FOCUS FAMILY

Findings and Remediation Plan

Technical annex to the ISO Conformity Profile

Software Notion · Technical annex · Product state 1.3.0 · Assessed 1 Nov 2026

Contents

1. **Purpose and audience**
2. **Scope and method**
    1. Components assessed
    2. Procedure
    3. Test evidence
    4. Not assessed
3. **Summary of findings**
4. **Verified protection mechanisms in detail**
    1. Protection of the audit log
    2. Key management and rotation
    3. Local deletion chain and deletion orders
    4. Reporting function and data protection guard
    5. Sign-in, permission ceiling and revocation
    6. Release and privacy boundaries of the interfaces
5. **Findings in detail**
    1. F-02 · No supplied alerting, application logs without forwarding
    2. F-05 · Two-factor authentication off by default
    3. F-06 · Refresh token valid for 365 days
    4. F-07 · Screen protection on iOS and desktop only limited
    5. F-08 · Container hardening incomplete
    6. F-13 · Accessibility not tested with assistive technology
    7. F-20 · Dictation with cloud fallback of speech recognition
    8. F-23 · Administration session idle for 180 minutes
    9. F-24 · Desktop key storage without a system keyring
    10. F-27 · Server information retrievable without sign-in
    11. F-28 · Passkey accounts discoverable through the sign-in redirect
    12. F-30 · Integration check scripts outdated
    13. F-32 · Connection pool without back-pressure, SQL logging in production
    14. F-33 · XTA 2 and FIT-Connect only tested against specification mocks
6. **Remediated findings**
    1. F-01 · Retention periods for the change feed and deletion markers
    2. F-03 · Third-party inventory and bill of materials
    3. F-04 · Static code analysis and image scanning
    4. F-11 · Clock synchronisation as an operating requirement
    5. F-12 · Requirements specification part of the project
    6. F-14 · Machine-readable interface description
    7. F-15 · Load and capacity evidence
    8. F-16 · Pagination in the log view
    9. F-17 · Backoff in the synchronisation client
    10. F-18 · Measured test coverage
    11. F-19 · Licence notice

12. F-21 · Daily overview for Watch and widgets
13. F-22 · Licence server addresses in the deployment template
14. F-25 · Self-service deletion of the pseudonymous licence identity
15. F-26 · Secure attribute and encrypted session key material
16. F-29 · Storage quota credit
17. F-31 · Unmaintained Docker CLI in the server and updater images

7. **Operator decisions**
8. **Remediation plan**
9. **Coverage of ISO/IEC 27001 Annex A**
    1. A.5 Organisational controls
    2. A.6 People controls
    3. A.7 Physical controls
    4. A.8 Technological controls
10. **Coverage of ISO/IEC 27034**
11. **Coverage of ISO/IEC 25010**
12. **Coverage of ISO/IEC 29147 and 30111**
13. **Remaining strengths in summary**
14. **Validity**

Purpose and audience

This document is the technical annex to the ISO Conformity Profile. It is addressed to the **IT administration, the information security officer and the data protection officer** of an operating organisation, as well as to external assessors.

While the conformity profile describes *what* the software provides, this document states *exactly where* deviations exist, *how they take effect* and *what is required to remedy them*. It deliberately also contains findings that are uncomfortable for the vendor.

The assessment is based on a review of the source code, not of the product documentation. Every security-relevant statement in the documentation was held against the implementation; where the two diverge, that is recorded as a finding.

Scope and method

Components assessed

Component	Scope
Mobile application	.NET MAUI, iOS/iPadOS and Android
Desktop application	Avalonia, macOS and Windows
Companion app	watchOS
Organisation server	ASP.NET Core 10, PostgreSQL 17
Administration interface	Blazor, server-side rendered
Sign-in through identity provider	OpenID Connect and SAML 2.0, emergency access for local system administration
Interfaces to third-party systems	REST/JSON, webhooks, CMIS 1.1 (client and repository), XÖV/xdomea over directory, SFTP, XTA 2 and FIT-Connect, SCIM 2.0, iCalendar, SIEM forwarding via syslog and CEF
Licence server	separate service
Deployment	Docker Compose, reverse proxy templates, backup scripts
Documentation	more than 200 documents, including security, operations, permission and data protection documentation

Size of the code base: 207,796 lines of production code, 130,159 lines of test code in 612 test files, 17 production and 7 test projects, 141 user interface definitions (counted as C# files excluding build outputs and generated files, and XAML/AXAML files; the same basis as in the profile).

Procedure

1. **Structural survey** – project layout, dependencies, deployment artefacts.
2. **Documentation review** – security documentation, threat model, authorisation and protocol descriptions, incident runbooks, backup procedures.
3. **Verification against the code** – every security-relevant statement in the documentation was checked against the actual implementation rather than taken at face value.

4. **Configuration review** – container, compose definitions, reverse proxy, application and build settings, build pipeline.
5. **Test run** – execution of all seven test suites to confirm the current state, the server and licensing suites against real PostgreSQL.
6. **Standards mapping** – allocation of findings to ISO/IEC 27001 Annex A, 27034, 25010 and 29147/30111.

Test evidence

Metric	Value
Tests executed (all seven projects)	6,061
Passed	6,047
Skipped	14 (environment-dependent tests: live-server cases in the sync client; keychain, signature and live-server cases on the desktop; on the server one live test against a real identity provider and two FIT-Connect revocation tests that do not run on macOS)
Failed	0
Breakdown	Core 1,506, Infrastructure 796, Sync client 149 (6 skipped), Desktop 577 (5 skipped), local AI 137, Licence server 645, Server 2,251 (3 skipped)
Test coverage	88.0 % of lines (76,000 of 86,362), 68.1 % of branches (21,397 of 31,439) (`tools/coverage.sh` across all seven suites)
Further checks	Static security analysis (`tools/sast.sh`): 0 findings, reviewed exceptions justified at the code (`[SuppressMessage]` , 12 places), allowlist empty; image scan of the three shipped images (admin panel amd64 and arm64, licensing server) with Trivy 0.75.0: no fixable critical or high findings, exception list empty, runtime images on `aspnet` 10.0.14 and `sdk` 10.0.403 with the security updates of 25 Oct 2026; secret scan with self-test passed; integration check scripts: all nine fail at sign-in (F-30)
Execution	One complete run of all seven test suites on 1 Nov 2026; the server and licensing suites ran against two PostgreSQL 17 instances, including the upgrade test from a version 1.0.0 database. The interface tests ran against an Alfresco test system for CMIS and a test server for SFTP; XTA 2 and FIT-Connect were tested against our own simulations of the specification (F-33)
Assessment	All suites green; migration, privilege, chain, rotation, concurrency and upgrade tests ran against real PostgreSQL and were not skipped. A single non-reproduced failure of a test on revoking a case share (`SharingAssignRightTests`), observed once, did not occur in this run and remains an observation point

Not assessed

Runtime behaviour against a production installation, the mobile applications executing on physical devices, the licence server in detail, the operating environment of any organisation, the connection to a production identity provider, and effectiveness testing through attack simulation (penetration testing), transfers over XTA 2 and FIT-Connect to real counterparts (F-33), and the interfaces in interaction with an organisation's production third-party systems. The device logic of trip recording (location sources, timers, automatic end) and of the Apple Watch controls is verified through builds and component tests; a test on real devices is pending.

