LONIA AI / PALLAS

Data Processing Agreement

GDPR Article 28 processor terms for Pallas by Lonia AI

The processor terms under which Lonia AI processes personal data on a customer's behalf in Pallas, covering the subject matter and duration of processing, the security measures relied on, the subprocessor chain, breach notification, assistance with data subject rights, international transfers, audit rights, and the return or deletion of personal data at termination.

Version. 1.0. Published 2026-08-30 by Lonia AI.

Jurisdictions addressed. EU GDPR Article 28 primarily. UK GDPR, the Swiss FADP, PIPEDA and Quebec Law 25, and the Australian Privacy Act 1988 are addressed where their obligations diverge from the GDPR baseline. The Standard Contractual Clauses are incorporated by reference; the SCC text and the Transfer Impact Assessment template are issued separately.

Every control claim in this document is rendered from the Pallas control inventory at src/data/control-inventory.ts, with its status, its evidence basis, and its disclosure text unaltered. Appendix A lists every control cited.

This render is informational; signable version issued on request. Request one at support@lonia.ai, giving your legal entity name and the jurisdiction of your establishment.

This document is provided for procurement and vendor-onboarding review. It is not legal advice, and both parties should have any contractual artifact reviewed by counsel before execution. Questions: support@lonia.ai

Contents

Parties and status of this document

This Data Processing Agreement (the "DPA") forms part of the agreement between the customer (the "Controller") and Lonia AI (the "Processor") for the use of Pallas by Lonia AI (the "Service"). It governs the processing of personal data that the Processor carries out on the Controller's behalf and is designed to meet the requirements of Article 28 of the EU General Data Protection Regulation and of the UK GDPR. Where the two conflict, this DPA prevails over conflicting terms in the main agreement with respect to data processing.

This public render names Lonia AI as the party and is the template a buyer reviews. Once terms are agreed, the executed version is issued on request at support@lonia.ai.

The Processor's public informational copy of these terms is also published as a web page at pallas.lonia.ai/legal/dpa. Where that page and this document describe the same obligation, they are generated and maintained together; where this document goes further, it does so by naming the specific technical control behind an obligation rather than by adding a new obligation.

1. Definitions

Terms defined in the GDPR carry their GDPR meaning: personal data, processing, controller, processor, subprocessor, data subject, supervisory authority, and personal data breach. In addition:

- **Service** means Pallas by Lonia AI, comprising the marketing site at pallas.lonia.ai, the authenticated application at app.pallas.lonia.ai, and the supporting Cloudflare Workers and Supabase infrastructure described in Section 6.
- **Controller Personal Data** means personal data that the Processor processes on the Controller's documented instructions under the main agreement.
- **Applicable Data Protection Law** means the GDPR, the UK GDPR, the Swiss Federal Act on Data Protection, the Personal Information Protection and Electronic Documents Act, the Act respecting the protection of personal information in the private sector as amended by Quebec Law 25, the Australian Privacy Act 1988, and any other data protection law applicable to a party's processing under this DPA.
- **Control Inventory** means the Processor's internal register of security and privacy controls at src/data/control-inventory.ts in the pallas-marketing repository, from which every control claim in this document is rendered.

2. Subject matter, duration, nature and purpose

The subject matter is the provision of the Service. The Processor processes Controller Personal Data for as long as the main agreement is in effect, plus any post-termination period required to return or delete data as described in Section 10.

The Processor processes Controller Personal Data solely to provide the Service: accessibility scanning of Controller-designated websites and documents, remediation tracking, governance and audit logging, reporting, and account administration. The Processor processes Controller Personal Data only on documented instructions from the Controller, including with regard to international transfers, unless required to do otherwise by a law to which the Processor is subject, in which case the Processor will inform the

Controller of that requirement before processing unless the law prohibits it.

3. Categories of data subjects and personal data

Data subjects. The Controller's authorised users, being administrators, managers, contributors, and viewers, and any individuals whose personal data appears incidentally within scanned websites or uploaded documents.

Categories of personal data. User identity data, being name, email address, profile photograph, and organisation; usage and audit metadata; scan targets and findings; and any personal data incidentally contained in Controller-submitted content. The Controller must not intentionally submit special categories of personal data within the meaning of Article 9 for scanning.

The enumeration of data classes the Service holds is a control in its own right, and it carries a disclosure that a Controller carrying out its own assessment should read rather than summarise.

Pallas holds account identity, organisation records, scan findings, generated reports, transiently uploaded source files, audit events, and marketing list membership. It holds no payment card data, no special category data by design, and no health or student records.

> Control data-classes-enumerated. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Disclosure. Pallas does not collect special category data as a matter of design, but it scans customer-supplied URLs and accepts customer-uploaded documents. Content a customer submits is outside Pallas control, so the accurate statement is that Pallas does not solicit or intentionally process special category data, not that none can ever transit the system. Uploaded source files carry a 7 day default retention precisely because they are the class most likely to contain something unintended.
>
> Implementation. app: The 29 tables enumerated in tenant-rls-every-table, plus the three Storage buckets in migration 020.
>
> Verification. marketing: src/pages/trust.astro, What Pallas holds and What Pallas never collects.

4. Obligations of the Controller

The Controller warrants that it has a lawful basis for the processing it instructs, that its instructions comply with Applicable Data Protection Law, and that it has provided any notices and obtained any consents required from data subjects. The Controller is responsible for the accuracy, quality, and legality of Controller Personal Data and for the means by which it acquired that data.

Where the Controller configures retention periods shorter or longer than the Service defaults, the Controller is responsible for ensuring those periods comply with its own retention obligations. The Service enforces a seven-year floor on audit log retention that the Controller cannot configure below, for the reason given at the control in Section 6.

5. Obligations of the Processor

The Processor will process Controller Personal Data only on the Controller's documented instructions; ensure that personnel authorised to process Controller Personal Data are bound by confidentiality; implement the technical and organisational measures described in Section 6; assist the Controller in responding to data subject requests as described in Section 7; assist the Controller with its obligations under Articles 32 to 36, taking into account the nature of processing and the information available to the Processor; make available the information necessary to demonstrate compliance with Article 28; and delete or return Controller Personal Data as described in Section 10.

The Processor will inform the Controller immediately if, in its opinion, an instruction infringes Applicable Data Protection Law.

