

Linked APIs

Mohsen Taheriyani

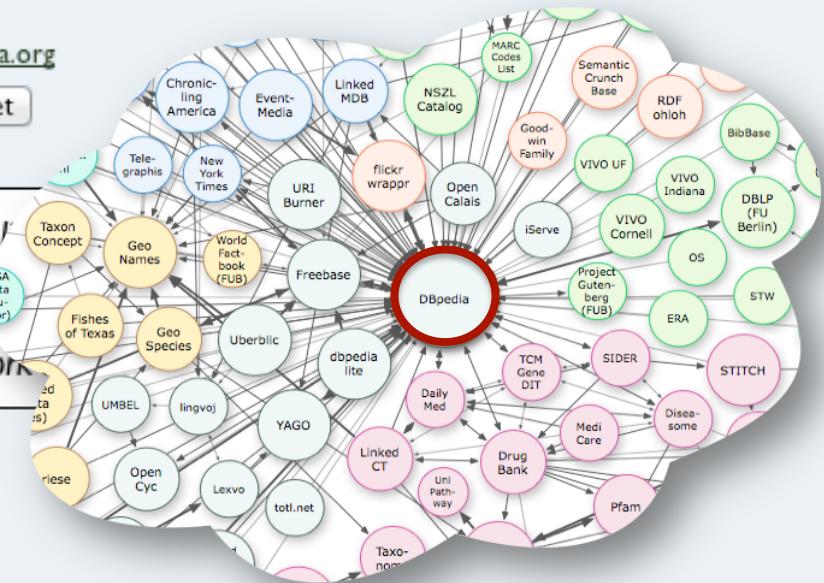
About: Downtown Los Angeles [Sponge](#) [Permalink](#)

An Entity of Type : [Neighborhoods in Los Angeles, California](#), within Data Space : [dbpedia.org](#)

Type: [yago:NeighborhoodsInLosAngeles,California](#)

Constrain facet

Downtown Los Angeles is the central business district of Los Angeles, California, the center of the metropolitan area. The area features many of the city's major opportunities, a variety of skyscrapers and associated large multinational shopping opportunities. Downtown is the hub of the city's freeway network.



leader title

City Council
U.S. House
State Senate
State Assembly

population total

39537(xsd:integer)

area total (km2)

8.538
13.73988692533248

foaf:homepage

<http://www.downtownla.com>

foaf:depiction



About: Downtown Los Angeles [Sponge](#) [Permalink](#)
 An Entity of Type : [Neighborhoods in Los Angeles, California](#), within Data Space
 Type: [yago:NeighborhoodsInLosAngeles,California](#)

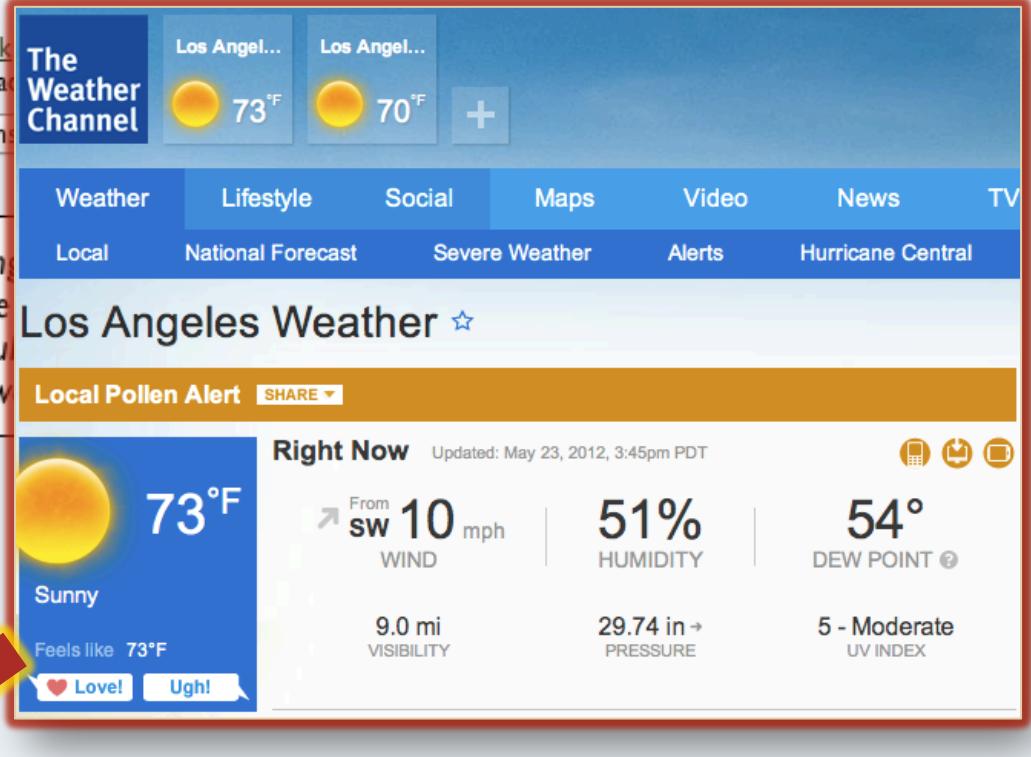
Downtown Los Angeles is the central business district of Los Angeles, California, the center of the metropolitan area. The area features many of the opportunities, a variety of skyscrapers and associated large murals, and shopping opportunities. Downtown is the hub of the city's freeway system.

leader title
 City Council
 U.S. House
 State Senate
 State Assembly

population total
 39537(xs:integer)

area total (km2)
 8.538
 13.73988692533248

Current temperature	73°F
Wind	10 mph SW
Humidity	51%
Dew Point	54°



Live weather

About: Downtown Los Angeles [Sponge](#) [Permalink](#)

An Entity of Type : [Neighborhoods in Los Angeles, California](#), within Data Space : [dbpedia.org](#)

Type: [yago:NeighborhoodsInLosAngeles,California](#)

Downtown Los Angeles is the central business district of the city of Los Angeles, California. It is the financial center of the metropolitan area. The area features many opportunities, a variety of skyscrapers and associated shopping opportunities. Downtown is the hub of the city's business, culture, and entertainment.

leader title	City Council U.S. House State Senate State Assembly
------------------------------	--

population total	39537(xsd:integer)
----------------------------------	--------------------

area total (km2)	8.538 13.73988692533248
----------------------------------	----------------------------

Current temperature	73°F
Wind	10 mph SW
Humidity	51%
Dew Point	54°

Events	Transparent Cities 5/1/2012 REDCAT Theater Saving Our Sons : A Community Conversatio... ...
--------	---

Events for May. 2012

[◀](#) [▶](#)

Film

[Transparent Cities](#)

5/1/2012

REDCAT Theater

Meeting

[Saving Our Sons: A Community Conversation](#)

5/2/2012

Los Angeles Trade Tech College

Special Event

[Autumn Lights Night](#)

5/3/2012

401 s Main Street

Concert

[Martin Hasselbeck Leads LACO's Baroque Conversations](#)

5/3/2012

Colburn School of Performing Arts

Ongoing Events

Exhibit

[A Nation Emerges: The Mexican Revolution Revealed](#)

9/8/2011 - 6/3/2012

LA Public Library - Central

Museum

[Justice, Balance and Achievement: African Americans in the California Courts](#)

1/12/2012 - 5/27/2012

California African American Museum

Exhibit

[Trouble In Paradise: Music and Los Angeles, 1945-1975](#)

2/22/2012 - 6/3/2012

GRAMMY Museum

Exhibit

[Xploration Lab 2012](#)

3/17/2012 - 6/17/2012

Events

About: Downtown Los Angeles [Sponge](#) [Permalink](#)

An Entity of Type : [Neighborhoods in Los Angeles, California](#), within Data Space : [dbpedia.org](#)

Type: [yago:NeighborhoodsInLosAngeles,California](#)

[Constrain facet](#)

Downtown Los Angeles is the central business district of Los Angeles, the center of the metropolitan area. The area features many of the city's opportunities, a variety of skyscrapers and associated large multi-story shopping opportunities. Downtown is the hub of the city's freeway system.

leader title

City Council
U.S. House
State Senate
State Assembly

population total

39537(xsd:integer)

area total (km2)

8.538
13.73988692533248

Current temperature

73°F

Wind

10 mph SW

Humidity

51%

Dew Point

54°

Events

Transparent Cities 5/1/2012 RED
Saving Our Sons : A Community...

Twitter Feed

Los Angeles becomes the largest U.S. city to ban...
Plastic grocery bags will now be banned in ...



Yale Environment 360 @YaleE360

Los Angeles becomes the largest U.S. city to ban **plastic bags** bit.ly/JA0PdX #LosAngeles #plasticbagban

[Expand](#)



DestinationTampaBay @DestTampaBay

Plastic grocery bags will now be banned in #LosAngeles. What do you think of that concept for #Florida?

[Expand](#)



annmarie anderson @AnnmarieCloud

"@HuffingtonPost: The nation's second largest city has just banned **plastic bags** in grocery stores huff.to/Jygs5N" #LosAngeles #green

[Expand](#)



gomaam @gomaam

#LosAngeles is now the largest U.S. city to **ban plastic bags** dlvr.it/1cX0bW

[Expand](#)



Ozzyopolis @Ozzyopolis

Los Angeles Becomes Largest U.S. City to Ban **Plastic Bags**: bit.ly/Jpl9xw via @TreeHugger #LosAngeles #plasticbagban

[Expand](#)   



Live Twitter Feed

The information is
available in Web APIs

About: Downtown Los Angeles

An Entity of Type : Neighborhoods in Los Angeles, California

Type: yago:NeighborhoodsInLosAngeles,California

Downtown Los Angeles is the central business district of Los Angeles, California, United States, located close to the geographic center of the metropolitan area. The area features many of the city's major arts institutions and sports facilities, sightseeing opportunities, a variety of skyscrapers and associated large multinational corporations and an array of public art and unique shopping opportunities. Downtown is the hub of the city's freeway network and growing Metro rapid transit system.

leader title

City Council
U.S. House
State Senate
State Assembly

population total

39537(xsdt:integer)

area total (km2)

8.538
13.73988692533248

Current temperature

73°F

Wind

10 mph SW

Humidity

51%

Dew Point

54°

Events

Transparent Cities 5/1/2012 REDCAT Th...
Saving Our Sons : A Community Conversatio...

...

Twitter Feed

Los Angeles becomes the largest U.S. ci...
Plastic grocery bags will now be banned

...

World Weather Online API

[Summary](#)

[Mashups \(1\)](#)

[How-To \(1\)](#)

[Developers \(1\)](#)

[Comments](#)



World Weather Online provides weather forecast and we...
for worldwide city and towns. The Developer Weather AP...
developers and programmers to access free 5 day weather

Eventful API

[Summary](#)

[Mashups \(46\)](#)

[How-To](#)

[Developers \(30\)](#)

[Comments](#)



Eventful is the world's largest collection of events, taking...
markets throughout the world, from concerts and sports...
events and political rallies. Eventful.com is built upon a

Twitter API

[Summary](#)

[Mashups \(709\)](#)

[How-To \(34\)](#)

[Developers \(432\)](#)

