KRX↔NXT Cross-Venue Arbitrage — System Blueprint (V1)

Prepared: August 23, 2025
Timezone: Asia/Seoul

Executive Summary

Purpose. A complete blueprint for building a simple, robust cross-venue arbitrage engine that captures spreads between KRX (exchange) and NXT (alternative trading system). This is written for readers who were not part of the original discussion.

Philosophy. Keep V1 minimal and deterministic. Trade tiny slices, always end flat, obey Kiwoom OpenAPI+ constraints, and instrument the core loop for safety and iteration. Prioritize stability over throughput.

Scope. Session timing, fees/thresholds, data ingestion, spread evaluation, throttling, routing, execution state machine, telemetry, GUI wireframe, config schema, and a runbook.

Scope & Success Criteria

- **Trading style:** Pure cross-venue arb on the same stock (KRX vs NXT) intraday. No overnight risk.
- **Sizing:** Start with micro-slices (1 share per leg per attempt). In V1: max **1 active slice per symbol**; **1–2 symbols** concurrently.
- **Success (initial):**
  (a) Flat and safe—no runaway exposure.
  (b) Low reject/timeout rates.
  (c) Hedge within ~1s when needed.
  (d) Net positive PnL after fees on feasible 1-tick edges.
  (e) Clear telemetry for iteration.

Key Constraints & Assumptions

- **Platform:** Windows with Kiwoom OpenAPI+. Keep all Kiwoom calls in a single process/thread (32-bit behavior/constraints).
- **Rate limits:**
  - Orders ≤ **5/sec** globally (cancels count).
  - Data requests ≤ **5/sec** (separate bucket).
  - Real-time registration ≤ **100 symbols per screen number**.

- **Venues & orders:** Direct routing only. KRX uses normal SendOrder(). NXT uses ATS order types (e.g., 21=buy, 22=sell). NXT mid-price uses **hoga=29** with **price=0**. SOR disabled for production. AL (통합) feed not used.
- **Edge requirement:** Global default ≥ **1 tick net after fees** (per-symbol overrides later).
- **Concurrency:** Start with **1–2 symbols** concurrently; **1 outstanding slice per symbol** in V1.

Sessions & Trading Window

- **Trade only in overlap** of:
  - **KRX day:** 09:00–15:20
  - **NXT Main:** 09:00:30–15:20
- **Guard window (engine armed): 09:00:32 → 15:19:50**. Outside this window the engine is disarmed.
- **Session signals:** Subscribe to NXT FID-215 (P…V) to react to session changes; use wall-clock guard for safety.
- **Out of scope (V1):** NXT Pre and After sessions.

Fees & Thresholds (Baseline)

- **Baseline fees (editable in config):**
  - **KRX broker:** ~0.015% (1.5 bps) per side
  - **NXT broker:** ~0.0145% (1.45 bps) per side
  - **NXT regulatory/agency example:** ~0.0031833% (0.31833 bps) per side
- **Thresholding:** Required edge ≥ **fees(buy+sell) + buffer** (buffer defaults to **+1 tick net**; may tune per symbol).

Architecture Overview (Modules)

- **SessionState** — Guards trading by overlap times + FID-215 signals.
- **SymbolMap** — Maps KRX code ↔ NXT code (no AL).
- **MarketData** — Per-venue L1 (bid/ask/size); screen sharding; tiny per-symbol snapshot; DIRTY tracking.
- **SpreadEngine** — Micro-batch (~10 ms); two candidate edges; fee/tick aware; size checks; **cooldown (100–200 ms)**.
- **Throttler** — Global buckets: **orders 5/sec**, **queries 5/sec**; preserve **2 tokens** for cancel/hedge.
- **Router** — Deterministic venue & order-style choice (take rich; post cheap; SOR off).
- **ExecutionGateway** — Send/cancel; correlate TR acks + Chejan fills; enforce **cancel-then-new** for type changes.
- **PairManager** — Per-pair timers (**t_hedge=1000 ms**), escalation (limit→IOC/market), always flatten.
- **Risk** — Per-symbol & global concurrency caps; simple freeze/mute hooks (minimal in V1).
- **FeesPnL** — Per-leg fees; per-pair & session PnL (KRW & bps).
- **Telemetry** — SLOs, rejects/timeouts, orders/sec, unhedged time; Slack (fills + major red flags).
- **GUI** — Ops view; Active Symbols, Pair Monitor, Event Feed, Config; Reports view post-session.

