Standardised Semantic Information Model (SSIM)

Version C 06 December 2018

Contents

[1 SSIM Overview 2](#_Toc531840916)

[2 SSIM Basics 3](#_Toc531840917)

[2.1 Digital Id or DigId 3](#_Toc531840918)

[2.2 Data Types Directory 4](#_Toc531840919)

[2.3 Facts Directories 4](#_Toc531840920)

[2.4 SSIM Id or SID 6](#_Toc531840921)

[3 Pacio Transactions with SSIM 7](#_Toc531840922)

[4 Pacio Application Data with SSIM 8](#_Toc531840923)

[4.1 Data Elements or Delements 8](#_Toc531840924)

[4.2 Data Sets 9](#_Toc531840925)

[4.3 SSIM Ontologies 9](#_Toc531840926)

[4.4 SSIM Smart Reporting Objects or Ssros 12](#_Toc531840927)

[4.5 SSIM Import Export Objects or Simeos 12](#_Toc531840928)

[5 Points still to cover 13](#_Toc531840929)

Preamble

This is a technical document describing the Standardised Sematic Information Model (SSIM), intended for people interested in the details of SSIM. It is a living document, to be updated as development proceeds.

Why SSIM is needed, the advantages it offers versus alternatives, how it will be used, and how it will gain adoption are discussed in the separate document “The Case for the Standardised Semantic Information Model (SSIM)” [Writing started 2018.12.06]

# SSIM Overview

SSIM (Standardised Sematic Information Model) is a method of storing, categorising, and reporting on 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 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 and ESG (Environmental Social Governance) reporting 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.

SSIM provides semantic information by a flexible data description system based on many directories of facts e.g. countries, currencies, languages, roles etc, with support for multiple human languages and jurisdictions, plus reporting 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 small size. No long or variable length tags are involved at the transaction and data storage levels.

SSIM covers all storage, classification, and reporting needs, but will also interact with or interface with other 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 underlying 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.

The facts directories, which are a major part of SSIM, are intuitive and simple to use.

Ontologies and other parts of SSIM are more complex, but modelling a complex world inevitably requires some degree of complexity. However, the complexity has been made easier to understand and work with than in other systems, by avoiding arcane mathematics and jargon. Spreadsheets are used as the visualisation and development tool – a tool familiar to most people who are likely to use SSIM. (The spreadsheets do not present an integrity danger – their inputs are checked for validity and integrity. They are just a familiar tool, and 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.

# SSIM Basics

SSIM provides a precise and concise way to describe any item of data by means of a single number called a SSIM Id or SID. Even though the description of an item in words might be lengthy e.g. “International Business Corporation incorporated in Saint Lucia, which is a member of the Organisation of Eastern Caribbean States, and a member of the Eastern Caribbean Currency Union”, the SID would be just 8 bytes long, the same length as a SID for “rent”.

The avoidance of lengthy and variable length tags helps with blockchain and database efficiency.

The SID can be used in searches, and by apps to identify data in a standard way, semantically, and accurately.

The precise definition of the descriptions making up a SID improve reporting and search accuracy.

To achieve this, SSIM at its starting or basic level uses just four components:

Digital Ids or DigIds to identify the person, entity, app/dapp involved

A Data Type to define the structure or nature of the data being described

Directories of facts to be used to semantically describe a data item, with as many facts being used as is needed to fully describe it, in a process similar to tagging, in most cases done automatically by the app involved

The resulting SSIM Id or SID which combines all the above references into a single number

For business/financial applications SSIM goes further, but just this basic start is powerful in its own right. SSIM at this level can be used independently of Pacio.

Descriptions of the four components follow.

## Digital Id or DigId

People, entities, and apps using SSIM need to be identified uniquely. SSIM and Pacio are not about anonymous data, but rather ensuring that people or entities have ownership and control over their data, and to profit from it, not others such as the infamous FANGs (Facebook, Apple, Amazon, Netflix and Google). This does require that the creators of data be identifiable. It does NOT mean that the data has to be public. Whether the data is public or private is a choice of the creator.

