

Intelligenza Artificiale 2: Linked Open Data

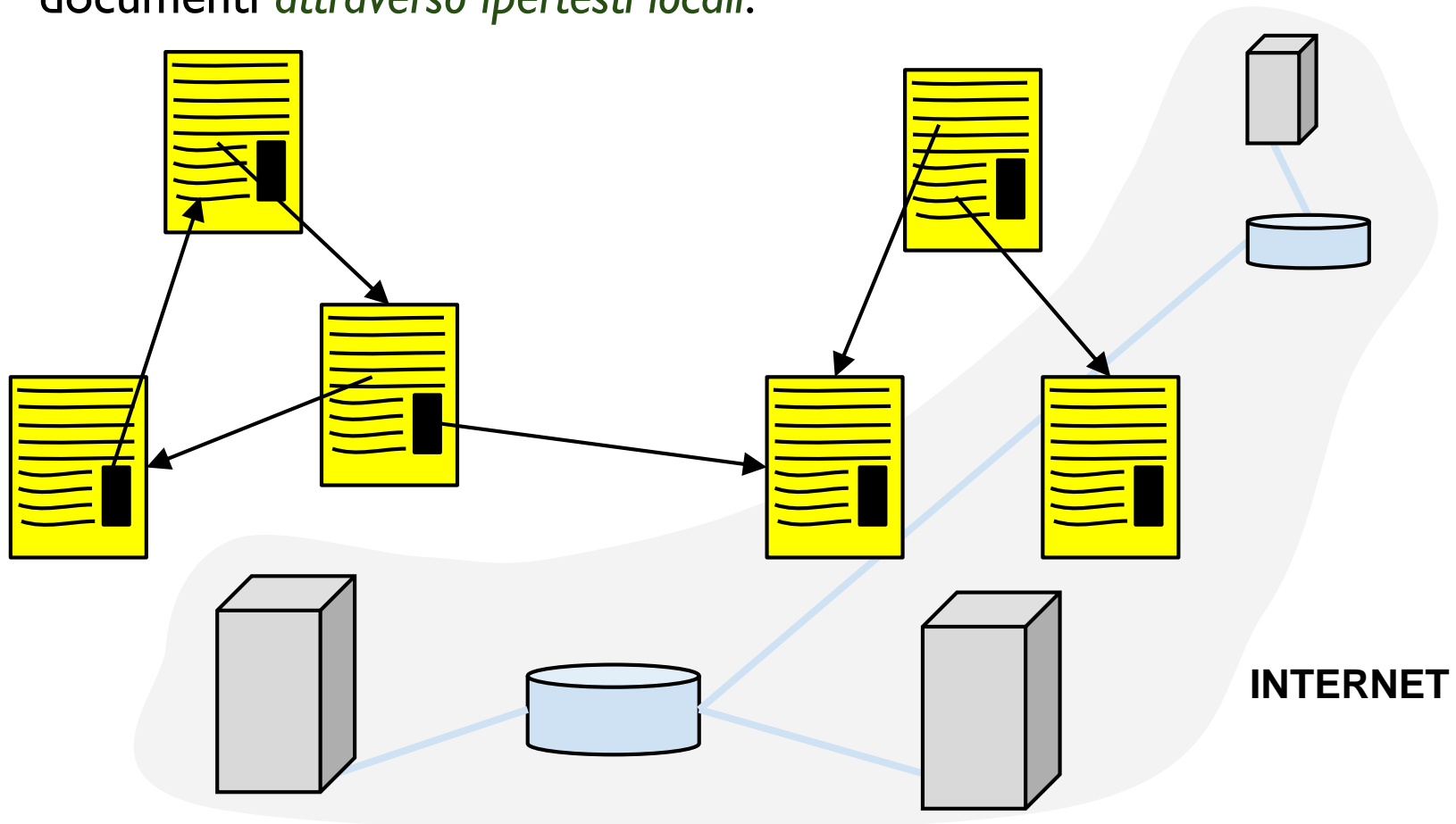
**“Linked Open Data:
Il Web Semantico fatto bene”
(Tim Berners-Lee)**

Armando Stellato
<stellato@uniroma2.it>

- Dal Web dei Documenti...
- ...al Web dei Dati
- Le regole dei Linked Data
- Pubblicazione dei Linked Data
- Consumare Linked Data
- Note finali

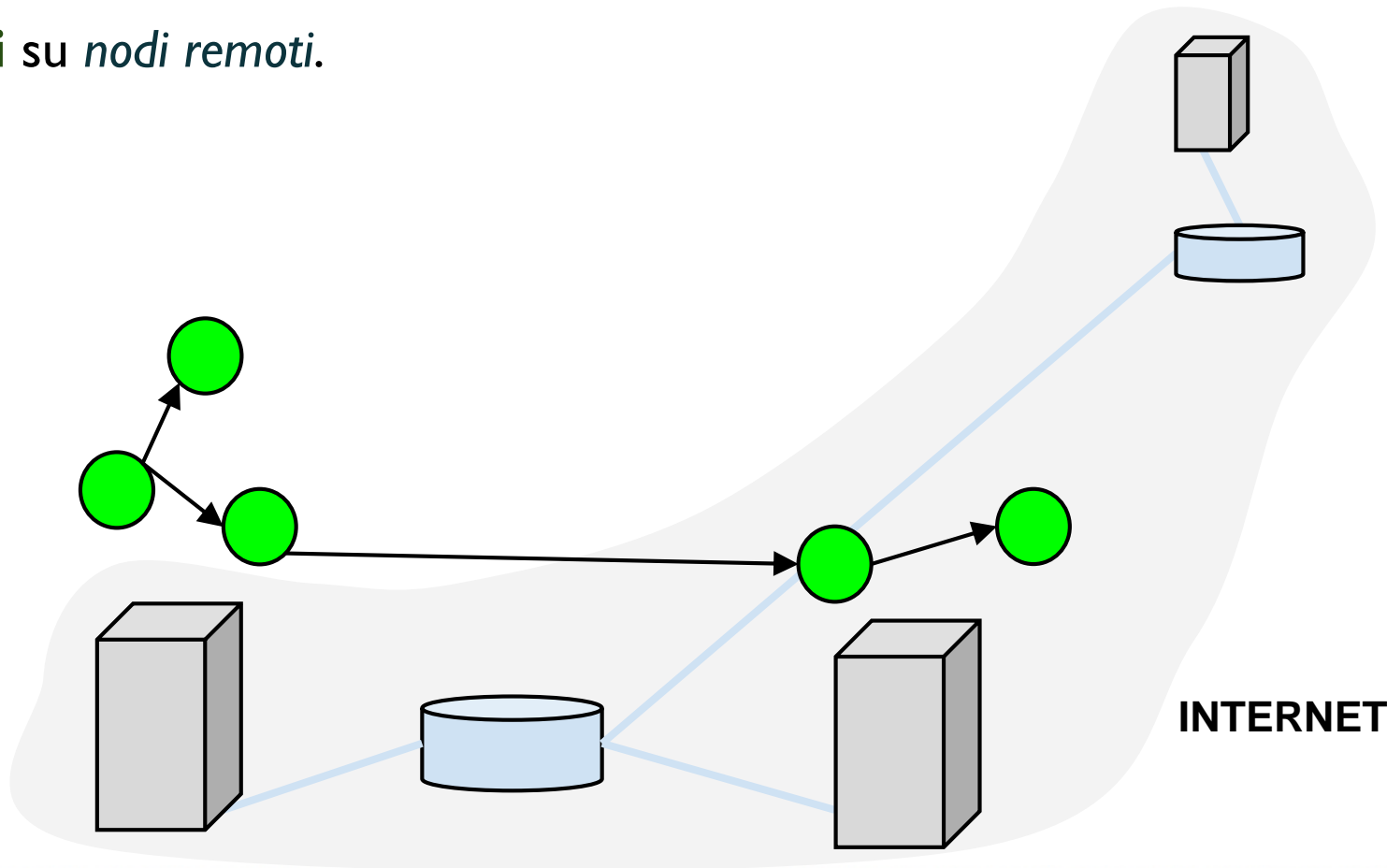
Dal Web dei Documenti

Identificatori globali (vedi [RFC 3986](#)) sono usati per *cucire* insieme documenti *attraverso ipertesti locali*.



...al Web dei Dati

Similmente, possiamo usare le *tecnologie web* per **spargere il grafo dei dati** su *nodi remoti*.



...usando le regole dei Linked Data

Ovviamente, ci sono *molti modi per collegare i dati* insieme, ma in genere facciamo riferimento alle seguenti regole dei **Linked Data** ([Tim Berners-Lee, 2006](#)):

1. Usa **URI** come nomi per le cose
2. Usa **URI HTTP** in modo che le persone possano consultare (look up) quei nomi.
3. Quando qualcuno **consulta uno URI**, fornisci *informazioni utili*, usando gli standard (RDF*, SPARQL)
4. Includi **collegamenti (link) ad altri URI** in modo che si possano *scoprire più cose*

Regola #1: URI come nomi per le cose (2)

Citando la *Architecture of the World Wide Web* ([Jacobs & Walsh, 2004](#)):

Il World Wide Web (WWW, o semplicemente **Web**) è uno *spazio informativo* (*information space*) nel quale le entità di interesse, cui ci si riferisce come risorse, sono identificate da *identificatori globali* chiamati **Uniform Resource Identifier** (URI).

Il primo *passo verso il Web dei Dati* consiste nell'assegnare uno URI ai soggetti di interesse.

Si potrebbe obiettare che il **Web** riguarda le *risorse informative* (*information resource*), mentre il **Web dei Dati** si interessa principalmente delle *entità* nell'*universo del discorso* (sia concrete che astratte).

Tuttavia, l'*RFC 3986* chiarisce che lo URI in sé *fornisce soltanto l'identificazione*; l'accesso alla risorsa non è garantito né implicato dalla presenza di uno URI.

Regola #1: URI come nomi per le cose (2)

La **Sintassi Astratta di RDF** permette l'uso di *URI* come pure di *blank node* (e literal).

La *prima regola* è rilevante in quanto *promuove l'uso degli URI al posto dei blank node*, per il bene della **identificazione globale** permettendo "*a chiunque di parlare di qualsiasi cosa in qualunque luogo*".

Avendo **ambito (scope) globale**, gli URI dovrebbero denotare la stessa risorsa a prescindere dal contesto in cui appaiono, *permettendo così alle persone di parlare di risorse di terze-parti (third-party)*.

La *creazione di uno URI* per una risorsa è chiamata **coniazione (minting)**; per convenzione sociale, soltanto il *proprietario di uno URI* o i suoi delegati possono assegnarlo (*evitando collisioni* nell'assegnazione degli identificatori in una *maniera federata*):

e.g. <http://art.uniroma2.it/fiorelli>

Regola #2: URI HTTP

Una volta che uno URI è stato coniato, il suo proprietario deve comunicare cosa esso dovrebbe significare.

La **seconda regola** richiede che questi *URI usino lo schema HTTP* (e.g. *http://purl.org/dc/elements/1.1/creator*) in modo che essi possano essere **consultati** per recuperare *informazioni utili circa i loro referenti*.

