IT Project

Guidance:   
Design of Domain Driven Data Schemas

Version:

0.2

## Purpose

To demonstrate the application of domain driven data schema development towards developing data schemas, using the education sector as an example.

## Synopsis

To reduce unnecessary novelty and system design, development and delivery risks, sector-specific domains should build on more general, reusable domains of human interaction—such as identity, relationships, coordination, planning, and work. By relying on these general patterns, domain modelling becomes more stable and requires only modest effort to extend or customise for a particular sector or policy context.

## Contents

[Purpose 1](#_Toc194904243)

[Synopsis 1](#_Toc194904244)

[Contents 2](#_Toc194904245)

[Introduction 4](#_Toc194904246)

[Background 4](#_Toc194904247)

[Domains 5](#_Toc194904248)

[Demonstration 8](#_Toc194904249)

[Domain & Elements Overview 8](#_Toc194904250)

[System Domain 8](#_Toc194904251)

[Social Domain 9](#_Toc194904252)

[Work Domain 10](#_Toc194904253)

[Coordination Domain 10](#_Toc194904254)

[Artefact Domain 11](#_Toc194904255)

[Support & Funding Domain 12](#_Toc194904256)

[Education Domain 12](#_Toc194904257)

[Demonstration of Use Cases 14](#_Toc194904258)

[Service Design (Curriculums) 14](#_Toc194904259)

[Sector Design (Regions, Communities of Learning) 14](#_Toc194904260)

[Sector Provisioning (Schools) 14](#_Toc194904261)

[Sector Analysis 15](#_Toc194904262)

[Teacher Provisioning and Delivery 15](#_Toc194904263)

[Home Schooling 15](#_Toc194904264)

[Māori Medium Schooling 15](#_Toc194904265)

[Private and International Schools 16](#_Toc194904266)

[Education Journey 16](#_Toc194904267)

[Pre-enrolment 16](#_Toc194904268)

[Multi-school Enrolment 16](#_Toc194904269)

[Attendance and Participation 16](#_Toc194904270)

[Learning and Progress 16](#_Toc194904271)

[Qualifications 17](#_Toc194904272)

[Vocational and Tertiary 17](#_Toc194904273)

[Support 17](#_Toc194904274)

[Edge Cases 17](#_Toc194904275)

[Cross-cutting, Flexible Scenarios: 18](#_Toc194904276)

[Model Query Examples 18](#_Toc194904277)

[Teaching Workforce & Governance 19](#_Toc194904278)

[Learner Participation & Enrolment 19](#_Toc194904279)

[Attendance & Coordination 19](#_Toc194904280)

[Curriculum Coverage & Progress 20](#_Toc194904281)

[Qualification & Assessment Outcomes 20](#_Toc194904282)

[Support & Equity 20](#_Toc194904283)

[Artefacts, Evidence & Moderation 21](#_Toc194904284)

[API Representation 22](#_Toc194904285)

[Conclusion 23](#_Toc194904286)

[Appendices 24](#_Toc194904287)

[Appendix A - Document Information 24](#_Toc194904288)

[Versions 24](#_Toc194904289)

[Images 24](#_Toc194904290)

[Tables 24](#_Toc194904291)

[References 24](#_Toc194904292)

[Review Distribution 24](#_Toc194904293)

[Audience 24](#_Toc194904294)

[Structure 24](#_Toc194904295)

[Diagrams 25](#_Toc194904296)

[Acronyms 25](#_Toc194904297)

[API 25](#_Toc194904298)

[GUI 25](#_Toc194904299)

[ICT 25](#_Toc194904300)

[IT 25](#_Toc194904301)

[UI 25](#_Toc194904302)

[Terms 25](#_Toc194904303)

[Application Programming Interface 25](#_Toc194904304)

[User 25](#_Toc194904305)

[User Interface 25](#_Toc194904306)

# Introduction

This paper presents a conceptual and technical model for representing the data structures necessary to support the full operation of the education sector. It is designed to serve as a foundation for **interchange of data between systems**, including systems operated by the Ministry and those operated by education providers, without prescribing any specific software platform.

The model supports stable, long-lived core entities while allowing APIs and interfaces to evolve to fit sector-specific needs. Its aim is to allow systems to interoperate, adapt to policy and service delivery changes, and reduce the need for wholesale system redevelopment. It draws from contemporary architectural patterns including Domain-Driven Design and Code-First schema development, reflecting both sector requirements and modern software practices.

