

DOCUMENT DE CONCEPTION TECHNIQUE

Projet : DigitalBank France - Plateforme de Gestion Post-Cyberattaque

Programme : ESIS/CPDIA -ESIC

Date :20/01/2026

Fait par :

NAJIMI Sanae

SAOUDI Céline

SADOUDI Abdelhamid

Kémy DIAVOU Rêve

ABID Lydia

1. INTRODUCTION ET CONTEXTE

1.1. Contexte du projet :

DigitalBank France a subi une cyberattaque majeure le 15 décembre 2025, résultant en un chiffrement complet de sa base de données contenant 850 000 comptes clients, 95 millions de transactions bancaires, données de cartes de paiement et historiques de connexions. Suite à la restauration partielle au 8 décembre 2025, l'établissement doit mettre en place une plateforme de gestion complète intégrant surveillance, détection de fraude et conformité réglementaire.

1.2. Objectifs stratégiques :

Le système doit répondre à cinq objectifs principaux :

- OBJECTIF 1 - Restauration et Sécurisation : Restaurer l'accès aux données avec authentification multi-facteurs, chiffrement AES-256 et contrôle d'accès RBAC granulaire.
- OBJECTIF 2 - Détection de Fraude Temps Réel : Système automatisé ML avec précision supérieure à 85% et latence inférieure à 500ms.
- OBJECTIF 3 - Automatisation : Réduire le temps de réponse aux incidents de 72h à moins de 15 minutes via alertes automatisées.
- OBJECTIF 4 - Conformité RGPD : Traçabilité complète avec conservation des logs pendant 3 ans minimum
- OBJECTIF 5 - Monitoring Infrastructure : Disponibilité 99.9%, temps de réponse API inférieur à 200ms.

2. ARCHITECTURE GLOBALE DU SYSTÈME

2.1. Vue d'ensemble :

L'architecture suit un modèle trois tiers modifié intégrant automatisation et monitoring. La couche présentation comprend Power BI (dashboards Analyste, Service Client) et Grafana (monitoring). La couche logique métier inclut Xano API, Make.com (workflows automatisés) et Prometheus (métriques). La couche données regroupe Xano Database PostgreSQL et stockage modèles ML.

3. COUCHE BACKEND : XANO

3.1. Choix technique

Xano sélectionné pour backend PostgreSQL complet sans infrastructure, génération automatique API REST avec Swagger, authentification JWT avec MFA intégré, RBAC natif, webhooks temps réel, interface visuelle logique métier, déploiement global CDN, plan gratuit 10 000 requêtes/mois.

3.2. Architecture de la base de donnée :

TABLE customers

- customer_id : SERIAL PRIMARY KEY, identifiant unique auto-incrémenté
- email : VARCHAR(255) UNIQUE, email unique pour login
- password_hash : VARCHAR(255), hash bcrypt mot de passe

- first_name, last_name : VARCHAR(100), identité client
- date_of_birth : DATE, date naissance
- phone : VARCHAR(20), numéro téléphone
- address, city, postal_code, country : informations postales
- created_at, last_login : TIMESTAMP, dates traçabilité
- status : VARCHAR(20), valeurs active/suspended/closed

TABLE accounts

- account_id : SERIAL PRIMARY KEY
- customer_id : INT FOREIGN KEY vers customers
- account_number : VARCHAR(34) UNIQUE, format IBAN
- account_type : VARCHAR(20), valeurs checking/savings/business
- balance : DECIMAL(15,2), solde actuel
- currency : VARCHAR(3), devise EUR par défaut
- opened_at : TIMESTAMP, date ouverture
- status : VARCHAR(20), active/frozen/closed

