RBAC Permissions - Multi-Tenant Multi-Company Architecture

Architecture Overview

Ce document décrit le système de permissions RBAC pour l'architecture multi-tenant, multi-company du système payroll-saas.

Structure des Permissions

Format : `resource.action.scope`

- **resource** : L'entité concernée (company, bankAccount, user, contract, etc.)
- **action** : L'opération (create, read, update, delete, list, approve, etc.)
- **scope** : Le périmètre d'application
- `global` : Accès à toutes les ressources du tenant (Platform Admin)
- `ownCompany` : Accès aux ressources de sa company (Agency Admin)
- `own` : Accès à ses propres ressources ou celles créées (Agency User, Contractor)
- `parent` : Accès aux ressources créées par soi (hiérarchie parent-enfant)

Permissions par Rôle

1. Platform Admin

Companies
- `company.list.global` - Voir toutes les companies (tenant + agency)
- `company.create.global` - Créer n'importe quel type de company
- `company.update.global` - Modifier n'importe quelle company
- `company.delete.global` - Supprimer n'importe quelle company
- `company.*.global` - Toutes les opérations sur les companies

Bank Accounts
- `bank.list.global` - Voir tous les bank accounts
- `bank.create.global` - Créer n'importe quel bank account
- `bank.update.global` - Modifier n'importe quel bank account
- `bank.delete.global` - Supprimer n'importe quel bank account
- `bankAccount.*.global` - Toutes les opérations sur les bank accounts

Users
- `user.list.global` - Voir tous les users du tenant
- `user.create.global` - Créer n'importe quel user
- `user.update.global` - Modifier n'importe quel user
- `user.delete.global` - Supprimer n'importe quel user

Contracts
- `contract.list.global` - Voir tous les contrats

- `contract.create.global` - Créer n'importe quel contrat
- `contract.update.global` - Modifier n'importe quel contrat
- `contract.delete.global` - Supprimer n'importe quel contrat
- `contract.approve.global` - **Approuver tous les MSA/SOW** ⭐

MSA (Master Service Agreement)
- `contract_msa.list.global` - Voir tous les MSA
- `contract_msa.create.global` - Créer des MSA
- `contract_msa.update.global` - Modifier des MSA
- `contract_msa.delete.global` - Supprimer des MSA

---

2. Agency Admin

Own Company Management
- `company.create.own` - Créer sa propre agency company
- `company.update.own` - Modifier sa propre agency company
- `company.read.own` - Voir sa propre agency company
- `company.list.own` - Voir les tenant companies (version simplifiée)

Own Bank Account Management
- `bankAccount.create.own` - Créer le bank account de sa company
- `bankAccount.update.own` - Modifier le bank account de sa company
- `bankAccount.read.own` - Voir le bank account de sa company

Company Users
- `user.list.ownCompany` - Voir tous les users de sa company
- `user.create.ownCompany` - Créer des users dans sa company (héritage automatique de companyId)
- `user.update.ownCompany` - Modifier les users de sa company

Contracts
- `contract.list.own` - Voir ses propres contrats
- `contract.create.own` - Créer des contrats (SOW uniquement, MSA si permission spéciale)
- `contract.update.own` - Modifier ses propres contrats (draft uniquement)
- `contract.read.own` - Voir les détails de ses contrats

Visibility
- Peut voir les **Tenant Companies** (liste simplifiée, sans détails sensibles)
- Ne peut **pas** voir les autres Agency Companies
- Ne peut **pas** voir les Platform Admins (sauf dans la liste des approvers assignés automatiquement)

---

3. Agency User (Employee)

Read-Only Company
- `company.read.own` - Voir la company de l'agency (read-only)

Read-Only Bank Account
- `bankAccount.read.own` - Voir le bank account de la company (read-only)

Limited Users
- `user.read.own` - Voir son propre profil
- `user.list.parent` - Voir les users créés par lui (si applicable)

Contracts
- `contract.read.own` - Voir les contrats auxquels il participe
- `contract.list.own` - Lister ses contrats

---

4. Contractor (Individual)

No Company
- Pas de company associée (companyId = null)
- Affiché comme “User Name (Individual Contractor)” dans les contrats

Limited Access
- `user.read.own` - Voir son propre profil
- `contract.read.own` - Voir ses contrats
- `invoice.read.own` - Voir ses factures
- `timesheet.*.own` - Gérer ses timesheets

Workflows Spéciaux

Assignation Automatique des Approvers (MSA)

Trigger : Quand une Agency envoie un MSA (passage en `pending_platform_review`)

Logique :
1. Le backend cherche un Platform Admin avec la permission `contract.approve.global`
2. Sélectionne le plus ancien (critère : `createdAt ASC`)
3. Crée automatiquement un `ContractParticipant` avec :
- `role: "approver"`
- `approved: false`
- `requiresSignature: false`

Résultat :
- L’Agency Admin ne sélectionne **pas** l’approver manuellement
- Un message est affiché : “Un approver de la plateforme sera automatiquement assigné”
- Le contrat passe en état `pending_approval`

