Wevo GL Analytics

# Technical Architecture and Integration Overview

The major components of Wevo GL Analytics, how they fit together, and what your IT and GL teams need to provide

Professional Software Ltd

Copyright 2026 Professional Software Ltd. Distribute freely.

# 1. About this document

This overview is written for the IT architects, infrastructure teams and GL system owners who assess Wevo GL Analytics during an evaluation, and who will later install and run it. It answers three questions: what are the major components, how do they fit together, and what does your organisation need to provide.

In one sentence: Wevo is a read-only management reporting layer that receives three GL extract files from your General Ledger, summarises them through your organisational and account structures, and serves the results to users in a browser and by internal email, running entirely inside your own environment.

Section 2 shows all nine components on one page. Section 3 takes each in turn: what it does, what it connects to, what it runs on, and your part in it. Sections 4 and 5 cover the GL feed and how reporting is kept current, and section 6 brings together everything your organisation needs to do. The remaining sections cover server layouts, security, operation and scale.

Security is covered in depth in the Security and Controls Overview. The full installation prerequisites for each module, down to package level, are set out in the Components and Installation Prerequisites document, which is supplied at implementation.

# 2. Wevo at a glance

Wevo is made up of six modules installed in your environment, which connect to three systems you already have. The table below lists all nine, and Figure 1 shows how they connect. The same numbering is used throughout this document.

Copyright 2026 Professional Software Ltd. Distribute freely.

## 2.1 The components in summary

| No. | Component | What it does | Provided by |
|---|---|---|---|
| 1 | Cube Development transactions | Maintenance application: hierarchies, cubes, users, profiles, reports and system settings | Wevo |
| 2 | Drilldown Enquiry transactions | Analytics application: drill from any consolidated figure to the lowest level of detail | Wevo |
| 3 | Email Reporting Engine | Sends report request forms and returns finished reports by email | Wevo |
| 4 | Application Database | Holds the GL data, hierarchies, cubes and pre-calculated roll-ups | Wevo schema on your SQL Server database platform |
| 5 | Summarisation Engine | Validates, loads and summarises the GL files | Wevo |
| 6 | Logging Databases | Record every file, build, request and dispatch | Wevo schema on MySQL |
| 7 | Mailbox | The address report requests go to and reports come from | You |
| 8 | General Ledger | Source of every figure; exports three CSV files | You |
| 9 | Landing Folder | Receives the GL files and presents them to the Summarisation Engine | You |

Copyright 2026 Professional Software Ltd. Distribute freely.