Summary of findings

Fourteen open findings: none of high, four of medium and ten of low criticality. Sixteen findings are remediated in the state of version 1.3.0 (F-01, F-03, F-04, F-11, F-12, F-14 to F-19, F-21, F-25, F-26, F-29, F-31); the section

"Remediated findings" describes the implemented mechanism and the test evidence for each, as well as number F-22, remediated in earlier assessment rounds. F-09 and F-10 are not deviations but operator decisions with a secure default (section "Operator decisions"). F-07 is partially remediated and narrowed to the remaining part: screen protection is centrally enforceable but only prevents screenshots on Android. F-30 is open: the integration check scripts currently fail at sign-in already. Several points identified in earlier assessment rounds are described in the section "Verified protection mechanisms" as existing mechanisms (local deletion chain, opt-out for crash reports, privacy declarations of the store manifests, sign-in and revocation). The numbers of remediated and reclassified findings stay reserved so that references remain stable.

No.	Finding	Criticality	Standard
F-02	No supplied alerting, application logs without forwarding	**medium**	27001 A.8.15, A.8.16
F-05	Two-factor authentication off by default	**medium**	27001 A.8.5
F-06	Refresh token valid for 365 days	**low**	27001 A.5.17
F-07	Screen protection on iOS and desktop only limited	**low**	27001 A.8.12, A.8.1
F-08	Container hardening incomplete	**low**	27001 A.8.9
F-13	Accessibility not tested with assistive technology	**medium**	25010, EN 301 549
F-20	Dictation with cloud fallback of speech recognition	**medium**	27001 A.5.14, A.5.19; GDPR Art. 28
F-23	Administration session idle for 180 minutes	**low**	27001 A.8.5, A.5.17
F-24	Desktop key storage without a system keyring	**low**	27001 A.8.24
F-27	Server information retrievable without sign-in	**low**	27001 A.8.9
F-28	Passkey accounts discoverable through the sign-in redirect	**low**	27001 A.8.5
F-30	Integration check scripts outdated	**low**	27001 A.8.29
F-32	Connection pool without back-pressure, SQL logging in production	**low**	25010 performance efficiency; 27001 A.8.6
F-33	XTA 2 and FIT-Connect only tested against specification mocks	**low**	25010 compatibility; 27001 A.8.29

Verified protection mechanisms in detail

Six mechanisms are described more closely below, because they carry the assessment of several controls and because the effectiveness of each is backed by its own test.

Protection of the audit log

Two independent layers, because either alone could be circumvented:

1. **Privilege revocation at the database level.** The application role holds neither UPDATE nor DELETE on `AuditEvents` and `PrivacyRequestEvents`. The revocation sits in the same procedure as the grant and is re-applied on every migration run.
2. **Hash chain.** Every entry carries a SHA-256 checksum over its content **and** over its predecessor's checksum. On PostgreSQL a transaction lock serialises appends, so concurrent writes cannot fork the chain.

Administration verifies the chain in the panel under *Audit → Verify chain*. The check names the first disturbed position with its sequence number and cause (entry altered, removed or reordered), and is itself logged. It covers the whole log, not only the excerpt shown in the interface. Each successful run writes a checkpoint (sequence number and checksum of the last entry) into a table the application may likewise only append to, so truncating the end of the chain afterwards is noticed as well.

A limit we state explicitly: the chain proves the internal consistency of the entries present and detects truncation since the last check. It is **not** an independent safeguard against someone who can bypass the privilege revocation and restore a complete older database state — only an export into storage outside this database protects against that. The retrieval interface for it exists.

Evidence: three tests against real PostgreSQL show that update and delete attempts fail with "insufficient privilege" while ordinary tables remain writable. Nine further tests cover the chain and the panel, among them a check proving for every single logged field that altering it changes the checksum.

Key management and rotation

Every envelope of data at rest carries a key id and is resolved through it:

Surface	Format	Rotation
Local case data on devices	`ffenc2:<keyId>:…`	setting in the app; rewraps the local store in batches
Local attachments (photos, audio, sketches, documents)	header `FFATT2` with key id	in the same run
Open synchronisation conflicts (local and server version in `sync.db3`)	`ffenc2:<key id>:…`	in the same run, before the old key is retired
File attachments on the server	header version 2 with key id	admin action under *System*, logged
Encrypted copies of access requests	as attachments	in the same run

Writing always uses the active key; reading resolves the key id of the respective envelope. On a rotation the new key is added to the ring and persisted **before** anything is rewrapped, so an interrupted run leaves every record readable and a later run continues. Old keys may only be retired once nothing uses them — the application refuses otherwise.

Writes during a run are not lost. For records in the database, one is overwritten only if its content is unchanged since it was read; otherwise it is left alone and picked up by the next run. For files — local attachments as well as server files — the run checks immediately before the replacement whether the file still exists and whether its size and modification time still match what they were when it was read. A file edited in the meantime therefore keeps its newer version, and a file deleted in the meantime is not restored. The same holds on the server for the copies of access requests, so a concurrent redaction or retention purge is not overwritten.

A read error is not interpreted as "nothing left". An envelope whose header cannot be read counts as outstanding; a copy whose key is missing does not abort the run but is reported as skipped. The server reports "complete" only

once no files, no copies and no unreadable envelopes are left — only then may an old key leave the configuration.

A metadata comparison alone would not carry this — a window would remain between the check and the replacement. Rotation, ordinary writes and deletions therefore take **the same per-file lock** for the whole operation: on the devices every write and delete path for attachments goes through the same place, and on the server file storage and rotation share one coordinator. The metadata comparison stays as a second barrier against writers outside the application.

Open synchronisation conflicts are stored in the synchronisation database encrypted with the same data key as the rest of the data set. Legacy plaintext rows are encrypted on open, after which the file is cleaned with `secure_delete` and `VACUUM`; if the cleanup fails, it is retried on the next open and after a reset. A conflict that can no longer be decrypted stays visible and blocks sending the record. If the server version is taken for it, that decision is bound to the local state at the time of the decision: a later local change raises a new conflict instead of being overwritten, and unsent or private content of a case that is missing or deleted on the server is kept. Removing the conflict, dropping pending operations and saving the synchronisation state happen in one transaction.

Three further assurances concern concurrency:

- **On the devices, files:** writes to attachments take a usage marker for their duration. Retiring old keys waits until no marker is outstanding, and after a waiting period it aborts rather than remove a key that a running write still needs.
- **On the devices, database:** the same marker covers a row from encryption through persistence. A record prepared under the old key therefore cannot be stored after that key was retired. The write key is moreover determined only inside the acquired permission, so even long-held references to the encryption service can no longer use a retired key. Markers are re-entrant per flow so that nested writes do not block — but a nested call receives its marker only while the outer call actually holds the permission, so it cannot slip past a running key retirement. If the previous marker has already finished or been aborted, a new outer marker is created and published in the caller's own flow; two calls in the same flow that are not awaited count as independent writers. Neither can a re-entry land on a dead marker — which would otherwise show up as mutual waiting with the key retirement — nor can a sibling call slip past it.
- **On the devices, rotation run:** the key change, the rewrapping and the key retirement run under one shared lock. A second job is refused rather than run in parallel — including when the user closed and reopened the settings window in between. No run can therefore write back a key ring derived from an older state and remove a key another run is currently using.
- **On the server:** a rotation run holds a database lock for its duration. If a second instance starts the same operation, it does not work in parallel but reports to the panel that a rotation is already running. Work files additionally carry the id of their run; leftovers are removed at the beginning of the next run, but only once they are twelve hours old, so that a concurrently running operation loses nothing.

