

# A Scalable Approach to Incrementally Building Knowledge Graphs

Gleb Gawriljuk (KIT), Andreas Harth (KIT), Craig A. Knoblock (USC), Pedro Szekely (USC)

INSTITUTE AIFB, CHAIRS OF KNOWLEDGE MANAGEMENT AND WEB SCIENCE

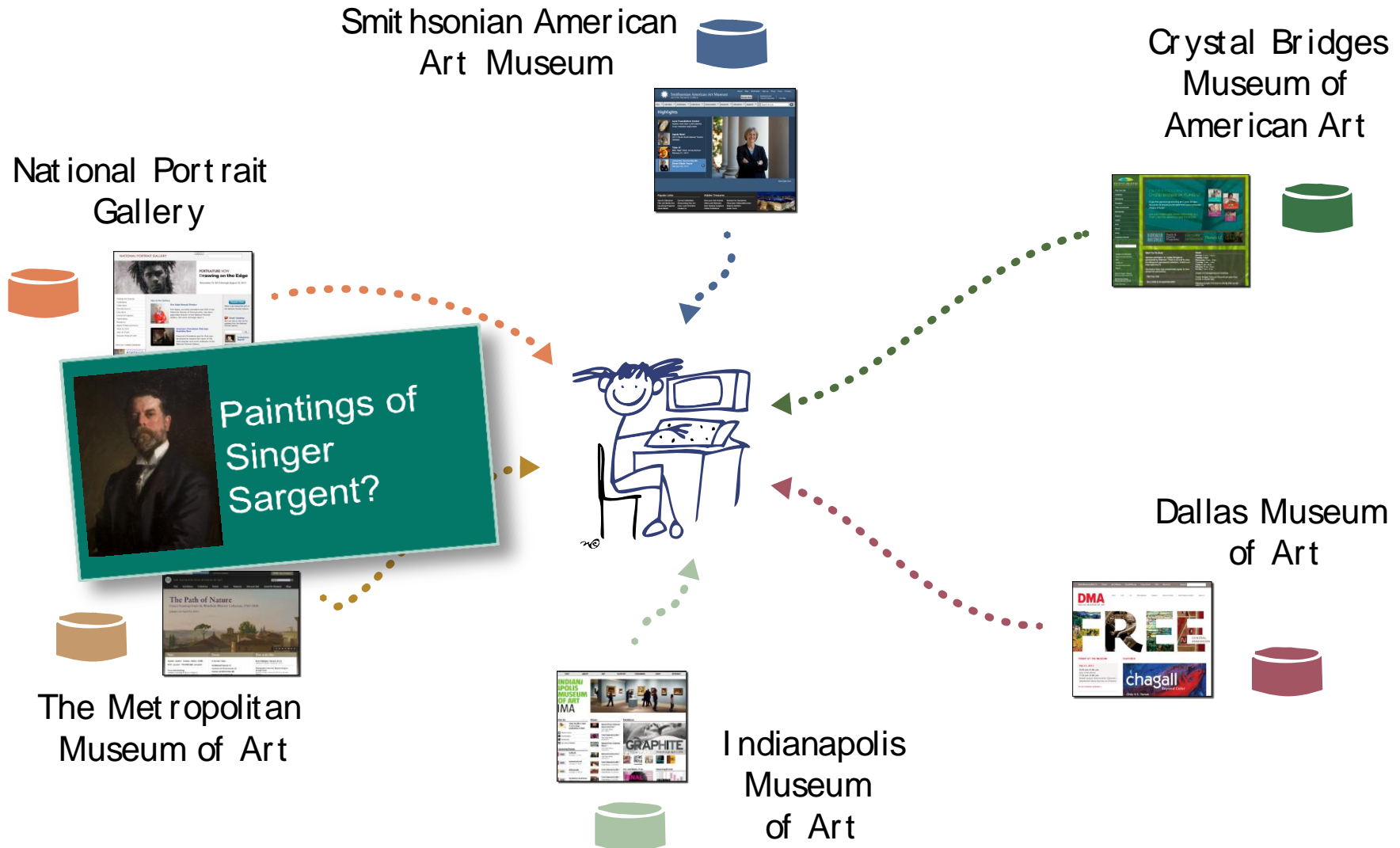


[http://www.imageduplicator.com/main.php?decade=70&year=79&work\\_id=1042](http://www.imageduplicator.com/main.php?decade=70&year=79&work_id=1042)

# Outline

- **Motivation**
- Overview of Approach
- Building and Extending a Knowledge Graph
- Evaluation
- Conclusion