6. Security measures

The Processor maintains technical and organisational measures appropriate to the risk. Rather than describing them in general terms, this section renders them from the Control Inventory with their status, their evidence basis, and their disclosures unaltered. A Controller assessing the Processor under Article 32 is assessing the rows below, including the ones that say a control is partial or that the evidence for it is a subprocessor's word.

6.1 Authentication

Pallas authenticates users through Google and Microsoft OAuth only. It stores no passwords, offers no password login, and has no password reset flow.

> Control auth-oauth-only. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Implementation. app: src/contexts/AuthContext.tsx:172 and :181; app: src/pages/Onboarding.tsx:123.
>
> Verification. app: Search src/ for signInWithPassword and signUp(.

The trigger that provisions a Pallas profile on first sign-in cannot block a sign-in if it fails.

> Control auth-profile-trigger-fail-safe. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Implementation. app: handle_pallas_new_user, migration 001:105, corrected in 003 and 021:260.
>
> Verification. app: supabase/migrations/021_reconcile_live_state.sql section 5.

Sessions are JWT-based through Supabase Auth, and refresh tokens rotate on use.

> Control auth-session-jwt. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.
>
> Disclosure. Token rotation behaviour is a property of Supabase Auth. Pallas configures it and relies on it; no Pallas code implements it, and neither Pallas repository can demonstrate it independently.

Implementation. vendor: Supabase Auth refresh token rotation.

Verification. marketing: src/pages/trust.astro, Security posture, Session security.

No authentication state, entitlement, or quota is trusted from browser storage. Authorisation decisions are made server side.

Control auth-no-browser-storage-for-entitlements. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: Row-Level Security policies on every pallas_ table, enumerated in the tenant-isolation rows below; app: pallas-app-api.

Verification. marketing: Rule sweep for browser storage API calls across src/.

6.2 Encryption

Data is encrypted at rest with AES-256 through Supabase-managed infrastructure.

Control encryption-at-rest. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.

Disclosure. Encryption at rest is provided and attested by Supabase. No Pallas code implements it and neither Pallas repository can demonstrate it. A buyer who requires independent evidence should request the Supabase SOC 2 Type II report rather than a Pallas assertion.

Implementation. vendor: Supabase managed PostgreSQL and Storage encryption at rest.

Verification. vendor: Supabase SOC 2 Type II report, available to the operator under NDA.

All connections are TLS 1.3, enforced by Cloudflare and Supabase.

Control encryption-in-transit. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.

Disclosure. TLS termination and version are properties of Cloudflare and Supabase. What this repository demonstrates is the header set and the build gate that refuses to ship without it, not the negotiated cipher.

Implementation. vendor: Cloudflare edge TLS and Supabase connection TLS; marketing: public/_headers.

Verification. marketing: scripts/csp-hashes.mjs.

Production secrets are held in the platform secret store and never in either repository. Worker configuration files carry variable names only.

Control encryption-no-secrets-in-repo. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Disclosure. Two Cloudflare KV namespace identifiers are committed, at workers/pallas-scan-email/wrangler.toml and in the pallas-app Workers. A KV namespace id is an identifier rather than a credential: reaching the namespace still requires authenticated access to the Cloudflare account. This is recorded as a deliberate decision, with the note that the ids should move to deploy-time values if either repository is shared with a buyer, an auditor, or a contractor.

Implementation. marketing: workers/pallas-scan-email/.dev.vars.example and

workers/pallas-email-gate/.dev.vars.example; marketing: .gitignore.

Verification. marketing: Rule sweep for key, token, price identifier, and webhook secret strings.

Finding comments, evidence descriptions and code snippets, and evidence files are encrypted at rest by the database and storage provider under a key the provider holds, and in transit by TLS. They are not client-side encrypted: every member of the organization can read them, they are included in the organization's data export, and the Processor can read them as any SaaS operator can.

Control encryption-member-prose-provider-held-key. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: docs/DATABASE_ENFORCEMENT.md:86, What is and is not encrypted; app: src/audit/auditMetadataKeys.ts, AUDIT_METADATA_KEYS.

Verification. app: tests/audit/auditMetadataKeys.test.ts.

6.3 Tenant isolation

Row-Level Security is enabled on every Pallas application table. Twenty-nine tables carry it in the migration file set.

Control tenant-rls-every-table. Status: Implemented. True of the migration file set. Live database state is not provable from source, and the operator pre-check that resolves it is named in the disclosure.

Disclosure. Twenty-nine tables carry RLS in the migration file set (twenty-four before migrations 057 and 059 added the digest-run and rule-override tables on 2026-09-16, twenty-six before migration 067 added the two crawl-state tables on 2026-09-17, and twenty-eight before migration 072 added the API-token table). Two of them, pallas_subscribers and pallas_subscriber_suppression, were created directly on the live database on 2026-08-30 and captured into migration 046 only afterwards, so any count taken before that migration was written disagrees with this one. The 2026-08-30 inventory pass recorded twenty-two on the live project. That difference has not been reconciled against the live database, because code in these repositories does not execute SQL. Before this claim is stated to a buyer with a number attached, the operator should run a live count of pg_tables joined to pg_class.relrowsecurity for tables matching pallas_%, and either correct this row or correct the 2026-08-30 inventory record. Until then the safe published form of this claim omits the count and states only that RLS is enabled on every application table.

Implementation. app: ENABLE ROW LEVEL SECURITY on 29 tables across migrations 001, 002, 005, 010, 017, 021, 024, 029, 034, 041, 046, 057, 059, 067, 072.

Verification. app: Search supabase/migrations/*.sql for ENABLE ROW LEVEL SECURITY and compare the count against CREATE TABLE.

Every tenant-scoped policy resolves membership through two SECURITY DEFINER helpers with a pinned search_path, rather than by re-entering the membership table under its own RLS.

Control tenant-org-scoping-helpers. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_user_has_org_access, migration 002:8, hardened in 031:54; app: pallas_user_org_role, migration 002:17, hardened in 031:68.

Verification. app: supabase/migrations/031_low_severity_hardening.sql.

The evidence, source-files, and reports Storage buckets are private and carry per-organisation RLS policies on storage.objects, with read, write, update, and delete scoped to organisation role.

Control tenant-storage-per-org. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: supabase/migrations/020_storage_bucket_rls.sql; app: pallas_storage_org_access, migration 020:50.

