



Interlinking Distributed Social Graphs

Matthew Rowe
OAK Group
Department of Computer Science
University of Sheffield, UK



Organisations,
Information and
Knowledge

Outline

- Problems and Motivation
- Requirements
- Approach
 - Social Graph Exportation
 - Social Graph Enrichment
 - Social Graph Aggregation
 - Graph Reasoning
 - Producing Linked Data
 - Social Graph Control
- Experiments
 - Datasets
 - Results
- Conclusions



Problems/Issues

- Social web and web 2.0 platforms and services allow an individual to enrich their online persona
 - Lack of functionality to export social graphs from such platforms
 - Access to data is restricted, hidden within a **walled garden**
- Web users maintain a profile on many different web platforms
 - Decentralisation of identity details
 - Each platform contains a different facet of their online identity
 - Different subsets of contacts, with some overlap
 - Lack of functionality to link together such information from multiple locations



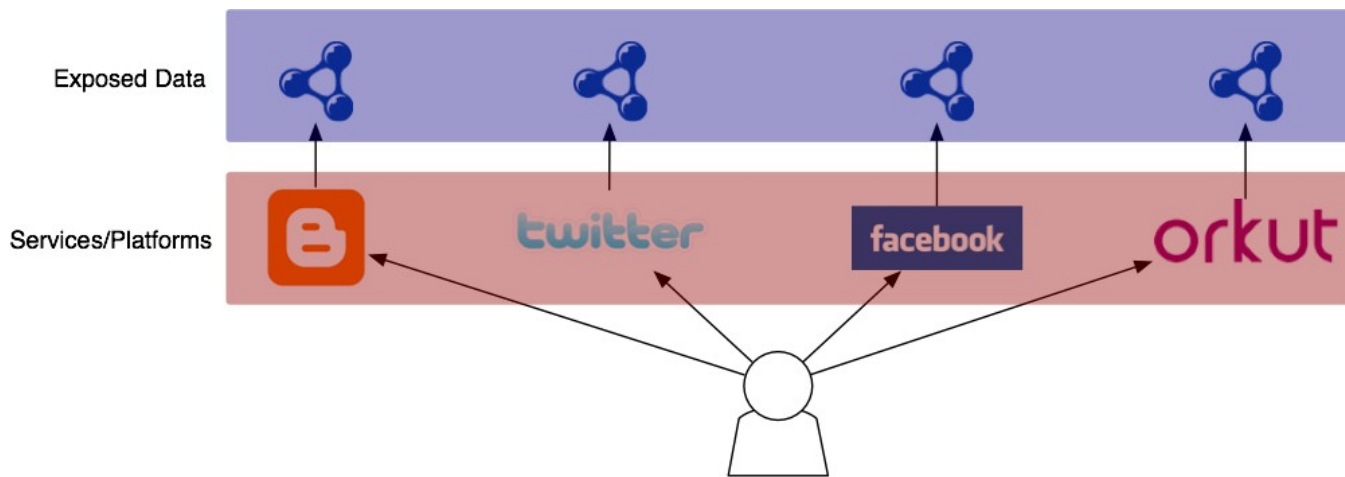
Motivation

- Interlinked social graphs would allow:
 - Importing existing contact lists when signing up for a new service
 - Establishing trust networks through transitive relationships
 - Recommendations and suggestions could be made using the interlinked data
 - Ability to break down the [wall](#)
- An interlinked social graph maintains a decentralised description of a person's online identity
 - Individual social graphs are linked together from multiple locations
 - URIs provide references to additional information without duplicating data
 - Able to maintain a rich representation of a person's online identity



