

Interopérabilité

Enterprise Application Intergration

ESB

Abdessamad Mektoubi

IGA
5ILTI/M2TIW

Enterprise Application Integration(EAI)

- Les entreprises disposent souvent d'un grand nombre d'applications différentes fonctionnant à plusieurs niveaux et sur différentes plates-formes en raison de la difficulté de créer une application d'entreprise unique et complète.
- Les entreprises préfèrent choisir la meilleure application pour chaque fonction spécifique, plutôt que de s'en remettre à une solution unique.
- Les fournisseurs ont réagi en proposant des applications ciblées pour des fonctions essentielles spécifiques, mais cela a entraîné un débordement de fonctionnalités entre les différentes applications.
- Il peut être difficile de définir clairement la séparation fonctionnelle entre les systèmes et les utilisateurs ne pensent pas aux frontières des systèmes lorsqu'ils interagissent avec une entreprise.

Le partage des données entre les applications nécessitent une intégration pour assurer un échange de données efficace, fiable et sécurisé.

- L'intégration d'entreprise concerne de multiples applications fonctionnant sur des plateformes et des sites différents, ce qui en fait une tâche complexe.
- Les suites EAI permettent une intégration multiplateforme et multilingue, mais les véritables défis de l'intégration sont à la fois commerciaux et techniques.
- L'intégration d'entreprise nécessite un changement significatif de la politique de l'entreprise et exige une communication entre plusieurs systèmes informatiques ainsi qu'entre les unités commerciales et les départements informatiques.
- Les efforts d'intégration ont des répercussions considérables sur l'entreprise, et une solution d'intégration défaillante ou qui se comporte mal peut être coûteuse.

- Les développeurs de l'intégration ont un contrôle limité sur les applications participantes et doivent souvent contourner les déficiences ou les différences entre les applications.
- Peu de normes se sont établies dans le domaine de l'intégration, et même les normes existantes ne traitent qu'une fraction des défis.
- L'exploitation et la maintenance d'une solution EAI peuvent être décourageantes en raison du mélange de technologies et de la nature distribuée de la solution.
- Les solutions EAI sont essentielles aux stratégies des entreprises d'aujourd'hui, mais elles peuvent rendre la vie des TI plus difficile.

- Information Portals
- Data Replication
- Shared Business Functions
- Service-Oriented Architectures
- Distributed Business Processes
- Business-to-Business Integration

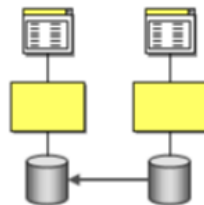
- Un portail d'information est un outil qui permet aux utilisateurs d'accéder à des informations provenant de plusieurs sources et de les visualiser à un endroit central,
- Il peut s'agir de simples affichages qui divisent l'écran en différentes zones, chacune montrant des informations provenant d'un système différent
- Ces portails peuvent améliorer l'efficacité et la productivité en permettant aux utilisateurs d'accéder plus facilement aux informations dont ils ont besoin et d'interagir avec elles.



EAI Categories

Shared Business Function

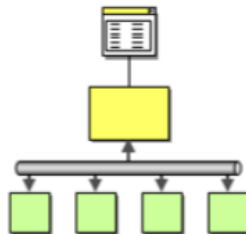
- La réplication des données est une méthode de synchronisation des données entre plusieurs systèmes, qui garantit que toutes les copies des données sont cohérentes et à jour.
- C'est important dans les situations où plusieurs systèmes doivent accéder aux mêmes données car les modifications apportées dans un système doivent être répercutées dans tous les autres systèmes qui utilisent ces données.
 - La réplication de base de données
 - ETL
 - Middleware



EAI Categories

Service-Oriented Architecture

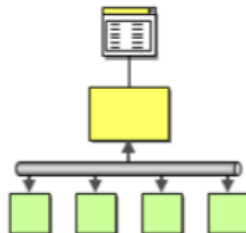
- Une architecture orientée services (SOA) est une façon de concevoir et de construire des applications en les composant à partir de services.
- Les services sont des fonctions bien définies qui sont universellement disponibles et répondent aux demandes d'autres applications.



