Documentation technique

Plateforme événementielle : API, back-office, application mobile. Les adresses, identifiants et secrets sont masqués.

1. Architecture

Trois dépôts git séparés :

Dépôt	Rôle	Technologies
api	API REST, synchronisation WordPress, push, Wallet, billetterie	Node.js 20, TypeScript, Hono 4, Drizzle ORM, PostgreSQL, Zod, Better Auth
backoffice	Pilotage de l'événement et de l'app	Next.js 16 (App Router), React 19, Tailwind CSS 4, shadcn/ui
mobile	Application des visiteurs	React Native 0.88, Expo SDK 58, Expo Router, TanStack Query, Zustand, modules natifs Swift

2. Installation locale

API

```
cd backend
corepack enable                 # pnpm 10
pnpm install
createdb lsm_dev                # PostgreSQL local
cp apps/api/.env.example apps/api/.env
pnpm db:push                    # applique le schéma (apps/api/src/db/schema.ts)
pnpm api:seed
pnpm api:dev                    # http://localhost:9100
```

Back-office

```
cd backoffice
pnpm install
cp env.local.example .env.local
pnpm dev                        # http://localhost:3001
```

Application mobile

```
cd mobile
pnpm install
pnpm ios                        # Metro + simulateur iOS (build de développement)
```

3. Variables d'environnement (valeurs masquées)

Variable	Brique	Rôle
`PORT`, `DATABASE_URL`	API	Port d'écoute, connexion PostgreSQL
`BETTER_AUTH_SECRET`, `BETTER_AUTH_URL`	API	Signature des sessions ; refus des secrets faibles en production
`ADMIN_API_KEY`	API	Clé d'administration, comparée en temps constant
`CORS_ORIGIN`	API	Origines autorisées (back-office)
`STORAGE_DRIVER`, `S3_*`	API	Stockage local ou S3 / MinIO des médias
`WALLET_*`	API	Certificats Apple Wallet (chemins, hors dépôt)
`STRIPE_*`, `TICKET_QR_SECRET`	API	Paiement et signature des QR codes
`SMTP_*`	API	E-mails transactionnels
`WP_SYNC_USERNAME`, `WP_SYNC_APP_PASSWORD`	API	Facultatif ; sinon réglé dans le back-office et stocké chiffré (AES-256-GCM)
`NEXT_PUBLIC_API_URL`, `INTERNAL_API_URL`	Back-office	Adresse de l'API côté navigateur et côté serveur
`NEXT_PUBLIC_APP_URL`, `BETTER_AUTH_URL`	Back-office	Origine publique (cookies, contrôle d'origine)
`EXPO_PUBLIC_API_URL`, `APP_VARIANT`	Mobile	API en développement ; un build Release utilise toujours l'API de production

4. Routes de l'API

Plus de 150 routes, en trois familles. Extrait :

Méthode	Route	Accès	Description
GET	`/health`	public	Santé et version déployée
GET	`/store/event`	public	Édition active, compte à rebours, lieu, bannière
GET	`/store/sessions?day_id=`	public	Programme d'un jour
GET	`/store/speakers`, `/store/speakers/:id`	public	Intervenants
GET / POST	`/store/live`, `/store/live/chat`	lecture publique, écriture connectée	Live et chat
GET	`/store/notifications`	connecté	Notifications de l'utilisateur
GET	`/store/ticket/wallet/apple.pkpass`	connecté (Bearer)	Billet Apple Wallet
GET	`/admin/dashboard`	ADMIN	Indicateurs du back-office

PATCH	`/admin/event`	ADMIN	Réglages de l'édition et du lieu
POST	`/admin/notifications`	ADMIN	Envoi d'une notification push
*	`/api/auth/*`	public	Connexion, sessions (Better Auth)
POST	`/webhooks/stripe`	signature Stripe	Confirmation des paiements

Sécurité transverse : limitation de débit par IP, en-têtes HTTP durcis, CORS restreint, taille des corps bornée, validation Zod, protection SSRF des appels sortants (test automatisé).

5. Intégration WordPress

L'API lit le site par une API REST dédiée (`/wp-json/…/v2`) avec un mot de passe d'application WordPress. Une tâche planifiée (10 minutes par défaut, 5 au minimum) importe éditions, sessions, salles, intervenants, sponsors et agenda. Une empreinte du contenu évite toute écriture si rien n'a changé. Les réglages et l'état de la dernière synchronisation sont visibles dans le back-office.

6. Versionnement et déploiement

Les versions sont calculées par `semantic-release` à partir des messages de commit (Conventional Commits, vérifiés par commitlint) : `fix:` → correctif, `feat:` → version mineure. Aucun push ne déploie automatiquement.

```
Merge request          : test → build
push alpha / main      : test → build → release (manuel)
tag vX.Y.Z[-alpha.N]   : publish (manuel) → deploy (manuel)
```

Étape	Contenu
test	lint, typecheck, `pnpm audit` (non bloquant), détection de secrets
build	compilation TypeScript, image Docker de validation
release	version, CHANGELOG, tag git, release GitLab
publish	image Docker taguée dans le registre GitLab
deploy	SSH vers le VPS, `deploy.sh` : mise à jour du schéma (changements additifs seulement), remplacement du conteneur derrière Traefik, attente de `/health`

Application mobile : Xcode → Product → Archive → Distribute App → TestFlight / App Store Connect. Le numéro de build doit être supérieur au dernier envoyé.

7. Mode démonstration

La branche locale `demo/portfolio` (jamais fusionnée ni déployée) ajoute `DEMO_MODE=1` : données fictives, synchronisation WordPress coupée, tout appel réseau sortant hors `localhost` bloqué. Elle a servi à produire les captures de ce document.