

How Aldra protects participant information

This paper sets out the security controls that are built into Aldra and running today, the obligations they exist to satisfy, how each one is verified on every change, and what Aldra does not yet claim. It is written for the people who have to sign off on a system that holds participant information: providers, their auditors and their privacy officers. It is general product information, not legal advice.

Version 1.0, draft · September 2026

Draft

This paper is a draft. The controls it describes are implemented and running, and each is traceable to the code that implements it, but the paper itself has not completed internal review or independent assessment. Treat it as a statement of how Aldra is built, not as a certified assurance report, and check with us before relying on it in a tender, an audit or a procurement decision.

Contents

1. What this is built to satisfy
2. How the duties map onto the platform
3. Where Aldra runs
4. Signing in
5. Separation between organisations is enforced by the database
6. Access to a participant is held by a named person
7. The request boundary
8. Files
9. Links shared outside Aldra
10. AI and automation
11. The audit trail
12. How we know the controls still hold
13. Backup, recovery and incidents

14. Keeping and removing information
15. What this document does not claim

What this is built to satisfy

Aldra holds participant information for registered NDIS providers and allied health practices. That work is regulated, and the controls in this paper exist to let a provider meet duties it already carries rather than to demonstrate engineering for its own sake.

- National Disability Insurance Scheme Act 2013 (Cth) and the rules made under it, including the NDIS (Incident Management and Reportable Incidents) Rules 2018 and the NDIS (Complaints Management and Resolution) Rules 2018, which set the records a registered provider must keep and for how long.
- NDIS Practice Standards and Quality Indicators. The rights and responsibilities outcome on privacy and dignity is that “each participant accesses supports that respect and protect their dignity and right to privacy”. The information management outcome requires each participant’s information to be identifiable, accurately recorded, current and confidential, accessible to the participant, and used only by the workers who need it.
- NDIS Code of Conduct, under which workers must respect the privacy of people with disability.
- Privacy Act 1988 (Cth) and the Australian Privacy Principles. Health information is sensitive information. APP 6 limits use and disclosure to the purpose of collection, APP 11 requires reasonable steps to protect information and to destroy or de-identify it once it is no longer needed, and APP 12 and APP 13 give a person rights of access and correction.
- The Notifiable Data Breaches scheme, which requires an eligible breach to be assessed and, where the threshold is met, notified to the Office of the Australian Information Commissioner and to the people affected.

How the duties map onto the platform

Each of those duties lands on a specific control rather than on a policy statement.

- Confidential, used only by the workers who need it: access is granted to a named person for a named participant and enforced by the database on every query, not by hiding things in the interface.
- Identifiable and accurately recorded: records carry their author, the responsible practitioner, the provider accountable for them and their revision history.
- Accessible to the participant, and rights of access and correction: information is held in one Case per participant that survives a change of practitioner or provider, and can be read and exported by the organisation that is accountable for it.

- Records that must be kept: deletion of a person's account removes their identity and access without erasing the organisation's regulated records, which are held under least-privilege access until they can lawfully be destroyed or de-identified.
 - Assessing a breach: an append-only audit trail that cannot be revised, plus an incident process that establishes tenant, participant and time scope and treats uncertain scope as affected.
 - Reasonable steps to protect: the rest of this paper.
-

Where Aldra runs

Aldra is built for Australian disability and allied health providers and runs on Australian infrastructure. The people who process information on our behalf are a short, controlled list, and adding one is an architectural decision recorded in the codebase rather than a quiet operational change.

- The application, the database and the file store are hosted in Australia. The file store is self-hosted S3-compatible object storage rather than a third-party consumer service.
 - All requests go through an Australian sovereign gateway running in Australian data centres.
 - Outbound email uses an Australian sovereign transport, chosen on the same basis as the AI gateway.
 - Test and production are physically separate environments with separate data directories, separate credentials and separate backups.
-

Signing in

Aldra does not hold passwords. Identity is handled by a dedicated OpenID Connect provider, and the application only ever sees a signed token that it verifies for itself on every request.

- Passkeys are the default. Registration is passwordless, credentials are discoverable, and user verification is required. One-time codes remain available only as a migration fallback and are not treated as phishing-resistant.
- Only ES256 and RS256 are accepted, duplicate registration of the same authenticator is refused, and the passkey is bound to the sign-in hostname.
- The web and iOS applications cannot exchange a username and password for a token; that grant type is disabled on both clients.
- Every request re-verifies the token signature, issuer and audience against the provider's published signing keys. A token that carries no valid workspace claim is rejected outright rather than falling back to any default scope.

