

Funktionsübersicht

Was AuditMail Server kann, und wozu die einzelnen Funktionen da sind. Die rechtliche Einordnung steht in GOBD.md, die Schnittstelle in API.md, die Einrichtung in DEPLOYMENT-STRATO.md.

Die Oberfläche hat vier Bereiche: **Archiv**, **Integrität**, **Protokoll**, **Einstellungen**.

1. Erfassung

Mehrere Postfächer

Beliebig viele IMAP-Konten, jedes mit eigenem Zugang, eigenem Zeitfenster und eigener Ausschlussliste. Jeder archivierte Eintrag trägt die Kennung seines Herkunftskontos und den Quellordner — jede Mail ist also einem Postfach zuzuordnen, auch Jahre später.

Ein Konto lässt sich **pausieren** (nimmt am Sammelabruf nicht mehr teil) oder **einzelnen abrufen**. Wird es entfernt, bleiben seine archivierten Mails erhalten; sie sind Teil der Hash-Kette und dürfen nicht verschwinden.

Nur lesender Zugriff

Der Abruf öffnet Ordner ausschließlich mit `EXAMINE` und liest mit `BODY.PEEK[]`. Im Postfach wird nichts verändert: nichts gelöscht, nichts verschoben, nichts als gelesen markiert. Ungelesene Mails bleiben ungelesen — am Postfach ist nicht erkennbar, dass ein Archiv mitliest.

Der IMAP-Client spricht das Protokoll direkt über einen TLS-Socket, weil die PHP-Erweiterung `imap` bei Strato nicht verfügbar ist. Zertifikatsprüfung bleibt aktiv.

Verbindung prüfen

Zeigt vor dem Speichern, ob Anmeldung und TLS funktionieren, welche Ordner der Server meldet und welche davon ausgenommen sind. Ein Konto mit falschem Passwort fällt damit sofort auf, statt erst beim nächtlichen Abruf.

Dublettenschutz

Zwei Stufen: die `Message-ID` der Nachricht (RFC 5322) gegen den Index, dazu die Liste bereits abgeholter IMAP-UIDs je Konto. Die UIDs allein genügen nicht, weil sie sich beim Verschieben zwischen Ordnern ändern. Ein Abruf lässt sich daher beliebig oft wiederholen, ohne Doppeleinträge zu erzeugen.

Unterbrechbarkeit

Pro Durchlauf gilt eine Höchstzahl neuer Mails; ein Rückstand arbeitet sich über mehrere Läufe ab. Das Zeitfenster (Standard 90 Tage rückwärts) ist gleichzeitig das Aufholfenster: Fällt der Abruf mehrere Tage aus, holt der nächste Lauf alles nach, was innerhalb des Fensters liegt.

Übergabe von außen

Sendende Systeme können eine Mail über die Schnittstelle direkt übergeben, unabhängig davon, ob sie in einem Postfach liegt. Gedacht für Ausgangspost aus eigener Software.

2. Ablage

Das Original bleibt unangetastet

Jede Mail wird bytewise unverändert als `.eml` gespeichert und danach nie wieder angefasst. Anzeige, Suche und Auswertung entstehen durch Parsen zur Laufzeit. Alle Kopfzeilen, Signaturen und Verschlüsselungshüllen bleiben erhalten — daran hängt die Beweiskraft.

Neben jeder Datei liegen die Metadaten als JSON, sodass der Bestand auch ohne Datenbank lesbar bleibt. Ablage nach Jahr und Monat.

Index in der Datenbank

Absender, Empfänger, CC, Betreff, Datum, Richtung, Kategorie, Referenz, Anhangsnamen, Größe, Hashes, Frist, Herkunftskonto und Quellordner. Der Index ist auswertbar und veränderbar; das Original ist es nicht.

Der Zeitpunkt einer Mail

Maßgeblich ist der `Date:`-Kopf — mit zwei Vorbehalten. Er trägt die Zeitzone des Absenders; das Archiv rechnet ihn deshalb in seine eigene Zone um, sonst wären die Einträge untereinander nicht vergleichbar und die Sortierung falsch. Und er ist die einzige Angabe hier, die ein Fremder frei bestimmt: Falsch gestellte Uhren sind Alltag, ein Datum in der Zukunft bei Werbemail ein bekannter Trick.