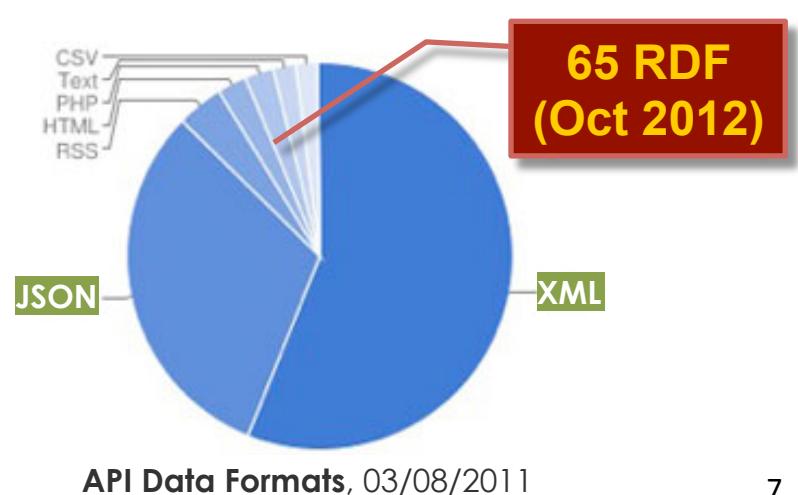
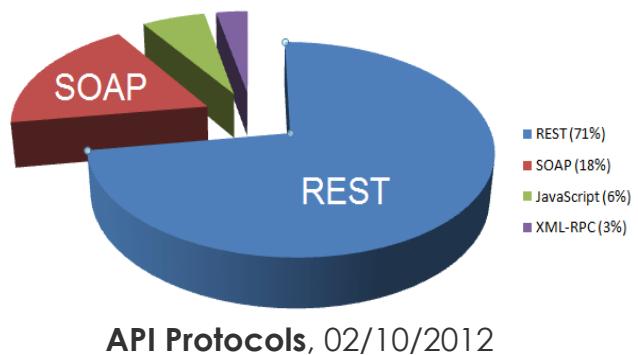
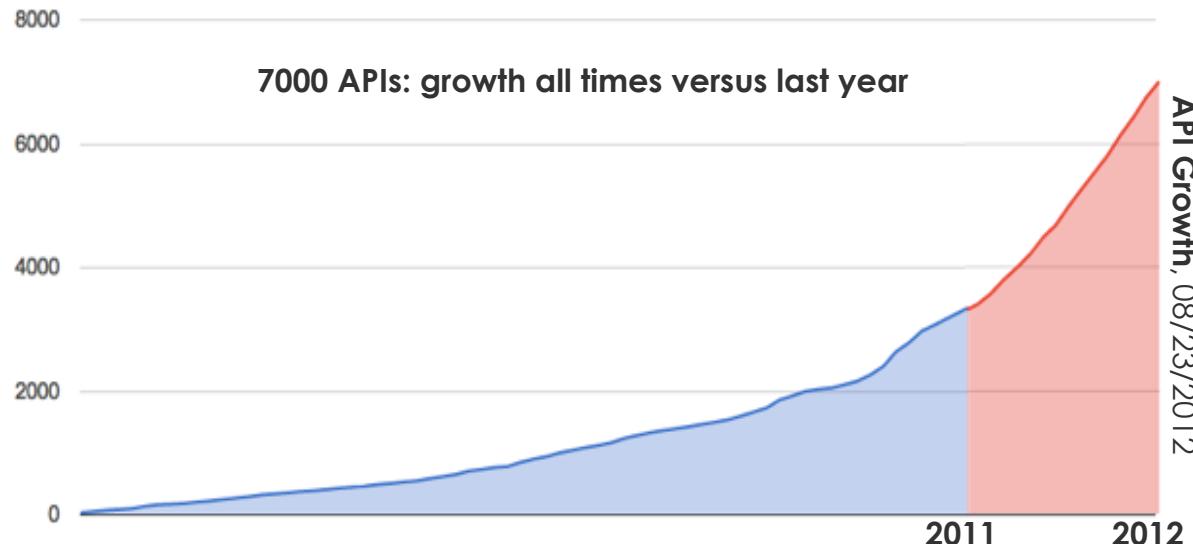
N

AP



The Twitter micro-blogging service includes two RE...
Twitter REST API methods allow developers to acc...

Web APIs



Problem 1: Finding APIs

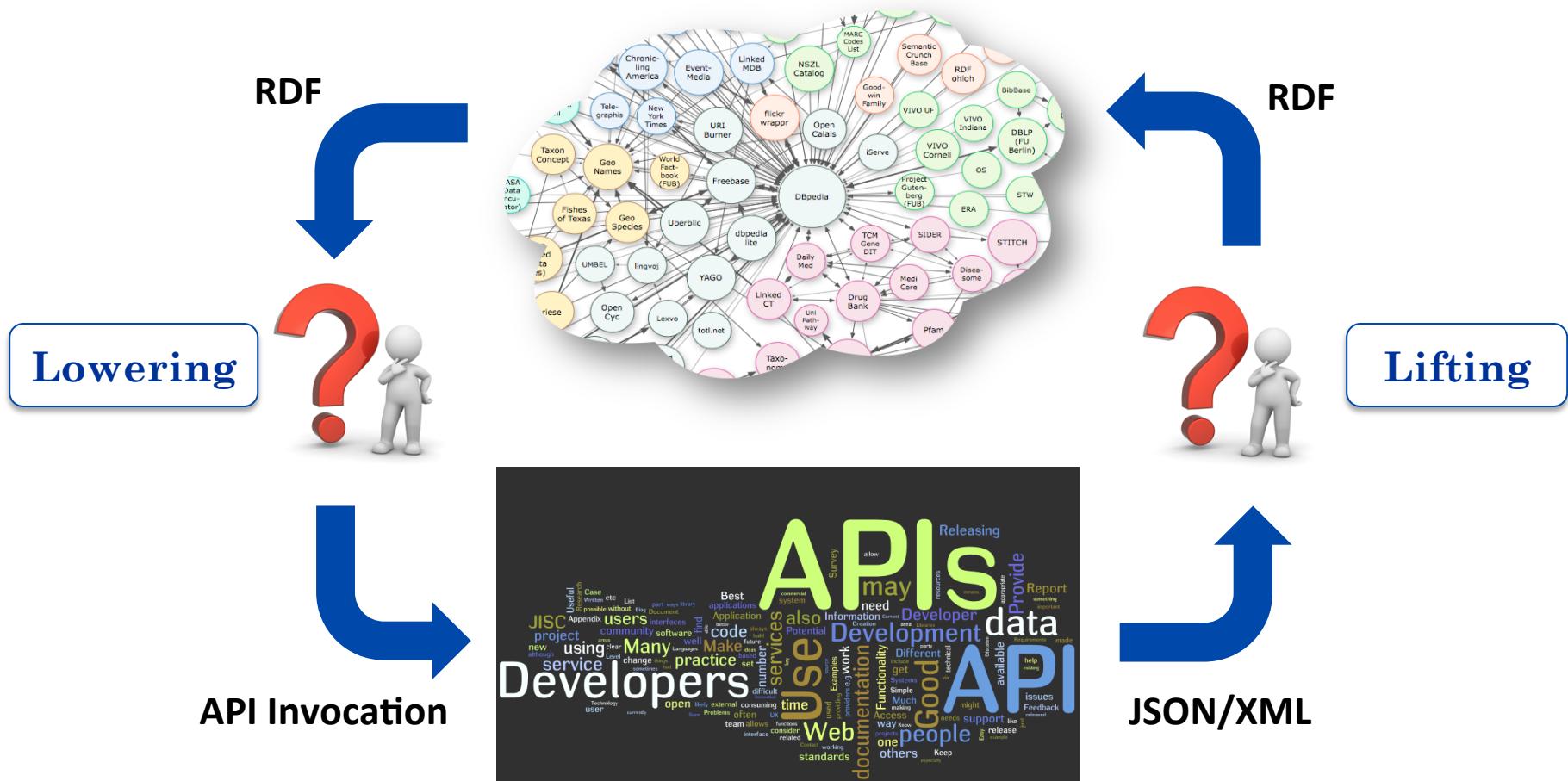


```
<http://sws.geonames.org/5145067>
a gn:Feature ;
wgs84:lat "40.78343" ;
wgs84:long "-73.96625" .
```

- Finds all APIs that take latitude and longitude as inputs
- Find a services that returns the nearby city of the given the latitude and longitude

For most of the Web APIs, only a textual description is available

Problem 2: Consuming and Producing Linked Data



Most of the Web APIs do not consume or produce linked data (RDF)

Integrating Services and LOD

- Publishing Service as Linked Data
 - C. Pedrinaci, D. Liu, M. Maleshkova, D. Lambert, J. Kopecký, J. Domingue: [iServe: a Linked Services Publishing Platform](#). ESWC 2010.
- Linked Open Services (LOS)
 - B. Norton, R. Krummenacher: [Consuming Dynamic Linked Data](#). 1st International Workshop on Consuming Linked Data, 2010.
- Linked Data Services (LIDS)
 - S. Speiser, A. Harth: [Towards Linked Data Services](#). ISWC 2010.
- Modeling Services in Karma
 - M. Taheriyan, C. A. Knoblock, P. Szekely, J. L. Ambite: [Rapidly Integrating Services into the Linked Data Cloud](#). ISWC 2012.

Publishing Service as Linked Data

Semantic Web Services (SWS)

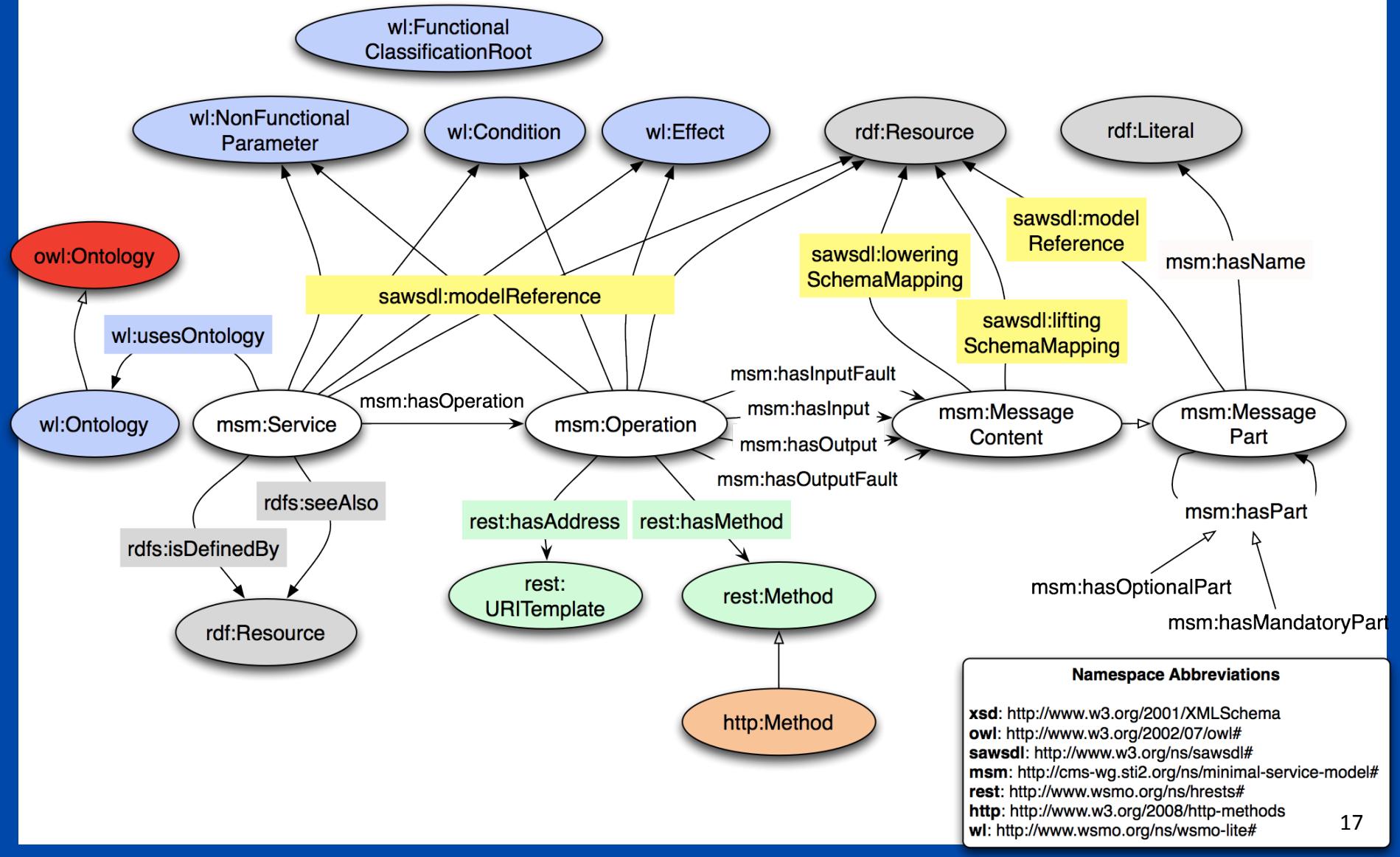
- Building **semantic descriptions** for services
- Providing support for service discovery
- Different (incompatible) vocabularies are proposed so far
 - OWL-S, WSMO-Lite, MicroWSMO, SAWSDL
- Complex formalisms

