Standardised Sematic Information Model (SSIM)

25 November 2018

Contents

[1 SSIM Overview 2](#_Toc530805211)

[2 Transactions 3](#_Toc530805212)

[3 Data Sets 3](#_Toc530805213)

[4 Data Elements or Delements 3](#_Toc530805214)

[5 Data Types Directory 4](#_Toc530805215)

[6 Facts Directories 5](#_Toc530805216)

[7 SSIM Id or SID 5](#_Toc530805217)

[8 Digital Id or DigId 6](#_Toc530805218)

[9 SIMM Ontologies 6](#_Toc530805219)

[10 SSIM Smart Reporting Objects or Ssros and Ssro Sets 9](#_Toc530805220)

[11 SSIM Export Import Objects or Seximos 9](#_Toc530805221)

[12 Points to cover 10](#_Toc530805222)

# SSIM Overview

SSIM (Standardised Sematic Information Model) is a method of storing, categorising, and accessing data in a standardised, semantic way which is powerful, open ended, easy to understand and use at a user level, while being efficient for scaling to large scale use.

SSIM as a whole may not be “simple” as it has a complex world to model, and that necessitates some complexity. However, it is proposed that the way in which SSIM is constructed makes it “simple” at each step, and “easy” at the level of an app using it, while still being complete and powerful.

SSIM is being developed by Pacio and will be used by Pacio but is intended to be an open source standard which can be used independently of Pacio.

SSIM applies to all data, but the initial development focus is on business data.

SSIM takes a bottom up approach to providing standardisation so that raw data categorised by SSIM can be recast or aggregated to suit any desired reporting requirement. In this way SSIM applies from a single raw transaction such as the sale of one can of beans all the way through to financial statements or <IR> integrated reports according to the desired accounting standard target, or targets e.g. US GAAP and IFRS, with the whole process fully automated from one end to the other.

SIIM provides semantic information by a flexible data description system based on directories of facts e.g. countries, currencies etc, and data objects structured according to ontologies. Any single piece of data can be fully described by a single 64 bit (8 byte) number called a SSIM Id or SID. SIDs are totally flexible yet efficient for blockchain/database use with their fixed and relatively small size.

SSIM encompasses relational information (how one item of data relates to another), how numeric data should be summed, how data is to be processed through changes of period, what is valid where, when data can be pruned, and how data should be presented in reports.

SSIM covers all storage, classification, and reporting needs, but will also interact with or interface with other current systems or data description languages or protocols such as XBRL, ODI, UBL, Open EDI, RDF, Ocean etc as required.

Users of apps making use of SSIM will not need to know anything about the details. In most cases apps will be able to classify data themselves. In the few cases where a manual selection might be required, an app will be able to present a selection list for a user to choose from.

For developers of SSIM ontologies or data objects, or for those who are curious about the details, SSIM keeps things understandable by avoiding complicated or unfamiliar concepts like arcs in XBRL or triples in WC3 OWL ontologies and semantic knowledge graphs.

Instead, maintenance or development of the ontologies and data objects underlying SSIM are done using spreadsheets which are a familiar and comfortable tool for anyone who might be interested. This use of spreadsheets does not lead to potential data or reporting holes or data integrity errors, as is the case when spreadsheets are used for generating reports, since the SSIM spread sheets are purely a visualisation and editing tool, which SSIM checks for validity before acceptance. The spreadsheets are never used directly.

All SSIM components will be open source and public, developed and maintained through a Pacio moderated folksonomy[[1]](#footnote-1) process. They will provide the equivalent of many currently scattered non-integrated directories, knowledge graphs, and ontologies/taxonomies.

The following sections describe the details.

# Transactions

A transaction is the basic or lowest level record or data item involved in SSIM. Everything starts as a transaction. SSIM and Pacio are transaction driven.

A transaction always has a date and time, called a datetime. All SSIM datetimes use GMT/UTC time. Datetimes may be converted to local or other times for reporting purposes, but internally to keep things clear, GMT/UTC rules.

In addition to its datetime, a transaction has:

A digital Id or digid (section 8) of who or what (entity/person, app/dapp, user, account) created the transaction

A digid of who or what the transaction is going to if different from the creator, including for transactions with other blockchains

A SSIM ID or SID (section 7) to define what the transaction is about (its standardised semantic content), and its data type (section 5). The SID is generated by the app which creates the transaction. For intrinsic crypto transactions the SID involves just the cryptos and optionally other blockchains involved from the Facts Directories (section 6). For other transactions involving data storage, the SID is generated via the facts directories plus the ontology (section 9) in use.

