Dealer Specification for Bilancio

November 22, 2025

1 Readiness of Dealer Specification for Bilancio Implementation

1.1 Overview

The current input—the dealer specification for the Kalecki ring, the worked Examples, and the v1.0 "Money modeling software" document—is sufficient to implement a one–security, three–bucket dealer module inside Bilancio.

The documents jointly provide:

- A complete per–bucket dealer kernel ("L1") with explicit formulas for capacity, ladder, layoff probability, inside width, inventory–sensitive midline, interior quotes, clipping and pass–through pins.
- Double–entry event tables for all balance–sheet mechanics: trader–dealer trades, pass–through to VBTs, settlement with proportional recovery, and internal dealer/VBT rebucketing across buckets.
- A time–stepped event loop for the Kalecki ring with tickets, including maturity/bucket updates, dealer pre–computation, randomized one–ticket order flow, settlement, and (optionally) VBT anchor updates.
- A battery of programmatic assertions (C1–C6) and worked Examples that can be turned directly into unit and integration tests.

The remaining choices for v1.0 are parameter selections and one policy toggle, not missing structure. Once these are fixed (or sensible defaults adopted), the coding work can proceed.

1.2 Elements that are already specified

1.2.1 Dealer kernel (per bucket)

For each bucket b (Short, Mid, Long), with ticket size $S > 0$, outside mid M and spread O, outside quotes and dealer kernel are given by:

$$\begin{aligned}
A &= M + \frac{O}{2}, \qquad B = M - \frac{O}{2}, \\
V &= Ma + C, \\
K^* &= \left\lfloor \frac{V}{M} \right\rfloor, \\
X^* &= SK^*, \\
\lambda &= \frac{S}{X^* + S}, \\
I &= \lambda O, \\
p(x) &= M - \frac{O}{X^* + 2S}\left(x - \frac{X^*}{2}\right), \\
a(x) &= p(x) + \frac{I}{2}, \qquad b(x) = p(x) - \frac{I}{2}, \\
a_c(x) &= \min\{A, a(x)\}, \qquad b_c(x) = \max\{B, b(x)\}.
\end{aligned}$$

Feasibility and pins:

- Customer BUY (dealer sells): an interior sale is feasible iff $x \geq S$. Otherwise, the operational ask is pinned to A and the trade is routed as pass–through to the VBT; the dealer's state (x, C) is unchanged.
- Customer SELL (dealer buys): an interior buy is feasible iff $x + S \leq X^*$ and $C \geq b_c(x)$. If not, the bid is pinned to B and the trade is passed through to the VBT, leaving (x, C) unchanged.

Guard: if $M \leq M_{\min}$, the Guard sets $X^* := 0$, collapses the ladder to $\{0\}$, and pins quotes to (A, B); only pass–through executions are possible.

1.2.2 Environment and actors

- Kalecki ring with N agents, ticket size S, and three maturity buckets: *Short* $\tau \in \{1, 2, 3\}$, *Mid* $\tau \in \{4, \ldots, 8\}$, *Long* $\tau \geq 9$.
- Exactly three dealers and three VBTs: one pair per bucket.
- Traders:
  - One debt liability per trader, owed to a single counterparty with a single maturity.
  - On the asset side, tickets on at most one issuer.
  - Equity measured at face (assets minus liabilities).
- Dealers/VBTs:
  - Assets: cash and tickets (bucket–eligible).
  - No explicit debt liabilities; equity is residual $E = C + Mx$ (marked to VBT mid).

1.2.3 Event mechanics

Settlement uses proportional recovery: for issuer i at time t,

$$R_i(t) = \min\left\{1, \frac{C_i(t)}{D_i(t)}\right\}, \qquad D_i(t) = SN_i(t).$$

Each holder of q_h maturing tickets receives $q_h S R_i(t)$; all such tickets are deleted and issuer cash becomes $C_i(t^+) = C_i(t) - R_i(t)D_i(t)$.

Event tables cover:

- Trader–dealer trades (Events 1–2).
- Settlement to dealer, VBT, trader with full repayment (Events 3–5).
- Settlement with default / partial recovery to each holder type (Events 6–8).
- Dealer pass–through to VBT at B/A (Events 9–10), leaving (x, C) unchanged.
- Internal dealer/VBT rebucketing at bucket boundaries (Events 11–12).

1.2.4 Initialization

Per bucket:

1. Choose a target capacity X^*_{target} and outside parameters M, O.
2. Set
$$\lambda = \frac{S}{X^*_{\text{target}} + S}, \qquad I = \lambda O.$$
3. Initialize inventory at the midpoint of the shelf:
$$x_0 = \frac{X^*_{\text{target}}}{2}, \qquad a_0 = \frac{x_0}{S}.$$
4. Set initial cash $C_0 = Ma_0$, so that $Ma_0 = C_0$ and $V_0 = Ma_0 + C_0 = 2Ma_0$.
5. Recompute capacity; this yields $X^* = X^*_{\text{target}}$ by construction.
6. Reassign ownership in the bucket: dealer $\approx 25\%$, VBT $\approx 50\%$, remainder with traders, all booked at M_0.

