

# BandTrack – Cahier des charges version 3.1

## 1 – Objet et évolutions

BandTrack est une application destinée aux groupes de musique pour organiser leur activité : partage de suggestions de morceaux, suivi des répétitions, planification des prestations, gestion collaborative et centralisation des documents.

Cette version 3.1 corrige les faiblesses architecturales identifiées en v3.0 et introduit une stack technique robuste pour la production, une meilleure expérience utilisateur et des fonctionnalités avancées de collaboration temps réel.

## 2 – Architecture technique générale

### 2.1 – Stack applicative

- **Frontend mobile** : Capacitor + React + TypeScript, installable sur Android/iOS
- **Backend** : Python 3.11 avec FastAPI + Pydantic pour la validation
- **Base de données principale** : PostgreSQL 15+ avec extensions JSON
- **Cache et sessions** : Redis 7+ pour les sessions, cache applicatif et files d'attente
- **Stockage de fichiers** : MinIO (S3-compatible) avec CDN CloudFront/CloudFlare
- **Files d'attente** : Celery + Redis pour les tâches asynchrones
- **Communication temps réel** : WebSocket via Socket.IO

### 2.2 – Architecture déployée

[Mobile App] ↔ [Load Balancer] ↔ [API Gateway]

↓

[FastAPI Instances] ↔ [Redis] ↔ [PostgreSQL]

↓

↓

[Celery Workers] ↔ [MinIO] ↔ [Backup Storage]

### 2.3 – Conteneurisation

- Chaque service dans son conteneur Docker
- Orchestration via Docker Compose (dev) ou Kubernetes (prod)
- Volumes persistants pour PostgreSQL et MinIO
- Configuration via variables d'environnement

## 3 – Authentification et gestion des groupes

### 3.1 – Système d'authentification

- **JWT** avec refresh tokens (durée : 15min access / 7j refresh)
- **Mots de passe** : Argon2id (recommandation OWASP 2024)
- **Sessions** : stockées dans Redis avec TTL automatique
- **2FA optionnelle** : TOTP via authenticator apps
- **Authentification biométrique** : empreinte digitale, Face ID, reconnaissance vocale
- **Device binding** : association sécurisée device/utilisateur via clé cryptographique

## 3.2 – Gestion des groupes

- Création de groupe avec **code d'invitation unique** (8 caractères alphanumériques)
- **Rôles hiérarchiques** : Owner > Admin > Membre > Invité
- **Système de récupération** : 2+ admins obligatoires, transfert d'ownership possible
- **Limites** : 50 membres max par groupe, 5 groupes max par utilisateur
- **Isolation complète** : row-level security PostgreSQL + middleware API

## 3.3 – Gestion des invitations

- Lien d'invitation temporaire (24h) avec code unique
- Notification push/email lors de nouvelles invitations
- Approbation manuelle optionnelle par les admins

## 4 – Suggestions musicales

### 4.1 – Fonctionnalités de base

- **Champs** : Titre (obligatoire, 200 chars max), Artiste (optionnel, 100 chars), Lien YouTube/Spotify
- **Métadonnées automatiques** : extraction titre/artiste depuis APIs YouTube/Spotify
- **Vote système** : +1/-1 par membre, tri par score
- **Tags personnalisés** : genre, difficulté, priorité

### 4.2 – Fonctionnalités avancées

- **Import par lot** : playlist Spotify/YouTube
- **Suggestions IA** : recommandations basées sur l'historique du groupe
- **Recherche full-text** : PostgreSQL FTS + filtres avancés

## 5 – Répétitions et morceaux

### 5.1 – Gestion des morceaux

- **Versioning** : historique des modifications avec restore possible
- **Niveaux personnels** : curseur 0-10 + courbe de progression

- **Annotations temporelles** : commentaires liés à des timestamps sur les enregistrements
- **Partage sélectif** : niveaux visibles/masqués selon les préférences

## 5.2 – Enregistrements audio

- **Formats supportés** : MP3, WAV, M4A (conversion automatique)
- **Compression intelligente** : qualité adaptée selon usage (demo/archive)
- **Traitement asynchrone** : normalisation audio, suppression silences
- **Limite** : 50MB par fichier, 500MB total par groupe

