The nature.com ontologies portal

---

nature.com/ontologies

Tony Hammond, Michele Pasin
Macmillan Science and Education

Who we are

We are both part of Macmillan Science and Education*

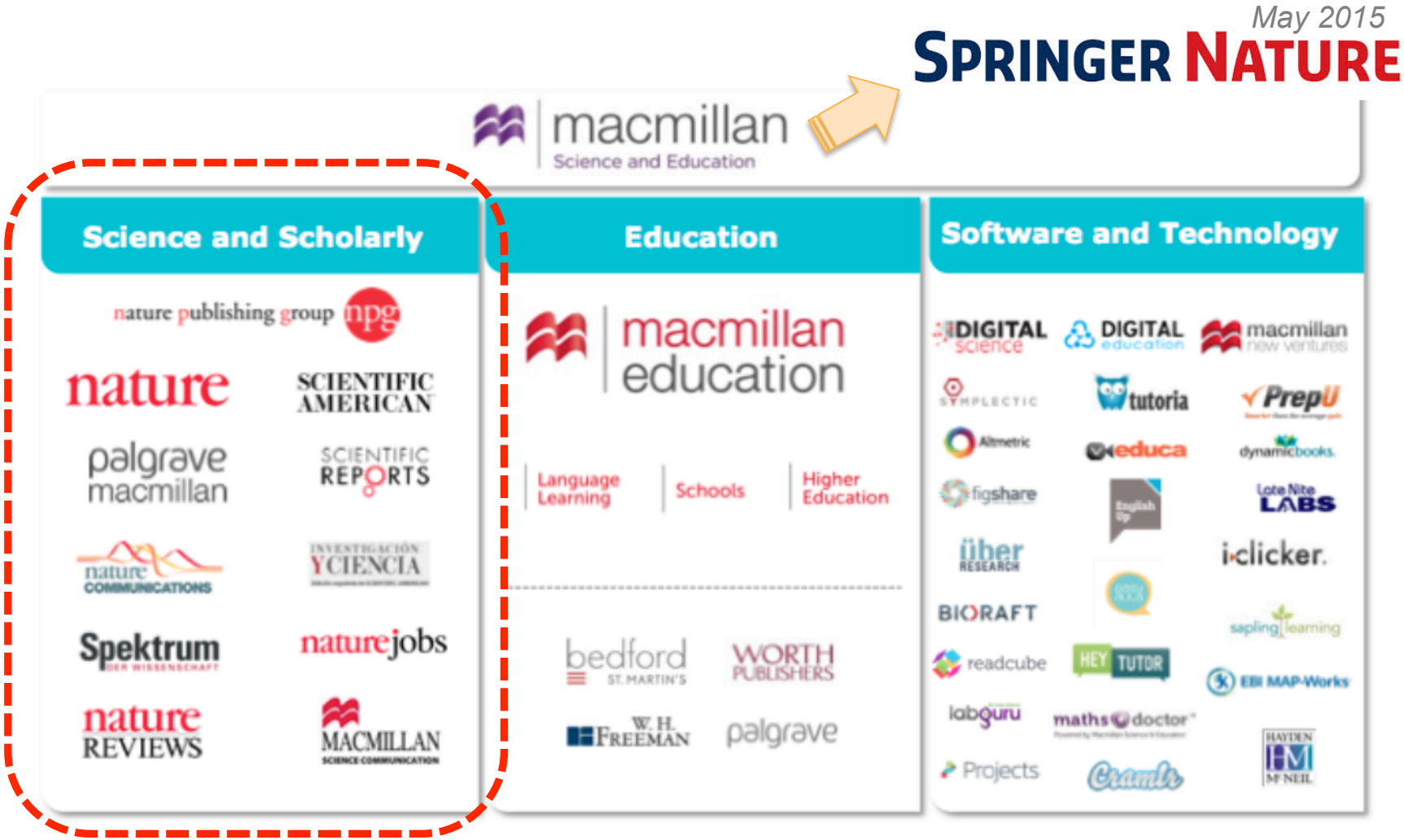
- Macmillan S&E is a global STM publisher

-  Tony Hammond is Data Architect, Technology
@tonyhammond

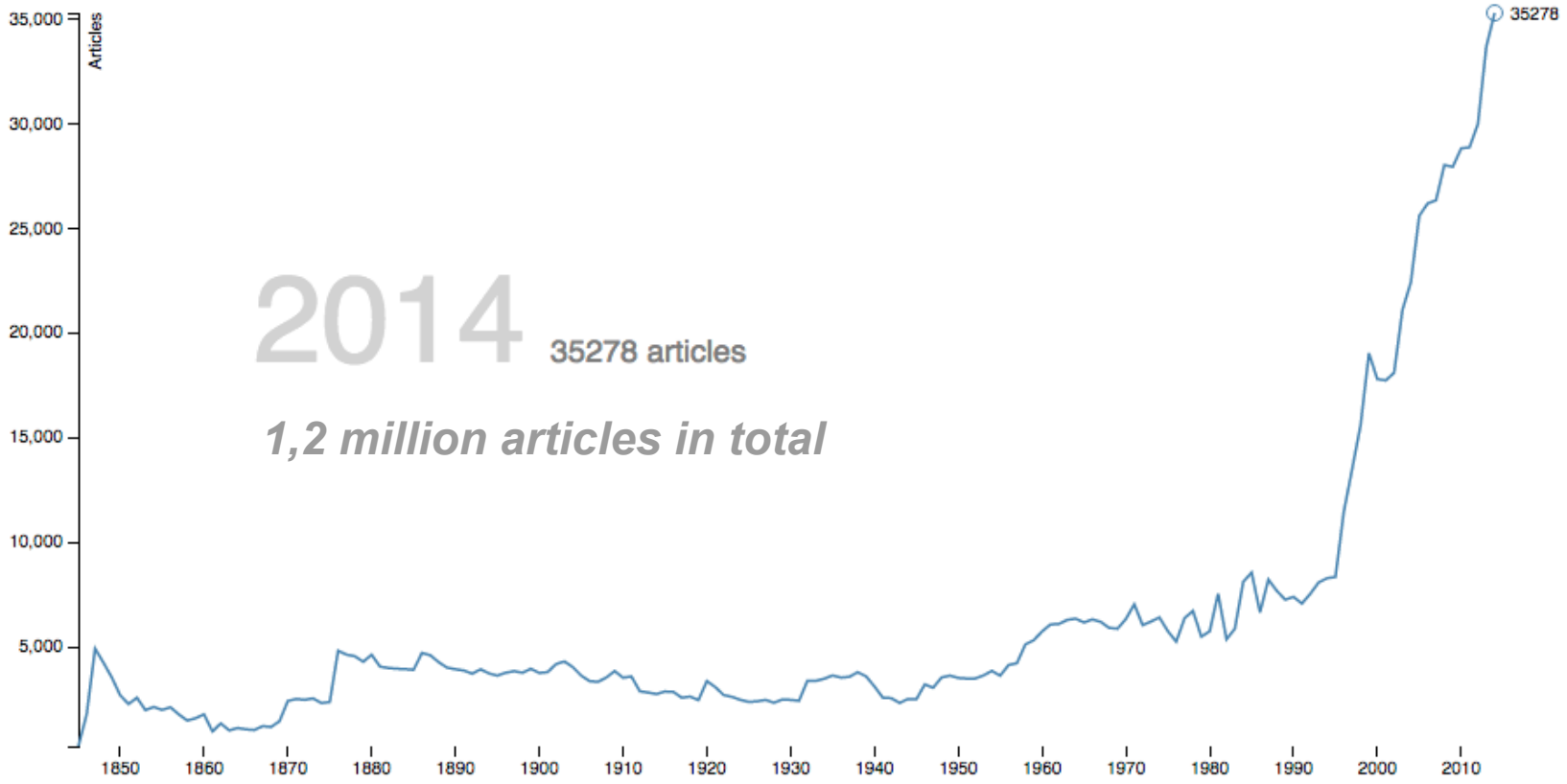
-  Michele Pasin is Information Architect, Product Office
@lambdaman

