

« Stratégies de développement des Systèmes d'Information Opérationnels de l'entreprise » (5)



Bernard ESPINASSE
Professeur à Aix-Marseille Université (AMU)
Ecole Polytechnique Universitaire de Marseille



Septembre 2014

- Introduction
- La stratégie développement spécifique
- La stratégie ERP
- La stratégie EAI (Enterprise Application Integration)

Sommaire

1. Introduction
2. La stratégie développement spécifique
 - Une méthode de conception spécifique : Merise
 - Origine et évolution
 - Niveaux d'abstraction et modèles
 - Démarche et décision
 - Forces et faiblesse de la stratégie développement spécifique
3. La stratégie ERP
 - Architecture modulaire d'un ERP
 - L'offre ERP
 - Forces et faiblesses de la stratégie ERP
4. La stratégie EAI (Enterprise Application Integration)
 - Introduction aux EAI
 - Architecture et composants d'une EAI
 - Exemple de fonctionnement d'une EAI
 - Type d'architecture d'EAI
 - Forces et faiblesse de la stratégie EAI

Références bibliographiques et emprunts

Ouvrages :

- P. Vidal, V. Petit, F. Lacroux, M. Augier, V. Merminod, M. de Gibon, C. Mangholz, *Systèmes d'information organisationnels*, 2e édition, Pearson Editeur, 2009.
- D. Nanci, B. Espinasse, B. Cohen, H. Hechenroth, J.C. Asselborn, *Ingénierie des Systèmes d'information : Merise 2° génération*, Vuibert, 2002.
- K. Laudon, J. Laudon, *Management des systèmes d'information*, 13e édition, Adapté par E. Fimbel, S. Costa, S. Canevet-Lehoux, Pearson Editeur, 2013.
- C. Morley, J. Hugues, B. Leblanc, O. Hugues, *Processus Métiers et systèmes d'information : Evaluation, modélisation, mise en oeuvre*, Dunod, 2005.
- Octo technology : Livre blanc des EAI. <http://www.octo.com/>

Cours :

- Cours de G. Rivière, ESTIA, 2014
- Note de synthèse de C. Plumejeaud, « Urbanisation des Systèmes d'Information : l'EAI », 2008
- Cours de L. Stumpf, « Enterprise Application Integration », CNAM 2006.

1 – Introduction

- Intégration des SI opérationnels
- Grandes stratégies pour le développement de SI opérationnels

Intégration des SI opérationnels

- Les **SI opérationnels** sont principalement dédié à supporter une **fonction particulière** de l'entreprise
 - Le développement de SI opérationnels conduit de plus en plus à l'émergence de **standards métier** :
 - **Enterprise Resource Planning (ERP)**,
 - **Customer Relationship Management (CRM)**,
 - **Supply Chain Management (SCM)**
 - **EDI : (Echange de Données Informatisées/Electronic Data Interchange)**
 - La tendance est à **intégrer** ces divers SI opérationnels selon diverses stratégies
- ⇒ **INTEGRATION DES SI Opérationnels = ERP, échanges de données informatisée (EDI), EIA, ...**

Grandes stratégies pour le développement de SI opérationnels

- **Développement spécifique** de logiciels (dans ou en dehors l'entreprise)
- **Acquisition et paramétrage d'un ERP** : 1 seul logiciel pour l'entreprise (Progiciel)
- **Agrégation/interfaçage/intégration de logiciels** :
 - Usage d'**intergiciels** (middleware)
 - **IAE** : Intégration d'Application d'Entreprise ou **EAI** : Enterprise Application Integration
- **Externalisation** : le SII est hébergé chez une autre entreprise

2 – La stratégie développement spécifique

- **Une méthode de conception spécifique : Merise**
- **Origine et évolution**
- **Niveaux d'abstraction et modèles**
- **Démarche et décision**

Une méthode de conception spécifique : Merise

- Méthodes de conception de SI
- La plus connue est **MERISE**
- **1978 : Merise 1^{ière} génération** :
 - développée sur l'impulsion du **Ministère de l'industrie**
 - concevoir et définir une **méthode d'intérêt national**
 - en collaboration avec les principales sociétés de service et le CETE d'Aix-en-Provence (H.Tardieu - A.Rochfeld)
 - s'appuie sur une **approche systémique**
 - définit différents **niveaux de préoccupation ou d'abstraction** (conceptuel, organisationnel/logique, physique)
 - propose de **nombreux modèles complémentaires**
 - propose une **démarche** garantissant la rigueur de la méthode et sa facilité d'application sur le terrain

Origine de Merise

Merise propose :

- un **schéma de réflexion** :
 - reposant sur des **concepts propres**
 - dans un langage commun à l'organisateur et l'informaticien
- un **guide normalisé** :
 - pour l'analyse et la définition des spécifications des futurs SI
 - définissant un découpage en **étapes cohérentes**
 - fournissant des **points de repères** permettant éventuellement de diversifier les intervenants de chaque étape (division du travail)
- un **support continu et adapté à la conduite de projet**
- des **standards** dans les domaines de :
 - la conception, l'analyse (fonctionnelle)
 - la réalisation (organique)
- des **outils** :
 - **conceptuels** ou/et **informatisé**
 - permettant de guider ou d'assurer le passage du point de départ au point d'arrivée de chaque étape.

