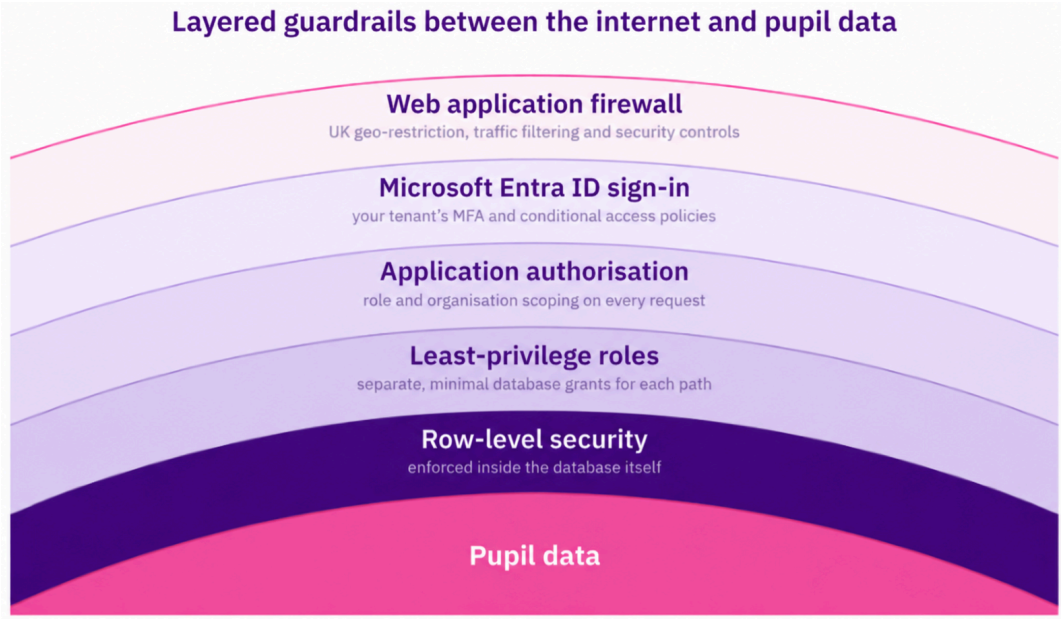
# Security Overview

Trellis · v2.3 · October 2026

*Every part of Trellis is built in the awareness that data we process contains sensitive information about vulnerable children. We have engineered our platform to protect this data with rigorous, multi-layered security controls, ensuring that pupil privacy is never compromised. Each organisation's data is separated by row-level security inside the database itself, so one school's staff cannot reach another school's records, and row-level filtering runs per user within a school.*

## Identity and access




*Every layer must let the request through. A bug in one layer is caught by the next.*

Staff sign in through our Microsoft Entra ID tenant. The Trellis application itself never sees, stores or transmits a password. Where sign-in is federated via SSO, MFA and other authentication policies are decided in your tenant and Trellis inherits those decisions automatically. Until federation is in place, accounts are hosted in the Trellis tenant and we enforce multi-factor authentication on every account, including administrators. SSO will be rolled out through the autumn term.

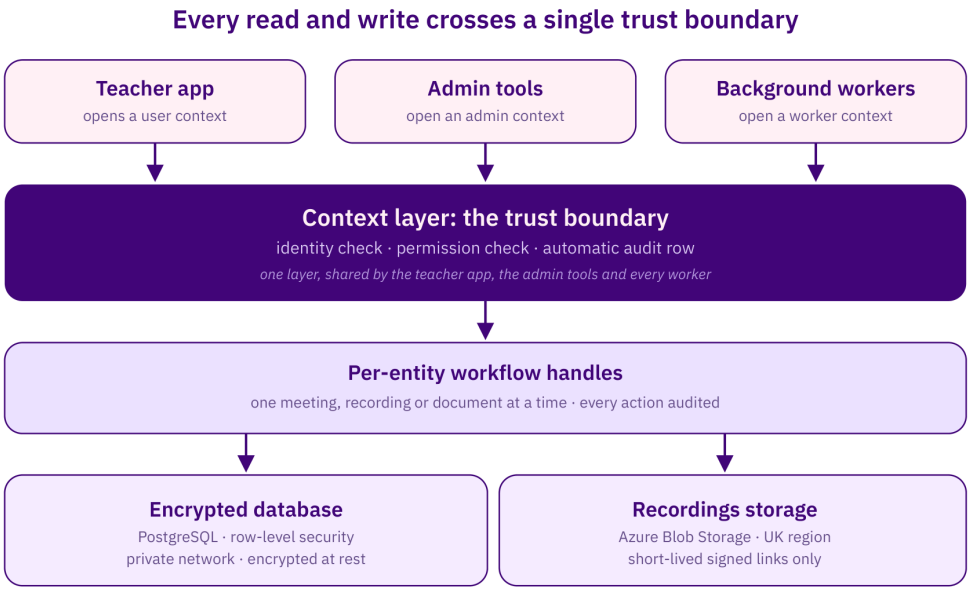
Entra External ID stores staff identity data in the EU (no UK tenant option). It does not hold pupil data, recordings, transcripts or documents.

