Knowledge discovery in the Semantic Web

Danai Symeonidou

INRA Montpellier

February, 22th 2017

The Web today

Google

Marion Cotillard

Web Images News Videos Maps More Search tools

About 10,100,000 results (0.83 seconds)

In the news

Cannes: Xavier Dolan Drama With Marion Cotillard Snags Germany, Australia, Russia Deals
Hollywood Reporter - 1 day ago
Seville International has sold Xavier Dolan's next French-language film, Juste La Fin Du ...

A Look Back at French It Girls Through the Ages, From Marie Antoinette to Marion Cotillard
E! Online - 2 days ago

Cannes: Justin Kurzel on What Makes 'Macbeth' a Western (Q&A)
Hollywood Reporter - 2 days ago

More news for marion cotillard

Marion Cotillard - IMDb
www.imdb.com/name/nm0182839/
Academy Award-winning Actress **Marion Cotillard** was born on September 30, 1975 in Paris. Cotillard is the daughter of Jean-Claude Cotillard, an actor, ...

Marion Cotillard - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Marion_Cotillard**
Marion Cotillard (French pronunciation: [ma.ʁjɔ̃ kɔ.ti.jaʁ]; born 30 September 1975) is a French actress and singer-songwriter. She is also an ...
Guillaume Canet - La Vie en rose (film) - List of awards and ... - Rust and Bone
You've visited this page 2 times. Last visit: 5/20/15

Marion Cotillard — Wikipédia
fr.wikipedia.org/wiki/**Marion_Cotillard** Translate this page
Marion Cotillard à l'avant-première parisienne de Public Enemies en 2009. Données clés. Surnom, The French Mermaid (« La sirène française »), Simone.

Marion Cotillard

Film actress

Marion Cotillard is a French actress and singer-songwriter. She is also an environmentalist and spokesperson for Greenpeace. Wikipedia

Born: September 30, 1975 (age 39), Paris
Height: 1.69 m
Partner: Guillaume Canet (2007–)
Children: Marcel Canet
Awards: Academy Award for Best Actress, more

Movies

View 35+ more


Two Days, One Night
2014


The Dark Knight Rises


Rust and Bone
2012


La Vie en rose
2007

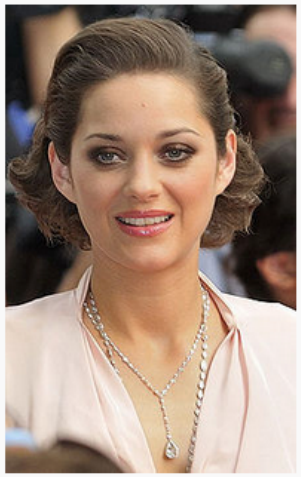

Inception
2010

The Web today

- Different sources containing common or complementary information

Wikipedia.com

Marion Cotillard



Marion Cotillard à l'avant-première parisienne de *Public Enemies* en 2009.

Surnom	*The French Mermaid* (« La sirène française »), Simone
Naissance	30 septembre 1975 (39 ans) Paris, France
Nationalité	Française
Profession	Actrice
Films notables	*Taxi* (saga) *Un long dimanche de fiançailles* *La Môme* *Inception* *De rouille et d'os* *The Dark Knight Rises* *Deux jours, une nuit*

Wikipedia.fr

Marion Cotillard

Ordre des Arts et des Lettres

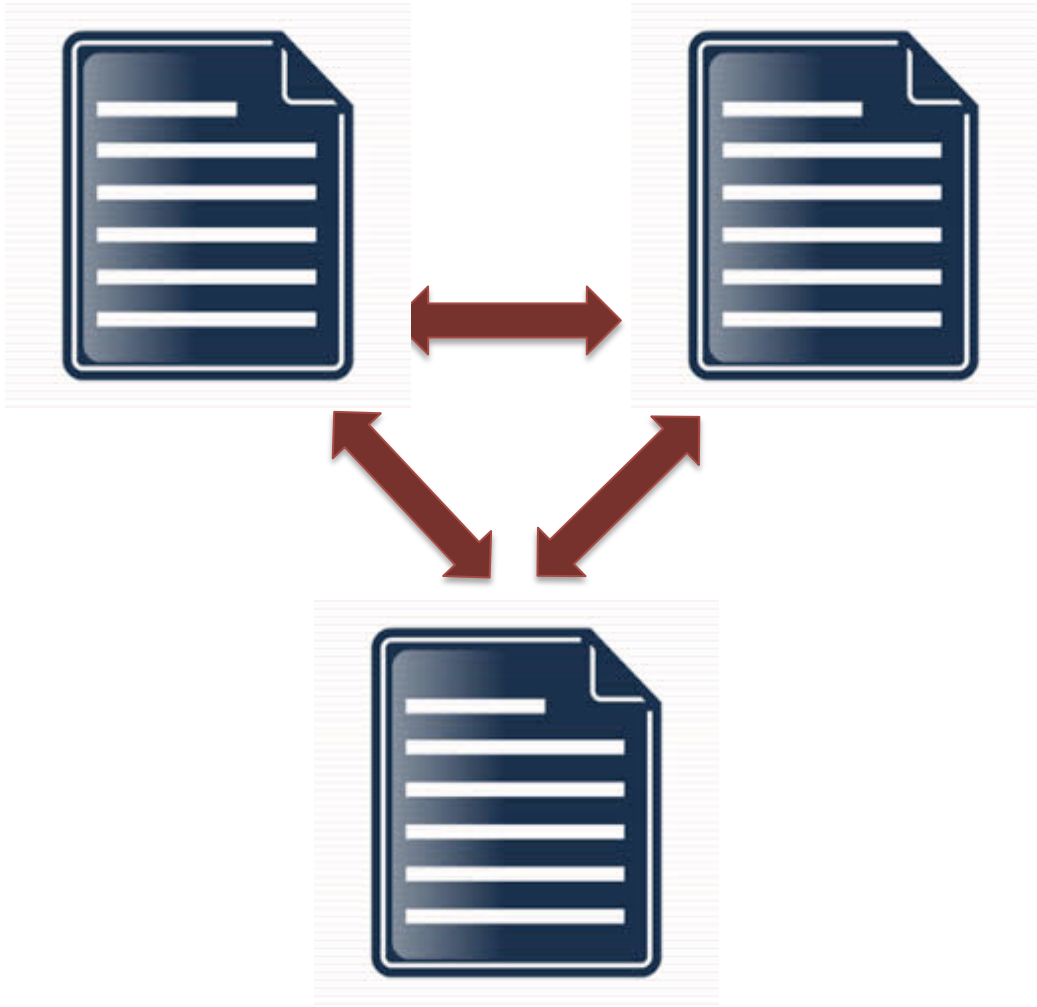
Cotillard at the 2012 Cannes Film Festival.

Born	30 September 1975 (age 39) Paris, France
Occupation	Actress, singer
Years active	1993–present
Partner(s)	Guillaume Canet (2007–present)
Children	1

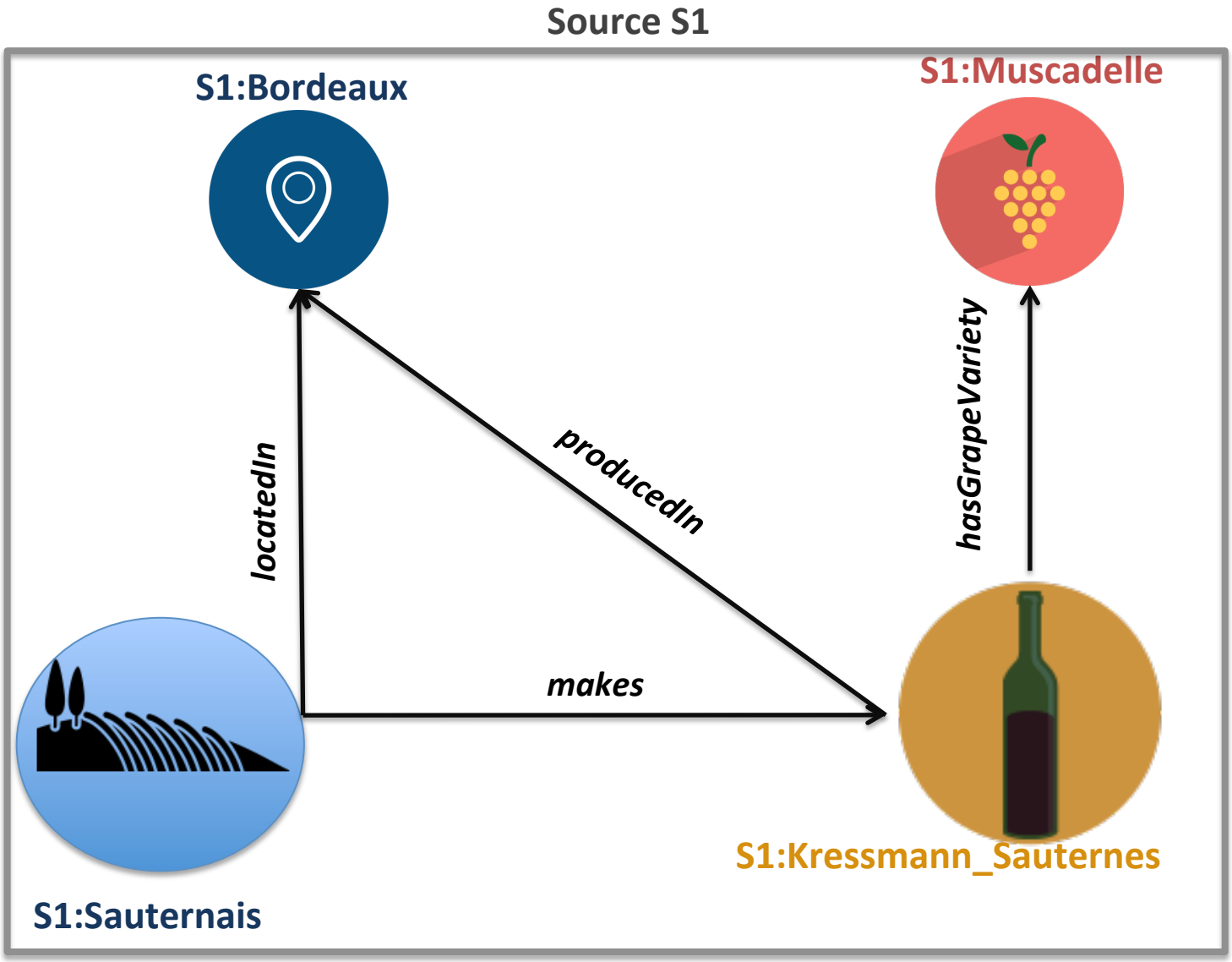
The Web today

- Web usually contains unstructured information

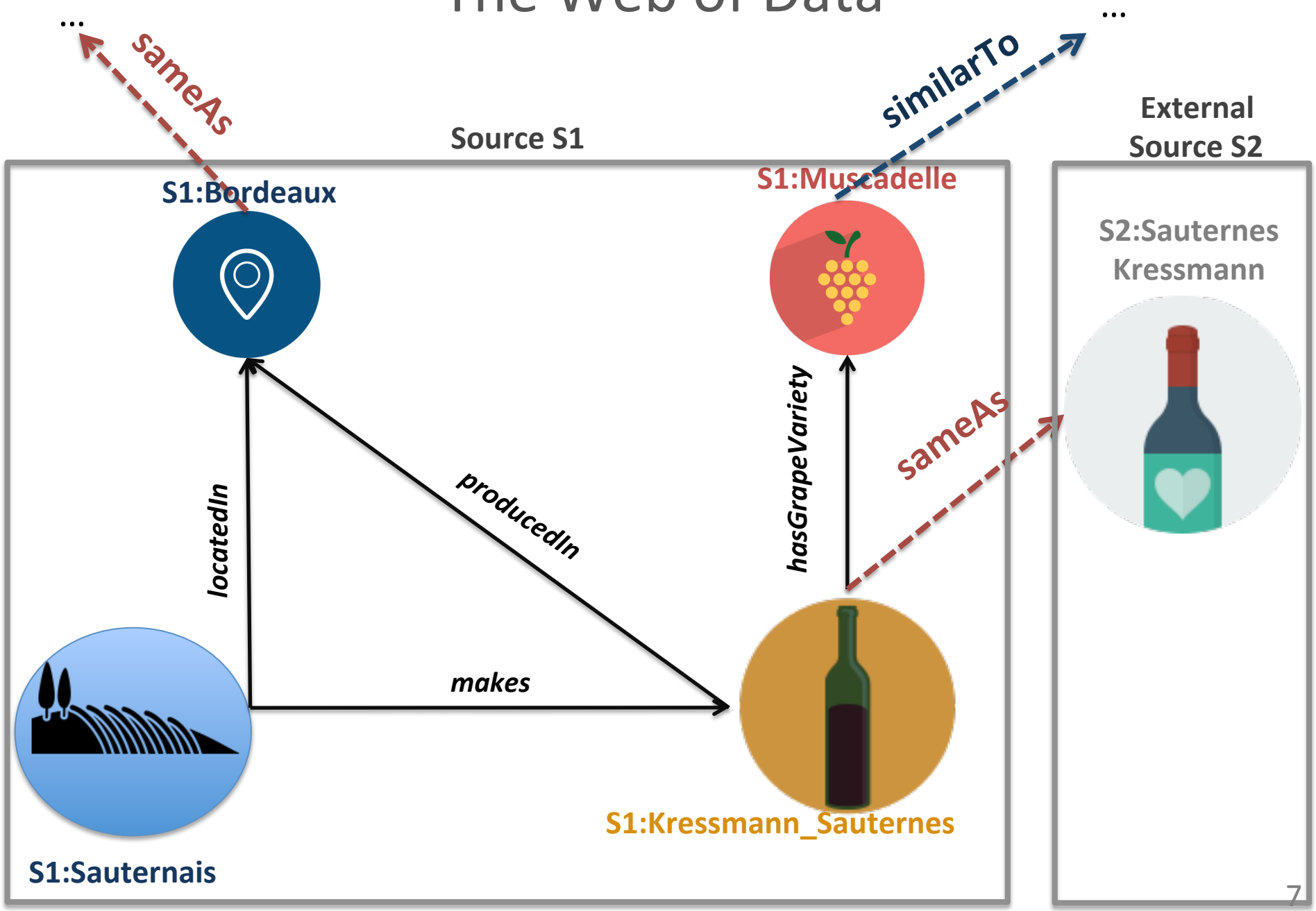
The Web today..