Evolutions de Merise

- **1992 : Merise 2ième génération**
- **évolution du cadre de modélisation** :
 - extension de 3 à 4 **niveaux d'abstraction** (conceptuel, organisationnel, logique et physique)
 - émergence de **nouveaux modèles** :
 - modèle logique de traitements (**MLT**)
 - modèle organisationnel de données (**MOD**),
 - distinction de **2 missions distinctes** de l'ingénierie des SI :
 - conception du **Système d'Information Organisationnel (SIO)**
 - conception du **Système d'Information Informatisé (SII)**
- **évolution des outils et formalismes associés** :
 - extension du formalisme entité-relation, avec par exemple l'explicitation de types et sous-types, de contraintes d'intégrité, ...
 - clarification de la modélisation des traitements à l'aide du formalisme issu des réseaux de Pétri, à différents niveaux de préoccupation.

Les 4 niveaux d'abstraction de Merise 2

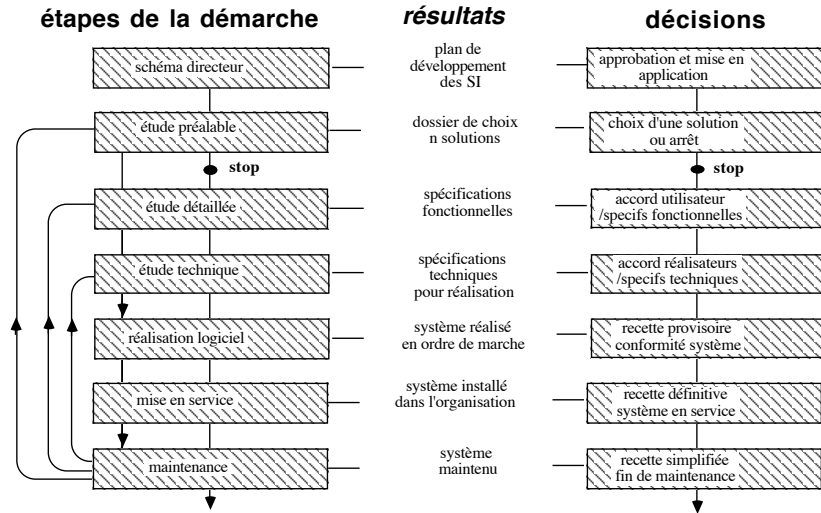
- **Système d'Information Organisationnel (SIO) :**
 - niveau **conceptuel** : exprime les choix fondamentaux de gestion : recherche des éléments stables indépendamment des moyens à mettre en oeuvre, de leurs contraintes et de leur organisation.
 - niveau **organisationnel** : exprime les choix d'organisation de ressources humaines et matérielles, au travers de la définition de sites, de postes de travail,...
- **Système d'Information Informatisé (SII) :**
 - niveau **logique** : exprime les choix de moyens et de ressources informatiques, en faisant abstraction de leurs caractéristiques techniques précises.
 - niveau **physique** : traduit les choix techniques et la prise en compte de leurs spécificités.

Les 8 Modèles principaux de Merise :

- **4 niveaux d'abstraction**
- **2 volets : données et traitements**
=> **8 modèles complémentaires**

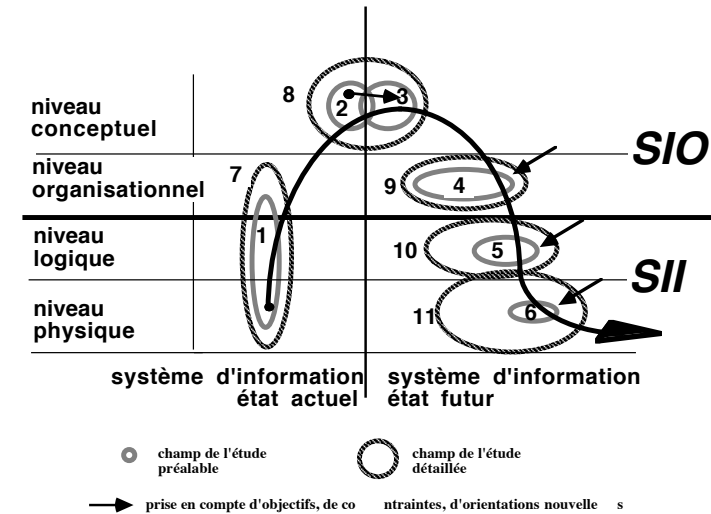
	Données	Traitements	
conceptuel	MCD <i>Modèle Conceptuel de Données</i>	MCT <i>Modèle Conceptuel de Traitements</i>	SIO Système d'Information Organisationnel
organisationnel	MOD <i>Modèle Organisationnel de Données</i>	MOT <i>Modèle Organisationnel de Traitements</i>	
logique	MLD <i>Modèle Logique de Données</i>	MLT <i>Modèle Logique de Traitements</i>	SII Système d'Information Informatisé
physique	MPD <i>Modèle Physique de Données</i>	MPT <i>Modèle Physique de Traitements</i>	

