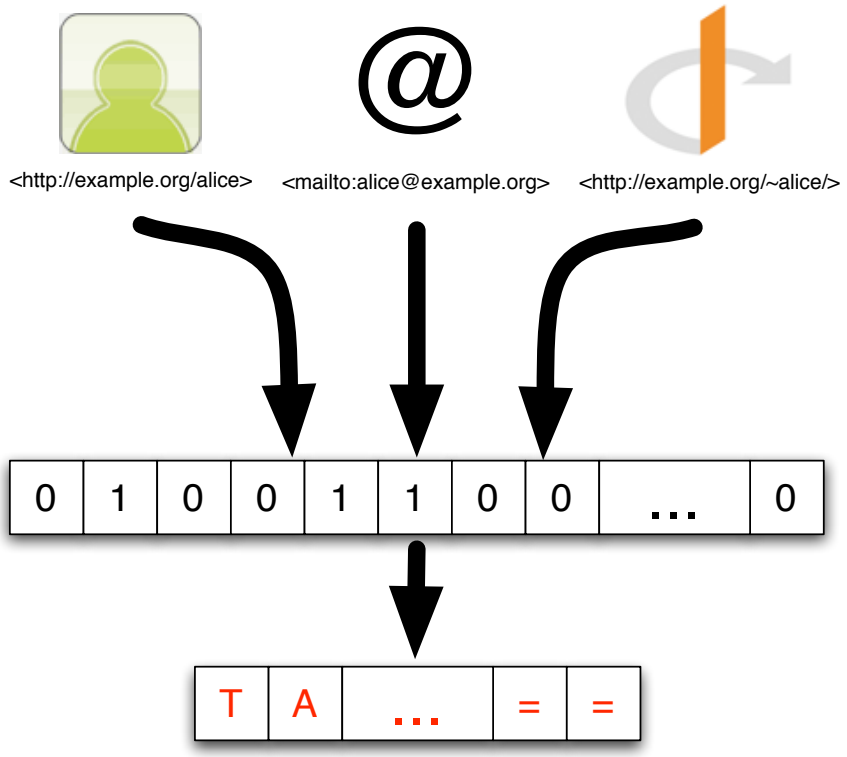
Toward Distributed Query Answering in SPARQL

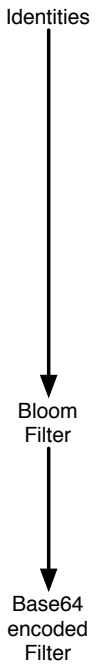
Gregory Williams

Past Research

- SPARQL Extensions
  - On-the-fly extension functions
  - Temporal Queries
  - Intermediate Result Passing

