REPUBLIQUE DU CAMEROUN
Paix – Travail – Patrie

UNIVERSITE DE DOUALA

ECOLE NATIONALE SUPÉRIEURE POLYTECHNIQUE DE DOUALA

REPUBLIC OF CAMEROON
Peace – Work – Fatherland

THE UNIVERSITY OF DOUALA

NATIONAL HIGHER POLYTECHNIC SCHOOL OF DOUALA

INGENIERIE ET CONCEPTION LOGICIELLE

Génie logiciel Niveau 4

ETUDE, ANALYSE ET CONCEPTION D'UN SYSTEME DE RECUPERATION DES DONNEES

(Cas d'un établissement bancaire)

Réalisé par :

Noms	**Matricules**
FOHOM MERVEILLES	22G00136
MEKOUNDE GUSTAVE	22G00253
MOHAMMED EL BACHIR	22G00262
NKAMOLOUM ANTHONY	22G00328
NTAKEU LEANE	22G00339

Sous la supervision de :

Dr. IHONOCK

Année académique 2025/2026

REMERCIEMENTS

Nous tenons à exprimer notre profonde gratitude à l'ensemble des personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail intitulé « Étude, Analyse et Conception d'un système de récupération des données ». Nous remercions particulièrement notre encadreur pédagogique pour son accompagnement constant, la qualité de ses orientations et la rigueur méthodologique qu'il nous a inculquée tout au long de ce projet. Nos remerciements s'adressent également aux responsables administratifs et techniques de notre établissement, dont la disponibilité et les ressources mises à notre disposition ont facilité l'avancement de nos recherches. Nous témoignons enfin notre reconnaissance à nos camarades de classe et aux membres du groupe pour leur collaboration, leur implication et l'esprit de partage qui ont permis d'aboutir à un travail collectif structuré et enrichissant. Ce projet n'aurait pu être mené à bien sans l'ensemble de ces contributions.

RESUME

Ce travail porte sur l'étude, l'analyse et la conception d'un système de récupération des données, en mettant particulièrement l'accent sur les enjeux de sécurité et de continuité de service dans un contexte bancaire. L'objectif principal est de comprendre les mécanismes fondamentaux de la perte de données, de présenter les techniques modernes de récupération et de proposer une architecture conceptuelle permettant d'assurer la restauration fiable d'informations sensibles, notamment des transactions bancaires et des fichiers critiques.

L'étude théorique explore les principales notions relatives au stockage, aux systèmes de fichiers, aux causes de corruption ou de suppression des données, ainsi que les approches techniques telles que le *signature scan*, le *file carving* et l'estimation du taux de récupérabilité. L'analyse fonctionnelle identifie ensuite les besoins spécifiques d'une institution bancaire, notamment la restauration des transactions supprimées, la récupération de fichiers à partir de supports corrompus, la traçabilité complète des opérations effectuées et l'interface décisionnelle via un tableau de bord.

Sur la base de ces exigences, une conception détaillée du système est proposée au moyen de diagrammes UML (cas d'utilisation, classes, séquence et déploiement), ainsi qu'un modèle de base de données adapté. Enfin, le travail propose une démarche de réalisation mettant en avant un prototype capable de démontrer trois fonctions clés : le *signature scan*, le *file carving* et l'estimation du taux de récupérabilité sur un support de stockage.

Ce projet offre ainsi une compréhension globale des principes de récupération de données et illustre leur application dans un environnement bancaire, où la fiabilité, la traçabilité et la précision constituent des impératifs majeurs.

Table de matières

LISTE DES TABLEAUX

LISTE DES FIGURES

INTRODUCTION

Dans un environnement numérique où les systèmes d'information occupent une place centrale dans la gestion des organisations, la disponibilité et l'intégrité des données constituent des enjeux stratégiques majeurs. Les institutions financières, et plus particulièrement les banques, dépendent fortement de la fiabilité de leurs bases de données et de leurs infrastructures de stockage pour assurer la continuité de leurs activités, la sécurité des transactions et la confiance des clients. Pourtant, aucune organisation n'est totalement à l'abri d'incidents pouvant entraîner une perte de données : erreurs humaines, défaillances logicielles, attaques malveillantes, corruptions de fichiers ou dysfonctionnements matériels. Ces situations peuvent avoir des conséquences critiques allant de la simple perturbation opérationnelle à l'interruption totale des services, voire à des impacts financiers et réputationnels importants.

Face à ces risques, les systèmes de récupération des données jouent un rôle déterminant. Ils permettent, selon le contexte, de restaurer des fichiers supprimés, de rétablir des données corrompues, de reconstruire une transaction perdue ou encore d'estimer les chances de récupération en cas d'incident grave. Dans le domaine bancaire, où les données sont particulièrement sensibles et souvent réglementées, la mise en place de mécanismes robustes, traçables et fiables représente une exigence incontournable.

Le présent travail s'inscrit dans cette problématique. Intitulé « Étude, Analyse et Conception d'un système de récupération des données », il vise à fournir une compréhension approfondie des fondements techniques et théoriques de la récupération de données, tout en proposant une solution conceptuelle adaptée au contexte bancaire. Après avoir examiné les différentes causes de perte de données et les technologies actuelles de récupération, nous analyserons les besoins spécifiques d'une banque en matière de restauration d'informations critiques. Cette analyse servira de base à l'élaboration d'une architecture conceptuelle intégrant des fonctionnalités essentielles telles que la restauration des transactions supprimées, le *signature scan*, le *file carving*, la traçabilité des opérations et l'estimation du taux de récupérabilité.

Au-delà de l'étude et de la conception, ce travail propose également une orientation vers la réalisation d'un prototype démonstratif, permettant d'implémenter certaines fonctionnalités fondamentales du système envisagé. L'objectif est ainsi de mettre en valeur la pertinence du projet, tant sur le plan théorique que pratique, et d'apporter une contribution à la compréhension des systèmes de récupération de données dans un cadre organisationnel aussi exigeant que le secteur bancaire.

CHAPITRE 1 : CADRE GÉNÉRAL, NOTIONS FONDAMENTALES ET ÉTAT DE L'ART DE LA RÉCUPÉRATION DES DONNÉES

1.1. Contexte général et problématique de la perte de données

L'essor du numérique a entraîné une croissance exponentielle du volume de données stockées et manipulées par les organisations. Dans les entreprises modernes, et plus particulièrement dans des secteurs critiques comme celui de la finance, les données représentent un actif stratégique indispensable au bon fonctionnement des opérations quotidiennes. Pourtant, ces données demeurent exposées à divers risques de perte ou de corruption. Une suppression accidentelle, une panne matérielle, un sinistre réseau, une attaque informatique ou encore une défaillance logicielle peuvent interrompre les processus internes et menacer la continuité des activités.

Dans le secteur bancaire, la perte de données peut entraîner des conséquences graves : incapacité de retracer une transaction, incohérences comptables, exposition aux sanctions réglementaires ou encore perte de confiance de la clientèle. La capacité d'une institution à restaurer rapidement et correctement ses données conditionne donc sa résilience opérationnelle. Cette problématique justifie la mise en œuvre de systèmes robustes de récupération des données capables non seulement de restaurer l'information perdue, mais aussi d'en assurer la traçabilité et l'intégrité.

Ainsi, la récupération des données n'est pas uniquement un processus technique ; elle représente un enjeu stratégique, organisationnel et sécuritaire. Ce chapitre vise à établir toutes les bases nécessaires pour comprendre son fonctionnement, ses acteurs, ses approches et ses limites.

1.2. Définitions et concepts clés

1.2.1. Données

Les données désignent l'ensemble des informations brutes stockées sous forme numérique. Elles peuvent prendre la forme de fichiers texte, de bases de données, d'images, de journaux de transactions ou de toute autre unité d'information exploitable par un système informatique. Dans un contexte bancaire, les données couvrent notamment les transactions financières, les historiques de comptes, les rapports d'audit, les justificatifs numérisés et les documents contractuels.

1.2.2. Récupération des données

La récupération des données est l'ensemble des techniques et procédés permettant de restaurer des informations perdues, supprimées ou corrompues, provenant d'un support de stockage endommagé ou non accessible. Cette discipline combine des notions de systèmes de

fichiers, de sécurité informatique, d'analyse hexadécimale, de forensic numérique et d'ingénierie logicielle.

1.2.3. Système de fichiers

Un système de fichiers est le mécanisme qui organise, structure et indexe les données stockées sur un support. NTFS, FAT32, exFAT, ext4 ou APFS constituent des exemples de systèmes de fichiers. La manière dont un système de fichiers gère l'allocation, la fragmentation et la suppression des données conditionne directement la facilité ou la difficulté de les récupérer.

1.2.4. Métadonnées

Les métadonnées sont des informations descriptives permettant d'identifier, de localiser ou d'interpréter un fichier : taille, date de création, emplacement, format, permissions. Leur corruption complique fortement la récupération.

1.3. Causes principales de perte de données

1.3.1. Erreurs humaines

La suppression accidentelle constitue la cause la plus fréquente. Un employé peut effacer un fichier sans vérifier sa criticité, ou encore détourner un dossier entier sans réaliser son importance. Dans un système bancaire, une mauvaise manipulation d'une base de données transactionnelle peut provoquer la disparition d'opérations importantes.

1.3.2. Défaillances matérielles

Les supports de stockage sont soumis à l'usure, à la surchauffe, à des chocs physiques ou à des défauts de fabrication. Un disque dur peut développer des secteurs défectueux, rendant illisibles certaines zones critiques. Ces pannes sont particulièrement dangereuses si le stockage ne dispose pas de réplications ou de sauvegardes régulières.

1.3.3. Attaques informatiques

Les attaques par ransomware, les effacements malveillants et les corruptions volontaires visent souvent à déstabiliser l'organisation ou à extorquer des fonds. Dans une banque, ce type d'incident peut affecter des données sensibles et compromettre la fiabilité des registres financiers.

