Wevo GL Analytics

Security and Controls Overview

For IT security, risk, and audit review

Version 6.0 | August 2026 | Document reference WEVO-ENG-OVW-002

1. Purpose and Scope

This document sets out the security architecture and operational controls of Wevo GL Analytics for review by IT security, risk, and audit functions. It covers the deployment boundary, data integration, access control, the governance of derived reporting output, the email reporting channel, application security, audit logging, retention, and vendor continuity.

It is written to be read in full by a reviewer with no prior knowledge of the product. Section 2 summarises the control model for readers who need only the position; the remainder provides the substantiation.

1.1 What this document does not claim

Wevo GL Analytics is a software product deployed within the client's own infrastructure. It is not a service, and Professional Software Ltd neither hosts nor processes client data. Statements in this document therefore fall into two categories, and the distinction is made explicit throughout:

- Capabilities of the product, which can be verified in the supplied source code and in a deployed instance.
- Responsibilities of the client, which the product supports but does not enforce, such as infrastructure hardening, identity provider configuration, and mail system policy.

Section 13 sets out the split formally. Section 14 states the residual risks that remain after the controls in this document are applied, and is intended to be read as part of the reviewer's assessment rather than as a disclaimer.

Wevo does not hold SOC 2 or ISO 27001 certification. It is designed to operate within environments certified to those standards, inheriting the client's controls under the shared responsibility model.

2. Control Model at a Glance

The security position rests on five structural properties. Each is a consequence of the architecture rather than a configuration option, which is what makes them durable.

Property	What it means for a reviewer
Nothing leaves the estate	Deployed entirely on client infrastructure. No SaaS dependency, no telemetry, no licence call-home. Data residency is met by construction, not by contract.
No connection to the source GL	Integration is file-based. Wevo holds no GL credentials and cannot write to, query, or affect the general ledger.
Deny by default for reporting	A reporting user sees nothing in a hierarchy until a security profile grants it. Access mirrors organisational responsibility rather than being maintained as a separate list.

Property	What it means for a reviewer
Both hierarchies must permit	A cube is visible only where the user holds a profile on its organisation hierarchy and on its accounts hierarchy. One profile alone grants nothing.
No dynamic SQL anywhere	All database access is through stored procedures with typed parameters. The SQL injection surface is eliminated by design, not mitigated by filtering.
Source code supplied	The client receives the complete source at installation and may review it. Every claim in this document is verifiable rather than asserted.
Protection that outlives delivery	Delivered reports are zipped and, where a password is allocated, encrypted. The finished report stays protected after it has left the application.

3. Deployment Boundary and Data Sovereignty

Wevo is deployed entirely within the client's own infrastructure: client-controlled cloud, hybrid, or fully on-premise. All financial data remains inside the client's environment at all times. There is no SaaS component, no third-party data processing, and no cross-border transfer mechanism introduced by the deployment.

The summarisation engine requires no internet connection of any kind. There is no licence verification service, no activation, and no telemetry, so the product is capable of running in an air-gapped environment. Data residency obligations under GDPR, CCPA, and equivalent regimes are therefore satisfied by the deployment model itself.

The licence includes both a development and a production instance at no additional cost. This allows hierarchy and cube development, training, and testing to be carried out on a separate instance, supporting segregation between change activity and live reporting.

3.1 Components and where they sit

Component	Placement	Notes
Application database	Client infrastructure	SQL Server. Holds hierarchies, summarised balances, security profiles, and audit records.
Web tier	Client infrastructure	Python (FastAPI). Serves the drill-down and maintenance interfaces to authenticated users.
Batch and summarisation engine	Client infrastructure	Monitors the landing folder, validates and loads extracts, rebuilds summaries. No internet access required.
Email reporting service	Client infrastructure, normally a separate server	Sends report request forms and returns completed reports through the client's own mail system.