Verification. app: supabase/migrations/020_storage_bucket_rls.sql header.

Server-side operator functions run under service credentials that bypass Row-Level Security by necessity. Their scope is limited, they are versioned in source, and their actions are covered by the audit trail.

Control tenant-service-role-bypass-disclosed. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Disclosure. Service-role credentials bypass Row-Level Security. This is a property of PostgreSQL and Supabase, not a Pallas design choice, and it means tenant isolation for service-role code paths rests on the correctness of those code paths rather than on the database refusing them. The mitigating controls are that the paths are few, versioned, and audit-logged, not that they are constrained by RLS.

Implementation. app: pallas-app-api; app: supabase/functions/enforce-retention/index.ts.

Verification. marketing: src/pages/trust.astro, Security posture, Tenant isolation.

6.4 Access control

Organisation membership carries one of four roles: org_admin, manager, contributor, viewer. Read is available to any member; write and update require contributor or above; delete requires org_admin.

Control access-org-roles. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_organization_users.role, migration 001:30; app: supabase/migrations/020_storage_bucket_rls.sql, role model section.

Verification. app: pallas_user_org_role, migration 002:17 hardened 031:68.

An organisation cannot be left without an administrator. Removing or demoting the last org_admin is refused at the database layer.

Control access-last-admin-guard. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_guard_last_admin, migration 010:106, rebuilt 025:162.

Verification. app: supabase/migrations/015_drop_duplicate_last_admin_trigger.sql.

When a user loses the privilege that let them invite others, their outstanding invitations are revoked.

Control access-privilege-loss-revokes-invites. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_revoke_invites_on_privilege_loss, migration 029:266, rebuilt 040:260.

Verification. app: supabase/migrations/029_medium_security_hardening.sql.

Authenticated endpoints carry per-user rate limits. Endpoints that move or expose data fail closed when the limiter is unavailable; read-only endpoints degrade to load.

Control access-rate-limiting-authenticated. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Disclosure. Cloudflare Workers KV is eventually consistent across colocations, so a KV-backed counter is a burst guard rather than a hard ceiling: concurrent requests can each read the same value before any write lands. Where a hard ceiling was required, the limit is transactional in PostgreSQL instead, as with the 50 invitations per organisation per 24 hours enforced by pallas_org_invites_today. Pallas does not claim KV-backed limits are exact.

Implementation. app: pallas-app-api, checkRateLimit() over the RATE_LIMIT_KV namespace.

Verification. app: workers/pallas-app-api/wrangler.toml, rate limiting comment block.

6.5 Audit logging and tamper evidence

The Processor maintains a cryptographically tamper-evident audit log with server-derived actor identity, in which each entry is hashed together with the previous entry's hash so that any modification, deletion, or reordering of historical events is detectable, and in which erasures carried out under Article 17 or under the Controller's retention policy are recorded as redaction entries carrying their reason.

Every audit event appends an entry to a SHA-256 hash chain. Each entry is hashed together with the previous entry hash, so modification, deletion, or reordering of historical events breaks the chain and is detectable.

Control audit-sha256-ledger. Status: Implemented. Observed in the production database by catalog query on 2026-09-16 and recorded in the pallas-app deployed-objects manifest. The migration that created the object is held out of the repository as a trade secret, so the observation, not a file, is the evidence; objects are cited by name only.

Implementation. app: supabase/migrations/041_audit_tamper_evidence.sql; app: hashing helpers inside migration 041; row_hash and prev_row_hash columns on pallas_audit_events; app: insert trigger on pallas_audit_events, migration 041.

Verification. app: pallas_verify_audit_chain, migration 041; app: pallas_verify_audit_row, migration 041.

The ledger table is append-only in the strong sense: Row-Level Security is enabled with zero policies, and INSERT, UPDATE, DELETE, and TRUNCATE are revoked from every role including service_role. Only SECURITY DEFINER trigger functions can write to it.

Control audit-ledger-append-only. Status: Implemented. Observed in the production database by catalog query on 2026-09-16 and recorded in the pallas-app deployed-objects manifest. The migration that created the object is held out of the repository as a trade secret, so the observation, not a file, is the evidence; objects are cited by name only.

Implementation. app: pallas_audit_chain, created by migration 041 with Row-Level Security enabled; app: SECURITY DEFINER trigger functions inside migration 041.

Verification. app: pallas_audit_chain, migration 041; the table answers permission denied to the anon role through the API.

Lawful removal of an audit row, whether under a retention policy or under GDPR Article 17, appends a redaction entry recording the sequence number, the hash before, the hash after, the reason, and the timestamp. The ledger outlives the row it describes, so a reviewer can distinguish a lawful purge from a concealment.

Control audit-lawful-redaction-recorded. Status: Implemented. Observed in the production database by catalog query on 2026-09-16 and recorded in the pallas-app deployed-objects manifest. The migration that created the object is held out of the repository as a trade secret, so the observation, not a file, is the evidence; objects are cited by name only.

Disclosure. The delete allowlist is a declaration, not an authorisation control. Anything holding service_role and a SQL connection can set the redaction reason. The control is not that setting it is hard; the control is that setting it does not buy silence, because the redaction entry lands in a table that role cannot subsequently edit or truncate. An audit row can be deleted. It cannot be deleted quietly. This distinction is stated here because a reviewer will press on it, and a document that presented the flag as an access control would be overclaiming.

Implementation. app: pallas_audit_delete_reason_allowed, migration 041; the redaction hash is computed inside the same migration; app: update and delete triggers on pallas_audit_events, migration 041.

Verification. app: migration 041, which carries the update and delete allowlists.

Audit rows carry a seven-year (2555 day) retention floor enforced by a database CHECK constraint. An administrator cannot configure a shorter audit retention than the regulatory minimum.

Control audit-seven-year-floor. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: chk_audit_log_retention_min_2555 on pallas_retention_policies, migration 016:34; app: pallas_apply_retention_policies, migration 042.

Verification. app: supabase/migrations/016_audit_retention_floor_check.sql.

Audit attribution survives user erasure. Actor name and email are snapshotted onto the audit row at write time, so severing the identity link under Article 17 does not orphan the audit trail.

Control audit-actor-attribution-survives-erasure. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_actor_snapshot 024:379, pallas_fill_audit_actor_snapshot 040:195; app: supabase/migrations/040_user_fk_set_null.sql.

