



# Architecture technique

- Besoins d'affaires:
  - « Que doit-on faire ? »
- Architecture:
  - « Comment allons-nous le faire ? »



# La valeur de l'architecture

- Encourage la satisfaction des besoins:
  - Les besoins techniques dérivent des besoins d'affaires;
  - Documents d'architecture.
- Facilite la communication:
  - Illustre les différents rôles au sein du système;
  - Communique la complexité du projet aux cadres supérieurs.
- Aide à la planification:
  - Regroupe tous les détails techniques;
  - Identifie des dépendances et de nouveaux de besoins.
- Flexibilité, productivité et maintenance:
  - Métadonnées, sélection d'outils, etc.

# Facteurs à considérer [1/2]



- L'interdépendance informationnelle entre les unités de l'entreprise
  - Ex: bonne intégration (ex: MDM) VS silos de données
- Les sources de données
  - Ex: 1 source VS 10 sources, ERP VS legacy, etc.
- La quantité des données
  - Ex: gigaoctets VS teraoctets
- La latence des données
  - Ex: mise-à-jour hebdomadaire VS temps-réel
- L'urgence d'obtenir une solution fonctionnelle
  - Ex: entrepôt d'entreprise (EDW) VS magasin de données

# Facteurs à considérer [2/2]



- Le nombre d'utilisateurs
  - Ex: 10-50 utilisateurs vs 50-200 utilisateurs
- La nature des tâches des utilisateurs finaux
  - Ex: rapports simples VS fouille de données
- Les contraintes sur les ressources
  - Ex: financières, main d'œuvre, biais technologique, etc.
- Les objectifs du projet
  - Ex: stratégique VS opérationnel
- Autres facteurs
  - Ex: politiques, habilités du personnel TI, etc.

# Architectures et métadonnées

- **Métadonnées:**

- **Définition:** « *information définissant et décrivant les structures, opérations et le contenu du système de BI* ».
- **Métadonnées techniques:**
  - ETL: sources et cibles pour les transferts de données, transformations, règles d'affaires, etc.
  - Stockage: tables, champs, types, indexes, partitions, dimensions, etc.
  - Présentation: modèle de données, rapports, cédules, privilèges d'accès, etc.
- **Métadonnées d'affaires:**
  - Décrit le contenu de l'entrepôt dans des termes compréhensibles par les utilisateurs d'affaires;
  - Ex: descripteurs de tables et champs.
- **Métadonnées de processus:**
  - Décrit le résultat de diverses opérations du système de BI;
  - Ex: logs ETL (début, fin, écritures disque, ...), statistiques sur les requêtes, etc.

# Architectures et métadonnées

- Bénéfices:
  - Découple la dépendance entre la technologie et son utilisation (ex: reconfigurer dynamiquement le système ETL pour modifier ou ajouter une source)
  - Permet de monitorer l'état et la performance de la solution BI
  - Sert de documentation au système
  - Permet de déterminer l'impact d'un changement
- Idéal:
  - Avoir un seul répertoire de métadonnées partagé par toutes les composantes de la solution BI

# Architectures et couches de service

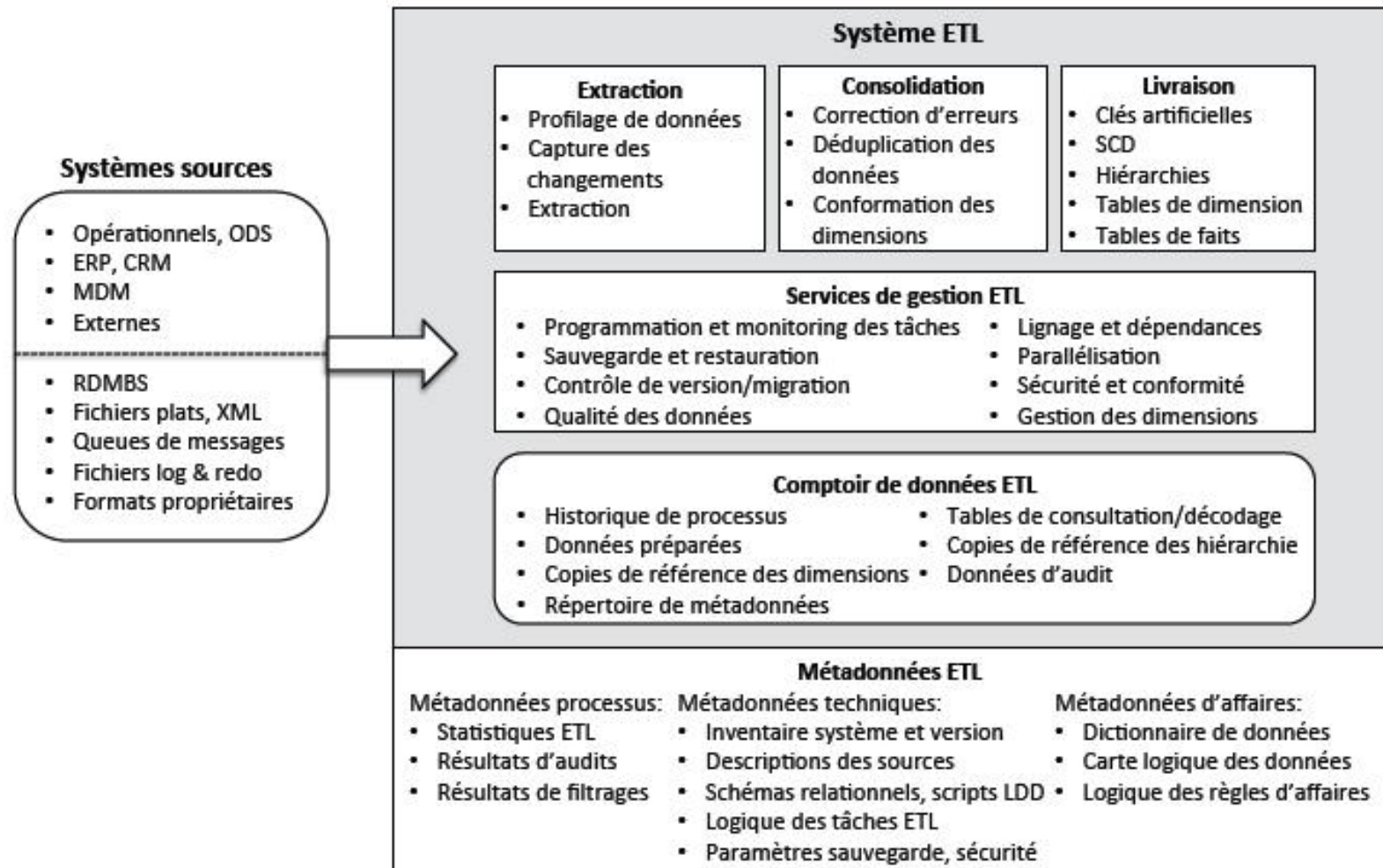
- Service oriented architectures (SOA):
  - Méthode d'intégration et de développement de systèmes dans laquelle les fonctionnalités sont regroupés autour de processus d'affaires et offerts sous la forme de services interopérables;
  - Permet la communication entre des systèmes qui n'ont pas été conçus dans cette optique, et leur participation conjointe dans des processus d'affaires.
- Dans le contexte des entrepôts de données:
  - Mécanisme pour échanger des données d'un système à un autre (ex: d'une application vers un ODS, d'un MDM vers une application, etc.);
  - Réduit les dépendances technique permettant une approche « *best-of-breed* ».