Questa pratica era *conosciuta per gli schemi* (es. per recuperare la definizione XSD di un namespace XML). Per contro, questa pratica è *nuova per i dati ground*.

Il modello di dati *RDF* usa gli URI *soltanto come nomi logici*.

Regola #3: Restituire informazioni usando standard (1)

La **terza regola** si focalizza sull'uso di *standard per codificare le informazioni* circa una risorsa.

La famiglia dei linguaggi basata su RDF è una scelta conveniente

- il **modello di dati orientato ai grafi** è ottimizzato per la *distribuzione* e *l'integrazione*;
- *Formati di serializzazione ampiamente accettati* (es. RDF/XML e Turtle);
- Dati *auto-descrittivi* attraverso definizioni formali dei vocabolari;
- **Assunzione di mondo aperto (open world)** e **semantica monotona** sono appropriate per essere usate sul Web
 - le informazioni circa una risorsa *non sono complete*
 - trattare *diversi livelli di comprensione* delle informazioni

Regola #3: Restituire informazioni usando standard (2)

Consultare lo URI di una risorsa in una base di dati RDF dovrebbe restituire un documento RDF che descrive quella risorsa.

Un approccio consiste nello generare la *Symmetric Concise Bounded Description (SCBD)*¹ della risorsa:

- Essa include tutti gli statement nei quali la risorsa compare come soggetto o oggetto;
- e, ricorsivamente, tutti gli statement che includono ogni blank node che compare nel sottografo calcolato fino a quel momento

Corollario: usando la SCBD, ogni tripla compare nella descrizione del suo soggetto come pure nella descrizione del suo oggetto.

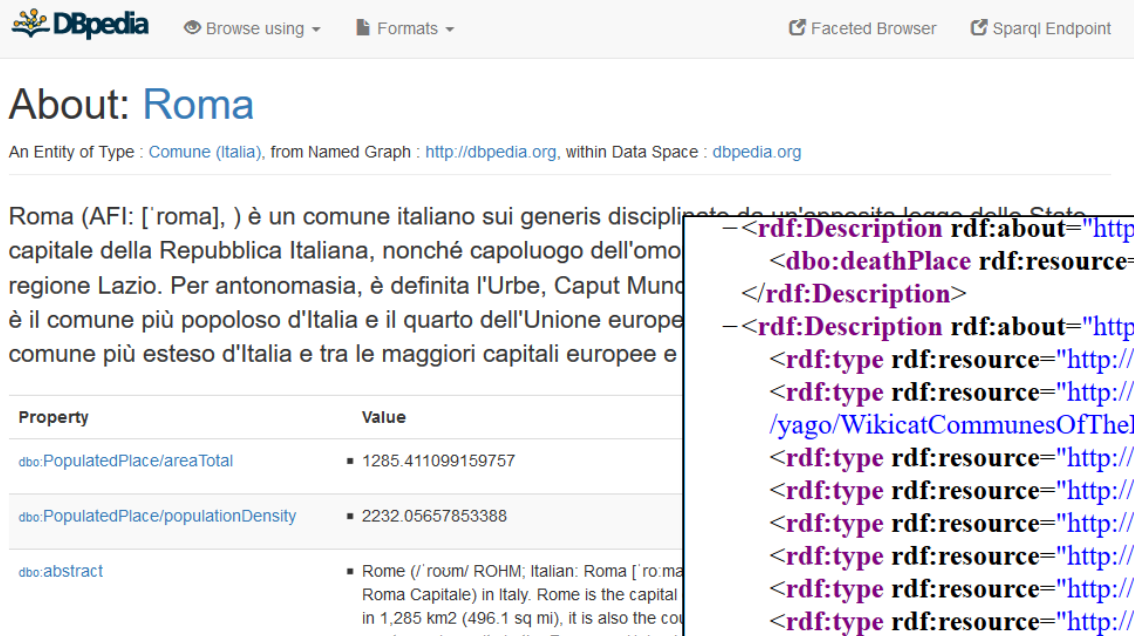
¹ <https://www.w3.org/Submission/CBD/>

Regola #3: Restituire informazioni usando standard (3)

<http://dbpedia.org/resource/Rome>

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

<http://dbpedia.org/page/Rome>



The screenshot shows the DBpedia web interface for the entity 'Roma'. It includes the DBpedia logo, navigation options like 'Browse using' and 'Formats', and a 'Faceted Browser' and 'Sparql Endpoint' link. The main heading is 'About: Roma'. Below it, there is a description of Roma as an Italian municipality and capital. A table lists properties such as 'dbo:PopulatedPlace/areaTotal' and 'dbo:PopulatedPlace/populationDensity' with their respective values.

Accept: application/rdf+xml

<http://dbpedia.org/data/Rome.xml>

```
-<rdf:Description rdf:about="http://dbpedia.org/resource/Romano_Mussolini">
  <dbo:deathPlace rdf:resource="http://dbpedia.org/resource/Rome"/>
</rdf:Description>
-<rdf:Description rdf:about="http://dbpedia.org/resource/Rome">
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatRomanTownsAndCities"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatCommunesOfTheProvinceOfRome"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatMuseumsInRome"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatHolyCities"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatAncientCities"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/Town108665504"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatCitiesAndTownsInItaly"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatCapitalsInEurope"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatCitiesWithMillionsOfInhabitants"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/WikicatWorldHeritageSitesInItaly"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/Depository103177349"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/Museum103800563"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/Region108630985"/>
  <rdf:type rdf:resource="http://dbpedia.org/class/yago/Location100027167"/>
```

Servirsi della *content negotiation* per richiedere la rappresentazione più adatta

Regola #4: Link

L'ultimo passo verso il Web dei Dati consiste nel collegare assieme le risorse per mezzo di **link**.

Questi link sono espressi come una **tripla** il cui *soggetto* e *oggetto* vivono in *dataset differenti*.

ex:fiorelli ex:livesIn dbpedia:Rome

dbpedia: è legato al namespace
<http://dbpedia.org/resource/>
 Posseduto dal dataset DBpedia

Link che attraversano i confini di un dataset supportano il **riuso**, l'**integrazione** e la **scoperta dei dati**.

Proliferazione degli identificatori

All'interno del framework dei Linked Data gli URI sono **sovraccaricati** essendo usati come:

- *identificatori* per parlare di qualcosa;
- un meccanismo per *recuperare* la descrizione (parziale) di qualcosa.

A chiunque dovrebbe essere permesso di *coniare un altro URI per una risorsa*; altrimenti, se le risorse avessero soltanto uno URI, ci sarebbe soltanto un posto per cercare informazioni su quella risorsa.

D'altra parte, la proliferazione di molteplici URI (*coreferenti*) per la stessa risorsa *contrastata l'aggregazione* della conoscenza disponibile su quella risorsa.

Riuso degli identificatori

Si dovrebbero **riusare termini** da *vocabolari popolari* (es. FOAF, Dublin Core, ...) e individui da *dataset di terze parti* (e.g. DBpedia, geonames, ...).

Altrimenti, un *nuovo termine dovrebbe essere collegato* al resto della nuvola (cloud) dei Linked Data (es. una nuova classe estende una classe esistente, un nuovo individuo è in realtà la stessa risorsa – *owl:sameAs* – esistente in un dataset popolare, ...)

L'integrazione dei dati è un *processo in corso*

distribuito tra:

- **Chi pubblica i dati** e vuole che essi siano *massimamente accessibili*
- **Chi consuma i dati** ed ha bisogno di *combinare dataset disparati*

Riconoscere che *due risorse sono in realtà la stessa* è un compito difficile, che prende diversi nomi:

- al **livello di individui**
identity resolution / objection consolidation /
deduplication
- al **livello concettuale**
schema mapping / ontology matching

La terminologia variegata è giustificata dall'uso di approcci e algoritmi differenti, persino sviluppati da comunità diverse.

PUBBLICAZIONE

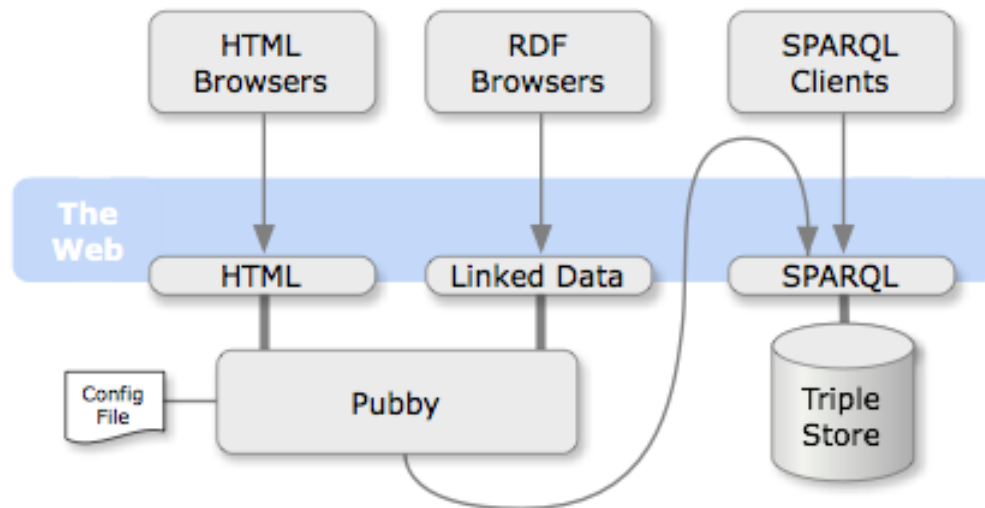
Le *descrizioni RDF* delle risorse trovate in un dataset possono essere pubblicate come **documenti statici** per mezzo di un server web.

La **manutenzione** di questi documenti è difficile e la necessità di immagazzinare le triple in più documenti (vedi il corollario della CBD simmetrica) può condurre a **inconsistenze**.

Publicazione: documenti dinamici (1)

La soluzione è *generare le descrizioni dinamicamente* interrogando un triple store.

Pubby: <http://www4.wiwiss.fu-berlin.de/pubby/>



Pubby serve URI dereferenziabili inviando una query **DESCRIBE** ad un endpoint SPARQL.

Publicazione: documenti dinamici (2)

LodView (<http://lodview.it/>) è un'alternativa Pubby, che ne riprende i principi di base, aggiungendo nuove caratteristiche e cambiandone alcune (che non piacevano agli sviluppatori di LodView).

Esso ha già molti utenti: <https://github.com/dvcama/LodView/wiki/LodView-users>

Loddy (<https://bitbucket.org/art-uniroma2/loddy/>) è un'altra soluzione ispirata a Pubby basata sul framework JSF: uno degli obiettivi del progetto è rendere facile la customizzazione dell'interfaccia utente usando competenze e tecnologie per il web design.

Publicazione: RDB2RDF (1)

La maggior parte delle applicazioni web sono alimentate da database relazionali, che vale la pena di pubblicare come Linked Data.