Intermediate Result Passing: Bloom Filters





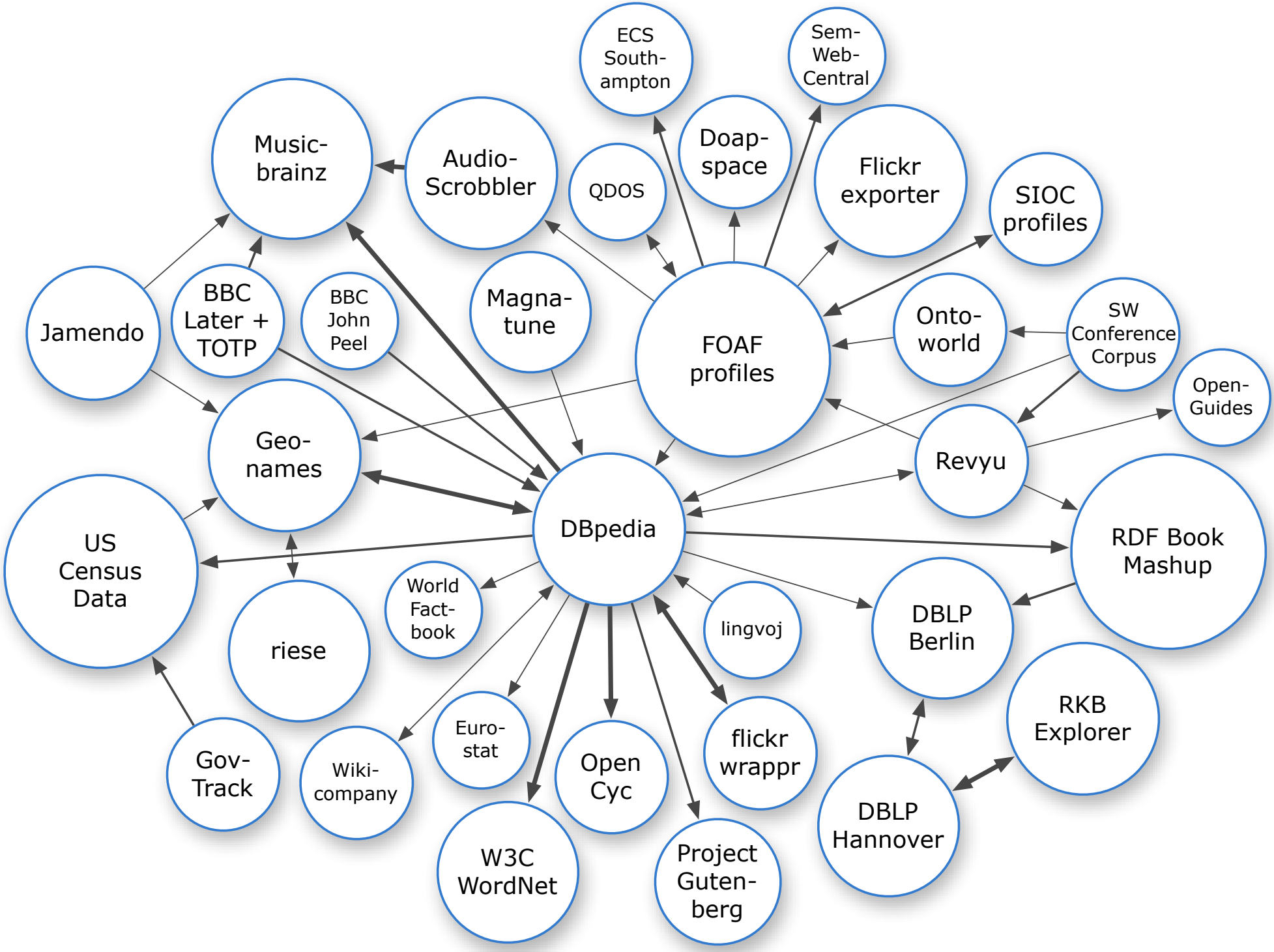

```
SELECT ?person ?phone
WHERE {
  ?person a foaf:Person ;
          foaf:phone ?phone .
  FILTER(
    :bloom(
      ?person, "TA...=="
    )
  )
}
```

Current Research

- Distributed Query Answering
  - Pose queries to a set of endpoints
  - Optimize for:
    - Co-located data
    - Early results

Optimization Strategy

- Co-located data
  - Send optimistic sub-queries to individual endpoints
  - Join sub-query results to produce early query results
- Early results
  - Expect sub-queries to be answerable quickly (data provider knows more about data than we do)


ECS South-ampton
Sem-Web-Central
Music-brainz
Audio-Scrobbler
QDOS
Doap-space
Flickr exporter
SIOC profiles
Jamendo
BBC Later + TOTP
BBC John Peel
Magna-tune
FOAF profiles
Onto-world
SW Conference Corpus
Open-Guides
Geo-names
Revyu
US Census Data
DBpedia
RDF Book Mashup
World Fact-book
lingvoj
DBLP Berlin
riese
RKB Explorer
Euro-stat
Open Cyc
flickr wrappr
Gov-Track
Wiki-company
DBLP Hannover
W3C WordNet
Project Guten-berg

Example

```
SELECT ?university ?city ?map
WHERE {
      []  dbp:type dbpedia:Private_university ;
          rdfs:label ?university ;
          dbp:city [
              rdfs:label ?city ;
              geo:locationMap ?map
          ] .
}
```

Example

```
SELECT ?university ?city ?map
WHERE {
    []  dbp:type dbpedia:Private_university ;
        rdfs:label ?university ;
        dbp:city [
            rdfs:label ?city ;
            geo:locationMap ?map
        ] .
}
```

Example

```
SELECT ?university ?city ?map
WHERE {
    []  dbp:type dbpedia:Private_university ;
        rdfs:label ?university ;
        dbp:city [
            rdfs:label ?city ;
            geo:locationMap ?map
        ] .
}
```

Example

```
SELECT ?university ?city ?map
WHERE {
    []  dbp:type dbpedia:Private_university ;
        rdfs:label ?university ;
        dbp:city [
            rdfs:label ?city ;
            geo:locationMap ?map
        ] .
}
```

Example: Sub-query

```
DBPedia:
SELECT ?university ?c
WHERE { [] dbp:type dbpedia:Private_university ;
          rdfs:label ?university ;
          dbp:city ?c }

Geonames:
SELECT ?c ?city ?map
WHERE { ?c rdfs:label ?city ;
          geo:locationMap ?map }
```

Current Work

- Develop cost model for executing (possibly remote) SPARQL patterns
- Use cost model to compare potential sub-queries
- Develop and analyze heuristics for sub-query execution (which and how many sub-queries to execute before using naïve query)
- Use intermediate result passing work to allow more efficient query execution