Verification. app: migration 041, whose update allowlist requires actor_name_snapshot and actor_email_snapshot unchanged.

6.6 Application security

The public scan tool validates every submitted URL against a hostname and IP-literal denylist before fetching it, blocking loopback, link-local, unique-local, IPv4-mapped private addresses, cloud metadata endpoints, and the private, CGNAT, and multicast IPv4 ranges.

Control appsec-ssrf-guard. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Implementation. marketing: src/lib/ssrf-guard.ts; app: workers/pallas-cors-proxy/src/index.ts, isBlockedHost and

isBlockedIpv4.

Verification. marketing: tests/ssrf-guard.test.ts.

Fetched third-party responses are capped at 10 MB, enforced against the declared Content-Length and again against the bytes actually received.

Control appsec-response-cap. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Implementation. marketing: src/lib/response-cap.ts; app: workers/pallas-cors-proxy/src/index.ts:261, MAX_RESPONSE_BYTES.

Verification. marketing: tests/response-cap.test.ts.

Scanned third-party markup renders on a dedicated origin with nothing in its storage, pinned shut with Content-Security-Policy, so retrieved markup cannot reach the application session, cookies, or IndexedDB.

Control appsec-sandboxed-rendering. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: workers/pallas-scanner-sandbox/src/index.ts.

Verification. app: docs/OPERATOR_ACTIONS.md section 1.1, mitigations table.

Untrusted content is escaped before it reaches HTML or email. The only place untrusted input meets raw HTML on the marketing site is JSON-LD, fed exclusively by a serializer that escapes the closing angle bracket and the U+2028 and U+2029 line separators.

Control appsec-output-escaping. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Implementation. marketing: src/lib/jsonld.ts, serializeJsonLd(); marketing: src/lib/html-escape.ts; marketing: src/lib/safe-url.ts.

Verification. marketing: tests/safe-url.test.ts and tests/scan-report.test.ts; marketing: SECURITY_ADVISORY_TRIAGE.md, section titled The one place untrusted input meets HTML.

The build command itself runs every gate: a type check, the typography and published-document freshness checks, the six written-claim verification gates, the unit test suite over the security and validation code, the build, an automated accessibility test against every built page, a Turnstile configuration check against the built artifact, and a sweep of the built output for forbidden claims. The hosting platform runs that command on every push, so a non-zero exit at any step fails the build and nothing deploys; the failing guards remove the dist directory so a failed build leaves nothing to deploy.

Control appsec-release-gate-fails-closed. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Implementation. marketing: package.json, the build script.

Verification. marketing: scripts/test-a11y.mjs, scripts/test-turnstile.mjs, scripts/check-dashes.mjs,

scripts/verify-doc-artifact-freshness.mjs.

6.7 Data residency

Controller Personal Data at rest resides in the Processor's primary Supabase region in the United States. Cloudflare edge locations are global and handle data in transit, being request routing, caching of static assets, and bot mitigation; they are not the system of record and do not hold Controller Personal Data at rest.

The default processing location is the United States. Supabase hosts the database and Storage in us-east-1; Cloudflare serves from its global edge network.

Control data-location-default-us. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.

Implementation. vendor: Supabase project region us-east-1; marketing: src/pages/legal/subprocessors.astro.

Verification. marketing: src/pages/trust.astro, Data residency options.

Processing takes place in the United States only. No European Union hosting exists and none is offered.

Control data-location-eu-option. Status: Not implemented. A commitment the operator makes contractually. No code enforces it.

Disclosure. No customer data is processed in the European Union and no EU infrastructure subprocessor is engaged or planned. Any procurement document, questionnaire response, or sales conversation must present United States processing as the only option and must not present EU hosting as available, on request, or on a roadmap.

Implementation. marketing: src/pages/trust.astro, Data residency options; marketing: src/pages/legal/subprocessors.astro, six operative entries and no planned entry.

6.8 Independent assurance, and what the Processor does not hold

The Processor holds no SOC 2 report, no ISO 27001 certificate, and no PCI DSS attestation of its own. No certificate is held today and the Processor does not claim one. Where a subprocessor holds its own certifications, those are listed on the subprocessor page and belong to that subprocessor.

Pallas holds no SOC 2, ISO 27001, or other independent security certification.

Control vuln-no-independent-certification. Status: Not implemented. A commitment the operator makes contractually. No code enforces it.

Disclosure. Pallas holds no certification of its own. Certifications named anywhere in Pallas materials belong to subprocessors. Where a buyer requires a converted assurance rather than an assertion, the mechanism is a contractual term in the Institutional Terms of Service, the signable agreement, not a certificate Pallas can produce. There is no separate master services agreement.

Verification. marketing: src/pages/trust.astro, Compliance frameworks, Planned.

Pallas has not commissioned an independent penetration test.

Control vuln-no-penetration-test. Status: Not implemented. A commitment the operator makes contractually. No code enforces it.

Disclosure. No third party penetration test has been performed and no report exists. Security review to date has been internal, through repeated structured audits recorded in this repository. Any questionnaire field asking for the date of the last penetration test or for a report must be answered as none performed. This is a real gap for institutional buyers and it should be presented as one rather than deflected to the internal audit history.

Verification. marketing: src/pages/trust.astro, questionnaire answers section.

Dependency advisories are triaged by whether a request path actually reaches the affected code, and the triage is recorded in version control with a review date and the reasoning rather than the conclusion alone.

Control vuln-advisory-triage-artifact. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Disclosure. Three advisories are currently open in npm audit, reported as one low and two high. All ten individual advisories were assessed as unreachable in this build configuration, on the basis that the site is a fully static build with no server runtime, no client hydration, and no build-time image processing. The advisories remain open rather than fixed: the proposed remediation is a two major version Astro upgrade that npm audit labels breaking, and that upgrade is scheduled on its own timeline rather than taken as an emergency for unreachable findings. A buyer running npm audit against this repository will see the counts, and this file is the answer to what they mean.

Implementation. marketing: SECURITY_ADVISORY_TRIAGE.md.

Verification. marketing: SECURITY_ADVISORY_TRIAGE.md, section titled What would invalidate this triage.

Every change reaches production through version control, and the build fails closed rather than degrading.

