Student Online Teaching Advice Notice

The materials and content presented within this session are intended solely for use in a context of teaching and learning at Trinity.

Any session recorded for subsequent review is made available solely for the purpose of enhancing student learning.

Students should not edit or modify the recording in any way, nor disseminate it for use outside of a context of teaching and learning at Trinity.

Please be mindful of your physical environment and conscious of what may be captured by the device camera and microphone during videoconferencing calls.

Recorded materials will be handled in compliance with Trinity's statutory duties under the Universities Act, 1997 and in accordance with the University's policies and procedures.

Further information on data protection and best practice when using videoconferencing software is available at https://www.tcd.ie/info_compliance/data-protection/.

University of Dublin
Trinity College

CSU22041: Information Management I
Uniform Resource Identifier
Resource Definition Framework

2020-2021

Gaye Stephens gaye.stephens@tcd.ie

Web of Documents vs. Web of Data

	Web of Documents	Web of Data
Analogy	Global file system	Global database
Primary objects	Documents	(Descriptions of) Things
Links between	(Parts of) Documents	Things
Degree of structure	Low	High
Semantics between links and content	Implicit	Explicit
Designed for	Human consumption	Both human and computer-based agents

Compiled from http://www.w3.org/2008/Talks/WWW2008-W3CTrack-LOD.pdf

Representing Information: Data and Relationships

So far, for representation of data and their relationships, you have already covered :

- As *objects*, using the well-accepted techniques of object-oriented analysis and design to capture a model
- As *UML*, but not easily machine processable
- As *a tree-like* representation (e.g. Binary, AVL, XML etc.)

For the semantic web, we are looking for infinitely extensible, mergeable and scalable solutions. We need the above plus something more….

Representing Information: Data and Relationships

What more do we need….

We need to be able to describe what we **mean** by the data we represent in our information system in a way that another system can understand it.

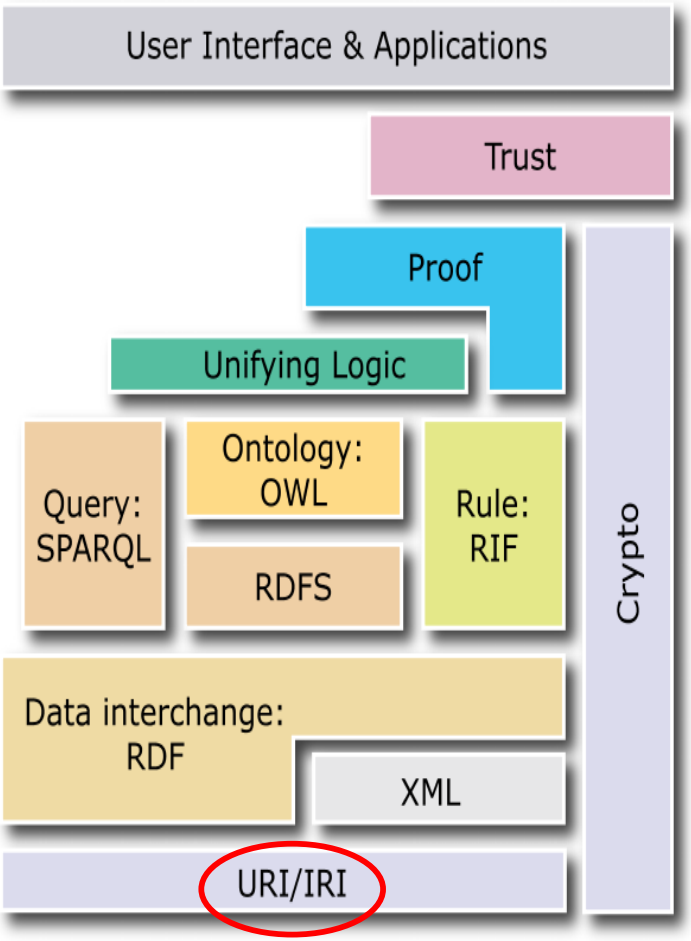
In addition to the meaning we also need a way to describe how the different parts of the data **relate** to each other.

We also need a means to identify the **location of the data.**

The Semantic Web Stack

Traditional Web
URI/IRI
XML
Cryptography

Semantic Web
Resource Description Framework (RDF)
RDF Schema
Web Ontology Language
SPARQL
Rules: RIF (and SWRL)



(from https://www.w3.org/)

https://www.w3.org/TR/cooluris/

University of Dublin
Trinity College

Uniform Resource Identifier (URI)

URIs

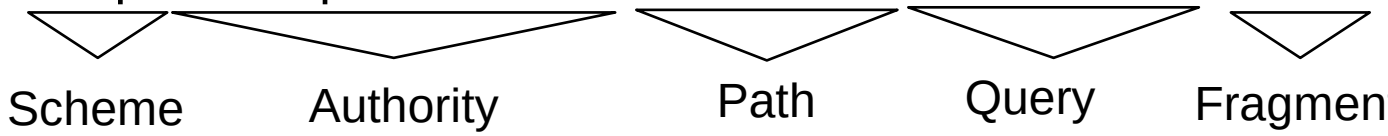
A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource [RFC3986]

Syntax:

URI = scheme ":" hier-part ["?" query] ["#" fragment]

Example

http://example.com:8042/over/there?name=ferret#nose



Note: scheme not the same as protocol

From: http://www.slideshare.net/mediasemanticweb/linked-data-michael-hausenblas-2009-03-05

URI look-ups

When entities are identified by URIs that use the `http://` scheme, these entities can be looked up simply by dereferencing the URI over the HTTP protocol.

Information is provided that is suitable for the “consumer”

- Suitable for browsers vs. suitable for machines
- Humans rather see HTML pages, PDFs, pictures,
- Machines want machine-readable formats such as RDF

Simple, standardized mechanism for retrieving **resources** via these URIs.

(Non-)Information Resources

Information resources(IR) are documents – referred to by a URI – that describe

non-information resources(NIR) – named with a URI – that represent things such as cars, people, etc.

Either a NIR or an IR is returned depending on what you need. How?

URI pointing to a IR returns the representation.

URI pointing to a NIR returns a redirect to an IR describing that NIR.

Information Resources and Non Information Resources

“The vast majority of **resources** on the Web are **information resources**. Representations of these resources are made available through the appropriate interaction with their Web presence. However, increasingly there is interest in being able to use URIs to identify uniquely resources whose essence is not information. We term such resources **non-information** resources.

