Wevo GL Analytics

Business Introduction

Near-Real-Time Hierarchical GL Reporting

1. Purpose

This document provides a business introduction to Wevo GL Analytics and how it transforms General Ledger reporting for finance teams and business users. It is intended for business stakeholders, finance managers, and technical teams who need to understand what Wevo does, how it works with GL systems, and the value it delivers.

Wevo GL Analytics is a near-real-time hierarchical reporting platform that allows organisations to define multi-dimensional views of their financial data, navigate from high-level summaries down to segment-value-level detail, and produce management reports that reflect the true and current structure of the business.

2. What is Wevo GL Analytics

Wevo GL Analytics is a hierarchical reporting solution, also called responsibility reporting, designed specifically for General Ledger systems. It solves a fundamental problem in financial reporting: GL data is stored as flat, linear transactions, but finance teams need to view and report on that data through the lens of organisational structures, product lines, cost centres, and management hierarchies in a human-readable way.

Traditional GL systems produce fixed reports based on predefined account groupings. Wevo goes beyond this by allowing users to define logical hierarchies that model how the business actually operates, then match those hierarchies together to create multi-dimensional analytical views of the financial data. The result is a flexible reporting system where users can drill down from high-level summaries to the lowest level of segment-value detail, navigate across multiple dimensions, and produce customised reports without requiring IT intervention.

2.1 Key Capabilities

- Define unlimited hierarchies that reflect organisational structures, product groups, cost centres, or any business dimension
- Match hierarchies together to create multi-dimensional analytical cubes
- Drill down from summary nodes to segment-value-level detail, near-real-time and minutes behind the GL
- Compare actuals against budgets with variance analysis
- Navigate between nodes horizontally and vertically to explore data from different perspectives
- View data across multiple record types, years, versions, periods, and currencies
- View GL balances, movements since the last full build, or daily movements - with the cube current to within minutes of GL activity
- Hold budgets alongside actuals, summarised through the same hierarchies, with variance at every node and multiple budget versions side by side
- Carry the close: 16 periods including 4 close periods, so year-end and audit adjustments flow through every hierarchy automatically
- Restructure freely - prior years instantly read through the new hierarchy for like-for-like comparison, while each year's structure is retained for history as originally reported
- Deliver formatted reports by email on request to any authorised user, with no licence seat and no system access required
- Receive the complete source code at installation under the Wevo Continuity Licence
- Monitor key organisation and account node pairs on a watchlist - a time-stamped snapshot of movements and balances is written on every engine run, building a longitudinal history for trend and anomaly analysis
- Run both a development and a production instance - both included in the licence - to prepare and rehearse complex changes safely before they reach the live system

- Maintain a Wevo database kept fully synchronised with the GL, so that reporting is a fast read against pre-aggregated data at any level of detail, with no on-demand computation against the source system

3. How Wevo Works with GL Systems

Wevo does not replace your General Ledger system. It complements it by providing advanced hierarchical reporting and analysis capabilities that sit on top of your existing GL database.

The GL system remains the authoritative source of transactional data. Wevo receives extracts of GL balances and movements through a batch engine that processes received CSV files automatically. These files contain the financial data along with the segment values (dimension codes) that classify each transaction - for example, Business Unit codes, Cost Centre codes, and Chart of Accounts codes.

3.1 The Data Flow

The diagram below illustrates the end-to-end data integration model. The GL system remains the authoritative system of record throughout; Wevo operates as a read-only analytical layer, receiving periodic extracts through a standard file-based interface with no direct connection to the source database.

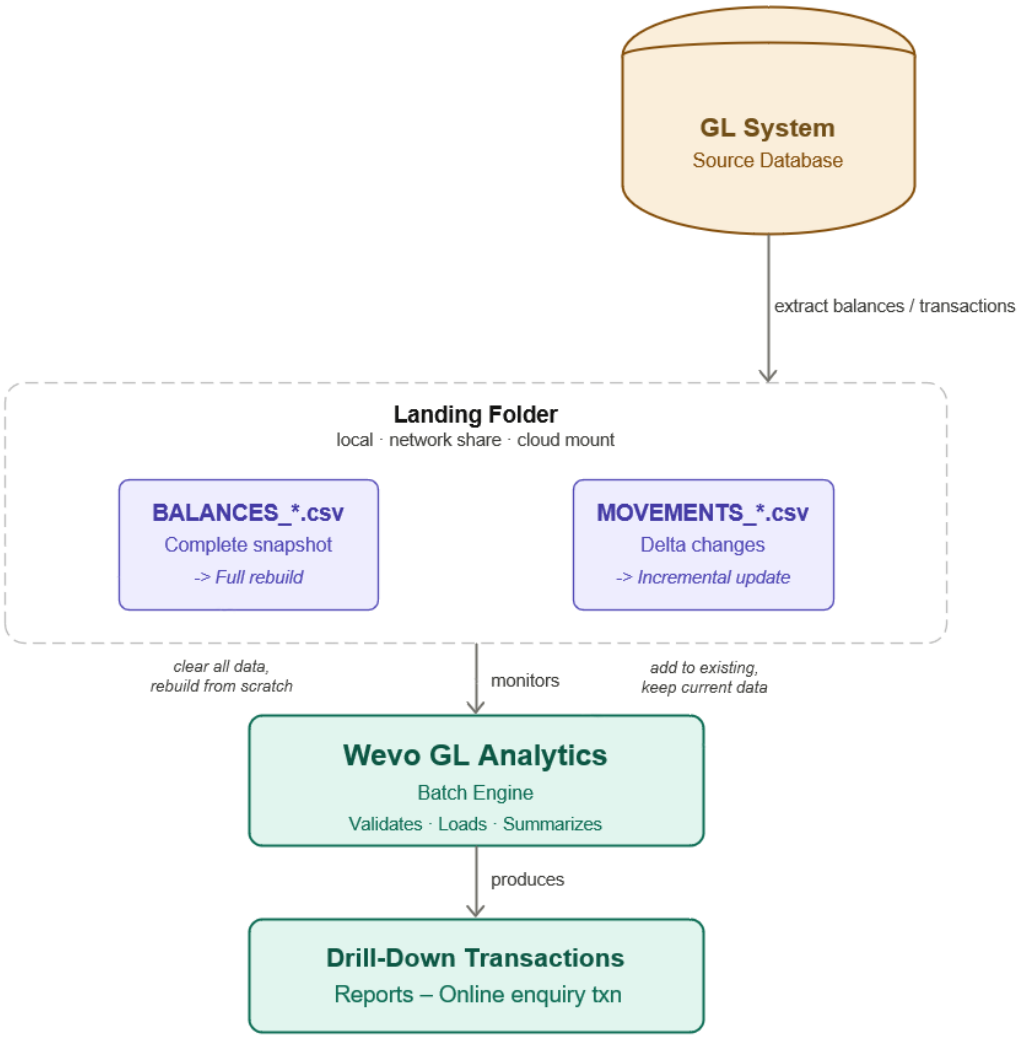


Figure 1: Wevo GL Analytics - data integration architecture

The interaction between Wevo and the GL system follows a simple pattern:

Step	What Happens
1. GL Extract	The GL system extracts balances and movements to CSV files and places them in a designated landing folder.
2. Batch Engine	Wevo's batch engine monitors the landing folder, validates incoming files, and loads the data into the Wevo database automatically.
3. Summarisation	The batch engine aggregates transaction data up through the defined cubes (matching hierarchies), creating summarised views at every node level.
4. User Access	Users access Wevo through a web interface or reporting system to view the summarised data, drill down to transactions, and navigate across hierarchies.

The GL system continues to operate independently. Wevo receives periodic snapshots of the data and processes them into analytical views. This design ensures that Wevo does not impact GL system performance and allows the finance team to maintain full control over their source-of-truth GL databases.

4. Understanding Hierarchies

Hierarchies are the foundation of how Wevo organises and presents financial data. A hierarchy is a tree-like structure of parent-child relationships that models a business dimension-for example an organisational structure, a chart of accounts, or a product line.

4.1 GL Key Structure

Each entry in a General Ledger consists of a key and data segments that together store the financial transactions of an organisation. Understanding this structure is essential to understanding how Wevo processes and organises GL data.

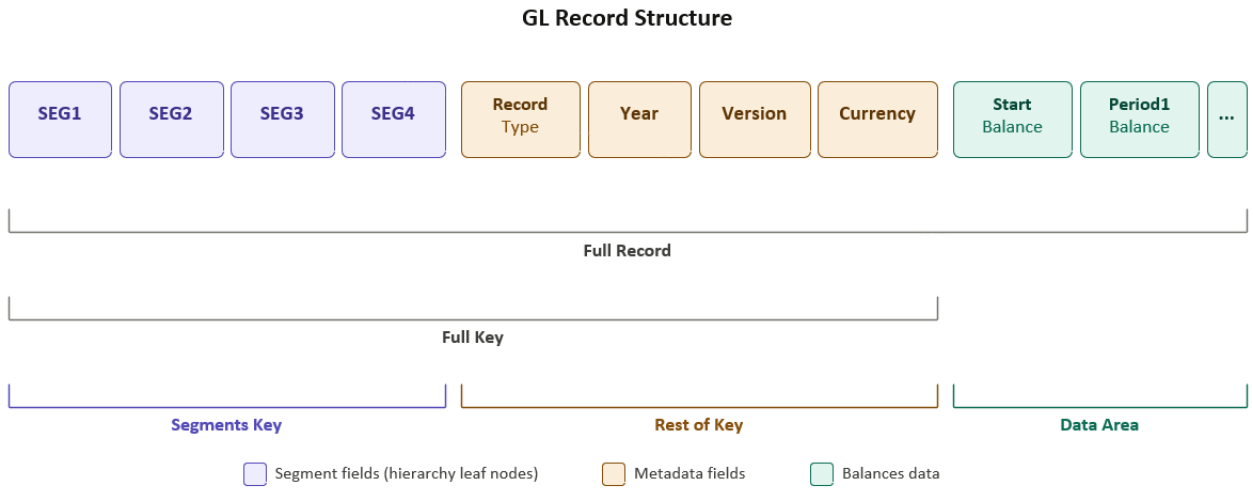


Figure 2: General Ledger key structure - segment fields, record type, year, version, currency, start balance, and period value layout

A typical General Ledger key is defined with Accounting Control Key segments (SEG1, SEG2, SEG3, SEG4, etc.) that provide the logical separation of transactions. Each segment represents a dimension of the business - for example, Business Units, Cost Centres, Regions, Products, or

Financial Account Codes. The number and meaning of segments vary by organisation, but the principle remains the same: each segment classifies the transaction from a different perspective.

Record Type defines the different types of entry in the GL - for example, Actuals (posted transactions), Budgets, Forecasts, or Statistical data. This allows the same hierarchies to be used for analysing different categories of financial information side by side.

Year represents the calendar year for which the data applies. Wevo can hold multiple years of data simultaneously, allowing users to compare current performance against prior years using the same hierarchy structures.

Version is a numeric identifier that differentiates between multiple versions of the same record type and year. For example, you might have Version 1 for the original budget, Version 2 for a revised budget, and Version 3 for a final approved budget. This provides flexibility for scenario analysis and what-if modelling.

Currency indicates the currency in which the transaction amounts are denominated. Wevo supports multi-currency environments where transactions can be recorded in both a base currency (for consolidated reporting) and a source currency (the original transaction currency). This is critical for multinational organisations that need to consolidate financials across multiple currencies.

Start Balance holds the opening balance carried forward from the previous year. This ensures continuity when reporting across year boundaries and allows period-based reports to show accurate year-to-date totals.

Period Values contain the transactional amounts for each period within the year. Periods are defined by the client according to how their GL system operates - typically 12 monthly periods, 13 four-weekly periods, or 52 weekly periods. Each period holds the net movement or balance for that time interval.

When Wevo receives GL data, it uses these key segments to validate transactions against the defined hierarchies, aggregate values up through the hierarchy structures, and create the multi-dimensional analytical cubes that users navigate through the web interface.

4.2 Hierarchy Structure

A hierarchy consists of multiple horizontal levels, with nodes positioned at each level representing logical groupings. In a validated hierarchy, every node is logically connected to a parent in parent-child relationships (except the top node called the root node).

