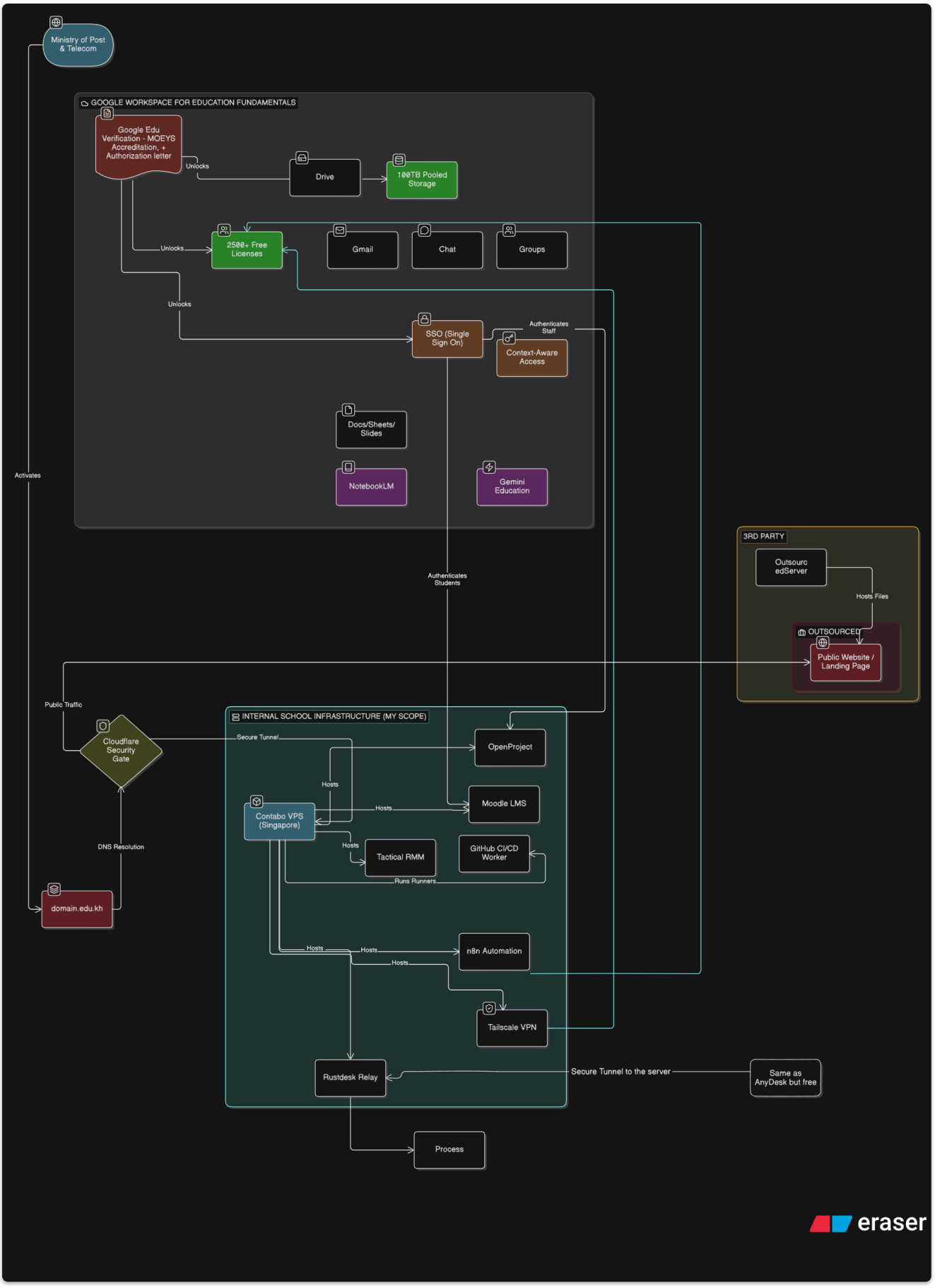
Education IAM & Infrastructure Architecture

Planning Identity Before Applications with Google Workspace for Education

Project Overview

This was a greenfield infrastructure and identity architecture project for a planned education organisation.



Note

The project did not progress beyond the initial domain-registration stage under my tenure, but the architecture was designed before deployment so that identity,

authentication, DNS, public services, internal systems, and administrative access would have clearly defined dependencies from the beginning.

The central design question was:

> *How does identity flow through the system, and where does that trust begin and end?*

Rather than starting with individual applications, I mapped the infrastructure relationships that needed to exist before students, teachers, or administrators could safely use them.

The Challenge

A new education organisation needs much more than a website and a learning-management system.

Before users can be provisioned, the organisation needs a trusted institutional identity.

Before applications can use Single Sign-On, an identity provider must exist.

Before educational cloud licensing can be established, the institution and its domain must satisfy provider verification requirements.

And before internal systems are exposed to users, public-facing and administrative infrastructure need clearly separated trust boundaries.