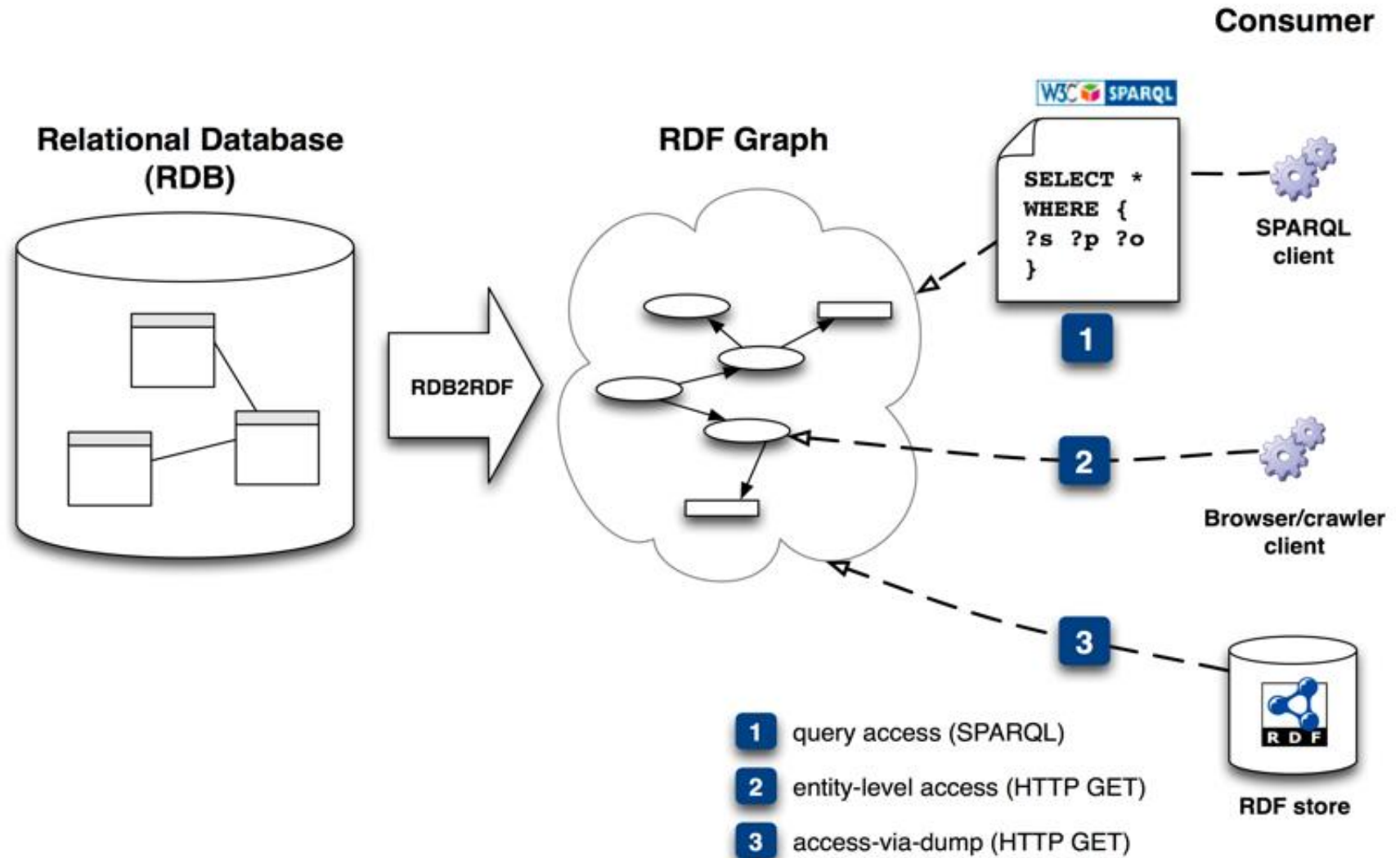
Il W3C RDB2RDF Working Group¹ ha definito **R2RML (RDB to RDF Mapping Language)**², per la specificazione di *mapping da database relazionali a dataset RDF* da servire attraverso un endpoint SPARQL or URI dereferenziabili.

Una situazione simile si è verificata alle origini del Web, quando la creazione dello schema FTP ha dato un impulso alla crescita del Web attraverso l'improvvisa pubblicazione delle numerose risorse FTP già esistenti su Internet.

¹ <http://www.w3.org/2001/sw/rdb2rdf/>

² <https://www.w3.org/TR/r2rml/>

Publicazione: RDB2RDF (2)



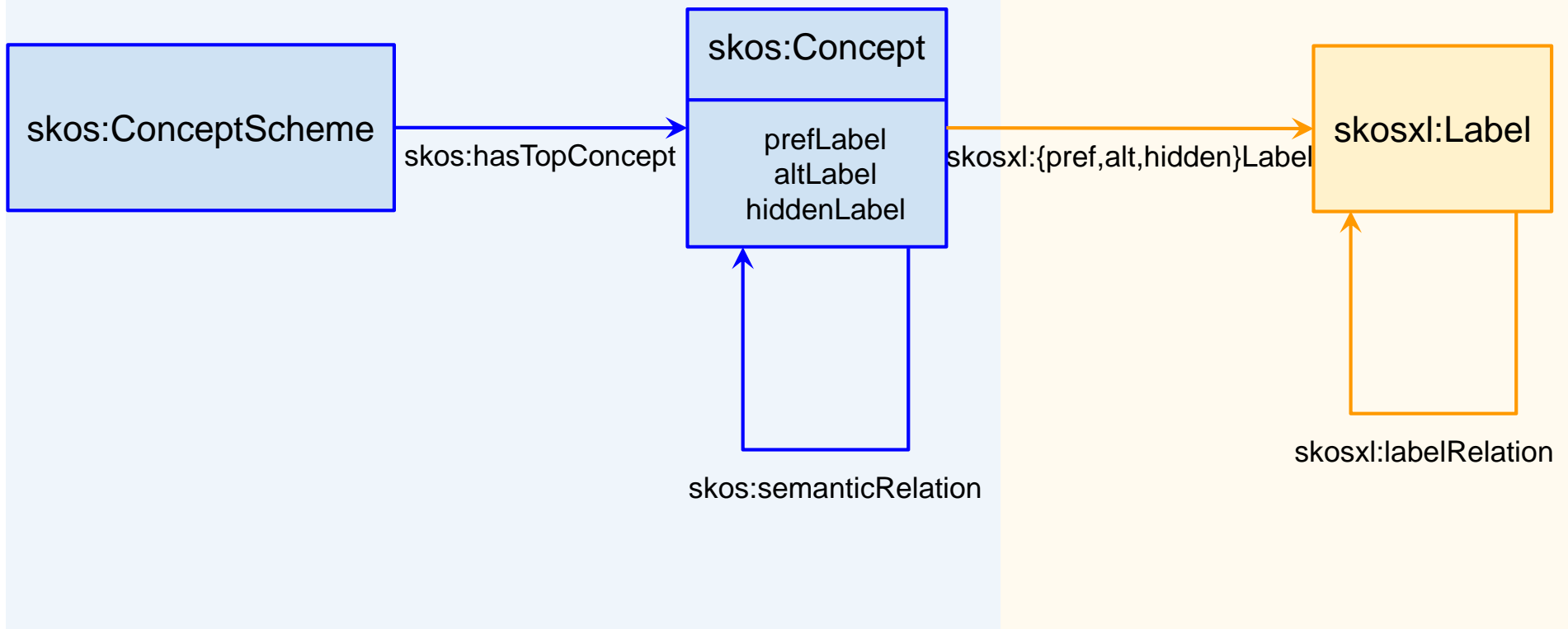
SKOS (Simple Knowledge Organization System) è un vocabolario per rappresentare tesauri, glossari e altri tipi di KOS in RDF:

- SKOS fornisce un percorso veloce per la migrazione di KOS esistenti alle tecnologie del Web Semantico e dei Linked Open Data
- SKOS non sostituisce le linee guida per la compilazione di KOS

SKOS (2/2)

Concept Level
SKOS

Lexical Level
SKOS-XL



Data Cube è un *vocabolario RDF* (sviluppato dal W3C Government Linked Data Working Group) per la pubblicazione di **dati multidimensionali** (es. statistiche) sul web dei dati.

Un dataset multidimensionale consiste di una collezione di **misure** fatte in punti lungo un gruppo di **dimensioni**.

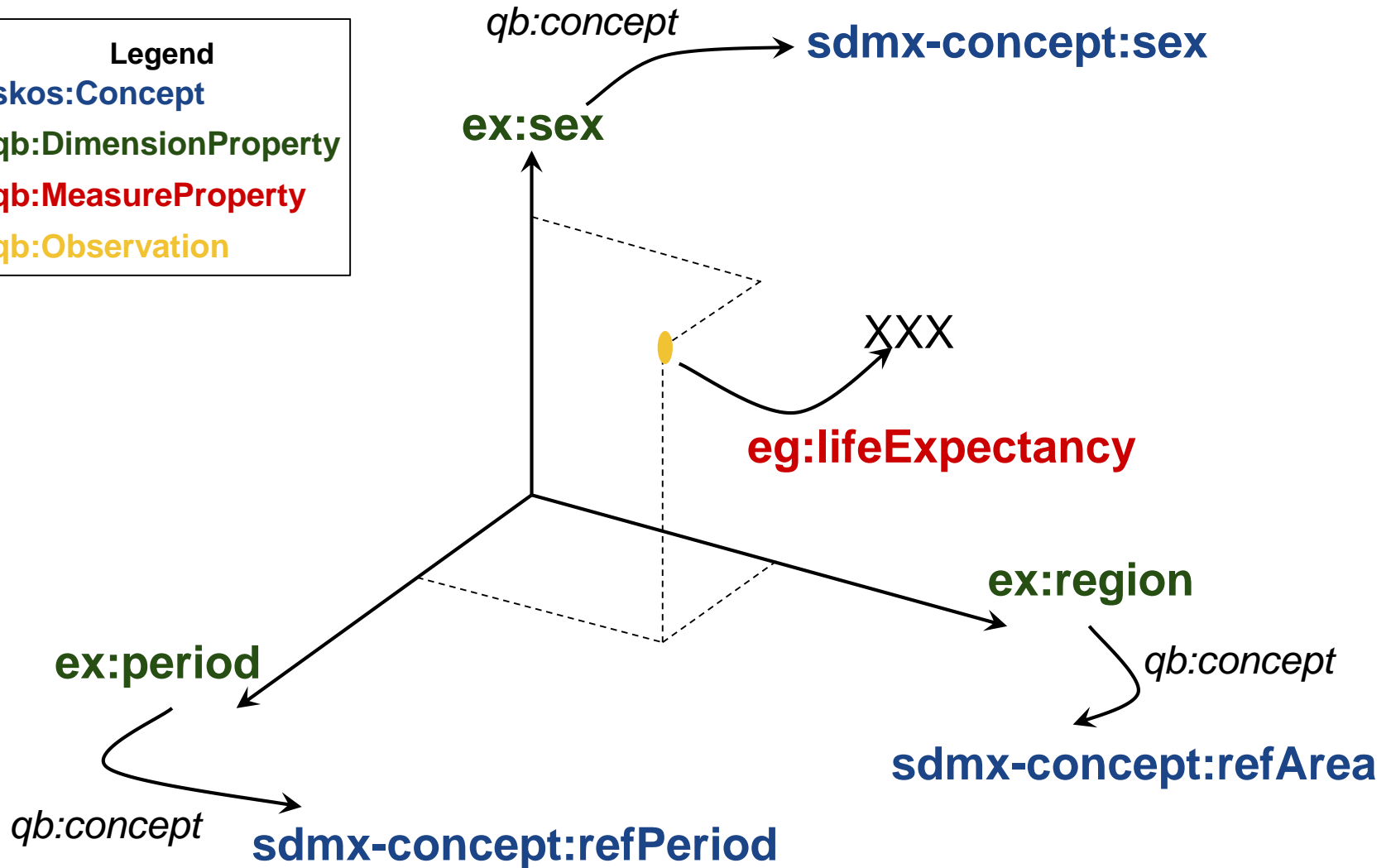
I metadati relativi alle misure (come unità, stato, etc.) sono espresse da **attributi**.

