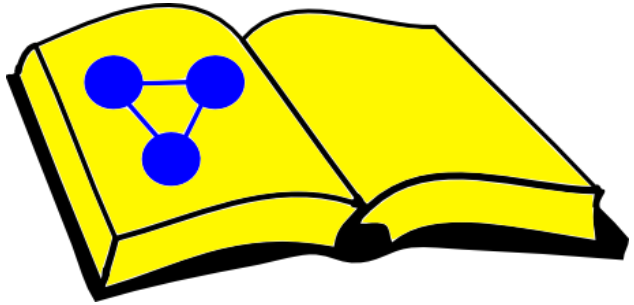


SchemEX

Creating the Yellow Pages of the LOD Cloud

Mathias Konrath, Thomas Gottron, Ansgar Scherp

WeST

UNIVERSITÄT KOBLENZ · LANDAU

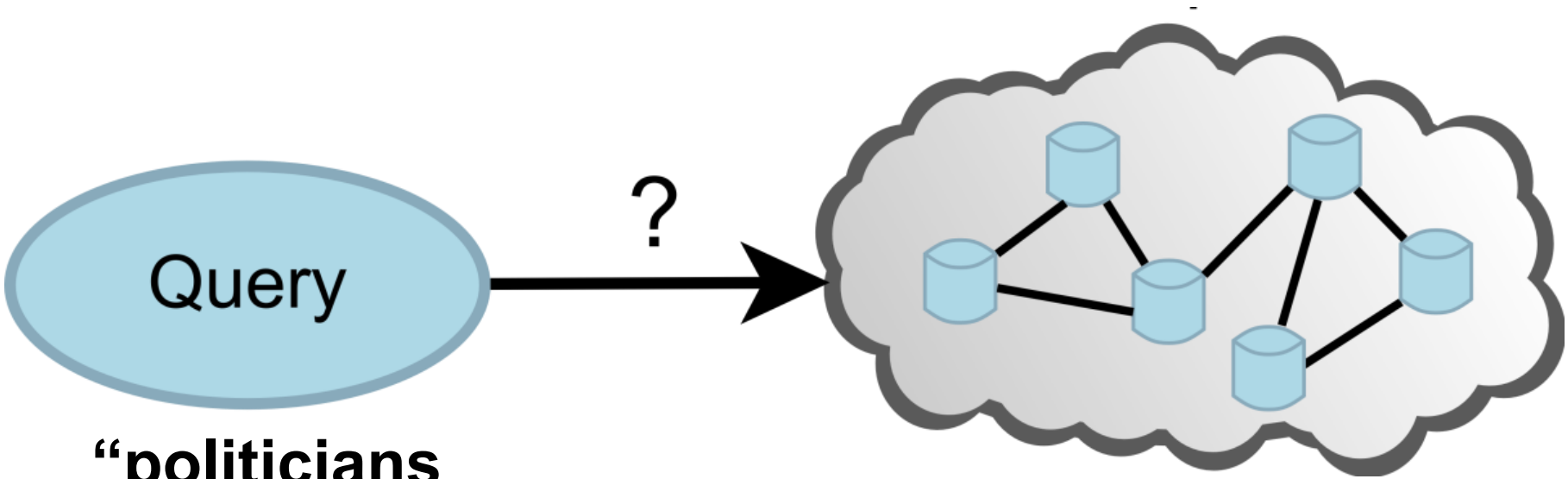
Scenario

- People who are **politicians** and **actors**

- Who else?
- Where do they live?
- Whom do they know?

