

CAHIER DES CHARGES TECHNIQUE

Application SaaS pour Avocats

Plateforme collaborative de gestion contractuelle

Version : 1.0
Date : Octobre 2025
Statut : Version synthétique

SDEN - Sami Dzogang Engineering

TABLE DES MATIÈRES

1	Architecture Technique	4
1.1	Stack Technique Global	4
1.1.1	Frontend	4
1.1.2	Backend	4
1.1.3	Services Externes	4
1.2	Architecture Base de Données	5
2	Fonctionnalités Principales	6
2.1	Comparaison de Documents	6
2.2	Éditeur Collaboratif Temps Réel	6
2.3	Conversion PDF vers Format Éditable	6
2.4	Suggestions Juridiques par IA	7
2.5	Gestion des Rôles et Permissions (RBAC)	7
3	Sécurité et Conformité	8
3.1	Chiffrement des Données	8
3.2	Authentification et Autorisation	8
3.3	Conformité RGPD	8
3.4	Sauvegarde et Résilience	8
3.5	Protection et Rate Limiting	9
4	API REST	9
4.1	Structure Générale	9
4.2	Endpoints Principaux	9
4.2.1	Authentification	9
4.2.2	Utilisateurs	10
4.2.3	Workspaces et Projets	10
4.2.4	Documents et Comparaison	10

4.2.5	Commentaires et Chat	11
4.2.6	Intelligence Artificielle	11
4.2.7	Notifications et Facturation	11
5	Roadmap de Développement - MVP 4 Semaines	12
5.1	Approche Agile Accélérée	12
5.2	Semaine 1 : Fondations & Authentification	12
5.2.1	Objectifs	12
5.2.2	Livrables	12
5.2.3	User Stories Validées	12
5.3	Semaine 2 : Projets, Documents & Upload	13
5.3.1	Objectifs	13
5.3.2	Livrables	13
5.3.3	User Stories Validées	13
5.4	Semaine 3 : Commentaires, Chat & Invitations	14
5.4.1	Objectifs	14
5.4.2	Livrables	14
5.4.3	User Stories Validées	14
5.5	Semaine 4 : Comparaison, Paiement & Déploiement	15
5.5.1	Objectifs	15
5.5.2	Livrables	15
5.5.3	User Stories Validées	15
5.6	Fonctionnalités Reportées (Post-MVP)	15
5.7	Livrable Final MVP (Fin Semaine 4)	16
5.7.1	Application Fonctionnelle avec :	16
5.8	Conclusion Roadmap MVP	16
6	Risques et Mitigation	17
6.0.1	Performance Technique	17

ARCHITECTURE TECHNIQUE

Stack Technique Global

Le projet s'appuie sur une architecture moderne et éprouvée garantissant scalabilité, maintenabilité et sécurité.

1.1.1 Frontend

- **Framework** : Next.js 14+ (App Router) avec React 18+
- **Interface utilisateur** : Tailwind CSS + Shadcn/ui (composants)
- **Animations** : Framer Motion
- **Éditeur de texte** : Tiptap avec édition collaborative (Yjs)
- **Gestion d'état** : Zustand (léger et performant)
- **Validation** : Zod + React Hook Form

1.1.2 Backend

- **Runtime** : Node.js 20 LTS
- **Framework API** : Express.js ou Fastify
- **Base de données** : PostgreSQL 15+
- **Cache** : Redis 7+ (sessions et performances)
- **ORM** : Prisma (gestion schema et migrations)
- **WebSocket** : Socket.io (chat et collaboration temps réel)
- **File d'attente** : BullMQ (traitements asynchrones)

1.1.3 Services Externes

- **Authentification** : NextAuth.js v5 (Auth.js)
- **OCR / Parsing PDF** : Google Document AI
- **Intelligence Artificielle** : OpenAI GPT-4o (suggestions juridiques)
- **Paieement** : Stripe (via Lemon Squeezy)
- **Emails** : Resend
- **Stockage documents** : AWS S3 ou compatible
- **Monitoring** : Sentry (erreurs) + PostHog (analytics produit)

Architecture Base de Données

Le schéma relationnel s'articule autour des entités suivantes :

users	Utilisateurs de la plateforme (id, email, password_hash, full_name, role)
workspaces	Espaces de travail (id, name, owner_id, subscription_tier)
projects	Projets contractuels (id, workspace_id, name, client_name, contract_type, status)
documents	Versions de documents (id, project_id, version_number, file_path, parsed_content)
comments	Annotations collaboratives (id, document_id, user_id, content, position_data)
chat_messages	Messagerie projet (id, project_id, user_id, content)
notifications	Système de notifications (id, user_id, type, related_entity_id, is_read)

Tables complémentaires : **templates**, **document_changes**, **project_members**, **workspace_members**, **support_tickets**.