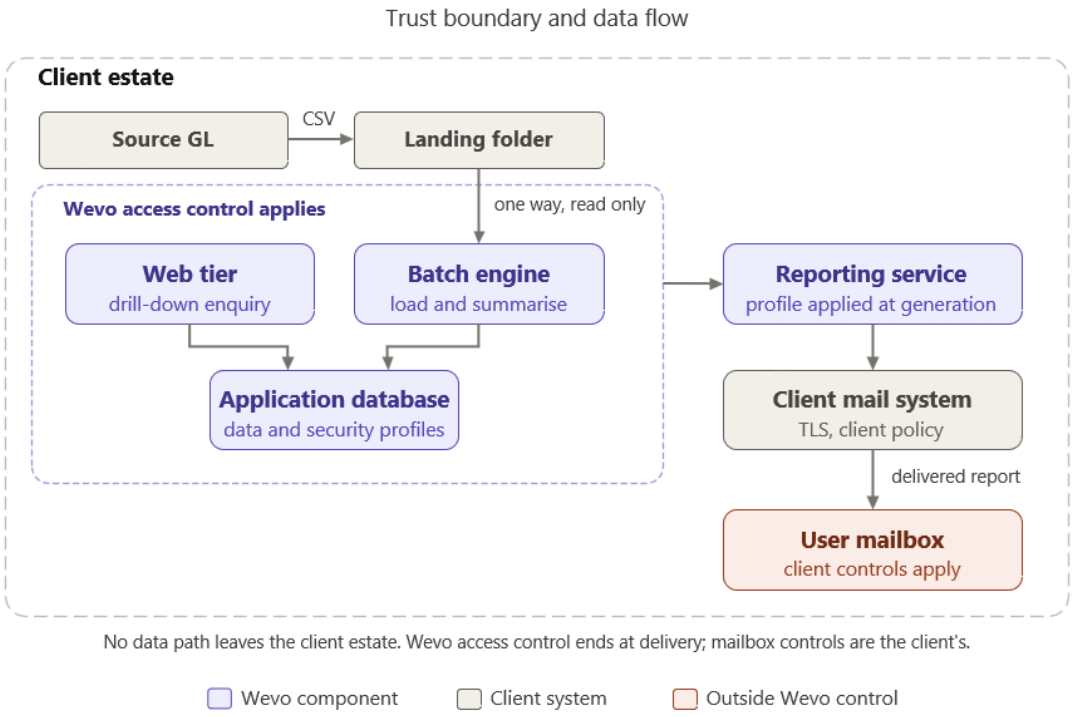

Figure 1 - Trust boundary and data flow. Wevo access control ends at delivery; mailbox controls are the client's.

4. Integration Security

Wevo has no live connection to the source general ledger. The GL system exports CSV extracts to a landing folder, and the Wevo batch engine monitors that folder, validates the content, and loads it. The file types on the feed are segment values, hierarchies, balances, and movements.

Three consequences matter for a security review:

- Wevo requires no credentials to the GL database, so no privileged account is created, stored, or rotated on the client's most sensitive financial system.
- Wevo operates as a read-only analytical layer. It has no code path that writes to, queries, or otherwise affects the GL. A compromise of Wevo cannot propagate into the ledger.
- The attack surface of the integration is a file share, governed by the client's existing access controls and monitoring, rather than a network service or an API surface exposed by the product.

Extract frequency is controlled by the client and may be time-based, amount-based, count-based, manually triggered, or a combination. The loading process uses an A/B table pointer so that users continue reading a consistent set of data while the inactive side is being updated.

5. Access Control

Application access is deny by default. A user sees nothing within a hierarchy unless a security profile has been assigned for that hierarchy. There is no implicit or inherited access, and no administrative shortcut that grants visibility without a profile.

5.1 Two populations, two different controls

Wevo distinguishes between the people who consume reporting and the small number who maintain the structures behind it. The two are governed differently, and the distinction matters when assessing this section.

Population	What they use	How access is controlled
Reporting users	Drill-down enquiry and the email reporting channel	The client's access management and Wevo application-level security profiles, as described in the remainder of this section. Deny by default.
Wevo administrators	Maintenance transactions: hierarchy development, cube definition, user and profile administration	The client's own access management. These transactions sit outside the profile model by design.

The reporting population is typically the whole of finance and beyond, since there are no licence seats to limit it. The administrative population is small and named: the people who own the reporting structure. Where that function sits is a matter for the client, and is most often within finance.

Everything described in sections 5.2 to 5.7 governs the reporting population. Administrative access is addressed at section 5.7 and in the shared responsibility model at section 13.