Problem

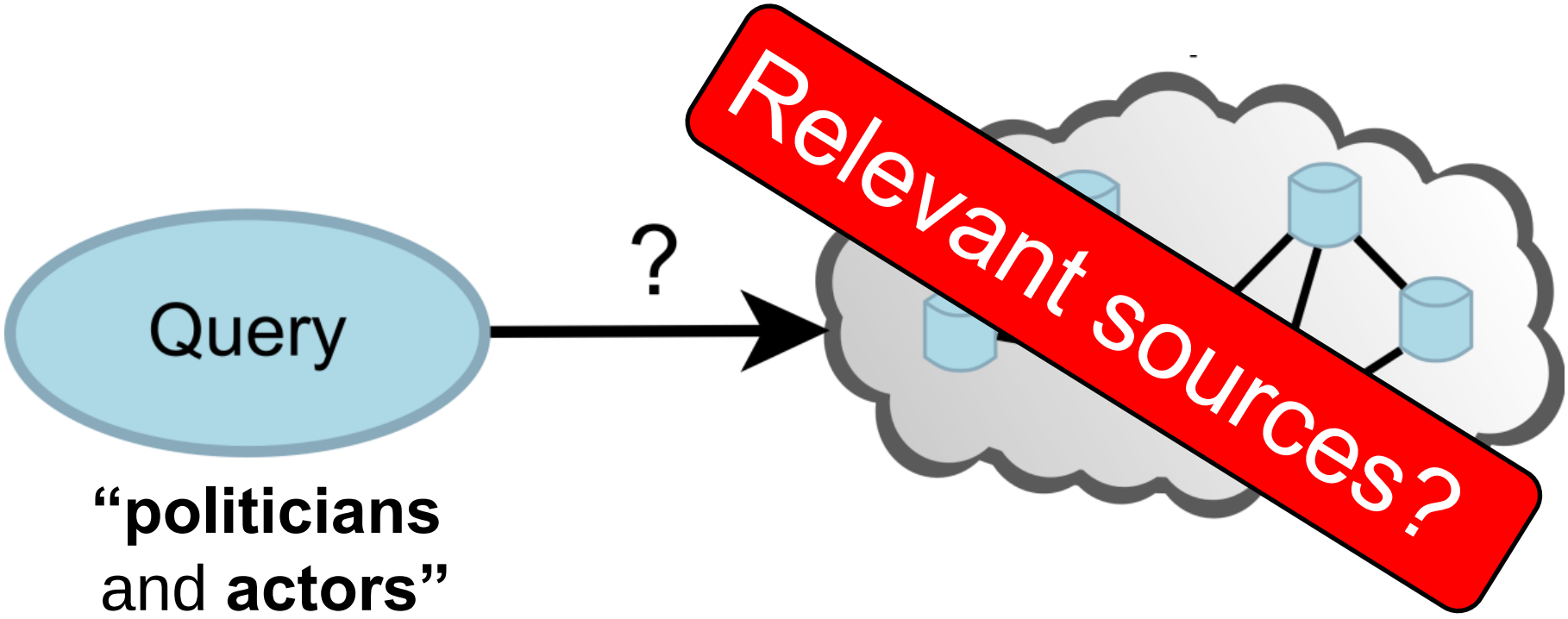
- Execute those queries on the LOD cloud
- No single federated query interface provided