Dimensioni, attributi and *misure* sono chiamate collettivamente **componenti**. Ogni componente può opzionalmente essere collegato al **concetto** che esprime. Questi concetti devono essere SKOS concept.

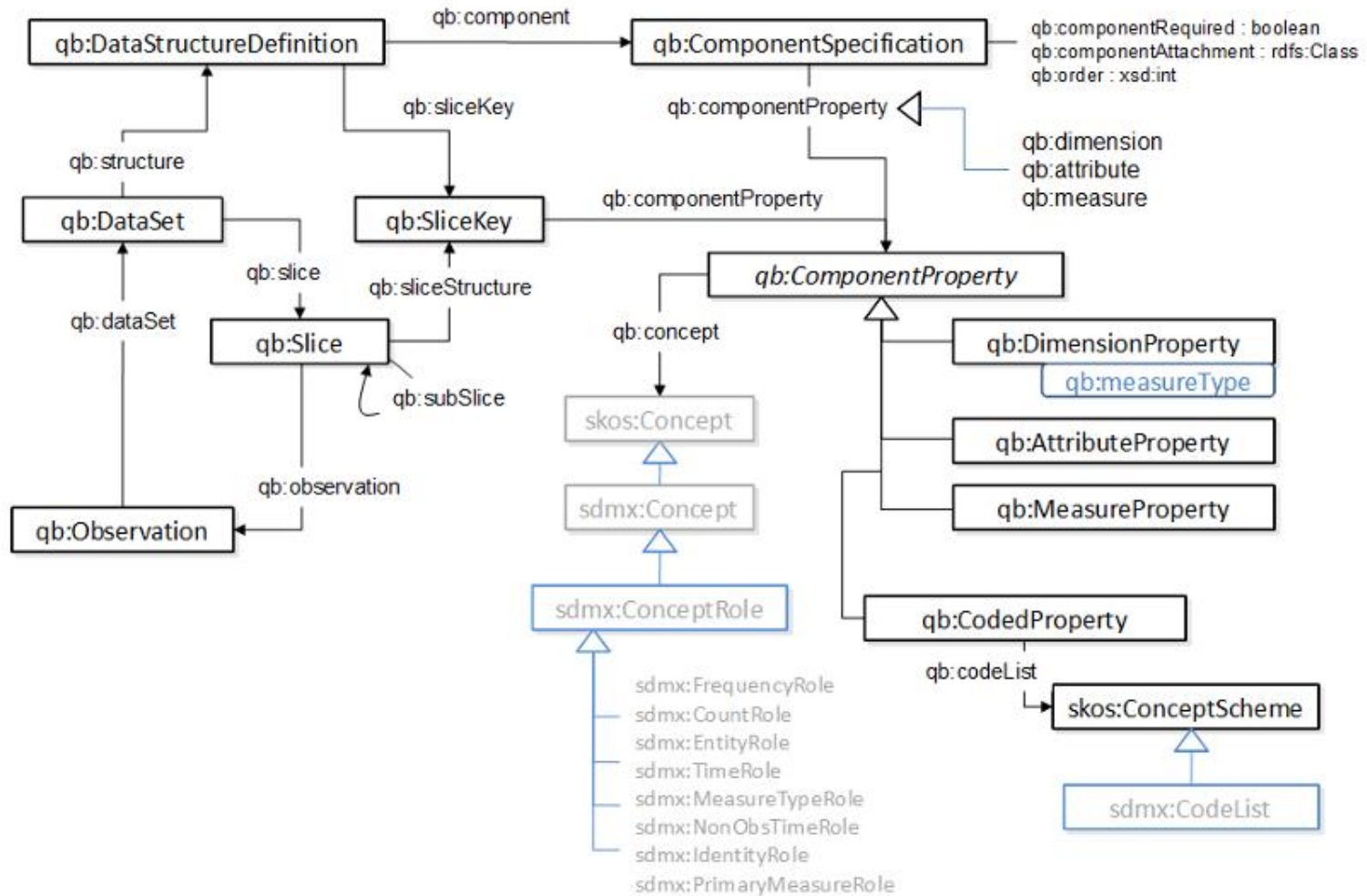
Data Cube (2/3)

Legend

- skos:Concept
- qb:DimensionProperty
- qb:MeasureProperty
- qb:Observation



Data Cube (3/3)



Publicazione: "embedded RDF" (1)

Ci sono vari standard per incorporare dati strutturati all'interno di pagine (X)HTML.

- Annotazioni con dati strutturati inframezzate al contenuto visibile
 - *RDFa*: XHTML e HTML 5 (a partire dalla versione 1.1)
 - *microformat*: (X)HTML
 - *microdata*: HTML5
- Dati strutturati isolati dal contenuto visibile
 - JSON-LD: una sintassi RDF leggera basata su JSON. Può essere usata nella API Web, come pure per rappresentare dati strutturati circa il contenuto di una pagina web (all'interno di un tag `<script>`)

Microformat e *microdata* non sono legati esplicitamente al modello di dati RDF

Publicazione: "embedded RDF" – RDFa (2)



Fonte: <https://www.youtube.com/watch?v=vioCbTo3C-4>

Un esempio che usa RDFa

```
xmlns:dc="http://purl.org/dc/elements/1.1/"    about="http://www
.example.com/books/wikinomics">
```

```
In his latest book <cite property="dc:title">Wikinomics</cite>,
<span property="dc:creator">Don Tapscott</span> explains deep
changes in technology, demographics and business. The book is
due to be published in <span property="dc:date" content="2006-
10-01">October 2006</span>.
```

```
</p>
```

E la sua traduzione in Turtle

```
@prefix : <http://www.example.com/books/> .
```

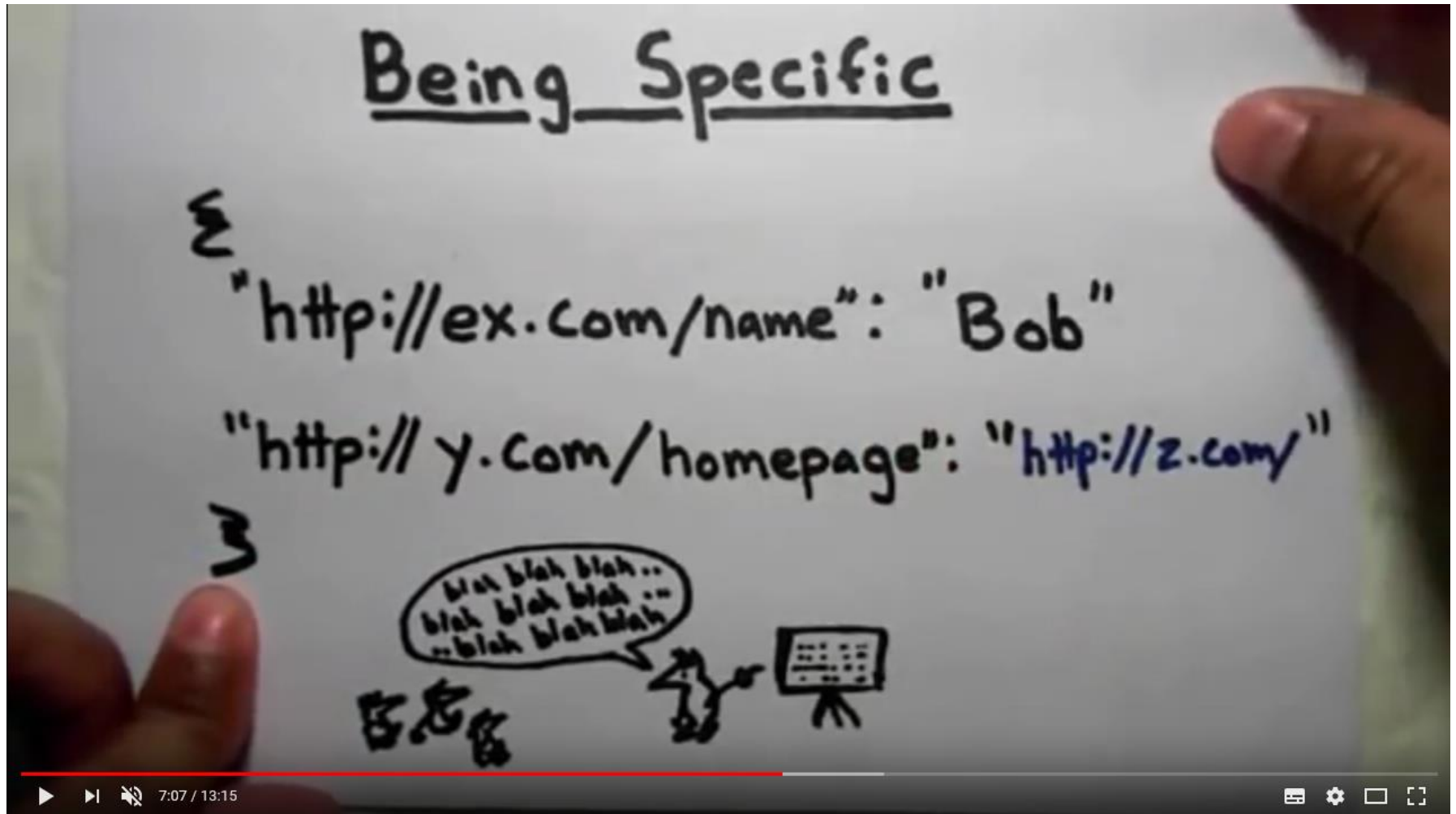
```
@ prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
:wikinomics dc:title "Wikinomics" ;
```

```
dc:creator "Don Tapscott" ;
```