* We merged earlier this year (May 2015) with Springer Science+Business Media to become **Springer Nature**. We are currently actively engaged in integrating our businesses.

Macmillan: science and education brands



We publish a lot of science! (1845-2015)



http://www.nature.com/developers/hacks/articles/by-year

Why we're here today: to ask some questions

We have been making semantic data available in RDF models for a number of years through our **data.nature.com** portal (2012–2015)

Big questions:

- Is this data of any use to the Linked Science community?
- Should Springer Nature continue to invest in LOD sharing?

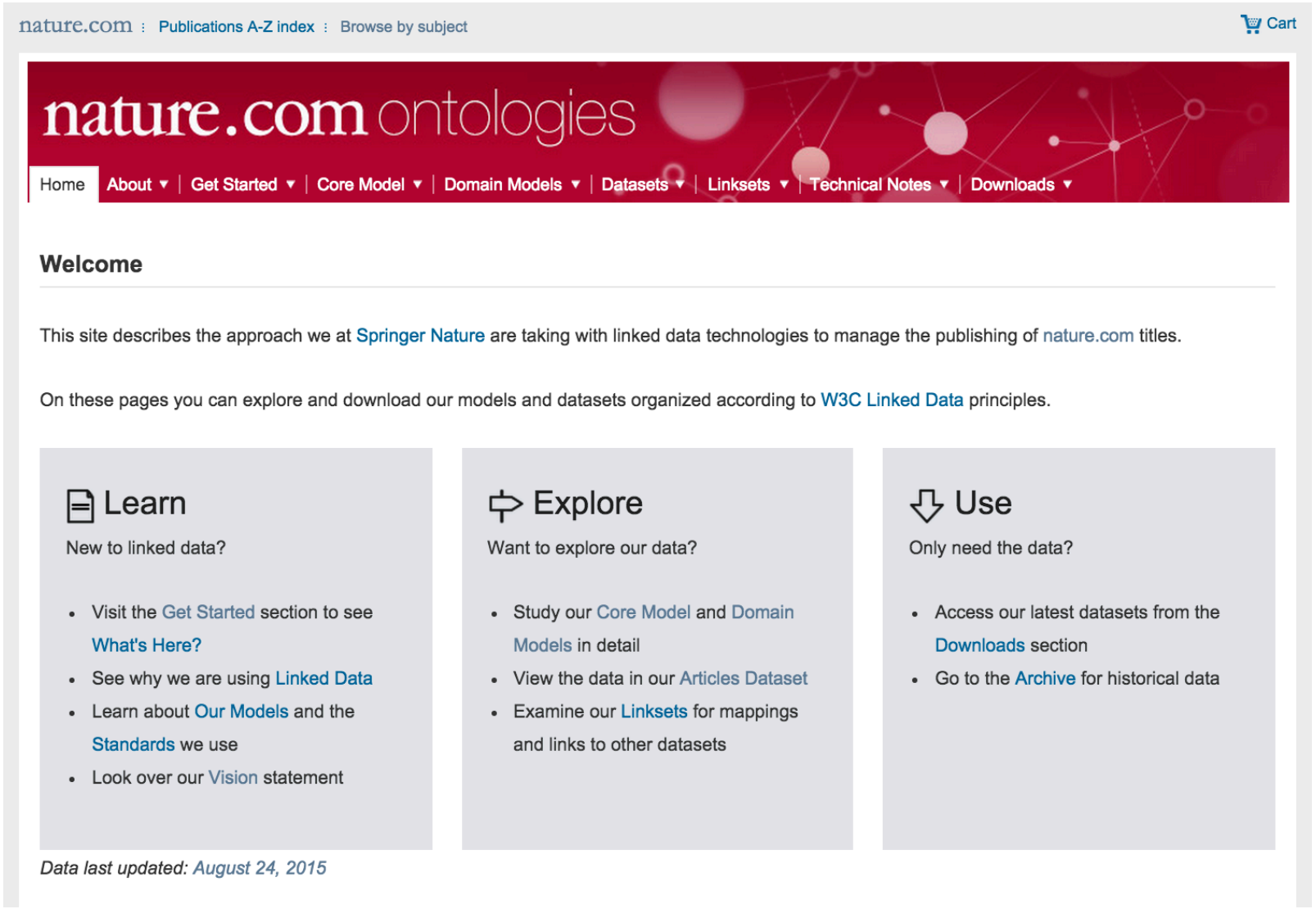
More specifically:

- Does the data contain enough items of interest? *[Content]*
- Are the vocabularies understandable and useful? *[Structure]*
- Are the data easy to get and to reuse? *[Accessibility]*
    - Is dereference / download / query the preferred option?