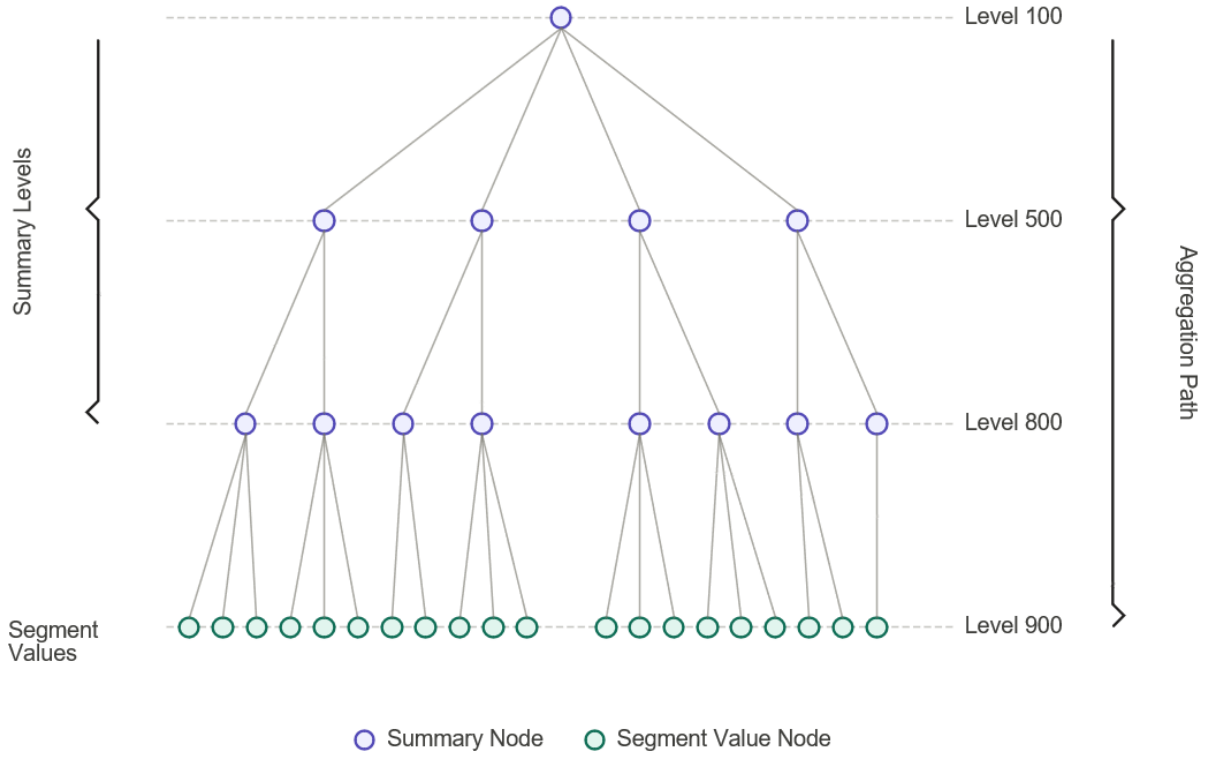


Figure 3: Example three-level hierarchy - summary nodes at upper levels aggregate the segment value nodes at the base

A hierarchy is designated to a particular segment of the GL accounting control key, identified by a segment number such as SEG1, SEG2, and so on. Each segment represents a specific dimension of the chart of accounts - for example, SEG1 might represent Cost Centre, SEG2 might represent Natural Account, and SEG3 might represent Legal Entity.

Leaf nodes - GL segment values

At the bottom level of the hierarchy sit the leaf nodes. Each leaf node represents a single, real GL segment value - that is, the actual code that appears on journal lines posted in the source GL system. The node name (ID) in Wevo is identical to the segment value code in the GL; there is no transformation or aliasing. If the GL defines Cost Centre codes CC001, CC002, and CC003, then those three codes appear as leaf nodes with exactly those names.

Summary nodes - arbitrary aggregation points

Above the leaf nodes sit one or more levels of summary nodes. These are Wevo-defined groupings that have no counterpart in the GL itself - they exist purely to aggregate their child nodes for reporting purposes. Their names (IDs) are arbitrary, chosen by whoever built the hierarchy, and must be unique within that hierarchy. A Cost Centre hierarchy might group CC001 and CC002 into a summary node called LONDON, group CC003 into MANCHESTER, and then roll both LONDON and MANCHESTER up into a single root node called ALL-COST-CENTRES.

Worked example: Cost Centre hierarchy for SEG1

The GL defines three cost centre codes: CC001, CC002, CC003. A Wevo hierarchy called MGT-CC is built for SEG1 as follows:

```
ALL-COST-CENTRES     (root summary node, Wevo-defined)
+-- LONDON           (summary node, Wevo-defined)
|   +-- CC001        (leaf node = GL segment value)
|   +-- CC002        (leaf node = GL segment value)
+-- MANCHESTER       (summary node, Wevo-defined)
    +-- CC003        (leaf node = GL segment value)
```

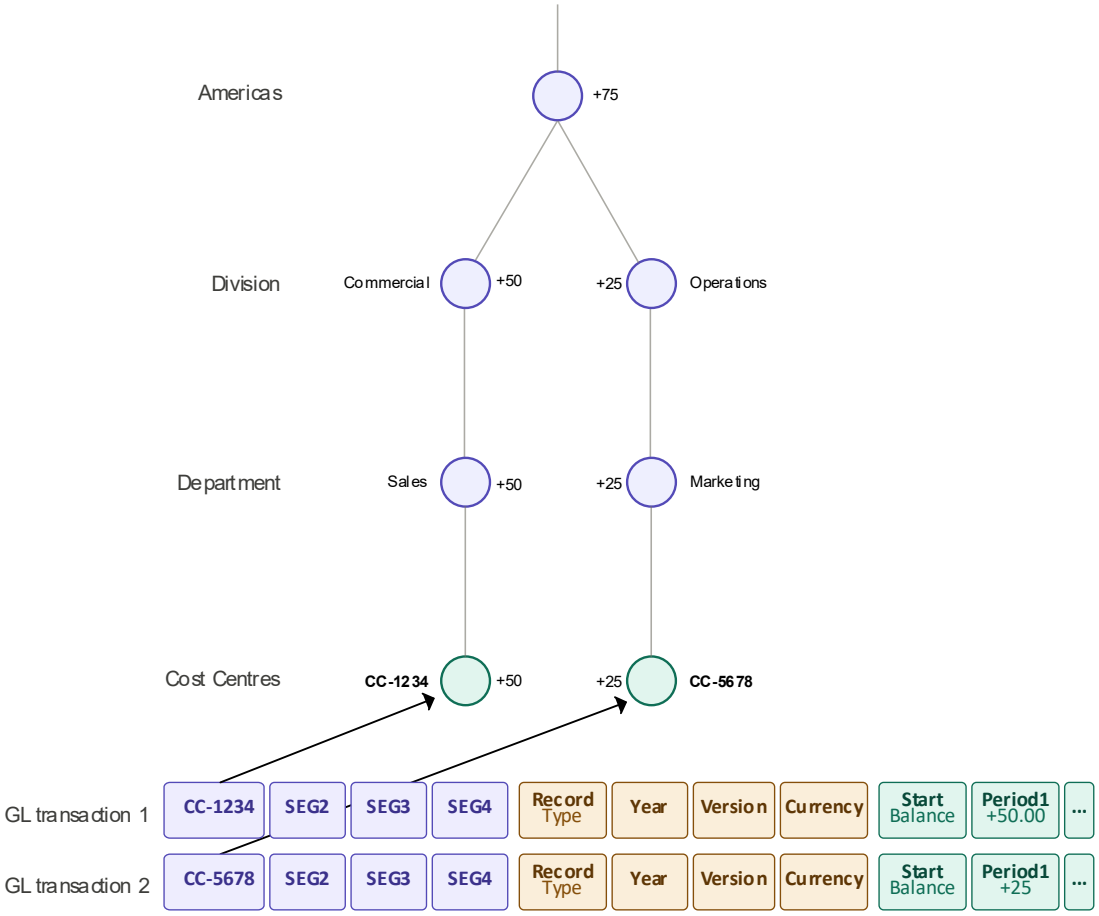


Figure 4: Segment value alignment - leaf-level node names match source GL segment values exactly, providing the linkage between hierarchy structure and GL transaction data

For example, a Cost Centre hierarchy might hold segment values at the bottom (CC-1234, CC-5678), department nodes in the middle (Sales, Marketing), and division nodes at the top (Commercial, Operations). When transaction data flows in, amounts posted to CC-1234 roll up to Sales, which rolls up to Commercial, creating a natural aggregation path. Because each leaf node carries the same code as the corresponding value in the source GL, the leaf node is the precise point at which Wevo matches incoming GL transactions to the hierarchy, allowing their values to be rolled up through every parent node above.

4.3 Matching Hierarchies Together

The real power of Wevo comes from matching two hierarchies together to create a multi-dimensional analytical cube. When you create a cube consisting of a Cost Centre hierarchy with a Chart of Accounts hierarchy, you create a two-dimensional grid where every node on one hierarchy intersects with every node on the other.

The diagram below illustrates a two-hierarchy analytical cube. Every node on the Cost Centre axis intersects with every node on the Chart of Accounts axis, creating a navigable two-dimensional financial view. Users drill down or navigate horizontally within either hierarchy to focus the data view.

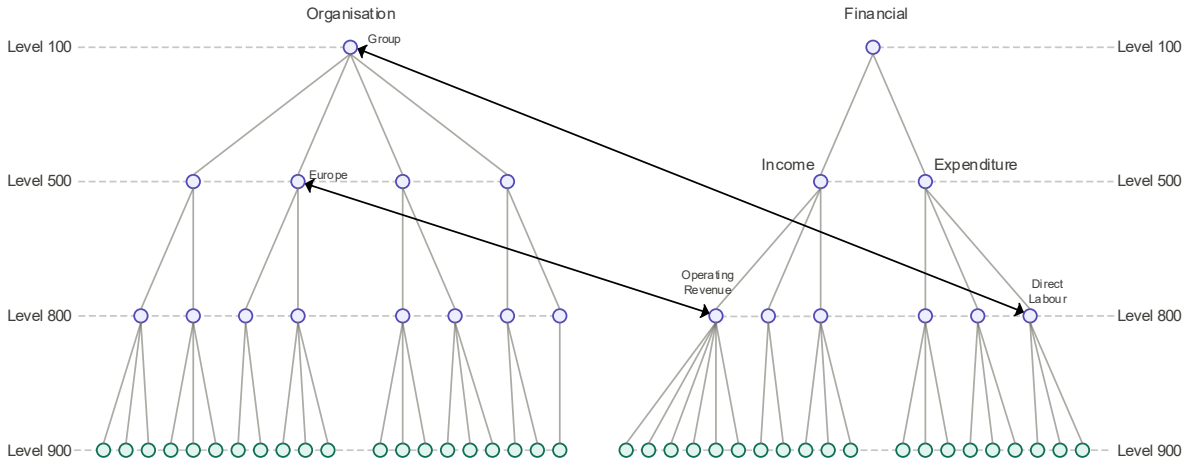


Figure 5: Two-hierarchy analytical cube - intersecting Cost Centre and Chart of Accounts hierarchies

This gives you the ability to answer questions like:

- What is total Operating Revenue for Europe?
- What are the Group Direct Labour costs?
- The value at any node-to-node intersection between the two hierarchies.

The data can be pivoted across many further combinations: financial values shown side by side across multiple years, different record types, accumulated period balances, net movements between periods, individual period values, and budget comparisons - for GL balances, weekly movements (or whatever interval is defined at installation), and daily movements.

Each intersection point in the cube represents a specific view of the data, and users can navigate between these views instantly by drilling up or down within a hierarchy, or by moving horizontally to a different node.

4.4 Hierarchy Lifecycle and Promote Status

Because finance teams maintain hierarchies themselves, Wevo builds governance into the maintenance lifecycle. Changes are developed and validated in a maintenance state - a hierarchy can be copied, restructured, and tested freely - while the production hierarchies continue to serve users and the summarisation engine, completely unaffected.
When a changed hierarchy is complete and passes validation, it is set to promote status: approved and awaiting deployment. On the next full balance build, promoted hierarchies are copied to production and the new structure takes effect in a single controlled event. Complex reorganisations can therefore be prepared over days or weeks, at any time, with production untouched until the moment of deployment.