EAI Categories

Service-Oriented Architecture

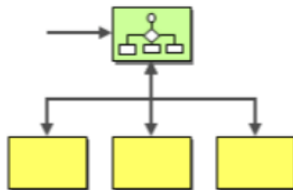
- Les éléments clés d'une SOA sont la découverte et la négociation de services, qui permettent aux applications de trouver des services et de communiquer avec eux par le biais d'un répertoire de services centralisé et d'une interface standardisée.
- Comme tous les services sont disponibles de manière cohérente, les SOA sont souvent appelées "architectures de bus de services".



EAI Categories

Service-Oriented Architecture

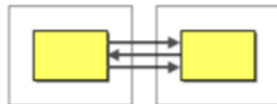
- La gestion des processus d'entreprise est également importante pour coordonner l'exécution des fonctions d'entreprise entre plusieurs systèmes,
- la distinction entre SOA et entreprise distribuée peut devenir floue.



EAI Categories

Business-to-Business Integration

- L'EAI se produit au sein d'une entreprise, entre différents départements ou fonctions commerciales, ou entre partenaires et fournisseurs externes.
- Les architectures orientées services (SOA) fournissent un moyen de gérer ces services et d'y accéder de manière cohérente.
- Lors de l'intégration avec des partenaires externes, des considérations telles que les protocoles de transport, la sécurité et les formats de données standardisés deviennent plus importantes.



- Couplage d'applications
- Simplicité d'intégration
- Technologie d'intégration
- Format des données
- Actualité des données
- Données ou fonctionnalité
- Asynchronicité

- Transfert de fichiers
- Base de données partagée
- L'intégration point à point
- Middleware
- Architecture orientée services
- Architecture pilotée par les événements (EDA)
- Intégration cloud

- Transfert de fichiers
 - Demandez à chaque application de produire des fichiers de données partagées pour que les autres puissent les consommer, et de consommer les fichiers que d'autres ont produits.
- Base de données partagée
 - Demandez aux applications de stocker les données qu'elles souhaitent partager dans une base de données commune
- Intégration point à point
 - Cette approche consiste à connecter deux applications directement à l'aide d'une interface personnalisée.
 - Il peut s'agir d'une solution simple et rapide, mais elle devient difficile à gérer et à adapter à mesure que d'autres applications sont ajoutées au système.

- Middleware

- Cette approche consiste à utiliser une couche logicielle qui se situe entre les applications et gère la communication entre elles.
- L'intergiciel peut fournir une variété de services tels que l'acheminement des messages, la transformation des données et la sécurité.
- Cette approche est plus évolutive et plus facile à gérer que l'intégration point à point, mais elle peut être complexe à mettre en place et à maintenir.

- Architecture orientée services (SOA)
 - Cette approche consiste à exposer les fonctions commerciales sous forme de services accessibles.
 - Les applications peuvent ensuite interagir avec ces services par le biais d'une interface bien définie, ce qui permet une plus grande flexibilité et une meilleure réutilisation.
 - L'architecture SOA peut être plus complexe à mettre en place et à gérer que l'intégration point à point ou les Middlewares, mais elle peut offrir une plus grande évolutivité et une meilleure flexibilité à long terme.

- Architecture pilotée par les événements (EDA)
 - Cette approche implique que les applications envoient et reçoivent des messages d'événement, plutôt que d'effectuer des appels directs les uns vers les autres.
 - Cela permet une intégration plus souple, facilitant l'ajout et la suppression d'applications dans le système.
 - L'EDA peut être plus complexe à mettre en place et à gérer que l'intégration point à point ou les intergiciels

- Intégration basée sur le cloud
 - Cette approche implique l'utilisation de plateformes et de services basés sur le cloud pour intégrer les applications.
 - Cette approche peut être plus évolutive que les solutions traditionnelles sur site, mais il peut être difficile de sécuriser et de gérer les données dans le nuage.

- La messagerie permet un couplage souple entre les applications en communiquant de manière asynchrone.
- La communication asynchrone augmente la fiabilité car les deux applications ne doivent pas nécessairement fonctionner en même temps.
- Le système de messagerie est responsable du transfert des données entre les applications, ce qui permet aux applications de se concentrer sur les données à partager plutôt que sur la manière de les partager.