Our work so far

- Step 1: Linked Data Platform (2012–2014)
  - datasets
  - downloads + SPARQL endpoint
  - linked data dereference
- Step 2: Ontologies Portal (2015–)
  - datasets + models (core, domain)
  - downloads
  - extensive documentation

The Ontologies Portal



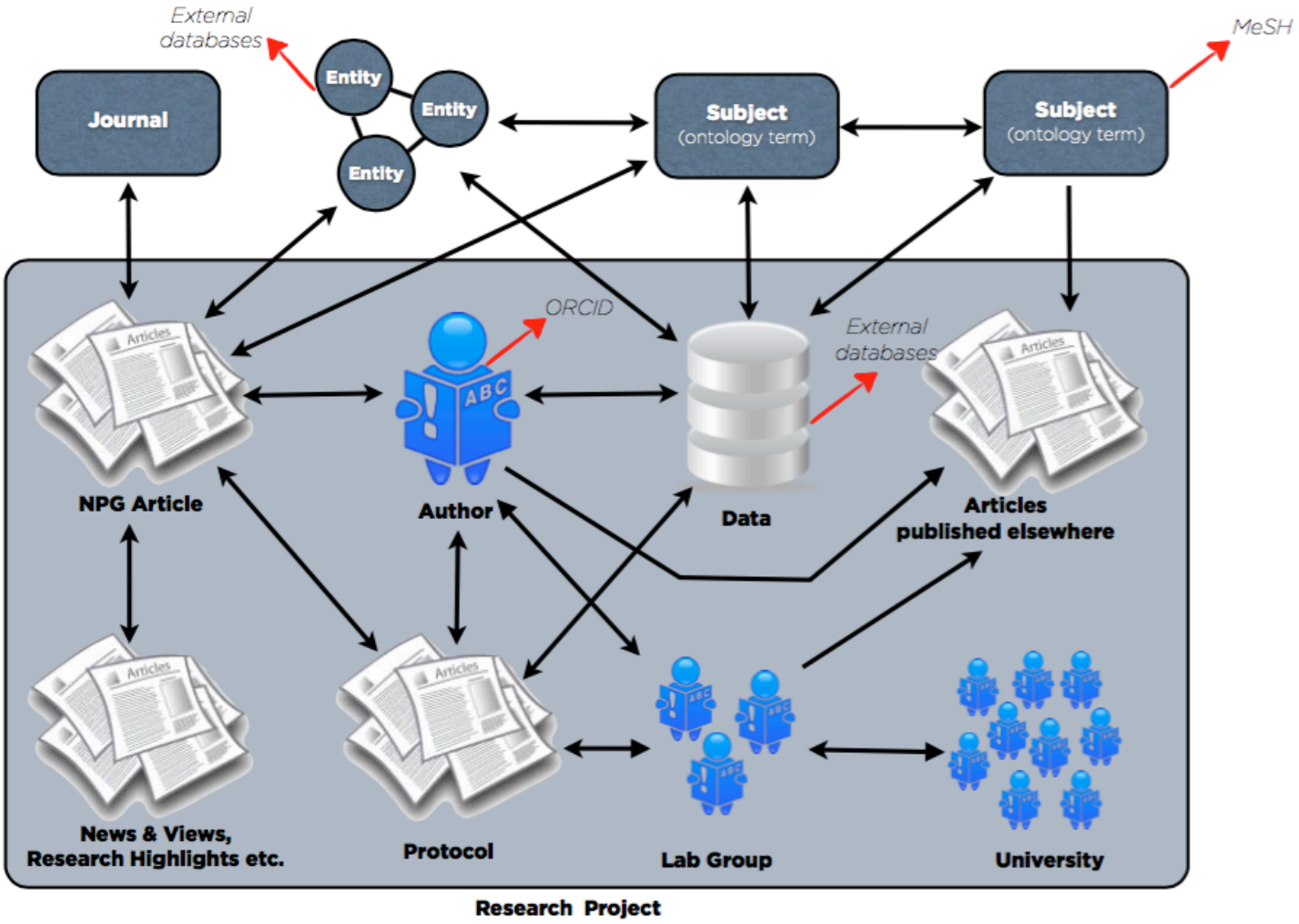
www.nature.com/ontologies

Our goals and rationale

- Semantic technologies are an effective way to do enterprise **metadata management** at web scale
  - Initially used primarily for data publishing / sharing (data.nature.com, 2011)
  - Since 2013, a core component of our digital publishing workflow (see ISWC14 paper)

- Contributing to an emerging **web of linked science** data
  - As a major publisher since 1845, ideally positioned to bootstrap a science ‘publications hub’
  - Building on the fundamental ties that exist between the actual research works and the publications that tell the story about it