Control vuln-version-control-and-review. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Disclosure. Pallas is built and operated by a single operator, so there is no second-party code review before merge and no segregation of duties between the person who writes a change and the person who deploys it. The compensating control is the automated build chain, which the hosting platform runs on every push, is identical for every change, and cannot be selectively skipped. Any questionnaire asking about mandatory peer review or segregation of duties must be answered no.

Implementation. marketing: package.json build script; marketing: scripts/csp-hashes.mjs and scripts/test-turnstile.mjs.

Verification. marketing: src/pages/trust.astro:121.

7. Assistance with data subject rights

Taking into account the nature of the processing, the Processor will assist the Controller by appropriate technical and organisational measures, insofar as possible, in fulfilling the Controller's obligation to respond to requests to exercise data subject rights: access, rectification, erasure, restriction, portability, and objection. The Service provides self-service export of findings, reports, and audit records, and a full organisation export covering organisations, members, assets, scans, findings, reports, subscriptions, audit events, and retention policies.

Where the Processor receives a request directly from a data subject in respect of Controller Personal Data, it

will not respond to the request itself except on the Controller's documented instruction, and will refer the data subject to the Controller without undue delay.

Any signed-in user can export the personal data Pallas holds about them, across every organisation they belong to. This serves GDPR Article 15 and Article 20, CCPA and CPRA right to know, PIPEDA Principle 9, and Australian Privacy Principle 12.

> Control gdpr-art15-art20-user-export. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Implementation. app: pallas_export_user_data, migration 039:268, extended to ten collections in migration 042; app: pallas-app-api, POST /export-user-data, 3 per hour per user, fails closed.
>
> Verification. app: supabase/migrations/039_gdpr_user_rights.sql header, controller split section.

Any signed-in user can erase their own account. This serves GDPR Article 17, CCPA and CPRA right to delete, and PIPEDA withdrawal of consent.

> Control gdpr-art17-user-erasure. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Disclosure. Erasure of a user does not erase the organisation content that user created. Findings, comments, and evidence belong to the organisation as a separate controller with its own retention obligation. The identity link is severed and the audit trail keeps the attribution snapshot. A data subject asking for erasure receives this explanation rather than a silent partial deletion. The request also returns HTTP 409 rather than proceeding if the caller is the sole org_admin of any organisation, because completing it would leave that organisation unadministrable.
>
> Implementation. app: pallas_delete_user_account, migration 039:506; app: pallas-app-api, POST /delete-user-account, requires the X-Confirm-Deletion header, 3 per hour per user, fails closed.
>
> Verification. app: supabase/migrations/040_user_fk_set_null.sql.

Pallas distinguishes the two controller roles in its schema, not only in its policy text. Lonia AI is controller for user profile data; the customer organisation is controller for its own scan and compliance data. The user export and the organisation export are separate functions, not one filtered by the other.

> Control gdpr-controller-split-documented. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Implementation. app: pallas_export_user_data 039:268 for the individual, pallas_export_org_data 009:13 rebuilt in 032:67 and 037:297 for the organisation.
>
> Verification. app: supabase/migrations/039_gdpr_user_rights.sql, controller split section.

An organisation administrator can export the full organisation data set for a data subject access request.

> Control gdpr-org-dsar-export. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.
>
> Implementation. app: pallas_export_org_data, migration 009:13, rebuilt 032:67 and 037:297; app: pallas-app-api, POST /export-org-data, org_admin only, 3 per hour per user and per org, fails closed.

Verification. app: supabase/migrations/032_export_completeness.sql.

An organisation administrator can delete the organisation. Deletion cancels the Stripe subscription, wipes Storage under the organisation prefix, and removes the database rows, archiving a defined subset of audit actions first.

Control gdpr-org-deletion. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_delete_org, migration 008:70, rebuilt 022:71, 024:702, 025:245, and again in migration 041; app: pallas-app-api, POST /delete-org, org_admin plus a typed confirmation phrase, deliberately not rate limited; app: pallas_retained_audit_events, migration 021:119.

Verification. app: pallas_record_post_delete_failure, migration 025:337.

The two individual rights functions are callable only by the service role. The acting user id is derived from a verified JWT and is never accepted from a request body.

Control gdpr-rights-service-role-only. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: supabase/migrations/039_gdpr_user_rights.sql, grant section; app: pallas-app-api.

Verification. app: supabase/migrations/039_gdpr_user_rights.sql header, service_role only section.

8. Subprocessors

The Controller provides general authorisation for the Processor to engage the subprocessors published at pallas.lonia.ai/legal/subprocessors. As at the version date of this document those are Supabase, Cloudflare, Stripe, Resend, Google, and Microsoft, with one further provider published as planned and not currently engaged.

Advance notice and objection. The Processor will inform the Controller of any intended addition or replacement of a subprocessor at least 30 days in advance, giving the Controller the opportunity to object within 14 days of notification on reasonable, good-faith data protection grounds. The Processor will work with the Controller to address an objection, for example by describing additional safeguards or an alternative configuration where one exists. If the objection cannot be resolved, the Controller may terminate the affected subscription without penalty for the unused, prepaid portion of the term.

Obligations imposed on subprocessors. The Processor imposes on each subprocessor, by written contract, data protection obligations no less protective than those in this DPA, including obligations of confidentiality, security measures appropriate to the risk, assistance with data subject requests, breach notification to the Processor without undue delay, and deletion or return of personal data at the end of the engagement. The Processor remains fully liable to the Controller for a subprocessor's performance of those obligations.

Objection at the individual level. For the two OAuth identity providers, Google and Microsoft, an individual user may opt out by choosing the other supported provider at sign-in. No other subprocessor can be selectively disabled for one Controller while the Service continues to be provided, because each delivers a core function: hosting, database, payments, or email.

The full subprocessor list is published, with each provider purpose, data categories, processing location, and published certifications.

Control subprocessors-published-list. Status: Implemented. Demonstrated in the pallas-marketing repository and checked by a release gate.

Disclosure. The 2026-08-30 inventory record names four subprocessors in use: Supabase, Cloudflare, Stripe, and Resend. The published page carries six operative entries, adding Google and Microsoft as OAuth identity providers. Both are accurate about different things: Google and Microsoft receive an authentication assertion at sign-in and store no Pallas customer data. Procurement artifacts should carry all six operative entries, because a buyer assessing identity provider dependency needs to see the two that the four-entry description omits.

Implementation. marketing: src/pages/legal/subprocessors.astro.

