Te Rito

Longitudinal Sector Data Hub Integration API  
DRAFT

Version:

0.2

Author:

Sky Sigal, Solution Architect

## Purpose

To describe the objective, outcomes and elements required of an API to transfer current and past educational state information in between varied systems via the use of a sector longitudinal data hub.

## Synopsis

Integration development is often complex in breadth and depth, hence costly to implement. Strategies to reduce the cost include making it simpler, less time consuming and costly for others to implement the integrations. This is achievable by requiring only the simplest of widely used API protocols, removing the requirements to implement the full breadth of the APIs models before it can be used, increasing reuse of more abstract elements as opposed to requiring the use of case specific entities.

Related to that, the length of service of longitudinal data hubs implies being able to communicate with an ever-evolving set of systems, which in all probability cannot all be modified or evolved every time a new source system is integrated with it. To accommodate this reality, the data schema that requires development must be the developed with elements that are the abstracted, reusable, and flexible manner that can be reasonably designed at the time of need.

## Contents

[Description 1](#_Toc154927865)

[Synopsis 1](#_Toc154927866)

[Contents 2](#_Toc154927867)

[Purpose 3](#_Toc154927868)

[Background 3](#_Toc154927869)

[Outcomes [Objectives] 3](#_Toc154927870)

[Options [Considered & Selected] 3](#_Toc154927871)

[Constraints 3](#_Toc154927872)

[Assumptions 3](#_Toc154927873)

[Dependencies 3](#_Toc154927874)

[Decisions 3](#_Toc154927875)

[Deliverables/Outputs 3](#_Toc154927876)

[Heading Level 3 3](#_Toc154927877)

[Heading Level 4 4](#_Toc154927878)

[Appendices 5](#_Toc154927879)

[Appendix A - Document Information 5](#_Toc154927880)

[Versions 5](#_Toc154927881)

[Images 5](#_Toc154927882)

[Tables 5](#_Toc154927883)

[References 5](#_Toc154927884)

[Review Distribution 5](#_Toc154927885)

[Audience 5](#_Toc154927886)

[Structure 5](#_Toc154927887)

[Diagrams 6](#_Toc154927888)

[Terms 6](#_Toc154927889)

## Background

Integrations between systems is often costly to implement due to several factors.

Making it less costly is prudent to improve uptake.

## Objectives

The desired long term effects include:

* The information be usable by an evolving set of services over the full s

## Risks

The development of a datahub that is intended to integrate between many known and unknown systems, over a long duration has many risks. Many of them avoidable.

In no specific order, the following are technical risks that can be addressed.

### Information Model Too tied to Current Use

A relatively common mistake is to design an API that is too specific to a current understanding of the problem. For at least two reasons this is a risk.

For one, whereas a single organisation can develop a common understanding of the problem, having multiple organisations agree becomes exponentially more difficult to achieve.

Secondly, things change – sometimes dramatically -- over time. Even the most certain aspects can change due to cultural and/or political reasons. For example, only 10 years ago, it was inconceivable that a government agency had the capability of defining a person as having any other gender than M, F. X is a much more recent change, and has quickly proven itself to be inadequate to be fit for purpose. Before the pressing need for integrating special character schools, there was absolutely no intention of managing any other school than state schools. Assessment grading has shifted over the decades from percentage based, to A-F based, to standardized to judgement based, to various combinations thereof. For all these changes, and any future ones that cannot be determined beforehand, it is important to expose APIs that are as un-specific as possible, allowing for the widest range of modifiability.

### Information Model too Specific

A model that is too specific -- and not sufficiently abstract -- is a risk. Developing an API for Students and Schools, as opposed to the more abstract Persons and Groups – limits the use of the API to only pedagogical entities.

Note:  
For example, Google Maps APIs don’t return Restaurant or Garage types. It returns collections an abstract Place, of Type Restaurant, Garage, etc. permitting an infinite addition of types without requiring redeploying a later version of the API schema.

### Too Specific Protocols

A long term, externally accessible, [data hub](#Term_DataHub) must account for the reality that Integration protocols become retired for technical reasons or simply fall out fashion and currency.

For example, 20 years ago, CSV was possibly the most prevalent non-hierarchical data exchange format, often reliant on filesharing, and was an accepted inter-system integration pattern. This was later replaced with XML based SOAP over multiple transport protocols before stabilising on mostly HTTP -- just in time to be dropped in favour of the less complicated but more limited JSON based data exchange format. Only for the REST pattern to be sidelined by GraphQL, and the story goes on.

The majority of service consumers -- for example schools, but also internal departments of an enterprise – do not have the expertise or budget to update service clients to keep up with changes to data exchange, encoding or transport protocols done at the server end. A naïve understanding of these real constraints would be a critical issue.

To work around this issue there are only a few options:

* choose a single pattern and continue to use it, possibly for a long after it is no longer fit for purpose on a functional, maintainable and/or security level, or
* design a system that can continue to evolve as the IT industry evolves, while continuing to make available legacy communication and encoding patterns far longer than one would in a controlled homogenous environment.

The first option is hard to get right from both a technical viewpoint as well as business point of view.

For example, SOAP has proven to have most longevity, but it is often too complicated for most organisations – i.e., most service consumers – to do inexpensively and correctly. This results in diminished market penetration.

On the other hand, JSON based REST is not universal and is still evolving, and yet is simple enough that most service consumers can figure it out sufficiently to meet their needs and the cost of updating is less.

Note:  
For the best intersection of longevity versus capability versus simplicity, our current recommendation (Q3/2024) is to use JSON – possibly the most used hierarchical[[1]](#footnote-2) data interchange format -- over OASIS defined OData, as well as the non-standard but more used GraphQL.

### Incorrect Data Modelling

Entity Models commonly make mistakes in way entities are related to each other. An example might be the development of a Student, with a child PedagogicalProfile object – as opposed to a PedagogicalProfile that is linked to a Person.

### APIs Clients cannot be used by all

While APIs is the correct means of integration, not all sector service consumers can implement them at reasonable cost. An example might be a small school who is using Spreadsheets to run their business. Others might be using Microsoft Access, or similar consumer grade software. Developing API clients into these simple systems is not a reasonable expectation.