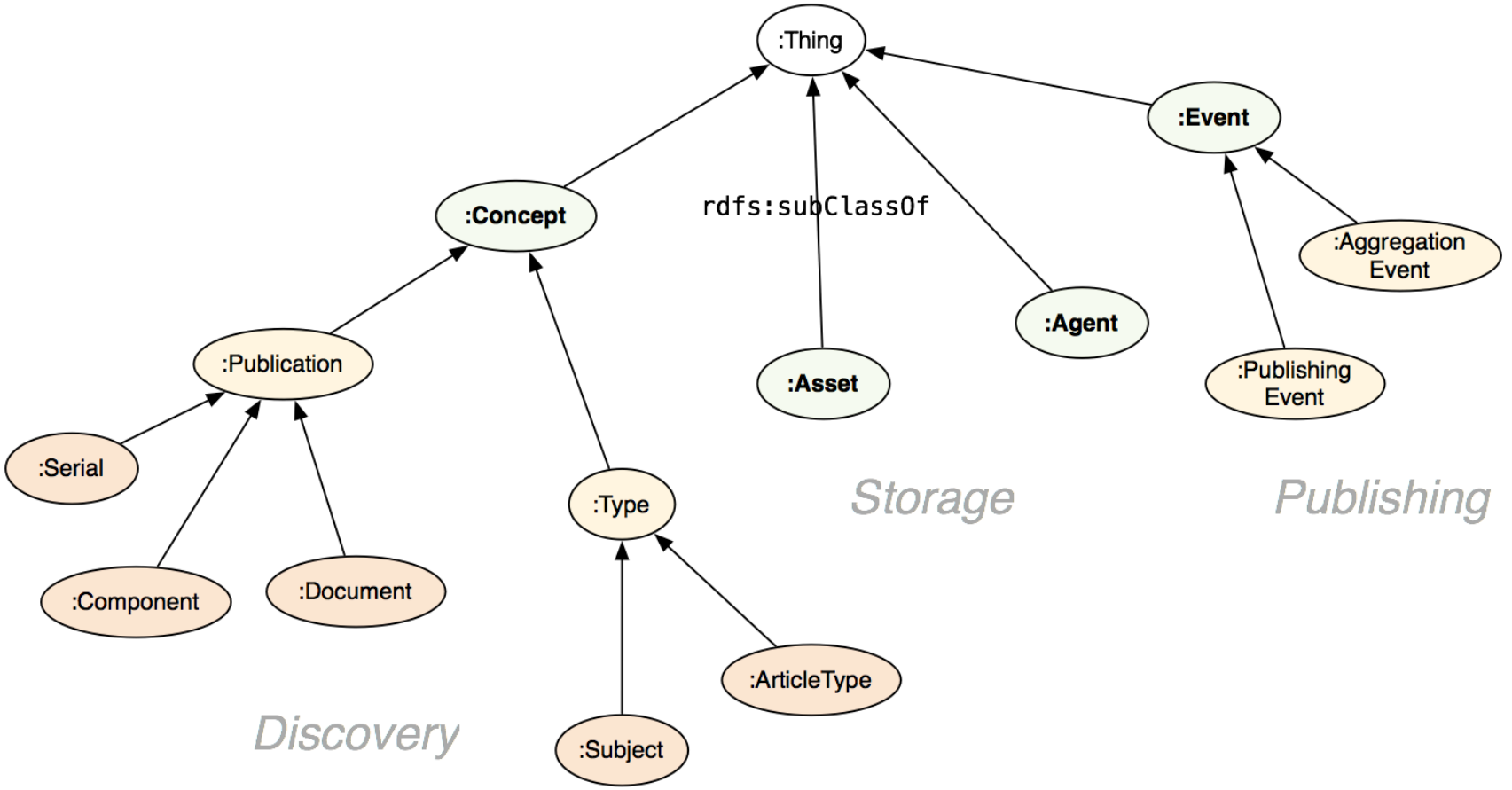
The vision of a science graph



What's available

Model	Description	Entities	Triples
Ontology			
Core	Formal model of key concepts	–	1,217
Taxonomies			
ArticleTypes	Genres for scholarly articles	64	977
Subjects	Subject areas for publications	2,636	48,500
Vocabularies			
Blogs	Master catalogue of blogs	32	270
Journals	Master catalogue of journals	236	3,831
PublishStates	Publish states for a publication	7	69
Relations	Relationships between publications	30	365
ReviewStates	Review states for a publication	5	53
SeverityLevels	Severity-levels used in build rules	8	93
SummaryTypes	Summary-types for summaries	4	45
Datasets			
Articles	Biblio data for scholarly articles	1,206,039	25,212,725
Contributors	Contributors to scholarly articles	2,698,052	11,052,560

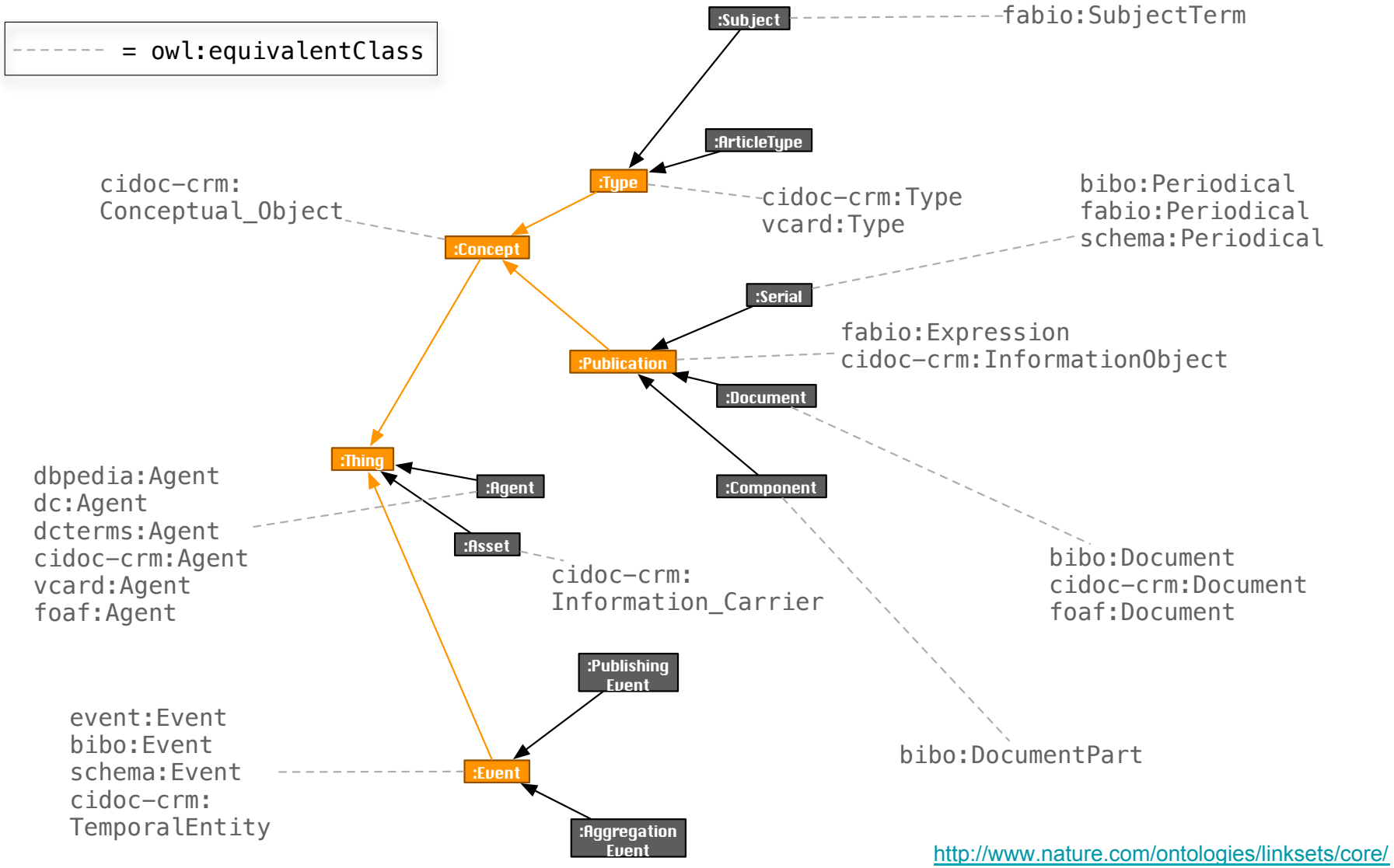
The core ontology



- **Language**: OWL 2, Profile: *ALCHI(D)*
- **Entities**: ~50 classes, ~140 properties
- **Principles**: Incremental Formalization/ Enterprise Integration / Model Coherence