FONCTIONNALITÉS PRINCIPALES

Comparaison de Documents

La plateforme permet la comparaison intelligente de multiples versions contractuelles :

- Upload jusqu'à 6 versions simultanées et possibilité d'ajouter X versions supplémentaires.
- Algorithme de différenciation (bibliothèque `jsdiff`)
- Affichage côte à côte ou fusionné
- Coloration automatique des modifications (ajouts, suppressions, modifications)

Éditeur Collaboratif Temps Réel

L'édition collaborative repose sur des technologies éprouvées :

- Édition simultanée multi-utilisateurs sans conflit
- CRDT (Conflict-free Replicated Data Type) via Yjs
- Curseurs visibles en temps réel avec identification utilisateur
- Historique complet avec restauration de versions antérieures
- Synchronisation < 200ms entre clients

Conversion PDF vers Format Éditable

Pipeline de traitement automatisé :

1. Upload du document PDF
2. Extraction par OCR (Google Document AI)
3. Parsing de la structure documentaire (sections, clauses, articles)
4. Conversion en JSON standardisé
5. Import dans l'éditeur Tiptap
6. Interface de correction manuelle des erreurs OCR

Objectif de précision : > 95% sur documents de qualité standard.

Suggestions Juridiques par IA

Assistance intelligente à la rédaction :

- Sélection de texte + prompt utilisateur contextuel
- Exemples de requêtes : "rendre plus formel", "simplifier le langage", "traduire FREN"
- Génération de plusieurs suggestions par requête
- Workflow Accept / Reject / Edit manuel
- Rate limiting : 20 requêtes/minute par utilisateur

Gestion des Rôles et Permissions (RBAC)

Système de permissions granulaires à deux niveaux :

Niveau Workspace :

- **Owner** : Droits complets + gestion facturation
- **Admin** : Gestion membres et projets
- **Member** : Accès selon permissions projet

Niveau Project :

- **Creator** : Administrateur du projet
- **Editor** : Modification des documents
- **Commenter** : Annotations uniquement (pas d'édition)
- **Reader** : Consultation seule

SÉCURITÉ ET CONFORMITÉ

Chiffrement des Données

En transit :

- TLS 1.3 obligatoire sur toutes connexions
- HSTS (HTTP Strict Transport Security) activé

Au repos :

- Chiffrement AES-256 des documents (S3)
- Chiffrement transparent base de données (PostgreSQL TDE)
- Secrets stockés dans vault sécurisé

Messagerie :

- Chiffrement de bout en bout (E2EE)
- Implémentation Signal Protocol ou équivalent
- Clés générées côté client

Authentification et Autorisation

- Tokens JWT avec mécanisme de refresh
- Hashing Argon2 ou bcrypt pour mots de passe
- MFA disponible (TOTP - applications d'authentification)
- Middleware de vérification des permissions sur chaque endpoint

Conformité RGPD

- Export de données utilisateur (endpoint `/api/users/:id/export`)
- Droit à l'oubli (suppression complète des données)
- Consentement explicite pour analytics
- Conservation des logs d'audit : 1 an minimum

Sauvegarde et Résilience

- Backup automatique quotidien de la base de données
- Backup incrémental toutes les 6 heures
- Rétention : 30 jours glissants
- Test de restauration mensuel obligatoire

Protection et Rate Limiting

Configuration par endpoint pour prévenir les abus :

- `/api/auth/login` : 5 tentatives / 15 minutes
- `/api/projects` : 60 requêtes / minute
- `/api/ai/suggest` : 20 requêtes / minute (coûts API)
- Endpoints standards : 100 requêtes / minute

API REST

Structure Générale

- **Base URL** : `https://companion.com/v1`
- **Format** : JSON exclusivement
- **Authentication** : Bearer token (JWT dans header)
- **Pagination** : Paramètres `?page=1&limit=50`
- **Filtrage** : Paramètres query string (`?status=active&sort=-created_at`)

Endpoints Principaux

4.2.1 Authentication

Route	Description
POST <code>/auth/register</code>	Inscription nouveau compte
POST <code>/auth/login</code>	Connexion utilisateur
POST <code>/auth/logout</code>	Déconnexion
POST <code>/auth/refresh</code>	Renouvellement token
POST <code>/auth/forgot-password</code>	Réinitialisation mot de passe