# Current State of Cultural Heritage Data: Get Info from Web Pages



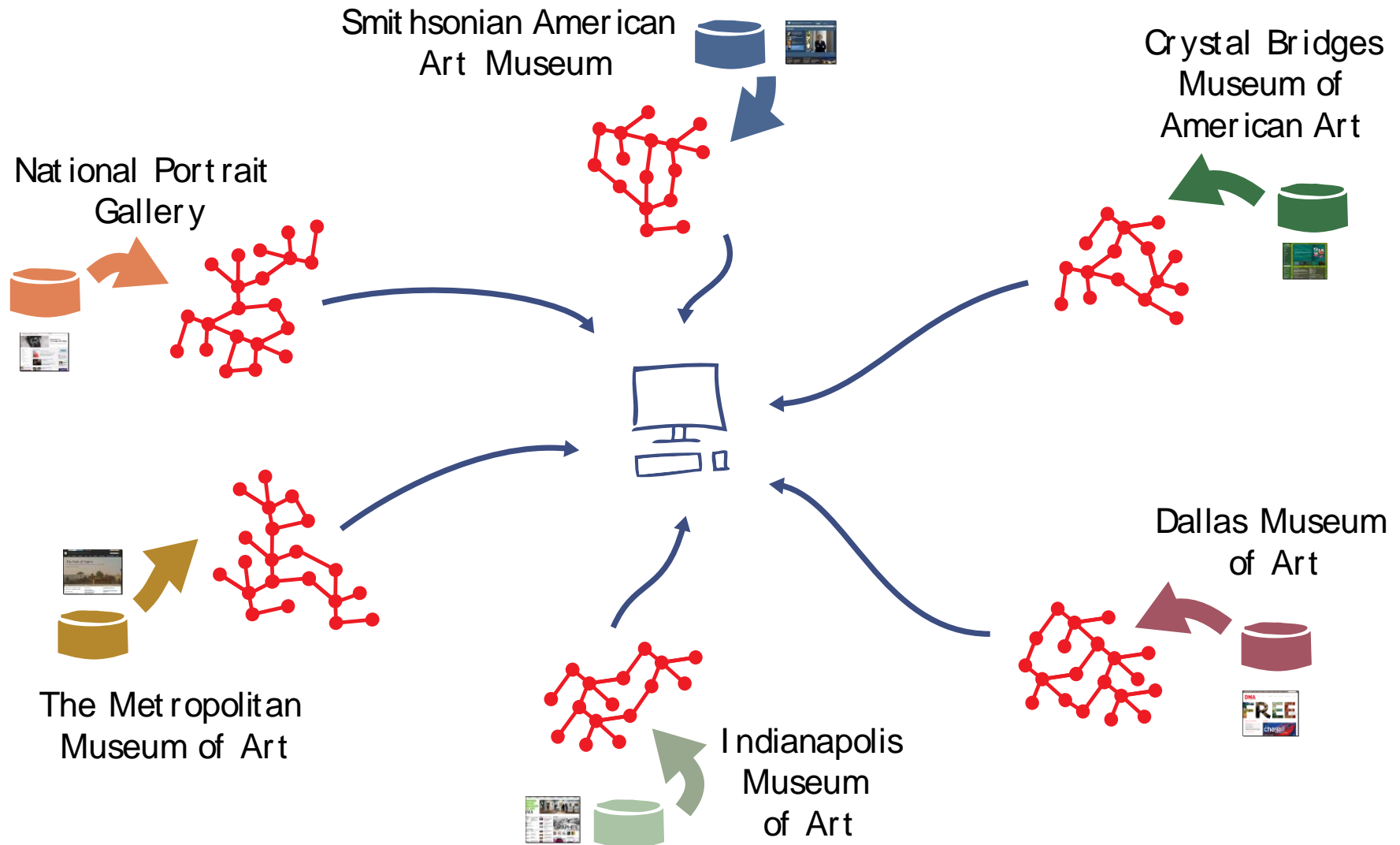
# Problem

web pages are machine processable,  
but not machine understandable

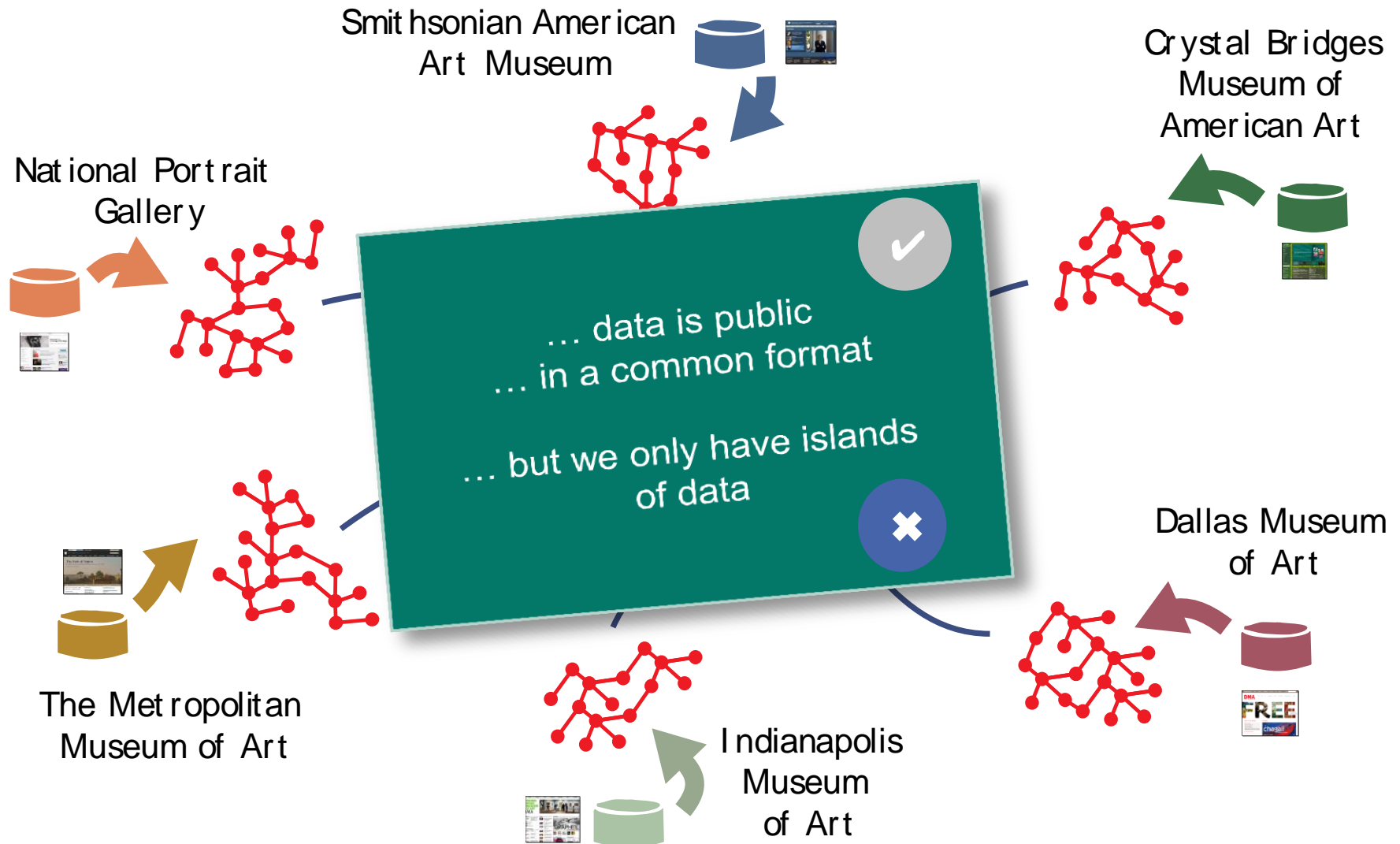
impractical for building applications using the data