iServe

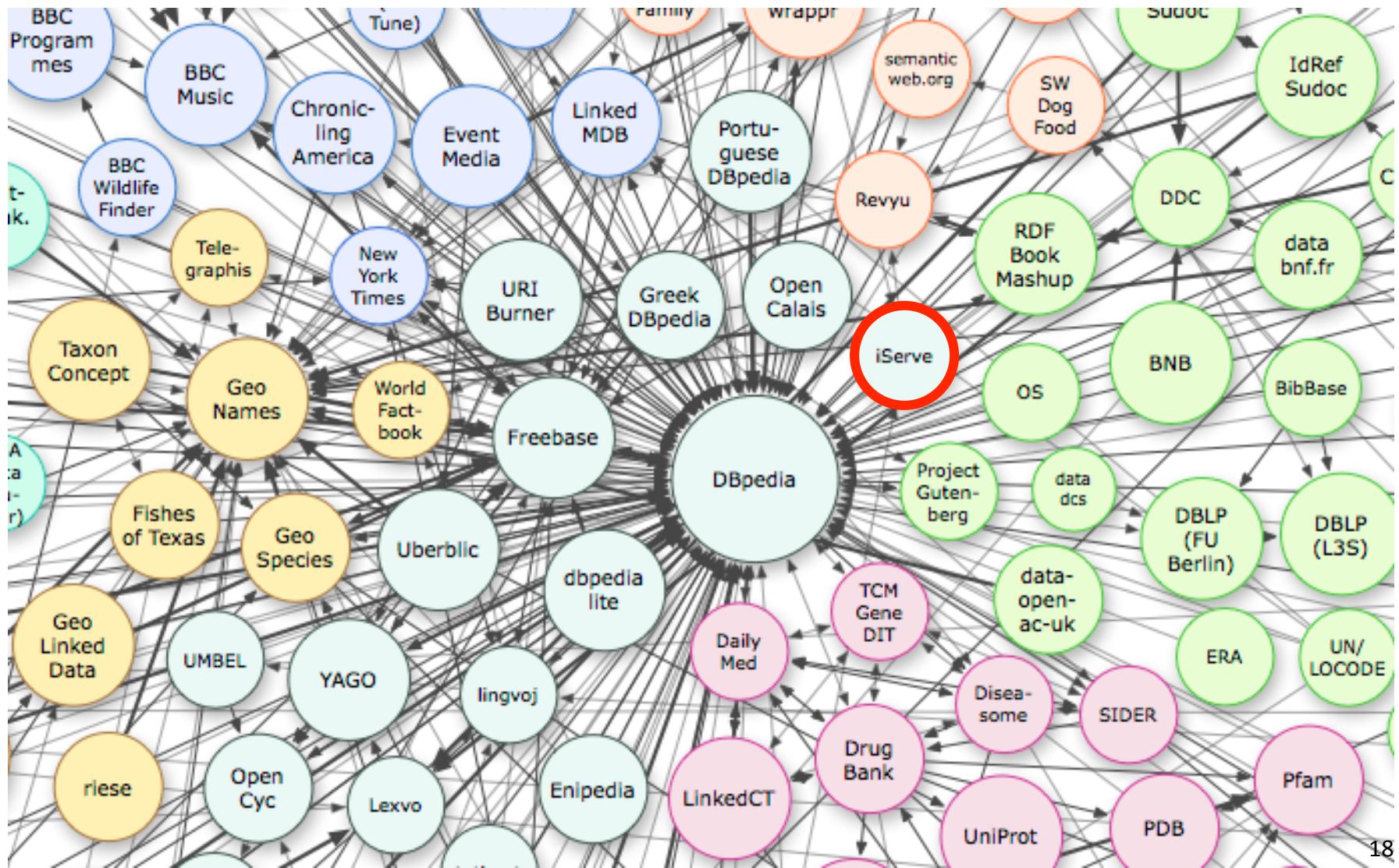
- iServe: A Linked Services publishing platform
- Unifying service **discovery** and **publication**
- Annotating services using a simple RDF ontology called **MSM** (Minimal Service Model)
- Exposing descriptions as Linked Data



Minimal Service Model



Publish as Linked Data



Service discovery Example

Find all services taking a dbpedia:City as input

```
SELECT ?url WHERE {  
    ?s      rdf:type msm:Service;    msm:hasOperation ?o.  
    ?o      rdf:type msm:Operation; rest:hasAddress ?url;  
            msm:hasInput ?in.  
    ?in     rdf:type msm:MessageContent; msm:hasPart ?par.  
    ?par    rdf:type  msm:MessagePart; sawsdl:modelReference ?m.  
    ?m      rdf:type dbpedia:City.  
}
```

How can we find a service that takes a City and returns its US State?

iServe Status

- One of the largest publicly accessible Semantic Service registries
 - More than 2000 services
- Supports access via
 - Web application - [iServe Browser](#)
 - Read&Write RESTful API
 - SPARQL endpoint
- Support for several SWS formalism
 - WSMO-Lite, MicroWSMO, SAWSDL, OWL-S
- [**SWEET**](#): a tool to help users in building MSM descriptions of services from their textual descriptions in HTML pages



Linked Open Services (LOS)

Linked Open Services (LOS)

- Services that communicate at the semantic level without lifting and lowering
- LOS services directly consume and produce linked data
- Linked Open Services = Linked Open Data + REST



LOS Principles

- I. Describe services as LOD prosumers with input and output descriptions as SPARQL graph patterns
2. Communicate RDF by RESTful content negotiation
3. The output should make explicit its relation with the input

LOS Example

1 - input and output as SPARQL graph patterns

Input: ?x geo:name ?name

Output: ?x wgs84:lat ?lat; wgs84:long ?lng.

HEAD

POST /examples/getLatLngByName HTTP/1

HOST: www.isi.edu

Content-Type: application/rdf+xml

BODY

```
<rdf:RDF  
    xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"  
    xmlns:geo="http://ws.geonames.org/">  
  
<rdf:Description about="http://...#parc">  
    <geo:name>Palo Alto Research Center</geo:name>  
</rdf:Description>  
</rdf:RDF>
```

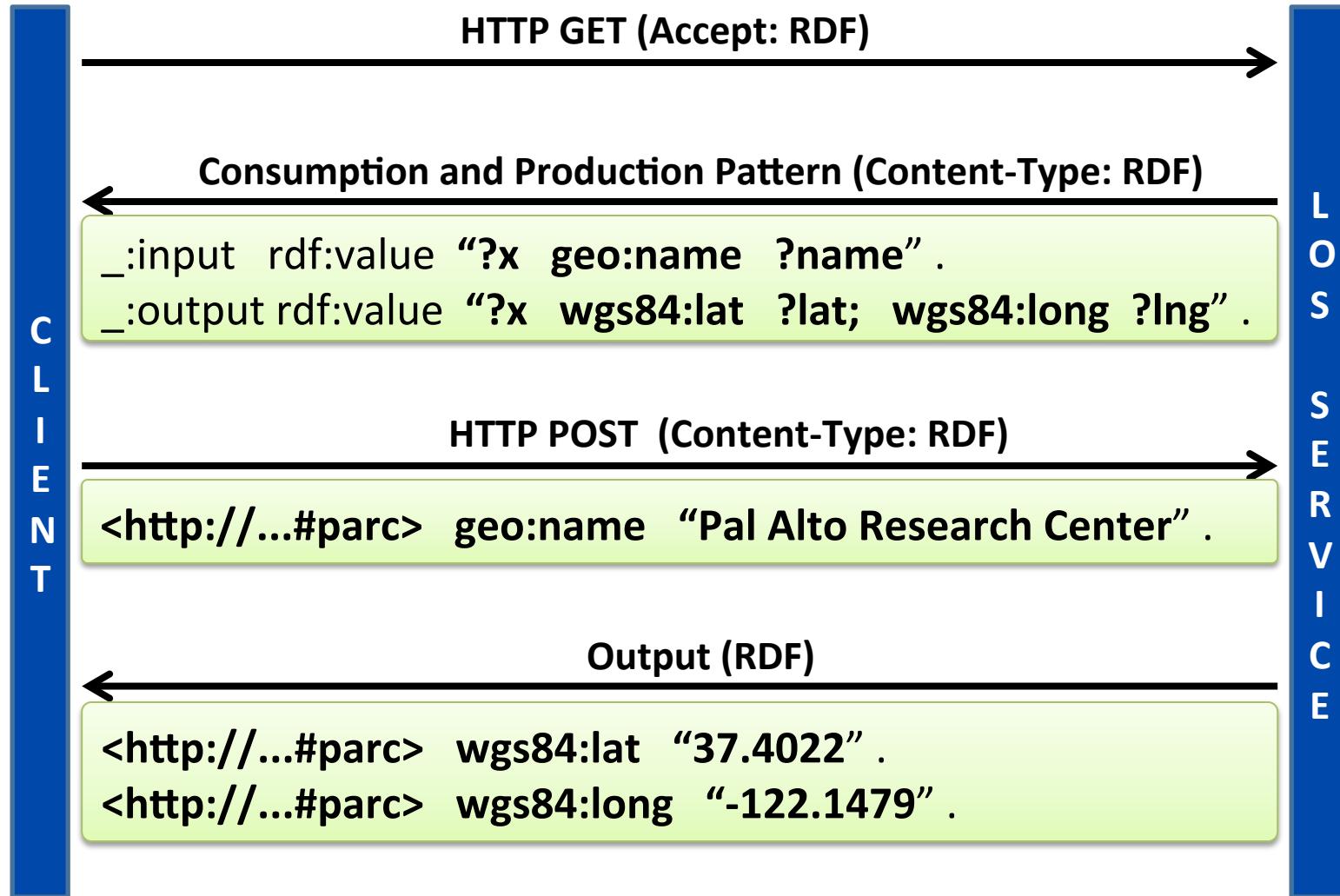
...

[**<http://...#parc>**](#)
wgs84:lat "37.4022" .
[**<http://...#parc>**](#)
wgs84:long "-122.1479" .
...

2 - communicate RDF

3 - explicit relation between output and input

Invoking LOS



What about existing services?

25	postalCodeCountryInfo	XML	JSON
26	postalCodeLookup		JSON
27	postalCodeSearch	XML	JSON
28	rssToGeo	RSS	KML
29	search		
30	siblings		
31	srtm3		
32	timezone	Note-1	
33	weather		
34	weatherIcao		
35	wikipediaBoundingBox		
36	wikipediaSearch		

Weather Stations with most recent Weather Observation

Webservice Type : REST

Url : api.geonames.org/weatherJSON?

Parameters :

north,south,east,west : coordinates of bounding box

callback : name of javascript function (optional parameter)

maxRows : maximal number of rows returned (default = 10)

Result : returns a list of weather stations with the most recent weather observation

Example <http://api.geonames.org/weatherJSON?north=44.1&south=-9.9&east=-22.4&west=55.2&username=demo>



JSON

```
{"weatherObservation":  
  {"clouds":"broken clouds",  
   "weatherCondition":"drizzle",  
   "observation":"EDDT 031520Z 25013KT 9999 ...",  
   "countryCode":"DE",  
   "windDirection":30, ...}
```

Lifting to RDF?

LOS Wrapping

JSON

```
{"weatherObservation":  
  {"clouds":"broken clouds",  
   "weatherCondition":"drizzle",  
   "observation":"EDDT 031520Z 25013KT 9999 ...",  
   "countryCode":"DE",  
   "windDirection":30, ...}
```



Generic lifting after RDF

```
[ temp:weatherObservation [  
    temp:louds "few clouds" ;  
    temp:weatherCondition "frizzle" ;  
    temp:countryCode "DE"  
    temp:windDirection "30" ;  
    ... ]]
```

LOS Wrapping (cont.)

Generic lifting after RDF

```
[ temp:weatherObservation [
    temp:louds "few clouds" ;
    temp:weatherCondition "frizzle" ;
    temp:countryCode "DE"
    temp:windDirection ...
]]
```

SPARQL
CONSTRUCT

```
CONSTRUCT {
?obs met:weatherObservation ?f .
?f weather:hasVisibilityEvent ?clouds ;
  weather:hasWindEvent [weather:windDirection ?windDirection] ;
  ...
} WHERE {
?obs temp:weatherObservation ?f .
  temp:windDirection ?windDirection ;
  OPTIONAL { _:c weather:hasVisibilityEvent ?clouds }
  ...
}
```

RDF

```
[
  weather:hasVisibilityEvent metar:BrokenClouds ;
  weather:hasWeatherEvent metar:Drizzle ;
  metar:observation "LESO 251..." ;
  weather:hasWindEvent
    [weather:windDirection "30"^^xsd:integer]
]
```