1.3.4. Corruptions logicielles

Une erreur dans le code, une mise à jour échouée ou une fermeture soudaine du système peut entraîner une incohérence des données, notamment dans les bases transactionnelles qui nécessitent atomicité et cohérence (ACID).

1.3.5. Catastrophes physiques

Incendies, inondations, surtensions électriques, catastrophes naturelles : ces événements peuvent endommager les machines et rendre les supports partiellement ou complètement illisibles.

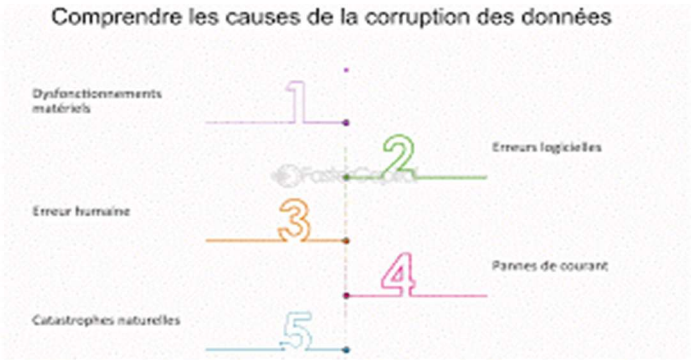

Figure 1 : Synthèse des différentes causes de corruption et perte des données

1.4. Fonctionnement interne d'un support de stockage

1.4.1. Structure logique du stockage

Les disques durs, SSD ou clés USB sont structurés en blocs ou clusters. Une donnée n'est jamais stockée d'un seul tenant ; elle est découpée en fragments dispersés. Le système de fichiers maintient une table d'allocation indiquant dans quels blocs se trouvent les fragments d'un fichier.

1.4.2. Suppression logique vs suppression réelle

Lorsque l'on supprime un fichier, celui-ci n'est pas effacé immédiatement. Le système de fichiers se contente de marquer son espace comme « disponible », sans écraser réellement les données. Elles restent présentes physiquement jusqu'à ce qu'un nouveau fichier recouvre ces blocs. C'est cette propriété qui rend possible la récupération.

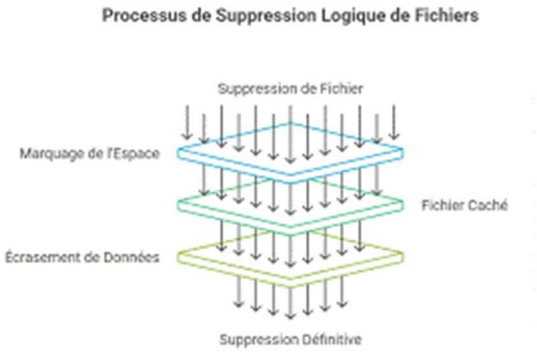

Figure 2 : Mécanisme de suppression logique

1.5. Les techniques modernes de récupération des données

1.5.1. Récupération à partir des métadonnées

Lorsque les entrées du système de fichiers sont intactes, il est possible de restaurer le fichier presque instantanément. Cette méthode est la plus fiable, mais elle dépend fortement de l'état des métadonnées.

1.5.2. Scan par signature (Signature Scan)

Cette technique consiste à rechercher dans la mémoire brute les signatures caractéristiques d'un type de fichier. Par exemple, un document PDF débute toujours par les octets « 25 50 44 46 ». Le logiciel balaie donc le disque à la recherche de ces signatures pour identifier des fragments de fichiers.

Tableau 1 : Quelques signatures hexadécimales de formats de fichiers critiques

Type de Fichier	Extension	En-tête (Header)	Pied de page (Footer)
Image	.jpg, .jpeg	FF D8 FF	FF D9
Document PDF	.pdf	25 50 44 46	0A 25 25 45 4F 46

1.5.3. File Carving

Le *file carving* reconstitue un fichier lorsque ses métadonnées sont perdues. À partir de signatures, de tailles approximatives et de cohérences internes, l'algorithme tente de recomposer les fragments dans le bon ordre. Il est particulièrement utile lorsque la table d'allocation est détruite.

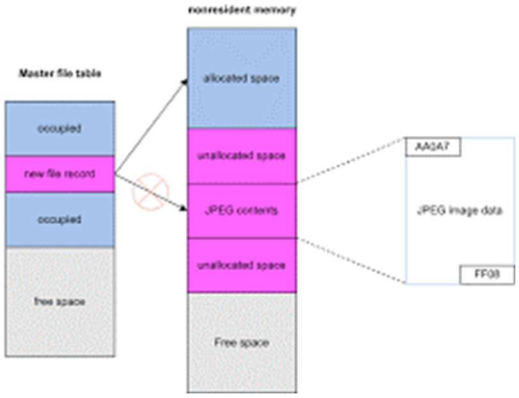

Figure 3 : Processus File Carving JPEG

1.5.4. Analyse forensique avancée

Cette approche, utilisée dans les enquêtes judiciaires, combine tracing, imagerie bit à bit, analyse heuristique et reconstruction d'artefacts pour retrouver des données dissimulées ou volatiles.

1.6. État de l’art : solutions actuelles de récupération

Les outils actuels se divisent en trois grandes catégories :

1.6.1. Outils grand public (RecuvA, EaseUS, DiskDigger)

Simples d’usage, ils permettent de restaurer rapidement des fichiers supprimés de manière basique. Leur puissance est limitée lorsque les données sont fortement corrompues.

1.6.2. Outils professionnels (R-Studio, Ontrack, GetDataBack)

Ils combinent analyse hexadécimale, carving, reconstruction RAID, imagerie et récupération avancée. Ils sont utilisés dans les grandes entreprises et dans les centres de forensic.

1.6.3. Solutions intégrées en entreprise

Certaines organisations disposent de systèmes internes spécialisés, intégrant journaux de transactions, sauvegardes automatisées, réplications, et procédures de restauration conformes à leur architecture opérationnelle.

1.7. Enjeux et limites des systèmes de récupération

1.7.1. Enjeux

- Garantir l'intégrité des données restaurées
- Minimiser les interruptions d’activité
- Assurer la traçabilité des opérations
- Prévenir la perte totale de fichiers critiques
- Renforcer la résilience organisationnelle

1.7.2. Limites

- Les données écrasées sont irrécupérables
- Le carving peut générer des fichiers incomplets
- Les coûts en temps et en ressources peuvent être élevés
- Certaines attaques modernes rendent volontairement la récupération impossible

Ce premier chapitre a permis de comprendre les fondements théoriques de la récupération des données, les mécanismes internes des systèmes de fichiers, les causes de perte

d’information, ainsi que les techniques et outils actuellement disponibles. Cette base conceptuelle est essentielle pour aborder, dans le chapitre suivant, l’analyse approfondie des besoins spécifiques d’un système de récupération appliqué à un contexte bancaire. Nous étudierons les exigences fonctionnelles, les contraintes techniques et sécuritaires, ainsi que les objectifs opérationnels que devra remplir un tel système pour répondre efficacement aux réalités du terrain.

CHAPITRE 2 : ANALYSE DES BESOINS, EXIGENCES ET SPÉCIFICATIONS DU SYSTÈME DE RÉCUPÉRATION DES DONNÉES EN MILIEU BANCAIRE

2.1. Introduction

Après avoir établi les fondements théoriques de la récupération des données dans le chapitre précédent, il est essentiel de s'intéresser désormais à l'analyse des besoins et des exigences propres à un environnement bancaire. Cette étape constitue une phase déterminante, car elle permet de traduire les concepts généraux en attentes concrètes, mesurables et adaptées au contexte réel dans lequel le système sera conçu. Le milieu bancaire se distingue par de fortes contraintes en matière de sécurité, d'intégrité et de continuité d'activité. L'analyse suivante vise donc à identifier les fonctionnalités incontournables, les acteurs impliqués, ainsi que les contraintes techniques et organisationnelles qui structureront la future conception.

2.2. Caractéristiques du contexte bancaire

Le secteur bancaire se caractérise par une manipulation quotidienne d'informations sensibles, notamment des soldes de comptes, des transactions financières, des justificatifs numériques et diverses données personnelles. La moindre incohérence ou absence d'information peut compromettre la fiabilité des résultats comptables et l'intégrité des services. La banque doit également assurer une conformité stricte aux réglementations (KYC, audit interne, lois financières), ce qui implique une traçabilité rigoureuse.

2.2.1. Sensibilité et criticité des données

Les données bancaires ne sont pas simplement importantes ; elles sont vitales. La perte d'une transaction, d'un justificatif ou d'une ligne d'historique peut remettre en question la confiance du client, perturber le bilan financier ou créer un risque juridique. La récupération des données, dans ce contexte, vise donc à prévenir toute rupture de la chaîne informationnelle.

2.2.2. Exigences de sécurité et de conformité

Le système doit répondre à des impératifs de confidentialité, d'intégrité et de disponibilité (CIA). De plus, il doit respecter les standards imposés par les autorités de contrôle, qui exigent une traçabilité complète des opérations et un archivage sécurisé.

Figure 4 : Les trois piliers fondamentaux de la sécurité informatique

2.2.3. Continuité de service

La banque doit rester opérationnelle 24 h/24. Par conséquent, un système de récupération doit permettre une restauration rapide afin d'éviter un arrêt prolongé des opérations. Cela implique des mécanismes d'analyse en parallèle et des outils permettant une estimation rapide du taux de récupérabilité.

2.3. Identification des acteurs du système

Afin de structurer les exigences du système, il est nécessaire d'identifier les différents acteurs qui interagiront avec lui.

2.3.1. Administrateur système

Il s'agit du responsable ayant un accès avancé aux fonctionnalités internes du système. Il gère la configuration, supervise les opérations critiques et valide les restaurations sensibles.

2.3.2. Analyste de données ou Agent d'audit

Cet acteur utilise le système pour récupérer des transactions, suivre l'historique des opérations et analyser les journaux en cas d'incident ou d'audit interne.

2.3.3. Utilisateur standard (bancaire)

Il peut demander un rapport de récupérabilité ou vérifier l'état d'un fichier. Ses droits sont limités afin d'éviter tout accès non autorisé.