## Background

Earlier system design prioritised readability of the underlying database schema. Tables were named after business entities—students, schools, teachers—and data was accessed directly. This made simple queries easy for DBAs to go in for maintenance. However it also narrowly locked the system to its original use case. Even modest schema changes often required full system redevelopment.

As service expectations expanded and integration became essential, this model faltered.

The rise of NoSQL reinforced the idea that systems no longer need fixed, queryable structures. Around the same time, ORMs (Object-Relational Mappers) began to abstract database access, making schemas less visible to developers while making the automated output more opaque to DBAs.

Modern systems now treat the database as only an internal concern not accessible to users or even maintenance stakeholders, including DBAs. Instead, APIs are the sole point of interaction.

Additionally, entities within the database are abstracted to permanent roles rather than transient business case specific (e.g. Person or Group and Group Role instead of Student, School, Enrolment). However, *externally*, via APIs, the system exposes Student, Teacher, or School API endpoints—shaped for the *user*, and not the *database*. This is interface-based design: flexible, layered, and resilient to change.

The model presented here demonstrates the fruits of this evolution. Its domains remain abstract internally, but each supports specific, recognisable use cases externally—without locking the system to them. That separation is what makes the architecture sustainable, flexible, without the maintenance cost and lack of evolvability of the more legacy approach of Database-First or Model-First system design.

## Domains

To demonstrate the value of internal abstraction, we define a set of layered domains that together support complex sector-specific use cases—without coupling logic to the underlying datastore. These domains reflect common patterns found across systems, and when composed together, they allow for rich yet stable service architectures.

At the foundation is the System domain, which handles essential infrastructure like user identity, session management, configuration, logging, auditing, and Users. This supports the operational integrity of any service.

Above that sits the Social domain, which models people, groups, roles, and the relationships between them. This layer defines people in general, how they are organised, and how they relate to one another, without assuming a specific context such as education or health. Some People will be associated to Users in the previous System Domain. Both people and groups can be contacted via postal, phone, email communications channels.

The Coordination domain introduces time and participation. Events, schedules, attendance or excuses are all universal constructs that describe how people engage with time-bound or repeatable activities. Coordination is concerned with who is involved, when, and in what capacity.

The Work domain captures action and output—projects, tasks, assessments, outcomes. It models effort planning, execution and its evaluation, whether that be an assignment in a classroom, a task in a workplace, or a service milestone in a government programme.

The Artefacts domain manages the media used to inform Work, as well as capture its ouput in the form of individually or collaboratively developed documents, etc.

Only once these foundational - almost universal - domains are in place do we go on to define a business case domain. In this case, the Education domain. This domain does not reinvent what has already been described, but instead builds lightly upon it: a learner is a person who’s personas have an education profile; a provider is just a group with a provider role; enrolments are simply service subscriptions contextualised to education; courses are structured collections of tasks and events; assessments are evaluated tasks; grades are standardised outcomes.

This layered model allows core abstractions to be reused across domains while providing the flexibility to apply sector-specific terminology and logic where needed. It enables stability in the core, adaptability at the edge, and long-term maintainability for systems that must evolve without constant reinvention.

In education, as for most activities, grouping is essential: teachers and learners are associated to schools, year levels, classes, subject groups, courses, and cohorts. These groupings overlap, change over time, and often but not always exist in hierarchies. The proposed model handles this not through rigid entity types, but through a flexible, typed **Group** construct with the ability to nest, label, and distinguish between **systems** and **logical collectives,** whether long term or temporary and/or ad-hoc.

At its core, every **Group** in the Social domain is defined by its purpose, its required roles, the members who have them, and whether it functions as a **System**. A Group that is a System issues identifiers (e.g.: student numbers) that are internally unique and contextually significant. Examples include:

* A **School** is a *System* ***group***—it assigns local learner IDs and manages enrolments.
* The **National Education System** is a System **group**—it assigns NSNs and maintains sector-wide identifiers.
* A **Business** is a System **group** —it assigns employee numbers and manages business units.
* The **Teachers Council** is a system – it assigns numbers to their members.
* The **National Health System** is a System group – just as the National Education System, it issues unique identifiers to each member of that group.
* Etc. (health system, clinics, military, etc.) all are *systems* versions of groups.

