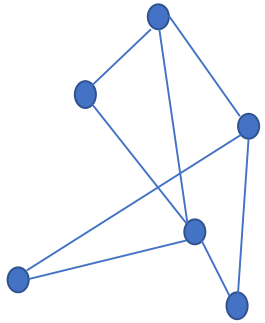
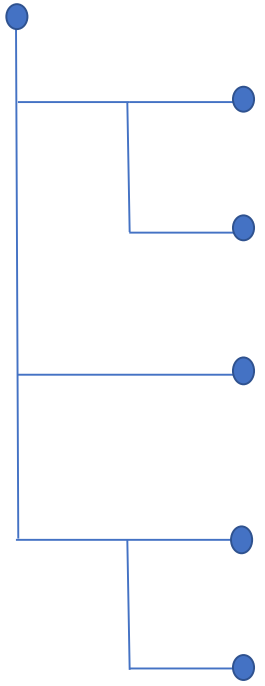


How to make a Storage Service Definition



Nicholas Car*

Data Storage Service Definitions WG, RDA Plenary 11

* Senior Experimental Scientist
CSIRO, Australia
nicholas.ar@csiro.au

Modern data model types 1/2

- UML
  - Standardised, platform-independent models
  - Need to be interpreted into another form for direct use (e.g. XML)
- Database schema
  - Entity-relationship diagrams etc.
  - Good for relational models only, not good for exchange
- Data exchange schema
  - XML, JSON etc
  - Good for exchange, not good for storage (but getting better: JSON DBs)

Modern data model types 2/2

- Semantic Web
  - Can be used for system-independent modelling & system-specific models
    - Due to exact implementations of RDF & SPARQL standards
  - Use sophisticated, standardized, RDF or OWL modelling
    - Better than UML class/object models!
  - Should inherit from/extend existing models
    - Part of the “Semantic Web”

Semantic Web model types 1/2

- Vocabularies v. Ontologies
  - Technically all instances of ontologies are vocabularies and all vocabularies are instances of an ontology
- Ontologies
  - A set of classes and relationships about an area of interest
  - Can use any one of a number of Semantic Web languages
    - All are based on RDF
    - In this list, each extends on the last
      - RDFS – basic claases & subclasses (hierarchies)
      - RDFS+c – plus a few OWL properties: xxxxxxxx
      - OWL – set theory-based modeling: unions, intersects etc.
      - OWL2 – improved OWL

Semantic Web model types 2/2

- Vocabularies
  - Usually, not always, purely hierarchical
  - Tend to use SKOS
    - SKOS itself uses OWL
    - A fairly simple ontology focused on term hierarchies
    - Contains only a few semantic relations for Concepts: closeMatch, exactMatch, xxx
    - Easy to cater for in tooling due to limited options

Voc or Ont for SSDefn?

- Can’t use just a vocab if we want to relate a series of very different concepts in incommensurate ways or in non-hierarchical ways
  - e.g. system geographic placement, latency, cost, policy features
- Could use SKOS vocabs for collections of commensurate terms
  - e.g. a hierarchy of different types of policy

Suggestion: use an OWL2 ontology for a main Storage System model, vocabularies for terms as needed

Catering for non-Sem Web systems

Like TOSCA templates, iRODS definitions, onedata system defns, vendor descriptions

- Implement an OWL model and provide mappings to others
- Publish the mappings as parts of the data model
- Implement converters, based on the mappings

I have done this many times before as all domains I've worked in have important non-Sem Web models that need supporting!

Straw man model: http://purl.org/storagesys

Storage Systems ontology

IRI:
http://purl.org/storagesys
Version IRI:
http://purl.org/storagesys/0.1
Authors:
Nicholas Car
http://orcid.org/0000-0002-8742-7730
Contributors:
Mikael Borg
Paul Millar
Research Data Alliance Working Group on Storage Service Definitions
Ontology source:
in turtle
in RDF/XML
Further documentation & examples:
This ontology's full documentation on GitHub

