Data Warehouse Release 3

Dossier d'architecture logicielle (SAD)

Version 1.2

Historique des modifications

Date	Version	Description	Auteur
18/11/2010	0.1	Première version pour revue interne CSC	Romuald Dumonceaux
21/11/2010	0.2	Corrections selon remarques de Michael Delire	Romuald Dumonceaux
15/12/2010	0.3	Adaptations après revue par SPF Finances	Romuald Dumonceaux
07/01/2011	1.0	Adaptations après seconde revue par SPF Finances et Logica	Romuald Dumonceaux
12/01/2011	1.1	Précisions au sujet de l'usage de MS SQL Server pour le stockage optionnel de data marts (6.4 et 7.3.3.1).	Romuald Dumonceaux
01/03/2011	1.2	Introductions des principes de l'auditing/logging et corrections après comité tactique.	Romuald Dumonceaux

Tables des matières

1. Présentation du document

1.1 Objet du document

Ce document fournit les informations nécessaires à la compréhension de l'architecture technique du projet de data warehouse pour la gestion des risques liés aux entreprises.

Ce projet est également connu sous les noms « Data warehouse II » ou « Data warehouse release 3 ».

1.2 Documents de référence

Sigle utilisé dans le document	Référence	Intitulé
SUPDEV_NOMMAGE	Convention_de_nommage-3.doc	Données/Conventions de nommages
DCC_ETL_ACCEPTANCE	20100120_DCC ETL Technical Acceptance Criteria_v110.pdf	Critères techniques d'acceptation pour projet ETL
DCC_DS_NAMING	20101129_v200_DataStage_Naming_Conventions.pdf	Conventions de nommage pour DataStage
DWH2-<*X*>[1]-REQ	DWH2_<X>_UserRequirements_vy.z.doc ***StarTeam path:*** *\DWH2\ FUP_02_Requirements*	User Requirements
DWH2-<*X*>[1]-SRC	DWH2_<X>_DescriptionSources_vy.z.doc ***StarTeam path:*** *\DWH2\ FUP_03_Analysis*	Analyse des sources
DWH2-<*X*>[1]-LOGICMOD	DWH2_<X>_ModèlesLogiques_vy.z.doc ***StarTeam path:*** *\DWH2\FUP_03_Analysis*	Modèle logique des données
DWH2-<X>[1]-ETLSPEC	DWH2_<X>_SpecificationsETL_vy.z.doc ***StarTeam path:*** *\DWH2\ FUP_04_Implementation*	Spécifications détaillées des ETL
DWH2-OPSMANUAL	DWH2_ProceduresOperationnelles_vx.x.doc ***StarTeam path:*** *\DWH2\ FUP_09_Environment\ FUP_01_Guidelines*	Manuel d'opérations
IAM_LOGGING_DESIGN	AIM Logging Design - V1.4.odt	Identity and Access Management Logging Design Specifications

[1]: <X> correspond au code de release.

L'ensemble de la documentation relative à ce projet est stocké sous StarTeam, dans le projet « DWH2 ».

2. Introduction

2.1 Scope du document dans sa version actuelle

Le SPF Finances a lancé au début des années 2000 un projet de modernisation baptisé Coperfin, qui s'inscrit dans le cadre de la réforme Copernic de l'administration fédérale.

Divers programmes ont été mis sur pied pour mettre en pratique ces nouvelles modalités. Un des seize programmes du projet de réorganisation est le programme Gestion des risques (en matière d'assistance, de contrôle et de recouvrement). Le projet *Data Warehouse / Datamining – Analyse des risques* fait partie de ce programme.

Dans ce contexte, une pré-étude a été réalisée, non seulement pour préciser les besoins exacts en termes d'utilisateurs et pour évaluer les avantages escomptés d'un environnement analytique pour la Gestion des risques, l'assistance, le contrôle et le recouvrement, mais aussi pour dresser un premier inventaire des sources d'information et définir une architecture de haut niveau ainsi qu'un modèle de données général.

Le présent document traite des aspects techniques de la mise en œuvre d'une deuxième version du Data Warehouse pour la gestion des risques, l'assistance, le contrôle et le recouvrement. Cette version couvre le sujet "Entreprise" qui comprend les aspects liés aux thèmes Personne, Aspect déclaratif, Traitement spécifique et Risque.

2.2 Limitations

Les projets "Data Warehouse/Business Intelligence" sont des projets dont la mise en œuvre est très différente des projets de logiciels transactionnels (plus courant au sein du SPF Finances). Avec pour conséquence la plus importante, le fait que la méthodologie FUP est partiellement applicable pour ce projet.

Ce document tente de respecter au mieux la méthodologie FUP. Certaines déviations sont cependant nécessaires et précisées dans les chapitres correspondants.

2.3 Définitions et Abréviations

Termes / Abréviations	Explication
ODS	Operational Data Store
DWH	Data Warehouse (Entrepôt de données)
DM	Data Mart (Magasin de données)
FTP	File Transfert Protocol
ETL	Extract-Tranform-Load (Intégration de données)
OLAP	On Line Analytical Processing
BI	Business Intelligence
SSRS	Microsoft SQL Server Reporting Services
SSAS	Microsoft SQL Server Analysis Services

3. Objectifs, contraintes et mécanismes de l'architecture

3.1 Respect des Standards

Mécanisme de Design	*Mécanismes d'Implémentation*	*Version*	*Service Responsable*
Intégration de données	IBM InfoSphere DataStage	8.1 FixPack 3	DCC
Stockage des données	IBM DB2	9.5	DCC
Business Intelligence	Microsoft SQL Server	2008 R2	DCC
Scheduling	Absyss Visual TOM	-	OPS
Identity Authentication Management	IAM	-	IAM
Modélisation de données	Embarcadero ER/Studio Data Architect	>8.5	SupDev
Gestion du code et de la documentation	Borland StarTeam	-	SupDev
Suivi des problèmes	HP Quality Center	-	SupDev

Les outils suivants ne sont pas utilisés car ils ne conviennent pas à ce type de projet:

- Borland Caliber
    - Logiciel pour récolte des "requirements" fonctionnels de logiciels
    - Orienté vers des scénarios (cas d'utilisation), pas adapté à l'intégration de données ou à la production de rapports.
- Borland Together
    - Logiciel de modélisation d'applications logiciel
    - Orienté essentiellement vers la modélisation de logiciels via UML
    - Des capacités de modélisations de bases de données sont disponibles mais sont limitées par rapport à Embarcadero ER/Studio. Notamment:
        - Pas de possibilité de documenter les transformations de données dans le modèle (data lineage)
        - Modèle logique en notation UML: moins clair que notation Entités-Relations, utilisation massive de stéréotypes, etc…
        - Nombreuses fonctionnalités de productivité absentes comparé à la solution de Embarcadero
- HP LoadRunner
    - Logiciel pour automatiser des tests
    - Surtout adapté pour tester des logiciels transactionnels, les résultats ne seraient pas suffisamment représentatifs pour ce projet.
    - LoadRunner ne permettrait pas d'automatiser complètement les tests des outils BI (pas d'interface de commande disponible)

3.2 Ergonomie

Dans ce projet, l'ergonomie est définie par les outils utilisés (IBM InfoSphere Information Server et Microsoft SQL Server) et ne peut pas être modifiée.

3.3 Performance

La solution doit pouvoir traiter des volumes importants de données dans des fenêtres de temps limitées.

L'impact sur les systèmes sources de données doit être aussi faible que possible. Pour cela, les données seront échangées via des fichiers plats ou via des bases de données de références (copie des systèmes de production).

Les objectifs de performances à atteindre seront définis pour chaque flux de données. Les critères à renseigner sont donc:

- ***Pour les flux d'intégration de données***
    - Temps de traitement et de chargement d'un volume défini de données sources. Ce volume sera choisi pour être représentatif du volume de données qui sera traité habituellement.
- ***Pour les bases de données data warehouse et data mart***
    - Temps maximum admissible pour exécuter des requêtes prédéfinies. Ces requêtes seront représentatives d'un usage typique du data warehouse et de chaque data marts.
- ***Pour les usages BI***
    - Temps maximum admissible pour réaliser des scénarios BI prédéfinis. Ces scénarios seront représentatifs d'un usage typique, par exemple: Affichage d'un rapport précis.

Les tests de performance se feront dans des conditions de charge permettant un fonctionnement optimal des traitements (CPU, mémoire et I/O).

Les mesures des performances se feront de la façon la plus appropriée pour chaque type de test, à savoir:

- ***Pour les flux d'intégration de données***
    - Utilisation des métadonnées opérationnelles (voir chapitre 5: "Vue données" pour plus de détails)
- ***Pour les bases de données data warehouse et data mart***
    - Utilisation de outils standard des clients DB2 ou de la commande UNIX "time"
- ***Pour les usages BI***
    - Mesure au chronomètre

3.4 Capacité

Le dimensionnement pourra être adapté au fur et à mesure des développements des flux en fonction des volumes de données, du nombre d'utilisateurs et des performances nécessaires.

Le tableau suivant donne les ordres de grandeur qui sont pris en compte pour la mise au point de l'architecture et des développements.

Métrique	Unité	Ordre de grandeur
Volume de données dans le Data Warehouse	Gb	~500
Volume de données par data marts	Gb	~100

Nombre d'utilisateurs total	Utilisateurs	<50

3.5 Disponibilité et Fiabilité

Le data warehouse est conçu pour offrir une disponibilité élevée, spécialement pendant les heures habituelles de bureau. Le data warehouse n'est toutefois pas considéré comme un système critique. Il n'est donc pas conçu pour garantir une disponibilité permanente (type 24/7).