RDFS

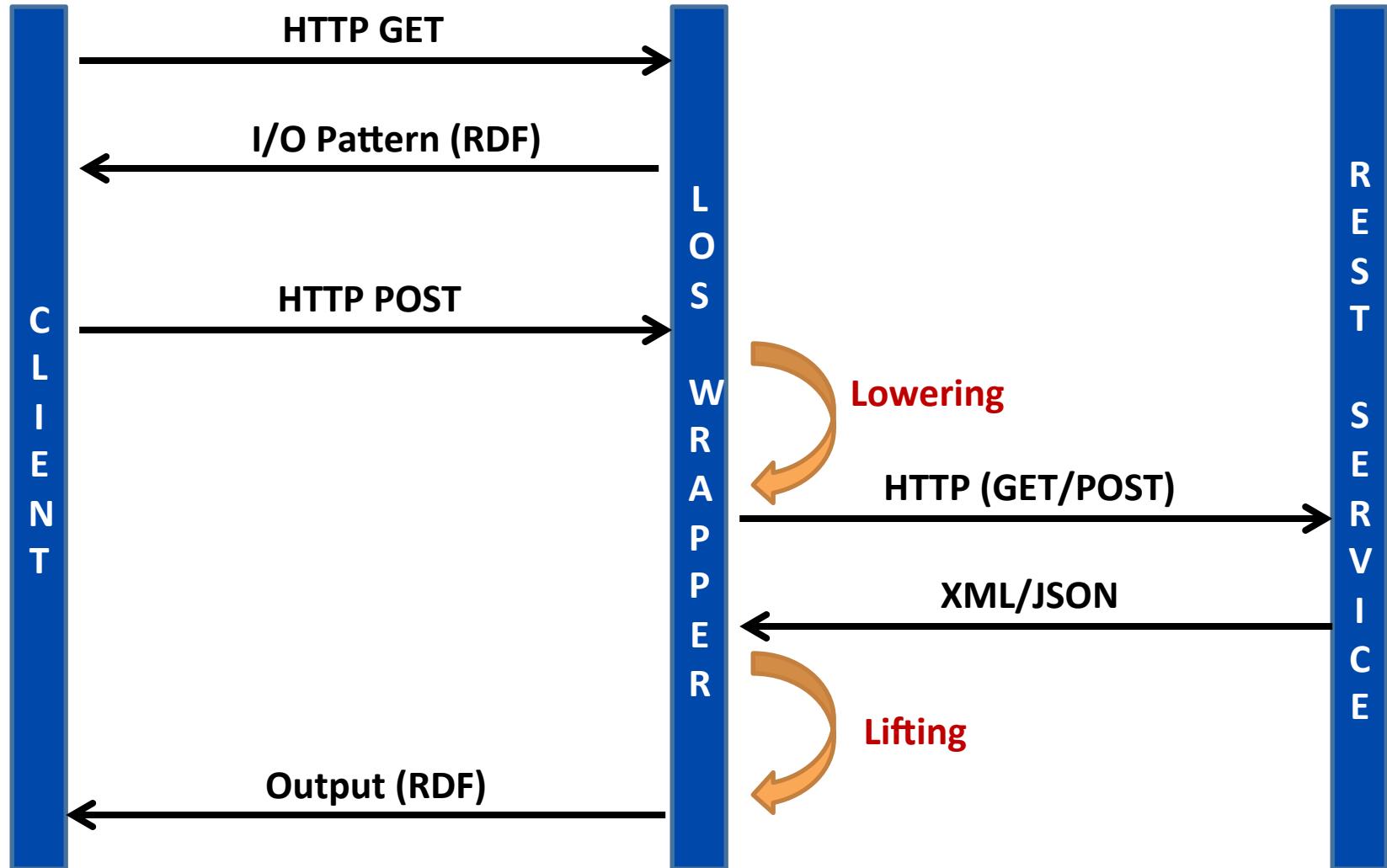
WeatherObservation



Report

METAR Or

Invoking LOS Wrapper



Exposing LOS as Linked Data

```
@prefix sparql: <http://www.w3.org/TR/rdf-sparql-query/#> .  
@prefix sawsdl: <http://www.w3.org/ns/sawsdl#> .  
@prefix wsl: <http://www.wsmo.org/wsmo-lite#> .  
@prefix posm: <http://www.wsmo.org/ns/posm/0.1#> .  
@prefix geo: <http://ws.geonames.org/> .  
geo:search rdf:type posm:Service;  
    posm:hasInputMessage [sawsdl:modelReference _:input];  
    posm:hasOutputMessage [sawsdl:modelReference _:output];  
    sawsdl:loweringSchemaMapping  
        <http://www.LinkedOpenServices.com/ex/geo/search.sparql> .  
    _:input rdf:value '?x geo:name ?name ."^^sparql:GraphPattern .'  
    _:output rdf:value "?x wgs84:lat ?lat; wgs84:long ?lng ."^^...  
    
```

Input and output are encoded as SPARQL strings. How can we query SPARQL patterns?

Linked Data Services (LIDS)

Linking Data and Services

- Many of the Web APIs are just “Data Services”
 - Do not alter the state of some entity, nor modify data
- How to create link between Linked Data and Data Services?

Motivation Example

Data (RDF): #PARC lat "37.4022"; long "-122.1479".

Data Service: <http://ws.geonames.org/findNearbyWikipedia>

`http://ws.geonames.org/findNearbyWikipedia?lat=37.4022&lng=-122.1479`

```
<geonames>
<entry>
<title>Hewlett-Packard</title>
<summary>Hewlett-Packard Company , commonly referred to as HP, ...</
summary>
<.wikipediaUrl>http://en.wikipedia.org/wiki/Hewlett-Packard</wikipediaUrl>
...
</entry>
<entry>
<title>Gunn High School</title>
<summary> ..</summary>
```

How to link #PARC (input) and its nearby locations (output)?

Linked Data Services (LOS)

- <http://openlids.org/>
 1. The input for a service invocation with given parameter bindings must be identified by a URI
 2. Resolving that URI must return a description of the input entity, relating it to the service output data
 3. The description must be returned in RDF format

LIDS Example

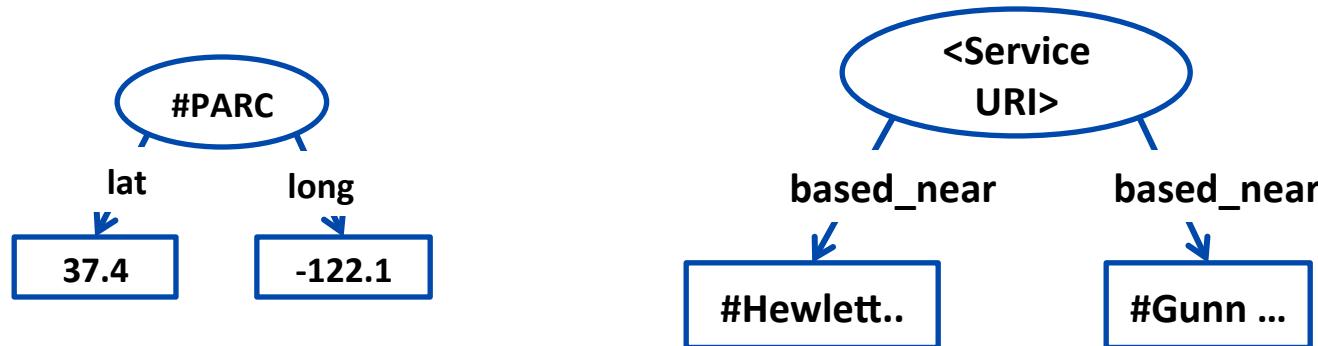
- Input is given as URI

`http://km.aifb.kit.edu/services/geonameswrap/findNearbyWikipedia?
lat=37.4022&lng=-122.1479` ← Parameters
`#point` ← Input Identifier

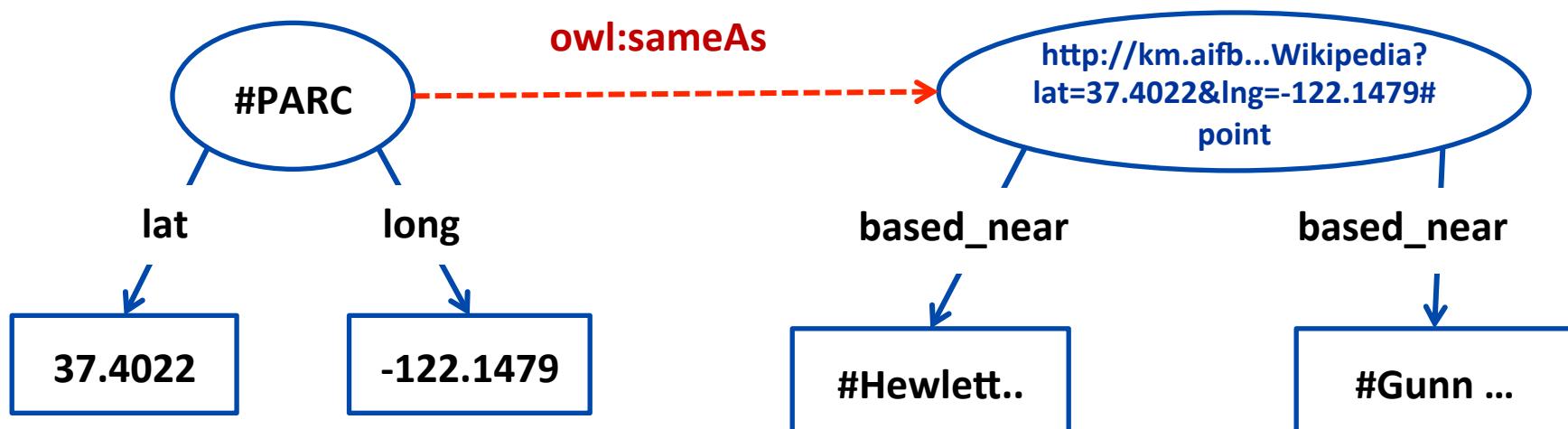
Service Endpoint

- Resolving the URI yields RDF

`http://km.aifb...Wikipedia?lat=37.4022&lng=-122.1479#point` ← Input
Relation → `foaf:based_near dbp:Hewlett-Packard ;
foaf:based_near dbp:Gunn High School .` ← Output



Integrating Linked Data and Services



Publishing LIDS as Linked Data

- One possible vocabulary to make LIDS discovery easy

```
@prefix lids: http://openlids.org/vocab#
:GeowrapNearbyWikipedia a lids:LIDS;
  lids:lids_description[
    lids:endpoint
      <http://km.aifb.kit.edu/services/geowrap/findNearbyWikipedia>;
    lids:service_entity "point" ;
    lids:input_bgp "?point a Point; geo:lat ?lat ; geo:long ?long";
    lids:output_bgp "?point foaf:based_near ?feature" ;
    lids:required_vars "lat;long" ].
```

Modeling Services in Karma

Motivation Example

RDF Input

```
<http://sws.geonames.org/5145067/>
  a gn:Feature ;
  wgs84:lat "40.78343" ;
  wgs84:long "-73.96625" .
```

neighbourhood?

Motivation Example

RDF Input

```
<http://sws.geonames.org/5145067/>
  a gn:Feature ;
  wgs84:lat "40.78343" ;
  wgs84:long "-73.96625" .
```



Lowering

Linked RDF

```
<http://sws.geonames.org/5145067/>
  gn:neighbour [
    a gn:Feature ; gn:name "Woodside";
    gn:nearby [
      a gn:Feature ; gn:name "...Queens";
      gn:parentCountry [ ...
```



Lifting

Neighbourhood / reverse geocoding

The neighbourhood for US cities. Data provided by [Zillow](#) under co

Webservice Type : REST

Url : api.geonames.org/neighbourhood

Parameters : lat,lng

Result : returns the neighbourhood for the given latitude/longitude

Example [http://api.geonames.org/neighbourhood?
lat=40.78343&lng=-73.96625&username=demo](http://api.geonames.org/neighbourhood?lat=40.78343&lng=-73.96625&username=demo)

Invocation Example

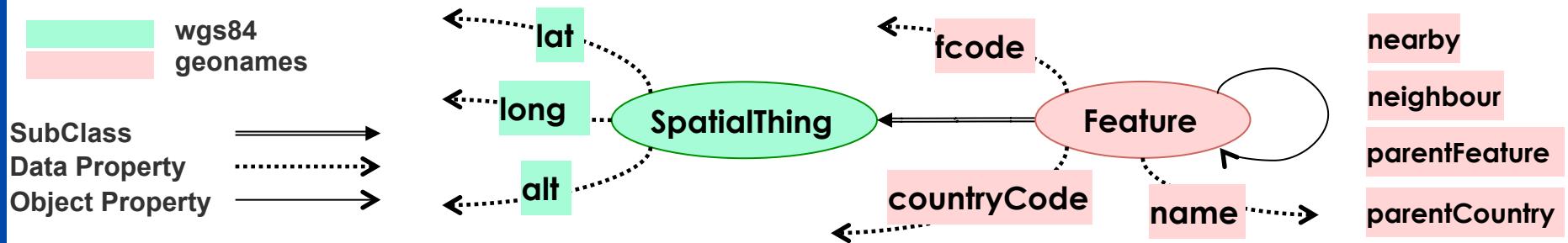
```
<geonames>
  <neighbourhood>
    <countryCode>US</countryCode>
    <countryName>United States</countryName>
    <adminCode1>NY</adminCode1>
    <adminName1>New York</adminName1>
    <adminCode2>061</adminCode2>
    <adminName2>New York County</adminName2>
    <city>New York City-Manhattan</city>
    <name>Central Park</name>
  </neighbourhood>
</geonames>
```