The Web of Data

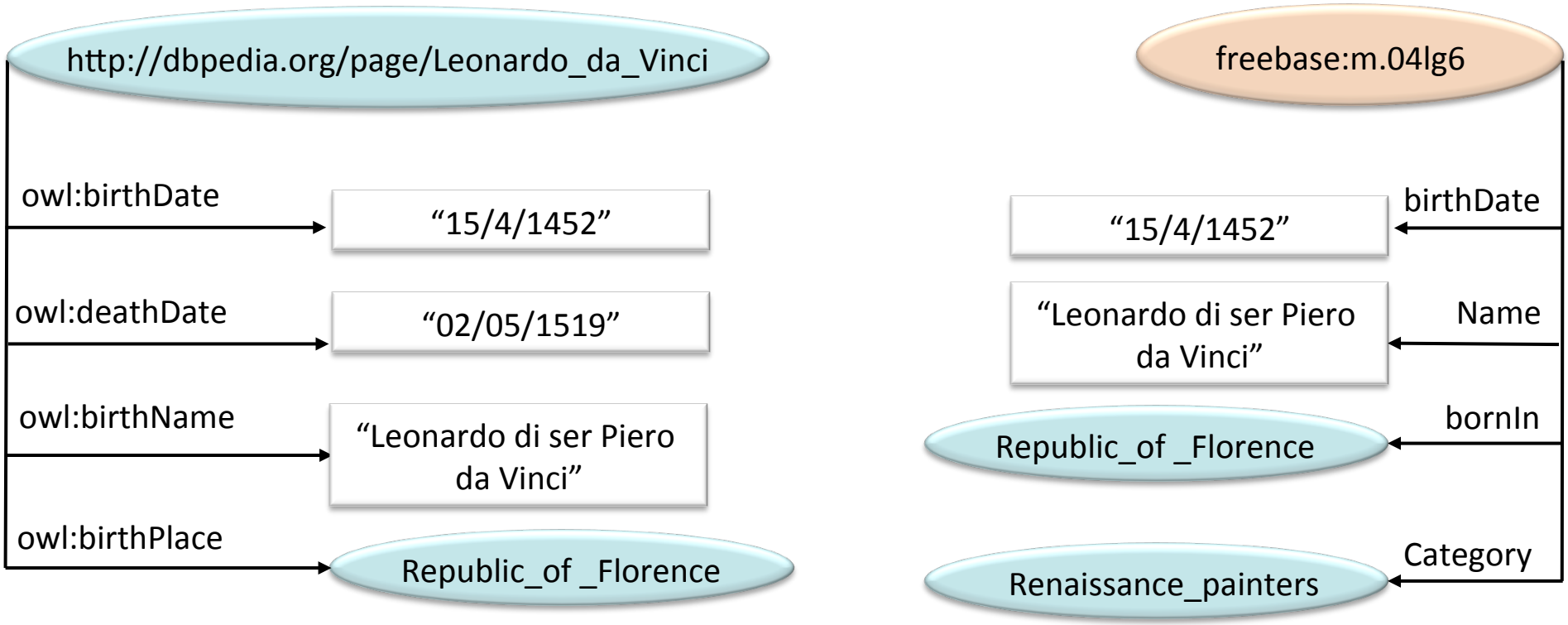


The Web of Data



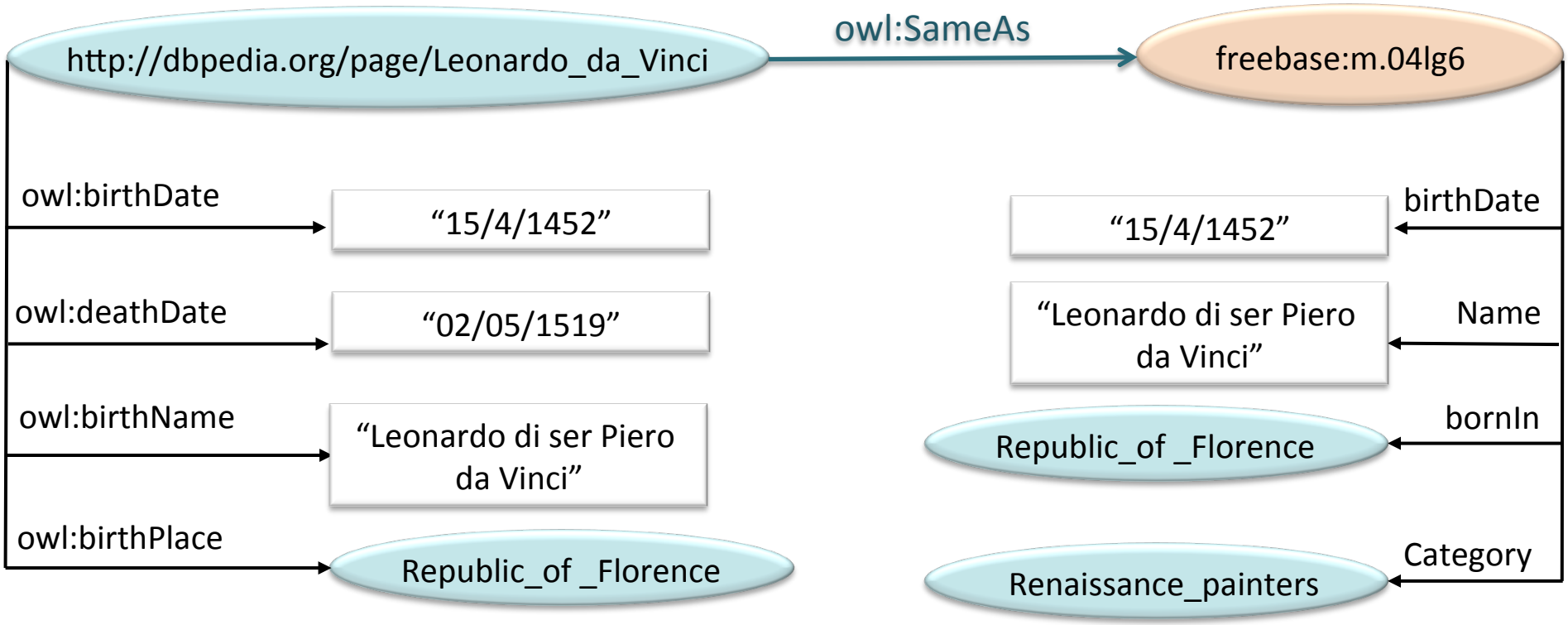
Data Linking: SameAs links

- **SameAs links**: connect instances referring to the same real world object



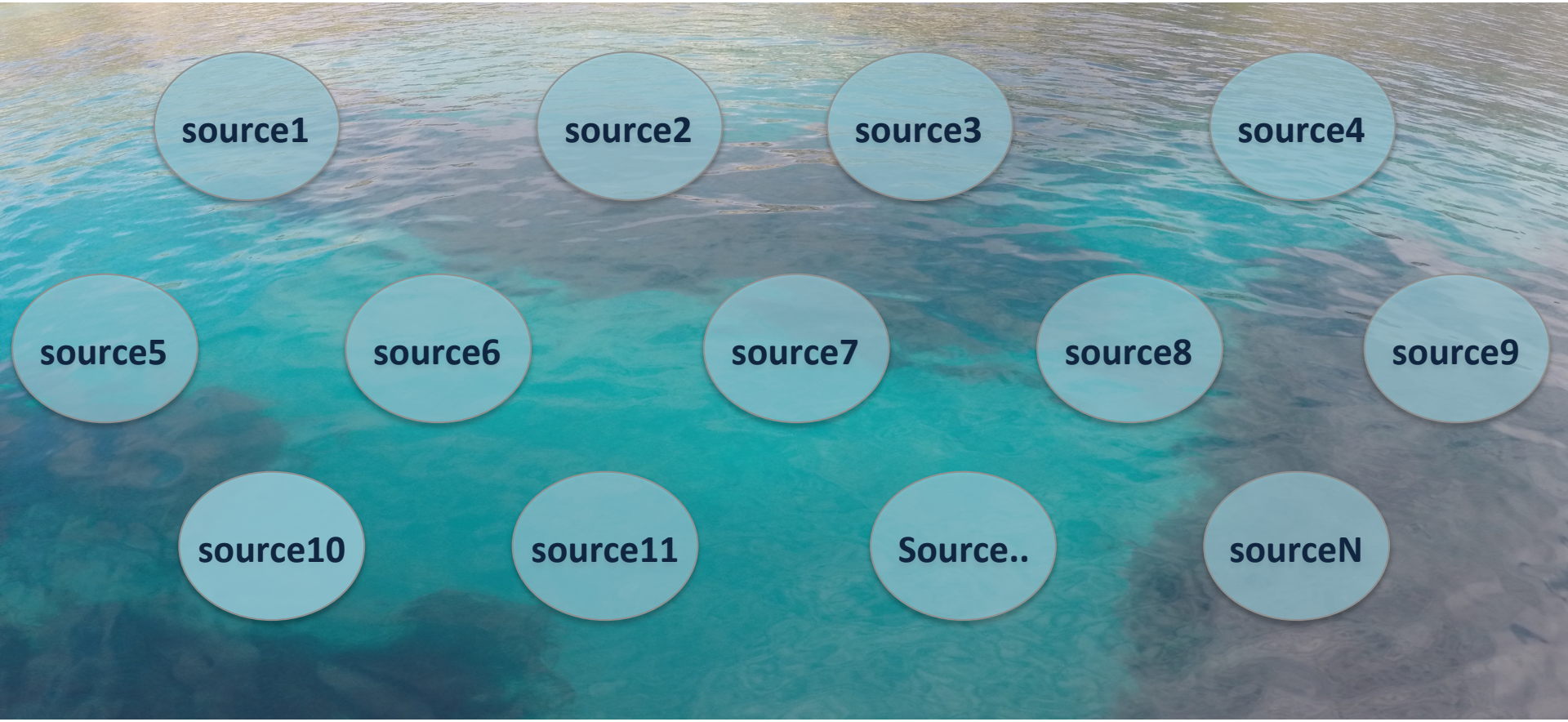
Data Linking: SameAs links

- **SameAs links**: connect instances referring to the same real world object



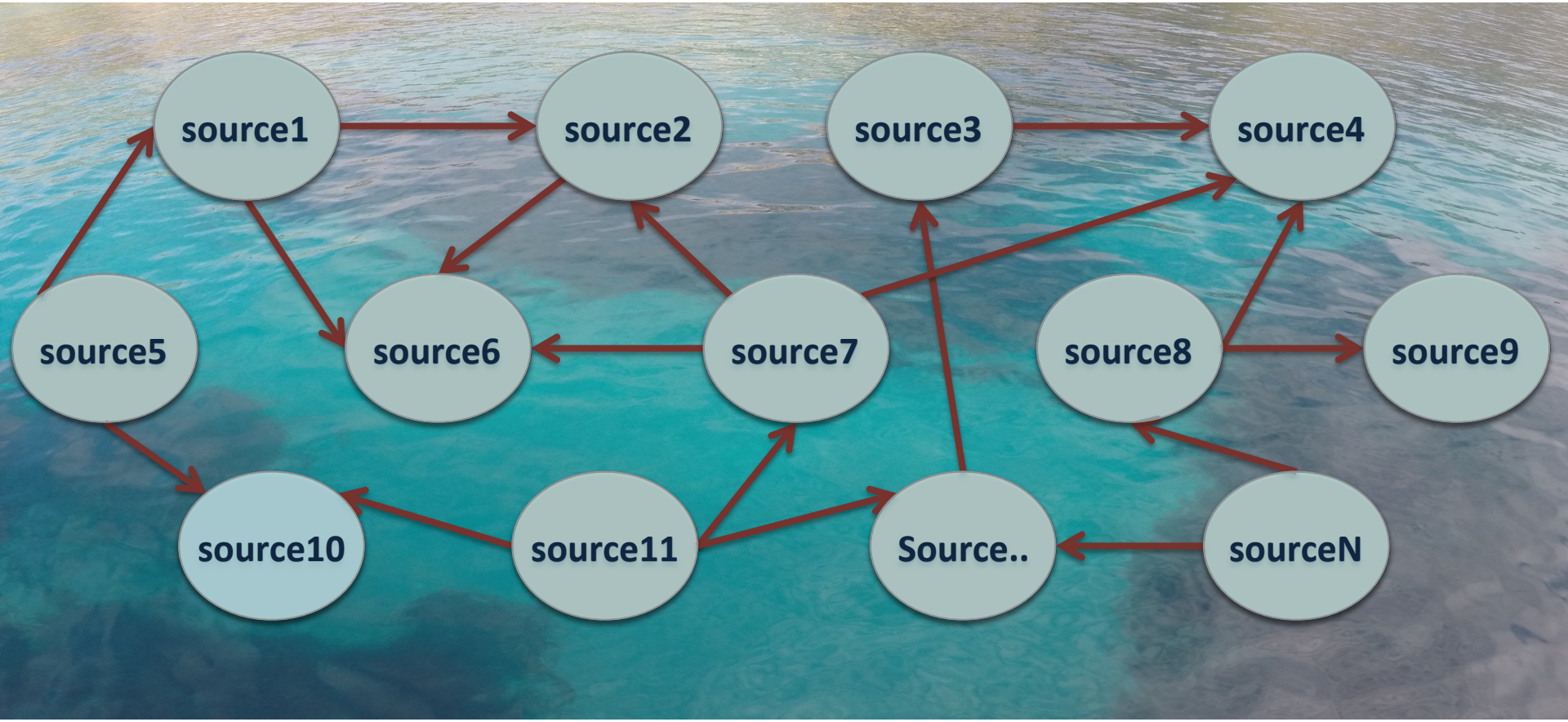
Isolated data

- Is the use of RDF and ontologies enough to obtain a Web of Data?



GOAL - Linked Data

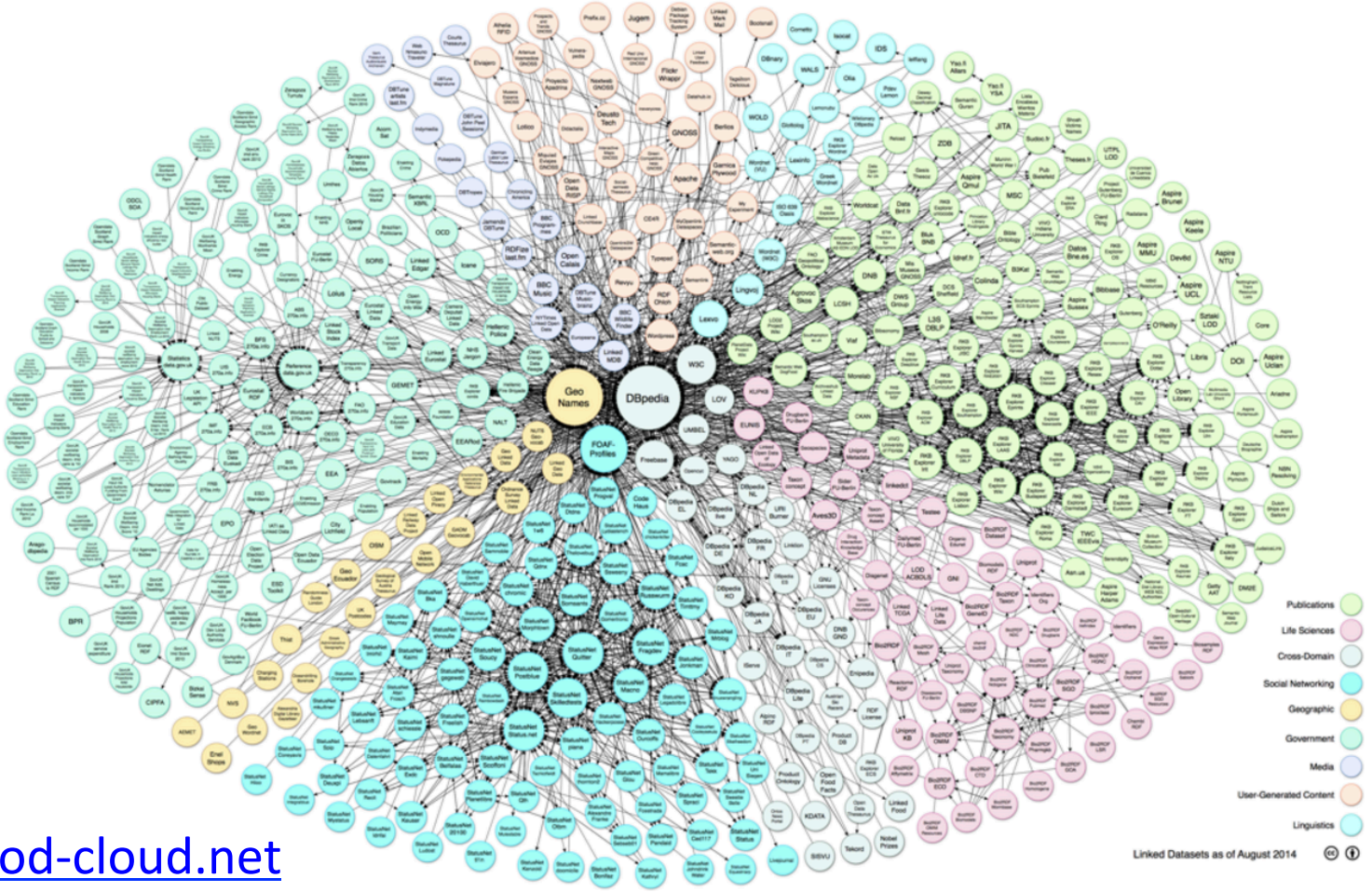
- Is the use of RDF and ontologies enough to obtain a Web of Data?



Linked data today

Linked Open Data cloud (LOD)

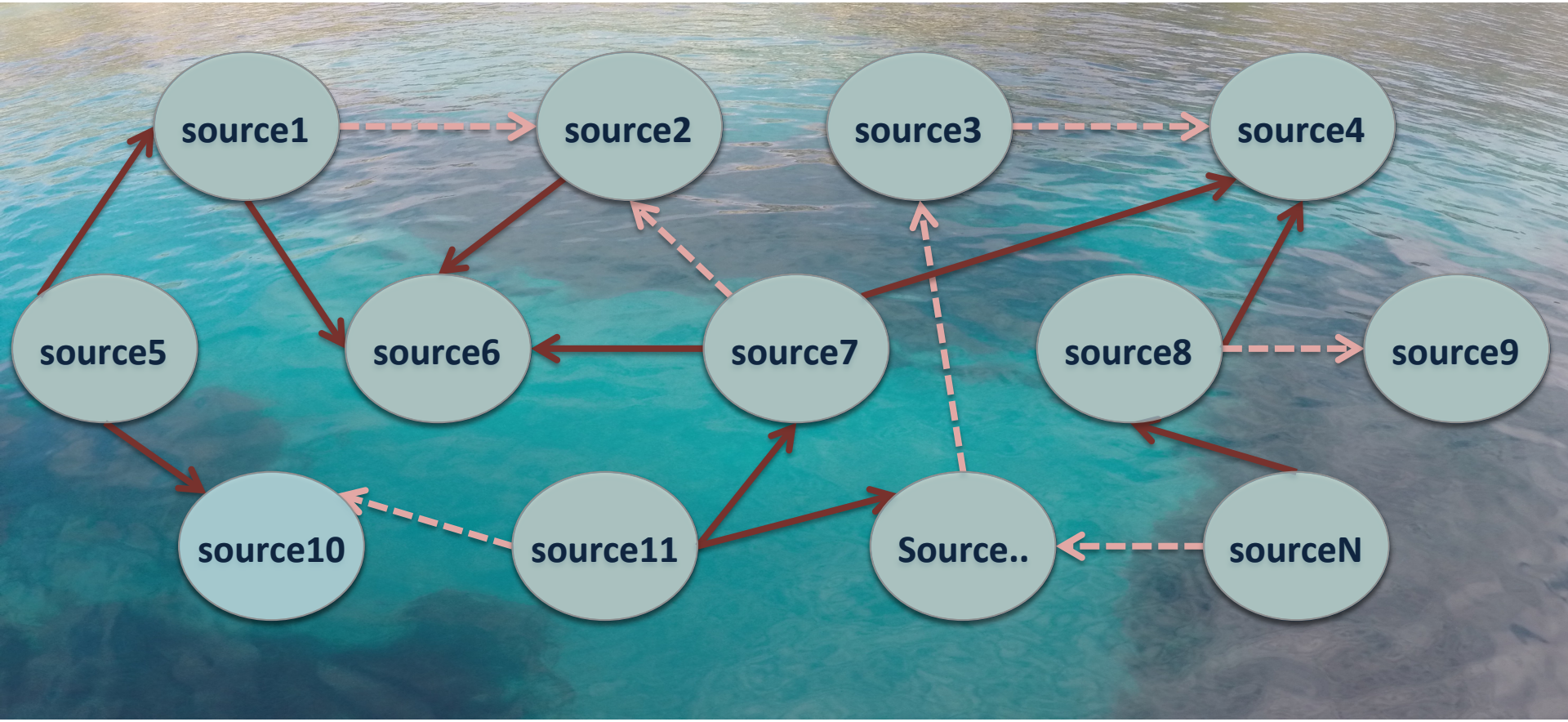
- 130+ billion triples and ≈ 0.5 billion links



http://lod-cloud.net

Linked data today

It is estimated that only a small number of data available on the Web is linked today



Links not defined

Existing links

Data linking approaches

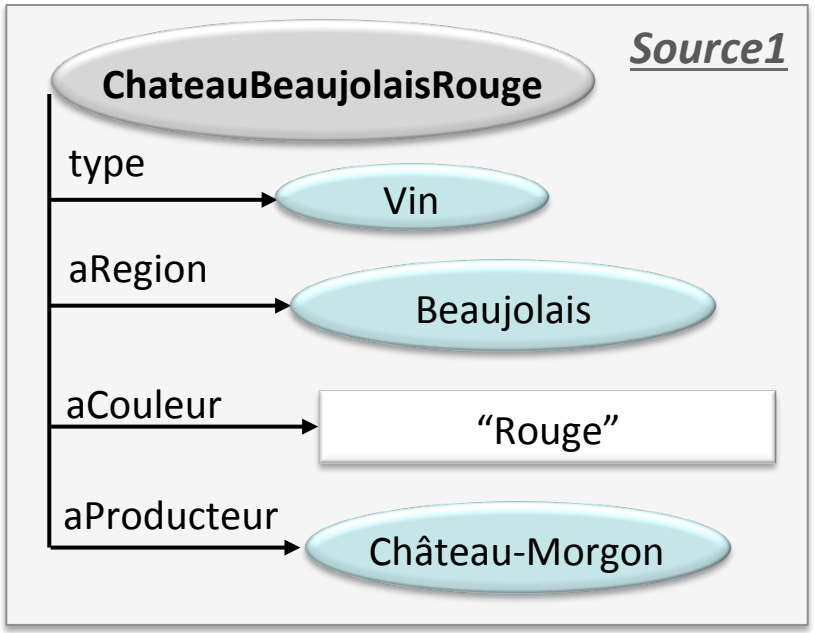
- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)
- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)
- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

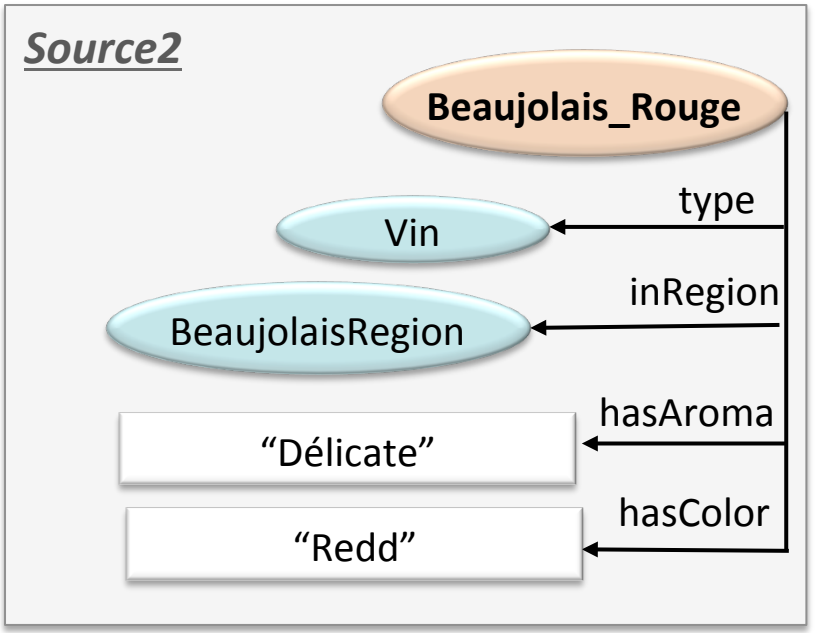
Data linking approaches

- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)
- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)
- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