Etapes de la démarche et décisions



Cheminement du processus de conception dans Merise

Courbe dite du « soleil » :



Forces et faiblesses de la stratégie développement spécifique

Forces :

- Repose sur une **analyse très fine des besoins de l'organisation**
- Doit **parfaitement répondre à ces besoins**
- Peut permettre un **véritable avantage concurrentiel** en se démarquant de la concurrence
- Peut ainsi constituer le **levier d'une stratégie spécifique** de l'organisation
- ...

Faiblesses :

- Peut conduire à des **coût et délais importants de développement**
- Engendre des **coûts pour la maintenance** tant corrective qu'évolutive
- Nécessite en général des **compétences en conception et en réalisation informatique** dans l'organisation
- ...

3 – La stratégie ERP

- **Définition d'un ERP**
- **Architecture modulaire d'un ERP**
- **L'offre ERP**
- **Forces et faiblesses de la stratégie ERP**

Définition d'un ERP

- Les ERP (en anglais **Enterprise Resource Planning**), aussi appelés Progiciels de Gestion Intégrés (PGI)
- "ERP" provient du nom de la méthode **MRP** (Manufacturing Resource Planning) utilisée depuis les années 70 pour la gestion et la planification de la production industrielle.
- Ce sont des applications dont le but est de **coordonner, intégrer**, l'ensemble des **activités verticales** d'une entreprise comme :
 - la production
 - l'approvisionnement
 - ...
- ou les **activités horizontales** comme :
 - le marketing
 - les forces de vente
 - la gestion des ressources humaines
 - ...
- autour d'un **même système d'information**.

La stratégie ERP

ERP : Enterprise Resource Planning (en Français **PGI** : Progiciel de Gestion Intégré)

- Solution logicielle qui **regroupe** en son sein les principales composantes fonctionnelles de l'entreprise :
 - gestion production, gestion commerciale, logistique, RH, comptabilité/gestion, paie, vente, distribution, approvisionnement, stock, e-commerce, ...
 - gestion du processus de planification/ordonnancement, ...
 - suivi de fabrication et de la traçabilité, ...
 - gestion sous-traitance, maintenance, qualité, ...



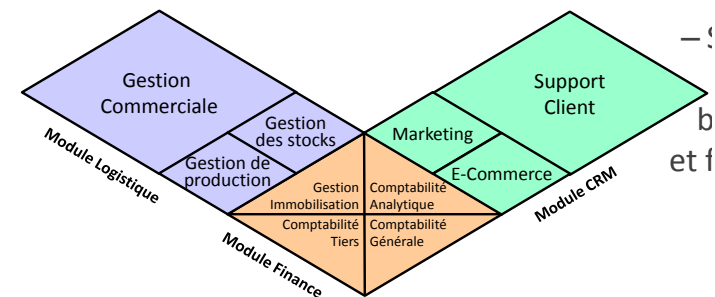
ERP et spécialisations

Certains sont dédiés à des secteurs d'activité particuliers (ou surcouches) :

- Aéronautique
- Assurances
- Automobile
- Banques
- BTP
- Cosmétiques
- Electroménager
- Filière Agroalimentaire
- Grande distribution
- Hôpital
- Imprimeurs
- Prêt-à-porter
- Téléphonie
- ...

Architecture modulaire d'un ERP (1)

- À **chaque fonction** de l'entreprise correspond un **module indépendant**
- Ces modules **partagent la même base de données** et sont **compatibles** entre eux (pas besoin de vérification)
- Ces modules s'imbriquent comme des **blocs de Lego** et fonctionnent ensemble



Architecture modulaire d'un ERP (2)

- Moteur de Workflow intégré :
 - Après saisie ou m.à.j, **propagation de l'information** dans tous les **modules** qui en ont besoin (synchronisation)
 - **Automatisé** (et **paramétrable**)
 - **Transparent** pour l'utilisateur

- L'ERP permet de gérer :
 - Plusieurs devises
 - Plusieurs langues (utilisateurs, clients, fournisseurs)
 - Plusieurs législations

