RECON-AI

Process Flow Specification

Event → NAV Dashboard → Trial Balance (Ledger Drill-Down) → Position Drill-Down

Version:	1.0
Date:	February 2026
Domain:	Fund Administration
Context:	Production Parallel Phase
Classification:	Internal – Confidential

1. Overview

1.1 Purpose

This specification defines the complete user interaction flow for the RECON-AI Control Center drill-down experience. It covers the four-screen progressive disclosure path that takes a user from the Event-level portfolio view down through NAV comparison, trial balance ledger decomposition, and finally into position-level detail. Each screen is designed to answer a specific question in the analyst's investigation workflow.

1.2 Navigation Model

The drill-down follows a strict hierarchical path with breadcrumb navigation. Each level narrows the analytical scope and increases data granularity:

Level	Screen	Primary Question	Data Grain	Key Entity
L0	Event Dashboard	Which events have breaks?	Event / Fund	Event, Fund
L0–L1	NAV Dashboard	Which funds break and where?	Fund / Share Class	NAV, GL Bucket
L1–L2	Trial Balance	Which GL accounts drive the variance?	GL Account	Ledger, SubLedger
L2–L3	Position Drill-Down	Which positions and lots cause the break?	Position / Lot	Position, TaxLot

1.3 Design Principles

- Progressive Disclosure: Each screen loads only the data relevant to the user's current investigative scope. Deeper detail is fetched on-demand via click-through.
- Contextual Propagation: The selected Event, Fund, Valuation Date, and Share Class propagate as filter context through every downstream screen. Breadcrumbs encode this context.
- Validation-Aware Rendering: Every numeric comparison carries an inline validation status (pass/break) derived from the data_validation rules. Breaks are color-coded and sortable.
- AI Commentary Persistence: AI agent analysis surfaces at every level. Commentary propagates downward (e.g., an L0 NAV comment about unrealized variance is visible when drilling into the relevant ledger account).
- Bi-Directional Cross-Checks: Where a drill-down row has a secondary validation (e.g., Ledger BS Compare Check alongside NAV Compare), the cross-check result is displayed inline as expandable detail.

2. Screen 1: Event Dashboard

2.1 Purpose

The Event Dashboard is the landing page of the RECON-AI Control Center. It provides a portfolio-level view of all active conversion events, their fund counts, validation health, and recent activity. The user selects an event to begin investigation.

2.2 URL & Routing

```
/events
```

2.3 Layout Structure

2.3.1 Event Cards Grid

Each conversion event is represented as a card in a responsive grid. Cards provide at-a-glance status and direct actions.

Component	Type	Description
Event Name + ID	Text	Descriptive name and system ID (e.g., EVT-2026-001: Vanguard Fixed Income Migration)
Status Badge	Enum Badge	DRAFT (gray), ACTIVE (blue), PARALLEL (amber), SIGNED_OFF (green), COMPLETE (teal)
Fund Progress Bar	Progress	Horizontal bar: funds passed (green) / attention (amber) / failed (red) / total. Label: "12 of 15 funds passed"
Incumbent Provider	Text	Source system name (e.g., State Street, Northern Trust)
Target Go-Live	Date	Planned production cutover date with countdown (e.g., "Mar 15, 2026 – 31 days")
7-Day Break Trend	Sparkline	Mini line chart showing total break count across all funds over trailing 7 valuation dates
Last Validation	Timestamp	Most recent validation run completion time with relative label (e.g., "Today 08:35 AM")
Quick Actions	Buttons	Primary: [Run Validation] → triggers validation modal. Secondary: [View Details] → navigates to NAV Dashboard

2.3.2 Quick Filters Bar

- Status: Multi-select chips for DRAFT, ACTIVE, PARALLEL, SIGNED_OFF, COMPLETE
- Date Range: Date picker filtering events by target go-live window
- Assigned To Me: Toggle filter showing only events where current user is on assignedTeam
- Search: Full-text search across event name, fund names, and incumbent provider

2.3.3 Activity Feed (Right Panel)

A chronological feed of system events scoped to the user's accessible events:

- Validation run completions with pass/fail summary
- AI analysis completions with confidence score

- Human review actions (approved, modified, escalated)
- Status transitions (event or fund status changes)

Each feed item is clickable and navigates to the relevant screen context.

2.4 User Actions

Action	Behavior	Navigation
Click Event Card	Selects the event and loads NAV Dashboard context	→ /events/{eventId}/nav-dashboard
Click [Run Validation]	Opens validation configuration modal (date picker, check suite selector, fund filter)	Modal overlay, stays on page
Click [View Details]	Direct navigation to event's NAV Dashboard	→ /events/{eventId}/nav-dashboard
Click Activity Feed Item	Navigates to the specific context (validation run, break, etc.)	→ context-dependent route

2.5 Data Requirements

Data Element	Source	Refresh	Notes
Event metadata	Event entity store	On page load	eventId, name, status, dates, team
Fund roll-up counts	ValidationRun aggregation	Real-time (WebSocket)	Pass/fail/attention counts per event
Break trend data	BreakRecord time series	On page load + daily	7-day rolling aggregation
Activity feed	Audit trail + AI analysis	Real-time (SSE)	Scoped to user's assigned events

3. Screen 2: NAV Dashboard

3.1 Purpose

The NAV Dashboard is the primary operational workspace for a single conversion event. It displays the NAV Compare (TF) validation, which is the aggregate-of-all-classes comparison between Incumbent TNA and BNY TNA for every fund in the event on a selected valuation date. This is the L0 reconciliation entry point. Each row is expandable to reveal inline cross-check validations (Ledger BS Compare Check, Ledger INCST Compare Check) before drilling deeper.