5.2 A cube is visible only where both hierarchies are permitted

Every figure in Wevo sits at the intersection of two structures: an organisation hierarchy and an accounts hierarchy. A cube is defined against a specific pair of them.

A user therefore requires a security profile on the organisation hierarchy and a security profile on the accounts hierarchy of that cube. Holding one without the other grants nothing at all. A cube for which the user lacks a profile on either hierarchy does not appear to them: it is absent from selection lists, cannot be navigated, and cannot be reported on.

This is enforced at the point data is requested, not only in the interface. A report or enquiry that names such a cube is refused before any data is read, and the refusal identifies which hierarchy is missing a profile rather than returning an empty result that might be mistaken for a nil balance.

The practical consequence for a reviewer is that access is granted along two independent axes and both must be satisfied. Granting a user visibility of a new part of the organisation does not, by itself, expose any additional account detail, and vice versa.

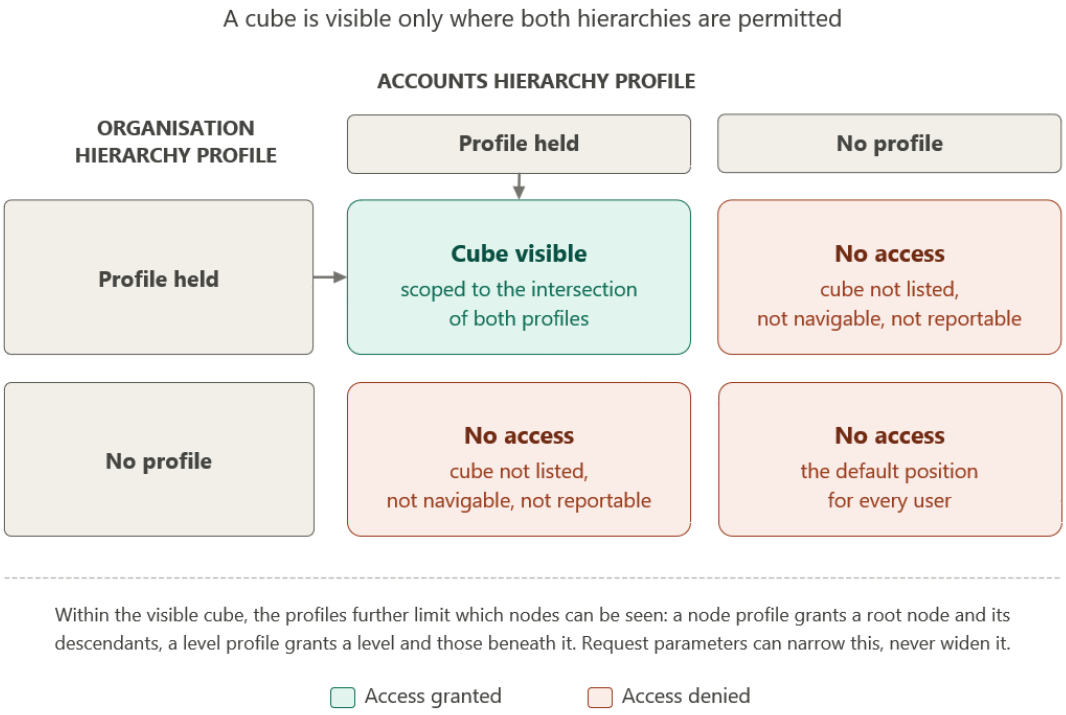

Figure 2 - The two-axis access model. A profile on one hierarchy alone grants nothing.

5.3 Security profile types

Profile type	Grants	Typical use
Node profile	A defined root node and all of its descendant nodes.	A cost centre manager who should see their branch and everything beneath it.
Level profile	All nodes at a specified hierarchy level and the sub-level nodes beneath.	A divisional reviewer who should see every unit at a given tier of the organisation.

Because visibility is expressed against the same hierarchy the business itself uses, access mirrors organisational responsibility by construction. There are no separate report variants, filtered copies, or extracts to maintain and control, which removes the most common source of drift between intended and actual access.