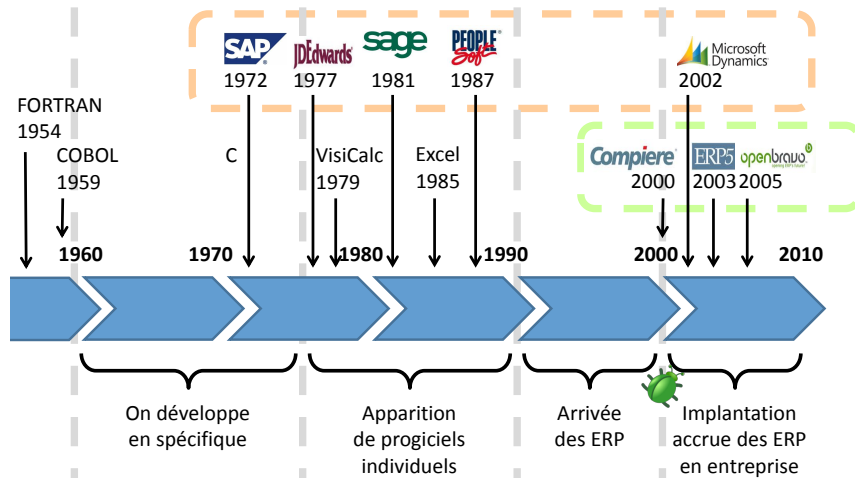
Mise en place d'un ERP

- C'est un **véritable projet** demandant :
 - une **intégration totale d'un outil logiciel** au sein d'une organisation
 - une **structure spécifique**
 - des **coûts importants d'ingénierie**

- Elle entraîne des **modifications importantes des habitudes de travail** d'une grande partie des employés.

- On considère que le **coût de l'outil logiciel** représente **moins de 20%** du coût total de mise en place !

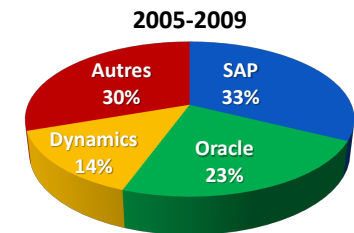
Historique de l'offre ERP



Pour rappel : C : 1972, C++ : 1983, HTML : 1992, PHP : 1994, Java : 1995

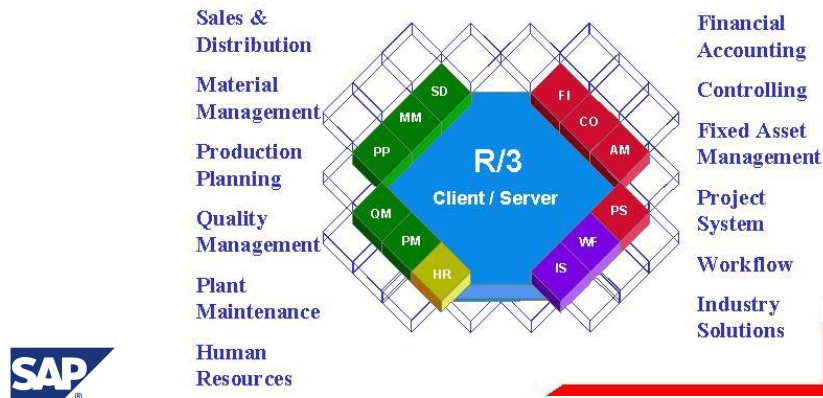
Offres commerciales en ERP

- Une centaine
- Principaux acteurs du marché :
 1. SAP (1972)
 2. ORACLE (v1 en 1978)
 - E-BUSINESS SUITE
 - PEOPLESOFT
 - JD EDWARDS
 3. SAGE ERP (1981)
 4. MICROSOFTDYNAMICS

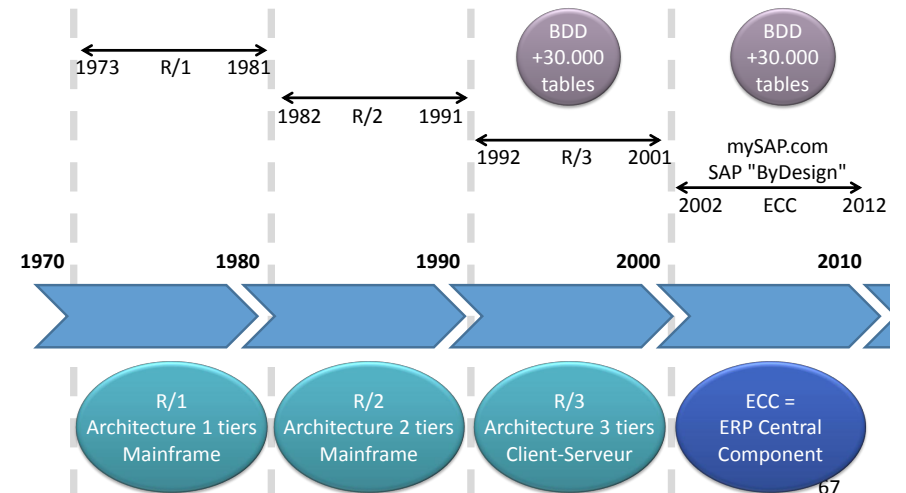


L'offre ERP SAP R/3 (1992-2001)

R/3 Core Business Processes



Evolution de l'offre ERP SAP R/3



L'offre ERP Sage x3

• Sage ERP x3



ERP : l'offre en logiciels libres

• Une trentaine

• Les principaux :