Abstract

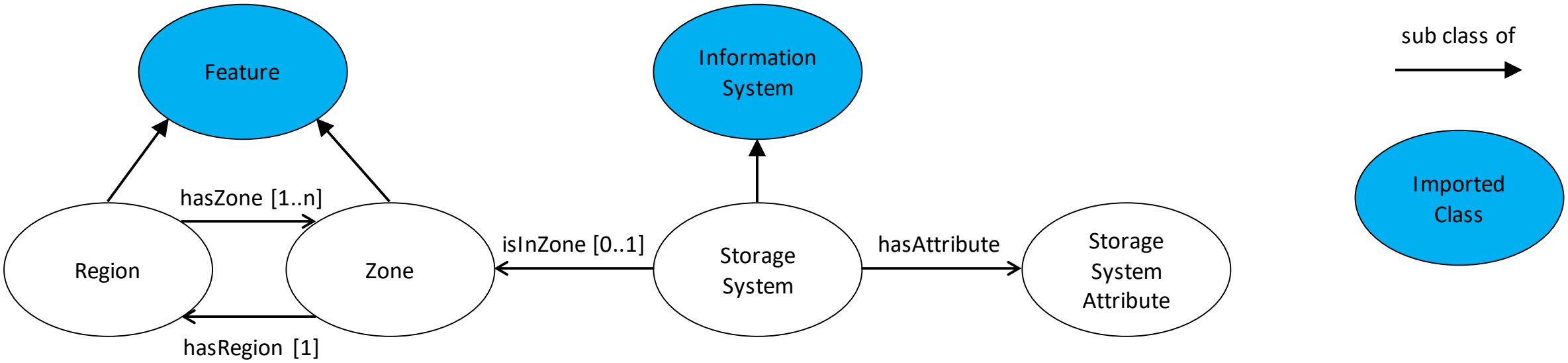
This ontology is designed to describe digital artefact storage systems in a vendor-neutral way so that gauged. This ontology fits within the broader context of other ontologies designed to describe records ma