L'infrastructure et les logiciels utilisés sont constitués des standards du SPF Finances (Serveurs, réseau, bases de données…) et sont considérés comme suffisamment fiables.

L'architecture du projet se concentre donc sur la robustesse des traitements de données et les capacités de redémarrage en cas de problèmes.

La robustesse des traitements de données est notamment assurée par l'intégration en plusieurs étapes successives. Chacune de ces étapes remplit des fonctions élémentaires et de complexité croissante (préparation des données, stockage avec historique, transformation vers modèles de données DWH et DM). Les traitements présentant le moins de risques sont exécutés en priorité. En cas de problèmes graves ne pouvant pas être gérés par les méthodes standards de l'outil DataStage, les traitements peuvent reprendre à partir de l'étape précédente, ce qui évite de recommencer ce qui a déjà été exécutés avec succès.

Pour faciliter les diagnostiques, des métadonnées opérationnelles supplémentaires viennent compléter celles générées en standard par DataStage. Des informations de haut niveau sur l'état des flux de données sont donc disponibles.

3.6 Sécurité

La plateforme utilisée dans ce projet est constituée de plusieurs composants distincts qui ont des comportements sensiblement différents vis-à-vis de la sécurité. La suite de cette section distingue systématiquement les composants suivants:

- Outils de développement et de maintenance pour l'intégration de données (Exemples : DataStage Designer, DataStage Director, Information Server Console…). Référencés "Clients Information Server" ci-dessous.
- Base de données du data warehouse (IBM DB2), référencé "DWH DB2" ci-dessous
- Objets BI accessibles via web (Exemples : Rapports Microsoft SQL Server Reporting Services Web Service - SSRS). Référencé "BI Web reports" ci-dessous.
- Objets BI accessible via des clients lourds (Exemples : Cubes Microsoft SQL Server Analysis Services - SSAS). Référencé "Other BI objects" ci-dessous.

3.6.1 Identification

Le tableau suivant résume les méthodes d'indentification ("login") utilisées par chaque composant de la solution:

Composant	Identification
Clients Information Server	Logiciels clients
DWH DB2	Logiciels clients
BI Web reports	IAM Access Manager
Other BI objects	Logiciels clients

3.6.2 Authentification

La structure actuelle des services d'authentification du SPF Finances et les logiciels utilisés imposent des contraintes importantes pour l'authentification.

Par conséquent, deux catégories d'utilisateurs peuvent être distinguées :

- Les développeurs "ETL" et les utilisateurs techniques
    - Ces utilisateurs accèdent aux ressources techniques via un login/mot de passe propre aux plateformes UNIX/Solaris.
- Les utilisateurs consommateurs de données
    - Ces utilisateurs accèdent aux ressources via le login/mot de passe du service IAM ou du domaine Microsoft Windows du SPF Finances (Les utilisateurs et les groupes définis dans l'Active Directory sont provisionnés avec les informations d'IAM).
    - Si nécessaire, l'accès aux données du data warehouse se fait via des utilisateurs techniques gérés de façon transparente par les serveurs procurant les services aux utilisateurs.

Le tableau suivant résume les méthodes d'authentification utilisées par chaque composant de la solution :

Composant	Authentification
Clients Information Server	Operating system (LDAP/OPS)
DWH DB2	Operating system (LDAP/OPS)
BI Web reports	IAM Policy Manager
Other BI objects	Windows Active directory

3.6.3 Autorisation

Pour les même raisons que pour l'authentification, des contraintes importantes sont également présentes ici. Les logiciels utilisés ne permettent pas de gérer finement les autorisations via un répertoire tiers.

Le tableau suivant résume les méthodes d'autorisation utilisées par chaque composant de la solution :

Composant	Autorisation
Clients Information Server	Serveur "IBM Information Server"
DWH DB2	Base de données DB2
BI Web reports	IAM Access Manager
Other BI objects	Windows Active directory

3.6.4 Confidentialité des données signalétiques

Par soucis de protection de la vie privée, il est nécessaire de limiter l'accès aux données permettant d'identifier des personnes (physiques ou morales).

Au niveau du data warehouse, seules les fonctions de sécurité avancées de la base de données IBM DB2 (LBAC) seront utilisées pour restreindre l'accès à ce type d'information.

Pour les data marts, plusieurs techniques peuvent être utilisées en fonction des besoins et des spécificités du data mart: sécurité avancée (DB2/LBAC ou fonctions similaires de SQL Server), anonymisation par agrégations des données, absences de données nominatives, chiffrage ou hachage des données nominatives...

3.6.5 Audit

Les actions des utilisateurs vis-à-vis des données sensibles doivent être stockées dans le service de logging d'IAM. Des informations détaillées sur ce service sont disponibles dans un document séparé. Voir paragraphe 1.2 pour la référence vers cette documentation.

Le tableau suivant résume les actions notées et la technique de récolte de l'information.

Composant	Action(s) auditées	Récolte de l'information
Information Server	Accès aux données	Via DWH DB2
DWH DB2	Accès aux données	Via fonctions d'audit de DB2 ou triggers sur tables importantes (à définir)[1]
BI Web reports	Accès aux rapports	Via logging de SSRS[2]
Other BI objects	Accès aux rapports et cubes	Via logging de SSRS et tracing de SSAS[2]

[1]: Dans le cas des accès aux données via les services SQL Server, il n'est pas possible de connaitre l'utilisateur initial. Seul l'utilisateur technique utilisé par SQL Server pourra être identifié.

[2]: L'utilisateur initial sera identifié dans ce cas.

3.7 Administration

L'administration du data warehouse dépend essentiellement des outils utilisés (IBM InformationServer, IBM DB2, MS SQL Server). Les manuels d'administration de ces outils sont donc les premières références :

Pour IBM InfoSphere Information Server 8.1 :
http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r1/index.jsp

Pour IBM DB2 9.5 :
http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp

Pour Microsoft SQL Server 2008 :
http://technet.microsoft.com/en-us/library/bb418439%28SQL.10%29.aspx

Les procédures d'administration opérationnelle seront détaillées dans un document séparé, disponible sous StarTeam. Voir paragraphe 1.2 pour la référence vers cette documentation.

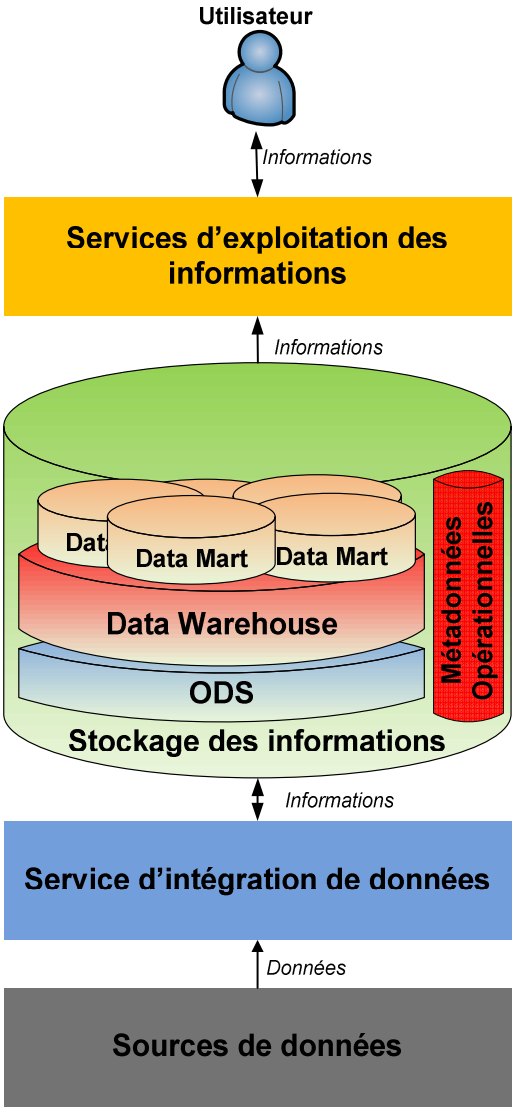
4. Vue Cas d'utilisation

Les cas d’utilisation ne sont pas applicables dans le cadre de ce projet.

5. Vue Logique

5.1 Architecture logique

Le système peut être représenté à haut niveau selon le schéma suivant.



Sources de données

Les sources de données représentent tous les systèmes externes du projet et qui lui fournissent des données "brutes". Ces sources sont considérées comme des "boites noires" et seules leurs interfaces avec le service d'intégration de données sont connues du projet.

Ces interfaces seront détaillées pour chaque source dans le document de description des sources, disponible sous StarTeam (voir paragraphe 1.2 pour la référence complète). Différents types sont possibles, par exemple fichiers plats via FTP, tables de base de données, etc…

Les flux de données sont à sens unique, depuis les sources de données vers le service d'intégration de données. Seul l'accès en lecture aux sources est donc permis.

Service d'intégration de données

Ce service a pour mission d'extraire, transformer et charger (ETL) les données des sources en informations consolidées destinées aux utilisateurs.

Stockage des informations

Les informations préparées par le service d'intégration de données sont stockées à ce niveau-ci.

Les données y sont organisées en trois couches successives :

L'**ODS** (Operational Data Store) est une couche technique –non accessible par les utilisateurs- qui permet de stocker toutes les données des sources en y ajoutant des informations historiques.

