



Implémentation Tenant Company - Documentation

Cette documentation décrit les modifications apportées pour implémenter le système de **Tenant Companies** et l'assignation d'admin/approver aux contrats MSA/SOW.



Vue d'ensemble des modifications

Problème Initial

Lors de la création d'un MSA ou SOW, les champs "Admin principal" et "Approver" étaient liés à des users individuels. Cependant, ces rôles devraient être liés au **tenant (la plateforme)** et assignés ultérieurement par les administrateurs.

Solution Implémentée

1. **Identification des Tenant Companies** : Un nouveau champ `tenantCompany` permet de distinguer les companies appartenant à la plateforme des companies clientes
2. **Création simplifiée des contrats** : Les MSA/SOW ne demandent plus de sélectionner admin/approver lors de la création
3. **Assignation différée** : Un nouveau composant permet aux admins d'assigner les rôles après la création du contrat



Fichiers modifiés

1. Schéma Prisma (`prisma/schema.prisma`)

Modification du modèle Company

```

model Company {
  // ... champs existants ...

  // 🔥 NEW Identify tenant companies (companies owned by the platform)
  tenantCompany Boolean @default(false)

  // ... reste des champs ...
}

```

Migration SQL

Un fichier de migration SQL a été créé : `prisma/migrations/manual/add_tenant_company_field.sql`

⚠ **IMPORTANT** : Cette migration doit être exécutée manuellement sur votre base de données :

```

ALTER TABLE "companies"
ADD COLUMN "tenantCompany" BOOLEAN NOT NULL DEFAULT false;

CREATE INDEX "companies_tenantCompany_idx" ON "companies"("tenantCompany");

```

2. Composants de création de contrats

`components/contracts/MSACreateModal.tsx`

Changements :

- ☒ Suppression des champs de sélection pour `clientAdminId` et `approverId`
- ☒ Ajout d'un message informatif expliquant que le contrat sera lié au tenant
- ☒ Création du contrat avec uniquement le participant `client` (utilisateur actuel)
- ☒ Les admins pourront assigner admin/approver ultérieurement

Ancien comportement :

```
// ❌ ANCIEN - Sélection obligatoire des admin/approver
const [clientAdminId, setClientAdminId] = useState("");
const [approverId, setApproverId] = useState("");
```

Nouveau comportement :

```
// ✅ NOUVEAU - Pas de sélection, assignation différée
participants: [
  {
    userId: currentUserId,
    role: "client",
    requiresSignature: false,
    isPrimary: true,
  },
]
```

`components/contracts/SOWCreateModal.tsx`

Changements identiques au MSACreateModal :

- ☒ Suppression de la validation des admin/approver du MSA parent
- ☒ Message informatif ajouté
- ☒ Création simplifiée avec uniquement worker et client

3. Composant de gestion des Companies

`components/modals/company-modal.tsx`

Ajouts :


- ☒ Nouveau champ `tenantCompany` dans le type `CompanyFormValues`
- ☒ Toggle/Switch pour marquer une company comme Tenant Company
- ☒ Interface utilisateur avec styling distinctif (gradient bleu/indigo)

Interface :

```
<div className="flex items-center justify-between p-4 bg-gradient-to-r from-blue-50 to-indigo-50 border border-blue-200 rounded-lg">
  <div className="flex-1">
    <Label>Tenant Company</Label>
    <p className="text-sm">Cette company appartient-elle à la plateforme ?</p>
  </div>
  <Switch checked={formData.tenantCompany} ... />
</div>
```

`app/(dashboard)/(modules)/settings/companies/page.tsx`

Ajouts :

- ☒ Affichage visuel distinctif pour les Tenant Companies
- ☒ Badge “ Tenant Company” avec gradient purple/indigo
- ☒ Icône avec ring et couleurs spéciales

Rendu visuel :

- Tenant Companies : Fond gradient indigo/purple avec ring, icône indigo
- Companies normales : Fond bleu standard

4. Nouveau composant d'assignation

`components/contracts/ContractAssignmentModal.tsx` ✨ NOUVEAU

Fonctionnalités :

- ☒ Sélection d'une Tenant Company pour le contrat
- ☒ Assignation d'un Admin principal (avec signature requise)
- ☒ Assignation d'un Approver (sans signature)
- ☒ Affichage des assignations existantes avec avertissement
- ☒ Validation complète avant soumission