A data element or delement (section 4) Id if the transaction creates/updates a delement, as a unique reference for an entity, the app, the delement, and the data set (section 3) if applicable. A delement id does not define a period, even if the entity uses periods, as the period is set by the transaction’s datetime.

One or more binary data fields according to its data type

# Data Sets and Data Set Directory

Transactions for entities may be grouped into Data Sets where that suits or is a requirement of the application.

For example, accounting or financial reporting apps would use Data Sets for each set of balancing double entry money transactions, a set of journal entries, or one group of non-monetary postings.

Another example is the set of line items making up a purchase order or an invoice, which itself would be a member of a double entry data set.

Applications which involve periods, as discussed in Data Types (section 5), would use Data Sets by period.

Data sets will be identified by a delement Id for the data set header, as a unique reference for an entity, the app, and the header delement.

Transactions not stored in data sets will update delements immediately, but in the case of data set transactions, the transactions will not be considered final or fully committed until the set has been closed. If closure should never happen due to an app or user problem, the non-final transactions would be discarded.

Data Set Directory

Pacio will maintain a Data Set Directory of data set types which will be extended as necessary to meet the needs of app developers in a Pacio moderated folksonomy1 like process.

# Data Elements or Delements

Transactions which are not just a crypto transfer from one account to account, create or potentially update data elements or delements. A delement is like an account in a financial system, but as they are also used for non-financial data, the term ‘data element’ or ‘delement’ is used for them rather than ‘account’. Another reason for avoiding the term ‘account’ here is that PIO holders and holders of other crypto currencies have accounts for their tokens.

Delements have the same data type as transactions which create or update them for non periodic data types, or an allowed superset of the transaction data type for a periodic or data set header delement.

A delement possesses:

A delement id as a unique reference for the delement itself plus the app, the entity if applicable, and the data set if applicable

two datetimes: when the delement was created and when it was last updated

one or more binary data fields according to its data type

the SID for the delement

Delements inherit optional attributes from their ontology element for:

Static – cannot change after being created

Deprecatable – can be deprecated and then removed from service

Dynamic by replacement where a repeat transaction replaces the current value(s)

Dynamic by summation where a further transaction adds to the current value(s)

Double entry accounting delement meaning that a set of transactions involved in updating the delement must sum to zero, with the set also updating one or more other double entry delements

* Triple entry accounting delement

# Data Types Directory

Pacio will create and maintain a public directory of data types for data to be stored using SSIM. These data types will be somewhat akin to database schema, but are intended to be universal, and easier to understand for non-technical people.

This directory will help with standardisation by defining data types in one place for use by all SSIM Ontologies, without ontologies needing to repeat the definitions for the data types they use.

A data type can specify:

a single item such as a number, a money item, a datetime, or some text etc

a number of components e.g. a money amount, a quantity number, a units number, an inventory reference etc as needed.

larger items also such as document, image, video, file … any digital thing.

* fields repeated n times for periodic data as for financial data in monthly or other periods, where the periods and dates of the periods are an entity property

optional restrictions e.g. number must be positive, or must have 6 digits etc according to rules or patterns, but done without using complicated regular expressions that most people wouldn’t understand

* formatting information where relevant, potentially varied according to country or jurisdiction and/or human language.

The data types directory will be extended as necessary to meet the needs of ontology creators and app developers in a Pacio moderated folksonomy1 like process.

# Facts Directories

Pacio will develop and maintain directories of world facts or information, to be used as part of the semantic classification of transactions and delements.

These directories are for information which does not change, or which changes infrequently. (An example of a ‘fact” which could change is a country grouping as for the UK and Brexit.) These directories will use existing e.g. ISO classification where possible, but extend them to be more encompassing, while allowing them to be integrated into a SID.

Pacio will also provide dynamic data via its “Authenticated Accounting and Real Time Feed for Oracles or Apps” service, including links with public knowledge graphs. Such data will be categorised according to this directory service plus SSIM ontologies, but not itself form a “directory”.

The facts directories will be open source and accessible to all as a Directories Service, maintained via a Pacio moderated folksonomy1 like process.

Any combination of facts can be referenced via a single Fact Id or FarctId.

