

Spécification Technique - US03 : Détection de changements sur candles

Objectif

Mettre en place un système de détection de changements significatifs sur les bougies Binance pour les symboles actifs. Trois types de changements sont visés : `VolumeSpike`, `PriceJump`, `VolatilitySpike`.

Règles métier

1. Détection active pour chaque timeframe

- Lorsqu'un `CandleReceivedEvent` est reçu :
- Le `CandleSnapshot` est ajouté au buffer circulaire (20 par défaut) pour le couple (`symbol`, `timeframe`)
- Si le buffer contient au moins N bougies :
 - Calculer les indicateurs (moyennes glissantes, ratios)
 - Appliquer les règles de détection propres au timeframe
 - Si au moins une règle est satisfaite :
 - Créer un `MarketDataChangeDetectedEvent`
 - Le publier via SignalR (groupe par symbole)

2. Mise à jour du `SymbolMetadata` pour les bougies `H1`

- Si l'événement reçu correspond à un intervalle `H1` :
- Mettre à jour la propriété `CurrentPrice` du `SymbolMetadata` concerné avec la valeur de `Close`

Composants à mettre à jour / créer

`InMemoryCandleBufferRegistry` (ajouts)

- `CandleSnapshot? GetLast(string symbol, KlineInterval interval)`
- `bool TryGetBuffer(string symbol, KlineInterval interval, out IReadOnlyList<CandleSnapshot> buffer)`

Event

```
public class CandleReceivedEvent : IEvent
{
    public CandleSnapshot Snapshot { get; set; }
}
```

Handler : CandleReceivedEventHandler

Responsabilités :

- Récupérer le buffer circulaire via `ICandleBufferRegistry`
- Ajouter le nouveau snapshot
- Si le buffer est complet ($\geq N$) :
- Détecter les changements
- Publier un `MarketDataChangeDetectedEvent` si besoin
- Si intervalle == H1 :
- Mettre à jour `CurrentPrice` dans `SymbolMetadata`

Service : IChangeDetectionService

Interface à définir avec :

- `List<string> Detect(CandleSnapshot current, IReadOnlyList<CandleSnapshot> history)`
- `object BuildContext(...)` (pour le champ `Context` de l'événement)

Configuration

Définie dans `appsettings.json` :

- `BufferWindow` : taille du buffer
- Seuils pour chaque critère :
- `VolumeSpike` : `RatioThreshold`, `MinAbsoluteVolume`
- `PriceJump` : `RatioThreshold`
- `VolatilitySpike` : `RatioThreshold`

Événements émis

MarketDataChangeDetectedEvent

Champs :

- Symbol
- Timeframe
- Timestamp
- ChangeTypes (ex: ["VolumeSpike"])
- Candle (snapshot)
- Context (moyennes, ratios)
- Source = "MarketDataCollector"