TABLE transactions

- transaction_id : SERIAL PRIMARY KEY
- account_id : INT FOREIGN KEY vers accounts
- transaction_type : VARCHAR(20), deposit/withdrawal/transfer/payment/fee
- amount : DECIMAL(15,2), montant négatif pour débit
- merchant_name, merchant_category : informations marchand
- location : VARCHAR(255), lieu transaction
- timestamp : TIMESTAMP, date heure exacte
- status : VARCHAR(20), pending/completed/failed/reversed
- is_fraud : BOOLEAN, marqueur fraude ML
- fraud_score : DECIMAL(3,2), score 0.00-1.00

TABLE audit_logs

- log_id : SERIAL PRIMARY KEY
- user_id, user_role : identification utilisateur
- action : VARCHAR(100), VIEW_CUSTOMER/UPDATE_ACCOUNT etc.
- table_name, record_id : ressource concernée
- ip_address : VARCHAR(45), adresse IP source
- timestamp : TIMESTAMP, horodatage précis

3.3. API Layer et endpoints :

Authentification :

- POST /auth/signup : Création compte utilisateur
- POST /auth/login : Connexion retourne JWT
- POST /auth/logout : Déconnexion
- POST /auth/refresh : Rafraîchir token JWT
- POST /auth/mfa/enable : Activer MFA TOTP
- POST /auth/mfa/verify : Vérifier code MFA

Gestion de données :

- GET /customers : Liste clients paginée filtrée
- GET /customers/{id} : Détails client
- GET /accounts : Liste comptes
- GET /accounts/{id}/transactions : Transactions compte
- POST /transactions : Créer transaction
- PATCH /transactions/{id} : Modifier transaction admin only

Détection de Fraude :

- POST /fraud/predict : Prédire fraude, body amount/category/location/hour, réponse is_fraud(score/risk_level)

Analytics :

- GET /analytics/fraud-stats : Statistiques globales fraude
- GET /analytics/top-fraud : Top 10 transactions suspectes
- GET /analytics/by-category : Répartition par catégorie

Audit :

- GET /audit/logs : Logs audit admin only
- GET /audit/user/{id} : Logs utilisateur spécifique

3.4. Sécurité :

- Authentification JWT : Token accès 15 minutes, refresh token 7 jours, signature HS256 clé 256 bits, payload user_id/email/role/exp/iat.
- MFA : TOTP période 30 secondes, applications Google Authenticator ou Authy, 10 codes backup générés activation.
- Rate Limiting : 100 requêtes/minute par IP, 1000 requêtes/heure par utilisateur authentifié, blocage temporaire 15 minutes si dépassement.
- Chiffrement : Données transit TLS 1.3, données repos AES-256-GCM, password_hash bcrypt cost 12, card_number et cvv chiffrés AES-256-GCM, rotation clés tous les 90 jours.

4. COUCHE PRÉSENTATION : POWER BI

4.1. Architecture des dashboards :

DASHBOARD 1 - ANALYSTE SÉCURITÉ DÉTECTION FRAUDE

- **Objectif :** Surveillance temps réel et analyse patterns fraude.
- **Composants KPIs :** Total Clients, Transactions Jour, Fraudes Déetectées, Taux Fraude pourcentage. Affichage cards en en-tête.
- Carte Géographique : Visualisation transactions par localisation, bulles proportionnelles nombre transactions, couleur gradient vert à rouge selon taux fraude.

DASHBOARD 2 - SERVICE CLIENT GESTION CLIENTS

- **Objectif :** Recherche consultation rapide informations client.
- **Barre Recherche :** Input email client avec bouton rechercher, saisie semi-automatique.
- **Profil Client :** Affichage nom complet, email, téléphone, ville code postal, statut compte actif/suspendu/fermé, date membre depuis.
- **Statistiques Client :** Cards nombre comptes, solde total agrégé, transactions par mois moyenne, nombre fraudes détectées.
- Comptes Associés : Liste comptes avec numéro IBAN, type compte courant/épargne/professionnel, solde actuel, statut.

