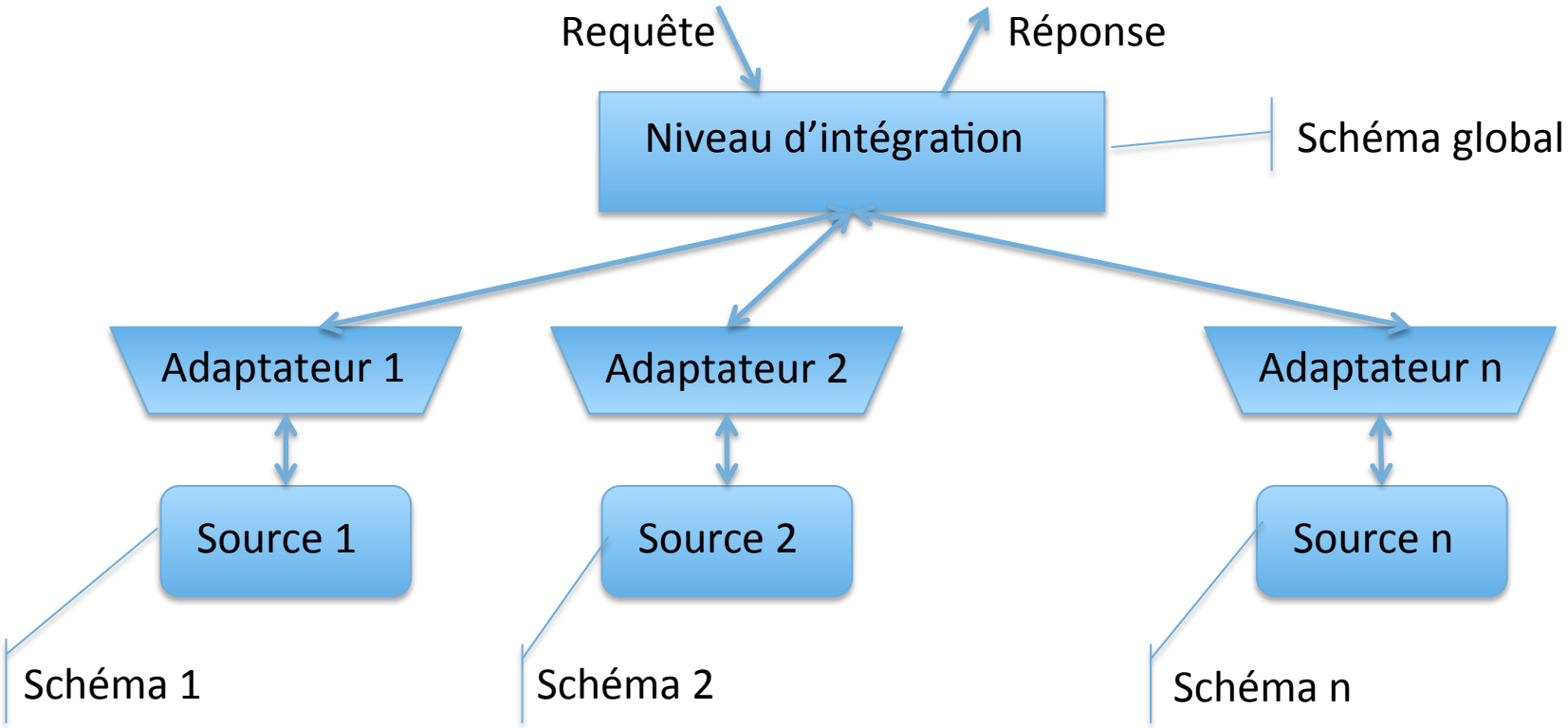
Intégration des données et ETL

Intégration de données

L'intégration de données a pour objectif de permettre un accès uniforme à des données hétérogènes provenant de différentes sources.



