Cahier des charges

Système : BrokerX — Plateforme de courtage en ligne pour investisseurs particuliers

Contexte

Le secteur des services financiers connaît une transformation rapide, portée par la numérisation des échanges, l'exigence de transparence réglementaire et l'augmentation du volume d'ordres traités en temps réel. Dans ce contexte, les courtiers en ligne doivent offrir à leurs clients des plateformes fiables, performantes et sécurisées, capables de traiter un grand nombre de transactions tout en garantissant la conformité aux normes de surveillance et de gestion du risque.

Le projet BrokerX — Plateforme de courtage en ligne pour investisseurs particuliers s'inscrit dans cette dynamique. Il vise à concevoir et à développer une solution logicielle qui permet à des clients de détail (investisseurs individuels ou petites entreprises) d'accéder à une interface de courtage moderne. Les utilisateurs doivent pouvoir créer un compte, vérifier leur identité, approvisionner un portefeuille virtuel, et interagir avec les marchés financiers simulés en passant des ordres d'achat ou de vente (marché ou limite). Le système offre également la possibilité de consulter en temps réel les carnets d'ordres, les cotations, ainsi que l'évolution des portefeuilles et positions.

Du point de vue opérationnel, BrokerX intègre plusieurs domaines critiques :

- **Ordres et exécutions** : réception, routage et appariement interne avant la connexion à des bourses simulées.
- **Portefeuilles et règlements** : mise à jour des positions, génération de rapports et clôture journalière.
- **Conformité et surveillance** : application de contrôles pré-trade (pouvoir d'achat, restrictions de prix, interdictions de ventes à découvert) et détection post-trade d'activités suspectes.
- **Observabilité et audit** : traçabilité complète via journaux immuables, métriques de performance et traçage distribué.

L'architecture de BrokerX est conçue pour évoluer progressivement : d'une base monolithique pour un prototype rapide, elle migre vers une architecture microservices puis vers une approche orientée événements. Chaque étape augmente les capacités de performance (latence, débit),

de disponibilité et de résilience du système. De plus, l'intégration de mécanismes modernes d'observabilité permet de surveiller et d'analyser en continu la santé du système selon les quatre signaux fondamentaux (latence, trafic, erreurs, saturation).

Problème à résoudre

- Développer une plateforme de courtage sécurisée permettant aux clients de :
- Passer des ordres (marché, limite),
- Consulter leurs portefeuilles,
- Recevoir exécutions et confirmations,
- Effectuer le règlement et la compensation des transactions.

La plateforme doit se connecter à un ou plusieurs marchés boursiers simulés et satisfaire à des objectifs non fonctionnels (disponibilité, latence, débit, observabilité).

1. Vision et Contexte

Développer une plateforme de courtage sécurisée permettant aux clients de :

- Passer des ordres
- Consulter leurs portefeuilles
- Recevoir exécutions et confirmations
- Effectuer le règlement et la compensation des transactions

La plateforme doit se connecter à un ou plusieurs marchés boursiers simulés et satisfaire à des objectifs non fonctionnels (disponibilité, latence, débit, observabilité).

Parties prenantes

- **Clients** : utilisateurs via interface web/mobile.
- **Opérations Back-Office** : gestion des règlements, supervision.
- **Conformité / Risque** : surveillance pré- et post-trade.
- **Fournisseurs de données de marché** : cotations en temps réel.
- **Bourses externes** : simulateurs de marché pour routage d'ordres.

2. Périmètre fonctionnel

Domaines principaux (bounded contexts DDD)

1. **Client & Comptes**
    - balances, authentification et autorisation, intégrations AML.

2. **Ordres & Appariement**
    - Routage broker + moteur d'appariement interne.

3. **Données de marché**
    - Quotations, transactions, OHLC (Open, High, Low, Close), diffusion en continu aux clients.