DASHBOARD 3 - MONITORING INFRASTRUCTURE

- **Objectif :** Surveillance santé système performances.
- Métriques Santé Système : Cards Uptime pourcentage cible 99.9, CPU pourcentage utilisation, RAM gigabytes utilisés sur total, Disk mégabytes utilisés sur total.
- **Graphique CPU Usage :** Ligne temps réel refresh 1 minute, seuil alerte 80 pourcentage ligne rouge horizontale.

4.2. Modèle de données

Modèle étoile star schema pour optimisation performances analytiques.

- Tables faits : fact_transactions table centrale toutes transactions, fact_logins tentatives connexion, fact_api_calls logs appels API monitoring.
- Tables de dimensions: dim_customers dimension client SCD Type 1, dim_accounts dimension comptes, dim_time dimension temporelle date/heure/jour semaine/mois/trimestre, dim_location dimension géographique pays/ville, dim_merchant_category dimension catégories marchands.

Relations : fact_transactions account_id vers dim_accounts account_id Many-to-One, dim_accounts customer_id vers dim_customers customer_id Many-to-One, fact_transactions timestamp vers dim_time datetime Many-to-One.

5. AUTOMATISATION : MAKE.COM

5.1. Workflows critiques :

WORKFLOW 1 - DÉTECTION FRAUDE TEMPS RÉEL

Déclencheur : Webhook Xano nouvelle transaction créée.

- Étape 1 : Réception données transaction input
- Étape 2 : Appel HTTP API Flask ML POST /predict avec body
- Étape 3 : Router évaluation score fraude, Route A score supérieur ou égal 0.8 fraude critique, Route B 0.6 inférieur ou égal score inférieur 0.8 fraude probable, Route C score inférieur 0.6 transaction normale.
- Étape 4a Route A : Email alerte critique destinataire security@digitalbank.fr sujet ALERTE Fraude critique avec détails score/montant/marchand/localisation action bloquer carte immédiatement.
- Étape 4b : Update Xano API PATCH /transactions avec fraud_score et is_fraud.
- Étape 4c : Notification Slack ou Discord webhook message transaction suspecte détectée avec score.
- Étape 5 : Log audit POST /audit/log action FRAUD_DETECTION.

WORKFLOW 2 - ALERTE TRANSACTION ÉLEVÉE

Déclencheur : Webhook Xano transaction créée montant supérieur 5000 euros.

- Étape 1 : Réception transaction.
- Étape 2 : Filter vérifier montant supérieur 5000.
- Étape 3 : Récupération infos client GET /customers.
- Étape 4 : Email notification client avec message transaction importante détectée montant/marchand/date, si non origine contacter immédiatement numéro urgence.
- Étape 5 : Enregistrement datastore table high_value_transactions.

WORKFLOW 3 - ALERTE ÉCHECS CONNEXION RÉPÉTÉS

Déclencheur : CRON toutes les 5 minutes.

- Étape 1 : Requête Xano GET /auth/failed-attempts depuis 5 minutes.
- Étape 2 : Aggregator grouper par IP address.
- Étape 3 : Filter garder seulement IP avec supérieur ou égal 5 échecs.
- Étape 4 : Iterator pour chaque IP suspecte bloquer temporairement POST /security/block-ip durée 3600 secondes, alerte Slack channel security-alerts message attaque force brute IP/nombre tentatives/action bloquée 1h.
- Étape 5 : Log incident datastore table security_incidents.

WORKFLOW 4 - RAPPORT QUOTIDIEN AUTOMATIQUE

Déclencheur : CRON tous les jours 8h00 GMT+1.

- Étape 1 : Récupération statistiques GET /analytics/daily-stats.
- Étape 2 : Récupération top fraudes GET /analytics/top-fraud période 24h.
- Étape 3 : Text Aggregator créer rapport formaté avec date, statistiques transactions totales/frauduleuses/taux/montant, top 5 marchands risque, top 5 localisations risque.