Facts directors will cover:

Countries, states/provinces, jurisdictions, regions, cities/towns, and groupings of these

Human languages

FIAT currencies

Crypto currencies and their blockchains or other distributed ledger systems

Functional roles – expense, sale, equity, fixed asset etc

People’s roles - director, partner, officer, remote worker etc

Entity types from sole proprietorship to public limited company including charities, NGOs, and Government Departments/agencies, for all the variations in the world

Stock and futures exchanges where public companies may be listed

Crypto exchanges

Industry/business classifications

Units of weight and measure

Activities – all business and human activities e.g. ‘retail sale’, ‘accounting’, programming, exercising etc, keyed to units where applicable e.g. truck driving and tonne kilometres

Report names or headings such as “Balance Sheet” allowing for language and jurisdictional variations

* Other groupings of facts that may be required

# SSIM Id or SID

A transaction or delement is classified or using a single 64 bit (8 byte) number called an SSIM Id or SID. SIDs allow totally flexible data description in just 8 bytes, which will help make data storage and transaction transfers efficient. No long or variable length tags are involved.

A SID is a key into a global database of sets of references or keys into the Data Types Directory, Facts Directories, and a SSIM Ontology. Each set of references used by an app results in a single SID. A SID does not point to data directly. It is purely to provide context and semantic content information.

A SID provides the equivalent of XBRL context plus concept tags and attributes.

64 bit SIDs allows for 264 – 1 or 18,446,744,073,709,551,615 different sets of references or 1.8 Billion for every person on earth at projected peak population of 10 Billion people, which should cover the world’s needs for decades. If ever that limit should be approached, then extending SIDs to just 10 or so bytes would allow for centuries.

SIDs once used will exist “forever”. They may become deprecated or no longer valid for new data, but will be kept forever for historical analysis purposes.

A SID by itself would not tell a human observer anything, but software will be easily able to show its references, and to use it in searches or reports via a Pacio SID Service which Pacio will operate.

A SID which uses only Facts would equal the FactId.

SIDs are not specific to Pacio i.e. they could also be used by non-Pacio systems.

# Digital Id or DigId

Pacio will provide a digital id or digid service for people, entities, apps, and dapps which works in conjunction with other services:

Open source technology and standards as listed in [Decentralized Digital Identities and Blockchain](https://cloudblogs.microsoft.com/enterprisemobility/2018/02/12/decentralized-digital-identities-and-blockchain-the-future-as-we-see-it/):

* [Decentralized Identity Foundation (DIF)](http://identity.foundation/)
* [Decentralized Identifiers (DIDs)](https://w3c-ccg.github.io/did-spec/) – a W3C spec that defines a common document format for describing the state of a Decentralized Identifier
* [Identity Hubs](https://github.com/decentralized-identity/hubs/blob/master/explainer.md) – an encrypted identity datastore that features message/intent relay, attestation handling, and identity-specific compute endpoints.
* [Universal DID Resolver](https://medium.com/decentralized-identity/a-universal-resolver-for-self-sovereign-identifiers-48e6b4a5cc3c) – a server that resolves DIDs across blockchains
* [Verifiable Credentials](https://w3c.github.io/vc-data-model/) – a W3C spec that defines a document format for encoding DID-based attestations.

Existing digital identity participants:  
[Civic](https://www.civic.com/), [DID](https://decentralized.id/) (Decentralized ID), [Essentia](https://essentia.one/), Estonia, [Legal Entity Identifier (LEI)](https://www.globallei.com/), [OpenID](http://openid.net/connect/), [Persona](https://persona.im/), [uPort](https://www.uport.me/), the United Nations ID2020 programme, [VeriMe](https://www.verime.net/)

Other initiatives that gain traction

As with all aspects of Pacio, the Digital Identity service will be open, and designed to readily accommodate new standards or services as they become available.

DigIds will be used in Pacio transactions, with PIO holder accounts, and for referencing entities, apps, dapps, and people using Pacio.

DigIds are not specific to Pacio i.e. they could also be used by non-Pacio systems via a DigId Service which Pacio will operate.

# SIMM Ontologies

SSIM uses ontologies as part of organising the storage of data in SSIM Smart Reporting Objects or Ssros, and Ssro Sets described in the next section.

SSIM ontologies are simpler than other ontology/taxonomy systems because of the information content delegated to other parts of SSIM, namely the Data Types Directory, Facts Directories, and Ssros plus Ssro Sets. Simply put, SSIM ontologies provide the framework, while Ssros and Ssro sets plus data types and facts fill in the details.

Any number of ontologies may be created. There will be many of them, ultimately thousands of them for all types of storage and reporting requirements.

