

# Cahier des charges - Application Trading ETF Court Terme

## 1. Présentation générale

### 1.1 Objectif

Développer une application web progressive (PWA) d'analyse et de trading court terme d'ETF avec signaux d'achat/vente automatisés, optimisée pour compte-titres ordinaire.

### 1.2 Périmètre fonctionnel

- Analyse technique et fondamentale des ETF
- Génération de signaux de trading automatisés
- Système d'alertes push en temps réel
- Interface responsive (desktop/mobile/tablette)
- Gestion des risques et optimisation fiscale

### 1.3 Utilisateurs cibles

Investisseurs particuliers expérimentés souhaitant optimiser leurs stratégies court terme sur ETF.

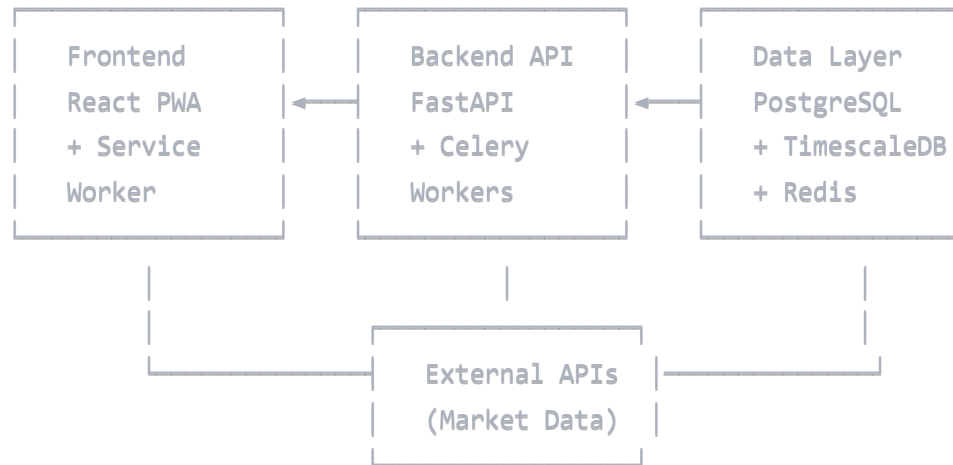
## 2. Architecture technique

### 2.1 Stack technologique

- **Backend** : Python 3.11+ / FastAPI / Uvicorn
- **Frontend** : React 18+ / TypeScript / Tailwind CSS
- **Base de données** : PostgreSQL + TimescaleDB (time-series)
- **Cache** : Redis 7+

- **Message Queue** : Celery + Redis
- **Authentication** : JWT + OAuth2
- **Déploiement** : Docker / Docker Compose

## 2.2 Architecture système



## 3. Fonctionnalités détaillées

### 3.1 Collecte et traitement des données

#### 3.1.1 Sources de données

- **APIs financières** : Alpha Vantage, Yahoo Finance, Quandl
- **Données macro** : FRED, BCE, INSEE
- **Bourses européennes** : Euronext, Deutsche Börse
- **Données ETF** : iShares, Vanguard, Amundi APIs

#### 3.1.2 Job scheduling (Celery)

- **Collecte temps réel** : Prix, volumes (toutes les 5 minutes)
- **Collecte journalière** : Données fondamentales, NAV
- **Collecte hebdomadaire** : Composition ETF, dividendes
- **Nettoyage des données** : Détection anomalies, normalisation

## 3.2 Moteur d'analyse

### 3.2.1 Indicateurs techniques

- **Moyennes mobiles** : SMA, EMA (20, 50, 200)
- **Oscillateurs** : RSI, MACD, Stochastique
- **Volatilité** : Bollinger Bands, ATR
- **Volume** : OBV, Volume Profile, VWAP
- **Momentum** : Rate of Change, Williams %R

### 3.2.2 Analyse des flux

- **Flux ETF** : Création/rachat de parts
- **Volumes relatifs** : Comparaison historique
- **Écarts NAV** : Premium/discount tracking
- **Corrélations** : Inter-ETF et avec indices