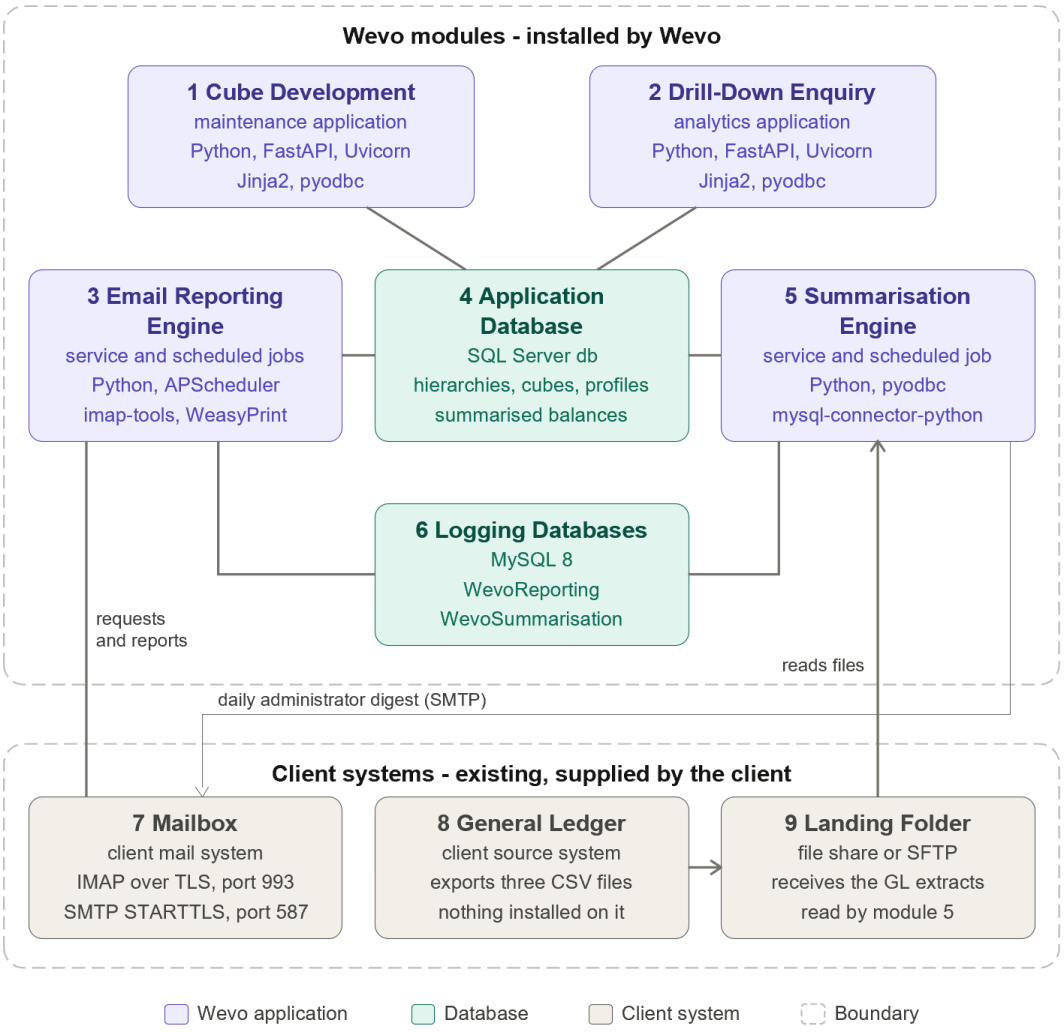



*Figure 1 - Wevo GL Analytics - modules, components and connections*

## 2.2 How the pieces fit together

Four paths through the diagram explain the whole system:

- **Data in.** An extract process on your General Ledger (8) writes three CSV files to the landing folder (9). The Summarisation Engine (5) validates each file, loads it into the application database (4), and summarises it through every active cube.
- **Structure.** Finance maintains the hierarchies and cubes in Cube Development (1). Changes are prepared in development and reach production at the next full build.
- **Reporting out.** Users drill into the figures in Drilldown Enquiry (2). The Email Reporting Engine (3) sends report request forms through the mailbox (7) and returns finished reports the same way.
- **Record.** Each engine writes every event to its own logging database (6).

## 2.3 Two principles behind the design

**Read-only by design.** Wevo holds no credentials for your General Ledger, has no connection to it, and has no code path that writes to it. The ledger writes files; Wevo reads them. The GL remains

Copyright 2026 Professional Software Ltd. Distribute freely.

the system of record, and introducing Wevo changes nothing about how the ledger is run, secured or audited.

**Structure held separately from data.** GL data is stored at leaf level - each segment value against each account - and the hierarchies are held as separate, versioned objects. Roll-ups are calculated by applying a structure to the leaf-level data, so the same data can be read through today's structure or through the structure in force at the time. A reorganisation is a rebuild, not a data migration.

# 3. The components

Each component is described in the same way: what it does, what it connects to, what it runs on, and what your organisation provides. The modules are independent: any of them can share a server with any other, or each can have its own (section 7).

## 3.1 Cube Development (1)

| Aspect | Detail |
|---|---|
| What it does | The maintenance application. Finance and Wevo administrators maintain hierarchies, cubes, segment values, financial years, the watchlist, system parameters, report definitions, users, security profiles and report assignments. Hierarchy changes are promoted to production and structures are preserved from here. |
| Connects to | Application Database (4) |
| Runs on | Python 3.10 or later, FastAPI and Uvicorn, Jinja2, and the Microsoft ODBC Driver for SQL Server. All scripts, styles and fonts are served locally, with no external calls. |
| Your part | A host, which may be shared, and HTTPS access for administrators only. This application holds administrative capability and is not limited by the reporting security profiles, so restricting who can reach it is part of your access management. |

Copyright 2026 Professional Software Ltd. Distribute freely.