In contrast, many Groups are **non-systems**:

* A **Classroom group**, a **Course**, a **Cohort**, a sports team, or a **Community of Learning (CoL)** may organise people and activities but do not issue persistent identifiers.
* A **Family**, **Friend Group**, or **Study Group** may be stable and ad hoc, but exist only by social agreement or temporary purpose.
* An **artefact development group** isn’t a system – it just provides roles to different Personas to work on documents (Creator, Contributor, Reviewer, Publisher, Maintainer, Consumer, etc.)

To account for this, the model includes:

* Group.IsSystem – indicates whether the group functions as a system that can issue internal identifiers.
* Group.Type – allows semantic distinction (e.g. School, Class, SubjectGroup, Family, Adhoc, Artefact).
* Group.ParentGroupId – allows optional nesting and containment (e.g. Classroom within Course within School).
* GroupRole – links people to groups (e.g. a Person can have a Student or Teacher role in a Maths Level 2A Group).

As will be demonstrated later within the Demonstration of Use Cases section, this model accommodates even complex groupings:

* A **learner** may belong to multiple Groups (e.g. Ed Sector, School ABC, Room 12 Maths, Year 11 Māori, Rugby 2), only some of which are Systems.
* A teacher may belong to multiple Groups as well – multiple schools, multiple professional organisations, peak bodies, etc.
* An education provider will be a system Group, and may exist in multiple contexts—its own system, part of a Kāhui Ako, a reporting structure, a funding collective – as well as host multiple internal Groups (e.g. classes, cohorts, etc).
* **Families/Whānau** are handled as social groups, where relationships may or may not be reciprocal, named, or stable. The Relationship entity connects People within or across Groups, allowing for nuanced modelling: birth parent, caregiver, shared guardian, informal carer, or associated whānau member. Relationships are used everywhere – for example between Teachers and Learners.

**Ad hoc Groups** exist temporarily (e.g. a one-off field trip group) and do not have continuity, persistence, or system behaviour. They’re still useful for attendance, coordination, or communication, but they don’t carry the weight of systemic data structures.

# Demonstration

A demonstration of the above concept entails first defining the distinct domains and their entities, then showing real-world use cases the model can support, followed by example queries that can be run across the model, and finally API representations that present the abstract data in recognisable forms for users. Each of these areas is explored in the sections below: Domain & Elements Overview, Use Case Demonstration, Model Queries, and API Representation.

## Domain & Elements Overview

The following are a list of domains, first base ones, then culminating in a business case domain (e.g., “Education”).

In each domain a number of core entities are chosen to exemplify the domain’s purpose, and support the conversation[[1]](#footnote-2).

## System Domain

This most basic domain is for the runtime needs of technology, irrespective of their business purpose.

This domain is practically universal across most IT systems.

It contains the elements to diagnose, track errors, save system configurations, save system preferences, track sessions, operations, permissions and users – amongst other system specific concerns.

It exists to support application integrity, *beneath* system specific business-level logic.

* Logs / Errors
* System Preferences
* Immutable Configuration & mutable Settings
* Session
* User
* Permissions
* Operation Audits
* Etc (Accounts[[2]](#footnote-3), Categories, Routes, Queues, Workflows, etc.)

## Social Domain

This domain captures persons, their personas and their relationships, as well organisational structure and roles within.

At the core is the distinction between a Person (the human being), a Persona (how that person operates within a system), and their Profiles (domain-specific facets, such as Education, Health, or Employment). Most individuals have only one Persona, which may hold multiple Profiles.

Their Personas are what are invited to have Roles within Groups.

Persons are identified by 3rd party systems they belong to (Identity Providers (IdPs).

Groups represent any organisation or social unit—schools, classrooms, whānau, communities of learning. Some groups are of type “system” (e.g. schools, government agencies), meaning they issue identifiers (e.g. NSNs, school specific ids, etc.). Others are non-systems (e.g. cohorts, families) used for coordination and grouping.

Groups can be nested as needed (e.g. School, Class, Course, Cohort) and typed (e.g. System, Service, Ad-hoc, etc.)

* **Person**: a unique human being.
* **Persona**: how a Person is known within a system.
* **Relationship**: between Personas (*not* Persons).
* **Group**: a collection of people or nested groups. May be a System (e.g. a school) if it issues identifiers, may be a Service (more on that later), or Ad-Hoc temporal group.
* **Group Role**: the single point of linkage between a Persona and a Group. It includes the Role name, start and end dates, status (e.g. active, suspended), and defines permissions. It may draw from a preconfigured Role template or have custom permission overrides.
* **Profile**: domain-specific representation of a Persona (e.g. EducationProfile, HealthProfile). A person who has multiple Personas will have multiple Profiles.

