

Analyse Complète de la Structure RBAC Actuelle

Date: 17 Novembre 2025

Repository: <https://github.com/StrealyX/payroll-saas/tree/dev>

Objectif: Refactorisation complète de l'architecture RBAC



Résumé Exécutif

Problèmes Identifiés

1. Structure basée sur les rôles plutôt que sur les fonctionnalités

- Dossiers nommés d'après les rôles : `/contractor`, `/agency`, `/payroll-partner`
- Rend l'architecture rigide et difficile à maintenir

2. Permissions trop larges et ambiguës

- `contractors.view` utilisée pour DEUX contextes différents :
 - Contractor voit SES propres informations → `/contractor/information`
 - Admin voit TOUS les contractors → `/contractors`
 - Pas de séparation claire entre permissions de visualisation et d'action

3. Imbrication excessive de sous-dossiers

- Structure complexe et difficile à naviguer
- Trop de niveaux de profondeur

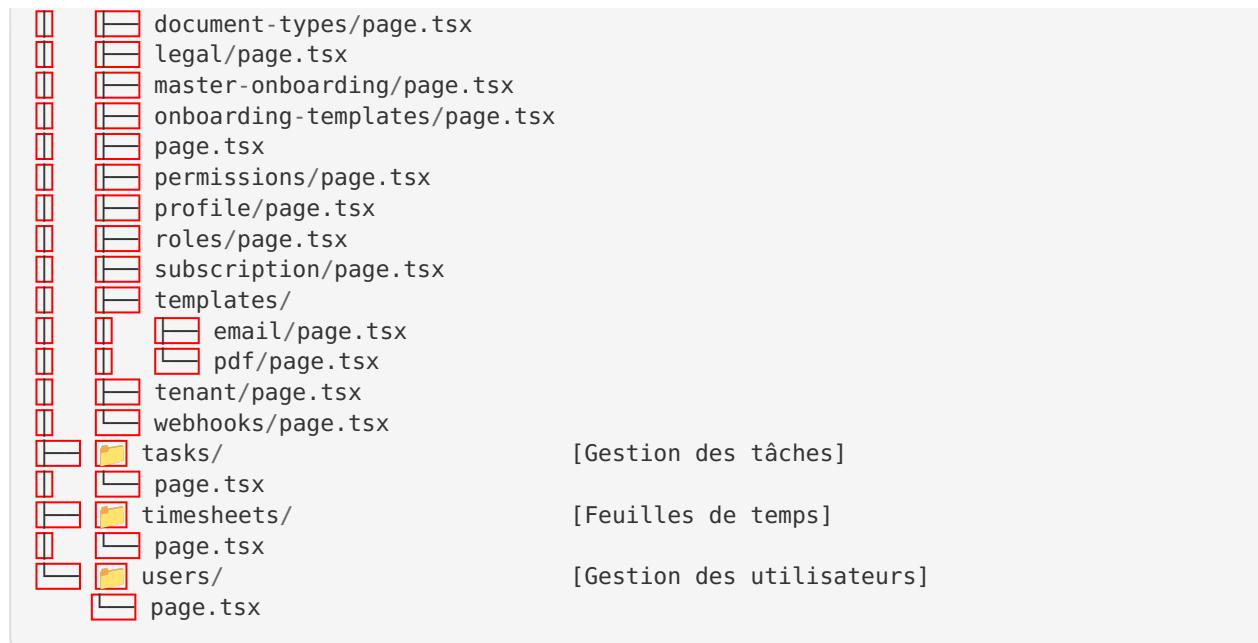
4. Système RBAC partiellement implémenté

- Progrès : 39% selon `RBAC_PROGRESS.md`
- Permissions existent mais vérification incohérente
- Pas de guards/middleware uniformes

Structure Actuelle des Dossiers

Hiérarchie Complète (app/(dashboard)/(modules)/)