XML Response

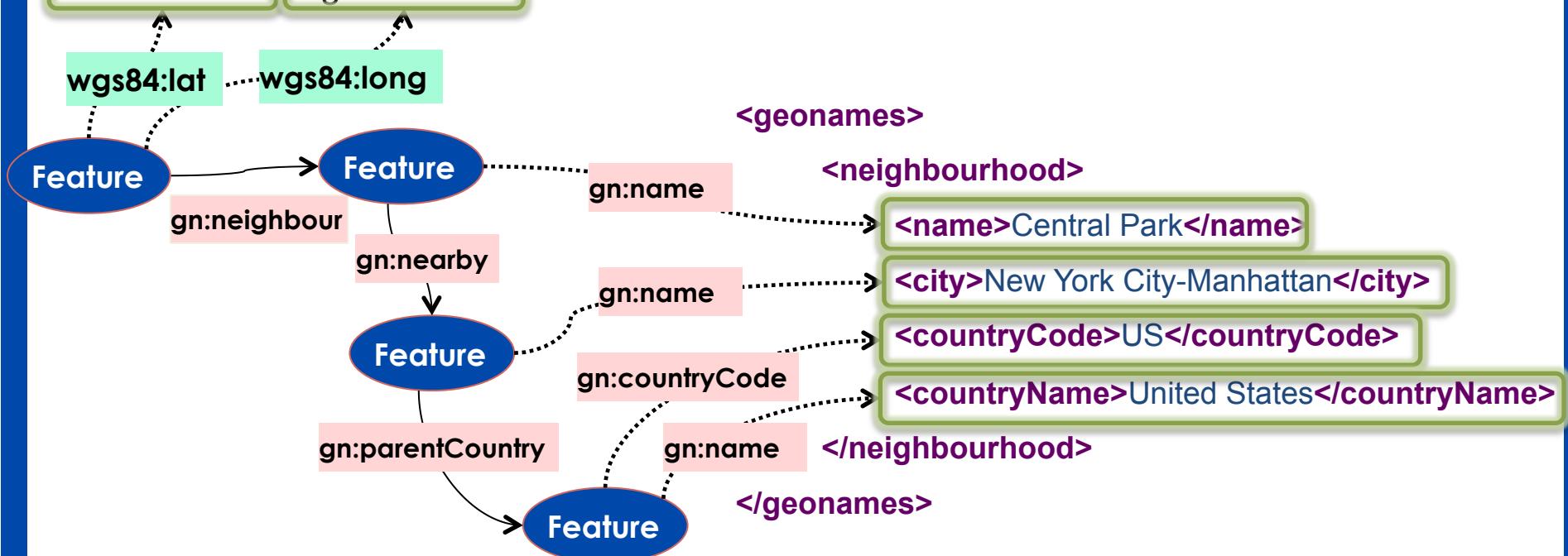
Outline

- Rapidly Integrating APIs with the LD
 - **Building API Semantic Model**
 - Representing API Descriptions
 - Building Linked APIs
- Evaluation

The Semantic Model



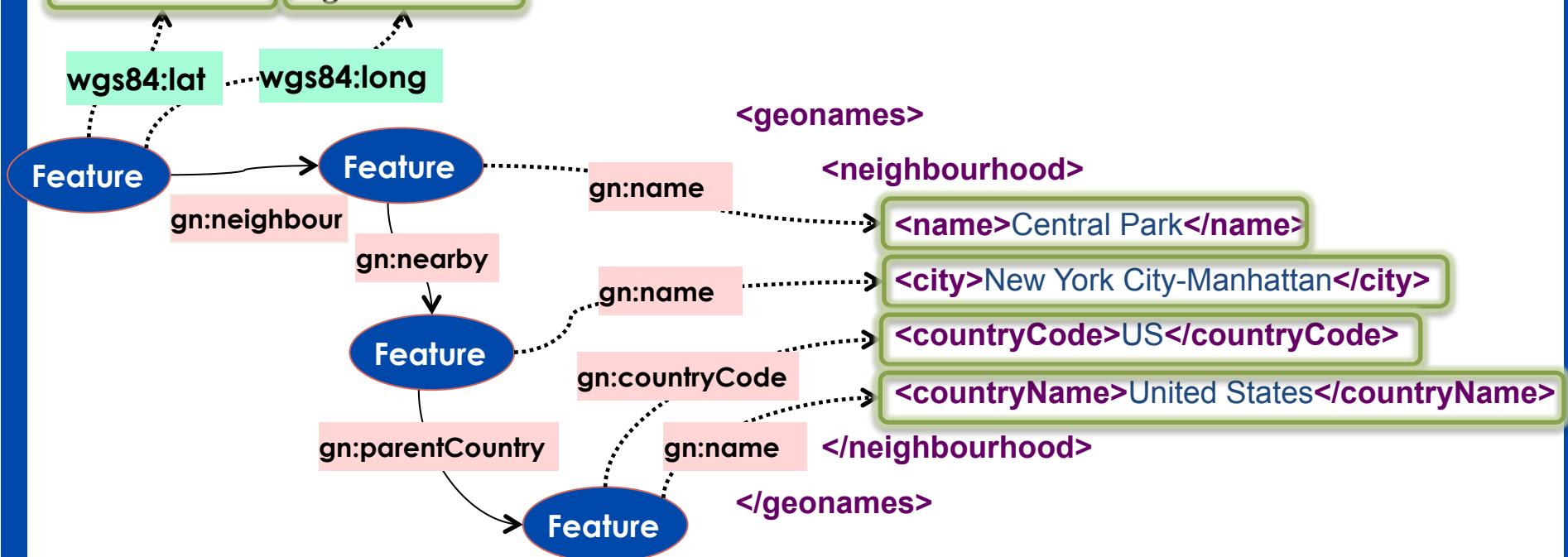
[http://api.geonames.org/neighbourhood?
lat=40.78343 &lng=-73.96625 &username=demo](http://api.geonames.org/neighbourhood?lat=40.78343&lng=-73.96625&username=demo)



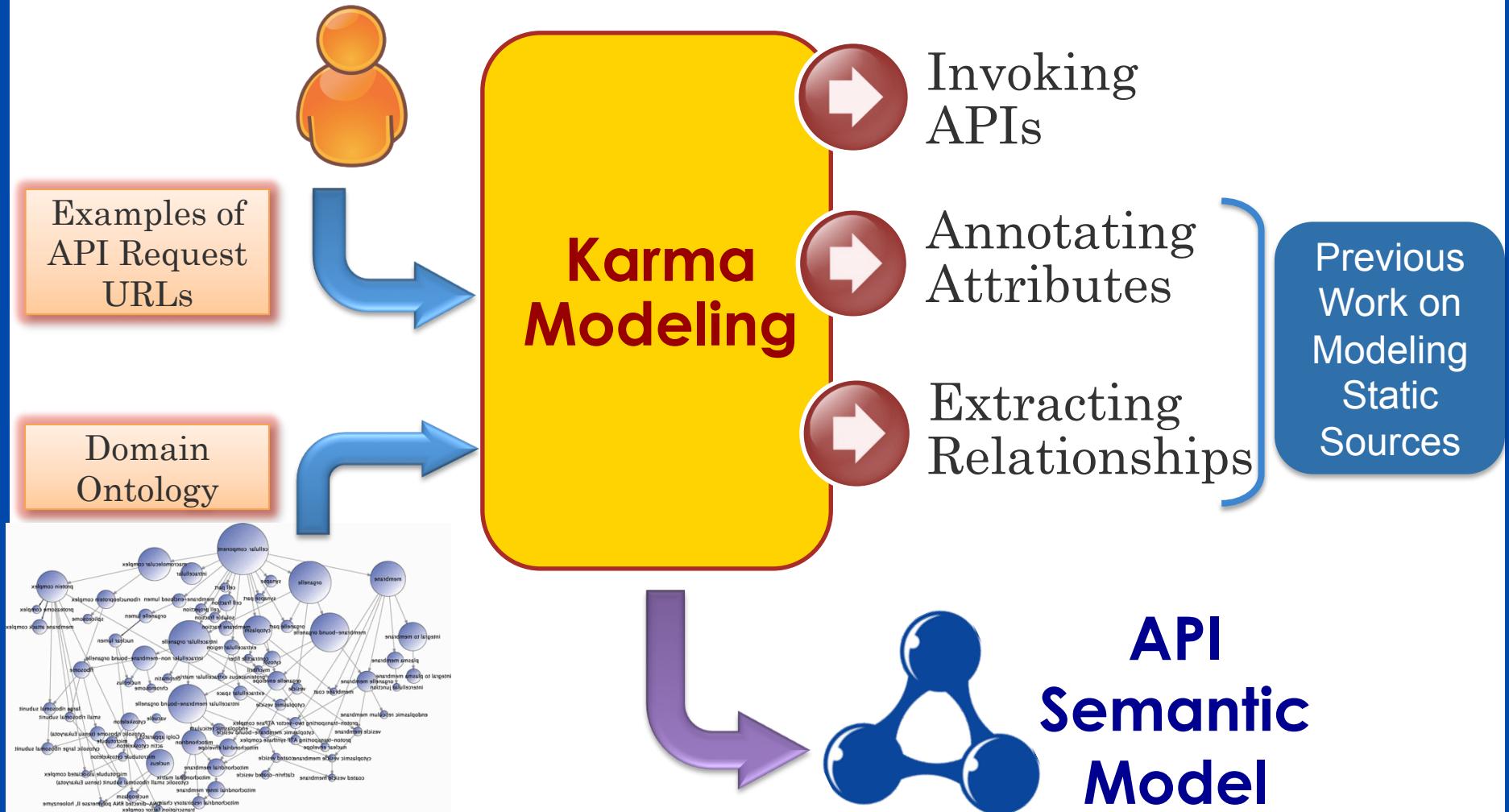
The Semantic Model

```
neighbourhood($lat, $long, @countryCode, @countryName, @city, @name) →  
  gn:Feature(v1) ∧ wgs84:lat(v1, $lat) ∧ wgs84:long(v1, $long) ∧ gn:neighbourhood(v1, v2) ∧  
  gn:Feature(v2) ∧ gn:name(v2, @name) ∧ gn:nearby(v2, v3) ∧  
  gn:Feature(v3) ∧ gn:name(v3, @city) ∧ gn:parentCountry(v3, v4) ∧  
  gn:Feature(v4) ∧ gn:countryCode(v4, @countryCode) ∧ gn:name(v4, @countryName)
```

[http://api.geonames.org/neighbourhood?
lat=40.78343 &lng=-73.96625 &username=demo](http://api.geonames.org/neighbourhood?lat=40.78343&lng=-73.96625&username=demo)



Semi-Automatically Modeling APIs



Invoking APIs

- User provides examples of the API request URLs

Examples of API Invocation URLs

The screenshot shows a web browser window with a title bar 'neighbourhood'. Below the title bar is a search bar labeled 'request url'. Three URLs are listed in a list box:

- http://api.geonames.org/neighbourhood?at=40.78343&lng=-73.96625&username=karma
- http://api.geonames.org/neighbourhood?at=40.71012&lng=-73.90078&username=karma
- http://api.geonames.org/neighbourhood?at=40.90145&lng=-73.90815&username=karma

At the bottom of the browser window, there is a 'Show' dropdown menu with options: 10, 20, 50 records.

Invoking APIs

- User provides examples of the API request URLs
- Karma extracts the input values from the sample request URLs

The diagram illustrates the process of extracting API invocation URLs and their corresponding input attributes. A yellow speech bubble labeled "Examples of API Invocation URLs" points to the "request url" column of a table. Another yellow speech bubble labeled "Input Attributes" points to the "lat", "lng", and "username" columns.

request url	lat	lng	username
http://api.geonames.org/neighbourhood?at=40.78343&lng=-73.96625&username=karma	40.78343	-73.96625	karma
http://api.geonames.org/neighbourhood?at=40.71012&lng=-73.90078&username=karma	40.71012	-73.90078	karma
http://api.geonames.org/neighbourhood?at=40.90145&lng=-73.90815&username=karma	40.90145	-73.90815	karma

Show: 10 20 50 records

Invoking APIs

- User provides examples of the API request URLs
- Karma extracts the input values from the sample request URLs
- Karma invokes the API and extracts the output attributes from the API response (XML/JSON)

The diagram illustrates the process of invoking APIs. It shows three yellow speech bubbles pointing to a table:

- The first bubble, labeled "Examples of API Invocation URLs", points to the "request url" column of the table.
- The second bubble, labeled "Input Attributes", points to the "lat", "lng", and "username" columns.
- The third bubble, labeled "Output Attributes", points to the "countryCode", "countryName", "city", and "name" columns.

neighbourhood

request url	lat	lng	username	countryCode	countryName	city	name
http://api.geonames.org/neighbourhood?at=40.78343&lng=-73.96625&username=karma	40.78343	-73.96625	karma	US	United States	New York City-Manhattan	Central Park
http://api.geonames.org/neighbourhood?at=40.71012&lng=-73.90078&username=karma	40.71012	-73.90078	karma	US	United States	New York City-Queens	Ridgewood
http://api.geonames.org/neighbourhood?at=40.90145&lng=-73.90815&username=karma	40.90145	-73.90815	karma	US	United States	New York City-Bronx	Riverdale

Show: 10 20 50 records Previous Next

Annotating Inputs and Outputs

[Goel, Knoblock, Lerman, 2012]

- A CRF-based model to assign a **Semantic Type** to each column from its data
- Semantic Type
 - Ontology Class
 - Data Property + Domain