In such cases the information may require being either one way (only exporting data) or less than optimal solutions to import data (e.g.: importing JSON or CSV files over email, etc.) to import data.

While not optimal, [CSV](#Acronym_CSV) can be used to import/export [denormalised](#Term_Denormalised) hierarchical data.

[API](#Acronym_API) technology stack cannot be implemented by all 3rd parties at reasonable effort and cost.

[API](#Acronym_API) initially designed for primarily one domain are difficult to extend to integrate to accommodate the requirements of other business domains as time progresses.

[API](#Acronym_API) schema design managed by 3rd parties limit the evolution of the [API](#Acronym_API) to meet needs not in line with the strategies or needs of 3rd parties.

### Lack of Additional SDKs

Lack of supporting information: for 3rd parties to develop clients around an [API](#Acronym_API) they require documentation that is current, accurate and clear. It is common to also supply an [Software Development Kit](#Term_SoftwareDevelopmentKit) ([SDK](#Acronym_SDK)) to demonstrate that the functionality exists.

## Outcomes

To facilitate attaining the above defined Outcomes, a few outcomes are proposed.

* The data hub information must be usable over the longest duraction
* The data

## Options [Considered & Selected]

### Use an Already Accepted Integration API

One option is to use an integration standard already in use by most systems.

This is currently not feasible as there is no common standard already in use between sector systems. They vary per system, while also using different protocols, for different purposes.

Note:  
Internally, ELI, FIRST, NSI and ENROL all offers proprietary APIs[[2]](#footnote-3).  
Externally, KAMAR, Edge, Hero, etc. all offer custom APIs[[3]](#footnote-4) while consuming to varying degrees the API end points made available by the just mentioned internal services.

### Use an Already developed Integration Standard

An option is to research for and use an integration standard that is recognised, appropriate, usable and maintained.

#### SIF

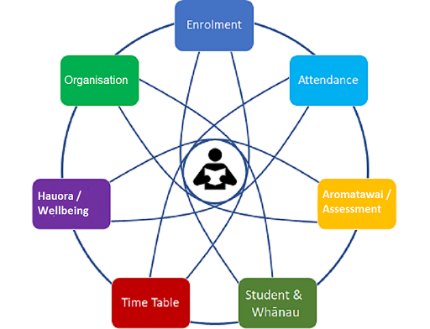


Figure 1: SIF Domains (HL)

SIF is recognised in the education industry.

However, SIF has poor market penetration and is not used beyond mandated circumstances.

The poor market penetration can be attributed to one or more of the following:

* the technical protocols it officially uses (SOAP and XML) are both dated and more complex than more recent options (REST and JSON),
* its data schema is overly specific and prescribed, with both low extensibility and limited ability of being partially and incrementally implement.
* The domain schema is specific to a subset of the full education spectrum – compulsory education - without solutions or guidance on how to extend into other aspects of education (preschool, vocational, tertiary, PLD) or dissemination (via on-prem/in-person, remote, self-guided, etc.). Nor provide convincing and usable solutions for common issues (dual-enrolment, CoLs, demographics, pedagogical, health and finance profiles, etc.)

The above aspects contribute to a high cost to implement, while also being found incomplete to model the whole sector in its many forms.

Attempts have been made to address these issues, leading to different incompatible versions of the SIF protocol (US, AU, GB,NZ) – which defeats the purpose of a common standard.

Note:  
At best, a fractured protocol that is not universally agreed upon or used or implemented provides material to consider when developing a new framework, and solid material to analyse as to why the protocol never gained universal or even wide traction.

### Other standard frameworks

Other frameworks have been developed and gained some level of market adoption. Common to all is keeping the scope of the protocol narrow, focusing on a single aspect of the business needs, leaving it to the systems at both ends of the integration channel to integrate this data with data sourced from other purposes.

In some cases, the protocols avoid defining transport, encoding, or even structure, to permit the widest adoption by whatever means available to each system. For example, one protocol only defines the attributes, leaving it up to implementation to decide in what format to transfer the data (CSV, SOAP, JSON, etc.).

### Development of a Fit for purpose API

The primary purpose is to enable an API for New Zealand’s Education needs, resilient

NZ’s education sector is uniquely different from those of other countries in many ways. It’s dual culture involves multiple curriculums as opposed to a single national ones, schools are managed in a devolved manner, where the national agencies provide support to schools, rather than demand accountability via information.

## Recommended Option

The TODO

## Decisions

## Design Objectives

Capable of mapping people, their relationships, their interactions, events, and the subscriptions to services, plans, assessments, improvements, and attainment of objectives, whether they occur in an educational or not context.

By being capable of mapping the above in a general sense, rather than specific education sense, the schema will provide the resilience necessary to remain useful through a long service lifespan, with the least breaking changes that would require service clients to have to update their code.

## Information Schema

A key objective is to reduce the number of education specific entities – or any other business domain for that matter – by keeping the entities as abstract as possible, reusable in as many future use case areas as possible.

## Domains

### Common

#### Attributes

##### Validity Range

: most relations between entities require additional metadata to define from when to when the relationship is valid. Best practice is the addition of FromUTC/ToUTC values to the table used to join the two entities.

Note:  
The advantages of validity ranges include:

* Decreasing security access risks by ensuring that by default relationships (eg: employment, roles, etc.) have a termination date.
* Increasing maintainability by permitting the development of relationships before they are needed, only making them available to end users at a predefined future date.

Remote Systems Identifiers

State

#### Entities

Remote Systems

[Group]

[Group] Available Roles

[Group] Role

[Group Membership] Application

[Group Role] Invitation

Acceptance

### System Domain

While a system is intended automate business domains, described next, it requires a number of entities that are specific to the domain of IT systems themselves.

##### User

: a user is an [identity](#Entity_Identity) that authenticates itself to a system

Note:  
Not all person’s managed by a system are users (e.g. Grandparents, extended family members, persons that work for service and product suppliers, etc.)