An app will select the ontology or ontologies applicable to its needs, via the Ssro Sets it uses.

General purpose ontologies will be developed and maintained by Pacio in a moderated folksonomy1 process, but entities may also create specific purpose ontologies if they so wish.

Ontologies may be created for any purpose, but the initial ones to be built by Pacio will be intended for business and financial data. These ontologies could be structured in accordance with an accounting standard e.g. a US GAAP focused one, or an IFRS focussed one, but this is not necessary or even desirable given the SSIM Export Import Objects or Seximos of section 11. It will, in fact, be a goal of Pacio to produce more general or fundamental ontologies without the anglo-american biases of the current IFRS and US GAAP views of world, and which can be used to generate reports according to any desired accounting standard.

Ontologies use realms, domains, and elements to define the framework, with an element being the lowest level or most basic. An element corresponds to a Ssro or Ssros. There will typically be more Ssros than ontology elements, because Ssros can be categorised in other ways than just their ontology element, plus Ssros can be replicated as optionally filtered slaves for summing and presentations in various ways.

Ontologies are maintained in spreadsheets which are imported into Pacio. No arcs as in XBRL or triples as in WC3 web 3 specifications and knowledge graphs are involved. The import performs validity checks to check for possible errors or inconsistencies. Spreadsheet use makes it easy for accountants and business people to understand how SSIM ontologies are structured.

Ontologies provide the equivalent of the XBRL definition view. Presentation and calculation or other XBRL views are provided by Ssros and Ssro Sets in the SSIM case.

Apps will use ontologies to categorise the data they are creating, and, if alternatives should exist for a particular item, to prompt a user for a decision, all without users (other than ontology creators/maintainers) needing to know anything about the ontology being used.

Ontologies will use or have:

A status property for ‘under development’, ‘issued’, ‘deprecated’, or ‘not to be used for new data’, with dates for ‘issued’, ‘deprecated’, or ‘not to be used for new data’. Once an ontology has been issued, with an issue date, it cannot be changed. It can only be deprecated or set to ‘not to be used for new data’, typically on replacement by a new or upgraded version.

Facts directory references which can be applied to any ontology component, within the limitations that the component might apply. For example, facts directory functional roles can be used for a similar purpose to roles in XBRL roles but are universal, not particular to any one ontology.

Realms, which define groupings of domains, next item. Realms are similar to the XBRL concept of hypercubes. Realms can be deprecated, with a deprecated date and a date for when the realm becomes ‘not to be used for new data’. If a deprecated realm is the only realm for a domain, the domain and its member elements also become deprecated with the same dates.

Domains provide the next level down of categorisation, and provide a ‘home’ for elements, the next item. A domain can be a member of multiple realms and must be a member of at least one. Domains can be deprecated, with a deprecated date and a date for when the domain becomes ‘not to be used for new data’. The deprecated property and dates are inherited by all of a domain’s members. In XBRL terms SSIM domains encompass both dimensions and domains.

Elements or concepts are the lowest level of categorisation. Elements can stand alone or be a member of a domain. An element can be a member of only one domain. Elements in the ontology correspond to delements of stored data i.e. each delement has an associated ontology element specified via its SID.

Elements cover XBRL elements, concepts, members, scalars, facts, primary items, and facts.

Elements have attributes for:

Data type (number, string etc) from the Data Type Directory, which could also provide formatting information

Whether concrete or not, where concrete elements can hold data. Non-concrete elements can provide structure or headings etc but not hold data.

Level from 0 upwards for organisational purposes. These levels would normally be followed by Ssros for presentation and summing purposes, with optional mutual exclusive rules applied, but do not have to be. Ontology elements levels are just a guide.

Whether Read only (or Report only) or not. Read only elements would not be used for delements as there is no point in having a delement which can’t be written to, but a read only element can be used with a Ssro of an appropriate data type as the target of a summing operation.

Whether the element allows for dynamic entity extensions e.g. for specific directors, inventory items etc

* Whether deprecated, with a deprecated date and a date for when the element becomes ‘not to be used for new data’. This attribute is automatically set for an element which is a member of a deprecated domain.

Financial ontology elements have additional attributes covering:

Numerical element financial type:

* Profit and Loss type
* Balance Sheet type
* Appropriations type

Numerical element sign:

* Expected to be Dr
* Expected to be Cr
* Must be Dr
* Must be Cr

Numerical element accounting type:

* Double entry
* Double entry with an associated triple entry
* Notes/info – not double entry

The relatively simple SSIM ontologies in conjunction with the other SSIM components will be much more powerful than other methods of expressing semantic data relationships such as W3C's OWL ontologies or XBRL Taxonomies yet will be easier to work with and understand at the user level.