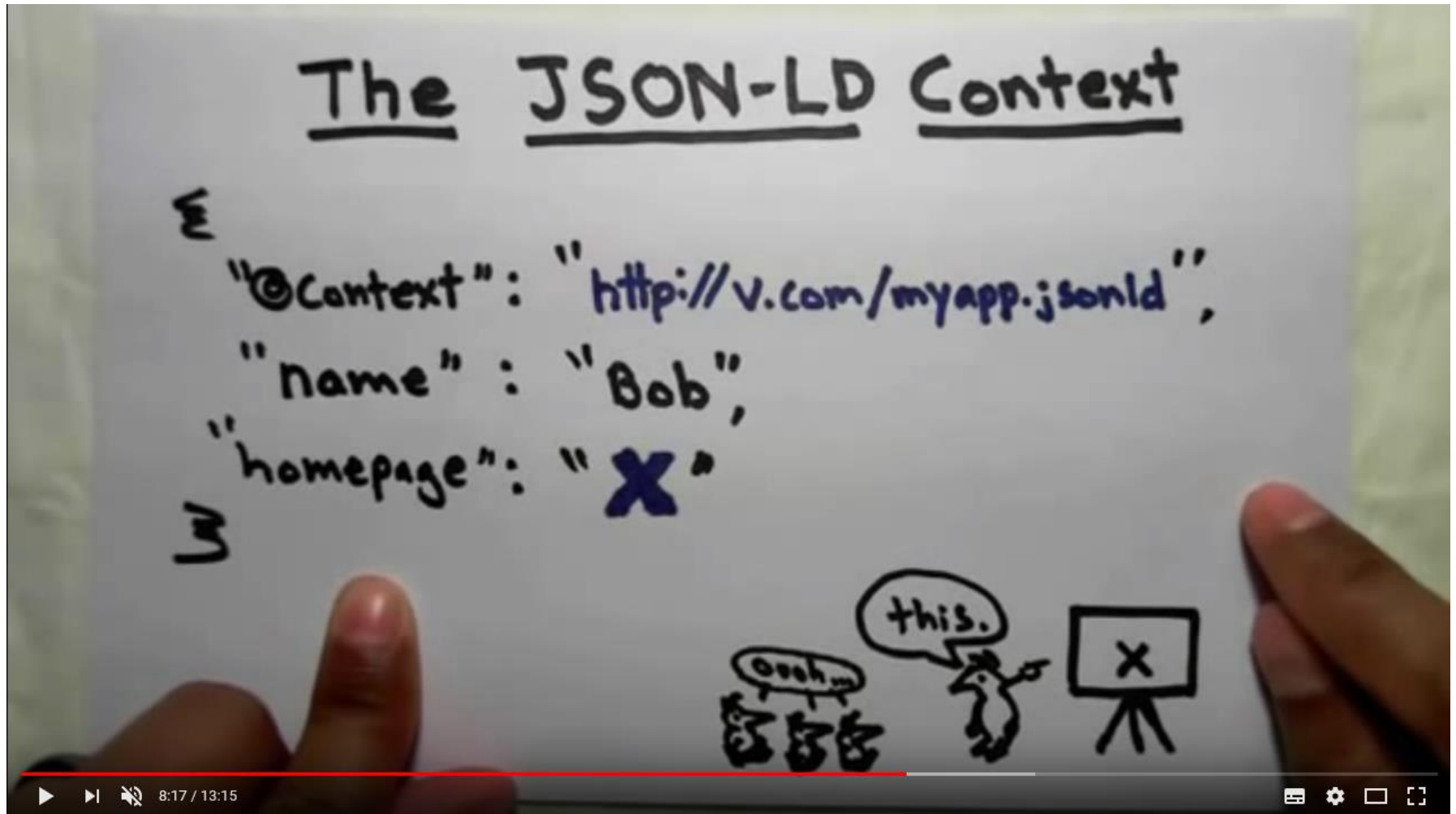
```
dc:date "2006-10-01"^^<http://www.w3.org/2001/XMLSchema#date> .
```

Publicazione: "embedded RDF" – JSON-LD (4)



Fonte: <https://www.youtube.com/watch?v=vioCbTo3C-4>

Publicazione: "embedded RDF" – JSON-LD (5)



Fonte: <https://www.youtube.com/watch?v=vioCbTo3C-4>

Publicazione: "embedded RDF" (5)

L'incorporamento di dati strutturati è utile quando *non è possibile pubblicare altro che pagine (X)HTML*.

I metadata incorporati possono anche *aiutare i motori di ricerca* a indicizzare i siti web e, successivamente, a produrre *risultati migliori* per le ricerche degli utenti.

Bing, Google, Yahoo! e Yandex hanno avviato il progetto schema.org per stabilire degli schemi comuni per l'annotazione di pagine web.

Annotazione semantica di siti web (1)

I dati strutturati sono di primaria importanza nell'ottimizzazione per i motori di ricerca (search engine optimization, SEO):

<https://developers.google.com/search/docs/guides/>

PAGINA INIZIALE

GUIDES

REFERENCE

CASE STUDIES

APIS

TOOLS

SUPPORT

Introduction

Structured data

About Search features

Search feature gallery

Introduction to structured data

Enhance your site's attributes

Mark up your content items

Build, test, & release structured data

Structured data general guidelines

▶ Feature guides

Structured data code lab 

Introduction

This documentation covers technical issues affecting your pages on Google Search. For information about structured data that can affect the appearance of your page on Search, or technical details about how to mark up pages to appear in Google Search with features that are specific to Google Search.

These documents focus on the technical "how". For documentation about "why" you might want to use certain features, or a broader view of how Google Search works, please visit the [Search Console help](#), [structured data guides](#), [overviews](#), and more general information for site owners, SEOs, and content providers.

These guides cover the following topics:

Structured data	Use structured data to help Google understand the content of your site and enable your pages.
---------------------------------	---

[AMP](#) Learn how to get the most out of AMP in Google Search. Implement AMP-specific...

Annotazione semantica di siti web (2)

Google sfrutta i *dati strutturati* per meglio *comprendere il contenuto delle pagine web*, attivando specifiche *Search result feature* per esse.

- *Feature specifiche per un tipo di contenuti* come arricchire un risultato con figure, voti ed altri elementi (a volte interattivi), o inserire una pagina in carosello di ricette, una lista di eventi, etc.
- *Miglioramenti* che si applicano a più di un tipo di contenuti, es. logo, breadcrumb, etc.

La maggior parte dei dati strutturati dovrebbe essere espresso con il *vocabolario Schema.org*, ed immagazzinata preferibilmente come contenuto *JSON-LD* all'interno di tag `<script>`.

Annotazione semantica di siti web (3)

The image shows a Google search interface for the query "spaghetti cacio e pepe". The search results are filtered to "Tutti". The top result is from "GialloZafferano" and includes a picture of the dish, a rating of 4.5 stars, and a 20-minute preparation time. Annotations highlight these elements: a green circle around the picture, a yellow circle around the rating, a red circle around the preparation time, and a blue oval around the dish description. A "Video" section is visible below the main result.

Google

spaghetti cacio e pepe

Tutti Video Immagini Notizie Maps Altro Impostazioni Strumenti

Circa 1.440.000 risultati (0,56 secondi)

Rating

Picture

Ricetta Spaghetti Cacio e Pepe - La Ricetta di GialloZafferano

<https://ricette.giallozafferano.it/Spaghetti-Cacio-e-Pepe.html>

★★★★★ Valutazione: 4,5 - 228 voti

Total time 20 min

Dish description

Gli Spaghetti cacio e pepe sono uno dei piatti forti della tradizione romana: Pecorino grattugiato e grani di pepe, una ricetta veloce e saporita!

Video

Annotazione semantica di siti web (4)

```

isPushing = false,
is_bnzm_pdown = false;
</script>

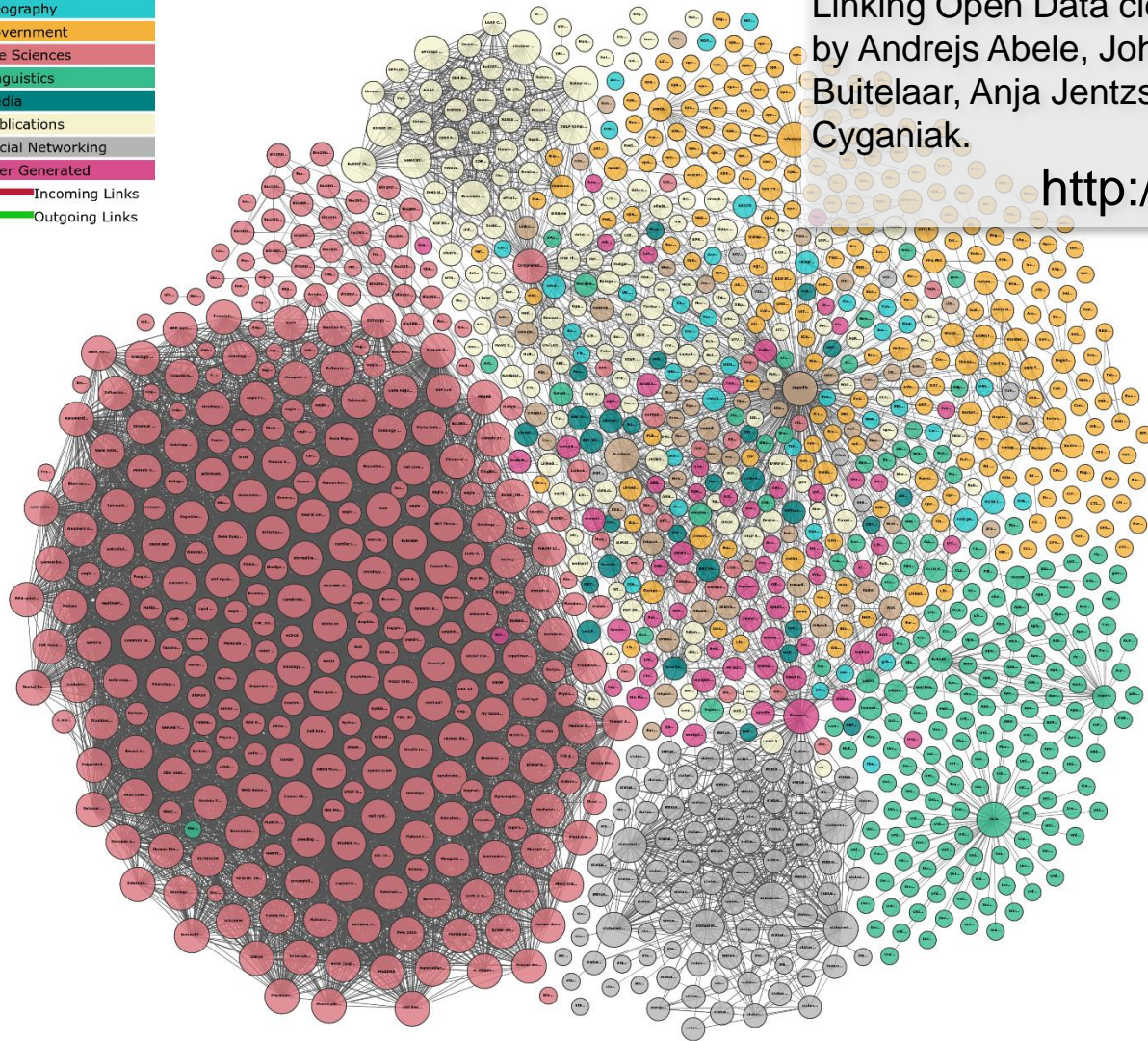
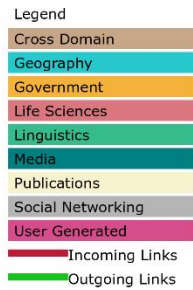
<script
type="application/ld+json">
{"@context":"http://
/schema.org","@type":"Recipe","name"
:"Spaghetti Cacio e Pepe","author":
{"@type":"Organization","name":"Giall
oZafferano"},"image":"https://
/www.giallozafferano.it/images
/ricette/176/17617/foto_hd
/hd650x433_wm.jpg","description":"Gl
i Spaghetti cacio e pepe sono uno dei
piatti forti della tradizione romana:
Pecorino grattugiato e grani di pepe,
una ricetta veloce e
saporita!","prepTime":"PT10M","totalT

```

@type	Recipe
name	Spaghetti Cacio e Pepe
image	https://www.giallozaff erano.it/images /ricette/176/17617 /foto_hd /hd650x433_wm.jpg
description	Gli Spaghetti cacio e pepe sono uno dei piatti forti della tradizione romana: Pecorino grattugiato e grani di pepe, una

<https://search.google.com/structured-data/testing-tool>

Il digramma della LOD cloud



Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak.