2.3.4. Système de fichiers / Base de données bancaire

Bien qu'il ne s'agisse pas d'un acteur humain, ce système agit comme une source et une destination de données à restaurer. Il interagit avec la plateforme pour fournir ou recevoir les informations récupérées.

2.4. Analyse des besoins fonctionnels

Les besoins fonctionnels représentent les actions que le système doit permettre.

2.4.1. Restauration des transactions supprimées

Le système doit permettre d'identifier des opérations supprimées ou perdues et de les restaurer. Cela se fait par analyse des journaux transactionnels, analyse du journal Write-Ahead Logging (WAL), ou récupération directe à partir des pages de la base de données.

2.4.2. Récupération de fichiers supprimés

Cette fonctionnalité utilise deux techniques principales :

- **Signature Scan** : recherche des empreintes binaires caractéristiques.
- **File Carving** : reconstruction de fragments lorsque les métadonnées sont détruites.

Le système doit pouvoir analyser une clé USB, un disque dur interne ou externe.

2.4.3. Estimation du taux de récupérabilité

Cette fonctionnalité permet d'analyser l'état du support (taux de fragmentation, cohérence des blocs, présence de signatures intactes) et d'estimer la probabilité de récupération. Elle guide l'utilisateur dans la prise de décision.

2.4.4. Traçabilité des opérations de récupération

Toute action effectuée doit être enregistrée dans un journal sécurisé :

- date de l'opération,
- utilisateur concerné,
- type de récupération,
- résultats obtenus,
- fichiers restaurés.

Cela garantit la conformité réglementaire.

2.4.5. Tableau de bord de suivi

Une interface de visualisation doit permettre d'afficher :

- l'état du support scanné,
- le pourcentage de données récupérables,
- les fichiers identifiés par signature,
- les opérations de restauration effectuées,

- les alertes critiques.

2.5. Analyse des besoins non fonctionnels

2.5.1. Sécurité

Le système doit intégrer :

- un contrôle d'accès basé sur les rôles,
- une gestion stricte des permissions,
- un cryptage des journaux et des résultats de récupération,
- un isolement des espaces sensibles.

2.5.2. Performance

Le scan doit être optimisé pour analyser rapidement les supports volumineux, tout en évitant la surcharge de la machine.

2.5.3. Fiabilité

Le système doit fournir :

- un taux d'erreur minimal,
- une gestion des incidents,
- des mécanismes de vérification de l'intégrité des données récupérées.

2.5.4. Disponibilité

Le système doit être accessible en permanence et capable de fonctionner même sous forte charge.

2.5.5. Traçabilité et auditabilité

Chaque opération doit être enregistrée de manière horodatée, sécurisée et infalsifiable.

2.6. Contraintes du système

2.6.1. Contraintes techniques

- Compatibilité avec les systèmes de fichiers courants (NTFS, FAT32, ext4).
- Capacité à lire un support endommagé.
- Intégration avec la base de données bancaire.

- Capacité à gérer de grands volumes de données.

2.6.2. Contraintes organisationnelles

- Respect des processus internes de la banque.
- Conformité avec les politiques de sécurité.
- Gestion des droits utilisateurs.

2.6.3. Contraintes juridiques et réglementaires

- Respect des lois sur la protection des données.
- Conformité aux audit trails exigés par les régulateurs.
- Conservation sécurisée des traces de récupération.

2.7. Analyse des risques

2.7.1. Risques techniques

- Récupération partielle ou incorrecte.
- Nouveaux fichiers écrasant des données à récupérer.
- Support trop endommagé pour une lecture complète.

2.7.2. Risques de sécurité

- Accès non autorisé aux données sensibles restaurées.
- Falsification des journaux de récupération.
- Exploitation des outils de récupération à des fins malveillantes.

2.7.3. Risques opérationnels

- Temps de scan trop élevé.
- Dépendance à des ressources matérielles spécifiques.

2.7.4. Risques humains

- Mauvaise manipulation du système.
- Utilisation en dehors des normes établies.

2.8. Spécifications fonctionnelles détaillées

Chaque besoin fonctionnel est traduit en exigences précises.

1. Le système doit identifier automatiquement les signatures de fichiers connues.

Par exemple : PDF, DOCX, PNG, JPEG, TXT, SQL, MDF.

2. Le système doit reconstruire un fichier lorsque seules des signatures sont trouvées.

Ce carving doit être fait bloc par bloc.

3. Le système doit effectuer une estimation en pourcentage du taux de récupérabilité.

Basée sur :

- taux de fragmentation,
- blocs lisibles,
- blocs corrompus.

4. Le système doit journaliser toute action.

5. Le système doit restaurer, lorsque possible, les transactions supprimées à partir de journaux internes.

6. Le système doit afficher un tableau de bord synthétique.

7. Le système doit fonctionner sur Windows comme sur Linux.

Ce deuxième chapitre a permis d'identifier et d'analyser de manière détaillée les besoins, les acteurs, les contraintes et les risques associés à la mise en place d'un système de récupération de données en milieu bancaire. Cette analyse constitue une base solide pour passer à l'étape suivante : la conception du système. Dans le chapitre 3, nous formaliserons les résultats de cette analyse à travers des modèles UML, notamment les diagrammes de cas d'utilisation, de classes, de séquence et de déploiement, ainsi que le modèle conceptuel de données. Ce passage à la conception permettra de structurer précisément la future architecture et les interactions internes du système.

CHAPITRE 3 : CONCEPTION D'UN SYSTÈME DE RÉCUPÉRATION DES DONNÉES EN MILIEU BANCAIRE

3.1. Introduction

Après avoir identifié et défini les besoins fonctionnels et non fonctionnels du système dans l'analyse, il est maintenant indispensable de traduire ces exigences en une architecture logique et structurée. La phase de conception permet de formaliser les composantes du système, leurs relations, les fonctionnalités qu'elles doivent assurer, ainsi que la manière dont les acteurs interagiront avec ces composantes. Pour cela, nous nous appuyons sur des modèles UML, qui constituent un langage standard et universel utilisé en ingénierie logicielle afin de représenter visuellement un système complexe.

Ce chapitre présente donc l'ensemble des diagrammes nécessaires à la conception du système : le diagramme de cas d'utilisation, le diagramme de classes, le diagramme de séquence et le diagramme de déploiement. Enfin, nous proposons un modèle de base de données cohérent avec les besoins du système de récupération en contexte bancaire.

3.2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les interactions entre les acteurs et les principales fonctionnalités du système. Il décrit ce que le système doit faire, sans préciser comment. Il constitue l'une des premières vues logiques du système.

3.2.1. Acteurs identifiés

- **Acteur** (Utilisateur principal) : Peut être un administrateur système, un analyste bancaire ou un agent de récupération de données qui interagit avec le système de récupération.

3.2.2. Principaux cas d'utilisation

- **Restaurer Fichier** : Récupérer des fichiers bancaires supprimés ou corrompus
- **Restaurer Transaction** : Restaurer des transactions bancaires perdues
- **Estimer le taux de récupérabilité** : Calculer la probabilité de succès de récupération
- **Consulter** : Consulter les informations et l'état des récupérations
- **Journaliser** : Enregistrer toutes les opérations effectuées dans le système
- **Générer Rapport** : Produire des rapports détaillés sur les opérations de récupération
- **Scan** : Scanner les supports de stockage pour détecter les données
- **Signature Scan** : Identifier les signatures de fichiers bancaires
- **File carving** : Extraire et reconstruire les fichiers à partir des secteurs bruts
- **Generer Alert** : Générer des alertes en cas d'anomalies ou d'événements critiques

3.2.3. Relations entre cas d'utilisation

- **Restaurer Fichier** <<include>> **Scan** : La restauration de fichier nécessite obligatoirement un scan préalable
- **Restaurer Transaction** <<include>> **Scan** : La restauration de transaction inclut un scan du support
- **Restaurer Fichier** <<include>> **Signature Scan** : La restauration utilise les signatures pour identifier les fichiers
- **Scan** <<include>> **File carving** : Le scan inclut le processus de file carving pour extraire les données
- **File carving** <<extend>> **Generer Alert** : Le file carving peut déclencher des alertes si nécessaire
- **Générer Rapport** <<extend>> **File carving** : Un rapport peut être généré suite au file carving
- **Signature Scan** <<extend>> **File carving** : La détection de signatures peut étendre le processus de file carving

3.2.4. Description textuelle du fonctionnement

Lorsqu'un utilisateur souhaite effectuer une récupération de données bancaires, il lance d'abord un Scan du support de stockage (disque dur, clé USB, etc.). Le système procède alors à l'identification des Signatures Scan pour reconnaître les fichiers bancaires présents. Une fois les signatures détectées, le système applique la technique de File carving pour reconstruire les fichiers à partir des blocs de données brutes. Parallèlement, l'utilisateur peut Estimer le taux de récupérabilité pour évaluer les chances de succès de l'opération. Selon le besoin, l'utilisateur peut ensuite Restaurer Fichier (pour récupérer des documents bancaires) ou Restaurer Transaction (pour récupérer des enregistrements de transactions). Toutes les opérations sont automatiquement Journalisées dans le système pour assurer la traçabilité. L'utilisateur peut Consulter l'état et les détails des opérations en cours ou terminées. En fin de processus, le système permet de Générer Rapport détaillant les résultats de la récupération. Si des anomalies sont détectées durant le file carving, le système peut Generer Alert pour avertir l'utilisateur des problèmes potentiels.