Instance-based approaches

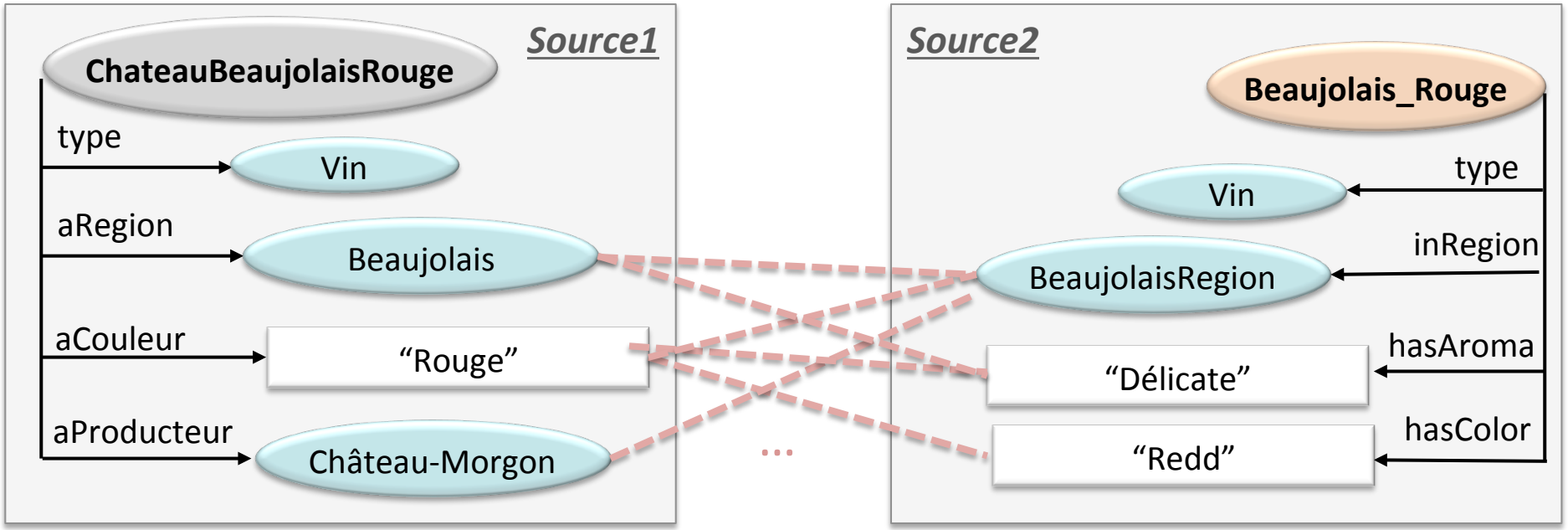
- **Instance-based approaches:** consider only data type properties
  - String comparison





Instance-based approaches

- **Instance-based approaches:** consider only data type properties
  - String comparison

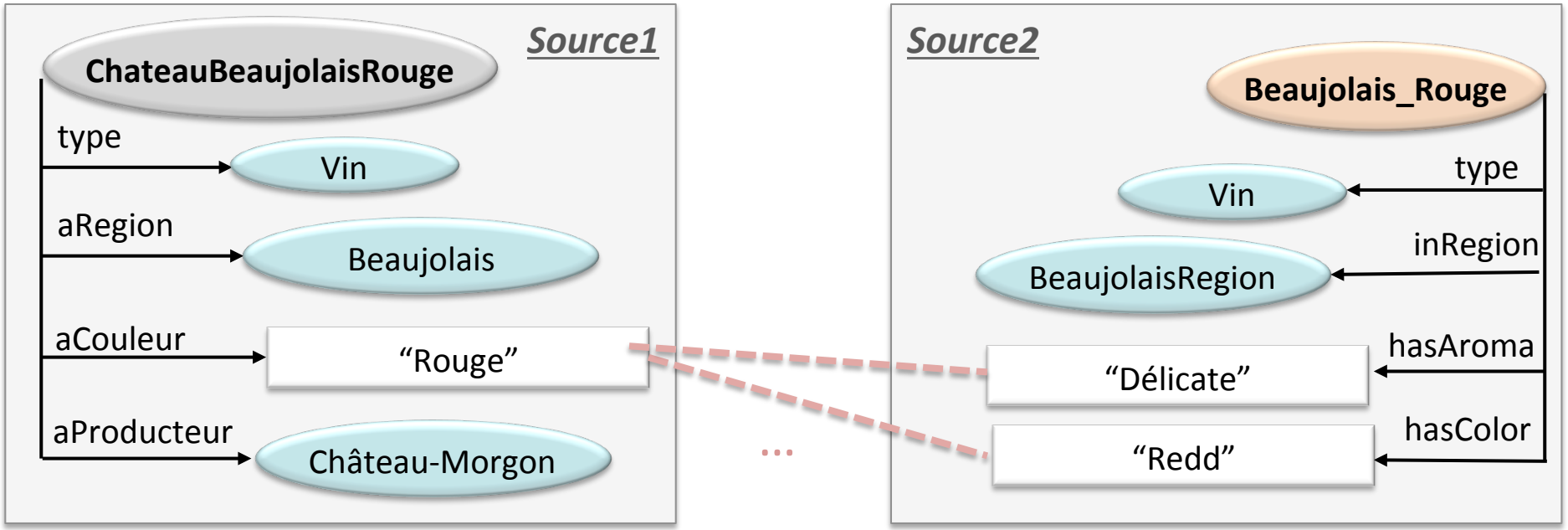


Data linking approaches

- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)
- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)
- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

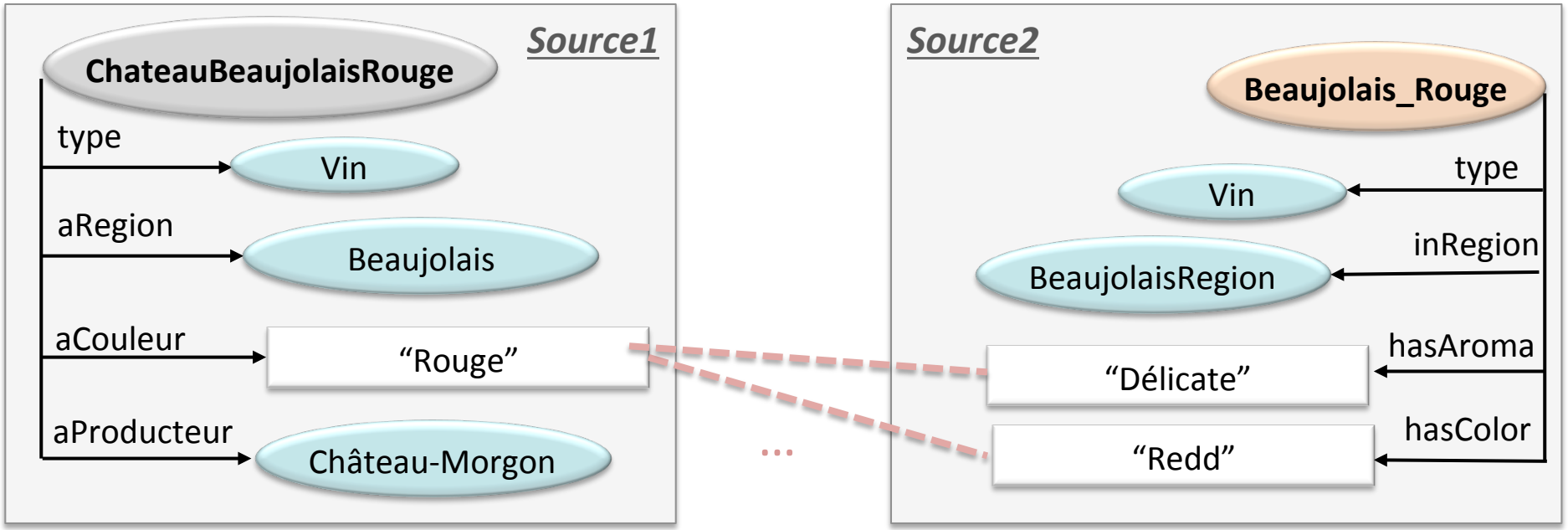
Graph-based approaches

- **Graph-based approaches:**
  - Data type properties – String Comparison



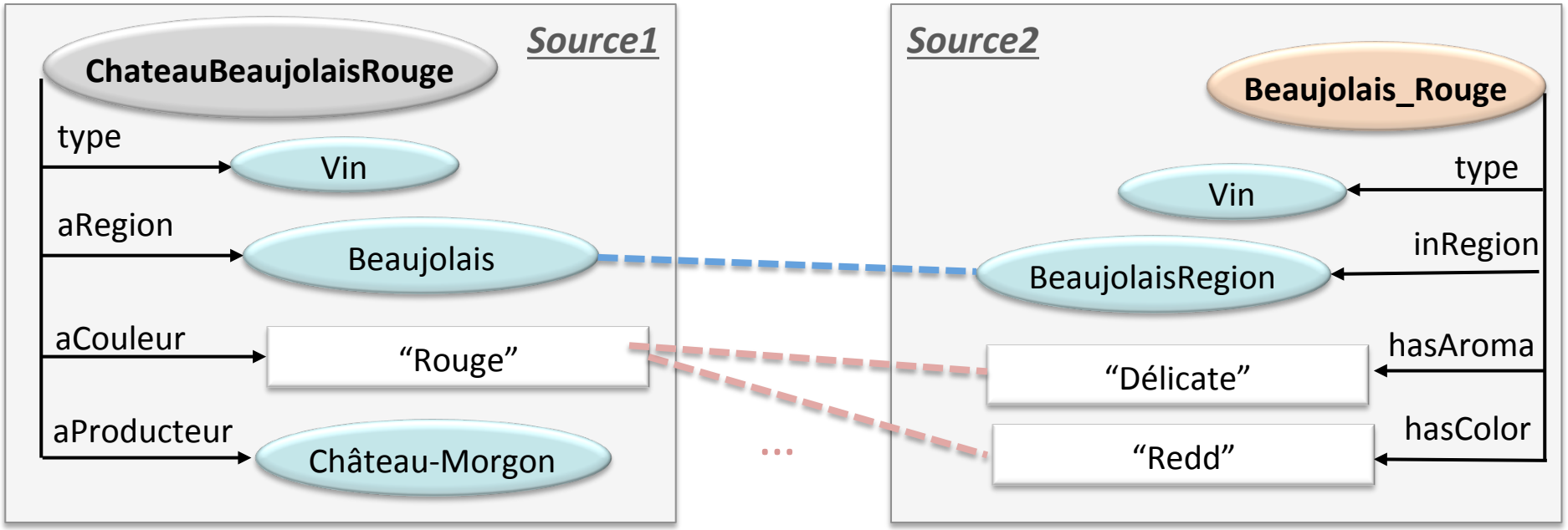
Graph-based approaches

- **Graph-based approaches:**
  - Data type properties – String Comparison
  - Object properties – Object comparison (propagate similarity scores/linking decisions)



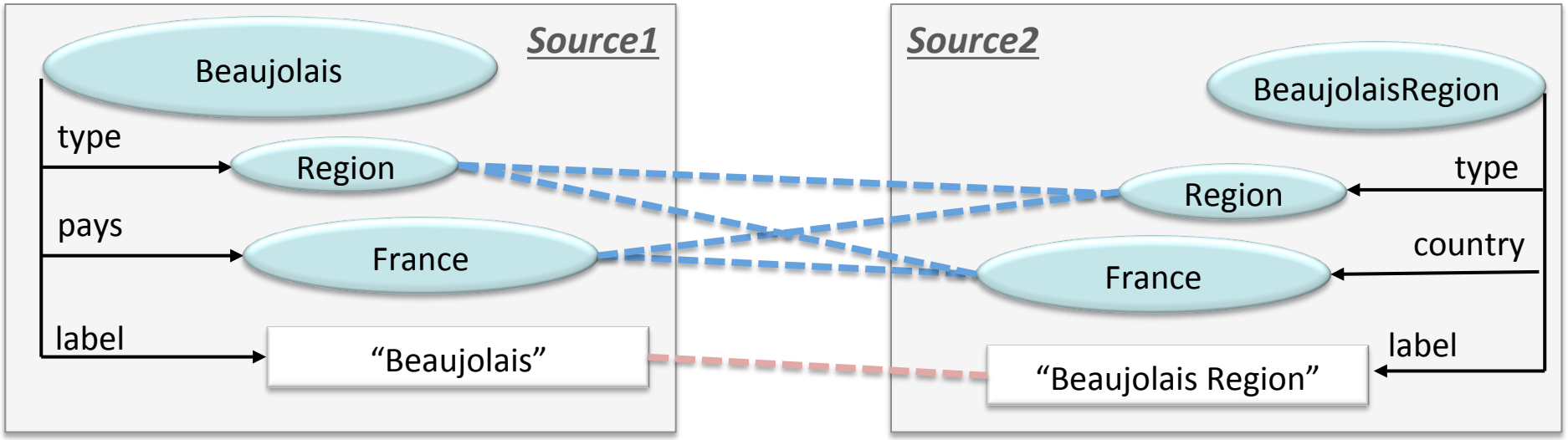
Graph-based approaches

- **Graph-based approaches:**
  - Data type properties – String Comparison
  - Object properties – Object comparison (propagate similarity scores/linking decisions)



Graph-based approaches

- **Graph-based approaches:**
  - Data type properties – String Comparison
  - Object properties – Object comparison (propagate similarity scores/linking decisions)



Data linking approaches

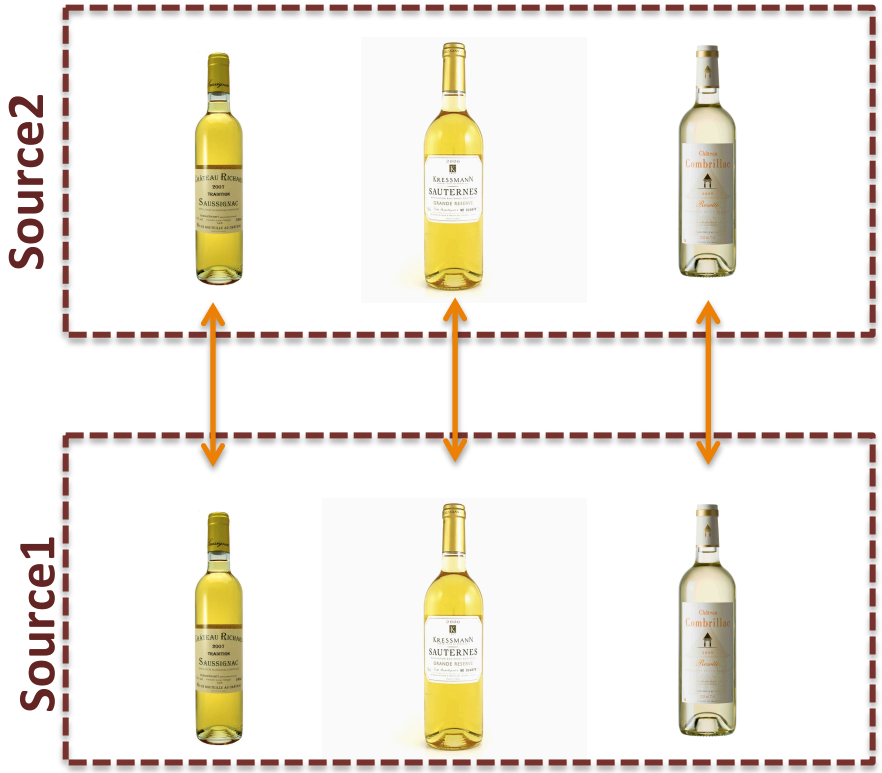
- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)
- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)
- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

Supervised approaches

- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)

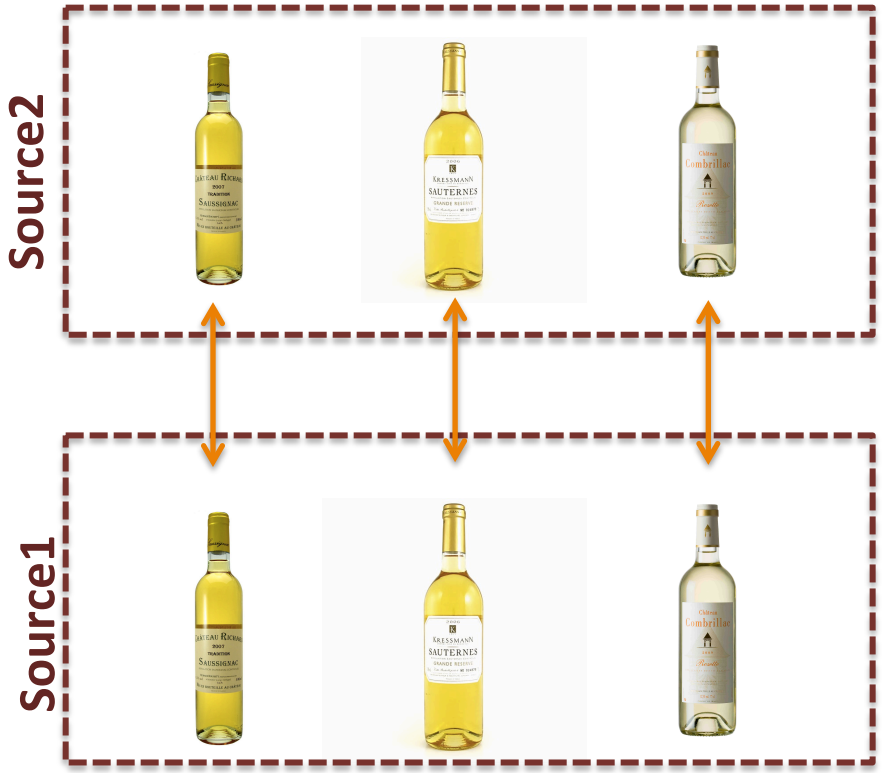
Supervised approaches

- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)



Supervised approaches

- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)
  - Ex. Name(v1, y) ⋀ Name(v2, y) ➔ sameAs(v1,v2)



Data linking approaches

- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)
- **Supervised approaches:** require an expert to build samples of linked data to train models (manual and interactive approaches)
- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

Data linking approaches

- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)

Are all the properties equally important in the data linking??

- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

Data linking approaches

- **Instance-based approaches:** consider only data type properties
- **Graph-based approaches:** consider data type properties and object properties to propagate similarity scores/linking decisions (collective data linking)

Most of the existing approaches use rules to link data

- **Informed approaches:** require knowledge to be declared in the ontology or in other format given by an expert

Data linking using rules

- Rules
  - **Logical Rules**
    - Ex. For instances of the class Restaurant

      homepage(w1, y) ∧ homepage(w2, y) ➔ sameAs(w1, w2)

Data linking using rules

- homepage(w1, y) ∧ homepage(w2, y) ➔ sameAs(w1, w2)

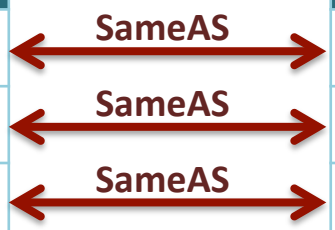
	…	homepage
Restaurant11		www.kitchenbar.com
Restaurant12		www.jardin.fr
Restaurant13		www.gladys.fr
Restaurant14		…

homepage	…	
www.kitchenbar.com		**Restaurant21**
www.jardin.fr		**Restaurant22**
www.gladys.fr		**Restaurant23**
…		**Restaurant24**

Data linking using rules

- homepage(w1, y) ∧ homepage(w2, y) ➔ sameAs(w1, w2)
  - *sameAs(Restaurant11, Restaurant21)*
  - *sameAs(Restaurant12, Restaurant22)*
  - *sameAs(Restaurant13, Restaurant23)*

	…	homepage
Restaurant11		www.kitchenbar.com
Restaurant12		www.jardin.fr
Restaurant13		www.gladys.fr
Restaurant14		…



homepage	…	
www.kitchenbar.com		**Restaurant21**
www.jardin.fr		**Restaurant22**
www.gladys.fr		**Restaurant23**
…		**Restaurant24**

Data linking using rules

- Rules
  - **Logical Rules**
    - Ex. For instances of the class Restaurant

      homepage(w1, y) ∧ homepage(w2, y) ➔ sameAs(w1, w2)

  - **Complex Rules**
    - Ex. For instances of the class Restaurant

      max(Jaccard(lat(w1, n);lat(w2, m));jaro(long(w1, x);long(w2, y)))

      > 0.8 ➔sameAs(w1, w2)

Data linking using rules

- Rules
  - **Logical Rules**
    - Ex. For instances of the class Restaurant

      homepage(w1, y) ∧ homepage(w2, y) ➔ sameAs(w1, w2)

      {homepage}: discriminative property
  - **Complex Rules**
    - Ex. For instances of the class Restaurant

      max(Jaccard(lat(w1, n);lat(w2, m));jaro(long(w1, x);long(w2, y)))

      > 0.8 ➔sameAs(w1, w2)

      {lat, long}: discriminative property set

Data linking using rules

- Rules
  - **Logical Rules**
    - Ex. For instances of the class Restaurant

      homepage(w1, y) ∧ homepage(w2, y) ➔ sameAs(w1, w2)

      ↓

      {homepage}: discriminative property

  - **Complex Rules**
    - Ex. For instances of the class Restaurant

      max(Jaccard(lat(w1, n);lat(w2, m));jaro(long(w1, x);long(w2, y)))

      > 0.8 ➔sameAs(w1, w2)

      ↓

      {lat, long}: discriminative property set

Rules contain discriminative properties => keys

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName] a key?