Le **data warehouse** stocke l'information consolidée

Les **data marts** stockent des informations dérivées du data warehouse et préparées pour des besoins précis.

Le flux de données avec le service d'intégration de données peut-être à double sens puisque le stockage est organisé en plusieurs couches internes.

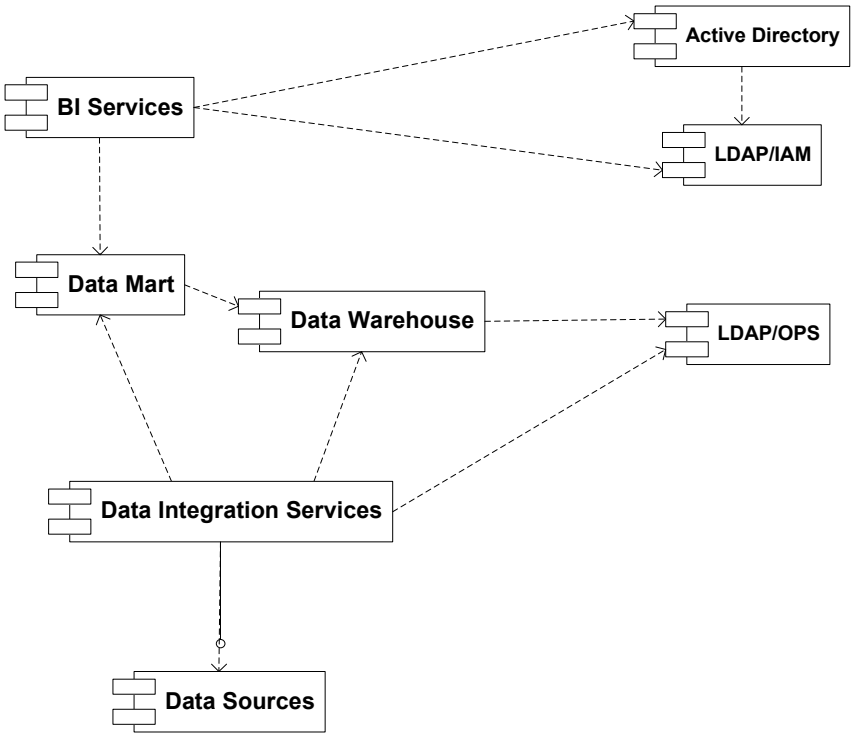
La zone des **métadonnées opérationnelles** stocke des informations au sujet du suivi des traitements de données ainsi que sur la qualité des données.

Services d'exploitation des informations

Ces services permettent aux utilisateurs d'exploiter les données de façon intuitive, sans connaitre nécessairement les détails techniques sous-jacents. Exemples de services: Rapportage, Analyse ("OLAP"), Data Mining (projet distinct de ce projet mais client des données), etc…

6. Vue Implémentation

6.1 Aperçu global – Diagramme UML des Composants



Le service d'intégration de données traite les données provenant des sources via son interface. Ce service est assuré par un outil dédié (IBM Information Server – DataStage), la structure interne est donc indépendante de ce projet.

Les données traitées et consolidées sont stockées dans le data warehouse selon une structure relationnelle et normalisée.

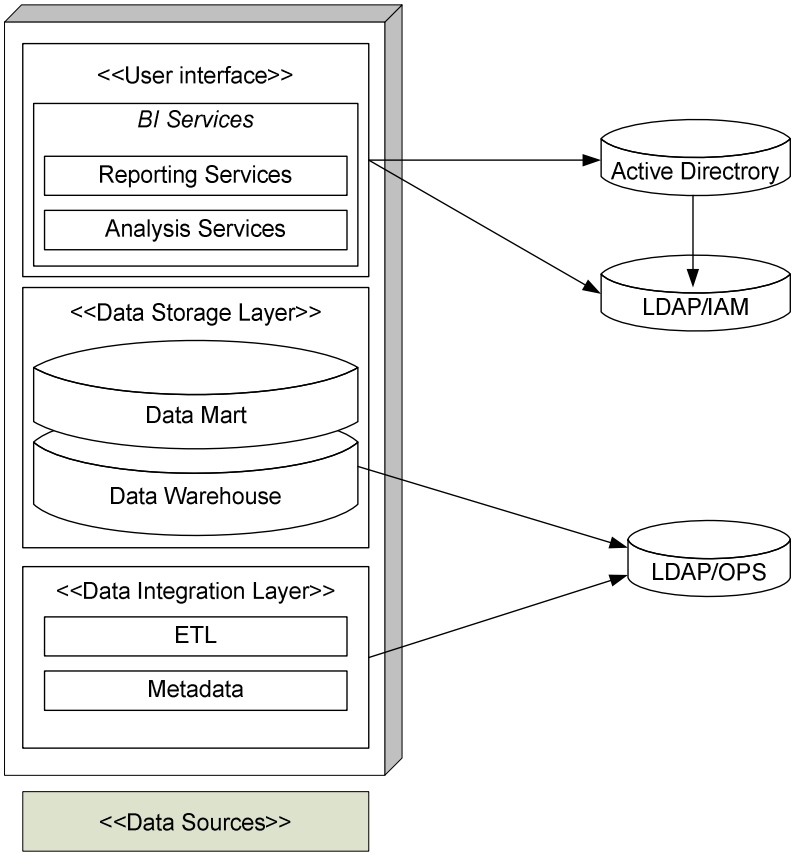
Une partie des données est ensuite dérivée des données du data warehouse par le service d'intégration de données pour être stockées dans les data marts. Les structures de stockages sont adaptées à l'usage final du data mart et reflètent généralement des aspects métiers particuliers des consommateurs de données.

Les utilisateurs accèdent aux données au moyen des services BI. Ces services peuvent être accessibles via des clients lourds ou via le web. Ces services sont assurés par un outil dédié (Microsoft SQL Server), la structure interne est donc indépendante de ce projet.

L'authentification des utilisateurs du service d'intégration de données (développeurs de flux ETL) et des utilisateurs du data warehouse est assurée par le système d'exploitation (Solaris) qui se base sur un répertoire propre (LDAP/OPS).

L'authentification des utilisateurs de services BI se fait via un "active directory" Windows si un client lourd est utilisé et via IAM lorsque l'accès se fait via le web. L' "active directory" est synchronisé avec l'annuaire LDAP associé au service IAM.

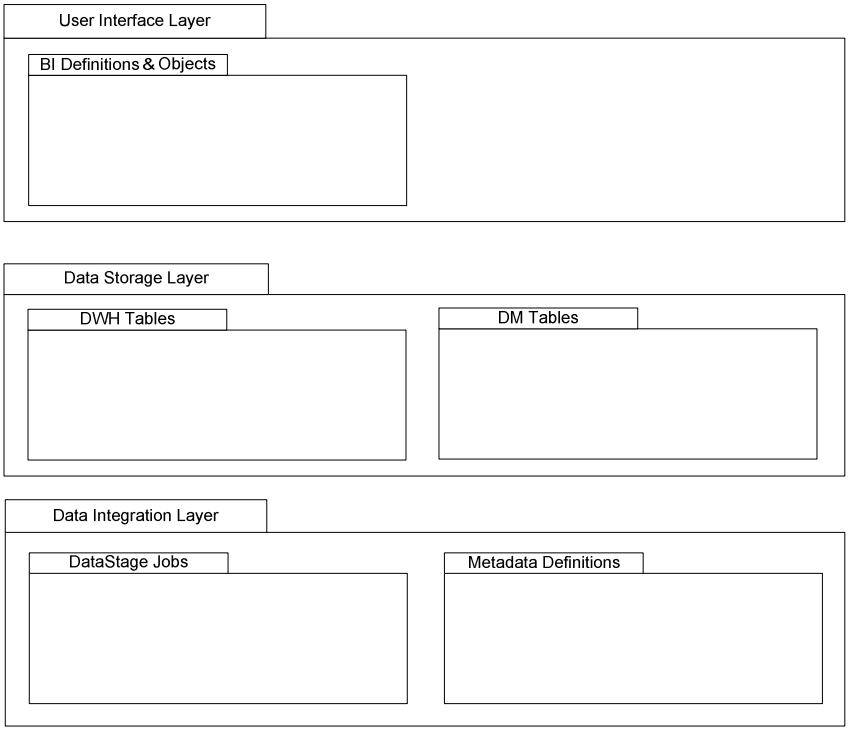
6.2 Couches – Diagramme UML des Composants



L'application DWH2 peut-être divisée en trois couches principales :

- **La couche "Intégration de données"**
  - Fonctions de traitement de données (ETL)
  - Métadonnées associées à ces traitements
- **La couche "Stockage des données"**
  - Data warehouse
  - Data marts
- **La couche "Interface utilisateur"**
  - Services de rapportage
  - Service d'analyse de données
  - Autres services d'exploitation des données

6.3 Structuration en packages



Data Integration Layer

Contient les objets qui permettent l'intégration des données dans le data warehouse et les data marts.

DataStage Jobs

Contient tous les processus DataStage de traitement des données

Metadata Definitions

Contient les définitions techniques et métier des données du data warehouse et des data marts

Data Storage Layer

Contient les objets permettant de stocker les données.