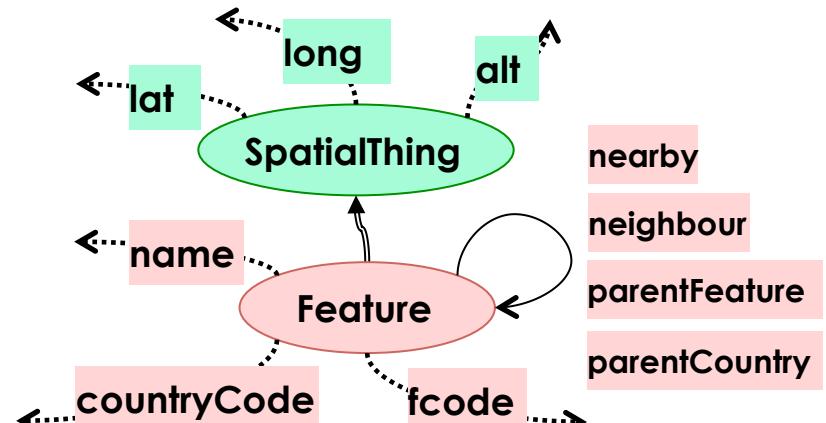
(wgs84:lat, gn:Feature)

lat	long	username	countryCode	countryName	city	name
40.78343	-73.96625	karma	US	United States	New York City-Manhattan	Central Park
40.71012	-73.90078	karma	US	United States	New York City-Queens	Ridgewood

(wgs84:long, gn:Feature)

(gn:name, gn:Feature)

(gn:countryCode, gn:Feature)



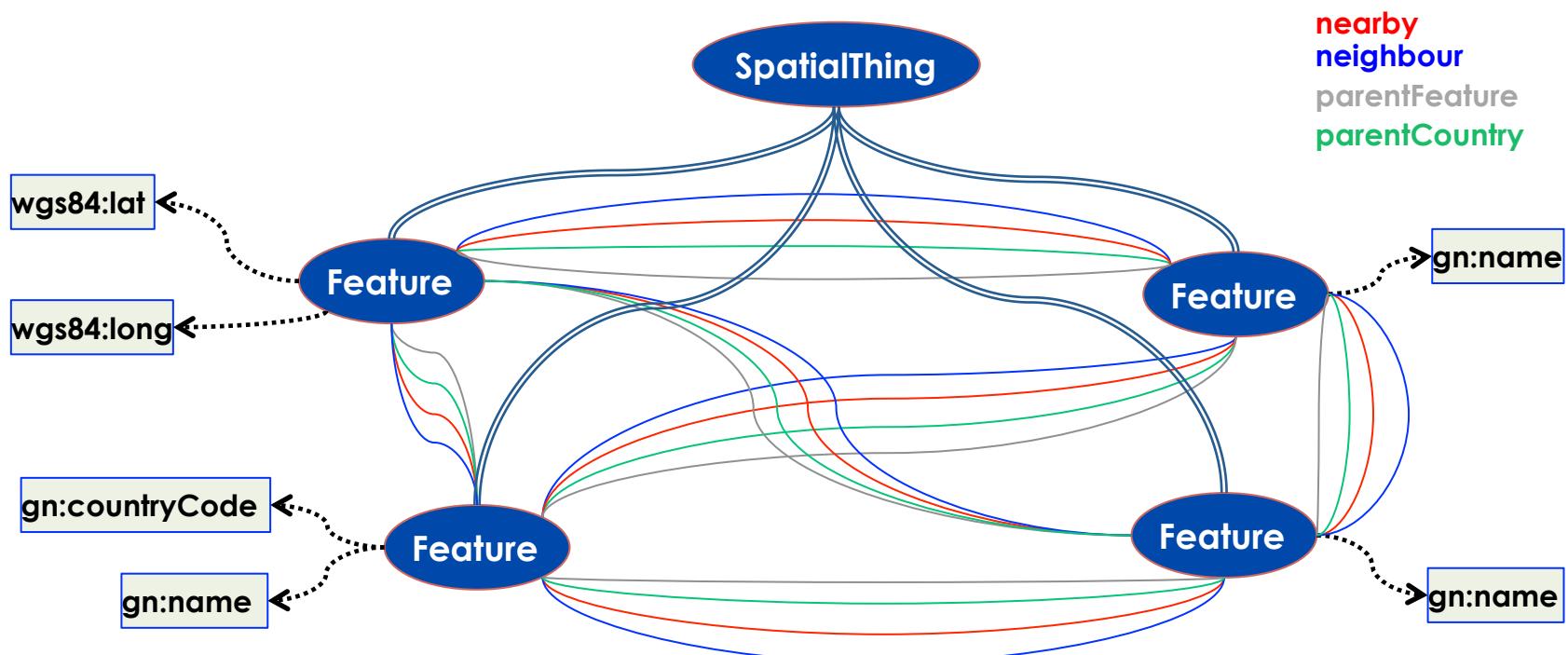
SubClass →
Data Property →
Object Property →

wgs84
geonames

Extracting Relationships

[Knoblock et al., ESWC 2012]

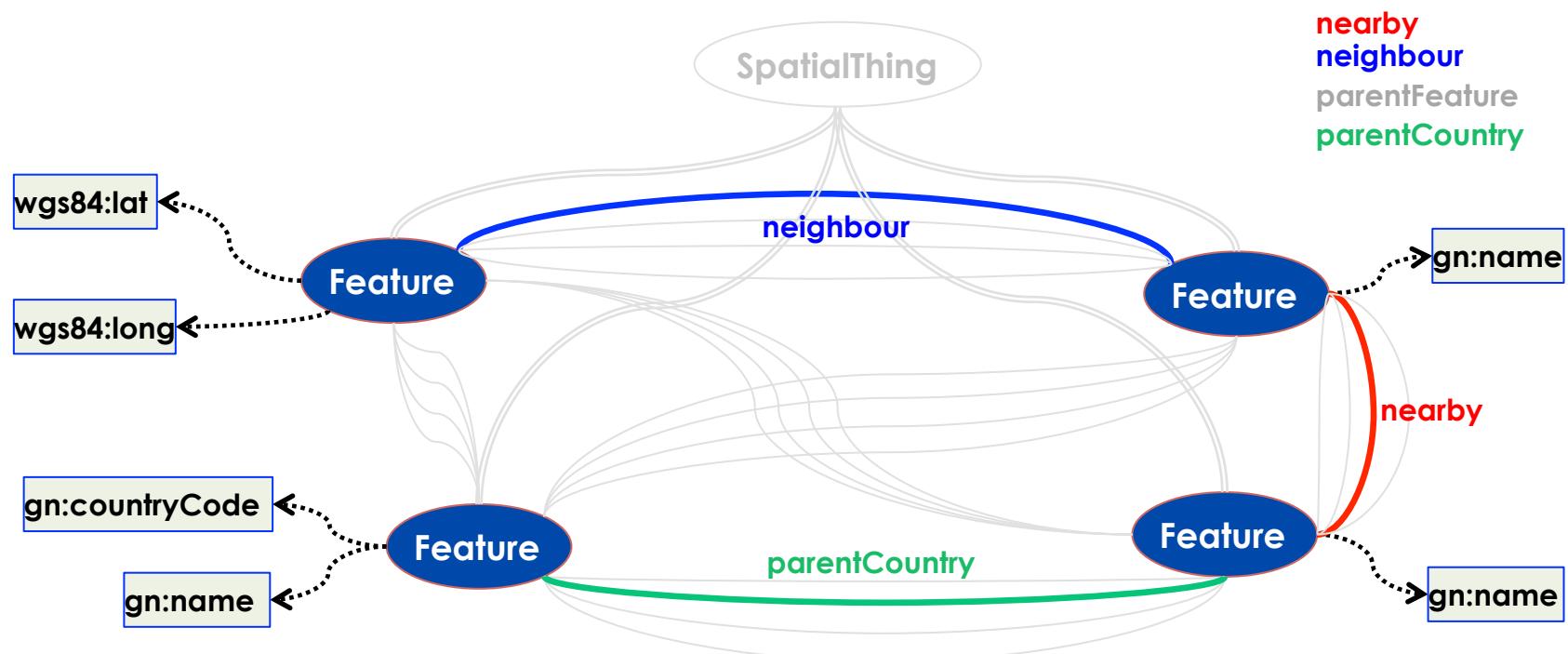
- Construct a graph from semantic types and ontology graph
- Select minimal tree that connects all semantic types
 - A customized **Steiner tree algorithm**



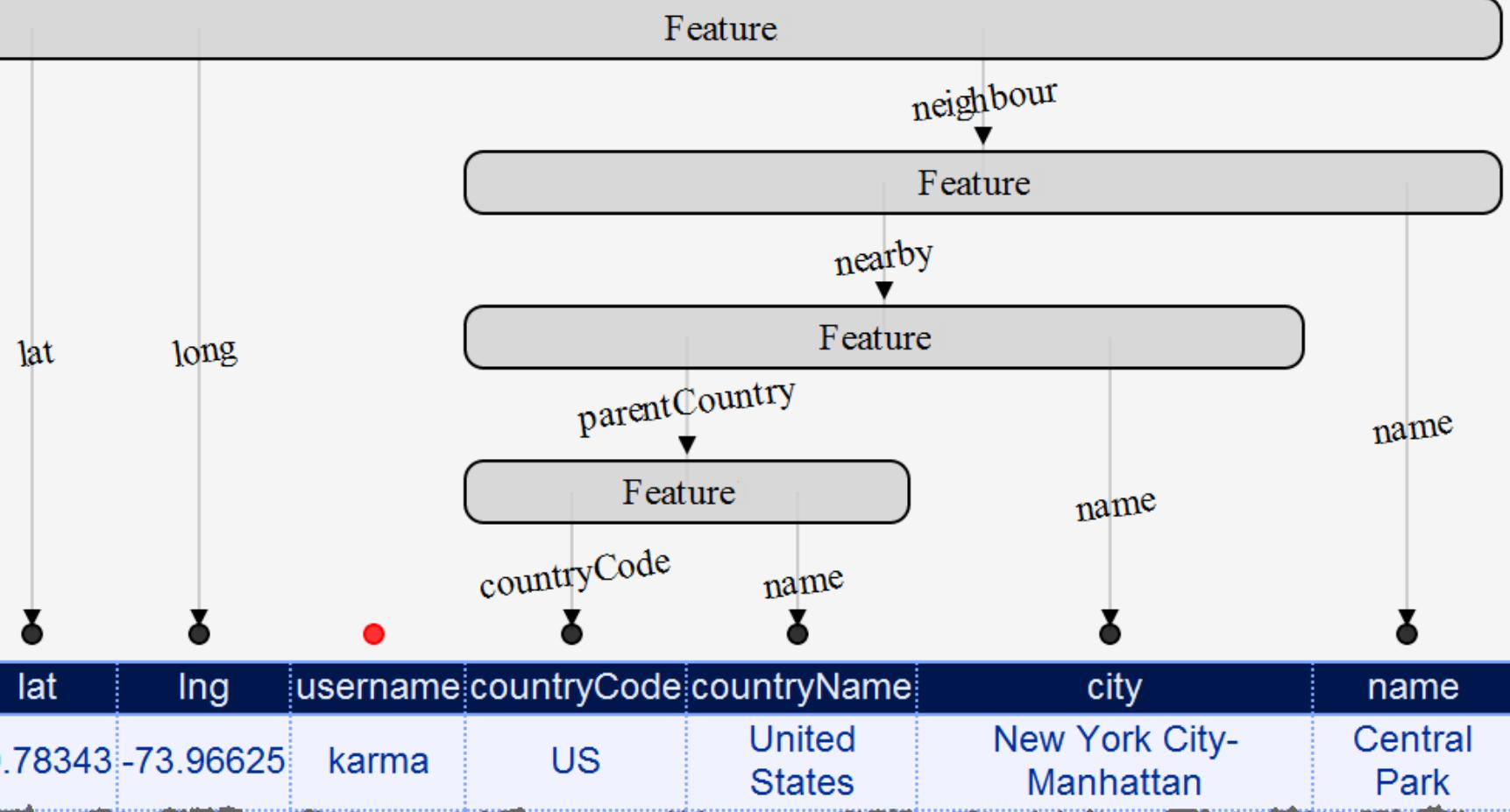
Extracting Relationships

[Knoblock et al., ESWC 2012]

- Construct a graph from semantic types and ontology graph
- Select minimal tree that connects all semantic types
 - A customized **Steiner tree algorithm**