- **COMPIERE** (2000, Java) 2008-2009 - www.compiere.com
- Réf : Yves Rocher, La poste, ...
- **OPENBRAVO** (2005, Java) - www.openbravo.com
- **ERP5** (2003, Python/Zope) 2006 - www.erp5.org
- Réf : EADS, ...
- **OFBIZ** (2001, Java) - www.ofbiz.apache.org
- **OPENERP** (2002, Python) - www.openerp.com
- **NEOGIA** (2004, Java) - <http://neogia.org>



Forces et faiblesses des ERP libres

Forces :

- Développés en **étroite collaboration avec des utilisateurs**
- **Temps mise en œuvre inférieur** aux ERP commerciaux
- **Très faible taux d'échec** (car adaptable)
- Pas de formation conçue et gérée exclusivement par un vendeur (pratique parfois discutable)

Faiblesses :

- Concurrents commerciaux implantés depuis plusieurs décennies
- Encore très récents (jeunesse)

Forces de la stratégie ERP

Un système **unifié** permet de faire travailler des utilisateurs de différents métiers dans un **environnement applicatif identique** :

- **1 seule BD, cohérence et homogénéité des données**
- **Intégrité et unicité du SI, non-redondance**
- **Minimisation des coûts** :
 - pas d'interface entre modules,
 - synchronisation des traitements,
 - corrections assurées par l'éditeur
- **Globalisation de la formation** (même logique et ergonomie)
- **Coûts et des délais de mise en œuvre** sont connus (souvent de 3 à 36 mois)

Faiblesses de la stratégie ERP

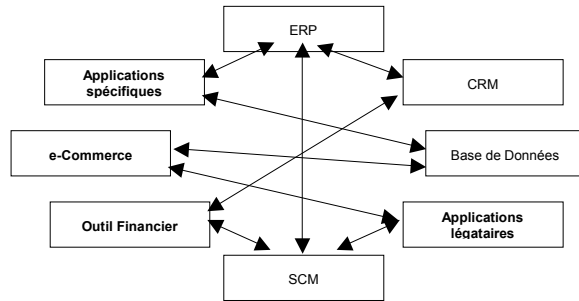
- **Coût élevé** (investissement lourd)
- **Couvre rarement tous les besoins** : nécessite souvent des développements supplémentaires
- **Couverture fonctionnelle plus large que les besoins** : nécessite une bonne connaissance des processus de l'entreprise
- L'entreprise doit parfois **adapter ses processus** à l'ERP
- **Dépendance vis-à-vis de l'éditeur** (code source)
- **Lourdeur et rigidité de mise en œuvre** : difficulté d'appropriation par utilisateurs

3 – La stratégie EAI (Enterprise Application Integration)

- **Introduction aux EAI**
- **Composants d'une plate-forme EAI**
- **Architecture fonctionnelle d'une EAI**
- **Exemple de fonctionnement d'une EAI**
- **Type d'architecture d'EAI**
- **Forces et faiblesse de la stratégie EAI**

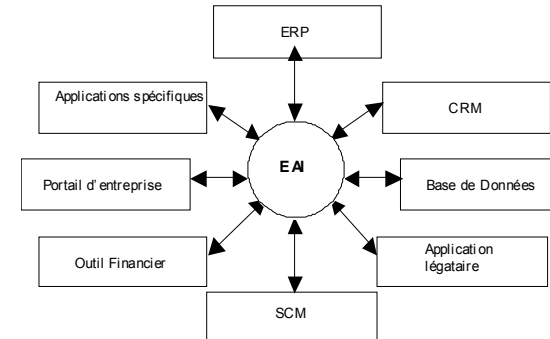
Introduction aux EAI - Enterprise Application Integration (1)

- Depuis les années 80, **complexité des SI** n'a cessé d'augmenter :
 - ajout de **nouveaux applicatifs** toujours plus nombreux
 - complexification du réseau inter-applicatif**
 - ⇒ **plat de spaghettis inextricable ...**



Introduction aux EAI - Enterprise Application Integration (2)

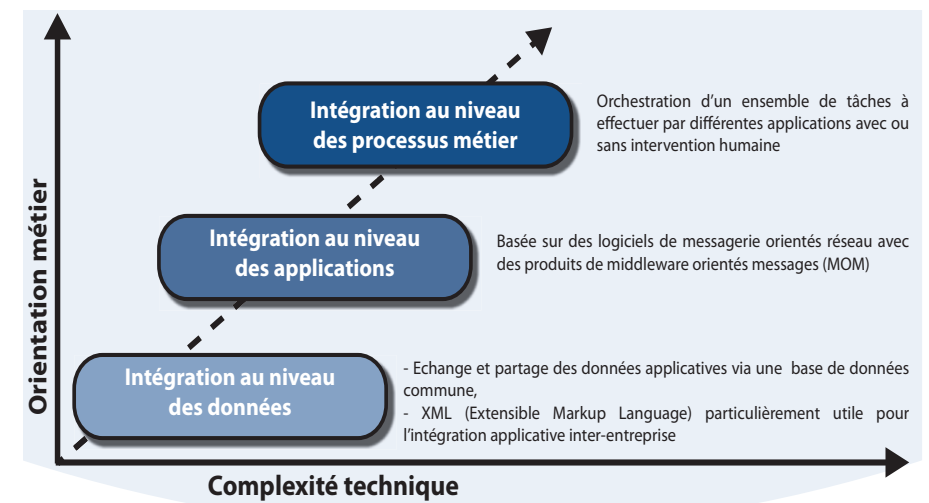
- L'objectif des plates-formes logicielles EAI** (Enterprise Application Integration) est de simplifier ce « plat de spaghettis » :
 - En **centralisant les messages inter-applicatifs**
 - En **homogénéisant la couche de communication** entre applications