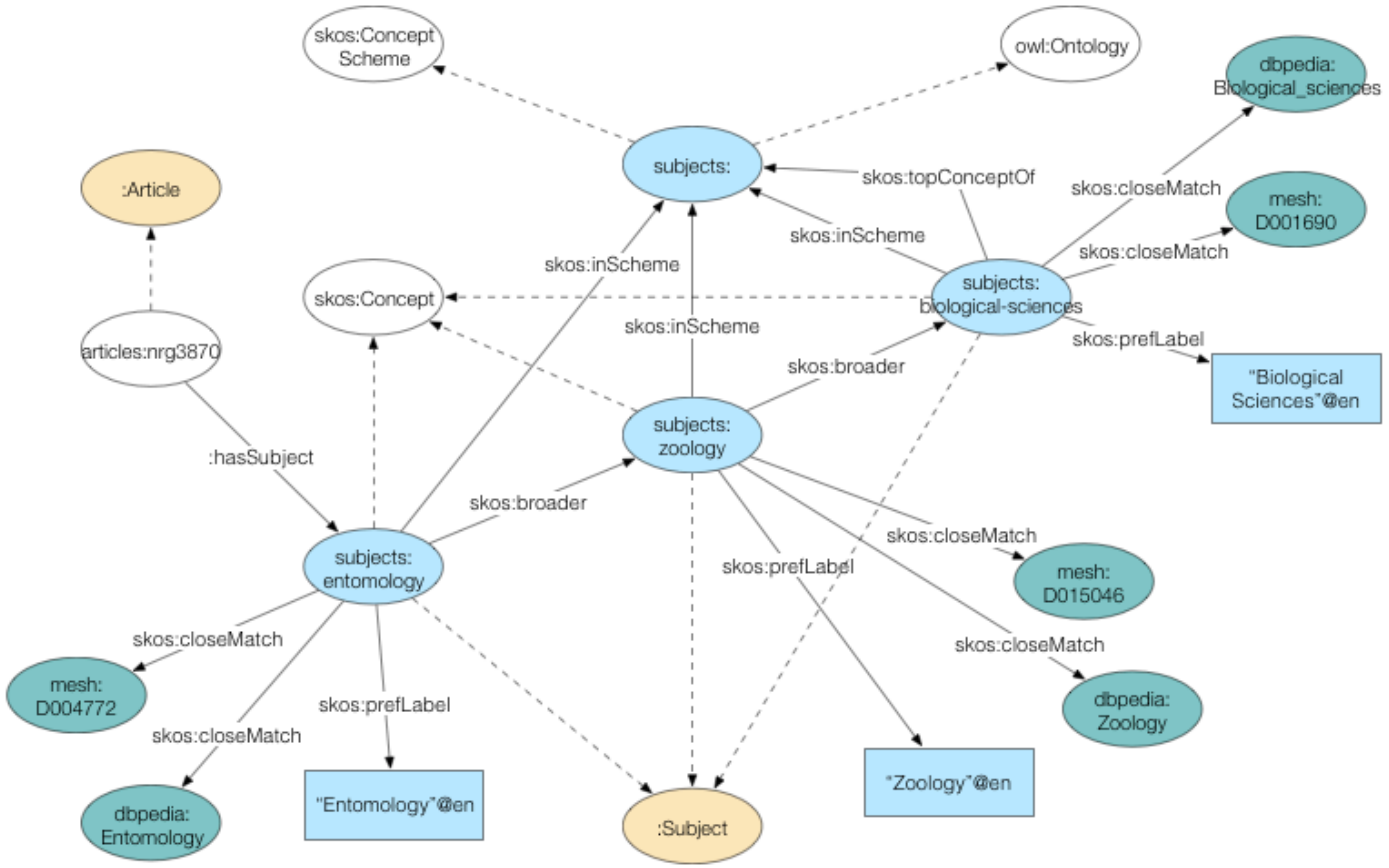
http://www.nature.com/ontologies/core/

The core ontology: mappings



http://www.nature.com/ontologies/linksets/core/

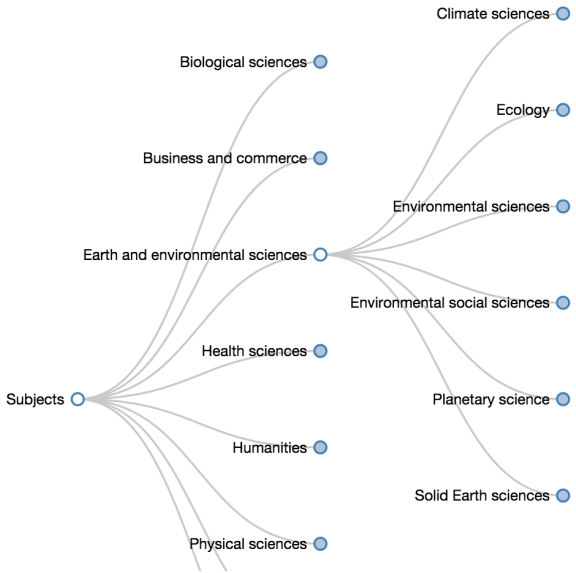
Domain models: subjects ontology



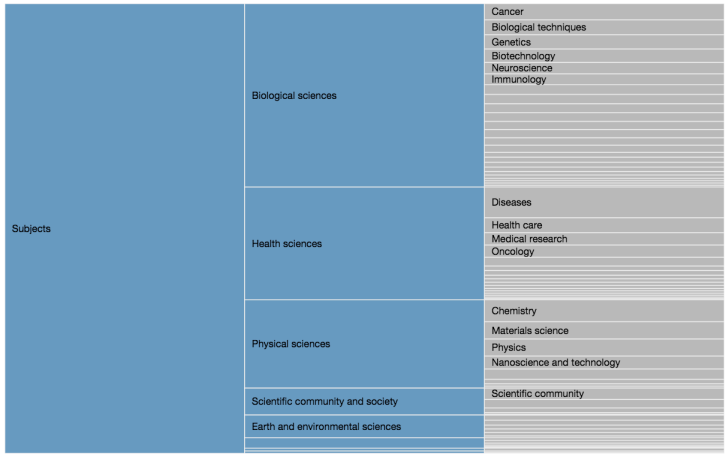
- **Structure**: SKOS, multi hierarchical tree, 6 branches, 7 levels of depth
- **Entities**: ~2500 concepts
- **Mappings**: 100% of terms, using *skos:broadMatch* or *skos:closeMatch,* (Dbpedia and MESH)