3.2 URL & Routing

```
/events/{eventId}/nav-dashboard?valuationDt={date}
```

3.3 Breadcrumb

Events > {Event Name}

3.4 Layout Structure

3.4.1 Validation Control Panel

A sticky panel at the top of the screen providing validation execution controls:

Control	Type	Behavior
Valuation Date	Date Picker	Selects the NAV date for comparison. Defaults to most recent available date. Calendar highlights dates with existing validation runs (dot indicator). Changing date triggers data refresh.
Check Suite	Multi-select Checkboxes	NAV to Ledger (L0), Ledger BS to INCST (L1), Ledger TF to Class (L1), Position to Lot (L2), Ledger to Subledger (L2), Basis Lot Check (L2). All checked by default.
Fund Filter	Radio + Multi-select	All Funds (default) or Selected Only with searchable fund picker. Persists selection across date changes.
Run Validation	Primary Button	Triggers validation execution. Shows loading state with progress bar ("Processing fund 3 of 15..."). Disabled during active run.
Schedule	Secondary Button	Opens scheduling modal for recurring daily validation at configured time.

3.4.2 NAV Compare Grid (Primary Table)

This is the core data grid implementing the NAV to Ledger validation (Section 1.1 of Data Validation Rules). It shows aggregate-of-all-classes NAV comparison for each fund.

Grid Configuration:

- Component: AG-Grid Enterprise with server-side row model
- Default Sort: Validation status (breaks first), then by absolute TNA Difference descending
- Row Selection: Single-click selects row, double-click opens Trial Balance drill-down

- Expandable Rows: Chevron expands to show inline cross-check validations

Column Definitions:

Column	Field Source	Data Type	Format	Notes
Valuation Dt	dataNav.valuationDt	Date	YYYY-MM-DD	Key field from canonical model
Account	dataNav.account	String	Left-aligned	Portfolio account ID, links to refFund
Account Name	refFund.accountName	String	Left-aligned	Display field joined from refFund
Incumbent TNA	dataNav.netAssets (Incumbent)	Decimal(18, 2)	#,##0.00	LHS: Net assets from incumbent source (userBank = incumbent)
BNY TNA	dataNav.netAssets (BNY)	Decimal(18, 2)	#,##0.00	RHS: Net assets from BNY source (userBank = BNY)
TNA Difference	Calculated	Decimal(18, 2)	(#,##0.00) red for negative	Incumbent TNA - BNY TNA
TNA Difference BP	Calculated	Decimal(10, 4)	#,##0.00 bp	Difference as basis points of Incumbent TNA
Validation	Computed	Enum	Traffic light icon	Green: |diff| < threshold. Amber: marginal. Red: material break. Threshold configurable per event.

3.4.3 Expandable Row: Inline Cross-Check Validations

When a user expands a NAV Compare row, two inline cross-check tables appear beneath it. These correspond to the secondary validations that verify internal consistency of the BNY data before comparing to the incumbent.

Cross-Check A: Ledger BS Compare Check

Validates that the ledger balance sheet (GL accounts starting with 1 or 2, class = TF) ties to the reported NAV netAssets. This implements validation rule 1.1 (NAV to Ledger) at the individual fund level.

Column	LHS Source	RHS Source	Format	Validation
Incumbent TNA	dataNav.netAssets (Incumbent)	—	#,##0.00	—
BNY TNA	—	dataNav.netAssets (BNY)	#,##0.00	—
TNA Difference	Calculated	Calculated	(#,##0.00)	Traffic light

Cross-Check B: Ledger INCST Compare Check

Validates that the income statement ledger (GL accounts NOT starting with 1 or 2) ties to the balance sheet remainder. This implements validation rule 1.2 (Ledger BS to INCST).

Column	LHS Source	RHS Source	Format	Validation
Incumbent TNA	dataLedger (INCST filter)	—	#,##0.00	—
BNY TNA	—	dataLedger (BS filter)	#,##0.00	—

TNA Difference	Calculated	Calculated	(#,##0.00)	Traffic light

3.4.4 AI Commentary Panel

A collapsible right-hand panel (350px default width, resizable) that displays AI agent commentary for the selected fund row:

- Trend Summary: AI-generated narrative summarizing break trends across recent valuation dates (e.g., "Price breaks on bond securities has caused unrealized variance on the level 3 position recon, leading to ledger differences in unrealized on the level 2 ledger recon, leading to the net asset variance on the level 1 NAV recon.")
- Pattern Recognition: Links to similar breaks in other funds within the same event
- Confidence Score: Visual gauge with percentage
- Recommended Next Step: Suggested drill-down path (e.g., "Investigate Accrued Income in Trial Balance")

3.5 User Actions

Action	Behavior	Navigation
Click fund row	Selects row, updates AI panel context	Stays on NAV Dashboard
Expand row (chevron)	Shows inline Ledger BS and INCST cross-check tables	Stays on NAV Dashboard
Double-click fund row	Opens Trial Balance for selected fund	→ /events/{eventId}/funds/{account}/trial-balance?valuationDt={date}
Click break status icon	Opens AI Analysis slide-out panel with full root cause detail	Slide-out panel overlay
Right-click → Export	Exports NAV Compare grid to Excel with all visible columns	Download initiates
Change Valuation Date	Refreshes all grid data for new date. Preserves fund selection if available.	Stays on page, data refreshes

3.6 Data Requirements

The NAV Dashboard executes the following data validation queries derived from the validation rules specification:

NAV Compare (TF) – Primary Grid:

- LHS: SELECT valuationDt, account, class, netAssets FROM dataNav WHERE isPrimaryBasis = 'Y' AND userBank = {incumbent}
- RHS: SELECT valuationDt, account, class, netAssets FROM dataNav WHERE isPrimaryBasis = 'Y' AND userBank = 'BNY'
- Join: ON valuationDt, account, class
- Aggregate: SUM by account (aggregate of all classes) for grid display