## 3.2 Drilldown Enquiry (2)

| Aspect | Detail |
|---|---|
| What it does | The analytics application. Reporting users choose a cube, set their selection criteria and drill from any consolidated figure to the lowest level of detail, typically in under a second. Each user sees only the part of the business their security profile allows. |
| Connects to | Application Database (4) |
| Runs on | The same technology as Cube Development, but a separate application with its own address. The two can share a host, or be separated so that maintenance is reachable only from an administrative network. |
| Your part | HTTPS access from users' browsers. Nothing is installed on user machines. |

## 3.3 Email Reporting Engine (3)

| Aspect | Detail |
|---|---|
| What it does | Sends a report request form to authorised users on a schedule, for example each Monday morning. It reads the requests that come back, generates each report from the pre-calculated cubes, and returns it as a ZIP attachment to the address registered against the user, optionally AES-encrypted with a per-user password. Output formats are TXT, PDF, HTML and CSV. |
| Connects to | Application Database (4), with permission to run Wevo stored procedures only; its logging database WevoReporting (6); the Mailbox (7) |
| Runs on | Python 3.10 or later, running as an operating system service, with two scheduled housekeeping jobs. It needs the PDF rendering libraries of the host operating system. |
| Your part | The mailbox described under component 7. No inbound network ports are needed: the engine starts every connection itself. |

Copyright 2026 Professional Software Ltd. Distribute freely.

## 3.4 Application Database (4)

| Aspect | Detail |
|---|---|
| What it does | Holds the leaf-level GL data, the hierarchies, the cubes, the pre-calculated roll-ups, the security profiles, the report definitions and the application audit log. It keeps two copies of each cube's summarised tables, so that users read one while the other is rebuilt. |
| Connects to | Nothing. Components 1, 2, 3 and 5 connect to it. |
| Runs on | Microsoft SQL Server. MySQL is also supported. The SQL is written to the standard by design, for portability. |
| Your part | The database platform, backups, disaster recovery and encryption at rest. Accept connections from components 1, 2, 3 and 5 only, each with its own login holding only the permissions it needs. |

## 3.5 Summarisation Engine (5)

| Aspect | Detail |
|---|---|
| What it does | Watches the landing folder, then validates, loads and summarises each GL file and archives it. It also carries out instructions such as the movements cleardowns and a controlled stop, writes a heartbeat so that your monitoring can confirm it is running, and sends a daily administrator digest. |
| Connects to | Landing Folder (9); Application Database (4); its logging database WevoSummarisation (6); optionally, the mail system (7) for the daily digest |
| Runs on | Python 3.10 or later, running as an operating system service, with one scheduled job for the daily movements cleardown. |
| Your part | Read and write access to the landing folder. No inbound ports and no internet access are needed. For a fully isolated installation, the daily digest can be switched off. |

Copyright 2026 Professional Software Ltd. Distribute freely.

## 3.6 Logging Databases (6)

| Aspect | Detail |
|---|---|
| What it does | Two small databases, one for each engine. WevoReporting records every report request and dispatch; WevoSummarisation records every file received and every engine event. |
| Connects to | Nothing. The Email Reporting Engine (3) and the Summarisation Engine (5) each write to their own database. |
| Runs on | MySQL 8, normally on the same host as the engine that writes to it, so that it has no network exposure at all. |
| Your part | Include both databases in your backup regime. |

## 3.7 Mailbox (7)

| Aspect | Detail |
|---|---|
| What it does | The address users send report requests to, and the address finished reports come from. |
| Connects to | Nothing. The Email Reporting Engine (3) connects to it. |
| Runs on | Your own mail system. |
| Your part | One mailbox with IMAP over TLS (port 993) and SMTP submission with STARTTLS (port 587); the Processed and Quarantine folders are created automatically. SPF, DKIM and DMARC are recommended for the sending domain. The same mailbox account also sends the Summarisation Engine's daily digest. |

## 3.8 General Ledger (8)

| Aspect | Detail |
|---|---|
| What it does | The source of every figure Wevo reports. It is numbered here so that it appears in the component list, but it is not a Wevo component and nothing is installed on it. |
| Connects to | Landing Folder (9), which it writes the extract files to |
| Runs on | Any GL that can export structured data, including SAP, Oracle, Microsoft Dynamics and Sage. |
| Your part | A scheduled export producing three CSV files: segment values, balances and movements (section 4). |

Copyright 2026 Professional Software Ltd. Distribute freely.

### 3.9 Landing Folder (9)

| Aspect | Detail |
|---|---|
| What it does | Receives the GL extract files and presents them to the Summarisation Engine. It is the entire integration surface between your ledger and Wevo. |
| Connects to | Written to by the General Ledger export (8); read by the Summarisation Engine (5) |
| Runs on | A file share, an SFTP target, or a folder written directly by the GL export, on your infrastructure. |
| Your part | Provide the folder and govern it with your own access controls: the GL export writes to it, and the Summarisation Engine reads files and moves them to its archive. |

### 3.10 Users

Users need only a web browser, and report recipients need an email address registered in Wevo. Where a report is protected with a file password, the recipient needs an archive tool that supports AES-encrypted ZIP files, such as 7-Zip, NanaZip, PeaZip or WinRAR. The archive handlers built into Windows and macOS cannot open AES-encrypted files.