Evidence: sixty-eight tests, among them: after retiring an old key exactly the envelope written with it becomes unreadable; an attachment stays readable after rotation and a restart of the services; a record changed in the meantime is not overwritten; a deletion that starts while a file is being rewrapped cannot resurrect that file; a file changed or deleted behind the application's back is not overwritten, respectively not brought back; a key is retired neither while a file write is outstanding nor while a database write is outstanding; a second write in the same flow cannot slip past a held key retirement while a genuinely nested write — including after an earlier finished or aborted marker — is granted without waiting; a caller that gives up waiting takes away neither the next caller's lock nor its permission; a second rotation job is refused and a rotation start during a key retirement waits, so no stale key ring is written back; a protector obtained earlier writes with the new key after the retirement; an unreadable or truncated envelope — including one in a privacy copy — prevents the "complete" result; and a second run against real PostgreSQL reports the held lock instead of writing into the same store. For each of these assurances it was additionally verified that the test fails without the respective safeguard.

For the synchronisation conflicts, 19 tests (`SyncConflictEncryptionTests`, `SyncConflictKeyLifecycleTests`) show that conflict content never reaches the synchronisation database in plaintext, legacy rows are encrypted and remnants removed, rotation rewraps open conflicts before the old key is retired, a conflict without its key stays blocked and is never pushed, and unsent or private case content survives taking the server version.

Not evidenced: a run against a production-sized data set. Expect a corresponding runtime with large attachment stores and perform the rotation in a maintenance window.

Local deletion chain and deletion orders

Deleting a case on a device is a coordinated sequence in four steps:

1. **Inventory.** All files of the case (photos with thumbnails, audio, sketches with drawing data, generated documents, signature images on time entries, journal entries and documents, generic attachments) are derived from the records. Files still referenced by another entry are kept.
2. **Durable orders.** Every deletion order is written as its own file **before** any record is removed. An abort between record and file deletion therefore leaves a resumable state; the next start processes open orders and re-checks whether the file has become referenced again.
3. **Operation lock.** Enqueuing, record deletion and processing run under a shared lock, so a parallel clean-up run cannot discard an order that has not been released yet. Deletions arriving through synchronisation (tombstones, access revocation) use the same chain.
4. **Honest reporting.** "Missing", "failed" and "unreadable" are counted separately; corrupted orders are quarantined instead of deleted. A deletion counts as complete only without open orders.

"Delete all data" additionally pauses background synchronisation, waits for running write and rotation operations, writes a durable full-wipe order and only then removes database and files. If the wait times out, nothing has been deleted.

Evidence: Fourteen tests cover the cascade, reference check, tombstones, resumption after a failed file deletion, interleaved clean-up runs, quarantine of unreadable orders and the full-wipe order; two more prove that background work is paused.

Reporting function and data protection guard

The report is composed **locally** and shown in full before it can leave the device. There is no silent transmission; passing it on is always a deliberate act (save as a file, copy, open a mail window).

The content is limited to a fixed field list: category, severity, affected area, title, description, steps, an optional reply address, plus product version, build, platform, OS version, device model, interface language and — in the admin panel only — server version and instance id. A reflection test fails as soon as a field is added that is not on that list.

The free text passes two stages:

1. **Redaction** of e-mail addresses, phone numbers, IBANs, dates, long digit runs, user paths and credentials inside URLs.
2. **Hard block**: if the text names a person, alias, address or case label held in the installation, **no report is produced**. Matching is accent- and case-insensitive at word boundaries, so "Meier" matches but "Meierei" does not.

The guard is **fail-closed**: if the personal-data check cannot be carried out, no report is produced. The finished text is checked a second time before output.

Limits we state explicitly: the check is a term comparison, not a guarantee. Names shorter than three characters are not matched, because they would otherwise hit ordinary words. Case-related free text that names nobody held

in the installation passes the filter. The report therefore does **not** claim to be free of case data; it describes what was checked and asks the reader to read it again before sending. The decision stays with the person passing it on.

Evidence: 22 tests on the checking logic and 7 through the panel's real HTTP interface, among them: a report naming a person from the installation is refused and recorded as a failed attempt; another account cannot retrieve a foreign draft.

Sign-in, permission ceiling and revocation

Sign-in through the identity provider. Sign-in can be delegated to the organisation's identity provider (OpenID Connect or SAML 2.0); the server continues to issue its own tokens afterwards. An account is only linked by e-mail address if the provider verifies it (OpenID Connect `email_verified`, tenant-bound claims for Microsoft Entra ID, SAML only after explicit opt-in); otherwise administration approves the first sign-in once for 24 hours. Each account has at most one link. Re-authentication only counts through an explicitly repeated sign-in at the provider no older than five minutes. When administration switches on **Enforce SSO**, local sessions and tokens are revoked at once, addresses in the SSO domains are always routed to the provider, and the apps only offer sign-in through the organisation. Local system administration accounts keep a password as emergency access – to the admin panel only; the last such account cannot be switched. The strong authentication requirement (passkey, MFA or SSO) is evaluated under the same rule for the panel and for the device check of synchronisation.

Uniform sign-in answers. Unknown, locked and password-less accounts pass through the same password check as an existing account, so neither the answer nor the computing time reveals whether a password account exists (exception: F-28). Account, MFA and emergency access endpoints are rate-limited and lock after failed attempts; MFA enrolment does not return an already active secret.

Permission ceiling. Every account change in the panel – edit, deactivate, password, MFA, passkeys, setup QR code, sign-in method – is refused if the target account holds permissions outside one's own grant scope. Administration limited to areas only changes accounts whose areas and area-bound teams lie entirely within its own scope, and never grants organisation-wide operational administration or area management.

Immediate revocation. Deactivation revokes sessions and tokens; synchronisation rejects deactivated accounts and stale security stamps at once and requires a device ID. All administrative actions are protected against request forgery. The Content Security Policy only permits the own server, the licence server and the registered app callback addresses as form targets; script-capable schemes are never admitted.

Evidence: a regression test sends a request without forgery protection to every administrative action and expects it to be refused; tests count the password checks for unknown, locked and password-less accounts and fail without the equalisation; further tests cover the permission ceiling, area boundaries, enforced SSO, the shared MFA rule for panel and synchronisation, the rejection of deactivated accounts in synchronisation and the permitted form targets (`AppCallbackFormActionTests`). A live test against a real identity provider (Keycloak) exists and runs only with a provided test environment; it was skipped in the run for this assessment.