##### User Identifier

: A system [user](#Entity_User) identifies themselves to a system using a 3rd party external system identifier (i.e., an email address within a public domain email system (e.g.: *Gmail*, *outlook.com*, etc.) or organisation specific domain (e.g., *someschool.govt.nz*).   
A system [user](#Entity_User) can be associated to multiple 3rd party digital identifiers.

Note:  
consultants, relief teachers, etc. can belong to multiple organisation and/or school domains. Students may be dual enrolled for learning purposes, or belong to one school, but also be attending a technical centre managed by another school.

##### Role Invitation

: authorised system users invite persons to accept a specified role (admin, tenant admin, user, etc.) within a tenancy within a system.

##### Role Invitation Acceptance

: upon accepting an invitation to a [system role] within a system (and/or a tenant within it, and/or group within it), a user is assigned [system roles](#Entity_SystemRole).

##### Session

: [users](#Entity_User) start or continue interactive sessions, during which they make multiple requests.

##### Request

: a single operation – request and response -- initiated by a [user](#Entity_User) during a [session](#Entity_Session).

##### Security Profile

: a system [user](#Entity_User) has a security profile which is the collection of system roles and ad-hoc permissions associated to the user.

##### [System] Roles

: a system user can apply and be is invited to accept a system role, which is collection of system permissions and system obligations.

##### [System] Permissions

: a system role is a logical collection of included (and optionally excluded) system permissions used to authorise users’ requests.

##### [System] Obligations

: mature systems require users to accept obligations in return for assigning [permissions](#Entity_SystemPermission).

##### [System Role] Application

: a request by a user and/or non-user person to become a system [user](#Entity_User) associated to a system role.

##### [System Role] Invitation

: an invitation sent by an authorised user to a person to first become a system user if not already one, and accept a proposed role.

### Remote System Roles

TODO

### Person Domain

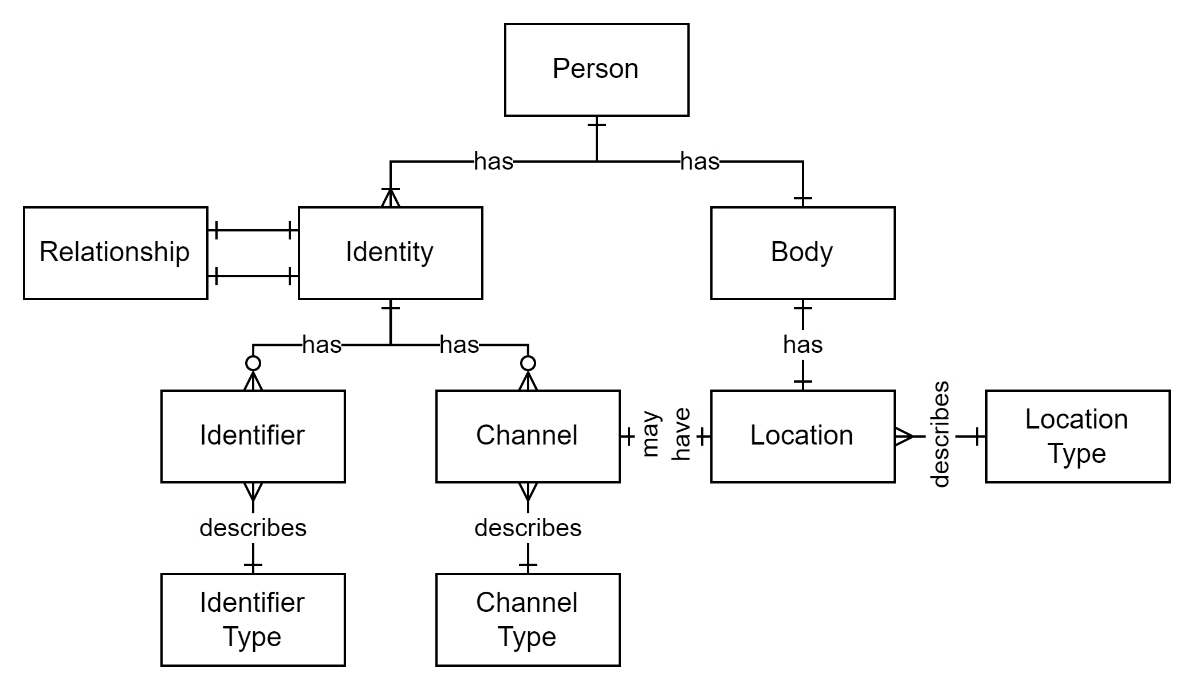


Figure 2: Person Domain

##### Person

: a [person](#Entity_Person) is an entity that has a single [body](#Entity_Body) and multiple [identities](#Entity_Identity).

##### Body

: a [body](#Entity_Body) gains immutable factual (as opposed to contextual) attributes at birth (sex, dob, genetics, and similar).

Note:   
Birth facts can later be used to support the development of a person’s health profile or demographic profile, but are not to be confused with these profiles.

##### Identities

: a [person](#Entity_Person) can develop multiple identities over time. A well-known example is when a person has an identity prior to marriage, and then another when married, changing one of their [identifiers](#Entity_Identifier) (their name). Another example is when a [person](#Entity_Person) transitions gender. Some identities are used as the basis of legal identification, others are just transitional experimentations.

##### Identifiers

: a person’s [identity](#Entity_Identity) can have one or more identifiers associated to it. An Identifier is a combination of [Identity type](#Entity_IdentityType) and value(s). In the case of an [identity type](#Entity_IdentityType) being ‘Name’, the value(s) will be their name(s). If the [Identity type](#Entity_IdentityType) were – for example, “[NSN](#Acronym_NSN)” -- the identity value would be the identity’s [national student number](#Term_NSN).

Note:  
For many reasons[[4]](#footnote-5) it is possible for an identity to have multiple values of the same type. In such cases one should be defined as the preferred identifier of that type.