A **Service** is a **Group** with Type = Service. It may or may not be a System that issues identity. People enrol in Services (which is a type of Group Role). Services can be structured, tracked, funded, and evaluated across the same domains as core education delivery. People can be PreEnrolled – another type of Role with different Permissions than the latter Enrolled role.

Examples:

* Year 11 Science → Group of Type = Service, offered by a School
* ESOL Support → Group of Type = Service, offered by a Cluster
* PLD Course → Group of Type = Service, offered by a PLD provider

## Work Domain

Models purposeful effort, tasks, and evaluation.

Work includes tasks and assessments, projects grouping tasks, and outcomes linked to people or artefacts. Assessments are special types of tasks that expect an evaluated outcome. Outcomes may optionally be mapped to curriculum or policy frameworks.

* **Project**: a larger structured effort, typically spanning multiple Tasks and Outcomes, often with a clear goal or timeframe (e.g. a science fair, unit of inquiry, or moderation cycle).
* **Task**: an actionable unit of work undertaken by a person or group, such as completing an exercise, writing a report, or preparing materials.
* **Assignment**: a Task that has been formally issued to a Persona, with expectations for delivery (e.g. homework, staff responsibility).
* **Assessment**: a special form of Task where the output is evaluated against defined criteria. Assessments may be internal or external and are typically linked to CurriculumMilestones.
* **Outcome**: the result of a completed Task or Assessment. It may include a grade, status, or commentary and is optionally linked to a CurriculumMilestone or SupportGoal.

## Coordination Domain

Organises and tracks participation in structured or time-bound activities.

People participate in time-bound events such as classes, meetings, or programmes.

Coordination links Personas to Events through participation records, and tracks attendance and explanations for absence.

This domain supports both coarse-grained structures (e.g. term calendars, exam blocks) and fine-grained scheduling (e.g. PLD workshop scheduling, visits, trips – and even down to classroom periods if required).

It also enables pre-term coordination, such as the linking of Events to Work (lesson plans), and integration with teaching Artefacts.

This domain provides the patterns to organise and observe how people and groups interact with time: events, classes, meetings, attendance, and justification of absence.

* **Event**: a scheduled or recorded occurrence, such as a class, meeting, exam, or intervention. Events serve as temporal anchors for Participation, Tasks, and Attendance.
* **Schedule**: defines a pattern of Events (e.g. weekly timetable, term plan, rotational block). Schedules can be linked to Services, Groups, or institutional calendars.
* **Participation**: links a Persona to an Event, establishing who is expected to attend or contribute.
* **Attendance**: records whether a Persona attended a specific Event.
* **Excuse / Justification**: provides the reason for non-attendance (e.g. illness, leave, discipline).
* **Schedule**: a pattern of Events over time.
* **Participation**: a Persona's involvement in an Event.
* **Attendance**: actual presence or absence.
* **Excuse**: reason for non-attendance.

Appointments (such as learning support sessions, evaluations, or pastoral meetings) are modelled as Tasks that are scheduled via Events and Schedules. These typically involve one or more Participants and may produce Outcomes. No additional entity is required to support appointments—they are handled through coordinated Tasks.

## Artefact Domain

Handles outputs and evidence related to Work.

Artefacts are digital or physical materials produced during tasks or assessments. They are linked to Tasks, Outcomes, and/or directly to Personas, and may include learning materials, evidence, or products. They support moderation, auditing, portfolios, and review.

* **Artefact**: a standalone object representing evidence or output (e.g. a document, video, scanned worksheet, image, or code snippet). Artefacts are durable, referenceable, and may be student- or teacher-generated.
* **MediaReference**: a link to where the Artefact is stored (e.g. a file location, URL, or external media system). Supports future-proofing by decoupling content from storage.
* **ArtefactLink**: connects Artefacts to other entities (e.g. a specific Task, Outcome, Event, or Persona). Enables multiple uses or reuse of the same Artefact across contexts.
* **ArtefactType**: categorises the Artefact (e.g. Written, Audio, Image, Performance, Planning Document, Moderation Sample) to support filtering, validation, and expected format checks.
* **Tags / Metadata**: shared meta-domain that provides additional classification or descriptive context (e.g. subject, year level, language, keywords). Tags are optional but powerful for discovery, search, and analytics.

## Support & Funding Domain

Manages targeted learning support and non-structural funding.

