ProcessMind Audit Log and SIEM Statement

Effective date: November 12, 2026

This statement describes how ProcessMind B.V. records security-relevant and administrative actions performed against Customer Data (the "audit log"). It supplements the Security Measures, the Data Processing Addendum, the Customer Agreement and the Sub-processor List.

---

1. Scope

1.1 One audit log. ProcessMind maintains a single audit log for security-relevant and administrative actions performed against Customer Data and the ProcessMind Service. The audit log is separate from operational (application) logging, from HTTP access logging and from the product's activity feed; those streams serve different purposes and are not represented as an audit trail.

1.2 What it is not. The audit log is not a record of every read of Customer Data, and it is not the product's change history for process models, diagrams, documentation or simulations. Those are recorded in the process change history available inside the product.

---

2. What each audit record contains

2.1 Fields. Each record carries the fields below. The set is fixed by the record schema: a record cannot grow a field that is not on this list, and an action cannot add one it did not declare in the registry.

- *schemaVersion* (number): record schema version, currently 1.
- *eventId* (string): unique record identifier. New records use a ULID; rows migrated from the earlier store keep their UUID.
- *occurredAt*, *ingestedAt* (timestamps): when the action happened, and when the audit writer stored the record. Both in UTC.
- *eventClass* (enum): security, auth, admin, data, api, ai or platform.

- *action* (enum): one code from the closed, reviewed action registry, for example auth.login.blocked or api.key.revoked.
- *outcome* (enum): success, failure, denied or error.
- *severity* (enum): info, notice, warning or critical, derived from the outcome. A successful action uses its registry default (info unless the action says otherwise), failure is recorded as notice, and denied or error as warning.
- *actor* (object): who acted. type (user, apiKey, service, system or anonymous), id, email, displayName, and onBehalfOf plus onBehalfOfEmail where the action ran for another account.
- *actorContext* (object): the acting session. sessionId, authMethod (oidc, password, loginToken, apiKey or none) and a support-session flag.
- *organizationId*, *tenantId* (identifiers, nullable): the organization and the environment the action belongs to. Null for records that concern no customer, or that sit above the environment.
- *resource* (object, nullable): the affected object. type, id and name, where type is one of apiKey, artifact, connection, database, datatable, export, lead, model, organization, process, role, session, setting, storage, tenant or user.
- *source* (object): where the record was produced. service (api, externalApi, superadminApi, worker, cron or webSocket), function and version.
- *stage*, *region* (strings): the deployment stage that produced the record (for example prod) and the AWS region (eu-central-1 for customer records).
- *correlation* (object): requestId, traceId and parentEventId, which tie the record to the request that caused it and to the records before it in the same flow.
- *network* (object, nullable): sourceIp and userAgent. Captured only for the actions whose registry entry opts in: sign-in, access decisions, administrative and settings changes, data uploads, downloads and exports, API-key changes and AI-policy changes.

- *detail* (object): the action's own payload, validated against that action's schema. Values are primitives only (string, number, boolean, null), so the record stays queryable and redactable.

2.2 A closed vocabulary. Actions are drawn from a fixed, reviewed list of action codes; there is no free-text message in an audit record. Each action also declares which additional fields it may carry, so a record cannot silently grow a new kind of content.

---

3. Which actions are recorded

3.1 Recorded today. ProcessMind records, among others:

- **Authentication:** sign-in success and failure, blocked sign-in (locked account, disallowed domain, unverified provider email), sign-out, login-token issuance and handoff.
- **Authorization:** access denials, including every rejected external-API credential, scope and tenant state, and origin (CORS) rejections.
- **Support access:** the start of a ProcessMind support session on a customer account, naming both the operator and the affected account.
- **API credentials:** creation, scope change and revocation of API keys, including the granted scopes.
- **Account and role administration:** user creation, invitation, administrator profile edits, deletion, role grants and revocations, and tenant or organization membership changes.
- **Organization lifecycle:** creation, deletion, restore, deletion scheduling and cancellation, and completed erasure.
- **Data upload, download and export:** artifact upload, download and deletion, tabular-data upload and update, export downloads, data table deletion, and webhook (connection) creation, change and deletion.
- **Billing state:** plan changes, payment-driven suspension and recovery, and subscription cancellation.

- **ProcessMind's provider console:** privileged query execution, user and organization administration, settings and feature changes, billing synchronisation requests and migrations.
- **Administrative and destructive actions by ProcessMind's scheduled jobs:** tenant, organization, user and lead deletions, retention sweeps, and pipeline failures.
- **Audit-log integrity:** failures of the audit log's own write path.

3.2 Workspace edits. Edits inside a workspace (process models, documentation, simulations, dashboards, comments and bookmarks) are recorded in the product's own process change history, not in the audit log. The audit log records who uploaded, downloaded, exported or deleted data; the change history records what changed inside a model. The current list of recorded action codes is available from ProcessMind on request.