DWH Tables

Contient les tables du data warehouse

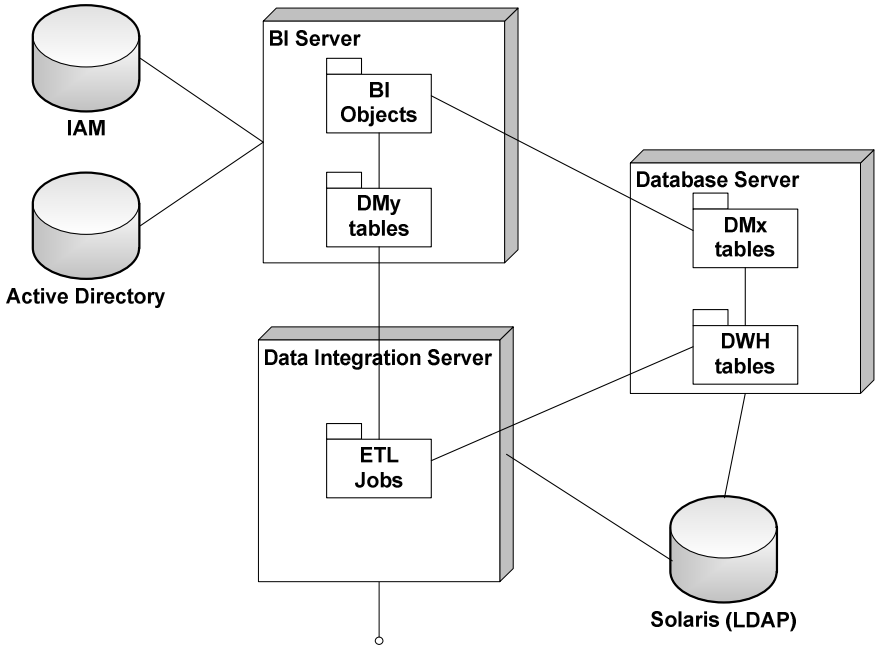
DM Tables

Contient les tables des data marts

User interface Layer

Contient les objets permettant d'exploiter les données

6.4 Packaging / Structuration en unité de déploiement



Le packaging sera constitué d'éléments différents à déployer sur chaque nœud du système.

Base de données data warehouse

Le paquet à déployer consiste en un fichier texte contenant la description des tables et des autres objets de la base de données (fichier DDL).

Base de données data mart

Le paquet à déployer consiste également en un fichier texte contenant la description des tables et des autres objets de la base de données (fichier DDL).

Les data marts doivent normalement être stocké au niveau du serveur de base données du data warehouse. Cependant, ils peuvent être éventuellement stockés au niveau du serveur BI pour garantir une intégration optimale avec les services de la plateforme BI. Dans ce cas, il faut garantir qu'aucune fonction avancée de sécurité n'est nécessaire au niveau du serveur BI et que l'administration des données reste minimum.

Dans le cas des cubes, une partie des données est toujours stockée au niveau du serveur BI (structure et données agrégées ou pré-calculées)

Le choix de la position de chaque data mart sera précisé et justifié dans les documents relatifs au modèle de données de chaque release (voir paragraphe 1.2 pour les références).

Serveur d'intégration de données

Le paquet à déployer est un paquet standard de l'outil DataStage contenant la définition des tables du data warehouse et des data marts ainsi que la description des processus traitant les données. Ces paquets sont produits et consommés via les fonctions d'export/import de l'outil DataStage Designer.

Serveur BI

Le paquet à déployer est un paquet standard de l'outil SQL Server contenant la définition des tables des data marts ainsi que et la description des objets BI (rapports, cubes…). Ces paquets sont produits et consommés via les fonctions d'export/import de l'outil.

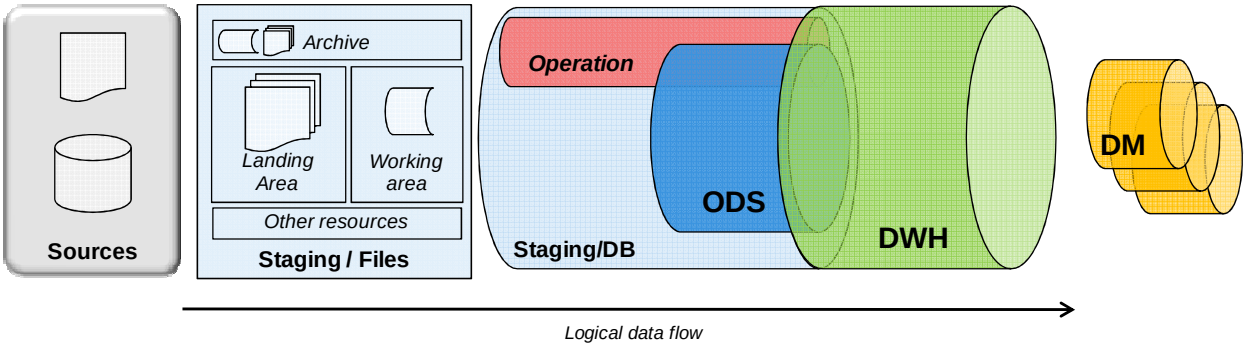
6.5 Réalisations de Cas d'Utilisation structurants - Diagrammes UML des Classes et d’Interaction (obligatoire)

Pas applicable pour ce projet.

7. Vue Données

7.1 Architecture logique des données

Le schéma suivant montre l'organisation des données.



Quatre zones logiques de données sont définies:

Sources de données

Les sources de données sont toutes les données externes au projet DWH2. Cela comprend toutes les sources de données internes et externes du SPF Finances, y compris la "zone FTP".

Données intermédiaires (Staging/Files et Staging/DB)

Les données intermédiaires sont les données et métadonnées techniques (et temporaires) nécessaires aux traitements des données.

Cette zone est composée d'un système de fichiers et d'une base de données et comprend différentes parties:

- La zone d'atterrissage (***landing area)*** est le tampon d'entrée des données sous forme de fichier à traiter.
- La zone des fichiers de travail (***working area***) stocke les fichiers temporaires nécessaires aux traitements.
- La zone d'***archivage*** stocke des données qui peuvent être nécessaires pour des besoins de maintenance et réparation.
- La zone intermédiaire base de données (***staging database area)*** stocke des données temporaires pour les traitements intra-base de données et des données techniques persistantes.
- L'ODS (***Operational Data Store)*** stocke les données sources en les historisant sous un format qui simplifie les traitements ultérieurs, notamment en cas de problème.
- La zone des métadonnées opérationnelles (***operational metadata area)*** est utilisée pour stocker des métadonnées au sujet du traitement des données qui ne sont pas proposées en standard par la plateforme d'intégration de données (IBM Information Server)

Data warehouse

La zone DWH est la zone de stockage principale. Les données consolidées y sont stockées.

Data marts

Les data marts sont dédiés au stockage de données préparées pour des usages spécifiques tels que rapportage et analyses OLAP.

7.2 Organisation technique des données

7.2.1 Systèmes sources

Les descriptions précises de chaque système source seront détaillées dans des documents spécifiques, disponible sous StarTeam. Voir paragraphe 1.2 pour la référence vers cette documentation.

Les paragraphes suivant donnent cependant quelques recommandations pour garantir une intégration fiable des données.

7.2.1.1 Formats préférés

Une grande variété de formats de données sources est techniquement possible. Certains formats donnent cependant de meilleurs résultats en termes de fiabilité et de performance.

A titre d'information, le tableau suivant donne la hiérarchie des formats dans l'ordre de préférence. L'objectif de ce tableau n'est pas d'imposer des modifications dans un système source pour obtenir le format préféré mais plutôt de choisir le meilleur format disponible sans modifications lourdes.

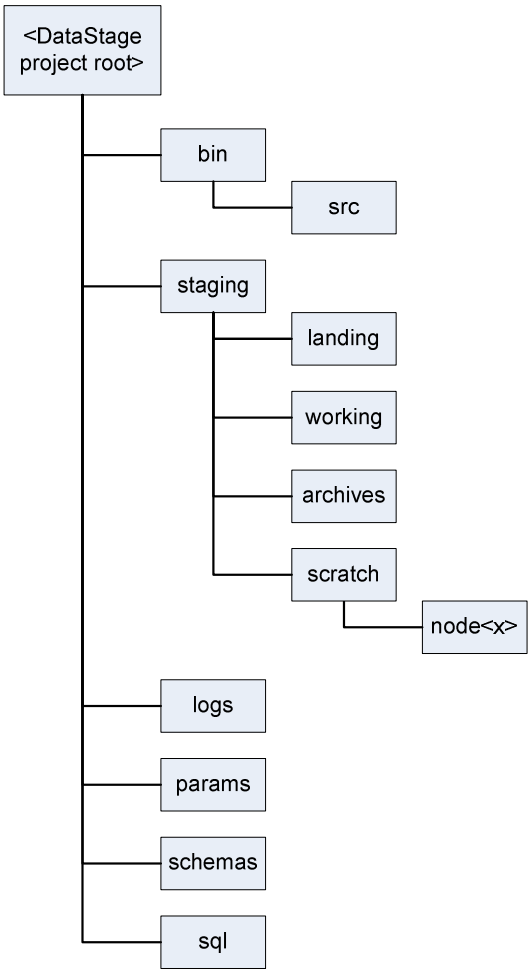
Préférence	Type de source	Format/Version	Commentaire
1	Flux MQ	Flux CDC (Change Data Capture)	Recommandé pour intégration en temps réel.
2	Bases de données	Vues ou tables DB2 9.5	Les vues sont recommandées (si possible)
3		Vues ou tables DB2 9.7	
4		Vues ou tables sur autres bases de données accessibles par DataStage	
5	Fichiers plats	Champs fixes (fixed width)	
6		Champs délimités	Risque de retrouver le délimiteur dans le contenu des champs
7		Autres formats complexes	

Les fichiers binaires tels que les fichiers Microsoft Excel ne sont pas supportés.