“politicians
and **actors”**

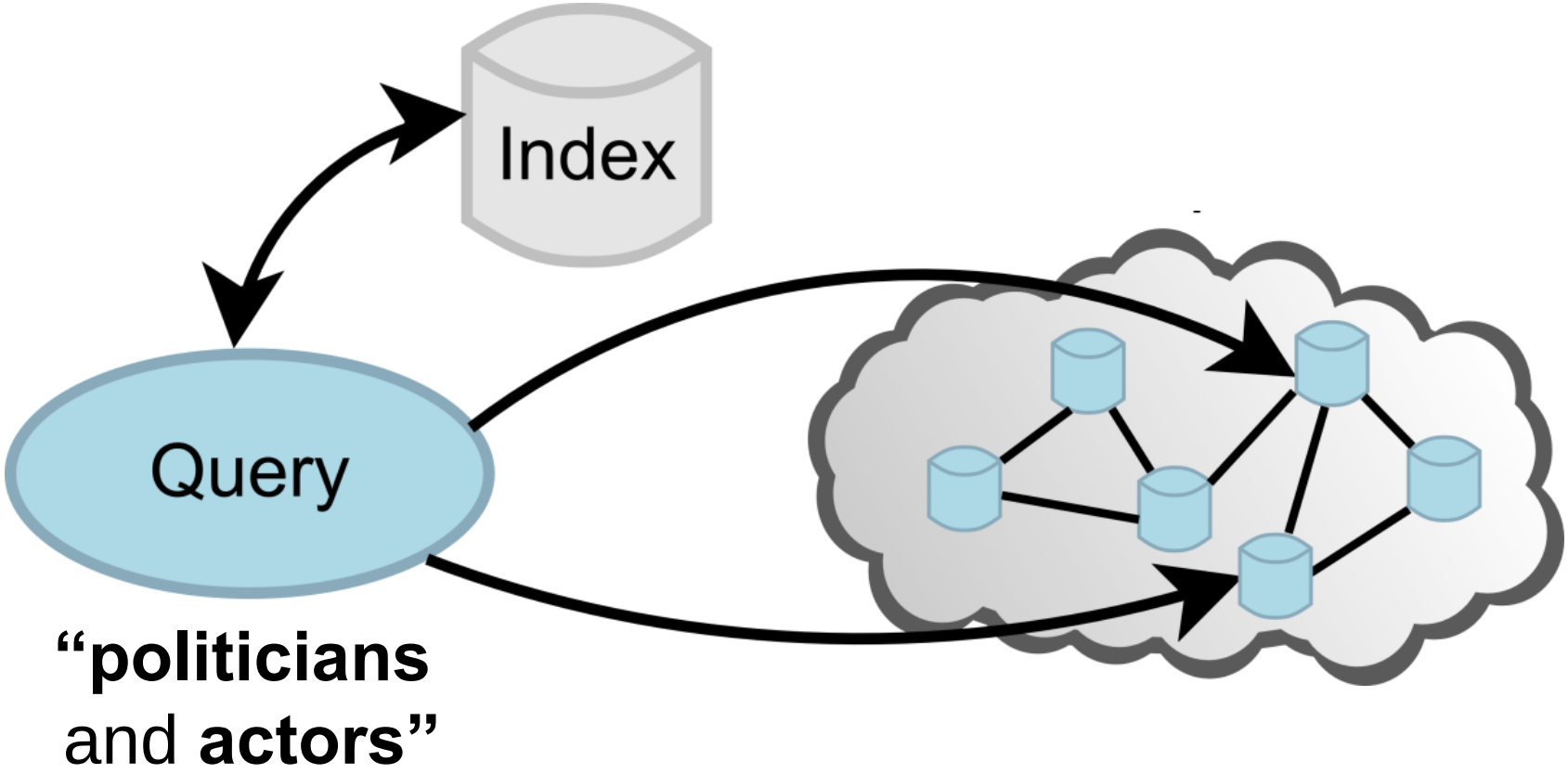
Problem

- Execute those queries on the LOD cloud
- No single federated query interface provided



Principle Solution

- Suitable index structure for looking up sources



The Naive Approach

1. Download the entire LOD cloud
2. Put it into a (really) large triple store
3. Process the data and extract schema
4. Provide lookup

- Big machinery
- Late in processing the data
- High effort to scale with LOD cloud

The Naive Approach

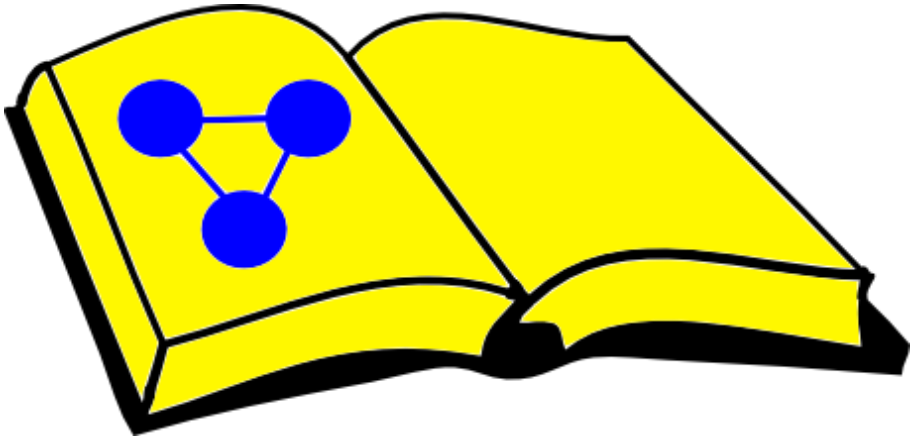
1. Download the entire LOD cloud
2. Put it into a (really) large triple stor
3. Process the data and extra
4. Provide lookup

- Big n
- ssing the data
- effort to scale with LOD cloud

Can we do smarter?