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName] a key? ✖

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName] a key? ✖

Is [LastName] a key?

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName] a key? ✖

Is [LastName] a key? ✖

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName] a key? ✖

Is [LastName] a key? ✖

Is [FirstName,LastName] a key?

Classic keys

- **Key**: A set of properties that uniquely identifies every instance in the data

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName] a key? ✖

Is [LastName] a key? ✖

Is [FirstName,LastName] a key? ✔

Classic keys - Key Monotonicity

- **Key monotonicity:** When a set of properties is a key, all its supersets are also keys
- **Minimal Key:** A *key* that by removing one property stops being a key

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName,LastName, Birthday] a key?

Classic keys - Key Monotonicity

- **Key monotonicity:** When a set of properties is a key, all its supersets are also keys
- **Minimal Key:** A *key* that by removing one property stops being a key

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName,LastName, Birthday] a key? ✔

Classic keys - Key Monotonicity

- **Key monotonicity:** When a set of properties is a key, all its supersets are also keys
- **Minimal Key:** A *key* that by removing one property stops being a key

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName,LastName, Birthday] a key? ✔

*Is [FirstName,LastName, Birthday] a **minimal** key?*

Classic keys - Key Monotonicity

- **Key monotonicity:** When a set of properties is a key, all its supersets are also keys
- **Minimal Key:** A *key* that by removing one property stops being a key

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName,LastName, Birthday] a key? ✔

*Is [FirstName,LastName, Birthday] a **minimal** key?* ✖

Classic keys - Key Monotonicity

- **Key monotonicity:** When a set of properties is a key, all its supersets are also keys
- **Minimal Key:** A *key* that by removing one property stops being a key

	FirstName	LastName	Birthdate	Profession
Person1	Anne	Tompson	15/02/88	Actor
Person2	Marie	Tompson	02/09/75	Researcher
Person3	Marie	David	15/02/85	Teacher
Person4	Vincent	Solgar	06/12/90	Teacher

Is [FirstName,LastName, Birthday] a key? ✔

*Is [FirstName,LastName, Birthday] a **minimal** key?* ✖

***Minimal keys**: [[FirstName,LastName],[FirstName,Profession] [Birthdate], [LastName,Profession]]*

Keys declared by experts for data linking

- Not an easy task:
  - Experts are not aware of all the keys

    Ex. {SSN}, {ISBN} easy to declare

    Ex. {Name, DateOfBirth, BornIn} **is it a key for the class Person?**

  - Erroneous keys can be given by experts

  - As many keys as possible
    - More keys => More linking rules

Automatic key discovery

- **Naive automatic way to discover keys**
  - Examine all the possible combinations of properties
  - Scan all instances for each candidate key

  Example: Class described by 15 properties ➔ 2^{15}-1 = 32767 candidate keys

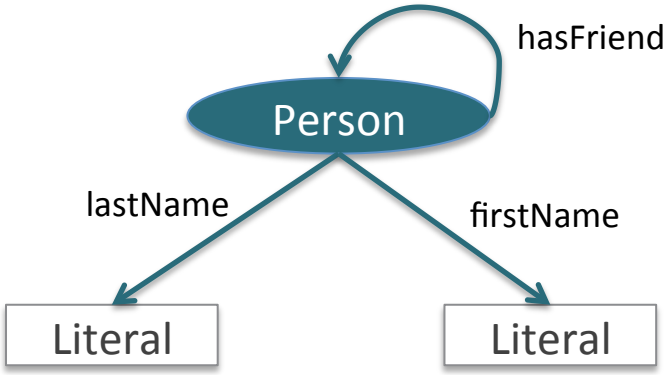
- Discover keys efficiently are necessary

Key Discovery - Related Work

Relational Databases					
Approach	**Composite keys**	**Complete set of keys**		**Approximate keys**	**Incomplete data heuristics**
[SBNHR06]	✓	✓			
[AN11]	✓	✓			
[VLM12]	✓				
[HJAQR+13]	✓	✓			
[KLL13]	✓	✓			✓
Semantic Web					
Approach	**Composite keys**	**Complete set of keys**	**OWL2 keys**	**Approximate keys**	**Incomplete data heuristics**
[SAS11]			✓	✓	
[ADS12]	✓	✓		✓	
[PSS13]	✓	✓	✓		✓
[SAPS14]	✓	✓	✓	✓	✓
[SMN15]	✓	✓		✓	

Data in the Web today

- **RDF Data conform to ontologies**



	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria	
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	

Data in the Web today

- **RDF Data are multi-valued**

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	**Person2, Person3**
Person2	Tompson	Maria	
Person3	David	George	**Person2, Person4**
Person4	Solgar	Michel	

Ex.
<Person1> <hasFriend> <Person2>
<Person1> <hasFriend> <Person3>

<Person3> <hasFriend> <Person2>
<Person3> <hasFriend> <Person4>

Data in the Web today

- **RDF Data in the Web are often incomplete**
  - Produced automatically
  - Specific purpose data

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria	
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	

Person1=?= Person2=?=Person3=?=Person4

hasFriend(Person2,Person3) …. ?

hasFriend(Person1, Person4) …. ?

Data in the Web today

- **RDF Data in the Web are often incomplete**
  - Produced automatically
  - Specific purpose data

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria	
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	

Person1=?= Person2=?=Person3=?=Person4

hasFriend(Person2,Person3) …. ?

hasFriend(Person1, Person4) …. ?

Assumptions

- **Unique Name Assumption (UNA)**: different identifiers refer to distinct real world objects
  - Person1<> Person2 <> Person3 <> Person4

  Fulfilled in datasets extracted from RDB, YAGO

- **Uniform Vocabulary**: Syntactically different ➔semantically different in one dataset
  - “UK” and “United Kingdom” never in the same dataset

Data in the Web today

- **RDF Data in the Web are often incomplete**
  - Produced automatically
  - Specific purpose data

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria	
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	

Person1=?= Person2=?=Person3=?=Person4

hasFriend(Person2,Person3) …. ?

hasFriend(Person1, Person4) …. ?

Open world VS closed world

- **Open World Assumption (OWA)**

  What is not in the data is not false

  - Ex. hasFriend(Person2, Person3) => ???

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	**Person2, Person3**
Person2	Tompson	Maria	
Person3	David	George	**Person2, Person4**
Person4	Solgar	Michel	

Open world VS closed world

- **Open World Assumption (OWA)**

  What is not in the data is not false

  - Ex. hasFriend(Person2, Person3) => ???

- **Closed World Assumption (CWA)**

  The data are complete, no information is missing

  - Ex. hasFriend(Person2, Person3) => false

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	**Person2, Person3**
Person2	Tompson	Maria	
Person3	David	George	**Person2, Person4**
Person4	Solgar	Michel	

Open world VS closed world

- **Open World Assumption (OWA)**

  What is not in the data is not false

  - Ex. hasFriend(Person2, Person3) => ???

- **Closed World Assumption (CWA)**

  The data are complete, no information is missing

  - Ex. hasFriend(Person2, Person3) => false

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	**Person2, Person3**
Person2	Tompson	Maria	
Person3	David	George	**Person2, Person4**
Person4	Solgar	Michel	

The Open World Assumption is more adapted to the data

OWL2 Key, Key in the Open World

- **OWL** (Web Ontology Language)

- **OWL2 Key for a class**: a combination of properties *k={p1,..,pn}* that uniquely identify each instance of a class
  - If for every property in *k* two instances share at least one value then these instances are the same

hasKey(Restaurant(latitude) (longitude)) means:

$\text{Latitude}(r_1, x) \wedge \text{latitude}(r_2, x) \wedge \text{longitude}(r_1, y) \wedge \text{longitude}(r_2, y) \rightarrow \text{sameAs}(r_1, r_2)$

Keys in the Open World

- **Key in the Open World**
  - **Key discovery**: For a set of properties that is a key, there is no instance that shares values with another instances

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria, Hellen	Person1
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	Person3

Keys in the Open World

- **Key in the Open World**
  - **Key discovery**: For a set of properties that is a key, there is no instance that shares values with another instances

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria, Hellen	Person1
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	Person3

  - Ex. firstName is a **key in the Open World** since
    - There do exist any people that share first names

Keys in the Open World

- **Key in the Open World**
  - **Key discovery**: For a set of properties that is a key, there is no instance that shares values with another instances

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	**Person2, Person3**
Person2	Tompson	Maria, Hellen	Person1
Person3	David	George	**Person2,** Person4
Person4	Solgar	Michel	**Person3**

  - Ex. hasFriend is **not** a key in the Open World since
    - There exist at least two people that share a friend

Keys in the Closed World

- **Key in the Closed World**
  - For each property that participate in a key, every set of values for a property and an instance should be unique

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria	Person1
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	Person3

Keys in the Closed World

- **Key in the Closed World**
  - For each property that participate in a key, every set of values for a property and an instance should be unique

	lastName	firstName	hasFriend
Person1	Tompson	Manuel	Person2, Person3
Person2	Tompson	Maria	Person1
Person3	David	George	Person2, Person4
Person4	Solgar	Michel	Person3

  - Ex. hasFriend is a **key in the Closed World** since
    - **{Person2,Person3} <> {Person1} <> {Person2,Person4} <> {Person3}**

Keys in the Closed World

- **Key in the Closed World**
  - For each property that participate in a key, every set of values for a property and an instance should be unique
  - Ex. hasFriend is a **key in the Closed World** since
    - **{Person2,Person3} <> {Person1} <> {Person2,Person4} <> {Person3}**
  - **Data linking**
    - To produce a **sameAs link** between two instances i1 and i2 using a key, all the sets of values for every property in the key should match.
      - Ex. If the property has friend is a key, two instances can be linked if the set of their friends are identical

Closed World approaches
VS
Open World approaches

Key Discovery Approaches

- **CWA approaches**
  - Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking
  - ROCKER: A Refinement Operator for Key Discovery

- **OWA approaches**
  - KD2R: a Key Discovery method for reference reconciliation
  - SAKey: Scalable almost key discovery in RDF data
  - Linkkey: Data interlinking through robust Linkkey extraction

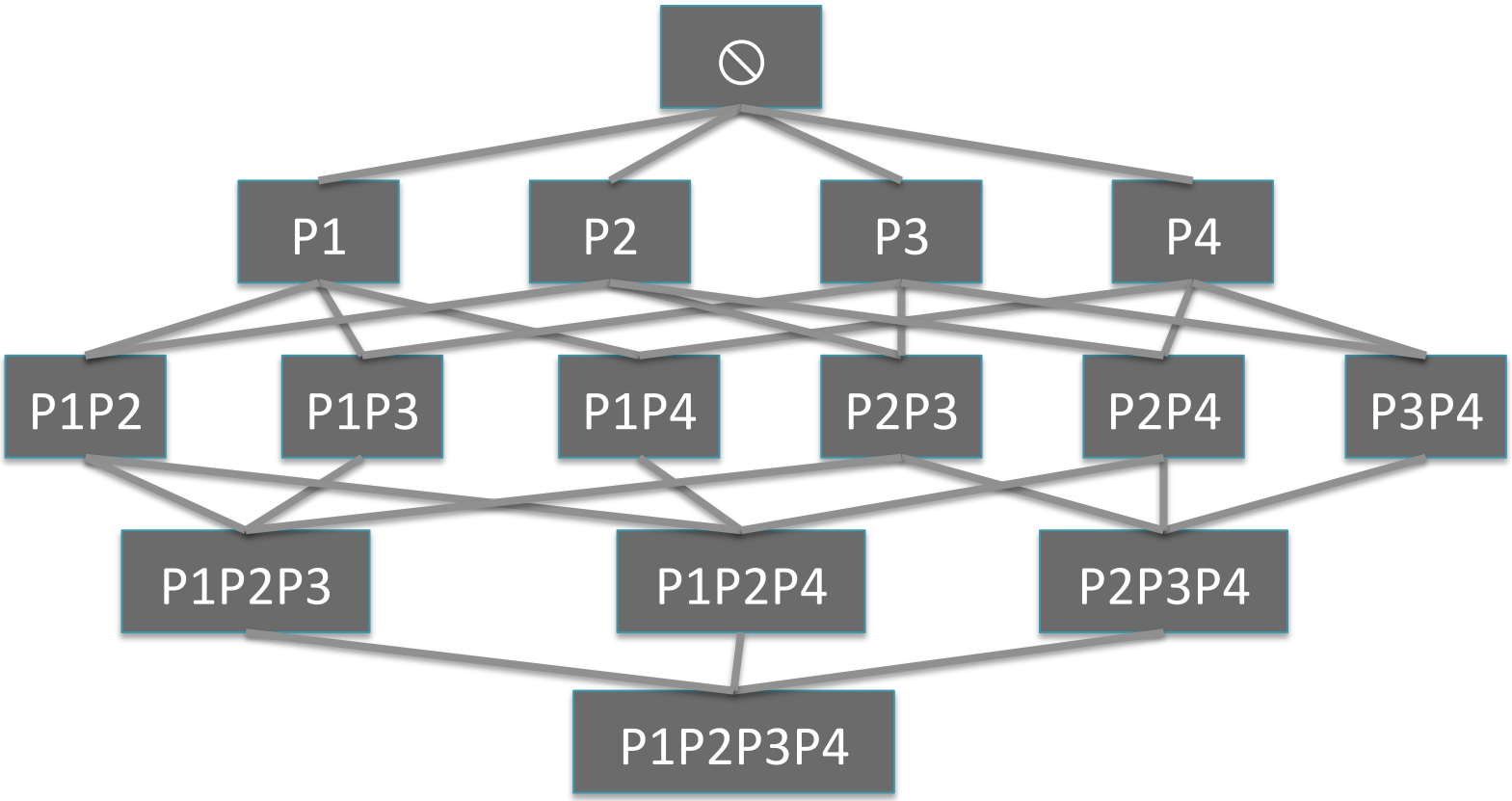
Key Discovery Approaches

- **CWA approaches**
  - **Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking**
  - ROCKER: A Refinement Operator for Key Discovery

- **OWA approaches**
  - KD2R: a Key Discovery method for reference reconciliation
  - SAKey: Scalable almost key discovery in RDF data
  - Linkkey: Data interlinking through robust Linkkey extraction

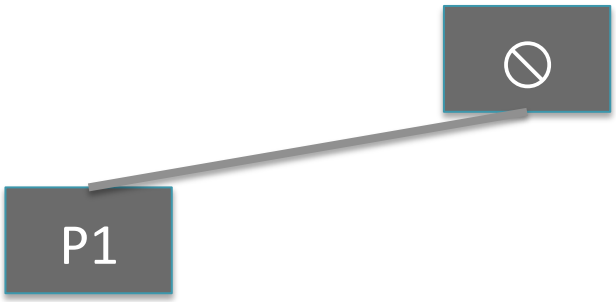
Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



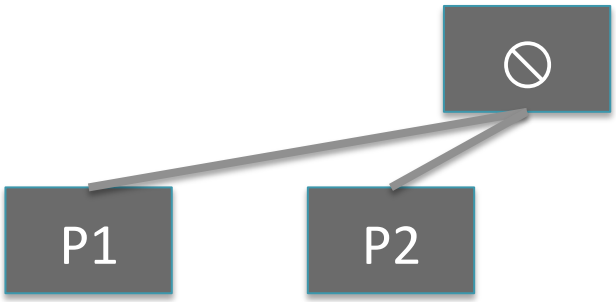
Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



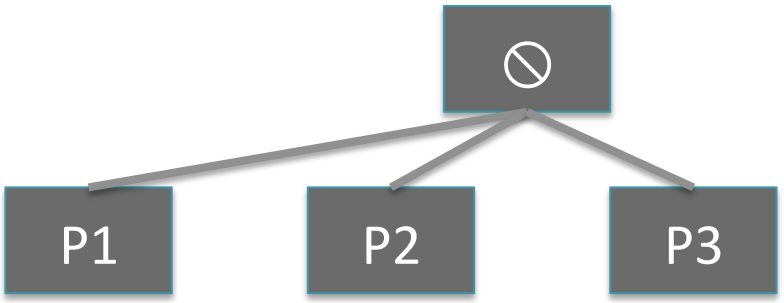
Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



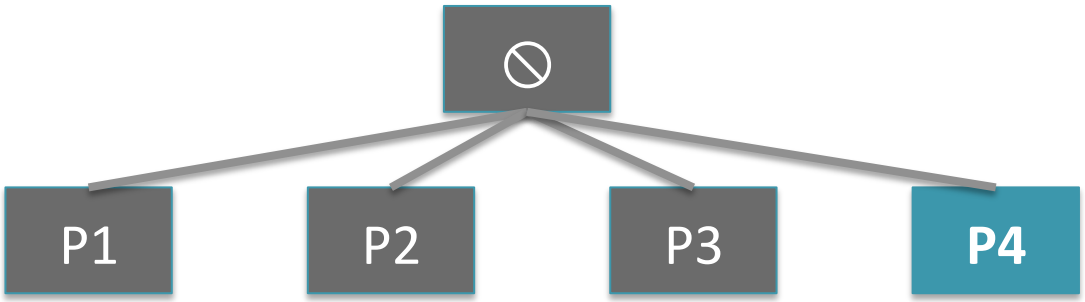
Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



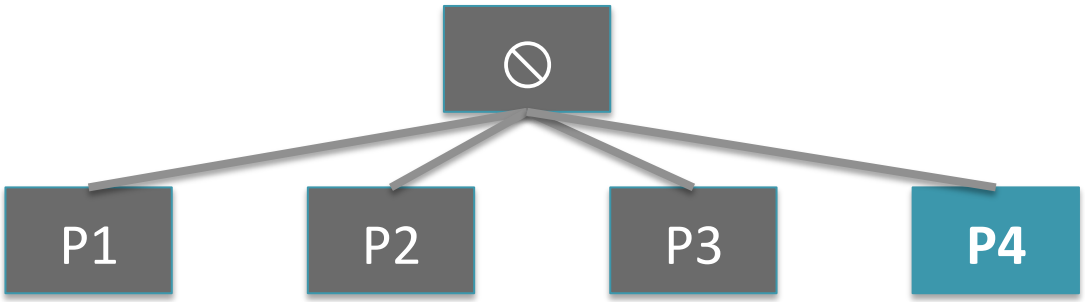
Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