Messaging Systems

Composants d'un Messaging Systems

- Channels

- Les applications de messagerie transmettent des données par le biais d'un canal de message, un tuyau virtuel qui relie l'émetteur au récepteur.
- Un système de messagerie nouvellement installé ne contient pas de canaux ;
- Vous devez déterminer comment vos applications doivent communiquer, puis créer les canaux pour faciliter cette communication.

Messaging Systems

Composants d'un Messaging Systems

- Messages

- Un message est un paquet atomique de données qui peut être transmis sur un canal.
- Pour transmettre des données, une application doit diviser les données en un ou plusieurs paquets, emballer chaque paquet comme un message, puis envoyer le message sur un canal.
- De même, une application réceptrice reçoit un message et doit extraire les données du message pour le traiter.
- Le système de messagerie essaiera de livrer le message (par exemple, le transmettre de l'expéditeur au récepteur)

Messaging Systems

Composants d'un Messaging Systems

- Multi-step delivery

- Dans le cas le plus simple, le système de messagerie remet un message directement de l'ordinateur de l'expéditeur à l'ordinateur du destinataire.
- Cependant, des actions doivent souvent être effectuées sur le message après qu'il a été envoyé par son expéditeur initial mais avant qu'il ne soit reçu par son destinataire final.
- Par exemple, le message peut devoir être validé ou transformé parce que le récepteur s'attend à un format de message différent de celui de l'expéditeur.

Messaging Systems

Composants d'un Messaging Systems

- Routage

- Dans une grande entreprise comptant de nombreuses applications et des canaux pour les connecter, un message peut devoir passer par plusieurs canaux pour atteindre sa destination finale.
- L'itinéraire qu'un message doit suivre peut être si complexe que l'expéditeur d'origine ne sait pas quel canal fera parvenir le message au récepteur final.
- Au lieu de cela, l'expéditeur d'origine envoie le message à un Message Router, un composant applicatif qui déterminera comment naviguer dans la topologie du canal et diriger le message vers le récepteur final

Messaging Systems

Composants d'un Messaging Systems

- Transformation

- Diverses applications peuvent ne pas s'entendre sur le format des mêmes données conceptuelles.
- L'expéditeur formate le message d'une certaine manière, mais le récepteur s'attend à ce qu'il soit formaté d'une autre manière.
- Pour concilier cela, le message doit passer par un filtre intermédiaire, un traducteur de message, qui convertit le message d'un format à un autre.

Messaging Systems

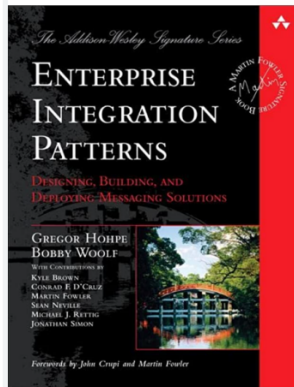
Composants d'un Messaging Systems

- Endpoints

- Une application n'a pas de capacité intégrée pour s'interfacer avec un système de messagerie.
- Elle doit plutôt contenir une couche de code qui connaît à la fois le fonctionnement de l'application et celui du système de messagerie.

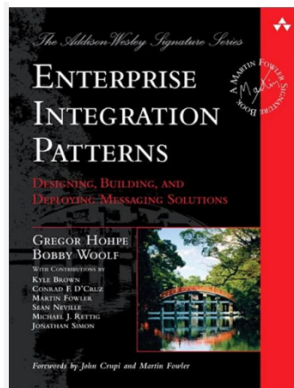
Enterprise Integration Patterns (EIP)

- Langage de patrons composé de 65 patrons d'intégration aide les développeurs à concevoir et à créer des applications distribuées ou à intégrer des applications existantes.
- Les patrons fournissent un vocabulaire indépendant de la technologie et une notation visuelle tirée de solutions éprouvées à des problèmes récurrents.



