Trausto — Client-Side Encryption

Technical whitepaper for architecture, security and procurement review

Version 1.1 · 12 August 2026 · RiskZone GmbH, Rothenburg LU, Switzerland

Contact for questions on this document: security@trausto.com

What this document is, and what it is not

This is a description of how Trausto encrypts customer content, written for people whose job is to disbelieve it. It states the mechanism, the properties that follow from it, and — in the same detail — the properties that do **not** follow. Where the implementation is weaker than the architecture, the section says so rather than describing the target state.

It is **not** an audit. Nobody outside RiskZone GmbH has reviewed it at the time of writing. That review is planned as one of three independent evidence steps (external penetration test, external review of this document, ISO/IEC 27001 certification), and this page will name the reviewer and the date once it exists. Until then, treat this as a vendor statement that can be checked against the product, not as third-party assurance.

Derived from the internal architecture documentation (`docs/security/e2ee.md`), which remains the normative source. Where the two disagree, the internal document is correct and this one is stale — please report it.

---

1. The claim, in one paragraph

Assessment content — assets, zones, conduits, threat scenarios, risk ratings, uploaded evidence — is encrypted in the customer's browser before it is transmitted. The server stores ciphertext and cannot decrypt it: it holds no private key, no data encryption key, and no recovery secret. A database compromise, a rogue operator, or a lawful order served on RiskZone GmbH therefore yields ciphertext for that content. **One exception exists and is described in section 6.**

2. Key model

Element	Mechanism
User identity key	One X25519 keypair per user
Per-device protection	Private key wrapped separately per device via WebAuthn PRF
Content keys	Per-project data encryption key (DEK), wrapped for each member and device
Recovery	24-word BIP39 phrase, Argon2id key derivation

Element	Mechanism
Symmetric encryption	AES-256-GCM
DEK wrapping	AES-KW (256-bit)
Key agreement derivation	HKDF

One keypair, many devices. Each device holds its own PRF-wrapped copy of the same private key. Losing a device does not compromise the others; removing a device revokes that copy — it is no longer served, but retained.

The unlock is hardware-bound. The wrapping secret comes from a WebAuthn PRF ceremony against a credential bound to the origin and held in the device's secure element. It is phishing-resistant by construction: a credential registered for one origin does not respond to another.

Enterprise SSO does not change this. A federated login proves *identity*. It does not unwrap a key. After an SSO login the client still performs the WebAuthn PRF ceremony or uses the recovery phrase. Deriving key material from an ID token is explicitly excluded — an identity provider compromise must not become a content compromise.

3. Every ciphertext is bound to its purpose

All AES-GCM operations pass through one function that computes additional authenticated data as:

```
AAD = SHA-256( len(prefix)‖prefix ‖ len(purpose)‖purpose ‖ len(ownerId)‖ownerId
             ‖ len(bindingId)‖bindingId ‖ len(version)‖version )
```

Fields are length-prefixed with four big-endian bytes so that field boundaries cannot be shifted — `("ab","c")` and `("a","bc")` produce different AAD.

The practical consequence: a ciphertext produced for one purpose, owner or object **fails to authenticate** when presented in another position. A compromised server cannot serve object A's ciphertext in place of object B and have the client accept it. This is the property that turns "the server stores ciphertext" into "the server cannot rearrange what it stores".

Two constraints follow, and both are design rules we hold ourselves to:

- **AAD inputs must be known to the client independently.** An identifier that the client learns only from the server's own response cannot bind anything — the server would be authenticating its own claim.
- **Row-per-object tables need per-object binding.** A binding of the form `(orgId, kind)` is sound only for data that exists once per organisation. Using it for a table with many rows per organisation re-creates the substitution weakness the AAD exists to prevent.

4. Membership is a key operation, not a flag

Adding someone to a project wraps that project's DEK for their key. Removing them rotates the DEK and re-wraps it for the remaining members. Deleting a project destroys the envelopes, which is what makes the stored ciphertext unrecoverable — there is no soft-delete in the cryptographic sense.

The limit, stated plainly: what a removed party already decrypted and retained is outside the reach of any vendor. Rotation controls future access, not past copies. Anyone claiming otherwise is describing a system that cannot exist.

A second limit, currently open — and being closed: rotation of the organisation-level key on organisation-member removal is **not automatic in the version this document describes**. A manual rotation endpoint exists, re-wraps the remaining holders atomically, and is offered to administrators after a removal. Until rotation is automatic, a removed organisation member who retained a cached organisation key keeps transitive unwrap capability for material wrapped under it.