End-to-End Flow Diagram (Text)

```
┌──────────────────────────────────────────────────────────────────────────────┐
│ 0) LAUNCH → LOGIN → ARM (must be in this exact order)                        │
└──────────────────────────────────────────────────────────────────────────────┘
[Start GUI] → [Load config.yaml] → [Init Kiwoom] → [CommConnect() login]
   ├─ if login fails → show error & retry
   └─ post-login → GetLoginInfo; open Account-PW window if needed; continue
[Session bootstrap (disarmed)] → subscribe FID-215 heartbeat
[Overlap passes?] (KRX day ∧ NXT Main) → [ARM TRADING (09:00:32→15:19:50)]

┌──────────────────────────────────────────────────────────────────────────────┐
│ 1) FEEDS & STATE (per-venue L1; no AL; sharded screens)                      │
└──────────────────────────────────────────────────────────────────────────────┘
[Register L1 feeds] → shard ~200 symbols across 3–4 screens (≤100 per screen)
[Per-symbol snapshot] {krx_bid, krx_ask, krx_sz, nxt_bid, nxt_ask, nxt_sz,
t_krx, t_nxt}
On each tick: update snapshot; if best price changed → mark DIRTY

┌──────────────────────────────────────────────────────────────────────────────┐
│ 2) MICRO-BATCH DECISION LOOP (every ~10 ms)
│
└──────────────────────────────────────────────────────────────────────────────┘
For each DIRTY symbol not in cooldown:
  Compute A) Buy KRX ask vs Sell NXT bid  and  B) Buy NXT ask vs Sell KRX bid
  Require visible size ≥ slice; edge ≥ fees(buy+sell)+buffer (≥1 tick net)
  If no → arm cooldown 100–200 ms; If yes → emit SIGNAL(symbol, bestPair, qty=1)

┌──────────────────────────────────────────────────────────────────────────────┐
│ 3) ADMISSION & ROUTING
│
└──────────────────────────────────────────────────────────────────────────────┘
[SIGNAL] → Risk (armed? caps ok?) → Throttler (orders bucket: 5/s; ≥4 tokens
free)
  If admitted → Router chooses direct venues & styles:
    Rich side = TAKE (IOC/Market; price=0)
    Cheap side = POST (Limit or NXT Mid; hoga=29; price=0)
  Else → queue or drop (log reason)

┌──────────────────────────────────────────────────────────────────────────────┐
│ 4) EXECUTION PIPE (TR ack + Chejan lifecycle)
│
└──────────────────────────────────────────────────────────────────────────────┘
[ENTRY_TAKE_SENT] → TR 주문번호? If none → REJECT → cooldown
```

```
If accepted → Chejan 접수/체결… On fill → [HEDGE_POST_SENT]
  If hedge fills before t_hedge=1000 ms → [PAIRED_DONE → Flat]
  Else at t_hedge → [CANCEL_POST_SENT] → [HEDGE_IOC_SENT] → [PAIRED_DONE → Flat]

┌──────────────────────────────────────────────────────────────────────────────┐
│ 5) TELEMETRY / PnL / ALERTS
│
└──────────────────────────────────────────────────────────────────────────────┘
SLO tiles: Tick→Signal p95 < 25 ms; Signal→Send p95 < 15 ms; Send→Ack p95 < 150
ms
Orders/sec utilization meter (auto-pause new entries ≥80% for 5s; advisory)
Slack (fills + major red flags only): BUY_FILL, SELL_FILL, PAIR_DONE,
  AUTO-PAUSE ON/OFF, HEDGE TIMEOUT, REJECT SPIKE

┌──────────────────────────────────────────────────────────────────────────────┐
│ 6) DISARM / SHUTDOWN
│
└──────────────────────────────────────────────────────────────────────────────┘
At 15:19:50 → disarm new entries; finish hedges; export report; disconnect.
```

MarketData & SpreadEngine (Performance Design)

MarketData (ingestion) - Screen sharding: ~200 symbols across 3–4 screens (≤100 per screen). - Keep tiny in-place snapshots; mark DIRTY only on best-price change. - Avoid on-demand TRs during trading; rely on real-time L1.

SpreadEngine (decision) - Micro-batch cadence: ~10 ms (tunable). Single pass over DIRTY then clear. - Two candidate edges per symbol; tick- & fee-aware; require visible size ≥ slice. - Threshold: ≥ fees(buy+sell) + buffer (default +1 tick net after fees). - Cooldown: 100–200 ms after a just-miss/reject to prevent thrash.

Execution State Machine (V1)

- **States (per tiny slice):** `IDLE → CANDIDATE → ENTRY_TAKE_SENT → HEDGE_POST_SENT → PAIRED_DONE → IDLE`
  Branches: `ENTRY_REJECTED/COOLDOWN` and `CANCEL_POST_SENT → HEDGE_IOC_SENT`.
- **Rules:**
  - Take on the rich side (IOC/Market) first to secure hedge quickly.
  - Post on the cheap side (Limit or NXT Mid; mid uses hoga=29; price=0).
  - If hedge not filled by **t_hedge=1000 ms**: cancel rest → send IOC/Market to flatten.
  - Changing order type requires **cancel-then-new** (정정 can't flip type).
  - Treat empty TR 주문번호 as reject; lifecycle/fills from Chejan are authoritative.

Global Throttling Budget (V1)

- **Global buckets:**
  - **Orders:** 5/sec (hard cap). Cancels count.
  - **Data requests:** 5/sec (separate; avoid during trading).
  - **Real-time:** ≤ 100 symbols per screen number.
- **Reservations & admission:**
  - Always keep **2 order tokens reserved** for cancel/hedge.
  - Admit new entries only if **≥ 4 tokens free** (2 legs + 2 reserve).
  - **Auto-pause** new entries if orders/sec **≥ 80%** for **≥ 5 s** (advisory; hedges still allowed).
  - Concurrency (V1): **≤ 1–2 symbols** active; **1 outstanding slice per symbol**.

