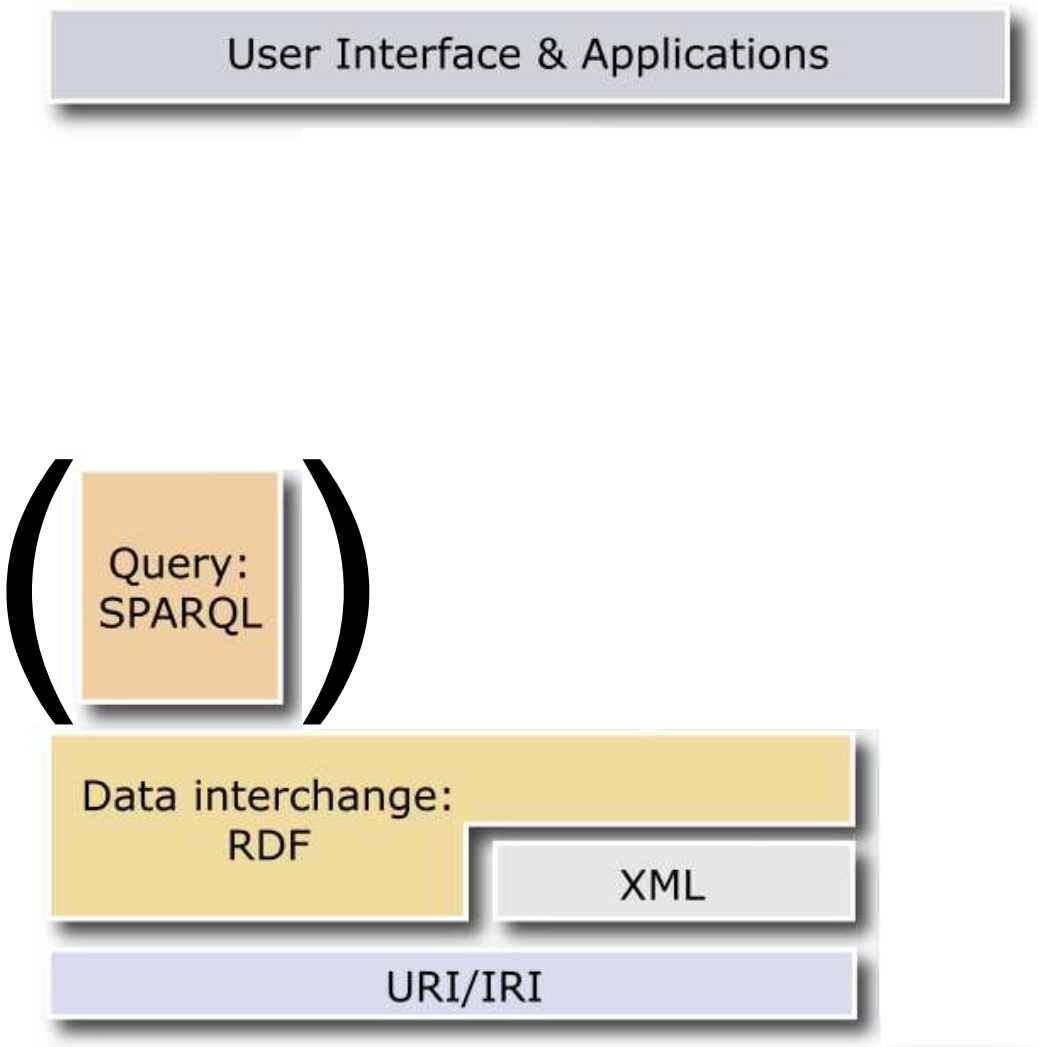
Karlsruhe Institute of Technology

Linked Data Principles for Services and Streams