The architecture therefore had to answer questions such as:

- What establishes the institution's digital identity?
- Which system should be authoritative for users?
- Where should authentication take place?
- Which services should be public?
- Which administrative systems should remain private?
- How should applications inherit identity and access decisions?
- What happens when individual services are replaced?

Identity Before Applications

A common failure mode in greenfield deployments is allowing every application to maintain its own user database.

That creates:

- duplicate accounts
- multiple password stores
- inconsistent authentication policies
- fragmented onboarding and offboarding
- duplicated permission management
- unnecessary administrative overhead

The proposed architecture instead used Google Workspace for Education as the authoritative identity provider.

Applications would consume that identity using federated authentication such as OpenID Connect rather than creating independent identity systems.

This centralises:

- user lifecycle management
- organisational units
- group membership
- authentication policy
- Multi-Factor Authentication
- account suspension and offboarding

Applications therefore do not need to independently determine who a user is.

They trust an identity decision that has already been made upstream.

Designing the Trust Chain

The infrastructure was designed as a dependency graph rather than a collection of servers.

At a simplified level, the trust chain was:

Institutional recognition
↓
Educational domain
↓
Verified Google Workspace for Education tenant
↓
User identity and authentication
↓
Federated applications
↓
Private operational infrastructure

This matters because every component inherits trust from something above it.

For example:

- infrastructure trusts the private administrative network
- the private network trusts the identity provider
- the identity provider trusts the verified institutional domain
- the domain ultimately derives legitimacy from the organisation itself

By the time an administrator reaches an internal service, several independent trust decisions have already occurred.

Separating Public and Internal Systems

Another important architectural decision was separating public-facing infrastructure from operational systems.

A school website and an internal learning or administration platform have fundamentally different trust models.

The public website must be accessible to anyone.

Internal services such as:

- learning platforms
- project-management systems
- automation services
- remote administration
- infrastructure management

should not automatically share the same exposure.

Separating the two environments reduces attack surface and allows them to fail independently.

A compromised public website should not provide a direct path into internal administration.

Likewise, maintenance of internal systems should not unnecessarily affect the public site.

Planned Infrastructure

The architecture considered several operational components, each with a defined role:

- Google Workspace for Education — authoritative identity provider

- Cloudflare — DNS and external infrastructure controls
- Moodle — learning platform
- OpenProject — internal project management
- n8n — workflow automation
- GitHub Actions — deployment automation
- Tailscale — private administrative network
- RustDesk / remote-management tooling — technical support
- Containerised infrastructure — application hosting and service isolation

The individual products were not the most important design decision.

Their relationships were.

Each component needed a clear answer to:

- What does this system trust?
- What trusts this system?
- Should this service be publicly reachable?
- How is the user authenticated?
- What happens if this component fails?
- Can this product later be replaced without redesigning the entire environment?

Designing for Replacement

Technology products change much faster than organisational identity.

That means infrastructure should not be designed around a particular application surviving indefinitely.

The architecture deliberately placed identity above applications because applications can be replaced while users and organisational roles persist.

Similarly:

- public and operational systems remain separate regardless of hosting platform
- administrative access remains private regardless of remote-access product
- applications consume central identity regardless of application vendor
- deployment processes can change without changing the trust model

The dependency graph is therefore more durable than the specific software used to implement it.

Educational Licensing as an Infrastructure Dependency

An unusual part of the project was recognising that administrative and institutional requirements were themselves technical dependencies.

Eligibility for an educational domain and verification for education-specific cloud services needed to occur before the planned identity architecture could be established.

The proposed Google Workspace for Education licensing model also represented a substantial potential saving compared with equivalent commercial Workspace licensing.

This meant accreditation, domain registration, public institutional presence, DNS, identity, and application deployment could not be treated as unrelated tasks.

They formed one dependency chain.

Architecture Principles

Identity first

Applications should consume centrally managed identity rather than creating independent account systems.

Trust should be explicit

Every service should have a clear understanding of what it trusts and why.

Public does not mean internal

Public-facing services and operational infrastructure should have different exposure and security assumptions.

Private administration

Infrastructure-management interfaces should not be unnecessarily exposed to the public Internet.

Lifecycle over login

Identity design must account for onboarding, role changes, suspension, offboarding, and permission changes, not simply authentication.

Replaceable applications

Applications should be replaceable without rebuilding the institution's identity architecture.

Dependencies before products

Good architecture begins by understanding relationships between systems rather than choosing software first.

My Role

I designed the planning architecture and dependency model for the proposed environment.

The work included:

- identity architecture
- Google Workspace for Education planning
- Single Sign-On strategy
- federated authentication design
- DNS and domain dependencies
- public/private service separation
- administrative network design
- infrastructure trust relationships
- application dependency mapping
- deployment and automation planning

The project was intentionally designed before implementation so that technical dependencies could be understood before they became production problems.

Key Takeaway

Good infrastructure architecture is not primarily about selecting products.

It is about deciding where identity originates, how trust moves through the environment, which systems are allowed to communicate, and which dependencies must exist before anything else can work.

Servers are replaceable.

Applications are replaceable.

The trust relationships between them are much harder to redesign after deployment.