- One audit record is written for each sign-in session, which is what the administration console reports as last login.

Separation between organisations is enforced by the database

The boundary between one organisation and another is not a filter in application code that a future mistake could forget. It is a set of PostgreSQL row level security policies, and the application has no way around them.

- The API connects as a role that is not a superuser and cannot bypass row level security.
- Every table holding organisation data has row level security enabled and forced. Forced means the rule applies even to the table's owner.
- Policies compare the row's organisation against the organisation pinned to the current transaction. With no context set the comparison yields nothing and the query returns no rows, so the default is to deny.
- That context is set as a transaction-local setting in the first statement of the transaction, so scope cannot leak between requests that share a pooled database connection.
- There is one data-access module and no unscoped query path. Every read and write states whose data it is opening.

Access to a participant is held by a named person

Belonging to an organisation does not by itself open a participant's Case. Each person holds their own link to each participant they work with, and the same database policies that separate organisations also apply that link.

- A person sees only the participants they are linked to. Owners and administrators of an organisation see the participants in their own organisation, which is what makes them accountable for it.
- The check runs inside the policies on participants, files, plans, invoices, timesheets and every other participant-scoped table, so it cannot be satisfied on one screen and missed on another.
- Removing a link takes effect on the next request, because the check runs in the database when the query runs rather than being baked into the signed-in session.

The request boundary

Everything that reaches the API is checked before a handler sees it, and everything that leaves it is checked before a client sees it.

- Every route validates its input against a shared schema package. Anything that does not match is refused with a validation response before any work begins.
 - Security response headers are applied to every response.
 - Cross-origin access to the authenticated API is allow-listed to Aldra's own applications. The endpoints behind shared links are separately open by necessity and are rate limited instead, at 120 reads and 30 writes a minute per client per capability, with the edge proxy as the outer layer.
 - An unexpected error returns a generic message and a request identifier. Stack traces, database errors and internal identifiers do not reach the client.
-

Files

File bytes never travel through the API. The API authorises an upload or a download and records the metadata; the browser talks to the object store directly using a short-lived signed URL.

- Upload URLs last five minutes and download URLs two minutes.
 - Object keys are assembled from validated identifiers and a validated folder name, so a path cannot be manipulated to reach another organisation's file.
 - Only inert content types — images, PDF and plain text — are ever served for viewing in the browser. Everything else is forced to download, so an uploaded file cannot run as a page on the storage origin. Active document types are refused at upload.
 - Every file has a maximum size enforced in the shared contract.
-

Links shared outside Aldra

Some things have to be readable by someone with no Aldra account: an invoice sent to a plan manager, a document sent out for review, a behaviour observation form given to a support worker. Those links are bearer credentials and are treated as such.

- A link carries 256 bits of random token. Only the SHA-256 hash of the token is stored, so a leaked database row cannot be turned back into a working link.
 - Links expire on a bounded schedule and can be revoked, which rotates the stored hash.
 - The endpoint behind a link returns only the fields the recipient needs. It does not expose internal identifiers, other participants or anything about the organisation.
 - Tokens are stripped from the request log before it is written, so access to logs never becomes access to the records.
-

AI and automation

AI in Aldra drafts records, extracts from documents and answers questions about a Case. It runs as a queued task through one gateway, and no provider call happens without a budget check and an audit record.

- The gateway record keeps the provider, the model, token counts, latency and whether the call succeeded. It does not store the prompt or the response.
 - The assistant reads through the same scoped access path as everything else, so it cannot surface a Case the person asking is not entitled to open.
 - Document extraction runs in a private processing service that holds no database credentials and no tenant credentials. It downloads only from configured object-store hosts, using a short-lived URL the worker creates. A URL supplied by the browser is never accepted.
 - Extraction returns a proposal. Structured writes to a participant's record stay behind a person confirming them.
-

The audit trail

Actions that matter are recorded where they cannot be quietly revised, including by us.