5.4 What the profile governs in drill-down enquiry and reporting

For a reporting user, the security profile is not a report filter. It governs every route by which that user can reach information through drill-down enquiry and the reporting channel, and it applies equally to the three distinct things they might learn:

- Values. A user can only see balances and movements for nodes inside their profile, whether reached through drill-down enquiry, an on-screen report, or the email reporting channel.
- Navigation. Drill-down movement is bounded by the profile. A user cannot step upward, sideways, or downward to a node outside it, so no path exists through the interface that arrives at data the profile does not permit.
- Structure. A user can view hierarchy definitions only within their own security scope. The shape of the organisation beyond their permitted branch is not displayed, and neither are the names or descriptions of the nodes it contains.

The third point is the one most often overlooked in reporting tools and is worth stating plainly. An organisational structure is itself confidential: the existence of a subsidiary, the naming of a new division, or the presence of a cost centre for an unannounced venture can each be sensitive before any figure is attached to it. Because drill-down enquiry scopes the structure to the profile as strictly as it scopes the values, a reporting user cannot enumerate the organisation beyond their own responsibility.

5.5 Request parameters overlay the profile, they never widen it

A report request can narrow what is returned but can never extend beyond the profile. For each hierarchy a request may specify a node, a level range, or both:

- Node only: only that node's data is reported.
- Level range only: only nodes within that range which also fall within the security profile are reported.
- Node and level range: the children of the specified node, within the requested levels and within the security profile, are reported.

The profile is applied at the point the data is assembled, not as a filter applied to a wider result set. A request naming a node outside the user's profile is rejected and returns no data.

5.6 Per-parameter control on assigned reports

Where a user is assigned a report for the email channel, each parameter carries four independent settings: whether it is used at all, whether the user can see it, whether the user may change it, and whether it is required. A parameter can therefore be applied silently, displayed as a read-only value, or offered as a field the user may adjust within their permitted range.

This is the mechanism by which an administrator fixes the organisational scope of a recurring report while leaving the reporting period open for the user to select. A value the user is not permitted to edit is re-imposed from the stored assignment when the request is processed, whatever the request contains.

Whenever a user modifies the report request parameters via the email subject line, those parameters are always validated against the user's assigned report definition and security profile. The system will only process the request if all parameters fully comply with these established settings.

5.7 Administrative access to maintenance transactions

Maintenance transactions are a separate set of functions from drill-down enquiry. They cover hierarchy development, cube definition, system parameters, and the administration of users, profiles, and report assignments. They are intended for a small, named group of Wevo administrators: the function that owns the reporting structure, wherever the client chooses to place it.

These transactions are deliberately outside the security profile model. An administrator maintaining a hierarchy must be able to see and edit the whole of it, in the same way that the person who owns a chart of accounts must be able to see all of it. Restricting a maintainer to a branch of the structure they are responsible for building would make the function unusable.

Access to the maintenance transactions is therefore a client responsibility, secured through the client's own access management and identity systems in the same way as any other privileged administrative function. Wevo's position is:

- The maintenance transactions are separated from the enquiry and reporting interfaces, so administrative capability is not reachable from the interface a reporting user is given.
- Every maintenance action is recorded in the audit log with the acting user and a time stamp, so administrative activity is evidenced even though it is not restricted by profile.
- The develop, validate, promote lifecycle described at section 9 means a hierarchy change made by an administrator does not reach production until a separate, logged promotion event, providing segregation between the act of changing and the act of releasing.
- The separate development instance included in the licence allows maintenance work to be carried out away from production entirely.

This is recorded as a residual risk at section 14, so that a reviewer assesses it against the client's own privileged access controls rather than assuming the profile model covers it.

5.8 Identity and authentication

Identity provider integration points are available, including Azure AD and Okta. Multi-factor policy is enforced within the client's existing identity infrastructure rather than by the product, so authentication strength is governed by the client's own standards and is not weakened by the introduction of Wevo.

6. Derived Output: the Most Sensitive Artefact

Most security assessments of a reporting platform concentrate on the database. For a management reporting product that emphasis is incomplete, and Wevo is designed on the opposite assumption: the finished report is more sensitive than the data it was built from.

6.1 Why the output carries higher impact than the input

A general ledger extract is a flat file of segment values, account codes, and amounts. Interpreting it requires the chart of accounts, the organisational hierarchy, the period conventions, and knowledge of the sign convention in use. Its confidentiality matters, but its immediate intelligibility to an unintended recipient is low.