For business applications especially a data system like SSIM needs to be able to uniquely identify each participant. This is needed to correctly handle inter entity transactions. Example: ACME ltd sends an invoice to Widget Inc. Only when both entities are uniquely and irrevocably identified in the system can fraudulent invoicing be avoided. Company names can change or conflict. An Entity Id is intended to be unique and permanent – essential for efficient and safe business.

Pacio takes the process a step further than others by also identifying the app or dapp which was involved in creating a piece of data. That improves security and auditability.

Pacio will provide a digital id or digid service to identify 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 by Pacio but could also be used by non-Pacio systems as DigIds are not specific to Pacio.

## Data Types Directory

In a standardised semantic environment, every item of data needs to be classified, whether that be a knowledge base entry about the universe, or a business record (invoice, purchase order, journal etc). Classification has two main attributes, what the item is in a generic sense (number, word, file etc), and a description what the content is about. Example: the item could be “currency number with 2 places of decimals” for which the description is “USD, rent”. The Data Types Directory defines the possible types of data items to be classified by SSIM.

Pacio will create and maintain a public directory of data types for data to be described by SSIM. The data types directory will be extended as needed in a Pacio moderated folksonomy1 like process.

This directory will help with standardisation by defining data types in one place for use by all SSIM components, including the more complex ones described in following sections, without those components needing to repeat the definitions for the data types they use.

A data type can specify:

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

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

a number of items e.g. a money amount, a quantity number, an activity units number, a barcode (inventory) reference etc as needed.

* 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.

A data type is referenced by an Id into the data types directory.

## Facts Directories

To continue the example from above: while the Data Type Directory contains the definition of items such as “currency number” the Facts Directories contains a list of 192 world currencies, and “facts” to describe what the currency transaction is about. The business app will choose the correct currency from the currencies directory, and it will choose appropriate further facts such as “sale” or “rent” to complete the description. The Facts Directories are intended to cover or model business (and human) activities and interests.

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.)

Pacio will develop and maintain the facts directories as an open source service accessible to all, maintained via a moderated folksonomy1 like process.

Directories will use existing classifications where possible e.g. ISO country codes.

All directories will allow for groupings or classes within them e.g. Asian countries.

All facts will allow for language and jurisdictional variations.

Facts can be flexibly combined as a set of references and relationships which results in a single SID as described in the next section.

The vast majority of directory uses, certainly for business needs, will be for the simple description of an item of data using one or more references without need for the complicated semantic syntax of systems such as [OWL](https://en.wikipedia.org/wiki/Web_Ontology_Language) (Web Ontology Language). An example is “this is an XCD money rent expense”.

Relationships

The relationship for a data item being described with a “fact” is self evident as adjectival (descriptive), or “is a”, “is in”, “has the property of”, “is a member of”, or “has as a member” according to the data type of the item and the type of the fact. Adjectival and “is a “ are the commonest. For “this is an XCD money rent expense”, the data type would be “money” with “XCD”, “rent” and “expense” all being descriptive or having a “is a “relationship to the data item.

For “Saint Lucia is a country in the OECS (Organisation of Eastern Caribbean States)” the relationship of Saint Lucia to the OECS is understood to be “is a member of” because OECS would be defined as a fact that is a group or collection of countries. Whereas “OECS” plus the fact “Saint Lucia” are linked by a “has as a member” relationship.

Such simple self-evident relationships, or relationships derived from the data type of the item and the type of a fact, meet business reporting needs, if not all those of other more esoteric domains such as metaphysics, pure mathematics, or religion. SSIM development will start with just these simple relationships.

Support for other relationships such as “is not”, “is not a member of set/class x“, plus logical combinations using “and” or “or” could be added as requirements arise or are defined. The reason for deferring such additions is that some other semantic and data description language attempts have suffered from trying to define everything at the start, with the result that options became over complicated and have never been used in practice, resulting in later specifications advising against their use, and their ultimate deprecation. With SSIM the approach will be to add more complicated options only if there is proven to be a real need for them.