# LES COMPOSANTES DE L'ARCHITECTURE



## Couche de préparation de données (*back-room*)





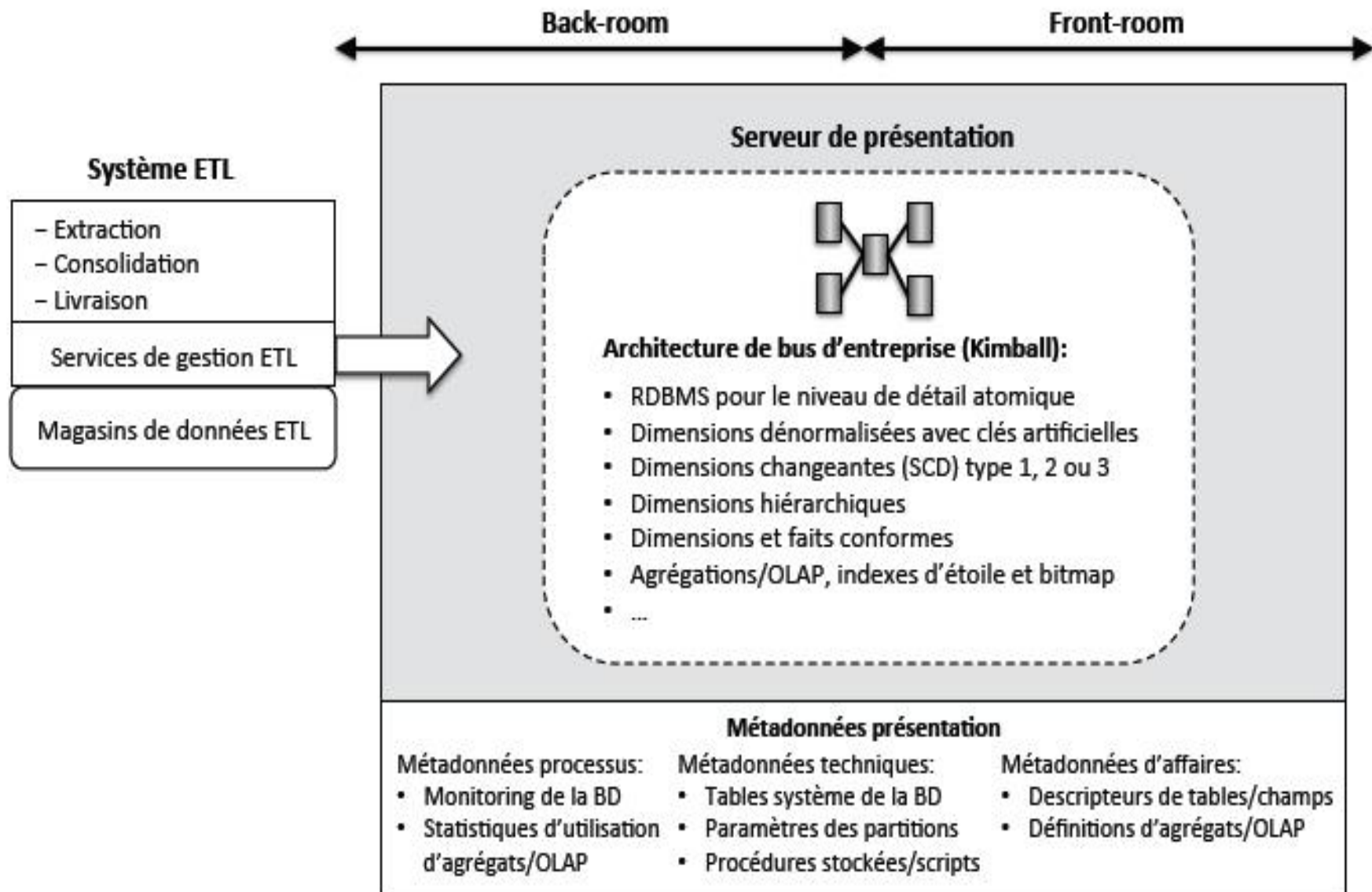
# Couche de préparation de données (*back-room*)

- Besoins généraux:
  - Support à la productivité (ex: environnement de développement)
  - Convivialité (ex: interface graphique simple)
  - Flexibilité (ex: métadonnées)
- Fonctionnalités ETL:
  - **Extraction:**
    - Ex: profilage des données, capture des changements, copie des données
  - **Consolidation:**
    - Ex: règles de transformation, résolution d'incohérences, intégration
  - **Livraison:**
    - Ex: insertion dans les tables de faits/dimensions, gestion des changements (SCD)

# Couche de préparation de données (*back-room*)

- Services de gestion ETL:
  - Planification de tâches (*job scheduler*)
  - Sauvegarde/restauration
  - Sécurité
  - etc.
- Comptoir de données ETL (*data store*):
  - Données temporaires d'extraction (*staging area*)
  - Historique du processus ETL (métadonnées processus, QA)
  - Sauvegarde des références ETL (métadonnées techniques)
  - etc.