Release and privacy boundaries of the interfaces

Four-eyes release. API clients and connections are created as drafts. Data flows containing case data become active only when a person holding a data protection permission releases them with a local account and a fresh sign-in confirmation, and that person neither created nor last changed the interface. The release is bound to a server-computed revision of the reviewed state; if the interface was changed after the page was opened, the release is refused. Any change to data scope, target, privacy settings or inbound direction resets the interface to "awaiting release". Opening beyond the data-minimising presets (pseudonymised, final documents only, notes and confidential document types blocked) is reserved to the data protection role; technical administration may do so only while mandatory release applies. A configuration import contains no secrets, creates drafts only and never widens case access.

Visibility at every outflow. The REST API, the CMIS repository and the change feed check every record against its current state: private records, records with restricted processing or marked for deletion, and documents blocked by the document filters do not appear. Webhooks carry only identifiers and links and are checked again against area and privacy settings before every delivery attempt, including retries. Every successful delivery relating to a case writes an entry to the disclosure log with a snapshot of the recipient (name, type, purpose, legal basis); the subject access package lists these recipients.

Transport and secrets. Before sending, every outbound request refuses addresses without HTTPS (unless explicitly switched on) and addresses with embedded credentials, follows no redirects and checks the address actually connected to against SSRF. Stored credentials do not follow a changed target but must be entered again; OAuth tokens are never shared between connections. The CMIS client sends credentials only to the origin of the configured address. FIT-Connect encrypts only once the destination's key chains up to the public administration PKI and is not revoked.

Inbound data and erasure. Delivered files pass the same checks as attachments on receipt (type, size, file signature, virus scan); xdomea packages are unpacked within limits on scope and bytes read, XML is read without DTD processing. Only holders of the permission *Review and accept incoming data from interfaces* handle inbound items, and only for cases within their area; acceptance is atomic. Final erasure of a case or record clears the content of the related inbox entries and removes external references and the storage paths in the delivery log. SCIM manages only practitioner accounts without administrative permissions, never changes permissions, and reconciles group memberships transactionally and locked against concurrent changes.

Limits we name explicitly: Focus Family does not erase data already transferred to target systems; the delivery log names the recipients. Private networks are deliberately allowed as targets (DMS on the intranet), so a holder of *Manage interfaces* can address internal hosts, and with mandatory release switched off the second review of new data flows is dropped. Threats T22 to T28 of the threat model describe these residual risks in detail.

Evidence: 26 test classes in the server suite, among them `IntegrationPrivacyBoundaryTests` (four-eyes release against the creating and the last changing person, binding to the revision, erasure of interface copies), `IntegrationTransportGuardTests` and `CmisClientGuardTests` (HTTPS guard, SSRF, origin binding), `IntegrationDisclosureTests` (disclosure log and access package), `IntegrationLicenseGateTests`, `IntegrationHardeningTests` as well as `ScimConcurrencyTests` and `IntegrationInboundConcurrencyTests` against real PostgreSQL.

Findings in detail

F-02 · No supplied alerting, application logs without forwarding

Criticality: medium · **Standard:** ISO/IEC 27001 A.8.15, A.8.16 (monitoring activities)

Situation. The audit log can be forwarded to a SIEM as an interface (syslog per RFC 5424 over TLS per RFC 5425, or CEF, with an event filter, identifiers and chain checksum only, no case data in clear text); if forwarding fails repeatedly it pauses visibly without losing entries. The **application logs**, by contrast, are written to standard output. There is no forwarding, no retention period and no access protection for them; when the container is removed they are gone.

The **data sources**, by contrast, are complete. The server and the licence server each expose a monitoring interface in Prometheus format (`/metrics`) and the same numbers as JSON (`/monitoring/status`), ready for Grafana, Uptime Kuma and comparable tools. The scope covers the security-relevant signals: rejected actions in the last 24 hours, open emergency accesses, devices waiting for approval, disk and attachment usage, the age of the last audit-chain verification, unreadable encryption envelopes and **overdue data-subject requests**. Technical administration enables the interface in the admin area under *System → Monitoring* (its own permission *Configure*

monitoring), creates the access key there and can restrict access to particular address ranges. Without a key it answers "not found" rather than serving unprotected; it delivers numbers only, no names and no case content. If a single measurement fails, the interface says so explicitly (`degraded`, with the group named) instead of passing an incomplete collection off as unremarkable — a failure of the monitoring itself therefore stays visible.

Impact. What is missing is the **evaluating and alerting layer** and forwarding of the application logs. A.8.16 requires that anomalous behaviour be detected **and acted upon**. An attack attempt is currently logged and can be queried, but reaches nobody on its own. An organisation that switches on neither the monitoring interface nor SIEM forwarding has no monitoring.

Remediation. Two points remain open. First, supplied alerting rules for the obvious signals, so that every operating organisation need not phrase them itself. Second, a documented path for forwarding the application logs to a central system with a retention period and access protection. The metrics required are already available, including those of the interfaces; the technical manual lists which ones are worth alerting on.

F-05 · Two-factor authentication off by default

Criticality: medium · **Standard:** ISO/IEC 27001 A.8.5 (secure authentication), A.8.2, A.8.9

Situation. Defaults of the organisation policy:

Setting	Default	Assessment
App lock mandatory	on	appropriate
Four-eyes principle for access requests	on	appropriate
Break-glass enabled	off	appropriate
Two-factor authentication mandatory	**off**	**not appropriate**
Device approval by administration	off	defensible

Impact. The application processes social services data about families and minors – special categories of personal data under Art. 9 GDPR. A freshly provisioned instance permits sign-in without a second factor until an administrator actively changes the policy. All technical prerequisites exist and are mature: time-based one-time passwords with recovery codes, and passkeys with enforced user verification and a resident key.

Mitigating factor: credentials alone are in any case insufficient for interface access – every request must additionally pass the device gate. For browser sign-in to the administration interface, however, that additional barrier does not apply.

Remediation. Change the default to "mandatory"; alternatively enforce a deliberate decision during commissioning instead of a silent default.

F-06 · Refresh token valid for 365 days

Criticality: low · **Standard:** ISO/IEC 27001 A.5.17, A.8.5

Access tokens are valid for one hour, refresh tokens for one year. Understandable for an offline-capable application, but a long window of attack. Mitigated by binding the token to the registered device, rolling renewal with a short overlap, and server-side revocation checking. **Remediation:** tie validity to the configured maximum offline duration (default seven days) rather than a fixed year.

F-07 · Screen protection on iOS and desktop only limited

Criticality: low · **Standard:** ISO/IEC 27001 A.8.12, A.8.1

Partially remediated and narrowed to the remaining part. **Implemented** is central enforceability: the managed setting `ScreenProtectionEnforced` can be set in the admin panel for every scope and is delivered to connected devices. The mobile app then turns on "Hide content in app switcher" and locks the switch while the device is bound to the server. The actual protective effect, however, depends on the platform:

Platform	Screenshots and screen recordings	Preview in the app switcher
Android	prevented (`FLAG_SECURE`)	hidden (`FLAG_SECURE`)
iOS and iPadOS	**not prevented**; the app does not detect recordings either	hidden (blur layer when moving to the background)
Desktop (macOS, Windows, Linux)	**not prevented**; the setting is not evaluated	**not hidden** by the setting; only with an app lock set up does the app cover its content once no window is active

What the tests prove. `ScreenProtectionPolicyTests` (core) verify the evaluation of the setting: normalisation of values, locking of the user switch only while bound, behaviour when the enforcement is removed. `ScreenProtectionManagedSettingTests` (server) verify the catalogue, the input validation in the panel, delivery of the most specific value and the labels in all languages. The effect at operating-system level (setting `FLAG_SECURE`, blur layer on iOS) is not covered by an automated test; it was verified in the code.