A web application firewall sits in front of every request. Users can access the Trellis sign-in page from UK networks, including home broadband. Access from outside the UK is blocked unless the organisation has registered its IP ranges with Trellis. Additional security checks and rate limiting may apply to requests from unlisted IP addresses. Trellis strongly recommends that organisations maintain an up-to-date list of their egress IP addresses.

The internal admin tools and the database have no public route at all. They live on a private network reachable only through an audited, time-restricted VPN. Use of that route is logged against a named operator and a stated reason;

there is no anonymous standing access for anyone, including our own engineers. Recordings storage is never browsed directly: every read and write happens through short-lived signed links that the application issues one request at a time.

## The context system: every action is checked and recorded




*Identity, permission checks and the audit logger live in one place that every request passes through.*

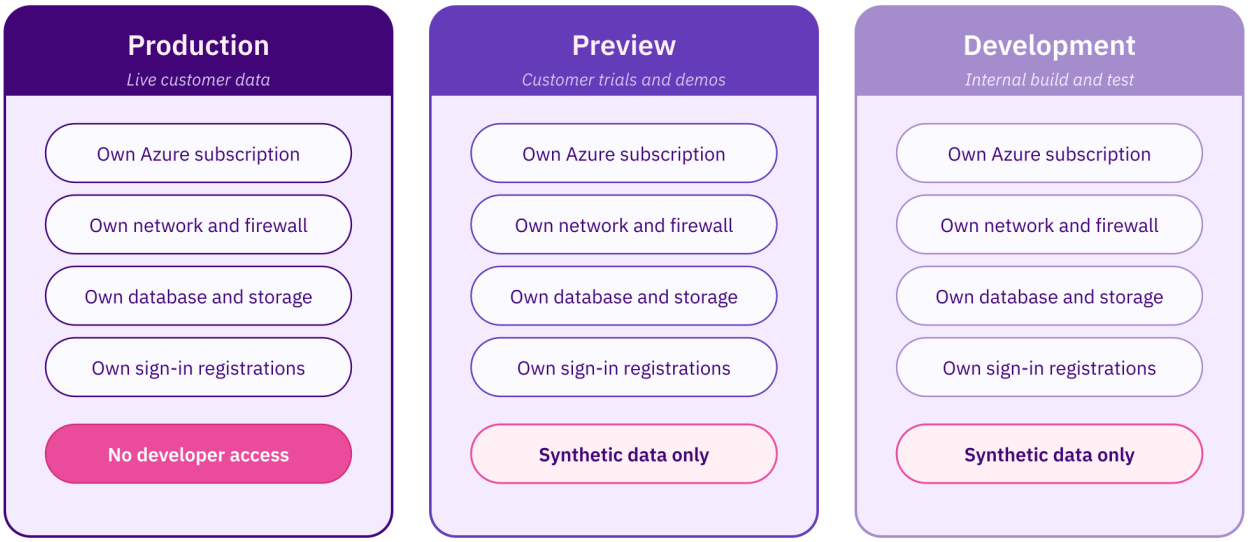
To prevent human error and ensure consistent coverage, permission checks and audit logging are not left to manual, per-feature implementation. Trellis solves the problem once, in a shared trust-boundary layer that every component goes through to reach storage: one place where identity is established, permissions are checked and the audit row is written, rather than a convention each feature has to remember.

A teacher's request opens a user context, which sets a session variable carrying that teacher's identity. Row-level security inside PostgreSQL filters every subsequent query against that variable, so that even a buggy SQL statement cannot return another teacher's row. The database is the last line of defence, not the only line.