Table of Content

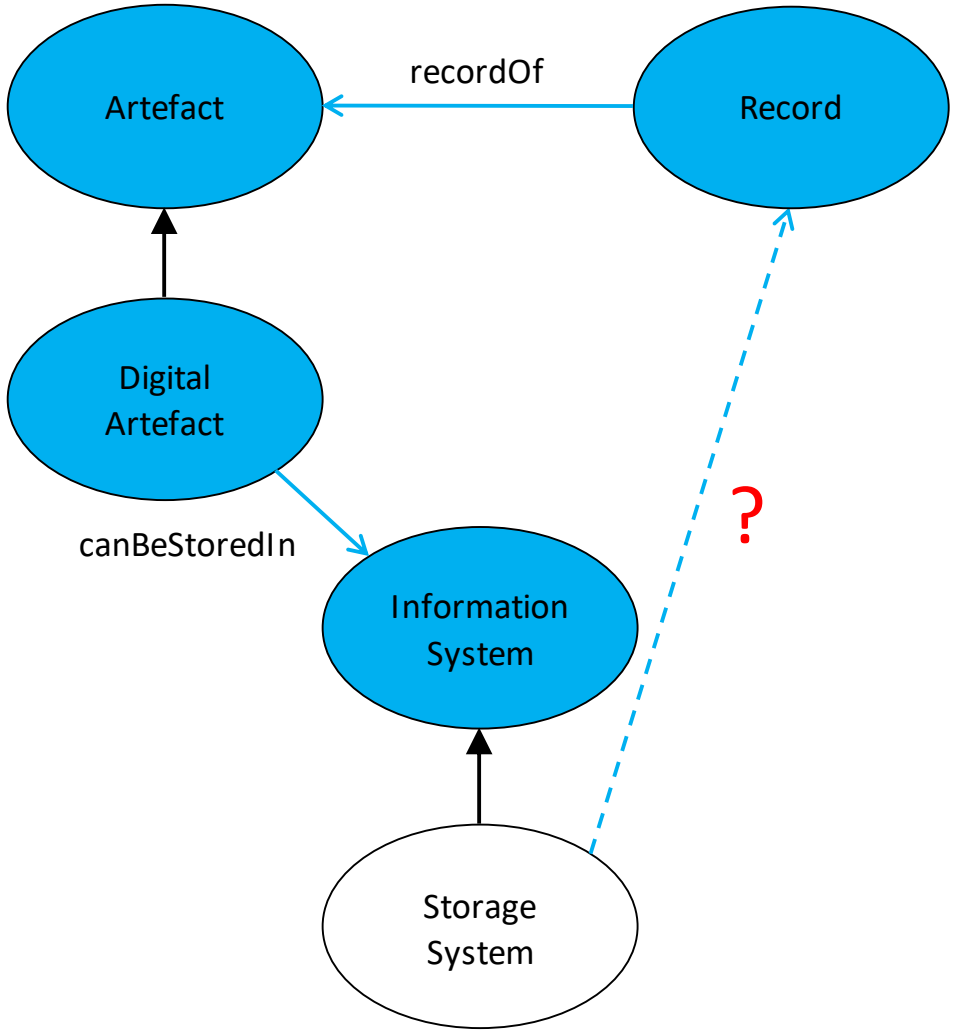
Classes

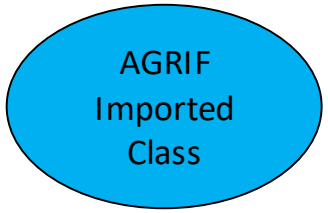
certification Digital Artefact feature geographic placement Information System in performance placement power network placement Redundancy retention Storage Region