4.2.2 Utilisateurs

Route	Description
GET /users/me	Profil utilisateur connecté
PATCH /users/me	Modification profil
DELETE /users/me	Suppression compte (RGPD)
GET /users/me/export	Export données personnelles

4.2.3 Workspaces et Projets

Route	Description
GET /workspaces	Liste workspaces
POST /workspaces	Création workspace
GET /projects	Liste projets (filtrable)
POST /projects	Création projet
GET /projects/ :id	Détails projet
POST /projects/ :id/members	Invitation collaborateur

4.2.4 Documents et Comparaison

Route	Description
GET /projects/ :id/documents	Versions du document
POST /projects/ :id/documents	Upload nouvelle version
GET /documents/ :id/content	Contenu structuré
PATCH /documents/ :id/content	Modification contenu
POST /documents/compare	Comparaison versions
POST /documents/ :id/convert	Conversion PDF éditabile

4.2.5 Commentaires et Chat

Route	Description
GET /docs/:id/comments	Liste commentaires
POST /docs/:id/comments	Ajout commentaire
POST /docs/:id/resolve	Résoudre commentaire
GET /projects/:id/chat	Historique messages
POST /projects/:id/chat	Envoi message

4.2.6 Intelligence Artificielle

Route	Description
POST /ai/suggest	Suggestions reformulation
POST /ai/analyze	Analyse juridique (future)

4.2.7 Notifications et Facturation

Route	Description
GET /notifications	Liste notifications
PATCH /notifications/:id/read	Marquer comme lu
GET /billing/subscription	Infos abonnement
POST /billing/checkout	Session paiement Stripe

ROADMAP DE DÉVELOPPEMENT - MVP 4 SEMAINES

Approche Agile Accélérée

Cette roadmap concentre le développement sur un **MVP fonctionnel en 4 semaines** avec les fonctionnalités essentielles. L'objectif est de valider rapidement le produit auprès des utilisateurs avant d'investir dans des fonctionnalités avancées.

Semaine 1 : Fondations & Authentification

5.2.1 Objectifs

Mettre en place l'infrastructure technique et le système d'authentification complet.

5.2.2 Livrables

Infrastructure (Jours 1-2)

- Repository Git + CI/CD basique (GitHub Actions)
- Next.js 14+ configuré avec App Router
- PostgreSQL + Prisma (schema initial + migrations)
- NextAuth.js configuré
- Design system Shadcn/ui (composants de base)
- Variables d'environnement (dev/staging)

Authentification (Jours 3-5)

- Pages : /login, /register, /forgot-password
- API routes auth complètes (inscription, connexion, reset MDP)
- Envoi emails (Resend) - templates basiques
- Dashboard layout (navbar + sidebar responsive)
- Protection routes authentifiées

5.2.3 User Stories Validées

- Je peux créer un compte avec email/mot de passe
- Je peux me connecter et réinitialiser mon mot de passe
- J'accède à un dashboard après connexion

Semaine 2 : Projets, Documents & Upload

5.3.1 Objectifs

Permettre la création de projets et l'upload de documents PDF avec visualisation.

5.3.2 Livrables

Projets (Jours 1-2)

- Modèle BDD : `projects`, `project_members`
- API CRUD projets (POST, GET, PATCH, DELETE)
- Dashboard : liste projets avec cards (dernière activité en haut)
- Modal création projet (nom, client, type contrat)
- Page détail projet (layout avec tabs : Document, Chat)

Documents (Jours 3-5)

- Modèle BDD : `documents` (versioning simple)
- Stockage S3 (ou équivalent) configuré
- Zone upload drag & drop
- API upload document (POST `/projects/:id/documents`)
- Viewer PDF intégré (react-pdf)
- Extraction texte basique (pdf-parse library - pas d'OCR avancé pour MVP)
- Affichage historique versions (liste simple)

5.3.3 User Stories Validées

- Je peux créer un projet (nom + client + type)
- Je peux uploader un document PDF dans un projet
- Je vois mes projets sur le dashboard
- Je peux consulter le PDF uploadé

Semaine 3 : Commentaires, Chat & Invitations

5.4.1 Objectifs

Activer la collaboration avec commentaires, chat projet et système d'invitations.

