Architectures for massive data management

Ioana Manolescu
INRIA Saclay & Ecole Polytechnique
ioana.manolescu@inria.fr
http://pages.saclay.inria.fr/ioana.manolescu/

M2 Data and Knowledge
Université de Paris Saclay

Who am I

- Senior researcher (DR) at INRIA Saclay
- Part-time professor at Ecole Polytechnique

- I work on **algorithms and systems** for efficient management of complex data
  - Complex data: complex structure (trees, graphs…) and/or complex semantics (meaning)
  - Data management: queries, updates
  - Efficiency: fast evaluation; low resource usage

- PhD: INRIA Rocquencourt, France, 2001
- Master: Ecole Normale Supérieure and U. Paris 6
- Undergraduate: Polytechnic U. Bucharest, Romania

What is your background in data management?

Please fill in the **questionnaire**

Course goal

1. Discuss the main characteristics (dimensions) of massive data management platforms
   – « Big Data »
2. Present the main classes of such systems, according to the above dimensions
3. Analyze advantage/disadvantage trade-offs
4. Introduce some open research issues

Architectures to be covered

- Distributed databases
- P2P systems
  - Structured
  - Unstructured
- Key-value stores
  - Redis, DynamoDB
- Data integration systems
  - Local-as-view, global-as-view
  - Mediator systems
  - Dataspaces
  - Data lakes
- MapReduce parallelism [covered in another course]
- [Spark covered in another course]
- Large distributed tables (Spanner, F1)
- Graph databases: Neo4J, a distributed graph system

Course organization

- **Instructors**:
  Ioana Manolescu (INRIA)
  Silviu Maniu (U. Paris Sud)


- **Location**: U. Paris Sud, PUIO
  - Lecture: mostly **B107**
  - Lab: mostly **D104**
  - **Reference information in Synapses**
- **Evaluation**: exam + lab work

Practical matters

- You will have access to the course material in PDF format
  - At least, we will send them on the list: mrdk2018@googlegroups.com
- Feel free to contact us
  - Ask questions during course days
  - Write (usable ☺) e-mail ioana.manolescu@inria.fr

Today's course plan

1. **Motivation**: Big Data
   - Characteristics
   - Applications
2. **From databases to architectures for Big Data management**
   - Databases: quick recall or crash course
   - What needs to change to handle Big Data?

MOTIVATION: BIG DATA

Defining Big Data: the V's

- Volume
  - Scale
- Velocity
  - Speed of producing and consuming the data
- Variety
  - Very different sources and data types
- Veracity
  - Is the data correct / certain / true?

Where does the data volume come from? (1)

- Human-produced data
  - Web content: Web pages, blogs, social networks, tweets…
  - Twitter: 7 Terabytes (1 tera = 10^{18}) per day
  - Facebook: 10 Terabytes per day

Where does the data volume come from? (1)

- Human-produced data
  - Web content: Web pages, blogs, social networks, tweets…
  - Twitter: 7 Terabytes (1 tera = 10^{18}) per day
  - Facebook: 10 Terabytes per day
- Machine-produced data
  - Log data from all kind of servers
  - Real world devices: payments, telecom, energy, weather, transportation, shipment…

    Sensors, including on (US) highways and trains

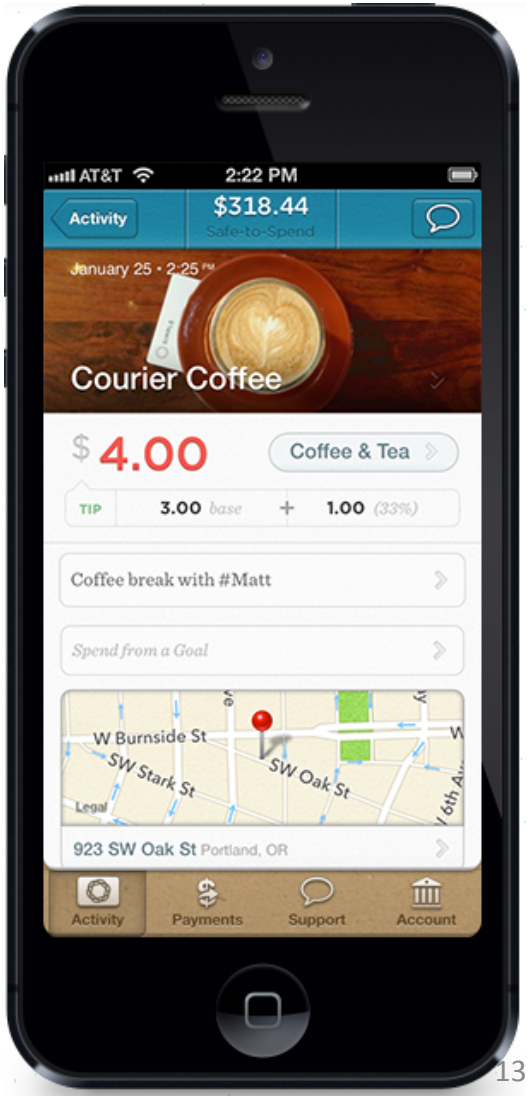
Gazpar (GDF)

Linky (EDF)

More data than we thought: bank transactions

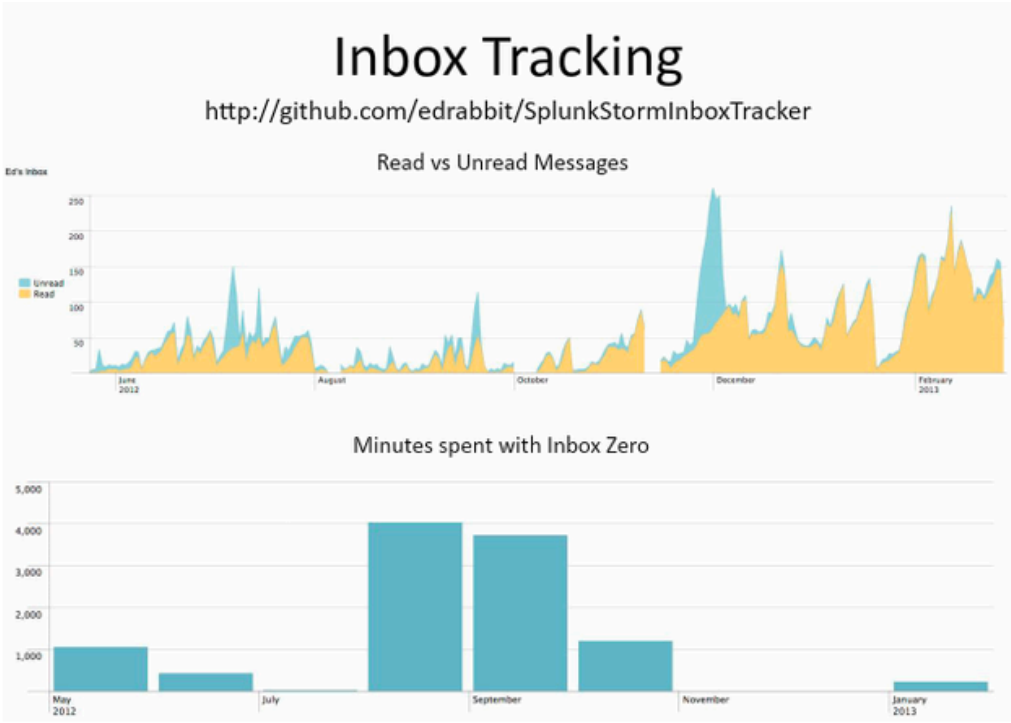
- The first historical database benchmark application
- (TPC benchmark: bank, branches, clients, accounts, tellers, transactions etc.)

- Simple.com: bare-bones financial services + access to transaction data
  - 80 attributes per transaction
  - Banks typically use only 4 ("Date/Libellé/ Euros")
  - Applications to exploit the complete transaction data



Human-produced data / Quantified Self

- "How big is your unread inbox?"
- "How did your heart rate change while exercising?"
- Social media content (including professional use), company intranets, photos etc.



Capturing and storing human user activity

- Gordon Bell (Microsoft), 2009
  http://edition.cnn.com/2009/TECH/09/25/total.recall.microsoft.bell/index.html?eref=rss_latest
- "video equipment, cameras and audio recorders to capture his conversations, commutes, trips and experiences. […] SenseCam that would hang around a person's neck and automatically capture every detail of life in photo form.

[…] saves everything -- from restaurant receipts (he takes pictures of them) to correspondence, bills and medical records. He makes PDF files out of every Web page he views.

[…] more than 350 gigabytes worth, not including the streaming audio and video -- is a replica of Bell's biological memory.
It's actually better, he says, because, if you back up your data in enough places, this digitized "e-memory" never forgets.

Gordon Bell, Microsoft, 2009

"It's like having a multimedia transcript of your life.

By about 2020 [...] our entire life histories will be online and searchable.

Location-aware smartphones and inexpensive digital memory storage in the "cloud" of the Internet make the transition possible and inevitable.

No one will have to fret about storing the details of their lives in their heads anymore. We'll have computers for that.