- Étape 4 : Email envoi rapport destinataires management/security/analyst avec pièce jointe CSV stats détaillées.
- Étape 5 : Google Drive archiver rapport dossier /DigitalBank/Rapports Quotidiens/2026/ nom fichier Rapport_YYYYMMDD.txt.

5.2. Intégration Connecteurs :

Util	Utilisation
Xano	CRUD transactions, customers, analytics endpoints
Email SMTP	Alertes, notifications, rapports
Slack	Notifications équipe temps réel
Discord	Alternative Slack équipe développement
Google Drive	Archivage rapports et logs
Google Sheets	Export statistiques analyse manuelle
Datastore	Tables temporaires agrégations

5.3. Gestion des erreurs

- **Niveau 1 Retry Automatique** : 3 tentatives maximum, délai exponentiel 1s/2s/4s, applicable erreurs réseau timeouts 5xx.
- **Niveau 2 Routes Alternatives** : API principale échoue basculement API backup, email échoue notification Slack, webhook échoue polling API fallback.
- **Niveau 3 Logging Alertes** : Chaque erreur loggée datastore error_logs, si supérieur 10 erreurs par heure alerte admin, dashboard dédié Make.com monitoring.
- **Niveau 4 Circuit Breaker** : Service échoue 5 fois consécutives désactivation temporaire 10 minutes, évite surcharge effet domino, réactivation automatique après cooldown.

6. MONITORING : PROMETHEUS

6.1. Architecture de surveillance

Stack complet Prometheus pour collection métriques stockage TSDB rétention 15 jours scrape interval 15 secondes, Grafana pour dashboards visualisation, Alertmanager pour routage notifications.

Exporteurs : Node Exporter métriques serveur CPU/RAM/Disk/Network, Custom Exporter métriques application API calls/fraud score/transactions par seconde/error rate.

6.2. MÉTRIQUES COLLECTÉES

INFRASTRUCTURE

- node_cpu_seconds_total
- node_memory_MemAvailable_bytes
- node_disk_io_time_seconds_total
- node_network_receive_bytes_total :

6.3. Alertes Notifications :

Configuration Alertmanager : SMTP smarthost smtp.gmail.com:587, groupe alertes par alertname/cluster/service, délai groupe 10 secondes, intervalle répétition 12 heures, routes conditionnelles severity critical vers pagerduty, severity warning vers slack-warnings.

7. SÉCURITÉ ET CONFORMITÉ

7.1. Chiffrement protection de données

Chiffrement Transit TLS 1.3 , Chiffrement Repos AES-256

Champs chiffrés : customers.password_hash bcrypt cost 12, cards.card_number AES-256-GCM, cards.csv AES-256-GCM, transactions.merchant_name pseudonymisation si analytique.

Gestion clés : Clé maître stockée HashiCorp Vault recommandé, rotation tous les 90 jours automatiquement, dérivation PBKDF2 avec 100 000 itérations, backup clés chiffré clé différente stockage off-site.

8. DÉTECTION DE FRAUDE PAR ML

8.1. Architecture ML

- **Pipeline complet** : Extraction données Xano Database historique transactions, labellisation is_fraud TRUE FALSE.
- **Feature Engineering** : Features numériques amount_abs valeur absolue, hour 0-23, day_of_week 0-6, is_weekend 0/1. Features catégorielles merchant_category location transaction_type Label Encoded. Features dérivées is_high_risk_category 0/1, is_high_risk_location 0/1, is_online 0/1, time_period morning/afternoon/evening/night.
- **Model Training** : Algorithmes testés Random Forest baseline n_estimators 100 max_depth 10 AUC-ROC 0.87, XGBoost champion n_estimators 200 max_depth 8 learning_rate 0.1 AUC-ROC 0.92, LightGBM challenger n_estimators 150 num_leaves 31 AUC-ROC 0.90. Validation 5-fold Cross-Validation, métrique AUC-ROC classe déséquilibrée.
- **Model Evaluation** : Métriques test set Precision 0.88, Recall 0.85, F1-Score 0.86, AUC-ROC 0.92. Matrice confusion True Negatives 18, False Positives 2, False Negatives 1, True Positives 9. Feature Importance amount_abs 0.35, is_high_risk_location 0.22, merchant_category_encoded 0.18, is_high_risk_category 0.12, hour 0.08.
- **Model Deployment** : Flask API endpoint POST /predict, input amount/category/location/hour, output is_fraud/fraud_score/risk_level, déploiement Render.com free tier URL <https://digitalbank-fraud-api.onrender.com>, artifacts exportés fraud_detection_model.pkl, label encoders, feature_columns.json.