Straw man model: http://purl.org/storagesys

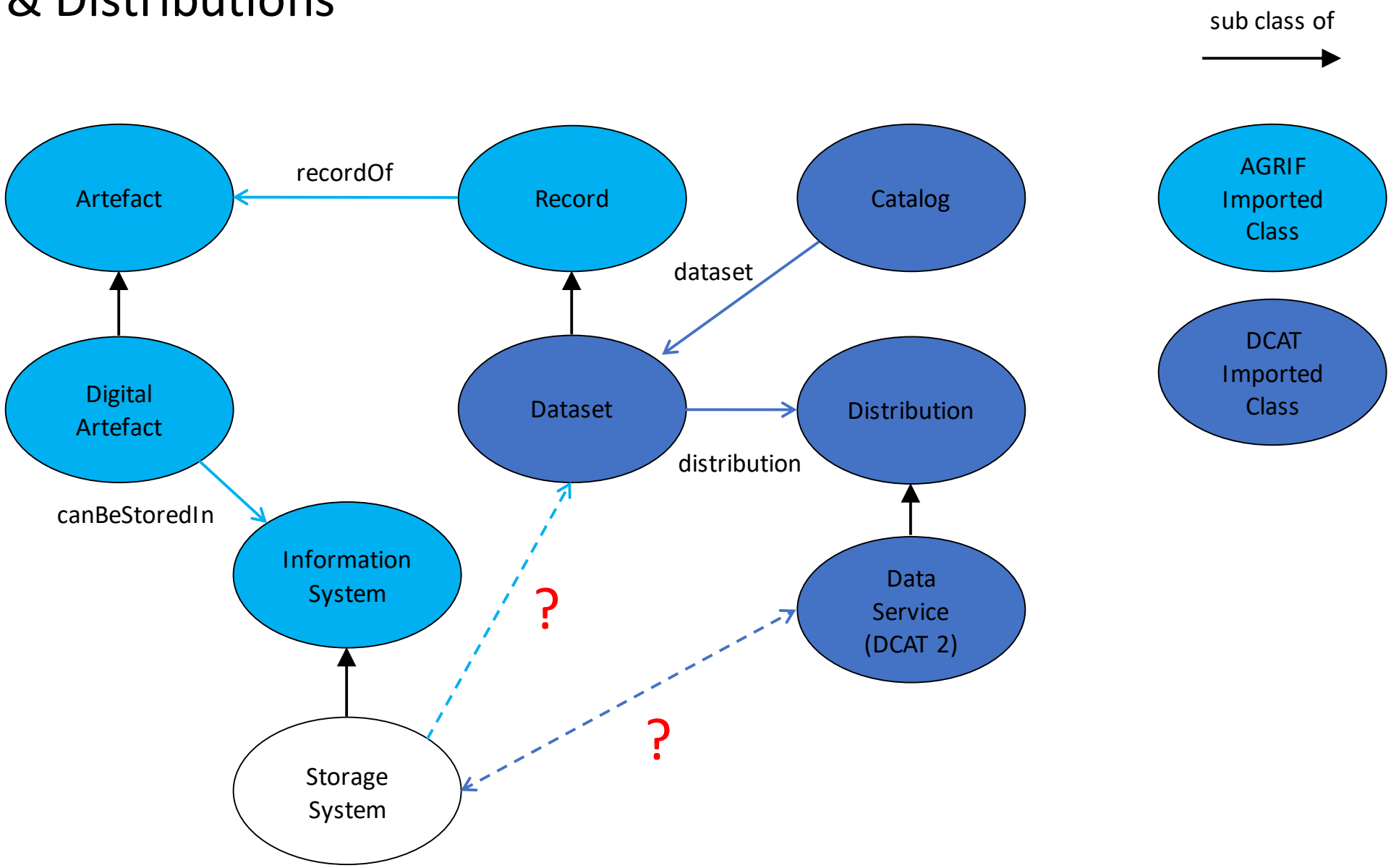


AGRIF: Artefacts & Records

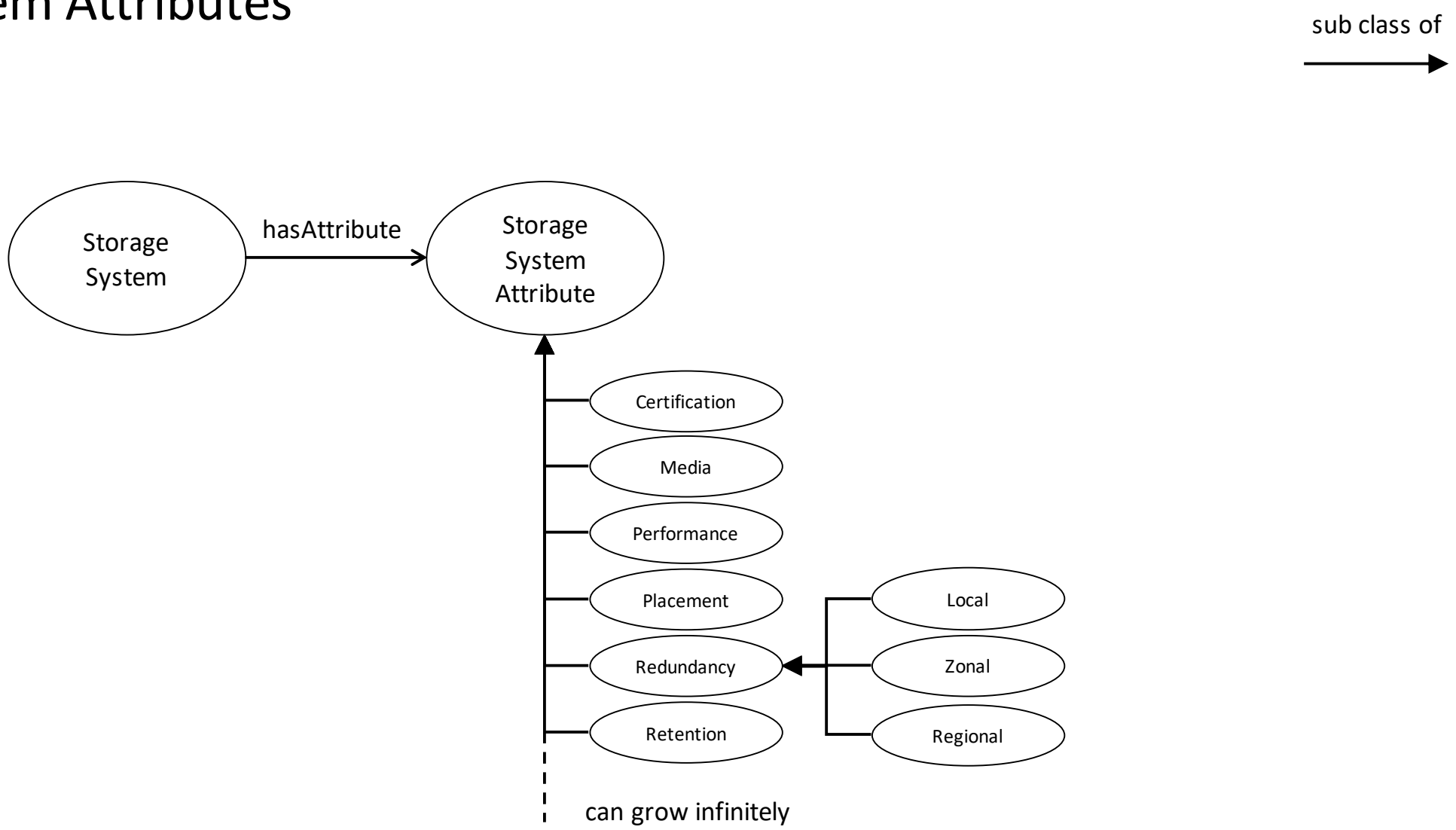