Deshalb der Abgleich mit der obersten `Received:`-Zeile, die der *empfangende* Server schreibt — also das Postfach des Betreibers. Liegt der angegebene Sendezeitpunkt mehr als eine Stunde nach dem Empfang, kann er nicht stimmen, und es gilt der Empfang. Fehlt `Date:`, ebenso.

Die Zone, in der gerechnet wird, steht unter *Einstellungen* → *Automatischer Abruf*; ohne Angabe ist es Europe/Berlin. Sie gilt für alles Zeitliche: Datum im Index, Protokoll, Fristbeginn und das Nachtfenster des Sammelaufrufs. Die `.eml`-Datei bleibt von alldem unberührt — dort steht der Kopf des Absenders unverändert.

3. Unverändertheit

Hash-Kette

Beim Archivieren wird der SHA-256 der Datei gebildet und mit dem Hash des Vorgängereintrags zu einem neuen Hash verkettet. Daraus folgt: Eine geänderte Datei passt nicht mehr zu ihrem Hash, ein entfernter Eintrag bricht die Verkettung des Nachfolgers, ein eingeschobener passt nicht in die Reihe.

Die Reihenfolge ergibt sich aus einer fortlaufenden Nummer der Datenbank, nicht aus einem änderbaren Datum. Beim Einfügen läuft eine Transaktion mit Zeilensperre auf dem letzten Eintrag — zwei gleichzeitige Abrufe können die Kette nicht gabeln.

Integritätsprüfung

Prüft den gesamten Bestand: Dateihash neu berechnen, gegen den gespeicherten Wert und gegen den Vorgänger vergleichen. Das Ergebnis nennt jeden abweichenden Eintrag einzeln und unterscheidet drei Fälle — intakt, abweichend, Datei fehlend. Jede Prüfung steht im Protokoll.

Kettenanker

Der Kettenkopf — der Hash des letzten Eintrags, der rekursiv den gesamten Bestand abdeckt — wird festgehalten und an eine Adresse **außerhalb dieses Servers** gesendet. Damit liegt ein Vergleichswert dort, wo ihn niemand mehr ändern kann, der nur diesen Server beherrscht.

Das schließt die Lücke, die jede reine Hash-Kette hat: Wer Datenbank und Dateien kontrolliert, kann beides ändern und die Hashes nachziehen. Gegen einen außerhalb verwahrten Kopf gelingt das nicht mehr unbemerkt.

Ein Bruch der Ankerfolge ist nicht automatisch verdächtig — eine fristgerechte Löschung baut die Kette neu auf und verändert den Kopf zwangsläufig. Aussagekräftig ist ein Bruch **ohne** zugehörigen Protokolleintrag.

Kette neu berechnen

Vorhanden, aber ausdrücklich keine Alltagsfunktion: Sie überschreibt den Nachweis. Gedacht für die Übernahme von Altbestand ohne Kette und für die Reparatur nach einem abgebrochenen Import — und für den Neuaufbau nach einer planmäßigen Fristlöschung, die zwangsläufig Lücken hinterlässt. Jeder Aufruf steht im Protokoll.

Fehlt eine Datei, ist der richtige Weg das Ergänzen aus der Sicherung, **nicht** der Neuaufbau: Der würde den Verlust nur verdecken.

4. Fristen

Berechnung

Buchungsbelege acht Jahre (§ 147 Abs. 3 AO in der Fassung des BEG IV, ab 01.01.2025), sonstige Geschäftspost sechs Jahre (§ 257 Abs. 4 HGB). Die Frist beginnt mit dem Schluss des Kalenderjahres, in dem die Mail entstanden ist (§ 147 Abs. 4 AO) — nicht am Tag der Archivierung. Der Löschtermin wird bei der Aufnahme berechnet und bei jeder Änderung der Kategorie neu.