# Couche de stockage de données (*presentation*)



## Couche de stockage de données (*presentation*)



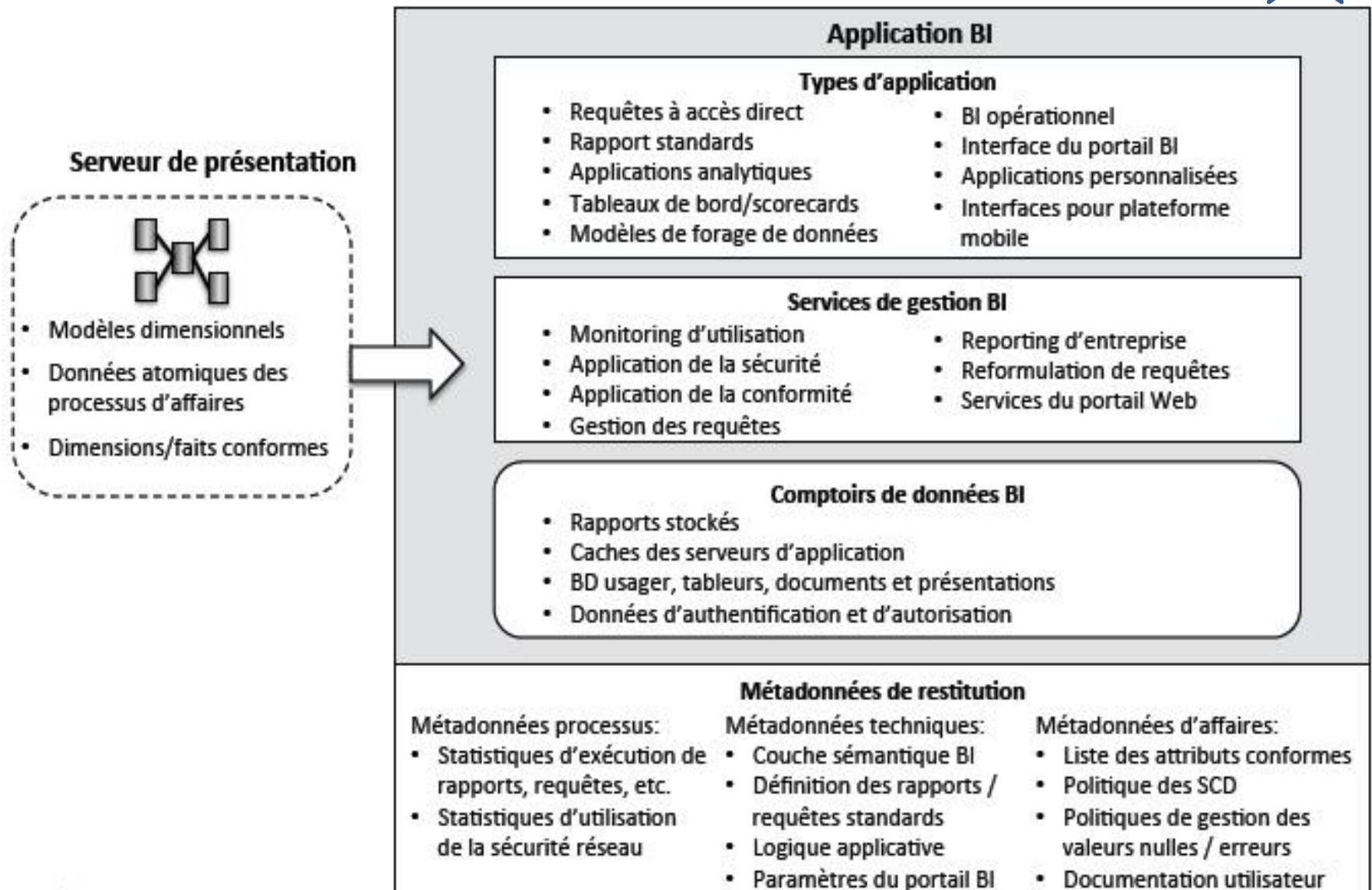
- Objectif:
  - Fournir un accès simplifié et rapides aux données, pour les utilisateurs (ex: requêtes ad hoc) et applications de BI.
- Caractéristiques souhaitées:
  - Données provenant des principaux processus d'affaires
  - Données atomiques ET agrégées
  - Source **unique** de données à tous les utilisateurs (peu importe l'emplacement physique des données)
  - Analyses variées avec les mêmes données



## Couche de stockage de données (*presentation*)

- Considérations:
  - Tables de dimensions dénormalisées (schéma en étoile)
  - Clés artificielles
  - Dimensions à évolution lente (SCD 1, 2, 3)
  - Dimensions conformes basées sur la matrice en bus de données
  - Données atomique au niveau des transactions
  - Stratégies d'agrégation (ex: OLAP, ROLAP, etc.)
  - Stratégies de performance (ex: index, partitionnement, etc.)
  - etc.

# Couche de restitution de données (*front-room*)



## Couche de restitution de données (*front-room*)



- Objectifs:

- Supporter les besoins analytiques des utilisateurs
  - Ex: rapports, analyse OLAP, fouille de données, etc.
- Offrir des interfaces d'accès simplifiés aux données
  - Ex: portail Web, service SOA
- Offrir une performance adéquate

- Services de gestion BI:

- Gestion des requêtes
  - Reformulation/optimisation
  - Redirection vers la bonne ressource informationnelle
  - Navigation d'agrégation
  - Gestion de priorité
- Gestion de la sécurité/accès
- Monitoring de la l'utilisation/performance



## Couche de restitution de données (*front-room*)

- Comptoirs de données BI:

- Modèles de rapports
- Cache du serveur d'application (performance)
- Magasin de données locaux (attentions aux silos de données)
- etc.



# ARCHITECTURES PARTICULIÈRES

## Les magasins de données

- Caractéristiques:
  - Contient une portion du contenu de l'entrepôt de données;
  - Se concentre sur 1 sujet d'analyse (ex: les ventes OU les livraisons, mais pas les deux);
  - Sert à faire des analyses simples et spécialisées (ex: les fluctuations des ventes par catégorie de produits);
  - Nombre de sources limitées, provenant la plupart du temps d'un même département;
  - Extraction et transfert de données rudimentaires, souvent fait par transfert de fichier ou du code maison;
  - Même processus de conception que les entrepôts de données, mais demande moins de ressources.