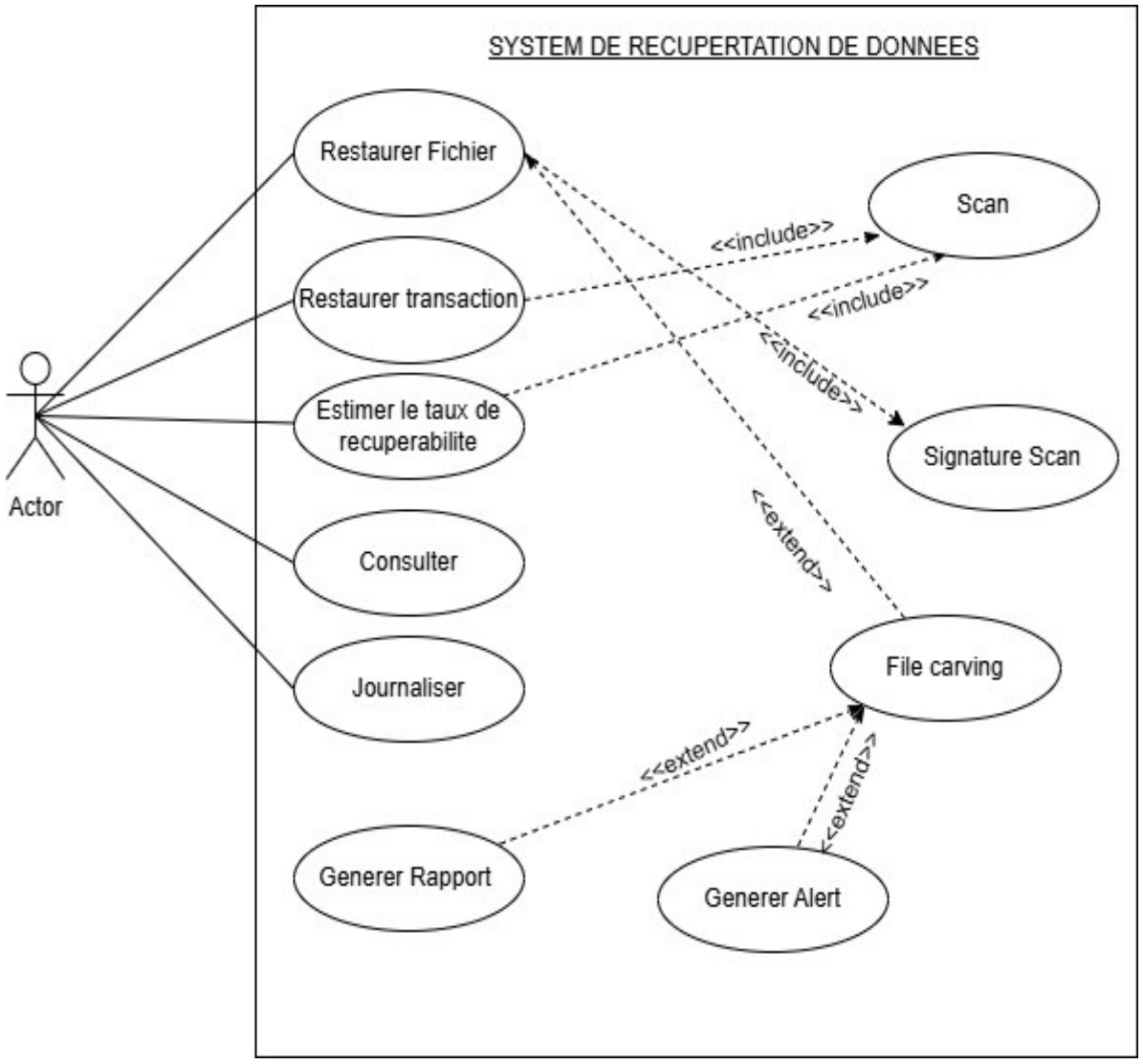

Figure 5 : Diagramme de cas d'utilisation

3.3. Diagramme de classes

Le diagramme de classes constitue le cœur du modèle conceptuel du système. Il représente les entités principales, leurs attributs, leurs méthodes et les relations structurelles entre elles.

3.3.1. Classes principales retenues

1. **CompteBancaire**
    - Attributs : idUtilisateur, utilisateur, solde
    - Méthodes : consulterCompte()
2. **SignatureFichier**
    - Attributs : idSignature, nomFichier, motifHex,descriptio
    - Méthodes : détecterSignature()

3. **ScanneurDonnee**
    - Attributs : idBloc, signature
    - Méthodes : scannerSecteur(),appliquerFileCarving()
4. **FichierBancaire**
    - Attributs : idFichier, nom, taille, chemin, statut
    - Méthodes : restaurer()
5. **TransactionBancaire**
    - Attributs : idTransaction, compte, date, montant, état
    - Méthodes : voirTransaction(), restaurer
6. **OperationRecuperation**
    - Attributs : idAction, utilisateur, dateAction, typeAction, résultat
    - Méthodes : enregistrer()
7. **Utilisateur**
    - Attributs : idUtilisateur, nom, rôle, motDePasse
    - Méthodes : seConnecter(), effectuerAction()
8. **TauxRecuperabilite**
    - Attributs : idTaux,taux
    - Méthodes : calculerTaux(), analyseEntropie()
9. **LogOperation**
    - Attributs : id,typeEvenement,description
    - Methodes : voirLog()
10. **Dashboard**
    - Attributs : nombreRecuperationEncours, nombreDeRecuperationTermine,tempsMoyen de recuperation, volumeDeDonneeRecupere
    - Methodes : afficherStats(),afficherAlerte()

3.3.2. Relations entre les classes

- CompteBancaire 1..1 → **TransactionBancaire** 0..1 (association "possède") : Un compte possède zéro ou plusieurs transactions.

- TransactionBancaire 0..* → **OperationsRecuperation** 0..* (association "concerne") : Une transaction peut être concernée par plusieurs opérations de récupération.
- OperationsRecuperation 1..1 → **Admin** 1..1 (association "effectue") : Chaque opération de récupération est effectuée par un administrateur.
- OperationsRecuperation 1..1 → **LogOperation** 1..1 (association "possede") : Chaque opération de récupération possede un historique de log d'operations.
- Operations**Récupération** 0..1 → **FichierBancaire** 1..* (association "concerne") : Une opération concerne un ou plusieurs fichiers bancaires.
- FichierBancaire 0..* → **SignatureFichier** 1..* (association "correspond") : Un fichier correspond à une ou plusieurs signatures.
- TauxRecuperabilite 1..1 → **OperationsRecuperation** 1..* (association "fournit") : Un taux est fourni pour chaque opération de récupération.
- ScanneurDonnee 1..1 → **TauxRecuperabilite** 1..* (dépendance) : Le scanneur utilise le taux de récupérabilité pour ses analyses.
- **Dashboard** → **OperationsRecuperation** (association implicite d'affichage) : Le dashboard agrège et affiche les statistiques des opérations de récupération
- Dashboard 1 → **Admin** 1 (composition) : Le dashboard est une composante intégrée de l'interface administrateur. Ces classes et relations assurent une cohérence parfaite avec les fonctionnalités analysées au chapitre précédent.
- **ScanneurDonnee** → **SignatureFichier** (association implicite via file carving) : Le scanneur utilise les signatures pour identifier et détecter les fichiers bancaires lors du scan des secteurs.
- **ScanneurDonnee** → **FichierBancaire** (association implicite de production) : Le scanneur produit/restaure des fichiers bancaires après avoir scanné les données et appliqué les techniques de file carving

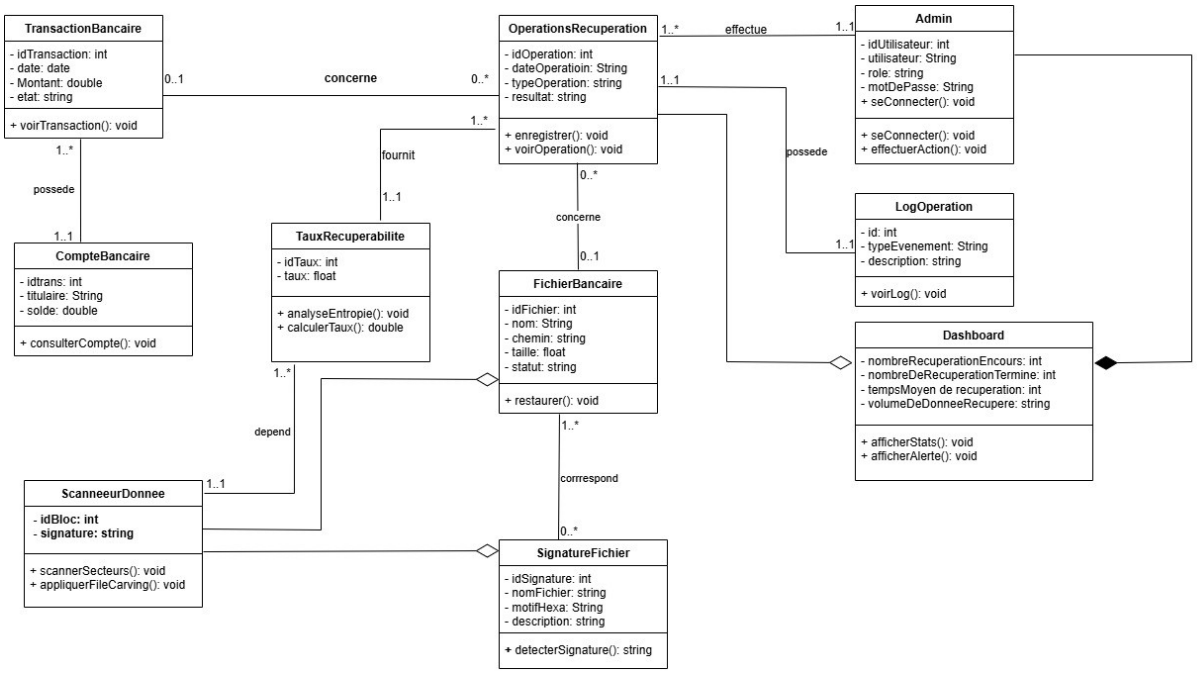

Figure 6 : Diagramme de classes

3.4. Diagramme de séquence

Le diagramme de séquence représente un scénario dynamique montrant les interactions entre objets au cours du temps. Nous choisissons ici le scénario : **“Analyse et récupération de fichiers supprimés”**. Ce diagramme de séquence illustre le **workflow complet de récupération de transactions bancaires** depuis la demande initiale de l'utilisateur jusqu'à la mise à jour du tableau de bord. Le processus se déroule en deux phases principales : la restauration de données et la récupération proprement dite.