Business Related Directories

The initial directories to be developed are those needed for business and financial reporting, though some of these will also be applicable to other needs:

Types of directory fact e.g. country, state/province/ political grouping of countries, set of x, etc for each of the following directories

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 and occupations - 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

Product categories e.g. household appliances > climate control appliances > fans > ceiling fans

Product barcodes including UPC (Universal Product Code), EAN (International Article Number, previously European Article Number), ISBN (International Standard Book Number), Code 128 etc. Of the order of 200 million such codes exist with new ones being created constantly. They may be accessed via online APIs for use by SSIM.

Service types – accounting, auditing, legal work etc

Facts and terms relevant to so called ESG (Envorinmental Social Governance) reporting

* Report names or headings such as “Balance Sheet”

General Data Directories

More general directories not so specific to business data will also be developed progressively, covering such things as:

Colours

Materials

Terms for the sciences: physics, chemistry, biology, botany, geology, genetics, mathematics, astronomy, palaeontology, ecology, oceanography, meteorology, zoology etc

Medical terms

Religions

Product names

People names

* Other groupings of facts that people may be interested in and are prepared to help build. This could include linking to other facts sources such as [The Data Commons Knowledge Graph (DCKG)](https://browser.datacommons.org/)

## SSIM Id or SID

Any item of data is classified or semantically described 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 processing 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 Facts Directories, and optionally for data or an app using it, an SSIM Ontology reference as described in section ??.

Each set of references used by an app results in a single SID. A SID provides 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 which is 1.8 Billion for every person on earth at projected peak population of 10 Billion people. That 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 of reference combinations.

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

The SID database will be a critical component of SSIM. Additions and updates will be logged for security and to enabling rebuilding in the event of catastrophic loss of the database.

A SID by itself would not tell a human observer anything, but software will easily show its references via the Pacio SID Service. Applications will be able to search or query by facts using indices.

SIDs could be shown as QR codes if an application wished to publish them.

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

# Pacio Transactions with SSIM

A transaction is the basic or lowest level Pacio record or data item. Everything starts as a transaction. Pacio is transaction driven, and SSIM applies at that level

A transaction may also be all that is involved. For a crypto transfer, or one piece of information, the transaction forms the complete record. For more complex storage needs discussed in the next section, the transaction also updates other storage elements.

A transaction includes:

An Id – the transaction Id

A date and time, called a datetime. All Pacio and 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.

Digital Ids or digids (section 2.1) for the entity (if there is one involved), the user/account, and the app/dapp which created the transaction

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

A data type Id

One or more binary data fields according to the transaction’s data type

A SID to describe the transaction. The SID is generated by the app which creates the transaction. For transaction only storage the SID involves just references from the facts directories as described in SSIM Basics above. For transactions which update another storage element (next sections), a delement, the SID also includes an ontology (section 4.3) reference.

A delement Id for the data set (section 4.2) if one is involved

Transactions will be retrievable by Id, and by indices for digids, SIDs, and for the references making up a SID. These indices will be large but stored totally off chain as they are non-critical and could be rebuilt if necessary.

Applications for FIAT or crypto transfers, and applications involving just storage of discrete items of knowledge will need northing more than transactions with SSIM. Business and other applications which involve more complex data storage and reporting, will, however, need to make use of the storage features described in the following sections.

# Pacio Application Data with SSIM

As an example of an application requiring more complex data storage and reporting let’s consider the production of company financial statements. That involves more than just transactions or single items of data. It involves aggregated data from numerous sources (today an average of over 800 spreadsheets in a Fortune 500 company) structured according to a set of rules – the accounting standard being followed, typically as defined via an XBRL taxonomy. The result is then put into a presentation format so that it can be read by stakeholders and be sent to the authorities.

Applications of that magnitude will use some or all of the more advanced aspects of SSIM described here. A business application involving raw transactions through to final financial statements or <IR> integrated reports, would use all of them.