# Magasins vs entrepôts de données



Caractéristique	Magasin de données	Entrepôt de données (EDW)
<b>Portée</b>	Un domaine d'analyse	Plusieurs domaines d'analyse
<b>Temps de développement</b>	Mois	Années
<b>Coûts de développement</b>	\$ 10,000 à \$ 100,000 +	\$ 1,000,000+
<b>Complexité de développement</b>	Faible à moyenne	Grande
<b>Taille des données</b>	Mb à plusieurs Gb	Gb jusqu'à plusieurs Pb
<b>Horizon des données</b>	Courantes et historiques	La plupart du temps historiques
<b>Transformation des données</b>	Faible à moyenne	Importante
<b>Fréquence des mises-à-jour</b>	Horaire, journalier ou hebdomadaire	Peu aller jusqu'à mensuel
<b>Nombre d'utilisateurs simultanés</b>	Dizaines	Centaines à milliers
<b>Types d'utilisateur</b>	Analystes dans le domaine spécifique et gestionnaires	Analyste d'entreprise et cadres seniors
<b>Objectifs d'affaires</b>	Optimisation des activités dans le domaine spécifique	Optimisation inter-fonctionnelle et support à la décision

Source: E. Turban, R. Sharda, D. Delen et D. King (2010). « Business intelligence: A managerial approach », Pearson.

# Magasins de données opérationnelles

- Operational data store (ODS):
  - Environnement informationnel et analytique reflétant à tout instant les données intégrées et consolidées provenant de sources hétérogènes.
- ODS vs entrepôts de données classiques:
  - Contiennent rarement des données historiques;
  - Met à jour les données au lieu de les ajouter;
  - Effectue les changements presque instantanément au lieu de les faire en lot;
- Utilisation des ODS:
  - Intégration permet d'avoir des règles d'affaires complexes impliquant des données de plusieurs sources;
  - Analyse OLAP.

## ODS: Exemple d'utilisation

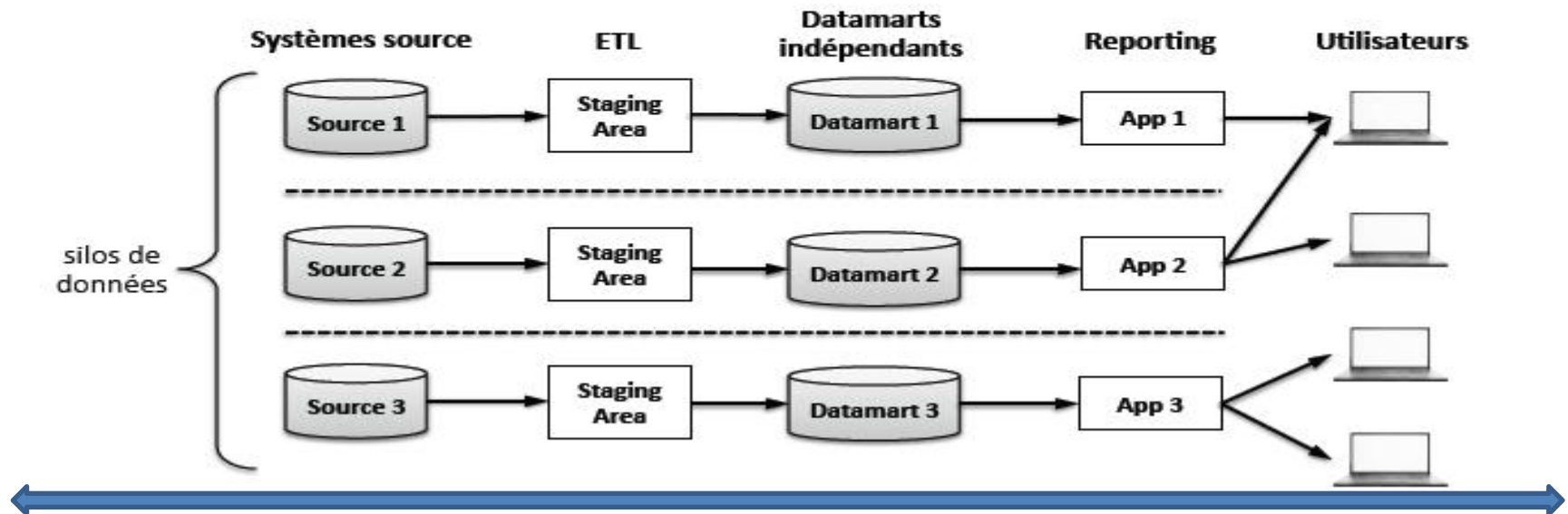
- Une entreprise bancaire vient de faire l'acquisition d'une compagnie d'enquête de crédit;
- Les comptes, placements, et dossiers de risque des clients sont gérés par des applications différentes;
- Afin d'approuver un nouveau prêt à un client l'entreprise doit s'assurer de la solvabilité de ce client;
- Cette règle d'affaires nécessite l'intégration et la consolidation de données provenant de plusieurs applications;
- Tout changement aux données doit être propagé presque en temps-réel afin d'appliquer la règle d'affaires sur les données *actuelles*.



## Les architectures d'entrepôts de données

1. Magasins de données indépendants
2. Architecture en bus de magasins de données
3. Architecture *Hub-and-spoke*
4. Entrepôt de données centralisé
5. Architecture fédérée