- **Phase 1 : Demande de restauration**

  Le processus débute lorsque l'**utilisateur** formule une **demande de restauration** via l'**Interface**. Cette requête est transmise au composant **Scanner données** qui analyse la demande. Le système sollicite ensuite le module **Signature fichier** pour lancer un scan des données disponibles. Une fois le scan effectué, le module **Estimation du taux de récupération** calcule le taux de récupération potentiel en se basant sur l'analyse des fichiers scannés. Ce taux estimé est ensuite renvoyé à l'utilisateur via l'interface, lui permettant d'évaluer la faisabilité de la récupération avant de poursuivre.

- **Phase 2 : Demande de récupération**

Si l'utilisateur confirme la **demande de récupération**, l'interface transmet cette requête au module **Opération de récupération**. Ce dernier coordonne alors plusieurs actions parallèles :

1. **Demande de signature** : Le système interroge le module **Signature fichier** pour obtenir les signatures nécessaires à l'identification et à la validation des fichiers à récupérer.
2. **Lancement du file carving** : Le module **File carver** est activé pour extraire les données des zones endommagées ou supprimées du support de stockage.
3. **Envoi des données** : Les données récupérées sont transmises au module **Opération de récupération** qui centralise les résultats.

- ### Phase 3 : Logging et mise à jour du Dashboard

Une fois les données récupérées, le système enregistre l'opération dans le module **Log opération** afin d'assurer la traçabilité et l'auditabilité du processus. Enfin, le **Dashboard** est mis à jour pour refléter les nouvelles données restaurées, affichant un **message de réussite** à l'utilisateur via l'interface et permettant un suivi en temps réel des opérations de récupération.

- ### Interactions et flux de données

Ce diagramme met en évidence les **interactions asynchrones** entre les différents composants du système, notamment :

- La communication bidirectionnelle entre l'interface utilisateur et les modules backend
- La coordination entre les modules d'analyse (Scanner, Estimation) et d'exécution (File carver, Opération de récupération)
- La persistance des opérations via le système de logging
- La mise à jour en temps réel du tableau de bord pour une expérience utilisateur optimale

Ce processus garantit une récupération de données **sécurisée, traçable et transparente**, tout en offrant à l'utilisateur un contrôle total sur les opérations effectuées.

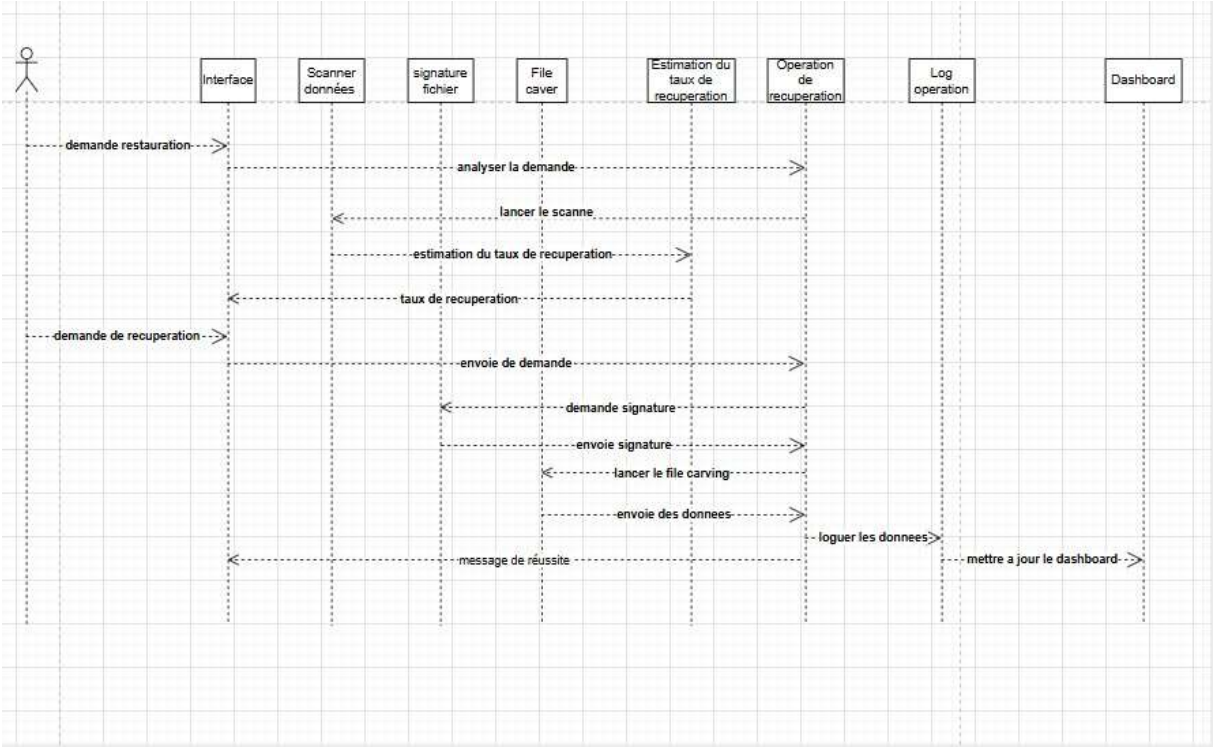

Figure 7 : Diagramme de séquence

3.5. Diagramme de déploiement

Le diagramme de déploiement présente l'architecture physique et technique du système de récupération de transactions bancaires. Il illustre la répartition des composants logiciels sur les différents nœuds matériels et les protocoles de communication qui les relient. Cette architecture multi-couches garantit la séparation des responsabilités, la scalabilité et la haute disponibilité du système.

Le système est organisé selon une **architecture en quatre couches distinctes** : la couche client (présentation), la couche application (logique métier) et la couche données (persistance). Cette séparation modulaire permet une maintenance aisée, une évolutivité optimale et une isolation des pannes potentielles.

3.5.1. Couche Client (Présentation)

La couche client constitue le point d'entrée du système et regroupe l'ensemble des dispositifs permettant l'interaction avec les utilisateurs et les sources de données externes.

Le **poste utilisateur** représente l'environnement de travail standard des opérateurs du système. Il s'agit d'une machine sous système d'exploitation Windows, MacOS ou Linux équipée d'un navigateur web moderne (Chrome, Firefox). Ce poste héberge l'**interface web** qui permet aux utilisateurs de formuler des demandes de restauration et de récupération de

données. La communication avec le serveur s'effectue exclusivement via le protocole HTTPS pour garantir la confidentialité et l'intégrité des échanges.

L'**appareil mobile administrateur** offre aux administrateurs système un accès mobile au tableau de bord de supervision. Disponible sous iOS et Android, cette application native permet le suivi en temps réel des opérations de récupération, la consultation des logs et la réception d'alertes critiques. La connexion s'établit via l'API REST du système et utilise des WebSockets pour assurer des mises à jour instantanées du dashboard.

Le **scanner matériel** constitue un dispositif spécialisé dédié à l'acquisition physique des données bancaires. Connecté au réseau via interface USB ou Ethernet, ce périphérique intègre les pilotes nécessaires à la numérisation et transmet les données brutes vers le serveur métier pour traitement ultérieur.

3.5.2. Couche Application (Serveurs Métier)

La couche application représente le cœur du système et se décompose en trois serveurs applicatifs spécialisés, chacun assumant des responsabilités distinctes dans le traitement des requêtes.

- Serveur Web / API

Le **serveur Web/API** constitue la façade du système et assure la fonction de point d'entrée unique pour toutes les requêtes externes. Déployé sur une machine Linux Ubuntu Server dotée de 32 GB de RAM et 8 cœurs de processeur, il héberge un serveur d'applications Tomcat ou Node.js selon les besoins technologiques.

Ce serveur accueille plusieurs composants critiques packagés sous forme d'archives déployables. Le composant **Interface.war** encapsule l'interface utilisateur web développée en technologies front-end modernes (React, Angular ou Vue.js). Le module **API_REST.jar** expose l'ensemble des endpoints RESTful permettant aux clients d'interagir avec le système selon les principes architecturaux REST. Le composant **Authentification.jar** gère le cycle de vie des sessions utilisateurs, l'authentification multi-facteurs et les autorisations d'accès. Le module **GestionSessions.jar** maintient l'état des sessions actives et assure leur persistance via le cache Redis. Enfin, le composant **Dashboard.war** fournit l'interface d'administration temps réel accessible aux administrateurs.

- Serveur Métier (Business Logic)

Le **serveur métier** orchestre l'ensemble de la logique applicative et coordonne les interactions entre les différents composants du système. Hébergé sur une infrastructure Linux CentOS équipée de 64 GB de RAM et 16 cœurs, ce serveur exécute un conteneur JBoss ou Spring Boot pour la gestion des composants métier.

Les composants déployés sur ce serveur implémentent les processus métier critiques du système. Le module **EstimationRecuperable.jar** analyse l'entropie des données et calcule le score de récupérabilité en exploitant des algorithmes d'analyse statistique avancés. Le composant **OperationRecuperation.jar** gère le cycle de vie complet des opérations de récupération, de leur initiation à leur clôture, en passant par le suivi de leur progression. Le module **FileCarver.jar** implémente les techniques de file carving permettant d'extraire des données exploitables à partir de supports endommagés ou partiellement corrompus. Le composant **SignatureFichier.jar** maintient la bibliothèque des signatures de fichiers et assure la validation de l'intégrité des données récupérées via calcul de hash cryptographiques. Enfin, le module **ScannerDonnees.jar** traite les flux de données provenant des scanners matériels et coordonne leur intégration dans le système.

La communication entre le serveur web et le serveur métier s'établit via le protocole RMI (Remote Method Invocation) ou EJB (Enterprise JavaBeans), garantissant des appels de méthodes distantes performants et fiables dans un environnement distribué Java.

