IoT Metadata Specification

Updated: Oct 25, 2022

Version 0.10.0.1 (draft for comment)

CONTENTS

FORWARD

This forward is not part of the IoT Metadata Specification; it is included for information only.

ABOUT THIS SPECIFICATION

This specification is the "*private draft*" version of the IoT Metadata Specification v.0.10.0.1 for review and discussion.

This at the moment is a working document and subject to change. In the near future Terbine will declare this as a public version and changes will be versioned and either backward compatible or a detailed migration path will be outlined.

AUDIENCE

This document is intended for anyone who will interact with the Terbine system, whether a data provider or user. This will allow an overview of what metadata is collected and the structure used for discovery and indexing of content within Terbine.

HOW THIS DOCUMENT IS ORGANIZED

This document is organized within these main sections.

- Metadata Definition - specific of the individual entities and their elements.
- Domain Types - standard reference types
- Metadata Elements - organization of the entities, their cardinality within the larger metadata structure.

CONVENTIONS USED IN THIS DOCUMENT

Entities will have a label, name, comment and list of elements (fields). Label is descriptive title for human consumption, name is a title for machine processing and a comment provides a long text explanation of the element.

TYPES

Types will be either concrete or referenced.

If a concrete type it will be one of the following.

- UUID
- TEXT
- CHAR
- NUMERIC
- DECIMAL
- CURRENCY
- ID – id to a domain type in numeric format such as 1 or 2.
- DATE
- DATETIME
- TIME
- FLAG - 1 is true and 0 as false
- BOOLEAN – true or false
- YESNO – yes or no

If referenced, they will be of two variations

- **type** - this means the entire type as defined is included in the current definition. Example is **type:createupdate** where all fields defined as belonging to createUpdate type are included in this definition.
- **ref** - this means a reference to the unique identifier for that type is included in the current definition. Example is **ref:createupdate** where only the identifier that references an instance of an createUpdate information is included in this definition.
- **domain** – this refers to a domain type, these are listed under the section Domain Types. Example is **domain:sensorType** where one of the ID values defined for that type will be included here.

Mandatory or Optional

Elements can be defined as mandatory or optional. Mandatory fields are required to have a valid value as defined by the element type. Default is mandatory if not specified. See Cardinality for further information.

Cardinality

Elements can have a defined cardinality which is the number of elements. This also can be used to show a mandatory vs. optional relationship.

- Fixed - fixed occurrence. Default is 1 if not specified.
- 0 .. 1 - Item is optional, can have zero or 1 elements.
- 1 - Item has 1 element and is mandatory
- 0 .. N - an element can have 0 to N number of elements. Denotes optional elements.
- 1 .. N - an element can have 1 to N number of elements. Denotes mandatory attribute.

What This Document Does Not Cover

This document does not cover certain aspects of metadata in relation to the contributors organization.

OVERVIEW

This document is the official specification for the IoT Metadata Definition v 0.9.9.9s.

RELATED DOCUMENTS

- Terbine IoT Data Exchange V1 Requirements
- Terbine API Overview

USES OF METADATA

Metadata is collected, tracked and analyzed within Terbine for a variety of reasons. These include the following:

- Provide unique identity.
- Provide originator/owner/source name information.
- Track data lifecycle information.
- Track transferring of ownership.
- Tracking of source type and source specific information (sensor).
- Allow identifying group and organizational information.
- Provide contextual information, including geospatial identification.
- Track grading (quality) information.
- Store schema and type information.
- Allow identifying data with predefined and user supplied categories and tags.

For a sensor data exchange to be usable by a range of entities (e.g., human via the Web interface, an IoT Data Platform or an AI acting on behalf of a Member organization) the following key elements need to be provided for within the following elements of the total offering:

- A Metadata Specification
- Indices built upon the store of metadata
- Web-based tools for authoring and discovering metadata instances
- APIs pertaining to Search, Creation and related

This document outlines the first of these, the specification of how metadata will be structured to allow the elements to be possible.
The world of IoT or sensor-born data introduces many unique challenges within the realm of metadata. These include that the Machine to Machine (M2M) communication

will rely on automatic update and discovery of metadata elements. Also, within IoT it is important to track these types of relationships within the data.

- **Container**: Another aspect of IoT is that often data is delivered within a larger ecosystem of related sensors used to track an entire environment. The value of the information often lies in the ability to identify and record this relationship. Metadata within Terbine allow storing and updating this relationship. Example of this are a group of sensors measuring various readings within a large HVAC system. Note the HVAC system could also be part of a larger monitoring system for an entire building.

- **Similarity**: Also, relationships that track similar data sources from disparate providers is required to allow cross referencing and identification of these links. Example of this is temperature sensors of a specific type across all same HVAC systems in all buildings.
- **Inheritance**: Data can be aggregated, combined, reduced, mapped etc. to derive new datasets that are useful for a specific use case. Therefore the data that was used to create other datasets should be tracked.

TYPES OF METADATA

Metadata is generally divided into two overarching types.

- **Structural** - defining the organization of container for the data.
- **Descriptive** - defining the content or context of the data

Terbine metadata specification is designed to cover both of these types and allows a complete description of the form, structure, content and context of the data.

TYPES OF ATTRIBUTES

Within the specification there are three types of attributes.

- **Delivered** – it is part of the metadata information directly linked to the described content format and classification. This is items that are known before any content is delivered.
- **Derived** – it is part of the metadata information derived based on content once delivered, such as quality.
- **Custom** – this is specific to a supplier/industry and enhances the metadata information with further descriptions. This is indexed and discoverable like other types.

SOURCE OF METADATA

Metadata is created from one of these sources.

- **User supplied** – metadata is entered through the Terbine API before or is created after ingestion of the associated data stream.
- **Data generated** – under certain conditions metadata may be machine generated from ingested or sample data.

- **Migrated** – metadata migrated from other source systems containing their own metadata. In this manner the attributes of the data can be transferred over and reflected within Terbine.

CONTENT VS. METADATA

There can be gray areas where the same information could be treated as content or metadata, depending on the workflow. In general, metadata should have value on its own without regard for the content. For example, if there is following record delivered to Terbine, this would be content.

```
2017-08-14T01:01:01,23.1,TEMPERATURE
```

The value “TEMPERATURE” which describes a sensor reading value is considered content.

If within the description of the data a user would enter the following tags to describe that record.

```
SENSOR;DAILY;TEMPERATURE
```

The tag “TEMPERATURE” would be considered metadata.

Note also metadata can be delivered embedded in or part of the delivery and will be treated as such. This can be used to create a new metadata record or update an existing one. An example of this is as follows.

```
1:   DATE,READING,TYPE
2:   2017-08-14T01:01:01,23.1,TEMPERATURE
3:   2017-08-14T01:03:11,17.1,TEMPERATURE
```

Here the first line is considered metadata which is used to describe the data and all lines following (in this example lines 2 and 3) are actual content.

IDENTIFICATION

Critical to any metadata system is to be able to uniquely identify and allow human and machine to machine (M2M) identification and querying. The Specification allows this using their ID registry.

The IDs allocated by this service will be assigned to each metadata configuration as well as for content and organization/users. This is an immutable identifier. It has a length of 36 and is a combination of numeric and alphanumeric characters.

Also part of the function of this service is to store other unique identifiers and types and associate with the main identifier. These various ids can be queried to find the main ID as well as add or update these within an organization's domain.

FORMAT OF AN IDENTIFIER WITHIN THE ID SERVICE

Name	Length	Type	Comment
Schema	1	Numeric	For now always 1.5.
Type	2	Numeric	Type of identifier (see table below for possible values).
Registrar	10	Numeric	Unique identifier for the owning or registering party.
Item	10	Numeric	Unique identifier for the item being identified.

Custom	13	Alpha	Client specific identifier they can use for their purpose. If not entered by client, it will auto-allocate this field.

TYPES

Id	Name
1	account
2	user
3	metadata
4	dataset
5	content
6	transaction

CUSTOM LISTS

This metadata specification is intended for tracking information about delivered data formats and structures that are cross-industry. These can be thought of as a flexible extension mechanism to the fixed portions of the specification.

To allow maintaining a catalogue of designators that is flexible, but still allows non-human guided M2M communication, is a significant challenge. To implement this, the concept of custom lists has been introduced and integrated into the metadata specification.

An organization can then maintain these named lists by assigning a variable number of entries with an id value, name and optional description and then assign these to a metadata configuration. The values will be indexed and searchable along with all fixed portions of the specification.

Custom lists have two elements. The list definition and the entries for that list:

List Definition

organization	UUID	Mandatory	The owning organization, not provided if a global list.
name	TEXT	Mandatory	Name of this list
description	TEXT	Optional	Description of this list

List Entries

id	ALPHA	Mandatory	A unique identifier within this list for this list entry
name	ALPHA	Mandatory	Name of this list entry
description	ALPHA	Optional	Description for this list entry

Example

List Definition is:

32ee4079-e562-45c8-8a63-ab11c56af3b9	ODB-II Codes

List Entries are:

P0600	Serial Communication Link Malfunction
P0601	Internal Control Module Memory Check Sum Error
P0602	Control Module Programming Error
P0603	Internal Control Module Keep Alive Memory (KAM) Error
P0604	Internal Control Module Random Access Memory (RAM) Error
P0605	Internal Control Module Read Only Memory (ROM) Error (Module Identification Defined by SAE J1979)
P0606	ECM/PCM Processor Fault
P0607	Control Module Performance
P0608	Control Module VSS Output A Malfunction
P0609	Control Module VSS Output B Malfunction
P0610	Control Module VSS Output C Malfunction
P0611	Fuel Injector Control Module Performance
P0612	Fuel Injector Control Module Relay Control Circuit

The person defining the metadata configuration then designates that a custom list will be used with the specification and can assign any number of the entries under that list as being relevant to that metadata configuration.

METADATA DEFINITION

The following top-level sections can be found in the Terbine Metadata Specification definition section.

- Internal Types – these are definitions for types that are used within definition of entities. Generally, these have a child relationship to an owning entity (example address).
- Meta – this is information about the metadata record such as name, description and create data information.
- Originator / Owner - Specific names of entities *generating* and *owning* the datasets to which a given metadata instance points
- Identifier - unique identifying information
- Sensor Type and Name
- Environment - System or subsystem within which physical sensors are situated. Also "interactors" that can impact or influence sensor outputs including a given group of sensors' data grade.
- Additional Attributes – this section allows structure definition and information related to additional industry specific attributes.
- Container - System or subsystem within which physical sensors are situated. In this case the word "container" is to be considered as literal.
- Industry Classification - ISIC
- Legal - this is legal type, regulatory and any required citation
- Schema - information pertaining to the schema and datasets format
- Grading - internal quality indicator
- Custom Lists - defined lists that allow expansion of the specification with corporate or industry specific lists
- Domain Types – these are definition of standard domain or reference data. These are predefined types that have an id, code and associated description.

Fields within top level definition include

- **requireLegalReview** - flag if entire metadata configuration requires legal review when created.