This domain enables tracking of support entitlements, programme delivery, and outcomes without requiring full accounting.

It supports equity-based policy delivery (e.g. ESOL, RTLB, ADHD support), temporary interventions, or needs-based funding at the learner or group level.

* **SupportEntitlement** – defines eligibility (e.g. ESOL, behavioural support)
* **SupportProgram** – structured intervention or funded service
* **SupportAllocation** – actual delivery of support, with start/end and provider
* **SupportOutcome** – optional evaluation or record of impact
* **FundingReference** – optional link to budget codes or external systems
* **TaggedGroupMembership** – allows learner or class-level tagging for support-linked funding

## Education Domain

Specialises other domains for sector-specific interpretation.

Handles enrolment, delivery, curriculum structure, progress tracking, and formal assessment.

This domain provides the education-specific logic layered atop general identity, work, and coordination. It models participation (via Enrolment), learning structure (via Curriculum), evaluation (via Grades and Outcomes), and now, longitudinal tracking (via Progress).

Curriculum defines what is to be achieved through **CurriculumMilestones** and structured via **CurriculumPaths**. These are classified using standard **Codesets** (e.g. SALT), enabling consistency across providers and systems. Assessments are linked to defined types. Outcomes may be formally graded, and all evidence can be linked to learning goals.

Progress allows summarising a learner’s (or teacher’s) movement along a defined curriculum or framework. It does not replace detailed outcomes but gives a usable, high-level picture of educational development.

* **EducationProfile**: a profile to describe a Persona that is going through the education phase.
* **ProviderProfile**: a profile that describes a Group that is of type School (that is a System).
* **Enrolment**: a logical type of membership to Groups of type School (a system).
* **AssessmentType:**
* **Grade:**
* **CurriculumPath :** a sequence of expected learning outcomes.
* **CurriculumMilestone**: a specific goal or standard within a CurriculumPath.
* **CurriculumLink**: connects Work or Outcomes to Milestones.
* **Codeset / Classification**:
* **Progress**:

# Demonstration of Use Cases

These examples show how the model handles identity, enrolment, activity, learning, teaching, and resource management across contexts. Each use case identifies which domain entities are involved, making their function and reuse explicit across contexts.

They include normal cases and known edge cases, showing the model is capable without overfitting.

## Service Design (Curriculums)

* A new national curriculum is published → CurriculumPath created, structured via CurriculumMilestones and classified using Codeset. This forms the foundation for all future tracking of learning progress and assessment alignment.
* A local variant (e.g. kaupapa Māori stream) is defined → additional CurriculumPath with reused or new Milestones. These may align with or diverge from national frameworks while remaining structurally compatible.
* Curriculum milestones are mapped to real-world assessments → CurriculumLink connects Milestones to Outcomes, allowing local assessments to reflect national or provider-level expectations.

## Sector Design (Regions, Communities of Learning)

* A Ministry-defined region is created → Group with Type = Region.
* A Kāhui Ako (Community of Learning) is formed → Group with Type = CoL, schools added via GroupMembership.
* A specialist provider network is established → Group with Type = Cluster or Partnership.

## Sector Provisioning (Schools)

* A new school opens → Group created, Type = School, marked as System. This enables local ID issuance and administrative autonomy.
* The school offers services (e.g. Courses) → Groups of Type = Service nested under School. These are structured to reflect learning offerings, each with their own Enrolments, Schedules, and Events.
* Students enrol → GroupMembership to School and to individual Services. This allows clear tracking of enrolment pathways, service usage, and overlaps across institutions.
* Teacher joins staff → GroupMembership with Role = Teacher, ProviderProfile updated. Teachers become active participants in the delivery of services through Work and Coordination.

## Sector Analysis

* Curriculum progress is analysed across cohorts → Progress entries summarise linked Outcomes against CurriculumPaths.
* Class size ratios calculated → GroupMembership and Role entities analysed per Group.
* School utilisation and regional trends → Groups, Enrolments, and Attendance evaluated over time.
* Demand for new schools identified → derived from Service access patterns, Enrolment growth, and Progress indicators.

## Teacher Provisioning and Delivery

* Teacher completes initial training → Assessment and Outcome recorded, linked to a Teaching CurriculumPath.
* Teacher joins Teachers Council → GroupMembership, Role = Registered Teacher.
* Teacher undertakes PLD → Enrolment in Service of Type = PLD, Outcome and Progress tracked.
* Teacher teaches at multiple schools → GroupMemberships in multiple Groups, Role = Teacher.
* Teacher under review → Outcome of Assessment, Role marked On Hold.
* Teacher stood down → Role status updated, Outcome retained.