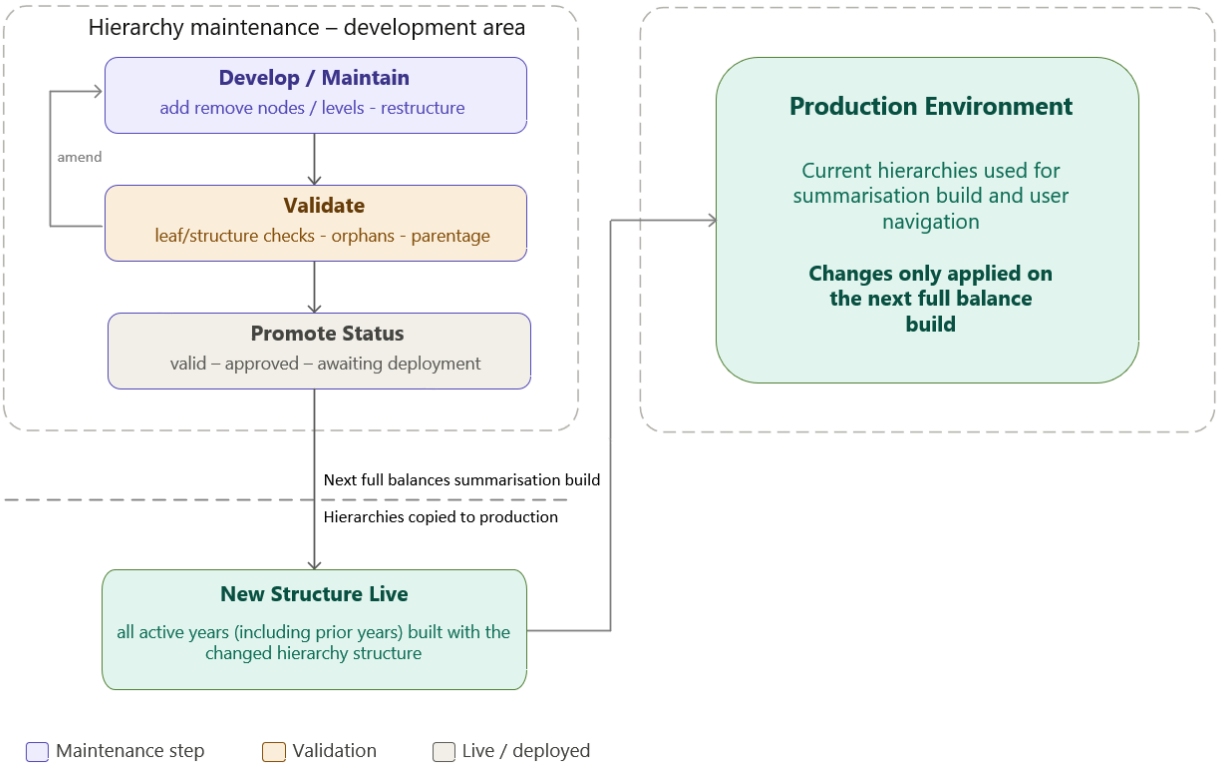


Figure 6: Hierarchy promote-status lifecycle - changes developed, validated, and promoted without affecting production

4.5 Year-Versioned Cubes and Hierarchies: As-Is and As-Was Reporting

At a defined point in the financial year - typically year end - an organisation can copy its current group cube(s) and hierarchies, which point to the current year, and embed that year in the cube and hierarchy names. Once the copied hierarchies are set to promote status, the cube is rebuilt in production at the next full build run. If these copied cubes and hierarchies are left unchanged once the new year begins, they preserve a fixed snapshot of the organisation as it stood for the previous year - and earlier years can be retained in the same way. Because the structure is frozen, the copied hierarchy keeps the leaf-node settings exactly as they were defined at that point in time; each full build processes only that year, producing a faithful snapshot of the organisation as it was then.

This gives finance both answers to the comparability question. Because financial data is stored at segment-value level and aggregated through the current structure, prior-year data is displayed through today's hierarchy - so after a restructure, an acquisition, or a disposal, year-on-year comparisons are instantly like-for-like, with no restatement exercise and no mapping tables. And when the question is "what did we report at the time?", the retained prior-year hierarchy presents

that year's data through the structure as it stood - as-was reporting for audit and review, from the same underlying data.

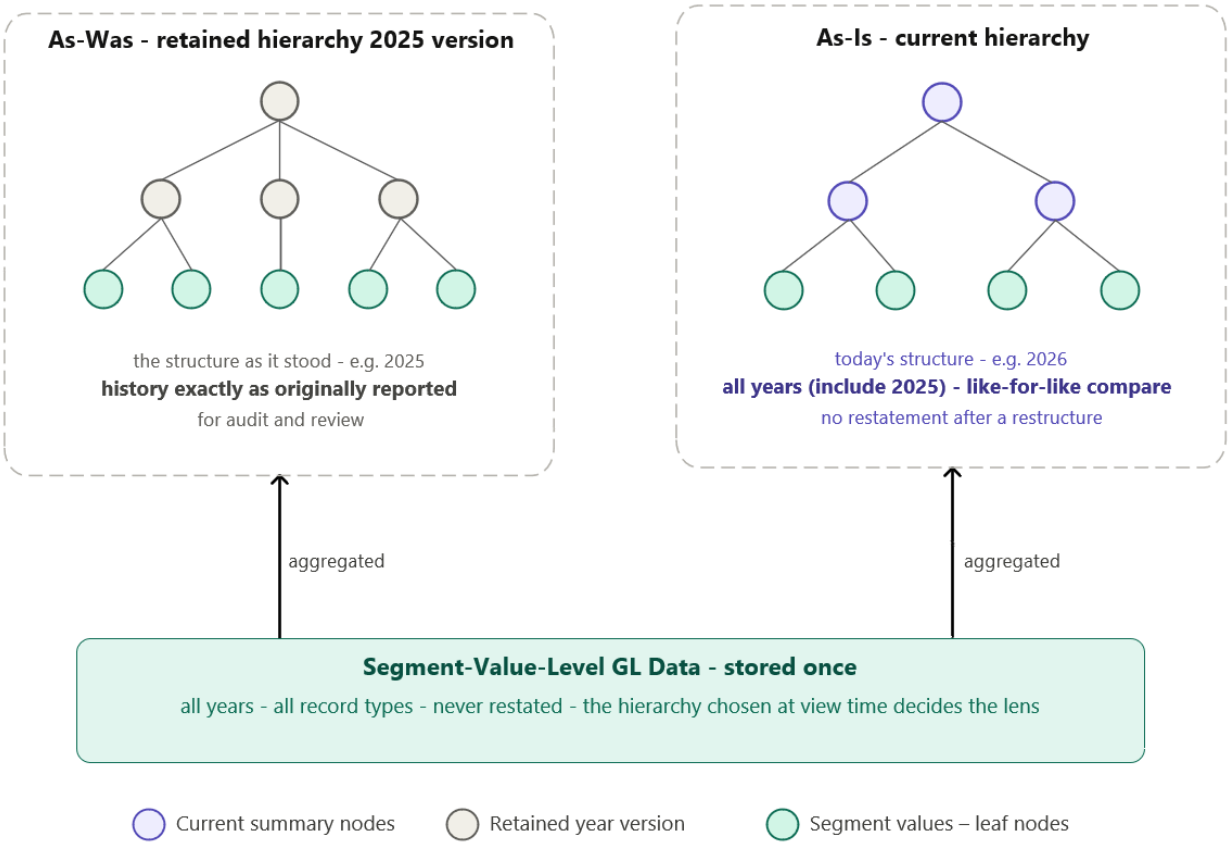


Figure 7: As-is and as-was reporting - one set of segment-value data, viewed through the current hierarchy or a retained year version

5. The Wevo Web Interface

Users interact with Wevo through a browser-based web application. The interface is designed for finance professionals who need fast access to summarised data and the ability to navigate from high-level summaries through to segment-value-level detail, without waiting for IT to run reports - making current figures available instantly in board meetings to support executive decision-making.

5.1 Cube Selection and Navigation

When a user logs in, they select which analytical cube they want to view (for example Cost Centre vs Chart of Accounts), choose the record type (Actuals or Budget), select the year and periods, and pick the currency. The system then displays the summarised data at the intersection of the two hierarchies.

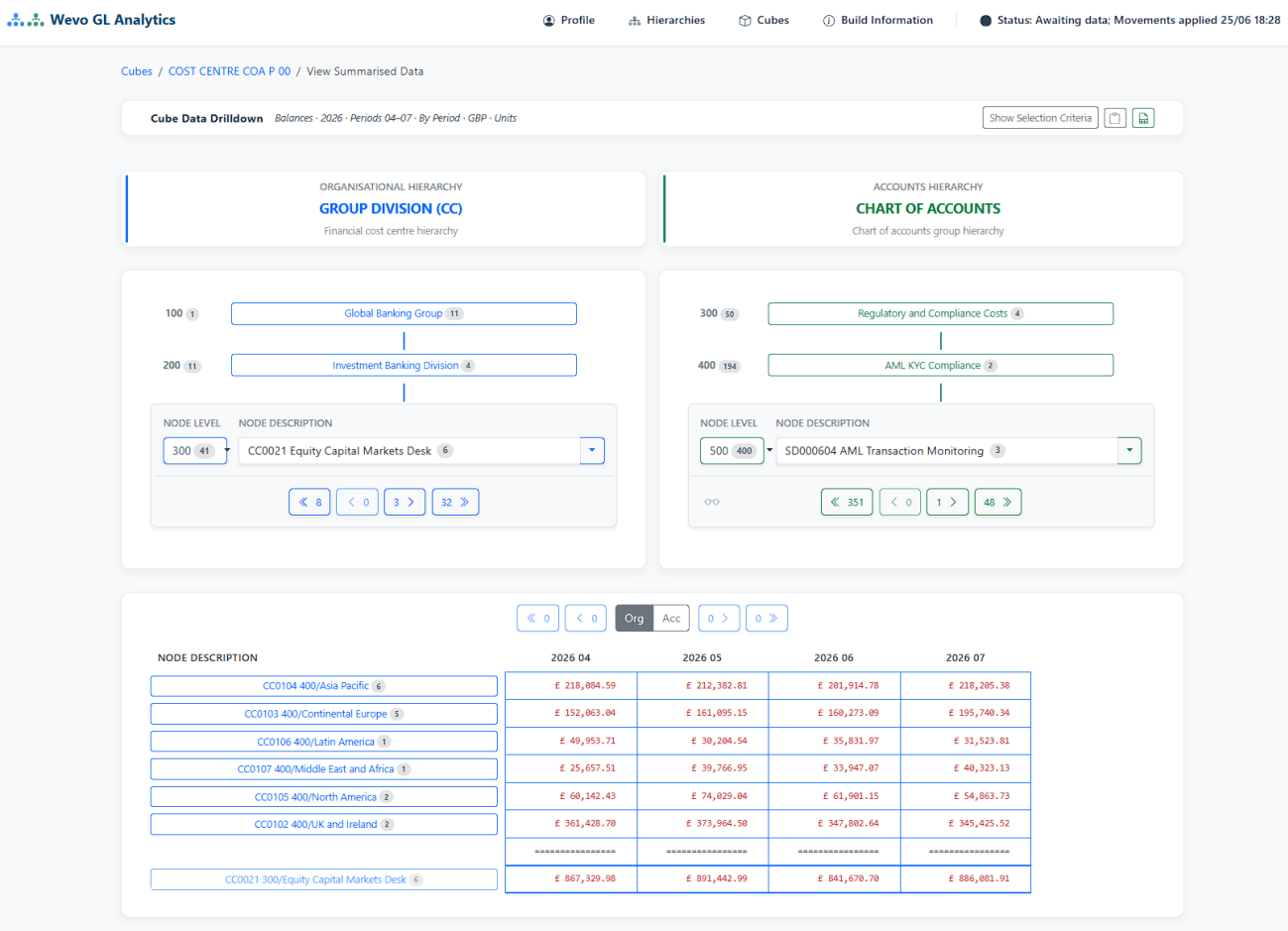


Figure 8: Cube selection - record type, year, period, and currency parameters are configured before entering the analytical view

Once inside the cube, users navigate the financial data by selecting nodes within either hierarchy. The view updates instantly as the user drills down through hierarchy levels or moves laterally to a different node.