Deux approches

- Intégration virtuelle
  - Les données restent dans les sources
  - Les requêtes sont exprimées sur le schéma global, puis décomposées en sous-requêtes sur les sources
  - Les résultats des sources sont combinés pour former le résultat final
- Intégration matérialisée
  - Les données provenant des sources sont transformées et stockées sur un support spécifique (entrepôt de données).
  - L'interrogation s'effectue sur la base matérialisée

Intégration matérialisée

- Nous nous limitons ici à l'intégration matérialisée, i.e. à l'alimentation d'un entrepôt de données.
- L'intégration matérialisée nécessite de
  - **normaliser** les données,
  - gérer le **référentiel** qui garantit l'intégrité des données,
  - connaître la sémantique des données, utiliser pour cela des **métadonnées**

Qualité des données

L'intégration matérialisée des données vers un entrepôt doit garantir la qualité des données :

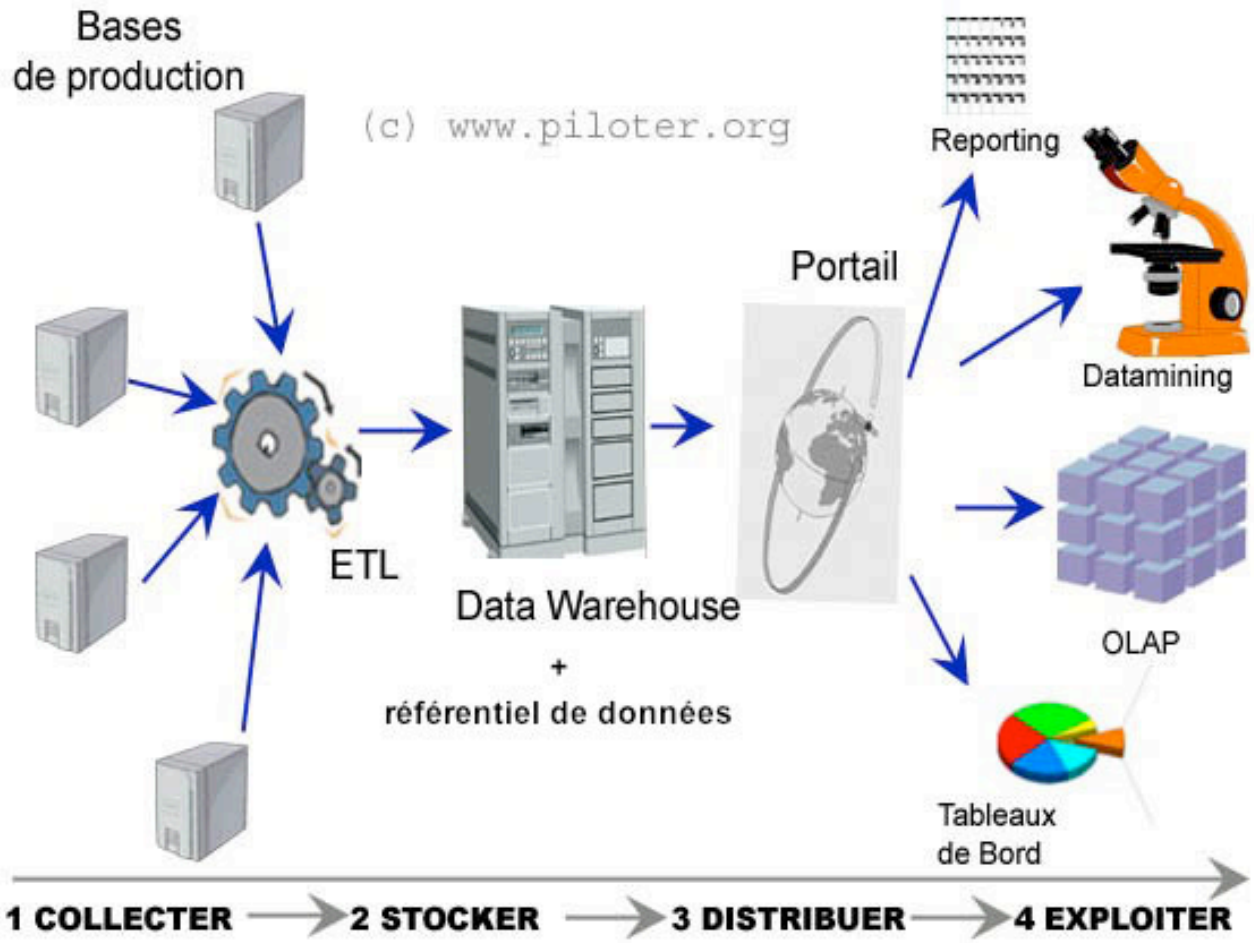
- **Complétude** : toutes les informations nécessaires sont disponibles
- **Consistence** : pas de conflit entre les sources de données
- **Validité** : les données sont correctes
- **Conformité** : par rapport à des formats spécifiques
- **Pertinence** : les données reflètent bien la réalité
- **Intégrité** : les données sont fiables, on retrouve les liens entre elles et on n'introduit pas de doublons.