Ledger BS Compare Check (Expandable):

- LHS: SELECT SUM(endingBalance) FROM dataLedger WHERE isPrimaryBasis = 'Y' AND eagleClass = 'TF' AND (LEFT(eagleLedgerAcct,1) = '1' OR LEFT(eagleLedgerAcct,1) = '2')

- RHS: dataNav.netAssets for same account/date

Ledger INCST Compare Check (Expandable):

- LHS: SELECT SUM(endingBalance) FROM dataLedger WHERE isPrimaryBasis = 'Y' AND eagleClass = 'TF' AND LEFT(eagleLedgerAcct,1) NOT IN ('1','2')
- RHS: Derived from BS total minus NAV, per rule 1.2

4. Screen 3: Trial Balance (Ledger Drill-Down)

4.1 Purpose

The Trial Balance screen decomposes a fund's NAV variance into its constituent ledger account categories. When a user drills into a fund showing a $17,837 NAV break, this screen shows the breakout of ledger categories that make up that amount, including any validation breaks at the ledger level. This is the L1 reconciliation layer that answers: "Which accounting categories drive this fund's NAV variance?"

4.2 URL & Routing

```
/events/{eventId}/funds/{account}/trial-balance?valuationDt={date}
```

4.3 Breadcrumb

Events > {Event Name} > {Fund Name} > Trial Balance

4.4 Context Header

A fixed context bar showing the inherited filter state from the parent NAV Dashboard:

- Fund: {accountName} ({account})
- Valuation Date: {valuationDt}
- NAV Variance: ${TNA Difference} ({TNA Difference BP} bp) with traffic light indicator
- AI Status: Analyzing / Complete ({confidence}%) / Needs Review

4.5 Layout Structure

4.5.1 Ledger BS Compare Grid (Primary Table)

This table implements the Ledger-to-Subledger validation (Section 1.5 of Data Validation Rules) displayed as a categorized trial balance. Each row represents a GL category with its incumbent vs. BNY balances, and the variance that contributes to the overall NAV break.

Column Definitions:

Column	Field Source	Data Type	Format	Notes
Valuation Dt	dataLedger.valuationDt	Date	YYYY-MM-DD	Inherited from NAV Dashboard context
Account	dataLedger.account	String	Left-aligned	Fund portfolio account ID
Category	Derived from eagleLedgerAcct	String	Left-aligned	GL category label: Cash, Investment Cost, Investment Urgl, Futures Margin, Dividend RecPay, Interest RecPay, Expense RecPay, etc.
Incumbent Balance	dataLedger.endingBalance (Incumbent)	Decimal(18, 2)	#,##0.00	Sum of GL accounts in category from incumbent source

BNY Balance	dataLedger.endingBalance (BNY)	Decimal(18, 2)	#,##0.00	Sum of GL accounts in category from BNY source
Balance Diff	Calculated	Decimal(18, 2)	(#,##0.00)	Incumbent Balance - BNY Balance
Balance Diff BP	Calculated	Decimal(10, 4)	#,##0.00 bp	Difference as basis points of fund TNA
Validation	Computed	Enum	Traffic light icon	Green if |diff| < threshold, else Red

Expected Category Rows (derived from Section 2 of Data Validation Rules):

Category	GL Account Pattern	Derived SubLedger Source
Cash	Cash GL accounts	Direct ledger load (ledgerLoad = 1)
Investment Cost	eagleCostLedgerAcct	posBookValueBase from dataSubLedgerPosition (Security Cost rule)
Investment Urgl	1011000101 (BS)	posMarketValueBase - posBookValueBase (Security URGL BS rule)
Futures Margin	Futures GL accounts	ltdVariationMarginBase (Future URGL INCST rule)
Dividend RecPay	Dividend receivable GL	Unsettled transaction amounts
Interest RecPay	eagleIntLedgerAcct	posIncomeBase from dataSubLedgerPosition (Security Interest rule)
Expense RecPay	Expense payable GL	Unsettled expense transaction amounts
Capital Subs	3002000110	subscriptionBalance * -1 (Capital Subs rule)
Capital Reds Pay	3002000210 / 2005003500	redemptionBalance / redemptionPayBase (Capital Reds rules)
Distribution	3004000100/110/120	incomeDistribution / stcgDistribution / ltcgDistribution
Distribution Payable	2006000700	distributionPayable * -1
Forward Cost	1007001100 / 2005002900	ABS(fwdBookValue) receivable/payable (Forward rules)
Forward URGL	1011000201 / 4004000401	fwdUnrealized BS/INCST (Forward URGL rules)
Income Unrealized	1011000300 / 3003000800	posIncomeMarket - posIncomeBase (Security Int URGL rules)

4.5.2 Expandable Row: Subledger Compare Check

When a user expands a category row in the Trial Balance grid, an inline table appears showing the Subledger Compare Check. This validates that the GL ending balance matches the derived subledger rollup value for that category. This implements validation rule 1.5 (Ledger to Subledger).

Column	LHS (Ledger)	RHS (Subledger)	Difference	Validation
Description	dataLedger.endingBalance	derivedSubLedgerRollup.subLedgerValue	endingBalance - subLedgerValue	Traffic light
Filter (LHS)	eagleClass = 'TF' AND isPrimaryBasis = 'Y'	—	—	—
Filter (RHS)	—	isPrimaryBasis = 'Y'	—	—

4.5.3 Reconciliation Roll-Up Summary

Below the grid, a summary footer row shows:

- Total Incumbent Balance: Sum of all category incumbent balances
- Total BNY Balance: Sum of all category BNY balances
- Total Variance: Should tie to the NAV TNA Difference from the parent screen
- Tie-Out Validation: A visual check confirming the sum of category variances equals the NAV-level variance. If it does not tie, a warning is displayed.