A Wevo report is the deliberate inverse. It is labelled, aggregated to management level, titled with the organisation node it covers, dated, denominated, and self-describing. A single page can disclose the profitability of a business unit, the cost base of a department, or the position of an entity in a form that any recipient understands immediately and can act upon or circulate without further work.

The product's core value, making financial structure legible, is therefore also the point of greatest exposure. Controls are concentrated accordingly: at the moment of generation, at the moment of delivery, and over the retention of what has been produced.

6.2 Controls applied to generated output

Control	Effect
Profile enforcement at generation	The security profile is applied while the report data is assembled. Output can only ever contain data the requester was entitled to see.
Parameter re-validation on receipt	Every parameter on an inbound request is validated against the stored assignment and the profile before any data is read.
Delivery to the registered address only	A completed report is returned to the address held for that user. A request that reaches the system from elsewhere gains the sender nothing.
Row limit per assignment	Each report assignment carries a maximum row count. Output that reaches the limit is truncated and marked explicitly as incomplete, so a partial extract cannot be mistaken for a full one.
Provenance printed on every output	Report identity, organisation node, period, currency, denomination, and zero-suppression setting appear on every format, so any figure can be traced to the parameters that produced it.
Full audit record	Every request, its validation outcome, the parameters used, and the dispatch of the finished report are recorded with a time stamp and the user identity.
Retained copy with scheduled disposal	Generated reports are archived on the reporting server for a configurable period and purged automatically thereafter.
Nothing silently discarded	A request that cannot be processed is moved to a quarantine folder for administrator review rather than deleted, so failures are visible rather than absent.

6.3 Protection of the delivered file

Every report returned to a user is delivered as a compressed archive rather than as a bare file. This keeps attachment sizes small, which matters for a large trial balance and for mail systems that enforce attachment limits.

Where a Wevo administrator has allocated a report file password to a user, that archive is encrypted and the recipient must supply the password to open it. The password is set by the administrator, not chosen by the recipient, and is held against the user rather than against an individual report, so it applies to every report that user receives. Where no password has been allocated, the archive is compressed but not encrypted.

This is the one control in this document that continues to operate after delivery. The residual risk recorded at section 14, that a report leaves the application's access control once it reaches a mailbox, is materially reduced where a password is in force: a report that is forwarded, copied to a shared drive, or left in an unattended mailbox cannot be opened without it.

Property	What it means for a reviewer
Password set by an administrator	Allocated on the user record through the maintenance screens. The recipient does not choose it and cannot change it.

Property	What it means for a reviewer
Encrypted archive	Where a password is set, the delivered archive is AES-encrypted using the pyzipper library.
Never displayed after entry	The maintenance screens show only whether a password is currently set. The stored value is never returned to the browser or shown to an administrator.
Stored encrypted, not in readable form	The password is encrypted before storage using Fernet from the Python cryptography library: AES-128 in CBC mode with HMAC-SHA256 authentication, so a stored value cannot be altered undetected.
Key held outside the database	The encryption key is read from a protected environment file on the server. It is not held in the application database, so database access alone does not yield the passwords.
Decrypted only at the point of use	The reporting engine decrypts the stored value at the moment the archive is written, and holds it only for that operation.

One practical consequence is worth stating. AES encryption is not supported by the archive handler built into Windows or by the macOS Archive Utility, so a recipient needs a tool such as 7-Zip, NanaZip, PeaZip or WinRAR to open a protected report. The delivery email says so explicitly and names those tools, but allocating a password does place that requirement on the user. The weaker legacy ZipCrypto scheme, which those built-in handlers do support, is deliberately not used: it is defeated by a known-plaintext attack and a Wevo report has a highly predictable header, so it would offer the appearance of protection rather than the substance.

Two operational points follow from the design. The key must be protected as carefully as the database itself, since it is what makes the stored values readable; it belongs in the same protected environment file as the engine's other credentials, with filesystem permissions to match. And because the stored values are encrypted under that key rather than derived from it, rotating the key renders existing stored passwords unreadable and they must be re-entered.

6.4 Where the client's controls take over

Once a report has been delivered it resides in the recipient's mailbox, which is outside the application's access control. This is an inherent property of email delivery rather than a weakness specific to Wevo, and it is stated plainly here because it is the point at which the client's own controls become the operative ones.