We record this as a limit rather than a footnote because the honest description of the current build is that organisation-level revocation depends on an administrator acting. The automatic path — rotation triggered by the removal itself, covering the member keys, the risk register including its retained snapshot history, the project-level organisation wraps, the recovery envelope and pending invitations in one resumable batch — is decided and scheduled ahead of general availability. **When it ships, this section changes and this document gets a new version number.** A reader comparing versions should be able to see exactly when the property changed; that is why it is written this way rather than as a forward-looking promise.

5. Organisation-wide views are computed on the client

Risk aggregation across an organisation happens after decryption, in the browser, over exactly the projects the current user can decrypt. Two users with different access therefore see different organisation totals — which is correct, and occasionally surprising.

Server-side aggregation over decrypted content is out of scope by construction. Where aggregate reporting is offered, it is an explicit, organisation-administrator opt-in publishing a declared, minimal set of computed values — never the underlying content.

6. The exception: server-routed AI

Trausto offers an optional AI assistant. It has two modes.

In-browser mode: the model runs client-side. No content leaves the device for this purpose. One nuance belongs here rather than in a footnote: in automatic mode the client uses the in-browser model when one is available and otherwise falls back to the server-routed mode. A user who wants to guarantee that no content is ever server-routed selects in-browser mode explicitly.

Server-routed mode: the request — which may include file contents, entered text and current project data — passes through our worker in the clear to reach the configured AI provider — the customer's own endpoint or, where one is set, a platform-wide default the organisation can disable. For the duration of that request, that content is not encrypted from our perspective.

This is the one place where the sentence "the server never sees plaintext content" does not hold, and it is stated on the public website in the same words. The feature is opt-in. Operational telemetry about these calls (which provider, success or failure, duration) is retained; request content is not.

If your threat model does not permit this, use in-browser mode or do not enable the assistant. We would rather lose the feature than the sentence.

7. What the server holds in the clear

Encryption is half a promise until you say what stays unencrypted — and for how long. Retention values are stated where they are fixed in code; a dash means the duration is operational and not asserted here. This table is generated from the same data source as the website's disclosure, and two checks hold that: the renderer refuses to produce this PDF while the table and the site data differ, and the website's build fails when the disclosure data changes after this render (the render's data fingerprint is published next to the PDF).

Category	Readable by us	Retention	Why it exists
Assessment content	No — ciphertext only	—	Assets, zones, conduits, scenarios and evidence are encrypted on your device before upload. The one exception is the server-routed AI path below.
Uploaded evidence	No — ciphertext only	—	Supplier documents and attachments follow the same path as assessment content.
AI processing	Yes, for the duration of the request — only if you use the server-routed AI path	—	If you use the server-routed AI assistant, the content of that one request passes through our worker in the clear to reach the configured provider — your own endpoint or, where one is set, a platform-wide default your organisation can disable. Running the model in the browser avoids this — that is the honest reason the option exists.
Key material	No — wrapped only	—	Your private key is wrapped separately per device by the device's secure element; your personal recovery phrase and the optional organisation one are generated in your browser and never leave it in the clear. We hold wrapped forms only. An organisation key can unwrap the projects in its permission chain — that is how an organisation keeps access when a person leaves, and it means the organisation recovery phrase reaches those projects too.
Account identity	Yes	—	Name and business email. Needed to authenticate you and to address you in support.
Project membership	Yes — structure, not content	—	Which record belongs to which project, and who has access to it. That controls who can retrieve; the content becomes readable only through the project key.
Timestamps	Yes	—	Created and last-changed times, so concurrent edits and version history work.

Category	Readable by us	Retention	Why it exists
Audit log	Yes	90 days (default) · SIEM export is the long-term archive	Who did what, when. Required for the evidence trail the product exists to produce.
Usage volume	Yes	—	Storage consumed and record counts, for billing and capacity.
Device and session metadata	Yes	Sessions valid 30 days, rows purged after 90 · device entries are revoked, not deleted	Per device: the device name you chose (plain text), its public key, when it was added and when its wrapped key was last retrieved; removed devices are revoked, not deleted. Per session: browser identifier, an IP prefix — never the full address — and which device opened the session.
Transactional email	Yes — sent via Mailgun (EU region)	—	Registration and recovery codes, invitations and account-change notices are delivered by the email provider Mailgun. These mails carry: confirmation codes, device name, organisation name (also in the subject line), role, and the address of the acting person; on an address change, the old address is told the new one. Recipient addresses and mail metadata therefore also sit with the delivery provider.
Support and administration access	Yes — metadata, never content	—	Platform administrators can read member lists including email addresses, audit logs and usage statistics; assessment content they cannot — it exists on our side only as ciphertext. Administrative writes are recorded in the audit log; reads currently are not recorded individually.
AI operational telemetry	Yes — metadata — only if you use the server-routed AI path	90 days (default), as part of the audit log	Which provider was called, whether it failed and how long it took. No request content.
SIEM export	Yes — audit metadata — only with a webhook you configure	—	If your org configures a SIEM webhook, audit events go to the endpoint you name. Assessment content is not part of that stream.