4.5.4 NAV Waterfall Chart

A D3.js waterfall chart positioned above or alongside the grid, visually decomposing the NAV variance:

- Starting bar: Incumbent NAV (full height, neutral color)
- Component bars: Each GL category's variance contribution as positive (green, upward) or negative (red, downward) bars
- Ending bar: BNY NAV (full height, neutral color)
- Interactive: Clicking a bar in the waterfall highlights/filters the corresponding row in the grid
- Tooltip: On hover, shows category name, amount, and percentage of total variance

4.6 User Actions

Action	Behavior	Navigation
Click category row	Selects row, shows subledger detail in expandable area	Stays on Trial Balance
Expand row (chevron)	Shows inline Subledger Compare Check for that GL category	Stays on Trial Balance
Double-click category row	Opens Position Drill-Down filtered to that GL category	→ /events/{eventId}/funds/{account}/positions?valuationDt={date}&category={cat}
Click waterfall bar	Highlights corresponding grid row, scrolls into view	Stays on Trial Balance
Click breadcrumb	Returns to parent screen with context preserved	→ NAV Dashboard or Event Dashboard
Export to Excel	Exports full trial balance with all categories and subledger detail	Download initiates

4.7 AI Commentary Integration

The AI commentary panel at this level shows category-level analysis:

- Identifies which categories are the primary variance drivers
- Correlates category breaks with position-level patterns (e.g., "Investment Urgl variance of ($22,972.58) is driven by price differences on 3 bond positions")
- Propagates upward commentary: summarizes how this category variance flows to the NAV-level break

- Suggests drill-down priority: recommends which categories to investigate first based on magnitude, confidence, and pattern matching

5. Screen 4: Position Drill-Down

5.1 Purpose

The Position Drill-Down is the deepest operational view, implementing L2 (Position to Lot) and L2 (Ledger to Subledger) validations. When a user drills into a GL category showing a variance (e.g., Investment Urgl with a ($22,972.58) difference), this screen shows every security position contributing to that category, with the ability to further expand into individual tax lots. This is where the analyst identifies the specific securities, accruals, or transactions causing the break.

5.2 URL & Routing

```
/events/{eventId}/funds/{account}/positions?
valuationDt={date}&category={category}
```

5.3 Breadcrumb

Events > {Event Name} > {Fund Name} > Trial Balance > {Category} Positions

5.4 Context Header

Inherited context from Trial Balance plus category filter:

- Fund: {accountName} ({account})
- Valuation Date: {valuationDt}
- Category: {GL Category Name} (e.g., Investment Cost, Investment Urgl, Interest RecPay)
- Category Variance: ${Balance Diff} ({Balance Diff BP} bp)
- NAV Variance: ${TNA Difference} (total fund-level for reference)

5.5 Layout Structure

5.5.1 Position Compare Grid (Primary Table)

This table implements the Position-to-Lot validation (Section 1.4 of Data Validation Rules) filtered to the selected GL category. It compares position-level aggregates between Incumbent and BNY data. The comparison fields vary by category context.

Column Definitions – Core Columns (all categories):

Column	Field Source	Data Type	Format	Notes
Asset ID	dataSubLedgerPosition.assetId	String	Left-aligned	Primary security identifier, links to refSecurity
Security Type	refSecurity.secType	String	Left-aligned	Joined from refSecType for display label
Issue Description	refSecurity.issueDescription	String	Left-aligned	Security name/description
CUSIP	refSecurity.cusip	String	Left-aligned	9-character CUSIP identifier
Long/Short	dataSubLedgerPosition.longShortInd	String	Center	L or S indicator

Share Class	dataSubLedgerPosition.shareClass	String	Left-aligned	Class identifier for multi-class funds

Comparison Columns by Category Context:

GL Category	Comparison Fields (Incumbent vs BNY + Variance)
Investment Cost	posBookValueLocal, posBookValueBase, posOrigCostLocal, posOrigCostBase, posShares, posOriginalFace
Investment Urgl	posUnrealizedLocal, posUnrealizedBase (derived: posMarketValueBase - posBookValueBase), posMarketPrice, posMarketValueBase
Interest RecPay	posIncomeLocal, posIncomeBase, posIncomeMarket, posPrevCouponDt, posNextCouponDt, dailyInterestLocal, dailyInterestBase
Dividend RecPay	Unsettled transaction fields: transAmountLocal, transAmountBase, transTradeDate, transSettleDate
Expense RecPay	Unsettled transaction fields: transAmountLocal, transAmountBase, transCode, transSettleDate
Forward Cost/Urgl	fwdLongAmount, fwdShortAmount, fwdBookValue, fwdUnrealized, fwdLongCurrency, fwdShortCurrency
Futures Margin	ltdVariationMarginLocal, ltdVariationMarginBase, dailyVariationMarginLocal, dailyVariationMarginBase

For each comparison field, the grid shows three sub-columns:

- Incumbent {field}: Value from incumbent data source (userBank = incumbent)
- BNY {field}: Value from BNY data source (userBank = BNY)
- Variance: Incumbent - BNY, with conditional formatting (red if |variance| > threshold)

5.5.2 Expandable Row: Tax Lot Detail

When a user expands a position row, the tax lot detail appears. This implements the Position-to-Lot validation (Section 1.4) at the individual lot level, using dataSubLedgerTrans as the lot-level source.