## 4. The GL feed

The GL feed is the one piece of integration work on your side. It is deliberately simple: three CSV files, written by your own scheduler to the landing folder. There is no connector, adaptor or API on your ledger, and no Wevo software near it.

Copyright 2026 Professional Software Ltd. Distribute freely.

## 4.1 The three GL extract files

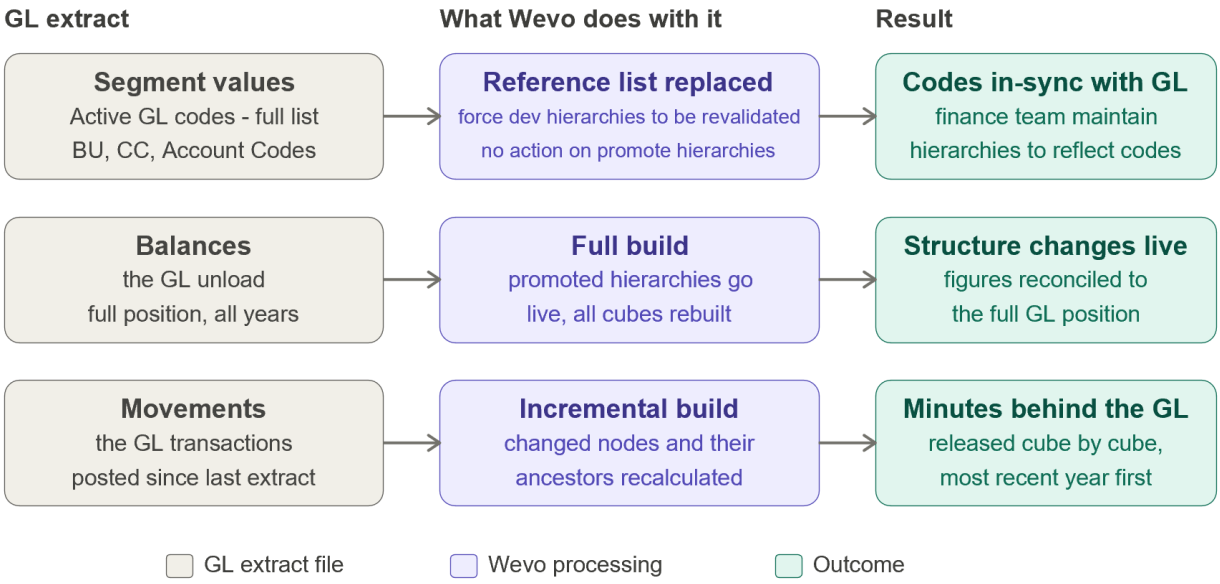



*Figure 2 - The three GL extracts and what each one drives*

| File | Name prefix | Contents | Typically sent |
|---|---|---|---|
| Segment values | SEGMENT | The codes and descriptions of the segment values in use in the GL, for each segment: business unit, cost centre and account. A full list each time; closed or inactive codes are not sent | When codes are added or changed, or on a regular schedule |
| Balances | BALANCE | Full GL unload: an opening balance and 16 period values for each segment combination, record type, year, version and currency. | On a cycle you choose, such as weekly or monthly |
| Movements | MOVEMENT | The GL transactions posted since the last extract: segment combination, record type, year, version, currency, period and amount | Frequently through the day |

Each file drives something different:

- **Segment values drive the leaf nodes.** Every leaf node of a hierarchy is a real GL code, named identically to it. Each segment values file replaces Wevo's reference list in full. It does not affect production reporting, but any validated hierarchy in development is marked

Copyright 2026 Professional Software Ltd. Distribute freely.

for revalidation, so that new or removed codes are validated in hierarchies before the next promotion.

- **Balances drive the full build.** Receiving a balances file triggers a full build: hierarchy changes promoted by finance are copied to production, and every active cube is rebuilt from the full GL position.
- **Movements drive the incremental build.** Only the nodes affected by the new postings, and their ancestors, are recalculated. This is what keeps reporting minutes behind the GL.