And this revolution will "*change what it means to be human*"

Where does the data volume come from? (2)

- Machine-produced data
  - Log data from all kind of servers
  - Real world devices: payments, telecom, energy, weather, transportation, shipment…

    Sensors, including on (US) highways and trains

Gazpar (GDF)

Linky (EDF)

Where does the data volume come from? (2)

- E.g. french railway system: surveillance trains for the normal and high-speed lines
- TGV specially equipped for measuring while circulating at 320 km/h:
  - rail geometry
  - train/rail interaction
  - rail signalization and communication devices
  - electric power availability etc.
  - 150 sensors, 20 cameras
  - complete tour of France's high speed network in 6 days

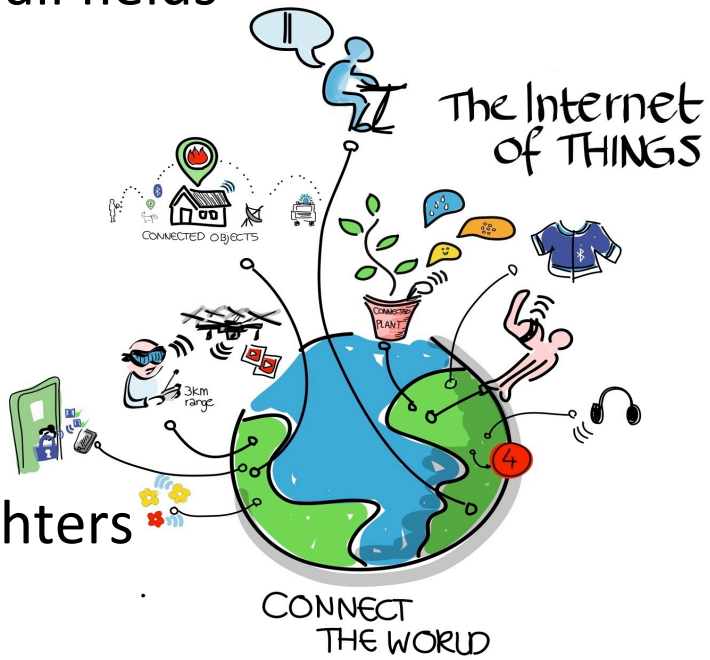
Machine-produced data: « Internet of Things » (IoT)

"The **Internet of Things** (**IoT**) is the interconnection of uniquely identifiable devices over the Internet

Expected to usher in **automation** in nearly all fields

Sample connected "things":

- Fridge → supermarket ☺
- heart monitoring implants
- biochip transponders on farm animals
  automobiles with built-in sensors
- field operation devices that assist firefighters



Already there:

- smart thermostat systems
- washer/dryers using wifi for remote monitoring.

Machine-produced data: « Internet of Things » (IoT)

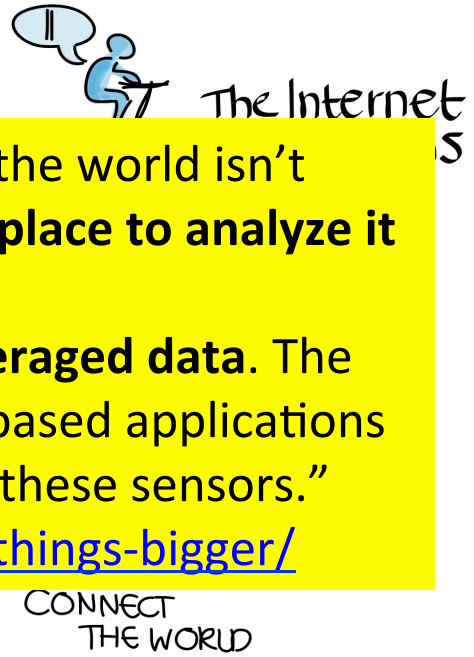
"The **Internet of Things** (**IoT**) is the interconnection of uniquely identifiable devices over the Internet

Expected to usher in **automation** in nearly all fields

Sample connected "things":

- Fridge
- heart
- biochi
  autor
- field o

"All the information gathered by all the sensors in the world isn't worth very much if there isn't **an infrastructure in place to analyze it in real time**.
Cloud-based applications are the key to using leveraged data. The Internet of Things doesn't function without cloud-based applications to interpret and transmit the data coming from all these sensors."
https://www.wired.com/2014/11/the-internet-of-things-bigger/



Already there:

- smart thermostat systems
- washer/dryers using wifi for remote monitoring.

Defining Big Data: the V's

- Volume
  - Scale
- Velocity
  - Speed of producing and consuming the data
- Variety
  - Very different sources and data types
- Veracity
  - Is the data correct / certain / true?

Big Data velocity

- How much data is produced e.g. per second
- Data enters a **pipeline** consisting of storage and/or processing
  - Store-then-process: for off-line data analysis. Storage by itself is a challenge sometimes, e.g. data links to/from clouds are rather slow
  - Process-then-store: for data whose interest is maximized upon arrival (real-time processing)
  - Process-then-discard: sensor/monitoring (if nothing happens)

Sample high-throughput data streams

- French IT company runs a data center of 2000 servers
- 5000 energy efficiency indicators (temperature, electricity consumption etc.) are measured every 20 seconds x 50 Kb per measure result = 170 Terabytes / year

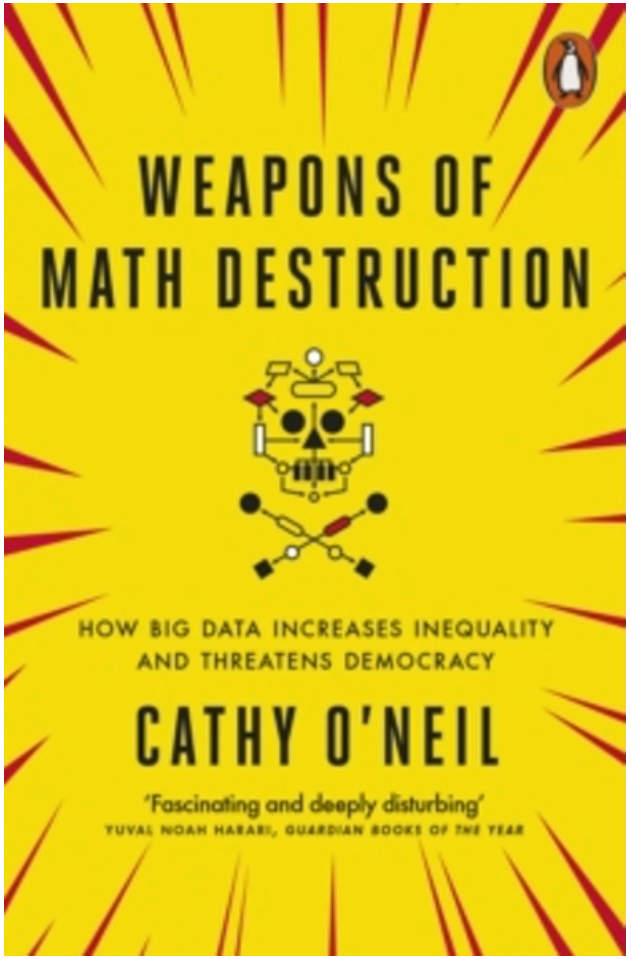
- Unable to store all data → sample (measure more rarely)
- May miss important things when they happen

The importance of current/recent data

- Real-time applications work only / mostly with **the latest data**
  - Embedded control mechanisms based on sensor data, e.g., "*this railway wagon component is breaking*" (now!)
  - Intrusion or malfunctioning detection…
- Keeping **humans engaged**
  - Customer relationship management *while the client is on the phone* with the customer service (see: “Ordering pizza in the future” video)
  - Recommending places where your friends are hanging out *now*

Important aspect of Big Data: ethics

- Current technologies for gathering and processing data raise **risks** wrt personal **freedoms** and **rights**
  - Discriminations, privacy violations, consumer rights, infringement on personal freedom (cf. geo-tagging), political manipulation (cf. focused FB ads)
  - Democracy and liberty may be at risk
  - Work starting on algorithm accountability (e.g., ParcoursSup)



Defining Big Data: the V's

- Volume
  - Scale
- Velocity
  - Speed of producing and consuming the data
- Variety
  - Different sources, data formats, data types
- Veracity
  - Is the data correct / certain / true?

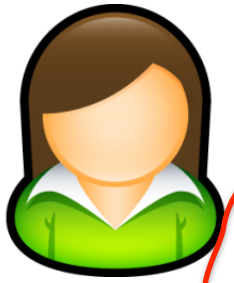
Big data heterogeneity (variety)

- Each new data type has added up on the old ones
- Enterprise data typically has high per-byte value ($/byte)
  - Hard to explain that "we will not need this database in the future"
- In many areas, legal obligation to keep old data (e.g. railway sensors, telecom, commercial…)

→ Data model & data management system soup

- hierarchical, relational, object-oriented, XML, RDF, JSON, key-value pairs…

Sample relational database



Clients

NumClient	Nom	Adresse	Ville	Age
1	Julie	1 rue Dugommier	Paris	22
…	…	…	…	…

Comptes

NumCompte	Type	Découvert	NumClient
12345	Courant	1000	1
…	…	…	…

Transaction

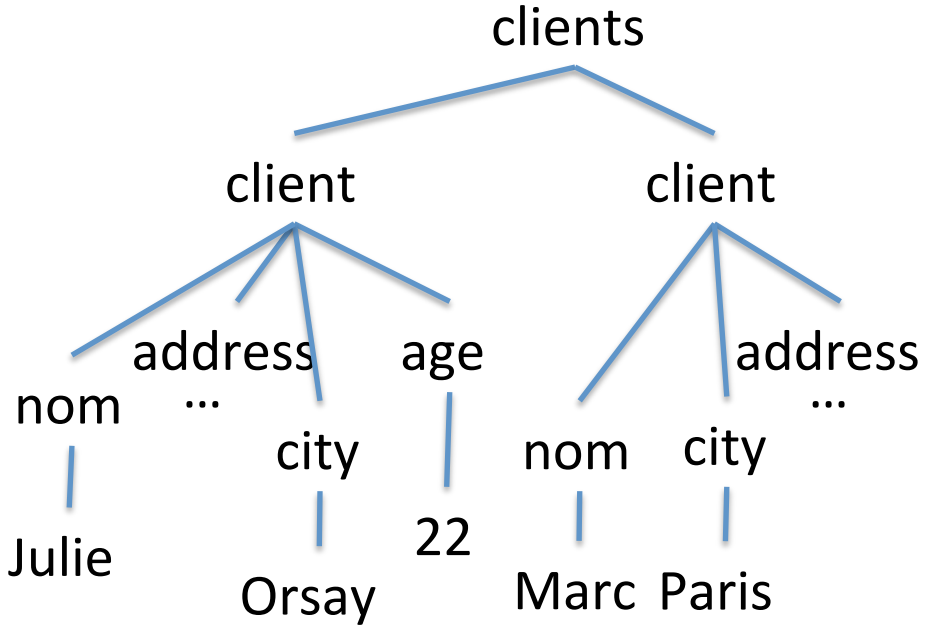
NumCompte	Montant	Date	Info
12345	-40,00	5/10/11	Retrait
12345	+23,45	6/10/11	Remb. MAIF
12345	-300,00	7/10/11	Chaussures