Tax Lot Columns:

Column	Field Source	Data Type	Format	Notes
Transaction ID	dataSubLedgerTrans.transactionId	String	Left-aligned	Unique lot identifier
Lot Trade Date	dataSubLedgerTrans.lotTradeDate	Date	YYYY-MM-DD	Original acquisition date
Lot Settle Date	dataSubLedgerTrans.lotSettleDate	Date	YYYY-MM-DD	Original settlement date
Shares	dataSubLedgerTrans.shares	Decimal(18,6)	#,##0.000000	Lot share quantity
Original Face	dataSubLedgerTrans.originalFace	Decimal(18,6)	#,##0.000000	For factor-based securities
Orig Cost Local	dataSubLedgerTrans.origCostLocal	Decimal(18,2)	#,##0.00	Original cost of lot (local)
Orig Cost Base	dataSubLedgerTrans.origCostBase	Decimal(18,2)	#,##0.00	Original cost of lot (base)
Book Value Local	dataSubLedgerTrans.bookValueLocal	Decimal(18,2)	#,##0.00	Amortized cost (local)

Book Value Base	dataSubLedgerTrans.bookValueBase	Decimal(18, 2)	#,##0.00	Amortized cost (base)
Market Value Local	dataSubLedgerTrans.marketValueLocal	Decimal(18, 2)	#,##0.00	Current market value (local)
Market Value Base	dataSubLedgerTrans.marketValueBase	Decimal(18, 2)	#,##0.00	Current market value (base)
Income Local	dataSubLedgerTrans.incomeLocal	Decimal(18, 2)	#,##0.00	Accrued income on lot
Broker Code	dataSubLedgerTrans.brokerCode	String	Left-aligned	Lot broker code

Each lot row follows the same Incumbent / BNY / Variance tri-column pattern for the relevant comparison fields.

5.5.3 Basis Lot Check (Secondary Validation)

An additional validation panel available via a tab or toggle, implementing rule 1.6 (Basis Lot Check). This compares shares between the primary accounting basis and non-primary basis for each position to detect basis reconciliation issues.

- LHS: dataSubLedgerTrans WHERE isPrimaryBasis = 'Y' → shares by assetId
- RHS: dataSubLedgerTrans WHERE isPrimaryBasis <> 'Y' → shares by assetId
- Variance: Any difference indicates a tax lot accounting discrepancy

5.5.4 Position Roll-Up Validation

A summary footer validates that the sum of all position-level variances ties to the GL category variance from the Trial Balance:

- Sum of Position Variances: Calculated from all visible position rows
- GL Category Variance: Inherited from Trial Balance context
- Tie-Out: Visual pass/fail indicator. Discrepancies may indicate unmapped positions or GL account classification issues.

5.6 User Actions

Action	Behavior	Navigation
Click position row	Selects position, updates AI panel with position-specific analysis	Stays on Position Drill-Down
Expand position row	Shows tax lot detail table for that security	Stays on page
Click security identifier	Opens security reference detail modal showing refSecurity fields	Modal overlay
Toggle Basis Lot Check	Switches between category view and basis lot comparison view	Tab switch on same page
Click breadcrumb: Trial Balance	Returns to Trial Balance with category context preserved	→ Trial Balance
Click breadcrumb: NAV Dashboard	Returns to NAV Dashboard with fund selection preserved	→ NAV Dashboard
Export to Excel	Exports position grid with lot-level detail	Download initiates

	for all expanded rows	
AI: Request Analysis	Triggers AI agent analysis for selected position/lot	AI panel updates

5.7 AI Commentary Integration

At the position level, AI analysis is most granular and actionable:

- Position-Level Root Cause: Specific explanation per security (e.g., "CUSIP 789456123: Day count convention mismatch. CPU uses ACT/ACT (32 accrual days), Incumbent uses 30/360 (30 accrual days). Expected variance: $15,208.33")
- Lot-Level Anomaly Detection: Flags specific lots with unusual variances relative to peers
- Cross-Position Pattern: Identifies if multiple positions share the same root cause (e.g., all fixed income positions with same day count issue)
- Resolution Recommendation: Specific action items (e.g., "Update day count convention in ElectronDSL rule for security type BOND to ACT/ACT")
- Confidence Scoring: Per-position confidence with evidence chain

5.8 Data Requirements

Position Compare (primary grid):

- LHS: SELECT * FROM dataSubLedgerPosition WHERE userBank = {incumbent} AND account = {account} AND valuationDt = {date} AND isPrimaryBasis = 'Y'
- RHS: SELECT * FROM dataSubLedgerPosition WHERE userBank = 'BNY' AND account = {account} AND valuationDt = {date} AND isPrimaryBasis = 'Y'
- Join: ON valuationDt, account, shareClass, assetId, longShortInd
- Filter: GL category mapping determines which positions are shown based on eagleLedgerAcct classification

Tax Lot Detail (expandable):

- LHS: SELECT * FROM dataSubLedgerTrans WHERE userBank = {incumbent} AND account = {account} AND assetId = {selected}
- RHS: SELECT * FROM dataSubLedgerTrans WHERE userBank = 'BNY' AND account = {account} AND assetId = {selected}
- Join: ON valuationDt, account, shareClass, assetId, longShortInd, transactionId

6. End-to-End Flow Example

This section traces a complete investigation scenario through all four screens, demonstrating the progressive disclosure pattern and data lineage.

6.1 Scenario

A Conversion Manager investigating a NAV break for account AC0002 on valuation date 2026-02-10 within the Vanguard Fixed Income Migration event.

6.2 Step-by-Step Flow

Step 1: Event Dashboard

- User sees EVT-2026-001 (Vanguard Fixed Income Migration) card showing 12 of 15 funds passed, 3 with attention.
- The 7-day sparkline shows increasing break trend.
- User clicks the event card to navigate to NAV Dashboard.