www.nature.com/ontologies/models/subjects/

Subjects visualizations



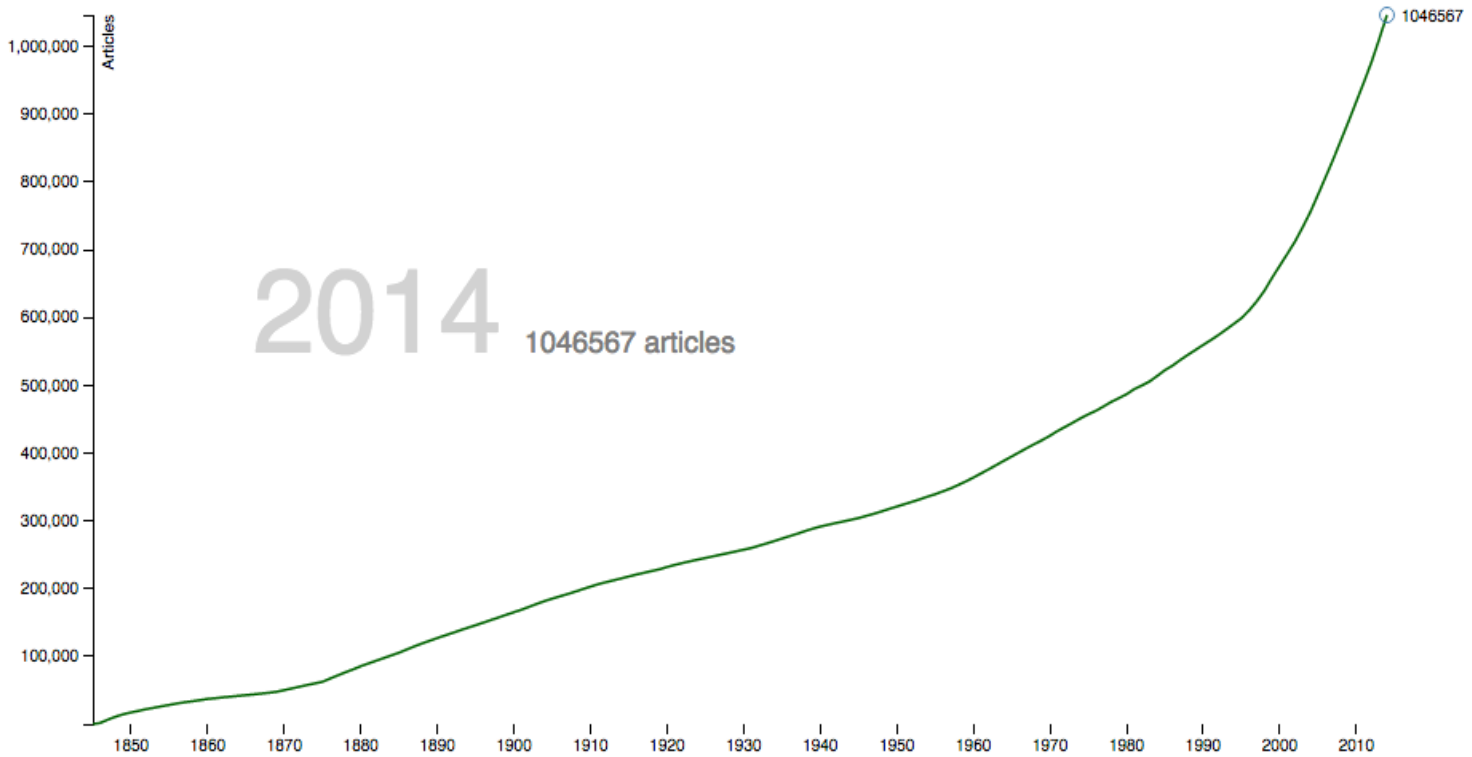






http://www.nature.com/developers/hacks/#1

Datasets



- **Articles**: 25m records (for 1.2m articles) with metadata like *title*, *publication* etc.. except authors
- **Contributors:** 11m records (for 2.7m contributors) i.e. the article's authors, structured and ordered but not disambiguated
- **Citations:** 218m records (for 9.3m citations) – from an earlier release

Datasets: articles-wikipedia links



238 Wikipedia mentions of NPG articles published in **1988** prev next download

npg article	wikipedia links	wikipedia page
10.1038/334287a0 : "High-Dilution" Experiments A Delusion **NATURE**, 1988	6	Nature (journal) | Water memory | John Maddox | Homeopathic dilutions | Jacques Benveniste | Homeopathy
10.1038/scientificamerican1188-118 : Evolution Of Human Walking **SCI AM**, 1988	6	Australopithecus | Australopithecus afarensis | Owen Lovejoy (anthropologist) | Bipedalism | Transitional fossil | Human skeletal changes due to bipedalism

Filter by journal:

Nature 205
Sci Am 16
J Invest Dermatol 6
Brit J Cancer 3

How: data extracted using wikipedia search API, 51,309 links over 145 years

Quality: only ~900 were links to nature.com without a DOI, rest all use DOIs correctly

Encoding: *cito:isCitedBy* => wiki URL, *foaf:topic* => dbPedia URI

http://www.nature.com/developers/hacks/wikilinks

Data publishing: sources

Sources:

Ontologies (small scale; RDF native)

- mastered as RDF data (Turtle)
- managed in GitHub
- in-memory RDF models built using Apache Jena
- models augmented at build time using SPIN rules
- deployed to MarkLogic as RDF/XML for query
- exported as RDF dataset (Turtle) and as CSV

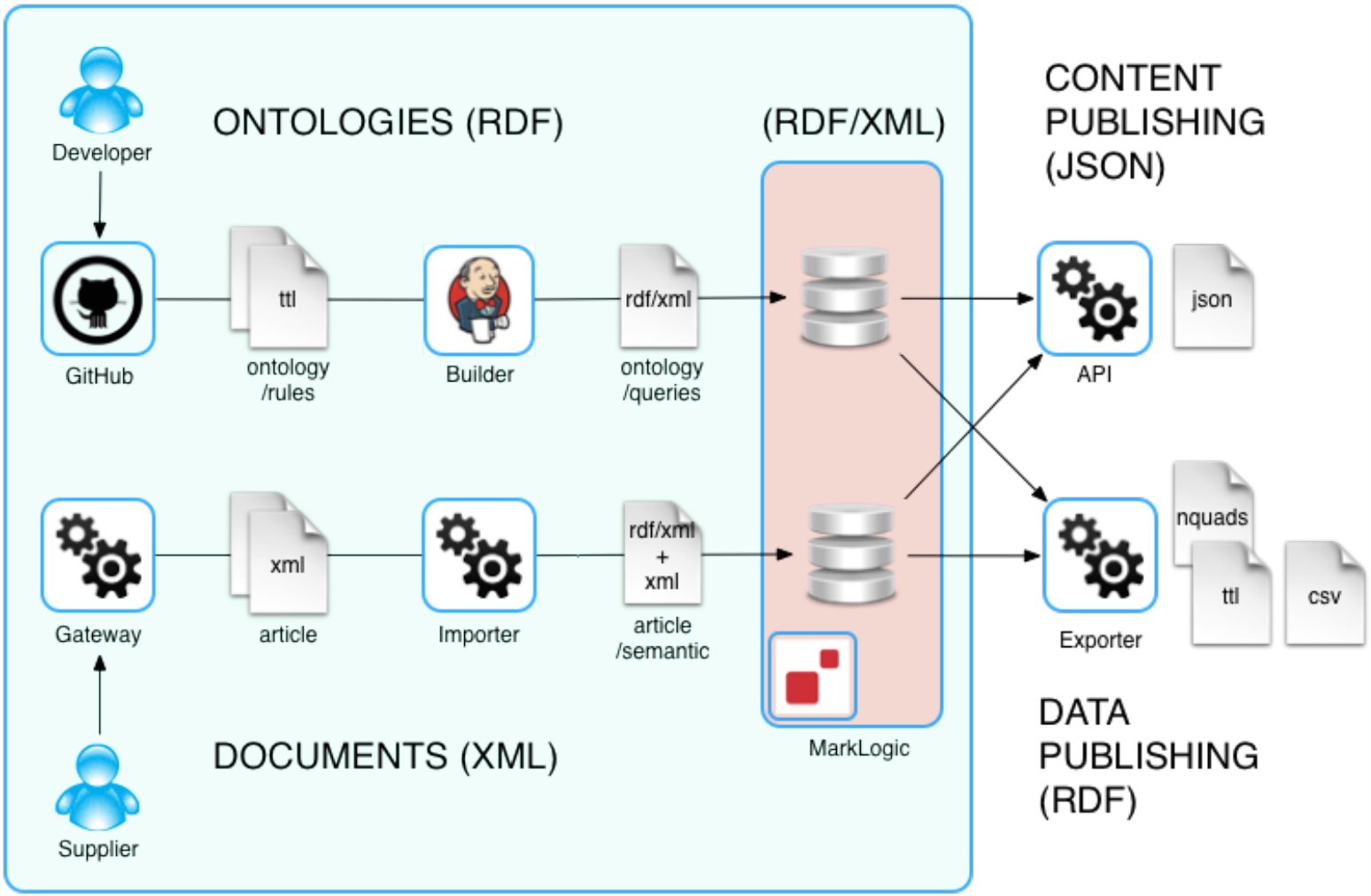
Documents (large scale; XML native)