INTERNAL TYPES

These are defined types that only have an instance with another identifiable piece of content. These types have no identifier.

CREATE INFO

Label:	**CreateUpdate**		
Name:	**createupdate**		
Comment:			This is a standard type used to hold created date and update information. Generally updated internally in the system and used for display purposes.
Elements			
createdate	Mandatory	DATETIME	Timestamp of creation
createuser	Mandatory	ref:user	Reference to user responsible for creating entity
updatedate	Optional	DATETIME	Timestamp of last update
updateuser	Optional	ref:user	Reference to user responsible for updating entity

LAT LON INFORMATION

Label:	**Lat Lon Info**		
Name:	**latLonInfo**		
Comment:			This is a standard type used to hold information about rights, legal information or copyright information.
Elements			
label	Mandatory	TEXT	Label for this lat/lon pair
latitude	Mandatory	double	
longitude	Mandatory	double	

LEGAL INFORMATION

Label:	**Legal Info**		
Name:	**legalInfo**		
Comment:			This is a standard type used to hold information about rights, legal information or copyright information.
Elements			
type	Mandatory	domain:legalType	Type of legal info

REGULATORY INFORMATION

Label:	**Regulatory Info**		
Name:	**regulatoryInfo**		
Comment:			This is a standard type used to hold information about regulatory information.
Elements			
name	Mandatory	TEXT	Name of regulatory body pertaining to the dataset defined by the metadata

LOCATION INFORMATION

Label:	**Location Info**
Name:	**locationInfo**

Comment:			This is a standard type used to hold information about location information.
Elements			
category	Mandatory	domain:locationCatType	Location category
type	Mandatory	domain:locationType	Type of location info
latitude	Optional	TEXT	Used if single coordinate of type Lat/Lon or GPS Coordinates
longitude	Optional	TEXT	Used if single coordinate of type Lat/Lon or GPS Coordinates
radius	Optional	TEXT	Radius of sensor reading from Lat/Lon Value
radiusUnit	Optional	TEXT	Unit of measure for Radius
latLonInfo	Optional	LIST type:latLonInfo	Array of labeled coordinates if type “Bounded Box” or “Polygon”
address	Optional	TEXT	Street address
stateterritory	Optional	TEXT	State or territory information
county	Optional	TEXT	County or province information
country	Optional	TEXT	Country code as defined in **ISO 3166-2**
postalcode	Optional	TEXT	Postal code
movingText	Optional	TEXT	For GPS and Algorithm Generated.
freeText	Optional	TEXT	When location type Other this field may be used. Also, for additional text for Moving subtype options,

SENSOR INFORMATION

Label:	**Sensor Info**		
Name:	**sensorInfo**		
Comment:			This is a standard type used to hold information about sensor information (source type and specifics).
Elements			
sourceName	Optional	TEXT	Manufacture. E.g. `Suburban Housing Project 14`
sourceType	Optional	TEXT	Manufacture. E.g. `temperature" or "wind speed`
sensorType	Mandatory	domain:sensorType	Type of sensor info
manufacturer	Optional	TEXT	Manufacture. E.g. `Supco`
make	Optional	TEXT	Make. E.g. `ToughSense`
model	Optional	TEXT	Model. E.g. `255C`
version	Optional	TEXT	Version. E.g. `1.119`

SCHEMA INFORMATION

Label:	**Schema Info**
Name:	**schemaInfo**

Comment:			This section contains information about schema of the content. This is highly dependent on the type.
Elements			
format	Mandatory	type:formatType	Type of format of data when delivered
type	Mandatory	type:schemaType	Type of schema if schema is present in body element.
body	Optional	TEXT	Schema body, dependent on type.
unit	Optional	TEXT	Unit of measure, e.g. meter

META

Label:	**Meta**			
Name:	**meta**			
Comment:			This contains information about the metadata. This is considered metamodel information.	
Elements				
name	Optional	TEXT	Name for Metadata. Also known as search name.	
description	Optional	TEXT	Description of the metadata in long form. Allowable 2500 characters.	
createupdate	Mandatory	type:createupdate		Create and update information for organization

IDENTIFIER

Label:	**Identifier**		
Name:	**identifier**		
Comment:			This section contains information that is used as an unambiguous reference to the information, title, and description.
Elements			
id	Supplied	UUID	UUID identifier, internal only
tid	Supplied	ALPHA(36)	This is the id provided from the Identification Service. Used for machine to machine identification and public identification of entities within TERBINE.
createupdate	Mandatory	type:createupdate	Create and update information for organization

ORIGINATOR

Label:	**Originator**		
Name:	**originator**		
Comment:			Specific names of entities *generating* the datasets to which a given metadata instance refers to.
Elements			
id	Mandatory	UUID	Global unique identifier
name	Mandatory	TEXT	Name for this entity.
type	Mandatory	domain:entityType	ID that references the type

OWNER

Label:	**Owner**		
Name:	**owner**		
Comment:			Specific names of entities *owning* the datasets to which a given metadata instance refers to.
Elements			
id	Mandatory	UUID	Global unique identifier
name	Mandatory	TEXT	External identifier for this source.
type	Mandatory	domain:entityType	ID that references the type

OWNERSHIP

Label:	**Ownership**		
Name:	**ownership**		
Comment:			All information pertaining to the specifics of where/how datasets associated with a given metadata instance are generated.
Elements			
originators	Mandatory	List of originators	1..N originators.
owners	Mandatory	List of owners	1..N owners.

RELATIONSHIP

Label:	**Relationship**		
Name:	**relationship**		
Comment:			If this metadata instance was created by combining datasets from other

			existing metadata instances these can be designated here.
Elements			
relationship	Optional	List of metadata identifiers	1..N metadata ids.

DATASET

Label:	**Dataset**		
Name:	**dataset**		
Comment:			This section contains information that is used to describe the dataset of the associated content the metadata is describing.
Elements			
id	Mandatory	UUID	Global unique identifier
type	Mandatory	domain:datasetType	ID that references the type of source (see DatasetType domain data)
sensorInfo	Mandatory	type:sensorInfo	Information on sensor type and sensor specifics as applicable and available.
schemaInfo	Mandatory	type:schemaInfo	Information on schema for this dataset
createupdate	Mandatory	type:createupdate	Create and update information for dataset information. This is create and update information on the dataset data record.

CONTAINER

Label:	**Container**		
Name:	**container**		
Comment:			System or subsystem within which physical sensors are situated.
Elements			
id	Mandatory	UUID	Global unique identifier
type	Mandatory	domain:containerType	ID that references the type of container (see Container Type domain data)
locationType	Mandatory	domain:locationType	This is the type of location info.
locationSubType	Optional	domain:locationSubType	This is the type of sub location, dependant on location type
location	List Mandatory	type:locationInfo	This is the location of the container, section is mandatory even if location information is unknown. See locationType.
name	Optional	TEXT	Container name, like ‘Aircraft’, ‘Vehicle’, or ‘Traffic Signal.

ENVIRONMENT

Label:	**Environment**
Name:	**environment**

Comment:			Environment in and around the location(s) of the sensors emitting data associated with a given metadata instance.
Elements			
Id	Mandatory	UUID	Global unique identifier
Type	Mandatory	List of domain:environmentType	ID that references the type of environment (see Environment Type domain data)
Interactor	Optional	TEXT	Interactor name, like 'Meteorological'
containerName	Optional	TEXT	Container name, like 'Aircraft', 'Vehicle', or 'Traffic Signal.

ADDITIONAL PROPERTY

Label:	**Additional**		
Name:	**additionalProperties**		
Comment:			Industry specific properties.
Elements			
Id	Mandatory	UUID	Global unique identifier
Type	Mandatory	List of domain:AdditionalType	ID that references the type of Additional Property (see Additional Type domain data)
definition	Mandatory	TEXT	JSON structure describing the additional property content
content	Mandatory	TEXT	JSON structure of attributes with values for additional property

INDUSTRY CLASSIFICATION

Label:	**Industry**		
Name:	**industry**		
Comment:			This section contains information about industry and associated tags for the content
Elements			
id	Mandatory	domain:ISIC	1..N ISIC Codes.

LEGAL

Label:	**Legal**		
Name:	**legal**		
Comment:			This section contains information about legal and copyright information related to the content.
Elements			
id	Mandatory	UUID	Unique identifier for this rights information.
legalInfo	Mandatory	type:legalInfo	Legal information internal type.
createUpdate	Mandatory	type:createupdate	

CITATION

Label:	**Citation**

Name:	**citation**		
Comment:			This section contains information about required citation for any dataset within this metadata configuration.
Elements			
id	Mandatory	UUID	Unique identifier for this citation information.
citation	Mandatory	type:text	Actual citation text.
createUpdate	Mandatory	type:createupdate	

REGULATORY

Label:	**Regulatory**		
Name:	**regulatory**		
Comment:			This section contains information about regulatory information.
Elements			
id	Mandatory	UUID	Unique identifier for this rights information.
regulatoryInfo	Mandatory	type:regulatoryInfo	Regulatory information internal type.
createUpdate	Mandatory	type:createupdate	

GRADING

Label:	**Grading**		
Name:	**grading**		
Type:	derived*		
Comment:			This section contains information about grading information for content.
Elements			
id	Mandatory	UUID	id of grading info.
score	Mandatory	INTEGER	Value between 0 and 999. This is populated by the system using a variety of proprietary criteria for measuring quality of the underlying data.
grade	Mandatory	TEXT	The highest level will consist of data generated by sensors that are calibrated and maintained by humans whose job it is to do so. Terbine will use a four-level scale, with each level being an order of magnitude more 'certain' than the one below it. The scale is derived from the score field.
comment	Optional	TEXT	Comment about grading.
createupdate	Mandatory	type:createupdate	Create and update information for grading information record. This is the create and update date of this record holding grading information.

** **Derived** denotes this is a read-only. Any attempt to update this externally will be ignored. This section is populated, maintained and continually reviewed by the internal system after the metadata and associated datasets have been loaded.*

CUSTOM LIST

Label:	**Custom List**

Name:	**customList**		
Comment:			This section contains information about custom lists that have been assigned to a configuration.
Elements			
id	Mandatory	UUID	0 or more id of custom lists that have been linked to this configuration..
entries	1..n	Entries of ID/Name pairs	Under each list the list entries that have been assigned to this configuration.

DOMAIN TYPES

Note that the values provided is not an exhaustive list and in most cases is meant to show an example of the type of domain data.

DATASET TYPE

Label:	**Dataset Type**
Name:	**datasetType**
Domain Type Id:	**3**
Comment:	This is a designator for type of dataset.

Values:

ID	CODE	DESCRIPTION
30	Sensor	Sensor sourced dataset

SENSOR TYPE

Label:	**Sensor Type**
Name:	**sensorType**
Domain Type Id:	**4**
Comment:	This is a designator for sensor type.

Values: Complete list in system can be maintained.

ID	CODE	DESCRIPTION

FORMAT TYPE

Label:	**Format Type**
Name:	**formatType**
Domain Type Id:	**5**
Comment:	This is a designator for format type.