<http://lod-cloud.net/>

Vocabulary of Interlinked Datasets (VoID)

VoID (<https://www.w3.org/TR/void/>) è un vocabolario per la descrizione di dataset RDF

Citando la specifica di VoID:

“Un dataset è un insieme di triple RDF che sono pubblicate, mantenute o aggregate da un singolo fornitore”

La descrizione di un dataset include:

- Metadati generali
 - Aiutano i potenziali utenti a stabilire se un dataset è rilevante per i propri scopi
- Metadati di accesso
 - Descrivono i metodi per accedere il contenuto di un dataset
- Metadati strutturali
 - Informazioni riguardanti lo schema e la struttura interna di un dataset
- Descrizione dei link tra dataset

VOID – Descrizioni di dataset (1)

Un approccio per la pubblicazione di metadati VOID consiste nel metterli in un file Turtle chiamato *void.ttl* e piazzato nella radice del sito del dataset.

In basso c'è un frammento del file *http://example.org/void.ttl*:

L'IRI relativo
"stringa vuota" <>
rappresenta l'IRI
del documento

Usare *foaf:topic*
se questo VOID
file descrive più
dataset

```
@prefix void: <http://rdfs.org/ns/void#> .
<> a void:DatasetDescription ;
    foaf:primaryTopic <#ExampleDataset> .
<#ExampleDataset> a void:Dataset ;
...
(ulteriori metadati sul dataset)
...
.
```


VOID – Descrizioni di dataset (2)

Per facilitare la scoperta dei metadata su un dataset, ogni risorsa definita al suo interno dovrebbe essere collegata alla descrizione del dataset.

```
@prefix ex: <http://example.org/resource/> .
ex:SomeResource void:inDataset <http://example.org/void.ttl#ExampleDataset>
```

VOID – Metadati Generali (1)

foaf:homepage è una proprietà funzionale inversa, che può essere quindi usata per la identity resolution

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<#ExampleDataset> a void:Dataset ;
foaf:homepage <http://example.org/> ;
foaf:page <http://datahub.io/...>
.
```

Qualsiasi pagina correlata

VOID – Metadati Generali (2)

VOID raccomanda l'uso di proprietà esistenti dal vocabolario Dublin Core Metadata Terms¹.

Nella prossima slide, il prefisso *dcterms* è legato al namespace *<http://purl.org/dc/terms/>*

¹ <http://dublincore.org/documents/2010/10/11/dcmi-terms/>

VOID – Metadati Generali (3)

dcterms:title	The name of the dataset.
dcterms:description	A textual description of the dataset.
dcterms:creator	An entity, such as a person, organisation, or service, that is primarily responsible for creating the dataset. The creator should be described as an RDF resource, rather than just providing the name as a literal.
dcterms:publisher	An entity, such as a person, organisation, or service, that is responsible for making the dataset available. The publisher should be described as an RDF resource, rather than just providing the name as a literal.
dcterms:contributor	An entity, such as a person, organisation, or service, that is responsible for making contributions to the dataset. The contributor should be described as an RDF resource, rather than just providing the name as a literal.
dcterms:source	A related resource from which the dataset is derived. The source should be described as an RDF resource, rather than as a literal.
dcterms:date	A point or period of time associated with an event in the life-cycle of the resource. The value should be formatted and data-typed as an xsd:date.
dcterms:created	Date of creation of the dataset. The value should be formatted and data-typed as an xsd:date.
dcterms:issued	Date of formal issuance (e.g., publication) of the dataset. The value should be formatted and datatyped as an xsd:date.
dcterms:modified	Date on which the dataset was changed. The value should be formatted and datatyped as an xsd:date.

VOID – Metadati Generali (4)

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<#ExampleDataset> a void:Dataset ;
    dcterms:creator <#John> ;
    dcterms:publisher <#ExampleCompany> ;
    dcterms:issued "29-10-2017"^^xsd:date
    .
[
<#John> a foaf:Person ;
    foaf:name "John" ;
    foaf:mbox <mailto:john@example.org>
    .
<#ExampleCompany> a foaf:Organization ;
    foaf:name "Example Company" ;
    foaf:mbox <mailto:info@example.org>
    .
]
```

Sia il *creator* sia il *publisher* sono rappresentate come risorse, la cui descrizione include informazioni utili come la loro natura (persona vs organizzazione), nome e contatto.

VOID – Metadati Generali (5)

La riusabilità di un dataset dipende chiaramente dalla sua licenza, che deve essere resa nota nella descrizione del dataset.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
<#ExampleDataset> a void:Dataset ;  
    dcterms:license <http://www.opendatacommons.org/licenses/by/>  
    .
```

VOID – Metadati Generali (6)

La proprietà *dcterms:subject* dovrebbe essere usata per indicare l'argomento (topic) del dataset.

L'argomento dovrebbe essere espresso come una risorsa definita da:

- DBpedia
- oppure, altro dataset largamente usato in una specifica comunità

```
<#ExampleDataset> a void:Dataset ;  
    dcterms:subject <http://dbpedia.org/resource/Rome>  
    .
```

VOID – Metadati Generali (7)

La proprietà *void:feature* può essere usata per esprimere feature tecniche (*void:TechnicalFeature*) del dataset (es. serializzazioni RDF supportate)

```
<#ExampleDataset> a void:Dataset ;  
    void:feature <http://www.w3.org/ns/formats/RDF_XML>  
    .
```

Nuove feature tecniche possono essere definite come istanze della classe *void:TechnicalFeature*, accompagnate da informazioni come una *rdfs:label*, un *rdfs:comment* o valori per la proprietà *rdfs:seeAlso*.

VOID – Metadati di Accesso

Un *void:Dataset* è soltanto un surrogato (proxy) di un dataset RDF finalizzato alla rappresentazione di metadati, tra i quali ci sono i meccanismi per accedere effettivamente alle triple.

```
<#ExampleDataset> a void:Dataset ;
  void:sparqlEndpoint <http://example.org/sparql>;
  void:dataDump <http://example.org/dump.nt> ;
  void:dataDump <http://example.org/dump.nt.gz> ;
  void:rootResource <http://example.org/resource/Top1>,
                    <http://example.org/resource/Top2> ;
  void:uriLookupEndpoint <http://example.org/lookup?uri=> ;
  void:openSearchDescription <http://example.org/opensearch.xml>
  .
```

Le radici di un dataset strutturato ad albero

Un dump (possibilmente compresso) di (parte) del dataset

L'endpoint SPARQL del dataset

Una Open Search Description dell'API di ricerca free-text

L'API per consultare un URI (che non è altrimenti dereferenziabile)

VOID – Metadati Strutturali (1)

Le *risorse di esempio* dovrebbero dare un senso immediato del tipo di dati che l'utente può aspettarsi di vedere nel dataset.

```
<#ExampleDataset> a void:Dataset ;
  void:exampleResource <http://example.org/resource/A> ;
  void:exampleResource <http://example.org/resource/B> ;
  .
```

Gli URI delle risorse descritte in un dataset condividono in genere un namespace comune.

```
<#ExampleDataset> a void:Dataset ;
  void:uriSpace "http://example.org/resource/" ;
  .
```

Scenari più complessi possono essere gestite attraverso la proprietà *void:uriRegexPattern* che può contenere una espressione regolare che è soddisfatta dagli URI delle risorse nel dataset.

VOID – Metadati Strutturali (2)

I metadati dovrebbero elencare i vocabolari più importanti
(specialmente, per interrogare il dataset)

```
<#ExampleDataset> a void:Dataset ;  
    void:vocabulary <http://xmlns.com/foaf/0.1/> ;  
    .
```

VOID – Metadati Strutturali (3)

La proprietà `void:subset` mette in relazione un dataset con un suo sottoinsieme, in modo da poter asserire metadati specifici per quest'ultimo.

```
<#ExampleDataset> a void:Dataset ;
  void:subset <#ExampleSubset>
  .

<#ExampleSubset> a void:Dataset ;
  void:datadump <http://example.org/subsetdump.ttl>
  .
```

I *subset* possono anche essere usati per descrivere aggregazioni di dataset.

```
<#Example2Dataset> a void:Dataset ;
  void:sparqlEndpoint <http://example2.org/sparql> ;
  void:subset <http://foo.example.org/void.ttl#FooDataset> ;
  void:subset <http://bar.example.org/void.ttl#BarDataset> ;
  .
```

VOID – Metadati Strutturali (4)

Una *class-based partition* è il sottoinsieme di un dataset che descrive le istanze di una classe particolare.

```
<#ExampleDataset> a void:Dataset ;  
  void:classPartition [  
    void:class foaf:Organization  
  ]  
.
```

Una *property-based partition* contiene soltanto le triple di un dataset che usano un certo predicato.

```
<#ExampleDataset> a void:Dataset ;  
  void:propertyPartition [  
    void:property foaf:homepage  
  ]  
.
```

VOID – Metadati Strutturali (5)

VOID definisce un certo numero di proprietà per rappresentare informazioni statistiche.

void:triples	The total number of triples contained in the dataset.
void:entities	The total number of entities that are described in the dataset. To be an entity in a dataset, a resource must have a URI, and the URI must match the dataset's void:uriRegexPattern, if any. Authors of VOID files may impose arbitrary additional requirements, for example, they may consider any foaf:Document resources as not being entities.
void:classes	The total number of distinct classes in the dataset. In other words, the number of distinct class URIs occurring as objects of rdf:type triples in the dataset.
void:properties	The total number of distinct properties in the dataset. In other words, the number of distinct property URIs that occur in the predicate position of triples in the dataset.

VOID – Metadati Strutturali (6)

VOID definisce un certo numero di proprietà per rappresentare informazioni statistiche.

void:distinctSubjects	The total number of distinct subjects in the dataset. In other words, the number of distinct URIs or blank nodes that occur in the subject position of triples in the dataset.
void:distinctObjects	The total number of distinct objects in the dataset. In other words, the number of distinct URIs, blank nodes, or literals that occur in the object position of triples in the dataset.
void:documents	If the dataset is published as a set of individual documents, such as RDF/XML documents or RDFa-annotated web pages, then this property indicates the total number of such documents. Non-RDF documents, such as web pages in HTML or images, are usually not included in this count. This property is intended for datasets where the total number of triples or entities is hard to determine.

VOID – Metadati Strutturali (7)

La maggior parte delle statistiche può essere calcolata automaticamente per mezzo di query SPARQL:

<http://code.google.com/p/void-impl/wiki/SPARQLQueriesForStatistics>

Le stesse proprietà possono essere usate per rappresentare informazioni statistiche circa *class/property-based partitions*, che sono un tipo di *void:Dataset*.

VOID – Descrivere Linkset (1)

Un *void:Linkset* è un *void:Dataset* contenente triple i cui soggetti ed oggetti sono descritti in dataset diversi.