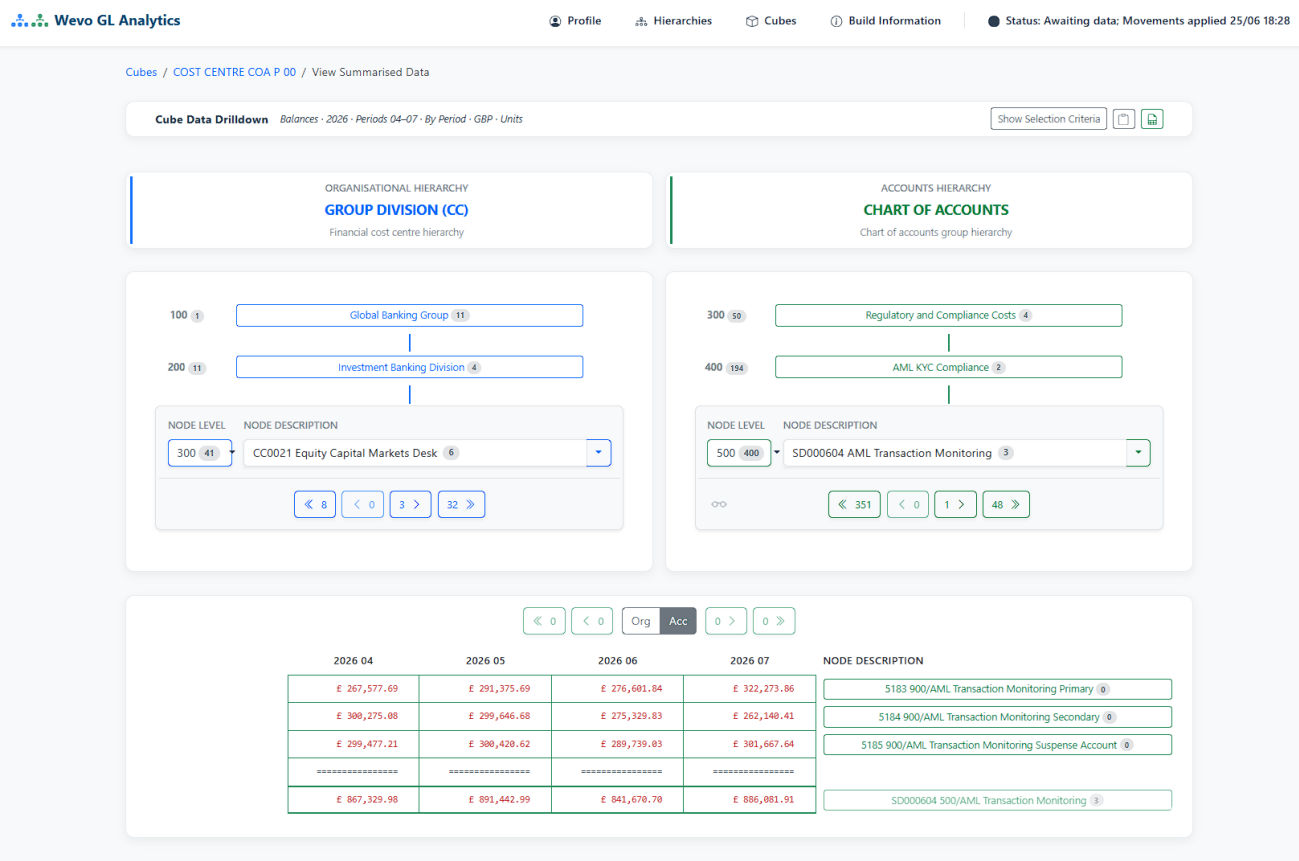


Figure 9: Cube navigation - drill-down interface showing summarised node values at the intersection of two hierarchies

Users navigate through the data by clicking on summary nodes to drill down to lower levels, or by clicking on different nodes to shift the view horizontally. Budget variance tooltips appear when hovering over values, showing the difference between actual and budget amounts. The interface supports dual-hierarchy navigation (H1 and H2) and allows users to switch between movement and balance views.

5.2 Selection Criteria

The Selection Criteria panel is accessible at any time and controls the precise data view presented within the cube. Users adjust the criteria without leaving the current navigation context. The following parameters are available:

- **Data Type** - Determines whether the displayed values represent GL balances, movements accumulated since the last full build, or movements recorded during the current day only.
- **Time Frame** - Selects the financial year or years to be displayed, enabling side-by-side comparison of data across multiple years using identical hierarchy structures.
- **Periods** - Specifies the accumulation range, from the year-start opening balance through to any nominated GL period. A Wevo year holds an opening balance plus 16 periods (including 4 close periods), giving 17 columns per year.
- **View Options** - Controls how period values are presented: as year-to-date accumulated balances, as the net sum of values between two selected periods, or as individual period values.
- **Currency** - Filters the data to a specific currency or combination of currencies, supporting both single-currency and consolidated multi-currency views.
- **Scaling** - Adjusts the unit of display between Units, Thousands, and Millions to present values at the appropriate level of detail for the reporting context.
- **Notation** - Sets the sign convention applied to debit and credit amounts, allowing users to present values in whichever accounting notation is standard for their organisation.

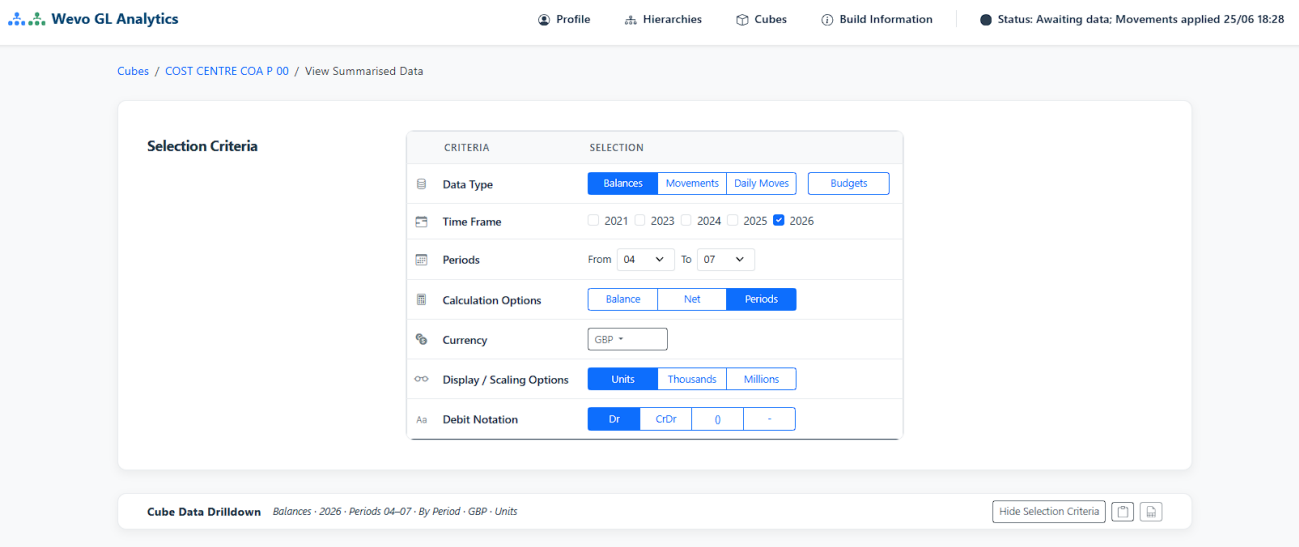


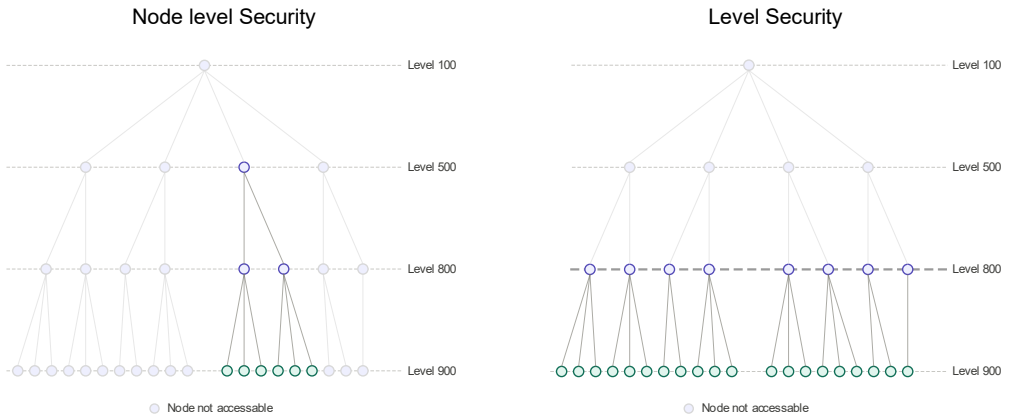
Figure 10: Selection Criteria panel - data type, time frame, period range, view options, currency, scaling, and notation controls

5.3 Drill-Down to Transaction Detail

From any summary node in the cube, users can drill down through the hierarchy levels to explore increasingly granular views of the financial data. Navigation continues until the lowest available level - the intersection of two segment value nodes - which displays the accumulated values for that specific combination across the selected record type, year, periods, and currency. This gives users a precise, targeted view of where values originate within the hierarchy without needing to interact with the GL system directly. For organisations that require visibility of the underlying individual GL transactions that compose those accumulated values, a direct query to the source GL system would be required.

5.4 Data Security

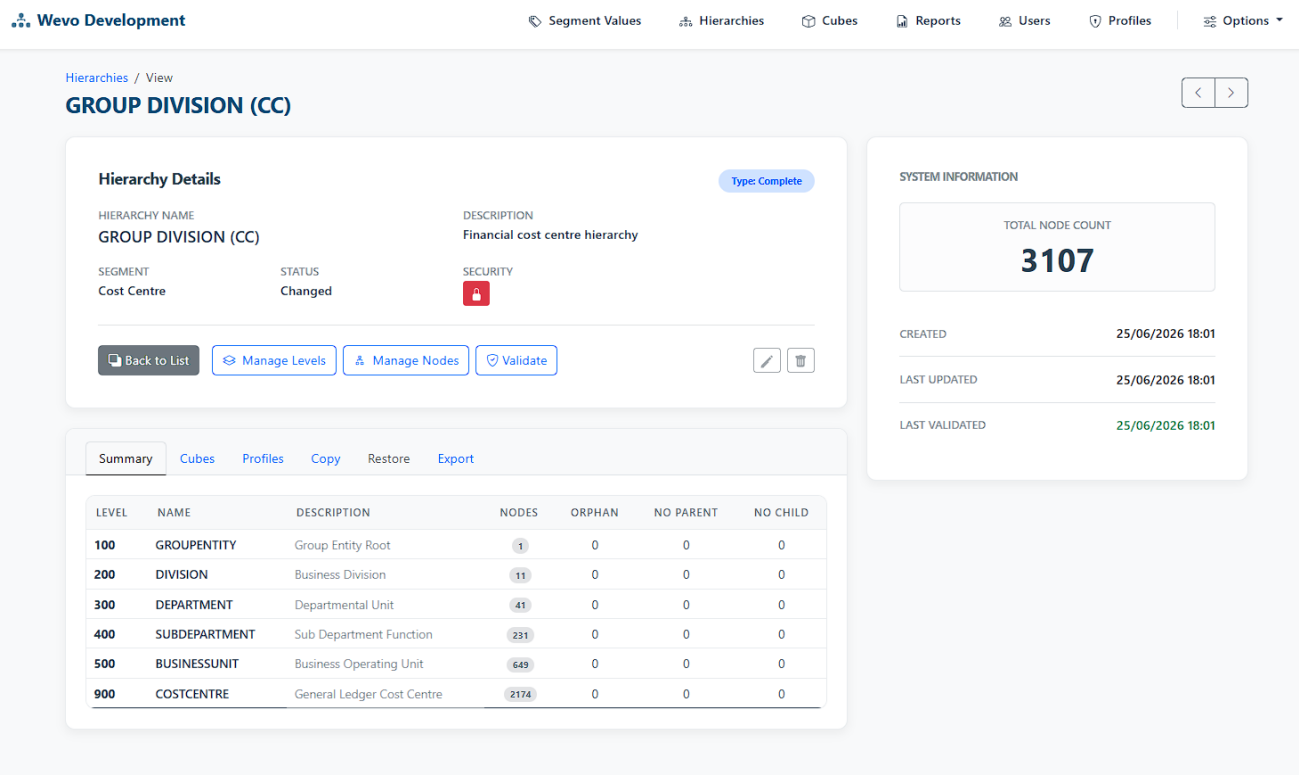
Wevo controls data access at the hierarchy level through security profiles assigned to each user. A profile defines which hierarchies a user can navigate and at what scope within it. There are two profile types: a Node profile, which restricts the user to a defined root node and its descendants, and a Level profile, which grants access to all nodes at a specified level and the sub-levels beneath them. A user can only view cube data where both hierarchies fall within the bounds of their assigned profile, ensuring that access to financial data is precisely controlled without requiring separate data extracts or report variants.