When a resource is not an information resource, it is important that it does not behave on the Web as if it is an information resource. To do so would be misleading. For example, it is important that non-information resources do *not* respond with representations if they have a Web presence. “

https://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14.html

Resource Description Framework (RDF)

The Semantic Web is envisioned as a decentralised world-wide information space for sharing machine-readable data with a minimum of integration costs. Its two core challenges are the distributed modelling of the world with a shared data model, and the infrastructure where data and schemas can be published, found and used. Users benefit from getting information *"raw and now"* [Give] and in portable data formats [DP]. Providers often publish data embedded in a fixed user interface, in HTML. A basic question is thus how to publish information about resources in a way that allows interested users and software applications to find and interpret them.

On the Semantic Web, all information has to be expressed as *statements* about *resources*, like *the members of the company Example.com are Alice and Bob* or *Bob's telephone number is "+1 555 262"* or *this Web page was created by Alice*. Resources are identified by *Uniform Resource Identifiers* (URIs) [*RFC3986*]. This modelling approach is at the heart of *Resource Description Framework* (RDF) [*RDFPrimer*]. A nice introduction is given in the N3 primer [N3Primer].

Using RDF, the statements can be published on the Web site of the company. Others can read the data and publish their own information, linking to existing resources. This forms a distributed model of the world. It allows the user to pick any application to view and work with the same data, for example to see Alice's published address in your address book.

At the same time, Web documents have always been addressed with URIs (in common parlance often referred as Uniform Resource Locators, URLs). This is useful because it means we can easily make RDF statements about Web pages, but also dangerous because we can easily mix up Web pages and the things, or resources, described on the page.

So the question is, what URIs should we use in RDF? As an example, to identify the frontpage of the Web site of Example Inc., we may use http://www.example.com/ But what URI identifies the company as an organisation, not a Web site? Do we have to serve any content—HTML pages, RDF files—at those URIs? In this document we will answer these questions according to relevant specifications. We explain how to use URIs for things that are not Web pages, such as people, products, places, ideas and concepts such as ontology classes. We give detailed examples as to how the Semantic Web can (and should) be realised as a part of the Web.

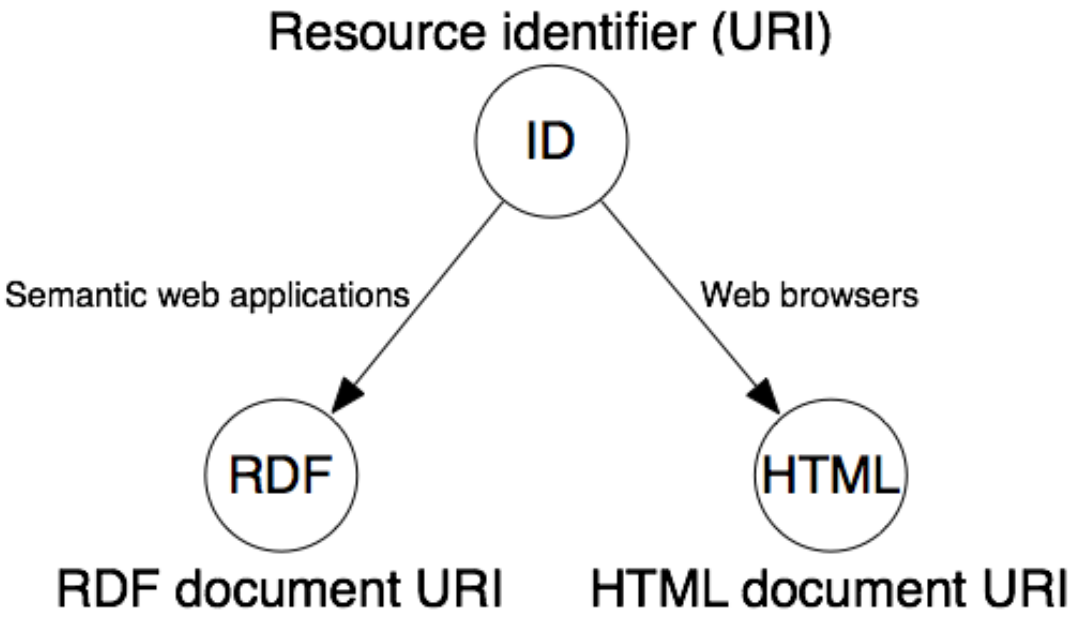
https://www.w3.org/TR/cooluris/

Process of rendering Information Resources

“Information resources make up the vast majority of the Web today. Their behavior is well understood. In particular, information resources have representations which are, in some sense, 'obvious'. The essence of an information resource is information. Consequently, the act of creating a representation is simply a transformation of that information into an appropriate form. Often that transformation will include formatting that allows the rendered representation to be used conveniently by a Web user.

As an example, let's consider the creation of a statement of activity for a particular month for a particular bank account. We'll suppose that a URI identifies the resource which, in this case, is a particular set of of binary data held in a relational database. To create a representation of the resource, the appropriate data is first extracted from the database and converted to textual form. Then it is embedded in a stream of HTML markup that also references appropriate styling information. This representation flows across the Web to a browser, where it is rendered. A user is able to perceive the rendered form and to understand the activity on the account for month in question.
The process of creating and rendering representations from information resources is so common that it is often either overlooked or considered to be completely ubiquitous. However, not all Web resources are necessarily associated with obvious representations.”

https://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14



https://www.w3.org/TR/cooluris/#oldweb

Data and Meta data returned

“The representation returned when a Web presence is accessed consists of two parts. **One part is the data, the other part is metadata.**
For a Web page, for example, the data is the part that contains the markup and associated instructions from which the rendered version is created.
The metadata includes the information within the HTTP headers that form part of the response. **This metadata includes the HTTP response code.** One particular value of this response code forms the mechanism by which HTTP itself can indicate associations.