Andreas Harth

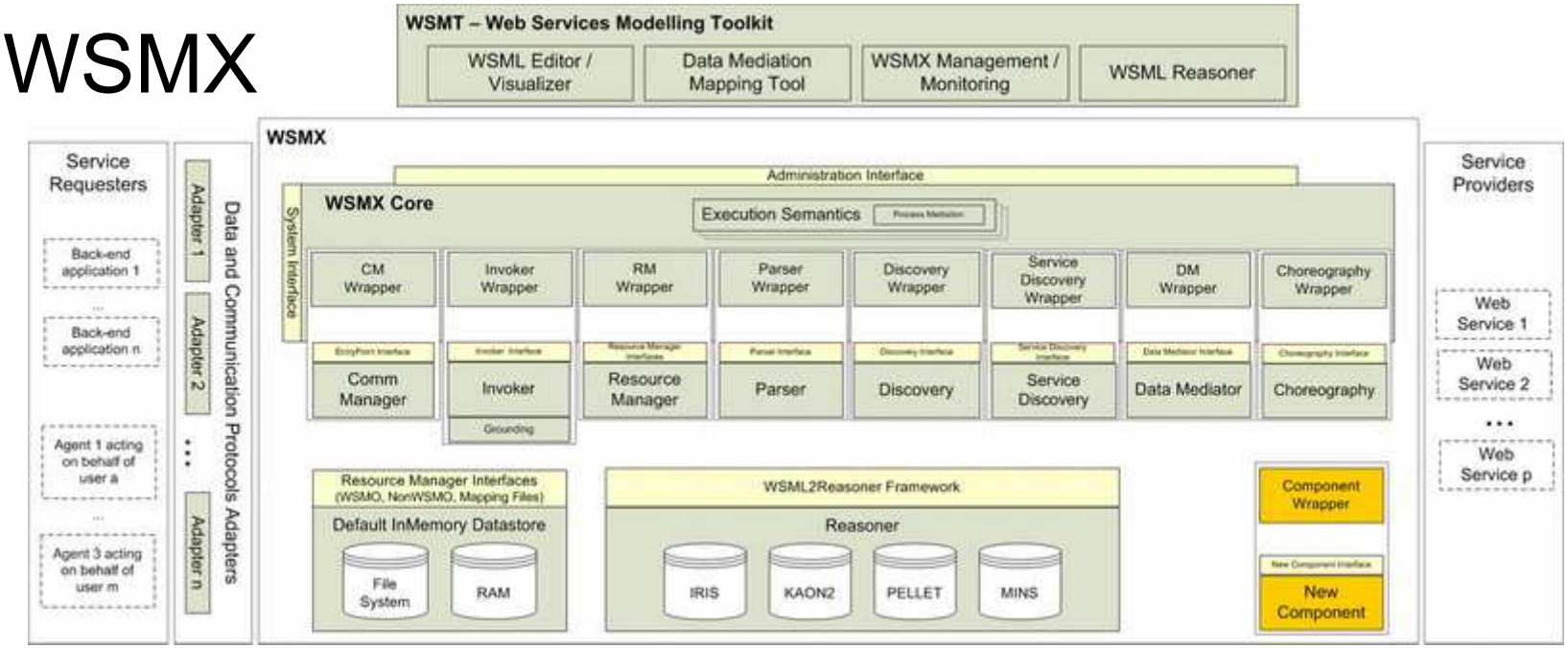
Institute AIFB, Group Prof. Studer, Knowledge Management