The components are:

Data elements or delements

Data Sets

Ontologies

SSIM Smart Reporting Objects or Ssros

SSIM Import Export Objects or Simeos.

## Data Elements or Delements

A data element or delement is an item of data created or updated by transactions. It is stored independently from the transaction or transactions which create or update it. All delements could be recreated from their transactions, but delements provide a means of coping with the numbers and complications of the raw transactional data. One delement could represent from one to millions of transactions.

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’.

A delement possesses:

A delement id as a unique reference for the delement

Digid for the entity or user ‘owning’ the delement

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

A data type id. The data type could be the same as the data type of the transaction which creates/updates the delement, or it could be a superset of it for an application involving periodic data e.g. a financial application with 12 or 13 periods in a year plus more periods used for end of year rollover.

One or more binary data fields according to the delement’s data type

* A SID to define the standardised semantic content of the delement. The SID is generated by the app which creates the transaction which creates the delement. The SID will use the facts directory references from the transaction SID plus an ontology (section 4.3) reference if an ontology is being used.

Delements with an ontology reference 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

Delements will be retrievable by Id, and by indices for digids, SIDs, and for the references making up a SID. These indices will be large but stored totally off chain as they are non-critical and could be rebuilt if necessary.

## Data Sets

Transactions for entities may be grouped into data sets where that 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 in turn be a member of a double entry data set.

Applications which involve periods 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 or sets have been closed. If closure should never happen due to an app or user problem, the non-final transactions would be discarded. [? How? Should data set transactions be cached and only written when closed? But that has real time issues e.g. re inventory quantities. Need to resolve this.]

Data Set Directory

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

## SSIM Ontologies

SSIM uses ontologies as part of standardised semantic organising of data and for use by SSIM Smart Reporting Objects (Ssros) and SSIM Import Export Objects (Simeos described in the next two sections.

Ontology use is not mandatory – Pacio and SSIM as described to this point can be used without an Ontology. However, ontology use is required for the more advanced SSIM uses provided by Ssros and Simeos. Business applications which produce XBRL reports or work with other data description languages will need to use ontologies.

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. Then Simeos provide interfaces to and from other systems.

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. 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 data organisational 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.

However, that is not necessary or even desirable given the capability of Simeos*.* 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. That is where the “standardised” part of SSIM comes into play.

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 visualise and understand how SSIM ontologies are constructed.

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 details of the ontology being used.

The relatively simple SSIM ontologies in conjunction with the other SSIM components will be 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.

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 but are universal, not particular to any one ontology.

All components have a name, labels (for reporting), and a description

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. 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. There can, however, be multiple delements for one ontology element, qualified by other references such as roles.

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. Concrete elements (as delelemts or Ssros) 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 definition framework. The Ssros import program checks for consistency.

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

Whether Start/End in nature i.e. with a start of period balance updated by transactions to give an end of period balance. Such elements would have start and end of period labels.

Numerical element accounting type:

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

For more on the SSIM components vs XBRL components see DFR Terminology.xlsx

## 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. They provide for intelligent, flexible reporting.

The Pacio report writer uses them. An apps own special purpose report writer could use them.

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, including from transactions (at a speed penalty), 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

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 only 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 Import Export Objects or Simeos

SSIM Import Export Objects or Simeos will allow export 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.

Simeos and Simeo 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 Simeos and Simeo 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 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 Simeo 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 testing

As for other SSIM components, Simeos and Simeo Sets are maintained in spreadsheets which are imported into Pacio, with extensive validity and consistency checking.