Différentes solutions d'intégration matérialisée

1. Scripts - *on laissera cette solution de côté*
2. Extract-Transform-Load (ETL)
3. Enterprise Application Integration (EAI)
4. Change data capture (CDC)

ETL

ETL - Rappel du contexte



Définition

- ETL signifie « Extract, Transform, Load »
- Wikipédia :

Il s'agit d'une technologie informatique intergicielle (comprendre middleware) permettant d'effectuer des synchronisations massives d'information d'une source de données vers une autre, et utilisée en particulier

pour le chargement régulier de données agrégées dans les entrepôts de données.

Définition

Un ETL est un système qui permet :

- de découvrir, analyser et **extraire** les données à partir de sources hétérogènes;
- de **nettoyer** et standardiser les données selon les règles de gestion établies par l'entreprise;
- de **charger** les données dans un entrepôt de données et/ou les propager vers les datamarts
- d'offrir un environnement de développement, des outils de gestion de ces opérations.

Extraction

- Hétérogénéité des données : plusieurs sources, qui ont des caractéristiques distinctes car correspondent à des métiers distincts (RH, gestion des stocks, …)
- Hétérogénéité technologique :
  - Sources : bases relationnelles, fichiers excel, ERP, source SaaS
  - Connecteurs : solution adaptée à chaque type de source.

Extraction - Méthode

- **Qui** : Identifier les sources de données et leurs structures
- **Comment** : décider de la façon d'extraire les données, i.e. pour chaque source et aussi de l'ordonnancement de ces tâches d'extraction
- **Quand** : Décider de la fréquence, et pour chaque source, de la fenêtre de temps durant laquelle se fera l'extraction
- Définir comment gérer les exceptions

Staging area – phases E et T

- Les données sont extraites et stockées dans une zone intermédiaire (par exemple sous la forme de fichiers plats), appelée staging area.
- C'est dans cette zone que les données vont pouvoir être nettoyées et transformées avant de les charger dans le datawarehouse.

Extraction

1. Extraction complète :
   - Ensemble des données de l'OLTP à un instant *t* (snapshot)
   - Chargement initial des données, ou rafraîchissement complet dû par exemple à la modification d'une source
   - Coûteuse en temps !
2. Extraction incrémentale
   - Capture uniquement les changements depuis la dernière extraction
   - Extraction différée : on utilise les timestamps des données sources (mais gestion des suppressions compliquées) ou une comparaison avec la précédente extraction (comparaison – coûteuse - de snapshots).

Nettoyage/Transformation

- Problème de **résolution d'entité** : on a la même donnée dans différentes sources mais sans référentiel commun (donc identifiants différents)
- Lié au problème précédent : résolution des **doublons**
  - Equivalence de champs
  - Equivalence d'enregistrements : fusion d'enregistrements
- On trouve des correspondances en utilisant des règles de résolution (par exemple 2 entités sont équivalentes si elles ont au moins N champs identiques)
- Le résultat de la transformation peut utiliser des **référentiels** connus, par exemple les CSP, des adresses conformes aux normes postales, …