1.2.5 VBT anchors

Per bucket b and period t:

$$\begin{aligned}
\ell^b_t &\in [0,1] \quad \text{(bucket loss rate on tickets in } b \text{ that mature at } t\text{)},\\
M^b_{t+1} &= M^b_t - \phi^b_M\, \ell^b_t,\\
O^b_{t+1} &= O^b_t + \phi^b_O\, \ell^b_t,
\end{aligned}$$

with optional clipping $O^b_{t+1} \leftarrow \max\{O^b_{\min}, O^b_{t+1}\}$ and $B^b_{t+1} \leftarrow \max\{0, B^b_{t+1}\}$.

1.3 Remaining choices for v1.0 (with suggested defaults)

For concreteness, the following defaults can be adopted:

1. **Default handling:**
   - Use proportional recovery and continue the simulation rather than stopping on first default. Introduce a switch allowing "stop on default" for special runs.

2. **Outside anchors and spreads:**

$$M_0^S = M_0^M = M_0^L = 1.0,$$

   with spreads, for example,

$$O_0^S = 0.20, \quad O_0^M = 0.30, \quad O_0^L = 0.40.$$

3. **Loss sensitivities:**
$$\phi_M^b = 1.0, \qquad \phi_O^b = 0.6 \quad \text{for all buckets } b.$$

4. **Guard threshold:** $M_{\min} = 0.02$, with outside–bid non–negativity clip $B \leftarrow \max\{0, B\}$ enabled by default.

5. **Ticket size:** $S = 1$.

6. **Order–flow controls:** $\pi_{\text{sell}} = 0.5$, $N_{\max} = 3$, with a fixed RNG seed.

7. **Trader policies:**
   - Investment horizon $H = 3$ (minimum distance to earliest liability).
   - Cash buffer $B_{\text{buffer}} = 1.0$.

8. **Initial dealer capacity:** $X^*_{\text{target}} = 4$ in each bucket; with $M = 1, S = 1$ this gives $x_0 = 2$, $C_0 = 2$.

9. **Ticket–ID tie–breaking:** deterministic, e.g. lowest maturity τ then smallest serial number.

1.4 Implementation plan for the coding agent

1.4.1 Data model

Tickets.

- Fields: issuer, owner, face = S, maturity_time, remaining_tau, bucket_id, serial_id.

Agents (traders).

- State: cash, list of tickets_owned, list of obligations_due, single asset–side issuer $k(i)$.

- Methods: `compute_shortfall`, `compute_earliest_liability`, `eligible_to_sell`, `eligible_to_buy`, `choose_ticket_to_sell`, `choose_bucket_to_buy`.

DealerBucket (per bucket).

- State: $\{S, M, O, A, B, a, C, x, X^*, \lambda, I\}$.
- Methods:
  - `recompute_capacity_and_quotes()`.
  - `quote_bid()`, `quote_ask()` returning $b_c(x)$, $a_c(x)$.
  - `can_buy_one()`, `can_sell_one()` implementing feasibility rules.
  - `execute_customer_sell()` (dealer buys one ticket).
  - `execute_customer_buy()` (dealer sells one ticket).

VBTBucket (per bucket).

- State: (M, O, A, B), deep inventory, cash, anchor–update parameters (ϕ_M, ϕ_O).
- Methods: anchor update from loss rate, pass–through execution at A/B, ticket transfer at bucket boundaries.

1.4.2 Event loop (per period)

For each time step t:

1. **Update maturities and buckets.**
   For each ticket:
   $$\text{remaining\_tau} \leftarrow \text{remaining\_tau} - 1,$$
   recompute bucket_id; migrate tickets across buckets. If a dealer/VBT holds a ticket that crosses a boundary, perform an internal sale at the new bucket mid (Events 11–12); if a trader holds it, only the bucket label changes.

2. **Dealer pre–computation (per bucket).**
   Given $(M_t^b, O_t^b, S, a_t^b, C_t^b)$, compute $X_t^{*,b}$, λ_t^b, I_t^b, $p^b(x_t^b)$, $a_c^b(x_t^b)$, $b_c^b(x_t^b)$, applying Guard if $M_t^b \le M_{\min}$.

3. **Eligibility sets.**
   For each trader i, compute:
   $$\text{shortfall}_i(t) = \max\{0, \text{payments\_due}_i(t) - \text{cash}_i(t)\},$$
   and earliest future liability date $T_i^{\min}(t)$ (or $+\infty$). Define:
   $$S_t = \{i : \text{shortfall}_i(t) > 0,\ i \text{ owns at least one ticket}\},$$
   $$B_t = \{i : T_i^{\min}(t) - t \ge H,\ \text{cash}_i(t) > B_{\text{buffer}}\}.$$
   Each agent may trade at most once per side per period; after a successful trade, remove i from the corresponding set.