## 5.3 – Collaboration temps réel

- **Synchronisation live** : modifications visibles instantanément
- **Curseurs partagés** : voir qui modifie quoi en temps réel
- **Résolution de conflits** : merge automatique ou choix manuel

# 6 – Prestations et planning

## 6.1 – Gestion des prestations

- **Typologie** : Concert, Répétition publique, Studio, Festival
- **Géolocalisation** : intégration Google Maps avec partage position
- **Setlist interactive** : réorganisation drag & drop, durées estimées
- **Post-mortem** : feedback collectif après prestation

## 6.2 – Analytics de groupe

- **Statistiques** : morceaux les plus joués, progression individuelle/collective
- **Rapports automatiques** : résumé mensuel d'activité
- **Objectifs** : définition de goals avec suivi de progression

# 7 – Partage de fichiers et documents

## 7.1 – Stockage et sécurité

- **Formats acceptés** : PDF, MP3, WAV, M4A, TXT, DOCX (scan antivirus)
- **Limites** : 100MB par fichier, 5GB total par groupe
- **Versioning** : 5 versions conservées par fichier
- **CDN** : mise en cache géographique pour accès rapide

## 7.2 – Organisation

- **Dossiers** : arborescence personnalisable par groupe

- **Tags** : système de marquage flexible
- **Recherche** : contenu full-text dans PDF et documents
- **Permissions** : lecture/écriture granulaire par membre

### 7.3 – Fonctionnalités avancées

- **Preview** : visualisation PDF/audio intégrée
- **Annotations collaboratives** : commentaires sur les partitions
- **Synchronisation offline** : téléchargement pour usage hors-ligne

## 8 – Calendrier collaboratif et notifications

### 8.1 – Planification intelligente

- **Création assistée** : suggestion créneaux libres selon disponibilités
- **Récurrence** : répétitions fixes avec gestion des exceptions
- **Intégration calendriers** : export iCal, sync Google Calendar/Outlook
- **Géolocalisation** : lieux fréquents suggérés automatiquement

### 8.2 – Système de présences

- **États** : Présent / Absent / Incertain / En retard
- **Notifications push** : rappels J-1 et H-2
- **Seuils d'alerte** : notification si <X% de présences confirmées
- **Remplacement** : système de musiciens de session

### 8.3 – Notifications multi-canal

- **Push mobile** : priorité haute/basse selon contexte
- **Email digest** : résumé hebdomadaire personnalisable
- **SMS** : urgences uniquement (facturable)
- **Préférences granulaires** : par type d'événement et canal

## 9 – Authentification biométrique et sécurité device

### 9.1 – Implémentation biométrique

- **Technologies supportées** :
  - Android : Fingerprint API + BiometricPrompt API
  - iOS : Touch ID + Face ID via LocalAuthentication
- **Fallback sécurisé** : code PIN 6 chiffres si biométrie indisponible
- **Stockage sécurisé** : Android Keystore / iOS Keychain pour clés cryptographiques

- **Validation côté serveur** : signature cryptographique pour prouver l'authenticité

## 9.2 – Flux d'authentification biométrique

1. Premier login : email/password traditionnel
2. Activation biométrie : génération paire de clés RSA sur device
3. Enregistrement : clé publique envoyée au serveur + hash device
4. Logins suivants : signature biométrique → JWT refresh → accès app
5. Fallback : PIN device ou re-authentification complète

## 9.3 – Sécurité renforcée

- **Device fingerprinting** : identification unique appareil (non-invasif)
- **Détection jailbreak/root** : blocage sur devices compromis
- **Certificate pinning** : protection contre attaques man-in-the-middle
- **Session binding** : JWT lié à l'empreinte device spécifique
- **Auto-logout** : après 30 jours d'inactivité ou changement device suspect

## 9.4 – Expérience utilisateur optimisée

- **Setup guidé** : tutoriel interactif configuration biométrie
- **Indicateurs visuels** : statut sécurité (biométrie active/inactive)
- **Gestion multi-device** : liste des appareils connectés avec révocation
- **Mode dégradé** : fonctionnement partiel si biométrie temporairement indisponible

# 10 – Interface utilisateur et expérience

## 9.1 – Navigation optimisée

- **Architecture** : 3 onglets principaux + menu hamburger
  - **Accueil** : dashboard personnalisé, prochaines échéances
  - **Morceaux** : suggestions, répétitions, setlists
  - **Planning** : calendrier, prestations, disponibilités
- **Menu secondaire** : Fichiers, Paramètres, Profil, Statistiques