Principes des EAI

- Récupérer, transmettre et traiter** les données issues des applicatifs :
 - par les **messages applicatifs**,
 - au fil de l'eau** et
 - de façon individuelle** (en pseudo temps-réel)
- Définir une **gestion des flux (workflow)** entre les applications :
 - Analyser les **flux d'échanges métiers** entre les applications
 - Définir les **règles de passage des messages** entre application
 - Implanter ces **règles** dans le **moteur de routage des messages** (Message Broker)

Les 3 niveaux d'intégration d'un EAI



Architecture fonctionnelle générale d'une EAI (2)

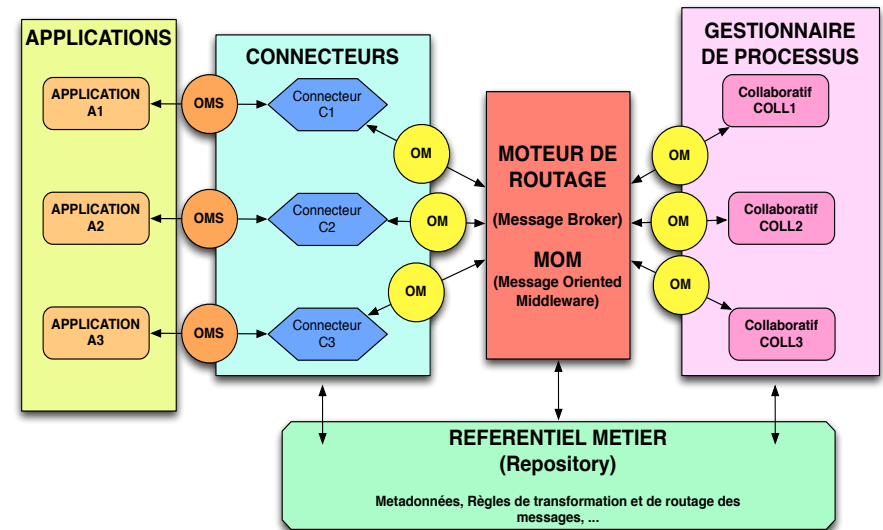
En fonction d'événements préalablement définis, une plateforme EAI :

- récupère les données d'une application
- les transforme en messages OMS - Objets Métiers Spécifiques à l'application
- convertit ces messages OMS dans un format adéquat, les messages OM - Objets Métiers
- puis les route vers leur destination (une autre application), selon une logique de processus métier

Elle se compose principalement de :

- de connecteurs (adaptateurs)
- d'un moteur de routage (Message Broker / MOM - Message Oriented Middleware)
- d'un gestionnaire de processus (ensemble de collaboratifs)
- d'un référentiel métiers (Repository : Metadonnées, Règles de transformation et de routage des messages, ...)

Architecture fonctionnelle générale d'une EAI (2)



Composants d'une EAI (1) : les messages

- Messages OMS - Objets de Métier Spécifiques (Application Specific Business Objects - ASBO) :
 - reflètent les données de l'application (nom du champ, format...)
 - sont construits à partir des données d'une application source par un connecteur (adaptateur) spécifique
 - seront ensuite transformés par ce connecteur en messages standards à l'EAI : les OM
- Messages OM (Objets de Métier Spécifiques) (Business Objects - BO) :
 - Messages standards à l'EAI reflétant le modèle de données global des différents processus de l'entreprise
 - sont transmis à des traitements appelés collaborations qui reflètent la logique de processus à appliquer sur un OM
 - avant de le transmettre à une ou plusieurs applications cible (compléter les infos par recherche dans une autre application, vérification de la validité du processus métier...).