Dans tous les cas, le format d'interface doit faire l'objet d'un contrat entre la source et l'ETL, ceci afin de garantir la stabilité des services d'intégration de données.

7.2.2 Système de fichiers

Le schéma suivant montre l'organisation du système de fichiers au sein d'un projet DataStage.



- *<DS project root>* est la racine du projet DataStage, par exemple "**/dsdata/dwmdsadd/DWT_Dev"** dans l'environnement de développement.
- *<x>* est le nom du nœud de traitement parallèle dans DataStage. Par exemple 1, 2, 3…

Les zones logiques de données type "fichiers" sont dès lors réparties comme suit:

Zone logique	Description logique	Chemin physique
Staging	Landing area	*<DS Project root>*/staging/landing
	Working area (*DataStage resources disks*) (*DataStage scratch disks*)	*<DS Project root>*/staging/working *<DS Project root>*/staging/scratch
	Archive	*<DS Project root>*/staging/archives

Le tableau suivant montre la répartition des autres ressources, propre au traitement de données:

Description logique	Chemin physique
DataStage Schema files	*<DS Project root>*/schemas
DataStage parameters and configuration	*<DS Project root>*/params
Sql scripts	*<DS Project root>*/sql

Log files	*<DS Project root>*/logs
Tools binaries	*<DS Project root>*/bin
Tools sources	*<DS Project root>*/bin/src

7.2.3 Bases de données

7.2.3.1 Données intermédiaires et data warehouse

Les données intermédiaires ainsi que les données consolidées du data warehouse sont stockées dans une base de données IBM DB2. Cette base de données est nommée DWT_<x>, où <*x*> donne le type d'environnement. Les valeurs possibles sont:

- “D” pour environnement de développement
- "A” pour environnement d'intégration et d’acceptation
- “P” pour environnement de production

Les zones logiques de données sont réparties dans des schémas de la base de données, comme indiqué dans le tableau suivant.

Zone logique	Description logique	Schéma
Staging	DB Working area	STGWRK
	Operational Data Store	ODS
Data Warehouse	Données consolidées	DWH

7.2.3.2 Data marts

Les data marts ont pour objectif de fournir des données sous des formats adaptés à l'usage final qui en sera fait par les utilisateurs. Une partie des problématiques de structure de données, de performances et de sécurité peuvent être résolues à ce niveau, par exemple en limitant l’historique, en agrégeant et/ou en dénormalisant les données (schéma en étoile).

L'organisation précise de chaque data mart sera spécifiée dans des documents spécifiques, disponible sous StarTeam. Voir paragraphe 1.2 pour la référence vers cette documentation.

7.2.3.3 Metadonnées opérationnelles

Les métadonnées opérationnelles sont stockées sur deux niveaux.

Les **métadonnées opérationnelles "natives**" sont gérées en standard par la plateforme IBM Information Server et exploitée via les outils DataStage Director et Metadata Workbench. Elles donnent des renseignements précis au niveau des traitements de données (Jobs et séquences DataStage). Ces métadonnées et les outils ne sont donc pas détaillés dans ce document.

Les **métadonnées additionnelles** renseignent sur l'état opérationnel des traitements entre les sources, l'ODS, le data warehouse et les data marts au niveau des flux de données. Les informations sur les données de l'ODS pour lesquels des problèmes ont été détectés sont également stockées dans cette zone.

Les détails de stockage sont renseignés dans le tableau suivant:

Zone logique	Description logique	Schéma
Operation	Métadonnées	OPMETA

	opérationnelles	

7.3 Diagramme E/R des Données Persistantes

Les diagrammes E/R des données sont détaillés dans des documents séparés, disponible sous StarTeam. Voir paragraphe 1.2 pour la référence vers cette documentation.

Les paragraphes suivants donnent les principes directeurs qui sont appliqués pour l'élaboration des diagrammes E/R.

7.3.1 ODS

7.3.1.1 Règles de modélisation

Le modèle de données de l'ODS doit être pratiquement identiques aux modèles de données source. Les différences résident dans l'historisation des modifications des données des systèmes sources. Chaque enregistrement unique dans un système source peut donc être stocké plusieurs fois dans l'ODS, en fonction du nombre de modifications captées ou détectées par le système d'intégration de données.

La clef unique originale doit donc être étendue à un champ donnant la date de modification associée. Un champ supplémentaire doit être prévu pour renseigner le type de modification.

L'exemple suivant illustre le cycle de vie complet d'un enregistrement unique dans un système source.

Description de la modification	Code du type de modification	Date
Insertion initiale	I	*T*
Première modification	U	*t+1*
Deuxième modification	U	*t+2*
Rafraichissement forcé des données[1]	R	*t+3*
N*ième* modification	U	*t+n*
Suppression	D	*t+n+1*

[1] *: Le rafraîchissement de données peut-être utilisé en cas de problèmes de synchronisation avec la source.*

Pour des questions d'efficacité de stockage et de performances, le modèle physique des données ne doit contenir aucune contraintes (tel que des "NOT NULL"), à l'exception de celles concernant les clés primaires. L'objectif est d'être capable de stocker une image aussi fidèle que possible des sources de données, y compris de possibles violations de contraintes.

7.3.1.2 Champs techniques

Chaque table de l'ODS contiendra en plus des champs de la table source les champs techniques suivants:

- **O_I_IDF** (Bigint): référence interne de l'enregistrement dans l'ODS.
- **S_I_ODS_INS** (Timestamp): date et heure de la création de l'enregistrement dans l'ODS.
- **C_I_MOD_TYPE** (char(1)) : code du type de modification. I = Insert, U = Update, D = Delete, R = Refresh (rafraichissement forcé de toutes les données).
- **O_INS_FLOW_IDF** (Bigint): Référence vers le flux qui a chargé l'enregistrement.

7.3.2 Data warehouse

7.3.2.1 Règles de modélisation

Le modèle de données du data warehouse doit être relationnel et normalisé au minimum jusqu'à la troisième forme normale.

Intégrité référentielle

L'intégrité référentielle entre les entités de données doit être assurée par des clés techniques (surrogate keys) et pas par des clés métiers (natural keys). Cette règle est généralement considérée comme une règle de bonne pratique pour les raisons suivantes:

- Réduction du volume des données clés (primaires et étrangères) et des indexes.
- Optimisation des performances des jointures entre tables
- Indépendance et stabilité accrue vis-à-vis des clés métiers (pas de risques de collisions et de dénormalisation accidentel, faible impact en cas de changement des clés métiers…)

Les tables décrivant des codes peuvent se contenter des clés naturelles et doivent donc pas contenir de clés technique.

Signalétique

DWH 2 doit être temporairement indépendant des signalétiques externes, tel que DWH 1. Des problèmes de chargement des données de ses signalétiques ne doivent donc pas impacter le fonctionnement de DWH2.

Pour répondre à ce besoin, le modèle de données du data warehouse contient une signalétique simplifiée (« mini signalétique »). Celle-ci stocke uniquement les données signalétiques nécessaires au fonctionnement du data warehouse 2.

Sécurité

La base de données DWH2 doit également être capable d'interdire l'identification de personnes pour des raisons de confidentialité. Ce besoin est couvert uniquement au moyen des fonctions de sécurité avancées de la base de données DB2 (fonctions LBAC). Les données permettant d'identifier une personne ou une entreprise sont donc impossibles à utiliser par les utilisateurs non autorisés. Exemple: numéro national, numéro d'entreprise…
Les utilisateurs avancés ayant accès au data warehouse peuvent utiliser les "surrogate keys" (clés techniques statiques) comme identifiants uniques.

7.3.2.2 Champs techniques

Chaque table du data warehouse contiendra au minimum les champs techniques suivants:

- **O_INS_FLOW_IDF** (Bigint): Référence vers le flux qui a créé l'enregistrement.
- **S_I_INS** (Timestamp): Date et heure de la création initiale de l'enregistrement
- **O_UPD_FLOW_IDF** (Bigint): Référence vers le dernier flux qui a modifié l'enregistrement.
- **S_I_UPD** (Timestamp): Date et heure de la dernière modification de l'enregistrement

7.3.3 Data marts

7.3.3.1 Règles de modélisation

Les règles suivantes devront guider l'élaboration des data marts.

- Les données des data marts doivent normalement être stockés dans des bases de données DB2. Cependant, ceux destinés à être utilisé par les services BI Microsoft peuvent éventuellement être stockés (partiellement pour les cubes) au niveau du serveur Microsoft SQL Server si cela permet de garantir une intégration optimale des fonctionnalités. Dans ce cas, il faut également garantir que la base de données SQL Server ne nécessitent qu'un minimum d'administration, y compris du point de vue de la sécurité. Les fonctionnalités avancées de sécurité ne peuvent être assurées que par DB2/LBAC.
- Les données des data marts sont toujours dérivées des données de référence du data warehouse. Aucune nouvelle données n'est introduite directement dans un data mart. De cette façon, un data mart peut toujours être reconstruit à partir du data warehouse en cas de désastre.
- Les data marts destinés au rapportage sont normalement modéliser en étoile pour des raisons de performances analytiques (tables de faits et dimensions). Les modèles relationnels normalisés et en "flocons de neige" (snowflake) sont déconseillés.

