DigitalBank France

Documentation des Rôles et Permissions

Date: 21/01/2026

1. Vue d'ensemble du Système RBAC

Le système DigitalBank France implémente un contrôle d'accès basé sur les rôles (RBAC - Role-Based Access Control) avec quatre niveaux de permissions distincts. Cette architecture garantit la sécurité des données bancaires sensibles tout en permettant une collaboration efficace entre les différents acteurs de l'organisation.

2. Définition des Rôles

Rôle	Description	Cas d'usage
Admin	Accès complet en lecture et écriture sur toutes les tables. Peut gérer les utilisateurs, modifier toutes les données, et accéder aux logs d'audit.	Direction IT, responsables sécurité, administrateurs système
Analyst	Lecture globale de toutes les transactions, comptes et tentatives de connexion. Aucune modification possible.	Data scientists, équipe anti-fraude, compliance, business intelligence
Customer Service	Lecture globale des clients, comptes et transactions. Modification limitée aux statuts (bloquer une carte, geler un compte).	Support client, service après-vente, gestionnaires de comptes
Client	Lecture uniquement de ses propres données (comptes, transactions, cartes). Aucune modification.	Clients finaux, application mobile, portail web client

3. Matrice des Permissions par Table

	Admin	Analyst	Customer Service	Client
customers	CRUD	—	R + U (statut)	R (soi-même)
accounts	CRUD	R	R + U (statut)	R (ses comptes)
transactions	CRUD	R	R + U (statut)	R (ses comptes)
cards	CRUD	—	R + U (bloquer)	R (ses cartes)
login_attempts	CRUD	R	—	—
audit_logs	CRUD	—	—	—

Légende: CRUD = Create, Read, Update, Delete | R = Read only | U = Update | — = Aucun accès

4. Implémentation Technique

4.1 Row Level Security (RLS)

La sécurité est implémentée au niveau de la base de données PostgreSQL via Row Level Security (RLS). Chaque table possède des policies qui vérifient automatiquement l'identité et le rôle de l'utilisateur lors de chaque requête. Cela garantit que même en cas de faille applicative, les données restent protégées.

```
-- Exemple de policy pour les clients:
CREATE POLICY transactions_client ON transactions
FOR SELECT TO authenticated
USING (
EXISTS (
SELECT 1 FROM accounts a
JOIN customers c ON a.customer_id = c.customer_id
WHERE transactions.account_id = a.account_id
AND c.email = auth.email()
)
);
```

4.2 Authentification et MFA

L'authentification est gérée par Supabase Auth avec les mécanismes suivants :

- Authentification par email/mot de passe avec hachage bcrypt
- Tokens JWT signés (HS256) avec durée de validité de 1 heure
- Multi-Factor Authentication (MFA) via TOTP (Time-based One-Time Password)
- Gestion automatique du refresh token pour maintenir la session
- Logging de toutes les tentatives de connexion dans login_attempts

4.3 Table de Gestion des Rôles

La table **profiles** fait le lien entre les utilisateurs Supabase Auth (auth.users) et leurs rôles :

```
CREATE TABLE profiles (
id UUID PRIMARY KEY REFERENCES auth.users(id),
role TEXT CHECK (role IN ('admin', 'analyst', 'customer_service', 'client')),
created_at TIMESTAMP DEFAULT NOW()
);
```

5. Scénarios de Test

Test Admin: Vérifier l'accès complet: doit voir les 30 transactions, tous les 10 clients, pouvoir modifier n'importe quelle donnée.

Test Analyst: Vérifier la lecture globale sans modification: doit voir toutes les transactions mais recevoir une erreur 403 en tentant un UPDATE.

Test Customer Service: Vérifier lecture + modification limitée: peut voir tous les clients/comptes, peut modifier le statut d'une carte (blocked), mais ne peut pas modifier un solde.

Test Client: Vérifier l'isolation des données: jean.dupont@email.fr doit voir uniquement ses 2 comptes et ~3-5 transactions, pas celles des autres clients.

6. Conformité RGPD et Audit

Le système respecte les principes du RGPD grâce à :

- **Minimisation des données** : Chaque rôle n'a accès qu'aux données strictement nécessaires
- **Intégrité et confidentialité** : RLS empêche tout accès non autorisé, même en cas d'erreur applicative
- **Transparence** : Tous les accès sont loggés dans audit_logs (qui a accédé à quoi, quand, depuis quelle IP)
- **Droit d'accès** : Les clients peuvent consulter toutes leurs données via l'API
- **Sécurité dès la conception** : Les policies RLS sont activées par défaut, pas de mode 'permissif'

7. Conclusion

Ce système RBAC à 4 niveaux garantit une séparation stricte des responsabilités tout en maintenant une flexibilité opérationnelle. Les policies RLS PostgreSQL assurent que la sécurité est appliquée au niveau de la base de données, rendant toute tentative de contournement impossible même en cas de faille applicative.

Document confidentiel - DigitalBank France

Version: 1.0 | Date: 21/01/2026

Contact: security@digitalbank.fr