Architecture fonctionnelle d'une EAI (2) : les connecteurs

- Un connecteur (adaptateur) :
 - sert d'interface entre l'EAI et une application avec ou sans intelligence métier
 - il scrute les événements fait l'extraction de données sous forme d'OMS depuis l'application et les transmet transformées en OM à l'EAI
 - il fournit à l'application les données provenant de l'EAI sous forme d'OMS
 - peut fournir des services complémentaires tels que la gestion des exceptions ou des mécanismes de remontée d'erreurs

Architecture fonctionnelle d'une EAI (3) : le Moteur de routage

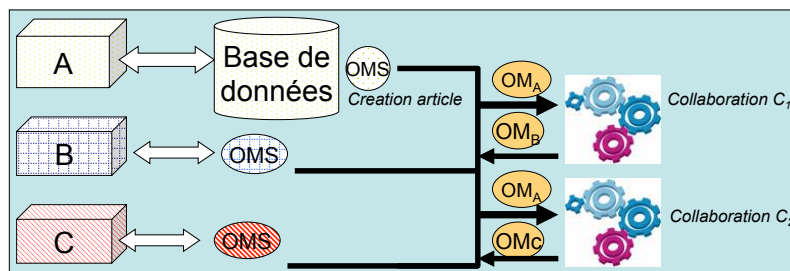
- Le **moteur de routage** (ou moteur d'intégration) :
 - ou « Message Broker », en général un intergiciel orienté messages (**Message Oriented Middleware - MOM**) et *asynchrone*
 - sert d'**interface** entre l'IAE et une application avec ou sans **intelligence métier**
 - **chef d'orchestre de l'EAI**, il administre les **règles** de routage des données, de transformation et de traitement issues du **Référentiel Métier** ou *Repository*
 - il assure les échanges asynchrones entre applications avec des **files d'attente de messages** (*message queues*) et un niveau de **tolérance de panne** : un message n'est pas perdu lorsqu'une application n'est pas prête à le recevoir
 - c'est une **couche logicielle non bloquante** : l'application émettrice du message redevient immédiatement disponible
 - il permet la **communication par publication / abonnement**.

Architecture fonctionnelle d'une EAI (4) : Référentiel et Gestionnaire de processus métier

- **Référentiel (Repository)** : c'est une base de données qui contient :
 - toutes les définitions des structures des données - ou **métadonnées** - échangées,
 - les **formats de messages**, les **règles de transformation** et de **routage** de ces messages, ... pour en faciliter leur maintenance au niveau de l'EAI
- **Gestionnaire des processus métiers (optionnel)** :
 - il permet de **modéliser** et faire **évoluer les processus d'intégration**
 - il **pilote** ces **processus d'intégration** selon une **logique de gestion des flux inter-applicatifs métier** définie dans le Référentiel
 - il **contrôle l'exécution** et le **cadencement** des **processus métiers** réalisés au travers de **collaborations** mises en œuvre par un **moteur de workflow**.

Exemple de fonctionnement d'une EAI (1)

- Soit une **application A** de gestion de commande qui **crée un nouvel article**
- elle veut le rendre disponible à :
 - une **application B** qui suit les anomalies techniques de cet article et à
 - une **application C** qui affiche l'article sur un portail Web

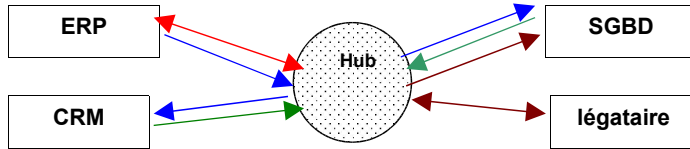


Exemple de fonctionnement d'une EAI (2)

1. L'**appli A** crée un **nouvel article** dans sa base de données.
2. Un **traitement automatique (trigger)** capture cet événement et l'archive dans une table d'événement avec la donnée associée (nouvel article)
3. Un **connecteur EAI JDBC (Base de données)** scrute cette table toutes les 10 secondes et découvre ce nouvel événement.
4. Il **récupère alors la donnée associée et la copie dans un OMS** en lui associant un verbe (création)
5. L'OMS spécifique à l'appli A contenant les données du nouvel article créé est **converti en un OM générique** « Article » reflétant toutes les informations nécessaires à l'entreprise pour représenter un article
6. l'OM « Article » est attendu par **2 collaborations (C1 et C2)** :
 - C1 récupère l'OM, analyse le verbe (création) et envoie l'OM en création vers **l'appli B** (Cet OM est remis en correspondance pour obtenir un article OMS destiné à B et est traité par le connecteur de l'appli B qui effectue la création).
 - C2 récupère l'OM original et l'envoie en création vers **l'appli C** (mappage, connecteur de l'appli C).

Types d'architecture d'EAI (1)

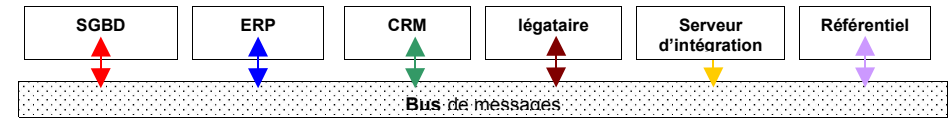
L'architecture HUB - "Hub and spoke" :



- **Modèle centralisé** : tout passe par un "hub" central qui concentre les services sur un seul serveur
 - **Aucun flux n'est possible sans l'entremise de ce hub**
 - Quand une application envoie un message, il est **expédié à destination du hub**
 - Le **référentiel** (la base où sont stockées les règles de routage et de transformation) est donc lui aussi **centralisé**
- **Avantage** : administration grandement facilitée
- **Inconvénient** : gestion de la charge complexe, la seule solution consiste en effet à multiplier les hubs sur les différents segments du réseau, mais il faut alors synchroniser les règles stockées sur ces différents nœuds.