Values:

ID	CODE	DESCRIPTION
500	CSV	Comma Separated
501	TAB	Tab Separated
502	JSON	JSON Format
503	XML	XML Format
504	Positional	Format of a record is defined by start and end position
505	Character	Non comma separated records such as semicolon

506	XLS	XLS Format
507	DOC	DOC/DOCX Format
508	PDF	PDF Format
509	ZIP	Zip
510	TAR	Tar
511	TARGZ	Tar Zipped
512	Unknown	Unknown Format
513	KMZ	Keyhole Markup Language Zipped
514	KML	Keyhole Markup Language

CONTAINER TYPE

Label:	**Container Type**
Name:	**containerType**
Domain Type Id:	**7**
Comment:	This is a designator for container type.

Values:

This is a designator for container type.

ID	CODE	DESCRIPTION
70	System	Container is part of a larger system.
71	Subsystem	Container is a subsystem of a larger system.

ISIC

Label:	**ISIC Code**
Name:	**isicCodes**
Domain Type Id:	**N/A**
Comment:	This is a designator for International Standard Industrial Classification (ISIC).

Values: Reference https://ilostat.ilo.org/resources/methods/classification-economic-activities/

SCHEMA TYPE

Label:	**Schema Type**
Name:	**schemaType**
Domain Type Id:	**8**

Comment:	This is a designator for schema type.

Values:

ID	CODE	DESCRIPTION
80	JSON	JSON Schema definition
81	AVRO	Avro Schema
82	XSD	XML Schema Definition
85	TEXT	Text Description of schema
83	None	
84	Unknown	

ENTITY TYPE

Label:	**Entity Type**
Name:	**entityType**
Domain Type Id:	**6**
Comment:	This is a designator for entity type. Used to classify originator and owner.

Values:

ID	CODE	DESCRIPTION
60	Corporation	Public traded or private corporation
61	Individual	Individual
62	Government	Government
63	Non profit	Non Profit Organization.
64	Education	Educational Institution
65	Unknown	Unknown Entity Type
66	NGO	Non-Governmental Organization

ENVIRONMENT TYPE

Label:	**Environment Type**	
Name:	**environmentType**	
Domain Type Id:	**16**	
Comment:		This is a designator for environment type. In this case the word “environment” is to literally the environment in and around the location(s) of the sensors emitting data associated with a given metadata instance.

Values:

ID	CODE	DESCRIPTION
160	Indoors	Indoor Environment Factor Type
161	Outdoors	Outdoor Environment Factor Type
162	Aerial	Aerial Environment Factor Type
163	Marine	Marine Environment Factor Type
164	Space	Space Environment Factor Type
165	Alpine	Alpine Environment Factor Type
166	Urban	Urban Environment Factor Type

ADDITIONAL TYPE

Label:	**Additional Type**	
Name:	**additionalType**	
Domain Type Id:	**27**	
Comment:		This is a designator of the structure used for describing an additional property.

Values:

ID	CODE	DESCRIPTION
300	CSV	CSV Field description
301	JSON	JSON Structure description for additional property

LEGAL TYPE

Label:	**Legal Type**	
Name:	**legalType**	
Domain Type Id:	**10**	
Comment:		This is a designator for legal type. This is a designator for a party or organization that had or has legal rights for the content.

Values:

ID	CODE	DESCRIPTION
103	Proprietary	Proprietary controlled data, public or private data.
104	Open Source	Open source licensed data.

REGULATORY TYPE

Label:	**RegulatoryType**	
Name:	**regulatoryType**	
Domain Type Id:	**14**	
Comment:		This is a designator for regulatory type. Currently not used, initially regulatory will be free text. Will be re-introduced as known values are accumulated and reviewed.

Values:

ID	CODE	DESCRIPTION
140	GDPR	Content is covered by General Data Protection Regulation.
141	Privacy Shield	Content is covered by Privacy Shield.
142	ITAR	Content is covered by International Traffic in Arms Regulation
143	UK-DPA	United Kingdom Data Processing Act
144	NULL	Not Applicable

LOCATION TYPE

Label:	**Location Type**
Name:	**locationType**
Domain Type Id:	**13**
Comment:	This is a designator for location type.

Values:

ID	CODE	DESCRIPTION
130	Fixed	Fixed location
131	Moving	Moving or mobile location
132	Other	Other or unknown

LOCATION SUB TYPE FIXED

Label:	**Location Subtype (Fixed)**
Name:	**locationSubType Fixed**
Domain Type Id:	**101**

Comment:	This is a designator for fixed location sub type.

Values:

ID	CODE	DESCRIPTION
10001	Address	Street address, street, city, state, postal, country.
10003	LAT/LON	Latitude / Longitude
10007	Bounded Box	Two coordinates designating NW and SE coordinates.
10008	Polygon Coordinates	Polygon Coordinates (set of 5 lat/lon values)

LOCATION SUB-TYPE MOVING

Label:	**Location Subtype (Moving)**
Name:	**locationSubType Moving**
Domain Type Id:	**102**
Comment:	This is a designator for moving location sub type.

Values:

ID	CODE	DESCRIPTION

10004	Device GPS	Location, via GPS on sensor
10005	System Level	Geo produced by system level tracking
10006	Algorithm Generated	Geo generated via algorithm

LOCATION CATEGORY TYPE

Label:	**LocationCatType**
Name:	**locationCatType**
Domain Type Id:	**17**
Comment:	This is a designator for location category types.

Values:

ID	**CODE**	**DESCRIPTION**
170	Terrestrial	Operate on land
171	Aerial	Operate in the air.
172	Marine	Operate on or below the oceans or other large bodies of water

173	Spaceborne	Operate in orbit around the Earth, another celestial body (e.g., the Moon or Mars) or deep space.

FEED TYPE

Label:	**Type**
Name:	**type**
Domain Type Id:	**N/A**
Comment:	
	This is a designator for the feed type (Fixed, Archival or Group)

Values:

ID	CODE	DESCRIPTION
1	Archival	Archival/Fixed Dataset
2	Continuous	Continuous/Streaming Dataset
4	Group	Group Metadata Instance

LOAD FREQUENCY TYPE

Label:	**Load Frequency Type**
Name:	**loadFrequencyType**
Domain Type Id:	**N/A**
Comment:	
	This is a designator for the unit of time used to measure load frequency.

Values:

ID	CODE	DESCRIPTION
1	Milliseconds	Millisecond Load Frequency Type
2	Seconds	Seconds Load Frequency Type
3	Minutes	Minutes Load Frequency Type
4	Hours	Hours Load Frequency Type
5	Days	Days Load Frequency Type
6	Hz	Hertz

LOAD FREQUENCY TYPE

Label:	**Update Interval Type**
Name:	**updateIntervalType**
Domain Type Id:	**N/A**
Comment:	
	This is a designator for the unit of time used to measure the update interval.

Values:

ID	CODE	DESCRIPTION
1	Milliseconds	Millisecond Update Interval Type
2	Seconds	Seconds Update Interval Type
3	Minutes	Minutes Update Interval Type
4	Hours	Hours Update Interval Type
5	Days	Days Update Interval Type
6	Hz	Hertz

Metadata Elements

Section Name	**Occurrence ****	**Description**
Meta	1	Summary information about the metadata.
Identifier	1	Identifying information about the metadata.
Dataset	1	Dataset information about the metadata.
Ownership	1	One ownership record that contains one or more references each to an originator and owner
Grading	1	Grading information for this metadata configuration.
Relationship	1..N	List of identifiers for metadata instances that contained datasets used to create this instance.
Container	1..N	Container information related to larger systems of which the deliver is part of. These can link to other containers providing a hierarchy.
ISIC	1..N	This possibly contains reference to multiple ISIC codes.
Legal	0..N	Optional information pertaining to legal information.

Custom List	0..N	Optional information about custom lists linked to this metadata configuration. When a list is linked there will be 0..N entries from that list assigned to this metadata configuration.