The following are recommended and are the client's to implement:

- Mailbox retention and deletion policy covering delivered report attachments.
- Data loss prevention rules on outbound mail, if reports must not leave the organisation.
- Sensitivity labelling or classification applied to attachments by the mail platform.
- Filesystem permissions on the report archive directory, restricting it to the service account and named administrators.
- Inclusion of report recipients in the joiners, movers, and leavers process, so that a departing user is removed from report assignments as well as from the application.

Section 14 records the residual risk that remains after these measures.

7. The Email Reporting Channel

The email reporting service delivers report request forms to authorised users and returns completed reports to their inboxes. It is normally installed on a separate server and uses the client's internal mail system. It is the only part of the product that moves data beyond the application interface, and it is controlled accordingly.

7.1 How a request is authorised

Each inbound request passes a sequence of independent checks before any data is read. A failure at any stage stops the request:

1. The sender address must correspond to a registered, active user. Unrecognised senders are discarded without a reply and without an audit record, so the system discloses nothing about whether an address is known.
2. A per-user rate limit is applied over a rolling twenty-four hours, so no single account can saturate the service with requests.
3. The request must carry a validation token issued with the form. The channel does not trust the sending address alone.
4. Every parameter is validated against the stored report assignment and the user's security profile. Values the user is not permitted to change are re-imposed from the assignment.
5. The completed report is returned to the registered address held for that user, not to the address the request arrived from.

Requests that fail validation receive a plain rejection explaining the reason. They return no data, and the outcome is recorded in the audit log.

7.2 Content of a request form

The catalogue of reports in each form is scoped to the recipient's security profile, and the parameters offered for a report are typically a subset of what the user could otherwise access. A user cannot request a report that has not been assigned to them, and cannot widen the scope of one that has.

7.3 Transport

Mail is collected and sent over authenticated, TLS-protected connections to the client's mail infrastructure. Where the reporting service reaches the application database across a network boundary, the connection is made over an encrypted tunnel and the database port is not exposed publicly.

Publishing SPF, DKIM, and DMARC records for the sending domain, and enforcing DMARC on the mailbox that receives requests, is recommended. This is a client-side mail configuration and is the appropriate control against forged sender addresses, complementing the token validation described above.

7.4 Output formats

Reports are produced as plain text, PDF, HTML, or CSV. The format is selected by the user from those enabled on the assignment. Whatever the format, the finished report is delivered as a compressed archive, encrypted where a report file password has been allocated to that user (section 6.3).

8. Application Security by Construction

8.1 No dynamic SQL

There is no dynamically constructed SQL anywhere in the code base. All database access is performed through stored procedures with typed, bound parameters. This is a deliberate design constraint adopted for client security rather than a coding preference.

For a reviewer this has two practical consequences. The SQL injection surface is eliminated structurally rather than mitigated by input filtering, so it cannot be reintroduced by a missed sanitisation call. And security assurance is simplified: the complete set of database operations is enumerable from the stored procedure catalogue, which makes code review and penetration test scoping straightforward.

8.2 Input handling

Inputs are validated at the boundary and again in the database layer. Character validation is applied to identifiers, numeric parameters are range checked, and content received by the email channel is stripped of control characters and truncated before storage. Values that fail validation produce a defined error from the message catalogue rather than a system message, so internal detail is not disclosed to the user.

8.3 Source code availability

The complete source code is supplied at installation. A client's security function may review it, subject it to static analysis, or include it in an internal assurance programme without a separate agreement or a source code escrow release. Every statement in this document is verifiable against that code.

9. Hierarchy Change Governance

Hierarchy maintenance is self-service for finance but governed by a develop, validate, promote lifecycle. Changes are developed and validated in a maintenance state while production hierarchies continue to serve users and the summarisation engine unaffected.

A validated hierarchy is set to promote status, which locks it, and it is copied to production at the start of the next full build: a single, controlled, logged event. The separation between the person making a change and the moment it takes effect is structural rather than procedural.

This maps onto the change management controls tested under IT general controls reviews, providing segregation of duties, staged promotion, and an audit trail of what changed and when. The separate development instance included in the licence allows complex changes to be rehearsed before they reach the production environment at all.

Who may perform hierarchy maintenance is controlled by the client, as set out at section 5.7. The controls described here govern what happens to a change once it has been made, and apply irrespective of which administrator made it.