Zuordnung

Kategorie und Geschäftsvorfall-Referenz lassen sich je Mail setzen; die Frist wandert mit. Eine Heuristik erkennt bestimmte Nummernkreise im Betreff automatisch. Was sie nicht erkennt, bleibt ohne Kategorie und erhält die kürzere Frist — die Kennzahl *Ohne Zuordnung* zeigt diese Einträge, sie gehören regelmäßig durchgesehen.

Frist-Löschung

Entfernt Originaldatei und Indexeintrag der Einträge mit abgelaufener Frist. Nur auf ausdrückliche Anweisung eines Administrators, nie automatisch. Der Protokolleintrag über die Löschung bleibt. Anschließend wird die Kette neu aufgebaut.

Löschsperre

Ein Schalter mit Anlass, der die Frist-Löschung verweigert, statt nur zu warnen. Nötig, weil § 147 Abs. 3 Satz 5 AO den Ablauf der Frist hemmt, solange die Unterlagen für eine begonnene Außenprüfung oder ein Rechtsbehelfs- oder Steuerstrafverfahren erheblich sind — eine scheinbar abgelaufene Frist ist dann nicht abgelaufen, und eine Löschung wäre nicht rückholbar.

Setzen, Aufheben und jeder abgewiesene Löschkversuch stehen mit Anlass im Protokoll. Die Sperre lässt sich auch fest in der Konfiguration verankern; dann kann sie über die Oberfläche nicht aufgehoben werden.

Fristen neu berechnen

Bestimmt Aufbewahrung und Löschkdatum aller Einträge neu. Nötig für übernommenen Altbestand, dessen Frist ab dem Archivierungstag gerechnet wurde, und nach einer Änderung der Aufbewahrungsdauer. Ändert nur Metadaten.

5. Suchen und Lesen

Suche

Über Absender, Empfänger, Betreff, Referenz und Anhangsnamen, kombinierbar mit Filtern nach Kategorie, Richtung, Herkunftskonto, Jahr, Datumsbereich, sowie zwei Sonderfiltern: *nur mit abgelaufener Frist* und *nur ohne Zuordnung*. Seitenweise Anzeige.

Detailansicht

Kopfdaten, Nachrichtentext, Anhänge und ein Abschnitt **Nachweis** mit Kennung, eigenem Hash, Vorgänger-Hash, Aufbewahrung, Löschdatum, Größe und dem Dateipfad. Oben steht, ob die Integrität bestätigt ist.

HTML-Nachrichten werden in einem abgeschotteten Rahmen dargestellt: ohne Skripte, ohne Zugriff auf die Anwendung, ohne Nachladen externer Bilder. Fehlt die Originaldatei, sagt die Anzeige das ausdrücklich und nennt den erwarteten Pfad, statt eine leere Nachricht vorzutäuschen.

Anhänge

Werden erst auf Anforderung geladen, damit eine Mail mit großem Anhang nicht bei jedem Öffnen den Speicher belastet. PDF, Klartext und Bilder werden im Browser angezeigt, alles andere ausschließlich als Download — ein archivierter Anhang darf nie als HTML im Kontext der Anwendung laufen.

Originaldatei herunterladen

Die unveränderte `.em1`, lesbar mit jedem Mailprogramm. Das ist der Nachweis für Betriebsprüfung oder Gericht. Jeder Abruf steht im Protokoll.

Export als ZIP

Originaldateien eines Zeitraums mit beigelegtem Prüfprotokoll, das die Rechenvorschrift der Hash-Kette und das Prüfergebnis je Eintrag nennt. Für den Datenträgerzugriff einer Prüfung.

Kennzahlen

Gesamtzahl, Ein- und Ausgang, Speicherbedarf, ältester und neuester Eintrag, Verteilung nach Jahr, Kategorie und Konto, Zahl der löschbaren und der nicht zugeordneten Einträge.