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

# Points still to cover

Digids for inter blockchain transactions

Handling other entity DBs

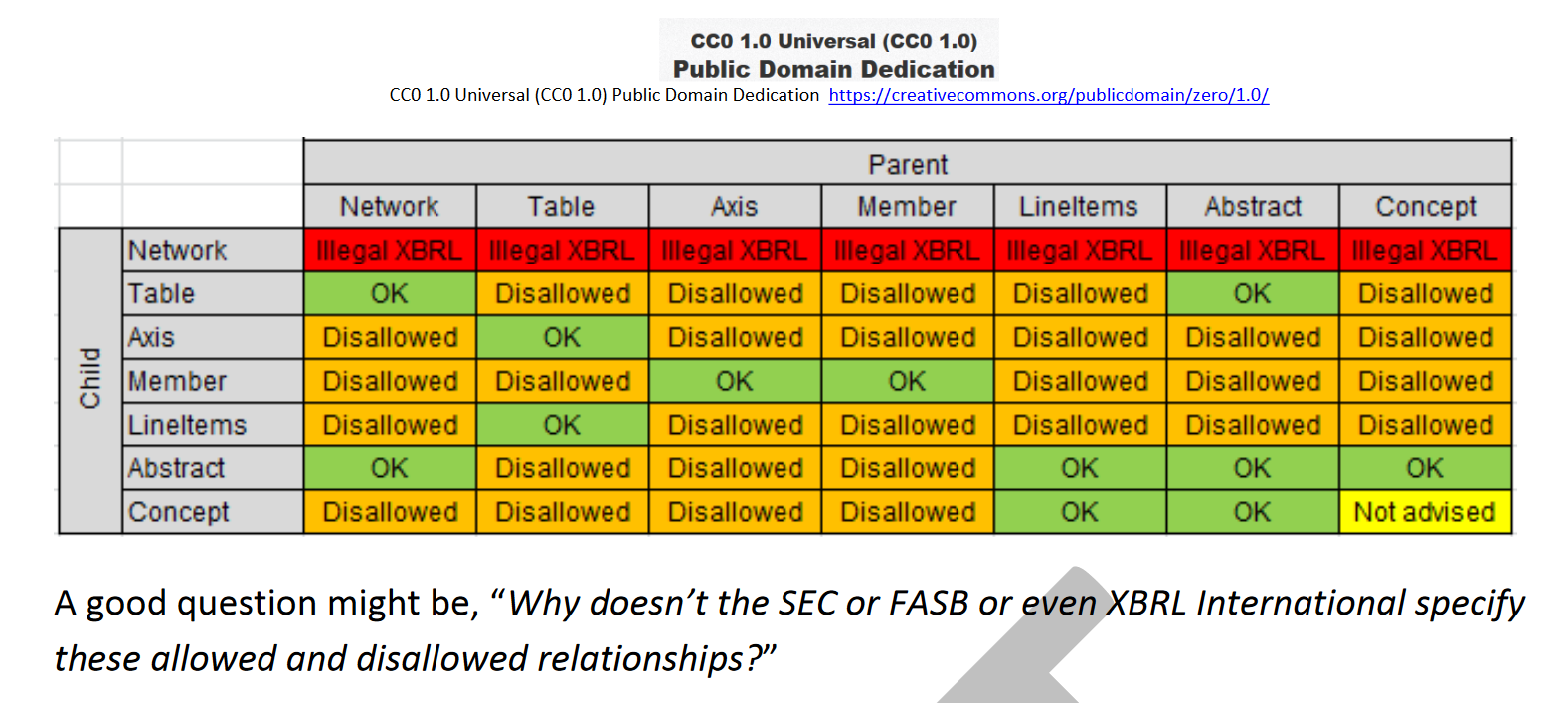
Entity privacy issues

Specify how privacy settings will be handled for transactions and delements

Entity data in own DB linking

Ssro access to entity specific stuff

Include an equivalent to Charlie’s following table?



Add a report writer section, and in that explain how tables would work

Cover nested data sets better

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

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

Decide how to handle transactions not committed to a data set due to the set not being closed.

Specify the Binary format. ION or JsonB or RFC 7049 Concise Binary Object Representation?

Cover how data can be marked as outdated (without being deprecated)

Cover how data can be deleted for GDPR requirements

Give more details of the various services

Interaction between SSIM databases and data pruning

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)