HTTP response code 303, named 'See Other', indicates that there is no representation for the accessed URI, but that associated information may be available. Importantly, this response also provides a URI which may be accessed in pursuit of the associated information. **This type of redirection of the request to a different URI is exactly the behavior that we need in order to indicate that there may be additional information associated with a non-information resource.**”

https://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14.html

Table 1: Summary of inferences that can be made when dereferencing an HTTP URI

Code	Meaning	Material Returned	Inference
200	OK	A representation	The resource is an information resource and a representation of it has been returned.
301	Moved Permanently	A URI	The URI specified in the request and the URI returned in the response are aliases and refer to the same resource. The resource might be an information resource or a non-information resource.
302	Found	A URI	The URI specified in the request and the URI returned in the response are aliases and refer to the same resource. The resource might be an information resource or a non-information resource.
303	See Other	A URI	The resource is a non-information resource. There is an associated resource whose URI has been returned. The associated resource might or might not be an information resource.
406	Not Acceptable	Nothing	The resource is an information resource, but no representation could be returned.
Other 4XX	Error	Nothing	Nothing can be inferred about the nature of the resource.
5XX	Error	Nothing	Nothing can be inferred about the nature of the resource.

https://www.w3.org/2001/tag/doc/httpRange-14/2007-08-31/HttpRange-14.html

2 URI processing Styles for NIR

303 Redirect (e.g. http://example.uk/people/dave-smith)

- Used for large, dynamic data sets
- Flexible because redirection can be separately configured for each resource eg can store data in multiple files or DB. Can change this at deployment/run-time.
- Typically used for resource descriptions in large data-sets

#URIs (e.g. http://example.uk/people#dave-smith)

- Used for small, static data sets
- Reduced number of HTTP round-trips => reduced latency
- A single HTTP request retrieves the entire document
- May transmit unnecessary data across the web
- Typically used for vocabulary definitions

Resource Description Framework (RDF)

“The *Resource Description Framework* RDF allows users to describe both Web documents and concepts from the real world—people, organisations, topics, things—in a computer-processable way. **Publishing such descriptions on the Web creates the *Semantic Web*.**

URIs (Uniform Resource Identifiers) are very important, providing both the core of the framework itself and the link between RDF and the Web.

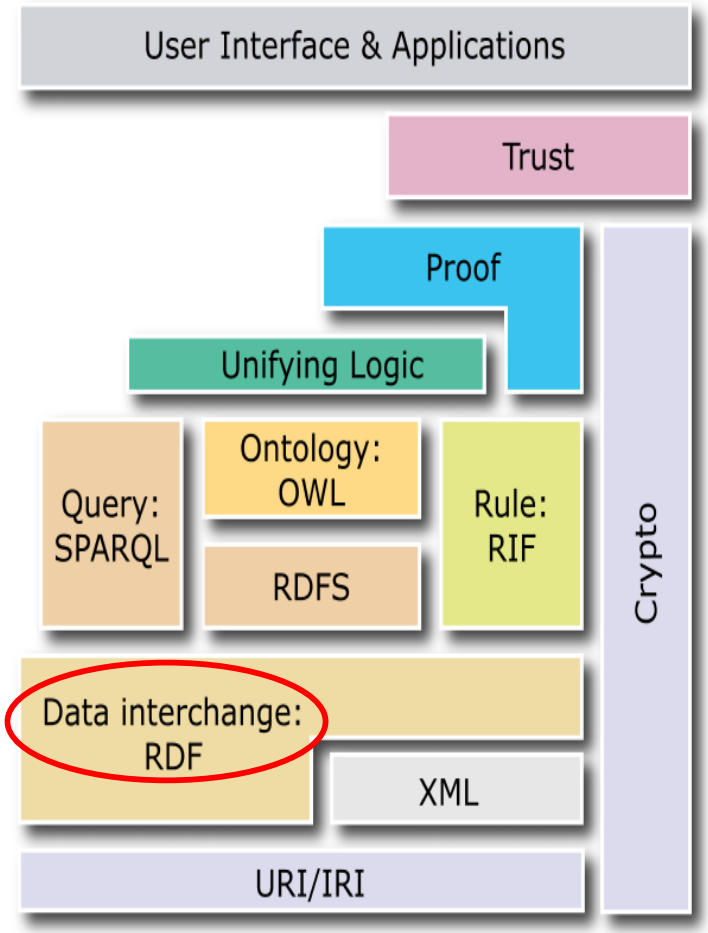
This document presents guidelines for their effective use. It discusses two strategies, called *303 URIs* and *hash URIs*. It gives pointers to several Web sites that use these solutions, and briefly discusses why several other proposals have problems.”

https://www.w3.org/TR/cooluris/

The Semantic Web Stack

Traditional Web
URI/IRI
XML
Cryptography

Semantic Web
Resource Description Framework (RDF)
RDF Schema
Web Ontology Language
SPARQL
Rules: RIF (and SWRL)



(from https://www.w3.org/)

https://www.w3.org/TR/cooluris/

Representing Information: Data and Relationships

So far, for representation of data and their relationships, you have already covered :

- As *objects*, using the well-accepted techniques of object-oriented analysis and design to capture a model
- As *UML*, but not easily machine processable
- As *a tree-like* representation (e.g. Binary, AVL, XML etc.)

For the semantic web, we are looking for infinitely extensible, mergeable and scalable solutions. We need the above plus something more….

Information graphs…. Expressed in RDF format

Basic structure – triples

RDF represent information using a triple structure

- Subject
- Predicate
- Object

Remember, a triple structure is one way of viewing a graph, so RDF essentially defines an information graph

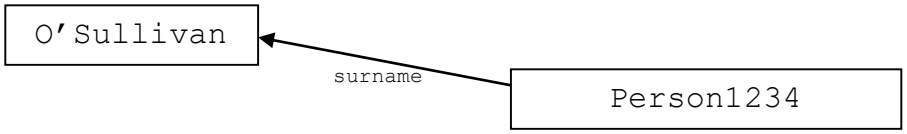
Information is built up as a collection of these triples, contained within an XML file

An RDF triple is conventionally written in the order subject, predicate, object.

Two ways to view a graph

As nodes and arcs

- Nodes store facts, arcs store relationships between them

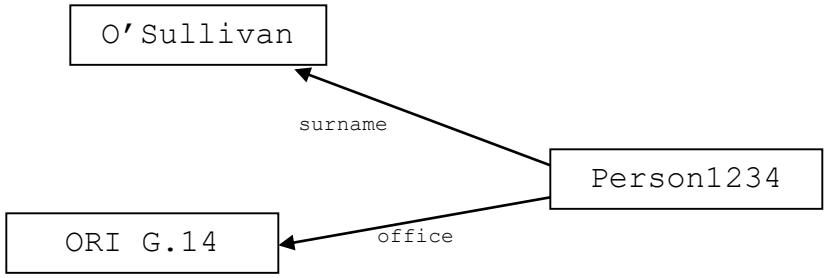


As triples

- A three-place relationship of 'subject, predicate, object'

```
Person1234 surname O'Sullivan
```