Step 2: NAV Dashboard

- NAV Compare grid loads for valuation date 2026-02-10.
- AC0002 shows: Incumbent TNA = $2,233,943.99, BNY TNA = $2,257,400.99, TNA Difference = ($17,857.00), red status.
- User expands AC0002 row to see inline cross-checks:
  - Ledger BS Compare Check shows TNA Difference of 2,673.56 — indicating an internal BNY ledger discrepancy.
  - Ledger INCST Compare Check shows same 2,673.56 variance.
- AI Commentary panel notes: "Price breaks on bond securities has caused unrealized variance on the L3 position recon, leading to ledger differences in unrealized on the L2 ledger recon."
- User double-clicks AC0002 to drill into Trial Balance.

Step 3: Trial Balance (Ledger Drill-Down)

- Context header shows: AC0002, 2026-02-10, NAV Variance: ($17,857.00).
- Ledger BS Compare grid shows categorized balances:
  - Cash: Incumbent $142,666.00, BNY $138,498.00, Diff $4,168.00, Validation: amber
  - Investment Cost: Incumbent $1,420,817.00, BNY $1,420,620.00, Diff $197.00, Validation: green
  - Investment Urgl: Incumbent $394,423.44, BNY $417,396.02, Diff ($22,972.58), Validation: red
  - Futures Margin: Incumbent $85,935.00, BNY $85,940.00, Diff ($5.00), Validation: green
  - Dividend RecPay: Incumbent $43,698.01, BNY $41,362.00, Diff $1,777.02, Validation: amber
  - Interest RecPay: Incumbent $139,778.00, BNY $143,461.00, Diff ($3,673.00), Validation: amber

    - Expense RecPay: Incumbent ($9,760.01), BNY ($9,758.03), Diff ($1.98), Validation: green
- Waterfall chart visually shows Investment Urgl as the dominant red bar.
- User double-clicks "Investment Urgl" row to drill into positions.

Step 4: Position Drill-Down

- Context header shows: AC0002, 2026-02-10, Category: Investment Urgl, Category Variance: ($22,972.58).
- Position grid shows all securities with unrealized gain/loss comparison:
    - CUSIP 789456123 (Corporate Bond 5.25%): posUnrealizedBase Incumbent = $42,350.00, BNY = $27,150.00, Variance = $15,200.00
    - CUSIP 456789012 (Govt Bond 3.75%): posUnrealizedBase Incumbent = $18,900.00, BNY = $12,100.00, Variance = $6,800.00
- User expands CUSIP 789456123 to see 3 tax lots with individual unrealized variances.
- AI panel shows: "Day count convention mismatch. CPU uses ACT/ACT, Incumbent uses 30/360 for corporate bonds. This affects the accrual calculation which flows through to unrealized. Confidence: 91%. Similar breaks found in Fund DEF (Jan 2026) and Fund GHI (Dec 2025)."

7. Data Pipeline & Validation Execution

7.1 Validation Execution Sequence

When a user triggers "Run Validation" from the NAV Dashboard, the following checks execute in sequence for each fund in the event:

Order	Check	Screen	LHS → RHS	Dependency
1	NAV to Ledger	NAV Dashboard	dataNav → dataLedger	None
2	Ledger BS to INCST	NAV Dashboard (expand)	dataLedger BS → dataLedger INCST	Check 1
3	Ledger TF to Class	Trial Balance	dataLedger TF → dataLedger Class	Check 1
4	Ledger to Subledger	Trial Balance (expand)	dataLedger → derivedSubLedgerRollup	Check 3 + rollup calc
5	Position to Lot	Position Drill-Down	dataSubLedgerTrans → dataSubLedgerPosition	Check 4
6	Basis Lot Check	Position Drill-Down (tab)	dataSubLedgerTrans primary → non-primary	Check 5

7.2 Derived SubLedger Rollup Calculation

The derivedSubLedgerRollup dataset is computed at validation time using the rules defined in Section 2 of the Data Validation Rules. This is the critical transformation layer that enables the Trial Balance to compare GL ledger balances against position-level aggregates. Each rule maps specific canonical model fields to GL account numbers, applying sign conventions and aggregation logic.

The rollup is computed per valuationDt, account, and eagleLedgerAcct. Results are cached in the validation run context and made available to both the Trial Balance grid and the Position Drill-Down for tie-out validation.

8. State Management & Context Propagation

8.1 Redux State Shape

The application uses Redux Toolkit with the following state slices for the drill-down flow:

```
drillDown: {
  context: {
    eventId: string,
    eventName: string,
    account: string | null,
    accountName: string | null,
    valuationDt: string | null,
    category: string | null,
    assetId: string | null,
  },
  navDashboard: {
    funds: Fund[],
    selectedFund: string | null,
    expandedRows: Set<string>,
    crossChecks: Record<string, CrossCheckResult>,
  },
  trialBalance: {
    categories: LedgerCategory[],
    selectedCategory: string | null,
    expandedRows: Set<string>,
    subledgerChecks: Record<string, SubledgerCheckResult>,
    waterfallData: WaterfallBar[],
  },
  positionDrillDown: {
    positions: Position[],
    selectedPosition: string | null,
    expandedRows: Set<string>,
    taxLots: Record<string, TaxLot[]>,
    basisLotCheck: BasisLotResult[],
  },
  aiAnalysis: {
    currentAnalysis: AIAnalysis | null,
    loading: boolean,
    history: AIAnalysis[],
  }
}
```

8.2 Context Propagation Rules

Transition	Context Set	Context Preserved	Context Cleared
Event → NAV	eventId, eventName	—	account, category, assetId
NAV → Trial Bal	account, accountName	eventId, valuationDt	category, assetId
Trial Bal → Position	category	eventId, account, valuationDt	assetId
Any → Back	—	All parent context	Child-level context

8.3 Real-Time Updates

WebSocket subscriptions are scoped to the current drill-down context:

- Event Dashboard: Subscribe to all events in user's scope
- NAV Dashboard: Subscribe to validation updates for current event
- Trial Balance: Subscribe to AI analysis updates for current fund
- Position Drill-Down: Subscribe to AI analysis for current fund + category

When a validation run completes or AI analysis finishes, the affected grid rows update in-place without full page refresh. New break status indicators animate briefly (pulse effect) to draw attention.