publish the data as Linked Open Data

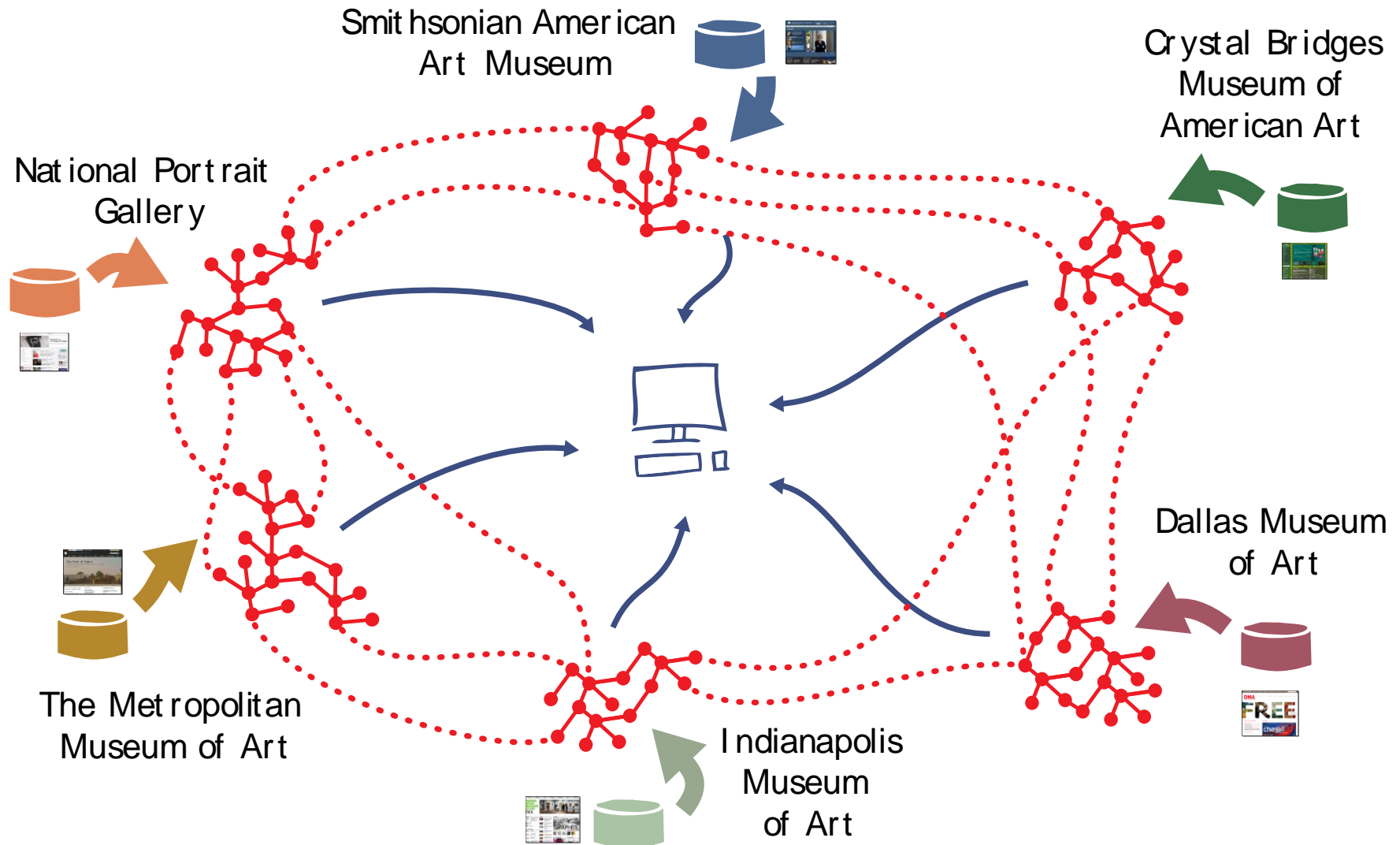
# Cultural Heritage “Linked” Open Data



# Cultural Heritage “Linked” Open Data

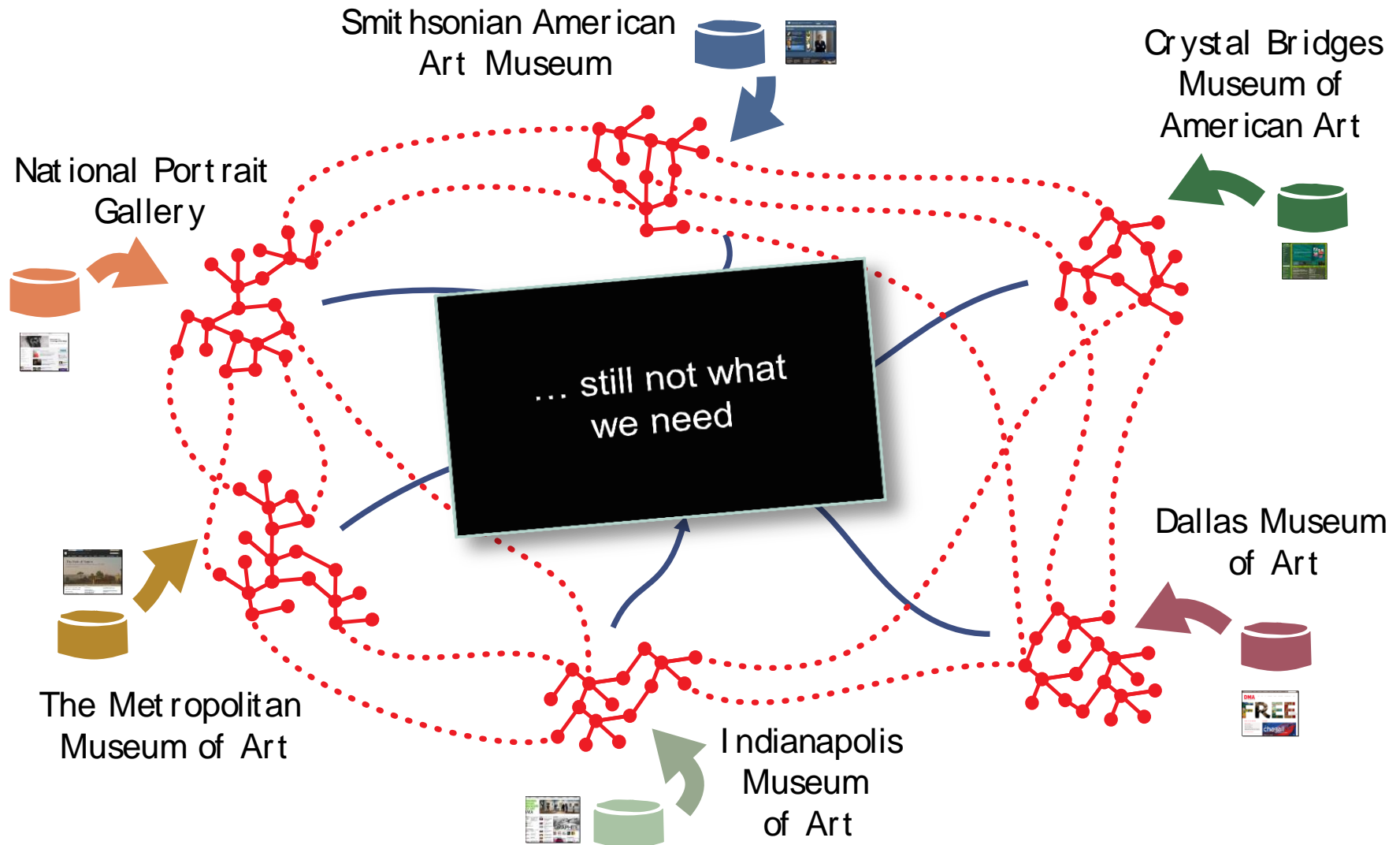


# Cultural Heritage Linked Open Data

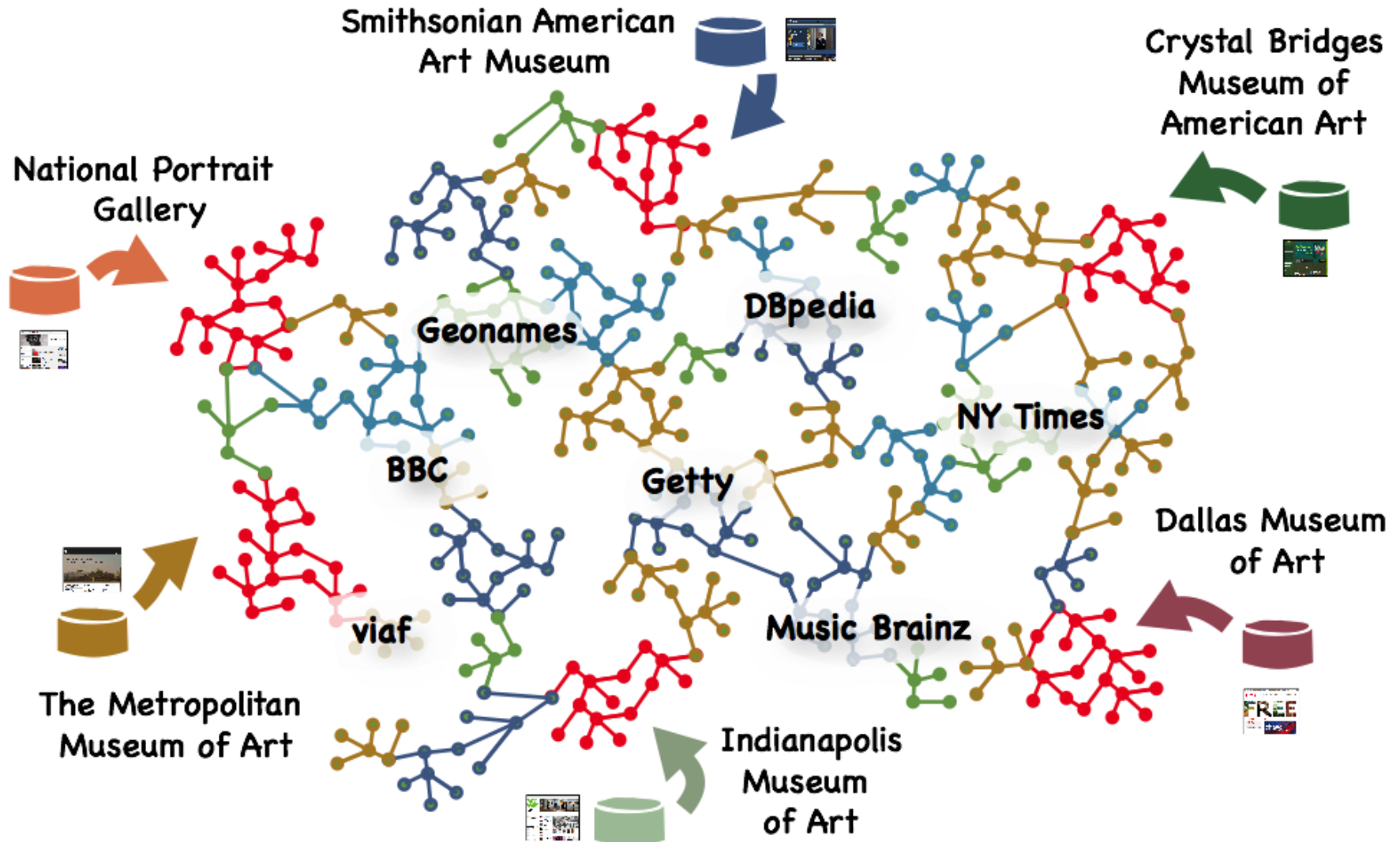




# Cultural Heritage Linked Open Data



# Linked Open Data



# Integrated Querying based on owl:sameAs Links

<http://d-nb.info/gnd/118547739>

<http://id.loc.gov/authorities/names/n50019335>



<http://viaf.org/viaf/12466780>

[http://dbpedia.org/resource/John\\_Singer\\_Sargent](http://dbpedia.org/resource/John_Singer_Sargent)