4. **Portefeuilles & Positions**

Fonctionnalités

- Inscription/login utilisateur (MFA optionnelle), simulation de financement.
- Consultation portefeuille et solde.
- Abonnement aux données de marché (liste de titres).
- Placement, modification et annulation d'ordres ; carnet d'ordres par symbole.
- Routage vers moteur interne (phase 1), puis vers bourse externe simulée (phase 2+).
- Mise à jour des positions suite aux exécutions ; confirmations et notifications.
- Rapports quotidiens (EOD snapshot), traçabilité complète (audit trail).

3. Exigences de qualité (ou non-fonctionnelles)

Performance (par phases évolutives)

- Latence P95 (ordre → ACK) :
    - ≤ 500 ms (monolithique)
    - ≤ 250 ms (microservices)
    - ≤ 100 ms (architecture événementielle)

- Débit (throughput) :
    - ≥ 300 ordres/s (monolithique)
    - ≥ 800 ordres/s (services)
    - ≥ 1200 ordres/s (event-driven)

Disponibilité (stimé)

- 90,0 % (monolithique)
- 95,5 % (microservices)
- 99,9 % (event-driven)

Observabilité

- Logs + métriques (4 Golden Signals) dès la phase 2.
- Traces distribuées à partir de la phase 3.

3. Conformité et Opérations

- Idempotence pour soumission d'ordres.
- Garantie d'**exactly-once** (ou effectively-once) pour règlements.
- Journal d'audit immuable (flux append-only).
- Contrôles réglementaires (pré- et post-trade).

4. Contraintes techniques

- **Les plateformes/langages serveur privilégiées** : Java, C#, Go, Rust ou C++
- **Python et Javascript/Typescript** sont fortement déconseillées (sévèrement évalué par soucis de qualité et performance)

5. Cas d’utilisation

UC-01 — Inscription & Vérification d’identité

Objectif :
Permettre à un nouvel utilisateur de créer un compte sur la plateforme en fournissant ses informations personnelles, de vérifier son identité selon les exigences réglementaires (KYC/AML) et d’activer son accès à la plateforme. Ce cas établit la relation de confiance initiale entre l’utilisateur et BrokerX.

Acteur principal : Client
Déclencheur : L’utilisateur souhaite créer un compte.
Pré-conditions : Aucune.
Postconditions (succès) : Compte créé en état *Pending* et changer à *Active* après validation.
Postconditions (échec) : Compte non créé ou marqué *Rejected* avec raison.

Flux principal (succès)

1. Le Client fournit email/téléphone, mot de passe, données personnelles requises (nom, adresse, date de naissance).

2. Le Système valide le format, crée un compte en *Pending*, envoie un lien de vérification email/SMS.

3. Le Client confirme le lien OTP (one-time passwords)/MFA (multi-factor authentication).

4. Le Système passe le compte à *Active* et journalise l’audit (horodatage, empreinte des documents).

Alternatifs / Exceptions
A1. **Vérif email non complétée** : compte reste *Pending* (rappel, expiration après X jours).
E1. **Doublon** (email/tel déjà utilisés) : rejet, proposition de récupération de compte.

UC-02 — Authentification & MFA

Objectif :

Garantir un **accès sécurisé** à la plateforme en permettant aux clients de s'authentifier avec identifiant/mot de passe et, le cas échéant, via un mécanisme de multi-facteurs (OTP, TOTP, WebAuthn). Ce cas protège les comptes contre les accès non autorisés.

Acteur principal : Client (investisseur particulier souhaitant créer un compte).
Déclencheur : Le Client se connecte.
Préconditions : Compte *Active*.
Postconditions (succès) : Session valide établie (token JWT/opaque) avec rôle *Client*.
Postconditions (échec) : Aucune session créée.

Flux principal

1. Le Client saisit identifiant/mot de passe.
2. Le Système valide (anti-brute force, IP reputation).
3. Si MFA activée/obligatoire, le Système demande OTP (TOTP/SMS/WebAuthn).
4. Le Client valide l'OTP.
5. Le Système émet le jeton de session, enregistre l'audit (IP, device, succès).

Alternatifs / Exceptions

A1. **Appareil de confiance** : MFA step-up seulement pour opérations sensibles.
E1. **MFA échoue** (3 tentatives) : verrouillage temporaire.
E2. **Compte suspendu** : accès refusé, contact support.