##### Identity Type

: an [identity](#Entity_Identity) is identified by one or more [types](#Entity_IdentityType) of identifiers. The most common is a person’s name (a social identifier). But it is important to note that names come in many formats, and that it is highly common for people to have many names – from nicknames, to known aliases, to multiple legally recognised names in different jurisdictions in different scripts (a Chinese person may have 4 or more legally recognised names).

In addition to social *names*, an [identity](#Entity_Identity) has other forms of [identifiers](#Entity_Identifier), such as system ids. For example: education, army, health system ids.

They are all valid -- but associated to different use cases.

The [identifier type](#Entity_IdentityType) defines what attributes are collected. For example, a student identifier only requires persisting one value (e.g.: the [National Student Number](#Term_NSN) ([NSN](#Acronym_NSN))), whereas a full name may require persisting multiple attributes (e.g.: the Given, Middle, Surname).

Note: the persisting of data in different fields provides no significant advantage and should not be a design deciding factor[[5]](#footnote-6).

Note:  
an [identity identifier](#Entity_Identifier) is distinct from an entity identifier, described elsewhere within this View.

##### Channels

: beyond face-to-face person to person communication, one communicates with an [identity](#Entity_Identity) via channels they make available. A channel is defined by its [channel type](#Entity_ChannelType) (e.g., “Postal”, “Email”, “Fax”, “Mobile”, “LinkedIn”, “X”, etc.) and Value(s) (“+64 21 123 4567”, etc.), which is a identifier unique to the channel type: i.e., an email address, a phone number, a LinkedIn or Twitter/X handle, etc.

Note:  
in current times, most channels are digital and only require a single value -- whereas historical, -- i.e. pre-digital -- ones such as postal, are composites of multiple values (unity, street, neighbourhood, city, code, country).

##### Channel Type

: the type of [channel](#Entity_Channel) (e.g., Postal, Email, Fax, Mobile, LinkedIn, X, etc.)

##### Location

: locations are the combination of [location type](#Entity_LocationType) and GPS coordinates which can be associated to an organisation’s assets and addresses – but can be associated to a person’s body[[6]](#footnote-7).

Note:   
It is a logical error to define Addresses as locations. PO boxes addresses demonstrate this most evidently: they remain a communication [Channel](#Entity_Channel), although they can be associated to a Location.

##### Location Type

: the customisable type of [location](#Entity_Location) (e.g.: “school”, “home”, “work”, etc.).

## Social Domain

Often poorly distinguished from a person domain, a social domain[[7]](#footnote-8) extends the individual domain, providing the entities to model the relationships of individuals to other individuals and groups and/or organisations.

Individuals belong to multiple groups at one time, where a group can be a family, a school, an organisation, or simply an ad-hoc gathering or circle of friends. Within each group individuals have a role to play. In a family group, they can have a member or responsible role. In a classroom group, they can the role of learner or teacher.

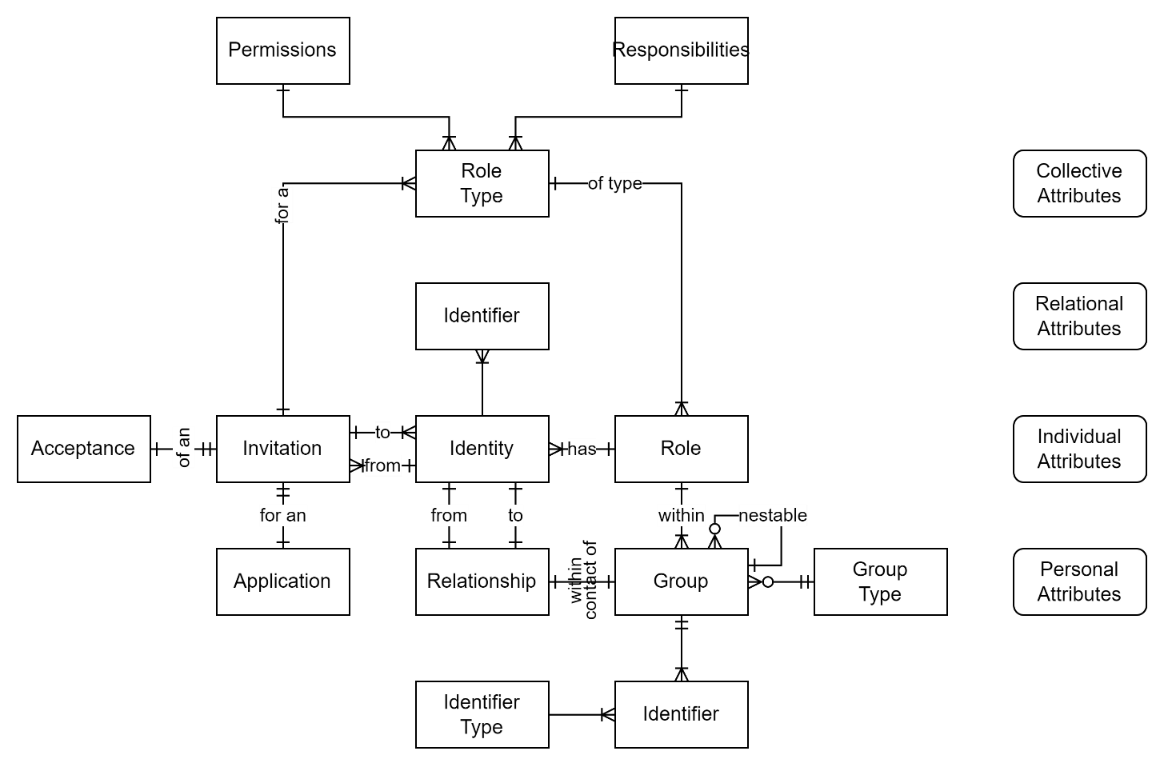


Figure 3: Social Domain Entities & Relationships

##### Group

: a group is a collection of Roles associated to Identities of a Person. Each identity has a Role following [RASCI](#Acronym_RASCI) characteristics (Accountable, Responsible, Informed, Consulted, Ignored, Excluded) even if they have localised labels (Parent, Cousin, restricted access).

[Groups](#Entity_Group) can be nested.