app/(dashboard)/(modules)/	
└── agencies/	[Admin: Gérer les agences]
└── page.tsx	
└── agency/	[Agency: Portail agence]
└── contracts/page.tsx	
└── information/page.tsx	
└── invoices/page.tsx	
└── page.tsx	
└── roles/page.tsx	
└── settings/page.tsx	
└── users/page.tsx	
└── analytics/	[Analyses et rapports]
└── page.tsx	
└── contractor/	[Contractor: Portail personnel]
└── information/page.tsx	
└── invoices/page.tsx	
└── onboarding/page.tsx	
└── page.tsx	
└── payslips/page.tsx	
└── refer/page.tsx	
└── remits/page.tsx	
└── time-expenses/page.tsx	
└── contractors/	[Admin: Gérer les contractors]
└── page.tsx	
└── contracts/	⚠️ BUG: utilise contractors.view
└── page.tsx	
└── expenses/	[Gestion des contrats]
└── page.tsx	
└── invoices/	[Gestion des dépenses]
└── agency/page.tsx	
└── contractor/page.tsx	
└── payroll-partner/page.tsx	
└── leads/	[Gestion des factures]
└── page.tsx	
└── onboarding/	[Gestion des prospects]
└── page.tsx	
└── payroll-partner/	[Processus d'onboarding]
└── contracts/page.tsx	
└── information/page.tsx	
└── invoices/page.tsx	
└── page.tsx	
└── payslips/page.tsx	
└── remits/page.tsx	
└── roles/page.tsx	
└── settings/page.tsx	
└── users/page.tsx	
└── payroll-partners/	[Payroll Partner: Portail]
└── page.tsx	
└── payslips/	
└── page.tsx	
└── reports/	
└── activity-logs/page.tsx	
└── email-logs/page.tsx	
└── page.tsx	
└── sms-logs/page.tsx	
└── user-activity/page.tsx	
└── settings/	[Admin: Gérer les payroll partners]
└── banks/page.tsx	
└── branding/login/page.tsx	
└── companies/page.tsx	
└── countries/page.tsx	
└── currencies/page.tsx	
	[Gestion des bulletins de paie]
	[Rapports système]
	[Paramètres système]



Total: 62 pages

👥 Rôles Existantes et Permissions

1. Admin (Tenant Admin)

- **Home Path:** /admin
- **Permissions:** TOUTES (accès complet)
- **Total:** ~100+ permissions

2. HR Manager

- **Home Path:** /hr
- **Permissions:**
 - contractors.*
 - agencies.*
 - onboarding.*
 - companies.view
 - tasks.view/create/assign

3. Finance Manager

- **Home Path:** /finance
- **Permissions:**
 - invoices.*
 - banks.*
 - payroll.*
 - contracts.view

4. Agency Owner

- **Home Path:** /agency
- **Permissions:**

- `agencies.*`
- `contractors.*`
- `contracts.*`
- Toutes les permissions `.view`

5. Payroll Manager

- **Home Path:** `/payroll`

- **Permissions:**

- `payroll.*`
- `payslip.*`
- `contracts.view`
- `invoices.view`

6. Recruiter

- **Home Path:** `/recruitment`

- **Permissions:**

- `contractors.*`
- `leads.*`

7. Contractor PROBLÉMATIQUE

- **Home Path:** `/contractor`

- **Permissions:** (21 actuellement)

```

onboarding.responses.view_own
onboarding.responses.submit
contracts.view           ← Trop large
payslip.view
contractors.update      ← Doit être contractors.profile.update
contractors.documents.upload
contractors.documents.view
timesheet.view/create/submit
expense.view/create/submit
invoices.view/create
payroll.view
referrals.view/create/track

```

8. Viewer

- **Home Path:** `/home`

- **Permissions:** Toutes les permissions `.view` seulement



Bug Critique Identifié

Le Problème : `contractors.view`

Symptôme: La permission `contractors.view` est utilisée pour deux contextes incompatibles :

1. **Page personnelle du contractor** (`/contractor/information`)

- Objectif : Voir SES propres informations

- Utilisateur : Le contractor lui-même
- Doit afficher : Nom, email, téléphone, adresse, etc.

2. Page admin de gestion (/contractors)

- Objectif : Gérer TOUS les contractors
- Utilisateur : Admin, HR Manager
- Doit afficher : Liste de tous les contractors, actions CRUD

Conséquence:

- Un contractor avec `contractors.view` peut voir la page admin
- Un admin pourrait être redirigé vers la page personnelle
- Confusion dans la logique de navigation

Solution Requise:

- Séparer en deux permissions distinctes :
 - * `profile.view` ou `contractors.profile.view_own` → Pour le contractor
 - * `contractors.view` ou `contractors.manage.view_all` → Pour les admins
-

Mapping Complet : Pages → Permissions Actuelles

Pages Basées sur les Rôles (À Refactoriser)