XML: extensible markup language

W3C, 2008

clients.xml:

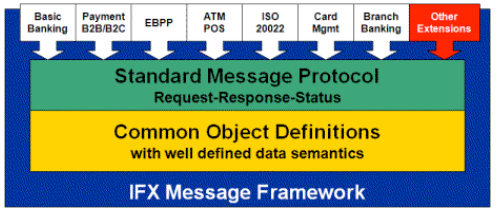
```
<clients>
 <client><nom>Julie</nom>
  <address>1, rue Dugommier</address>
  <city>Paris</city><age>22</age>
 </client>
 <client><nom>Marc</nom>…
 </client>
</clients>
```

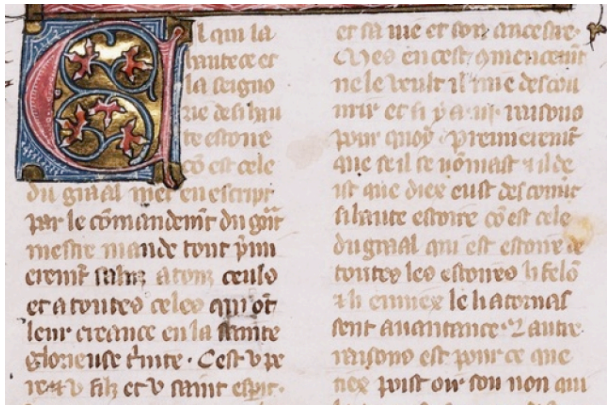


Flexible
Platform-independent
Separate content from presentation
Schema possible (not compulsory)

XML applications

- Main language for the Web: **XHTML, XML Schema, SVG, RSS,** …
- Web Services: **SOAP, WSDL, BP4WS**
- **MathML** (mathematical markup language)
- **CML** (chemical markup language)
- **SMILE** (synchronized multimedia integration language)
- Financial Exchange (**IFX**)
- The Text Encoding Initiative (**TEI**)

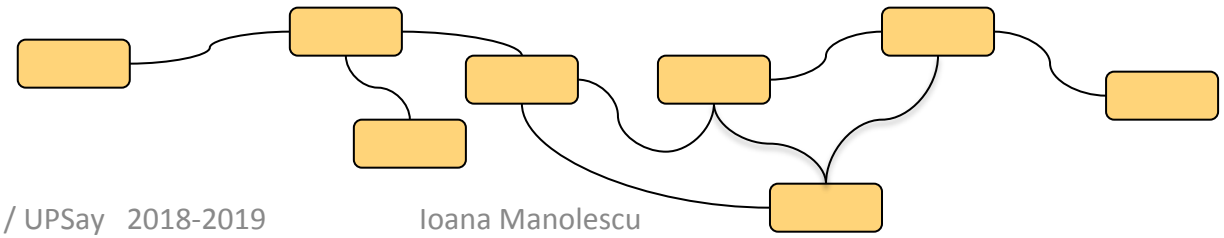




Critique of XML

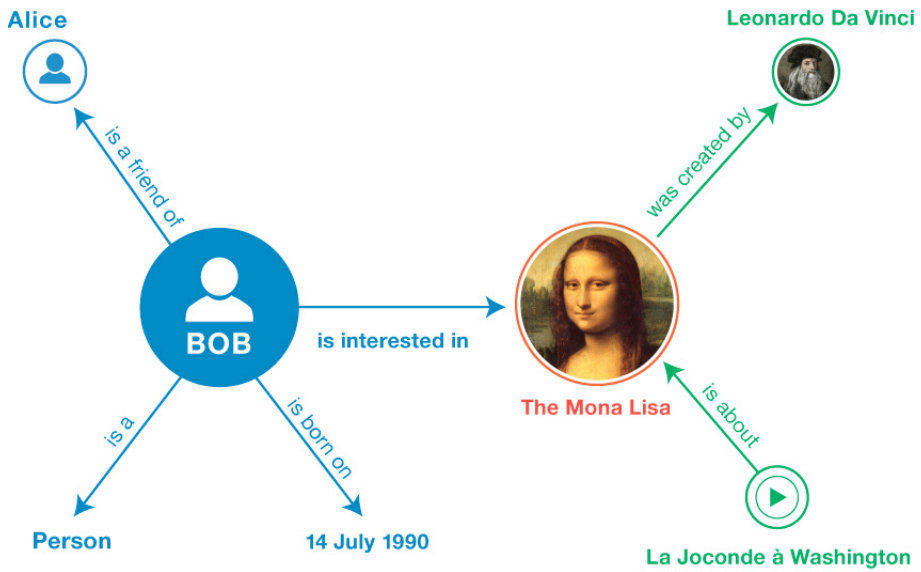
- Each information ends up in only one place
- OK for "classification" applications, structured text
- Fundamentally restrictive for **data = real world**!

Tim Berners-Lee, WWW proposal, CERN, 1998:
"*Many systems are organised hierarchically. A tree has the practical advantage of giving every node a unique name. However,* ***it does not allow the system to model the real world***."
(On newsgroups)*: "Typically, a discussion under one newsgroup will develop into a different topic, at which point* ***it ought to be in a different part of the tree***."



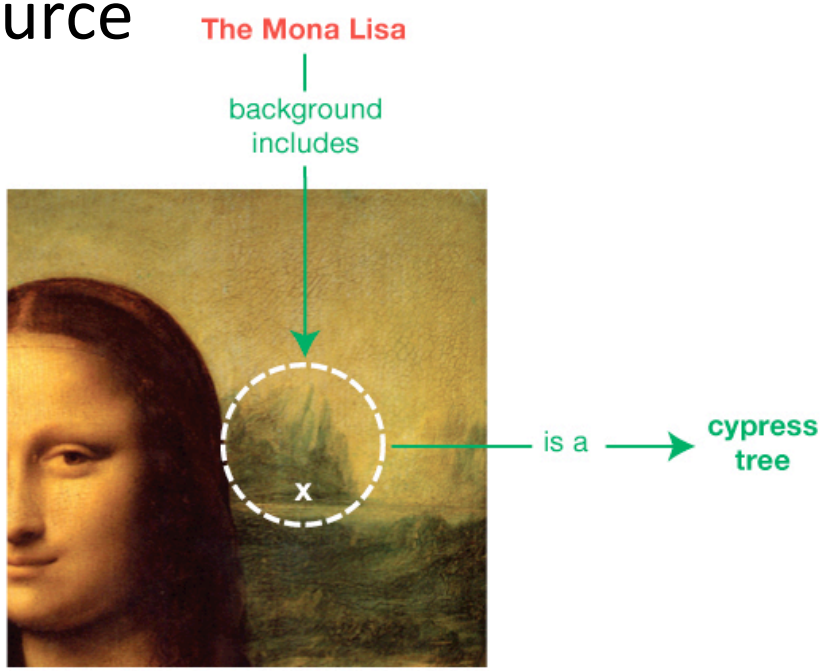
Graph data format for the Web: RDF (see also pre-requisite RDF course!)

- Resource Description Format, W3C, 2003
- Resources have properties with values.
- URIs (Universal Resource Identifiers) identify resources
- Resources, properties, or values may be specified by an URI.
- Properties and values may be constants