Residual risk. On iPhone, iPad and the desktop, screen content relating to cases can be captured by screenshot or screen recording even when the organisation enforces screen protection. The hint in the admin panel and the handbooks state this limit.

Remediation. iOS: detect ongoing screen recordings (`UIScreen.isCaptured`) and cover the content meanwhile. Desktop: tie display exclusion (`SetWindowDisplayAffinity` on Windows, `NSWindow.sharingType` on macOS) to the setting. Until then, block screenshots on managed iOS devices with an MDM restriction and include the limit in the staff instructions.

F-08 · Container hardening incomplete

Criticality: low · **Standard:** ISO/IEC 27001 A.8.9, A.8.6

The deployment definition sets neither a read-only root file system, nor the dropping of kernel capabilities, nor the block on privilege escalation, nor limits for memory and CPU. Without resource limits, a single container can impair the host.

Well solved and explicitly to be acknowledged: execution without system privileges, health check, all durable data on named volumes, database and virus scanner not reachable externally, application bound to the loopback interface only, base images pinned to exact versions.

F-13 · Accessibility not tested with assistive technology

Criticality: medium · **Standard:** ISO/IEC 25010 (interaction capability, inclusivity); EN 301 549; German BITV 2.0; European Accessibility Act

Situation. In place: screen reader labels on **every** symbol-only control (60 in the mobile app, 30 in the desktop application), labelled form fields, a skip-to-content link and a visible keyboard focus in the admin panel, localisation of the user interfaces in seven languages, text scaling following the system setting, light and dark mode. An accessibility statement is enclosed (`docs/ACCESSIBILITY.md`). A regression test fails as soon as a symbol-only control without a label is added.

Not implemented or not evidenced:

1. **No testing with real assistive technology.** Labels exist and are demonstrable in the code, but a complete pass with VoiceOver, TalkBack, NVDA or JAWS is outstanding. A present label does not by itself guarantee a sensible reading order.
2. **Contrast ratios have not been measured.**
3. **Reading order and heading hierarchy** of the mobile views have not been assessed.
4. **Generated PDF documents are not tagged.**
5. **Drawing surfaces** (sketches, handwriting, signatures) require a pointing device; there is no equivalent alternative.

Impact. Directly relevant for public-sector bodies, which are bound by accessibility regulations for the information technology they deploy. Focus Family is **partially conformant** with EN 301 549; the five points above belong in the operator's own statement as non-accessible content. Irrespective of the legal position, practitioners with a visual or motor impairment remain excluded from individual functions.

Remediation. Testing with VoiceOver and TalkBack across the daily workflows, measuring and correcting contrast ratios, heading hierarchy of the mobile views, tagged PDF output, then an external assessment.

F-20 · Dictation with cloud fallback of speech recognition

Criticality: medium · **Standard:** ISO/IEC 27001 A.5.14, A.5.19; GDPR Art. 28

Dictation uses the operating system's speech recognition. On iOS, on-device recognition is only enforced when the device supports it; otherwise Apple processes the recording on its own servers. Android merely sets a preference for offline recognition. Dictated conversation content can therefore reach Apple or Google on devices without local recognition, without the organisation being able to control this. The app privacy policy names the fallback. **Remediation:** disable dictation on devices without local recognition or make it controllable as an organisation policy; until then, show a notice before the first dictation.

F-23 · Administration session idle for 180 minutes

Criticality: low · **Standard:** ISO/IEC 27001 A.8.5, A.5.17

The administration session expires after 180 minutes of inactivity (configurable, sliding). For an interface with access to case data, 15 to 30 minutes are customary. **Remediation:** lower the default to 30 minutes; note in the operations documentation.

F-24 · Desktop key storage without a system keyring

Criticality: low · **Standard:** ISO/IEC 27001 A.8.24

Facts. On Linux the master key lives exclusively in the system Secret Service (libsecret interface, the same path as for the licence secrets). If no Secret Service is reachable, the application shows a notice and exits without creating a key or a database; there is no plaintext fallback (evidence: `DesktopSecureKeyStoreBackendTests`, installation test on Ubuntu 24.04 and Fedora 42 with and without a keyring). On Windows DPAPI protects the key. On macOS the key is passed to the `security` tool as a process argument and is briefly visible to processes of the same user; that is the remaining finding. **Remediation:** storage through the Keychain programming interface.

F-27 · Server information retrievable without sign-in

Criticality: low · **Standard:** ISO/IEC 27001 A.8.9

The interface through which apps learn about the server before signing in returns, without sign-in, the organisation name, server version, instance ID, supported sign-in methods and, with SSO active, the identity provider's display

name. Part of this is needed to set up the apps; version and organisation name, however, ease reconnaissance of a publicly reachable server. **Remediation:** reduce the pre-sign-in output to what setup requires.

F-28 · Passkey accounts discoverable through the sign-in redirect

Criticality: low · **Standard:** ISO/IEC 27001 A.8.5

Sign-in answers are uniform for password accounts. Accounts switched to passkey sign-in, however, are redirected to the passkey page, which reveals that a passkey account exists for an address. The redirect is needed for the apps' passkey flow and is therefore kept deliberately. Mitigated by the rate limit on sign-in. **Remediation:** offer passkey selection independently of the address entered.

F-30 · Integration check scripts outdated

Criticality: low · **Standard:** ISO/IEC 27001 A.8.29 (security testing in development and acceptance)

The nine integration check scripts (`tests/integration`, started through `run-all.sh`) run as a stage of their own in the build pipeline against a freshly migrated server with its own PostgreSQL instance. Currently all nine fail at sign-in through OpenID Connect already: the scripts' sign-in flow obtains no access token (`missing_token`), so the actual checks are never executed. Further deviations lie behind this: the scripts use former role names and send empty payloads that the server rightly rejects. If a script fails, the stage fails; if no Linux Docker engine is available on the build agent, the stage is skipped with a warning. Every run labels its database container with a run ID and removes the run's container, server process and temporary files, also after failures and cancellations. The flows concerned, including sign-in and token issuance, are also covered by the xUnit suites against real PostgreSQL; for now the scripts are therefore no independent evidence. **Remediation:** adapt the scripts' sign-in flow to the current sign-in route and update roles and payloads.

F-32 · Connection pool without back-pressure, SQL logging in production

Criticality: low · **Standard:** ISO/IEC 25010 (performance efficiency, reliability); ISO/IEC 27001 A.8.6

The load test (`docs/server/CAPACITY.md`) shows two points. First, the database connection pool (100 connections) is as large as PostgreSQL's `max_connections` (100); under overload the database rejects new connections and the server answers HTTP 500 instead of letting requests wait or rejecting them with 503. Second, in production the server logs every executed SQL command at Information level, which increases log volume and CPU demand under load. **Remediation:** set the pool size below `max_connections` and answer pool exhaustion with HTTP 503; set the command logging of the database layer to Warning in production.