Route	Permission Actuelle	Utilisateur Cible	Problème
/contractor/information	contractors.view	Contractor	 Trop large
/contractor/invoices	invoices.view	Contractor	Devrait être invoices.view_own
/contractor/onboarding	onboarding.responses.view_own	Contractor	 Correct
/contractor/payslips	payslip.view	Contractor	Devrait être payslips.view_own
/contractor/remits	payroll.view	Contractor	Devrait être payroll.view_own
/contractor/refer	referrals.view	Contractor	 Correct
/contractor/time-expenses	timesheet.create , expense.create	Contractor	 Correct
/agency/information	agencies.view	Agency	 Trop large
/agency/contracts	contracts.view	Agency	 Trop large
/agency/invoices	invoices.view	Agency	 Trop large
/payroll-partner/information	Inconnue	Payroll Partner	 Non documenté

Pages Fonctionnelles (Mieux Structurées)

Route	Permission Actuelle	Type	Statut
/contractors	contractors.view	Admin	⚠️ Conflit avec contractor
/agencies	agencies.view	Admin	✓ OK
/contracts	contracts.view	Admin	✓ OK
/invoices/agency	invoices.view	Admin	✓ OK
/invoices/contractor	invoices.view	Admin	✓ OK
/timesheets	timesheet.view	Admin	✓ OK
/expenses	expense.view	Admin	✓ OK
/payslips	payslip.view	Admin	✓ OK
/tasks	tasks.view	Tous	✓ OK
/reports/*	audit_logs.view	Admin	✓ OK
/settings/*	Diverses	Admin	✓ OK

🎯 Permissions Actuelles (Analyse Détailée)

Structure de `PERMISSION_TREE`

```
PERMISSION_TREE = {
    tenant: { view, update, branding, billing, roles, users, ... }
    companies: { view, create, update, delete }
    agencies: { view, create, update, delete, assignContractor, manageTeam, notes }
    contractors: {
        view, ⚠ PROBLÉMATIQUE
        create,
        update, ⚠ Trop large (admin + contractor)
        delete,
        documents: { upload, view, delete },
        onboarding: { start, update, review, validate },
        assignToAgency,
        changeStatus
    }
    contracts: { view, create, update, delete, send, approve, reject, ... }
    invoices: { view, create, update, delete, send, markPaid, export }
    payroll: { view, generate, update, send, markPaid, create, delete }
    payslip: { view, generate, update, send, markPaid, create, delete }
    timesheet: { view, create, update, delete, approve, submit }
    expense: { view, create, update, delete, approve, submit, listAll, reject, pay }
    referrals: { view, create, update, delete, track }
    tasks: { view, create, update, delete, assign, complete }
    leads: { view, create, update, delete, export }
    audit: { view, export }
    settings: { view, update }
    onboarding: {
        templates: { view, create, update, delete },
        questions: { add, update, delete },
        responses: {
            view, ⚠ Admin view all
            viewOwn, ✓ Contractor view own
            submit,
            review
        }
    }
    ...
}
```

Permissions Manquantes (Nécessaires)

1. Propriété (Ownership)

- `contractors.profile.view_own` - Voir son propre profil
- `contractors.profile.update_own` - Mettre à jour son profil
- `contracts.view_own` - Voir ses propres contrats
- `invoices.view_own` - Voir ses propres factures
- `payslips.view_own` - Voir ses propres bulletins
- `payroll.view_own` - Voir ses propres paiements

2. Actions Granulaires

- `contractors.manage.view_all` - Admin voir tous les contractors
- `contractors.manage.create` - Admin créer des contractors
- `contractors.manage.update` - Admin modifier des contractors
- `contractors.manage.delete` - Admin supprimer des contractors

3. Permissions de Gestion d'Équipe

- `team.view` - Voir son équipe
 - `team.manage` - Gérer son équipe
 - `team.invite` - Inviter des membres
-

Proposition de Nouvelle Architecture

Principe : Structure Fonctionnelle

Au lieu de :

```
/contractor/
/agency/
/payroll-partner/
```

Nous aurons :

<pre>/profile/ /dashboard/ /contracts/ /invoices/ /timesheets/ /expenses/ /payments/ /team/ /onboarding/ /reports/ /settings/</pre>	<ul style="list-style-type: none"> ← Profil personnel (tous les rôles) ← Dashboard personnalisé par rôle ← Gestion des contrats ← Gestion des factures ← Feuilles de temps ← Dépenses ← Paiements (remits, payslips) ← Gestion d'équipe (agencies, contractors) ← Processus d'onboarding ← Rapports et analytics ← Paramètres système
---	--