9. Performance & Accessibility

9.1 Performance Targets

Metric	Target	Measurement	Screen
Event Dashboard load	< 2s	Time to First Contentful Paint	Event Dashboard
NAV Compare grid render	< 1s	500 fund rows with status	NAV Dashboard
Trial Balance load	< 500ms	30 GL categories with balances	Trial Balance
Position grid render	< 1s	1000 positions with grouping	Position Drill-Down
Tax lot expansion	< 300ms	50 lots per position	Position Drill-Down
Cross-check expansion	< 200ms	Inline validation display	NAV Dashboard
Waterfall chart render	< 200ms	D3.js SVG with 15 bars	Trial Balance
AI panel update	< 300ms	After row selection change	All screens
WebSocket latency	< 200ms	Server event to UI update	All screens

9.2 Accessibility Requirements

- WCAG 2.1 AA compliance across all four screens
- Full keyboard navigation: Tab through grid rows, Enter to expand, Escape to collapse, Arrow keys for cell navigation within AG-Grid
- Screen reader: ARIA live regions for validation status changes, AI analysis updates, and real-time feed items
- Color contrast: 4.5:1 minimum for all text. Break status uses both color AND icon (checkmark, warning triangle, X) for color-blind accessibility
- Focus indicators: Visible focus ring on all interactive elements, follows WCAG 2.4.7

10. Classification Mapping Configuration

The Position Drill-Down and Trial Balance screens rely on three classification mapping tables to correctly categorize positions into GL ledger buckets, determine which comparison columns to display, and route unsettled transactions to the correct RecPay categories. These mappings are source-system-specific (e.g., InvestOne, Eagle) and must be maintainable by operations teams through a dedicated configuration screen in the Control Center.

10.1 Mapping Architecture Overview

The classification pipeline uses three mapping tables in sequence to transform raw position and transaction data into the categorized ledger view shown in the Trial Balance:

- Asset Classification (convAssetClass): Maps each security type code to a canonical asset class (equity, fixedIncome, cash, future, swapTrs, etc.). This determines which GL categories a position feeds into.
- Transaction Classification (convTransClass): Maps each transaction code to a canonical transaction class (longBuy, longSell, dividend, coupon, reclaim, etc.). This determines how unsettled transactions are categorized into RecPay buckets.
- Ledger Category Derivation: The combination of asset class + transaction class determines the GL category assignment (Investment Cost, Investment Urgl, Interest RecPay, Dividend RecPay, etc.) and which comparison columns are shown in the Position Drill-Down.

10.2 Asset Type Classification Mapping

This mapping classifies security type codes from each source system into canonical asset classes. The asset class drives GL category assignment in the Trial Balance and determines which comparison fields are relevant in the Position Drill-Down.

Table: convAssetClassification

key_source	key_secType	convAssetClass	GL Category Impact
investone	S	equity	Investment Cost, Holdings Unrealized
investone	MF	mf	Investment Cost, Holdings Unrealized
investone	TI	fixedIncome	Investment Cost, Holdings Unrealized, Interest RecPay
investone	FI	future	Future Margin
investone	CU	cash	Cash
investone	CA	fixedIncome	Investment Cost, Holdings Unrealized, Interest RecPay
investone	RP	fixedIncome	Investment Cost, Interest RecPay
investone	W	equity	Investment Cost, Holdings Unrealized
investone	R	equity	Investment Cost, Holdings Unrealized

investone	B	fixedIncome	Investment Cost, Holdings Unrealized, Interest RecPay
investone	XX	invalid	Excluded from reconciliation
investone	FH	fixedIncome	Investment Cost, Holdings Unrealized, Interest RecPay
investone	DN	fixedIncome	Investment Cost, Holdings Unrealized
investone	TN	fixedIncome	Investment Cost, Holdings Unrealized, Interest RecPay
investone	G	fixedIncome	Investment Cost, Holdings Unrealized, Interest RecPay
investone	TRS	swapTrs	Swap Income RecPay, RecPay Urgl

Usage: When the Position Drill-Down loads for a given GL category (e.g., Investment Cost), the system uses this mapping to filter dataSubLedgerPosition to only positions whose secType maps to an asset class that feeds that category.

10.3 Transaction Code Classification Mapping

This mapping classifies transaction codes from each source system into canonical transaction classes. The transaction class determines how unsettled transactions are categorized into the RecPay buckets shown in the Trial Balance and Position Drill-Down.

Table: convTransClassification

key_source	key_transCode	convTransClass	RecPay Category
investone	RECL+	reclaim	Reclaim RecPay
investone	RECL	reclaim	Reclaim RecPay
investone	RECL-	reclaimNeg	Reclaim RecPay
investone	SHORT	shortSell	Investment RecPay
investone	COVER	buyCover	Investment RecPay
investone	BUY	longBuy	Investment RecPay
investone	SELL	longSell	Investment RecPay
investone	EXP	ignore	Excluded
investone	WEXP	divWHT	Dividend RecPay
investone	RBRECL	reclaim	Reclaim RecPay
investone	DIV	dividend	Dividend RecPay

10.4 Ledger Category Derivation Matrix

The combination of convTransClass and convAssetClass determines the full set of GL ledger categories each position or transaction feeds into. This matrix drives both the Trial Balance category rows and the Position Drill-Down's context-aware column selection.