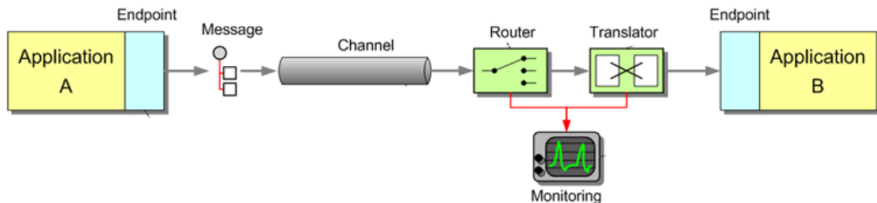
Enterprise Integration Patterns (EIP)

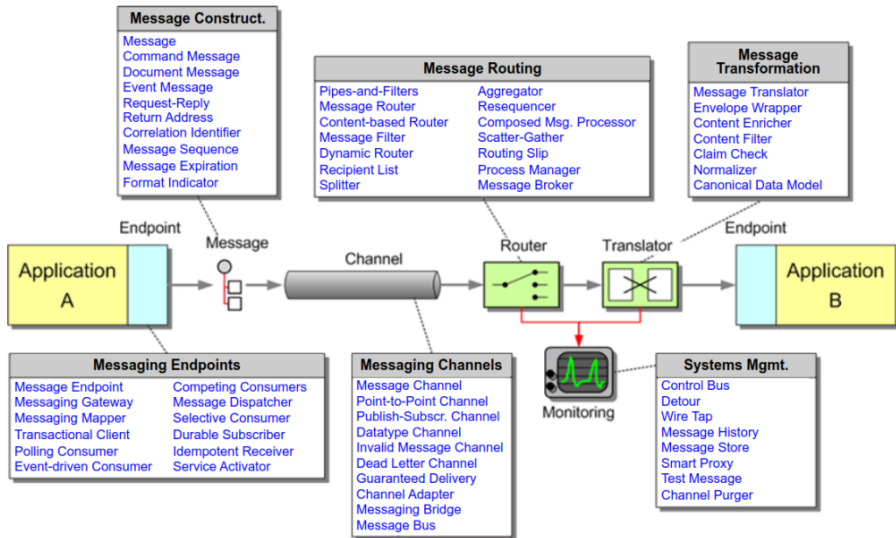
- Ces patterns ont stimulé le développement de toute une génération d'implémentations de bus de service d'entreprise (ESB) open source, comme Apache Camel, Mule, WSO2, Oracle Service Bus, Open ESB, SonicESB ou Fuse ServiceMix.



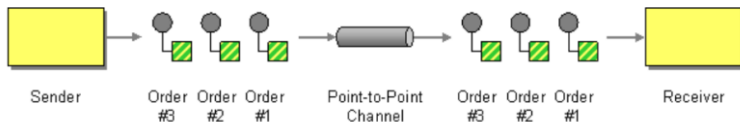
- Un modèle de conception est une solution générale reproductible à un problème courant dans la conception de logiciels.
- Un design pattern n'est pas une conception finie qui peut être transformée directement en code. Il s'agit d'une description ou d'un modèle de la façon de résoudre un problème qui peut être utilisé dans de nombreuses situations différentes.

- Les modèles de conception sont des meilleures pratiques formalisées qui peut etre utiliser pour résoudre des problèmes courants lors de la conception d'une application ou d'un système.
 - Problème récurrent
 - Pourquoi et Comment
 - Nom



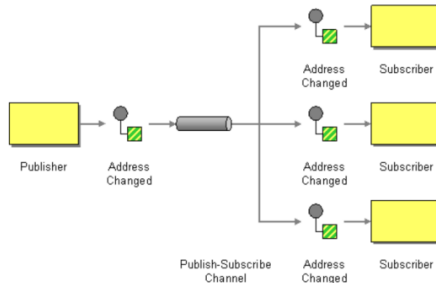


- Point-to-Point Channel.
 - Ce modèle décrit comment envoyer et recevoir des messages à l'aide d'un canal point à point, qui permet à un seul récepteur d'envoyer et de recevoir des messages.



- Publish-Subscribe Channel

- Ce modèle décrit comment envoyer et recevoir des messages à l'aide d'un canal de publication et d'abonnement, qui permet d'envoyer des messages à plusieurs récepteurs à la fois.



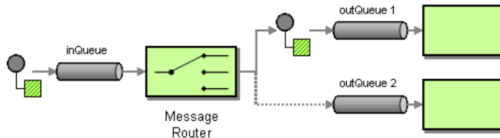
- Pipes and Filters

- Ce modèle est utilisé pour décomposer une tâche importante en tâches plus petites et plus faciles à gérer.
- Chaque filtre exécute une fonction spécifique et les données passent par un pipeline de filtres pour obtenir le résultat souhaité.