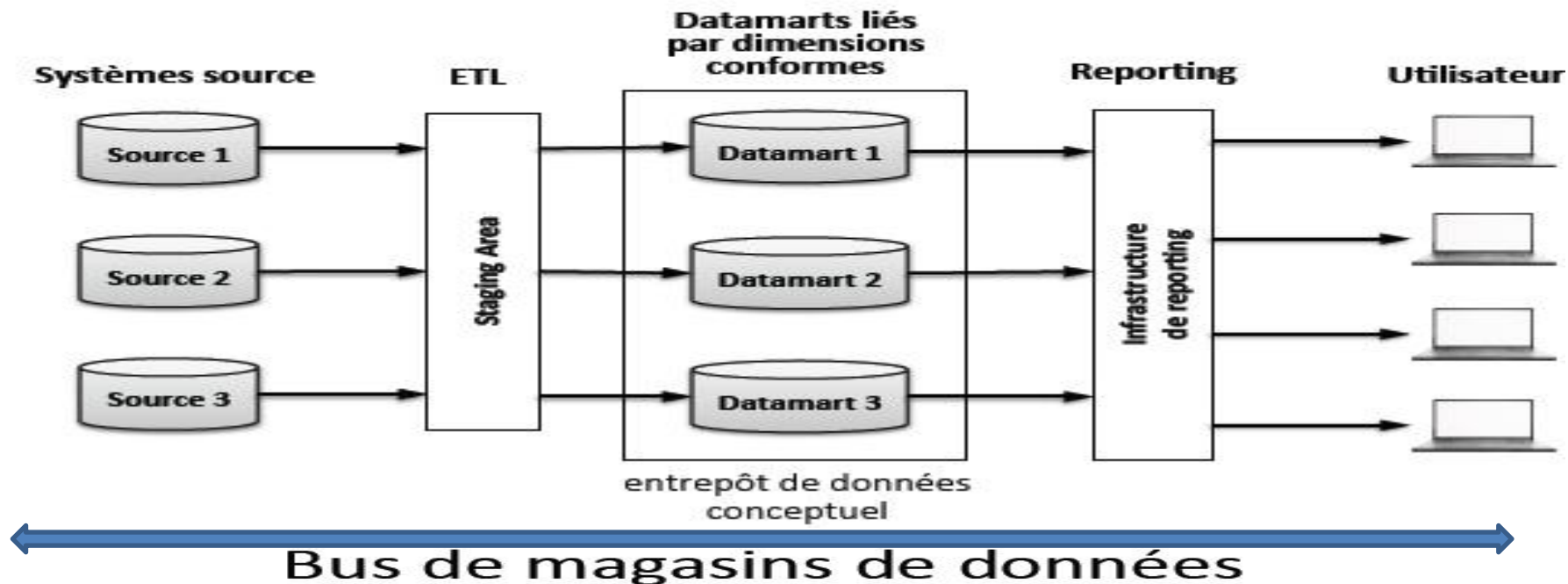
# Magasins de données indépendants



## Magasins de données indépendants

- **Caractéristiques:**
  - Les datamarts sont développés et opèrent de manière indépendante;
  - Les données sont disposées en « silos fonctionnels »;
  - Pas de dimensions conformes.
- **Avantage:**
  - Architecture la plus simple et la moins coûteuse à développer;
- **Inconvénients:**
  - Incohérences et redondances entre les datamarts (ex: dimensions, définitions, mesures, types, etc.);
  - Il n'y a pas *une seule* version de la vérité;
  - Analyse inter-fonctionnelle difficile ou impossible;
  - Vision limitée, pas extensible.

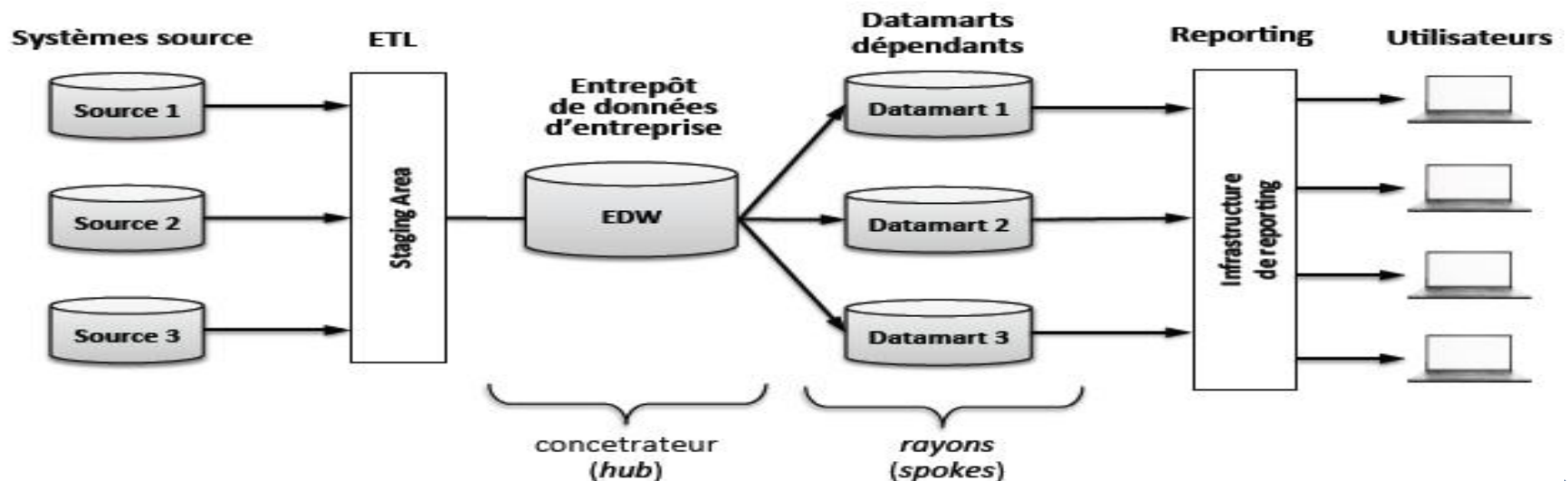
# Bus de magasins de données



- **Caractéristiques:**
  - Approche *bottom-up*, proposée par R. Kimball;
  - Datamarts développés par sujet/processus d'affaires, en se basant sur des dimensions conformes;
  - Modélisation dimensionnelle (*schéma en étoile*), au lieu du diagramme entité-relation;
  - Entrepôt de données conceptuel, formé de magasins de données inter-reliés à l'aide d'une couche d'intergiciels (*middleware*).
- **Avantages:**
  - Intégration des données assurée par les dimensions conformes;
  - Approche incrémentale (processus les plus importants d'abord);
  - Donne des résultats rapidement.
- **Inconvénients:**
  - Itérations futures difficiles à planifier;
  - Performance sous-optimale des analyses impliquant plusieurs datamarts.



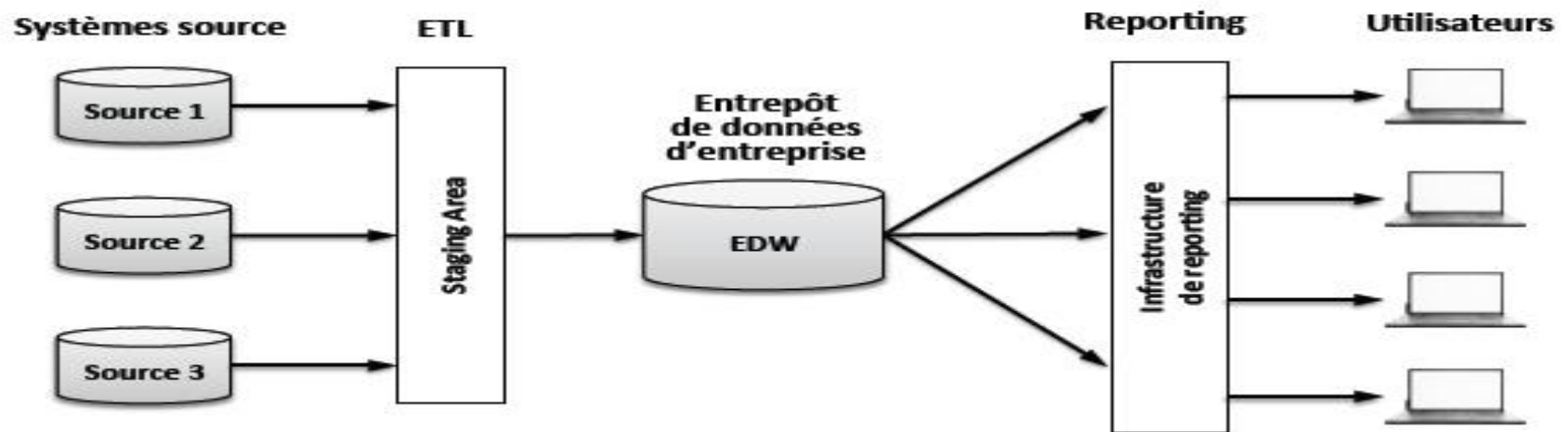
# Architecture Hub-and-spoke (Corporate Information Factory)