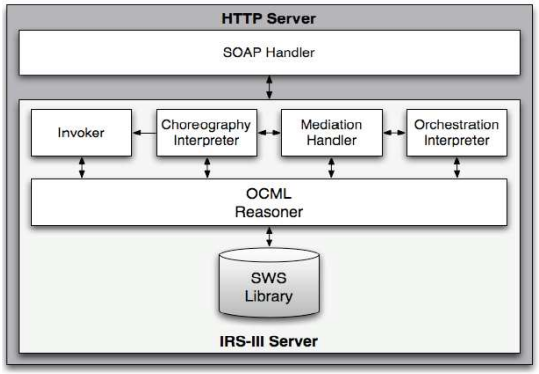
Semantic Web Application Architecture



Services

WSMX





IRS-III

Motivation

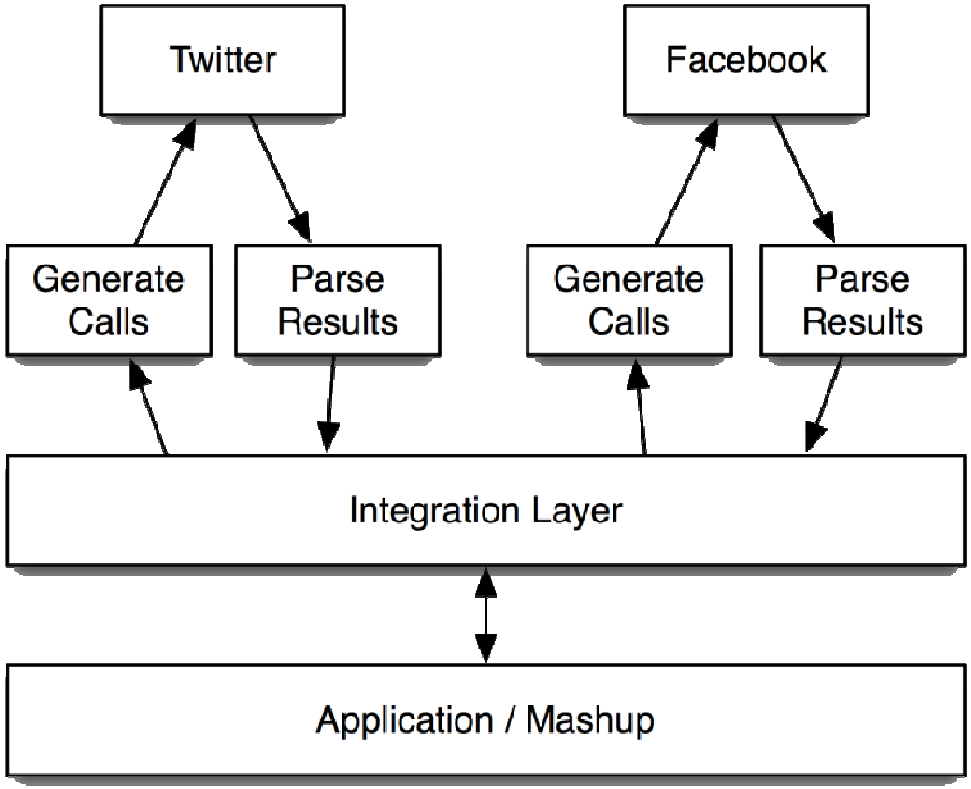
- Linked Data (LD) makes a lot of data available in the Web
- Applications typically rely on data from different sources
- With LD, integration of data sources is easy:
  - `<http://data.semanticweb.org/conference/eswc/2010>`
    `foaf:based_near`
    `<http://dbpedia.org/resource/Heraklion>`

- There are lot of applications and mashups on the net which do not have this comfort because they rely on Web APIs
  - Typically based on JSON or XML retrieved through a custom URI scheme
    Out of 3274 APIs from ProgrammableWeb, only 37 based on RDF

  - Typically not interlinked
  - Typically the information that users want: Tweets, Facebook friends, eBay auctions, Flickr images and YouTube Videos