4. **Randomized one–ticket order flow.**
   With parameters $\pi_{\text{sell}} \in (0, 1)$ and $N_{\max} \in \mathbb{N}$:

```
n := 1
while n <= Nmax and (S_t  B_t  ):
    draw Z_n ~ Bernoulli(_sell)
    if Z_n = 1:   # SELL-preferred
        if S_t  :
            pick i  S_t
            let i choose ticket, bucket
            query dealer bid b_c(x)
            if feasible (interior or pass-through):
                execute SELL; update states
                remove i from S_t
        else if B_t  :
            treat as BUY (fallback)
    else:         # BUY-preferred
        if B_t  :
            pick i  B_t
            let i choose bucket
            query dealer ask a_c(x)
            if feasible (interior or pass-through and cash_i  p):
                execute BUY; update states
                remove i from B_t
        else if S_t  :
            treat as SELL (fallback)
    if trade executed:
        recompute dealer state/quotes in affected bucket
    n := n + 1
```

5. **Settlement with proportional recovery.**
   For each issuer i:
   - Collect tickets with maturity_time $= t$; compute $N_i(t)$ and $D_i(t)$.
   - Compute $R_i(t)$ as above; pay $q_h S R_i(t)$ to each holder h.
   - Delete all maturing tickets and update issuer cash to $C_i(t^+)$.
6. **VBT anchor update (optional).**
   Compute bucket loss rates ℓ_t^b from defaults in t and update (M_{t+1}^b, O_{t+1}^b) (with clipping, if enabled). Use these in dealer pre–computation at $t+1$.

1.5 Configuration stub

A convenient configuration stub (e.g. YAML) for v1.0 can be:

```
seed: 42
ticket_size_S: 1

buckets:
  short:  {tau: [1,3],  M0: 1.0, O0: 0.20, phiM: 1.0, phiO: 0.6}
  mid:    {tau: [4,8],  M0: 1.0, O0: 0.30, phiM: 1.0, phiO: 0.6}
```

```
  long:   {tau: [9,inf],M0: 1.0, O0: 0.40, phiM: 1.0, phiO: 0.6}

guard:
  Mmin: 0.02
  clip_nonneg_B: true

dealers:
  init_target_capacity_Xstar: 4

order_flow:
  pi_sell: 0.5
  Nmax: 3

trader_policy:
  H_min_invest: 3
  buffer_cash_B: 1.0

initial_allocation:
  dealer_share: 0.25
  vbt_share: 0.50
  traders_share: 0.25

ring:
  n_traders_min: 100
```

1.6 Test plan aligned with Examples and assertions

Tests should systematically enforce the following assertions after each scripted event:

- **C1 (Double–entry and conservation):** sum of cash changes and ticket count changes across parties is zero; registry size changes only at settlement.
- **C2 (Quote bounds and pins):** $b_c(x) \geq B$, $a_c(x) \leq A$, with explicit pin flags when equalities hold.
- **C3 (Feasibility checks):** interior execution only if feasibility conditions hold (no shorting, no borrowing).
- **C4 (Pass–through invariants):** at pins, the dealer's $(x', C') = (x, C)$.
- **C5 (Equity basis by role):** dealers/VBTs marked to M; traders at face.
- **C6 (Anchor timing):** anchors (M_{t+1}, O_{t+1}) depend only on losses in t.

Each worked Example can be implemented as a dedicated scenario test:

- Capacity jumps across an integer K^* and associated discrete jumps in λ and I.
- Boundary pass–through on bid and ask with dealer invariants.
- Guard regime with $M \leq M_{\min}$ and pass–through on both sides.

- Multi–claimant defaults with proportional recovery and bucket loss rate computation.
- Ticket–ID transfer on BUYs (interior and pinned), with no generic ticket creation.
- Minimal event–loop harness for arrivals, verifying removal from eligibility sets, fallback, and convergence of executed SELL share towards π_{sell} (subject to feasibility).

1.7 Integration with v1.0 money–modeling engine

The dealer module is consistent with the v1.0 engine in the following sense:

- All interactions are represented as balance–sheet events with double–entry updates.
- Time is discrete; events occur at timepoints when entries activate, in line with the underlying debt ring.
- The dealer/VBT subsystem adds a secondary market and price formation layer, while preserving the base architecture of agents, tickets, and settlement.

The only nontrivial choice relative to the v1.0 text is whether to stop the simulation on first default or to continue using proportional recovery; for the Dealer module as currently specified, continuation with proportional recovery is the natural baseline.