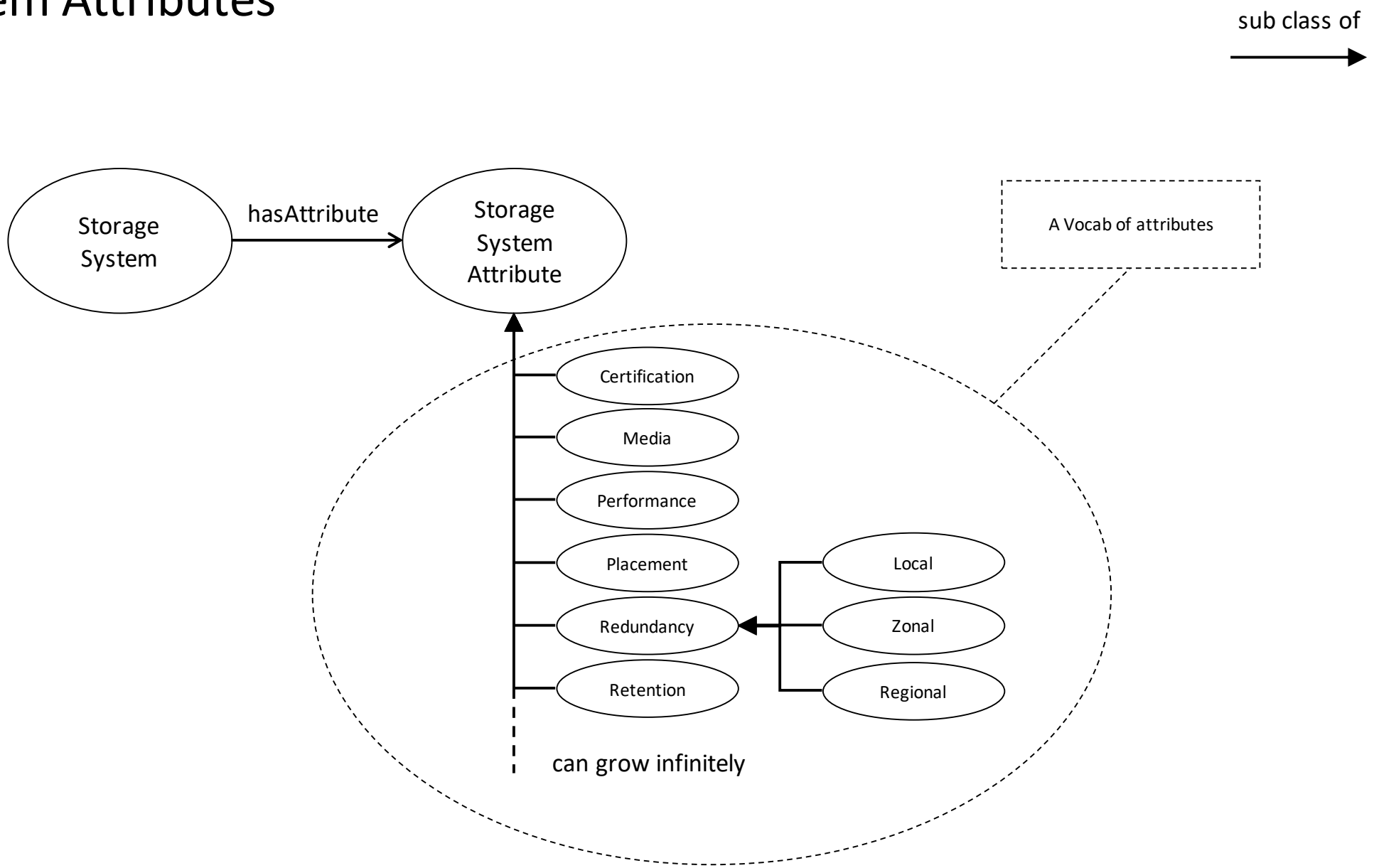
DCAT: Datasets & Distributions



Storage System Attributes

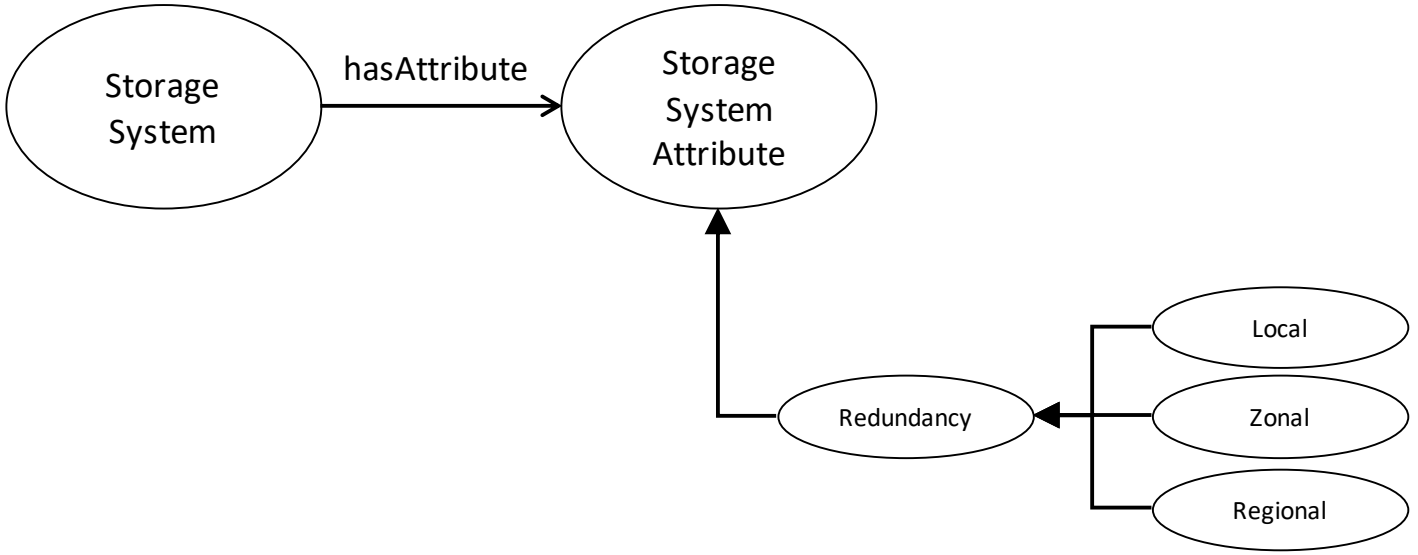


Storage System Attributes