8.2. Pipeline de prediction :

- Étape 1 Réception Transaction : Input amount -2500.00, merchant_category Electronics, location Dubai UAE, timestamp 2026-01-20T14:30:00Z.
- Étape 2 Feature Extraction : Extraction amount_abs 2500, hour 14, day_of_week calcul, is_weekend 0 ou 1, encodages catégoriels, flags high_risk_category et high_risk_location, is_online détection.
- Étape 3 Prédiction : Array features vers modèle, predict_proba retourne probabilité classe 1 fraude, is_fraud booléen si probabilité supérieur ou égal 0.6, risk_level CRITICAL si supérieur ou égal 0.8, HIGH si supérieur ou égal 0.6, MEDIUM si supérieur ou égal 0.4, LOW sinon.

- Étape 4 Réponse : Output is_fraud true, fraud_score 0.87, risk_level CRITICAL, recommendation BLOCK, timestamp 2026-01-20T14:30:02Z.

Latence cible inférieur 200ms percentile 95, disponibilité 99.9 pourcentage.

8.3. Intégration de MAKE.COM

Workflow Make.com consomme API ML : Webhook Xano nouvelle transaction, Make.com appelle POST /predict avec features, réception score fraude, routage conditionnel score supérieur ou égal 0.8 alerte critique blocage carte, 0.6 inférieur ou égal score inférieur 0.8 investigation manuelle, score inférieur 0.6 transaction approuvée, update transaction fraud_score Xano, log audit_logs.

Gestion erreurs : Timeout API 10s transaction marquée pending_review, API indisponible fallback règles heuristiques simples, retry 3x backoff exponentiel.

9. PLAN DE DÉPLOIEMENT ET MAINTENANCE

9.1. Procédure de déploiement

- Phase 1 Préparation J-1 : Validation tests unitaires intégration, review code par pair, backup complet base données, notification équipes ops/support/management, vérification monitoring alertes.
- Phase 2 Backend: Activer mode maintenance Xano API, exécuter scripts migration base données, déployer nouvelles API functions, tester endpoints critiques, vérifier RLS policies, désactiver mode maintenance.
- Phase 3 Dashboards: Publier rapports Power BI workspace prod, configurer refresh schedules, tester connexions datasources, assigner permissions groupes AD, validation UAT utilisateurs pilotes.
- Phase 4 Automatisation: Activer scenarios Make.com un par un, tester déclencheurs webhooks, vérifier logs exécution, valider alertes email Slack, monitoring erreurs 1ère heure.
- Phase 5 Monitoring: Démarrer Prometheus Grafana, configurer scrape targets, importer dashboards Grafana, tester alertes Alertmanager, baseline metrics CPU RAM latency.
- Phase 6 Validation: Tests end-to-end complets, vérification KPIs métier, monitoring dashboards 2h, réunion débrief équipe.

9.2 Rollback Disaster recovery :

Plan Rollback : Trigger taux erreur supérieur 10 pourcentage pendant 5 minutes, latence API p95 supérieur 2 secondes, données corrompues détectées, bug critique fonctionnel.

Disaster Recovery : RTO Recovery Time Objective 4 heures, RPO Recovery Point Objective 1 heure.