Background workers open a worker context and administrators open an admin context. Each context stamps its own actor type on every audit row, so it is always possible to tell whether a row was written by a teacher, an administrator, a background worker or the platform itself. Beneath the contexts, the database schema grants each path only the operations it needs. The audit writer is the strictest example: it holds a write-only grant on the audit log, so the code that records history is itself unable to read or rewrite it.

## Environment isolation

### Three environments, three subscriptions, no cross-talk



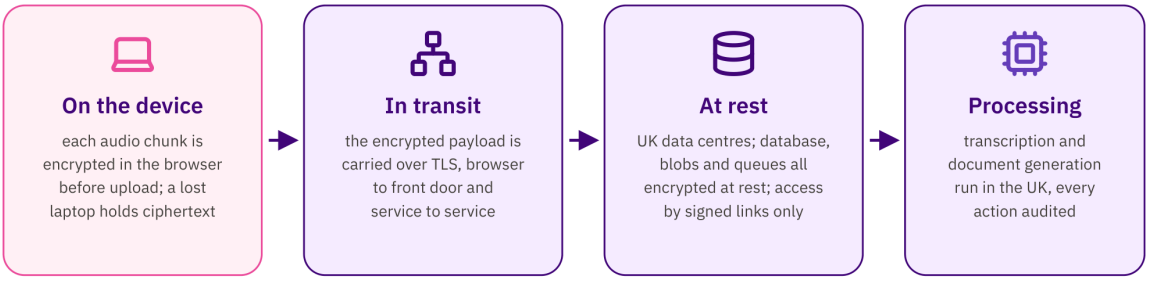

Production, customer preview and internal development each occupy a separate Azure subscription, with separate virtual networks, separate databases, separate storage accounts and separate Microsoft Entra application registrations. Infrastructure is held in Terraform and reviewed before each change applies. There is no shared admin console where a misclick in one environment could change another.

Developers do not have access to production data. Bug investigation happens in the preview environment, which runs the same software against fabricated data drawn from a published library of mock cases. The production database is reachable only through the audited ‘break-glass’ route, and only when an incident genuinely requires it.

Every change ships through automated testing. Unit tests cover the smaller pieces of logic, integration tests exercise the database and storage layers against real Postgres and real blob containers, and end-to-end tests drive a browser through the full recording-to-document pipeline. A deployment that fails any of these checks does not reach production. The gate is enforced in the pipeline rather than by convention; the narrow emergency route around it is itself audited and reviewed after the fact.

## Encryption, malware scanning and residency

### A recording is encrypted before it leaves the device




*A device that goes missing mid-meeting holds nothing readable; inside the platform, access is issued one signed link at a time.*

Encryption is layered rather than relied on at a single point. Audio is captured in short chunks, and each chunk is encrypted on the teacher's device (with a key issued per user for that recording) before it is written to the browser's local store or uploaded. A laptop that is lost, stolen or shared mid-meeting holds only ciphertext, and the upload payload stays protected independently of the connection carrying it. Every link between every browser, every internal service and the front door runs over TLS. Audio is decrypted only once it is inside the platform, where the database, the storage account and the message bus all encrypt at rest in Microsoft's UK data centres, and access is granted request by request through short-lived signed links.

Every file that reaches Trellis, whether uploaded documents or the audio itself, is scanned for malware at the storage layer by Microsoft Defender, and the verdict is recorded against the file. The document pipeline is gated on that verdict: a recording is not assembled and an uploaded document is not processed until scanning marks the file clean, and a malicious verdict stops the workflow and flags it for review. This adds a second, independent opinion to the application's own controls on which file types are accepted, which are enforced in the browser and again at the API.

Trellis content (pupil and meeting data, including the application database, recordings, transcripts and generated documents) is stored at rest and processed in Microsoft Azure UK South, with backups in a second UK region (UK West). Staff identity and authentication data (name, work email, unique ID, and sign in and sign out logs) is held by Microsoft Entra External ID in the EU; no pupil data sits there. Under the UK’s adequacy arrangements for the EEA, that EU hosting is not a restricted transfer.