F-33 · XTA 2 and FIT-Connect only tested against specification mocks

Criticality: low · **Standard:** ISO/IEC 25010 (compatibility, interoperability); ISO/IEC 27001 A.8.29

File transfer per xdomea can be sent over four transports. The directory transport works on a mounted file system; SFTP has been tested against a real test server, including host key pinning and time limits. XTA 2 version 4 and FIT-Connect are implemented according to the specification but tested only against our own simulations (`XoevGatewayTests`: mTLS, MTOM, technical faults, import confirmation, JWE encryption, certificate chain and revocation check of the destination). Acceptance against a real XTA 2 counterpart or the FIT-Connect test environment is outstanding. Two tests of the FIT-Connect revocation check run only on Linux or Windows and were skipped in the run for this assessment. Differences between simulation and a real counterpart would only surface in operation; a failed attempt does not lose data but leads to a retry and to the connection pausing automatically. FIT-Connect is implemented for sending only. **Remediation:** acceptance of both transports against real counterparts; until then, run a test transfer with your own counterpart before productive use.

Remediated findings

The following findings are remediated in the state of version 1.3.0. Described are the implemented mechanism and the evidence; the numbers stay reserved.

F-01 · Retention periods for the change feed and deletion markers

A daily background service (`SyncRetentionService`) removes entries of the synchronisation change feed and deletion markers older than the periods configured in the organisation policy and records every run in the audit log; the clean-up state per organisation is kept in a table of its own (step 31 of the migration ledger). Devices whose last synchronisation precedes the cleaned range are asked to perform a full resync; they reload the server state and reconcile their data with it, and unsent changes to records removed in the meantime are shown as conflicts. Revocation and deletion entries are removed in batches; each batch raises the resync floor in the same transaction, so an abort never loses a revocation that no device has received. Older app versions without this support transparently receive the complete state. Limitation: records deleted or revoked on the server within the cleaned range remain on such devices until the app is updated. The generated deletion concept describes the automatic enforcement. **Evidence:** `SyncRetentionTests` (including an abort after deletion on SQLite and PostgreSQL), `SyncFullResyncEndpointTests`, full resync in the sync client and desktop suites, upgrade test from a version 1.0.0 database.

F-03 · Third-party inventory and bill of materials

`tools/sbom.sh` generates CycloneDX bills of materials for the server, licence server, app and desktop application; the build pipeline stores them as an artefact. The component overview (`THIRD-PARTY-NOTICES.md`) has been rewritten, and `ThirdPartyInventoryTests` fails as soon as it deviates from the centrally pinned or actually resolved package versions. The two outdated risk statements are corrected.

F-04 · Static code analysis and image scanning

`tools/sast.sh` builds the shipped code with all security rules of the .NET analysers and the EF Core rules against SQL injection; every finding fails the build unless it is suppressed at the affected code with `[SuppressMessage]` and a justification, or listed with a justification in the allowlist. In the assessed state all reviewed exceptions are kept at the code (12 places: MD5 for stable IDs, search paths of native calls, certificate revocation checks, migration SQL), the allowlist is empty, and 0 findings are blocking. `tools/scan-images.sh` checks the built images and base images with Trivy; fixable "Critical" and "High" vulnerabilities fail the scan, exceptions carry a justification and an expiry date. The integration check scripts run as a stage of their own in the pipeline (for their state see F-30). At the end of every pipeline run, also after failures and cancellations, the containers, server processes, reports and temporary files of that run are removed. Overview: `docs/security/QUALITY-GATES.md`.

F-11 · Clock synchronisation as an operating requirement

Time synchronisation of the host is an explicit requirement (NFR-OPS-01 of the requirements specification). The technical handbook lists it among the installation requirements and describes in the section "Checking time synchronisation" which mechanisms depend on the server clock – device proof (±5 minutes), offline grants, tokens, two-factor codes, validity windows of SSO responses, licence assertions (2 minutes tolerance), audit log timestamps and deadline jobs – and how the operator verifies it (`timedatectl status`, `chronyc tracking`, comparison with the container time). The operations runbook and the deployment guide name it as a prerequisite. Running it remains the operator's task; the admin panel does not show clock offset. **Evidence:** `docs/handbooks/admin-technical.en.md` (Requirements, "Checking time synchronisation"), `docs/deploy/RUNBOOK.md` (golden rule 8), `deploy/docker/README.md` (Prerequisites), `docs/requirements/REQUIREMENTS.md` (NFR-OPS-01, section 5.8.1); each with a German version.

F-12 · Requirements specification part of the project

`docs/requirements/REQUIREMENTS.md` (German `REQUIREMENTS.de.md`) defines purpose and scope, stakeholders and roles, system context, functional requirements per domain (cases, documentation and e-file, planning, time and billing, trips, sync, admin panel, data protection workflows, licensing, local AI) and non-functional requirements. Every requirement carries a stable identifier, a status and, as acceptance criterion, the test class that proves it. Security requirements are kept as application security controls under ISO/IEC 27034 with a verification measure (authorisation, input validation, logging without content, key management, signed update chain, pipeline gates). A traceability matrix links requirement, design document and test evidence; unmet requirements are shown with their finding number; the change process defines maintenance per release and audit round.

The original documents behind the paragraph references have not been preserved. A concordance resolves the references of the server, licensing and app specifications from their context and maps them to the new identifiers, so the meaning of every number is traceable for auditors. **Limitation:** the concordance is a reconstruction, not the original. Control library and target level of trust under ISO/IEC 27034 remain open (measures plan). **Evidence:** `docs/requirements/REQUIREMENTS.md`, `docs/requirements/REQUIREMENTS.de.md`.

F-14 · Machine-readable interface description

`docs/api/openapi.v1.json` describes the app and management interface in OpenAPI 3.1 format (67 paths, authentication schemes, error codes; without the administration interface), `docs/api/openapi.integration-v1.json` the integration API (23 paths, scopes, error codes `integration.*`). `OpenApiDocumentTests` and `IntegrationApiTests` fail as soon as a file no longer matches the interface; in production the server exposes no OpenAPI endpoint, and the description of the integration API can be downloaded in the admin panel.

F-15 · Load and capacity evidence

k6 load tests (`tools/loadtest/run.sh`) with realistically signed-in devices and signed requests; results in `docs/server/CAPACITY.md`. With 50 concurrent virtual practitioners the pull took 232 ms at p95 without errors, with 100 it took 4.9 s at p95 with 0 % errors, with 200 the database connections were exhausted (see F-32). Sizing statement: up to 200 concurrently working practitioners with the default synchronisation interval are covered with ample headroom on 4 vCPUs and 8 GB RAM; beyond that the measurement should be repeated on the target hardware.

F-16 · Pagination in the log view

The log view pages through all entries while keeping the filters, and notices of a broken chain position link to the page containing the affected entry. **Evidence:** `AuditPagingTests`.

F-17 · Backoff in the synchronisation client

After network errors, timeouts and overload responses the interval until the next attempt grows exponentially with random jitter up to at most one hour; a `Retry-After` value from the server is honoured. **Evidence:** `SyncRetryBackoffTests`.