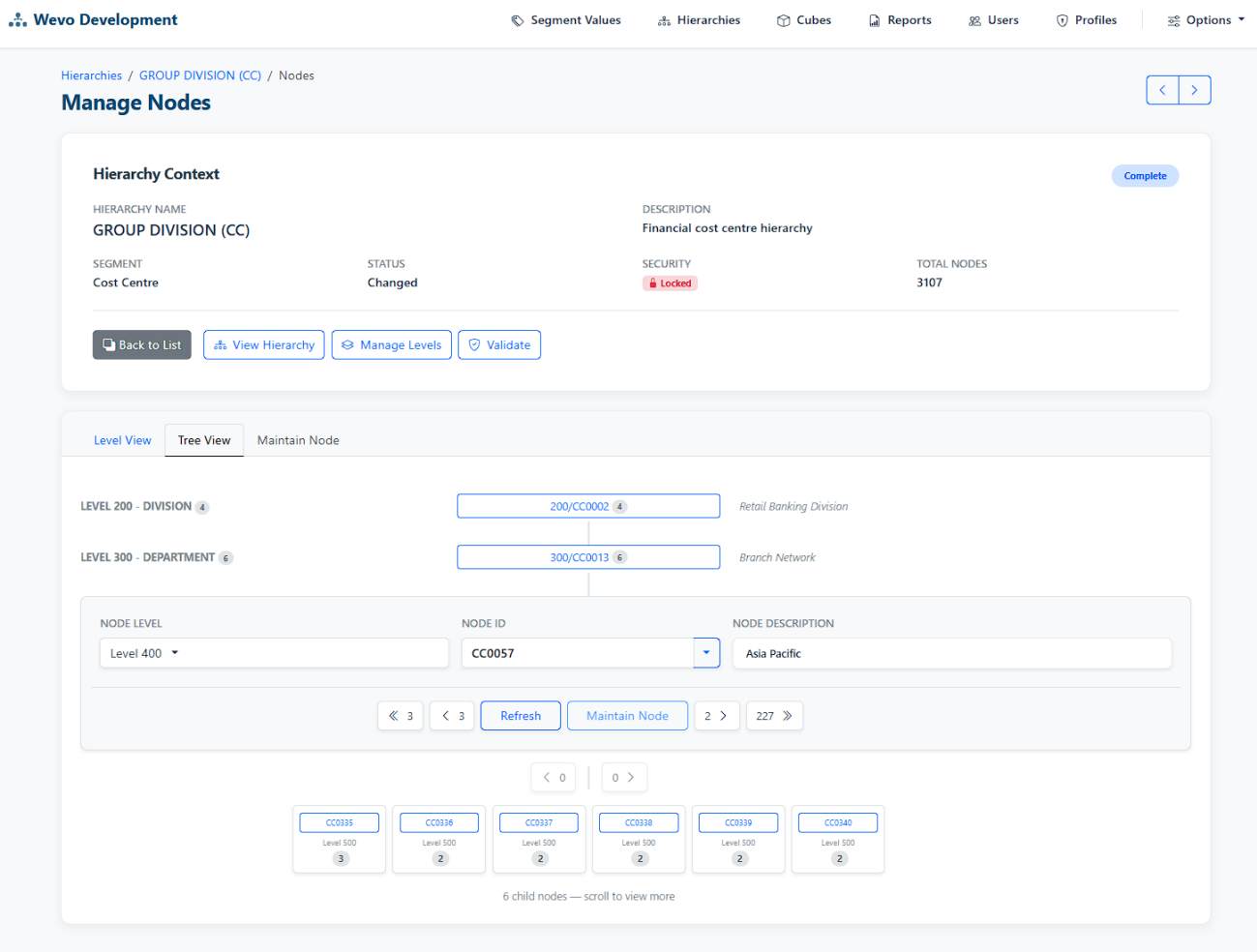
The diagrams above illustrate the two security profile types. Nodes outside the user's permitted scope are shown in grey and are inaccessible; permitted nodes are highlighted in colour.

5.5 Hierarchy Maintenance

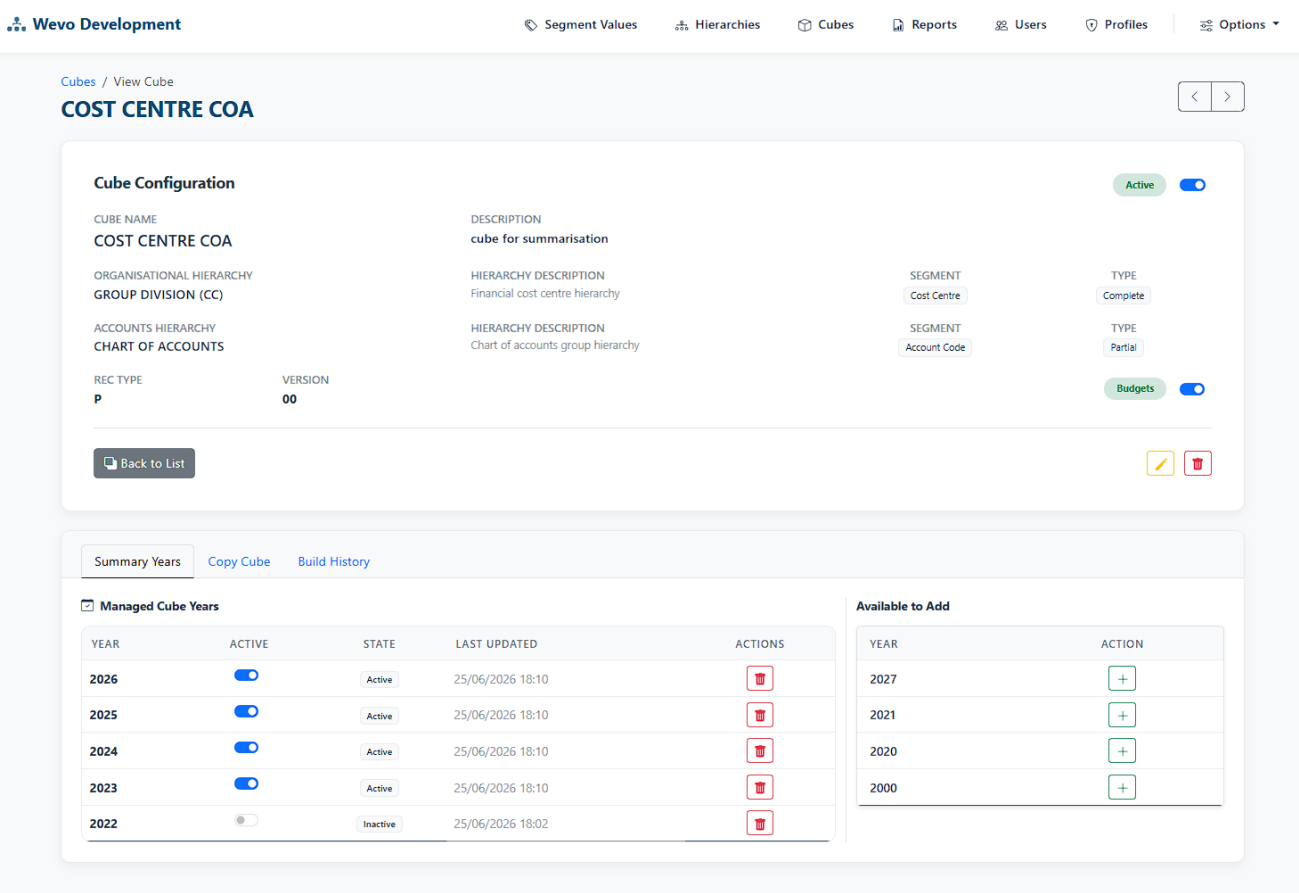
Wevo includes a separate maintenance application (CubeMaint) that allows administrators to create and modify hierarchies, add new nodes, link parent-child relationships, and validate hierarchy structures. This ensures that hierarchies can be updated to reflect organisational changes without requiring developer intervention.



The hierarchy maintenance screen allows administrators to view and edit hierarchy properties, including name, description, and the associated GL segment.



Parent-child node relationships within the hierarchy can be created, modified, and validated to ensure structural integrity before the hierarchy is published.



The cube maintenance screen allows administrators to view and configure cube definitions, including the two hierarchies that form each analytical cube and their associated data settings.

5.6 Budgets, Versions and the Close

Budgets load through the same file interface as actuals and are summarised through the same organisational and account hierarchies. Actual, budget, and variance therefore sit side by side at every node - like for like by construction - with budget variance shown against any figure in the drill-down view. Multiple budget versions (for example original, revised, and final approved) are held simultaneously for scenario comparison.

The Wevo year carries 16 periods, of which 4 are close periods. Year-end adjustments, audit adjustments, and post-close corrections are posted into the close periods and flow through every hierarchy exactly as trading-period postings do - the close calendar lives inside the product, not in a spreadsheet workaround alongside it.

5.7 Node Pair Watchlist Monitoring

Beyond on-demand enquiry, Wevo can build a longitudinal record of how key figures evolve over time. Any organisation and account node pair - an intersection of the two hierarchies - can be placed on a watchlist for a cube. Thereafter, every time the summarisation engine runs and data moves for that pair, Wevo writes a time-stamped snapshot: the opening position, all 16 period movements, and the corresponding 16 period balances, captured across every active cube year and every currency combination present in the data.

Node pairs are added by org node name and account node name through an administration screen. Because the watchlist stores names rather than internal indexes, an entry survives hierarchy rebuilds where node indexes may change. A pair whose org or account node no longer exists in the validated hierarchy is automatically soft-deleted and removed by a scheduled cleardown job, so the watchlist is self-maintaining. During ordinary drill-down, a pair of watchlist icons against the node

pair in focus indicates when that intersection is being monitored, giving an at-a-glance view of which figures are accumulating history.
Snapshots accumulate into a granular, time-stamped history that is available for export and offline analysis - trend analysis, anomaly detection, period-on-period variance, and AI-driven insight - without any additional query to the source GL. In effect, the watchlist turns Wevo from a point-in-time reporting tool into a passive, zero-effort data-collection layer for the financial indicators that matter most, accumulating value the longer it runs.

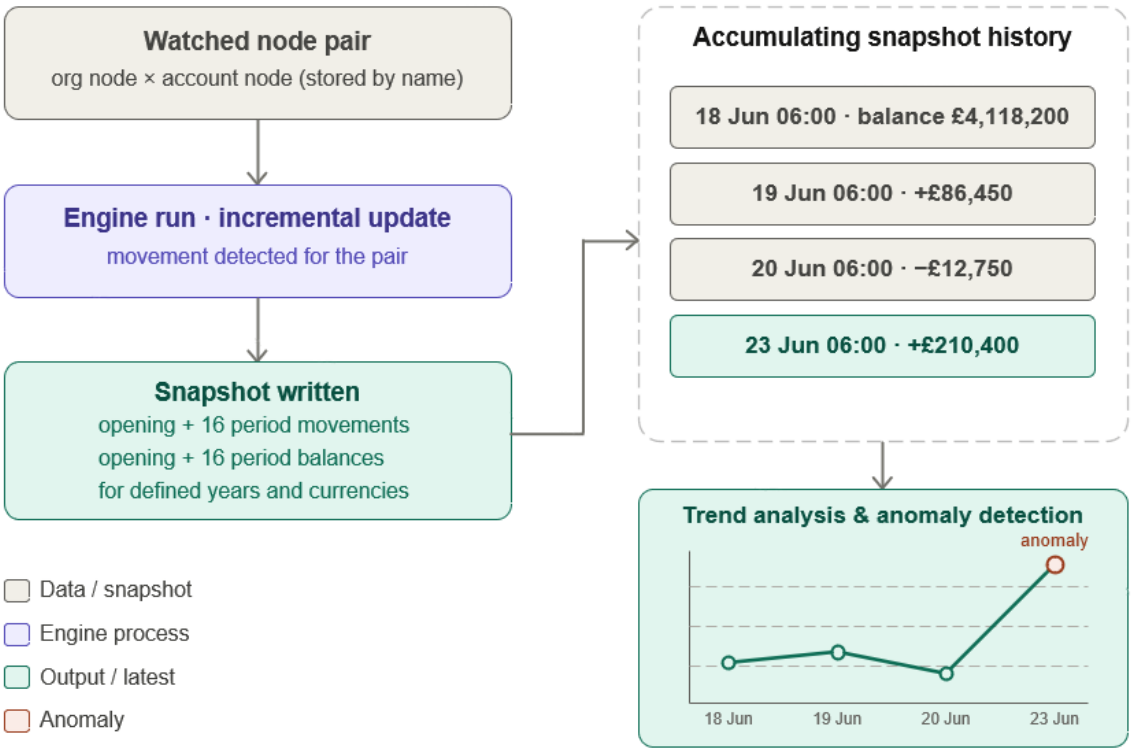


Figure 17: Node pair watchlist - a snapshot of movements and balances is written on every engine run for each watched intersection, building a longitudinal history available for export

6. Fixed Reporting System

To support stakeholders who do not require full access to the live analytical interface-whether due to security protocols or a need for a simplified data view-Wevo provides an automated, email-based reporting service. At scheduled intervals (e.g., every Monday morning), the system delivers an interactive HTML form to designated users, such as Cost Centre managers, containing only the specific reports assigned to their profile. Users can interact with this form to define report parameters and filters; upon clicking the request button, Wevo instantly generates the report using the most current, pre-calculated financial cube data and delivers the completed document directly to the user's inbox.