---

4. What is never recorded

Passwords, authentication tokens, API-key values, cookies, authorization headers, request or response bodies, file contents, presigned URLs, and AI prompts or completions are never written to the audit log. Audit records may contain personal data necessary for attribution: an actor's email address, and the source IP address and user agent for the actions whose registry entry opts in (sign-in, access decisions, administrative and settings changes, data movement, API-key and AI-policy changes). The permitted fields are defined per action and reviewed when an action is added.

---

5. Integrity

5.1 Append-only. The audit log is append-only. No application role holds an update or delete privilege on it, and a database constraint rejects both operations.

5.2 One writer. Records are written by a single component that validates the action code and its payload before writing. A record that does not validate is rejected and reported rather than written in part.

5.3 Failure handling. A failed audit write never fails the business action it describes. It is recorded on an alarmed operational path, and the record itself is preserved in a secondary copy so that it can be recovered.

---

6. Retention and location

6.1 Retention. Audit records are retained for at least ten (10) years from the event date.

6.2 Location and encryption. Audit records are stored in the European Union (Frankfurt, Germany, `eu-central-1`) and are encrypted at rest with a customer-managed AWS KMS key. A live copy is additionally written to an access-controlled Amazon CloudWatch log group so that an incident can be investigated while it is happening.

6.3 No cross-region replication. Audit records are not replicated outside the European Union.

---

7. Access to audit records

7.1 Entitlement. The in-app audit log is included in the Process Architecture plan and above. An organization administrator reads the whole organization; a tenant-scoped administrator reads their own tenant. The scope is derived from the authenticated session and checked before the query, so a request cannot widen it to another tenant or organization.

7.2 ProcessMind personnel. ProcessMind personnel access audit records only for support, incident response or a legal obligation, on a need-to-know basis. Cross-customer access is restricted to a dedicated support role, and such access is itself an administrative action.

7.3 How to obtain records. Organization and tenant administrators can read their audit records in the in-app audit log, and pull or export them through the external API. For a larger or assisted request, ProcessMind support provides the records in CSV or JSON for a specified organization, tenant and time range, targeting five (5) business days for a standard request.

---

8. Deletion of audit records

8.1 On request. On a verified written request, ProcessMind erases the audit records belonging to a customer organization from the audit store and confirms the erasure in writing. The operational CloudWatch copy cannot be edited record by record; it expires under its own retention window.

8.2 What is retained. The record that the erasure took place (organization identifier, date, requester, reason, and the number of records removed) is retained as a platform-level audit record and contains no Customer Data. Records that belong to no customer (ProcessMind platform events) are not affected by a customer erasure request.

8.3 Individual erasure requests. Where a request concerns an individual within a customer that remains active, the security record is retained for the establishment, exercise or defence of legal claims as permitted by Article 17(3)(b) and (e) GDPR; that basis is recorded in the record of processing activities and in the Privacy Policy.

---

9. SIEM integration and export

9.1 Availability. The immutable audit archive is an Enterprise option, enabled for a customer organization on request. It writes daily compressed NDJSON objects under a per-organization prefix in an Object-Locked (WORM) S3 bucket, with a manifest and nightly reconciliation.

9.2 Current delivery. ProcessMind does not currently offer a native SIEM connector or a push feed. Delivery is pull, from three surfaces: the read API

(`GET /v1/tenant/{tenantId}/audit`), the bounded export job (create it with `POST /v1/tenant/{tenantId}/audit/exports` and collect the result from its status call, which returns a short-lived download URL), and, with the Enterprise archive, the daily objects under the organization's own prefix in the Object-Locked bucket, which the customer's SIEM reads through an access arrangement. ProcessMind does not deliver records to a location of the customer's choosing on its own schedule.

9.3 ProcessMind's own monitoring. ProcessMind's security monitoring consumes the same records: audit-write failures, authentication-failure bursts, privilege changes and destructive administrative actions raise internal alarms.

---

10. Evidence available to customers

On request, and under confidentiality where applicable, ProcessMind provides this statement, the Security Measures, the Data Processing Addendum, the Sub-processor List, a redacted sample of audit records, and the procedure used for access and erasure requests.

---

11. Limitations

For the avoidance of doubt, ProcessMind claims an append-only audit store for every organization and, where the audit-archive entitlement is enabled, an Object-Locked (WORM) copy with nightly manifest reconciliation. ProcessMind does not currently claim a hash chain, a native real-time SIEM push connector, per-user redaction of audit records, an ISO 27001 or SOC 2 attestation covering this control, or coverage of every read of Customer Data. Customer self-service read and export are available through the in-app audit log and the external API.