[Groups](#Entity_Group) are not limited to identity roles, and can contain other types of entities (e.g., folders, assets, resources) which the group has a relationship to (generally manages).

[Groups](#Entity_Group) can be logical in that they automatically include identities as Members that have certain qualities (e.g.: members of a school AND are Age 11 are Members of a cohort group).

Note:  
It is important to note that Identities are not members of a Group directly but each have a Role within a Group. In flexible systems, Roles can be assigned to both Identities and Organisations.

##### Group Type

: [Groups](#Entity_Group) cover a tremendous range of ways that [identities](#Entity_Identity) of [persons](#Entity_Person) can collaborate. They can be ad-hoc, family, or any one of many educational group types (e.g., ECL, school, tertiary, [CoL](#Acronym_CoL), MOE Local office, industry training Organisation, etc.) as well as external groups (supplier, consumers, employers, consultants, etc.)

Note:   
It is a design consideration whether to permit different tenancies (e.g.: SMS) to refer to different types, but that have common Code Set values, or to force all to agree to the same type. We argue that flexibility of the first option is more prudent to improve take up by suppliers with the least amount of upfront configuration (the Code set references can always be added later).

##### Group Identifiers

: groups, like identities, can have multiple identifiers in different systems. For example, a Group can have a Business Number Identifier, education provider identifier, religious identifier, etc.

##### Group Type

: groups can be typed (e.g., family, school district, school, class, etc.). The type is used to indicate its purpose (family, class, etc.), but also logically constrains the type of individual and/role that can be associated to it, based on attributes associated to the [identity](#Entity_Identity) (e.g., qualifications).

##### Group Role

: an individual [identity](#Entity_Identity) has a role within a [group](#Entity_Group).

Note:  
even if the relationship is labelled with a localised name (e.g.: “parent”, “teacher”, “friend”, etc.) often aligning closely with the rights and responsibilities of traditional [RASCI](#Acronym_RASCI) roles.

While informed by group roles, [relationships](#Entity_Relationship) remain distinct and supplemental to them.

##### Relationship

: a relationship is a single-direction link between a group member and the group, and/or between members of a group. For example, “brother” within a family group, or “teacher of” within a school’s class group. While relationships can cover both directions (“brothers”, “siblings”, etc.) they are in general less valuable than having two distinct relationships (each being a ‘brother’ to another individual) and their used should be avoided.

Note:  
Relationships are not constrained by group and span between identities in different groups (e.g. from purchasing organisation to supplier organisation, etc.)

##### Relationship Type

: a [relationship](#Entity_Relationship) can be developed between an active (affecting) and passive entity (affected). The effect is generation, destruction, increase, diminution, alteration,

### Profiles

Individuals have multiple profiles, for use in different domains.

For example, most individuals have demographic profiles, health profiles, and education profiles.

##### Education Profile

: an individual’s education profile is composed of portfolio, assessments, reports, and qualifications.

##### [User] Security Profile

: system [users](#Entity_User) – being associated to Identities – have the above profiles, but in addition have a system security profile, which is the user’s collection of roles and permissions.

##### 

### Portfolios

Portfolios are media record stores, which are related to Profiles, but are not subset to them.

Important:  
A person may share a portfolio of their creations with people who the do not share their education profile with).

##### Education Profile Portfolio

: individuals can curate a presentation of work they believe best describes their capabilities.

##### Medical Profile Portfolio

A record keep that permits uploading of media supporting health decisions, which in turn can support funding decisions.

### General Categorisation Domain

Discovery of elements in a system is improved by being able to organise Individuals and entities into logical groupings based on intrinsic or associated attributes.

Note:  
[Categorisation](#Term_Categorisation) differs from [classification](#Term_Classification), and is usually satisfiable with tagging.

##### Tag

: the default way of applying [categorisation](#Term_Categorisation) attributes to entities.   
[Tags](#Entity_Tag) can be nested (e.g.: “Teacher/Physics” can be under “Teacher”).

[Tags](#Entity_Tag) can also be limited to being associated to specific types of entities (e.g., a [person](#Entity_Person) or [identity](#Entity_Identity) probably should not be tagged as a “Farm/Animal”).

##### Tag Type

: a [tag](#Entity_Tag) can be typed. Examples to consider include: “is a”, “quantity”, “has a” “quality/having”, “location”, “when”, “duration”, etc.

##### Grouping

: logical grouping is already covered earlier, under the Social Domain’s group entity.

### Event Scheduling Domain

[Identities](#Entity_Identity) come together physically or virtually for a single event or a series of events.

In an educational context, this is for a single course (e.g., an evening PLD presentation) or series of courses (e.g., a semester’s classes).

The scheduling and resourcing of events may be complex or simple. For example, early learning centres only need to offer one teacher to a room for the whole day, whereas secondary schools will have more complex and fine-grained scheduling needs.

People who come together can be from the same organisation (a school’s learners and teachers) or from various organisations (e.g.: from COLs, for regional PLD, etc.).

Note:  
A [data hub](#Term_DataHub) may only be required to record events that occurred, rather than what may happen (requiring full scheduling). If only the first case, the number of tables required by the hub are fewer, but it is our professional advice to develop the entities required to capture planned events as well. Below are listed entities required of a more mature system that can be used to record both past and planned records.

##### Event

: describes a fixed or recurring set of scheduled occurred or to occur events.

##### Event Type

: Events can be organised for a myriad of reasons. In an educational context, the events can be for Education (e.g.: classes), Informing (e.g.: parent-teacher meetings), etc. The Type can be used to inform Scheduling.

##### Event Schedule

: the set of parameters that describe the algorithm type and associated set of parameters. Includes but is not limited to recording whether the event is a Fixed (single date) or Recurring event, the start date, and if recurring, the nullable end date, the recurrence type (Daily, Weekly, Monthly, Yearly, etc.) and – if not a Daily event – the offset from the start (e.g., 3rd day of the week or month), the desired number of iterations, instructions on how to handle iterations that conflict with omissions (skip, choose the next nearest period, etc.), whether to align to Periods or skip straight to time.