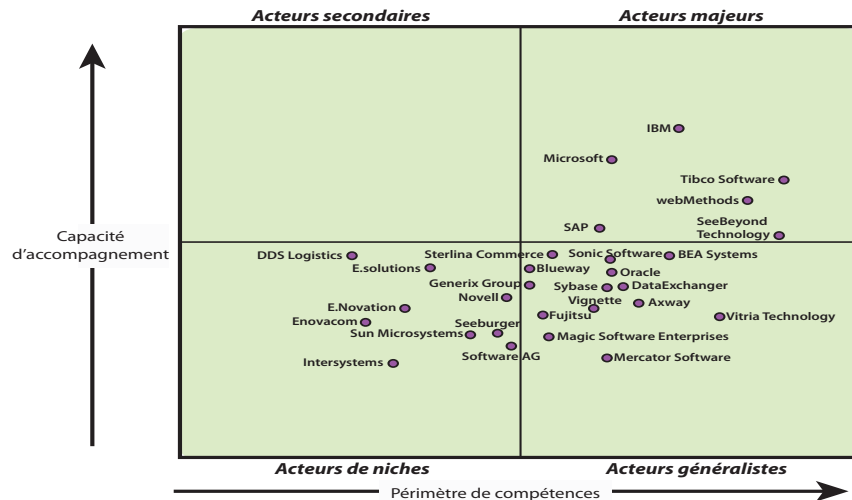
Types d'architecture d'EAI (2)

L'architecture BUS - "Network Centric" ou "Bus Applicatif" :



- **Modèle décentralisée** : l'architecture « bus applicatif » distribue les services sur plusieurs serveurs.
 - Des **référentiels** de règles et des **gestionnaires** de messages sont **disséminés sur l'ensemble des nœuds** (point de connexion à une application).
 - Quand une application émet un message, il est **traité par le référentiel du nœud correspondant** afin que les applications abonnées à ce type de messages le reçoivent.
- **Avantages** : la charge est donc répartie sur l'ensemble des nœuds, meilleures performances que le modèle « Hub »
- **Inconvénient** : mise en œuvre est plus complexe et plus difficile à administrer que le modèle « Hub ».

Les éditeurs d'EAI



Source : Gartner Group (mai 2003)
Révision : CRITT&L (2011)

Les éditeurs d'EAI (2)

Quelques produits commercialisés	Quelques produits libres (open sources)
<ul style="list-style-type: none"> ▪ IBM ▪ NEON ▪ BEA ▪ TIBCO ▪ TSI ▪ Aactiva ▪ Software Technologies ▪ Microsoft : BizTalk server ▪ Crossworld ▪ Vitria ▪ SOPRA ▪ Forté ▪ Template ▪ Viewlocity ▪ ... 	<ul style="list-style-type: none"> ▪ Openadaptor EAI compatible java/tomcat/jdbc ▪ OpenSyncro EAI compatible java ▪ Mule iae compatible java ▪ Proteus EAI compatible java/xalan/jdbc/jms/ftp/tibco ▪ J-EAI de Process One ▪ OpenEAI ▪ ...

Forces et faiblesse de la stratégie EAI

▪ Forces :

- On gagne en **souplesse** et en **réactivité**, en ne développant plus d'interfaces spécifiques point à point entre les applications, au profit d'une **collaboration des applications** autour d'une plate-forme d'EAI
- On réduit les coûts de développement et de maintenance de ces interfaces
- Les **flux sont traités "au fil de l'eau"** ce qui réduit le débit de traitement
- Les flux sont **réutilisables** et **extension aisée du système** à une autre application

▪ Faiblesses :

- **Pas adaptée aux flux massifs**
- **Coût initial élevé**
- **Maintenance de la cohérence des bases pas toujours facile** (pb de synchronisation)

Conclusion sur la stratégie EAI

- Elle constitue ainsi une **alternative aux ERP** (Enterprise Resource Planning) avec une approche plus **modulaire**
- **La mise en œuvre de la stratégie EAI nécessite :**
 - Tout d'abord que des **analystes métier cartographient** le SI de l'entreprise et **modélisent les flux** de données au regard de ses **processus** fonctionnels
 - Ensuite les **architectes de SI** mettent en œuvre EAI en définissant les **composants métier**, l'**extraction des données**, leur **routage** et leur **transformation**
- En incorporant une brique de **modélisation métier**, les plates-formes EAI **séparent la modélisation métier et l'implémentation technique** des processus.
- **L'EAI rentre dans la philosophie de l'«Urbanisation» des SI en simplifiant** le SI et en permettant de le faire plus facilement **évoluer pour suivre la stratégie et l'entreprise.**