** For an explanation of the Occurrence column see the section in the Preface titled **Cardinality**.

SAMPLE METADATA

Below is a complete example metadata instance.


```
{
    "categories": [],
    "citation": {
        "createUpdateInfo": {
            "createDate": "2020-03-01T11:41:31Z",
            "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
            "updateDate": null,
            "updateUser": null
        },
        "id": "c9745b31-e653-4280-8e16-ae251a38f570",
        "text": "None Specified"
    },
    "cloneId": "2efbcd06-99c9-480c-bcff-e2e00299fc0e",
    "containers": [
        {
            "address": null,
            "altitude": null,
            "createUpdateInfo": {
                "createDate": "2020-03-01T11:41:31Z",
                "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
                "updateDate": "2020-02-29T05:09:12Z",
                "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
            },
            "description": null,
            "endDate": null,
            "extId": null,
            "id": "4ad103ac-e1bf-4c17-8a95-72b3e7995839",
            "latitude": null,
            "legalType": null,
            "legalTypeName": null,
            "location": {
                "address": "5757 Wayne Newton Blvd.",
                "altitude": null,
                "city": "Las Vegas",
                "country": "United States",
                "countryId": 226,
                "county": null,
                "freeText": null,
```

```
                "latLonInfo": [],
                "latitude": null,
                "longitude": null,
                "postalCode": "89119",
                "radius": null,
                "radiusUnit": 0,
                "radiusUnitName": null,
                "sensorText": null,
                "stateTerritory": "Nevada"
            },
            "locationCategory": 170,
            "locationCategoryName": "Terrestrial",
            "locationSubType": 10001,
            "locationSubTypeName": "Address",
            "locationType": 130,
            "locationTypeName": "Fixed",
            "locations": null,
            "longitude": null,
            "name": null,
            "nameId": 0,
            "parentId": null,
            "startDate": null,
            "type": 70,
            "typeName": "System"
        }
    ],
    "customLists": [],
    "dataset": [
        {
            "comment": null,
            "createUpdateInfo": {
                "createDate": "2020-03-01T11:41:31Z",
                "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
                "updateDate": "2020-02-29T05:09:12Z",
                "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
            },
            "datasetUrl": "https://aspm.faa.gov/tfms/sys/OPSNET.asp",
            "extId": null,
            "externalType": 201,
            "id": "c4c6a8e4-f079-4205-9da3-dc92b2fb6fdb",
            "schemaInfo": {
                "body": "#, Date, Airport, Hour, Arrivals:AC+AT,GA,MIL,Total",
                "createUpdateInfo": {
                    "createDate": "2020-03-01T11:41:31Z",
                    "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
                    "updateDate": "2020-02-29T05:09:12Z",
                    "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
                },
                "format": 506,
                "formatName": "XLS",
                "id": "1283b52c-0d58-40fe-9e3e-39ea9770278d",
                "property": null,
                "type": 81,
                "typeName": "JSON",
                "unitOfMeasure": "Numeric Count",
                "unitOfMeasureId": 1955
            },
            "sensorInfo": {
                "comment": null,
                "createUpdateInfo": {
                    "createDate": "2020-03-01T11:41:31Z",
                    "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
```



```
                    "updateDate": "2020-02-29T05:09:12Z",
                    "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
                },
                "id": "46bfd06d-2f97-46dc-be6d-f1a1a771fc7b",
                "make": null,
                "manufacturer": null,
                "manufacturerId": 0,
                "model": null,
                "sourceName": "Traffic Flow Management System Counts (TFMSC)",
                "sourceNameId": 11781,
                "sourceType": "Aircraft Count",
                "sourceTypeId": 11782,
                "type": 452,
                "typeName": "Radar",
                "version": null
            },
            "type": 30,
            "typeName": "Sensor"
        }
    ],
    "deliveries": [],
    "environment": {
        "containerName": null,
        "containerNameId": 0,
        "environmentInfos": [
            {
                "createUpdateInfo": {
                    "createDate": "2020-02-29T04:56:19Z",
                    "createUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c",
                    "updateDate": null,
                    "updateUser": null
                },
                "id": "5df24829-5f71-4a3d-b97b-6ac0b3b667e2",
                "recordType": 1,
                "type": 1601,
                "typeName": "Outdoor"
            }
        ],
        "interactors": []
    },
    "events": null,
    "gicsCodes": [],
    "gradings": [
        {
            "comment": null,
            "createUpdateInfo": {
                "createDate": "2020-03-01T11:41:31Z",
                "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
                "updateDate": null,
                "updateUser": null
            },
            "grade": "Bronze",
            "id": "5b731805-410e-4e10-8598-dac8dfb254f2",
            "score": 680,
            "type": 1
        }
    ],
    "hasPrice": 0,
    "id": "22ee6391-cad5-42d5-8a5c-79c1a2e2811f",
    "identifier": {
        "createUpdateInfo": {
            "createDate": "2020-03-01T11:41:31Z",
```



```
            "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
            "updateDate": "2020-03-01T11:52:14Z",
            "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
        },
        "extId": null,
        "id": null,
        "tid": null,
        "uri": null,
        "urn": null
    },
    "isicCodes": [],
    "lastDataUpdate": null,
    "legal": [
        {
            "comment": "Opensource",
            "createUpdateInfo": {
                "createDate": "2020-03-01T11:41:31Z",
                "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
                "updateDate": "2020-02-29T05:09:12Z",
                "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
            },
            "endDate": null,
            "externalUrl": null,
            "id": "83ffbbf9-a8de-4011-8444-7cbbf2c32bd0",
            "startDate": null,
            "type": 104,
            "typeName": "Opensource"
        }
    ],
    "loadFrequency": null,
    "loadFrequencyType": null,
    "meta": {
        "createUpdateInfo": {
            "createDate": "2020-03-01T11:41:31Z",
            "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
            "updateDate": "2020-03-01T11:52:14Z",
            "updateUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c"
        },
        "description": "Flight arrivals data is collected by Federal Aviation
Administration (FAA) which is managed by the U.S. Department of
Transportation. Traffic Flow Management System Counts (TFMSC) is designed to
provide information regarding the total arrivals and departures of aircrafts,
which are counted by the number of aircrafts per hour in 2019. Flights are
detected via radar by the National Airspace System (NAS) and is assembled by
the Federal Aviation Administration (FAA). Flight arrivals and departures
dataset are collected to conduct research to ensure that commercial and
general aviation is safest in the world.",
        "endDate": null,
        "imageUrl": null,
        "name": "Flight Arrivals / McCarran International Airport, Las Vegas,
Nevada, United States / 2019",
        "startDate": null,
        "tid": null,
        "version": "1.0"
    },
    "numberDatasets": 1,
    "orgId": "6d94ff88-4b41-41d0-b3a4-d14d89da0939",
    "orgName": "TERBINE",
    "owners": [],
    "ownership": {
        "originators": [
            {
```



```
                "createUpdateInfo": {
                    "createDate": "2020-02-29T04:56:19Z",
                    "createUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c",
                    "updateDate": null,
                    "updateUser": null
                },
                "id": "30b3e0d6-09f4-4d42-89e7-b495991eb864",
                "name": "Federal Aviation Administration",
                "originatorId": "b91fb9eb-91e9-44e5-8749-3285da4b1795",
                "type": 1,
                "typeName": "Originator"
            }
        ],
        "owners": [
            {
                "createUpdateInfo": {
                    "createDate": "2020-02-29T04:56:19Z",
                    "createUser": "caf4e7ca-e1f3-49bd-974d-72a3f78b9a4c",
                    "updateDate": null,
                    "updateUser": null
                },
                "id": "3d961310-d45a-4536-b59b-ddf496cf0f58",
                "name": "U.S. Department of Transportation (USDOT)",
                "originatorId": "66d91fb9-c8f5-437a-95ef-4fad8130ecd6",
                "type": 2,
                "typeName": "Owner"
            }
        ]
    },
    "priceText": "No price",
    "regulatory": [
        {
            "comment": "NULL",
            "createUpdateInfo": {
                "createDate": "2020-03-01T11:41:31Z",
                "createUser": "aab864c1-74fc-44c1-b0d3-531828717729",
                "updateDate": null,
                "updateUser": null
            },
            "endDate": null,
            "id": "9484d53b-2595-4c1a-ad20-39f8c35a5c97",
            "startDate": null,
            "type": 144,
            "typeName": "NULL"
        }
    ],
    "relations": [],
    "requireLegalReview": 0,
    "sourceIndicator": 1,
    "sourceInfo": null,
    "status": 1,
    "tags": [],
    "type": 1,
    "updateInterval": null,
    "updateIntervalType": null
}
```


APPENDIX A - GLOSSARY

Attribute - named characteristic of an Entity.

Channel – Unique delivery within an organization. An organization can have many channels and each channel has a metadata instance related to it.

Company - *See organization.*

Dictionary - list of terms and possibly their occurrence.

Entity - person, place, or thing about which data is stored.

Extract - subset of data that is pulled a main data set.

Instance - an actual concrete example of

Join - combining data based on a common attributes or defined set of attributes.

Metadata - detailed description of the instance data, format, content, source, and modification history. Within Terbine this is used to track all information about content delivered to the Terbine Ingestion API , Persisted within Terbine and displayed within the Terbine Marketplace.

MIME - Internet Media Types

Namespace - The use of namespaces avoids conflict between properties in different schemas that have the same name but different meanings. For example, two metadata entities might have an Owner property: in one, it might mean the person who owns a resource; in another context, the application used to create the resource.

Originator– Specific names of entities *generating* the datasets to which a given metadata instance points.

Owner - Specific names of entities *owning* the datasets to which a given metadata instance points.

Sensor Details (Specifics) - Manufacturer/make/model/version or other appropriate designation(s) for sensor(s) or subsystem(s) pointed to by a given metadata instance.

Source Name - Specific name of product or system generating the datasets to which a given metadata instance points.

Source Type - Specific typing of sensors or subsystems generating the datasets to which a given metadata instance points.

APPENDIX B - REVISIONS

Version	Date	Name	Description
Draft	2015/10/02	Brian Enochson	For Initial Review
Draft	2015/10/20	Brian Enochson	Added Regulatory Section
Draft	2016/01/30	Brian Enochson	Updated with recent additions for Meta and Legal sections.
Draft	2016/03/01	Brian Enochson	Updated Reference Domain Information.
Draft	2016/03/30	Brian Enochson	General Updates
Draft	2016/10/22	Brian Enochson	Citation and Legal Enhancement
Draft	2017/02/01	Brian Enochson	Multiple identifier updated
Draft	2017/07/23	Brian Enochson	TERBINE identifier, refactor spec and add industry specific information. Remove content attributes from metadata. Remove events.
Draft	2017/08/22	Brian Enochson	Minor modifications around id and provenance tracking
Draft	2017/09/17	Brian Enochson	Changes for domain values, identifier and schema changes.
Draft	2017/09/21	Brian Enochson	Added GICS Code to Metadata
Draft	2018/03/23	Brian Enochson	Additions of several domain types and grading score.
Draft	2018/04/30	Brian Enochson	Reorg, Cleanup and addition of provenance and context values to specification.

Draft	2018/05/18	Brian Enochson	Updated sections related to sensor type/source type/source name and added Geo info for Boundex Box and Polygon. Regulatory type free text, units of measure, remove GPS Coordinates
Draft	2020/03/25	Ramon Murao	Updated sections to add Type, Load Frequency and Update Interval.
Draft	2022/03/21	Brian Enochson	Updated sections on continuous feeds and smaller enhancements on specific attributes.
Draft	2022/10/25	Brian Enochson	Updated sections for handling Additional Properties within Existing Specification.

APPENDIX C – LABELLING GUIDELINES

IOT METADATA SPECIFICATION

UNIFORM LABELLING GUIDELINES

PURPOSE

This document intends to provide guidance and instructions for users authoring/creating metadata instances for use within the Terbine IoT Data Exchange and/or its derivations, i.e., TerbineLink for the mobility and energy sectors.

INTENT

Everything found in the IoT metadata being encoded into the Terbine system, pertains to **allowing finding of correct datasets or streams**, assignment of rights and other key issues, plus other "behind the scenes" elements needed to make the system usable. It's important that the fields provided within metadata instances are sufficiently rich to allow for deep and specific searching, while leading to the most accurate results possible. Similarly, duplications can be avoided through careful attention to naming, especially

when manually entered i.e., free form. Note that as time moves forward, AI-based programs will increasingly be "doing the searching" thus the need for resolute naming guidelines is paramount to ensure the highest-likelihood results and outcomes.

AUDIENCE

This document is intended for anyone who will be generating metadata instances for ingestion into the Terbine system, whether internal to Terbine or a client/data provider.

VERSIONING

This is currently a working document and therefore subject to change. In the near future Terbine may elect to declare this a public document under open-source standards.

PHILOSOPHY

The overarching reasons for having metadata instances are twofold: (1) to provide an abstraction layer away from the data itself, e.g. many datasets can be mapped to a given metadata instance; and (2) to provide searchability (whereas the datasets or streams themselves do not offer sufficient information to make them searchable). Therefore it is incumbent upon the author creating metadata instances, to do so within guidelines that offer the highest probability of a desired search outcome. Without the instillment of a uniform labelling schema for the various metadata fields, the likelihood of "false positives" resulting from a search increase. As technology moves forward, there will be a large number of systems based on artificial intelligence algorithms, which will be querying metadata – the opportunities for sending back incorrect search results could potentially increase when a non-human interactor makes the query, thus reinforcing the need for a consistent and clear labelling schema.

TERMINOLOGY

Any terms not accepted in common use, or for which the usage herein is different than found in typical usage, will be called out in the Glossary section.

ALL NOMENCLATURE ENTERED INTO METADATA FIELDS MUST BE IN AMERICAN ENGLISH UNLESS A SPECIFIC EXCEPTION IS AUTHORIZED.