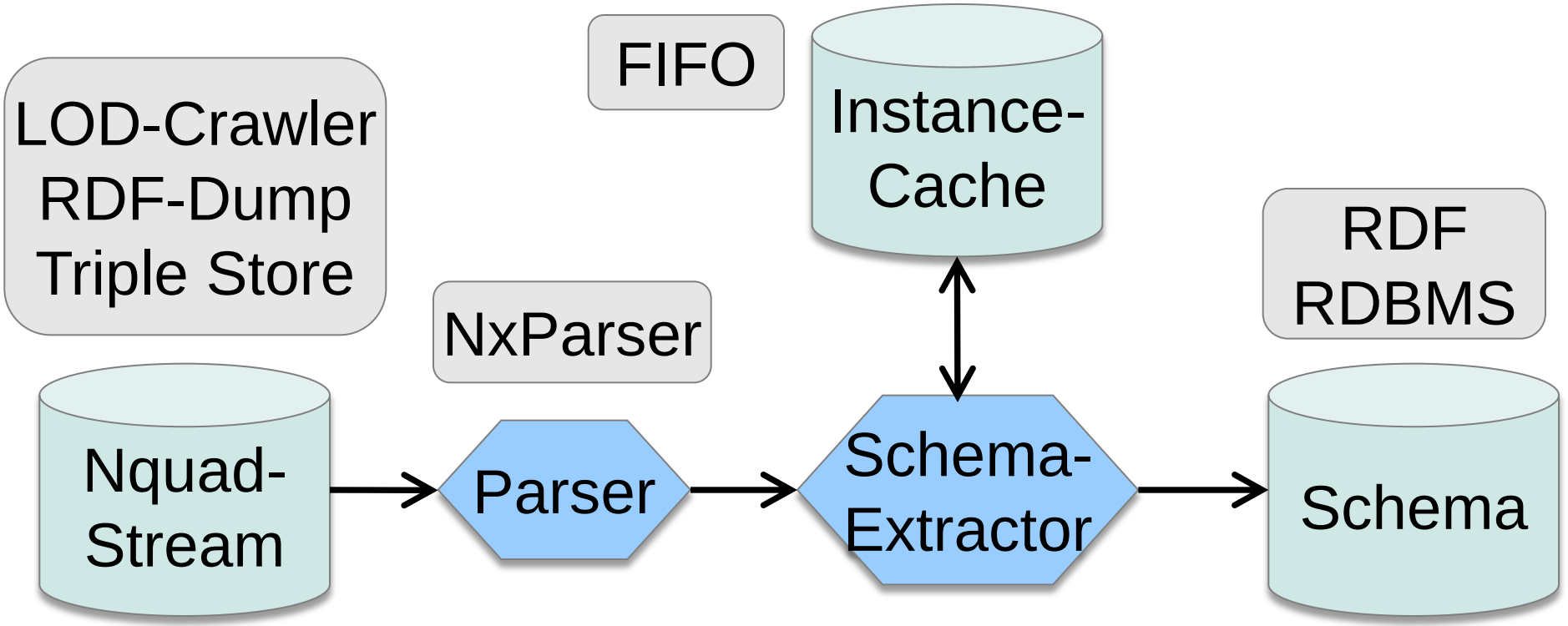
UNIVERSITÄT KOBLENZ · LANDAU WeST

Yes, we can ...



