PROJET CRÉSUS

POC — Spécifications Techniques

BI Prédictive Finance

« Comment l'analyse prédictive et la simulation peuvent-ils être des facteurs de gains dans la gestion de trésorerie d'une banque ? »

Commanditaire
ZF Banque
Direction des Systèmes d'Information

Document
Spécifications Techniques AMOE
Proof of Concept

SOMMAIRE

INTRODUCTION

But du document

> *Cette fiche a pour but de présenter les spécifications techniques du Proof of Concept (POC) pour le projet « Crésus ». Elle détaille l'architecture, les technologies et la structure des données qui seront mises en œuvre pour démontrer la faisabilité et la valeur d'une solution de BI prédictive pour la gestion de trésorerie du groupe ZF Banque.*
>
> *Plus concrètement, le document rassemble les livrables clés du POC, à savoir :*

- Les fichiers sources utilisés
- L'architecture technique détaillée de la solution
- Le dictionnaire de données (KPI..)
- Les Modèles Conceptuel et Logique de Données (MCD / MDL)
- Le mapping de données.

SPÉCIFICATIONS TECHNIQUES

Solutions choisies

- **Base de données :** PostgreSQL
- **Orchestrateur de flux :** Apache Airflow
- **Modélisation et génération de données :** Scripts Python
- **Data Visualisation et Simulation :** Microsoft Power BI Desktop

Architecture technique

L'architecture proposée suit un flux en quatre étapes :

1. **Sources de données :** Génération de données synthétiques (flux de trésorerie) via un script Python et intégration de données de marché externes (ex: taux de change via fichiers CSV).
2. **Ingestion & stockage :** Centralisation des données brutes et structurées dans une base de données PostgreSQL.
3. **Modélisation & simulation :** Orchestration par Apache Airflow de pipelines (DAGs) pour extraire, transformer les données, entraîner les modèles prédictifs et stocker les résultats (prévisions, simulations) dans PostgreSQL.

4. **Restitution & utilisation :** Connexion de Power BI à PostgreSQL pour visualiser les données historiques, les prévisions et permettre aux utilisateurs de lancer des simulations interactives (scénarios « what-if »).

Fichiers sources

Les fichiers sources constituent le point d'entrée principal du projet. Ils rassemblent l'ensemble des données nécessaires à la mise en œuvre de la solution de BI prédictive appliquée à la gestion de la trésorerie.

Ces données proviennent à la fois de sources internes, correspondant aux opérations financières du groupe, et de sources externes, issues des marchés monétaires et financiers.

Deux types de données sont exploités :

- **Les flux de trésorerie** (interne), représentant les opérations financières réalisées par les filiales du groupe.
- **Les cours de change des devises** (externe), permettant de prendre en compte les variations de taux dans les analyses et les simulations de trésorerie.

L'ensemble des fichiers sources a été généré ou importé sous format CSV. Les jeux de données internes ont été produits à l'aide d'un script Python simulant des transactions financières réalistes, tandis que les données externes proviennent de fichiers de marché contenant l'historique des taux de change.

Ces fichiers constituent la première étape du flux de traitement des données. Ils sont ensuite intégrés dans la base de données PostgreSQL, structurés selon le modèle de données défini, documentés dans le dictionnaire de données, puis mappés vers les différentes tables cibles utilisées pour les analyses et visualisations. Cette chaîne assure la cohérence, la traçabilité et la fiabilité des informations exploitées dans l'outil Power BI.

Base de données

La base de données, gérée sous PostgreSQL, a été choisie pour sa robustesse, sa flexibilité et ses capacités avancées en SQL. Elle centralisera l'ensemble des données du projet, qu'elles soient internes (générées) ou externes, et contiendra aussi bien les données brutes que les données préparées pour l'analyse, ainsi que les résultats des prévisions et des simulations. Sa structure est pensée pour garantir la cohérence, la traçabilité et la performance des requêtes effectuées, notamment depuis Power BI.

Ce choix s'explique par le fait que PostgreSQL est reconnu comme un standard de l'industrie, compatible avec de nombreux outils, ce qui en facilite l'intégration. C'est une base performante, capable de gérer la montée en charge et de traiter de grosses volumétries. En tant que base relationnelle, PostgreSQL permet de diviser les données en plusieurs tables et de les lier via des requêtes plus ou moins complexes. Enfin, sa compatibilité avec l'écosystème Python et sa gestion avancée des types JSON offrent la possibilité de stocker et manipuler des données semi-structurées si besoin.

Structure de la base

1. Table de faits

FACT_FLUX_TRESORERIE — Contient l'ensemble des transactions financières.