Graphs

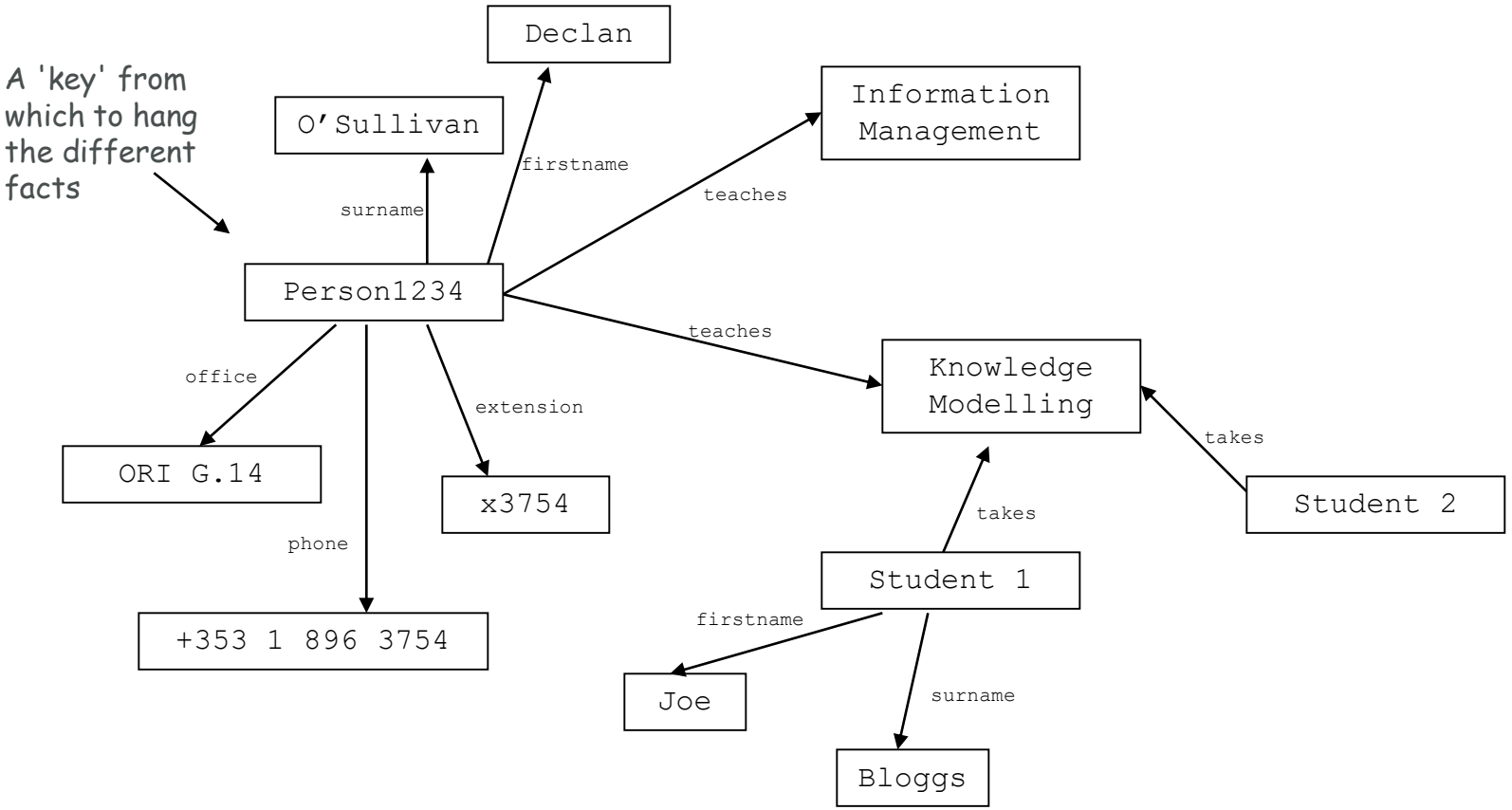


We can use the nodes of a graph for facts and the arcs as (binary) directed relationships between them

- Arcs are typically called predicates or relationships in this view
- The set of arcs intersecting a node tells us the information we know about that node

Graphs of information– 1

How do we use graphs to represent information?



Namespace

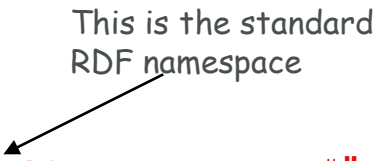
RDF uses XML namespaces to identify predicates

- Define a namespace representing a family of predicates we want to use, for example a model of location
- Define a set of elements and attributes within the namespace to model the phenomenon we're interested in
- Use the fully-qualified name of the elements to specify the predicate we want uniquely

Basic RDF structure

RDF data can live inside other XML documents

Identified by a top-level element from the well-known RDF namespace




```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  …
</rdf:RDF>
```

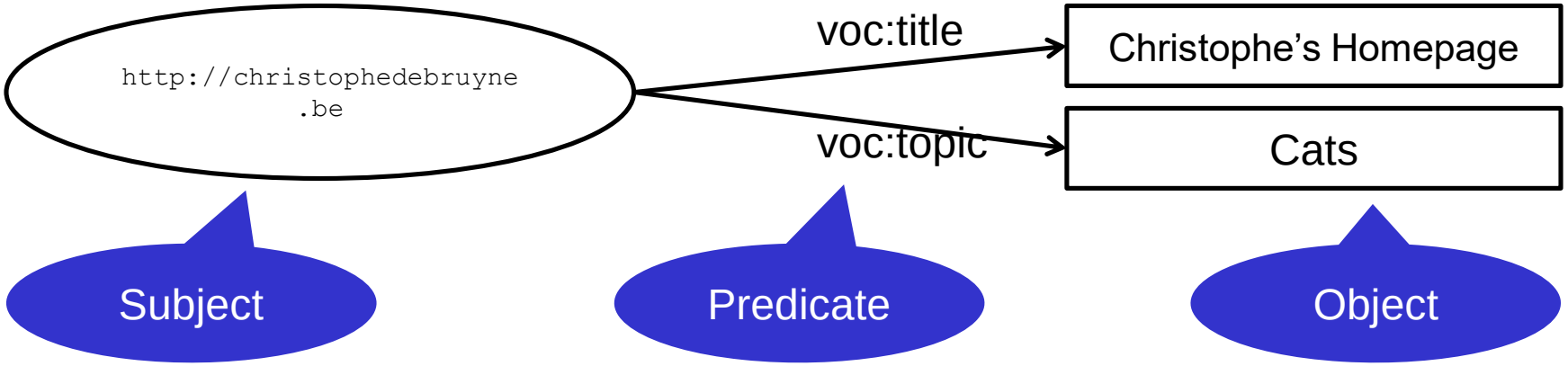
Within the `RDF` element we can define triples

- Co-exists well with the rest of XML
- Readable enough for people to build by hand