7.3.3.2 Champs techniques

Chaque table des data marts devrait contenir au minimum les champs techniques suivants:

- **O_INS_FLOW_IDF** (Bigint): Référence vers le flux qui a créé l'enregistrement.
- **S_I_INS** (Timestamp): Date et heure de la création initiale de l'enregistrement
- **O_UPD_FLOW_IDF** (Bigint): Référence vers le dernier flux qui a modifié l'enregistrement.
- **S_I_UPD** (Timestamp): Date et heure de la dernière modification de l'enregistrement

7.3.4 Métadonnées opérationnelles

7.3.4.1 Statuts opérationnels des flux

Les statuts opérationnels sont stockés dans la table **OPMETA.FLOW_STATUS**. Cette table contient également les statistiques de chaque flux de données (nombre de données extraites, rejetées, transformées, etc…)

L'information sur les données de l'ODS qui contiennent des erreurs est stockée dans la table **OPMETA.ODS_DATA_ERROR**. Cette table référence des enregistrements dans l'ODS, le flux de données qui à rejeté l'enregistrement et la raison du rejet. Les raisons des rejets sont surtout des raisons métiers, telle que décrite dans la documentation de chaque flux de données.

La description détaillée de ces tables est donnée dans un document séparé, disponible sous StarTeam. Voir paragraphe 1.2 pour la référence vers cette documentation.

7.4 Backups

L'ensemble du système doit pouvoir être restauré après un désastre. Les paragraphes suivant décrivent les éléments qui doivent être sauvegardés.

7.4.1 Service d'intégration de données

La plateforme IBM InfoSphere Information Server doit être sauvegardée selon les règles d'administration décrite dans la documentation du logiciel ("IBM Information Server 8.1 : Administration Guide").

Une période de rétention des données de 15 jours minimum est recommandée pour pouvoir couvrir les plus longues périodes sans support opérationnel (long week-ends, ponts, etc…)

7.4.2 Service BI

La plateforme Microsoft SQL Server doit être sauvegardée selon les règles d'administration décrite dans la documentation du logiciel.

Une période de rétention des données de 15 jours minimum est recommandée pour pouvoir couvrir les plus longues périodes sans support opérationnel (long week-ends, ponts, etc…)

7.4.3 Système de fichiers

Les répertoires suivants doivent être sauvegardés :

Répertoire	Description	Fréquence minimum recommandée	Période de rétention minimum recommandée
<*DS Project root*>/staging/archives	Archive	Tous les jours	60 jours
<*DS Project root*>/schemas	DataStage Schema files	A chaque release	1 release
<*DS Project root*>/params	DataStage parameters and configuration	A chaque release	1 release
<*DS Project root*>/sql	Sql scripts	A chaque release	1 release
<*DS Project root*>/logs	Log files	A chaque release	1 release
<*DS Project root*>/bin	Tools binaries	A chaque release	1 release

7.4.4 Bases de données

Les schémas suivants doivent être sauvegardés :

Schéma	Description	Fréquence minimum recommandée	Période de rétention minimum recommandée
OPMETA	Métadonnées opérationnelles	Tous les jours	15 jours
STGWRK	Working area (db)	Tous les jours	15 jours
ODS	Operational Data Store	Tous les jours	15 jours
DWH	Data Warehouse	Tous les jours	15 jours
Data marts	Data Marts	A chaque release	1 release

Les data marts peuvent être reconstruit à partir des données du data warehouse à n'importe quel moment. Seule leur structure doit donc être conservée.

Le data warehouse pourrait éventuellement être reconstruit à partir des données de l'ODS. Ceci pourrait cependant demander un très long temps de traitement et rendre la plateforme indisponible un long moment. Il est donc recommandé de sauvegarder les données du data warehouse au moins tous les jours.

8. Vue Processus

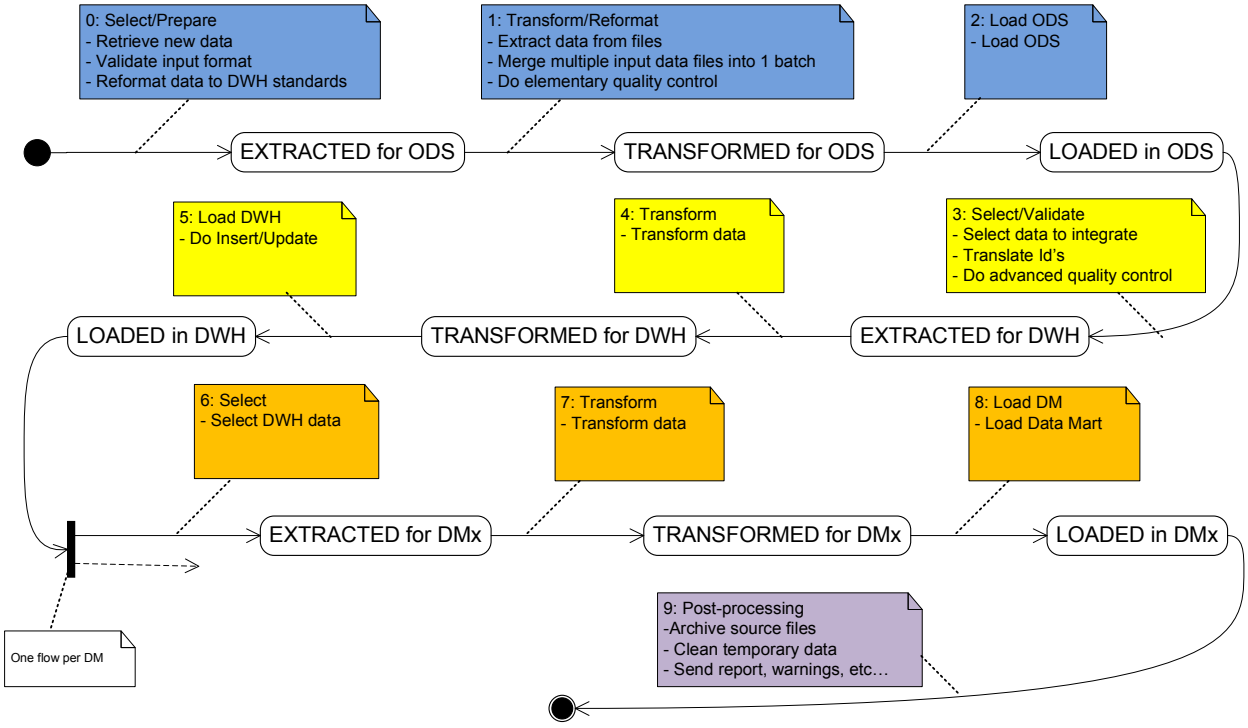
8.1 Flux standard

Cette section décrit le traitement standards des données pour leur intégration dans le data warehouse et les data marts. L'objectif est de favoriser la cohérence des traitements d'intégration de données afin de faciliter la maintenance.

Ce standard n'est pas une règle absolue. Certaines données peuvent nécessiter des traitements différents en contradiction avec ce standard. Dans ce cas, il faut surtout privilégier la clarté et la facilité de maintenance des traitements.

Le diagramme états-transitions suivant décrit à haut niveau le flux standard de chargement du data warehouse et des data marts.

Les états (EXTRACTED-TRANSFORMED-LOADED) décrits sont des états logiques uniquement. Si les traitements sont peu complexes, plusieurs transitions entre ces états peuvent être regroupées dans un seul job DataStage afin de garantir des performances élevées.



Les trois étapes majeures sont l'ODS, le data warehouse et les data marts. Ces étapes sont des "points d'arrêts", c.à.d. que si certains traitements échouent ou que des données sont corrompues, il est toujours possible de relancer les traitements à partir de l'étape précédente.

Bien que les traitements entre ces étapes soient assez différents, le flux suit toujours la logique Extraction – Transformation - Chargement.

8.1.1 Etape 0: Préparation des données - Sélection

8.1.1.1 Description

Cette étape à pour objectif d'isoler des traitements ultérieurs la complexité des sources de données en termes de codage et de format.

8.1.1.2 Entrées

Fichiers sources ou base de données

8.1.1.3 Traitements logiques

1. Initialiser le flux dans les métadonnées opérationnelles avec le statut "**STARTED**"
2. *Pour sources de type "fichier":* Détecter les nouveaux fichiers dans la zone FTP (ou d'autres systèmes de fichiers)
   Pour sources de type "base de données": Extraire les (nouvelles) données
3. Rejeter dans des fichiers les données techniquement mal formatées et/ou illisibles (par exemple : format de date invalide dans un fichier texte, longueur excessive d'un champ texte,…)
4. Si nécessaire, calculer les différences (delta) par rapport à la dernière situation connue
5. Si fichier complexe, reformater les données au format d'entrée standard du data warehouse et écrire ces données dans la "landing area".
6. Si possible, enregistrer les informations des nouvelles données dans les métadonnées opérationnelles avec le statut "**EXTRACTED**"

8.1.1.4 Sorties

Si la source est une base de données ou un fichier simple: Flux de données DataStage (flux interne ou datasets)

Si la source est un fichier complexe: Fichiers d'entrée standards pour DWH.