Champs principaux :

- id_flux (clé primaire)
- montant
- date_operation
- type_operation
- statut
- Clés étrangères : id_temps, id_compte, id_devise, id_scenario, id_contrapartie

Cette table centralise les flux de trésorerie et permet les agrégations par période, compte, devise ou entité.

2. Tables de dimensions

- **DIM_TEMPS** — Gère la granularité temporelle des opérations (jour, mois, année). Permet les analyses par période.
- **DIM_COMPTE** — Contient les informations des comptes financiers (numéro, type de compte, filiale associée).
- **DIM_DEVISE** — Répertorie les devises utilisées, avec leur code ISO et libellé.
- **DIM_SCENARIO** — Définit les différents scénarios d'analyse (prévisionnel, réalisé, simulé…).
- **DIM_CONTREPARTIE** — Identifie les entités externes impliquées dans les opérations (fournisseurs, clients, partenaires).
- **DIM_FILIALE** — Contient les informations des filiales du groupe (nom, pays, région).

3. Relations

FACT_FLUX_TRESORERIE est au centre du modèle, liée à chaque dimension par des clés étrangères.

Chaque dimension entretient une relation de type 1,n avec la table de faits (une dimension peut concerner plusieurs flux).

Certaines dimensions (comme **DIM_COMPTE** et **DIM_FILIALE**) sont également liées entre elles via des clés étrangères pour représenter la hiérarchie organisationnelle.

Modèle de données

Le modèle de données est conçu en flocon, optimisé pour les analyses BI.

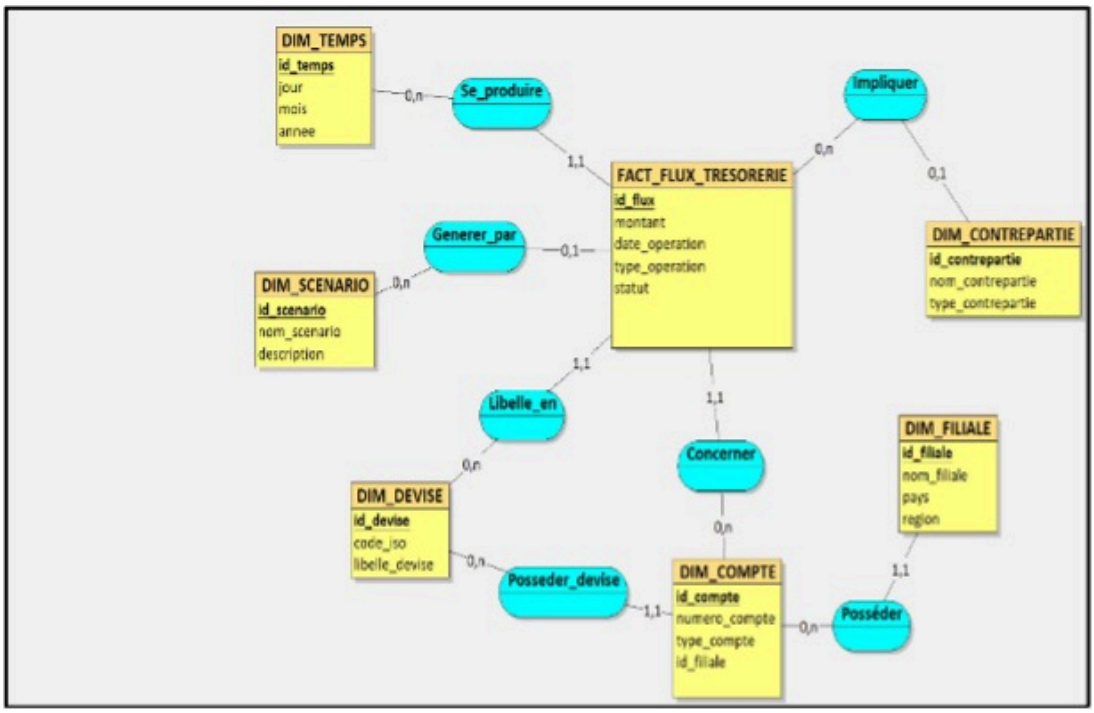

Fig. 1. – Modèle Conceptuel de Données (MCD)