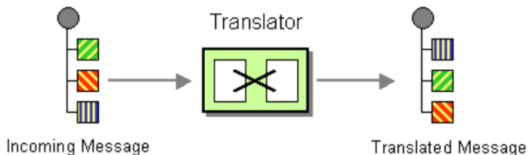
- Message Router

- Ce modèle est utilisé pour acheminer les messages en fonction de leur contenu.
- Le routeur examine le message et l'achemine vers la destination appropriée.
- Il est utile dans les situations où les messages doivent être envoyés à différentes destinations en fonction de certaines conditions.



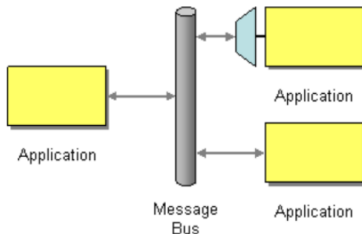
- Message Translator

- Ce motif est utilisé pour traduire les messages d'un format à un autre.
- Il est utile dans les situations où différents systèmes utilisent différents formats de messages et doivent communiquer entre eux.



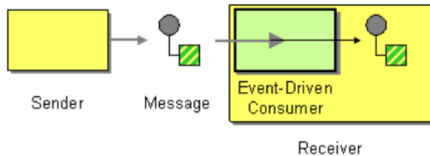
- Bus de messages

- Ce patron est utilisé pour créer une infrastructure de messagerie centralisée qui permet à différents systèmes d'envoyer et de recevoir des messages.
- Cela facilite la gestion de l'infrastructure de messagerie et la communication entre les différents systèmes.



- Event-Driven Consumer

- Ce modèle est utilisé pour traiter les messages de manière asynchrone, permettant à l'expéditeur et au destinataire de travailler indépendamment l'un de l'autre.
- IL est utile dans les situations où les messages doivent être traités en temps voulu, mais où l'expéditeur et le destinataire n'ont pas besoin de fonctionner en même temps.



<https://www.enterpriseintegrationpatterns.com/>

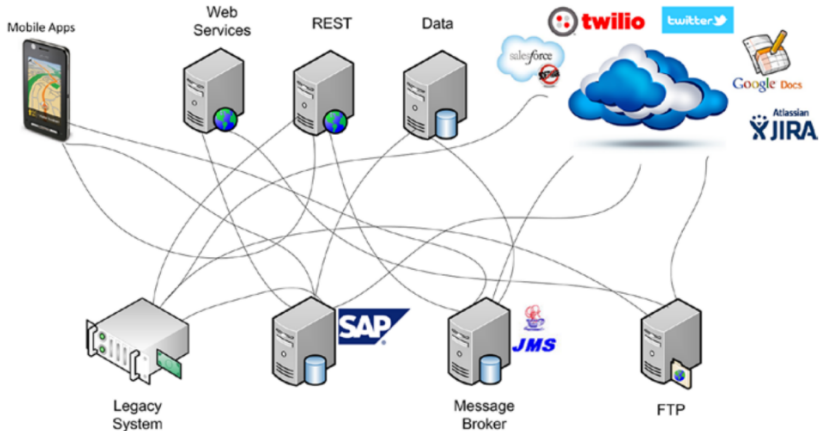
Enterprise service bus (ESB)

- ESB est un modèle d'architecture logicielle qui permet à différents systèmes et applications de communiquer entre eux par le biais d'un système de messagerie centralisé.
- Il agit comme une couche intermédiaire qui facilite l'intégration de diverses technologies et protocoles, permettant une communication et un échange de données transparents entre différents systèmes.
- Les ESB sont couramment utilisés dans les environnements d'entreprise pour prendre en charge des scénarios d'intégration tels que l'architecture orientée services (SOA) et les microservices.