### 3.2.3 Analyse sectorielle

- **Rotation sectorielle** : Performance relative
- **Heat maps** : Visualisation performances
- **Momentum sectoriel** : Tendances émergentes
- **Cycles économiques** : Positionnement optimal

### 3.3 Génération de signaux

#### 3.3.1 Algorithmes de trading

- **Breakout** : Cassures de résistances/supports
- **Mean Reversion** : Retour à la moyenne
- **Momentum** : Suivi de tendance
- **Arbitrage statistique** : Pairs trading ETF

#### 3.3.2 Scoring et ranking

- **Score technique** : Agrégation indicateurs (0-100)
- **Score fondamental** : Métriques ETF (frais, liquidité)
- **Score de risque** : Volatilité, corrélations
- **Ranking final** : Combinaison pondérée

#### 3.3.3 Signaux d'action

- **BUY** : Signal d'achat avec niveau de confiance
- **SELL** : Signal de vente avec justification
- **HOLD** : Maintien position avec monitoring
- **WAIT** : Attente d'opportunité

### 3.4 Interface utilisateur

#### 3.4.1 Dashboard principal

- **Vue d'ensemble** : Marchés, indices, secteurs
- **Alertes actives** : Signaux en cours

- **Portefeuille** : Positions et P&L
- **Watchlist** : ETF surveillés

### 3.4.2 Analyse détaillée ETF

- **Graphiques interactifs** : Prix, volumes, indicateurs
- **Données fondamentales** : Composition, frais, performance
- **Historique des signaux** : Backtest et performance
- **Outils de comparaison** : Benchmarks et peers

### 3.4.3 Gestion des risques

- **Position sizing** : Calcul automatique
- **Stop-loss** : Niveaux recommandés
- **Diversification** : Répartition sectorielle
- **Exposition** : Risque global du portefeuille

## 3.5 Système d'alertes

### 3.5.1 Types d'alertes

- **Signaux de trading** : Achat/vente avec priorité
- **Événements** : Dividendes, rebalancement
- **Risques** : Dépassement de seuils
- **Actualités** : News impactantes

### 3.5.2 Canaux de notification

- **Push notifications** : Via Service Worker

- **Email** : Alertes importantes
- **SMS** : Signaux critiques (optionnel)
- **In-app** : Notifications temps réel

### 3.5.3 Personnalisation

- **Seuils configurables** : Score minimum, volatilité
- **Fréquence** : Temps réel, horaire, journalier
- **Filtres** : Secteurs, géographies, capitalisation
- **Priorités** : Niveaux d'importance

## 4. Spécifications techniques détaillées

### 4.1 Backend API (FastAPI)

#### 4.1.1 Endpoints principaux

python

#### *# Authentication*

POST /auth/login

POST /auth/register

POST /auth/refresh

#### *# Données de marché*

GET /market/etfs

GET /market/etf/{isin}

GET /market/sectors

GET /market/indices

#### *# Signaux et alertes*

GET /signals/active

GET /signals/history

POST /alerts/create

PUT /alerts/{id}/update

#### *# Portefeuille*

GET /portfolio/positions

POST /portfolio/transaction

GET /portfolio/performance

#### *# Utilisateur*

GET /user/profile

PUT /user/preferences

GET /user/watchlist

### **4.1.2 Modèles de données**

python

*# ETF*

```
class ETF:
    isin: str
    name: str
    sector: str
    currency: str
    ter: float
    aum: float
```

*# Signal*

```
class Signal:
    etf_isin: str
    signal_type: SignalType
    confidence: float
    price_target: float
    stop_loss: float
    timestamp: datetime
```

## 4.2 Frontend React

### 4.2.1 Structure des composants



```
src/  
├── components/  
│   ├── Dashboard/  
│   ├── Charts/  
│   ├── ETFDetail/  
│   ├── Portfolio/  
│   └── Alerts/  
├── hooks/  
├── services/  
├── utils/  
└── types/
```

#### 4.2.2 Gestion de l'état

- **Redux Toolkit** : État global
- **React Query** : Cache des données API
- **Context API** : Thème, authentification

