@petalslink.com
+33 5 62 73 43 80
EBM Websourcing – Petals Link,
4 Rue Amélie,
31000 Toulouse, France