<http://www.wikidata.org/entity/Q155626>

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
SELECT ?object ?title ?picture WHERE {
  dbpedia:John_Singer_Sargent foaf:made ?object
  ?object dc:title ?title .
  ?object foaf:depiction ?picture .
}
```

Link traversal query processing with reasoning yields 16 results out of ~2900 paintings on data from the LOD cloud as of Jan 2015

# Steps to Create Linked Data

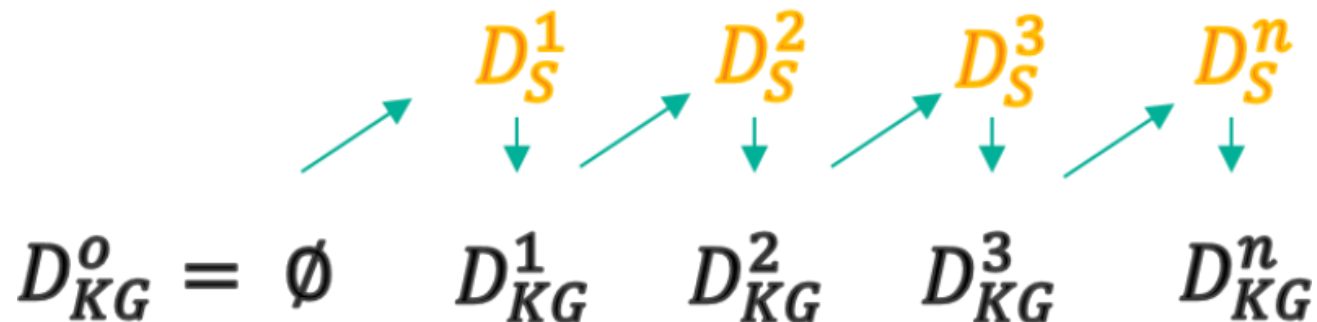
- Select ontologies
  - ... that define classes and properties for our data (e.g., DC, FOAF, CIDOC CRM...)
- Convert data to RDF
  - ... from the museum database to the ontologies
- Identify links to other Linked Data datasets
  - ... to other museums and Linked Data hubs

# Outline

- Motivation
- **Overview of Approach**
- Building and Extending a Knowledge Graph
- Evaluation
- Conclusion

# Goal: Integrate Artist Descriptions

- Getty Union List of Artist Names (ULAN): 109,415 artists
- Smithsonian American Art Museum (SAAM): 8,407 artists
- DBpedia: 1,176,759 people
- The Virtual International Authority File (VIAF): 16,244,546 people
  
- Goal: consolidate the data into a knowledge graph of artists

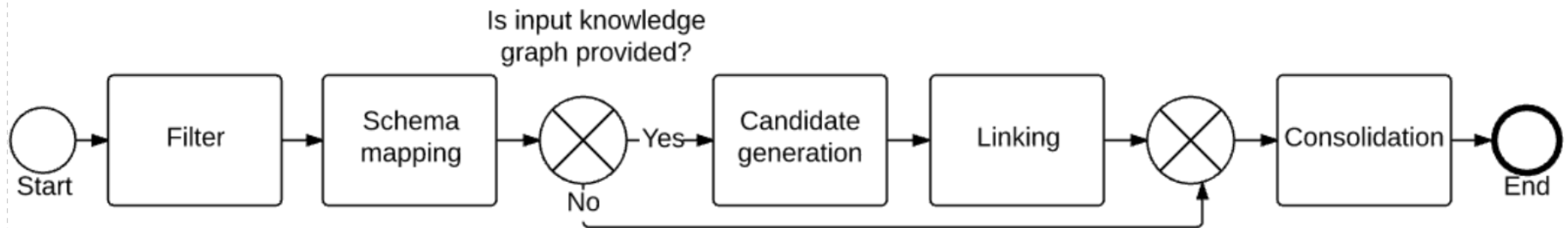


# Challenge: Scalability

- Object consolidation requires to compute the similarity of each entity with each other entity
- Impractical with our data size
  - DBpedia ~1.2m people (~900 MB), VIAF ~16.2m people (67 GB)
- How to reduce the number of pair-wise comparisons?

# Overview of Approach

1. Filter
2. Schema mapping
3. Candidate generation
4. Linking
5. Consolidation





# Outline

- Motivation
- Overview of Approach
- **Building and Extending a Knowledge Graph**
- Evaluation
- Conclusion

# 1. Filter

- We are interested in artists, but the data sources contain information about many more things
- In the filter step, we select all artists from DBpedia and VIAF via SPARQL queries
- We use a streaming query processor (Linked Data-Fu) to run a query that selects only people from the data and thus reduce the amount of data we have to process further

## 2. Schema Mapping

- We use the Karma tool to map the person descriptions in different ontologies to terms from schema.org

Property name	ULAN	SAAM	DBpedia	VIAF	Knowledge graph
name	X	X	X	X	X
alternateName		X		X	X
givenName		X	X	X	X
familyName		X	X	X	X
gender	X				X
nationality	X				X
birthDate	X	X	X	X	X
deathDate	X	X	X	X	X
birthPlace	X	X	X		X
deathPlace	X	X	X		X
description	X	X	X	X	X

# Karma in Action

**Karma** v1.41

Linked  
Data  
Mapping

- Command History**
- Import Excel File: crystal-bridges-records.xlsx
  - Show Model: crystal-bridges-records\_Sheet1
  - Set Semantic Type: title of SAAMCHO
  - Add User Link:
  - Set Semantic Type: dateOfBirth of SaamPerson
  - Set Semantic Type: dateOfDeath of SaamPerson
  - Set Semantic Type: medium of SAAMCHO
  - Set Semantic Type: format of SAAMCHO
  - Set Semantic Type: created of SAAMCHO
  - Set Semantic Type: title of SAAMCHO
  - Publish RDF:
  - Publish the Model:

crystal-bridges-records\_Sheet1 [RDF](#)

SAAMCHO

creator

SaamPerson

prefLabel\*      dateOfBirth      dateOfDeath      title\*      created

Attribution	Alpha Sort	Begin Date	End Date	Title	Dated	Begin Date	medium	Dimensions