#### 4.2.3 Progressive Web App

- **Service Worker** : Cache offline, notifications
- **Manifest** : Installation native
- **Responsive design** : Mobile-first

### 4.3 Graphiques et visualisations

#### 4.3.1 Librairies

- **Chart.js** : Graphiques standards
- **D3.js** : Visualisations custom

- **React-Chartjs-2** : Intégration React

#### 4.3.2 Types de graphiques

- **Candlestick** : Prix OHLC avec volumes
- **Line charts** : Indicateurs techniques
- **Heat maps** : Performance sectorielle
- **Scatter plots** : Corrélations

### 4.4 Optimisations performance

#### 4.4.1 Cache strategy (Redis)

- **Time-series data** : TTL 5 minutes
- **Static data** : TTL 1 heure
- **User sessions** : TTL 24 heures
- **Computed signals** : TTL 15 minutes

#### 4.4.2 Database optimization

- **Indexes** : Sur ISIN, timestamp, signal\_type
- **Partitioning** : Tables time-series par mois
- **Connection pooling** : SQLAlchemy async
- **Query optimization** : N+1 queries évitées

## 5. Sécurité et conformité

### 5.1 Authentification et autorisation

- **JWT tokens** : Access (15min) + Refresh (7 jours)

- **Rate limiting** : 100 req/min par utilisateur
- **CORS** : Domaines autorisés uniquement
- **Input validation** : Pydantic models

## 5.2 Protection des données

- **Chiffrement** : HTTPS/TLS 1.3
- **Données sensibles** : Hashage bcrypt
- **Logs** : Pas de données personnelles
- **RGPD** : Consentement et export données

## 5.3 Conformité financière

- **Disclaimer** : Pas de conseil en investissement
- **Risk warning** : Risques du trading
- **Data accuracy** : Limitation de responsabilité
- **Audit trail** : Traçabilité des signaux

# 6. Déploiement et monitoring

## 6.1 Infrastructure

- **Containerisation** : Docker multi-stage
- **Orchestration** : Docker Compose
- **Reverse proxy** : Nginx
- **SSL** : Let's Encrypt

## 6.2 Monitoring

- **Application** : Prometheus + Grafana
- **Logs** : ELK Stack (Elasticsearch, Logstash, Kibana)
- **Uptime** : Pingdom/UptimeRobot
- **Alerting** : Slack/Discord webhooks

## 6.3 Backup et récupération

- **Database** : Backup quotidien automatique
- **Redis** : Persistence AOF
- **Code** : Git avec CI/CD
- **Disaster recovery** : RTO < 1h, RPO < 15min

## 7. Planning et livrables

### 7.1 Phase 1 - MVP (8 semaines)

- Setup infrastructure et CI/CD
- Backend API de base
- Frontend avec dashboard simple
- Collecte de données basique
- Authentification utilisateur

### 7.2 Phase 2 - Analyse (6 semaines)

- Moteur d'analyse technique
- Génération de signaux simples
- Graphiques interactifs
- Système d'alertes basique

### 7.3 Phase 3 - Optimisation (4 semaines)

- Algorithmes avancés
- PWA et notifications push
- Optimisations performance
- Tests et debugging

### 7.4 Phase 4 - Production (2 semaines)

- Déploiement production
- Monitoring et alerting
- Documentation utilisateur
- Formation et support

## 8. Critères de succès

### 8.1 Performance technique

- **Latence API** : < 200ms (95e percentile)
- **Disponibilité** : > 99.5%
- **Temps de chargement** : < 3s first paint
- **Précision signaux** : > 60% sur 3 mois

### 8.2 Expérience utilisateur

- **Mobile friendly** : Score Lighthouse > 90
- **Notifications** : Délivrance > 95%
- **Uptime** : Alertes < 5min de délai
- **Satisfaction** : NPS > 50

### 8.3 Business metrics

- **Adoption** : 80% des signaux consultés
- **Rétention** : 70% utilisateurs actifs à 30 jours
- **Engagement** : 5+ actions par session
- **Performance** : Tracking vs benchmarks