Verification. marketing: src/pages/trust.astro, Sub-processors section.

A new subprocessor is published with 30 days advance notice before it processes any customer data.

Control subprocessors-change-notice. Status: Implemented. A commitment the operator makes contractually. No code enforces it.

Disclosure. This is a contractual commitment by the operator, not a technical control. Nothing in either repository enforces the notice period, and no new subprocessor has yet been added under it.

Implementation. marketing: src/pages/legal/subprocessors.astro, Change Notification section.

Certifications listed against each subprocessor are those published by that provider.

Control subprocessors-certifications-are-theirs. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.

Disclosure. Pallas holds no SOC 2, ISO 27001, or PCI DSS certification of its own. Every certification named on the subprocessor list belongs to the named provider and covers that provider infrastructure, not the Pallas application built on top of it. A procurement document must never present a subprocessor certification in a way that could be read as a Pallas certification. This is the single most consequential misreading available in this inventory, so the distinction is stated wherever the certifications appear.

Implementation. marketing: src/pages/legal/subprocessors.astro.

Pallas operates no data centre, server cage, office server room, or other facility in which institutional data is held. Physical and environmental security for every system that holds institutional data belongs to Supabase and Cloudflare.

Control physical-security-subprocessor-operated. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.

Disclosure. Every physical-security control a questionnaire asks about, including staffed facilities, cages, enclosing barriers, redundant and tested power, cooling, fire suppression, and carrier diversity, belongs to Supabase or Cloudflare. Pallas can neither demonstrate nor test any of them, and does not inherit them as its own controls. Answers to physical-security questions state reliance and name the provider. A buyer who needs evidence should read the provider audit reports; a Pallas assertion is not evidence of a provider control.

Implementation. vendor: Supabase and Cloudflare published data centre physical and environmental security

programmes.

Verification. marketing: src/pages/trust.astro, subprocessor list with per-provider processing locations.

Pallas operates no network perimeter, no host, and no network monitoring of its own. Edge filtering, DDoS mitigation, and TLS termination are Cloudflare functions, and the database network boundary is operated by Supabase.

Control network-perimeter-subprocessor-operated. Status: Implemented. A statement by a subprocessor. Pallas configures and relies on it; no Pallas code implements it and neither Pallas repository can demonstrate it.

Disclosure. Pallas runs no firewall of its own, no network intrusion detection or prevention system, and no host-based agent, because it operates no hosts. Stateful filtering, DDoS mitigation, and TLS termination happen at the Cloudflare edge and are answered as provider-operated. Intrusion detection, intrusion prevention tuned by Pallas, next-generation persistent threat monitoring, and 24x7 intrusion monitoring are answered no: no such capability is operated by Pallas or configured by Pallas at a provider. A buyer requiring a monitored perimeter under vendor control, or a staffed security operations centre, should treat this as an open item.

Implementation. vendor: Cloudflare edge network, web application firewall and DDoS mitigation; Supabase managed database network boundary.

Verification. marketing: workers/, whose Workers execute on the Cloudflare edge rather than on any Pallas-operated host.

9. Personal data breach notification

The Processor will notify the Controller without undue delay, and in any event within 72 hours of becoming aware, of a personal data breach affecting Controller Personal Data, and will provide the information reasonably necessary to allow the Controller to meet its own obligations under Articles 33 and 34. The clock starts on awareness, not on confirmation.

Australia. Where the Australian Privacy Act 1988 applies, the Processor will assist the Controller in assessing a suspected eligible data breach within the 30 days the Notifiable Data Breaches scheme allows for that assessment, and, where the Controller forms the belief that an eligible data breach has occurred, in notifying affected individuals and the Office of the Australian Information Commissioner as soon as practicable thereafter. The 30 days is the assessment window, not a notification allowance; the Processor's own notice to the Controller remains bound by the 72-hour commitment above.

Quebec. Where the Act respecting the protection of personal information in the private sector, as amended by Law 25, applies, the Processor will assist the Controller in notifying the Commission d'acces a l'information and affected individuals of any confidentiality incident that presents a risk of serious injury, and in maintaining the required incident register.

What the Processor can and cannot promise here. Every row below is an operator commitment rather than a technical control, and one of them is partial. A Controller should capture the notification window as a contractual term and should understand which leg of the cascade the Processor does not control.

Pallas notifies affected customers without undue delay and in any event within 72 hours of becoming aware of a breach affecting their data. The clock starts on awareness, not on confirmation.

Control incident-72-hour-notification. Status: Implemented. A commitment the operator makes contractually. No code

enforces it.

Disclosure. This is a contractual commitment. No technical control enforces it and no artifact in either repository can demonstrate it has been met, because no breach has occurred. A buyer should treat it as a term to be captured in the Data Processing Agreement rather than as a verified capability.

Implementation. marketing: src/pages/trust.astro, Breach commitments, Customer notification; marketing: src/pages/legal/dpa.astro.

Authority notification follows the jurisdiction: Ireland DPC, UK ICO, and EU supervisory authorities within 72 hours; Canada OPC under PIPEDA; Australia OAIC under the Notifiable Data Breaches scheme.

Control incident-authority-notification. Status: Implemented. A commitment the operator makes contractually. No code enforces it.

Disclosure. A contractual and regulatory commitment rather than a technical control. Pallas has not been required to make such a notification, so there is no track record to evidence.

Implementation. marketing: src/pages/trust.astro, Breach commitments, Authority notification.

Worker observability and logging are enabled on every deployed Worker so that failures surface rather than passing silently.

Control incident-detection-monitoring. Status: Partial. Demonstrated in the pallas-marketing repository and checked by a release gate.

Disclosure. Observability is enabled and logs are collected. There is no security information and event management system, no alerting pipeline, no on-call rotation, and no documented incident response runbook with named roles. Detection today means an operator reading the Cloudflare dashboard. Any questionnaire asking about 24 by 7 monitoring, SIEM, or a tested incident response plan must be answered no. Pallas is operated by a single operator and a procurement document that implied a staffed security operations centre would be false.

Implementation. marketing: workers/pallas-scan-email/wrangler.toml and workers/pallas-email-gate/wrangler.toml, observability and observability.logs enabled; app: All six pallas-app Workers, observability enabled in wrangler.toml.

Verification. marketing: workers/pallas-email-gate/wrangler.toml observability comment.