10. Audit Logging and Evidence

Wevo maintains time-stamped audit records across the platform: data loads, hierarchy changes, user administration, report requests, validation outcomes, and report dispatches. Records identify the acting user and are queryable at any time through the database.

10.1 What the audit trail supports

- Demonstrating who had access to which part of the organisation, and when, in support of access recertification.
- Evidencing the change history of a reporting hierarchy for an IT general controls review.
- Reconstructing which reports were produced, for whom, with which parameters, over any period covered by retention.

10.2 Integrity of the audit trail

Audit records are held in the application database and are protected by the database encryption and access controls applied by the client. They are not, in themselves, tamper-evident against a sufficiently privileged database administrator, and this document does not claim otherwise.

Where the client's control environment requires tamper evidence, the appropriate measure is to forward audit records to the client's existing SIEM or to write-once storage on a schedule. Wevo's records are queryable through standard SQL and are straightforward to ship in this way. Section 14 records this as a residual risk with its mitigation.

11. Encryption and Infrastructure

Data at rest is protected using the encryption available in the client's platform, typically transparent database encryption or cloud provider volume encryption, with AES-256. Data in transit uses TLS 1.2 or 1.3. Certificate management is handled by the client within their existing infrastructure.

Wevo runs on a moderate, standard server: a Python web tier and a SQL Server database, with MySQL used for web tier logging. There are no in-memory database licensing and no specialist hardware, so the deployment sits inside the client's existing patching, monitoring, vulnerability management, and backup regimes rather than requiring an exception to them.

12. Data Retention and Disposal

Retention is configurable and enforced by a scheduled housekeeping process rather than left to manual clearance. The categories below are the defaults supplied with the email reporting service and are expected to be set to the client's own policy at installation.

Data	Default retention	Disposal
Report request audit records	90 days	Automatic deletion of completed records after the retention period.
Report dispatch records	90 days	Automatic deletion after the retention period.
Incomplete or abandoned request records	1 day	Automatic deletion, so failed runs do not accumulate.
Archived report files	30 days	Automatic deletion of the stored copy of each generated report.
Delivered reports in user mailboxes	Client mail policy	Governed by the client's mail retention rules, not by Wevo.

Application audit records covering hierarchy changes, user administration, and data loads are retained according to the client's own policy, since these are the records most likely to be required for a control review.

13. Shared Responsibility Model

Wevo is installed and operated by the client. The division below is the practical basis for a control review: the left column is assessed against the client's own control environment, the right column against the product and the supplied source code.

Control area	Client responsibility	Wevo product capability
Infrastructure and network	Servers, segmentation, firewalls, patching, monitoring	Runs on standard infrastructure within existing controls; no exceptions required
Identity and MFA	Identity provider, MFA policy, session policy	Integration points for Azure AD and Okta
Database access	DBA permissions, separation of duties, privileged access management	Stored procedure architecture with no dynamic SQL and no ad-hoc data paths
Application access (reporting)	Profile assignment, periodic recertification, leaver removal	Deny-by-default node and level profiles enforced at data assembly
Administrative access (maintenance)	Restricting the maintenance transactions to named administrators through the client's access management	Maintenance functions separated from the enquiry interface; every administrative action audit-logged; staged promotion to production
Reporting output	Mail retention, DLP, classification, archive directory permissions	Profile enforcement at generation, delivery to registered address, scheduled purge of archived copies
Mail infrastructure	Mail server, SPF, DKIM, DMARC enforcement, transport policy	Authenticated TLS connections; token validation independent of sender address
Audit	Retention policy, SIEM forwarding where tamper evidence is required	Time-stamped logging of loads, changes, requests, and dispatches
Encryption	TDE or volume encryption, certificate management	Operates with AES-256 at rest and TLS 1.2 or 1.3 in transit
Backup and continuity	Backup schedules, DR infrastructure and testing	Full rebuild capability from source GL extracts
Report file passwords	Deciding which users require a password, communicating it to them, and rotating it	Per-user password, encrypted at rest under a key held outside the database; AES-encrypted delivery archive

14. Residual Risks and Mitigations

The following risks remain after the controls described above are applied. They are stated explicitly so that a reviewer can assess them against the client's own risk appetite rather than having to infer them.