- Audit records are append-only. The database grants allow insert and select and nothing else, and a trigger raises an error on any attempt to update, delete or truncate.
 - The record is written inside the same transaction as the action it describes, so the two commit or roll back together and there is no window where one exists without the other.
 - The actor is resolved from the verified token subject inside the SQL statement. A client cannot claim to be someone else.
 - Audit records sit behind the same organisation policies as everything else, so the trail is your organisation's to read.
 - The AI gateway keeps its own append-only record on the same terms.
-

How we know the controls still hold

Controls that are only checked by hand drift. Every change to Aldra goes through the same pipeline before it can be merged, and several of those jobs exist specifically to catch a security regression rather than a bug.

- The pipeline runs formatting, linting, a strict type check with no suppressions permitted, contract and design-system tests, the API suite, the web suite, a database suite, a secret scan, a filesystem vulnerability and misconfiguration scan, a build contract check, and end-to-end journeys through the real application.
- The database job runs the cross-organisation isolation suite against a real PostgreSQL instance rather than a mock, alongside every other integration suite.
- One of those is a structural gate: any table carrying an organisation column must have row level security enabled, forced, and at least one policy. A change that adds a table and forgets its policy fails the build rather than shipping.
- The secret scan reads the full commit history, on the basis that a credential is only gone once it is gone from every commit. The dependency scan blocks high and critical advisories that have a published fix, and falls back to an independent scan of the exact production dependency graph if the advisory service is unavailable, rather than treating an outage as a pass.
- Scanner exceptions must name a specific rule and component, carry a rationale, an accountable owner and an expiry date, and are published in the repository. Blanket package, directory or severity ignores are not allowed, and an expired exception fails review.
- The platform is reviewed adversarially against the running production configuration, across payments, the unauthenticated surface, authentication and tenancy, file upload and storage, and secrets and configuration. Findings are recorded and closed; the restriction on which content types may be viewed in the browser came out of one such review.

Backup, recovery and incidents

Availability and integrity are planned for rather than assumed.

- Backups run nightly on a timer that catches up a missed night rather than skipping it. They cover the application database, the identity database and the file store, with a SHA-256 manifest over both artefacts.
- The recovery point is a day or better and the restore is a single command on the host, measured in minutes. Test and production back up and restore independently of each other.
- Incidents are graded. Cross-tenant disclosure, credential compromise and destructive data loss are treated as the highest severity and responded to immediately; a suspected one is treated as real until disproved.
- Response preserves evidence. Audit records are never deleted or rewritten during an incident, and where the scope is uncertain it is treated as affected rather than assumed clear.
- A written review follows within two business days covering timeline, root cause, customer impact and corrective actions with owners and dates, and adds a regression test whenever the failure can be reproduced safely.

- Credentials are rotated on a schedule and immediately after suspected disclosure, replaced with least-privilege equivalents, and verified as revoked. Secret values are never placed in the codebase, tickets, chat or incident reports.
-

Keeping and removing information

What Aldra holds, and for how long, follows your organisation's obligations rather than a convenient default. Deleting an Aldra account removes that person's identity and their access; it is not authority to erase an organisation's participant, clinical, financial, incident, complaint or audit records, some of which must be kept for a set period under NDIS and other Australian obligations. Those are held with least-privilege access until they can lawfully be destroyed or de-identified. The privacy page sets this out in full.

What this document does not claim

No system is completely secure, and a security document that says otherwise is not worth reading. These are the edges as they stand.

- The platform data use agreement that organisations accept is a draft pending qualified Australian legal and privacy review.
 - Step-up to phishing-resistant authentication for high-risk actions such as granting access or exporting records is defined as a contract and configured in the identity provider, but is not yet enforced by the application itself.
 - Named on-call ownership, notification channels and regulatory and privacy contacts are configured per environment as part of go-live rather than being a property of the software.
 - Independent third-party certification has not been sought. The assurance described here is internal and automated, plus adversarial review.
-

Sources

The instruments referred to above are published by the NDIS Quality and Safeguards Commission, the Office of the Australian Information Commissioner and the Federal Register of Legislation. Check the current wording there rather than relying on this summary.

[NDIS Practice Standards and Quality Indicators](#)

[Privacy Act 1988 \(Cth\)](#)

Reporting a security issue

If you believe you have found a vulnerability in Aldra, or you suspect information has been accessed or disclosed without authority, write to us. Please include enough detail to reproduce the issue and do not include participant information in the report.

admin@aldra.au

Issued by

This paper is issued as a draft on behalf of the organisation that builds Aldra.

Shane Wheller

Co-founder and Chief Executive Officer
AlliedOS Pty Ltd, creator of Aldra.au

Issued 6 September 2026 Status Draft