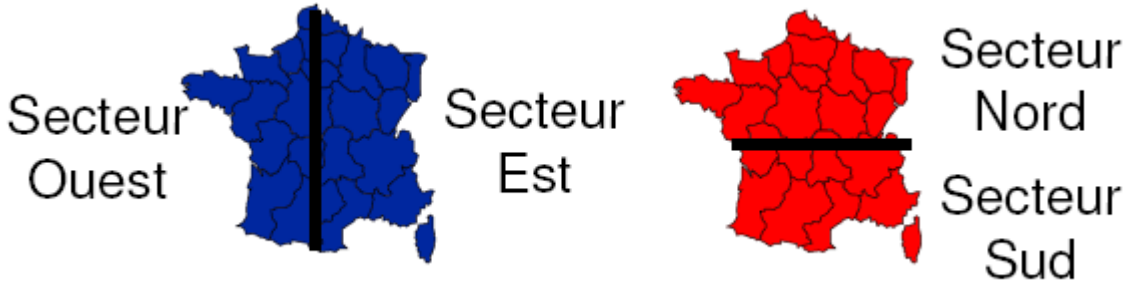
Nettoyage/Transformation

- Problème de **modélisation** : différents modèles de données sont utilisés
- Problème de **terminologie** : un objet est désigné par 2 noms différents, un même nom désigne 2 objets différents

Exemple : couleurs nommées différemment

Nettoyage/Transformation

- **conflit sémantique** : choix de différents niveaux d'abstraction pour un même concept
- **conflit de structures** : choix de différentes propriétés pour un même concept
- **conflit de représentation** : 2 représentations différentes choisies pour les mêmes propriétés d'un même objet



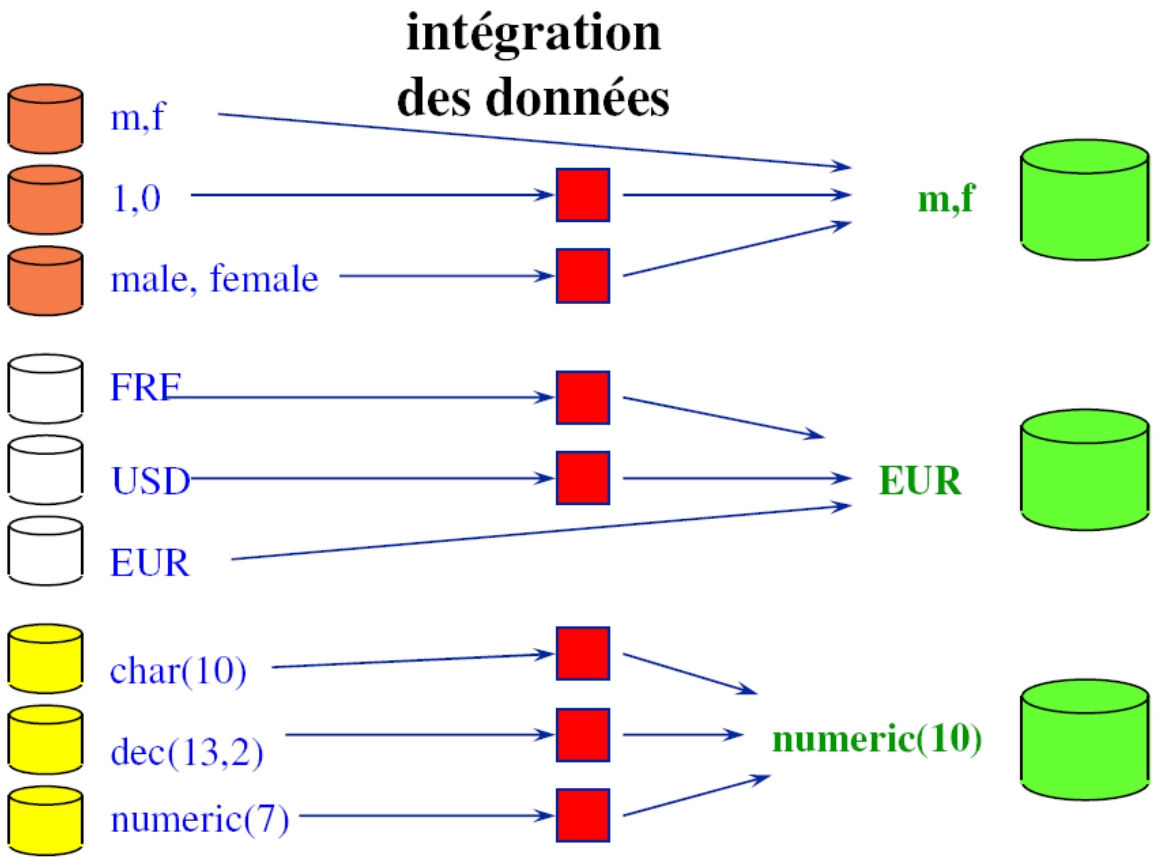
- **Incompatibilités de contraintes** : 2 concepts équivalents ont des contraintes incompatibles