6. Automatischer Betrieb

Ein Cronjob für alles

Auf gewöhnlichem Webspaces läuft PHP nur, wenn eine Adresse aufgerufen wird — die Anwendung kann sich nicht selbst wecken. Den Anstoß gibt deshalb ein Cronjob beim Hoster oder ein externer Aufrufdienst, und zwar **ein einziger**: `cron.php`, alle 15 Minuten, abgesichert über ein Token.

Was dabei geschieht, entscheidet die Anwendung nach ihrem Zeitplan — Abruf nach eingestelltem Intervall, Sicherung und Kontrolllauf einmal nachts. Damit liegt der Takt in der Oberfläche und nicht beim Hoster, und ein Auftrag genügt auch dann noch, wenn Aufgaben dazukommen.

Sicherung und Kontrolllauf lesen beide den gesamten Bestand; sie laufen deshalb nie im selben Aufruf. Die Sicherung hat Vorrang, die Kontrolle folgt im nächsten Takt.

Die Endpunkte bleiben einzeln erreichbar (`cron_fetch.php`, `cron_backup.php`, `cron_verify.php`) — für eine sofortige Ausführung von Hand und für Installationen, die den Takt lieber beim Hoster festlegen. Eine Sperrdatei

verhindert parallele Läufe. Ein Postfach mit abgelaufenem Passwort meldet den Fehler, hält die übrigen aber nicht auf.

Wächter über den Abruf

Genau daraus folgt eine Lücke: Wird der Cronjob abgeschaltet, das Token geändert oder der Tarif gewechselt, läuft alles weiter wie bisher, nur ohne neue E-Mails. Das Archiv wird still leer.

Deshalb hält jeder Cron-Lauf seinen Zeitpunkt fest, und unter *Einstellungen* → *Automatischer Abruf* steht das erwartete Intervall. Bleibt der Abruf länger als das Dreifache davon aus, meldet es die Startseite, der Kontrolllauf schreibt **ABRUF-AUSGEFALLEN** ins Protokoll und antwortet mit einem Fehlerstatus. Ein einzelner ausgefallener Lauf bleibt still — beim Hoster ist das Alltag, erst mehrere hintereinander sind ein Defekt.

Kontrolllauf

Einmal täglich prüft die Anwendung die Integrität des gesamten Bestands, protokolliert das Ergebnis, setzt den Kettenanker, erneuert die Lizenz und sieht nach dem Abruf-Wächter. Findet sie Abweichungen, antwortet der Cronjob mit einem Fehlerstatus — damit ein externer Aufrufdienst von selbst Alarm schlägt, statt einen stillen Erfolg zu melden.

Gemeldet wird dabei der **neue** Befund, nicht der Dauerzustand: Beim ersten Auftreten ein Fehlerstatus, danach wieder ein Erfolg, dessen Text den Befund weiterhin nennt, und einmal täglich erneut ein Fehlerstatus. Der Grund ist handfest — Aufrufdienste schalten einen Auftrag ab, wenn er oft genug hintereinander fehlschlägt. Ein Postfach mit abgelaufenem Passwort hätte damit binnen Stunden den ganzen Betrieb stillgelegt: kein Abruf, keine Sicherung, keine Kontrolle. Der Wächter hätte den Ausfall verursacht, den er verhindern soll.

Aus demselben Grund wird eine gescheiterte Sicherung oder Kontrolle nicht sofort wiederholt, sondern frühestens nach einer Stunde — sonst liefe bei jedem Takt ein weiterer Vollscan über den gesamten Bestand.

Das erfüllt die Anforderung an durchgeführte Kontrollen: Eine Prüffunktion, die niemand aufruft, ist kein internes Kontrollsystem.

7. Protokoll

Jeder archivrelevante Vorgang wird mit Zeitpunkt (auf Mikrosekunden), Benutzer und IP-Adresse festgehalten — Aufnahme, Öffnen, Anhang, Export, Zuordnung, Prüfung, Löschung, Kettenneuaufbau, Kettenanker, Löschsperre, Anmeldungen, Benutzerverwaltung, Abrufstörungen.

Doppelt geführt: in der Datenbank (auswertbar) und als Textdatei (lesbar, auch wenn die Datenbank ausfällt).