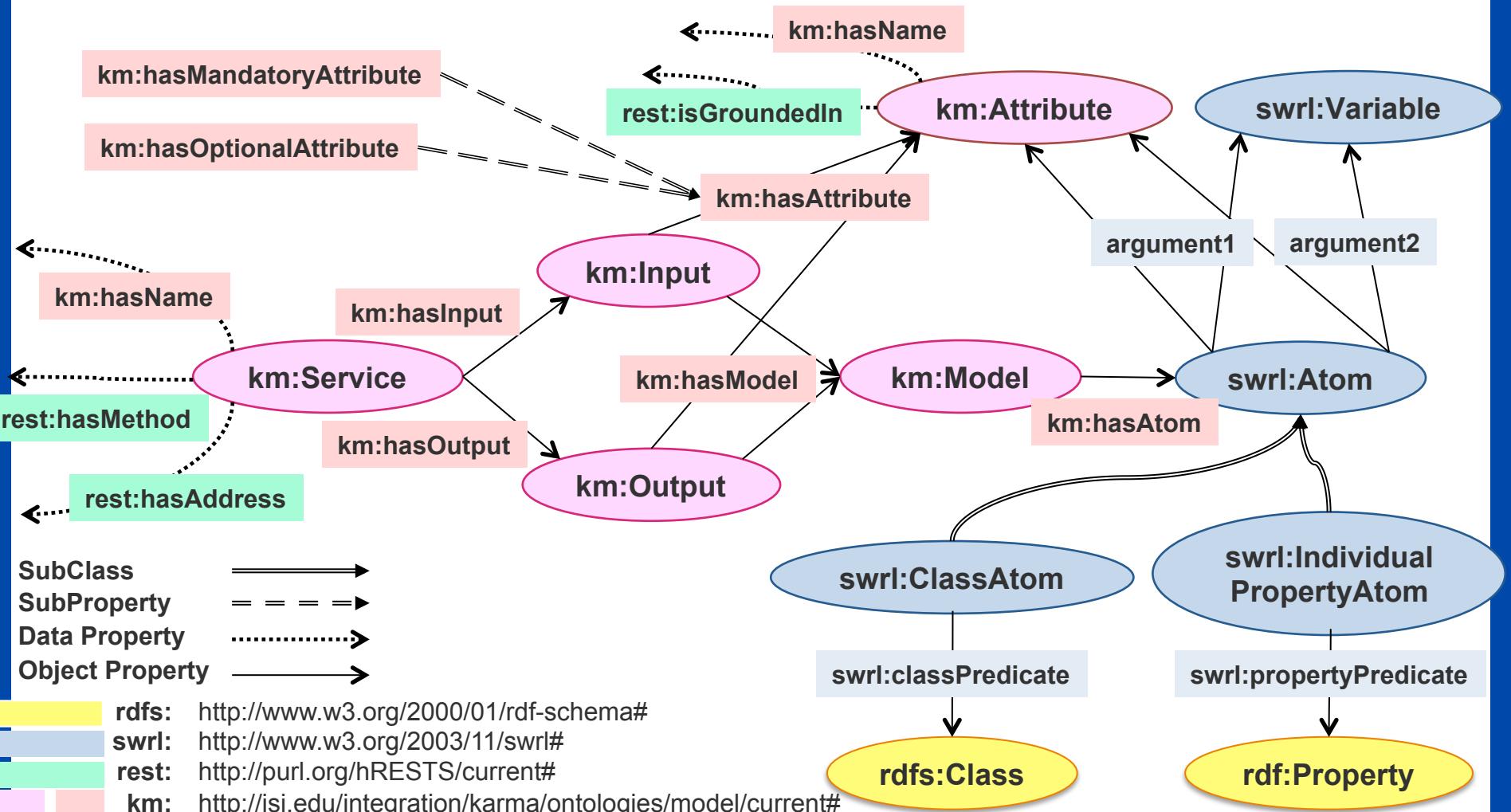
API Semantic Model in Karma



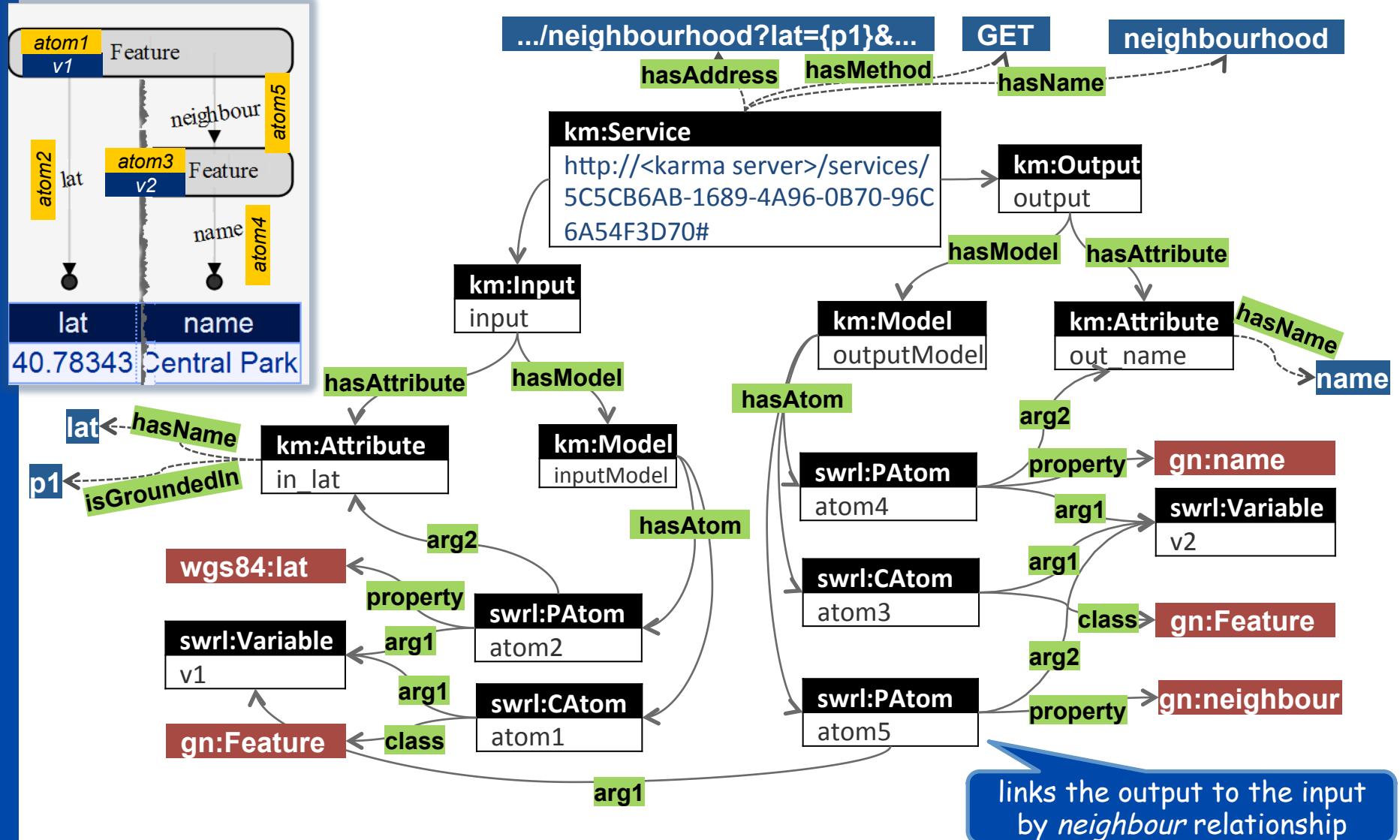
Outline

- Rapidly Integrating APIs with the LD
 - Building API Semantic Model
 - **Representing API Descriptions**
 - Building Linked APIs
- Evaluation

Service Modeling Ontology



Example: Service Description



Service Discovery

- Finds all services that take latitude and longitude as inputs
-

```
SELECT ?s WHERE {
    ?s km:hasInput [km:hasAttribute ?i1, ?i2].
    ?s km:hasInput [km:hasModel [km:hasAtom
        [swrl:classPredicate      gn:Feature; swrl:arg1 ?f1],
        [swrl:propertyPredicate wgs84:lat; swrl:arg1 ?f1; swrl:arg2 ?i1],
        [swrl:propertyPredicate wgs84:long; swrl:arg1 ?f1; swrl:arg2 ?i2]]]
}
```

Service Discovery

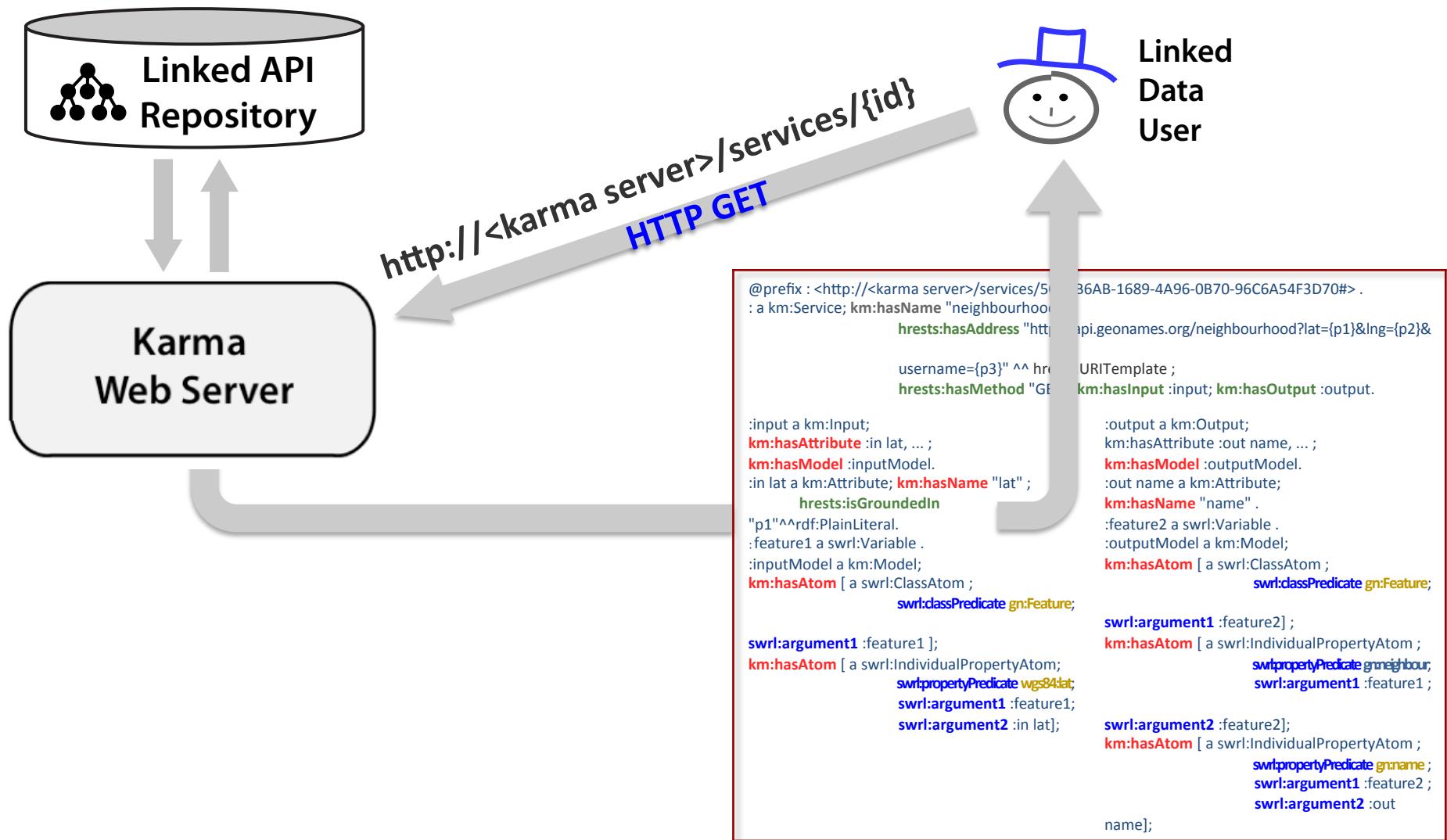
- Find services that return the neighbor feature given the latitude and longitude
-

```
SELECT ?s WHERE {
    ?s km:hasInput [km:hasAttribute ?i1, ?i2].
    ?s km:hasOutput [km:hasAttribute ?o1].
    ?s km:hasInput [km:hasModel [km:hasAtom
        [swrl:classPredicate gn:Feature; swrl:arg1 ?f1],
        [swrl:propertyPredicate wgs84:lat; swrl:arg1 ?f1; swrl:arg2 ?i1],
        [swrl:propertyPredicate wgs84:long; swrl:arg1 ?f1; swrl:arg2 ?i2]]].
    ?s km:hasOutput [km:hasModel [km:hasAtom
        [swrl:classPredicate gn:Feature; swrl:arg1 ?f2],
        [swrl:propertyPredicate gn:neighbour; swrl:arg1 ?f1;
        swrl:arg2 ?f2]]]}{}
```