Storage System Attributes: Redundancy

sub class of →



We can indicate that a system does, or that a system to be chose should, have Local, Zonal or Regional redundancy.

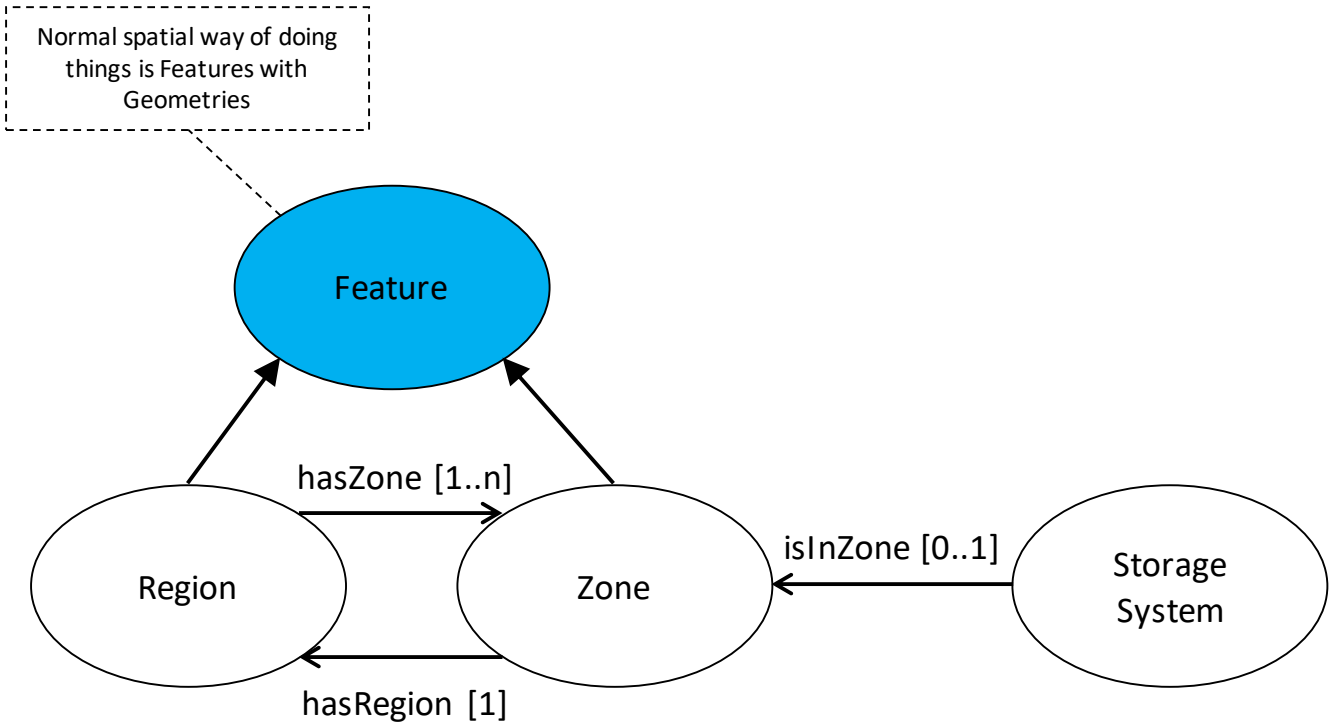
This may be purely numerical (“a Zonal redundancy of 2”) but may be Feature-specific (“…one copy in Rome, one in Paris”)