- mastered as XML data
- managed in MarkLogic XML database
- data mined from XML documents (1.2m articles) using Scala
- in-memory RDF models built using Apache Jena
- injected as RDF/XML sections into XML documents for query
- exported as RDF dataset (N-Quads)

Organization:

Named graphs – one graph per class

Data publishing: workflows



Data publishing: rules (enrichment)

```
construct {
  ?s npg:publicationStartYear ?xds1 .
  ?s npg:publicationStartYearMonth ?xds2 .
  ?s npg:publicationStartDate ?xds3 .
  ?s npg:publicationEndYear ?xde1 .
  ?s npg:publicationEndYearMonth ?xde2 .
  ?s npg:publicationEndDate ?xde3 .
}
where {
  ?s a npg:Journal .
  optional { ?s npg:dateStart ?dateStart }  optional { ?s npg:dateEnd ?dateEnd }
  {
    bind (if(regex(?dateStart, "^\\d{4}"), substr(?dateStart,1,4), "") as ?ds1)
    bind (xsd:gYear(?ds1) as ?xds1)
  } union {
    bind (if(regex(?dateStart, "^\\d{4}-\\d{2}"), substr(?dateStart,1,7), "") as ?ds2)
    bind (xsd:gYearMonth(?ds2) as ?xds2)
  } union {
    bind (if(regex(?dateStart, "^\\d{4}-\\d{2}-\\d{2}$"), substr(?dateStart,1,10), "") as ?ds3)
    bind (xsd:date(?ds3) as ?xds3)
  } union {
  …
  }
  filter (?xds1 != "" || ?xds2 != "" || ?xds3 != "" || ?xde1 != "" || ?xde2 != "" || ?xde3 != "")
}
```

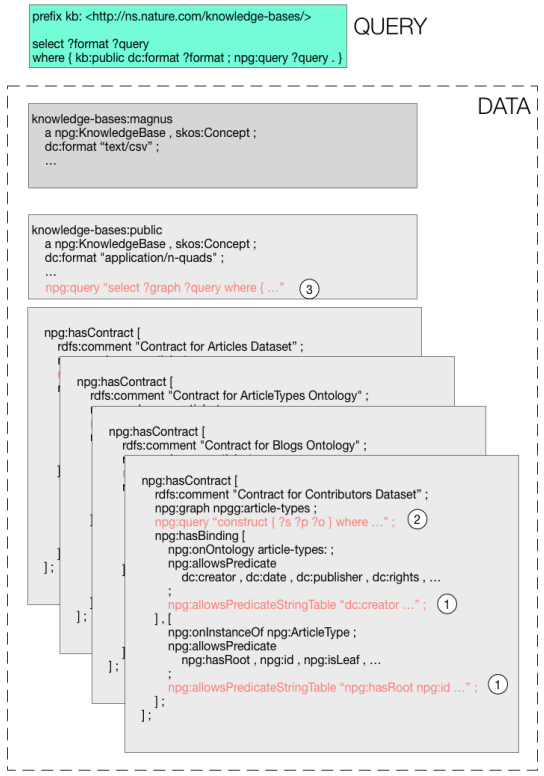
Data publishing: rules (validation)

```
construct {
  npgg:journals npg:hasConstraintViolation [
    a spin:ConstraintViolation ;
    npg:severityLevel "Warning" ;
    rdfs:label ?message ;
    spin:rule [ a sp:Construct ; sp:text ?query ; ] ;
  ] .
}
where {
  { select (count(?s) as ?count)
    where {
      ?s a npg:Journal .
      filter ( not exists { ?s bibo:shortTitle ?h . } ) }
  }
  bind (concat("! Found ", str(?count), " journals with no short title") as ?message)
  bind ("""
construct {
  npgg:journals npg:hasConstraintViolation [
    a spin:ConstraintViolation ;
    spin:violationRoot ?s ; … ] .
} where { … }
""" as ?query)
}
```