## Architecture Hub-and-spoke (Corporate Information Factory)

- **Caractéristiques:**
  - Approche *top-down*, proposée par B. Inmon et al.
  - Entrepôt (*hub*) contient les données **atomiques** (c.-à-d. le niveau de détail le plus fin) et **normalisées** (3FN);
  - Les datamarts (*spokes*) reçoivent les données de l'entrepôt;
  - Les données des datamarts suivent le modèle dimensionnel et sont principalement résumées (pas atomique);
  - La plupart des requêtes analytiques sont faites sur les datamarts.
- **Avantages:**
  - Intégration et consolidation complète et des données de l'entreprise;
  - Approche itération et facilement extensible.
- **Inconvénients:**
  - Peut avoir de la redondance de données entre les datamarts;
  - Performance sous-optimale des analyses impliquant plusieurs datamarts.

# Entrepôt de données centralisé

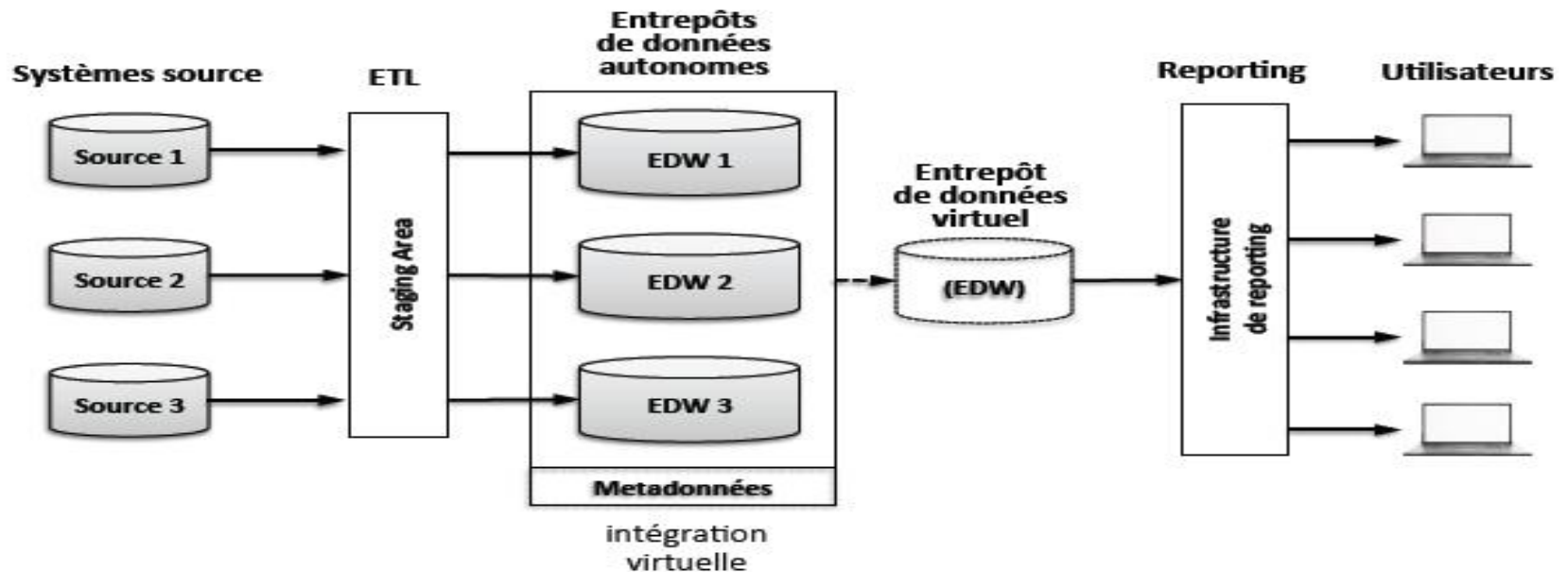


## Entrepôt de données centralisé

- **Caractéristiques:**
  - Similaire à Hub-and-spoke, mais sans les datamarts dépendants;
  - Un gigantesque entrepôt de données servant l'entreprise entière;
  - Les données peuvent être atomiques ou résumées.
- **Avantages:**
  - Les utilisateurs ont accès à toutes les données de l'entreprise;
  - Intégration (ETL) et maintenance facile car les données sont à un seul endroit;
  - Performance optimale (ex: appliance warehouse, *Teradata*).
- **Inconvénients:**
  - Long et coûteux à développer;
  - Pas incrémental;
  - Extensibilité limitée ou très coûteuse.



# Architecture fédérée



## Architecture fédérée

- **Caractéristiques:**
  - L'entrepôt de données est distribué sur plusieurs systèmes hétérogènes;
  - S'opère de manière transparente (l'utilisateur ne voit pas que les données sont réparties);
  - Les données sont intégrées logiquement ou physiquement à l'aide de méta-données (ex: XML);
  - Complément plutôt que remplace (selon les experts).
- **Avantages:**
  - Utile lorsqu'il y a déjà un entrepôt en place (ex: acquisitions ou fusions de compagnies);
  - Demande peu de ressources matérielles additionnelles.
- **Inconvénients:**
  - Très complexe: synchronisation, parallélisme, concurrence, etc.
  - Peu de contrôle sur les sources et la qualité des données;
  - Faible performance (... mais la technologie s'améliore).