# SSIM Smart Reporting Objects or Ssros

SSIM Smart Reporting Objects or Ssros pronounced “s-rows” and Ssro Sets are used for processing data for reporting or querying.

Ssros do not store data – they are in memory objects for processing data from delements, and optionally transactions, for reporting and querying purposes.

Ssros are organised according to an ontology, using realms, domains, and elements in a multi branch tree structure. Ssros embody knowledge about their environment derived from their ontology, facts, and attributes plus logic built into them by Ssro developers.

Ssros may be copied to another place in the tree as a slave of the master Ssro, with optional filtering, to permit summing or reporting in flexible ways without having to create multiple data sets. For example, sales by region and sales by product/service type could be reported on from just the one set of sales data.

Ssro Sets are sets or collections of Ssros.

For a business/financial application Ssros and Ssro Sets, plus the underling delement and transaction data, form the Pacio intelligent equivalent of subsidiary ledgers plus the general ledger and its chart of accounts.

Ssro Sets are modular to permit full Ssro Sets being built up from smaller sets, potentially catering for jurisdictional or standards difference in the process. An app generating a report or making a query would use a full Sssro Set.

Building block Ssro Sets are called Incl-Sssro Sets as they are included when building a full Ssro Set. Incl- Ssro Sets are intended to be used for particular reporting areas e.g. Fixed Assets. Incl-Ssro Sets may be dynamically included/excluded at the time of building a full Ssro Set for a particular jurisdiction (country), and entity type.

A Ssro Set can be complete in itself, and thus be a “full Ssro Set” but it is preferable for ease of development, maintenance, and reusability, for full Ssro Sets to take advantage of the dynamic inclusion/exclusion capability of Incl-Ssro Sets.

Ssros and Ssro Sets are maintained in spreadsheets which are imported into Pacio. No arcs as in XBRL or triples as in WC3 web 3 specifications and knowledge graphs are involved. The import performs validity checks to check for possible errors or inconsistencies. Spreadsheet use makes it easy for accountants and business people to understand how Ssros and Ssro Sets are structured.

Ssros and Ssro Sets are described fully in the separate Ssros.docx document.

# SSIM Export Import Objects or Seximos

SSIM Export Import Objects or Seximos will allow export of of data organised via one ontology to other systems, or the import of data from other systems.

Conversion will only be possible to the extent that other systems can replicate the information depth of SSIM. In cases where the target system lacks equivalents for some SSIM features, information content will necessarily be lost on conversion.

Seximos and Seximo Sets will work as extended Ssros and Ssro Sets that work with both the SSIM data and the target “ontology”.

Writing or the creation of transactions, data sets (if applicable), and delements, will be optional if the target is an SSIM ontology. Otherwise the data will be held in the Seximos and Seximo Sets like Ssros, and be available for reporting from there, as for normal Ssros and Sssros sets.

Use case examples are:

One SSIM ontology to another e.g. for conversion from the universal or fundamental SSIM ontology to a “US GAAP” or IFRS focussed one, with optional writing

Ontology upgrades e.g. from the 2018 ontology to the 2019 ontology, with writing expected unless during testing

SSIM ontology to an XBRL taxonomy for XBRL reporting – no writing

SSIM ontology to W3C type ontologies for reporting – no writing

SSIM ontology to schemas as in [Schema.org](https://schema.org/) for reporting – no writing

* SSIM ontology to any other community required "ontology" as per [Ontologies Ontologies Everywhere – but Who Knows What to Think?](https://protege.stanford.edu/conference/2006/submissions/slides/1.2_Uschold.pdf) for which a Seximo interface can be written
* Importing data from a non SSIM based app to a Pacio SSIM based app using a SSIM ontology, with writing expected unless during testing

As for other SSIM components, Seximos and Seximo Sets are maintained in spreadsheets which are imported into Pacio.