GENERAL NOTES

- When source information for a field cannot be determined, leave the field blank.
- **It is important to investigate prior entries for any fields that offer hand-entry.** If a term has already been utilized and is sufficiently descriptive to cover the new metadata instance being authored, then as a matter of good judgement it may be appropriate to employ the already-utilized term vs. entering a new one that is too close so as to confuse subsequent searches.
- **It is important to first ascertain** is the data archival or continuous. Archival is normally aggregated data from a previous time period. Dependent on the data it will be grouped in some time interval such as weekly, monthly or annual. Continuous data is sent in a real time fashion with continual updated. This has an update frequency property which is how often the reading is emitted from the sensor and a load frequency which is how often it is actual transmitted to Terbine.
- **All datasets or streams associated with a given metadata instance must contain only only data type**. If submitted data files have readings from different sensors they must be separated, each with its own metadata instance, prior to submission into the Terbine system. This can be done by the data provider, or via an Ingestion Adapter.

FIELD ENTRY GUIDELINES

NAME

Purpose: The Name field provides the primary means for an automated process or human user to find metadata instances that are being sought, therefore this field is the primary focus of these labelling guidelines. The Name field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user. With the latter, it will be rendered at the top of a Metadata Instance "tile" within the Terbine GUI and/or Branded Data Exchange GUI(s).

Convention: the first word(s) of a Name must provide a clear, unambiguous indication of what the datasets or streams associated with the given metadata instance are (see Examples). Names should be just long enough to be descriptive, and not so long that they are 'dense' but not so short as to be 'cryptic' and thus lead to false-positive search results. All concepts conveyed in the name should be independent of one another, and no "piece" of the name should be contingent or dependent on another aspect of the name or title.

Syntax: Per the preceding, the first word(s) of a Name will be an indication of the sensing type (see Notes below for definition and examples). Following this can be a categorical label for the machines/sensors emanating the datasets or streams that the given metadata instance maps to. After this a separator "/" can be applied and optionally an acceptable location reference can be included (see below for restrictions).

All words in a Name must have the first letter capitalized, e.g.:

Sensing Type / Common Location Name *(optional)* **/ Year** *(if applicable, per below)*

Example: **Sea State Levels / Indian Ocean / 2014**

Prohibited/Restricted Nomenclature:

- Naming of Sensors, Owners, Originators, date-ranges, addresses or other **elements that are covered by other/specific metadata fields**.
- **Not to be used in Names are the terms "data" "sensor(s)" "sets" "streams"** (as in data streams vs. physical streams ala water) **"readings" "measurements" "information" or casual/colloquial terms** (e.g. use "vehicles" not "cars" or "autos", use "pedestrian(s)" not "people", "precipitation" not "rain" or "rainfall", "sea state" not "swells" or "waves").
- **Acronyms or shorthand-terms** are to be avoided unless they are very commonly used and not industry-specific (such as "DC" "AC" or international airport codes, per below).

- **Participles, prepositions, conjunctions, pronouns, determiners**. For definitions of these grammatical terms please refer to the Oxford Dictionary or similar authoritative reference.
- **Locations are not to be included in the Name** unless commonly understood on a global basis, "New York City" or "JFK Airport" whereas "Manhattan" is not since there are many 'Manhattans' around the United States. Always assume that the program or person doing a search does not have other than a general understanding of location-names (or in the case of a programmatic search, the software making the inquiry make be matching against a list of cities, counties, states, regions, countries or widely known landmarks such as airports, e.g. "EWR" not "Newark Airport" or "New Jersey Airport" which are too broad, or "United States Nationwide" is acceptable whereas "America" or "U.S." are not).
- **Years should not be included in the Name** unless all datasets pertain specifically to that year and no further datasets will be added under that metadata instance (i.e. if an archive of datasets for the year 2015 then adding the year is acceptable, but not if it includes multiple years). **In no case can months or other dates or date-ranges be included in the Name field.** If a year is included it must be placed at the end of the name e.g. "Vehicle Counts on public roads / London Congestion District / 2013").
- **Years should not be included in the Name** if the type of dataset is streaming or continuous in nature. Continuous or streaming datasets are delivered in real time or at certain intervals and as such the contents and the range of time covered by the dataset increment automatically.

Examples:

Atmospheric Density / Satellite-Based

- In this case, it is salient that the readings are coming from satellites. Note that "atmospheric density" does not refer to a sensor type, rather a sensing type.

Vehicle Counts / London Congestion District / 2017

- In this case, the sensing-type is "vehicle counts" and the location can be included since London is commonly understood to mean London, England. All of the datasets that match to the metadata instance were generated during calendar 2017.

Ocean Depths / Global

- In this case, the sensing-type is "depth" and the target is "ocean" to indicate what the depth-readings are for. The addition of "Global" is acceptable if the Location field does not provide specific coordinates.

Vehicle-Sampled Barometric Air Pressure / United States Nationwide

- In this case, it is salient that the barometric air pressure readings are coming from moving vehicles. Note that "vehicle" does not refer to a sensor type, it is closer to a physical platform aka Container. A percentage of these will require judgement calls as to whether the inclusion of context belongs in the Name field, since those

elements should come up in a search when the appropriate fields are populated correctly (thus, the fact of the readings coming from vehicles would come up if ‘vehicle’ was entered into the search, and the Container field included the word ‘vehicle’ to ensure the metadata instance was found). Note that the inclusion of “United States Nationwide” is valid provided that all other metadata instances of similar sourcing and types, do not have conflicting names (such as “United States” without specifying “Nationwide”).

Notes:

- For the Name field, sensor-types are to be distinguished from types of sensing whereas the latter is acceptable in a Name, e.g., “temperature” is acceptable whereas “thermometer” is not. “Velocity” is acceptable where “speedometer” is not and so on.

- Being that the system is intended only for machine-generated data in numeric or alphanumeric formats, data that is produced via algorithmic processing aka analytics, e.g., vehicle counts culled from traffic cameras, are not to be referred to with “cameras” in the name. Instead the resulting data must be described ala “Vehicle Counts In Greater New York City” where the source is not important to the metadata name. Such information can be included within the Description field at any level of specificity.

- For metadata instances being produced for use within a Branded Data Exchange, additional information may be included in the Name field if requested by the client-organization. This could include specifics not found on the main Terbine system, e.g., “Chevy Volt Hard Braking / Greater Chicago / 2016” or similar.

DESCRIPTION

Purpose: The Description field provides a key means for an automated process or human user to find metadata instances that are being sought, however the main purpose is to provide a human user with additional contextual and bounding information relative to narrowing a search. The Description field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user. With the latter, it will be rendered within a Metadata Instance "tile" within the Terbine GUI and/or Branded Data Exchange GUI(s).

Convention: Descriptions can be verbose but must be as succinct as possible to convey salient information about the datasets or streams that match with the given metadata instance. All terms and place- or source-names must be commonly understood by native or fluent English speakers. This field is where details that will not be found in other fields can be included, such as machine type or grouping names (if not found in Originator, Owner or Container fields), e.g. "Sonobuoys anchored in the Chesapeake Bay" or "Vehicles employed in ride-sharing" or "Orbital inclinations around the Equator" etc. It is important to research 'around' the source data to find clues as to its provenance and context that can be conveyed via the Description field (i.e. those which cannot be readily entered into other fields). Suffixed to all Descriptions must be an indication of whether the datasets or streams have been processed, or are raw/unprocessed (see examples).

Syntax: There are no restrictions on Description syntax other than those specified below. Spelling, grammer and punctuation are to be rendered just as with an article in a scholarly paper. If an acronym is to be included, it must be defined (see examples). At all times common sense must be applied to authoring, assume that a program may be the reader.

Prohibited/Restricted Nomenclature:

- Naming of Sensors, Owners, Originators, date-ranges, addresses or other **elements that are covered by other/specific metadata fields**.
- **Use of conversational language or acronyms**, as a program accessing the system via the API may be performing the query (vs. a human user).
- **Terms or language that are so broad** as to potentially produce false-positives.
- **Citations**, i.e., information regarding acknowledging of sources, legal items not covered by the system-level master agreement and related, are not to be included in the Description field. **These belong solely in the Citation metadata field**.

Examples:

(Name field) Vehicle-sampled Ambient Air Temperature / United States nationwide

(Description field) Timestamped & geolocated samplings of outside air temperature on vehicles equipped with the Geotab On Board Diagnostics (OBD-II standard) capture / transceiver device. Fleet sampling of over 150.000 vehicles of varying manufacturers in various geographies. Vehicle identification codes anonymized, otherwise unprocessed.

(Name field) Vehicle Headlight Status / United States nationwide

(Description field) Timestamped & geolocated status (on/off) of headlights on vehicles equipped with the Voyomotive On Board Diagnostics (OBD-II standard) capture / transceiver device. Fleet sampling of over 500 vehicles in various geographies.
(Name field) Vehicle Headlight Status / United States nationwide. Unprocessed.

(Name field) Atmospheric Density / Satellite-Based

(Description field) Measurements of cloud density in the upper atmosphere captured by a fleet of twelve Low Earth Orbit (LEO) satellites operating in the near-infrared range. Readings cover 96% of the Earth's surface, captured on 90-minute intervals. Processed from raw inputs to an accuracy of +/-86% on average.

(Name field) Ocean Depths / Global

(Description field) Low-period depth soundings from surface vessels operating outside of international boundaries in all major oceans. Period varies by locale. Unprocessed.

Notes:

- Being that the system is intended only for machine-generated data in numeric or alphanumeric formats, data that is produced via algorithmic processing aka analytics, e.g., vehicle counts culled from traffic cameras, are not to be referred to with "cameras" in the name. Such information regarding analytical pre-processing can be included within the Description field at any level of specificity.

ORIGINATOR

Purpose: The Originator field provides specific names of entities responsible for the machines/sensors emanating the datasets or streams that the given metadata instance maps to. The Originator field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: The field can contain one or more names of originating entities. These will initially be hand-entered. Eventually a sufficient number of entity-names will be recorded within the system to allow for drop-down lists or selection of entity-names via corresponding codes via the API. Such codes will be the same as those generated by the system for organizations which are members of the system. The code will not be visible via the GUI in the general case (but can be in the supervisory case ala control consoles).

Syntax: Entity-names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Initials are to be avoided unless extremely common (e.g., "IBM" or "USPS" or "DHL"). Generally, it is desirable to utilize full names (e.g., "General Motors" or "US Department of Energy") so as to increase the likelihood that future users will recognize the entity. Multiple entity-names must be ordered ala parent-subsidiary-subsidiary.

Prohibited/Restricted Nomenclature:

- **Suffixes** such as "Inc." "Corp" "PLC" "Ltd" and other corporate designators, or those for public entities such as "Agency" (unless specifically part of the formal name). "City of" is not normally needed unless there is also a county or country of the same name.
- **Acronyms** (as opposed to initials per the syntax description above).
- **Slang or popular names** for entities (e.g., "Post Office" where the proper name is "US Postal Service" or "USPS").
- **Subsidiary names should not be used standalone** (e.g., "Chevrolet" alone would be inappropriate when the true corporate entity is General Motors).
    - An example involving a commercial organization would be finding that the datasets in question originate from the Pratt & Whitney division of United Technologies thus "United Technologies" followed by "Pratt & Whitney" where both should be included with the 'senior' or 'parent' organization being entered in the primary field, followed by the subsidiaries in nested order.
    - An example involving a governmental organization would be "US Department of Defense" followed by "US Navy"
    - An example involving an educational organization would be "University of Nevada, Las Vegas" followed by "Howard R. Hughes College of Engineering" followed by "Drones and Autonomous Systems Lab" designating the actual sub-organization responsible for generating the data

OWNER

Purpose: The Owner field provides specific names of entities who own the legal rights to the datasets or streams that the given metadata instance maps to. The Owner field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: The field can contain one or more names of owning entities. These will initially be hand-entered. Eventually a sufficient number of entity-names will be recorded within the system to allow for drop-down lists or selection of entity-names via corresponding codes via the API. Such codes will be the same as those generated by the system for organizations which are members of the system. The code will not be visible via the GUI in the general case (but can be in the supervisory case ala control consoles).