# Comparaison entre les architectures



## Popularité:

Architecture	Fréquence
Hub-and-spoke	39 %
Bus de datamarts	26 %
Entrepôt centralisé	17 %
Datamarts indépendants	12 %
Entrepôts fédérés	4 %

Source: T. Ariyachandra et H. Watson (2005). « Key factors in selecting a datawarehouse architecture », *Business Intelligence Journal*, vol. 10, no. 2.



## Critères:

- Qualité de l'information (précise, complète, cohérente);
- Qualité du système (flexible, extensible, intégration);
- Impact sur les individus (productivité, décisions, etc.);
- Impact sur l'entreprise (satisfaction des requis, ROI, etc.).

## Résultats:

Architecture	Qualité de l'information	Qualité du système	Impact sur les individus	Impact sur l'entreprise
Hub-and-spoke	5.35	5.56	5.62	5.24
Bus de datamarts	5.16	5.60	5.80	5.34
Entrepôt centralisé	5.23	5.41	5.64	5.30
Datamarts indépendants	4.42	4.59	5.08	4.66
Entrepôts fédérés	4.73	4.69	5.15	4.77

Source: T. Ariyachandra et H. Watson (2005). « Key factors in selecting a datawarehouse architecture », *Business Intelligence Journal*, vol. 10, no. 2.

# Approche *top-down* vs *bottom-up*

Caractéristique	Top-down (B. Inmon)	Bottom-up (R. Kimball)
<b>Objectifs</b>	Livrer une solution technologiquement saine basée sur des méthodes et technologies éprouvées des bases de données	Livrer une solution permettant aux usager d'obtenir facilement et rapidement des réponses à leurs requêtes d'analyse
<b>Complexité de la méthode</b>	Plutôt complexe	Plutôt simple
<b>Importance de la conception physique</b>	Importante	Peu importante
<b>Orientation du modèle</b>	Orienté données	Orienté processus d'affaires
<b>Accessibilité des utilisateurs finaux</b>	Faible	Forte
<b>Outils de conception</b>	Traditionnels (diagrammes entité-relation et flot de données)	Modélisation dimensionnelle (schéma en étoile)
<b>Auditoire principal</b>	Professionnels en TI	Utilisateurs finaux

Source: E. Turban, R. Sharda, D. Delen et D. King (2010). « Business intelligence: A managerial approach », Pearson.

# Entrepôts de données hébergés (cloud)

- Caractéristiques:

- L'infrastructure matérielle et informatique réside sur le site d'un fournisseur;
- L'entreprise loue l'infrastructure.

- Avantages:

- Minimisent l'investissement dans l'infrastructure;
- Libèrent les ressources matérielles et humaines de l'entreprise;
- Évitent les tâches de mise-à-jour et de maintenance.

- Inconvénients:


- Moins rentable à long terme;
- Sécurité et domaine privé des données.

# Solutions clé en main

- Appliance data warehouse:
  - Ensemble intégré de serveurs, dispositifs de stockage, DBMS, systèmes d'exploitation et de logiciels pré-installés et pré-optimisés pour l'entreposage de données;
  - Utilisent une architecture de traitement massivement parallèle;
  - Solution allant du *terabyte* au *petabyte*.
- Avantages:
  - Faibles coûts de mise-en-place et de maintenance;
  - Bonnes performance et extensibilité due à l'architecture parallèle;
  - Permet d'obtenir rapidement des bénéfices.

# PROCESSUS DE DÉVELOPPEMENT DE L'ARCHITECTURE

- Questions selon le niveau de détail:



Niveau de détail	Back-room	Front-room
Besoins d'affaires et audit de données	<ul style="list-style-type: none"> <li>Comment obtenir les données nécessaires aux besoins d'affaires ?</li> </ul>	<ul style="list-style-type: none"> <li>Comment mesurer, suivre, analyser et faciliter les opportunités d'affaires ?</li> </ul>
Implications architecturales et modèles	<ul style="list-style-type: none"> <li>Quelles sont les fonctions et composantes nécessaires pour obtenir les données dans la forme, l'endroit et le moment désirés.</li> <li>Quels sont les principaux magasins de données et services et où sont-ils situés ?</li> <li>Quel est la stratégie de métadonnées ?</li> </ul>	<ul style="list-style-type: none"> <li>Que requièrent les utilisateurs pour avoir l'information dans une forme utilisable ?</li> <li>Quelle est la stratégie de portail BI ?</li> </ul>
Modèles détaillées et spécifications	<ul style="list-style-type: none"> <li>Quel est le contenu spécifique de chaque magasin de données ?</li> <li>Quel sont les capacités spécifiques de chaque service ?</li> </ul>	<ul style="list-style-type: none"> <li>À quoi ressemblent les rapports standards ?</li> <li>Comment ceux-ci seront-ils présentés ?</li> <li>Quel est le design du portail BI ?</li> </ul>
Sélection de produit et implémentation	<ul style="list-style-type: none"> <li>Quels produits fournissent les capacités requises ?</li> <li>Comment ceux-ci seront-ils assemblés ?</li> </ul>	<ul style="list-style-type: none"> <li>Quels produits fournissent les capacités requises ?</li> <li>Comment ceux-ci seront-ils assemblés ?</li> </ul>

# Document d'implications architecturales

- Exemple:

Besoins d'affaires	Implication architecturale	Sous-système	Valeur / priorité
Améliorer le taux de réponse à l'aide d'une stratégie de vente croisée	Outils d'intégration permettant de coupler les clients avec les produits	ETL	H / 8
	Création de listes de vente croisée et monitoring de base à l'aide d'outils BI	App. BI	H / 7
	Traitement des offres et suivi des réponses par le système CRM	App BI	N/A
Améliorer le taux de réponse à la campagne par courriel en fournissant aux analystes des outils pour générer les listes de clients ciblés	Application analytique	App. BI	H / 7
Augmenter la précision des prédictions de vente à l'aide d'une meilleure historique de données et de meilleurs modèles analytiques	Application analytique avec prédiction de séries temporelles	App. BI / forage de données	N/A
	Extraire de l'information des systèmes externes pour le suivi des ventes	ETL	H / 8