We can select/search for systems from a set of them, based on redundancy.

We can define what redundancy means elsewhere.

Geography

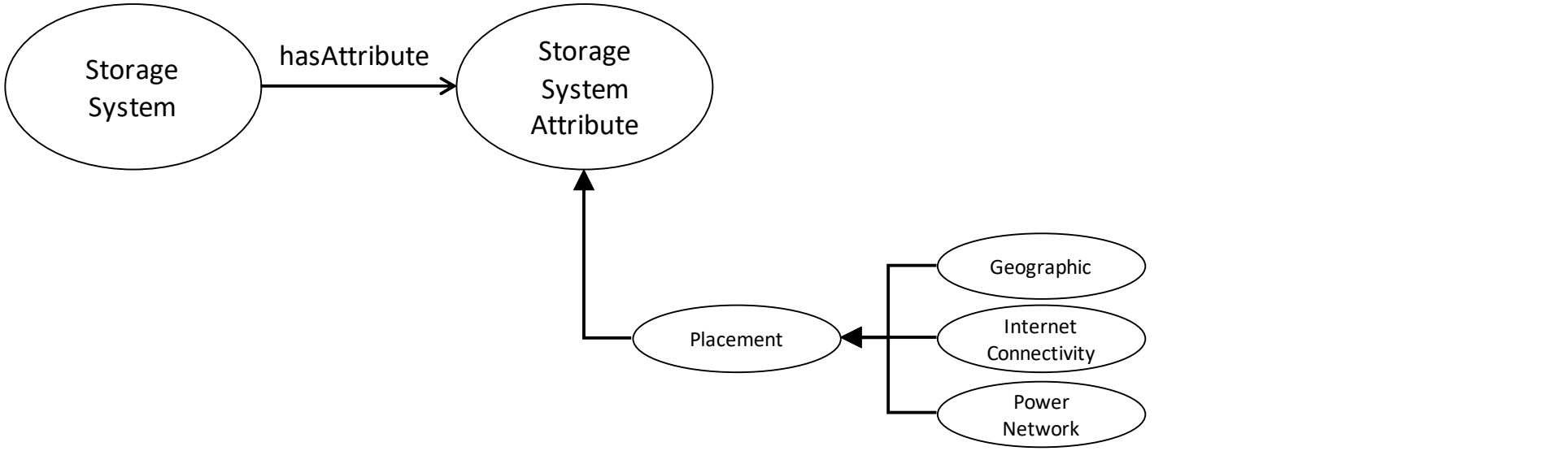
sub class of →



A particularly important system attribute

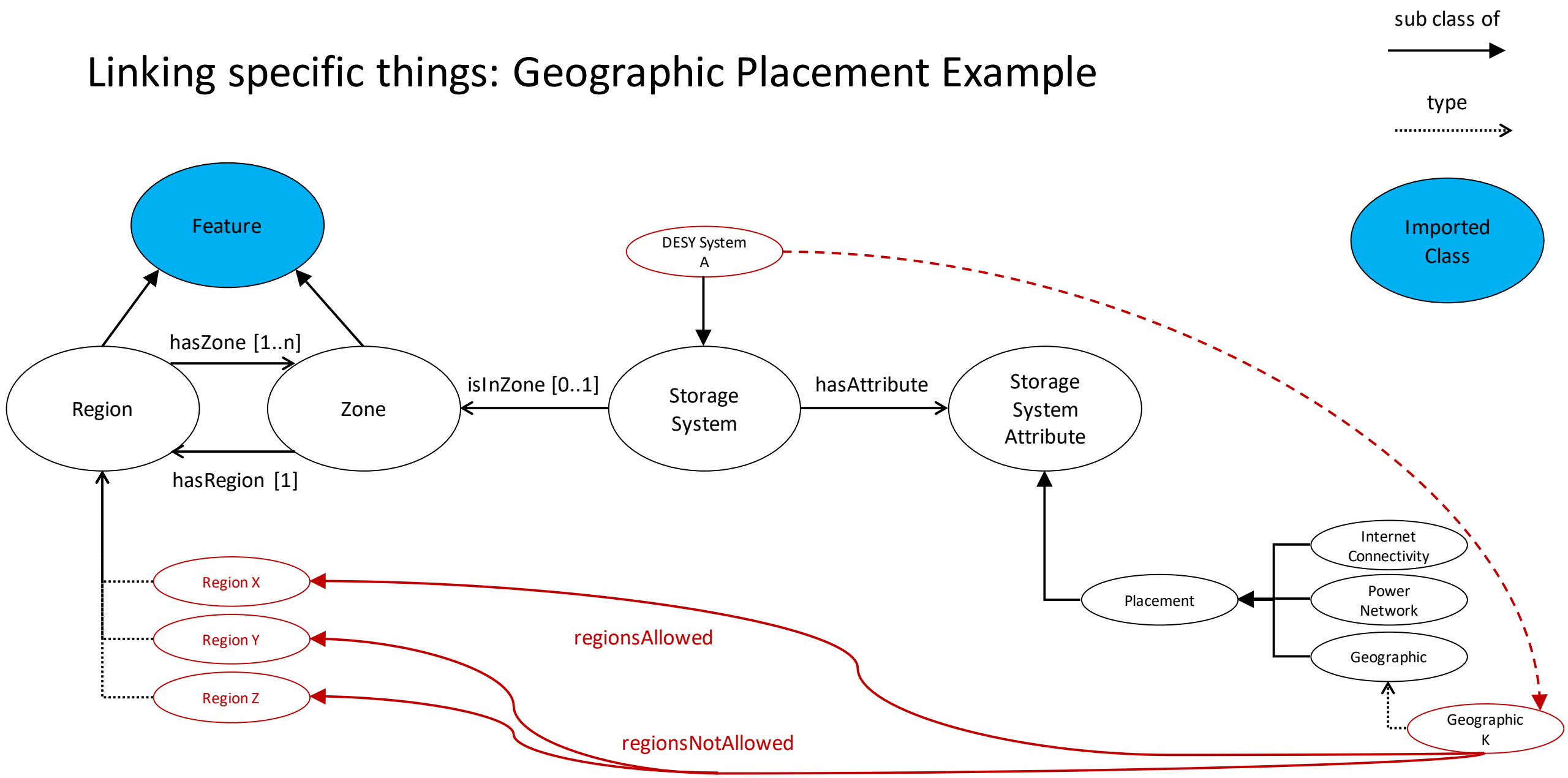
Should it be its own thing? Or just another Attribute?

Storage System Attributes: Placement



If Redundancy is numerical, we could use placement for specific Regions, Zones or other features like Power Grids etc.

Linking specific things: Geographic Placement Example



Requiring systems with specific characteristics

- If we have modelled in OWL, we can query for instances of a system with certain characteristics using SPARQL:
  - *A system with a zonal redundancy of 2 within the European East region and a write latency of less than 300 ms:*

```
SELECT ?sys
WHERE {
    # only geographic attributes can have regionsAllowed, and only Euro Easy wanted
    ?sys :hasAttribute/:regionsAllowed ex:EuropeanEastRegion .

    ?sys :hasAttribute [
        a :WriteLatency ;       # only systems with a :WriteLatency attribute recorded
        :maxLatency ?maxL       # we get the max write latency
    ]
    FILTER (?maxL < 300)        # only those max write latency < 300 (ms)
}
```

Mapping to TOSCA templates