Syntax: Entity-names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Initials are to be avoided unless extremely common (e.g., "IBM" or "USPS" or "DHL"). Generally, it is desirable to utilize full names (e.g., "General Motors" or "US Department of Energy") so as to increase the likelihood that future users will recognize the entity. Multiple entity-names must be ordered ala parent-subsidiary-subsidiary.

Prohibited/Restricted Nomenclature:

- **Suffixes** such as "Inc." "Corp" "PLC" "Ltd" and other corporate designators, or those for public entities such as "Agency" (unless specifically part of the formal name). "City of" is not normally needed unless there is also a county or country of the same name.
- **Acronyms** (as opposed to initials per the syntax description above).
- **Slang or popular names** for entities (e.g., "Post Office" where the proper name is "US Postal Service" or "USPS").
- **Subsidiary names should not be used standalone** (e.g., "Chevrolet" alone would be inappropriate when the true corporate entity is General Motors).
    - An example involving a commercial organization would be finding that the datasets in question originate from the Pratt & Whitney division of United Technologies thus "United Technologies" followed by "Pratt & Whitney" where both should be included with the 'senior' or 'parent' organization being entered in the primary field, followed by the subsidiaries in nested order.
    - An example involving a governmental organization would be "US Department of Defense" followed by "US Navy"
    - An example involving an educational organization would be "University of Nevada, Las Vegas" followed by "Howard R. Hughes College of Engineering" followed by "Drones and Autonomous Systems Lab" designating the actual sub-organization responsible for generating the data

SOURCE NAME

Purpose: The Source Name field generally provides the name of the product, system or bounded locale generating the datasets or streams associated with the given metadata instance. The Source Name field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Source Name can be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of source-names will be recorded within the system to allow for drop-down lists or selection of source-names via corresponding codes via the API.

Syntax: Source Names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Initials are to be avoided unless extremely common (e.g., "IBM" or "BMW"). Generally, it is desirable to utilize full names (e.g., "General Electric" or "Johnson Controls") so as to increase the likelihood that future users will recognize the entity. The producer or bounded locale of the product or system should come first, followed by a space and then the remainder of the string to provide further identification. Brevity is important as well as providing a complete name for search purposes (see examples).

Prohibited/Restricted Nomenclature:

- **Acronyms** (as opposed to initials per the syntax description above).

Examples:

- **Machine or system names** to include factory equipment, infrastructure components, vehicles e.g. "BigCo Model 255C Tractor Unit" or "Planet Labs Low Earth Orbit Infrared Satellite Constellation 12"
- **Bounded locales** taken as a whole are acceptable where the sensors emanating data are situated (only if all sensors are contained within that bounded locale, e.g. "Kennedy Space Center Launch Pad 39A" or "Atlanta Suburban Housing Project 114"). *Note this is naming of the Source as a bounded locale for indexing purposes, as opposed to Location and Container.*

SOURCE TYPE

Purpose: The Source Type field generally provides categorical (often using physics, chemical or environmental nomenclature) typing for sensors or subsystems (if the sensors cannot be separated from same) associated with the given metadata instance. The Source Type field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Source Type can be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of Sensor Types will be recorded within the system to allow for drop-down lists or selection of Sensor Types via corresponding codes via the API.

Syntax: Source Types must be researched to ensure that the most common representation is entered, for the avoidance of doubt in searching. These will typically consist of one or two words, occasionally three. See examples.

Prohibited/Restricted Nomenclature:

- **Acronyms or shorthand-terms** (unless in common usage, e.g. “ABS” or “DC”).
- **Sensor Types** which are covered by the Sensor Type field. The difference between “Source” aka “Sensing” and “Sensor” is exemplified as follows:
    - Source Type = *temperature* (as what’s being measured)
    - Sensor Type = *thermometer, thermocouple* (as the means to measure)

Examples:

- Physical attributes, e.g., “temperature” or “wind speed” or “audio levels”
- Subsystem typing is allowed, e.g., “ocean currents” or “traffic counts” wherein the output is gained from more than one sensor housed within the subsystem, and the output from the individual sensor-outputs are aggregated or otherwise collated into a single stream emanating from that subsystem.

SENSOR TYPE

Purpose: The Sensor Type field generally provides specific (often using electro-mechanical nomenclature) typing for sensors or subsystems (if the sensors cannot be separated from same) associated with the given metadata instance. The Sensor Type field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Sensor Type can be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of Sensor Types will be recorded within the system to allow for drop-down lists or selection of Sensor Types via corresponding codes via the API.

Syntax: Sensor Types must be researched to ensure that the most common representation is entered, for the avoidance of doubt in searching. These will typically consist of one or two words, occasionally three. See examples.

Prohibited/Restricted Nomenclature:

- **Acronyms or shorthand-terms** (unless in common usage, e.g. "LIDAR").
- **Sensing Types** which are covered by the Source Type field. The difference between "Sensor" and "Source" aka "Sensing" is exemplified as follows:
    - Sensor Type = *thermometer, thermocouple* (as the means to measure)
    - Source Type = *temperature* (as what's being measured)

Examples:

- Sensor typing ala "thermocouple" or "torque transducer"
- Subsystem typing ala "microphone array" or "sonobuoy" wherein the output is gained from more than one sensor housed within the subsystem, and the output from the individual sensor-outputs are aggregated or otherwise collated into a single stream emanating from that subsystem.

MANUFACTURER

Purpose: The Manufacturer field generally provides the name of the manufacturer or systems integrator for all of the sensors or sensor-platforms associated with the given metadata instance. The Manufacturer field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Manufacturer name can be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of manufacturer-names will be recorded within the system to allow for drop-down lists or selection of manufacturer-names via corresponding codes via the API.

Syntax: Manufacturer names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Initials are to be avoided unless extremely common (e.g., "IBM" or "BMW"). Generally, it is desirable to utilize full names (e.g., "General Electric" or "Johnson Controls") so as to increase the likelihood that future users will recognize the entity. Multiple entity-names must be ordered ala parent-subsidiary-subsidiary.

Prohibited/Restricted Nomenclature:

- **Acronyms** (as opposed to initials per the syntax description above).
- **Subsidiary names should not be used standalone** (e.g., "iPad" alone would be inappropriate when the true corporate entity is "Apple Computer").
    - An example involving a commercial organization would be finding that the datasets in question originate from the Harman division of Samsung thus "Samsung" followed by "Harman" where both should be included with the 'senior' or 'parent' organization being entered in the primary field, followed by the subsidiaries in nested order.

Examples:

- Manufacturer names such "Samsung" or "Bosch" or "Borg Warner"

MAKE

Purpose: The Make field generally provides the specific brand or subbrand of the manufacturer or systems integrator for all of the sensors or sensor-platforms associated with the given metadata instance. The Make field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Typically, one Make name will be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of make-names will be recorded within the system to allow for drop-down lists or selection of make-names via corresponding codes via the API.

Syntax: Make names must be researched to ensure that the most accurate representation is entered, for the avoidance of doubt in searching. Generally, it is desirable to utilize full names (vs. initials, acronyms or shorthand) so as to increase the likelihood that future users will recognize the entity. Multiple/nested make-names must be ordered as parent-subbrand.

Prohibited/Restricted Nomenclature:

- **Acronyms** (as opposed to common initials per the syntax description above).

Examples:

- “Galaxy” (where the Manufacturer field has “Samsung”)
- “SmarTemp” then “Pro Series” (where the Manufacturer field has “Honeywell”)

MODEL

Purpose: The Model field provides the precise model-designation for all of the sensors or sensor-platforms associated with the given metadata instance. The Model field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Model name will be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of model-names will be recorded within the system to allow for drop-down lists or selection of make-names via corresponding codes via the API.

Syntax: Model names must be researched to ensure that the most accurate representation is entered, for the avoidance of doubt in searching. The field must contain only the precise name as published by the entity named in the Manufacturer field, and subordinate to the name(s) provided in the Make field. Sub-models can be supplied in the order they are typically presented by the Manufacturer (see examples). Model names often include numbers, these must be associated with any alpha characters with appropriate spaces and/or punctuation as per the Manufacturer's publication of same (see examples).

Prohibited/Restricted Nomenclature:

- **Acronyms** (as opposed to model-name/number initials).

Examples:

- "S9 Edge" (where the Manufacturer field has "Samsung" and the Make field has "Galaxy")
- "201B" (where the Manufacturer field has "Honeywell" and the Make fields have "SmarTemp" sub "201 Series")

VERSION

Purpose: The Version field provides the exact version level for all of the sensors or sensor-platforms associated with the given metadata instance. The Version field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Version-name will be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of version-names will be recorded within the system to allow for drop-down lists or selection of version-names via corresponding codes via the API.

Syntax: *While the term "name" is used for consistency within these Guidelines, in practice versions tend to be expressed as numbers, often with multiple digits beyond a decimal point.* Some will contain one or more alpha characters as well or in lieu of numbers. These must be researched to ensure that the most accurate representation is entered, for the avoidance of doubt in searching. The field must contain only the precise version indication as published by the entity named in the Manufacturer field, and subordinate to the name(s) provided in the Make and Model fields. Version names must be entered precisely, with all characters having appropriate spaces and/or punctuation as per the Manufacturer's publication of same (see examples). Do not include "V" or "V." or "Version" or "Version No." or "#" etc. in this field, only the Version name itself.

Examples:

- "2.0101b" (where the Manufacturer field has "Samsung", the Make field has "Galaxy" and the Model field has "S9 Edge")
- "AF5.772" (where the Manufacturer field has "Honeywell", the Make fields have "SmarTemp" sub "201 Series" and the Model field has "201B")

INTERNATIONAL STANDARD INDUSTRY CLASSIFICATION (ISIC) CODES

Purpose: The ISIC field provides generally accepted codes indicating the field(s) of industry or industry sectors within which the Originator associated with the given metadata instance operates. The ISIC field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: One or more ISIC codes can be selected/specified for a given Originator. It is common with large multinational corporations that they operate in or across multiple industry sectors. An understanding of ISIC codes can be gained here.

Syntax: Only the code(s) are to be entered into the field. Multiple codes are to be separated by comma+space if done manually.

Exceptions:

If the rare case that the Originator cannot be fit into one of the ISIC categories, leave the field blank / no selection.