6.1 Report Form Emailed to Users

The following screenshot shows a report request form as it appears in the user's email inbox. The form is automatically generated and pre-populated with the reports assigned to the user's security profile. Users specify report parameters - including hierarchy scope, period range, and currency - directly within the form and submit the request by clicking the button. Wevo processes the request

using the current pre-calculated cube data and returns the completed report to the inbox within seconds.

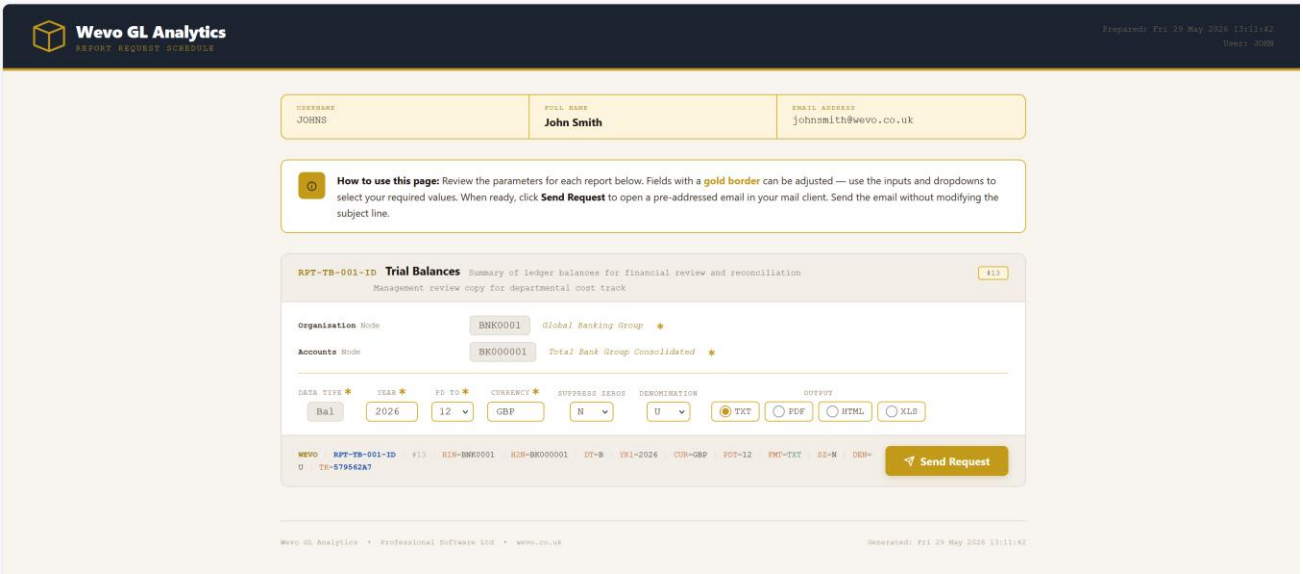


Figure 11: Email-delivered report request form - only reports assigned to the user's security profile are displayed

6.2 Report Delivery

The completed report is generated using live pre-calculated cube data and delivered directly to the requester's email inbox as an attachment.

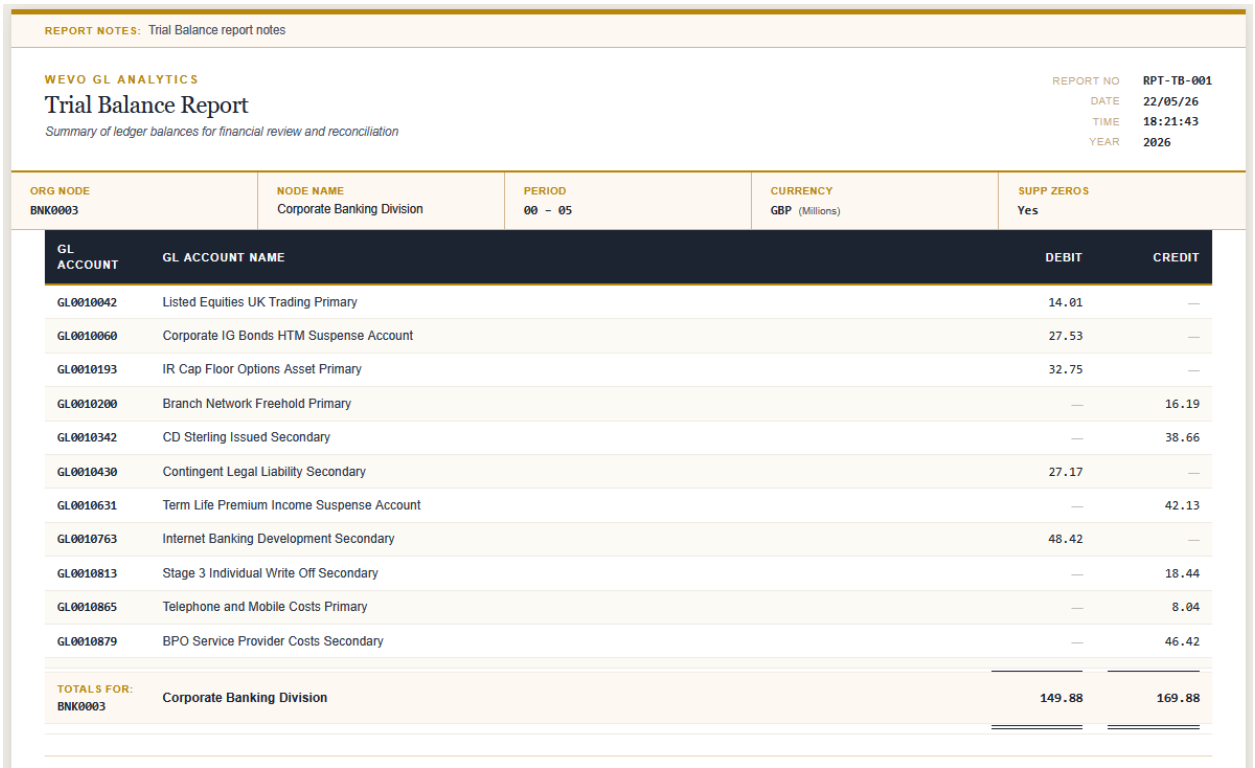


Figure 12: Sample delivered report - hierarchical trial balance showing aggregation from segment value nodes through to summary root

WEVO GL ANALYTICS

Trial Balance Report

Organisation Node: BNK0001 - Global Banking Group
Reporting Currency: GBP (In Thousands)

Report ID: RPT-TB-001
Date: 22/05/2026
Period: 00 - 05
Page: 1

GL ACCOUNT	GL ACCOUNT NAME	DEBIT	CREDIT
GL0010003	EUR Cash Vault Holdings	0.00	
GL0010015	Nostro MUFG JPY Tokyo		5,998.12
GL0010019	BOE Voluntary Reserves Secondary	79,929.10	
GL0010026	Overnight Money Market Placements Secondary	48,520.28	
GL0010027	Variation Margin Cash Posted Primary	89,253.86	
GL0010042	Listed Equities UK Trading Primary	14,015.80	
GL0010043	Listed Equities UK Trading Secondary	13,535.14	
GL0010044	Listed Equities Intl Trading Primary	0.00	
GL0010048	Structured Notes Trading Primary		44,794.66
GL0010051	UK Gilts HTM Portfolio Secondary	0.00	
GL0010060	Corporate IG Bonds HTM Suspense Account	49,281.07	
GL0010064	CMBS Senior Tranches HTM Secondary	0.00	
GL0010069	Supranational Bonds AFS Secondary		37,063.13
GL0010075	Financial Institutions AFS Secondary	0.00	
GL0010081	Money Market Fund AFS Primary	0.00	
GL0010082	Money Market Fund AFS Secondary	0.00	

Figure 13: Sample delivered report - cost centre variance report with budget comparison columns

6.3 Security of the Email Channel

An email-driven request channel naturally invites security questions, and Wevo answers them by design. Every request form carries a token that is validated alongside the sender and every parameter of the request - the channel never trusts the sending address alone. The report catalogue in each form is scoped to the recipient's security profile, and each parameter is re-validated against that profile when the request is received. Completed reports are delivered to the registered user, so a forged request gains an attacker nothing. Every request received, validation outcome, and report dispatched is time-stamped in the audit log.

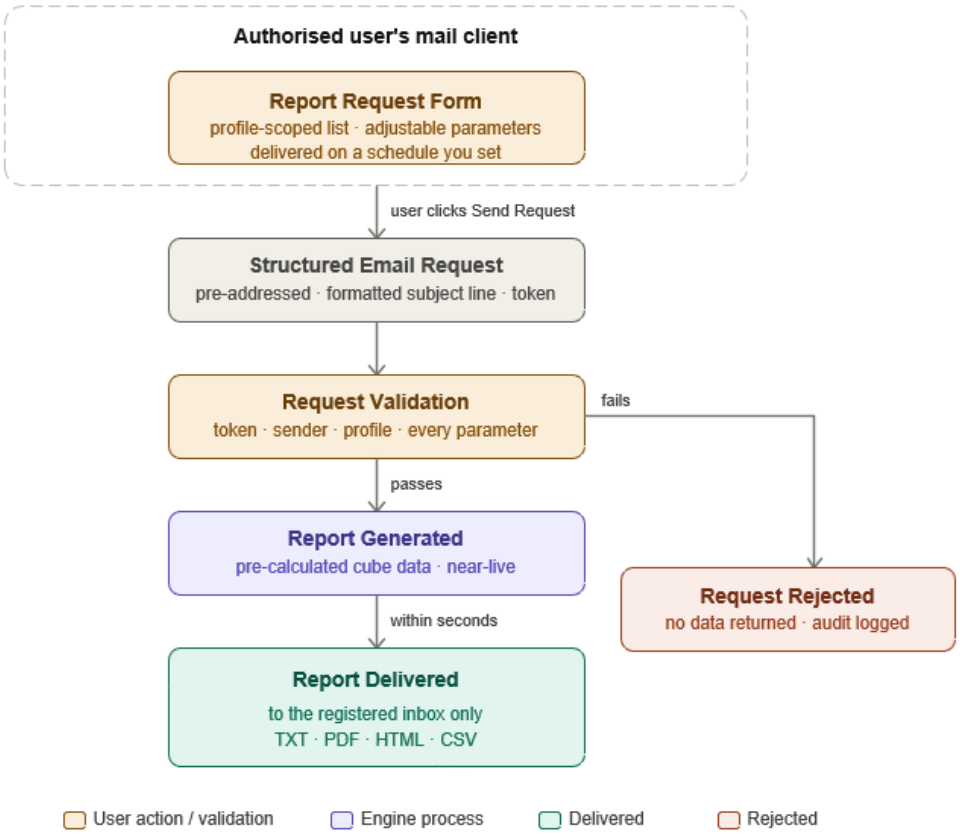


Figure 14: The email report request loop - token, sender, and profile validation with delivery to the registered inbox

7. Data Currency and Near-Real-Time Reporting

7.1 Understanding Near-Real-Time

The phrase "real-time" in financial reporting requires careful qualification. Wevo's web interface presents data from its pre-aggregated analytical cube, not by querying the source GL directly. This architecture is fundamental to Wevo's performance: because data is pre-aggregated rather than computed on demand, navigation through hierarchies of thousands of nodes is instantaneous and requires no waiting.

In practice, the currency of the data in the Wevo cube reflects two factors: when the GL system last produced an extract file, and how long the Wevo batch engine took to process and summarise that extract. For most organisations processing typical business-day volumes, this means the cube reflects the current GL state within minutes of a new extract arriving.

The summarisation process uses an incremental build: only the hierarchy nodes affected by new postings are recalculated in each run. The records being updated are the summary node rows stored in the Wevo database, each identified by its organisation and account hierarchy key combination. Because parent-child relationships within a hierarchy are fixed, the same set of ancestor nodes always aggregates the same children. A posting to any leaf node therefore propagates upwards through the entire ancestry chain - from the originating segment value node, through every intermediate summary node, all the way to the root.

This characteristic means that certain high-level summary nodes are recalculated on every single build run, regardless of how few leaf-node changes triggered it. The root node is the clearest example: it sits above the entire hierarchy and accumulates every posting in the system, so it is always included in a recalculation. If summarisation is triggered twice in quick succession - say, two

runs each processing £10 of new postings - the root node record is written twice: once showing the first £10, and again showing £20. The result after both runs is the same as if a single run had processed the combined £20. The second approach is the more efficient one: one record write at root level instead of two, with the per-run overhead of transaction management, audit logging, and index maintenance paid only once. The same logic applies to every ancestor node that is shared between the leaf nodes touched by each run - the higher the node in the hierarchy, the more likely it is to be recalculated unnecessarily when runs are too frequent.