## Home Schooling

* Learner is educated at home → EducationProfile exists, GroupMembership to School may be absent or nominal.
* Tasks, Assessments, and Progress still apply → same use of Work, Outcome, and Curriculum domains.

## Māori Medium Schooling

* School follows Te Marautanga o Aotearoa → CurriculumPath aligned to alternate Codeset.
* Students and teachers interact identically → same structures reused.

## Private and International Schools

* School follows IB, Bac, or Oxbridge curriculum → CurriculumPath imported or mapped.
* Outcomes and Progress tracked the same way → local systems link Tasks and Outcomes to Milestones.

## Education Journey

* Learner begins in ECE → GroupMembership in Provider, NSN issued if not already present.
* Transitions to school → EducationProfile reused, new GroupMembership added.

## Pre-enrolment

* Learner accepted to school in future term → Enrolment created in advance with future start date.
* Relationship to school formalised, without triggering funding or scheduling.

## Multi-school Enrolment

* Learner enrolled in both local school and Te Kura → concurrent GroupMemberships.
* Progress, Attendance, and Outcomes tracked per Group.

## Attendance and Participation

* School day is scheduled → Event created via Schedule.
* Learner expected to attend → Participation planned.
* Learner present or absent → Attendance recorded, with optional Excuse.

## Learning and Progress

* Learner completes homework → Task created, Artefact submitted, Outcome recorded. Artefacts might include files, photos, or online content linked directly to work.
* In-school and out-of-school assessments → AssessmentType defines structure, same Outcome model used. Results from formal exams or informal evaluations are tracked consistently.
* Arts, performance, multimedia → Artefacts (e.g. video, document) linked to Task or Outcome. Supports full breadth of creative and practical learning, not limited to text-based outputs.

## Qualifications

* Learner completes NCEA Level 2 → CurriculumPath fulfilled through Outcomes and Milestones.
* Learner completes Oxbridge exam → CurriculumPath mapped from imported Codeset, Outcomes recorded.

## Vocational and Tertiary

* Learner enrols in polytechnic course → GroupMembership in Provider, Enrolment to vocational Service.
* Learner accepted to university → GroupMembership created, EducationProfile reused.
* No change in schema needed between secondary and tertiary use.

## Support

* Learner identified with ESOL needs → SupportEntitlement created, SupportProgram assigned.
* Service delivered via pull-out or in-class model → SupportAllocation linked to learner and provider.
* Progress toward support goal tracked → SupportOutcome recorded, optionally linked to CurriculumMilestone.
* Support linked to funding → FundingReference and TaggedGroupMembership record funding impact.

## Edge Cases

The following are examples of edge cases that the above domains support.

* Learner has multiple Personas across systems → each Persona linked to a single Person, with separate Profiles for education, health, etc. NSNs may differ, identity continuity preserved.
* Learner is re-enrolled under a different name or gender → existing Person reused, new Persona created with updated details, linked to the same EducationProfile.
* Teacher changes roles mid-term (e.g. becomes Acting Principal) → Role on GroupMembership updated, historic role retained for audit.
* Learner is dual-enrolled in a school and a community-based education programme → multiple Enrolments tracked via GroupMembership in two Services, coordinated through Events and shared Outcomes.
* A learner exits the system (expelled, migrated) and returns years later → EducationProfile retained, new GroupMembership created, previous Progress preserved.
* A teacher participates in PLD delivered by their own school → School acts as both Provider and Service, GroupMemberships reflect dual role as Learner and Teacher.

## Cross-cutting, Flexible Scenarios:

* A **Kāhui Ako is created** → Group (not System), schools join via Group-to-Group membership
* A **student's essay is submitted and marked** → Artefact linked to Task, Outcome generated, Grade assigned
* A **school imports curriculum milestones** into its SMS → reads from shared CurriculumPath, applies internal mappings
* A **moderator reviews student work from three schools** → Artefacts pulled via linked AssessmentOutcome, classified, compared
* A **student is home-schooled** → Person with EducationProfile, no School Group, but still assigned Tasks, Outcomes, Artefacts

The model handles edge cases without special accommodation. Home schooling, Māori-medium schooling, suspended or expelled learners, seconded teachers, and multiple school enrolments all fit. No separate logic or exception tables are required.