Matrice de transformation

Cible			Source			Transformation
Nom table	Nom colonne	Type de donnée	Nom table	Nom colonne	Type de données	

- Ce document spécifie ce qui est attendu du processus de transformation, i.e. mapping entre les sources et la cible
- Souvent cette transformation peut être exprimée en SQL

Exemples


intégration
des données
m,f
1,0
male, female
m,f
FRF
USD
EUR
EUR
char(10)
dec(13,2)
numeric(7)
numeric(10)

Chargement

1. Chargement/rafraichissement complet :
   - Lors de la création de l'entrepôt, ou lorsque le nombre de modifications par rapport au précédent chargement est trop important
   - Les index et contraintes d'intégrité sont désactivées temporairement
   - Peut prendre plusieurs heures
2. Chargement incrémental
   - Tiens compte de la nature des changements (données historisées ou écrasées, …)
   - Peut être fait en temps réel ou en lots

Chargement

- il faut programmer le chargement par lots à une période creuse (par exemple la nuit)
- Définir la fréquence de rafraichissement de l'entrepôt. Elle est liée à la granularité de la dimension Temps de l'entrepôt.

Métadonnées

L'entrepôt – comme toute base de données – définit des métadonnées sur

- Les données cibles (dimensions, faits, hiérarchies)
- Les contraintes d'intégrité
- Les index, partitions
- Les vues matérialisées

Métadonnées

- Le processus ETL nécessite d'autres métadonnées sur :
  - Les données sources
  - Les règles de nettoyage et transformation
  - La politique de rafraichissement
  - La sécurité
  - La surveillance des process
  - …
- Il faut une BDD pour stocker toutes ces informations.
- Standard : **Common Warehouse Metamodel**, proposé par l'OMG, basé sur UML, XML, SOAP. Dernière version en 2003.

ETL - avantages

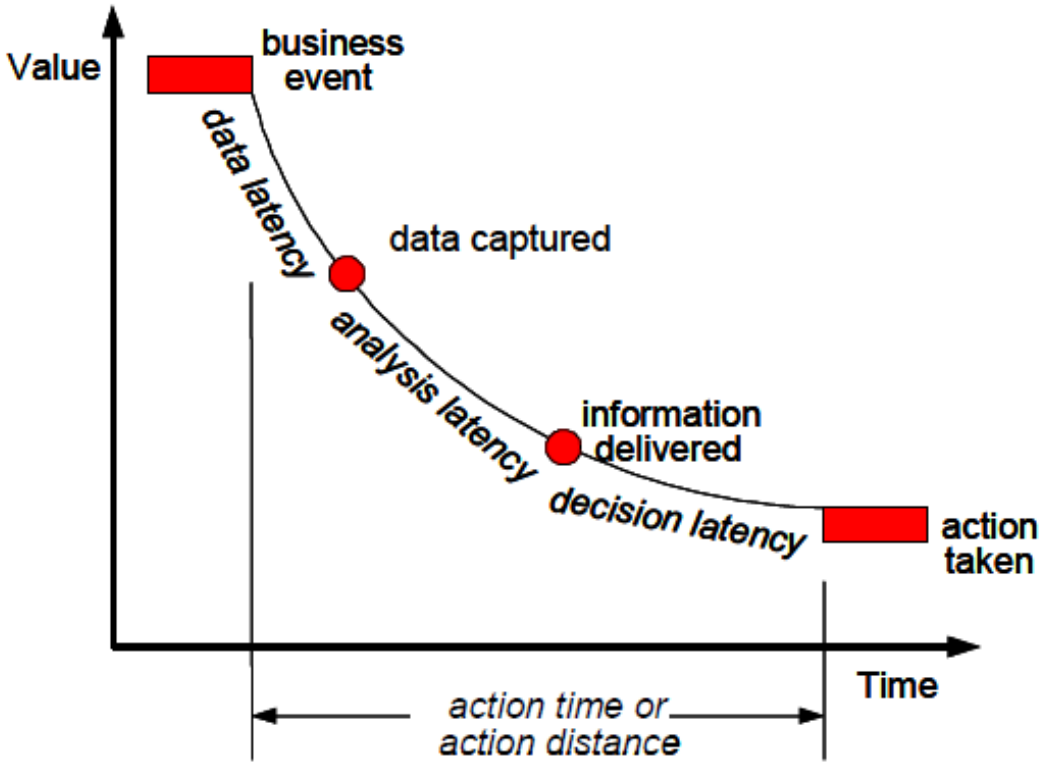
- Un processus ETL peut traiter une grande quantité de données (traitement en lots)
- Transformations complexes, agrégations sur les données
- Nombreux outils sur le marché, interfaces graphiques qui facilitent la prise en main et améliorent la productivité.