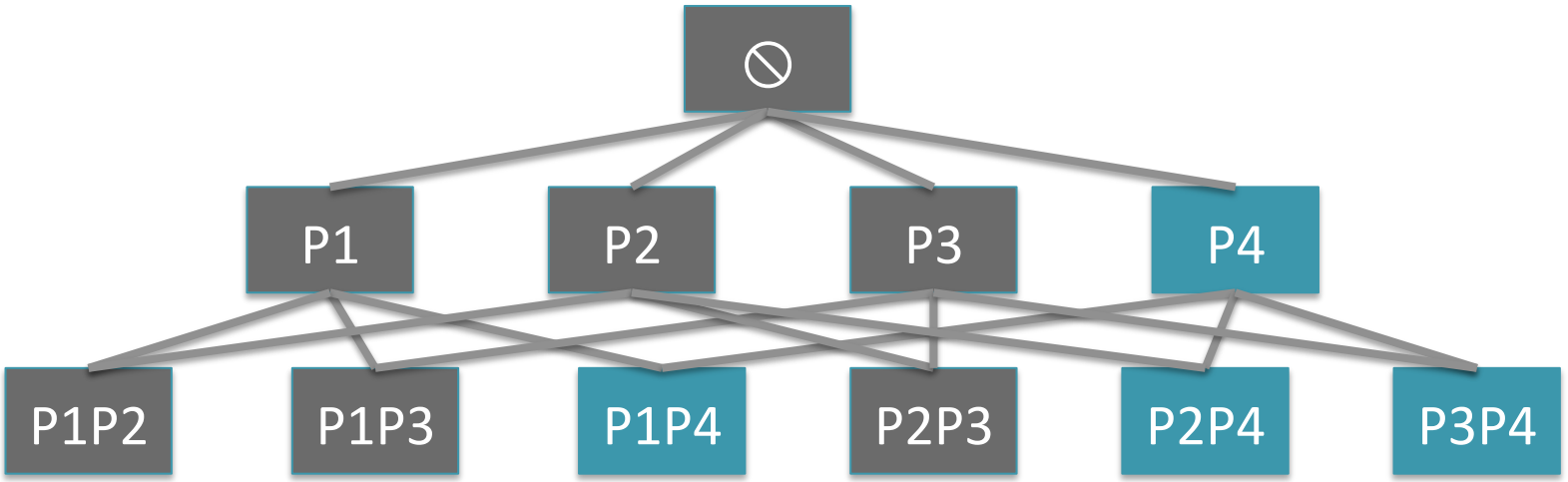
- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



- ***If P4 is a key*** *=> P1P4, P2P4,.., P1P2P3P4* ***are also keys***

Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

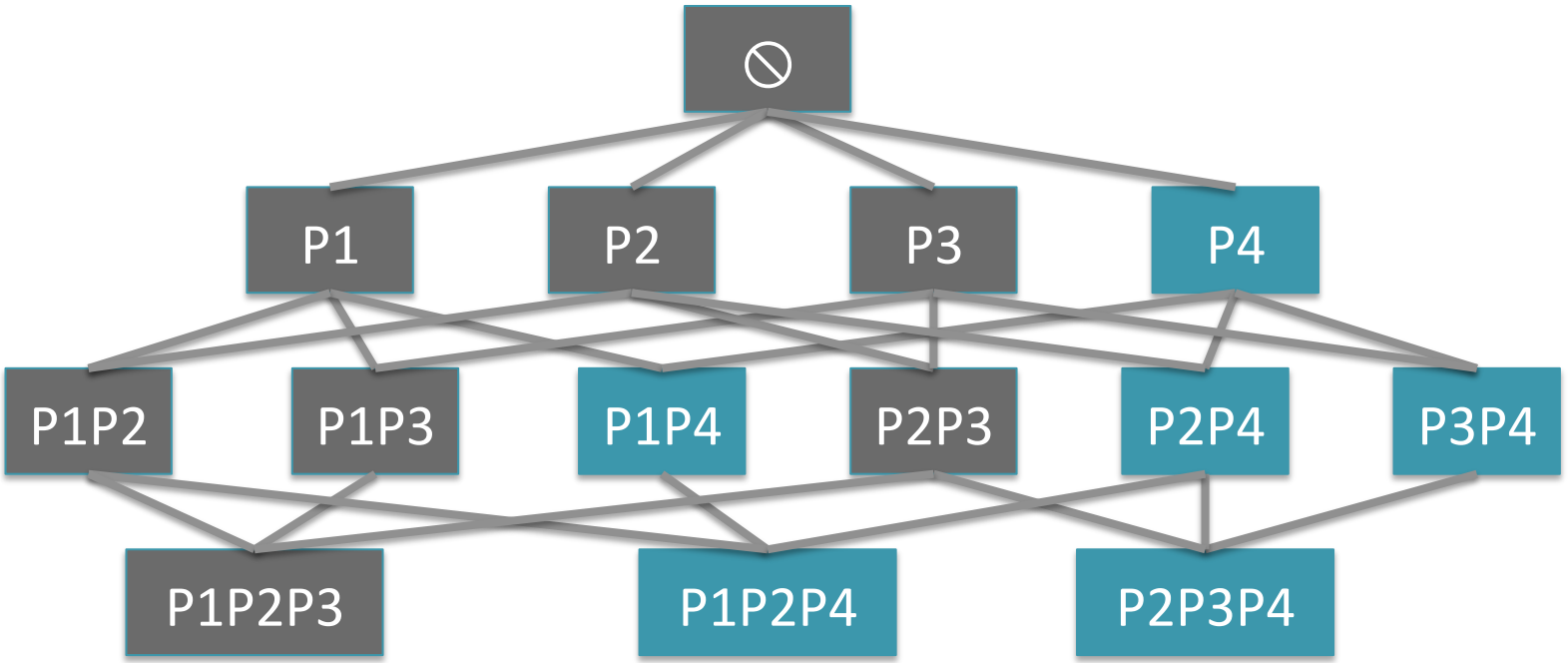
- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



- ***If P4 is a key*** *=> P1P4, P2P4,.., P1P2P3P4* ***are also keys***

Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

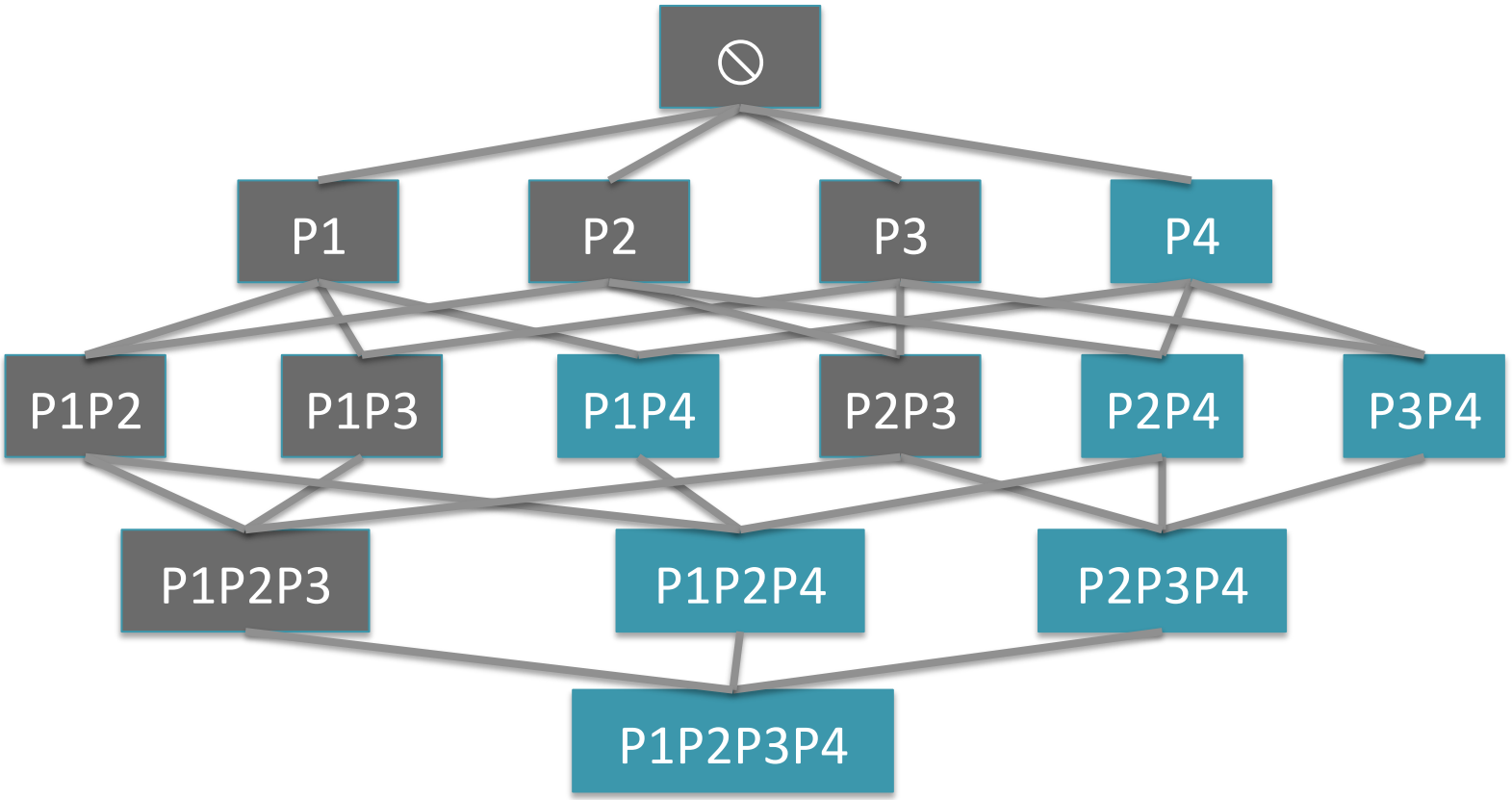
- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



- ***If P4 is a key*** *=> P1P4, P2P4,.., P1P2P3P4* ***are also keys***

Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- Bottom-up approach that discovers
  - **Keys in the Closed World**
  - **Pseudo keys** - keys that tolerate exceptions



- ***If P4 is a key*** *=> P1P4, P2P4,.., P1P2P3P4* ***are also keys***

Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking [ADS12]

- To verify if a set of properties is a key
  - ***Partition*** instances according to their sharing values
  - If each partition contains only one instances => **Key**

- Key quality measures
  - **Support** of a set of properties *P*:

$$\text{support}(P) = \frac{\#\ instances\ described\ by\ P}{\#\ all\ instances}$$

  - **Discriminability** of a set of properties *P* (pseudo-keys):

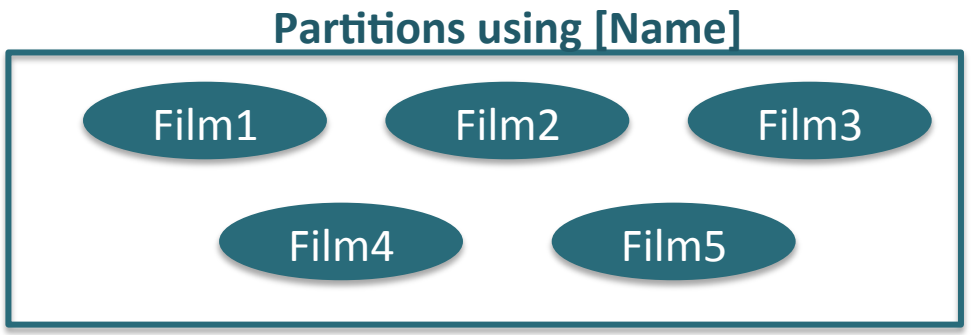
$$dis(P) = \frac{\#\ singleton\ partitions}{\#\ partitions}$$

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	“Ocean’s 11”	“B. Pitt” “J. Roberts”	“S. Soderbergh”	“3/4/01”	www.oceans11.com	---
film2	“Ocean’s 12”	“B. Pitt” “J. Roberts”	“S. Soderbergh” “R. Howard”	“2/5/04”	www.oceans12.com	“english”
film3	“Ocean’s 13”	“B. Pitt” “G. Clooney”	“S. Soderbergh” “R. Howard”	“30/6/07”	www.oceans13.com	“english”
film4	“The descendants”	“N. Krause” “G. Clooney”	“A. Payne”	“15/9/11”	---	“english”
film5	“Bourne Identity“	“D. Liman”	---	“12/6/12”	www.bourneIdentity.com	“english”

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	"B. Pitt" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"

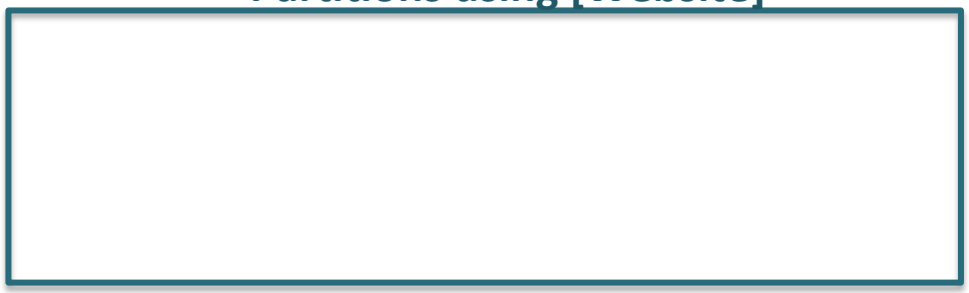


[Name]: key
support 5/5 =1
Discriminability = 5/5 =1

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	"B. Pitt" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"

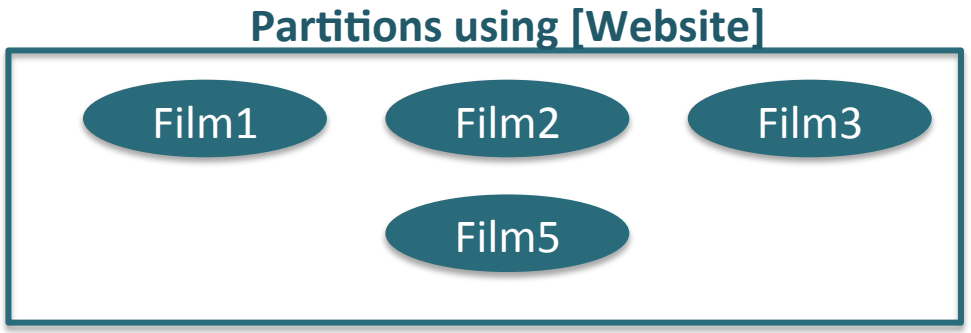
Partitions using [Website]



[Website]: ???

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	"B. Pitt" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"



[Website]: key
Support = 4/5 =0.8
Discriminability = 4/4 =1

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	"B. Pitt" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"

[ADS12] - Key example

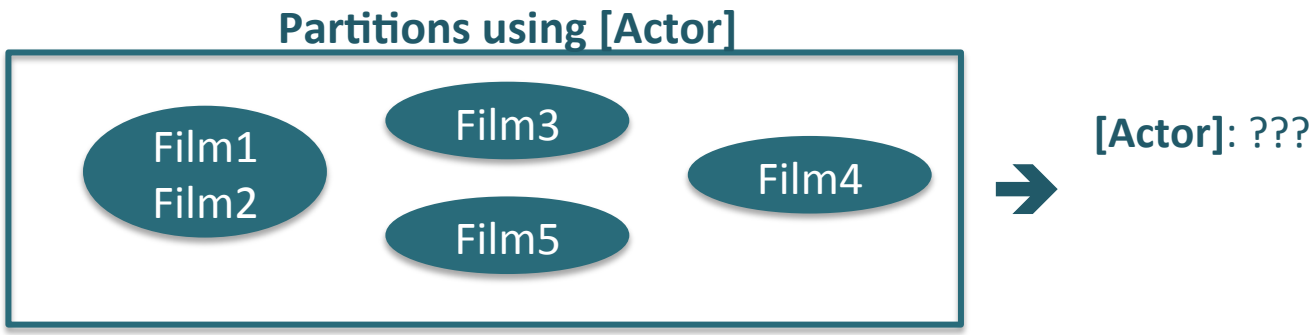
	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	"B. Pitt" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"

Partitions using [Language]

[Language]: pseudo key
Support = 4/5 =0.8
Discriminability = 0/1=0

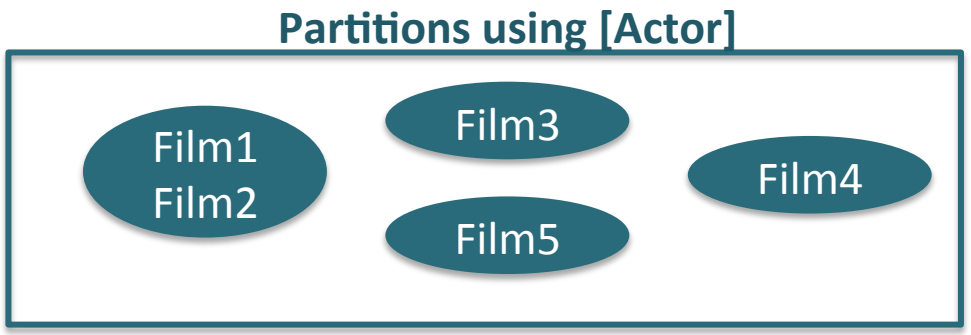
[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	**"B. Pitt"** **"J. Roberts"**	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	**"B. Pitt"** **"J. Roberts"**	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"



[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	**"B. Pitt" "J. Roberts"**	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	**"B. Pitt" "J. Roberts"**	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"



[Actor]: pseudo key
Support = 5/5 =1
Discriminability = 3/4=0.75

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	"B. Pitt" "J. Roberts"	**"S. Soderbergh"** **"R. Howard"**	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	**"S. Soderbergh"** **"R. Howard"**	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"

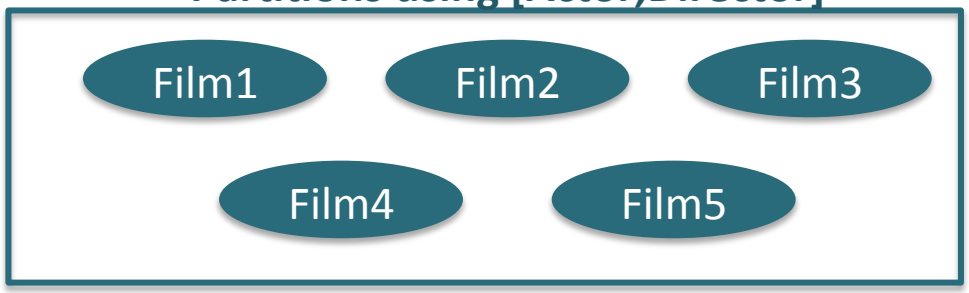
Partitions using [Director]

[Director]: pseudo key
Support = 4/5 =0.75
Discriminability = 2/3=0.66

[ADS12] - Key example

	Name	Actor	Director	ReleaseDate	Website	Language
film1	"Ocean's 11"	**"B. Pitt"** **"J. Roberts"**	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
film2	"Ocean's 12"	**"B. Pitt"** **"J. Roberts"**	**"S. Soderbergh"** **"R. Howard"**	"2/5/04"	www.oceans12.com	"english"
film3	"Ocean's 13"	"B. Pitt" "G. Clooney"	**"S. Soderbergh"** **"R. Howard"**	"30/6/07"	www.oceans13.com	"english"
film4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	---	"english"
film5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"

Partitions using [Actor,Director]



[Actor, Director]: key
Support = 4/5 =0.8
Discriminability = 5/5=1

Why pseudo-keys are useful?