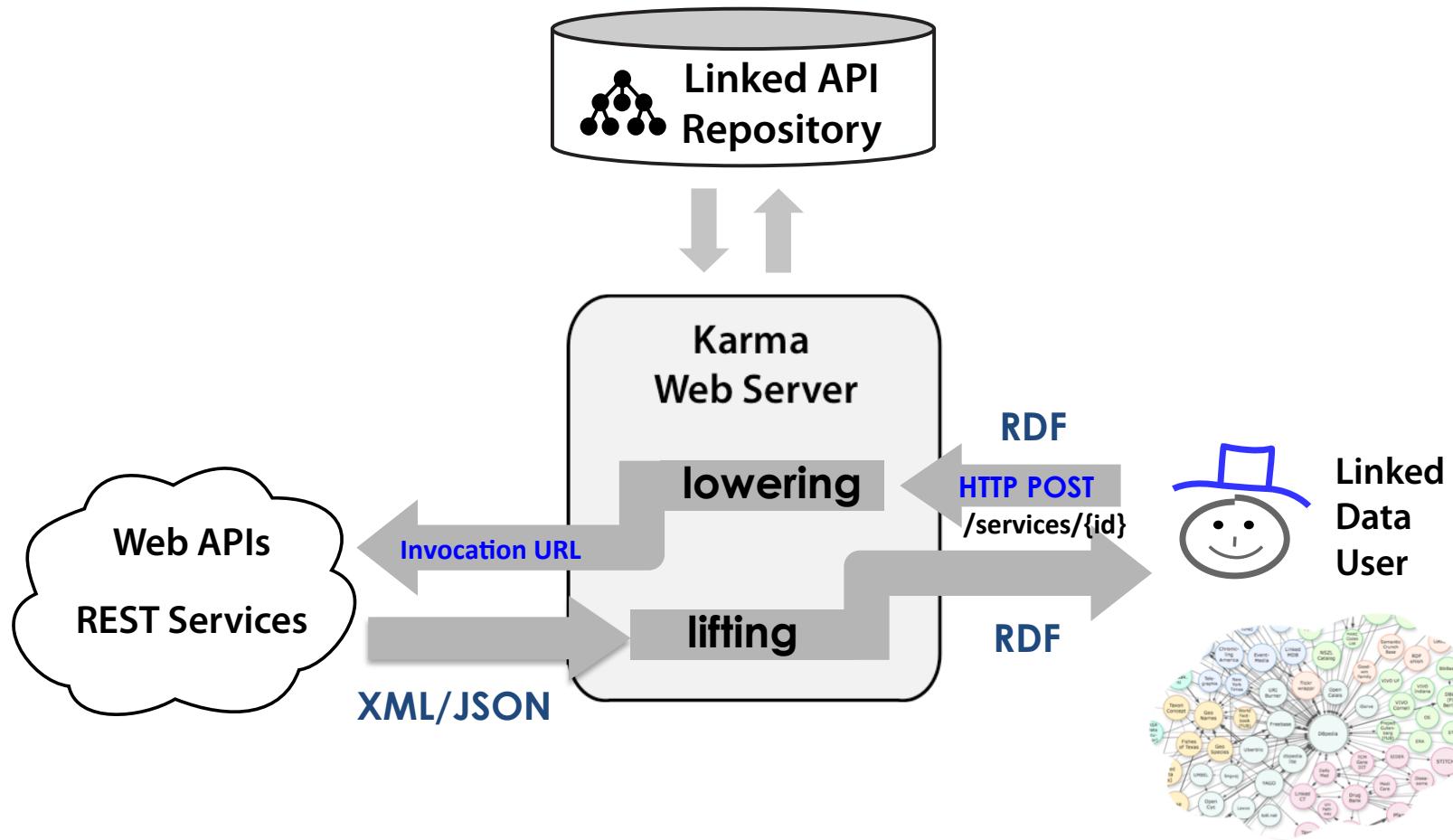
Outline

- Rapidly Integrating APIs with the LD
 - Building API Semantic Model
 - Representing API Descriptions
 - **Building Linked APIs**
- Evaluation

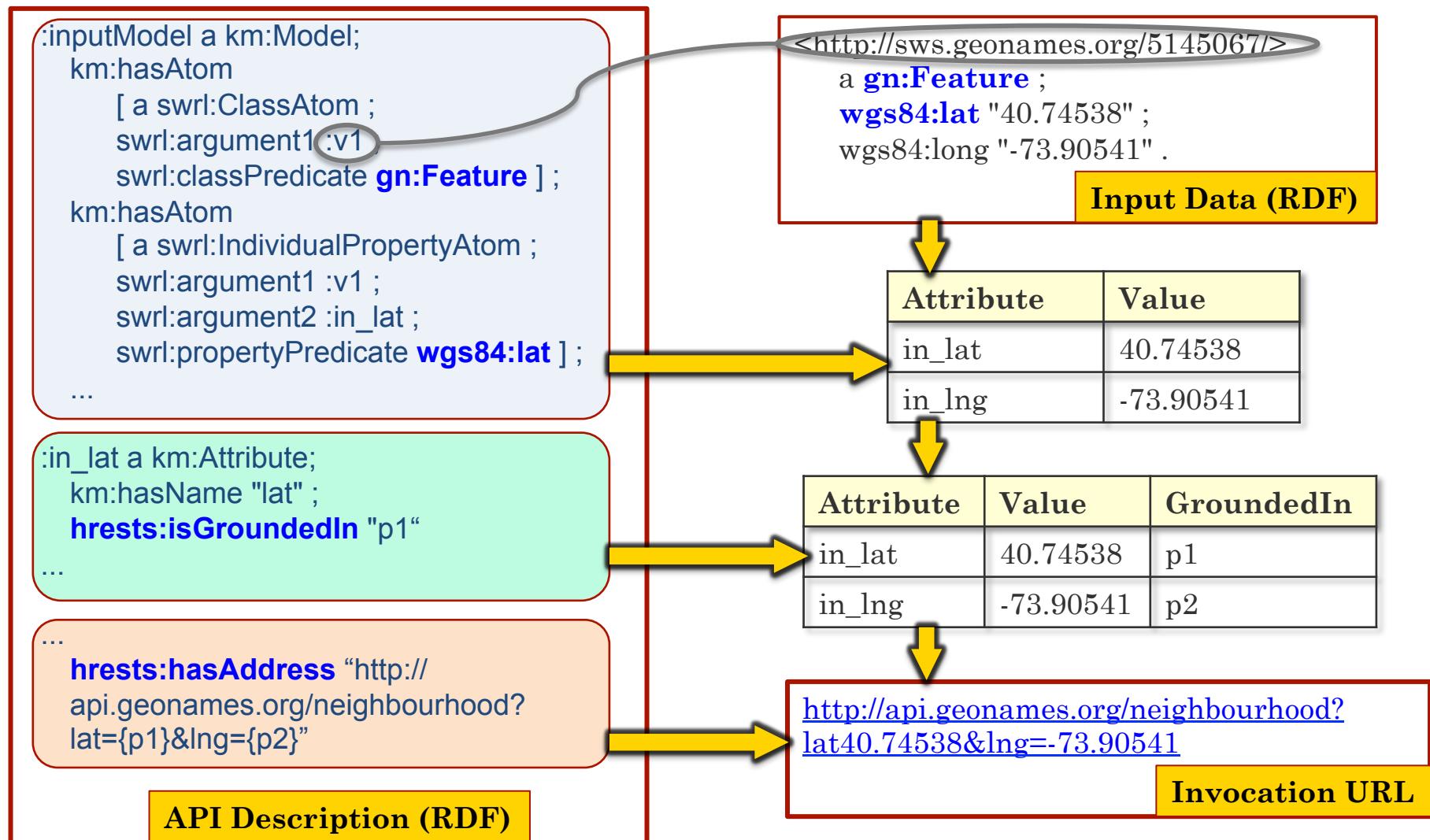
Invoking Linked APIs (GET)



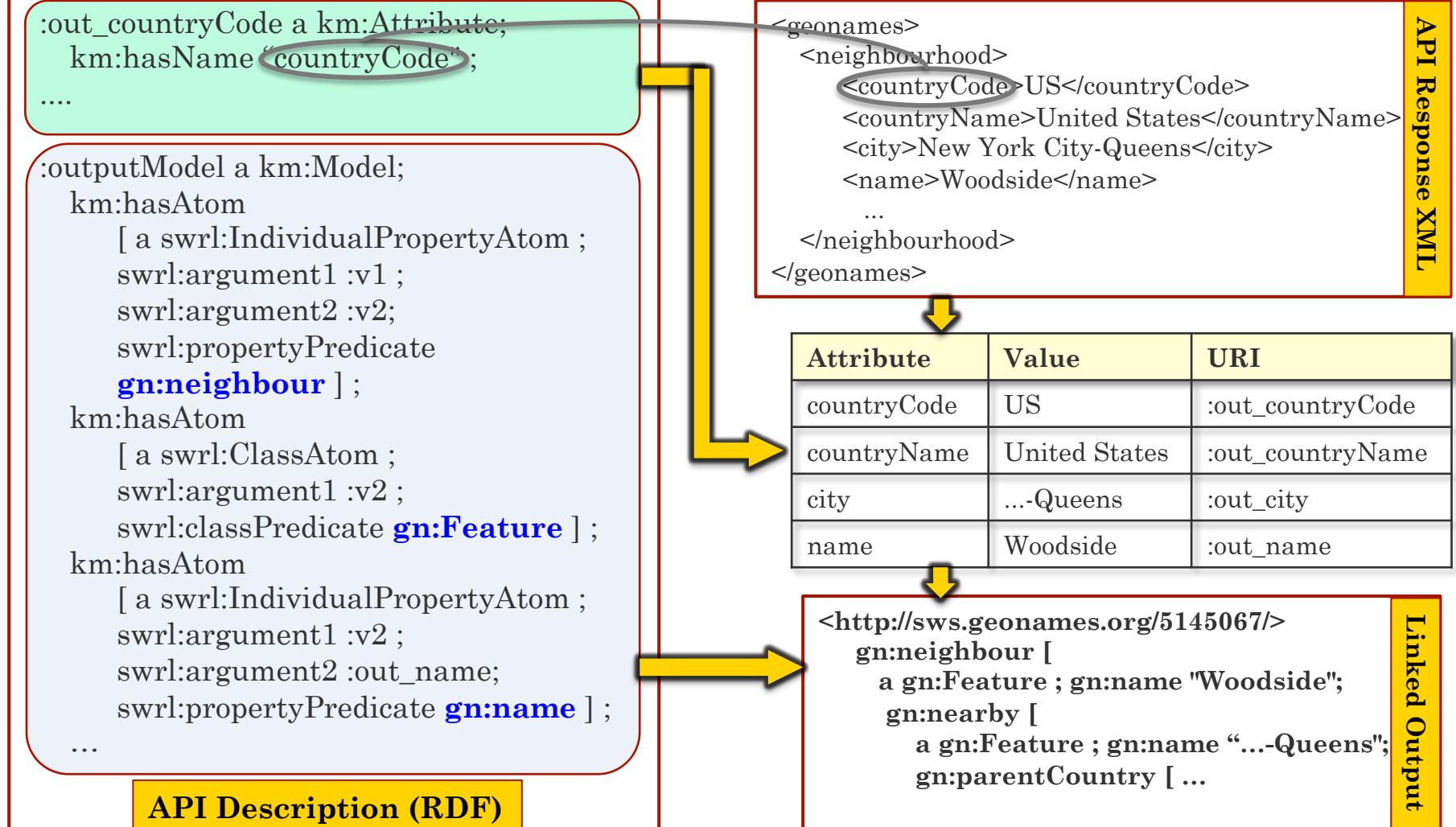
Invoking Linked APIs (POST)



Lowering



Lifting



Outline

- Rapidly Integrating APIs with the LD
 - Building API Semantic Model
 - Representing API Descriptions
 - Building Linked APIs
- Evaluation

Evaluation

Geonames API	#Examples	#Columns	Time (min)
<i>neighbourhood</i>	3	10	6
<i>neighbours</i>	2	9	5
<i>children</i>	2	10	3
<i>sibling</i>	1	9	3
<i>ocean</i>	2	3	1
<i>findNearby</i>	3	11	3
<i>findNearbyPostalCodes</i>	3	11	7
<i>findNearbyPOIsOSM</i>	3	7	3
<i>findNearestAddress</i>	3	14	6
<i>findNearestIntersectionOSM</i>	3	8	3
<i>postalCodeCountryInfo</i>	1	5	2
Total	26	97	42

Average 4 minutes to build a linked API

Summary

- Publishing Service as Linked Data [Pedrinaci et al, 2010]
 - Annotates inputs and outputs by concepts from ontologies
 - Publishes service descriptions into the LOD cloud using Minimal Service Model (MSM)
 - Helps to discover services easier
 - Cannot represent relationships between service attributes
- Linked Open Services (LOS) [Norton & Krummenacher, 2010] , Linked Data Services (LIDS) [Speiser & Harth, 2010]
 - Services that (consume and) produce linked data
 - SPARQL graph patterns to describe inputs and outputs
 - Service discovery is not straightforward
- Service Modeling in Karma [Taherian & Knoblock & Szekely & Ambite, 2012]
 - Semi-automatically building rich service descriptions
 - Publishing service descriptions into the LD cloud
 - Building linked APIs that consume and produce linked data
 - **Modeling process does not require expertise in SW technologies**