UC-03 — Approvisionnement du portefeuille (dépôt virtuel)

Objectif:
Donner aux utilisateurs la possibilité de **créditer leur portefeuille virtuel** en effectuant des dépôts simulés, afin de disposer de liquidités nécessaires pour placer des ordres d'achat. Ce cas assure la disponibilité des fonds pour les opérations boursières.

Acteur principal : Client
Secondaires : Service Paiement Simulé / Back-Office
Déclencheur : Le Client crédite son solde en monnaie fiduciaire simulée.
Préconditions : Compte *Active*.
Postconditions (succès) : Solde augmenté, écriture comptable ajoutée (journal immuable).
Postconditions (échec) : Solde inchangé.

Flux principal

1. Le Client saisit le montant.
2. Le Système valide limites (min/max, anti-fraude).
3. Le Système crée une transaction *Pending*.
4. Le Service Paiement Simulé répond *Settled*.
5. Le Système crédite le portefeuille, journalise et notifie.

Alternatifs / Exceptions
A1. **Paiement async** : passe *Pending*, solde crédite à confirmation.
E1. **Paiement rejeté** : état *Failed*, notification avec motif.
E2. **Idempotence** : si retry reçu avec même idempotency-key, renvoyer le résultat précédent.

UC-04 — Abonnement aux données de marché

Objectif:
Offrir aux clients un accès en **temps réel aux cotations et carnets d'ordres** pour les instruments suivis. Ce cas permet aux investisseurs de prendre des décisions éclairées grâce à des données actualisées.

Acteur principal : Client
Secondaires : Fournisseur de données de marché simulé
Déclencheur : Le Client ouvre la vue Marché ou s'abonne à des symboles.
Préconditions : Session valide.
Postconditions (succès) : Flux temps réel établi (WebSocket/Server-Sent Events (SSE)), latence < 200 ms.
Postconditions (échec) : Aucun flux établi.

Flux principal

1. Le Client demande l'abonnement à une liste de symboles.

2. Le Système autorise (quotas, rate-limit).

3. Le Système ouvre un canal (WS/SSE) et pousse cotations/ordre book snapshots + diff.

4. Le Client reçoit updates (top of book, OHLC, trades).

Alternatifs / Exceptions
A1. **Débit élevé** : passage en mode *throttled* (agrégation, batch).
E1. **Source indisponible** : bascule vers flux dégradé (snapshots moins fréquents).
E2. **Stale data** : alerte UI, bannière "données retardées".

UC-05 — Placement d'un ordre (marché/limite) avec contrôles pré-trade

Objectif:
Permettre aux clients de **soumettre des ordres d'achat ou de vente** (marché ou limite), qui seront validés par des contrôles pré-trade et insérés dans le moteur d'appariement. Ce cas constitue le cœur fonctionnel de la plateforme de courtage.

Acteur principal : Client
Secondaires : Moteur de Règles Pré-trade, Gestion des Risques, Comptes/Portefeuilles
Déclencheur : Le Client soumet un ordre.
Préconditions : Session valide, portefeuille existant.
Postconditions (succès) : Ordre accepté (ACK) et placé dans le carnet interne.
Postconditions (échec) : Ordre rejeté avec raison.

Flux principal (succès)

1. Le Client renseigne symbole, sens (Achat/Vente), type (Marché/Limite), quantité, prix (si limite), durée (DAY/IOC…).

2. Le Système normalise les données et horodate l'opération (timestamp système en UTC avec millisecondes ou nanosecondes).

3. **Contrôles pré-trade** :

    - Pouvoir d'achat / marge disponible,

    - Règles de prix (bandes, *tick size*),

    - Interdictions (short-sell si non autorisé),

    - Limites par utilisateur (taille max, notionals),

    - Sanity checks (quantité > 0, instrument actif).