Duplicates and errors

- Duplicates
  - No keys in this example

	Name	Actor	Director	ReleaseDate
Film1	“Intouchables”	“F.Cluzet” “O.Sy”	“O.Nakache” “E.Toledano”	“2/11/11”
Film2	“Intouchables”	“F.Cluzet” “O.Sy”	“O.Nakache” “E.Toledano”	“2/11/11”

- Erroneous data
  - {name, director} not a key using a strict key discovery method

	Name	Actor	Director	ReleaseDate
Film1	“Her”	“J.Phoenix” “S.Johansson”	“S.Jonze”	“10/1/14”
Film2	“Her”	“J.Lawrence”	“S.Jonze” “D.Russell”	“25/12/12”

Errors

Key Discovery Approaches

- **CWA approaches**
  - Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking
  - ROCKER: A Refinement Operator for Key Discovery

- **OWA approaches**
  - KD2R: a Key Discovery method for reference reconciliation
  - SAKey: Scalable almost key discovery in RDF data
  - Linkkey: Data interlinking through robust Linkkey extraction

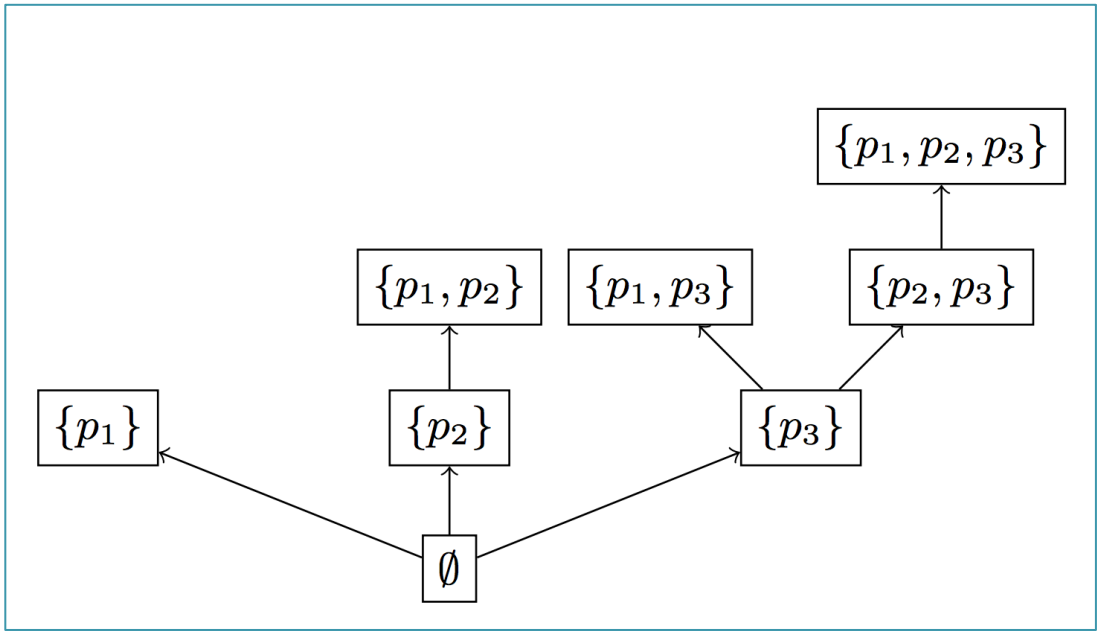
ROCKER [SMN15]

- Bottom-up approach that discovers in an efficient way
  - **Keys in the Closed World**
  - **Pseudo keys**

- Key quality measures
  - *Discriminant sets of properties*
    - ***Discriminability(P):*** # of instances that can be distinguished using the set *P*
    - ***Score(P)*** *= discriminability(P)/# instances*
      - Score values [0,1]
      - Key => **score = 1**
      - Pseudo key => **score < 1**

ROCKER [SMN15] - Steps

- Steps
  - Short properties in ascending order using their score
    - Ex. score(p1) = 1 , score(p2) = 0.9, score(p3) = 0.5



Closed World approaches
VS
Open World approaches

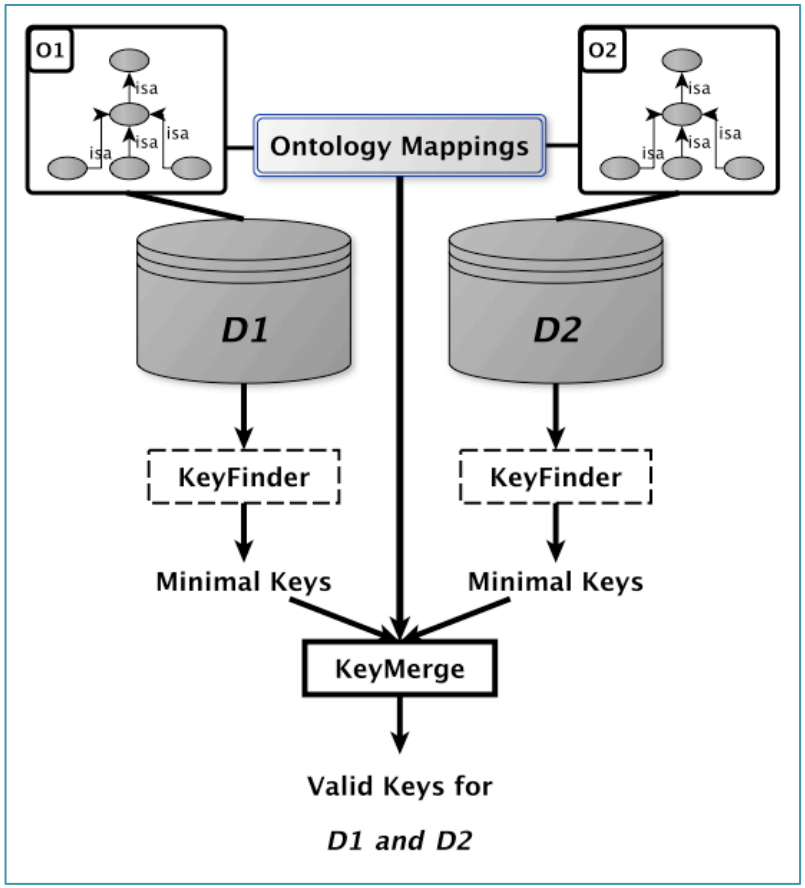
Key Discovery Approaches

- **CWA approaches**
  - Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking
  - ROCKER: A Refinement Operator for Key Discovery

- **OWA approaches**
  - **KD2R: a Key Discovery method for reference reconciliation**
  - SAKey: Scalable almost key discovery in RDF data
  - Linkkey: Data interlinking through robust Linkkey extraction

KD2R [PSS13]

- Key discovery approach
  - **Optimist keys**
  - **Pessimistic keys**



KD2R - Optimistic keys

- **Optimistic heuristic:**
  - only given values are considered

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3
i2	Tompson	Maria	
i3	David	George	i2, i4
i4	Solgar	Michel	

KD2R - Optimistic keys

- **Optimistic heuristic:**
  - only given values are considered

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3 ~~i4~~
i2	Tompson	Maria	~~i1, i3, i4~~
i3	David	George	i2, i4 ~~i1~~
i4	Solgar	Michel	~~i1, i2, i3~~

KD2R - Optimistic keys

- **Optimistic key**: set of properties that has a unique value for every instance described by this property set in the data

$$\forall X\ \forall Y\ ((X \neq Y) \wedge c(X) \wedge c(Y)) \Rightarrow (\forall Z\ \neg(p_j(X,Z) \wedge p_j(Y,Z)))$$

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3 ~~i4~~
i2	Tompson	Maria	~~i1, i3, i4~~
i3	David	George	i2, i4 ~~i1~~
i4	Solgar	Michel	~~i1, i2, i3~~

- **Keys**: {hasFriend, lastName}, {firstName}

KD2R - Pessimistic keys

- **Pessimistic heuristic:**
  - Not instantiated property ➔ value possibly one of the existing ones

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3
i2	Tompson	Maria	
i3	David	George	i2, i4
i4	Solgar	Michel	

KD2R - Pessimistic keys

- **Pessimistic heuristic:**
  - Not instantiated property ➔ value possibly one of the existing ones

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3
i2	Tompson	Maria	**i1, i3, i4**
i3	David	George	i2, i4
i4	Solgar	Michel	**i1, i2, i3**

KD2R - Pessimistic keys

- **Pessimistic heuristic:**
  - Not instantiated property ➔ value possibly one of the existing ones
  - Instantiated property ➔ only given values are considered

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3
i2	Tompson	Maria	**i1, i3, i4**
i3	David	George	i2, i4
i4	Solgar	Michel	**i1, i2, i3**

KD2R - Pessimistic keys

- **Pessimistic heuristic:**
  - Not instantiated property ➔ value possibly one of the existing ones
  - Instantiated property ➔ only given values are considered

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3 ~~i4~~
i2	Tompson	Maria	**i1, i3, i4**
i3	David	George	i2, i4 ~~i1~~
i4	Solgar	Michel	**i1, i2, i3**

KD2R - Pessimistic keys

- **Pessimistic key:** set of properties that has a unique value for every instance of the data

$$\forall X\ \forall Y\ ((X \neq Y) \wedge c(X) \wedge c(Y)) \Rightarrow \exists p_j\ (\exists U\ \exists V\ p_j(X,U) \wedge p_j(Y,V)) \wedge (\forall Z\ \neg(p_j(X,Z) \wedge p_j(Y,Z)))$$

id	lastName	firstName	hasFriend
i1	Tompson	Manuel	i2, i3 ~~i4~~
i2	Tompson	Maria	**i1, i3, i4**
i3	David	George	i2, i4 ~~i1~~
i4	Solgar	Michel	**i1, i2, i3**

- **Keys**: {firstName}

KD2R - Algorithm

- Non-key discovery first
  - Partially scan the data

	museumName	museumAddress	inCountry
Museum1	"Archaeological Museum"	"44 Patission Street"	"Greece"
Museum2	"Pompidou"		"France"
Museum3	"Musée d'Orsay"	"62, rue de Lille"	"France"
Museum4	"Madame Tussauds"	"Marylebone Road"	"England"
Museum5	"Vatican Museums"	"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "	"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"	"Archea Olympia"	"Greece"
Museum8	"Dalí museum"	"1, Dali Boulevard"	"Spain"

KD2R - Algorithm

- Non-key discovery first
  - Partially scan the data

	museumName	museumAddress	inCountry
Museum1	"Archaeological Museum"	"44 Patission Street"	"Greece"
Museum2	"Pompidou"		"France"
Museum3	"Musée d'Orsay"	"62, rue de Lille"	"France"
Museum4	"Madame Tussauds"	"Marylebone Road"	"England"
Museum5	"Vatican Museums"	"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "	"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"	"Archea Olympia"	"Greece"
Museum8	"Dalí museum"	"1, Dali Boulevard"	"Spain"

KD2R - Algorithm

- Non-key discovery first
  - Partially scan the data

key

	museumName	museumAddress	inCountry
Museum1	"Archaeological Museum"	"44 Patission Street"	"Greece"
Museum2	"Pompidou"		"France"
Museum3	"Musée d'Orsay"	"62, rue de Lille"	"France"
Museum4	"Madame Tussauds"	"Marylebone Road"	"England"
Museum5	"Vatican Museums"	"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "	"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"	"Archea Olympia"	"Greece"
Museum8	"Dalí museum"	"1, Dali Boulevard"	"Spain"

KD2R - Algorithm

- Non-key discovery first
  - Partially scan the data

key | non-key

	museumName	museumAddress	inCountry
Museum1	"Archaeological Museum"	"44 Patission Street"	"Greece"
Museum2	"Pompidou"		**"France"**
Museum3	"Musée d'Orsay"	"62, rue de Lille"	**"France"**
Museum4	"Madame Tussauds"	"Marylebone Road"	"England"
Museum5	"Vatican Museums"	"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "	"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"	"Archea Olympia"	"Greece"
Museum8	"Dalí museum"	"1, Dali Boulevard"	"Spain"

- Interested only in maximal non keys

KD2R - Algorithm

- Non-key discovery first
  - Partially scan the data

key | non-key

	museumName	museumAddress	inCountry
Museum1	"Archaeological Museum"	"44 Patission Street"	"Greece"
Museum2	"Pompidou"		**"France"**
Museum3	"Musée d'Orsay"	"62, rue de Lille"	**"France"**
Museum4	"Madame Tussauds"	"Marylebone Road"	"England"
Museum5	"Vatican Museums"	"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "	"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"	"Archea Olympia"	"Greece"
Museum8	"Dalí museum"	"1, Dali Boulevard"	"Spain"

- Interested only in maximal non keys
  - All the sets of properties that are not maximal non keys are keys

KD2R - Algorithm

- Non-key discovery first
  - Partially scan the data

key | non-key

	museumName	museumAddress	inCountry
Museum1	"Archaeological Museum"	"44 Patission Street"	"Greece"
Museum2	"Pompidou"		**"France"**
Museum3	"Musée d'Orsay"	"62, rue de Lille"	**"France"**
Museum4	"Madame Tussauds"	"Marylebone Road"	"England"
Museum5	"Vatican Museums"	"Piazza San Giovanni"	"Italy"
Museum6	"Deutsches Museum "	"Museumsinsel 1"	"Germany"
Museum7	"Olympia Museum"	"Archea Olympia"	"Greece"
Museum8	"Dalí museum"	"1, Dali Boulevard"	"Spain"

- Interested only in maximal non keys
  - All the sets of properties that are not maximal non keys are keys
    - **Example:** class described by the properties p1, p2, p3, p4

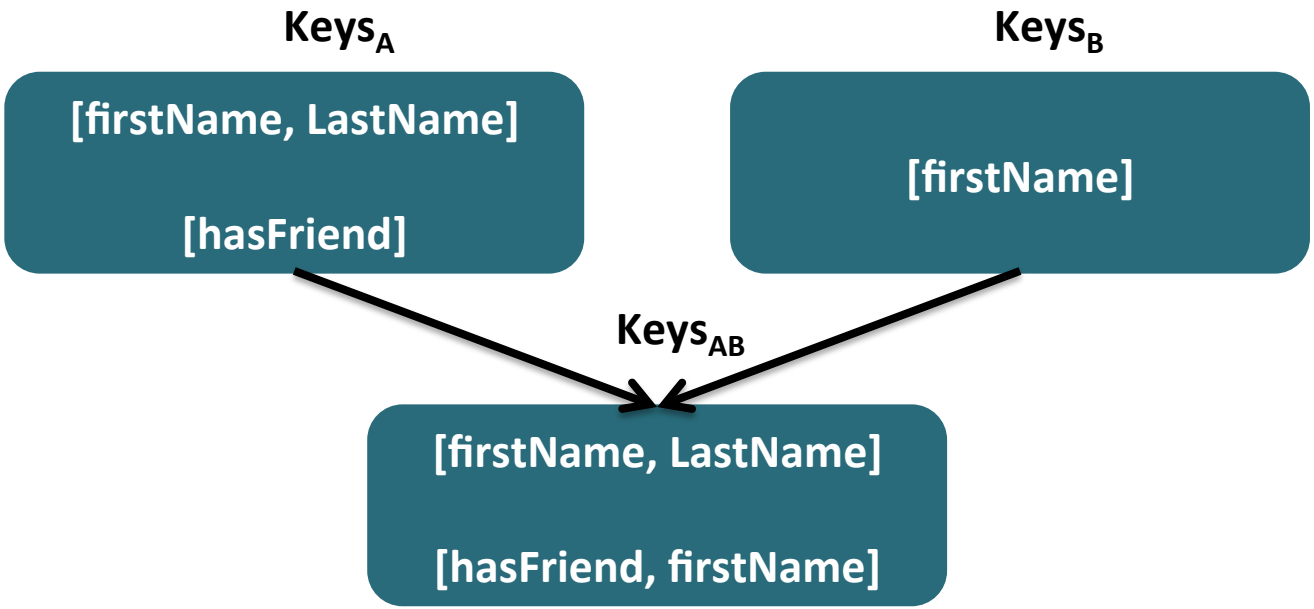
Maximal non key = [[p1, p2]] → keys = [[p3], [p4]]

Key merge

- **Goal**: Keys valid in both datasets
  - More sure keys
- **Intuition**: Computation of Cartesian product of sets of keys
  - Keep only minimal keys

Key merge

- **Goal**: Keys valid in both datasets
  - More sure keys
- **Intuition**: Computation of Cartesian product of sets of keys
  - Keep only minimal keys



Key Discovery Approaches

- **CWA approaches**
  - Keys and Pseudo-Keys Detection for Web Datasets Cleaning and Interlinking
  - ROCKER: A Refinement Operator for Key Discovery

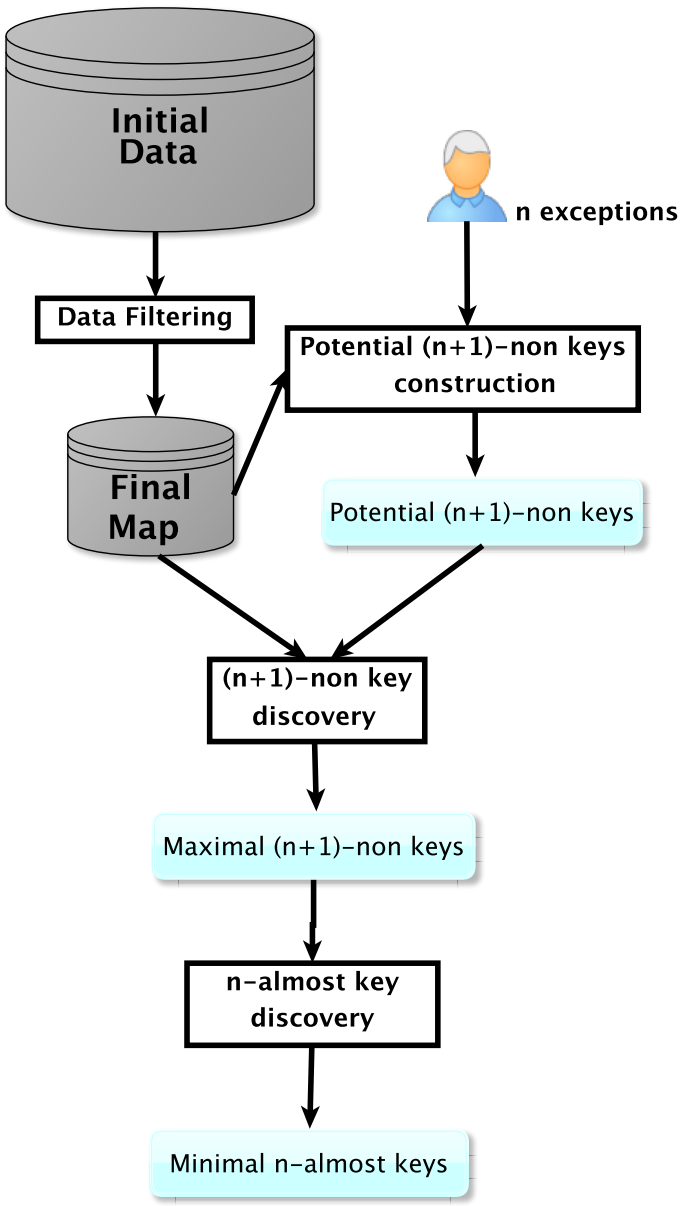
- **OWA approaches**
  - KD2R: a Key Discovery method for reference reconciliation
  - **SAKey: Scalable almost key discovery in RDF data**
  - Linkkey: Data interlinking through robust Linkkey extraction

SAKey [SAPS14]

- SAKey: Scalable Almost Key discovery approach for:
  - Incomplete data (optimistic heuristic)
  - Errors
  - Duplicates
  - Large datasets

- Discovers ***almost keys***
  - Sets of properties that are not keys due to few exceptions

SAKey - General Architecture



Key discovery

- **SAKey:** Key discovery approach for very large RDF datasets that may contain erroneous data or duplicates
  - ***n*-almost key:** A set of properties for which at most *n* instances are identical values

	Region	**Producer**	**Colour**
Wine1	Bordeaux	Dupont	White
Wine2	Bordeaux	Baudin	Rose
Wine3	Languedoc	Dupont	Red
Wine4	Languedoc	Faure	Red

- Examples of keys

  {Region, Producer}:

  0-almost key

Key discovery

- **SAKey:** Key discovery approach for very large RDF datasets that may contain erroneous data or duplicates
  - ***n*-almost key:** A set of properties for which at most *n* instances are identical values