LOCATION

Purpose: The Location fields provide for bounding of the geo-location area for all of the sensors or sensor-platforms associated with the given metadata instance. All of the Location fields (primary- and sub-fields) can be discovered via API-based queries. They can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Location Type, Subtype and Category can be referenced within a given metadata instance. All three fields will be contained within the system to allow for drop-down lists or selection of field-designators via corresponding codes via the API. Entries into each field will typically be hand- or API-entered.

Syntax & Usage:

- Location Type. A trinary choice of "Fixed" or "Moving" or "Other" to indicate that the machines, sensors, systems producing the datasets or streams associated with the given metadata instance are situated at permanent/stationary locations or mounted onto/inside of machines (e.g. automobiles, aircraft, trains) or platforms (e.g. barges) that can move. The third choice being "Other" is for when the ability of the given platform(s) to be mobile is unknown, or where Fixed or Moving don't make sense (primarily, satellites or other spacecraft whereas aircraft do count as Moving).
- Location Subtype (Fixed). Representing the specific location for the sensors or subsystems generating datasets or streams associated with the given metadata instance, e.g. the location of a wind or solar farm, a radar installation or a factory. When the Location Type "Fixed" is selected, the Subtype for fixed locations is invoked. This field provides two top-level selections along with subordinate choices, being
    - Address. A conventional street address as are used in postal deliveries and related. These are to be researched and tested prior to copy and pasting or hand-entering using commonly accepted tools such as Google Maps or (if situated within the United States) the usps.gov website, to ensure that the address information is complete and accurate.
        - Elements can include alphanumerical address, street name, city, state/province, postal code, country code (not country name / the country code must be provided in the standard ISO Alpha-2 format).
        - Any of the elements may be used without the others being present if the remaining address elements are unknown.
        - If the sensors or subsystems which generate the datasets or streams represented by the given metadata instance are found within a region that can be logically bounded via the use of a city, state/province, postal code or country code(s), then only those can be provided and the finer-detail fields such as street address deliberately omitted.

    - LAT/LON. Latitude and longitude must be supplied applying the Decimal Degrees system of coordinates. These to be entered precisely, applying the ISO6709 standard. The standard also allows for the encoding of altitude, height, and depth. Subordinate selections are:
        - Radius. Expressed in meters to describe the distance from a central point that sensors associated with the metadata instance are situated.
        - Bounding Box. These entries must provide coordinate-extents. An example can be found here.
        - Polygon Coordinates. Typically a suite of five (5) lat/lon values to define the extents of a region, facility or other area where the shape-definition does not fit a box or circle. A tool providing examples can be found here.
- Location Subtype (Moving). Representing the electromechanical means by which coordinates for the moving sensors or subsystems generating datasets or streams associated with the given metadata instance are produced. When the Location Type "Moving" is selected, the Subtype for moving sensors or subsystems is invoked. Given that all datasets or streams associated with this Subtype will provide individual coordinates with each data entry, coordinates are not included in the metadata instance itself when this Subtype is selected. This field provides three Subtype choices:
    - Device GPS. Indicating that the emanating coordinate-data is produced by an internal/onboard GPS receiver. Therefore all sensors or subsystems associated with the given metadata instance will provide each their own GPS readings to accompany individual data entries.
    - System Level. Indicating that the coordinate-data included with the sensor-data emanating from sensors or subsystems associated with the given metadata instance is generated externally to the sensor-platforms themselves, e.g. by signal-triangulation (other than GPS).
    - Algorithm Generated. Indicating that the coordinate-data included with the sensor-data emanating from sensors or subsystems associated with the given metadata instance is generated via deduction, interpolation or extrapolation within the context of the sensors or subsystems themselves but not using GPS, e.g., onboard ranging ala radar, LIDAR or ultrasound.
- Location Category. A choice of categorical references to indicate that the machines, sensors, systems producing the datasets or streams associated with the given metadata instance are operated at or within one of the four fundamental environments. This field is independent from whether the sensor data is being generated from fixed or moving locations. This field provides four choices:
    - Terrestrial. Indicating that the sensors or subsystems associated with the given metadata instance operate on land. This Category can also pertain to subterrean sources, e.g. mining equipment, pipelines or subway trains.

- Aerial. Indicating that the sensors or subsystems associated with the given metadata instance operate in the air. This Category pertain to any type of flying machine or machines to include powered and unpowered airplanes, UAVs, lighter-than-air craft such as diribles or sounding balloons, rockets that do not breach the atmosphere.
- Marine. Indicating that the sensors or subsystems associated with the given metadata instance operate on or below the oceans or other large bodies of water (e.g., the Great Lakes or the Caspian Sea). This Category pertain to any type of waterborne machine or machines to include engine-powered and wind- or solar-powered vessels, unmanned marine systems (UMS), submarines, buoys. The Marine category does not apply to large fixed infrastructural elements such as dams (these would fit under the Terrestrial category), however wind farms based in oceans are acceptable.
- Spaceborne. Indicating that the sensors or subsystems associated with the given metadata instance operate in orbit around the Earth, another celestial body (e.g., the Moon or Mars) or deep space. The Spaceborne category applies to spacecraft such as probes, satellites, manned capsules or spaceplanes or sounding rockets that breach the atmosphere.

Prohibited/Restricted Nomenclature:

- For Location Type (Fixed) the conventions noted in the description herein must be adhered to.
- For other fields associated with Location, entries are multiple choice only.

CONTAINER

Purpose: The Container field provides information regarding the physical encasement (aka "platform") for all of the sensors or sensor-platforms associated with the given metadata instance. **In this case the word "container" is to be considered as literal,** not in the programmatic sense. The Container field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Container-names will initially be hand-entered. Eventually a sufficient number of container-names will be recorded within the system to allow for drop-down lists or selection of container-names via corresponding codes via the API. It is critical that Containers and Subcontainers are only referenced (i.e., entered) when all of the sensors or sensor-platforms associated with the given metadata instance are 'contained' within the specified container-name(s). For example, if "building" is used, then all datasets or streams associated with that metadata instance must be generated by sensors contained solely within buildings. Furthermore, container-names must be commonly accepted terms (see examples) unless a specific exception is authorized.

Syntax: Container names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Initials are to be avoided unless extremely common (e.g., "HVAC"). Generally it is desirable to utilize full names so as to increase the likelihood that future users will recognize the entity (e.g., "Uninterruptible Power System" is preferable since "UPS" could also mean United Parcel Service or other depending upon context).

Where a subcontainer is named, it must be associated with a container in the same metadata instance *ala* Primary / Secondaries / Tertiaries (e.g. nested or chained containers which emanate all data associated with the metadata instance). Thus, nested container-names must be ordered ala outer container-subcontainer-subcontainer (i.e., "nested containers").

Prohibited/Restricted Nomenclature:

- **Acronyms** (as opposed to initials per the syntax description above).
- **Vendor or site names**, e.g. "Sony" or "Washington Monument"

Examples:

- (Primary) Container. Utilize commonly understood terms such as "Automobile" or "Building" or "Satellite" or "Aircraft" or "Traffic Signal" to describe the top-level container.
- (Secondary) Subcontainer. Terms which will tend to be of a technical nature and may be industry-specific are acceptable e.g. "Engine Control Unit" or "Elevator Controller" or "HVAC Unit" or "Anti-Lock Braking System".
- (Tertiary) Subcontainer. Aka 'drilling down towards the sensor,' i.e., to where the sensors are physically mounted or situated, e.g., "motherboard" or "probe" or "antenna" or "pod".

ENVIRONMENT

Purpose: The Environment field generally provides a succinct term for, or description of, the situation within which/where physical sensors are situated. In this case the word "environment" is to be considered as literal *ala* the environment in and around the location(s) of the sensors or sensor-platforms associated with the given metadata instance. The Environment field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: These will initially be hand-entered. Eventually a sufficient number of environment-names will be recorded within the system to allow for drop-down lists or selection of environment-names via corresponding codes via the API.

Syntax: Environment names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Multiple environment-names can be referenced within a given metadata instance, i.e., where a term for environment is provided, it can be associated with other environment-names within the same metadata instance, e.g. both "urban" and "indoors" can be true when the sensors are situated within a building located in a metropolitan area. Environment-names are not subordinated, i.e., if "mountain" and "tower" are both true, these are not parent-child to each other but are presented equally from a search perspective.

Prohibited/Restricted Nomenclature:

- **Acronyms** (unless very commonly understood).

Examples:

- Commonly understood terms are acceptable e.g. "Indoors" "Outdoors" "Aerial" "Marine" "Space" "Alpine" "Urban" or others such that a search will narrow the selections and improve the likelihood of a successful metadata discovery.

ADDITIONAL PROPERTY

Purpose: The additional property section provides the ability to define an industry specific extension to the standard metadata specification. It contains two sections. A structure definition that is a description of the schema of the industry specific section and a content section that contains the actual classifications in a structure as defined by the structure definition section.

Convention: These are highly dependent on the specific industry and input format and conventions are based on this definition.

Syntax: The form is based on an external definition or configuration for that section.

Prohibited/Restricted Nomenclature:

- **Free form configurations.** This must be strictly defined as per the specific industry classification.

Examples:

- An industry specific EV charging definition such as OCPP would fall under this section. The defined types would be described in the structure definition and the actual attributes used for classification would be in the content.

INTERACTOR

Purpose: The Interactor field provides a general catchall for environmental factors, machines and/or other "interactors" that can impact or influence sensor outputs including the sensors' data grade. Interactors will commonly be situated at, adjacent to, or encompassing the location(s) of the sensors or sensor-platforms associated with the given metadata instance. The Interactor field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: These will initially be hand-entered. Eventually a sufficient number of interactor-names will be recorded within the system to allow for drop-down lists or selection of interactor-names via corresponding codes via the API.

Syntax: Interactor names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Interactor names can consist of more than one word. Multiple interactor-names can be referenced within a given metadata instance, i.e., where a term for an interactor is provided, it can be associated with other interactor-names within the same metadata instance, e.g. both "sunlight" and "train noise" can be true when the sensors are subject to the effects of both. Interactor-names are not subordinated, i.e., if "airfield" and "winds" are both true, these are not parent-child to each other but are presented equally from a search perspective.

Prohibited/Restricted Nomenclature:

- **Acronyms** (unless very commonly understood).

Examples:

- General climatological conditions (as opposed to short-duration/weather), e.g., "windy" or "rainy" (with reference to the locale) or "high altitude" that can impact sensor outputs for farm equipment or road vehicles.
- Rapid vector-changes that can cause vibration affecting accelerometers in UAVs, therefore the interactor-name could be "directional shifts".
- Sun position causing exposed temperature sensors to skew, therefore the interactor-name could be "sun exposure".
- Large machinery that turns on and off could impact sensor readings on adjacent machines, e.g., the interactor-name could be "heavy machinery".