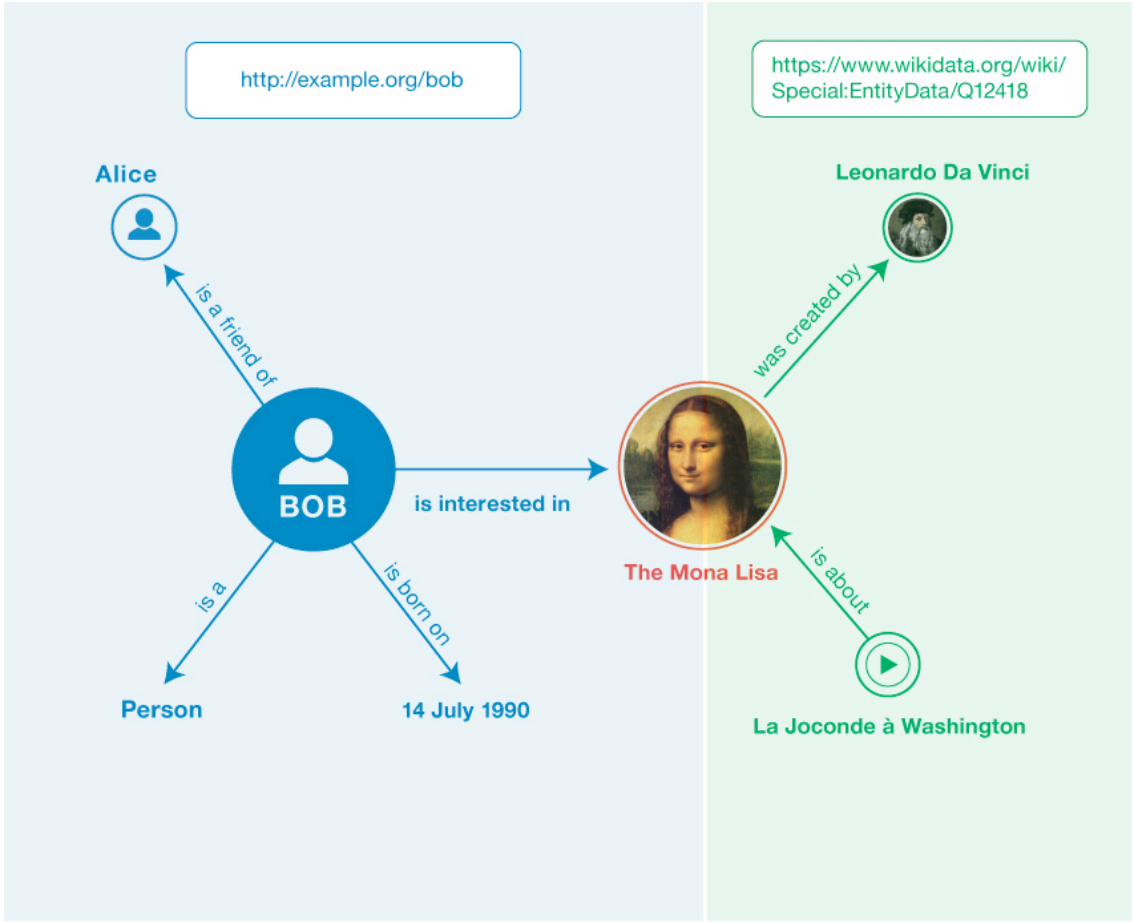
RDF feature: blank nodes

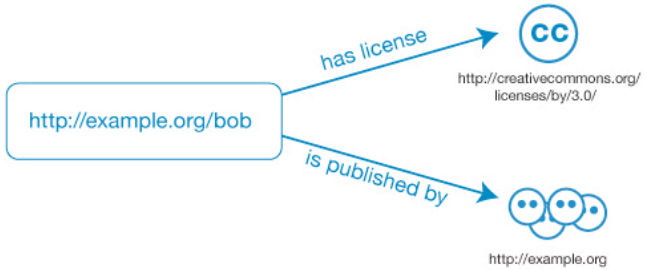
- Unnamed resource



- « Labeled null »

RDF graphs

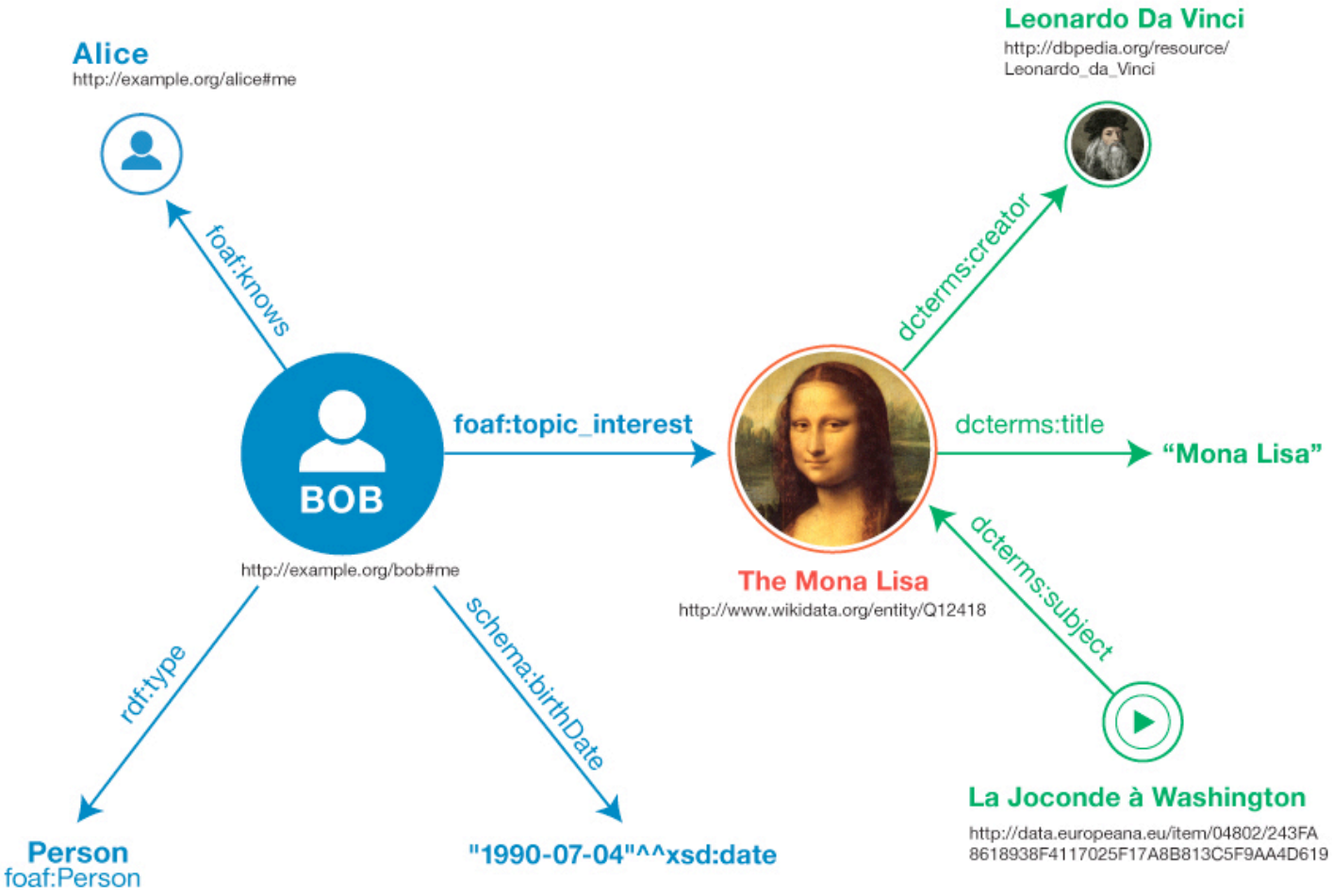

http://example.org/bob
https://www.wikidata.org/wiki/Special:EntityData/Q12418
Alice
Leonardo Da Vinci
is a friend of
was created by
BOB
is interested in
The Mona Lisa
is a
is born on
is about
Person
14 July 1990
La Joconde à Washington


has license
cc
http://creativecommons.org/licenses/by/3.0/
http://example.org/bob
is published by
http://example.org

RDF Schema constructs

Construct	Syntactic form	Description
Class (a class)	**C** rdf:type rdfs:Class	**C** (a resource) is an RDF class
Property (a class)	**P** rdf:type rdf:Property	**P** (a resource) is an RDF property
type (a property)	**I** rdf:type **C**	**I** (a resource) is an instance of **C** (a class)
subClassOf (a property)	**C1** rdfs:subClassOf **C2**	**C1** (a class) is a subclass of **C2** (a class)
subPropertyOf (a property)	**P1** rdfs:subPropertyOf **P2**	**P1** (a property) is a sub-property of **P2** (a property)
domain (a property)	**P** rdfs:domain **C**	domain of **P** (a property) is **C** (a class)
range (a property)	**P** rdfs:range **C**	range of **P** (a property) is **C** (a class)