ETL- inconvénients

- Espace disque supplémentaire (staging area)
- Unidirectionnel : des sources vers l'entrepôt
- Latence des données entre la source et l'entrepôt
- Complexité : on estime que le développement du système ETL compte pour 50% à 70% des efforts d'un projet de BI.

Vers des approches temps réel : EAI et CDC

Pourquoi ?



Plus on met de temps à capturer la donnée, moins elle a de la valeur.

Illustration Components of Action Time - Richard Hackathorn

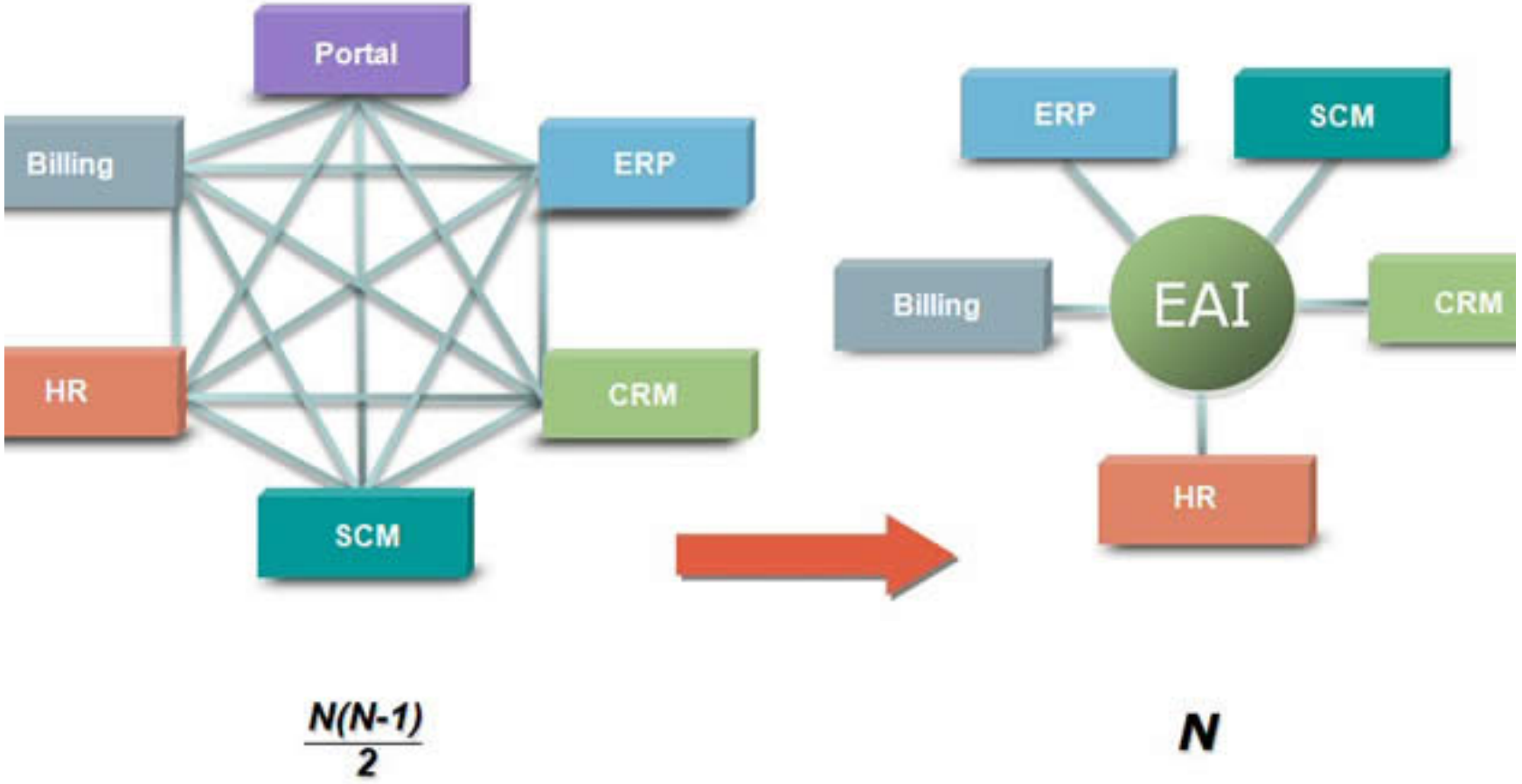
Exemples

- Suivi de ventes en temps réel dans les magasins – optimisation d'une campagne de marketing
- Détection de fraudes
- Optimisation de la logistique – (re)planification de livraisons en fonction du trafic routier.
- Analyse de données provenant de capteurs