F-18 · Measured test coverage

`tools/coverage.sh` measures and merges the coverage of all seven test suites; the build pipeline publishes the report. Measured: 88.0 % of lines and 68.1 % of branches.

F-19 · Licence notice

The project contains a licence file (`LICENSE`) governing the rights of use.

F-21 · Daily overview for Watch and widgets

While hiding is active, the app replaces every case alias with a placeholder and omits locations before writing, and the file is stored with the "complete" protection class; if that fails, it is discarded. **Evidence:** `CompanionSnapshotTests`, `WatchLogicTests` (Apple test target).

F-22 · Licence server addresses in the deployment template

The template only names HTTPS addresses, and outside development the server does not start with unencrypted licence addresses.

F-25 · Self-service deletion of the pseudonymous licence identity

An endpoint of the licence server deletes the pseudonymous identity together with its related records after proof of the recovery code; identities managed by an organisation and identities holding an active seat are not deleted, and the endpoint is rate-limited. The app offers the deletion in the licence settings and then removes the local licensing state. **Evidence:** 13 tests in `StandaloneIdentityDeletionTests`.

F-26 · Secure attribute and encrypted session key material

In production, session and anti-forgery cookies always carry the Secure attribute, even behind a proxy that forwards internally over HTTP; the only exception is a test installation explicitly permitted to use HTTP (see Operator decisions). The key ring of the session management is encrypted with the certificate `encryption.pfx`; existing unencrypted keys are re-encrypted in place at start-up. Backups taken before this step still contain the keys unencrypted and must be protected or replaced accordingly. **Evidence:** `CookieSecurityProductionTests`, `AntiforgeryCookieSecureTests`, `DataProtectionKeyRingTests`.

F-29 · Storage quota credit

Every deletion path – clean-up of orphaned files, permanent removal of archived activity records, deletion of a user's files – credits the freed storage back to the quota atomically within the same operation, never below zero. When cleaning up orphaned files and deleting a user's files, the physical deletion jobs enter the persistent deletion queue in the same transaction; if deletion on the volume fails or the process ends after the commit, the hourly run retries the job. **Evidence:** `QuotaCreditTests` (failing file deletion, stop after commit, SQLite and PostgreSQL).

F-31 · Unmaintained Docker CLI in the server and updater images

The server and prebuilt images (the updater also runs in the server image) take the Docker CLI from the maintained base image `docker:29-cli` instead of `docker:27-cli`; the standalone updater image, which was no longer shipped, has been removed. The Compose plugin is built from the source of the release tag during the image build (the commit is checked against its checksum, currently with `golang.org/x/net` 0.62.0), so that fixes in Go and `golang.org/x/net` are included even before upstream publishes a new release; the Buildx plugin is not shipped. For the Docker image, the pipeline's base-image scan checks only the CLI that is actually copied; it covers the self-built Compose through the module files kept in the repository, the Compose release version itself, the Go build image and a check that the version pins match. Base images are checked for both shipped platforms (`linux/amd64`, `linux/arm64`). Result: no fixable "Critical" or "High" vulnerability in the shipped images, and the scan's ignore list is empty. **Evidence:** image scan (`tools/scan-images.sh`, built images and base images) passed without exceptions; compatibility test of every Docker and Compose command used by the updater against Docker Engine 20.10, 23, 24, 25, 26, 27, 28 and 29.

Operator decisions

Numbers F-09 and F-10 do not denote deviations. Both are settings an operator chooses deliberately; the default is secure and the operations documentation describes the effect. The numbers stay reserved.

No.	Setting	Secure default	Recommendation
F-09	`OPENIDDICT_ALLOW_INSECURE_HTTP` permits sign-in over unencrypted HTTP	off; in production, session and anti-forgery cookies always carry the Secure attribute unless this setting is enabled	set only for test installations without TLS, never for installations holding case data
F-10	`REVERSE_PROXY_ALLOWED_HOST` restricts the accepted host names	all host names; forwarded sender information is accepted only from the fixed proxy network, and the supplied proxy templates fix the server name	set to the server's host name when operating under a domain

---

Remediation plan

Ordered by ratio of effect to effort. Remediated findings are not listed.

Rank	Measure	Finding	Effort	Standard
1	Two-factor authentication on by default	F-05	low	27001 A.8.5
2	Disable dictation without local recognition or control it by policy	F-20	low	27001 A.5.14
3	Container hardening and resource limits	F-08	low	27001 A.8.9
4	Limit the administration session to 30 minutes idle	F-23	low	27001 A.8.5
5	Connection pool below `max_connections` with HTTP 503 on exhaustion, command logging at Warning in production	F-32	low	25010, 27001 A.8.6
6	Adapt the integration check scripts to the current sign-in route, roles and payloads	F-30	low	27001 A.8.29
7	Reduce server information before sign-in, passkey selection independent of the address	F-27, F-28	low	27001 A.8.5, A.8.9
8	Tie screen protection on iOS (detect and cover recordings) and on the desktop (display exclusion) to the setting	F-07	medium	27001 A.8.12
9	Supply alerting rules, forward application logs	F-02	medium	27001 A.8.16
10	Accept XTA 2 and FIT-Connect against real counterparts	F-33	medium	25010, 27001 A.8.29
11	Derive control library and level of trust	—	medium	27034
12	Desktop key on macOS via Keychain interface (Linux part done: Secret Service mandatory)	F-24	medium	27001 A.8.24
13	Tie refresh token validity to offline duration	F-06	low	27001 A.5.17
14	Test accessibility with assistive technology, measure contrast, tag PDFs	F-13	high	25010, EN 301 549
15	Independent penetration test	—	high	27001 A.5.35

The low-effort ranks amount to a few working days in total. The accessibility item is the most laborious and, for public-sector bodies, the most consequential.

Coverage of ISO/IEC 27001 Annex A

Legend: met · partial · open · organisational (with the operator) · not applicable

A.5 Organisational controls

No.	Control	Status
5.1	Policies for information security	organisational
5.2	Roles and responsibilities	partial
5.3	Segregation of duties	met
5.4	Management responsibilities	organisational
5.5	Contact with authorities	met
5.6	Contact with special interest groups	met
5.7	Threat intelligence	partial
5.8	Information security in project management	met
5.9	Inventory of information and other associated assets	met
5.10	Acceptable use of assets	met
5.11	Return of assets	met
5.12	Classification of information	met
5.13	Labelling of information	met
5.14	Information transfer	met
5.15	Access control	met
5.16	Identity management	met
5.17	Authentication information	partial
5.18	Access rights	partial
5.19–5.22	Supplier relationships	partial
5.23	Use of cloud services	met
5.24–5.27	Incident planning, assessment, response, learning	partial
5.28	Collection of evidence	met
5.29	Security during disruption	met
5.30	ICT readiness for business continuity	partial
5.31	Legal and statutory requirements	met
5.32	Intellectual property rights	met
5.33	Protection of records	partial
5.34	Privacy and protection of personal data	met
5.35	Independent review of information security	partial
5.36	Compliance with policies and standards	organisational
5.37	Documented operating procedures	met

A.6 People controls

No.	Control	Status
6.1	Screening	**organisational**
6.2	Terms and conditions of employment	**organisational**
6.3	Awareness, education and training	**organisational**
6.4	Disciplinary process	**organisational**
6.5	Responsibilities after termination	**organisational**
6.6	Confidentiality agreements	**organisational**
6.7	Remote working	**technically supported**
6.8	Information security event reporting	**met**

A.7 Physical controls

No.	Control	Status
7.1–7.14	Secure areas, entry, equipment, cabling, maintenance, disposal	**entirely organisational**

The generated record of measures surveys the relevant points in a structured way and thereby forms the interface to the operator's ISMS.