Typed RDF graph

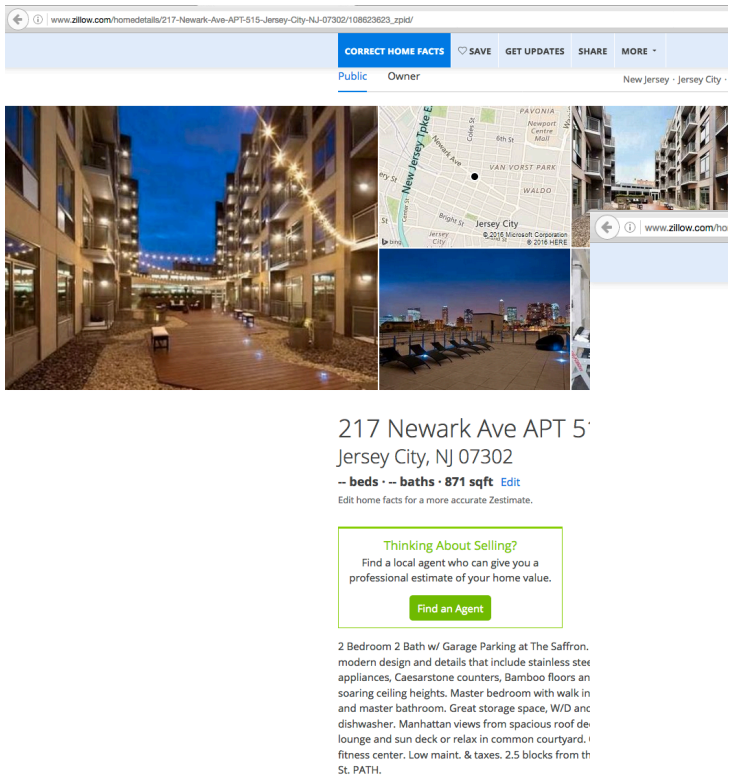


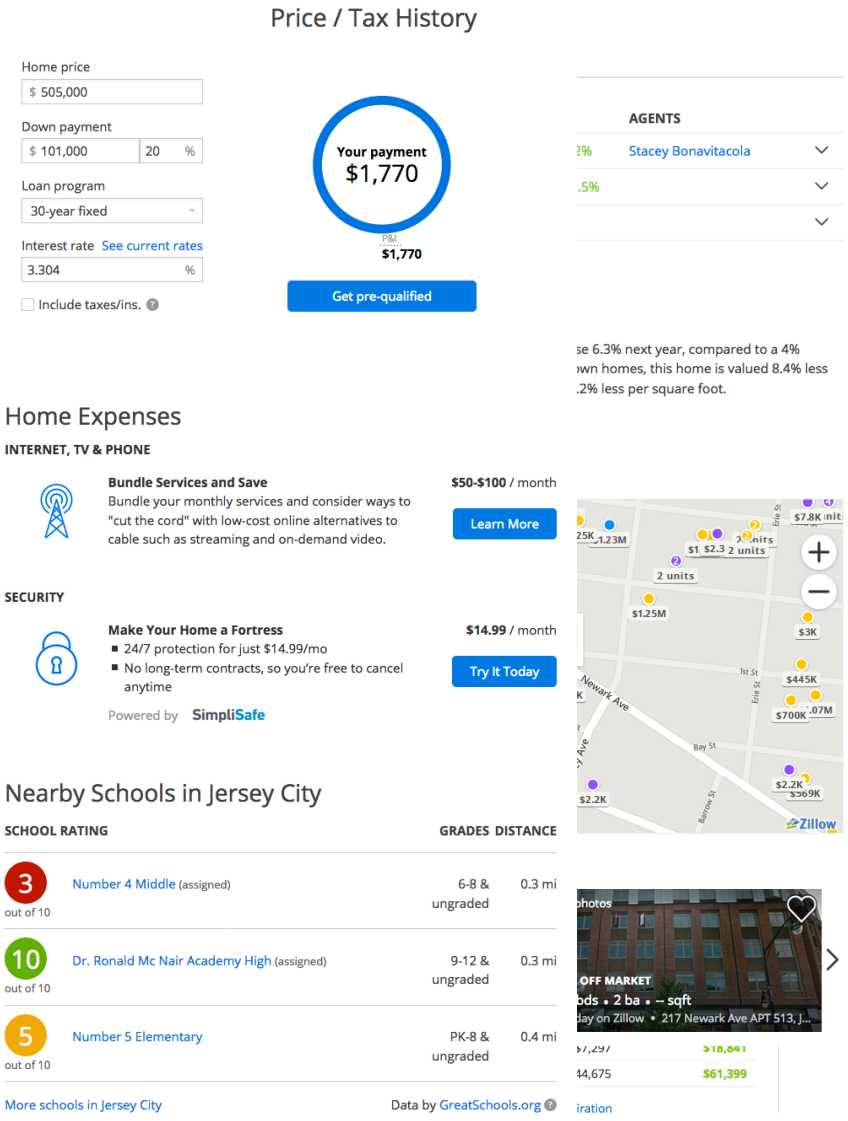
RDF reasoning

- RDF allow expressing **data *and* knowledge**
- Example:
  - if X *teaches a class*
  - then X *is a person*, *is an instructor*, *belongs to the school giving the class*, and *works for the university which includes the school*
- Reasoning exploits knowledge to infer *implicit data*

Varied Big Data has huge value potential

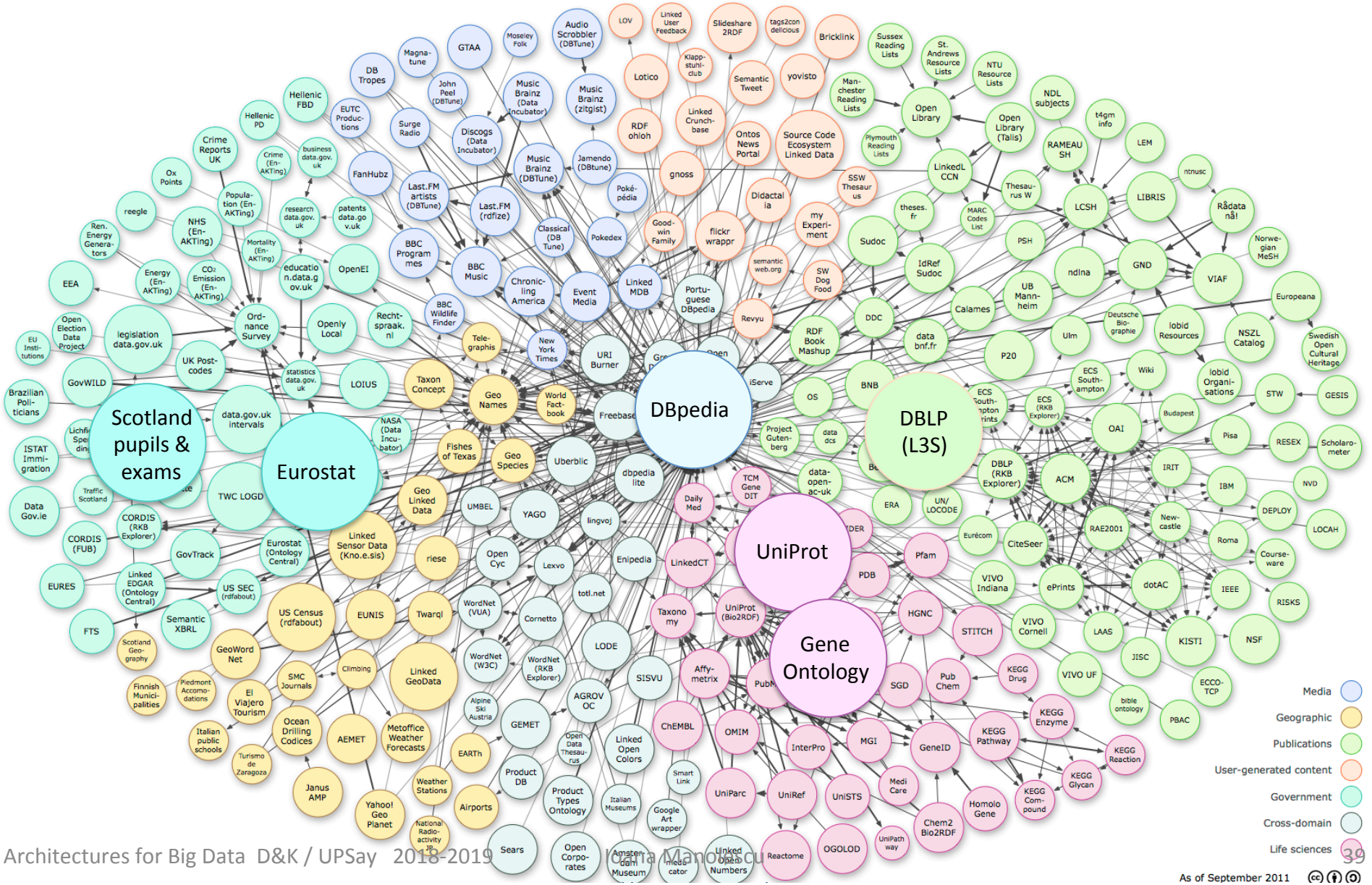
- Real estate ad from Zillow (US):