## Sub-processors and data processing

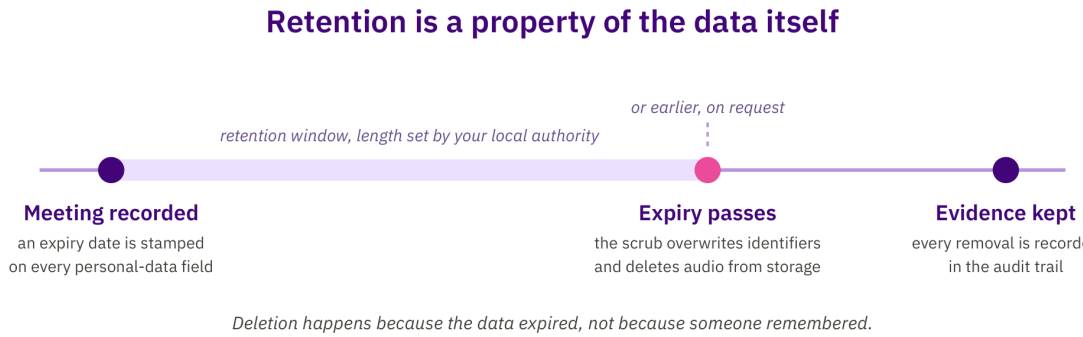
Trellis uses Microsoft Ireland Operations Limited as a sub-processor to host the application, database and recordings storage, and to provide speech transcription and document generation, within Microsoft Azure. Trellis also uses separate service providers for its own internal business administration. These services do not process Customer Data on behalf of schools.

## Audit and accountability

Every meaningful action writes a row to an append-only audit log. Meaningful actions include: a teacher signing in and opening a recording, an administrator updating a school, a background worker assembling a recording, the retention sweep clearing an expired record. Each row carries the actor, the action, the entity, the IP address, the user agent and the request trace identifier, which lets a single customer report be tied back to the application logs from the same request.

The audit trail is treated as compliance evidence first and a debugging tool second. It is written through a dedicated database role that can add rows but never read or change them, and if the originating database transaction rolls back, the failure row is written on a separate connection meaning the trail still records the attempt. Anomalous patterns (failed permission checks, rejected uploads, actions that did not complete) land in the same log and are reviewed on a regular cadence. Audit data is retained for the lifetime of the meeting and seven years afterwards.

# Retention and deletion




*Deletion happens because the data expired, not because someone remembered.*

Each personal-data record carries an expiry date that is computed from the retention period the owning local authority sets at onboarding. A background scrub checks for expired records every minute of every day: when an expiry passes, the identifying fields are overwritten, the related audio is deleted from blob storage, and each removal is itself written to the audit trail. Retention time is a property of the data itself.

Organisations can request earlier deletion at any time; it is carried out by the same scrub mechanism and leaves the same audit evidence. Because every removal is recorded, Trellis can provide a deletion report - what was removed, when, and under which policy - drawn directly from the audit trail, so the local authority has the evidence it needs for its own data-controller records.

## Backups, continuity and supply chain

Our database servers are configured for point-in-time recovery (PITR) within a rolling thirty-day window, with backups geo-replicated to a second UK region. Blob storage (audio, documents) is replicated across availability zones and to the same second UK region, so neither the loss of an Azure availability zone nor a regional failover loses data or moves it outside the United Kingdom. Restores are rehearsed against the preview environment, and the resulting timings feed the continuity plan.

Container images are built in our own pipeline, tagged to the exact release under deployment and pulled from a private registry over authenticated channels. Static security analysis runs on every pull request, and a change that fails any check cannot merge. Automated dependency monitoring runs weekly across every component and raises vulnerability alerts that are triaged into the next release.

## People and process

Code changes are reviewed by a second engineer before they can be merged. Production deployments require passing automated checks; the override route exists for true emergencies and is itself audited and reviewed after the fact. Engineers with access to customer environments are named, vetted to the Baseline Personnel Security Standard (BPSS) — DBS or Disclosure Scotland — and bound by a written acceptable-use policy. Joiners are onboarded through a single workflow with named approvers, and leavers are removed from every system on the day they leave.

## Independent assurance

| ISO 27001 | Independent Penetration Test | Continuous Controls |
|---|---|---|
| Trellis is certified to ISO 27001 (certificate 271956), covering the production service and supporting administrative systems. | An accredited third-party security firm tests the platform on a regular cadence. Findings are tracked to closure on a shared register and re-validated at the following test | Static analysis runs on every change, dependency monitoring runs across every component and infrastructure changes are reviewed before they apply. Production deploys request all checks to pass. |

Copyright © Trellis Education Ltd 2025–2026