Romare Bearden	Bearden, Romare	1911	1988	Sacrifice	1941	1941	Gouache and casein on paper	
George Wesley Bellows	Bellows, George Wesley	1882	1925	Excavation at Night	1908	1908	Oil on canvas	
George Wesley Bellows	Bellows, George Wesley	1882	1925	The Studio	1919	1919	Oil on canvas	
Thomas Hart Benton	Benton, Thomas Hart	1889	1975	The Steel Mill	1930	1930	Oil on canvas mounted on board	
Thomas Hart Benton	Benton, Thomas Hart	1889	1975	Ploughing It Under	1934,			
George de Forest Brush	Brush, G de Fore							
Dennis Miller Bunker	Bunker, D Miller						Oil on canvas	
Nick Cave	Cave, Nick	1959		Soundsuit	2010	2010	Appliqué found knitted and crocheted fabric, metal armature, and painted metal ...	97 x 48 x 42 in. (246.4 x 121.9 x 106.7 cm)

Crystal Bridges sample data

### 3. Candidate Generation

- MinHash/LSH operates over an n-gram representation of the name values, and hashes similar entities into the same cluster, based on the Jaccard similarity between the two sets of n-grams representing the two entities
- MinHash/LSH recall/precision performance depends on the number of used minhashes  $m$  and the number of items in the generated hashes  $I$
- LSH threshold  $t$  can be approximated as  $t = \frac{1}{i} \frac{1}{m}$

$t$	34%	40%	46%	52%
Precision	14.12%	15.69%	23.12%	26.96%
Recall	100%	99.50%	99.50%	98.01%

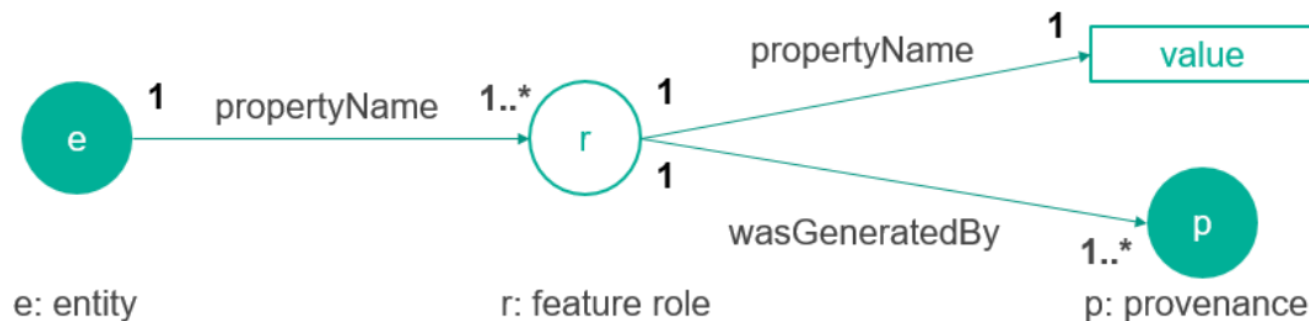
- We apply the MinHash/LSH with a low threshold of 46% to achieve high recall
- A low threshold leads to a low precision which we tolerate because the precision will be increased in the linking step

## 4. Linking

- Computes similarity based on matching functions on the found candidates
- When comparing people entities, we can define a matching function to
  - first check the similarity of the names and then
  - remove candidates with a different birth year
- Birth year might remove correct candidates (e.g., candidate “Pietro Aquila” has birth year “1592” in ULAN but “1650” in SAAM)

## 5. Consolidation

- Merge data from different sources while keeping provenance using the PROV ontology
- We use an n-ary representation to be able to keep provenance information within the triple data model (binary predicates)



# Outline

- Motivation
- Overview of Approach
- Building and Extending a Knowledge Graph
- **Evaluation**
- Conclusion



# Runtime Performance Results

- 161,465 artists consolidated from four data sources, based on 17,539,125 entities processed (link to dataset in paper)
- 4 AMD Opteron 62xx class 2GHz CPU cores and 32 GB RAM

Step	ULAN	SAAM	DBpedia	VIAF
Candidate generation	-	00:15:59	01:55:14	29:58:26
Linking	-	00:01:37	01:11:22	55:02:13
Consolidation	00:02:12	00:04:49	00:23:20	156:34:12
Total	00:02:12	00:22:25	03:29:56	229:00:39

# Quality Evaluation

- We manually build up a ground truth of links for the alphabetically first 200 artist entities which are represented in each of the four data sources and measured recall and precision

		T	Recall	Precision	
ULAN	Initial KG	-	-	-	
SAAM	Candidate generation	200	100%	10.87%	
	Linking	Hybrid-Jaccard	200	100%	92.59%
		birth year	187	93.50%	100%
DBpedia	Candidate generation	199	98.50%	12.83%	
	Linking	Hybrid-Jaccard	197	98.00%	91.59%
		birth year	187	93.50%	100%
VIAF	Candidate generation	271	97.84%	18.80%	