Risk	Assessment and mitigation
A delivered report leaves the application's access control once it reaches a mailbox.	Inherent to email delivery. Mitigated by profile enforcement at generation, delivery only to the registered address, and a full audit record of what was produced. Where a report file password has been allocated the delivered archive is encrypted, so the report remains protected after it has left the application (section 6.3). Residual exposure for users without a password is managed by the client's mail retention, DLP, and classification policy (section 6.4).
A sender address on an inbound request can be forged.	Mitigated in the product by the validation token and by returning output only to the registered address, so a forged request yields no data to the sender. The complementary control is DMARC enforcement on the receiving mailbox, which is a client mail configuration.
A sufficiently privileged database administrator could read or alter data, including audit records.	Not preventable by an application running on the client's own database. Managed by the client's separation of duties and privileged access management. The product provides no application path that bypasses security profiles, so this requires database-level privilege rather than misuse of the application.
Archived report copies reside on the reporting server filesystem.	Mitigated by scheduled automatic purge and by the absence of any application interface that exposes the archive. Directory permissions and volume encryption are client controls.
The reporting service depends on the client's mail infrastructure being available.	An availability rather than a confidentiality risk. Requests that cannot be processed are retried and, if they still cannot be completed, are quarantined rather than lost, and the administrator is notified through the daily activity digest.
A Wevo administrator using the maintenance transactions can see the whole hierarchy structure, outside the profile model.	By design: a structure owner must be able to maintain the whole structure. The population is small and named. Mitigated by separation of the maintenance transactions from the enquiry interface, by audit logging of every administrative action, and by staged promotion to production. Restricting who reaches those transactions is a client responsibility under its privileged access controls (section 5.7).
Endpoint and physical security of the machines on which reports are read.	Outside the product boundary entirely. Assessed under the client's existing endpoint and physical security controls.

15. Vendor Risk and Continuity

Wevo is supplied with its complete source code at installation, under a licence containing a continuity provision: if the product can no longer be supported by the vendor, the client retains the source and may continue the life of the product using in-house teams or AI-assisted tooling.

This is a materially stronger position than source code escrow. There is no release procedure to invoke, no trigger event to dispute, and nothing held by a third party that may have gone stale against the deployed version. The client already holds the code that is running.

For regulated buyers assessing ICT third-party risk under frameworks such as DORA, this provides a documented exit-strategy artefact and removes the dependency of the reporting service on the continuity of the vendor. It does not, by itself, constitute a complete exit plan: the client remains responsible for the transition arrangements, capability, and testing that those frameworks require.

16. Compliance Alignment

The table below describes how the deployment model and controls address common regulatory expectations. It is a statement of alignment, not of certification.

Framework or standard	How it is addressed
GDPR, CCPA, and regional privacy law	Financial data never leaves the client estate and no third party processes it, so residency and processor obligations are met by the deployment model. Wevo holds general ledger data; it does not require personal data beyond the identity of its own users.
SOX and internal control over financial reporting	Time-stamped audit records of data loads, hierarchy changes, and report production; a governed develop, validate, promote change lifecycle; deny-by-default access aligned to organisational responsibility.
DORA and NIS2	Client-owned infrastructure and BCDR, plus the Continuity Licence as a documented exit-strategy artefact (section 15).
SOC 2 and ISO 27001	Designed to operate within environments certified to these standards, inheriting the client's controls under the shared responsibility model. Wevo does not itself hold these certifications.
NIST CSF and CIS Controls	Deployment within client-controlled security infrastructure; AES-256 at rest and TLS 1.2 or 1.3 in transit; elimination of the injection surface by design; auditable access and change history.

Wevo processes general ledger and management accounting data. It is not a cardholder data environment and makes no claim of PCI DSS applicability.

17. Supporting a Security Review

The following are available to a client security function during evaluation, without additional agreement:

- The complete source code, for review or static analysis.
- The stored procedure catalogue, which enumerates every database operation the application can perform.
- A deployed instance for penetration testing within the client's own environment.
- An architecture review session covering any part of this document in detail.
- The installation guide, which documents the service account, credential handling, and file permissions used by the deployment.

Questions, and requests for an architecture review session:

Alan Webster | info@wevo.co.uk | wevo.co.uk