Héritage Automatique

Company Inheritance (User)

Quand un user crée un autre user :
- Le nouveau user hérite du `companyId` du créateur
- Les bank accounts sont partagés via la company
- Permet une gestion cohérente des ressources d’entreprise

Exemple :

```
// Agency Admin crée un Agency User
AgencyAdmin (companyId: "comp_123")
  └── creates AgencyUser (companyId: "comp_123" - hérité automatiquement)
```

Parent-Child Relationship

La relation `createdBy` (parentUserId) permet de tracer la hiérarchie :

```
PlatformAdmin
  └── AgencyAdmin (companyId: "comp_agency_1")
        └── AgencyUser1 (companyId: "comp_agency_1")
        └── AgencyUser2 (companyId: "comp_agency_1")
```

Visibilité des Données

Tenant Companies vs Agency Companies

Entité	Platform Admin	Agency Admin	Agency User
Tenant Companies (type: “tenant”)	✅ Tous les détails	✅ Liste simplifiée (nom uniquement)	❌ Non visible
Agency Companies (type: “agency”)	✅ Toutes	✅ Seulement la sienne	✅ Seulement la sienne (read-only)
Other Agency Companies	✅ Toutes	❌ Non visible	❌ Non visible

Users Visibility

Basé sur le helper `getUsersVisibleFor(user, scope)` :

Scope	Description	Utilisé par
`global`	Tous les users du tenant	Platform Admin
`ownCompany`	Tous les users de la même company	Agency Admin
`parent`	Seulement les users enfants (createdBy)	Agency User, Contractor

---

Implementation Notes

Backend Helpers

- `getTenantCompanies()` - Récupère les companies de type “tenant”

- `getAgencyCompanies()` - Récupère les companies de type “agency”
- `getUserCompany()` - Récupère la company d’un user
- `getUsersVisibleFor()` - Récupère les users visibles selon le scope RBAC
- `getParticipantDisplayName()` - Formate l’affichage Company vs Individual
- `assignPlatformApprover()` - Assigne automatiquement un approver pour MSA

Database Changes

Migration: `20251126090649_add_company_type_and_user_company_relation`

1. `companies.type` - “tenant” | “agency”
2. `users.companyId` - Relation directe avec Company
3. Indexes pour performance

Testing Scenarios

Scenario 1: Agency Admin Creates Company

1. Agency Admin se connecte
2. Appelle `company.createMyCompany`
3. Vérifie que `type = "agency"`
4. Vérifie que `user.companyId` est défini

Scenario 2: Agency Admin Creates Bank Account

1. Agency Admin a une company
2. Appelle `bank.setMyCompanyBank`
3. Vérifie que `company.bankId` est défini
4. Tous les users de la company peuvent voir le bank account

Scenario 3: MSA Auto-Approver Assignment

1. Agency Admin crée un MSA
2. Envoie le MSA (upload main document)
3. Vérifie qu’un Platform Admin est automatiquement assigné comme approver
4. Vérifie que `contractParticipant.role = "approver"` et `approved = false`

Scenario 4: User Hierarchy and Company Inheritance

1. Agency Admin crée un Agency User
2. Vérifie que Agency User a le même `companyId`
3. Vérifie que Agency User peut voir la company (read-only)
4. Vérifie que les deux users partagent les bank accounts

Security Considerations

1. **Agency Isolation** : Les agencies ne peuvent pas voir les données des autres agencies
2. **Sensitive Data Protection** : Les Tenant Companies sont affichées en version simplifiée pour les agencies
3. **Auto-Assignment Safety** : Les approvers sont assignés uniquement pour les MSA, pas les SOW

4. **Read-Only Enforcement** : Les Agency Users ont un accès read-only sur company et bank accounts

---

Migration Guide

Pour activer ces permissions sur un tenant existant :

```
-- Exécuter le script seed-permissions.sql
\i prisma/migrations/20251126090649_add_company_type_and_user_company_relation/seed-permissions.sql

-- Assigner les permissions aux rôles existants
-- Exemple : Assigner company.create.own au rôle Agency Admin
INSERT INTO role_permissions (roleId, permissionId)
SELECT
  r.id AS roleId,
  p.id AS permissionId
FROM roles r
CROSS JOIN permissions p
WHERE r.name = 'AGENCY_ADMIN'
  AND p.key IN ('company.create.own', 'company.update.own', 'company.read.own', 'bankAccount.create.own', 'bankAccount.update.own', 'bankAccount.read.own')
ON CONFLICT DO NOTHING;
```

🆕 NEW: Agency Portal & Payroll Partner Portal (Phase 4)

Agency Portal Permissions

Contractor Visibility

- `contractor.list.ownCompany` - Voir tous les contractors liés à l'agency
- `contractor.view.ownCompany` - Voir les détails d'un contractor
- `contractor.view_onboarding.ownCompany` - Voir le statut d'onboarding
- `contractor.view_dates.ownCompany` - Voir les dates de début/fin
- `contractor.view_payments.ownCompany` - Voir l'historique des paiements