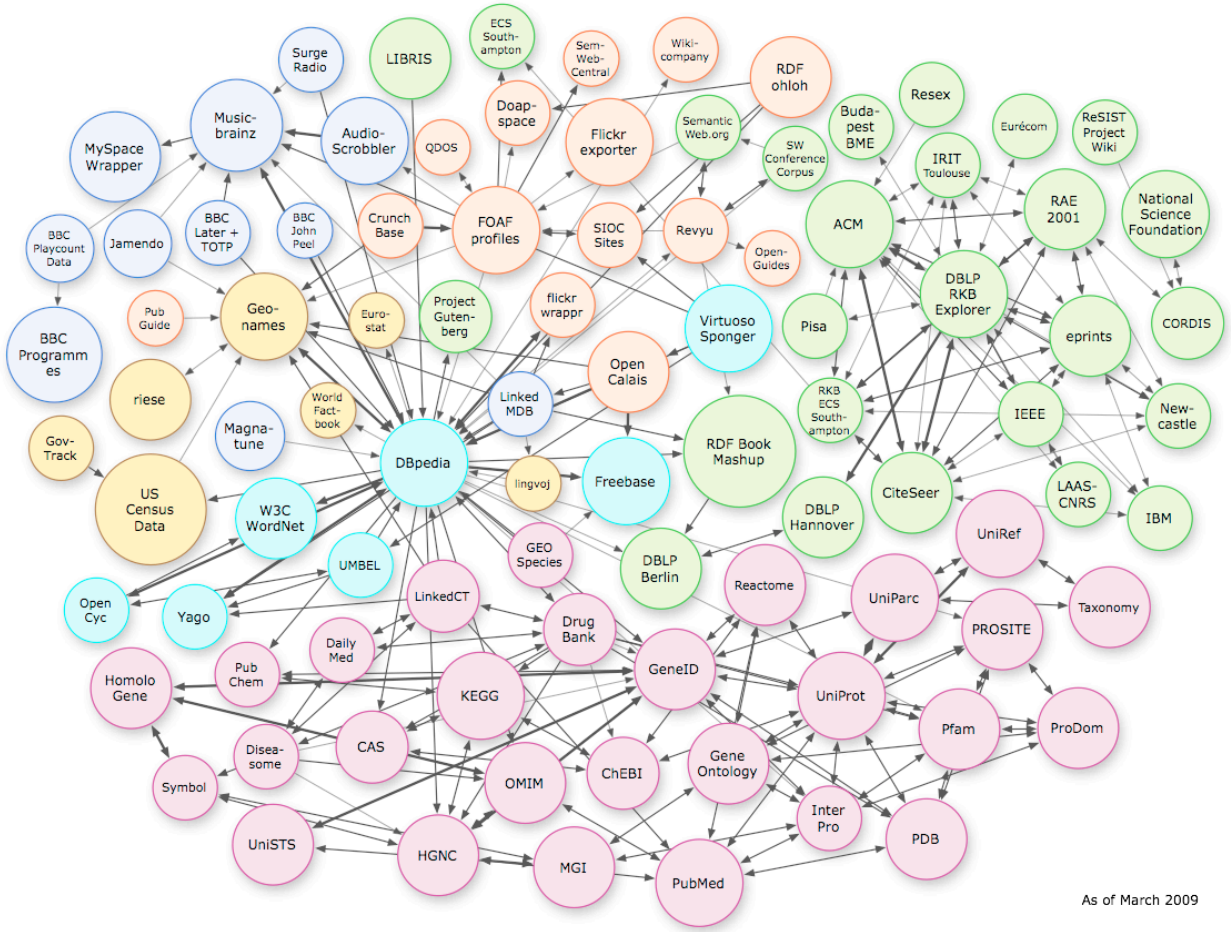


Linked Open Data (LOD) cloud

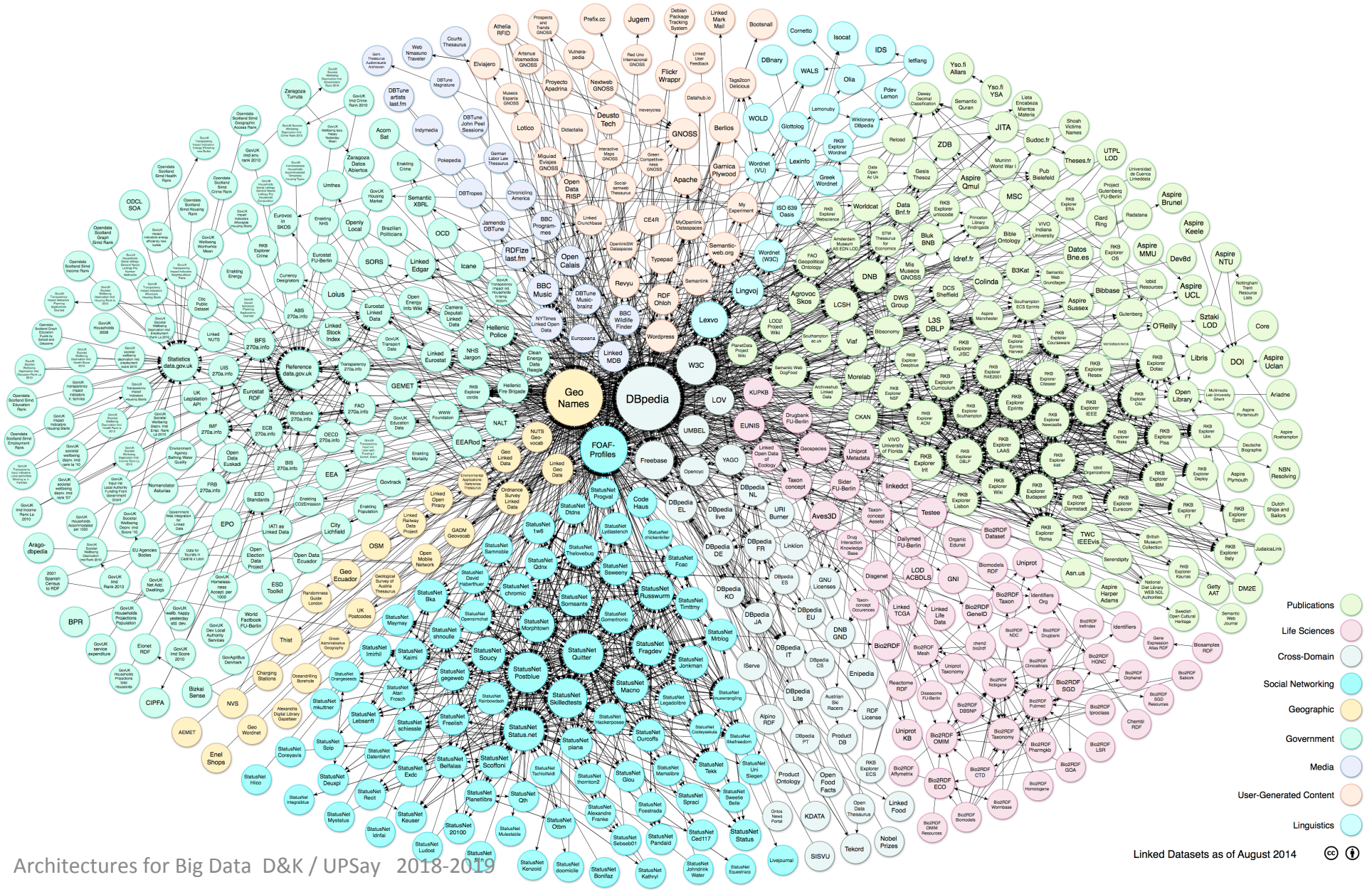
Linked Open Government Data project (logd.rw.rpi.edu): 10^{10} triples.



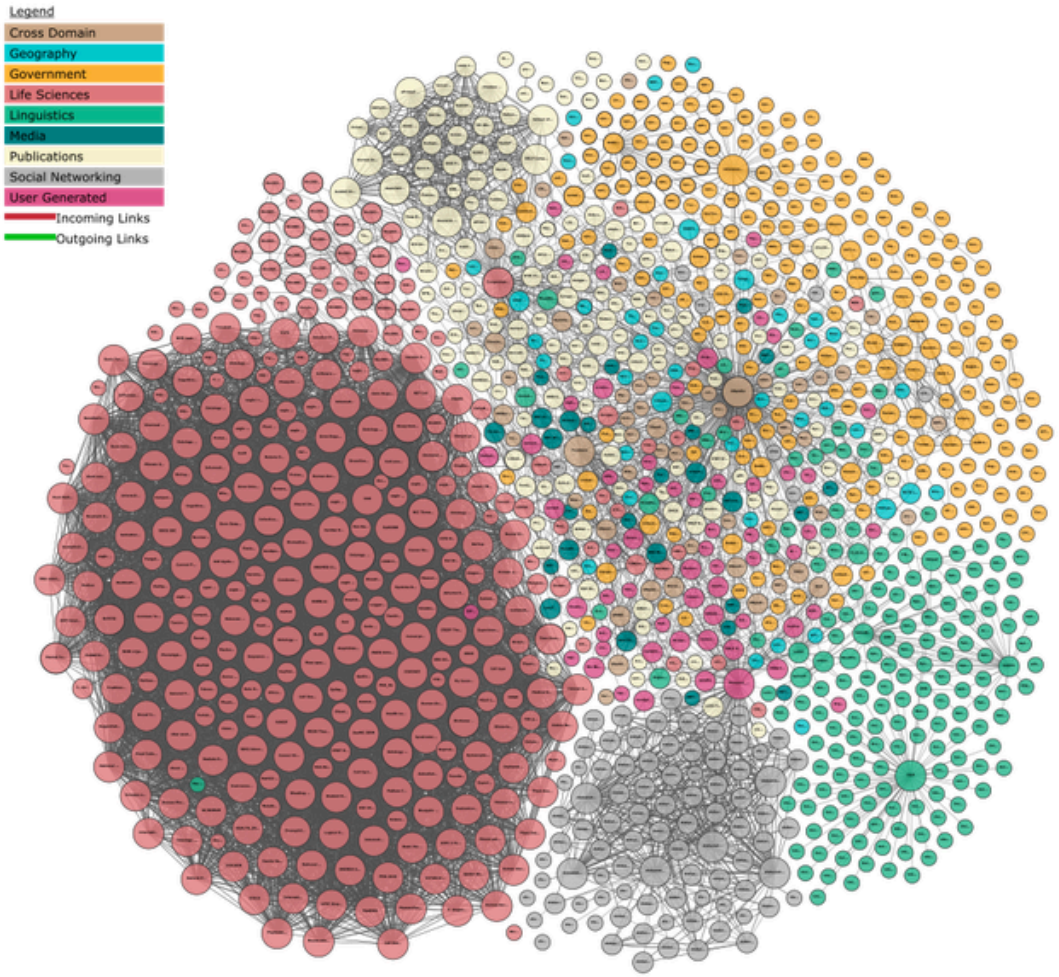
LOD cloud 2009



LOD cloud 2014



LOD cloud 2017



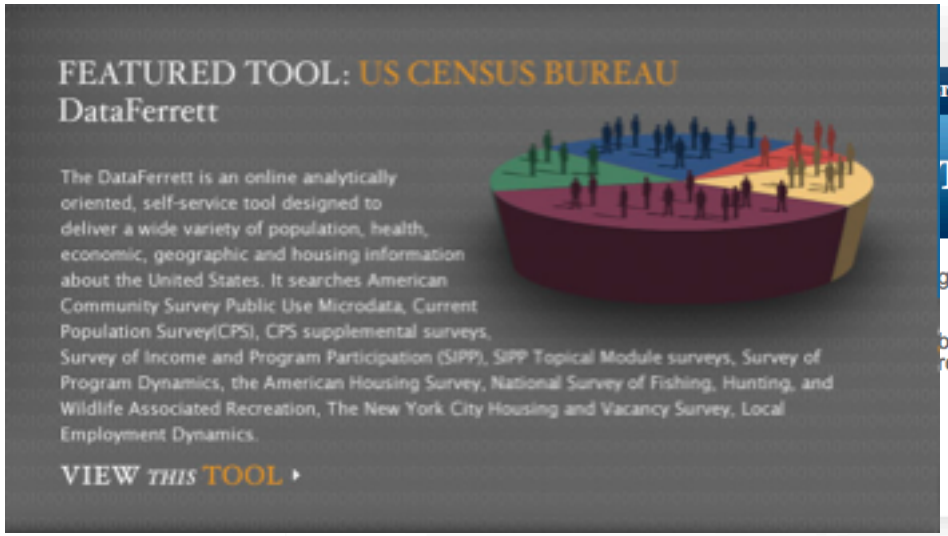
Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak.
http://lod-cloud.net/

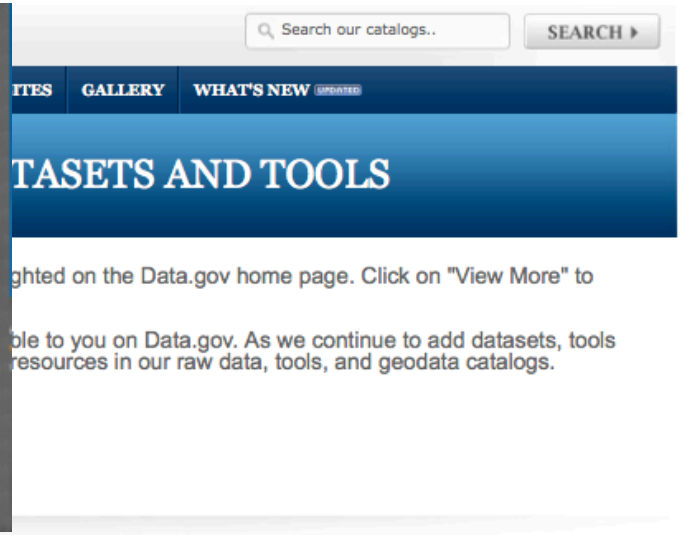
Open vs. linked data

1. **Linked Data**:
   "recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using **URIs** and **RDF**"
   - (Tim Berners-Lee) vision for the Web
2. **Open Data**:
   "idea that certain data should be freely available to everyone to use and republish as they wish, without restrictions from copyright, patents or other mechanisms of control"
   - In principle, orthogonal to the Linked aspect
   - In practice, Linked is a technical mean toward Open