Nouvelle Structure de Permissions

```

NEW_PERMISSION_TREE = {
    // Profil Personnel (tous les utilisateurs)
    profile: {
        view: "profile.view",           // Voir son profil
        update: "profile.update",      // Modifier son profil
        documents: {
            view: "profile.documents.view",
            upload: "profile.documents.upload",
            delete: "profile.documents.delete"
        }
    },
    // Gestion des Contractors
    contractors: {
        // Permissions de gestion (admin/hr)
        manage: {
            view_all: "contractors.manage.view_all",
            create: "contractors.manage.create",
            update: "contractors.manage.update",
            delete: "contractors.manage.delete",
            change_status: "contractors.manage.change_status"
        },
        // Permissions de visualisation propre
        view_own: "contractors.view_own",
        update_own: "contractors.update_own"
    },
    // Contrats
    contracts: {
        // Admin
        manage: {
            view_all: "contracts.manage.view_all",
            create: "contracts.manage.create",
            update: "contracts.manage.update",
            delete: "contracts.manage.delete",
            approve: "contracts.manage.approve",
            reject: "contracts.manage.reject"
        },
        // Utilisateur
        view_own: "contracts.view_own"
    },
    // Factures
    invoices: {
        manage: {
            view_all: "invoices.manage.view_all",
            create: "invoices.manage.create",
            update: "invoices.manage.update",
            delete: "invoices.manage.delete",
            mark_paid: "invoices.manage.mark_paid"
        },
        view_own: "invoices.view_own",
        create_own: "invoices.create_own"
    },
    // Feuilles de temps
    timesheets: {
        manage: {
            view_all: "timesheets.manage.view_all",
            approve: "timesheets.manage.approve",
            reject: "timesheets.manage.reject"
        },
    }
}

```

```
view_own: "timesheets.view_own",
create: "timesheets.create",
submit: "timesheets.submit"
},

// Dépenses
expenses: {
  manage: {
    view_all: "expenses.manage.view_all",
    approve: "expenses.manage.approve",
    reject: "expenses.manage.reject",
    mark_paid: "expenses.manage.mark_paid"
  },
  view_own: "expenses.view_own",
  create: "expenses.create",
  submit: "expenses.submit"
},

// Paiements
payments: {
  payslips: {
    view_all: "payments.payslips.view_all",
    view_own: "payments.payslips.view_own",
    generate: "payments.payslips.generate"
  },
  remits: {
    view_all: "payments.remits.view_all",
    view_own: "payments.remits.view_own"
  }
},
// ... autres modules
}
```



Nouvelle Structure de Dossiers Proposée

app/(dashboard)/(modules)/	
profile/	↳ Profil personnel [profile.view] [profile.update] [profile.documents.view]
page.tsx	
edit/page.tsx	
documents/page.tsx	
dashboard/	↳ Dashboard personnalisé [Dynamic routing by role]
page.tsx	
contracts/	↳ Gestion des contrats [contracts.manage.view_all OR con-
page.tsx	
tracts.view_own]	
[id]/page.tsx	[contracts.manage.update]
[id]/edit/page.tsx	[contracts.manage.create]
new/page.tsx	
invoices/	↳ Factures [invoices.manage.view_all OR invoices.view_own]
page.tsx	
[id]/page.tsx	
new/page.tsx	[invoices.create_own OR invoices.manage.create]
timesheets/	↳ Feuilles de temps [timesheets.manage.view_all OR
page.tsx	
timesheets.view_own]	
new/page.tsx	[timesheets.create]
[id]/page.tsx	
expenses/	↳ Dépenses [expenses.manage.view_all OR expenses.view_own]
page.tsx	
new/page.tsx	[expenses.create]
[id]/page.tsx	
payments/	↳ Paiements [payments.payslips.view_all OR view_own]
payslips/page.tsx	
remitts/page.tsx	[payments.remitts.view_all OR view_own]
team/	↳ Gestion d'équipe [contractors.manage.view_all]
contractors/	
page.tsx	
[id]/page.tsx	
new/page.tsx	
agencies/	[agencies.manage.view_all]
page.tsx	
[id]/page.tsx	
members/	[team.view]
page.tsx	
onboarding/	↳ Onboarding [onboarding.responses.view_own OR view_all]
page.tsx	
templates/page.tsx	[onboarding.templates.view]
referrals/	↳ Parrainages [referrals.view]
page.tsx	
reports/	↳ Rapports
page.tsx	
activity-logs/page.tsx	
analytics/page.tsx	
...	
settings/	↳ Paramètres
page.tsx	

```

roles/page.tsx
users/page.tsx
...