- Serveur de Traitement

Le **serveur de traitement** constitue la plateforme de calcul haute performance du système, dédiée aux opérations computationnellement intensives. Déployé sur une infrastructure Linux Red Hat dotée de ressources matérielles conséquentes (128 GB de RAM, 32 cœurs), ce serveur héberge un cluster Apache Spark pour le traitement distribué et parallèle des données.

Les moteurs de traitement déployés sur cette plateforme assurent les analyses complexes. Le module **AnalyseEntropie.jar** évalue le niveau de désorganisation des données et détermine la faisabilité technique de leur récupération. Le composant **CalculScore.jar** agrège les métriques d'analyse pour produire un score de récupérabilité exploitable par les opérateurs. Le moteur **MoteurRecuperation.jar** orchestre l'ensemble du processus de récupération en coordonnant les différentes étapes techniques (scan, carving, validation, reconstruction). Enfin, le module **LogOperations.jar** assure la journalisation exhaustive de toutes les opérations système à des fins d'audit et de traçabilité.

La communication entre le serveur métier et le serveur de traitement s'effectue via le protocole gRPC (Google Remote Procedure Call), offrant des performances optimales pour les échanges inter-services grâce à la sérialisation binaire Protocol Buffers.

3.5.3. Couche Données (Persistance)

La couche données assure la persistance fiable et durable de l'ensemble des informations du système à travers trois systèmes de stockage complémentaires, chacun optimisé pour des besoins spécifiques.

- Serveur de Base de Données Relationnelle

Le **serveur de base de données relationnelle** héberge le modèle de données structuré du système. Déployé sur Linux Ubuntu Server avec 64 GB de RAM et 2 TB de stockage SSD, ce serveur exécute PostgreSQL 15 ou MySQL 8 selon les contraintes techniques du projet.

Ce système de gestion de base de données stocke les entités relationnelles définies dans le modèle conceptuel : **TRANSACTION_BANCAIRE** (historique exhaustif des transactions), **COMPTE_BANCAIRE** (informations des comptes clients), **ADMINISTRATEUR** (profils des utilisateurs système), et **LOG_OPERATION** (journalisation des événements système). La configuration en réplication Master-Slave garantit la haute disponibilité du système en cas de défaillance du serveur principal, avec bascule automatique vers le serveur secondaire.

L'accès aux données s'effectue via les protocoles JDBC (Java Database Connectivity) pour les connexions bas niveau et JPA/Hibernate pour la couche d'abstraction objet-relationnel, simplifiant le développement et améliorant la maintenabilité du code d'accès aux données.

- Serveur de Fichiers (NAS/SAN)

Le **serveur de fichiers** constitue le système de stockage massif dédié aux fichiers bancaires volumineux issus des processus de récupération. Implémenté sur une solution NAS (Network Attached Storage) ou SAN (Storage Area Network) sous FreeNAS ou TrueNAS, ce serveur offre une capacité de 50 téraoctets configurée en RAID 10 pour allier performance et redondance.

Ce système stocke deux collections principales : **FICHIER_BANCAIRE** qui contient les données brutes scannées depuis les supports physiques, et **SIGNATURE_FICHIER** qui maintient les métadonnées associées (hash, taille, date, type MIME) permettant l'identification et la validation ultérieure des fichiers. L'accès à ce stockage partagé s'effectue via les protocoles réseau standard NFS (Network File System) pour les environnements Unix/Linux et SMB (Server Message Block) pour l'interopérabilité avec les systèmes Windows.

- Base de Données NoSQL (Cache)

La **base de données NoSQL** optimise les performances du système en fournissant un cache haute vitesse et un stockage temporaire pour les données fréquemment consultées. Déployée sur Linux Alpine avec 32 GB de RAM, cette instance Redis ou MongoDB fonctionne intégralement en mémoire pour garantir des temps de réponse inférieurs à la milliseconde.

Ce système stocke deux collections critiques pour la performance : **ESTIMATION_RECUPERABLE** qui cache les résultats d'estimation pour éviter les recalculs coûteux, et **OPERATION_RECUPERATION** qui maintient un historique rapide des opérations récentes pour consultation immédiate. Au-delà du cache applicatif, cette base NoSQL gère également les sessions utilisateurs actives et les résultats temporaires des calculs intermédiaires, allégeant ainsi la charge sur la base relationnelle principale.

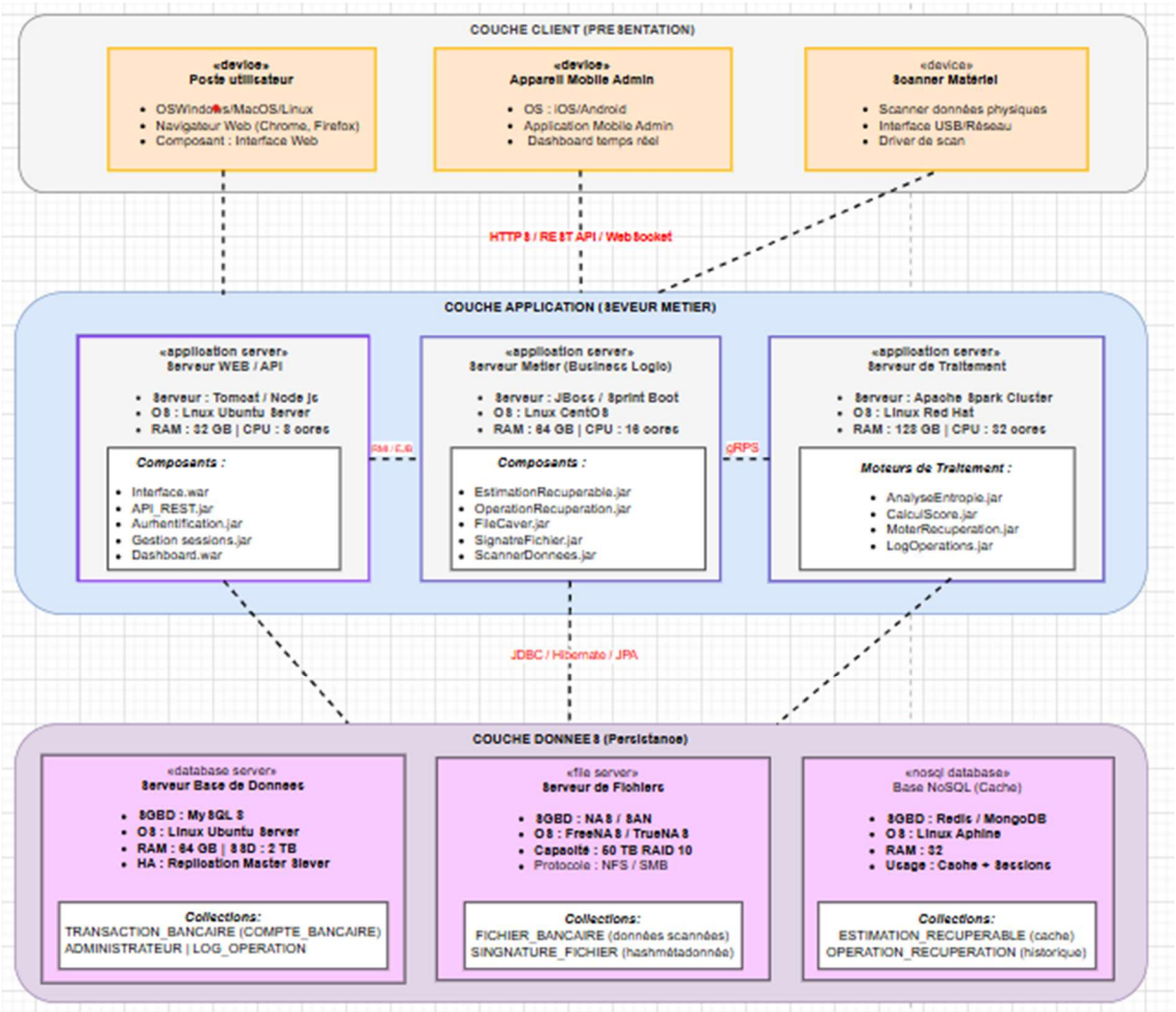

Figure 8 : Diagramme de déploiement

3.6. Modèle conceptuel de données (MCD)

Le Modèle Conceptuel de Données (MCD) de notre système de récupération de transactions bancaires est structuré autour de **huit entités principales** interconnectées par des **relations métier** reflétant les processus de gestion, d'analyse et de récupération des données bancaires.

3.6.1. Entités principales

- **TRANSACTION_BANCAIRE**

Cette entité centralise l'ensemble des transactions financières du système. Elle contient les attributs suivants :

- **idTrans** (clé primaire) : identifiant unique de la transaction
- **montant** : montant de la transaction
- **type_transaction** : nature de l'opération (débit, crédit, virement, etc.)

- **statut** : état actuel de la transaction (validée, en attente, échouée)

- **COMPTE_BANCAIRE**

Représente les comptes bancaires des utilisateurs avec :

- **idCompte** (clé primaire) : identifiant unique du compte
- **titulaire** : nom du propriétaire du compte
- **solde** : solde actuel du compte

- **ESTIMATION_RECUPERABLE**

Module d'analyse permettant d'évaluer la récupérabilité des données avec :

- **id** (clé primaire) : identifiant de l'estimation
- **score** : taux de récupération estimé (en pourcentage)
- Les méthodes **analyseEntropie()** et **calculerScore()** pour l'évaluation

- **FICHIER_BANCAIRE**

Gère les fichiers contenant les données bancaires :

- **idFichier** (clé primaire) : identifiant unique du fichier
- **nom** : nom du fichier
- **chemin** : emplacement physique du fichier
- **taille** : taille en octets
- **statut** : état du fichier (intact, corrompu, en cours de récupération)

- **SIGNATURE_FICHIER**

Stocke les signatures numériques pour l'identification et la validation des fichiers :

- **idSignature** (clé primaire) : identifiant de la signature
- **modHash** : empreinte cryptographique du fichier
- **description** : informations descriptives sur la signature

- **OPERATION_RECUPERATION**

Trace toutes les opérations de récupération effectuées :

- **idOperation** (clé primaire) : identifiant de l'opération
- **dateOperation** : date et heure de l'opération
- **typeOperation** : type de récupération effectuée
- **statut** : état de l'opération (en cours, réussie, échouée)
- **tauxReussi** : pourcentage de réussite de la récupération
- **recuperation** : montant ou volume de données récupérées

- **ADMINISTRATEUR**

Gère les utilisateurs administrateurs du système :

- **id** (clé primaire) : identifiant de l'administrateur
- **nom** : nom de l'administrateur
- **role** : rôle et permissions associés

- **LOG_OPERATION**

Assure la traçabilité des actions système :

- **id** (clé primaire) : identifiant du log
- **typeEvenement** : catégorie d'événement enregistré
- **description** : détails de l'événement

3.6.2. Relations et cardinalités

- POSSEDE (COMPTE_BANCAIRE ↔ TRANSACTION_BANCAIRE)

Cardinalités : (1,1) --- (1,*)

Un compte bancaire possède obligatoirement une ou plusieurs transactions, et chaque transaction appartient à un seul compte. Cette relation garantit l'intégrité référentielle entre les comptes et leurs opérations financières.