## Model Query Examples

This section provides examples of how the data model supports querying across different levels of the education system. Queries span product design, provider structure, service participation, staffing, curriculum coverage, funding, support, and learner outcomes.

**Sector Design & Provisioning**

* List all regions (districts) → query all Groups where Type = Region
* List all Communities of Learning (CoLs) → Groups where Type = CoL
* List all schools → Groups where Type = School
* List all services offered by a school → Groups where ParentGroup = [SchoolId] AND Type = Service
* List all schools in a CoL → GroupMemberships where Group = [CoLId]
* List all teachers at a school → GroupMemberships where Group = [SchoolId] AND Role = Teacher
* List all schools that offer physics → Services linked via CurriculumLink to CurriculumMilestone.Subject = Physics
* List all providers offering PLD → Groups where Type = Provider AND offers Services of Type = PLD

### Teaching Workforce & Governance

* List all teachers teaching maths → Teachers assigned to Services or Tasks linked to CurriculumMilestone.Subject = Maths
* List teachers approaching registration renewal → GroupMemberships in Teachers Council with EndDate within 90 days
* Compare number of active teachers in a school vs registered with Teachers Council → count GroupMemberships in School vs Teachers Council
* Identify schools with under-capacity staffing → ratio of Teacher roles to Enrolled learners under threshold
* List all teachers teaching in more than one school → GroupMemberships with Role = Teacher across multiple School Groups

### Learner Participation & Enrolment

* List all learners in a school → GroupMemberships where Group = [SchoolId] AND Role = Student
* List all learners in a CoL → aggregate all GroupMemberships in member schools
* List all learners enrolled in a Service (e.g. Level 3 Chemistry) → Enrolments to Group where Type = Service AND CurriculumMilestone.Subject = Chemistry
* List all dual-enrolled students → Personas with active Enrolments in more than one School-type Group
* Identify learners not enrolled in any school → EducationProfiles with no active GroupMembership in a School-type Group
* Identify new learners requiring NSNs → Personas without GroupMembership in National System

### Attendance & Coordination

* Daily roll check → Attendance records for Event = [Today]
* Learners with excessive unexplained absences → Attendance records where Status = Absent AND Excuse IS NULL
* Participation in national exams → Participation records linked to Events of Type = Exam
* Attendance by Event type (e.g. PLD, class, mentoring) → Attendance grouped by Event.Type

### Curriculum Coverage & Progress

* Learners behind expected milestone → Progress records where Status != Completed AND LastUpdated > expected threshold
* Learners reaching all Milestones in Level 2 Science → filter by CurriculumPath and Progress.Status = Completed
* Teachers delivering content mapped to specific CurriculumMilestone → Tasks/Assessments linked via CurriculumLink
* Artefacts linked to particular Curriculum areas → ArtefactLinks → Outcomes → CurriculumMilestones with subject tags
* Evaluate curriculum coverage by region → cross-reference Services and their CurriculumLinks by Provider region

### Qualification & Assessment Outcomes

* Learners eligible for NCEA Level 2 → CurriculumPath fulfilled via recorded Outcomes
* Learners failed external assessment → AssessmentType = External, Outcome below threshold
* Learners with incomplete Outcomes in enrolled Services → Enrolments without matching Outcomes for assigned Assessments
* Learners achieving a Merit or Excellence grade in Science → Outcomes linked to CurriculumMilestone.Subject = Science with Grade >= Merit

### Support & Equity

* Learners with ESOL entitlement → SupportEntitlement where Program = ESOL
* Support delivered in past month → SupportAllocation where Date within 30 days
* Schools with highest number of learning support needs → aggregate SupportEntitlements per School
* Link support outcome to progress → learners with SupportOutcome mapped to CurriculumMilestones

### Artefacts, Evidence & Moderation

* Learners with missing portfolio evidence → Tasks or Assessments without linked Artefacts
* Artefacts submitted for external moderation → ArtefactLinks where Type = ModerationSample
* Teachers with highest number of assessed Artefacts → count of ArtefactLinks by EvaluatorId

These examples illustrate how the data model enables practical, operational, analytical, and strategic queries across all layers of education delivery.

Conclusion

## API Representation

In an API-first system, internal entities—Person, Group, Task, Outcome, and so on—exist independently of how they are presented to external users. APIs are then layered over these entities to expose business-relevant concepts in language appropriate to the audience.