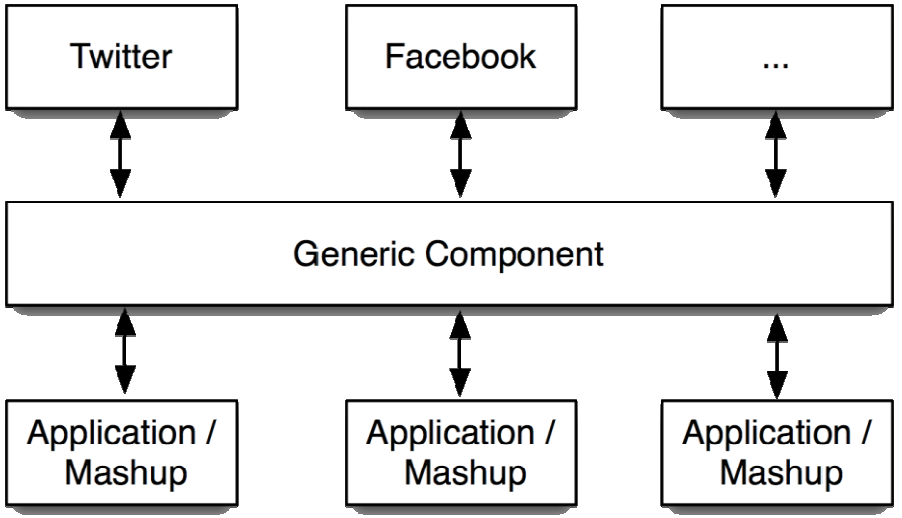
Example

- Example: Facebook and Twitter API
  - ProgrammableWeb lists 92 mashups using Twitter and Facebook
  - For both APIs: code to generate API call; code to parse JSON results into application's data model
  - Integration layer that connects information from both sources



Example

- Gluing code is written 92 times!
- For calls often libraries exist
  - But who maintains them for all different languages?
  - Wouldn't it be good to have the same call interface for all and then select the favourite generic access component?

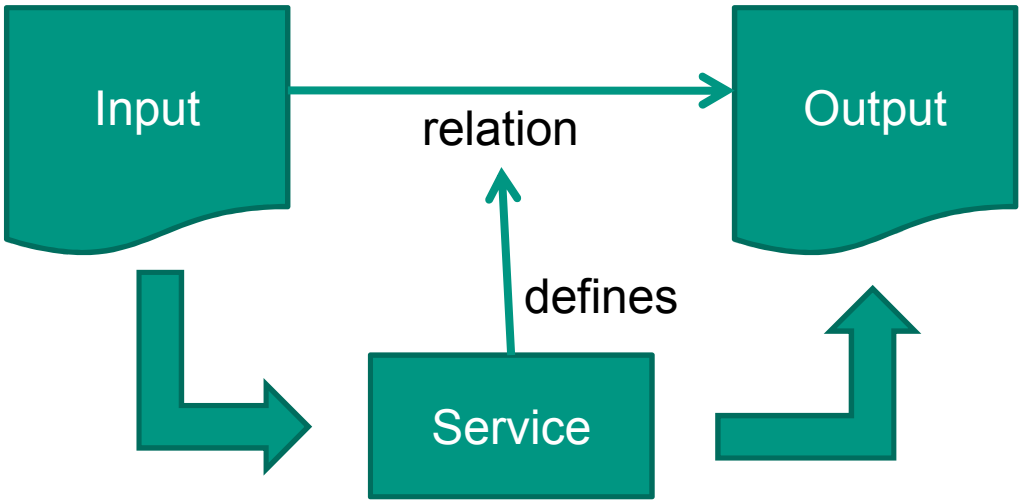


Data Services

- Not all data sources will be published as fully materialised data sets
- Reasons include:
  - Data is changing constantly (e.g. sensor data or stock quotes)
  - Data is calculated based on infinitely many inputs (e.g. route between two geographical locations)
  - Provider does not want arbitrary access (e.g. flight ticket prices, social networks)
- LIDS: Method to publish information services as Linked Data
  - SPARQL patterns describe input and output
  - URIs for service calls can be automatically created
  - Service calls can be directly interlinked with other data

Data Services Overview

- Given input, provide output
- Input and output are related in a service-specific way
- We do not consider state changes



- E.g. GeoNames findNearbyWikipedia service
  - Input: lat/lon
  - Output: places
  - Relation: output places that are *nearby* input place