Utilisation :

```
import { ContractAssignmentModal } from "@components/contracts/ContractAssignmentModal";

<ContractAssignmentModal
  open={isAssignmentModalOpen}
  onOpenChange={setIsAssignmentModalOpen}
  contract={selectedContract}
  onSuccess={() => {
    // Rafraîchir les données
    refetch();
  }}
/>
```

Logique :

1. Mise à jour du contrat avec la Tenant Company sélectionnée
2. Ajout du participant Admin principal (`client_admin` , `requiresSignature: true`)
3. Ajout du participant Approver (`approver` , `requiresSignature: false`)

5. Permissions

lib/permissions.ts

Nouvelles permissions ajoutées :

```
export const PERMISSIONS = {
  // ... permissions existantes ...

  // Contracts
  CONTRACTS_ASSIGN: "contracts.assign", // 🔥 NEW – Assign admin/approver

  // Companies
  COMPANIES_MANAGE_TENANT: "companies.manage_tenant", // 🔥 NEW – Manage tenant companies
}
```

Recommandation : Attribuer ces permissions aux rôles suivants :

- `CONTRACTS_ASSIGN` → Rôles : ADMIN, SUPER_ADMIN
- `COMPANIES_MANAGE_TENANT` → Rôles : ADMIN, SUPER_ADMIN



Guide d'utilisation

Étape 1 : Créer une Tenant Company

1. Aller dans **Settings** → **Companies**
2. Cliquer sur **"Add Company"**
3. Remplir les informations de la company
4. **Activer le toggle "Tenant Company"** 🔥
5. Sauvegarder

Étape 2 : Créer un MSA

1. Aller dans **Contracts**
2. Cliquer sur **"Nouveau MSA"**
3. Remplir les informations (titre, company cliente, devise, etc.)
4. **Note** : Pas besoin de sélectionner Admin/Approver
5. Voir le message : "Ce MSA sera lié au tenant (plateforme)"
6. Créer le MSA

Étape 3 : Assigner Admin & Approver

1. Dans la liste des contrats, sélectionner le MSA créé
2. Cliquer sur **"Assigner Admin & Approver"** (nouveau bouton à ajouter dans votre UI)
3. Dans le modal d'assignation :
 - Sélectionner une **Tenant Company**
 - Sélectionner un **Admin principal**
 - Sélectionner un **Approver**
4. Confirmer l'assignation

Étape 4 : Créer un SOW lié au MSA

1. Aller dans **Contracts**

2. Cliquer sur **“Nouveau SOW”**
3. Sélectionner le MSA parent
4. Sélectionner le Worker (contractor)
5. Remplir les détails du SOW
6. **Note** : Les admin/approver seront hérités du MSA ou assignés séparément
7. Créer le SOW

Intégration dans l'interface

Ajouter le bouton d'assignation dans la vue détail du contrat

Exemple : Dans `components/contracts/ContractView.tsx` ou la page de détail :

```
import { ContractAssignmentModal } from "@components/contracts/ContractAssignmentModal";

function ContractDetailPage() {
  const [isAssignmentModalOpen, setIsAssignmentModalOpen] = useState(false);

  return (
    <div>
      { /* Bouton visible uniquement si admin/approver non assignés */ }
      { !hasAdminAndApprover && (
        <Button onClick={() => setIsAssignmentModalOpen(true)}>
          <UserCheck className="mr-2 h-4 w-4" />
          Assigner Admin & Approver
        </Button>
      ) }
      <ContractAssignmentModal
        open={isAssignmentModalOpen}
        onOpenChange={setIsAssignmentModalOpen}
        contract={contract}
        onSuccess={() => refetch()}
      />
    </div>
  );
}
```

Vérification des permissions

Dans vos composants :

```
import { PERMISSIONS } from "@lib/permissions";
import { useSession } from "next-auth/react";

function MyComponent() {
  const { data: session } = useSession();
  const canAssign = session?.user?.permissions?.includes(PERMISSIONS.CONTRACTS_ASSIGN);
  const canManageTenantCompanies = session?.user?.permissions?.includes(PERMISSIONS.COMPANIES_MANAGE_TENANT);

  return (
    <>
      {canAssign && <AssignButton />}
      {canManageTenantCompanies && <TenantCompanyToggle />}
    </>
  );
}
```

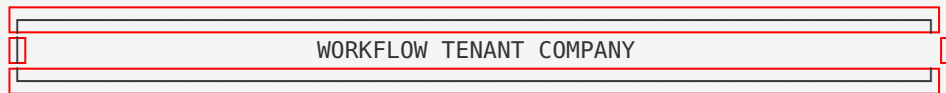
✓ Checklist de déploiement

Avant de déployer en production :