```

Avantages:

- Structure logique basée sur les fonctionnalités
- Pas de duplication de code entre rôles
- Facile d'ajouter de nouveaux rôles
- Permissions granulaires et claires
- Réutilisabilité maximale des composants

Matrice Rôles/Permissions (Proposée)

Contractor

Module	Permissions
Profil	profile.view , profile.update , profile.documents.*
Contrats	contracts.view_own
Factures	invoices.view_own , invoices.create_own
Feuilles de temps	timesheets.view_own , timesheets.create , timesheets.submit
Dépenses	expenses.view_own , expenses.create , expenses.submit
Paiements	payments.payslips.view_own , payments.remits.view_own
Onboarding	onboarding.responses.view_own , onboarding.responses.submit
Parrainages	referrals.view , referrals.create , referrals.track

Admin / HR Manager

Module	Permissions
Profil	profile.*
Contractors	contractors.manage.* (view_all, create, update, delete)
Contrats	contracts.manage.* (view_all, create, update, approve, etc.)
Factures	invoices.manage.*
Feuilles de temps	timesheets.manage.* (view_all, approve, reject)
Dépenses	expenses.manage.*
Paiements	payments.*.view_all , payments.*.generate
Équipe	team.* , agencies.*
Rapports	reports.* , audit.*
Paramètres	settings.* , tenant.*

Agency Owner

Module	Permissions
Profil	profile.*
Contractors	contractors.manage.view_all (assigned only)
Contrats	contracts.manage.* (own contracts)
Factures	invoices.manage.* (own invoices)
Feuilles de temps	timesheets.manage.view_all , timesheets.manage.approve
Équipe	team.view , team.manage (own team)

Plan de Refactorisation

Phase 1: Préparation (Étapes 1-4)

-  Cloner le repo
-  Analyser la structure
-  Identifier tous les rôles
-  Mapper toutes les pages/routes

Phase 2: Conception (Étapes 5-7)

-  Concevoir le nouveau système de permissions granulaires
-  Définir la matrice rôles/permissions
-  Concevoir la nouvelle structure de dossiers

Phase 3: Implémentation Backend (Étape 8)

-  Créer/améliorer les guards/middleware
-  Mettre à jour `server/rbac/permissions.ts`
-  Mettre à jour `scripts/seed/00-permissions.ts`
-  Mettre à jour `scripts/seed/01-roles.ts`

Phase 4: Restructuration (Étapes 9-11)

-  Déplacer les fichiers vers la nouvelle structure
-  Adapter tous les imports
-  Mettre à jour les routes et la navigation
-  Mettre à jour `lib/dynamicMenuConfig.ts`

Phase 5: Composants (Étape 12)

-  Créer `PermissionGuard` component
-  Créer `RouteGuard` component
-  Améliorer `usePermissions` hook

Phase 6: Documentation et Tests (Étapes 13-14)

-  Créer `RBAC_STRUCTURE.md`
-  Tester toutes les routes
-  Créer une PR

Notes Importantes

Fichiers Critiques à Modifier

1. Permissions:

- `server/rbac/permissions.ts`
- `scripts/seed/00-permissions.ts`
- `scripts/seed/01-roles.ts`

2. Navigation:

- `lib/dynamicMenuConfig.ts`
- `middleware.ts`
- `lib/routing/dynamic-router.ts`

3. Pages à Déplacer:

- Toutes les pages sous `/contractor/`
- Toutes les pages sous `/agency/`
- Toutes les pages sous `/payroll-partner/`

4. Composants à Créer:

- `components/guards/PermissionGuard.tsx`
- `components/guards/RouteGuard.tsx`
- `components/profile/*` (nouveaux composants de profil)

Risques et Mitigation

Risque	Impact	Mitigation
Casser les routes existantes	Haut	Créer des redirects temporaires
Perdre des permissions	Moyen	Tester avec tous les rôles
Imports cassés	Haut	Utiliser des alias TypeScript
Confusion utilisateurs	Moyen	Documentation claire

⌚ Prochaines Étapes

1. Valider cette analyse avec l'équipe
2. Créer une branche `refactor/rbac-architecture`
3. Implémenter le nouveau système de permissions
4. Créer des migrations de données si nécessaire
5. Déplacer les fichiers progressivement
6. Tester exhaustivement
7. Merger via PR

Statut: Analyse Complète

Date de Mise à Jour: 17 Novembre 2025

Prêt pour: Phase de Conception