	Linking	Hybrid-Jaccard	271	97.12%	77.36%
		birth year	262	93.53%	96.30%

Most links are correct

Only few links are missing

# Outline

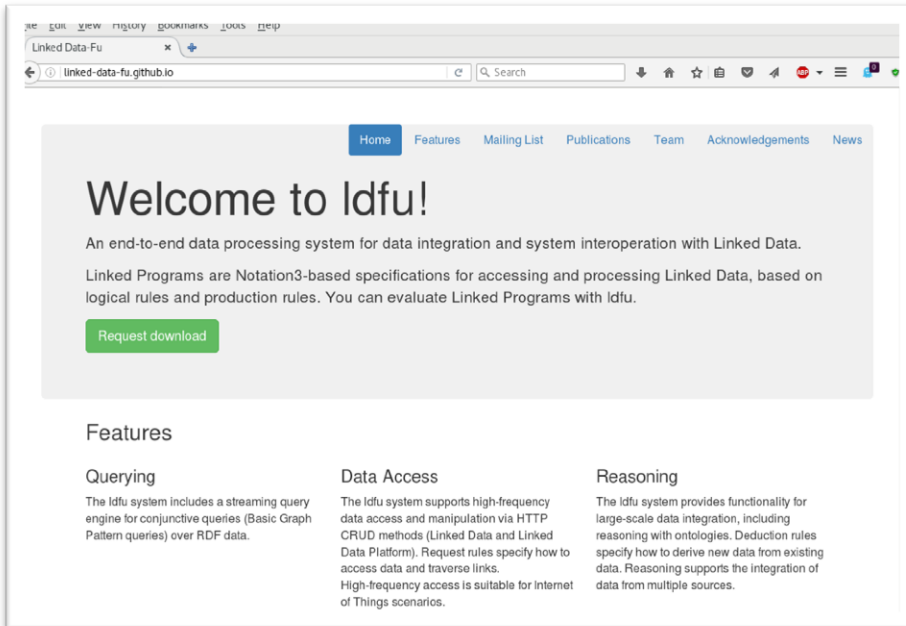
- Motivation
- Overview of Approach
- Building and Extending a Knowledge Graph
- Evaluation
- **Conclusion**

# Conclusion

- We have addressed the problem of efficiently building a consolidated knowledge graph out of multiple large data sources
- We have used the MinHash/LSH algorithm to identify candidate links to address the scalability challenge
- The approach can be used on different entity types and different datasets with minimal changes
- More elaborate matching functions could be used in conjunction with our approach
- We provide the used software as open source

<http://linked-data-fu.github.io/>

<http://www.isi.edu/integration/karma/>



The screenshot shows the homepage of Linked Data Fu. It features a navigation bar with links for Home, Features, Mailing List, Publications, Team, Acknowledgements, and News. The main heading is "Welcome to Idfu!" followed by a description: "An end-to-end data processing system for data integration and system interoperation with Linked Data. Linked Programs are Notation3-based specifications for accessing and processing Linked Data, based on logical rules and production rules. You can evaluate Linked Programs with Idfu." A green button labeled "Request download" is visible. Below this, there are three sections: "Features", "Querying", and "Data Access".

## Welcome to Idfu!

An end-to-end data processing system for data integration and system interoperation with Linked Data. Linked Programs are Notation3-based specifications for accessing and processing Linked Data, based on logical rules and production rules. You can evaluate Linked Programs with Idfu.

[Request download](#)

### Features

#### Querying

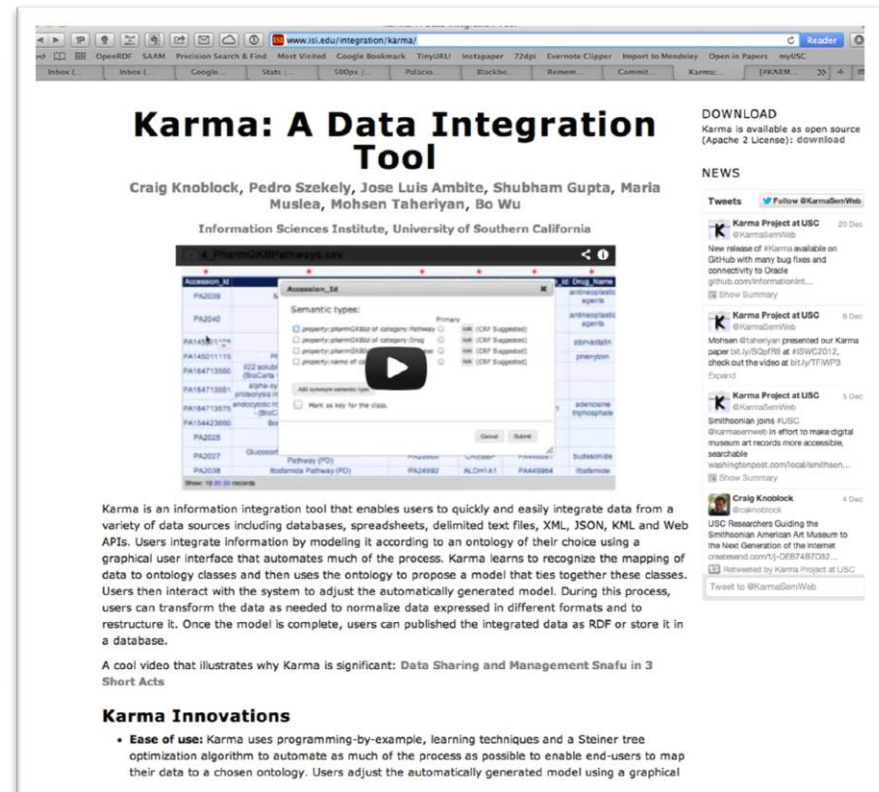
The Idfu system includes a streaming query engine for conjunctive queries (Basic Graph Pattern queries) over RDF data.

#### Data Access

The Idfu system supports high-frequency data access and manipulation via HTTP CRUD methods (Linked Data and Linked Data Platform). Request rules specify how to access data and traverse links. High-frequency access is suitable for internet of Things scenarios.

#### Reasoning

The Idfu system provides functionality for large-scale data integration, including reasoning with ontologies. Deduction rules specify how to derive new data from existing data. Reasoning supports the integration of data from multiple sources.



The screenshot shows the homepage of Karma: A Data Integration Tool. It features a navigation bar with links for Home, Features, Mailing List, Publications, Team, Acknowledgements, and News. The main heading is "Karma: A Data Integration Tool" followed by the authors: "Craig Knoblock, Pedro Szekeley, Jose Luis Ambite, Shubham Gupta, Maria Muslea, Mohsen Taheriyan, Bo Wu". Below this, there is a video player showing a screenshot of the Karma interface. The interface displays a table of data with columns for ID, Name, and Address. A video player is overlaid on the table. Below the video, there is a description of Karma: "Karma is an information integration tool that enables