

Spécification Technique - MarketDataCollector (Mise à jour)

1. Architecture générale

- Service d'arrière-plan (« BackgroundService ») intégré dans l'application SymbolDownloader
- Déploiement dans le même App Service Azure
- WebSocket unique multiplexé pour récupérer les bougies clôturées Binance
- Pas de stockage disque, fonctionnement 100% en mémoire

2. Comportement au démarrage

- Lecture des timeframes depuis `appsettings.json` (par défaut : M1, M5, M15)
- Appel direct à la couche SymbolDownloader pour les symboles actifs
- Construction de l'URL WS Binance multiplexée
- Ouverture d'une session WebSocket
- Initialisation des buffers FIFO par (symbol, timeframe)

3. Connexion WebSocket Binance

- Une seule connexion pour jusqu'à 1024 streams
- Messages traités uniquement si `x=true` (bougie clôturée)
- En cas d'échec, rotation planifiée chaque 12h relance la session

4. Pipeline de traitement kline

1. Désérialisation JSON en objet Candle
2. Validation de fermeture
3. Mise à jour des buffers FIFO (Volume, Close, Volatility)
4. Calcul des ratios et détection des critères (VolumeSpike, PriceJump, VolatilitySpike)
5. Si au moins un critère est rempli : création de `MarketDataChangeEvent`
6. Publication vers SignalR (dans le groupe du symbole concerné)
7. Mise à jour de la dernière bougie connue

5. SignalR - Publication des événements

- Utilisation du **même Hub SignalR que SymbolDownloader**
- Chaque symbole correspond à un **groupe SignalR**
- Les clients doivent appeler `JoinGroup(symbol)` pour recevoir les événements
- L'événement est envoyé en mode **fire-and-forget**, sans garantie de réception

6. Buffers circulaires internes

- Taille X paramétrable (par défaut : 5)
- Stockage en RAM des séries : Volume[], Close[], Volatility[]
- Aucune persistance disque ni backup

7. Gestion de la rotation (12h)

- Timer ou cron interne
- Si la récupération des symboles échoue : rotation annulée
- Recréation de la session WebSocket avec vidage des buffers
- Tolerance totale à la perte de données entre deux sessions

8. Configuration via appsettings.json

```
{
  "Timeframes": ["M1", "M5", "M15"],
  "VolumeSpike": {
    "Window": 5,
    "RatioThreshold": 1.5,
    "MinAbsoluteVolume": 10.0
  },
  "PriceJump": {
    "Window": 5,
    "RatioThreshold": 1.5
  },
  "VolatilitySpike": {
    "Window": 5,
    "RatioThreshold": 2.0
  },
  "RotationIntervalHours": 12,
  "ReconnectTimeoutSeconds": 10,
  "MaxSymbols": 200
}
```

9. Observabilité

- Logs structurés pour chaque événement détecté
- En option : expose un `/healthz/marketdata` retournant l'horodatage de la dernière bougie reçue
- Intégration possible App Insights ou Prometheus (métriques à définir)

10. Limitations assumées

- Pas de garantie de livraison SignalR (pas de retry)
 - Pas de sécurité sur le Hub pour le POC
 - Pas de logique d'abonnement dynamique (gestion manuelle des groupes côté client)
 - Pas de traitements statistiques complexes ou agrégations cross-symbols
-