- [] **Base de données** : Exécuter la migration SQL pour ajouter le champ `tenantCompany`
- [] **Permissions** : Créer les nouvelles permissions dans la base de données
 - `contracts.assign`
 - `companies.manage_tenant`
- [] **Rôles** : Attribuer ces permissions aux rôles appropriés (ADMIN, SUPER_ADMIN)
- [] **UI** : Ajouter le bouton "Assigner Admin & Approuver" dans la vue détail des contrats
- [] **Tests** : Tester le workflow complet :
 1. Créer une Tenant Company
 2. Créer un MSA sans admin/approuver
 3. Assigner admin/approuver via le nouveau modal
 4. Créer un SOW lié au MSA
 5. Vérifier que les relations sont correctes



Schéma de flux



1. CREATE TENANT COMPANY
 - ☐ Admin goes to Settings > Companies
 - ☐ Creates new company
 - ☐ ☒ Enables "Tenant Company" toggle
2. CREATE MSA
 - ☐ Client creates MSA
 - ☐ NO admin/approver selection required
 - ☐ Contract created with only "client" participant
 - ☐ Message: "Contrat envoyé à la plateforme"
3. ADMIN ASSIGNS ROLES
 - ☐ Admin opens contract detail
 - ☐ Clicks "Assigner Admin & Approver"
 - ☐ Selects:
 - ☐ Tenant Company
 - ☐ Admin principal (with signature)
 - ☐ Approver (no signature)
 - ☐ Participants added to contract
4. CREATE SOW
 - ☐ Client creates SOW linked to MSA
 - ☐ Selects worker/contractor
 - ☐ Same logic as MSA
 - ☐ Admin can assign roles separately if needed
5. CONTRACT WORKFLOW
 - ☐ Admin principal signs
 - ☐ Approver validates
 - ☐ Contractor signs (for SOW)
 - ☐ Contract activated



Problèmes connus et solutions

Problème : Prisma migration error

Symptôme : Erreur lors de la génération de la migration Prisma

Solution : Exécuter la migration SQL manuellement (fichier fourni)

Problème : tRPC method not found

Symptôme : `api.contract.addParticipant` undefined

Solution : Vérifier que le router contract est bien importé et que la méthode existe

Problème : Permissions non appliquées

Symptôme : Les utilisateurs voient des fonctionnalités qu'ils ne devraient pas voir

Solution :

1. Vérifier que les permissions sont créées dans la DB

2. Attribuer les permissions aux rôles appropriés
3. Utiliser les guards `hasPermission` dans les composants



Notes techniques

Architecture des participants

```
ContractParticipant {
  contractId: string
  userId: string
  role: "contractor" | "client_admin" | "approver" | "client" | ...
  requiresSignature: boolean
  isPrimary: boolean
  approved: boolean // Pour les approvers uniquement
}
```

Règles métier

1. **Admin principal** (`client_admin`) :
 - `requiresSignature: true`
 - Responsable de la signature du contrat côté tenant
 - Un seul par contrat
2. **Approver** :
 - `requiresSignature: false`
 - N'utilise PAS la signature, mais le champ `approved`
 - Valide le contrat avant activation
 - Peut être multiple (workflow multi-étapes)
3. **Tenant Company** :
 - `tenantCompany: true`
 - Représente une entité de la plateforme
 - Peut être multiple (ex: Aspirock Suisse, Aspirock France)
 - Utilisée pour structurer les contrats par région/entité



Prochaines étapes recommandées

1. **Workflow d'approbation avancé** :
 - Gérer plusieurs approvers en cascade
 - Historique des approbations/rejets
 - Notifications automatiques
2. **Dashboard Admin** :
 - Vue globale des contrats en attente d'assignation
 - Statistiques sur les tenant companies
 - Filtres avancés
3. **Audit Trail** :
 - Logger toutes les assignations

- Historique des changements de participants
- Traçabilité complète

4. **Tests automatisés :**

- Tests unitaires pour les composants
- Tests d'intégration pour le workflow complet
- Tests E2E avec Playwright/Cypress

Support

Pour toute question ou problème :

1. Vérifier cette documentation
2. Consulter les commentaires dans le code (🔥 NEW, ✅, etc.)
3. Contacter l'équipe de développement

Dernière mise à jour : 2025-11-26

Version : 1.0.0

Auteur : DeepAgent (Abacus.AI)