## 4.2 Company structures are not extracted

The summary nodes in the summary levels (above the leaf nodes) - divisions, departments, account groupings - exist only in Wevo, and finance builds and maintains them in Cube Development. Your GL team supplies the segment value codes; finance decides how they roll up.

If a structure already exists elsewhere, it can be loaded into Wevo once from a file at set-up. That is a one-off implementation task, not part of the GL feed.

## 4.3 What the files contain

Some points about the content that GL teams usually ask about:

- **Segments.** Each row carries the organisational segments in use (business unit, cost centre, or both) and the account segment. Which segments Wevo uses is set in the system parameters.
- **Amounts** are written as plain decimals in pounds and pence, or the equivalent for other currencies. Wevo converts them to whole minor units internally and exactly, so no floating-point rounding is introduced.
- **Periods.** A year holds 16 periods: 12 trading periods and 4 close periods, or 13 trading periods and 3 close periods. With the opening balance, that gives 17 values per year.
- **Record types** distinguish actuals from budgets and other versions, and are set in the Wevo system parameters.
- **Currency** is carried on every row as a three- or six-character code, exactly as held in the GL record.
- **Header record.** Balances and movements files begin with a header record carrying a description of the extract, which is recorded against the load.
- **Movements must net to zero.** A movements file that does not balance is rejected in full rather than partly loaded.

The exact column layouts are set out in the file layout specifications, which are supplied at implementation.

## 4.4 How a file is delivered and processed

A file is delivered in two parts. The export first writes the CSV data file, then writes a small companion file with the same name and the suffix .ready. The engine acts only on the .ready file, so it never picks up a data file that is still being written. There is nothing to call and no acknowledgement to wait for.

Copyright 2026 Professional Software Ltd. Distribute freely.

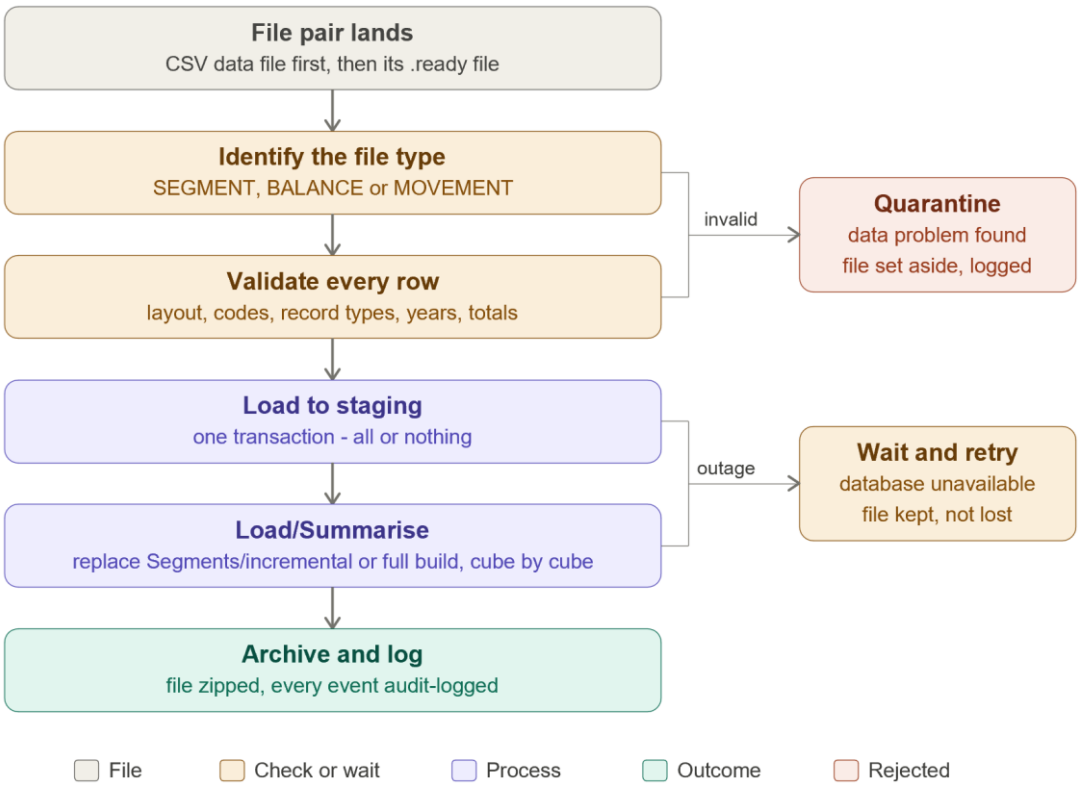



*Figure 3 - How each GL file is processed*

Each file passes through the same steps:

1. **Identify.** The file type is taken from the first seven characters of the file name. A file that cannot be identified is set aside and logged.
2. **Validate.** Every row is checked before anything is loaded: column layout, segment values, record types, years, currencies, amount formats and, for movements, the zero total. Validation stops after a configurable number of errors, so a badly formed file is reported quickly.
3. **Load.** Valid rows are loaded to staging in a single transaction. A load either completes or leaves nothing behind.
4. **Summarise.** The data is rolled up through every active cube (section 5).
5. **Archive.** The file is compressed and moved to an archive folder, which is cleared automatically after a configurable retention period.

Two kinds of failure are treated differently. A file with a data problem is quarantined: it is moved aside with the reason logged, nothing from it is loaded, and it is corrected and resent. A temporary infrastructure problem, such as the database being unavailable for maintenance, is not a data problem: the engine leaves the file in place, waits, retries automatically, and processes it normally once the database returns. A valid file is never quarantined because of an outage.

When a file is quarantined, its .ready file is renamed to .readyInERROR, which takes it out of further processing. The file stays in quarantine until it is deleted manually. Once the CSV file has been corrected, renaming .readyInERROR back to .ready is all that is needed for the engine to pick it up again automatically.

Copyright 2026 Professional Software Ltd. Distribute freely.

### 4.5 When to extract

How often movements are sent decides how current Wevo is. The movements extract can be triggered on whatever basis suits your ledger:

- time-based, at fixed intervals through the working day;
- amount-based, when the value posted since the last extract passes a threshold;
- count-based, when the number of postings since the last extract passes a threshold;
- manually, on demand;
- or a combination of these, whichever fires first.

Wevo does not impose a schedule for any of the three files. A typical pattern is a balances file each weekend for the full build, movements every 30 minutes through the working day, and a segment values file early each morning.

### 4.6 More than one General Ledger

A group running several GL systems, such as a parent ledger and separate subsidiary ledgers, presents them to Wevo as a single virtual GL. The one rule is that every segment value must be unique within its segment position across all the ledgers. A common way to achieve this is to prefix the codes from the non-main ledgers, so that cost centre 4100 in a subsidiary ledger arrives as A4100. How the extracts are combined is your choice. A group with a single GL needs none of this.

## 5. Keeping the reporting current

### 5.1 Incremental and full builds

| Build | Triggered by | What happens |
|---|---|---|
| Incremental | Each movements file | Only cubes with nodes affected by the new postings, and their ancestors, are recalculated. |
| Full | Each balances file | Hierarchy changes promoted by finance are copied to production, all data files are cleared down, then every active cube is rebuilt from the balances. |

Every financial year allocated to Wevo is kept in step with the GL. A prior-year adjustment or audit correction posted today reaches Wevo on the next movements update, at every level of every structure, in the same way as a current-year posting.

Wevo can also summarise prior years already held in your GL from the first day, using your current organisational hierarchy, so like-for-like comparatives are available at go-live.

### 5.2 Hierarchy changes reach production under control

Finance can edit any hierarchy at any time in development. When a change is ready, it is validated and promoted, which locks it. The promoted hierarchy is copied to production at the start of the next full build. A complex reorganisation can therefore be prepared over days or weeks and deployed in one controlled build, with the audit trail recording who changed what, and when.

Copyright 2026 Professional Software Ltd. Distribute freely.

### 5.3 Movements and daily movements

Alongside balances, Wevo holds movements, which accumulate from the last cleardown, and daily movements, which show what has reached the ledger today. The movements cleardown is system-wide and scheduled by you: a cleardown at 6am each Monday, for example, makes the movements figure a weekly view. The daily movements cleardown runs as a scheduled job each morning, with 6am suggested.

## 6. What you need to do

This section brings together everything your organisation provides. None of it is specialist: it is the same work your teams carry out for any internally hosted application.