## 9.2 – Personnalisation

- **Thèmes** : Clair, Sombre, Auto (selon système), + thèmes colorés
- **Dashboard personnalisable** : widgets drag & drop
- **Raccourcis** : actions fréquentes configurables
- **Accessibilité** : support lecteurs d'écran, contraste élevé

## 10.3 – Mode hors-ligne et synchronisation biométrique

- **Synchronisation intelligente** : données critiques en cache
- **Conflict resolution** : merge automatique au retour en ligne
- **Indicateurs visuels** : statut sync, données obsolètes
- **Stockage local** : 100MB max, purge automatique

## 11 – Sécurité et conformité

### 11.1 – Sécurité applicative

- **HTTPS obligatoire** : TLS 1.3, HSTS headers
- **Rate limiting** : par IP et par utilisateur (Redis)
- **Validation stricte** : Pydantic pour tous les inputs
- **Logs sécurisés** : audit trail, données sensibles masquées
- **Backup chiffré** : 3-2-1 strategy avec rotation automatique

### 11.2 – Protection des données et biométrie

- **RGPD compliant** : consentement explicite, droit à l'oubli
- **Chiffrement** : données au repos (AES-256) et en transit
- **Anonymisation** : statistiques sans données personnelles
- **Retention policy** : purge automatique après inactivité (2 ans)
- **Données biométriques** : JAMAIS stockées sur serveur, traitement local uniquement
- **Privacy by design** : templates biométriques chiffrés dans secure enclave device

### 11.3 – Monitoring et observabilité

- **Logs centralisés** : ELK Stack (Elasticsearch, Logstash, Kibana)
- **Métriques** : Prometheus + Grafana pour monitoring
- **Alerting** : PagerDuty/Slack pour incidents critiques
- **Health checks** : endpoints de santé pour tous les services

## 12 – Déploiement et infrastructure

### 12.1 – Environnements

- **Dev** : Docker Compose local, base SQLite pour rapidité
- **Staging** : réplique exacte de prod, données anonymisées
- **Production** : Kubernetes ou Docker Swarm selon échelle

## 12.2 – CI/CD Pipeline

- **Tests automatisés** :
  - Unitaires (>90% coverage)
  - Intégration (API endpoints)
  - End-to-end (Playwright/Cypress)
  - Performance (JMeter)
- **Déploiement** : Blue-green ou rolling selon criticité
- **Rollback** : automatique en cas d'erreur critique

## 12.3 – Scalabilité

- **Horizontal scaling** : FastAPI stateless + load balancer
- **Database sharding** : par groupe si nécessaire (>10k groupes)
- **CDN** : distribution géographique des assets statiques
- **Monitoring** : auto-scaling basé sur CPU/mémoire/latence

## 13 – Limites et contraintes techniques

### 13.1 – Limites système

- **Utilisateurs** : 50 membres max/groupe, 5 groupes max/utilisateur
- **Stockage** : 5GB/groupe, 100MB/fichier
- **API Rate limiting** : 1000 req/min/utilisateur, 10000/min/IP
- **Sessions concurrentes** : 3 devices max/utilisateur

### 13.2 – Performance targets

- **Latence API** : <200ms pour 95% des requêtes
- **Upload** : <30s pour fichiers 50MB
- **Sync offline** : <5s pour retour en ligne
- **Startup app** : <3s temps de démarrage
- **Authentification biométrique** : <1s validation empreinte/Face ID

## 14 – Roadmap et évolutions futures

### 14.1 – Phase 1 (MVP - 3 mois)

- Core features : groupes, morceaux, planning de base
- Architecture PostgreSQL + Redis
- Interface mobile responsive

- **Authentification biométrique** : implémentation empreinte digitale + Face ID

## **14.2 – Phase 2 (6 mois)**

- Notifications push
- Fichiers et stockage MinIO
- Collaboration temps réel

## **14.3 – Phase 3 (9 mois)**

- Analytics avancées
- IA/ML pour recommandations
- API publique pour intégrations tierces

Cette version 3.1 corrige les principales faiblesses architecturales tout en conservant la vision produit originale. L'accent est mis sur la robustesse technique, la scalabilité et une expérience utilisateur moderne.