Enter LIDS: Linked Data Services

- We'd like to integrate data services with Linked Data

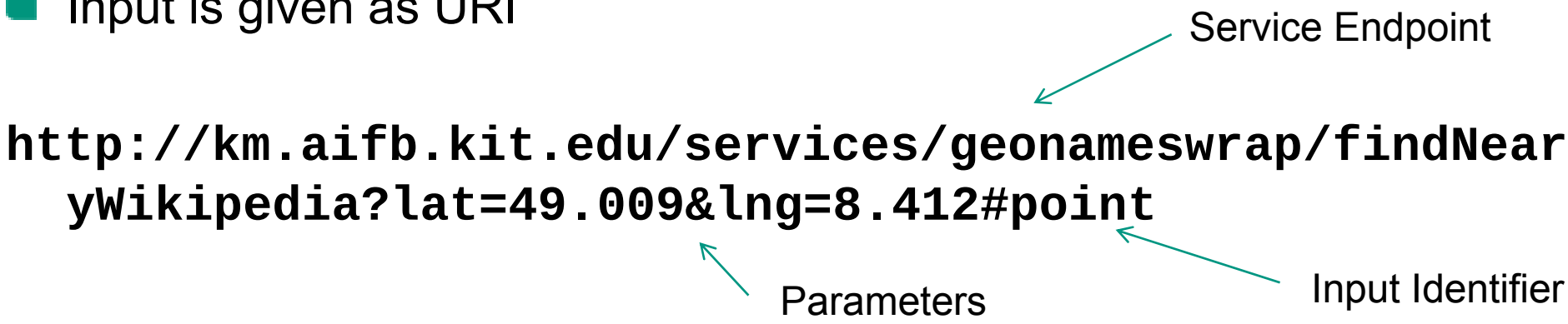
1. LIDS need to adhere to Linked Data principles

- We'd like to use data services in software programs

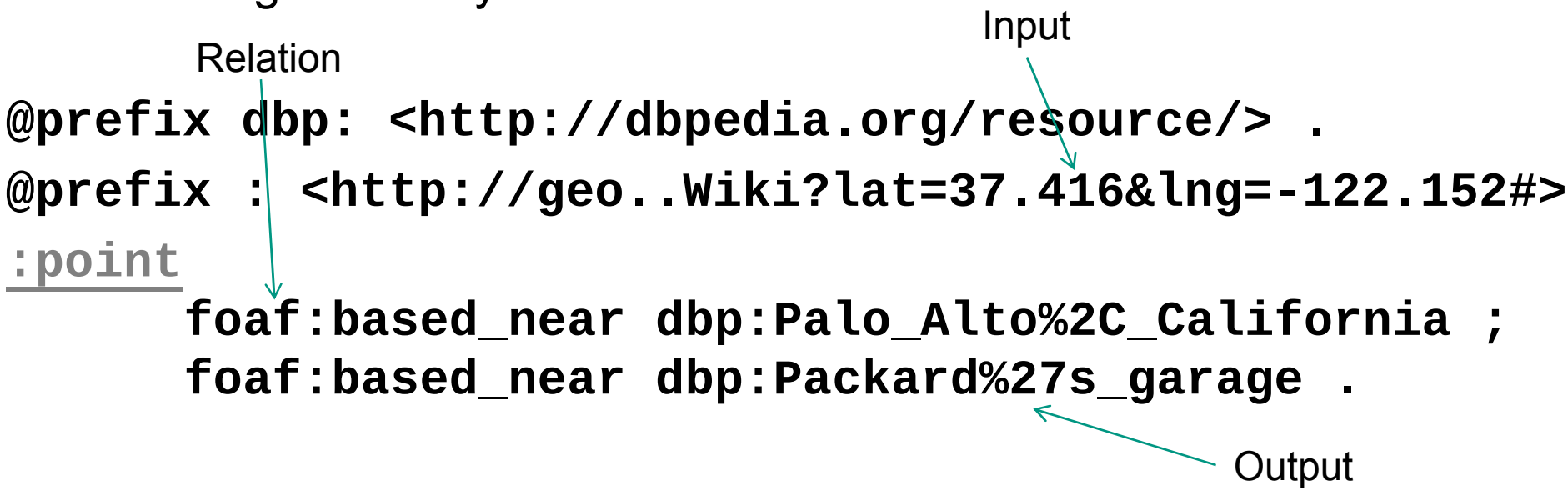
2. LIDS need machine-readable descriptions of input and output

1. Data Services as Linked Data

- Input is given as URI



- Resolving the URI yields RDF:



2. LIDS Descriptions

- LIDS characterised by
  - Endpoint URI ep, which is the base for all input entities
  - Local identifier i of input entity
  - List of parameters X_i
  - Basic graph pattern T_i describing conditions on parameters
  - Basic graph pattern T_o describing minimum output data

- Example:

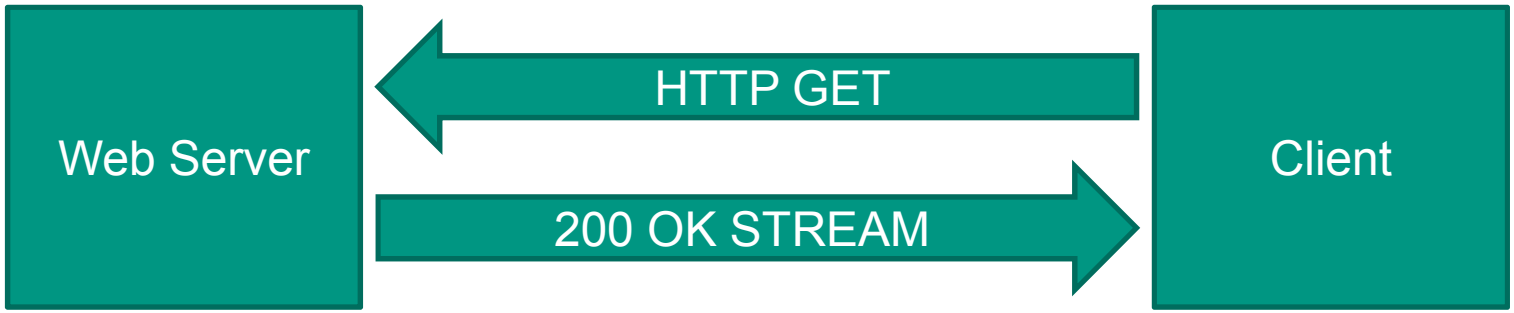
ep = **<http://geowrap.openlids.org/findNearbyWikipedia>**
i = point
X_i = {?lat, ?lng}
T_i = ?point a Point . ?point geo:lat ?lat .
?point geo:long ?lng
T_o = ?point foaf:based_near ?feature

Semantic Sensor Networks

- SSN ontology provides descriptions of sensors and observations
- Several stream engines (e.g., C-SPARQL) provide access to streams via SPARQL
- GSN (in combination with relational mapping tools such as D2R) provides access to archived sensor data
- Linked Sensor Data project provides historical data
- However, there is no defined ***protocol*** for ***live access*** to streams

Proposal: Linked Data Streams

- Use HTTP as access protocol for streams, as HTTP supports streaming of data
- Linked Data Streams use RDF as data encoding, and HTTP as access protocol
- Open HTTP connection, and then serve RDF triples ad infinitum



Example

- Source of stream

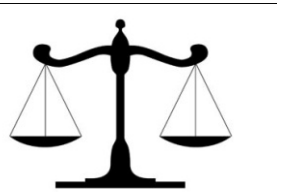
`http://events.play-project.eu/e1`

- Data (stream of triples)

```
e1:event a :avgTempEvent .
e1:event :startTime "2011-01-29"^^xsd:date .
e1:event :endTime "2011-01-31"^^xsd:date .
loc:Nice :avgTemp [ rdf:value "25" ; :event e1:event ] .
…
```

Spec draft at http://km.aifb.kit.edu/sites/lodstream

Pro's and Con's



- Use of standard HTTP servers and clients for streams
- Simple access using a web browser or other HTTP client
- Simple publication using a CGI script or Servlet
- Fits with Linked Data principles, and allows for reuse of tools and best practices (e.g., provenance tracking)
- Linking via use of URIs in the RDF stream
- Potential overhead in using HTTP and RDF (lower-level protocols and data formats might be more efficient)
  - *Javier D. Fernández, Miguel A. Martínez-Prieto, Claudio Gutierrez, and Axel Polleres. Binary RDF Representation for Publication and Exchange (HDT), W3C Member Submission 30 March 2011.*

Conclusion

- Increasing interest in real-time access to data
- Services and streams form the underlying mechanisms
- K.I.S.S. to keep barrier of entry low

- Encode parameters for services in URIs
- Stream data via HTTP
- Use RDF for returning data

Acknowledgements

- Joint work with Sebastian Speiser on Linked Data Services and Roland Stuehmer on Linked Data Streams
- Title slide image from http://www.flickr.com/photos/23209605@N00/2786126623/