Wiederholungen werden bewusst vermieden. Ein dauerhaft nicht erreichbares Postfach erzeugt einen Eintrag beim Eintritt der Störung und einen bei ihrem Ende, nicht hundert gleichlautende am Tag — sonst wären die Vorgänge, um die es geht, nicht mehr zu finden. Ergebnislose Abrufe erzeugen keinen Eintrag; dass sie stattfanden, steht am Konto.

Dieselbe Regel gilt für den Alarm des automatischen Betriebs: **BEFUND-GEMELDET** beim Eintritt, **BEFUND-BEHOBEN** am Ende.

8. Zugriff und Sicherheit

Anmeldung

Jeder Zugriff erfordert eine Anmeldung, auch der lesende. Die Prüfung steht an **einer** Stelle vor der Verteilung der Aktionen, nicht in jeder einzelnen — eine später ergänzte Funktion ist damit automatisch geschützt und kann nicht durch eine vergessene Prüfung offen bleiben.

Rollen

`admin` darf alles. `pruefer` darf lesen, suchen, prüfen und zuordnen — gedacht für Steuerberater und Betriebsprüfung. Löschung, Kettenneuaufbau, Einstellungen und Benutzerverwaltung bleiben dem Administrator.

Benutzerverwaltung

Anlegen, sperren, freigeben, löschen, eigenes Passwort ändern. Das eigene Konto lässt sich löschen — gedacht für die Übergabe an einen Kunden, bei der sich der Einrichtende am Ende selbst entfernt. Abgewiesen wird nur der letzte aktive Administrator, denn ohne ihn wäre das Archiv nicht mehr verwaltbar. Die Protokolleinträge eines gelöschten Benutzers bleiben bestehen.

Anmeldeschutz

Passwörter als bcrypt-Hash. Sperre nach mehreren Fehlversuchen, gezählt je Benutzername **und** IP-Adresse — so kann niemand fremde Konten aussperren, indem er absichtlich falsche Passwörter schickt. Auch bei unbekanntem Benutzernamen wird ein Hash geprüft, damit die Antwortzeit nicht verrät, welche Konten existieren.

Sitzung

Cookie mit `HttpOnly`, `Secure` und `SameSite=Strict`, neue Sitzungs-ID bei der Anmeldung, Verfall nach Untätigkeit.

Zugangsdaten

Postfach-Passwörter liegen mit AES-256-GCM verschlüsselt in der Datenbank und verlassen den Server nie im Klartext; in der Konfigurationsdatei steht keines. Die Konfiguration selbst gehört eine Ebene über die Dokumentwurzel, wo sie über HTTP nicht erreichbar ist — unabhängig von `.htaccess`.

Verteidigung in der Tiefe

Archivbestand, Programmbibliothek und Konfiguration sind per `.htaccess` gesperrt. Zusätzlich weist jedes Modul einen direkten Aufruf mit 403 ab, falls die Serverkonfiguration einmal nicht greift. HTTPS wird erzwungen, Sicherheits-Header sind gesetzt, Archivinhalte werden nicht zwischengespeichert.

9. Betrieb und Übernahme

Einrichtung

`setup.php` führt durch die gesamte Installation: Umgebung prüfen, Datenbankzugang abfragen und testen, Konfiguration schreiben, Anwendungsschlüssel und Cron-Token erzeugen, Schema anlegen, Administratorkonto erstellen. **Es muss keine PHP-Datei bearbeitet werden.** Darf PHP nirgends schreiben, zeigt die Seite den fertigen Dateiinhalt zum Hochladen an.

Die Schemaanlage ist wiederholbar und ergänzt bei einem Update fehlende Spalten und Tabellen. Nach dem Aufruf gehört die Datei vom Server — sie schreibt Zugangsdaten.

Mehrere Mandanten

Mehrere Installationen können als Unterverzeichnisse derselben Domain nebeneinander laufen, jede mit eigener Konfiguration, eigenem Archiv, eigenen Benutzern und eigener Lizenz. Konfigurationsdatei und Sitzungsname leiten sich dafür vom Verzeichnis ab; die Datenbanken trennt entweder ein eigener Zugang oder ein eigenes Tabellen-Prefix. Eine mandantenübergreifende Verwaltung gibt es bewusst nicht — siehe MANDANTEN.md.