8.1.1.5 Implémentation

N'importe quelle technique de programmation capable de préparer les données d'une façon simple, efficace et claire. Les outils suivants sont toutefois recommandés dans leur ordre de préférence:

1. DataStage parallel job
2. DataStage server job
3. Scripts Perl (Performances élevées et possibilités natives de traitement de données très étendues, syntaxe simple)
4. Scripts Unix Shell (sh, Ksh, Csh, Awk…) (Performances faible, possibilités étendues de traitement de données, syntaxe très simple)
5. Autres outils

8.1.2 Etape 1: Formatage (Transformation)

8.1.2.1 Description

Cette étape à pour but de préparer les données extraites pour le chargement dans l'ODS. C'est une étape qui doit être systématique et indépendante des traitements ultérieurs afin de limiter les adaptations quand les besoins évoluent.

8.1.2.2 Entrées

Flux de données DataStage (flux interne ou datasets)

Fichiers standards de la zone d'atterrissage.

8.1.2.3 Traitements logiques

1. Si fichiers préparés dans la "landing area", lire les fichiers et, si nécessaire, les fusionner en un seul lot à traiter
2. Ajouter les champs techniques obligatoires (date de chargements, id du flux de chargement…)
3. Si défini, ajouter les drapeaux de contrôle de qualité élémentaires, au niveau de chaque enregistrements. Exemple: données manquantes, hors bornes, etc…
4. Mettre à jour les métadonnées opérationnelles avec le statut "**TRANSFORMED**"

8.1.2.4 Sorties

Flux de données DataStage (flux interne ou datasets).

8.1.2.5 Implémentation

Uniquement des processus DataStage type parallèle.

8.1.3 Etape 2: Chargement de l'ODS

8.1.3.1 Description

Cette étape a pour but de stocker les nouvelles données dans l'ODS.

8.1.3.2 Entrées

Flux de données DataStage (flux interne ou datasets).

8.1.3.3 Traitements logiques

1. Lire les données préparées
2. Charger l'ODS
3. Mettre à jour les métadonnées opérationnelles avec le statut "**LOADED**"

8.1.3.4 Sorties

Tables de l'ODS

8.1.3.5 Implémentation

Uniquement des processus DataStage type parallèle.

8.1.4 Etape 3: Sélection et validation des données DWH

8.1.4.1 Description

Cette étape sélectionne et valide les données à intégrer au data warehouse.

8.1.4.2 Entrées

ODS

8.1.4.3 Traitements logiques

1. Initialiser le flux dans les métadonnées opérationnelles avec le statut "**STARTED**"
2. Sélectionner les données de l'ODS à intégrer dans le data warehouse (Nouvelles données et données précédemment déclarées en erreur, à recycler).
3. Traduire les clés métier en identifiants techniques
4. Si défini, faire les contrôles de qualité avancés (intégrité référentielle, validation des règles métier, etc…). Insérer les raisons du rejet et la références aux données erronées de l'ODS dans la table opérationnelle ODS_DATA_ERROR.
5. Si nécessaire, compléter les données de la mini signalétique pour assurer l'intégrité référentielle du data warehouse
6. Enregistrer les informations des nouvelles données dans les métadonnées opérationnelles avec le statut "**EXTRACTED**".

8.1.4.4 Sorties

Flux de données DataStage (flux interne ou datasets).

8.1.4.5 Implémentation

Uniquement des processus DataStage type parallèle.

8.1.5 Etape 4: Data transformation

8.1.5.1 Description

Cette étape transforme les données sélectionnées et validées pour les adapter à la structure du data warehouse. L'essentiel de la complexité des traitements est concentrée dans cette étape.

8.1.5.2 Entrées

Flux de données DataStage (flux interne ou datasets)

8.1.5.3 Traitements logiques

1. Transformer les données
2. Mettre à jour les métadonnées opérationnelles avec le statut "**TRANSFORMED**"

8.1.5.4 Sorties

Flux de données DataStage (flux interne ou datasets).

8.1.5.5 Implémentation

Uniquement des processus DataStage type parallèle.

8.1.6 Etape 5: Chargement du DWH

8.1.6.1 Description

Cette étape charge les données transformées dans le data warehouse.

8.1.6.2 Entrées

Flux de données DataStage (flux interne ou datasets).

8.1.6.3 Traitements logiques

1. Charger les données DWH
2. Mettre à jour les métadonnées opérationnelles avec le statut "**LOADED**"

8.1.6.4 Sorties

Tables data warehouse

8.1.6.5 Implémentation

Uniquement des processus DataStage type parallèle.

8.1.7 Etapes 6,7 et 8: Construction des Data Marts

8.1.7.1 Description

Ces étapes transforment les données du data warehouse pour les adapter aux formats des data marts. Ces traitements peuvent être très spécifique pour chaque data mart et doivent donc être documentés séparément.

8.1.7.2 Entrées

Tables data warehouse

8.1.7.3 Traitements logiques

1. Initialiser le flux dans les métadonnées opérationnelles avec le statut "**STARTED**"
2. Sélectionner les données du DWH à intégrer au data mart
3. Enregistrer les informations des nouvelles données dans les métadonnées opérationnelles avec le statut "**EXTRACTED**".
4. Transformer les données
5. Mettre à jour les métadonnées opérationnelles avec le statut "**TRANSFORMED**"
6. Charger les données dans le data mart
7. Mettre à jour les métadonnées opérationnelles avec le statut "**LOADED**"

8.1.7.4 Sorties

Tables data mart

8.1.7.5 Implémentation

Processus DataStage type parallèle.

8.1.8 Etape 9: Post-processing

8.1.8.1 Description

Cette étape regroupe tous les traitements qui ne sont pas directement nécessaires à l'intégration des données mais qui permettent de maintenir la plateforme dans un état sain.

8.1.8.2 Entrées

Paramètres opérationnels du système (historique des logs, etc…)

8.1.8.3 Traitements logiques

- Archiver les données sources
- Supprimer les données temporaires
- Gérer l'historique des données dans l'ODS, le data warehouse et les data marts
- Gérer l'historique opérationnel (logs, statuts, …)
- …

8.1.8.4 Sorties

Log des activités

8.1.8.5 Implémentation

N'importe quelle technique de programmation capable de remplir les fonctions nécessaires d'une façon simple, efficace et claire. Il peut s'agir de job DataStage, de script UNIX Shell ou d'autres outils.

8.2 Stratégie d'exécution des traitements

Les processus de traitement de données sont exécutés en série par des séquences DataStage afin de limiter les interdépendances et simplifier la maintenance opérationnelle en cas de problèmes. Ces séquences sont configurées pour pouvoir redémarrer au niveau du dernier traitement sans erreurs.

Les traitements se font en trois étapes « points d'arrêt » obligatoires : l'ODS, le Data Warehouse et les data marts. Une étape ne démarre pas tant que la précédente étape n'est pas entièrement terminée avec succès.

Le démarrage automatique des séquences est assuré par l'outil standard du SPF Finance : Absyss Visual TOM. Celui-ci est configuré selon les calendriers de démarrage nécessaires aux traitements (quotidien, hebdomadaire, mensuel…) et ne contient aucune logique liées aux données (celle-ci est gérée par les séquences DataStage).

Pour limiter l'impact d'éventuels problèmes d'exécution d'un « job » sur d'autres flux de données, il est possible de séparer les traitements en domaines indépendants (données et « jobs »). De cette façon, par exemple, une erreur au niveau de l'intégration de données dans l'ODS d'un domaine n'empêchera pas le data mart d'un autre domaine d'être chargé correctement.

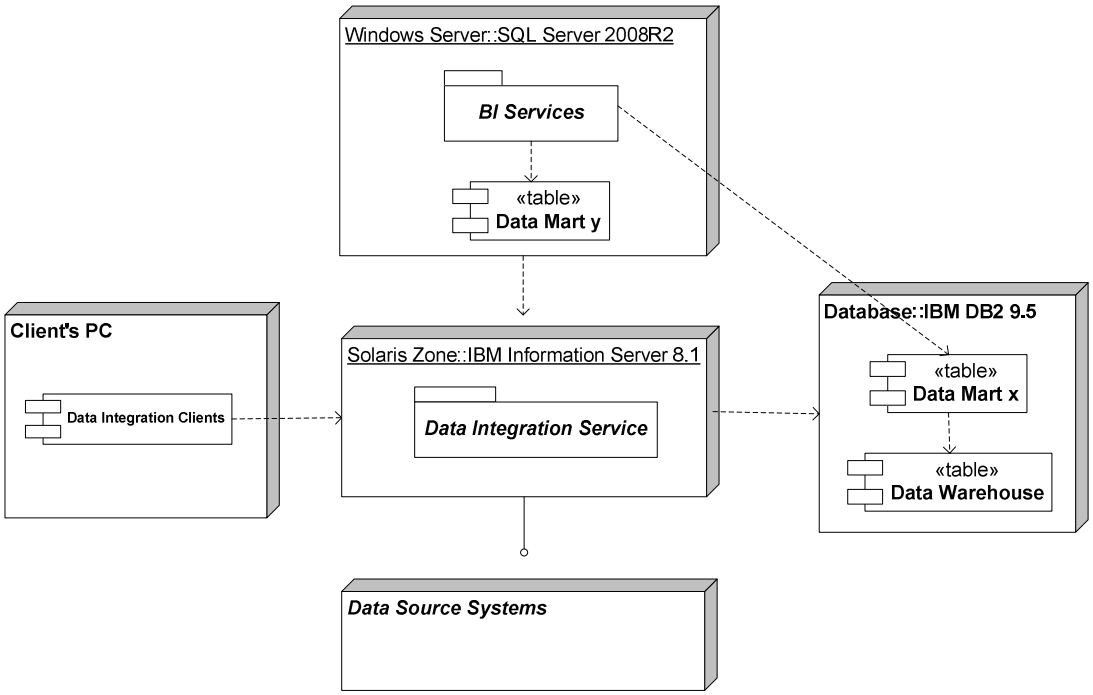
8.3 Diagramme UML de séquence

Pas applicable pour ce projet.