The scheduled iterations are checked against [Bounds](#Entity_EventBoundary).

##### Boundaries

: date ranges which can be defined to either typed as being one that contain (e.g., school terms) or excludes (e.g.: mid-term holidays) scheduled iterations.

##### Boundary Type

: a categorisation of a Boundary as being inclusive or exclusive boundary.

##### Holidays

: a specific type of [Boundary](#Entity_EventBoundary) is a list of known national (e.g.: Mother’s Day) or other holidays that happen regularly, and whether they impact event scheduling. Most national holidays are single days, although not all (e.g.: Statutory days between Christmas and New Year’s). Holidays can be national (specific to Samoa versus New Zealand) as well as regional (e.g., Wellington Anniversary Day versus Auckland’s Anniversary Day). Effort is required to provide the capability of including or excluding Anniversaries from a schedule that it could impact.

##### Prerequisites

: events can only be scheduled if they can be offered when an appropriate quantity of specific resources are available. For example, a teaching event will not be feasible without a teacher, overhead projector, microscopes, etc.).

##### Periods

: slices of time that may or may not align with hours. Schools use periods for classes (e.g.: a school may have 40-minute class duration, with 5 minutes in between for moving between classes, with a larger gap left for lunch). A Period is described by its [period type](#Entity_PeriodType). [Period](#Entity_Period)s are collected together as [Period Set](#Entity_PeriodSet)s.

##### Period Type

: the type of Periods: e.g., may be for classes, or for moving around between classes.

##### Periods Sets

: sets of [Period](#Entity_Period)s used to develop a whole duration (e.g.: a series of [Period](#Entity_Period)s of type “Learning”, followed by a period of type “lunch break”, and then resuming with more Periods of type “learning”. The whole set would be a set that could be titled “Teaching Day”.

Note:  
Resources are described in further detail in the next section.

Event Resource Type

Resource Pools

##### Prerequisites

: the list of prerequisites is dependent on the resources needed. For example, laptop resources may need prior charging. These activities can be allocated to Events for other people (technicians).

##### Assignment

: the concretisation of the reservation of a resource from a resource pool.

### Resource Availability Domain

Events are used to reserve resources.

##### Resource Type

: Resources can be locations, people, devices, media, etc., which in turn can be further subdivided (e.g.: classrooms, gyms, playing fields, etc., teachers, learners, parents, etc., laptops, projectors, etc., books, etc.)

##### Event Resource Pools

: an event requires resources of specific type.

While there might be a preference for a resource, it may not be available (e.g., either already taken by another event or is temporarily out of commission due to a leak). To accommodate flexibility, Events are scheduled against Pools of possible resources, which are reserved based on preference.

##### Resources

: metadata to describe other records (Persons) or external resources (school spaces).

##### Resource Identifiers

: as with persons, a Resource may have multiple identities. The same photo or artefact used as a teaching resource may be described with a Māori name, as well as an English name. The identifier/name may evolve over time. A teaching resource that is a book may also have ISBN numbers. It may be on shared with another organisation, and therefore have two distinct identifiers.

##### Classifiers

: whereas identifiers are generally unique to a single resource, multiple resources can be classified by 3rd party identifiers (e.g.: classified by an external curriculum part’s id).

##### Event Resource Limits

: resources have limits in terms of available times (e.g., business opening hours), quantity of people (e.g., microscopes and laptops can only be used by one or two people at a time, rooms can only accommodate a certain number of people without breaking fire regulations, etc.).

### Enrolment and Attendance Domain

Enrolment is used to reach a minimum to reserve resources. Attendance is a measure of the reserved resources being successfully utilised.

##### Enrolment

: similar to a “subscription” in that one is associated, implying possible service consumption, but making no statement of actual consumption (e.g., attendance). Used as a basis of reserving resources, employment contracts, funding required, etc.

##### Attendance/Absentee Type

: a type to identify whether an identity is still pending attendance, attended, virtually or not, or was absent.

##### Absentee Reason

: if absent, why, The Reason can be submitted beforehand (e.g.: Apologies or School passes) or afterwards.

### Education Domain

The hub’s baseline purpose is to facilitate the collection and transferring between service providers information that facilitates improving learning outcomes for a learner. A key entity managed within the hub is an education profile, associated to a Person’s Identity.

Note:  
the profile is intentionally separate from other profiles (health, etc.), and intentionally not placed as a subset of a Person or identity.

Appendices

Appendix A - Document Information

### Versions

* 1. Initial Draft

### Images

[Figure 1: Person Domain 9](#_Toc172195075)

[Figure 2: Social Domain Entities & Relationships 11](#_Toc172195076)

[Figure 3: ERD Crow's Feet Notation 19](#_Toc172195077)

[Figure 4: Example of ERD Crow's Feet Notation Use 19](#_Toc172195078)

### 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, ERD), is appropriate, diagrams are developed as simple “box & line” monochrome diagrams.

#### Entity Relation Diagrams

Entity Relational Diagrams were developed using Crow’s Feet Notation:

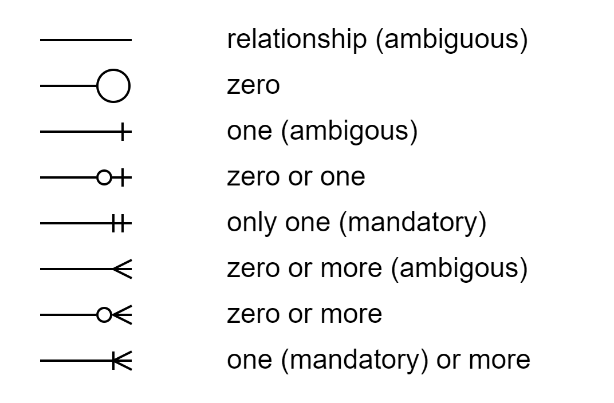


Figure 4: ERD Crow's Feet Notation

An example of how the above notation is used is demonstrated below:

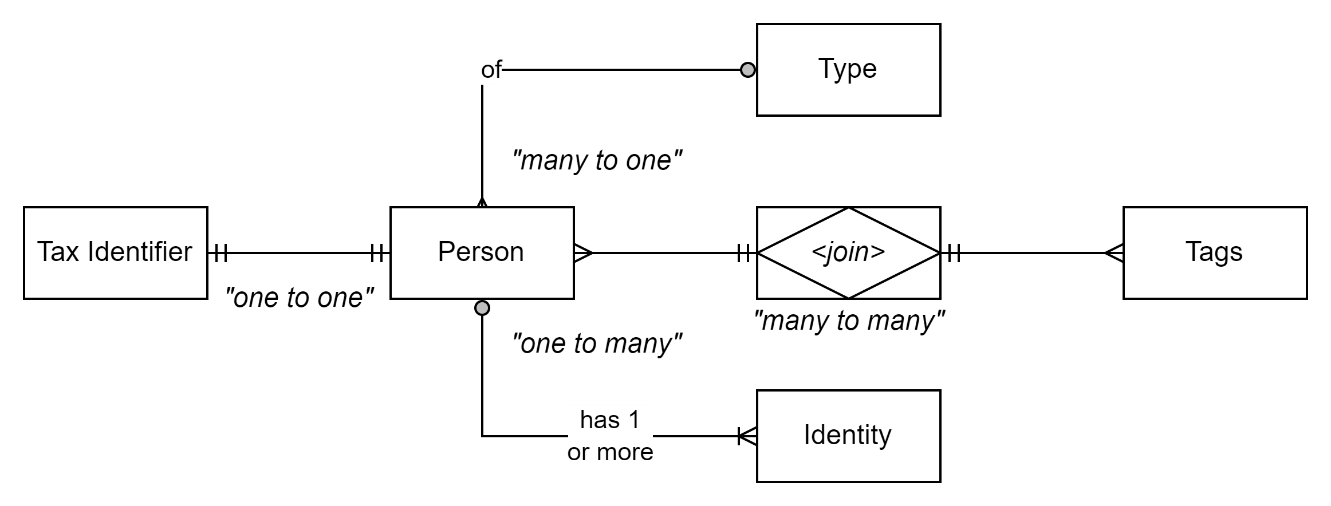


Figure 5: Example of ERD Crow's Feet Notation Use

### Standards

The following are international standards referenced within this document:

##### RFC-2119/8174

: Requirement level

##### RFC-9110

: HTTP

##### RFC-3986

: URI

##### RFC 4180

: CSV

##### ISO 21778

: JSON

##### RFC 9512

: YAML

##### ISO-8601

: Date, Time, Durations

##### ISO 20802

: OData

##### ISO-25010

: Qualities of a System

##### ISO-25012

: Qualities of a System’s Data

##### ISO-25022

: Qualities of the [Experienced] use of a System

##### W3C-SOAP

: SOAP

### Ports

The following are standard interoperability ports referenced directly or indirectly within this document:

##### 80

: HTTP

##### 25

: SMTP

##### 139

: [Windows] Intranet Filesharing over NetBios

##### 443

: HTTP/S

##### 587

: Default mail submission port

##### 445

: [Windows] Intranet Filesharing

##### 1433

: SQL Server

##### 5432/33

: Postgres port

##### 6379

: Redis cache

### Acronyms

Refer to the project’s Glossary.

##### API

: [Application Programming Interface](#Term_ApplicationProgrammingInterface).

##### CoL

: [Community of School](#Term_CommunityOfSchools)

##### CRUD

: Acronym for Create, Read, Update, Delete, the primary verbs for interacting with transactional datastores.

##### CSV

: character separated values. See [RFC 4180](#Standard_RFC_4180).

##### GUI

: [Graphical User Interface](#Term_GraphicalUserInterface).

GUID

: Globally Unique Identifier. See UUID.

##### IT

: acronym for [*Information* management](#Term_InformationManagement), using digital *technology* to automate and facilitate the operations required.

##### ICT

: acronym for [Information & Communication Technology](#Term_InformationCommunicationTechnology).

##### NSN

: the acronym for the [National Student Number](#Term_NSN).

##### RASCII

:

acronym for **R**esponsible (to do the work), **A**ccountable (allocating the work and reporting on its progress to Informed staked holders), **S**upporting (the work) **C**onsulting (on the requirements underpinning the work) stakeholders, and **I**nformed (kept abreast of progress) stakeholders, or **I**gnored/Excluded others.

##### RPC

: Remote Procedure Call

##### SDK

: [Software Development Kit](#Term_SoftwareDevelopmentKit).

##### SOAP

: [Simple Object Access Protocol](#Term_SOAP).

##### UI

: [User Interfaces](#Term_UserInterface) (see [GUI](#Acronym_GUI) and [API](#Acronym_API)).

##### UUID

: Universally Unique Identifier (see GUID).

##### WIMP

: “Windows, Icons, Mouse, Pointer” a windows-based approach to developing [UI](#Acronym_UI)s.

##### XML

: Xtensible Markup Language.

### Terms

##### Archive Datastore

: A separate system [transactional datastore](#Term_TransactionalDatabase) for physically removing records form an [operational database](#Term_OperationalDataStore). *Note: while widely used in the past, the act of physical archiving is now considered an anti-pattern (prefer instead the use of logical state flags over physical removal of records from an operational database).*

##### Application Programming Interface

: a form of interface for submitting and retrieving information, consumed by other systems rather than human users, who use User Interfaces.

##### Classification

: a mutually exclusive form of categorisation. For example, a document can be classified as either top secret or secret, but not both.

##### Categorisation

: a non-exclusive process of describing entities as having similar properties amongst themselves. For example, a person can be categorised (tagged) as both female and in preschool.

##### Community of Schools

: a [CoL](#Acronym_CoL) is a grouping of schools by aligned objectives, often in the same region.

##### Database

: a [relational] [datastore](#Term_Datastore).

##### Data Hub

: a centre of data exchange that provides [API](#Acronym_API) endpoints for services to connect with to provide or consumer data they are permitted to request.

##### Datastore

: a service to persist data in a relational (e.g.: relational database) on non-relational manner (e.g.: no-SQL based document or blob store).