```
<#ExampleDataset_AnotherExampleDataset> a void:Linkset ;
  void:target <#ExampleDataset> ;
  void:target <#AnotherExampleDataset>
  void:triples 1000
.
```

La proprietà *void:target* è ulteriormente specializzata in *void:subjectsTarget* e *void:objectsTarget* per indicare il dataset in cui sono definiti, rispettivamente, i soggetti e gli oggetti delle triple.

```
<#ExampleDataset_AnotherExampleDataset> a void:Linkset ;
  void:subjectsTarget <#ExampleDataset> ;
  void:objectsTarget <#AnotherExampleDataset>
  void:triples 1000
.
```

VOID – Descrivere Linkset (2)

Quando i link sono forniti da uno dei dataset «collegati», il linkset dovrebbe essere rappresentato come un subset del dataset in questione.

```
<#ExampleDataset> a void:Dataset;
  void:subset <#ExampleDataset_AnotherExampleDataset>
  .

<#ExampleDataset_AnotherExampleDataset> a void:Linkset ;
  void:target <#ExampleDataset> ;
  void:target <#AnotherExampleDataset> ;
  void:triples 1000
  .
```

La proprietà *void:linkPredicate* può essere usata per indicare la proprietà usata nel linkset

```
<#ExampleDataset_AnotherExampleDataset> a void:Linkset ;
  void:target <#ExampleDataset>, <#AnotherExampleDataset> ;
  void:triples 1000
  void:linkPredicate owl:sameAs
  .
```

CONSUMARE LINKED DATA

Dove cercare i dati? [non necessariamente LOD] (1)

Data Hub (<http://datahub.io/>)

Una *directory di dataset*; ha contribuito allo sviluppo della piattaforma dati CKAN

Linked Open Vocabularies (<http://lov.okfn.org/dataset/lov/>)

Una *rete di vocabolari* (es. Schemi RDF and Ontologie OWL)

Entity Name System

Un *Entity Name System* è una **authority** che assegna *identificatori canonici* alle risorse (specialmente agli *individui*) *descritti brevemente* in termini di *profili*. Poche informazioni circa la risorsa di interesse sono sufficiente per *consultare* l'ENS alla ricerca di un identificatore canonico.

Dove cercare i dati? *[non necessariamente LOD] (2)*

Portali dati governativi (spesso basati, o quantomeno compatibili con CKAN)

<https://data.gov> (US), <https://data.uk.gov> (UK), <https://www.dati.gov.it/> (IT),
<https://data.europa.eu/> (EU)

Istituti di statistica

<https://www.istat.it/>, <https://ec.europa.eu/eurostat>

Portali settoriali

<https://bioportal.bioontology.org/>

Consumare LD: Dereferenziazione

La descrizione RDF di una risorsa nella Linked Data cloud può essere ottenuta inviando una richiesta di tipo GET all'indirizzo HTTP della risorsa.

I Browser per i Linked Data (e.g. Tabulator, Disco, Marbles,...) fanno affidamento su questo approccio.

In aggiunta, questo meccanismo permette di rispondere a semplici interrogazioni formulate in linguaggi centrati sulle risorse ed orientate ai cammini (path) e.g. [LD Path](#) (Schaffert et al, 2012).

PRO	CONS
freschezza dell'informazione	latenza (per il consumer) carico sull'infrastruttura di chi pubblica i dati espressività limitata

Consumare LD: Federated SPARQL

Per supportare un accesso *efficiente*, un dataset dovrebbe fornire un **endpoint SPARQL** (pubblicizzato nella descrizione del dataset).

In SPARQL 1.1 è possibile interrogare più dataset in *maniera federata*¹.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?name
```

```
FROM <http://example.org/myfoaf.rdf>
```

```
WHERE
```

¹ <http://www.w3.org/TR/sparql11-federated-query/>

```
{
```

```
<http://example.org/myfoaf/!> foaf:knows ?person .
```

```
SERVICE <http://people.example.org/sparql> {
```

```
  ?person foaf:name ?name . }
```

```
}
```

L'esperienza maturata nel settore dei database (relazionali) federati ci dice che è *molto difficile ottimizzare* la valutazione delle query.

La maggiore **limitazione di SPARQL Federato** è la necessità di *decomporre esplicitamente la query* in frammenti che devono essere valutati da *dataset differenti* (spesso *menzionati esplicitamente nella query*)

Sfortunatamente, il paradigma dei Linked Data promuove l'interconnessione di dataset disparati cosicché risulta *molto difficile prevedere gli endpoint necessari*.

Un **approccio ibrido** (Hartig, Bizer & Johann-Christoph, 2009) permette di valutare una query SPARQL ordinaria usando delle *euristiche per determinare in modo incrementale i dati necessari* sulla base dell'ispezione della query e dei risultati parziali generati durante la valutazione.

Questo approccio consulta le risorse che man mano sono necessarie per rispondere ad una query SPARQL e, in generale, non può garantire la piena conformità alla semantica di SPARQL.

Consumare LD: Linked Data Fragments

Citando <http://linkeddatafragments.org/concept/>:

Un **Linked Data Fragment** (LDF) è caratterizzato da uno specifico **selettore** (*subject URI, SPARQL query, ...*), **metadati** (*nomi di variabili, conteggi, ...*), and **controlli** (*links o URIs verso altri fragment*).



Un **Triple Pattern Fragment** (Verborgh et al, 2014) corrisponde allr *triple selezionate da un triple pattern*

Un esempio è disponibile al seguente indirizzo:

<http://data.linkeddatafragments.org/dbpedia2014?subject=&predicate=rdf%3Atype&object=dbpedia-owl%3ARestaurant>

I *triple pattern fragment* cercano di trovare un equilibrio tra carico di lavoro sul client e sul server.

Consumare LD: Crawler (1)

Un **crawler LD** è un agente software (come i bot dei motori di ricerca) che *naviga nella Linked Data cloud per raccogliere informazioni*. [e.g. LDspider (Isele, Umbrich, Bizer & Harth, 2010)]

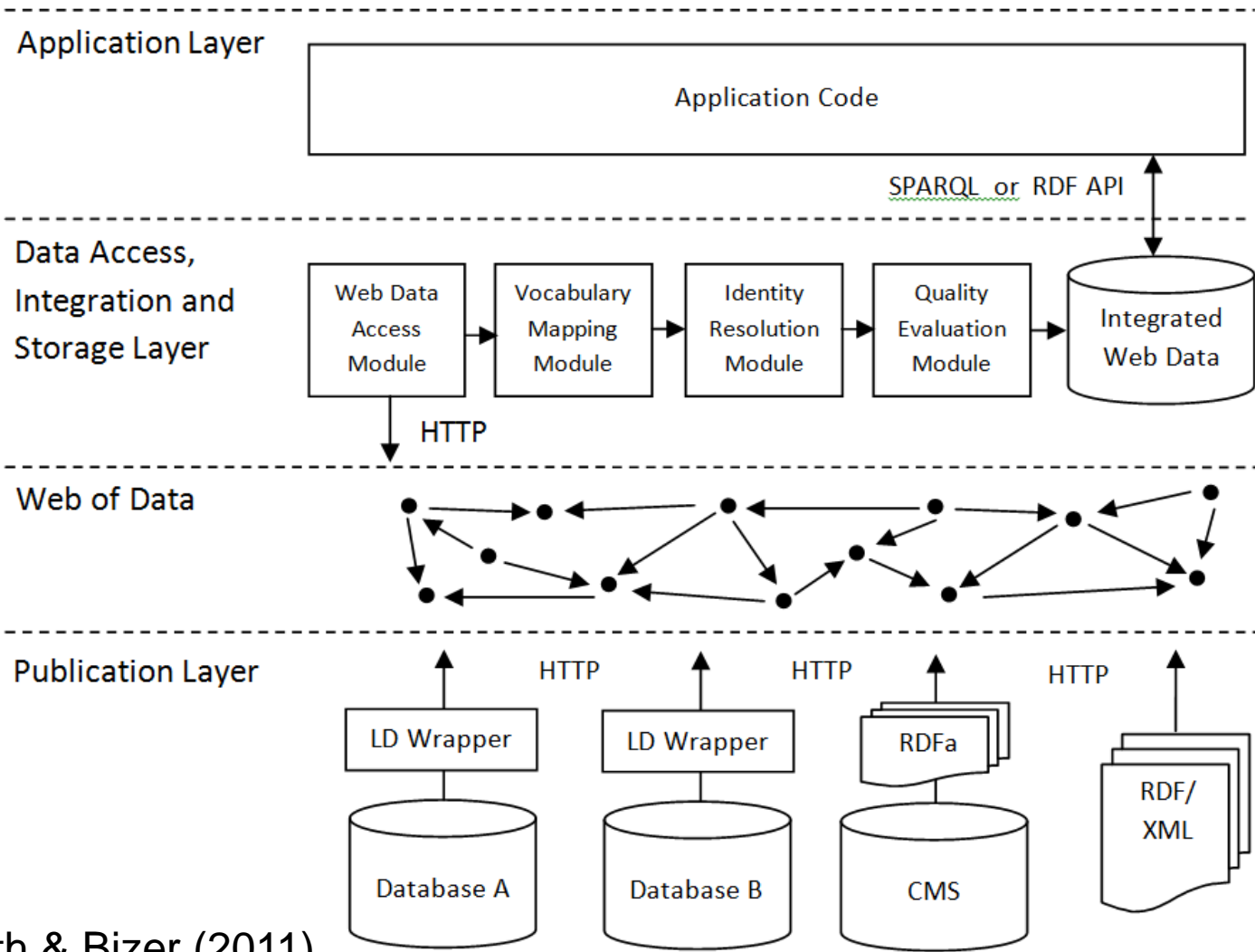
La *copia locale* può essere **elaborata** per

- schema mapping
- object consolidation
- quality management

Il risultato è una base di conoscenza locale che può essere gestita con *metodologie tradizionali*.