4. Si OK, le Système attribue un OrderID, persiste, achemine au Moteur d'appariement interne.

Alternatifs / Exceptions
A1. **Type Marché** : prix non requis, routage immédiat.
A2. **Durée IOC/FOK** : logique spécifique au matching (voir UC-07).
E1. **Pouvoir d'achat insuffisant** : **Reject** avec motif.
E2. **Violation bande de prix** : **Reject**.
E3. **Idempotence** : même *clientOrderId* → renvoyer résultat précédent.

UC-06 — Modification / Annulation d'un ordre

Objectif:
Offrir la possibilité de modifier ou d'annuler un ordre actif dans le carnet tant qu'il n'est pas totalement exécuté. Ce cas donne de la flexibilité aux clients pour gérer leurs stratégies de trading.

Acteur principal : Client
Secondaires : Moteur d'appariement, Portefeuilles
Déclencheur : Le Client modifie ou annule un ordre *Working*.
Préconditions : Ordre existant, non entièrement exécuté, modifiable.
Postconditions (succès) : Ordre mis à jour ou annulé; journal d'audit.
Postconditions (échec) : Aucune modification.

Flux principal

1. Le Client envoie *Cancel* ou *Replace* (nouvelles valeurs : quantité, prix).
2. Le Système verrouille l'ordre (optimisme/pessimisme), vérifie état.
3. Si *Replace*, repasse contrôles pré-trade; si *Cancel*, retire du carnet.
4. Retourne **Cancel/Replace ACK**.

Alternatifs / Exceptions
A1. **Exécution concurrente partielle** : seules quantités restantes modifiables.
E1. **Ordre déjà exécuté/annulé** : **Reject** (avec dernier état).
E2. **Conflit de version** : retry avec dernière révision.

UC-07 — Appariement interne & Exécution (matching)

Objectif:
Assurer l'**exécution automatique des ordres** en interne selon les règles de priorité (prix/temps) en rapprochant acheteurs et vendeurs. Ce cas fournit la mécanique centrale de traitement des transactions sur la plateforme.

Acteur principal : Moteur d'appariement interne
Secondaires : Données de Marché, Portefeuilles
Déclencheur : Nouvel ordre arrive dans le carnet.
Préconditions : Carnet maintenu (prix/temps), règles de priorité définies.
Postconditions (succès) : Transactions générées (partielles possibles), état d'ordre mis à jour.
Postconditions (échec) : Ordre reste *Working* (pas de contrepartie).

Flux principal

1. Le Moteur insère l'ordre dans le carnet (Buy/Sell).
2. Il recherche la meilleure contrepartie (price-time priority).
3. Si match, crée une ou plusieurs exécutions (fills), met à jour quantités.
4. Émet événements **ExecutionReport** (Fill/Partial Fill).
5. Met à jour top-of-book, publie update marché.

Alternatifs / Exceptions
A1. **Ordre marché sans liquidité** : exécution partielle possible, reste non exécuté → voir UC-08 (routage).
A2. **IOC/FOK** : IOC exécute le possible puis annule le reste; FOK exécute tout sinon annule.
E1. **Incohérence carnet** (rare) : déclenche rollback segmentaire et alerte Ops.

UC-08 — Confirmation d’exécution & Notifications

Objectif:
Notifier les clients de l’état final de leurs ordres (exécuté partiellement ou totalement, rejeté), en fournissant des informations précises et traçables. Ce cas garantit la transparence et la confiance dans les transactions.

Acteur principal : Système
Secondaires : Client, Back-Office
Déclencheur : Réception d’un *ExecutionReport*.
Préconditions : Ordre existant.
Postconditions (succès) : Client notifié (UI push/email), état ordre mis à jour.
Postconditions (échec) : Notification peut échouer mais l'exécution reste valide.

Flux principal

1. Le Système met à jour l’état de l’ordre (Partial/Filled).
2. Crée un enregistrement d’exécution (prix, qty, frais).
3. Notifie le Client (temps réel).
4. Journalise l’audit (horodatage, source).

Alternatifs / Exceptions
A1. **Agrégation** : pour multiples partial fills, regrouper notifications.
E1. **Échec de push** : retry, puis fallback email.