Document Management

- `document.upload_proof_of_payment.own` - Uploader proof of payment
- `document.upload_selfbill.own` - Uploader self-bill
- `document.upload_kyc.ownCompany` - Uploader KYC documents (future)

Access via:

- `AGENCY_ADMIN` : Toutes les permissions ownCompany
- `AGENCY_USER` : Permissions view seulement

Payroll Partner Portal Permissions

Worker Management
- `worker.list.ownCompany` - Voir tous les workers gérés
- `worker.view.ownCompany` - Voir les détails d'un worker
- `worker.view_onboarding.ownCompany` - Voir le statut d'onboarding
- `worker.view_dates.ownCompany` - Voir les dates de début/fin
- `worker.view_contract.ownCompany` - Voir le contrat local d'emploi

Payslip Management
- `payslip.upload.ownCompany` - Uploader payslip pour un worker
- `payslip.view.ownCompany` - Voir les payslips des workers

Invoice Management
- `invoice.upload_to_platform.ownCompany` - Uploader invoice vers Aspirock

Access via:
- `PAYROLL_PARTNER_ADMIN` : Toutes les permissions ownCompany
- `PAYROLL_PARTNER_USER` : Permissions view seulement

🆕 NEW: Reporting System (Phase 4)

Report Permissions
- `report.view_margin.global` - Voir le margin report (profit brut)
- `report.view_live_contractors.global` - Voir les contractors actifs
- `report.view_by_country.global` - Voir la répartition par pays
- `report.view_by_client.global` - Voir la répartition par client
- `report.view_income.global` - Voir les revenus
- `report.export.global` - Exporter les rapports (CSV/PDF/Excel)
- `report.view.ownCompany` - Voir rapports limités à sa company (future)

Available Reports:
1. **Margin Report**: Calcule le profit brut (fees) par période, compté uniquement quand le worker est payé
2. **Live Contractors**: Nombre de contractors actifs avec répartition par type (gross/employed/split)
3. **Contracts by Country**: Distribution géographique des contrats
4. **Contractors by Client**: Nombre de contractors par client/agency
5. **Income by Country**: Revenus par pays par période
6. **Dashboard Summary**: Vue d'ensemble avec métriques clés

Access via:
- `PLATFORM_ADMIN` : Toutes les permissions global
- `FINANCE_MANAGER` : Toutes les permissions global (custom role)

🆕 NEW: Enhanced Payment & Remittance (Phase 4)

Payment Management
- `payment.approve.global` - Approuver un paiement (passage à "processing")

- `payment.execute.global` - Exécuter un paiement (passage à “completed”)
- `payment.cancel.global` - Annuler un paiement pending

Remittance Management
- `remittance.generate.global` - Générer remittance advice
- `remittance.send.global` - Envoyer remittance au worker

Workflow:
1. Payment créé → status: “pending”
2. Finance Manager approve → status: “processing”
3. Finance Manager execute → status: “completed” + auto-création remittance
4. Remittance envoyée au worker avec détails

Access via:
- `PLATFORM_ADMIN` : Toutes les permissions
- `FINANCE_MANAGER` : approve, execute, generate, send (custom role)

🆕 NEW: Enhanced Document Management (Phase 4)

Granular Document Permissions
- `document.upload.own` - Uploader document pour soi
- `document.upload.ownCompany` - Uploader document pour sa company
- `document.upload_selfbill.own` - Uploader self-bill (Agency)
- `document.upload_proof_of_payment.own` - Uploader proof of payment (Agency)
- `document.upload_kyc.ownCompany` - Uploader KYC documents (future)
- `document.view.own` - Voir ses propres documents
- `document.view.ownCompany` - Voir documents de sa company
- `document.download.own` - Télécharger ses documents
- `document.download.ownCompany` - Télécharger documents de sa company

🆕 NEW: Suggested Roles to Create

PAYROLL_PARTNER_ADMIN

```
worker.list.ownCompany
worker.view.ownCompany
worker.view_onboarding.ownCompany
worker.view_dates.ownCompany
worker.view_contract.ownCompany
payslip.upload.ownCompany
payslip.view.ownCompany
invoice.upload_to_platform.ownCompany
user.create.ownCompany (créer users payroll partner)
```

FINANCE_MANAGER

```
payment.approve.global
payment.execute.global
payment.cancel.global
remittance.generate.global
remittance.send.global
report.view_margin.global
report.view_income.global
report.export.global
```

SALES_MANAGER

```
lead.list.global
lead.create.global
lead.update.global
lead.assign.global
lead.export.global
contractor.list.global (pour voir pipeline)
```

Future Enhancements

1. **Multi-Level Approval** : Support pour plusieurs niveaux d’approbation (approver1, approver2, etc.)
2. **Company Groups** : Regroupement de companies pour les grandes entreprises
3. **Custom Scopes** : Ajout de scopes personnalisés (team, department, etc.)
4. **Permission Templates** : Templates de permissions par industrie/use-case
5. **Agency Self-Service** : Agencies peuvent ajouter new joiners via SOW
6. **KYC Document Management** : Upload et validation automatique de documents KYC