For example, while the internal schema may define a Group and a Group Role, the API presents a School endpoint that returns learners, staff, calendar, classes, and enrolments—by projecting the correct structures from those abstract entities.

This approach preserves stability in internal logic while making the system usable by non-technical stakeholders. Here are examples of recognisable API endpoints made available through this model:

* /students → Personas with EducationProfiles and active School Group Roles
* /schools → Groups of Type = School
* /schools/{id}/students → Group Roles in a School Group where Role = Student
* /caregivers → Personas with Relationships to learners
* /classes → Groups of Type = Class or Course
* /classes/{id}/schedule → Schedule and Events linked to a Class Group
* /teachers → Personas with Group Roles in School or Service Groups, Role = Teacher
* /teachers/{id}/subjects → CurriculumLinks from Tasks delivered by this teacher
* /colleagues → All Personas in same School Group as current user
* /communities-of-learning → Groups of Type = CoL
* /communities-of-learning/{id}/schools → Member Groups in a CoL
* /events/today → Events occurring today that the current Persona participates in
* /attendance/{studentId} → All Attendance records for a given learner
* /pld/opportunities → Groups of Type = Service, tagged PLD
* /support/{studentId} → SupportEntitlements and Allocations for a learner
* /artefacts/{outcomeId} → Artefacts linked to a specific Outcome
* /curriculum/milestones → CurriculumMilestones across all paths
* /progress/{studentId} → Summary of learner’s Curriculum-linked Outcomes

This approach allows service consumers—whether educators, caregivers, analysts, or policy teams—to work with data in recognisable forms, while the underlying model remains abstract, flexible, and stable.

## Conclusion

Abstraction is essential for operating across a diverse and dynamic education sector—one where thousands of providers vary in structure, approach, curriculum, codeset use, and delivery method. This model enables that diversity without fragmentation.

The shift toward abstraction comes with an important consequence: data specialists must move away from traditional practices of direct database access and instead support environments where APIs become the interface to data. That transition elevates the responsibility of developers to provide robust, expressive APIs that expose abstract entities in precise, usable forms—allowing service consumers, including maintainers and analysts, to interact with data safely, securely, and meaningfully.

Direct access to data is not only risky from a security standpoint—it also tends to anchor the system design around the datastore rather than the logic or business needs it serves. This model reverses that by structuring around purpose and logic first, using internal schemas that can adapt over time, and external interfaces that remain understandable and stable for real-world stakeholders.

This is not a compromise. It is a requirement for long-lived, shared infrastructure in the modern public sector.

Appendices

Appendix A - Document Information

Authors & Collaborators

* Sky Sigal, Solution Architect

### Versions

* 1. Initial Draft

### Images

**No table of figures entries found.**

### Tables

**No table of figures entries found.**

### References

**There are no sources in the current document.**

### Review Distribution

The document was distributed for review as below:

|  |  |
| --- | --- |
| Identity | Notes |
|  |  |
|  |  |
|  |  |

### Audience

The document is technical in nature, but parts are expected to be read and/or validated by a non-technical audience.

### Structure

Where possible, the document structure is guided by either ISO-\* standards or best practice.

### Diagrams

Diagrams are developed for a wide audience. Unless specifically for a technical audience, where the use of industry standard diagram types (ArchiMate, UML, C4), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

### Acronyms

API

: [Application Programming Interface](#Term_ApplicationProgrammingInterface).

GUI

: [Graphical User Interface](#Term_ApplicationProgrammingInterface). A form of [UI](#Acronym_UI).

ICT

: acronym for Information & Communication Technology, the domain of defining Information elements and using technology to automate their communication between entities. [IT](#Acronym_IT) is a subset of ICT.

IT

: acronym for Information, using Technology to automate and facilitate its management.

UI

: User Interface. Contrast with [API](#Acronym_API).

### Terms

Refer to the project’s Glossary.

Application Programming Interface

: an Interface provided for other systems to invoke (as opposed to User Interfaces).

User

: a human user of a system via its UIs.

User Interface

: a system interface intended for use by system users. Most computer system UIs are Graphics User Interfaces ([GUI](#Acronym_GUI)) or Text/Console User Interfaces (TUI).

1. There *are* other supporting entities in each domain, but for the purpose of this document, they are not needed to be listed and/or brought into the discussion. [↑](#footnote-ref-2)
2. In passing, another design change that has occurred over the years is that – to permit collaboration across organisation and consultants, Accounts in SaaS has replaced the older notion of isolated Tenant. [↑](#footnote-ref-3)