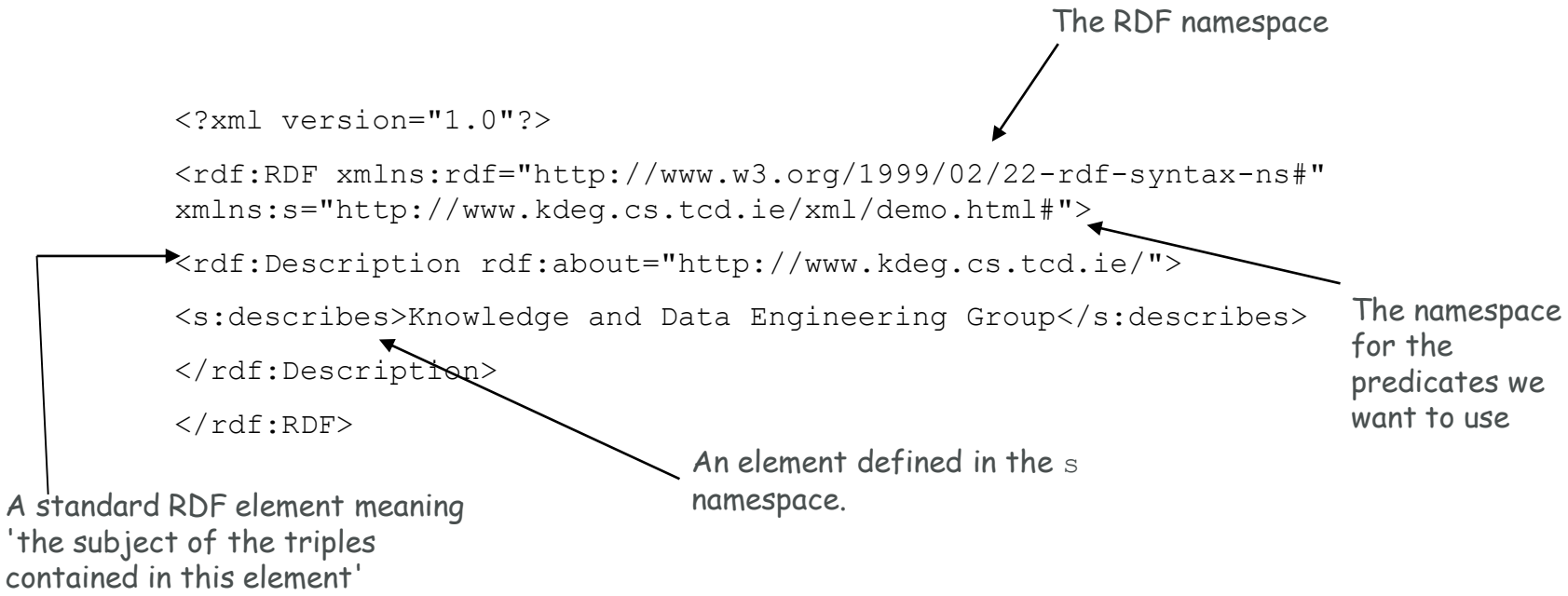
Simple Example and visualisation

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:voc="http://example.org/voc#">

  <rdf:Description rdf:about="http://christophedebruyne.be">
    <voc:title>Christophe's Homepage</voc:title>
    <voc:topic>Cats</voc:topic>
  </rdf:Description>

</rdf:RDF>
```

RDF Namespace



Another example



See how the predicate gets created from the namespace?

Subject: http://www.kdeg.cs.tcd.ie

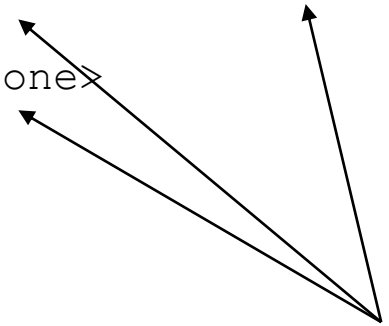
Predicate: http://www.kdeg.cs.tcd.ie/xml/demo.html#describes

Object: Knowledge and Data Engineering Group

A cluster of facts

Given a common subject we can build a cluster of facts using nested predicate elements

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:s="http://www.kdeg.cs.tcd.ie/xml/demo.html#">
  <rdf:Description rdf:about="http://www.kdeg.cs.tcd.ie/">
    <s:describes>Knowledge and Data Engineering Group</s:describes>
    <s:author>S. Punter</s:author>
    <s:phone>+353 1 123 4567</s:phone>
  </rdf:Description>
</rdf:RDF>
```



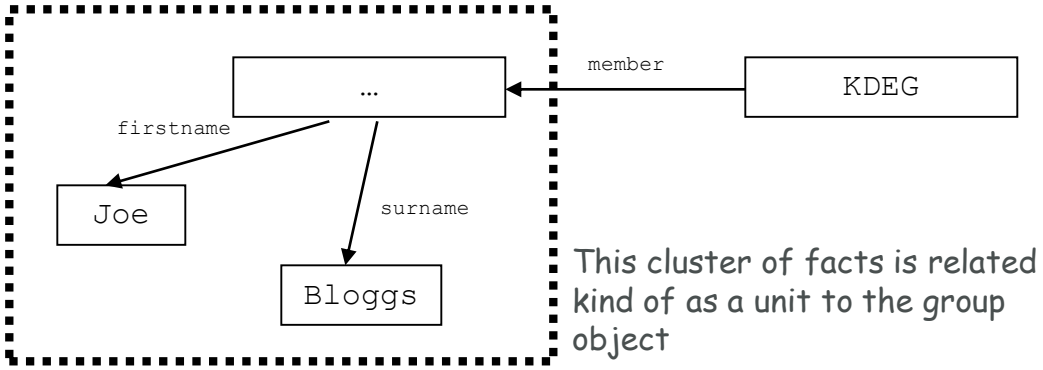
Each of these gives rise to a triple with the same subject (inherited from the containing `Description` element)

Structured values

Suppose we want to talk about a member of KDEG and capture their information too

- Not information about KDEG *per se*, so shouldn't be held as child elements of the `Description` element
- Plus it'd get very confusing when there were several people…

The solution is to define another cluster of facts and relate this *cluster* to KDEG node



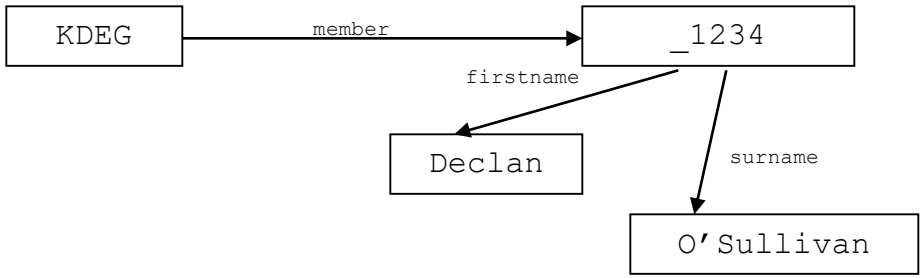
Synthetic structured values

Sometimes the structured value doesn't have an obvious identity to put in an `about` attribute