A full technical postmortem is provided to affected customers within 30 days.

Control incident-post-incident-report. Status: Implemented. A commitment the operator makes contractually. No code enforces it.

Implementation. marketing: src/pages/trust.astro, Breach commitments, Post-incident report.

A subprocessor breach triggers the same 72 hour notification cascade, from the subprocessor to Pallas to the customer.

Control incident-subprocessor-cascade. Status: Partial. A commitment the operator makes contractually. No code enforces it.

Disclosure. The Pallas leg of this cascade is a commitment Pallas controls. The subprocessor leg depends on each provider notifying Pallas, which is governed by that provider terms and is outside Pallas control. The end-to-end 72 hours is therefore contingent on subprocessor performance, and a buyer requiring a guaranteed end-to-end window

should raise it during Enterprise contracting rather than rely on this statement.

Implementation. marketing: src/pages/trust.astro, Breach commitments.

Pallas does not publish a cyber liability insurance policy or a certificate of cover.

Control insurance-no-published-cover. Status: Not implemented. A commitment the operator makes contractually. No code enforces it.

Disclosure. No cyber liability certificate is published. A questionnaire field asking for evidence of cover is answered as not published rather than left blank or answered from intent, and where a contract requires evidence of cover the matter is addressed in the agreement during procurement.

Verification. marketing: src/pages/trust.astro, Cyber liability insurance, status Not yet formalized.

10. Return and deletion of Controller Personal Data

On termination of the Service, the Controller may export its data during the 30-day post-cancellation period. After that period the Processor will delete or return Controller Personal Data in accordance with the Controller's configured retention policies and Applicable Data Protection Law, except where retention is required by law.

Two exceptions apply and are stated rather than buried. First, audit records carry a seven-year retention floor that the Controller cannot configure below, because an audit trail that a party can shorten on demand is not an audit trail. Second, erasure of an individual user does not erase the organisation content that user created, because that content belongs to the organisation as a separate controller with its own retention obligation; the identity link is severed and the audit trail keeps an attribution snapshot.

Retention is configured per organisation, with defaults of 365 days for scan data, 7 days for uploaded source files, 730 days for reports, and 2555 days (seven years) for the audit log.

Control retention-per-org-policies. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_retention_policies, migration 002:374.

Verification. app: uq_pallas_retention_org, migration 002:385.

A scheduled job applies retention policies daily at 03:00 UTC for every organisation with automatic deletion enabled.

Control retention-daily-sweep. Status: Implemented. True of the migration file set. Live database state is not provable from source, and the operator pre-check that resolves it is named in the disclosure.

Disclosure. The job is scheduled by a DO block that catches its own exception, so a database on which pg_cron was not enabled at migration time has no job and the migration still reported success. Whether the job exists and is running on the live database is an operator check against the cron.job and cron.job_run_details tables. This repository can show what was intended to be scheduled; it cannot show what is scheduled.

Implementation. app: pallas-daily-retention, '0 3 * * *', scheduled at migration 021:238; app: pallas_apply_retention_policies, migration 021:197, rebuilt 029:629, migration 041, and migration 042.

Verification. app: supabase/migrations/042_restore_retention_sweeps_and_gaps.sql.

Archived audit events past their retention window are purged nightly at 04:00 UTC, and each purge run is logged.

Control retention-audit-purge. Status: Implemented. True of the migration file set. Live database state is not provable from source, and the operator pre-check that resolves it is named in the disclosure.

Disclosure. Scheduled by the same self-catching DO block pattern as the 03:00 UTC sweep. Existence of the live job is an operator check against cron.job.

Implementation. app: pallas_nightly_retention_purge, '0 4 * * *', scheduled at migration 034:581; app: pallas_purge_expired_retained_audit_events, migration 034:460; app: pallas_retention_purge_log, migration 034:405, RLS enabled 034:416.

Verification. app: supabase/migrations/034_enforce_finding_limit_and_retention_purge.sql section 9.

Uploaded source files are purged from Storage daily at 05:00 UTC by a Supabase Edge Function invoked from the database.

Control retention-source-file-purge. Status: Partial. True of the migration file set. Live database state is not provable from source, and the operator pre-check that resolves it is named in the disclosure.

Disclosure. This job does no work unless two database settings are present: app.settings.functions_base_url and app.settings.cron_secret. When either is missing the job raises a warning and returns, so source files are NOT purged and nothing fails loudly. Migration 035:619 states this in the job body. The operator must confirm both settings are set on the live database before this control is claimed to a buyer. Separately, the 2026-08-30 inventory record describes this credential as being held in Supabase Vault under the name pallas_enforce_retention_bearer. No file in either repository references Vault or that name; the mechanism on disk is a Supabase Edge Function secret named CRON_SECRET plus a database setting named app.settings.cron_secret. The inventory description and the file set disagree, and the file set is what these repositories can show.

Implementation. app: pallas_daily_source_file_retention, '0 5 * * *', scheduled at migration 035:609; app: supabase/functions/enforce-retention/index.ts.

Verification. app: supabase/functions/enforce-retention/index.ts:239 to :252.

Retention settings cause actual deletion rather than being stored as a preference. The sweep issues DELETE statements against the tables it governs.

Control retention-deletion-actually-deletes. Status: Implemented. Backed by a file in the pallas-app repository, verified read-only on 2026-08-30, not gated from the marketing repository.

Implementation. app: pallas_apply_retention_policies, migration 042.

Verification. app: migration 041, which enumerates the three code paths that lawfully remove audit rows.

11. International transfers

Where processing involves the transfer of personal data out of the European Economic Area, the parties agree that the European Commission's 2021 Standard Contractual Clauses, and specifically Module Two, controller to processor, are incorporated into this DPA by reference and apply to those transfers. The docking

clause, the Processor's role as data importer, and the option selections default to those that provide the highest protection to data subjects.

For transfers out of the United Kingdom, the UK International Data Transfer Agreement, or the UK Addendum to the EU Standard Contractual Clauses, applies. For transfers out of Switzerland, the Standard Contractual Clauses apply with the amendments recognised by the Swiss Federal Data Protection and Information Commissioner. In each case the parties apply supplementary measures as appropriate following a transfer risk assessment.

The full Standard Contractual Clause text and the Transfer Impact Assessment covering the subprocessor chain, encryption controls, contractual safeguards, and jurisdictional posture are published at pallas.lonia.ai/trust.