### 6.1 Before installation

| Item | What is needed | For |
|---|---|---|
| Servers | One to three Linux hosts (Ubuntu 24.04 LTS in the reference deployment), sized during implementation from your data volumes and user numbers. No specialist hardware. | 1, 2, 3, 5, 6 |
| Application database | A Microsoft SQL Server instance, with a separate login for each connecting component, backups and encryption at rest. | 4 |
| Logging databases | MySQL 8, normally on the same host as each engine. | 6 |
| Mailboxes | A reporting mailbox with IMAP and SMTP access, and, if the daily digest is used, an outbound-only account for it. | 7 |
| Landing folder | A file share, SFTP target or folder that the GL export can write to and the Summarisation Engine can read and archive from. | 9 |
| Web access | HTTPS access to the two web applications, normally through your standard reverse proxy and certificate, with Cube Development restricted to administrators. | 1, 2 |
| Network rules | The connections listed in section 6.2, and nothing else. | All |
| Service accounts | An operating system account for each service. Credentials are held in protected configuration files readable only by that service; the Summarisation Engine refuses to start if its credential files are readable by anyone else. | 3, 5 |

Copyright 2026 Professional Software Ltd. Distribute freely.

## 6.2 Network connections

| From | To | Port | Purpose |
|---|---|---|---|
| User browsers | Cube Development (1), Drilldown Enquiry (2) | HTTPS | Administration and reporting |
| Components 1, 2, 3 and 5 | Application Database (4) | TCP 1433 (SQL Server) | Application data |
| Engines 3 and 5 | Logging Databases (6) | TCP 3306 | Engine logs; normally local to each host |
| Email Reporting Engine (3) | Mailbox (7) | TCP 993 and 587 | Report requests in, reports out |
| Summarisation Engine (5) | Mail system (7) | TCP 587 | Daily digest; optional |
| General Ledger (8) | Landing Folder (9) | Your transfer method | Extract files |
| Summarisation Engine (5) | Landing Folder (9) | Your file access method | Read and archive the files |

The two engines need no inbound ports at all, and no component needs internet access.

## 6.3 Building the GL extracts

Your GL team builds:

- a segment values extract of the codes in use for each segment;
- a balances extract (the GL unload) and a movements extract (the GL transactions) in the published layouts;
- a scheduled job that writes each file to the landing folder, followed by its .ready file.

Wevo supplies the file specifications, validates every test file on arrival, and reports precisely what is wrong with any file that fails, which makes the extracts quick to test. If you run more than one General Ledger, the extracts also need to present them as one virtual GL (section 4.6).

## 6.4 Once live

- **Backups** of the application database and both logging databases.
- **Monitoring.** Add the Summarisation Engine heartbeat and the daily digest to your existing monitoring.
- **Quarantined files.** When the digest reports a quarantined file, the GL team corrects and resends it.
- **Patching** of the operating system and database platforms, in line with your usual policy.
- **Joiners and leavers.** Your Wevo administrators keep users and security profiles in step with the organisation.

Copyright 2026 Professional Software Ltd. Distribute freely.

### 6.5 Who does what

| Area | Your organisation | Wevo |
|---|---|---|
| Servers | Provide the hosts to the agreed sizing | Install and configure the modules, in development and production |
| Databases | Provide SQL Server and MySQL 8, with backups | Install the Wevo database objects and logins |
| GL extract | Build and schedule the three extracts | Supply the layouts, validate test files and report errors precisely |
| Email | Provide the mailbox, and the digest account if used | Configure both engines to use them |
| Network and security | Network rules, web access, operating system hardening and service accounts | Configure access profiles with your finance administrators |
| Monitoring | Include the heartbeat and daily digest in your monitoring | Provide the monitoring hooks |

## 7. Server layouts and deployment

### 7.1 Example server layouts

Because the modules are independent, how many servers to use is your choice. Only components 1 to 6 are installed; the mailbox, the General Ledger and the landing folder are existing systems. Three common layouts are:

| Layout | What goes where | Suited to |
|---|---|---|
| Single server | All six modules on one host, with the application database on the same host or separate | Evaluation and smaller installations |
| Two servers | Server A: application database, Summarisation Engine and its logging database. Server B: both web applications, the Email Reporting Engine and its logging database | Separating the database and batch load from everything user-facing |
| Three servers (reference deployment) | Server A: application database, Summarisation Engine and its logging database. Server B: both web applications. Server C: Email Reporting Engine and its logging database | Keeping the mail-facing component apart from the web applications and the database |

Copyright 2026 Professional Software Ltd. Distribute freely.