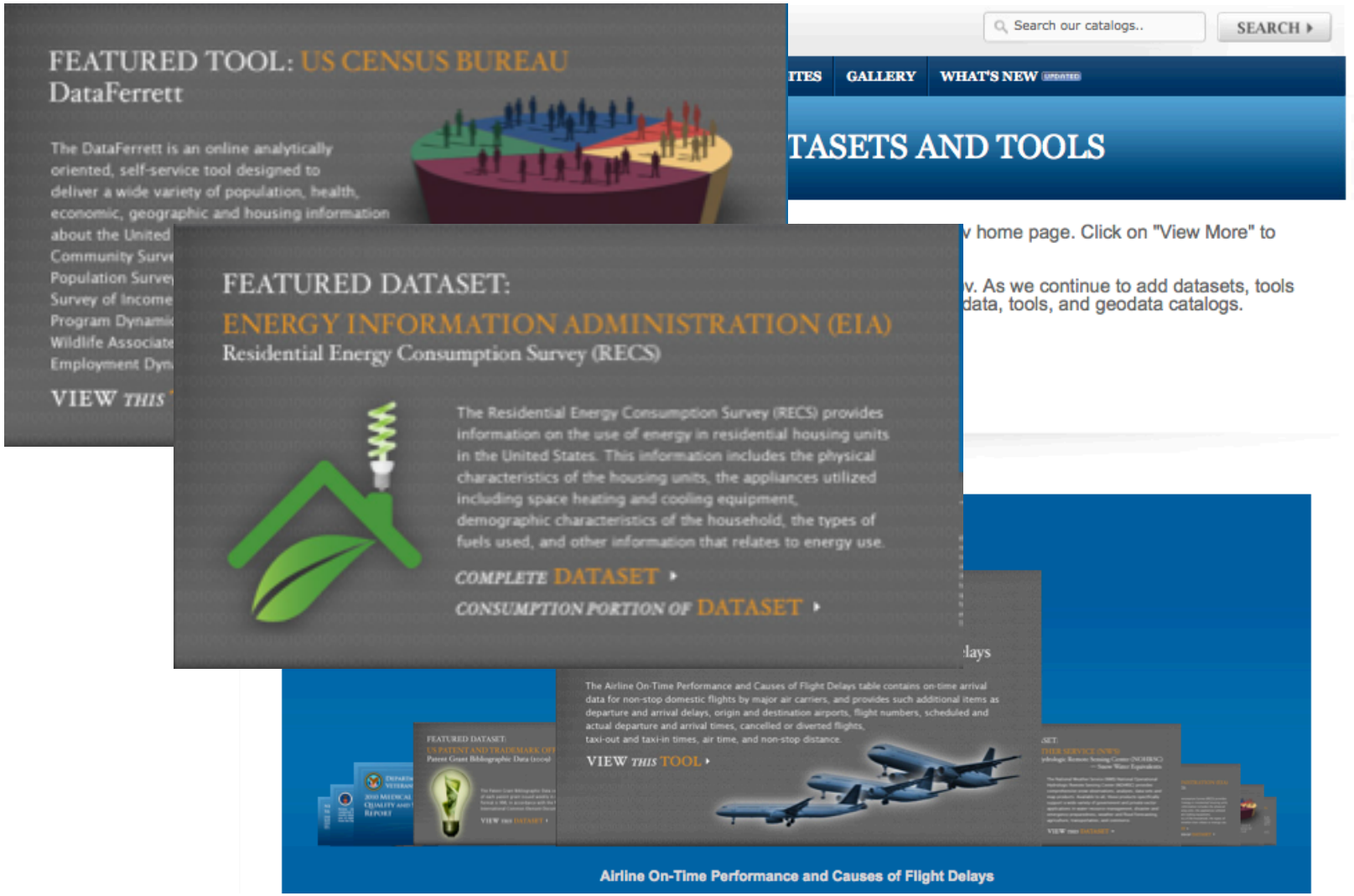
Open Data: data.gov (US)

Open Data: data.gov (US)





Open Data: data.gov (US)

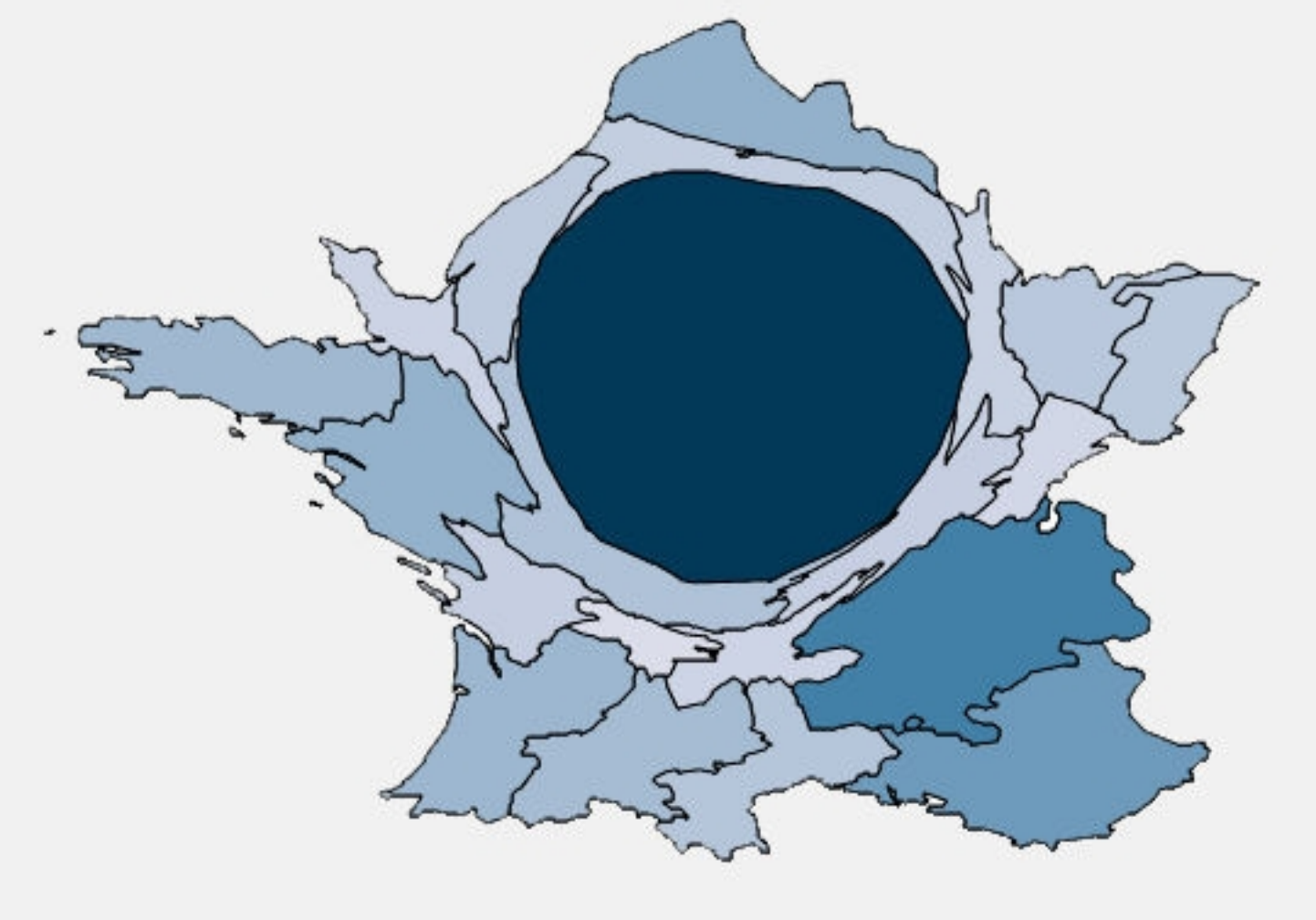


Open Data: data.gov (US)

Open Data: data.gov (US)