	Region	**Producer**	**Colour**
Wine1	Bordeaux	**Dupont**	White
Wine2	Bordeaux	Baudin	Rose
Wine3	Languedoc	**Dupont**	Red
Wine4	Languedoc	Faure	Red

- Examples of keys

  {Region, Producer}:

  0-almost key

  {Producer}:

  2-almost key

Tool available in https://www.lri.fr/sakey

n-almost keys

- **Exception of a key:** an instance that shares values with another instance for a given set of properties *P*

Films	HasName	HasActor	HasDirector	ReleaseDate	HasWebsite	HasLanguage
f1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
f2	"Ocean's 12"	"B. Pitt" "G. Clooney" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	---
f3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	---
f4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	www.descendants.com	"english"
f5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"
f6	"Ocean's 12"	---	"R. Howard"	"2/5/04"	---	---

n-almost keys

- **Exception of a key:** an instance that shares values with another instance for a given set of properties *P*
  - *f1, f2 and f3* are three exceptions for the property set {HasActor}

Films	HasName	HasActor	HasDirector	ReleaseDate	HasWebsite	HasLanguage
f1	"Ocean's 11"	**"B. Pitt"** "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
f2	"Ocean's 12"	**"B. Pitt"** "G. Clooney" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	---
f3	"Ocean's 13"	**"B. Pitt"** "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	---
f4	"The descendants"	"N. Krause" "G. Clooney"	"A. Payne"	"15/9/11"	www.descendants.com	"english"
f5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"
f6	"Ocean's 12"	---	"R. Howard"	"2/5/04"	---	---

n-almost keys

- **Exception of a key:** an instance that shares values with another instance for a given set of properties *P*
  - *f1, f2 and f3* are three exceptions for the property set {HasActor}
- **Exception Set E_P:** set of exceptions for *P*
  - E_P = {f1, f2, f3} ∪ {f2, f3, f4} = {f1, f2, f3, f4} for {HasActor}

Films	HasName	HasActor	HasDirector	ReleaseDate	HasWebsite	HasLanguage
f1	"Ocean's 11"	**"B. Pitt"** "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
f2	"Ocean's 12"	**"B. Pitt"** **"G. Clooney"** "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	---
f3	"Ocean's 13"	**"B. Pitt"** **"G. Clooney"**	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	---
f4	"The descendants"	"N. Krause" **"G. Clooney"**	"A. Payne"	"15/9/11"	www.descendants.com	"english"
f5	"Bourne Identity"	"D. Liman"	---	"12/6/12"	www.bourneIdentity.com	"english"
f6	"Ocean's 12"	---	"R. Howard"	"2/5/04"	---	---

n-almost keys

- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {HasActor} is a 4-almost key

n-almost keys

- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {HasActor} is a 4-almost key

- ***n*-non key:** a set of properties where $|E_P| \geq n$

n-almost keys

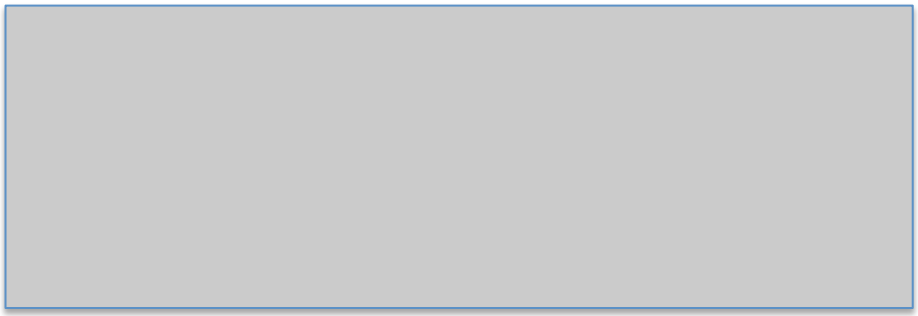
- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {HasActor} is a 4-almost key

- ***n*-non key:** a set of properties where $|E_P| \geq n$
  - Using all the maximal *n*-non keys we can derive all the minimal (*n*-1)-almost keys

n-almost keys

- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {HasActor} is a 4-almost key

- ***n*-non key:** a set of properties where $|E_P| \geq n$
  - Using all the maximal *n*-non keys we can derive all the minimal (*n*-1)-almost keys



All combinations of properties

n-almost keys

- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {HasActor} is a 4-almost key

- ***n*-non key:** a set of properties where $|E_P| \geq n$
  - Using all the maximal *n*-non keys we can derive all the minimal (*n*-1)-almost keys

All combinations of properties

All sets of properties that contain at least 5 exceptions

n-almost keys

- ***n*-almost key:** a set of properties where $|E_P| \leq n$
  - {HasActor} is a 4-almost key

- ***n*-non key:** a set of properties where $|E_P| \geq n$
  - Using all the maximal *n*-non keys we can derive all the minimal (*n*-1)-almost keys

All combinations of properties

All sets of properties that contain at least 5 exceptions

All sets of properties that contain less than 5 exceptions

n-non key discovery: Initial Map

n-non key discovery: Initial Map

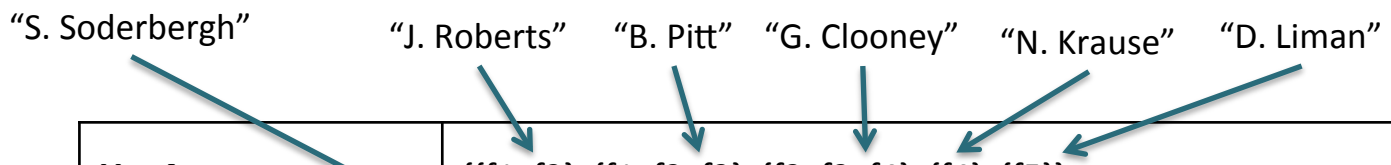
Films	HasName	HasActor	HasDirector	ReleaseDate	HasWebsite	HasLanguage
f1	"Ocean's 11"	"B. Pitt" "J. Roberts"	"S. Soderbergh"	"3/4/01"	www.oceans11.com	---
f2	"Ocean's 12"	"B. Pitt" "G. Clooney" "J. Roberts"	"S. Soderbergh" "R. Howard"	"2/5/04"	www.oceans12.com	---
f3	"Ocean's 13"	"B. Pitt" "G. Clooney"	"S. Soderbergh" "R. Howard"	"30/6/07"	www.oceans13.com	---
…						

"S. Soderbergh" "J. Roberts" "B. Pitt" "G. Clooney" "N. Krause" "D. Liman"

HasActor	{{f1, f2}, {f1, f2, f3}, {f2, f3, f4}, {f4}, {f5}}
HasDirector	{{f1,f2,f3}, {f2, f3, f6}, {f4}}
ReleaseDate	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasName	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasLanguage	{{f4, f5}}
HasWebsite	{{f1}, {f2}, {f3}, {f4}, {f5}, {f6}}

n-non key discovery: Data filtering

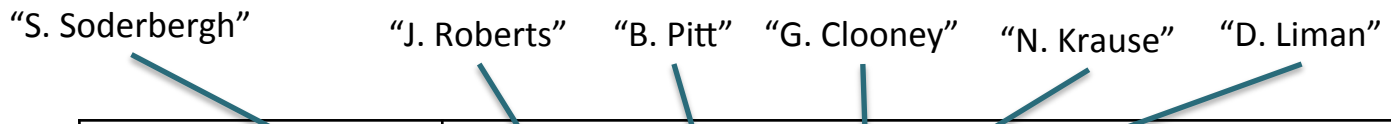
- Singleton filtering



HasActor	{{f1, f2}, {f1, f2, f3}, {f2, f3, f4}, {f4}, {f5}}
HasDirector	{{f1,f2,f3}, {f2, f3, f6}, {f4}}
ReleaseDate	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasName	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasLanguage	{{f4, f5}}
HasWebsite	{{f1}, {f2}, {f3}, {f4}, {f5}, {f6}}

n-non key discovery: Data filtering

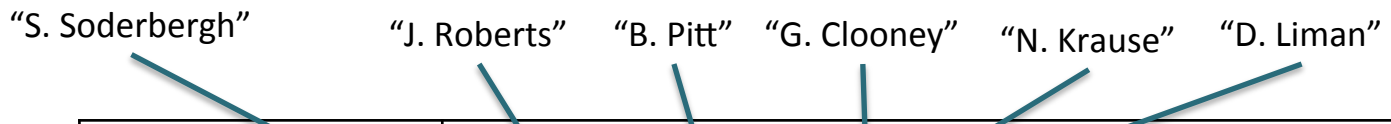
- Singleton filtering



HasActor	{{f1, f2}, {f1, f2, f3}, {f2, f3, f4}, {f4}, {f5}}
HasDirector	{{f1,f2,f3}, {f2, f3, f6}, {f4}}
ReleaseDate	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasName	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasLanguage	{{f4, f5}}
HasWebsite	{{f1}, {f2}, {f3}, {f4}, {f5}, {f6}}

n-non key discovery: Data filtering

- Singleton filtering



HasActor	{{f1, f2}, {f1, f2, f3}, {f2, f3, f4}, {f4}, {f5}}
HasDirector	{{f1,f2,f3}, {f2, f3, f6}, {f4}}
ReleaseDate	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasName	{{f1}, {f2, f6}, {f3}, {f4}, {f5}}
HasLanguage	{{f4, f5}}
HasWebsite	{{f1}, {f2}, {f3}, {f4}, {f5}, {f6}}

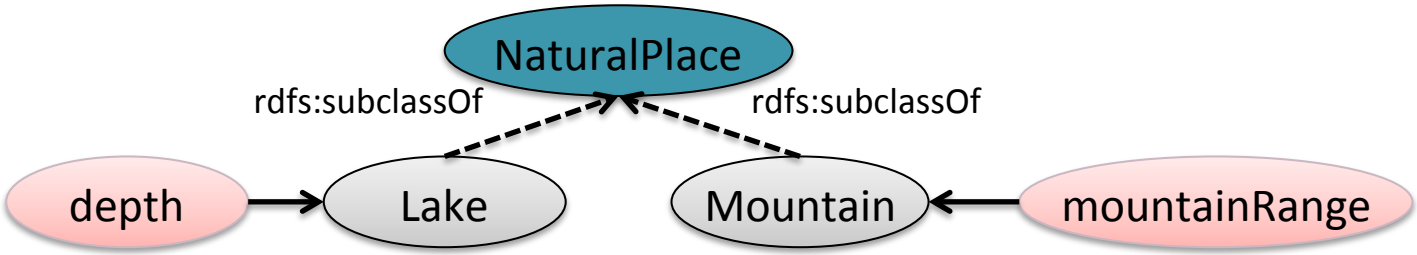
n-non key discovery: Data filtering

- Singleton filtering

HasActor	{{f1, f2, f3}, {f2, f3, f4}}
HasDirector	{{f1,f2,f3}, {f2, f3, f6}}
ReleaseDate	{{f2, f6}}
HasName	{{f2, f6}}
HasLanguage	{{f4, f5}}

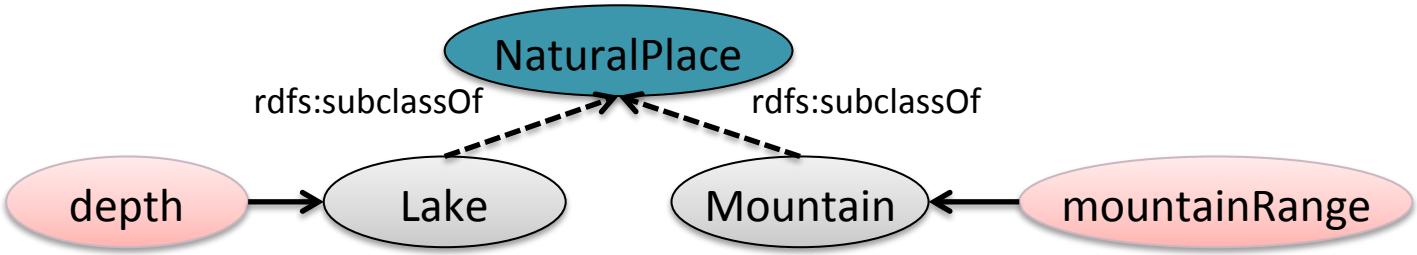
n-non key discovery: Potential *n*-non keys

- Combinations of properties not needed be explored
  - Incomplete data
  - Properties referring to different classes



n-non key discovery: Potential *n*-non keys

- Combinations of properties not needed be explored
  - Incomplete data
  - Properties referring to different classes



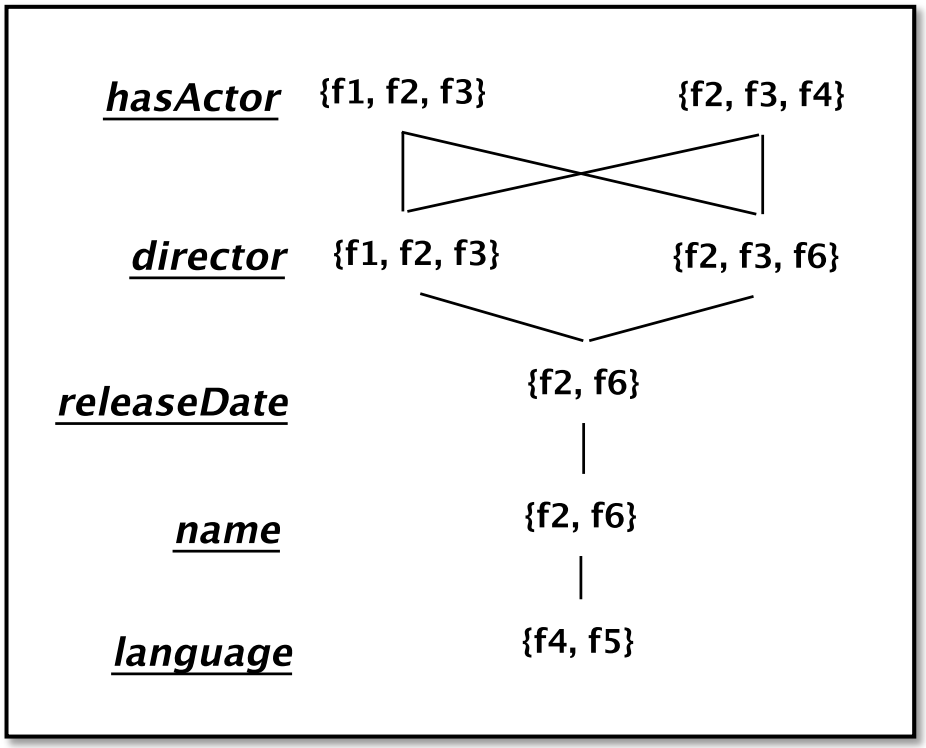
- **Potential *n*-non keys**: Sets of properties that possibly refer to *n*-non keys
  - Greedy algorithm

n-non key discovery

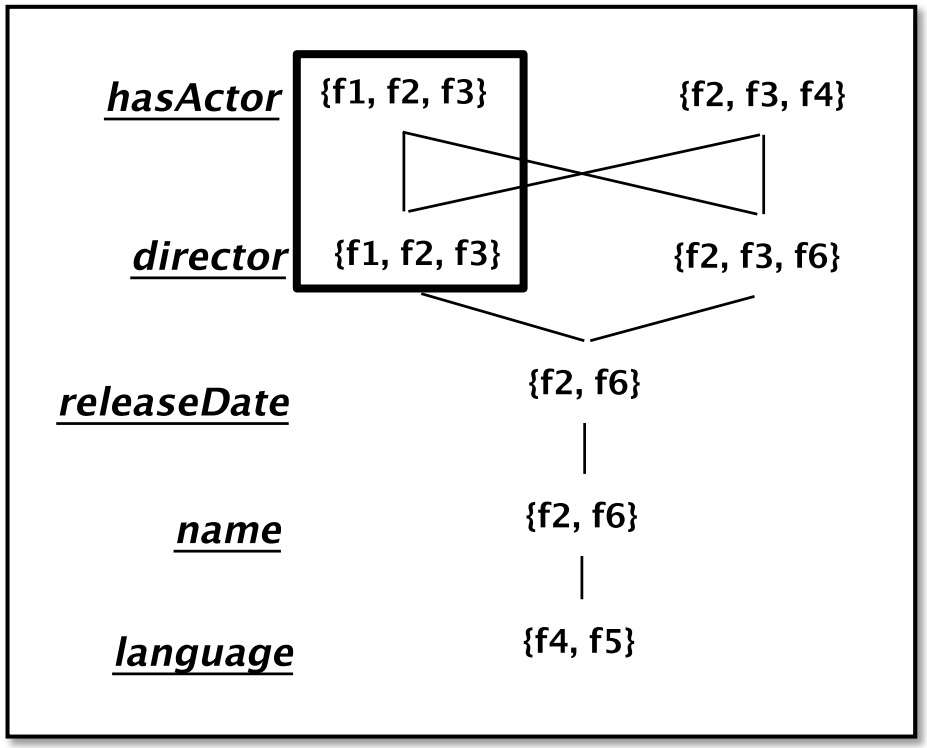
- HasActor
  - {f1, f2, f3} ∪ {f2, f3, f4} = {f1, f2, f3, f4} => 4-non key
- Composite *n*-non keys
  - Intersections between sets of different properties

HasActor	{{f1, f2, f3}, {f2, f3, f4}}
HasDirector	{{f1,f2,f3}, {f2, f3, f6}}
ReleaseDate	{{f2, f6}}
HasName	{{f2, f6}}
HasLanguage	{{f4, f5}}

n-non key discovery

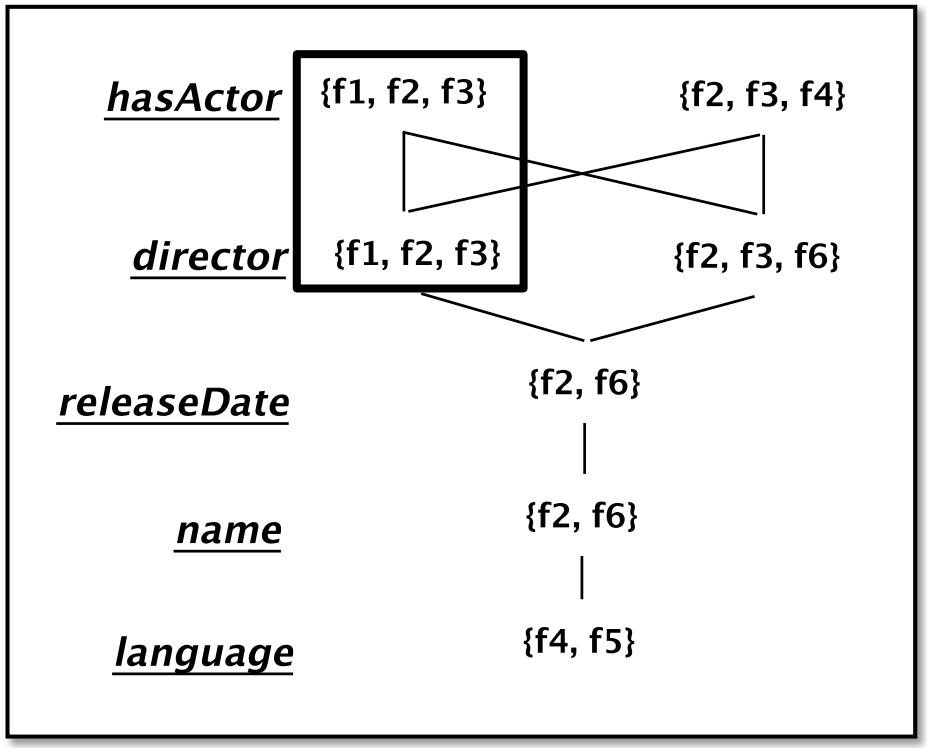


n-non key discovery

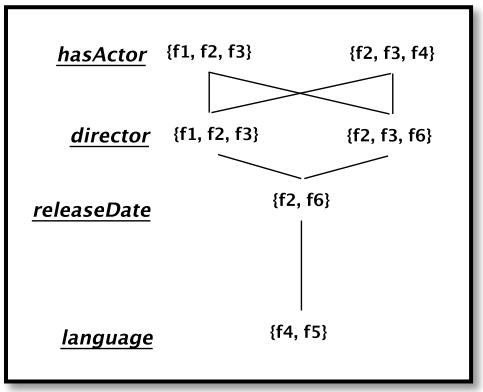


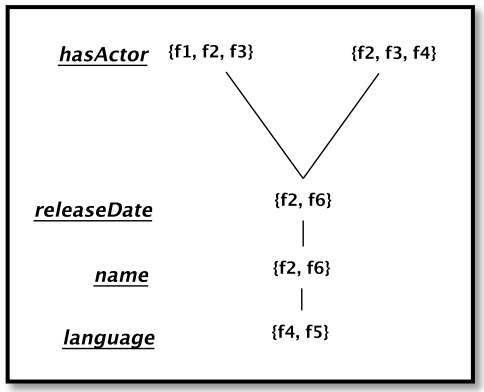
{hasActor, director} ➔ 3-almost key

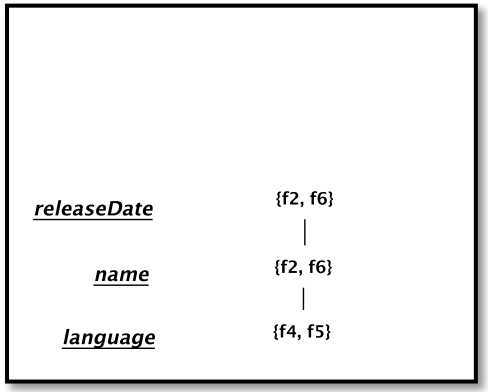
n-non key discovery



{hasActor, director} ➔ 3-almost key







...