users to quickly and easily integrate data from a variety of data sources including databases, spreadsheets, delimited text files, XML, JSON, KML and Web APIs. Users integrate information by modeling it according to an ontology of their choice using a graphical user interface that automates much of the process. Karma learns to recognize the mapping of data to ontology classes and then uses the ontology to propose a model that ties together these classes. Users then interact with the system to adjust the automatically generated model. During this process, users can transform the data as needed to normalize data expressed in different formats and to restructure it. Once the model is complete, users can published the integrated data as RDF or store it in a database." Below this, there is a section titled "Karma Innovations" with a bullet point: "Ease of use: Karma uses programming-by-example, learning techniques and a Steiner tree optimization algorithm to automate as much of the process as possible to enable end-users to map their data to a chosen ontology. Users adjust the automatically generated model using a graphical".

## Karma: A Data Integration Tool

Craig Knoblock, Pedro Szekeley, Jose Luis Ambite, Shubham Gupta, Maria Muslea, Mohsen Taheriyan, Bo Wu

Information Sciences Institute, University of Southern California

Karma is an information integration tool that enables users to quickly and easily integrate data from a variety of data sources including databases, spreadsheets, delimited text files, XML, JSON, KML and Web APIs. Users integrate information by modeling it according to an ontology of their choice using a graphical user interface that automates much of the process. Karma learns to recognize the mapping of data to ontology classes and then uses the ontology to propose a model that ties together these classes. Users then interact with the system to adjust the automatically generated model. During this process, users can transform the data as needed to normalize data expressed in different formats and to restructure it. Once the model is complete, users can published the integrated data as RDF or store it in a database.

A cool video that illustrates why Karma is significant: [Data Sharing and Management Snafu in 3 Short Acts](#)

### Karma Innovations

- Ease of use:** Karma uses programming-by-example, learning techniques and a Steiner tree optimization algorithm to automate as much of the process as possible to enable end-users to map their data to a chosen ontology. Users adjust the automatically generated model using a graphical

# American Art Collaborative

- Amon Carter Museum of American Art
- Archives of American Art, Smithsonian Institution
- Autry Museum of the American West
- Colby College Museum of Art
- Crystal Bridges Museum of American Art
- Dallas Museum of Art (DMA)
- Indianapolis Museum of Art (IMA)
- Thomas Gilcrease Institute of American History and Art
- National Portrait Gallery, Smithsonian Institution
- National Museum of Wildlife Art
- Princeton University Art Museum
- Smithsonian American Art Museum (SAAM)
- Walters Art Gallery
- Yale Center for British Art

<http://americanartcollaborative.org/about/members-of-the-american-art-collaborative/>