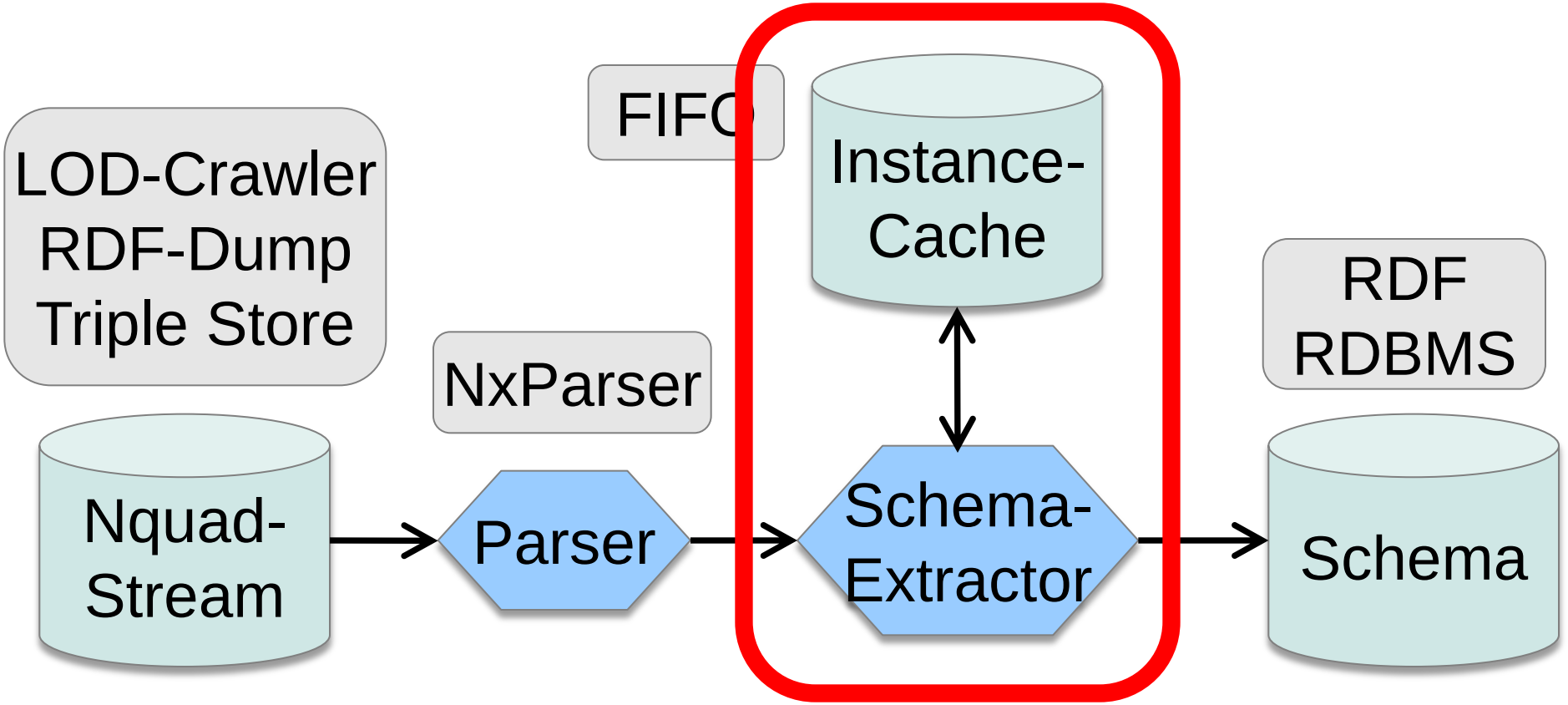
The SchemEX Approach

- Stream-based schema extraction
- While crawling the data



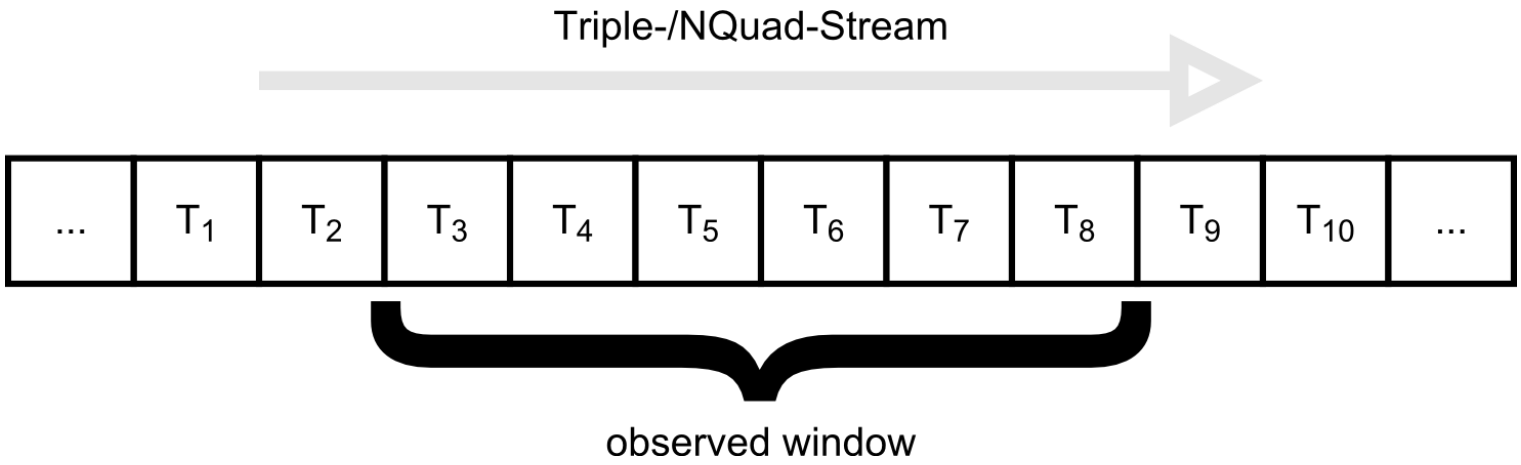
The SchemEX Approach