Reparaturfunktionen

Anhangsnamen nachtragen liest Einträge ohne Anhangsnamen erneut aus der Originaldatei und ergänzt den Suchindex; die Datei wird nur gelesen, Hash und Kette bleiben unberührt.

Zeitstempel neu einlesen bestimmt das Datum jeder E-Mail erneut aus der Originaldatei — auf demselben Weg wie bei der Aufnahme, mit Umrechnung in die Zeitzone der Anwendung und Abgleich gegen die `Received:` - Zeile. Nötig für Bestand aus der Zeit vor dieser Rechnung und nach einem Wechsel der Zeitzone. Das Löschedatum wird gleich mit neu bestimmt; Hash, Kette und Reihenfolge bleiben unberührt.

Werkzeuge für die Datenübernahme

`tools/adopt.php` übernimmt einen vorhandenen Bestand samt Zuordnungen, `tools/import.php` liest `.eml` - Dateien neu ein. Beide laufen nur über die Kommandozeile — auf Shared Hosting ohne Shell also nicht verwendbar.

Datensicherung

Zum Archiv gehören zwei Teile, die nur gemeinsam vollständig sind: die Originaldateien und die Datenbanktabellen mit Index, Kette und Protokoll. Eine Sicherung nur eines Teils macht die Integritätsprüfung wertlos.

`cron_backup.php` schreibt täglich einen SQL-Abzug nach `data/backup/` und räumt alte Stände auf. Damit erwischt ein einfacher Dateiabzug per FTP beide Teile — vorausgesetzt, er läuft **nach** dem Datenbankabzug. Die Anwendung schreibt jede `.eml` vor ihrem Indexeintrag; in umgekehrter Reihenfolge entstünden Indexeinträge ohne Datei, und die Prüfung meldete dauerhaft „Datei fehlt“. Der Kontrolllauf nennt das Alter der letzten Sicherung.

Lizenz

Die Installation aktiviert sich einmalig mit einem Schlüssel gegen den Lizenzserver und erhält ein signiertes Token, das sie danach selbst prüft — ohne Netzverkehr im laufenden Betrieb. Erneuert wird es täglich vom Kontrolllauf; fällt der Lizenzserver aus, läuft alles wochenlang weiter.

Ohne gültige Lizenz ruht die **Neuaufnahme**: kein Abruf, keine Zuordnung, keine Wartung. Suche, Anzeige, EML-Download, ZIP-Export, Integritätsprüfung und Protokoll bleiben erreichbar. Das ist bewusst so: Der Bestand ist der gesetzliche Nachweis des Betreibers (§ 147 AO, § 257 HGB) und darf nicht an einer Vertragsfrage hängen. Einzelheiten in `LIZENZ.md`.

Was das Programm nicht leistet

Ehrlichkeit gehört zur Verfahrensdokumentation; die vollständige Liste steht in `GOBD.md` unter *Grenzen*. Die wichtigsten Punkte:

- **Kein qualifizierter Zeitstempel.** Der Kettenanker ist ein Indiz mit fremdem Empfangszeitstempel, kein Zeitstempel eines Vertrauensdienstes nach eIDAS.
- **Keine Zertifizierung.** Kein Testat nach IDW PS 880. Die GoBD verlangen es nicht, ein Prüfer kann es erwarten.
- **Vollständigkeit hängt am Postfach.** Archiviert wird, was beim Abruf im Postfach liegt. Wer Vollständigkeit ab einem Stichtag zusichern will, lässt beim Mailprovider eine Kopie jeder ein- und ausgehenden Nachricht in ein separates Archivpostfach legen, auf das die Anwender keinen Zugriff haben.
- **Verschlüsselte Mails bleiben verschlüsselt.** S/MIME und PGP werden im Original aufbewahrt, sind im Archiv aber nicht im Klartext durchsuchbar.