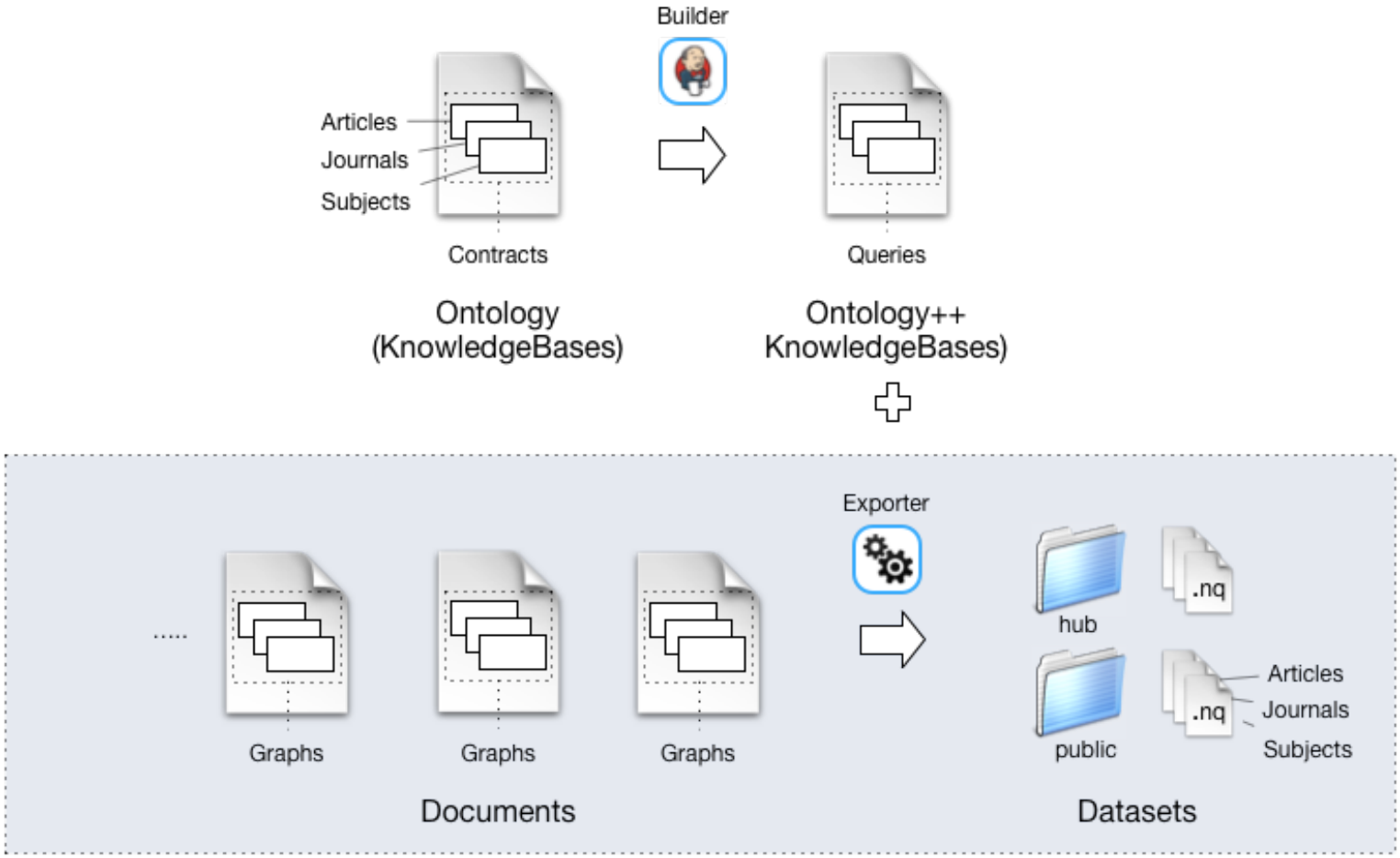
Data publishing: rules (contracts)

	class	query
1	npg:Article	construct { ?s ?p ?o } where { ?s a <http://ns.nature.com/terms/Article> . ?s ?p ?o . filter (str(?p) in ('http://ns.nature.com/terms/hasContributo ns#type','http://purl.org/ontology/bibo/pageStart','http://purl.org/ontology/bibo/volume')) filter (?o != skos:Concept) minus { ?s npg:isTestInsta
2	npg:ArticleType	construct { ?s ?p ?o } where { ?s a <http://ns.nature.com/terms/ArticleType> . ?s ?p ?o . filter (str(?p) in ('http://www.w3.org/2004/02/skos/core#topConceptOf','http://ns.nature.com/terms/id','http://ns.nature.com/terms/isRoot','http://www.w3.org/200 rdf-syntax-ns#type','http://www.w3.org/2004/02/skos/core#altLabel','http://www.w3.org/2004/02/skos/core#definition')) filter (?o != skos:Conc
3	npg:Journal	construct { ?s ?p ?o } where { ?s a <http://ns.nature.com/terms/Journal> . ?s ?p ?o . filter (str(?p) in ('http://www.w3.org/2004/02/skos/core#p minus { ?s npg:isPublishReady false } }
4	npg:Subject	construct { ?s ?p ?o } where { ?s a <http://ns.nature.com/terms/Subject> . ?s ?p ?o . filter (str(?p) in ('http://www.w3.org/2004/02/skos/core#topConceptOf','http://ns.nature.com/terms/id','http://ns.nature.com/terms/isRoot','http://www.w3.org/200 rdf-syntax-ns#type','http://www.w3.org/2004/02/skos/core#altLabel','http://www.w3.org/2004/02/skos/core#definition')) filter (?o != skos:Conc

```
knowledge-bases:public
   ...
   npg:hasContract [
      rdfs:comment "Contract for ArticleTypes Ontology" ;
      npg:graph npgg:article-types ;
      npg:hasBinding [
         npg:onOntology article-types: ;
         npg:allowsPredicate
            dc:creator , dc:date , dc:publisher , dc:rights , dcterms:license ,
            npg:webpage , owl:imports , owl:versionInfo , rdf:type , rdfs:comment ,
            skos:definition , skos:prefLabel , skos:note ,
            vann:preferredNamespacePrefix , vann:preferredNamespaceUri
         ;
      ] , [
         npg:onInstanceOf npg:ArticleType ;
         npg:allowsPredicate
            npg:hasRoot , npg:isPrimaryArticleType ,
            npg:id , npg:isLeaf , npg:isRoot , npg:treeDepth ,
            rdf:type , rdfs:isDefinedBy , rdfs:seeAlso ,
            skos:broadMatch , skos:broader , skos:closeMatch ,
            skos:definition , skos:exactMatch , skos:inScheme , skos:narrower ,
            skos:prefLabel , skos:relatedMatch , skos:topConceptOf
         ;
      ] ;
   ] ;
   ...
```



Data publishing: rules (contracts)



Next steps

More features:
- Linked data dereference
- Richer dataset descriptions (VoID, PROV, HCLS Profile, etc.)
- SPARQL endpoint?
- JSON-LD API?

More data:
- Adding extra data points (funding info, affiliations, …)
- Revamp citations dataset
- Longer term: extending archive to include Springer content

More feedback:
- User testing around data accessibility
- Surveying communities/users for this data

Looking ahead: how can a publisher make linked science happen?

From a business perspective:

- Finding adequate licensing solutions
- Justifying the effort to publishers
- What's the ROI?

From a communities perspective:

- Do we actually know who are the users?
- How do we get more feedback/uptake?
- Should we work more with non-linked-data communities?

Questions?