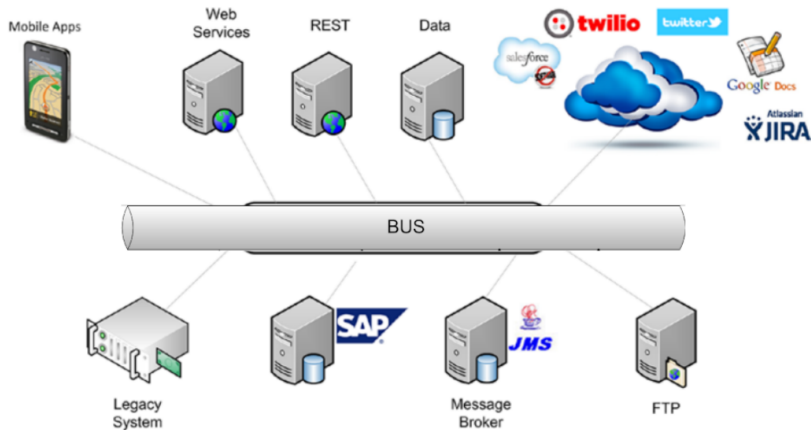
Enterprise service bus (ESB)

- À un haut niveau, un ESB agit comme un hub central pour toute communication entre différents systèmes.
- Il reçoit les demandes des clients, les achemine vers le service approprié, puis renvoie la réponse au client.
- L'ESB fournit également une variété de fonctionnalités supplémentaires telles que la transformation des messages, la conversion des protocoles et la sécurité.

Enterprise service bus (ESB)



Enterprise service bus (ESB)



Enterprise service bus (ESB)

- Les EIP et les ESB sont étroitement liés, car les EIP fournissent un ensemble de bonnes pratiques et de modèles pour concevoir et mettre en œuvre des solutions d'intégration, tandis que les ESB fournissent l'infrastructure nécessaire à la mise en œuvre de ces solutions.
- De nombreux ESB sont construits à partir des concepts et des modèles décrits dans les EIP, et ils offrent souvent un support intégré pour un large éventail d'EIP.

Enterprise service bus (ESB)

ESB vs ETL

- L'ESB est un système intergiciel basé sur la messagerie qui fait office de centre de communication pour les différents systèmes et applications d'une entreprise. Il facilite l'intégration de divers services et applications en fournissant des capacités de routage, de transformation et de médiation de protocole.
- L'ETL est un processus d'intégration de données qui extrait les données de diverses sources, les transforme pour répondre aux exigences du système cible et les charge dans le système cible.
- L'ETL est principalement utilisé pour l'entreposage de données, la veille stratégique et l'analyse.

Enterprise service bus (ESB)

ESB vs ETL

- L'ESB est davantage axé sur la communication et l'intégration de diverses applications et services, tandis que l'ETL se concentre sur l'intégration et la gestion des données.
- L'ESB permet une intégration en temps réel, tandis que l'ETL est généralement orienté vers le traitement par lots.
- L'ESB peut être utilisé conjointement avec l'ETL comme moyen de déplacer et de transformer les données entre les systèmes, mais ils ont des objectifs différents.

- **Apache ServiceMix** : Un bus de service d'entreprise (ESB) open-source construit au-dessus du Bbroker de messages Apache ActiveMQ et du moteur de routage et de médiation Apache Camel.
- **WSO2 ESB** : un ESB léger et open-source qui permet aux développeurs de connecter, gérer et transformer des API et des services.
- **Mule ESB** : Une plateforme ESB et d'intégration légère et open-source qui permet aux développeurs de connecter des applications ensemble rapidement et facilement, leur permettant d'échanger des données.

- **JBoss Fuse** : Un ESB open-source basé sur Apache Camel et ActiveMQ, conçu pour faciliter l'intégration et la gestion des applications et services d'entreprise par les développeurs.
- **Talend ESB** : Une plateforme ESB et d'intégration de données open-source qui permet aux développeurs de connecter et de gérer des applications, des services et des sources de données de manière unifiée.
- **OpenESB** : Une plateforme ESB et d'intégration open-source qui permet aux développeurs de connecter, gérer et orchestrer des services et des applications à l'aide d'une interface visuelle.
- **Petals ESB** : Une plateforme ESB et d'intégration open-source qui permet aux développeurs de connecter, gérer et orchestrer des services et des applications à l'aide d'une interface visuelle et d'une architecture à base de composants.