Designing the extract and summarisation trigger strategy is therefore an efficiency and cost calculation as much as a data currency one. The goal is to batch enough posting activity per run that the server cost of the summarisation is justified by the volume of new data being processed, without allowing the data to become stale enough to affect reporting decisions. Section 7.2 discusses the trigger mechanisms available to find that balance.

7.2 Designing the Extract Strategy

The frequency and trigger conditions for GL extracts are entirely within the client's control, since Wevo's batch engine simply monitors a landing folder for new CSV files. Rather than committing to a fixed schedule, organisations benefit from defining extract triggers that reflect their actual posting activity patterns. Three trigger types can be used individually or in combination:

Trigger Type	Description	Example
Time-based	Extract at regular intervals regardless of posting activity. Simple and predictable; ensures data is never older than the chosen interval. Recommended as a backstop in all configurations.	Every 30 minutes during business hours
Amount-based	Extract when the cumulative value of postings since the last extract reaches a defined threshold. Responsive to high-value activity without requiring a schedule change. Particularly effective during month-end and year-end periods when large postings occur.	When £1,000,000 has been posted since the last extract
Count-based	Extract when the number of individual transactions since the last extract reaches a threshold. Effective for organisations with high transaction volumes and smaller average posting values.	When 50 transactions have been posted since the last extract
Combination	All three triggers are active simultaneously; whichever fires first initiates the extract. Provides the most responsive and resource-efficient pattern for organisations with variable workload profiles.	30 minutes, OR £1,000,000 posted, OR 50 transactions - whichever occurs first
Immediate	Extract on demand, initiated manually or by an automated trigger when results are needed at once.	A critical update has been made to the General Ledger and the summarised results are required immediately.

Using a combination trigger gives a responsive and efficient extract cadence. High-volume business periods are handled by the activity thresholds; quiet periods are covered by the time-based

backstop. The practical effect is that data currency tracks posting activity automatically, without any manual intervention or schedule management.

The summarisation time is not materially affected by extract frequency. Because the incremental build only recalculates changed node values, a small extract processed every 15 minutes is typically faster than a large extract processed every hour. Frequent, small extracts are therefore often preferable to infrequent, large ones - both for data currency and for batch engine performance.

For clients with strict data currency service level requirements - for example, a commitment that the reporting cube will always reflect GL activity within the last 15 minutes - a combination trigger with a short time-based interval and appropriate activity thresholds provides the most reliable approach.

The combination pattern is summarised below: all three triggers run simultaneously, and whichever fires first produces the extract.

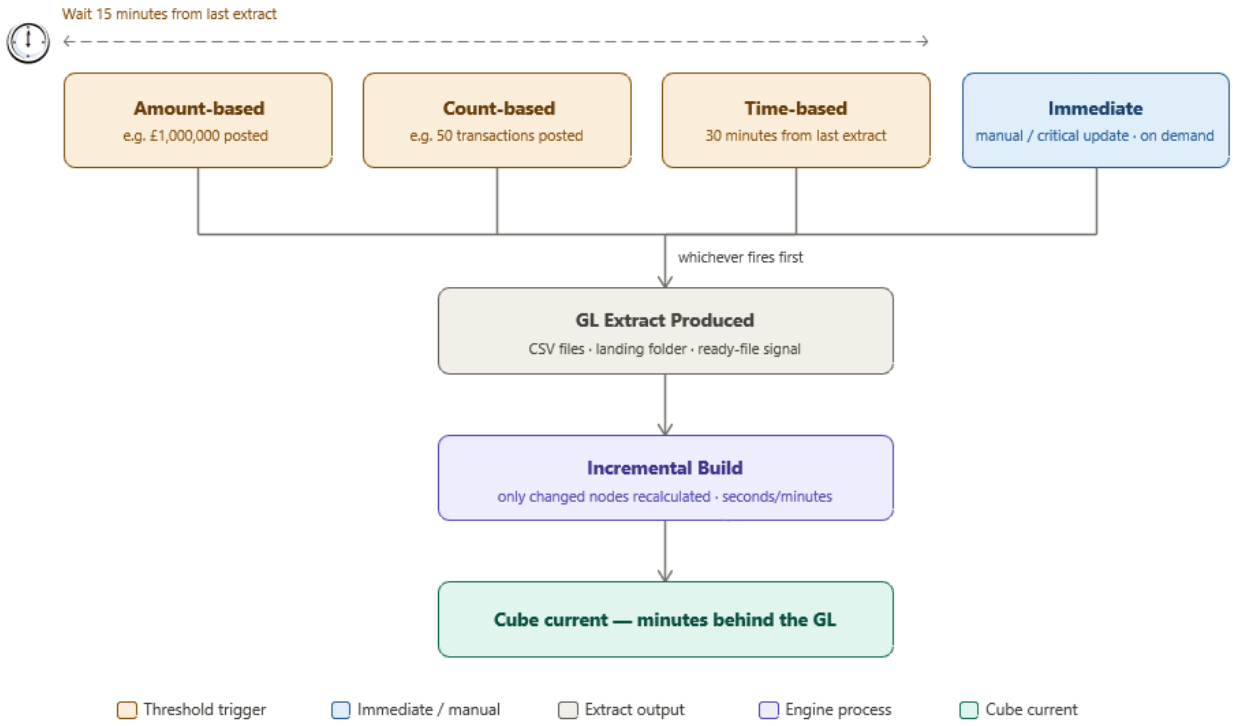


Figure 15: Extract trigger strategy - time, amount, and count triggers combined, with the incremental build keeping the cube minutes behind the GL

7.3 Usable During Updates - Cube-by-Cube Release

The incremental build does more than keep summarisation fast; it also keeps the system usable throughout a data load. Rather than holding all new data back until the entire run completes, Wevo applies roll-up updates one cube at a time, processing cubes in descending year order so the current year - the most-used data - refreshes first.
As each cube is updated to reflect the GL, the A/B table switch for that cube is made immediately: its new data becomes available to users while the next cube is still being processed. This continues until every active cube has been refreshed. The practical effect is that there is no maintenance window and no lock-out, even across millions of rows - users keep working, and the most relevant data appears first, which optimises the perceived processing time.

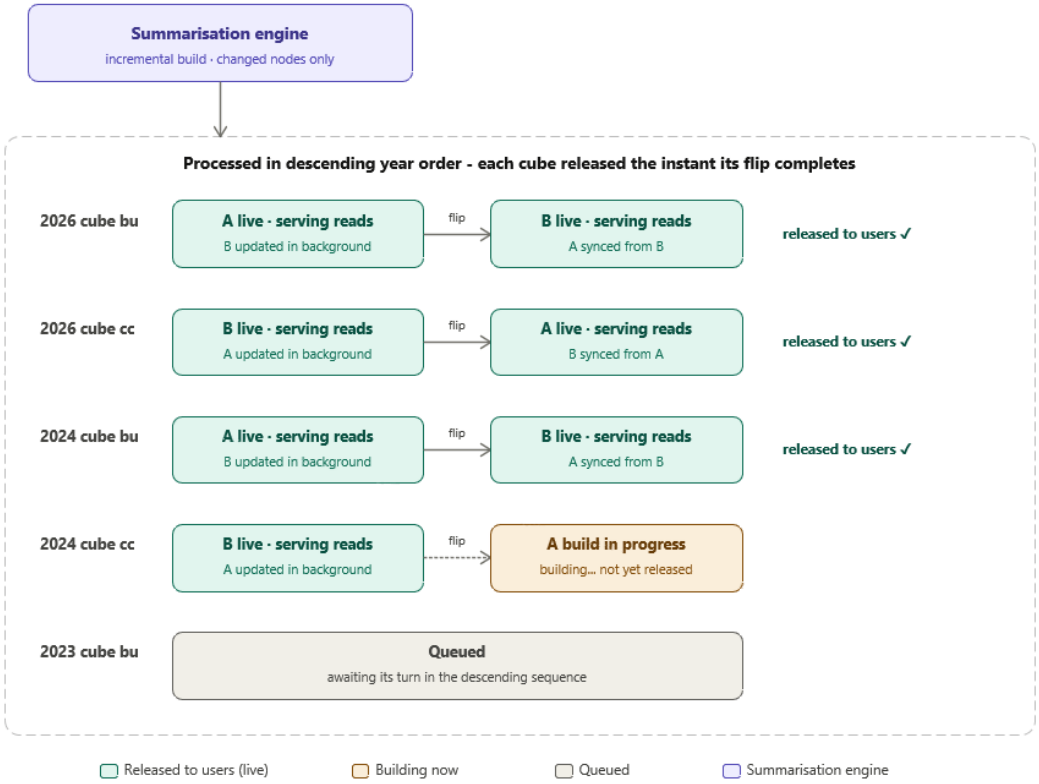


Figure 18: Cube-by-cube release - each cube is switched live the moment it completes, current year first, so users keep working throughout the load

8. System Architecture

Wevo GL Analytics is built on modern, enterprise-grade technology:

- **Web Application:** FastAPI (Python) with Jinja2 templating and Bootstrap front-end
- **Database:** SQL Server with stored procedure architecture for data processing
- **Batch Engine:** Python-based CSV processing with polling engine and validators
- **Data Integration:** File-based CSV feeds with ready-file signalling for zero manual intervention
- **Security by Design:** No dynamic SQL anywhere in the code base, removing an entire class of injection risk and simplifying security review in regulated and high-assurance environments

The platform is designed for performance and scalability, supporting hierarchies with thousands of nodes and transaction datasets with millions of GL entries. The A/B table-switching architecture ensures zero downtime during data loads, and the Summarisation engine processes data efficiently through SQL Server stored procedures.

The zero-downtime guarantee comes from the A/B table-switching pattern: while one table set serves every user query, the batch engine builds the new data in a parallel shadow set, and an instant pointer switch brings it live.

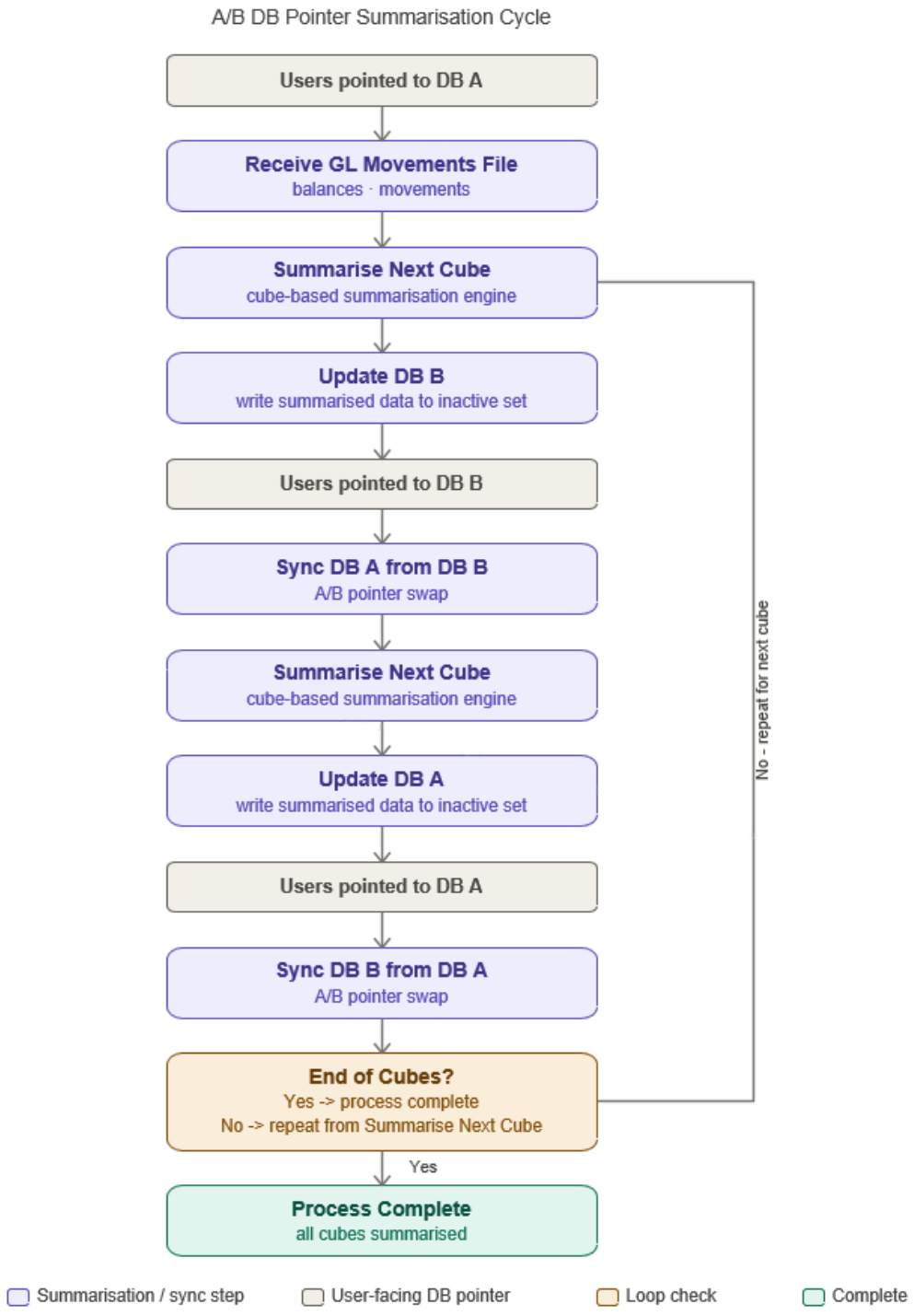


Figure 16: A/B table switching - users are served throughout the build; the pointer switch is instantaneous

9. Compliance and Regulatory Considerations

Wevo GL Analytics is deployed entirely within the Client's own managed infrastructure. This on-premise hosting model means that all financial data remains under the Client's direct control at all times, with no data transmitted to or processed by any third-party environment. As a result, the deployment inherits the Client's existing enterprise security controls and is designed to operate within the key regulatory frameworks applicable to financial systems.