- Stream-based schema extraction
- While crawling the data



Efficient Instance Cache

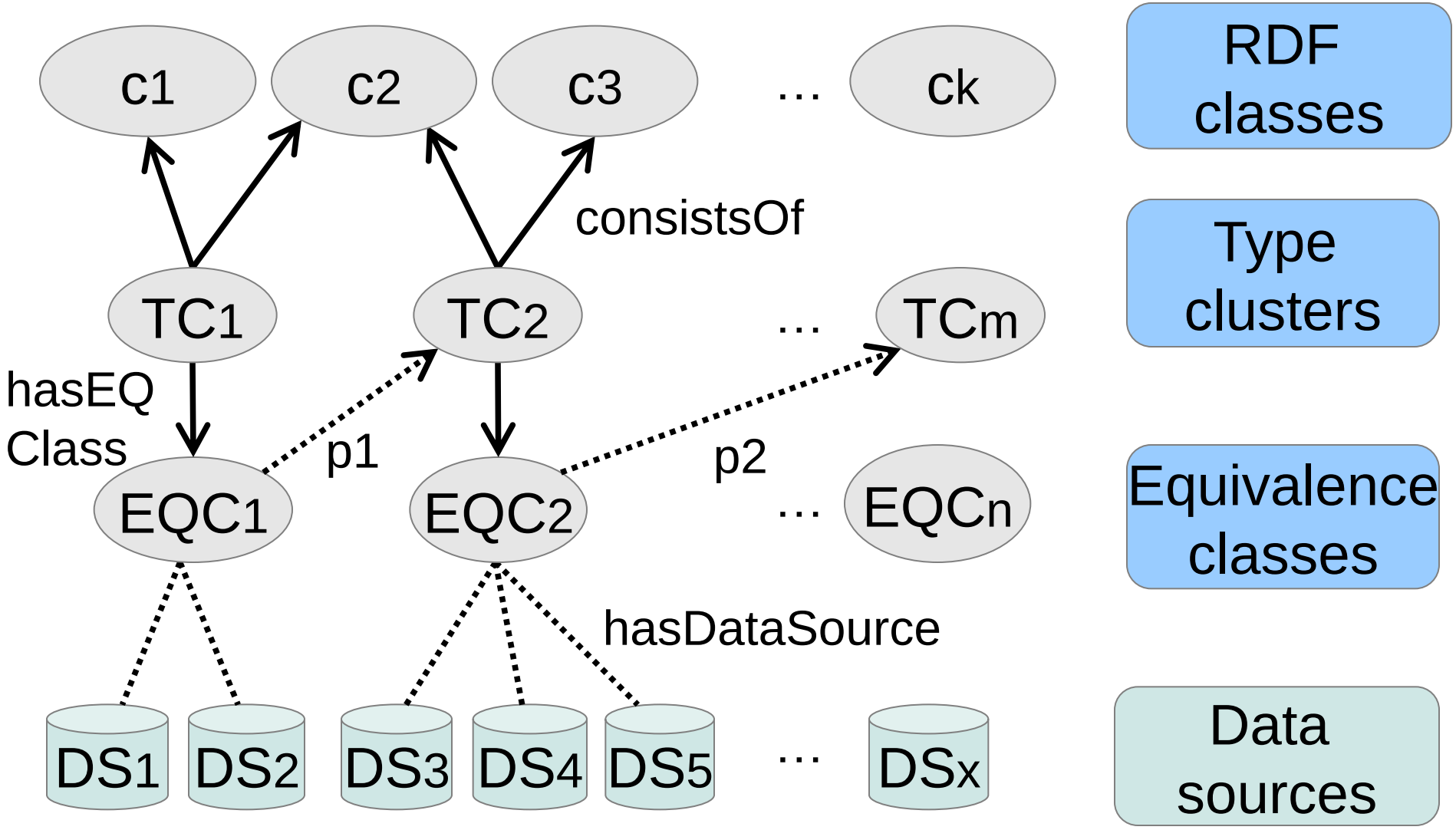
- Observe a quadruple stream from LD spider