9. Vue Déploiement

9.1 Diagramme de déploiement

Les différents composants de la plateforme peuvent être représentés de la façon suivante.



Les composants du système sont déployés sur différents serveurs.

Serveur d'intégration de données

Il s'agit d'une zone Solaris qui héberge les services d'intégration de données IBM Information server v8.1/DataStage.

Base de données Data Warehouse

Les données du data warehouse sont stockées dans une base de données IBM DB2 v9.5. Le serveur doit être considéré comme externe au serveur d'intégration de données, bien qu'il puisse techniquement être installé sur le même serveur physique.

Certains data marts peuvent être stockés sur ce serveur pour des raisons de performances.

Serveur BI

Les données des data marts peuvent être stockées sur le serveur offrant les services BI afin de garantir une intégration optimale de toutes les fonctionnalités BI.

Ces services sont fournis par SQL Server 2008R2

Logiciels clients

Les développeurs et administrateurs des processus d'intégration de données utilisent des logiciels clients qui communiquent avec le serveur d'intégration de données.

10. Taille et performances – FeedBack et Suggestions

Cette section sera complétée au fur et à mesure en fonction du feedback opérationnel obtenu lors des tests.

10.1 Flux d'intégration de données

10.1.1 Performance

La performance des flux d'intégration de données sera estimée en mesurant les temps de traitement et de chargement d'un volume défini de données sources. Ce volume sera choisi pour être représentatif du volume de données qui sera traité habituellement.

Ces performances pourront être mesurées objectivement via la table de suivi des flux de données dans les métadonnées opérationnelles.

Ces performances pourront être adaptées aux besoins notamment en ajustant le degré de parallélisme du moteur de traitement DataStage.

10.2 Bases de données

10.2.1 Performance

Les performances des bases de données seront estimées en mesurant les temps pour exécuter des requêtes prédéfinies. Ces requêtes seront représentatives d'un usage typique du data warehouse et de chaque data marts.

Ces performances pourront être mesurées objectivement via les outils clients de DB2 ou via des scripts UNIX utilisant la commande "time"

Ces performances pourront être adaptées aux besoins notamment en ajoutant des indexes ou en optimisant la structure des données et leur mode de stockage.

10.3 Services BI

10.3.1 Performance

Les performances des services BI seront estimées en mesurant les temps pour réaliser des scénarios BI prédéfinis. Ces scénarios seront représentatifs d'un usage typique, par exemple: Affichage d'un rapport.

Ces performances pourront être mesurées au chronomètre.

Ces performances pourront être adaptées aux besoins notamment en optimisant les paramètres du serveur BI, en dé-normalisant ou en optimisant le modèle de données des data marts.

11. Qualité

11.1 Extensibilité / Maintenance

11.1.1 Description

L'extensibilité/Maintenance est la capacité de mettre à jour l'application après la fin de la phase de développement (pour l'adapter aux besoins business changeants ou pour corriger certains bugs).

11.1.2 Solution

L'extensibilité/la maintenance est assurée par divers moyens :

- Un design technique séparant clairement les différentes fonctions de la solution
- La réutilisation systématique de procédures éprouvées pour des traitements similaires
- Une documentation complète permettant aux personnes non initiées de comprendre l'implémentation technique de la solution.
- L'utilisation d'outils avec interfaces graphiques permettant une vue claire et simple du fonctionnement intime de la solution
- L'utilisation d'un moteur d'intégration de données (ETL) parallèle permettant l'extensibilité des ressources de façon transparente vis-à-vis de la logique des traitements

11.2 Portabilité

11.2.1 Description

La portabilité est la capacité d'utiliser l'application dans un autre environnement.

11.2.2 Solution

Tous les composants de la solution peuvent être migrés sur des plateformes similaires du point de vue matériel et système d'exploitation avec un minimum d'adaptations.

Pour des migrations sur des matériels et systèmes d'exploitation différents, chaque composant est différent.

Base de donnée du data warehouse

IBM DB2 9.5 est disponible pour une large gamme de matériel (x86-64, POWER, System Z, SPARC…) et système d'exploitation (Windows, Linux, AIX, Solaris…). Se reporter à la documentation technique du produit pour une liste exhaustive.

Services d'intégration de données

IBM InfoSphere Information Server v8.1 est disponible pour une large gamme de matériel (x86, POWER, SPARC…) et systèmes d'exploitations (Windows, Linux, AIX, Solaris…). Se reporter à la documentation technique du produit pour une liste exhaustive.

IBM InfoSphere Information Server v8.1 supporte une large gamme de base de données ce qui permet des évolutions importantes de la base de données du data warehouse.

Services BI et bases de données des data marts

Microsoft SQL Server 2008R2 est uniquement disponible pour les plateformes avec un système d'exploitation de type Windows. Différentes versions et éditions de Windows peuvent être utilisées. Se

reporter à la documentation technique du produit pour une liste exhaustive.

Les services BI de SQL Server 2008R2 peuvent utiliser des données de différentes bases de données mais certaines limitations sont alors possibles.

Logiciels clients

Les logiciels clients ne supportent que des plateformes x86 avec un système d'exploitation Windows.

11.3 Monitoring

11.3.1 Description

Le monitoring est la capacité à contrôler les différents composants du système afin de s'assurer de leur bon fonctionnement, de détecter les erreurs et de produire des statistiques.

11.3.2 Solution

Le flux des données peut être contrôlé par les outils dédiés de la suite IBM Information Server v8.1 (DataStage Director, Metadata Workbench, …)

Des informations sont également transmises à HP OpenView pour un monitoring de haut niveau. Les outils en ligne de commande d'OpenView seront appelés par DataStage pour communiquer.

11.4 Logging

11.4.1 Description

Le logging est la capacité à contrôler individuellement les composants de l'application à des fins de debugging

11.4.2 Solution

Le logging se fait à deux niveaux:

- Le logging des traitements individuels des données est assuré en standard par les outils qui fournissent les services (IBM Information Server, DB2 et SQL Server)
- Le logging des flux de données permet un suivi à haut niveau des traitements de données dans leur ensemble. Il doit permettre d'identifier rapidement l'étape qui pose problème dans un traitement. Des statistiques de chargement peuvent également être extraites de ces logs.

12. Mise en œuvre des standards du SPF Finances

12.1 Conventions de nommage

12.1.1 Fichiers

12.1.1.1.1 Fichiers de la zone d'atterrissage

Les fichiers de cette zone sont nommés comme suit

FWH_<source name>_[<schema>_<table>]|[<source file name>]_<timestamp>.TXT

Où

<source name> est le nom abrégé de la source de données

<schema> et **<table>** sont les noms du schéma et de la table source de donnée dans le cas de sources de type « bases de données ».

<source file name> est le nom abrégé du fichier source de données dans le cas de sources de type « fichier ».

<timestamp> est la date et l'heure à laquelle le fichier à été créé, le format doit être "%C%y%m%d%H%M%S" (notation UNIX/POSIX).

12.1.2 Noms des objets pour le stockage de données

Les conventions de nommage pour les modèles de données doivent suivre le plus possible les standards du SPF Finances/SupDev. Cependant, ces conventions de nommage sont essentiellement prévues pour la conception de logiciels orientés objets en utilisant la notation UML. Ces conventions ne sont pas tout à fait adaptées à la conception Entité-Relation nécessaire dans un projet data warehouse. Quelques modifications sont donc nécessaires.

Les paragraphes suivant décrivent les différences entre les conventions de nommages standards de SupDev et les conventions utilisées dans ce projet.

Voir la section 1.2 pour la référence au document "SUPDEV_NOMMAGE" au sujet des conventions de nommage de SupDev.

12.1.2.1 Modèle logique de données

Les modèles logiques de données peuvent être considérés comme équivalents aux schémas conceptuels dans une conception orientée objet.

La table suivante donne les équivalences entre les concepts orientés objets et les concepts entité-relation (ER)

Concept OO	Concept ER	Convention de nommage en ER
Classe	Entité	Noms commencent par E *(Entity)* au lieu de K *(Klasse)*. Exemple: *EPersonne, ESituationJuridique*
Attribut	Attribut	Abréviations peuvent être utilisées dans les champs *Mn...n* mais devraient être évitées pour plus de lisibilité
Association	Relationship	Notation "Crow's foot". Pas de rôles.

12.1.2.2 Modèle physique de données

Les modèles physiques de données doivent suivre les conventions de SupDev et RDC.

Pour les data marts, il est toutefois possible d'adapter les noms des objets aux contraintes techniques ou métiers des utilisateurs finaux.

12.1.3 Noms des objets pour le traitement des données

Les traitements de données se font avec IBM DataStage.

Les objets DataStage doivent être nommés selon les conventions de DCC. Voir le document DCC_DS_NAMING dans le paragraphe 1.2 pour la référence.

12.1.4 Outil de business intelligence

L'outil BI utilisé pour l'exploitation des données est Microsoft SQL Server (Reporting services/SSRS et Analysis services/SSAS).

En accord avec les pratiques standards de DCC, aucune norme stricte n'est nécessaire pour le nommage des objets.

Le objets BI étant destinés aux utilisateurs finaux, il est néanmoins recommandé d'utiliser des noms qui décrivent l'objet de façon complète et précise auprès des utilisateurs.