### 7.2 Deployment models

Wevo runs entirely in your environment. There is no Wevo-hosted service, no software-as-a-service component and no third-party processing. You choose where that environment is:

| Model | Where it runs | Suited to |
|---|---|---|
| Cloud | Your own AWS, Azure or Google Cloud account | Organisations with a cloud-first policy that want the data in their own tenancy |
| Hybrid | Application servers in your cloud, data on premise | Organisations that keep financial data on premise but host applications in the cloud |
| On-premise | The whole stack inside your own boundary; air-gap capable | Regulated and secure sites, and any organisation that prefers no external dependency |

The full feature set is available in every model. No component needs an internet connection, and there is no licence engine or activation server. The licence includes a development instance as well as production at no extra charge, so hierarchy and cube work can be prepared and tested away from live reporting.

## 8. Security in brief

Security is covered in full in the Security and Controls Overview. The architectural points are:

- **One-way integration.** Wevo has no connection to the General Ledger and holds no GL credentials, so a compromise of Wevo cannot reach the ledger.
- **Deny by default.** A user with no profile for a hierarchy sees nothing in it. Access is granted by node or by level, so it mirrors organisational responsibility.
- **No dynamic SQL in any production procedure, function or trigger of the Wevo code. Database access uses stored procedures and parameterised statements.**
- **Least privilege.** Each component has its own database login holding only what it needs; the Email Reporting Engine can run Wevo stored procedures and read the system parameters, and nothing else.
- **Credentials outside the code**, in protected configuration files readable only by the service that uses them.
- **Reports protected after delivery.** Reports go only to the address registered against the user, never to the address a request came from. They are sent as ZIP attachments and can be AES-encrypted with a per-user password, which is stored encrypted.
- **Everything logged.** Every file load, build, instruction and report request is recorded.

Wevo is designed to operate within SOC 2 and ISO 27001 controlled environments. Because it runs entirely in your estate, the vendor never holds or processes your data.

Copyright 2026 Professional Software Ltd. Distribute freely.

# 9. Operation and monitoring

Wevo is designed to run with very little manual intervention:

| Facility | Purpose |
|---|---|
| Logging | Every file received, validation result, build, instruction, request and dispatch is recorded in the logging databases |
| Heartbeat | The Summarisation Engine updates a heartbeat file at a regular interval, so your monitoring can confirm that it is running |
| Daily digests | Each engine emails the administrator a daily summary of the previous day's activity and any problems, including a stale or missing heartbeat |
| Automatic retry | Temporary database or network problems are retried automatically; a valid file is never quarantined because of an outage |
| Quarantine | Files with data problems are set aside with the reason recorded, and nothing from them is loaded |
| Housekeeping | Archived files and logs are cleared automatically after configurable retention periods; messages in the Processed and Quarantine mailbox folders are kept until you set a period |
| Services | Both engines run as standard operating system services, so they start with the server and are managed with the tools your team already uses |

# 10. Scale and performance

| Measure | Figure |
|---|---|
| Drilldown response | Typically under 1 second |
| Currency of data | Near-real-time: minutes behind the GL, depending on how often you extract |
| Summarisation throughput | Around 4 million summarised rows calculated and updated per minute |
| Hierarchy depth | Up to 20 levels |
| Nodes and segment values | No architectural limit; 25,000 organisational nodes and 5,000 chart of accounts nodes demonstrated |
| Years side by side | Up to 9 |
| Periods | 16 per year, plus the opening balance |
| Users and report recipients | Unlimited |
| Downtime during data loads | None for incremental loads: data is released cube by cube using an A/B table configuration. The weekly full balance build runs out of hours |

Copyright 2026 Professional Software Ltd. Distribute freely.

Performance comes from pre-calculation. Every roll-up is calculated and stored when the data arrives, not computed when a user asks for it, so a Drilldown reads stored figures rather than aggregating the ledger on demand. The Summarisation Engine itself calculates and updates around 4 million summarised rows a minute, which is why a movements file is reflected in reporting so quickly.

## 11. Related documents

| Document | Read it for |
|---|---|
| Business Introduction | The complete overview of what Wevo does, illustrated end to end |
| Security and Controls Overview | Access control, governance of reporting output, audit trail and vendor continuity |
| From First Call to Live Reporting | How an evaluation and implementation run, and what each stage needs from you |
| Implementing Multi-Level Management Reporting | Designing hierarchies and cubes, and running the adoption |
| Components and Installation Prerequisites | Package-level installation requirements for each module; supplied at implementation |
| File layout specifications | The exact column layouts of the three GL files; supplied at implementation |

Copyright 2026 Professional Software Ltd. Distribute freely.