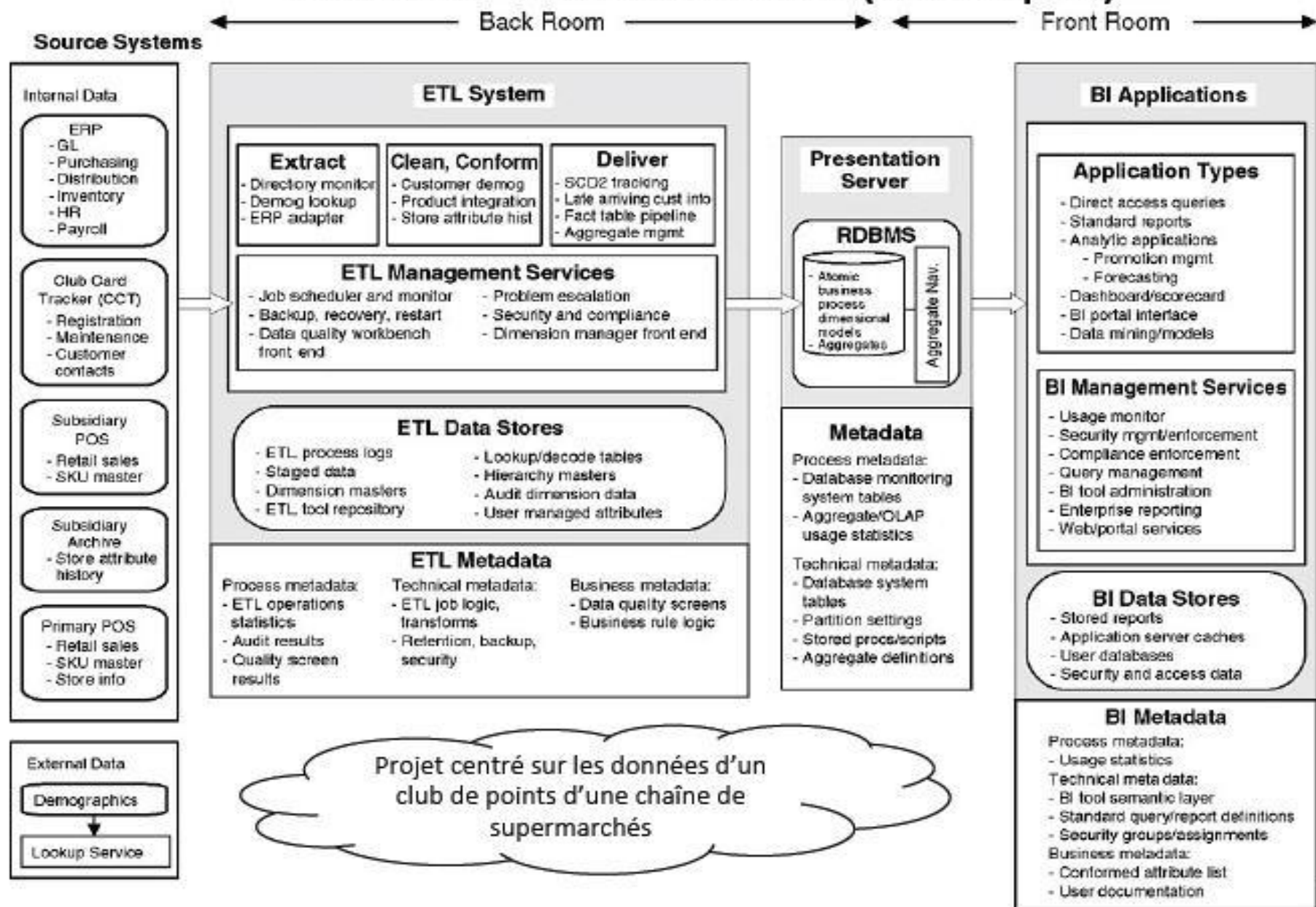


# Document de plan architectural

- Contenu:

1. Description sommaire du projet et ses objectifs;
2. Méthodologie;
3. Besoins et implications architecturales;
4. Survol de l'architecture
  - Ex: modèle haut-niveau, métadonnées, couches de service, etc.
5. Composantes architecturales principales
  - Ex: ETL, applications BI, sources de données, répertoire de métadonnées, infrastructure, etc.
6. Processus de développement de l'architecture
  - Ex: phases, preuve de concept, standards et sélection de produits, etc.
7. Modèle architectural.

# Modèle architectural (exemple)



# Sélection des produits

Guidée par les besoins d'affaires;

## Étapes:

1. Comprendre le processus d'achat de l'entreprise;
2. Faire une étude de marché:
  - **Sources:** internet, cours et séminaires, publications du domaine, consultants externes, etc.;
  - **Critères:** fonctionnalité, performance, productivité, support (technique, documentation, formation), etc.
3. Évaluer les solutions les plus prometteuses
  - Ex: rencontres avec les vendeurs, version d'essai, comparaison de prototypes, etc.
4. Rédiger un rapport de recommandation de produit;
5. Tester le produit retenu durant une période d'essai (ex: 90 jours);
6. Négocier le contrat (licences, support, formation, etc.).

## Exemple:

## Matrice d'évaluation de produits

ETL Tool Product Evaluation Worksheet Example						
	Project Weight	Vendor One	Vendor Two	Vendor Three	Vendor Four	Vendor Five
<b>Core Functionality</b>						
Ease of installation and maintenance	40					
Support for key sources (e.g. DB2, Oracle, SQL Server, ERP)	30					
Support for key targets (e.g. SQL Server, MOLAP engine)	10					
Full featured scripting language	10					
Extensible	10					
Execute steps across platforms	10					
Uses fast load facilities on target	10					
<b>Core Functionality</b>	<b>120</b>					
<b>Transformation Functionality</b>						
Slowly changing dimension mgmt (Type 2)	40					
Data quality screen management	40					
Fact table pipeline key substitution	30					
Late arriving dimension handling	30					
Lookups/validation against large tables	20					
Surrogate key management	20					
Late arriving fact handling	20					
Complex joins; outer joins	20					
Change data capture & propagation	20					
Built-in knowledge of ERP system internals	15					
Aggregation build and management	10					
Complex calculations	10					
Exception/error row handling	10					
Source filtering & validation	10					
<b>Transformation Functionality</b>	<b>295</b>					
<b>Performance</b>						
Test scores (standard platform and ETL script)	60					
Support for parallel execution	50					
Scalability options (add CPUs, clusters, distributed, etc.)	40					
Database drivers tuned for performance	30					
Performance management and monitoring	30					
<b>Performance</b>	<b>200</b>					
<b>Productivity (specific functionality hidden)</b>	<b>170</b>					
<b>Operations and Job Control (specific functionality hidden)</b>	<b>105</b>					
<b>Metadata (specific functionality hidden)</b>	<b>145</b>					
<b>Vendor Info (specific vendor requirements hidden)</b>	<b>190</b>					
<b>TOTAL</b>	<b>6125</b>					