- DEPEND (COMPTE_BANCAIRE ↔ ESTIMATION_RECUPERABLE)

Cardinalités : (1,1) --- (1,*)

Chaque estimation de récupération dépend d'un unique compte bancaire, mais un compte peut faire l'objet de plusieurs estimations au fil du temps. Cette relation permet de suivre l'évolution de la récupérabilité des données d'un compte.

- CONCERNE (ESTIMATION_RECUPERABLE ↔ FICHIER_BANCAIRE)

Cardinalités : (0,1) --- (0,*)

Une estimation peut ne concerner aucun fichier (estimation théorique) ou plusieurs fichiers bancaires. Inversement, un fichier peut ne pas être concerné par une estimation s'il n'a pas encore été analysé. Cette relation flexible permet une évaluation progressive des fichiers.

- CORRESPOND (FICHIER_BANCAIRE ↔ SIGNATURE_FICHIER)

Cardinalités : (1,*) --- (0,*)

Un fichier bancaire peut correspondre à plusieurs signatures (versions différentes, fragments), et une signature peut être associée à plusieurs fichiers (fichiers similaires ou dupliqués). Cette relation est essentielle pour le processus de file carving et d'identification des données.

- FOURNIT (ESTIMATION_RECUPERABLE ↔ OPERATION_RECUPERATION)

Cardinalités : (1,1) --- (1,1)

Chaque estimation aboutit à une unique opération de récupération, et chaque opération découle d'une seule estimation. Cette relation de type 1:1 matérialise le passage de l'analyse à l'action concrète de récupération.

- EFFECTUE (ADMINISTRATEUR ↔ OPERATION_RECUPERATION)

Cardinalités : (1,1) --- (0,*)

Un administrateur peut effectuer zéro ou plusieurs opérations de récupération, mais chaque opération est nécessairement effectuée par un seul administrateur. Cette relation assure la **traçabilité et la responsabilité** des actions de récupération.

- POSSEDE_LOG (ADMINISTRATEUR ↔ LOG_OPERATION)

Cardinalités : (1,1) --- (0,*)

Chaque log est associé à un unique administrateur, permettant ainsi un **audit complet** des actions effectuées par chaque utilisateur du système. Un administrateur peut avoir généré aucun ou plusieurs logs selon son activité.

3.6.3. Architecture et cohérence du modèle

Ce MCD respecte les **règles de normalisation** et assure :

- **L'intégrité référentielle** entre toutes les entités
- **La traçabilité complète** des opérations via les logs et les relations avec l'administrateur
- **La flexibilité** nécessaire pour gérer différents scénarios de récupération
- **La sécurité** par l'association systématique des actions à un administrateur identifié

Le modèle couvre l'ensemble du cycle de vie d'une récupération de données bancaires, depuis l'analyse initiale (Estimation) jusqu'à l'exécution (Opération) et la journalisation (Log), en passant par la gestion des fichiers et de leurs signatures pour une récupération efficace et sécurisée.

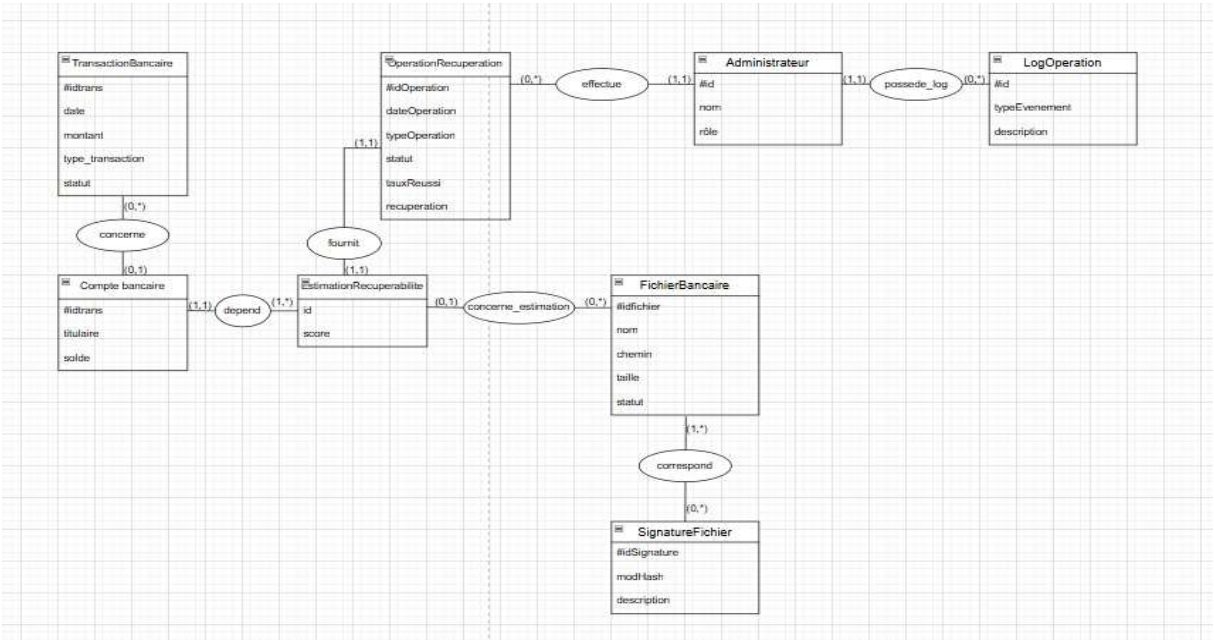

Figure 9 : Modèle conceptuel de données

CHAPITRE 4 : CONCEPTION DU PROTOTYPE DE SYSTÈME DE RÉCUPÉRATION DE DONNÉES

4.1. Introduction

Après avoir étudié le contexte général de la récupération de données, analysé les besoins spécifiques du milieu bancaire et identifié les contraintes techniques et organisationnelles, il devient nécessaire de traduire ces éléments en une architecture cohérente et réalisable. Toutefois, conformément au périmètre retenu pour la mise en œuvre, il ne s'agit pas de développer un système complet de récupération bancaire, mais **un prototype réduit**, ciblé sur quelques fonctionnalités essentielles démontrant les mécanismes fondamentaux d'un système réel.

Ce chapitre présente donc la conception du prototype, en mettant l'accent sur :

- les fonctionnalités réellement implémentées,
- les simplifications adoptées,
- les modèles UML nécessaires (cas d'utilisation, classes, séquences, déploiement),
- et le modèle de base de données minimaliste.

L'objectif est de produire une conception fonctionnelle, claire et cohérente, tout en restant adaptée à un projet académique.

4.2. Positionnement et périmètre du prototype

Le prototype ne cherche pas à couvrir tous les aspects d'un système bancaire complet. Il vise plutôt à démontrer une sélection de mécanismes clés et reproductibles qui constituent le cœur d'un système de récupération de données.

Les fonctionnalités retenues pour la réalisation

Le prototype intègre uniquement :

1. **Le Signature Scan**
   Capacité à analyser un support (disque dur, clé USB) pour retrouver des fichiers supprimés en identifiant leurs signatures (headers/footers).
2. **Le File Carving**
   Reconstructions simples de fichiers à partir de blocs consécutifs retrouvés.
3. **L'estimation du taux de récupérabilité**
   Petit module permettant d'évaluer la qualité de récupération d'un fichier trouvé.

4. **Un enregistrement des opérations réalisées**
   Pour assurer une traçabilité minimaliste.
5. **Un Rapport d'exécution**

Fonctionnalités non intégrées (mais couvertes dans la conception globale)

Elles sont décrites dans les chapitres précédents mais non implémentées car hors périmètre du prototype :

- restauration des transactions bancaires supprimées ;
- intégration complète à un SI bancaire ;
- reconstruction avancée de fichiers fragmentés ;
- gestion fine des permissions utilisateurs ;
- audit cryptographiquement signé.

Le chapitre 4 se concentre donc sur la **conception réaliste de ce prototype réduit**, tout en conservant une logique professionnelle.

4.3. Spécification fonctionnelle du prototype

4.3.1. Analyse d'un support externe

L'utilisateur sélectionne un disque ou une clé USB, et le système effectue :

- une lecture sectorielle simple,
- un balayage à la recherche de signatures de fichiers selon une liste prédéfinie (JPEG, PNG, PDF, CSV, DOCX).

4.3.2. Signature Scan

Le moteur vérifie la présence d'en-têtes connus :

- FF D8 pour JPEG,
- 25 50 44 pour PDF,
- 89 50 4E 47 pour PNG…

Il liste ensuite les fichiers potentiellement récupérables.

4.3.3. File Carving

Lorsque des signatures sont détectées, le prototype reconstitue :

- soit des fichiers entièrement contenus dans des blocs consécutifs,

- soit des fragments simples lorsque la fragmentation est faible.