Table: Ledger Category Derivation by convTransClass

convTransClass	amntConvCategory	urglBsConvCategory	intRecPayConvCat	intUrglIncstConvCat

buyCover	Investment RecPay	RecPay Urgl	—	—
coupon	Interest RecPay	RecPay Urgl	—	—
dividend	Dividend RecPay	RecPay Urgl	—	—
dividendNeg	Dividend RecPay	RecPay Urgl	—	—
dividendPosAdj	Dividend RecPay	RecPay Urgl	—	—
divWHT	Dividend RecPay	RecPay Urgl	—	—
futurePay	Future RecPay	RecPay Urgl	—	—
futureRec	Future RecPay	RecPay Urgl	—	—
longBuy	Investment RecPay	RecPay Urgl	—	—
longSell	Investment RecPay	RecPay Urgl	—	—
paydown	Investment RecPay	RecPay Urgl	—	—
payup	Investment RecPay	RecPay Urgl	—	—
reclaim	Reclaim RecPay	RecPay Urgl	—	—
reclaimNeg	Reclaim RecPay	RecPay Urgl	—	—
securityLending	Interest RecPay	RecPay Urgl	—	—
securityLendingNeg	Interest RecPay	RecPay Urgl	—	—
shortCoupon	Interest RecPay	RecPay Urgl	—	—
shortSell	Investment RecPay	RecPay Urgl	—	—

Asset Class to Ledger Category Derivation by convAssetClass (for settled positions):

convAssetClass	costConvCat	urglBsConvCat	dailyMarginCat	intRecPayCat	intUrglIncstCat
cash	Cash	—	—	—	—
equity	Investment Cost	Holdings Unrealized	—	—	—
mf	Investment Cost	Holdings Unrealized	—	—	—
fixedIncome	Investment Cost	Holdings Unrealized	—	Interest RecPay	Unrealized INCST
future	—	—	Future Margin	—	—
swapTrs	Investment Cost	Holdings Unrealized	—	Swap Income RecPay	Unrealized INCST

10.5 Unsettled Totals Derivation

The Trial Balance's RecPay categories (Dividend RecPay, Interest RecPay, Investment RecPay, Reclaim RecPay, Future RecPay) aggregate unsettled transactions by their convTransClass-derived category. The Position Drill-Down for these categories shows the individual unsettled transactions grouped by account and category.

Example Unsettled Totals Structure:

Account	Category	transCode	Amount	Notes
1	Dividend RecPay	DIV	682.98	

1	Reclaim RecPay	RECL	13,982.74	
1	Reclaim RecPay	RECL-	-21.69	Negative reclaim adjustment
1	Reclaim RecPay	RECL+	3,105.62	Positive reclaim adjustment
		Total	17,749.65	Ties to Trial Balance Dividend + Reclaim RecPay row
12	Dividend RecPay	DIV	2,990.43	
		Total	2,990.43	
3	Dividend RecPay	DIV	5,834.22	
3	Investment RecPay	BUY	-4,400,277.40	Pending settlement
3	Investment RecPay	COVER	0.00	Zero-amount cover
3	Investment RecPay	SELL	1,810,000.00	Pending settlement

10.6 Configuration Management Screen

The Control Center must provide a dedicated Mapping Configuration screen accessible to Conversion Managers for maintaining these classification tables. This screen is required because different incumbent source systems (InvestOne, Eagle, Geneva) use different security type codes and transaction codes that must be mapped to the canonical classification before reconciliation can execute correctly.

URL & Routing:

```
/admin/mappings?source={key_source}
```

Screen Requirements:

- Source System Selector: Dropdown to select key_source (investone, eagle, etc.). Filters all mapping tables to show only mappings for the selected source.
- Three Tabs: Asset Classification, Transaction Classification, Ledger Category Derivation. Each tab shows an editable AG-Grid with inline editing capability.
- Add/Edit/Delete: Users can add new mappings, edit existing values via inline cell editing, and soft-delete rows (mark inactive rather than hard delete for audit trail).
- Validation Rules: convAssetClass must be one of: equity, fixedIncome, cash, future, mf, swapTrs, invalid. convTransClass must be one of the defined canonical values. Duplicate key combinations (source + secType, source + transCode) are rejected.
- Import/Export: CSV import for bulk loading of mappings during initial source system onboarding. Excel export for review and offline editing.
- Audit Trail: Every mapping change is logged with timestamp, user, before/after values. Changes require confirmation modal before persisting.
- Effective Dating: Mappings support an effective date range so that historical validations continue to use the mapping that was active at the time of the original validation run.

Access Control:

- Conversion Manager: Full CRUD access to all mapping tables
- Fund Accountant: Read-only view access for reference during break analysis

- Auditor: Read-only with export capability

10.7 Impact on Existing Screens

The classification mappings feed directly into the existing drill-down screens:

- Trial Balance (Section 4): The Category column values in the Ledger BS Compare Grid are derived from the convAssetClass mapping. Each position's secType is looked up in convAssetClassification, and the resulting convAssetClass determines which GL category row the position's values roll up into.
- Position Drill-Down (Section 5): The Comparison Columns by Category Context table (Section 5.5.1) uses the convAssetClass to determine which fields to display. For example, fixedIncome positions show interest accrual fields while equity positions do not.
- Position Drill-Down RecPay Categories: When drilling into Dividend RecPay, Interest RecPay, or Investment RecPay from the Trial Balance, the Position Drill-Down uses convTransClassification to filter and group the unsettled transactions from dataSubLedgerTrans.
- Derived SubLedger Rollup (Section 7.2): The rollup calculation uses both mappings to correctly assign position and transaction values to GL account numbers for the Ledger-to-Subledger validation.

END OF DOCUMENT

RECON-AI Process Flow Specification v1.0 – February 2026