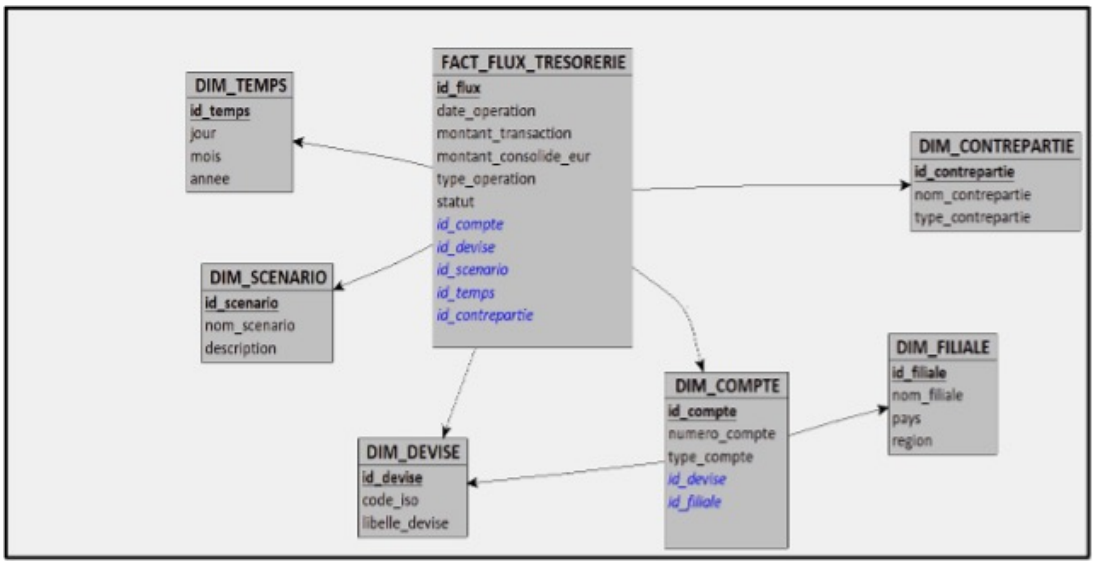

Fig. 2. – Modèle Logique de Données (MLD)

Dictionnaire de données

Le dictionnaire de données a pour objectif de définir de manière exhaustive l'ensemble des champs qui seront manipulés au sein du projet Crésus. Il sert de référentiel unique pour garantir une compréhension commune des informations utilisées, depuis leur source jusqu'à leur restitution.

Table : FACT_FLUX_TRESORERIE (Table de Faits)
Cette table est le cœur du modèle. Elle contient chaque transaction financière unitaire.

Champ	Type	Description
id_flux	INTEGER, PK	Identifiant unique de la transaction (flux).
date_operation	DATE	Date effective de l'opération.
montant_transaction	NUMERIC	Montant de la transaction dans la devise d'origine.
montant_consolide_eur	NUMERIC	Montant converti en EUR pour la consolidation.
type_operation	VARCHAR	Nature de l'opération (ex: « Virement émis », « Prêt »).
statut	VARCHAR	Statut de la transaction (ex: « Réalisé », « Prévisionnel »).
id_compte	INTEGER, FK	Clé étrangère liant à la table DIM_COMPTE.
id_devise	INTEGER, FK	Clé étrangère liant à DIM_DEVISE.
id_scenario	INTEGER, FK	Clé étrangère liant à DIM_SCENARIO.
id_temps	INTEGER, FK	Clé étrangère liant à DIM_TEMPS.
id_contrepartie	INTEGER, FK	Clé étrangère liant à DIM_CONTREPARTIE.

Tableau 1. – Structure de la table FACT_FLUX_TRESORERIE

Tables de dimensions

DIM_TEMPS
Gère la granularité temporelle des opérations.

Champ	Type	Description
id_temps	INTEGER, PK	Identifiant unique de la date.
jour	INTEGER	Jour du mois (1-31).
mois	INTEGER	Mois de l'année (1-12).
annee	INTEGER	Année (ex: 2024).

Tableau 2. – Structure de la table DIM_TEMPS

DIM_SCENARIO
Définit les différents scénarios d'analyse (réalisé, prévision, simulation).

Champ	Type	Description
id_scenario	INTEGER, PK	Identifiant unique du scénario.
nom_scenario	VARCHAR	Nom du scénario (ex: « Réalisé 2024 », « Simulation Taux +1% »).
description	TEXT	Description détaillée du scénario.

Tableau 3. – Structure de la table DIM_SCENARIO

DIM_DEVISE

Répertorie les devises utilisées pour les transactions et les comptes.

Champ	Type	Description
id_devise	INTEGER, PK	Identifiant unique de la devise.
code_iso	VARCHAR(3)	Code ISO 4217 de la devise (ex: « EUR », « USD », « GBP »).
libelle_devise	VARCHAR	Nom complet de la devise (ex: « Euro », « Dollar Américain »).