A.8 Technological controls

No.	Control	Status	Reference
8.1	User endpoint devices	partial	F-07
8.2	Privileged access rights	met	
8.3	Information access restriction	met	
8.4	Access to source code	organisational	
8.5	Secure authentication	partial	F-05, F-06, F-28
8.6	Capacity management	partial	F-08, F-32
8.7	Protection against malware	met	
8.8	Management of technical vulnerabilities	fulfilled	
8.9	Configuration management	partial	F-08, F-27
8.10	Information deletion	met	
8.11	Data masking	met	
8.12	Data leakage prevention	partial	F-07
8.13	Information backup	partial	
8.14	Redundancy of information processing facilities	partial	
8.15	Logging	partial	F-02
8.16	Monitoring activities	partial	F-02
8.17	Clock synchronisation	met	
8.18	Use of privileged utility programs	met	
8.19	Installation of software on operational systems	met	
8.20	Networks security	met	
8.21	Security of network services	met	
8.22	Segregation of networks	met	
8.23	Web filtering	not applicable	
8.24	Use of cryptography	met	
8.25	Secure development life cycle	met	
8.26	Application security requirements	met	
8.27	Secure system architecture and engineering	met	
8.28	Secure coding	met	
8.29	Security testing in development and acceptance	partial	F-30, F-33
8.30	Outsourced development	not applicable	
8.31	Separation of development, test and production	met	
8.32	Change management	met	
8.33	Test information	met	

No.	Control	Status	Reference
8.34	Protection during audit testing	organisational	

Summary: of 93 controls, 66 can be addressed by software. Of those, **40 are fully met and 26 partially, none open**. The remaining 27 rest with the operating organisation (25) or are not applicable (2). Some rows of the table group related controls into one area; the figures count every control individually.

Coverage of ISO/IEC 27034

Element	Status	Note
Requirements and environment specified	met	Requirements specification with stable identifiers, test evidence and a concordance of legacy references
Security risks assessed	met	28 threat scenarios with named residual risk, seven of them on the interfaces
Application normative framework	open	
Control library	present in substance	Security requirements as application security controls with verification measures in the requirements specification; not declared as a library with a level of trust
Controls paired with verification measurement	met	A test per finding that fails if the fix is reverted
Targeted level of trust defined	open	
Security across the life cycle	practised, not described	
Provisioning and operation	met	Deployment runbook, incident procedures, migration process
Application security audit	partial	Several vendor-commissioned assessment rounds, among them the full review of 3 Oct 2026, the tool-assisted second review of 8 Oct 2026 and review rounds of the integration interfaces from 28 Oct to 1 Nov 2026; no independent assessment

Assessment. The substance for ISO/IEC 27034 is largely present, and the requirements specification as the first step of the process is included. What is missing is the remaining structure the standard expects. From the threat model, the permissions documentation and the existing test suite, a control library with associated verification measurements can be derived with moderate effort. Of all four standards, this offers the best ratio of effort to achievable evidence.

Coverage of ISO/IEC 25010

Characteristic	Status	Principal basis or gap
Functional suitability	**met**	Complete operational workflow, four manuals and an interfaces manual, ~2,250 server tests
Performance efficiency	**partial**	Indexes, streaming, quotas, load evidence with sizing statement – overload behaviour of the connection pool open (F-32)
Compatibility	**met**	Five platforms, open export formats, interface descriptions in OpenAPI 3.1 format, documented standard interfaces (REST, webhooks, CMIS 1.1, XÖV/xdomea, SCIM 2.0, iCalendar, syslog/CEF) – acceptance of XTA 2 and FIT-Connect against real counterparts open (F-33)
Interaction capability	**partial**	Seven languages, screen reader labels and a statement in place – assistive-technology testing open (F-13)
Reliability	**met**	Offline capability, conflict surfacing, idempotency, verified restore
Security	**met**	See the Annex A table
Maintainability	**met**	17 sub-projects, test ratio 0.63, measured test coverage 88.0 % of lines and 68.1 % of branches, correlation identifiers
Flexibility	**met**	Five platforms from one code base, self-hosting, central settings
Safety	**partial**	No automated professional decisions, fail-closed – no operational hazard analysis

Coverage of ISO/IEC 29147 and 30111

Requirement	Standard	Status
Published disclosure policy	29147	**met**
Reporting channel and contact details	29147	**met**
Acknowledgement and deadlines	29147	**met**
Coordinated disclosure	29147	**deadline published, never yet applied**
Security advisories to operators	29147	**template and path defined**
Receipt of reports	30111	**mailbox in place**
Verification	30111	**practised**
Development of the remediation	30111	**practised**
Evidence of remediation effectiveness	30111	**met**
Release	30111	**path defined (signed update with downgrade protection), not yet used for a security fix**
Post-release review	30111	**described as mandatory, never yet performed**
Described process with roles and deadlines	30111	**met**
Severity rating	30111	**met**

Assessment. The substantive part of 30111 – verification, remediation, evidence of effectiveness – is demonstrably practised. The outward interface under 29147 is in place: policy, contact address with `security.txt`, published deadlines, advisory template and reporting function in the user interfaces. What remains open is solely the evidence from operation – coordinated disclosure, advisories to operators and the retrospective are described but have never been run, because no release has yet had to close a reported vulnerability.

Remaining strengths in summary

So that this report does not leave the wrong impression – the following properties are unusual in a product of this size and would stand out positively in an assessment:

1. **Honest delimitation.** The documentation states explicitly what the product does not provide: no end-to-end encryption, no remote wipe, no claim to certification, readability of content by the operator. Every security-relevant statement in the documentation was held against the code during this assessment; no assurance remained that the code does not deliver.

2. **Consistent fail-closed behaviour.** Virus scanner unreachable in production – upload rejected. Monitoring interface without an access token in production – not found. Signing key missing – rejection. Migration ledger unresolvable – abort. This pattern is consistent, not occasional.
3. **Evidence instead of assertions.** For every security control there is a test that fails as soon as the control is removed. That is the difference between "implemented" and "demonstrably implemented".
4. **Legal documents generated from reality.** The record of measures, the processor agreement template and the deletion concept are produced from the live configuration of the respective instance. A divergence between documented and actual state is thereby ruled out – a property rarely achieved in practice.
5. **Dependency vulnerability checking as a build error.** From severity "low" upwards, including transitive dependencies. Stricter than common practice.

Validity

This report describes product state 1.3.0 (build 26302) as assessed on 1 November 2026. It is a vendor self-assessment based on a structured review of the source code and does not replace an audit by an accredited body.

Software Notion · Sebastian Kruse · Kornblumenring 40 · 31600 Uchte · Germany · https://software-notion.de