The metadata layer is a relationship register: which organisation, which people, which projects, who may see what. It is not nothing, and a buyer should weigh it as what it is.

8. Threat model

Addressed by the design

- Database compromise — content is encrypted with keys the server never holds
- Rogue operator or compelled disclosure — the same

- Ciphertext substitution — purpose-bound AAD (section 3)
- Phishing — origin-bound, hardware-backed WebAuthn credentials
- Transport interception — TLS 1.3, HSTS
- Session hijacking — httpOnly cookies, short expiry, server-side invalidation, contextual session risk scoring with step-up re-authentication

Not addressed, and why

- **Cross-site scripting during an active session.** Script running in the page can read the decrypted key from memory. This is inherent to browser-based end-to-end encryption and is shared by every product in this category. Mitigations: content security policy, httpOnly cookies, automatic logout. The hardware-backed PRF secret itself is never exposed to JavaScript.
- **A compromised client device.** A keylogger can capture a recovery phrase as it is entered; malware with access to the browser process can read what the user can read. Under end-to-end encryption the endpoint is the perimeter — that is the trade, not an oversight.
- **Browser vulnerabilities.** Sandbox escapes are outside the application boundary.
- **Retained copies.** See section 4.
- **Future cryptanalysis.** See section 9.

9. Post-quantum status — read this carefully

The DEK envelope format is versioned and every version written today is tagged **pre-quantum**. Three exist: `x25519-pre-pq-v1` and `-v2` are legacy and read-only; `x25519-pre-pq-v3` is what the product writes, and it binds the recipient, the ephemeral key and the resource into the key derivation. In all three, the post-quantum input to the derivation is **32 zero bytes**: a placeholder, fixed as a compatibility contract so a later hybrid derivation can be introduced without silently reinterpreting existing envelopes.

This means Trausto offers no post-quantum protection today. A hybrid version combining X25519 with ML-KEM is designed and its identifier reserved as `x25519-mlkem768-hybrid-v1`, but the write and unwrap paths are not shipped. Pre-quantum unwrap logic rejects envelopes claiming a post-quantum algorithm, so a mismatch fails loudly instead of degrading silently. Existing envelopes will require a controlled re-wrap when the hybrid version ships. The key-size choice between ML-KEM-768 and ML-KEM-1024 is not yet settled.

We state this because the reserved identifier is visible in the codebase and an unexplained reservation invites exactly the wrong inference. Any Trausto material claiming post-quantum protection before that migration completes is wrong, and we would like to hear about it.

10. What you can check without asking us

- Swiss commercial register, UID **CHE-292.851.395**
- `https://www.trausto.com/.well-known/security.txt` (reachable from launch; until then the site is served from a staging host, where the same file answers) — disclosure contact, acknowledgement within five Swiss business days, and an explicit statement that no bug bounty is operated

- Transport headers and TLS configuration of `app.trausto.com` (from launch)
- The public disclosure table on the website, which matches section 7 of this document
- That no certification is claimed anywhere in our material that we do not hold

11. Data portability and exit

Export runs in the browser against the customer's own keys and produces JSON, CSV or XML — the same data the generated report is built from. It does not depend on our cooperation beyond the service being reachable. If the last key on the customer side is lost, the organisation recovery key takes over — provided the organisation has set it up and not disabled it. It is deliberately opt-in, because it lifts exactly the finality the rest of this document relies on.

Both are product functions, not contractual promises.

Open items named in this document

For a reviewer's convenience, the places where implementation trails architecture:

1. Automatic organisation-key rotation on member removal (section 4) — decided, scheduled before general availability
2. Post-quantum hybrid envelope (section 9) — designed, identifier reserved, not shipped
3. External review of this document (front matter) — planned, not yet commissioned

We publish this list rather than waiting until it is empty, because a whitepaper without one is either finished software or an incomplete document, and we are not finished software.

Version history

Version	Date	Change
1.0	11 August 2026	Initial version.
1.1	12 August 2026	Section 7 expanded to fourteen items and generated from the website's disclosure data (added: key material, device and session metadata, transactional email, support access; new scope and retention columns). Section 11 now states that the organisation recovery key is opt-in. Section 6 states the automatic-mode fallback explicitly and that the server-routed provider can be a platform-wide default until disabled. Section 2 corrected: removing a device revokes its wrapped copy rather than deleting it. Review status corrected from commissioned to planned.

A reader comparing versions sees exactly when a property changed. That is the point of this table.