4.3.4. Estimation du taux de récupérabilité

Le taux est calculé à partir de :

- la complétude estimée du fichier reconstitué,
- la présence ou non du footer,
- la taille obtenue vs taille attendue.

Taux = (Blocs valides / Blocs attendus) × 100

4.3.5. Historique des opérations

Chaque scan effectué génère un enregistrement (date, fichier trouvé, taux estimé).

4.3.6. Rapport d'exécution

Une fois la récupération effectué, l'utilisateur peut consulter et importer un rapport des opérations effectuées.

4.4. Choix techniques et environnement de développement

4.4.1. Langage de programmation

Le prototype a été réalisé en **Python**, principalement pour les raisons suivantes :

- il dispose de bibliothèques puissantes pour la lecture bas niveau des disques ;
- il est largement utilisé dans le domaine de la cybersécurité et du forensic ;
- sa syntaxe simple facilite la compréhension dans un cadre académique ;
- il permet une intégration rapide avec des interfaces graphiques et des fichiers Excel/CSV.

4.4.2. Interface graphique

Le choix s'est porté sur **Tkinter**, intégré nativement à Python, car :

- il permet de développer des interfaces simples et fonctionnelles,
- il est léger et adapté pour un prototype d'application locale,
- il ne nécessite aucune installation supplémentaire.

4.4.3. Système d'exploitation et matériel

- **Système d'exploitation :** Windows 10/11

- **Supports testés :** clé USB 8 Go et disque dur externe
- **Mode d'accès :** lecture brute (raw read) via Python

4.4.4. Format des fichiers pris en charge

Le prototype prend en charge un ensemble limité mais représentatif :

- JPEG
- PNG
- PDF
- CSV
- DOCX

Ces formats ont été retenus car ils disposent de signatures bien documentées et couramment manipulées.

4.5. Architecture logicielle du prototype

Le prototype s'organise autour de quatre modules majeurs :

1. Module de lecture disque

- effectue la lecture séquentielle du support ;
- accède aux blocs de données brutes ;
- isole les zones contenant des métadonnées ou des signatures reconnues.

2. Module Signature Scan

- contient une base de signatures (headers/footers) ;
- détecte l'emplacement des fichiers potentiels ;
- renvoie une liste des zones candidates à la récupération.

3. Module File Carving

- reconstitue les fichiers trouvés selon les blocs détectés ;
- gère la reconstruction simple sans fragmentation complexe ;
- sauvegarde les fichiers récupérés dans un répertoire local.

4. Module d'estimation du taux de récupérabilité

- mesure le taux de complétude des fichiers reconstruits ;

- évalue la correspondance header/footer ;
- calcule une estimation de qualité (%).

5. Module de persistance (base de données locale)

- enregistre les opérations dans SQLite ;
- stocke :
    - les fichiers récupérés,
    - leurs caractéristiques,
    - leurs taux de récupérabilité.

6. Interface graphique

- permet de lancer un scan,
- affiche les résultats,
- visualise les statistiques via un mini tableau de bord.

4.6. Tests et validation du prototype

4.6.1. Tests de détection des signatures

Réalisés sur :

- clé USB contenant des fichiers supprimés récents,
- clé USB formatée rapidement,
- disque dur externe avec des fichiers effacés.

Résultats :

- bonne détection JPEG/PNG,
- détection correcte des PDF,
- taux plus faible pour CSV en raison de signatures moins robustes.

4.6.2. Tests de reconstruction

- fichiers JPEG : reconstruction complète dans la majorité des cas, footer souvent présent ;
- fichiers PNG : fortune variable selon la fragmentation ;
- fichiers PDF : reconstruction partielle ou corrompue dans certains cas.

4.6.3. Tests sur l'estimation du taux de récupérabilité

Les résultats sont cohérents avec l'état réel des fichiers, avec une marge d'erreur acceptable pour un prototype.

4.6.4. Limitations observées

- incapacité à gérer la fragmentation avancée dans les disques réels ;
- vitesse limitée du scan pour les supports de grande taille ;
- signature CSV insuffisante pour usages réels ;
- pas de prise en charge des systèmes de fichiers très corrompus.

4.7. Limitations générales du prototype

Le prototype n'est pas :

- un outil complet de forensic,
- une solution adaptée à une banque en production,
- capable de gérer des transactions financières réelles,
- conforme à des normes industrielles telles que ISO 27040 ou NIST 800-86.

En revanche, il démontre :

- les mécanismes fondamentaux de la récupération de données,
- la faisabilité technique du carving et du signature scan,
- une capacité réelle à récupérer des fichiers supprimés,
- une structure d'architecture prête à être étendue.

La réalisation du prototype a permis de valider les concepts étudiés tout au long du projet et de démontrer la pertinence des approches de récupération de données dans un contexte bancaire. Malgré ses limites, l'outil développé constitue une base fonctionnelle solide qui pourrait, dans le futur, être améliorée et étendue vers un système plus complet, robuste et intégré.

Le chapitre suivant présentera la conclusion générale du travail, les apports du projet, les limites identifiées et les perspectives d'évolution possibles.

CONCLUSION GENERALE

Au terme de ce travail intitulé *« Étude, Analyse et Conception d'un système de récupération des données »*, l'objectif principal était de comprendre en profondeur les mécanismes de la récupération de données, d'en analyser les enjeux dans un contexte sensible tel que le milieu bancaire, puis de proposer une solution conceptuelle et technique adaptée à un cadre académique. Ce projet a permis d'aborder une problématique actuelle et stratégique, où la perte de données peut avoir des conséquences critiques tant sur le plan opérationnel que financier.

La phase d'étude a permis de poser les bases théoriques nécessaires à la compréhension des systèmes de récupération de données. Elle a mis en évidence les causes fréquentes de perte de données, les différents types de supports concernés, ainsi que les principales techniques utilisées dans le domaine, notamment le *signature scan* et le *file carving*. L'analyse du contexte bancaire a ensuite souligné des exigences spécifiques telles que la sécurité, la traçabilité, la fiabilité des opérations et la nécessité d'estimer la récupérabilité des données avant toute tentative de restauration.

Sur la base de cette analyse, une conception structurée du système a été proposée à l'aide de modèles UML (diagrammes de cas d'utilisation, de classes, de séquence et de déploiement). Cette modélisation a permis de formaliser les interactions entre les acteurs, les composants logiciels et les données manipulées, tout en tenant compte des contraintes propres à un système bancaire.

La réalisation d'un prototype réduit a constitué une étape clé du projet. Bien que volontairement limité à quelques fonctionnalités essentielles, ce prototype a permis de mettre en œuvre concrètement les concepts étudiés. Les modules de *signature scan*, de *file carving*, d'estimation du taux de récupérabilité, ainsi que l'enregistrement des opérations et l'affichage d'un tableau de bord simplifié, ont démontré la faisabilité technique de la solution proposée.

En perspective, plusieurs pistes d'amélioration peuvent être envisagées, notamment l'extension du prototype vers la récupération de transactions bancaires, l'intégration de techniques de récupération plus avancées, l'optimisation des performances pour les supports de grande capacité, ainsi que l'ajout de mécanismes de sécurité et d'audit conformes aux normes internationales. Ces évolutions permettraient de transformer le prototype en un système plus complet et opérationnel.

En conclusion, ce projet a permis d'allier théorie et pratique, en offrant une compréhension approfondie des systèmes de récupération de données et en mettant en valeur les compétences acquises en analyse, conception et développement logiciel. Il constitue ainsi une base solide pour de futurs travaux dans les domaines de la sécurité des systèmes d'information, du *digital forensics* et de la gestion des données critiques.

RÉFÉRENCES BIBLIOGRAPHIQUES

Les références suivantes ont été utilisées pour approfondir les notions théoriques, méthodologiques et techniques liées à la récupération de données, à la sécurité des systèmes d'information et à la conception logicielle.

Ouvrages (Livres)

1. **Carrier, B.** (2005).
   File System Forensic Analysis. Addison-Wesley Professional.
   → Référence majeure sur l'analyse des systèmes de fichiers et la récupération de données.

2. **Nelson, B., Phillips, A., & Steuart, C.** (2019).
   Guide to Computer Forensics and Investigations. Cengage Learning.
   → Ouvrage de référence en forensic numérique et récupération de données.

3. **Silberschatz, A., Korth, H. F., & Sudarshan, S.** (2020).
   Database System Concepts. McGraw-Hill.
   → Base théorique solide pour comprendre la gestion et la restauration des bases de données.

4. **Stallings, W.** (2018).
   Cryptography and Network Security: Principles and Practice. Pearson.
   → Utile pour le contexte bancaire et les exigences de sécurité.

Articles et documents techniques

5. **Casey, E.** (2011).
   Digital Evidence and Computer Crime. Academic Press.
   → Explique les méthodes de récupération et l'analyse des données supprimées.

6. **Palmer, G.** (2001).
   A Road Map for Digital Forensic Research. DFRWS.
   → Cadre méthodologique pour les systèmes de récupération et d'analyse.

7. **NIST Special Publication 800-86** (2006).
   Guide to Integrating Forensic Techniques into Incident Response.
   → Norme de référence pour les systèmes de récupération et d'audit.

Normes et bonnes pratiques

8. **ISO/IEC 27001**
   Information Security Management Systems.
   → Référence internationale pour la sécurité des systèmes d'information.

9. **ISO/IEC 27040**
   Storage Security.
   → Spécifique à la sécurité et à la gestion des données stockées.

Ressources techniques et outils

12. **Scalpel Documentation**
    File Carving Tool Documentation.
    → Outil de référence pour comprendre le file carving.

13. **Autopsy & Sleuth Kit Documentation**
    → Plateformes utilisées dans le domaine de la récupération de données et du forensic.

14. **Python Software Foundation**
    Python Documentation.
    → Référence pour l'implémentation technique du prototype.

Webographie (optionnelle dans un rapport scolaire)

15. **OWASP Foundation**
    Secure Application Development.
    → Bonnes pratiques de sécurité applicative.

16. **Microsoft Learn – File Systems Overview**
    → Compréhension des systèmes de fichiers courants.