If this is the case, RDF will synthesise one

```
<s:member>
  <rdf:Description>
    <s:surname>O'Sullivan</s:surname>
    <s:firstname>Declan</s:firstname>
    …
  </rdf:RDF>
</s:person>
```

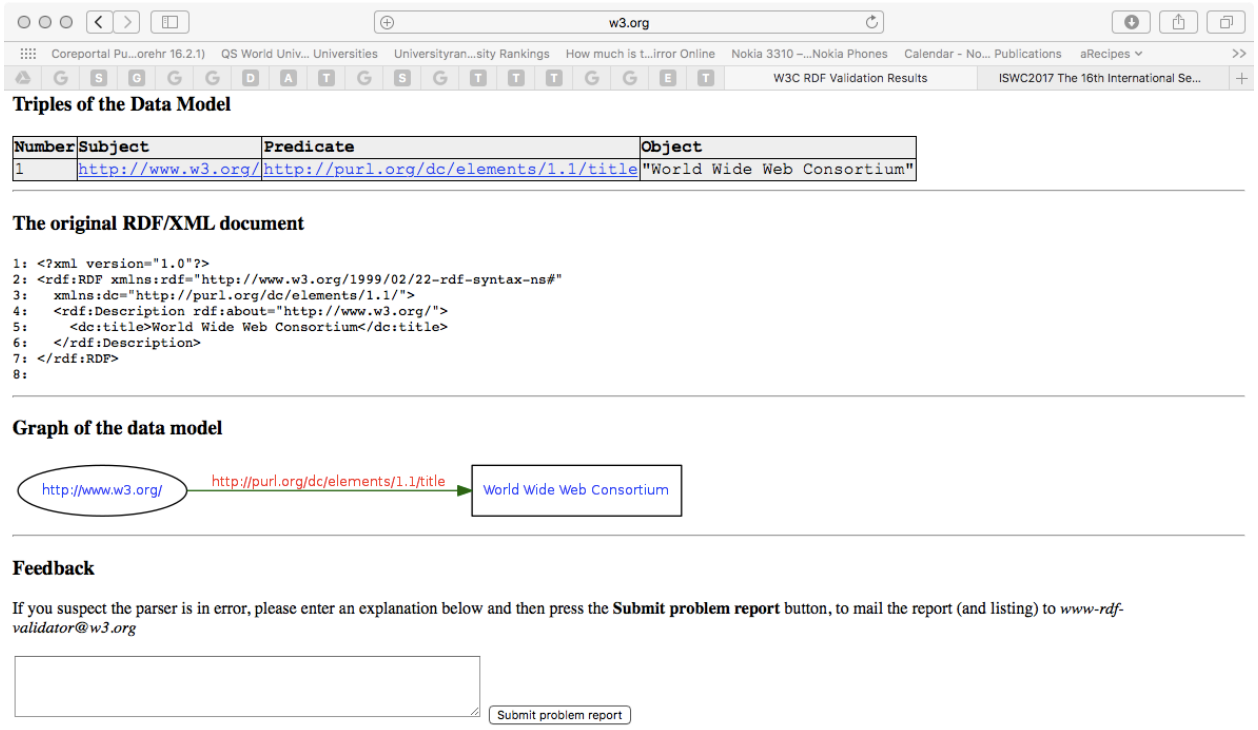
Description doesn't have an about attribute, so RDF synthesises a unique one for use in the graph



Validating RDF

Validating and Graphing

- http://www.w3.org/RDF/Validator/



Uplifting Tabular data into RDF

R2RML: RDB to RDF Mapping Language

- A W3C Recommendation since fall 2012
- http://www.w3.org/TR/r2rml/

Creating an R2RML file that annotates a relational database with existing vocabularies and/or ontologies (RDFS or OWL).

That R2RML file goes through an *R2RML Mapping Engine* to produce RDF.

Graphs of information– 2

Things to note

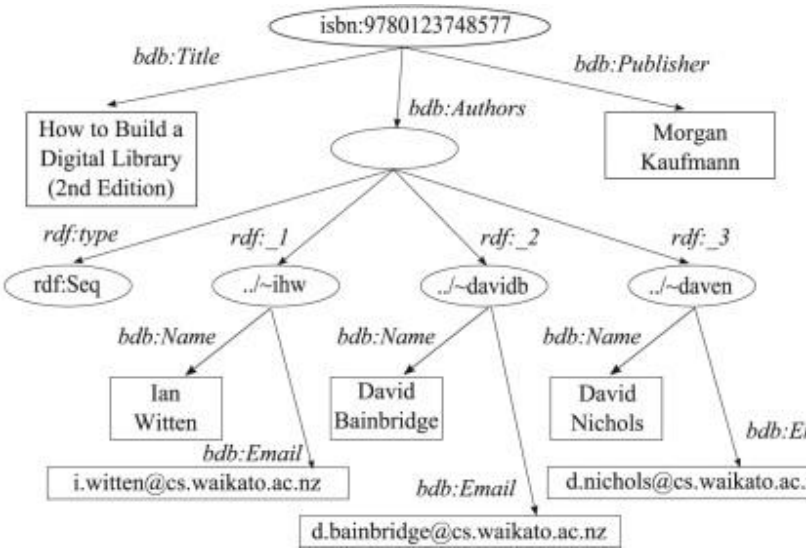
- Scaling – the same graph can represent a load of different information simultaneously
- Agreement – need to know what the various predicates 'mean'
- Structure – you need to know what nodes are related by a predicate
- Plurality – the same relationship may appear several times
- Symmetry – the same predicates can be used for common information, despite minor changes
- Asymmetry – relationships are inherently directed, which sometimes makes things awkward

…and this can get very tricky

…and this can be difficult to keep straight

For example both lecturers and students have names

RDF and Knowledge Graphs



(a)

(b)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:bdb="http://bookdatabase.org/schema/publications">

  <rdf:Description about="urn:isbn: 9780123748577">
    <bdb:Title>How to build a digital library</bdb:Title>
    <bdb:Authors>
      <rdf:Seq>
        <rdf:li>
          <rdf:Description about="http://www.cs.waikato.ac.nz/~ihw">
            <bdb:Name>Ian Witten</bdb:Name>
            <bdb:Email>i.witten@cs.waikato.ac.nz</bdb:Email>
          </rdf:Description>
        </rdf:li>
        <rdf:li rdf:resource="http://www.cs.waikato.ac.nz/~davidb"/>
        <rdf:li rdf:resource="http://www.cs.waikato.ac.nz/~daven"/>
      </rdf:Seq>
    </bdb:Authors>
    <bdb:Publisher>Morgan Kaufmann</bdb:Publisher>
  </rdf:Description>