5.4.2 Livrables

Commentaires (Jours 1-2)

- Modèle BDD : `comments` (avec position dans document)
- API commentaires (GET, POST, DELETE)
- Interface commentaires (sidebar latérale)
- Ajout commentaire sur section de texte
- Thread de réponses (1 niveau maximum pour MVP)

Chat Projet (Jour 3)

- Modèle BDD : `chat_messages`
- WebSocket setup basique (Socket.io)
- Interface chat (onglet dans projet)
- Envoi/réception messages temps réel

Invitations & Permissions (Jours 4-5)

- Système RBAC simplifié (2 rôles : Editor, Viewer)
- API invitation collaborateur par email
- Modal invitation dans projet
- Notifications in-app basiques (dropdown navbar)
- Email notification sur invitation

5.4.3 User Stories Validées

- Je peux ajouter un commentaire sur le document
- Je peux discuter en temps réel dans le chat projet
- Je peux inviter un collaborateur (Editor ou Viewer)
- Je reçois une notification quand je suis invité

Semaine 4 : Comparaison, Paiement & Déploiement

5.5.1 Objectifs

Finaliser avec comparaison de versions, système de paiement et déploiement production.

5.5.2 Livrables

Comparaison de Versions (Jours 1-2)

- Upload multi-versions (historique dans BDD)
- API comparaison : `POST /documents/compare`
- Algorithme diff basique (bibliothèque `jsdiff`)
- Interface split view (2 colonnes côte à côte)
- Coloration différences (ajouts en vert, suppressions en rouge)

Paiement (Jour 3)

- Intégration Stripe Checkout (mode test)
- Page `/pricing` simple
- Modal checkout (redirection Stripe)
- Webhook Stripe (activation abonnement)
- Limitation freemium : 1 projet max, levée après paiement

Déploiement & Tests (Jours 4-5)

- Déploiement staging (Vercel ou équivalent)
- Configuration environnement production
- Tests E2E critiques (signup → création projet → upload → invitation)
- Monitoring basique (Sentry pour erreurs)
- Documentation technique (README complet)
- Corrections bugs bloquants

5.5.3 User Stories Validées

- Je peux comparer 2 versions d'un document
- Les différences sont clairement identifiées
- Je peux souscrire à l'offre Standard (23€/mois)
- Mon abonnement est activé automatiquement après paiement

Fonctionnalités Reportées (Post-MVP)

Les éléments suivants sont **exclus du MVP 4 semaines** et seront développés en Phase 2 selon feedback utilisateurs :

- Édition collaborative temps réel (Yjs/Tiptap Collaboration)

- Suggestions IA (OpenAI GPT-4o)
- Conversion PDF intelligente (OCR avancé)
- Templates juridiques prédéfinis
- Onboarding interactif
- Mentions utilisateurs (@username)
- Système notifications avancé
- Analytics et exports avancés

Justification : Ces fonctionnalités nécessitent des intégrations complexes. Le MVP doit d'abord valider le besoin core : **collaboration autour de documents contractuels**.

Livrable Final MVP (Fin Semaine 4)

5.7.1 Application Fonctionnelle avec :

Authentification sécurisée (inscription, connexion, reset MDP)

Gestion de projets (création, consultation, archivage)

Upload & visualisation PDF (multi-versions)

Comparaison de versions (diff visuel côte à côte)

Collaboration basique (commentaires + chat temps réel)

Invitations collaborateurs (2 rôles : Editor, Viewer)

Païement Stripe (freemium → Standard 23€/mois)

Déployé en production (accessible publiquement)

Conclusion Roadmap MVP

Cette approche **Lean Startup** privilégie :

Vitesse : MVP fonctionnel en 4 semaines

Validation : Feedback terrain avant investissement massif

Flexibilité : Pivots possibles selon retours utilisateurs

Qualité : Focus sur features core, bien exécutées

RISQUES ET MITIGATION

Risque		Impact	Prob.	Mitigation
Précision OCR insuffisante		Élevé	Moyen	Interface correction + multi-API
Performance édition collaborative		Élevé	Moyen	Load testing précoce + Yjs
Complexité parsing PDF		Moyen	Élevé	Structure simple MVP puis itération
Coûts API IA élevés		Moyen	Moyen	Rate limiting + cache agressif
Adoption freemium faible		Élevé	Moyen	A/B testing + interviews utilisateurs
Sécurité données		Critique	Faible	Audit externe + bug bounty

6.0.1 Performance Technique

- Temps chargement dashboard : < 2 secondes
- Disponibilité (uptime) : > 99,5%
- Zéro incident sécurité majeur

Document établi par SDEN - Sami Dzogang Engineering

Version 1.0 - Octobre 2025