Data Linking using almost keys

- **Goal**: Compare linking results using almost keys with different *n*
- Evaluation of linking using
  - Recall
  - Precision
  - F-Measure
- Datasets
  - OAEI 2010
  - OAEI 2013
- Conclusion
  - Linking results using *n*-almost keys are the better than using keys

Example: Data Linking using almost keys

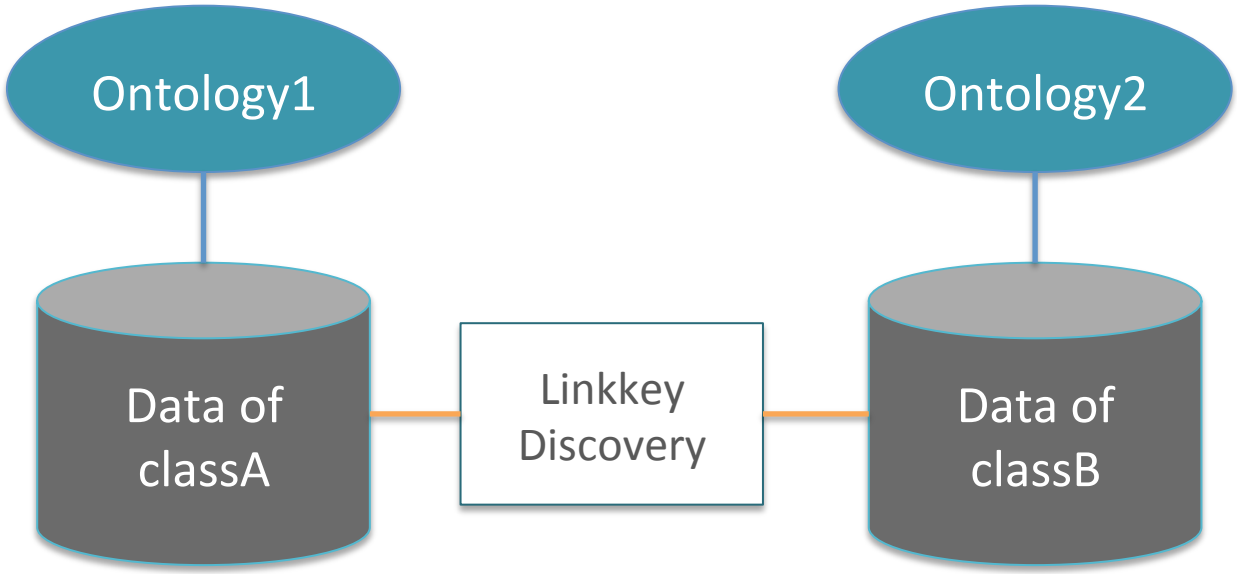
- OAEI 2013 - Person
  - BirthName, BirthDate, award, comment, label, BirthPlace, almaMater, doctoralAdvisor

	Almost keys	Recall	Precision	F-Measure
0-almost key	{BirthDate, award}	9.3%	100%	17%
2-almost key	{BirthDate}	32.5%	98.6%	49%

# exceptions	Recall	Precision	F-measure
0, 1	25.6%	100%	41%
2, 3	47.6%	98.1%	64.2%
4, 5	47.9%	96.3%	63.9%
6, ..., 16	48.1%	96.3%	64.1%
17	49.3%	82.8%	61.8%

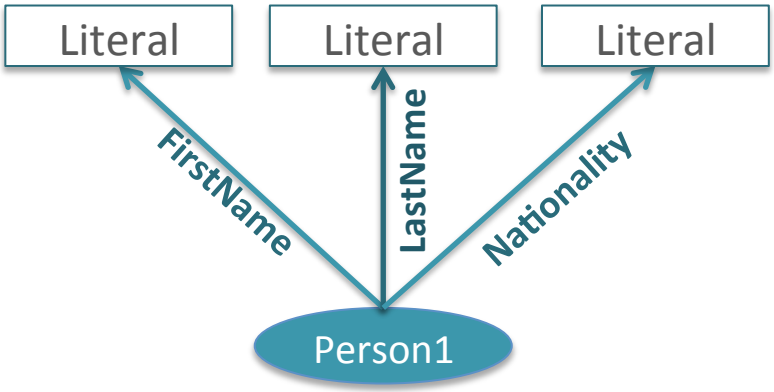
Linkkey [ADE14]

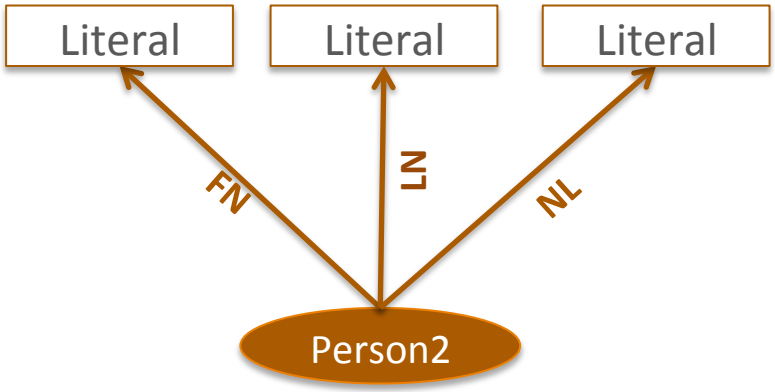
- Given a **pair of classes** in two datasets conforming to two different ontologies:
  - Discover ***Linkkeys*** – *sets of property pairs that can link instances of two different classes*



Linkkey [ADE14]

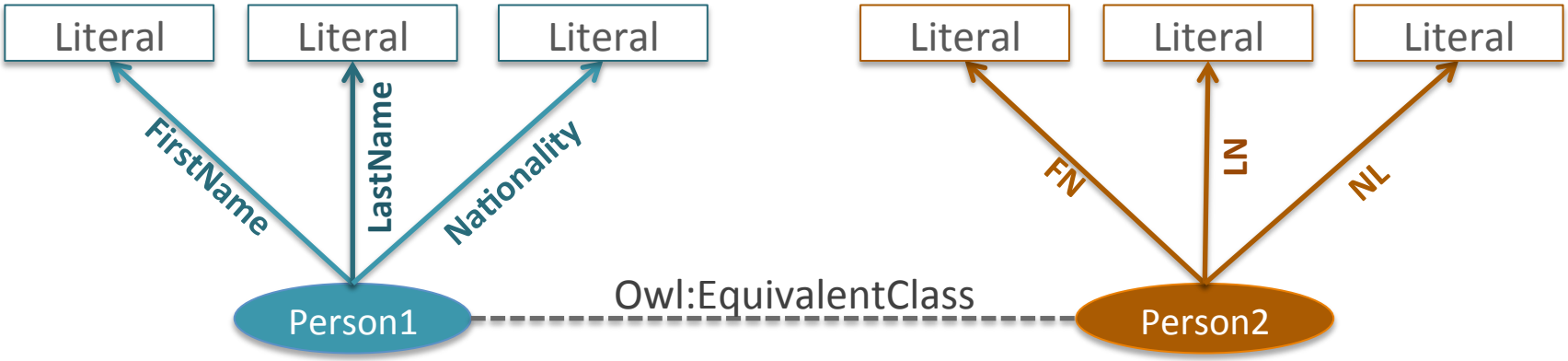
- Given a **pair of classes** in two datasets conforming to two different ontologies:
  - Discover ***Linkkeys*** – *sets of property pairs that can link instances of two different classes*





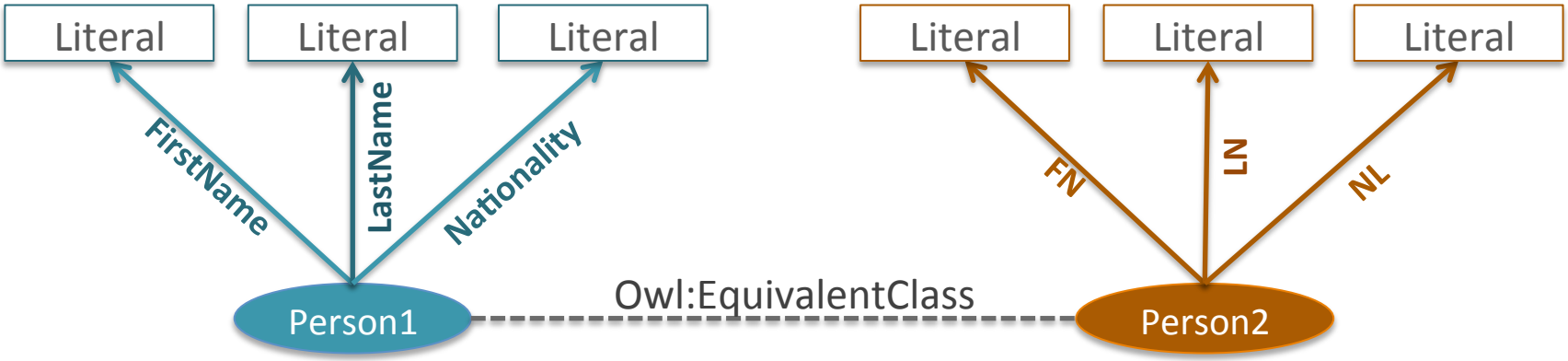
Linkkey [ADE14]

- Given a **pair of classes** in two datasets conforming to two different ontologies:
  - Discover ***Linkkeys*** – *sets of property pairs that can link instances of two different classes*



Linkkey [ADE14]

- Given a **pair of classes** in two datasets conforming to two different ontologies:
  - Discover ***Linkkeys*** – *sets of property pairs that can link instances of two different classes*

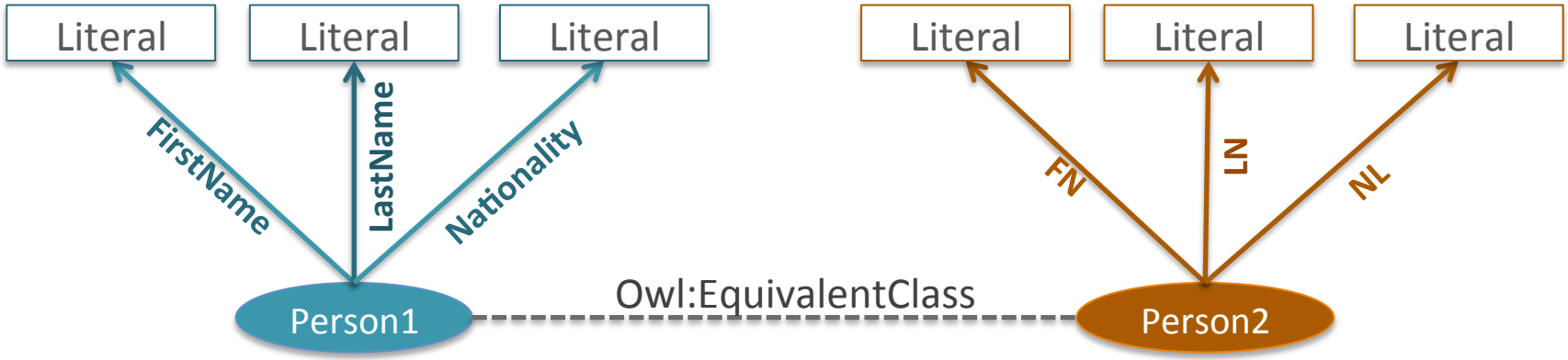


	LastName	Nationality	Profession
P11	Tompson	Greek	
P12	Dupont	French	Researcher

	LN	NL	PR
P21	Tompson	Greek	
P22	Tompson	Greek, French	Reseacher

Linkkey [ADE14]

- Given a **pair of classes** in two datasets conforming to two different ontologies:
  - Discover ***Linkkeys*** – *sets of property pairs that can link instances of two different classes*



	LastName	Nationality	Profession
P11	Tompson	Greek	
P12	Dupont	French	Researcher

	LN	NL	PR
P21	Tompson	Greek	
P22	Tompson	Greek, French	Reseacher

Ex. {<LastName,LN>,<Nationality,NL>}, {<Profession,PR>,<Nationality,NL>}

Linkkey [ADE14]

	LastName	Nationality	Profession
P11	Tompson	Greek	
P12	Dupont	French	Researcher

	LN	NL	PR
P21	Tompson	Greek	
P22	Tompson	Greek, French	Reseacher

Tompson: <P11, LastName>

Dupont: <P12, LastName>

Greek: <P11, Nationality>

French: <P12, Nationality>

Researcher: <P12, Profession>

Tompson: <P21, LastName>
<P22, LastName>

Greek: <P21, Nationality>
<P22, Nationality>

French: <P22, Nationality>

Researcher: <P21, Profession>

<P11,P12> => {LastName, LN}, {Nationality, NL}
<P11,P22> => {LastName, LN}, {Nationality, NL}

<P12,P22> => {Proffesion, PR}, {Nationality, NL}

Linkkey [ADE14]

	LastName	Nationality	Profession
P11	Tompson	Greek	
P12	Dupont	French	Researcher

	LN	NL	PR
P21	Tompson	Greek	
P22	Tompson	Greek, French	Reseacher

Tompson: <P11, LastName>

Dupont: <P12, LastName>

Greek: <P11, Nationality>

French: <P12, Nationality>

Researcher: <P12, Profession>

Tompson: <P21, LastName>
<P22, LastName>

Greek: <P21, Nationality>
<P22, Nationality>

French: <P22, Nationality>

Researcher: <P21, Profession>

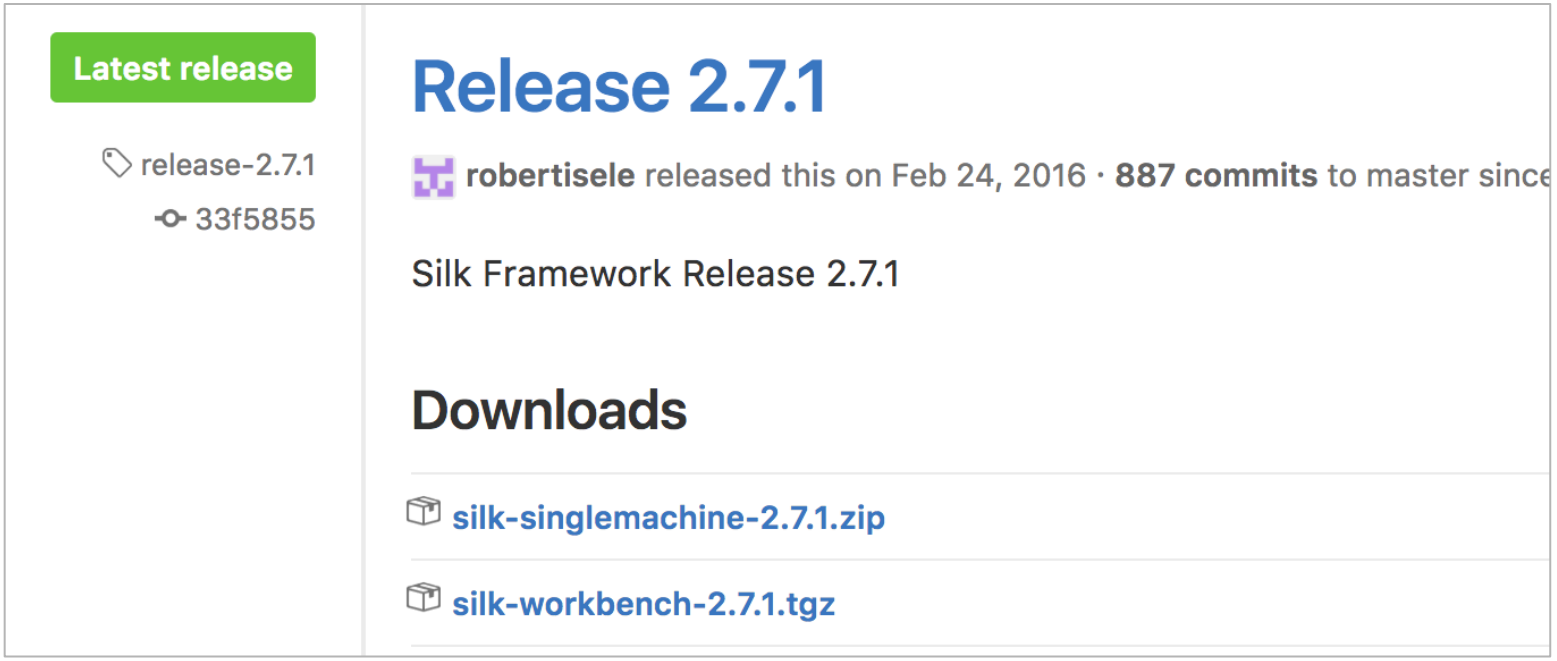
<P11,P12> => {LastName, LN}, {Nationality, NL}
<P11,P22> => {LastName, LN}, {Nationality, NL}

<P12,P22> => {Proffesion, PR}, {Nationality, NL}

SAKey and SILK

- SAKey
  - *Tool available in https://www.lri.fr/sakey*

- SILK - Graphical interface version
  - *Tool avalable in https://github.com/silk-framework/silk/releases*



QUESTIONS???

References

- **[PSS13]** Nathalie Pernelle, Fatiha Saïs, and Danai Symeonidou. An automatic key discovery approach for data linking. *Journal of Web Semantics*, 2013.

- **[ADS12]** Manuel Atencia, Jérôme David, and François Scharffe. Keys and pseudo-keys detection for web datasets cleansing and interlinking. *EKAW*, 2012.

- **[SAS11]** Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *VLDB*, 2011.

- **[SMN15]** Soru, Tommaso, Edgard Marx, and Axel-Cyrille Ngonga Ngomo. ROCKER: a refinement operator for key discovery. In *WWW*, 2015.

- **[ADE14]** Atencia, Manuel, Jérôme David, and Jérôme Euzenat. Data interlinking through robust Linkkey extraction. *ECAI,* 2014.

- **[SAPS14]** Symeonidou, Danai, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. "SAKey: Scalable almost key discovery in RDF data." In International Semantic Web Conference, pp. 33-49. Springer International Publishing, 2014.