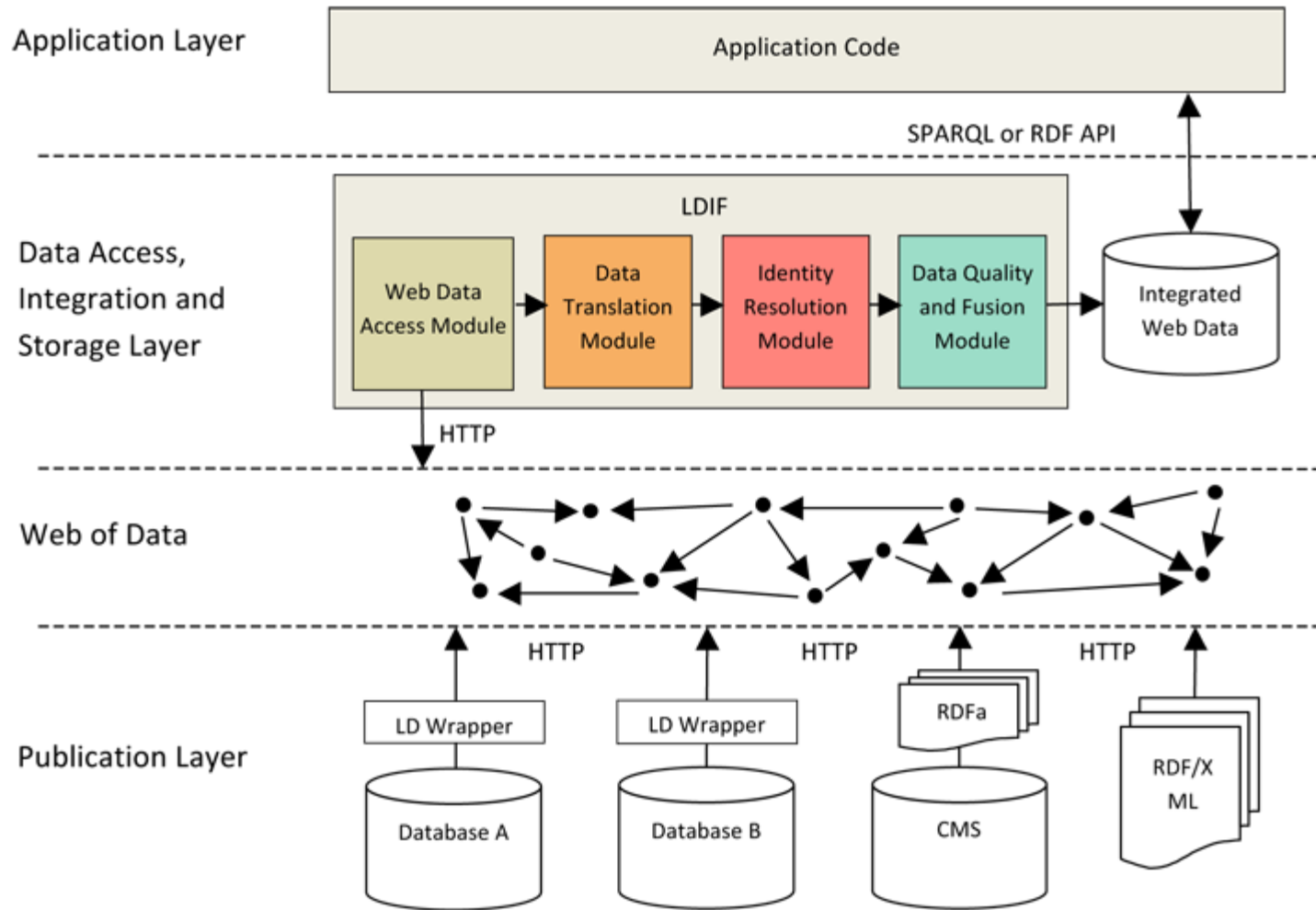
L'uso di una *copia cache di un frammento* della Linked Data cloud riduce il carico su chi pubblica i dati e permette di effettuare in modo effettivo ragionamenti complessi.

Consumare LD: Crawler (2)



Vedi Heath & Bizer (2011)

Consumare LD: Crawler (3)



Vedi Schultz et al (2012)

NOTE FINALI

Oltre la semantica RDF

La **semantica formale di RDF** assume che le *triple siano accurate*.

Chiaramente, questa *assunzione non vale* nel contesto dei Linked Data mentre si naviga tra *dataset disparati*.

Argomenti importanti:

- tracking *provenance*;
- judging *trustworthiness*;
- evaluating *data quality*.

Questi problemi sono **difficili da risolvere in generale** quando si ha a che fare con l'*intera Linked Data cloud*, mentre sono **semplici per applicazioni** che usano *pochi dataset (controllati)*.

Perché i Linked Data sono così rilevanti?

I grandi provider sono **felici** di fornire agli sviluppatori delle API per *invocare i propri servizi*, ma sono **timorosi** di *condividere i propri* (il cosiddetto "*database hugging*").

Al cuore dei Linked Data c'è un *movimento per la liberazione dei dati* nella propria **forma grezza** come presupposto per scenario d'uso innovativi.

Secondo <http://opendefinition.org/> **Open Data** è definito come segue:

Un pezzo di dati è aperto se chiunque è libero di usarlo, riusarlo, e ridistribuirlo — soggetto soltanto, al più, al requisito di attribuzione e/o condivisione nello stesso modo (share-alike)

Il movimento *Open Government* chiede alle pubbliche amministrazioni di aprire i propri database:

- che sono stati supportati dai contribuenti;
- per un'esigenza di trasparenza

data.gov (United States of America)

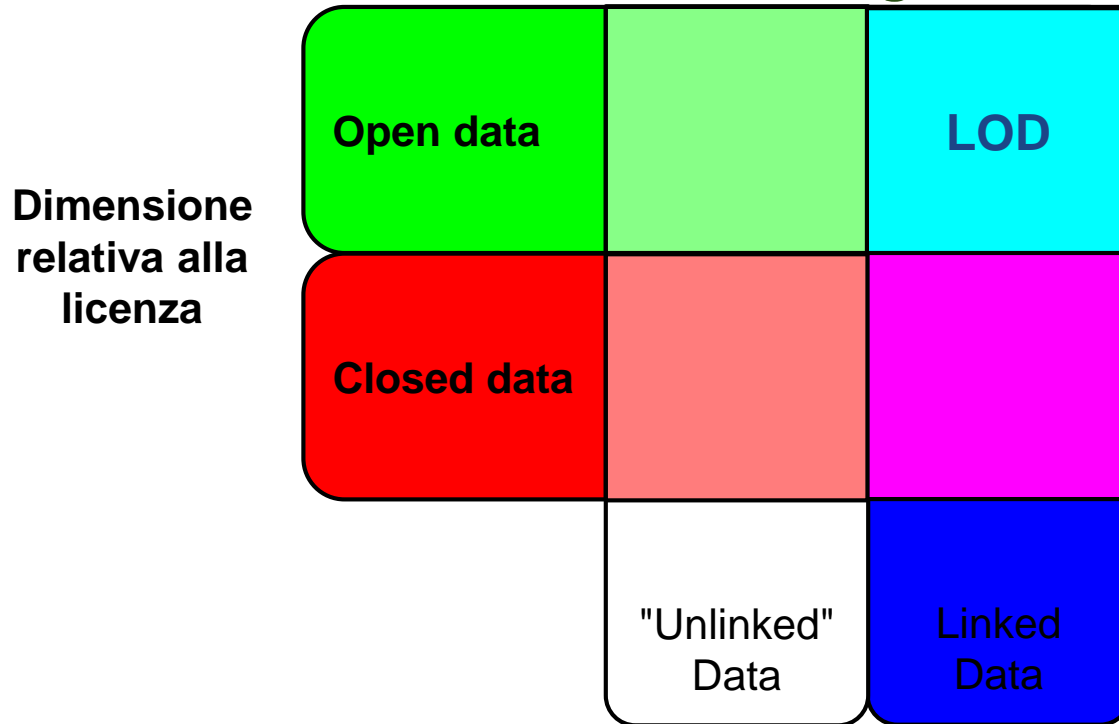
data.gov.uk (Regno Unito di Gran Bretagna e Irlanda del Nord)

dati.gov.it (Italia)

E molti altri

Open Data vs Linked Data

Open Data e Linked Data sono **concetti ortogonali**.



I Linked Data garantiscono i mezzi tecnici per il riuso degli Open Data

Dimensione relative alla pubblicazione

Gli Open Data garantiscono la massa critica per il successo del fenomeno

Linked Open Data (LOD) = Linked Data + Open Data

Tim Berners-Lee @ TED2009 su “the next Web”



http://www.ted.com/talks/tim_berniers_lee_on_the_next_web.html

Riferimenti (1)

1. Tim Berners-Kee and Mark Fischetti. *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*, 2000, HarperInformation.
2. Ian Jacobs and Norman Walsh. *Architecture of the World Wide Web, Volume One*, 15 December 2004. Available at <https://www.w3.org/TR/webarch/>
3. Tim Berners-Lee. *Design Issues: Linked Data*, 27 July 2006. Available at <https://www.w3.org/DesignIssues/LinkedData.html>
4. Talk at TED : "Tim Berners-Lee on the next Web", February 2009. Available at http://www.ted.com/talks/tim_berniers_lee_on_the_next_web.html
5. Antoine Isaac, and Ed Summers. *SKOS Simple Knowledge Organization System Primer*. August 2009. Available at <https://www.w3.org/TR/skos-primer/>
6. Alistair Miles, and Sean Bechhofer. *SKOS Simple Knowledge Organization System: Reference*. August 2009. Available at <https://www.w3.org/TR/skos-reference/>

Riferimenti (2)

7. Alistair Miles, and Sean Bechhofer. SKOS Simple Knowledge Organization System eXtension for Labels (SKOS-XL) Namespace Document - HTML Variant. August 2009. Available at <https://www.w3.org/TR/skos-reference/skos-xl.html>
8. Tom Heath and Christian Bizer (2011) *Linked Data: Evolving the Web into a Global Data Space* (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool. Available at <http://linkeddatabook.com/editions/1.0/>
9. Tom Heath. *Linked Data? Web of Data? Semantic Web? WTF*, March 2009. Available at <http://tomheath.com/blog/2009/03/linked-data-web-of-data-semantic-web-wtf/>
10. Sebastian Schaffert, Christoph Bauer, Thomas Kurz, Fabian Dorschel, Dietmar Glachs, Manuel Fernandez. *The linked media framework: integrating and interlinking enterprise media content and data*. Proceedings of 8th International Conference on Semantic Systems, I-SEMANTICS 2012, Graz, Austria, 5-7 September, 2012.

Riferimenti (3)

12. Olaf Hartig, Christian Bizer and Freytag Johann-Christoph. *Executing SPARQL Queries over the Web of Linked Data*. ISWC 2009, Springer Berlin Heidelberg, 2009.
13. Robert Isele, Jürgen Umbrich, Christian Bizer, and Andreas Harth. *LDSpider: An open-source crawling framework for the Web of Linked Data*. ISWC 2010. Proceedings of 9th International Semantic Web Conference (ISWC 2010) Posters and Demos, 2010.
14. Jeni Tennison, Richard Cyganiak, and Dave Reynolds Eds. *The RDF Data Cube Vocabulary*. April 2012. Available at <https://www.w3.org/TR/vocab-data-cube/>
15. Andreas Schultz, Andrea Matteini, Robert Isele, Pablo N Mendes, Christian Bizer and Christian Becker. *Ldif-a framework for large-scale linked data integration*. 21st International World Wide Web Conference (WWW 2012), Developers Track, Lyon, France

Riferimenti (4)

16. Ruben Verborgh, Olaf Hartig, Ben De Meester, Gerald Haesendonck, Laurens De Vocht, Miel Vander Sande, Richard Cyganiak, Pieter Colpaert, Erik Mannens and Rik Van de Walle. *Querying Datasets on the Web with High Availability*. The Semantic Web – ISWC 2014. ISWC 2014. Lecture Notes in Computer Science, vol 8796. Springer, Cham