Seximos and Seximo Sets are described fully in the separate Seximos.docx document. [?? To be written]

# Points still to cover

The Data Commons Knowledge Graph (DCKG)  
<https://browser.datacommons.org/>

‘Has a’ etc as well as ‘is a’

Digid re entity

Data set headers

Delement id systems

List all the fancy Ids

* factsId
* SID
* Data types Id
* Digid

Digid for inter blockchain transactions

Terminology SS

Other entity DBs

Dynamic delements e.g. for directories or inventory items

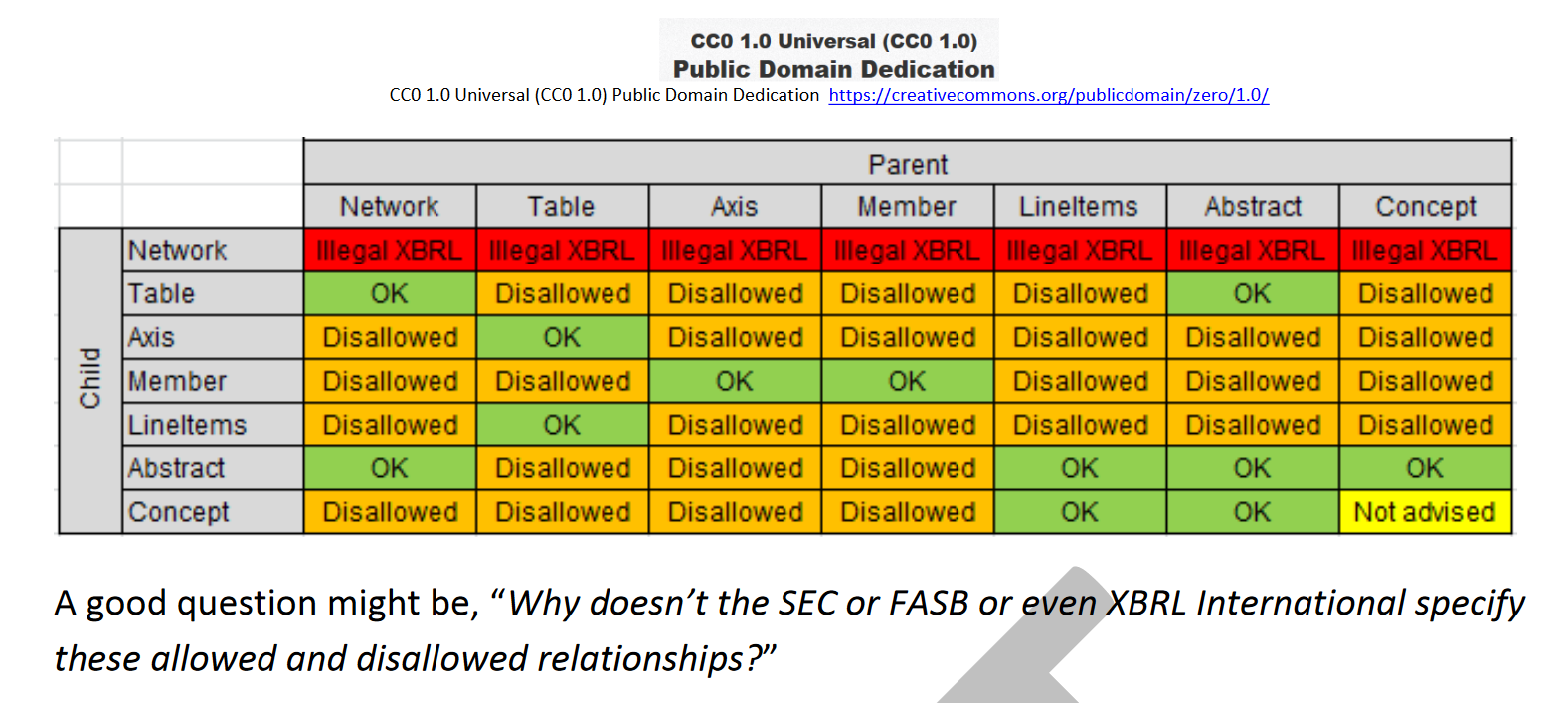
Mention descriptions

Cover entity privacy issues

Cover entity data in own DB linking

Cover entity specific stuff

Stress validity checking. Include a version of Charlie’s table?



Define how tables work

Cover nested data sets

Add a report writer section

Go into how comparative/prior period data will be handled especially re deprecated elements

Go into how Prior Period Adjustments (PPAs) will be handled

1. Folksonomy is a user driven system of classifying and organizing online content. It was used by [Freebase](https://en.wikipedia.org/wiki/Freebase), a large (1.9 Billion triple) public knowledge base prior to its acquisition by Google. [↑](#footnote-ref-1)