##### Denormalised [Data]

: data is stored in a document or other non-relational datastore, in doing so duplicating common data across multiple entries. See [Normalised data](#Term_Normalised).

##### Entity

: database and system models/objects that possess inherent [system identifiers](#Term_SystemIdentifier), as opposed to [value objects](#Term_ValueObject) which don’t.

##### Goal

: Strategies are developed to deliver them. Contrast with [objective](#Term_Objective)s.

##### Graphical User Interface

: form of [user interface](#Term_UserInterface) used by [human] users to interact a s system to provide and retrieve information managed by the system.

##### 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](#Acronym_ICT).

##### Information Management

: the tasks involved in the management of information over its full lifecycle, including collaborative creation, review, approval and endorsement, delivery to consumers, maintenance to evolving qualities, collecting user feedback as to applicability and usefulness, review as to currency and accuracy, and ultimately retirement from circulation.

##### Join Entity Type

: a [value object](#Term_ValueObject) used to develop a many to many relationship between [entities](#Entity_Event).[[8]](#footnote-9)

##### Longitudinal data

: data collected through a series of repeated observations of the same subjects over some extended time frame.

##### Name

: a social domain non-unique identifier. In only western cultures the name is usually composed of Prefix, Given, Middle and Surname, but this is not universal, especially in oriental and pacific communities and should be avoided in system design for it to be inclusive and usable over a long period by different cultures. For example, in many cultures, a person’s name might be a given name that means something (“He who rides the crest of the wave”), and they may be given other names to commemorate achievements or other events[[9]](#footnote-10).

##### National Student Number

: a numeric system identifier associated to a person’s [identity](#Entity_Identity) in the NZ compulsory education system.

##### Normalised

: data that has been persisted in a relational manner across multiple schemas and/or tables, decreasing duplication of stored data.

##### Objective

: a specific, measurable and constrained outcome to achieve a broader goal of a strategic plan.

##### Operational Datastore

: a [transactional datastore](#Term_TransactionalDatabase) used to persist records of state, usually keeping only the latest version of state, either physical deletion (see [CRUD](#Acronym_CRUD)) or removing previous archive records to archive datastores.

##### Opportunity

: a favourable set of circumstances that presents the potential for achieving objectives or creating value.

##### Outcome

: the long-term effects of a process, task or activity and may not be directly observable (i.e., differs from an [output](#Term_Output)). Risks include aiming towards them without rechecking occasionally they deliver in alignment to changing goals and strategies to meet them.

##### Output

: the tangible/direct results of a process, task or activity. As tangible, it is measurable (i.e., it differs from [outcome](#Term_Outcome)s). Risks include aiming towards them without rechecking occasionally, they deliver in alignment to changing goals and strategies to meet them.

##### Operational Database

: a system datastore, usually a [transactional database](#Term_TransactionalDatabase) or system specific reporting database. A specialised system database may also be a [longitudinal datastore](#Term_LongitudinalDataStore).

##### Reality

: the state of things as they exist, as opposed to idealistic or notional ideas of them, of which [truth](#Term_Truth)s are.

##### Simple Object Access Protocol

: [SOAP](#Acronym_SOAP) is an [XML](#Acronym_XML) based [RPC](#Acronym_RPC) messaging protocol.

##### Software Development Kit

: a collection of one or more functioning examples of code provided to 3rd parties, demonstrating how to consume API endpoints, making development and use of the API endpoints easier and less costly to develop.

##### Strategic Decision

: Contrast with [tactical decision](#Term_TacticalDecision)s.

##### System Identifier

: a unique identity within a system set (e.g., a database table). The identity is not used by outside the system.

##### Transactional Datastore

: a structured datastore used to persist records of state, usually keeping only the latest version of state, either deleting (see [CRUD](#Acronym_CRUD)) or removing previous archive records to archive datastores. Operational datastores longitudinal datastores which persist previous copies of state records for longitudinal analysis of actual and predicted change over time of these values.

##### Tactical [Decisions]

: TODO

##### Tag

: see [categorisation](#Term_Categorisation).

##### Truth

: a commonly agreed *notional* understanding of [reality](#Term_Reality) based on perceptions thereof. Truths may differ between groups. Truths are *not* equivalent to [reality](#Term_Reality).

##### User Interfaces

: a form of interface for submitting and retrieving information from a system, for use by human users. [UI](#Acronym_UI)s generally are either text-based interfaces (i.e., console interfaces) or graphical user interfaces ([GUI](#Acronym_GUI)s) that are built following [WIMP](#Acronym_WIMP) patterns.

##### Value Object

: object identified purely by their set properties, as opposed to a [system identifier](#Term_SystemIdentifier) (see [entity](#Term_Entities)). Note: Line items of an invoice entity, or addresses of an identified person are common examples of value objects.

1. Versus CSV, a non-hierarchical data transfer format. [↑](#footnote-ref-2)
2. FIRST’s APIs are REST, augmented by ODATA, ENROL is an unsupported early version SOAP used to wrap simpler CSV payloads, NSI’s externally accessible APIs are REST based. [↑](#footnote-ref-3)
3. Using a wide range of protocols (custom xml feeds, REST, SOAP). [↑](#footnote-ref-4)
4. It is totally possible that a person is provided multiple names or even NSNs, simply due to human error and later correction. [↑](#footnote-ref-5)
5. I.e., consider avoiding making database fields for first, middle, last, etc which are really only but a western civilisation concept as opposed to universal attributes. [↑](#footnote-ref-6)
6. For example tracking learners transit to school in a school chartered bus. [↑](#footnote-ref-7)
7. [individual self.pdf (southampton.ac.uk)](https://www.southampton.ac.uk/~crsi/individual%20self.pdf) [↑](#footnote-ref-8)
8. I.e., represents a “join table” in a relational datastore. [↑](#footnote-ref-9)
9. https://www.familysearch.org/en/wiki/Learn\_the\_History\_and\_Customs\_of\_Your\_Culture#Personal\_names [↑](#footnote-ref-10)