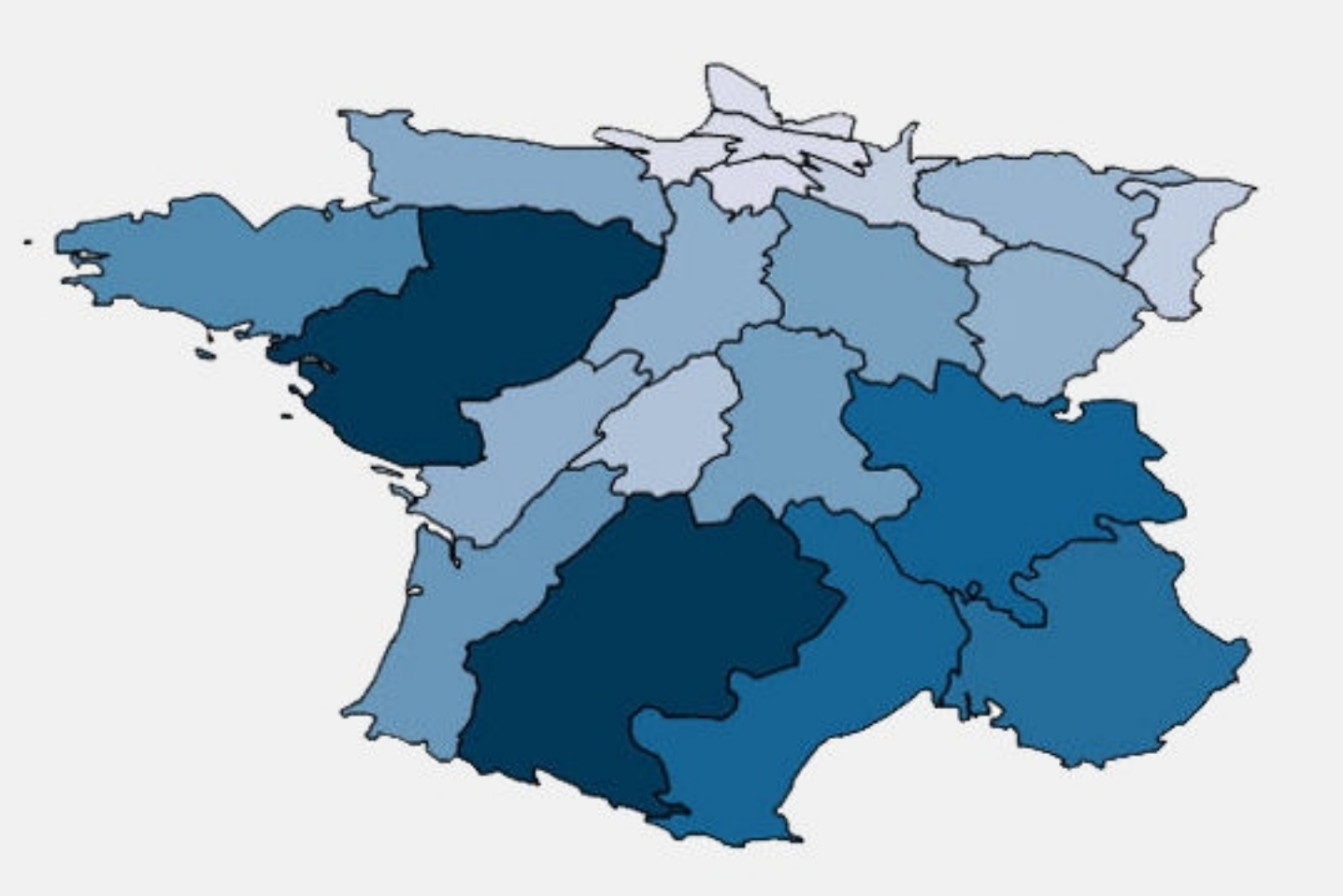
Open Data from Etalab (FR)

GDP per French region (Le Journal Du Net)



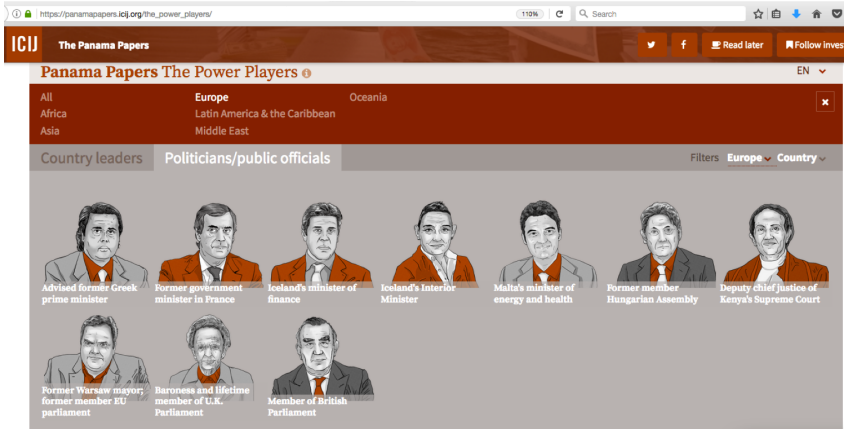
Open Data from Etalab (FR)

Organic agriculture per French region



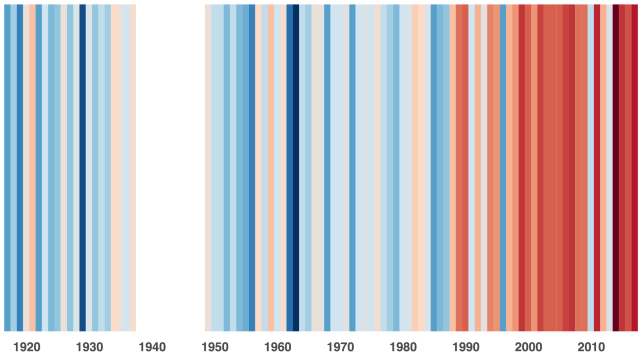
Open Data important for democracy

- Journalists and/or NGO workers play increasingly important role explaining society functioning
  - E.g., ICIJ Panama Papers investigation

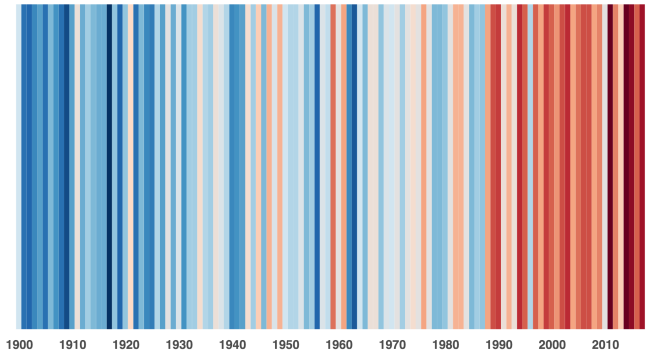


Open Data important for democracy

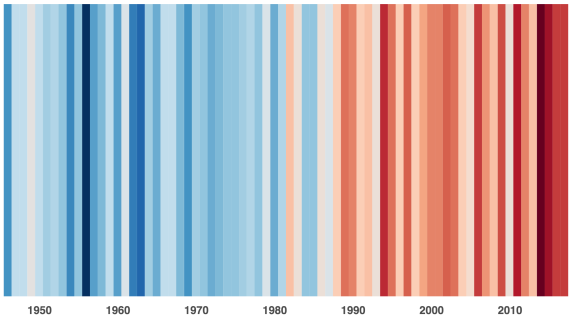
- Journalists and/or NGO workers play increasingly important role explaining society functioning
  - E.g., Météo France data on average yearly temperature, Les Décodeurs (Le Monde)



Evolution de la température par année depuis 1917
Dunkerque (59) : de 8,9 °C à 12,8 °C



Evolution de la température par année depuis 1900
Paris (75) : de 10 °C à 13,7 °C



Evolution de la température par année depuis 1946
Montpellier (34) : de 12,8 °C à 16,5 °C

JavaScript Object Notation (JSON)

Human-readable XML

1. Object = set of (attribute, value) pairs
2. Array = list of values.
3. Value = string | number | true | false | null | object | Array

```
{"menu": {
  "header": "SVG Viewer",
  "items": [
    {"id": "Open"},
    {"id": "OpenNew", "label": "Open New"},
    {"id": "ZoomIn", "label": "Zoom In"},
    {"id": "ZoomOut", "label": "Zoom Out"},
    {"id": "OriginalView", "label": "Original View"},
    null,
    {"id": "Quality"},
    {"id": "Pause"},
    {"id": "Mute"},
    {"id": "Help"},
    {"id": "About", "label": "About CVG Viewer..."}
  ]
}}
```

JavaScript Object Notation

- Among the most popular data interchange formats today
- There exist JSON notations for other data formats, e.g., RDF

```
{
  "http://example.org/about" : {
      "http://purl.org/dc/terms/creator" : [ { "value" : "_:anna",
                                               "type" : "bnode" } ] ,
  "_:anna" : {
      "http://xmlns.com/foaf/0.1/name" : [ { "value" : "Anna",
                                             "type" : "literal" } ]
  }
}
```

- Lab will cover MongoDB, most popular JSON database

Big data heterogeneity (variety)

Hierarchical, relational, object-oriented, XML, RDF, JSON, key-value pairs…

Data model & data management system soup

- hierarchical, relational, object-oriented, XML, RDF, JSON, key-value pairs…

Traditionally this has been solved (time and $ permitting) with **data migration** / **ETL** (extract-transform-load)

- Heterogeneous data and high throughput may make ETL impractical for Big Data →
  Need to **exploit big, heterogeneous data as is**

Defining Big Data: the V's

- Volume
  - Scale
- Velocity
  - Speed of producing and consuming the data
- Variety
  - Very different sources and data types
- Veracity
  - Is the data correct / certain / true?

Big Data veracity

- Is this true? (What is the probability?)
- Contradictory sources (1 vs. 2 clocks)
- Errors in the data
  - Humans introduce many errors
  - Sensors may have failures or erroneous readings
    - Light or heating sensors in a building
    - Wear and tear
- Tackled by **data curation / cleaning / quality** tools for regular (or at least homogeneous) data

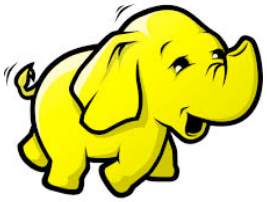
Big Data veracity

- Data reconciliation / entity extraction
- Large-scale ontologies such as YAGO, DBPedia, Google Knowledge Base (> Freebase)
  - Reference database of core facts

    - Places (city/country etc.), people (public figures, scientists, artists etc.), events (born, died, emigrated, was created…), time
  - Ontologies automatically extracted from Web and other specific sources → need for reconciliation

Big picture on big data

1. **Volume, velocity, variety, veracity**
2. Probably not all the data has the same **value** $/B
   - This is why existing databases will stay!
3. Very large, unstructured, uncertain-value data may fit in scalable, relatively slow systems (e.g. MapReduce/Hadoop)
   - Massive parallelism for dummies (to be seen)
   - Mining, batch, iterative processing



4. Large SW companies develop "database-style" layers on Map/Reduce, sometimes in Open Source
5. Many ways to distribute and parallelize the work; many data models