EAI – Enterprise Application Integration

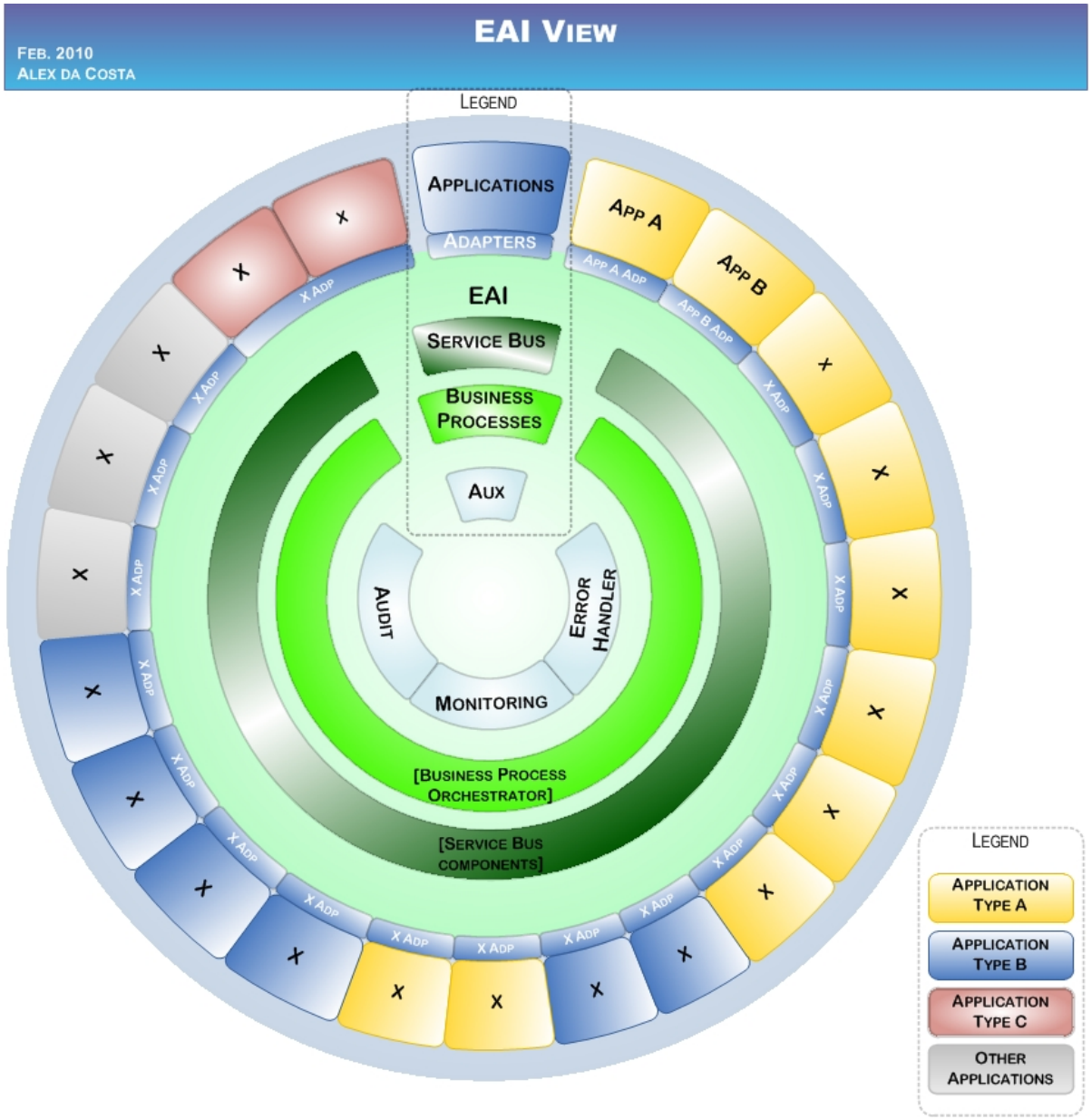
EAI

- La notion d'EAI dépasse largement le cadre de l'alimentation d'un entrepôt
- Avantages :
  - permet d'échanger des données en quasi temps réel
  - évite de développer des connecteurs entre toutes les applications qui doivent échanger des données (cf prochaine diapo).
- Inconvénient principal :
  - Trop lent pour des flux massifs (préférer un ETL dans ce cas)


Portal
Billing
ERP
HR
CRM
SCM
ERP
SCM
Billing
EAI
CRM
HR
N(N-1)
2
N

EAI - fonctionnement

- Des **connecteurs** font l'interface entre les applications et l'EAI. En fonction des événements des **applications sources**, les connecteurs fournissent à l'EAI des données sous la forme d'**ASBO** (Application Specific Business Objects)
- Ces ASBO sont transformés en données standards à l'EAI, les **BO**.
- Ces BO subissent un traitement avant d'être transmis aux **applications cibles**.
- L'acheminement des données entre les applications est effectué par une **couche de transport**.


EAI View
Feb. 2010
Alex da Costa
Legend
Applications
Adapters
App A
App B
App A Adp
App B Adp
X
X Adp
EAI
Service Bus
Business Processes
Aux
Audit
Error Handler
Monitoring
[Business Process Orchestrator]
[Service Bus components]
Legend
Application Type A
Application Type B
Application Type C
Other Applications

Change Data Capture (CDC)

Change Data Capture

- Principe : identification, capture et répercussion des changements des sources
- Une méthode possible : utilisation des journaux (logs) des bases sources
  - Pas d'impact sur la base source
  - Faible latence dans l'acquisition des changements
  - Scan des logs donc on rejoue les transactions en conservant l'ordre dans lequel elles ont été validées (commit)

Exemple : Oracle

Flux d'événements sur les données sources, gestion d'une file d'attente (producteur/consommateur)
La consommation peut être associée à un process de transformation

Comparatif des solutions d'intégration

	Scripts	ETL	EAI	Log-based CDC
Volume des données	moyen	très grand	faible	grand
Fréquence	Par intermitence	Par intermitence	En continu	Intermitence ou continu
Latence	Moyenne/ élevée	Moyenne/ élevée	faible	faible
Intégrité transactionnelle	non	non	non	Possible selon les produits
Transformations	intermédiaires	avancées	basiques	basiques
Coût traitement	Élevé par intermitence	Élevé par intermitence	Moyen mais continu	Faible et continu

Oracle white paper – Data integration architectures for operational data warehousing - sept 2012