Data sovereignty and residency requirements - including GDPR, CCPA, and equivalent regional privacy frameworks - are supported by design, as data never crosses organisational or geographic boundaries outside the Client's estate. Operational resilience obligations under DORA and NIS2 are addressed through the Client's own Business Continuity and Disaster Recovery infrastructure, with Wevo providing software built to security-by-design principles, including vulnerability management and MFA-ready integration points. Financial data integrity requirements under SOX and Basel III are supported through encrypted, time-stamped audit logging within the application, with database-level access permissions and separation of duties enforced by the Client's DBA team on their own servers. Technical security standards under PCI-DSS 4.x and the NIST Cybersecurity Framework are supported through the application's use of AES-256 encryption, with SSL/TLS certificate management and Identity Provider integration - including Azure AD and Okta - handled by the Client within their existing security infrastructure.

Under this Shared Responsibility Model, the Client assumes responsibility for infrastructure-level controls, and Wevo provides the application layer with the technical capabilities required to operate within those controls. No additional third-party data processing agreements, cross-border transfer mechanisms, or external audit obligations are introduced by the deployment of Wevo GL Analytics.

10. Deployment Options

Wevo can be deployed on-premise or in a cloud environment, depending on client requirements. The system requires SQL Server (2016 or later), a Python runtime environment, and a web server. All static assets (CSS, JavaScript, fonts) are served locally to support fully air-gapped deployments if needed.

For clients with existing GL systems running on platforms other than SQL Server, Wevo's batch engine interfaces through standard CSV files, making integration straightforward regardless of the source system.

10.1 Source Code and the Continuity Licence

Wevo is supplied with its complete source code at installation. The licence agreement includes a continuity provision: if the product can no longer be supported by the vendor, the client retains the source code and is free to continue the life of the product using in-house teams or modern AI-assisted development tooling.
This is a stronger guarantee than a conventional escrow arrangement - there is no release procedure, no trigger dispute, and nothing held by a third party to go stale. Because the codebase is Python and standard SQL on mainstream infrastructure, it is precisely the kind of system an in-house team can own. For regulated organisations, the provision also serves as a ready-made exit strategy for ICT third-party risk frameworks such as DORA: the continuity of the reporting service does not depend on the continuity of the vendor.

10.2 Development and Production Instances

The Wevo licence includes both a development and a production instance - one licence, two environments, with no surcharge for the non-production environment. The production instance serves live reporting and the email reporting service; the development instance supports training, testing, and ongoing hierarchy and cube development work without touching live processing. Finance teams can therefore prepare and rehearse complex changes - a major reorganisation, a new cube, a restructured chart of accounts - safely in development, then promote the validated result to production on the next full build.

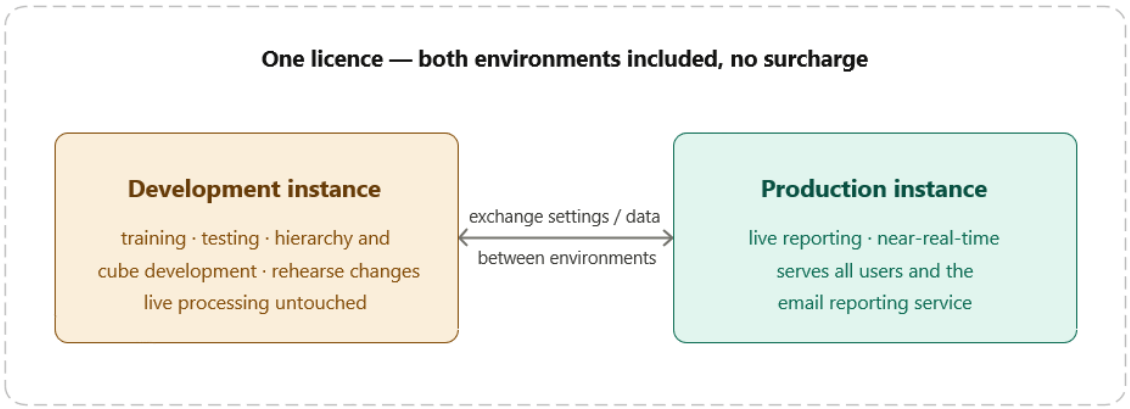


Figure 19: Development and production instances - both included in the licence; validated changes are promoted from development to production on the next full build

10.3 Year-End Read-Only Archive (optional)

An optional licence converts a year-end data backup, free of charge, into a read-only MySQL database. This archive carries the frozen hierarchies of that year, so the prior year can be read exactly as it stood at close, and supports drill-down enquiry through the standard interface. Because it has no update or build engine, it is highly portable: it can be packaged to run on a server, or even on laptops, as a self-contained way to interrogate a closed year.
For audit and review functions this provides a fixed, distributable view of a closed year with its hierarchies of the time, taking load off the live system. It is a low-cost way to retain and circulate historical positions without keeping the full build engine running for years that no longer change.

11. What Makes Wevo Different

Several aspects of Wevo GL Analytics distinguish it from conventional financial reporting tools, off-the-shelf BI platforms, and custom development approaches.

Responsibility Reporting Architecture. Wevo is built around the principle that financial data should be presented through the lens of the people and structures responsible for it. Rather than producing flat account-code reports, Wevo allows organisations to define hierarchies that precisely reflect their management structure, cost ownership model, or product groupings. This is not a configurable add-on; it is the architectural foundation of the system.

Non-Intrusive Integration by Design. Wevo has no direct connection to the source GL database. It receives periodic CSV extracts and processes them independently. This means Wevo can be deployed alongside any GL system - regardless of vendor or platform - without risk to the GL environment, without vendor agreements for direct database access, and without any impact on GL performance. The integration mechanism is the simplest possible interface, yet it supports all of Wevo's analytical capabilities.

Hierarchies as First-Class Citizens. In most BI platforms, hierarchical analysis is a feature added over a flat data model. In Wevo, the hierarchy is the data model. Cubes are defined by their hierarchies, security is enforced by hierarchy scope, and all reporting is structured around hierarchy navigation. When the organisation changes - a new department created, a cost centre reorganised, a product line added - the Wevo hierarchy is updated to reflect that change immediately, with no impact on existing data and no need to redefine any reports.

Zero-Downtime Data Loads. Wevo uses an A/B table-switching architecture for data updates. While one set of tables is serving live user queries, the batch engine builds the new data in a parallel shadow set. When the build is complete, the system switches transparently to the new data.

Users experience no interruption, even when processing millions of GL transactions, and there is no maintenance window required for data refreshes.

Channel-Agnostic Report Delivery. Beyond the interactive web interface, Wevo's reporting engine delivers formatted reports by email using structured request messages. Users who do not need - or should not have - direct access to the analytical interface can receive precisely the reports assigned to their profile, formatted and delivered automatically. This makes Wevo practical for cost centre managers, external stakeholders, and board-level recipients alike.

Adaptable Without Redesign. In conventional reporting systems, an organisational restructure typically requires changes to report definitions, account mapping tables, and often to the underlying data model. In Wevo, the hierarchy structure is maintained separately from the financial data. A reorganisation is reflected by updating the hierarchy; the existing data remains valid and fully navigable through the new structure on the next data build. No report definitions need to change.

Data Sovereignty by Design. Wevo is deployed entirely within the client's own infrastructure. There is no cloud dependency, no SaaS subscription, and no third-party data processing. All financial data remains inside the client's environment at all times, satisfying data residency, regulatory, and internal governance requirements without additional controls or agreements.

Comparatives That Keep Up. After a reorganisation, prior-year data is instantly comparable through the new structure - no restatement, no mapping tables - while retained year versions keep history available exactly as originally reported. Most tools force a choice between the two views; Wevo provides both from the same data.

Source Code From Day One. Wevo installs with its full source code under a continuity licence. Every other vendor asks you to trust their longevity; Wevo removes the question entirely.

Passive Longitudinal Data Collection. The node pair watchlist accumulates a granular, time-stamped history for the organisation and account intersections that matter most - written automatically on every engine run, across all active years and currencies, with no ongoing administration. The longer it runs, the more valuable the dataset becomes for trend analysis, anomaly detection, and AI-driven insight, all without any additional query to the source GL.

Hardened by Design. The Wevo code base contains no dynamic SQL anywhere. This removes an entire class of injection risk by construction and eases security review in regulated and high-assurance environments - a structural property of the product, not a configuration option that can be switched off.

12. Benefits Summary

Wevo GL Analytics delivers tangible benefits to finance teams:

- Near-real-time access to summarised GL data, updated automatically from regular GL source system extracts and current to within minutes of GL activity
- Multi-dimensional analysis through flexible hierarchy matching
- Drill-down capability from summary to segment-value-level detail
- Budget variance analysis built into every view
- Self-service reporting that reduces dependency on IT
- Adaptable hierarchies that reflect organisational changes without system redesign
- Non-intrusive integration that does not modify or impact the GL system
- Security profiles that enforce data access boundaries, ensuring users see only the financial data relevant to their role
- Enterprise-scale capacity supporting organisational hierarchies of up to 25,000 nodes and chart of accounts hierarchies of up to 5,000 nodes

- Zero-downtime hierarchy maintenance - hierarchies can be updated during business hours while the current production version remains fully operational, with changes promoted automatically on the next full balance build
- Budgets beside actuals through the same hierarchies, with variance at every node and 16 periods including 4 close periods
- Like-for-like comparatives after any restructure, with prior-year hierarchy versions retained for as-was reporting
- Email reporting service putting formatted reports in any authorised inbox - unlimited recipients, no licence seats
- Full source code supplied at installation under the Wevo Continuity Licence - a built-in vendor-independence guarantee
- Node pair watchlist building a granular, time-stamped history of key intersections automatically on every engine run - a passive, zero-effort longitudinal data store
- Development and production instances both included in the licence, so complex changes can be prepared and rehearsed safely before reaching the live system
- Optional year-end read-only MySQL archive - a portable, build-free record of a closed year with its hierarchies of the time, for drill-down off the live system
- No dynamic SQL anywhere in the code base, removing an entire class of injection risk and easing security review

13. Next Steps

This document has introduced the core concepts, capabilities, and architecture of Wevo GL Analytics. Organisations wishing to proceed with an evaluation or deployment should refer to the following supporting documentation:

- **Data Feed Specification** - defines the CSV file formats, column layouts, validation rules, and ready-file signalling protocol required to integrate the GL source system with the Wevo batch engine.
- **Deployment and Installation Guide** - covers system requirements, step-by-step installation, batch engine configuration, and the first-run checklist.
- **User Guide** - covers cube selection and navigation, hierarchy drill-down, budget variance analysis, and the fixed reporting email service.
- **Administrator Guide** - covers the CubeMaint application, including hierarchy creation and maintenance, cube management, user administration, and security profile configuration.
- **Technical Architecture and Deployment Guide** - provides detailed information on the system architecture, database schema, security model, and infrastructure requirements for on-premise and cloud deployments.

To arrange a demonstration or to discuss deployment requirements, please contact Wevo GL Analytics at info@wevo.co.uk.