  <rdf:Description about="http://www.cs.waikato.ac.nz/~davidb">
    <bdb:Name>David Bainbridge</bdb:Name>
    <bdb:Email>d.bainbridge@cs.waikato.ac.nz</bdb:Email>
  </rdf:Description>

  <rdf:Description about="http://www.cs.waikato.ac.nz/~daven">
    <bdb:Name>David Nichols</bdb:Name>
    <bdb:Email>d.nichols@cs.waikato.ac.nz</bdb:Email>
  </rdf:Description>

</rdf:RDF>
```

How to build a digital library David Bainbridge et al

XML and RDF

When considering RDF, you should keep in mind that it is a sister format to XML, not a subsidiary.

However, this medium bestows practical benefits: software support for parsing and editing, transparent handling of international characters, and so on.

(b)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:bdb="http://bookdatabase.org/schema/publications">

  <rdf:Description about="urn:isbn: 9780123748577">
    <bdb:Title>How to build a digital library</bdb:Title>
    <bdb:Authors>
      <rdf:Seq>
        <rdf:li>
          <rdf:Description about="http://www.cs.waikato.ac.nz/~ihw">
            <bdb:Name>Ian Witten</bdb:Name>
            <bdb:Email>i.witten@cs.waikato.ac.nz</bdb:Email>
          </rdf:Description>
        </rdf:li>
        <rdf:li rdf:resource="http://www.cs.waikato.ac.nz/~davidb"/>
        <rdf:li rdf:resource="http://www.cs.waikato.ac.nz/~daven"/>
      </rdf:Seq>
    </bdb:Authors>
    <bdb:Publisher>Morgan Kaufmann</bdb:Publisher>
  </rdf:Description>

  <rdf:Description about="http://www.cs.waikato.ac.nz/~davidb">
    <bdb:Name>David Bainbridge</bdb:Name>
    <bdb:Email>d.bainbridge@cs.waikato.ac.nz</bdb:Email>
  </rdf:Description>

  <rdf:Description about="http://www.cs.waikato.ac.nz/~daven">
    <bdb:Name>David Nichols</bdb:Name>
    <bdb:Email>d.nichols@cs.waikato.ac.nz</bdb:Email>
  </rdf:Description>

</rdf:RDF>
```

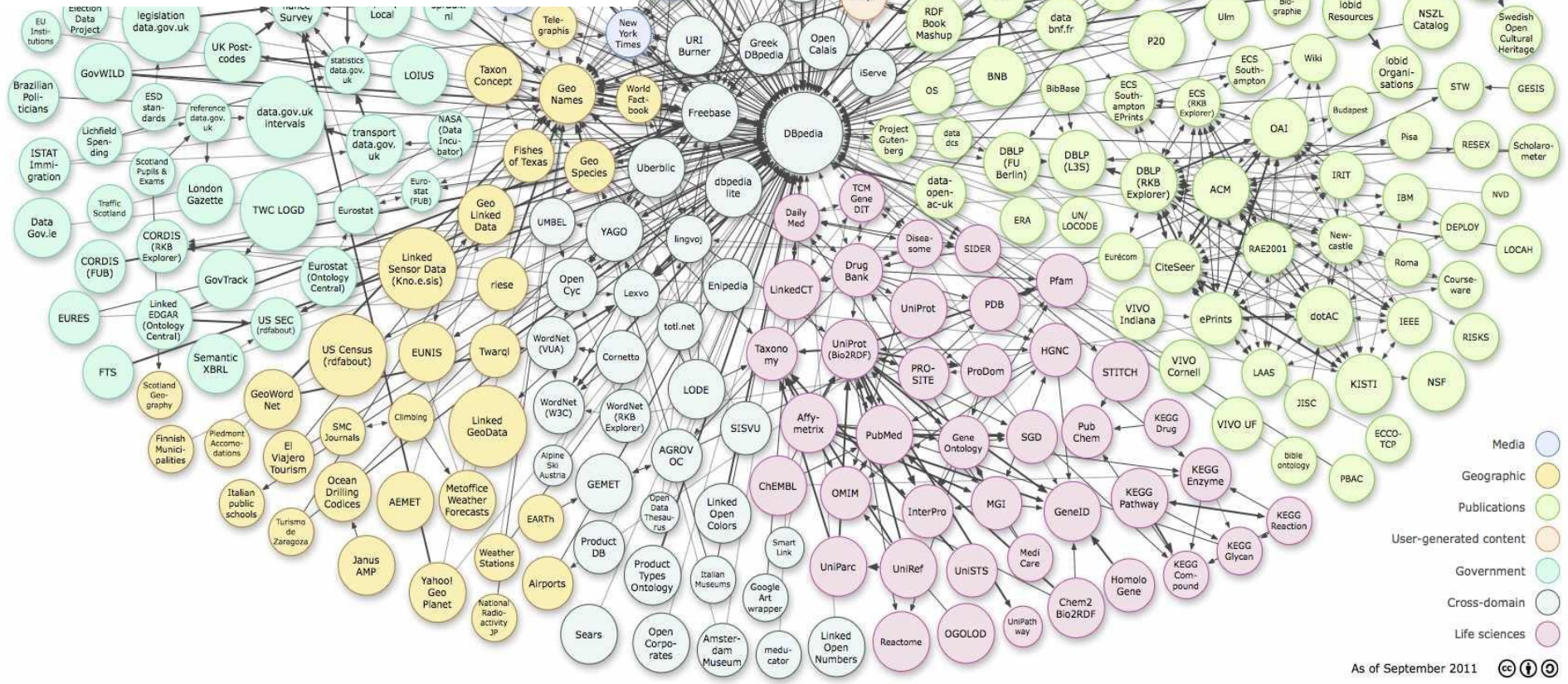
How to build a digital libraryDavid Bainbridge et al