- Ring queue, backed up by a hash map
- Organizes triples with same subject URI
- Dismiss oldest, when cache full (FIFO)

→ Runtime complexity O(1)

Building the Schema and Index



Computing SchemEX: TimBL Data Set

- Analysis of a smaller data set
- 11 M triples, TimBL's FOAF profile
- LDspider with ~ 2k triples / sec

- Different cache sizes: 100, 1k, 10k, 50k, 100k
- Compared SchemEX with reference schema
- Index queries on all Types, TCs, EQCs
- Good precision/recall ratio at 50k+

Computing SchemEX: Full BTC 2011 Data

	1st billion	2nd billion	full dataset
#triples	1 billion	1 billion	2.17 billion
#instances	187.7M	222.6M	450.0M
#data sources	13.5M	9.5M	24.1M
#type clusters	208.5k	248.5k	448.6k
#equivalence classes	0.97M	1.14M	2.12M
#triples index	29.1M	24.8M	54.7M
Compression ratio	2.91%	2.48%	2.52%
runtime (hh:mm)	6:51	6:05	15:16
average runtime per 10M chunk	247 s	219 s	252 s
standard deviation	80 s	12 s	57 s
#triples/sec.	40.5k	45.6k	39.5k

Cache size: 50 k

Computing SchemEX: Full BTC 2011 Data

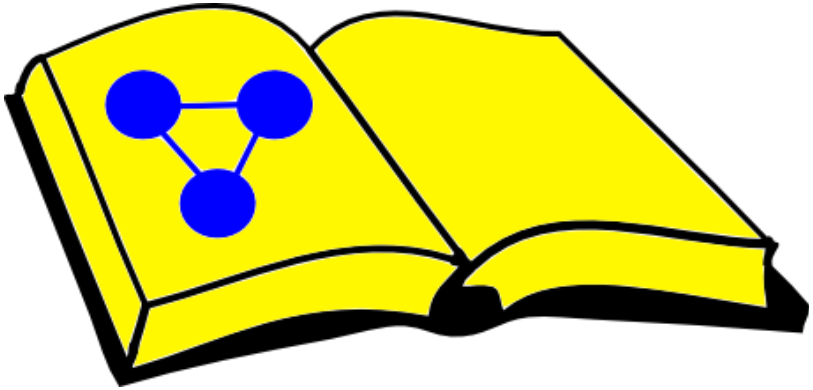
	1st billion	2nd billion	full dataset
#triples	1 billion	1 billion	2.17 billion
#instances	187.7M	222.6M	450.0M
#data sources	13.5M	9.5M	24.1M
#type clusters	208.5k	248.5k	448.6k
#equivalence classes	0.97M	1.14M	2.12M
#triples index	29.1M	24.8M	54.7M
Compression ratio	2.91%	2.48%	2.52%
runtime (hh:mm)	6:51	6:05	15:16
average runtime per 10M chunk	247 s	219 s	252 s
standard deviation	80 s	12 s	57 s
#triples/sec.	40.5k	45.6k	39.5k

Cache size: 50 k

UNIVERSITÄT KOBLENZ · LANDAU WeST

Conclusions: SchemEX

- Stream-based approach to schema extraction
- Scalable to arbitrary amount of Linked Data
- Applicable on commodity hardware
  (4GB RAM, standard single CPU)



- Lookup-index to find relevant data sources
- Support federated queries on the LOD cloud