Tableau 4. – Structure de la table DIM_DEVISE

DIM_COMPTE

Contient les informations des comptes financiers du groupe.

Champ	Type	Description
id_compte	INTEGER, PK	Identifiant unique du compte.
numero_compte	VARCHAR	Identifiant métier du compte (ex: IBAN ou numéro interne).
type_compte	VARCHAR	Type de compte (ex: « Compte courant », « Compte de prêt »).
id_devise	INTEGER, FK	Clé étrangère liant à DIM_DEVISE.
id_filiale	INTEGER, FK	Clé étrangère liant à DIM_FILIALE.

Tableau 5. – Structure de la table DIM_COMPTE

DIM_CONTREPARTIE

Identifie les entités externes ou internes impliquées dans les opérations.

Champ	Type	Description
id_contrepartie	INTEGER, PK	Identifiant unique de la contrepartie.
nom_contrepartie	VARCHAR	Nom de la contrepartie (ex: « Client A », « Fournisseur B »).
type_contrepartie	VARCHAR	Catégorie (ex: « Client », « Fournisseur », « Banque », « Interco »).

Tableau 6. – Structure de la table DIM_CONTREPARTIE

DIM_FILIALE
Décrit les entités et filiales du groupe ZF Banque.

Champ	Type	Description
id_filiale	INTEGER, PK	Identifiant unique de la filiale.
nom_filiale	VARCHAR	Nom légal ou usuel de la filiale (ex: « ZF Banque France »).
pays	VARCHAR	Pays d'implantation de la filiale.
region	VARCHAR	Zone géographique (ex: « Europe », « Amérique du Nord »).

Tableau 7. – Structure de la table DIM_FILIALE

Mapping de données

Le mapping de données décrit les correspondances entre les données sources et les tables PostgreSQL, ainsi que les transformations appliquées lors du chargement.

Mapping de la table de faits

Donnée source	Table.Colonne cible	Transformation
id_flux	FACT_FLUX_TRESORERIE.id_flux	Auto-incrément
date_operation	FACT_FLUX_TRESORERIE.date_operation	Conversion au format SQL DATE
montant	FACT_FLUX_TRESORERIE.montant	Nettoyage caractères non numériques
type_operation	FACT_FLUX_TRESORERIE.type_operation	Normalisation en minuscules
statut	FACT_FLUX_TRESORERIE.statut	Valeur par défaut : « confirmé »
simulation	FACT_FLUX_TRESORERIE.simulation	Dérivé des scénarios Power BI

Tableau 8. – Mapping des données vers FACT_FLUX_TRESORERIE

Mapping des dimensions

Donnée source	Table.Colonne cible	Transformation
code_iso	DIM_DEVISE.code_iso	Jointure avec DIM_DEVISE
libelle_devise	DIM_DEVISE.libelle_devise	Uniformisation majuscule
taux_change	DIM_SCENARIO (calcul)	Intégré dans la table scénario
nom_scenario	DIM_SCENARIO.nom_scenario	Généré automatiquement par Python
description	DIM_SCENARIO.description	Renseigné par l'utilisateur Power BI

Tableau 9. – Mapping des données vers les dimensions (1/3)

Donnée source	Table.Colonne cible	Transformation
nom_filiale	DIM_FILIALE.nom_filiale	Normalisation et suppression doublons
pays	DIM_FILIALE.pays	Nettoyage des libellés
region	DIM_FILIALE.region	Correspondance pays → région
numero_compte	DIM_COMPTE.numero_compte	Masquage partiel (RGPD)
type_compte	DIM_COMPTE.type_compte	Catégorisation automatique
devise_compte	DIM_COMPTE.devise	Jointure avec DIM_DEVISE
id_filiale	DIM_COMPTE.id_filiale	Clé étrangère DIM_FILIALE

Tableau 10. – Mapping des données vers les dimensions (2/3)

Donnée source	Table.Colonne cible	Transformation
nom_contrepartie	DIM_CONTREPARTIE.nom_contrepartie	Uniformisation (majuscules)
type_contrepartie	DIM_CONTREPARTIE.type_contrepartie	Normalisation via dictionnaire Python
jour, mois, annee	DIM_TEMPS	Extraction depuis date_operation
source_fichier	Métadonnée interne	Ajout automatique (Airflow)
date_integration	Métadonnée interne	Générée par le pipeline Airflow

Tableau 11. – Mapping des données vers les dimensions (3/3)

ANNEXE

Document de spécifications techniques pour le Proof of Concept du projet Crésus.

Pour toute question, contacter l'équipe MOE.