12. Audit rights

The Processor will make available to the Controller the information necessary to demonstrate compliance with Article 28 and will allow for and contribute to audits, including inspections, conducted by the Controller or an auditor mandated by the Controller, subject to reasonable confidentiality and scheduling arrangements.

In practice the Processor expects most audit requests to be satisfied by four things it can supply without an on-site visit: this DPA and its Control Inventory citations; the published security whitepaper, which renders the same inventory in full; the subprocessors' own SOC 2 Type II reports, which the Processor holds under NDA and can route a request for; and the output of the specific operator pre-checks named in the disclosure text of every control whose evidence basis is live unverifiable. Where a Controller requires more, the Processor will agree a scope in writing.

The Processor does not hold an audit report of its own, and an audit right in this section is not a substitute for one.

13. Liability

The liability of each party under this DPA is subject to the limitations and exclusions of liability set out in the main agreement. Nothing in this DPA limits either party's liability for a matter that cannot be limited under Applicable Data Protection Law, including liability to a data subject under Article 82 of the GDPR.

Where the Processor engages a subprocessor, the Processor remains fully liable to the Controller for the subprocessor's performance of its data protection obligations, as stated in Section 8.

14. Term, termination, and order of precedence

This DPA takes effect on the effective date of the main agreement and continues for as long as the Processor processes Controller Personal Data. Sections 9, 10, 12, and 13 survive termination to the extent necessary to give them effect.

In the event of a conflict, the order of precedence is: the Standard Contractual Clauses where they apply; this DPA; the main agreement; and any other document referenced by any of them.

15. Contact

Questions about this DPA, requests for the executed version, and subprocessor change notification subscriptions all go to support@lonia.ai. Include the legal entity name of the Controller and the jurisdiction of its establishment.

Security questions that are not specific to this DPA go to support@lonia.ai.

16. Execution

Processor. Lonia AI

- Signature: _____
- Name: _____
- Title: _____
- Date: _____

Controller. _____

- Signature: _____
- Name: _____
- Title: _____
- Date: _____

The signature lines above are reproduced so that a reviewer can see the shape of the instrument they would be asked to sign. This render is the public template; the executed version is issued once terms are agreed, on request at support@lonia.ai.

Appendix A. Controls cited

This document cites 55 of the 84 rows in the Pallas control inventory. Each is listed below with the status and evidence basis it carried at the 2026-08-30 inventory pass. A row absent from this list is not claimed by this document.

- **auth-oauth-only** (Authentication). Implemented.
- **auth-profile-trigger-fail-safe** (Authentication). Implemented.
- **auth-session-jwt** (Authentication). Implemented.
- **auth-no-browser-storage-for-entitlements** (Authentication). Implemented.
- **tenant-rls-every-table** (Multi-tenancy isolation). Implemented.
- **tenant-org-scoping-helpers** (Multi-tenancy isolation). Implemented.
- **tenant-storage-per-org** (Multi-tenancy isolation). Implemented.
- **tenant-service-role-bypass-disclosed** (Multi-tenancy isolation). Implemented.
- **audit-sha256-ledger** (Tamper-evident audit chain). Implemented.
- **audit-ledger-append-only** (Tamper-evident audit chain). Implemented.
- **audit-lawful-redaction-recorded** (Tamper-evident audit chain). Implemented.

- **audit-seven-year-floor** (Tamper-evident audit chain). Implemented.
- **audit-actor-attribution-survives-erasure** (Tamper-evident audit chain). Implemented.
- **gdpr-art15-art20-user-export** (Data subject rights). Implemented.
- **gdpr-art17-user-erasure** (Data subject rights). Implemented.
- **gdpr-controller-split-documented** (Data subject rights). Implemented.
- **gdpr-org-dsar-export** (Data subject rights). Implemented.
- **gdpr-org-deletion** (Data subject rights). Implemented.
- **gdpr-rights-service-role-only** (Data subject rights). Implemented.
- **retention-per-org-policies** (Retention and deletion). Implemented.
- **retention-daily-sweep** (Retention and deletion). Implemented.
- **retention-audit-purge** (Retention and deletion). Implemented.
- **retention-source-file-purge** (Retention and deletion). Partial.
- **retention-deletion-actually-deletes** (Retention and deletion). Implemented.
- **encryption-at-rest** (Encryption). Implemented.
- **encryption-in-transit** (Encryption). Implemented.
- **encryption-no-secrets-in-repo** (Encryption). Implemented.
- **encryption-member-prose-provider-held-key** (Encryption). Implemented.
- **access-org-roles** (Access control). Implemented.
- **access-last-admin-guard** (Access control). Implemented.
- **access-privilege-loss-revokes-invites** (Access control). Implemented.
- **access-rate-limiting-authenticated** (Access control). Implemented.
- **appsec-ssrf-guard** (Application security). Implemented.
- **appsec-response-cap** (Application security). Implemented.
- **appsec-sandboxed-rendering** (Application security). Implemented.
- **appsec-output-escaping** (Application security). Implemented.
- **appsec-release-gate-fails-closed** (Application security). Implemented.
- **data-classes-enumerated** (Data classes and processing location). Implemented.
- **data-location-default-us** (Data classes and processing location). Implemented.
- **data-location-eu-option** (Data classes and processing location). Not implemented.
- **subprocessors-published-list** (Subprocessors). Implemented.
- **subprocessors-change-notice** (Subprocessors). Implemented.
- **subprocessors-certifications-are-theirs** (Subprocessors). Implemented.
- **physical-security-subprocessor-operated** (Subprocessors). Implemented.
- **network-perimeter-subprocessor-operated** (Subprocessors). Implemented.
- **incident-72-hour-notification** (Incident response). Implemented.
- **incident-authority-notification** (Incident response). Implemented.
- **incident-detection-monitoring** (Incident response). Partial.
- **incident-post-incident-report** (Incident response). Implemented.
- **incident-subprocessor-cascade** (Incident response). Partial.
- **insurance-no-published-cover** (Incident response). Not implemented.

- **vuln-advisory-triage-artifact** (Vulnerability management). Implemented.
- **vuln-no-penetration-test** (Vulnerability management). Not implemented.
- **vuln-no-independent-certification** (Vulnerability management). Not implemented.
- **vuln-version-control-and-review** (Vulnerability management). Implemented.