Telemetry, GUI & Alerts (V1)

- **SLO targets (p95):** Tick→Signal **25 ms**; Signal→Send **15 ms**; Send→Ack **150 ms**.
- **Reliability KPIs:** reject rate < **0.5%** (5m), timeout rate < **0.2%** (5m), unhedged time p95 ≤ **1000 ms**.
- **GUI:** Top status (session, orders/sec, tokens free, SLO tiles), Active Symbols table, Pair Monitor, Event Feed.
- **Slack scope (V1):** BUY_FILL, SELL_FILL, PAIR_DONE, AUTO-PAUSE ON/OFF, HEDGE TIMEOUT, REJECT SPIKE.

Configuration Schema (V1)

`config.yaml` **(high-level keys)** - **app:** mode, timezone, logging
- **kiwoom:** server, account, screen_numbers {marketdata[], orders}, rate_limits (orders_per_sec=5, queries_per_sec=5, reserve_order_tokens=2), features {use_sor=false, use_al_feed=false}
- **sessions:** arm_only_in_overlap=true; overlap_window {start 09:00:32, end 15:19:50}; nxt_main {09:00:30–15:20}; use_fid_215_signals=true
- **symbols:** universe_file; per_symbol_overrides {edge_buffer_ticks, max_outstanding_pairs, t_hedge_ms}
- **market_data:** subscribe_top_of_book_only=true; shards=3; heartbeat_symbols=[...]
- **spread_engine:** batch_interval_ms=10; edge_rule {min_net_ticks_after_fees=1; also_require_min_visible_qty=1}; cooldown_ms=100
- **router:** entry_leg prefer ioc/market; hedge_leg prefer limit/mid (allow_nxt_mid_price=true; fallback_after_ms=t_hedge_ms)
- **execution:** t_hedge_ms=1000; cancel_then_new_on_type_change=true; max_concurrent_symbols=1–2; max_outstanding_pairs_per_symbol=1
- **throttling:** orders_bucket_per_sec=5; queries_bucket_per_sec=5; min_tokens_free_to_start_new_pair=4
- **fees:** krx.broker_bps=1.5; nxt.broker_bps=1.45; nxt.regulatory_bps=0.31833
- **telemetry:** slo_targets_ms {25,15,150}; orders_utilization_autopause {threshold=0.80, sustain_seconds=5, enabled=true}
- **alerts.slack:** send_on {buy_fill, sell_fill, pair_done, auto_pause_on, hedge_timeout, reject_spike}
- **persistence:** logs_dir, reports_dir, exec_log_format, retention_days

---

Cooldown — Design Note

- **Purpose:** Prevent flip-flop on borderline spreads that oscillate at the tick boundary.
- **Mechanism:** After a symbol drops below threshold or we back off, set `next_eligible_at = now + cooldown_ms`; skip evaluation until then. Default **100–200 ms**.
- **Impact:** Reduces cancel/reorder churn, protects 5/sec budget, improves hedge timeliness.
- **Future options:** Adaptive cooldown or hysteresis (enter/exit thresholds).

Operator Runbook (V1)

Before session 1) Edit `config.yaml` (fees, overlap window, shards, SLOs); set Slack webhook.
2) Prepare symbols file (KRX codes).
3) Launch app → Connect Kiwoom → complete login; save account PW (first run).
4) Confirm status bar: Connected ✓, Account ✓, Server (실/모의).

Arm for trading 5) Wait for NXT Main + guard start **09:00:32**; register feeds; Arm Trading.
6) Orders bucket shows **2 tokens reserved** (for cancel/hedge).

During session 7) Watch Active Symbols & Pair Monitor (V1: 1–2 symbols active).
8) If orders/sec ≥ **80%** for **5s**, auto-pause new entries; hedges/cancels still proceed.
9) If passive hedge not filled by **t_hedge=1000 ms** → cancel → IOC/Market to flatten.

Disarm & shutdown 10) At **15:19:50** stop creating new entries; finish hedges.
11) Export report; disconnect Kiwoom; exit.

Glossary

- **Take/Post:** Take = hit/lift immediately (IOC/Market). Post = place passive order (Limit or Mid).
- **Mid on NXT:** hoga=29 with price=0; passive midpoint order on NXT.
- **DIRTY set:** Symbols whose top-of-book changed since last micro-batch.
- **Micro-batch:** Fixed-cadence (~10 ms) evaluation pass; coalesces bursts.
- **t_hedge:** Max wait for passive hedge to fill before escalating to IOC/Market.

Open Items / Future Iterations

- Per-symbol buffers (½ vs 1 tick) based on volatility/liquidity.
- Adaptive cooldown/hysteresis toggles and strong-edge overrides.
- Kill-switch policies (auto-freeze on repeated rejects/timeouts).
- Extending to pre/after sessions; revisiting SOR in sandbox only.

End of Blueprint (V1)