LEGAL TYPE

Purpose: The Legal Type field specifies categorically whether the datasets or streams associated with a given metadata instance are provided under open source (i.e., "public" or free to use without restriction or cost) or are considered to be proprietary, i.e., sourced from an entity claiming ownership who asserts limitations on use or a right to be paid for it (i.e., not open source). Typically, commercial or private entities will assert proprietary rights over the data, but not always; data provided by a government agency or entity acting on behalf of a government agency will typically be open, but not always. Non-profits, sometimes referred to as non-governmental organizations (NGOs) may also offer data that could be open source but might not be. The Legal Type field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Legal Type is presented as a binary choice between Open Source and Proprietary. Only one may be selected for a given metadata instance.

Syntax: No syntax direction is required for this field.

Prohibited/Restricted Nomenclature:

- Not applicable.

Examples:

- The selection of **Proprietary** would be made when research into the Owner(s) of the source datasets or streams pertaining to the given metadata instance, are notated or listed as being offered commercially or with other claims to rights.
- The selection of **Open Source** would be made when research into the Owner(s) of the source datasets or streams pertaining to the given metadata instance, are notated as being offered as such. Evidence of this status is to be provided in the Citation field.

REGULATORY TYPE

Purpose: The Regulatory Type field specifies categorically whether the datasets or streams associated with a given metadata instance are covered or otherwise affected by a commonly referenced and applied body of governmental regulations. The Regulatory Type field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: These will initially be hand-entered. Eventually a sufficient number of regulatory-names will be recorded within the system to allow for drop-down lists or selection of regulatory -names via corresponding codes via the API. Regulatory-names must be researched to ensure that the most common and verbose representation is entered, for the avoidance of doubt in searching. Multiple regulatory-names can be referenced within a given metadata instance.

Syntax: Regulatory-names can consist of more than one word.

Prohibited/Restricted Nomenclature:

Not generally applicable, however since the Terbine system is not intended to handle Personal Identifying Information (PII), e.g., *people's names, social security numbers, state ID numbers, personal addresses, phone numbers, personal userids or passwords, car VINs,* and other such items that could be found in datasets, then regulatory regimes such as GDPR and HIPAA should not apply. There may be exceptions to these when paid-for Branded Data Exchanges that operate with closed user groups are provided by Terbine to third parties. If and when these are developed, the Guidelines will be appended accordingly.

CITATION

Purpose: The Citation field provides for textual descriptions of any legal or other key statements provided by the commercial or private entity claiming ownership of the datasets or streams associated with a given metadata instance. The intention is for these to be retrieved if and as desired or warranted by a human user and is typically intended for discovery via a GUI-based manual search. The Citation field can also be discovered via API-based queries.

Convention: Citation is presented in purely textual form, with an unlimited string length. It is **only to be populated via copy-and-paste from the website of the data originator where the datasets or streams pertaining to the given metadata instance are located**.

Syntax: No syntax direction is required for this field as it is for copy-and-paste only.

Prohibited/Restricted Nomenclature:

- No restrictions generally.

Examples:

- When **Open Source** is selected in the Legal Type field, a citation to evidence the open source status should be copied into the Citation field, e.g., “Covered by Creative Commons 2.0”.

SCHEMA BODY

Purpose: The Schema Body field provides for textual descriptions of the schema for the datasets or streams associated with a given metadata instance. The intention is for these to be retrieved and reviewed by a human user and is typically intended for discovery via a GUI-based manual search. The Schema Body field can also be discovered via API-based queries.

Convention: Schema Body is presented in alphanumeric form, with an unlimited string length. It is only **to be populated via copy-and-paste from the website of the data originator where the datasets or streams pertaining to the given metadata instance are located**. Alternatively the fields (e.g., column-headings for files that are provided in CSV or XLS formats) can be copied across from an example datafile. These will initially be hand-entered. Eventually a sufficient number of schema body-names will be recorded within the system to allow for drop-down lists or selection of schema body -names via corresponding codes via the API. Therefore it is important to provide consistency with entries to ensure uniformity and eventual codification into selection-lists.

Syntax: Where possible, headings/schema elements should be comma-separated for parsing by software programs.

Prohibited/Restricted Nomenclature:

- No restrictions generally.

Examples:

- “Timestamp, lat, lon, alt, reading”

MEASUREMENT UNIT

Purpose: The Measurement Unit field provides specific (typically using engineering or scientific nomenclature) typing for all measurements associated with the given metadata instance. The Measurement Unit field can be discovered via API-based queries. It can also be discovered via a GUI-based manual search by a human user.

Convention: Only one Measurement Unit can be referenced within a given metadata instance. These will initially be hand-entered. Eventually a sufficient number of Measurement Unit types will be recorded within the system to allow for drop-down lists or selection of Measurement Unit types via corresponding codes via the API.

Syntax: Measurement Unit types must be researched to ensure that the most common representation is entered, for the avoidance of doubt in searching. These will typically consist of one or two words, occasionally three. See examples.

Prohibited/Restricted Nomenclature:

- **Acronyms or shorthand-terms** (unless in very common usage, e.g., "RPM").
- **Multiple terms for the same measurement**. This is more difficult to police but is doable. An example of the same term for a given sensor data output would be "Hertz" which is identical to "Cycles Per Second". It important to look into the system and determine whether such a term has already been used, and therefore apply that term in the field to avoid future problems with searches.

Plural names. Always use the singular (e.g., "Foot" vs. "Feet" for length) unless the common usage requires the plural (e.g., "Miles Per Hour" for velocity). Assume that when written out, a Measurement Unit would typically precede or follow a numerical data value, e.g., "250 Volts" whereas for the purpose of indexing (and therefore parsing/searching) the singular "Volt" is preferred.

Examples:

- Meter
- Ampere
- Feet Per Second
- Millibar
- Angstrom
- Decibel
- Lumen
- Gauss
- Liter

GROUP INSTANCE

GROUP INSTANCE NAME

Purpose: The Name field for Group Instances provides the primary means for an automated process or human user to find metadata instances that are being sought. Similar to the Single Instance schema, Group Instance Name field can be discovered via API-based queries or via GUI-based manual search by a human user. With the latter, it will be rendered at the top of a Metadata Instance "tile" within the Terbine GUI and/or Branded Data Exchange GUI(s). The Group Instance 'tile' will have the "GROUP" indicator on the top-right corner. Group Instances are distinguishable as it would contain a payload of individual Feeds instead of datasets.

Convention: Similar to Single Instances, the first word(s) of a Group Instance's Name must provide a clear, unambiguous indication of what the datasets or streams associated with the given metadata instance are. Ideally, a Group Instance's Name should be based on its constituent Single Instances. Thus, all rules and conventions applicable to Single Instances (e.g rules on length and independence of information conveyed) must also apply to the naming of Group Instances.

Syntax: Per the preceding, the first word(s) of a Name will be an indication of the sensing type of the feeds contained inside the Group. Following this can be a categorical label for the machines/sensors emanating the datasets or streams that the given metadata instance maps to. After this a separator "/" can be applied and optionally an acceptable location reference can be included (see below for restrictions). The specific syntax on Group Instance Naming are shown below.

TIME BASED GROUPING

Grouping by Year: Source Type / Location / Year Range (20xx-20xx)

LOCATION BASED GROUPING

Geographical Based Group Naming: Source Type / *Specific County, City, State, Country (Outside the US)* / (20xx)

Bodies of Water: Source Type / Certain Body of Water / Year (20xx)

Interstate Highways: Source Type / Highway name / Year (20xx)

Other Generic Location: Source Type / *Specific Location* / Year (20xx)

GROUP INSTANCE DESCRIPTION

Purpose: The Description field provides a key means for an automated process or human user to find metadata instances that are being sought, however the main purpose is to provide a human user with additional contextual and bounding information relative to narrowing a search. Similar to the *Name* field for Group Instances, all rules applied to a Single Instance's *Description* apply to Groups. The main departure however would be that Group Instances would have their Description field derived from the Description of their constituent Single Instances.

Convention: Similar to how Description works for Single Instances, Group Instance Description can be verbose but must be as succinct as possible to convey salient information about the datasets or streams that match with the given metadata instance. All terms and place- or source-names must be commonly understood by native or fluent English speakers. This field contains details that will not be found in other fields that can be included, such as machine type or grouping names (if not found in Originator, Owner or Container fields). Group Instance description must also contain items that may explain the context or significance of the grouping. Lastly, suffixed to all Descriptions must be an indication of whether the datasets or streams have been processed, or are raw/unprocessed (see examples).

Syntax: There are no restrictions on Description syntax other than those specified below. Spelling, grammar and punctuation are to be rendered just as with an article in a scholarly paper. If an acronym is to be included, it must be defined (see examples). At all times common sense must be applied to authoring, assume that a program may be the reader. Group Description may generally follow the description of its Single Instances if applicable. Group Instances must not contain information relevant only to a single constituent instance.

GROUP INSTANCE LOCATION

Purpose: The Location fields provide for bounding of the geo-location area for all of the sensors or sensor-platforms associated with the given metadata instance. In the case of Group Instances, the location field would be the shared location of all the individual feeds inside the Group Instance. Similar to the mechanics of Location for Single Feeds, all of the Location fields (primary- and sub-fields) can be discovered via API-based queries and via a GUI-based manual search by a human user.

Convention: Only one Location Type, Subtype and Category can be referenced within a given metadata instance. All three fields will be contained within the system to allow for drop-down lists or selection of field-designators via corresponding codes via the API. Entries into each field will typically be hand- or API-entered.

Syntax & Usage:

- Location Subtype (Fixed). Representing the specific location for the sensors or subsystems generating datasets or streams associated with the given metadata instance, e.g. the location of a wind or solar farm, a radar installation or a factory. When the Location Type “Fixed” is selected, the Subtype for fixed locations is invoked. This field provides two top-level selections along with subordinate choices, being
    - Address. A conventional street address as are used in postal deliveries and related. These are to be researched and tested prior to copy and pasting or hand-entering using commonly accepted tools such as Google Maps or (if situated within the United States) the usps.gov website, to ensure that the address information is complete and accurate.

    - Elements can include alphanumerical address, street name, city, state/province, postal code, country code (not country name / the country code must be provided in the standard ISO Alpha-2 format).
    - Any of the elements may be used without the others being present if the remaining address elements are unknown.
    - If the sensors or subsystems which generate the datasets or streams represented by the given metadata instance are found within a region that can be logically bounded via the use of a city, state/province, postal code or country code(s), then only those can be provided and the finer-detail fields such as street address deliberately omitted.
    - Addresses for Group Instances must be representative of the shared location of all Single Instances
- LAT/LON. Latitude and longitude must be supplied applying the Decimal Degrees system of coordinates. These to be entered precisely, applying the ISO6709 standard. The standard also allows for the encoding of altitude, height, and depth. For Group Instances, LAT/LON must represent the shared location of all Single Instances. Subordinate selections are:
    - Radius. Expressed in meters to describe the distance from a central point that sensors associated with the metadata instance are situated.
    - Bounding Box. These entries must provide coordinate-extents. An example can be found here.

Polygon Coordinates. Typically a suite of five (5) lat/lon values to define the extents of a region, facility or other area where the shape-definition does not fit a box or circle. A tool providing examples can be found here.