Circle size **Triple Count**

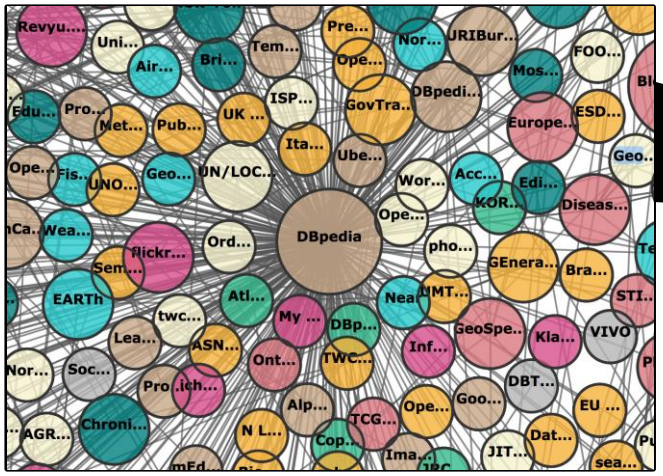
Very large >1B
Large 1B-10M
Medium 10M-500k
Small 500k-10k
Very small <10k

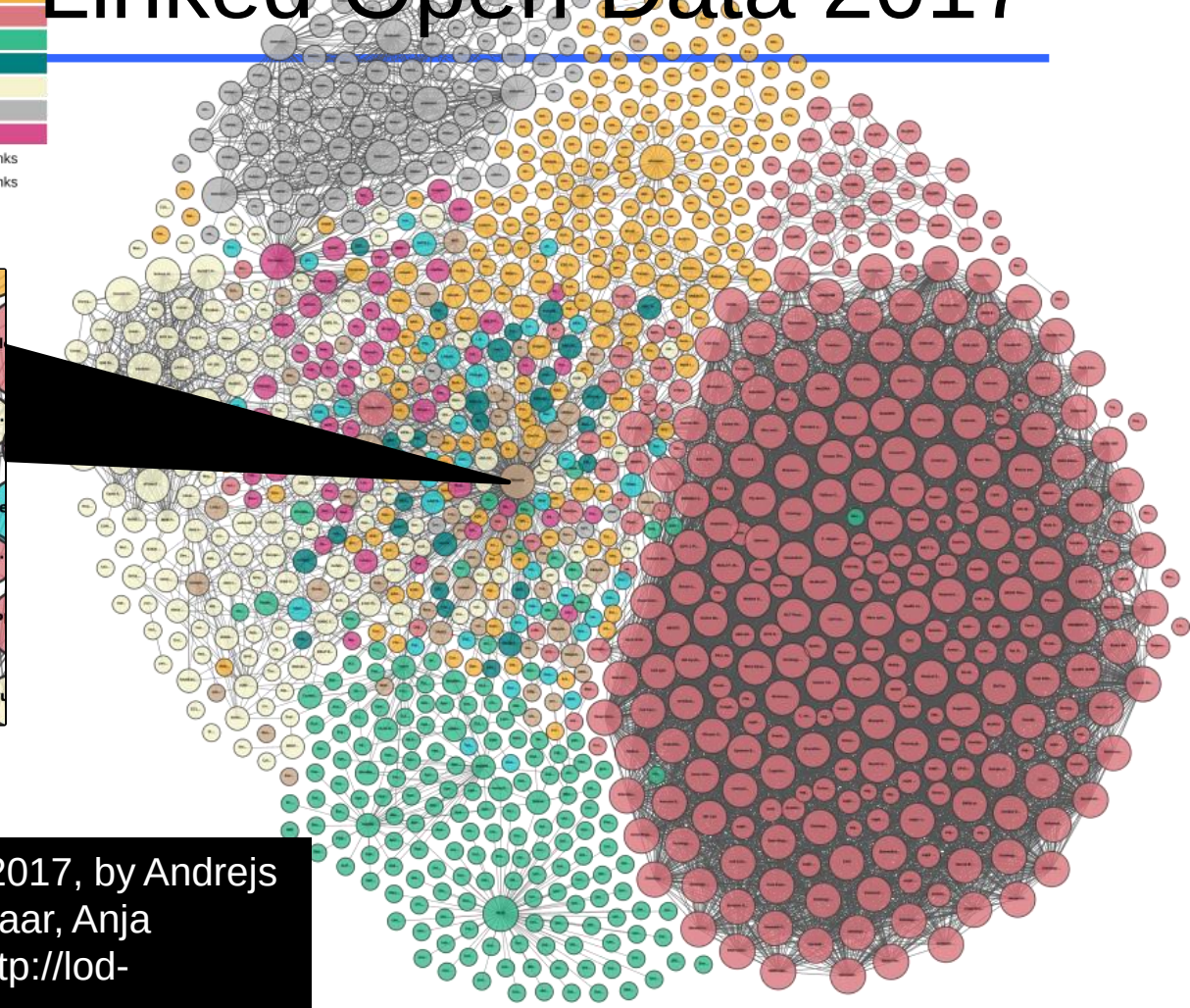
The arrows indicate the existence of at least 50 links between two datasets.

Linked Open Data 2012



Linked Open Data 2017





Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. http://lod-cloud.net/

About the diagram
This web page is the home of the *LOD cloud diagram*. This image shows datasets that have been published in the Linked Data format.
The dataset currently contains **1,239** datasets with **16,147** links (as of March 2019)

Maintainers

The diagram is maintained by John P. McCrae for the Insight Centre for Data Analytics. The following people have previously maintained the diagram or have also helped in current development of the diagram:

- Andrejs Abele
- Paul Buitelaar
- Richard Cyganiak
- Anja Jentzsch
- Vladimir Andryushechkin
- Jeremy Debattista

Contributing to the Diagram

First, make sure that you publish data according to the Linked Data principles. We interpret this as:

- There must be *resolvable http:// (or https://) URIs*.
- They must resolve, with or without content negotiation, to *RDF data* in one of the popular RDF formats (**RDFa, RDF/XML, Turtle, N-Triples**).
- The dataset must contain *at least 1000 triples*. (Hence, your FOAF file most likely does not qualify.)
- The dataset must be connected via *RDF links* to a dataset that is already in the diagram. This means, either your dataset must use URIs from the other dataset, or vice versa. We arbitrarily require at least 50 links.
- Access of the *entire* dataset must be possible via *RDF crawling*, via an *RDF dump*, or via a *SPARQL endpoint*.

Sample Linked Data Exam Question

1. The move to Linked Data (and eventually the Semantic Web) will bring benefits for application developers, compared to how data is currently available on the web.

   Discuss the statement above. Diagrams can be included to support or illustrate points made in your discussion.
   Include at least the following points in your answer.
   - Describe the benefits that Linked Data could bring;
   - Explain the concept of Linked Data;
   - Explain the concept of the Semantic Web;
   - Describe the Semantic Web Stack;
   - Explain in what way OWL builds on RDF and what benefits this brings.

   [Total 50 Marks]

Essay based answer

That's All Folks
Thank You for Listening