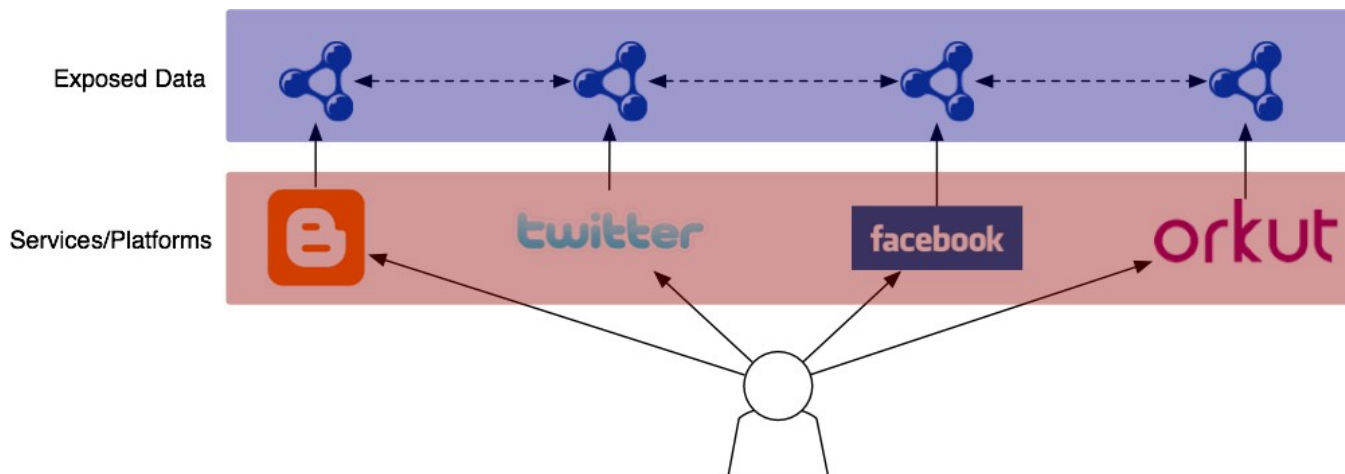
Requirements

- The approach to interlinking distributed social graphs is divided into two stages:
 - Creation of social graphs from individual social web platforms



Requirements

- The approach to interlinking distributed social graphs is divided into two stages:
 - Creation of social graphs from individual social web platforms
 - Interlinking of the created social graphs

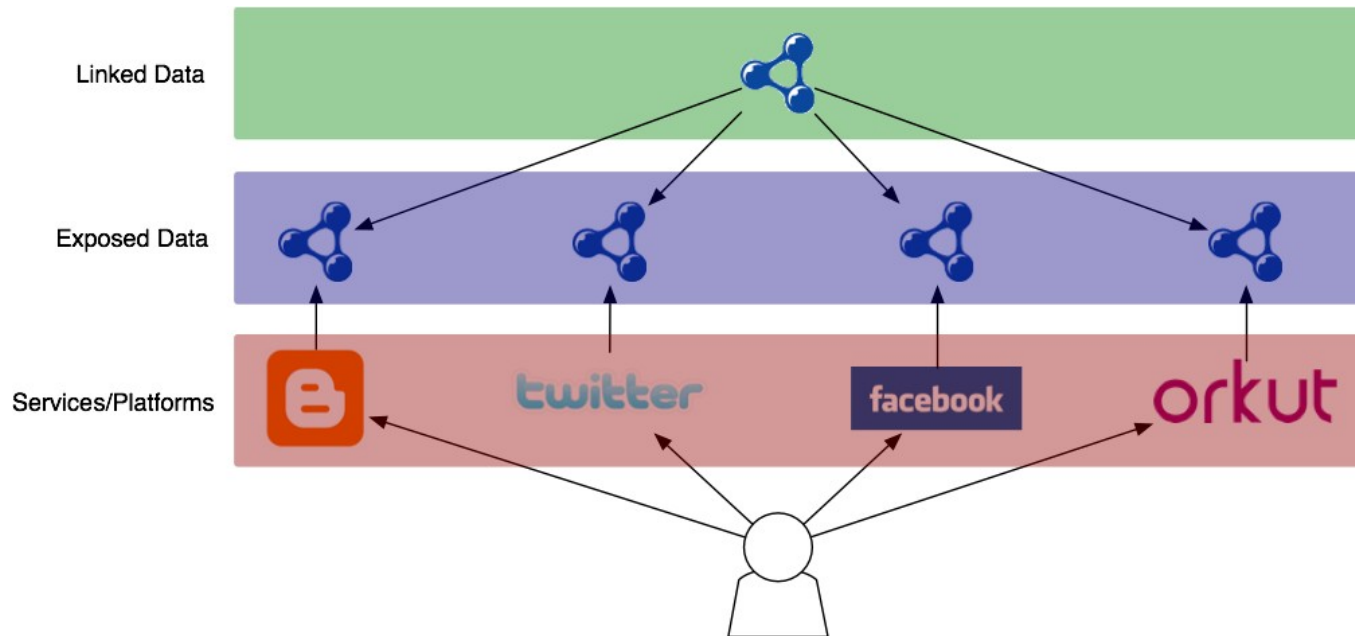


Requirements

- The approach to interlinking distributed social graphs is divided into two stages:
 - Creation of social graphs from individual social web platforms
 - Interlinking of the created social graphs
- Such an approach must meet the following requirements:
 - Export social data contained within data silos into the same semantic format
 - Link person instances from separate social networks referring to the same real world person
 - Maximise the number of correct links whilst minimising the number of incorrect links
 - Publish a decentralised linked social graph



Requirements



Social Graph Exportation

- The majority of social web and web 2.0 platforms store information within a 'walled garden' data silo
 - Prevents unwanted parties viewing my data
 - Hinders data exportation when I wish to transport it
- Climbing the wall involves interacting with the service's API and handling the received response
 - Authentication: Can this party access this data?
 - Return response: XML schema, JSON, etc



Social Graph Exportation

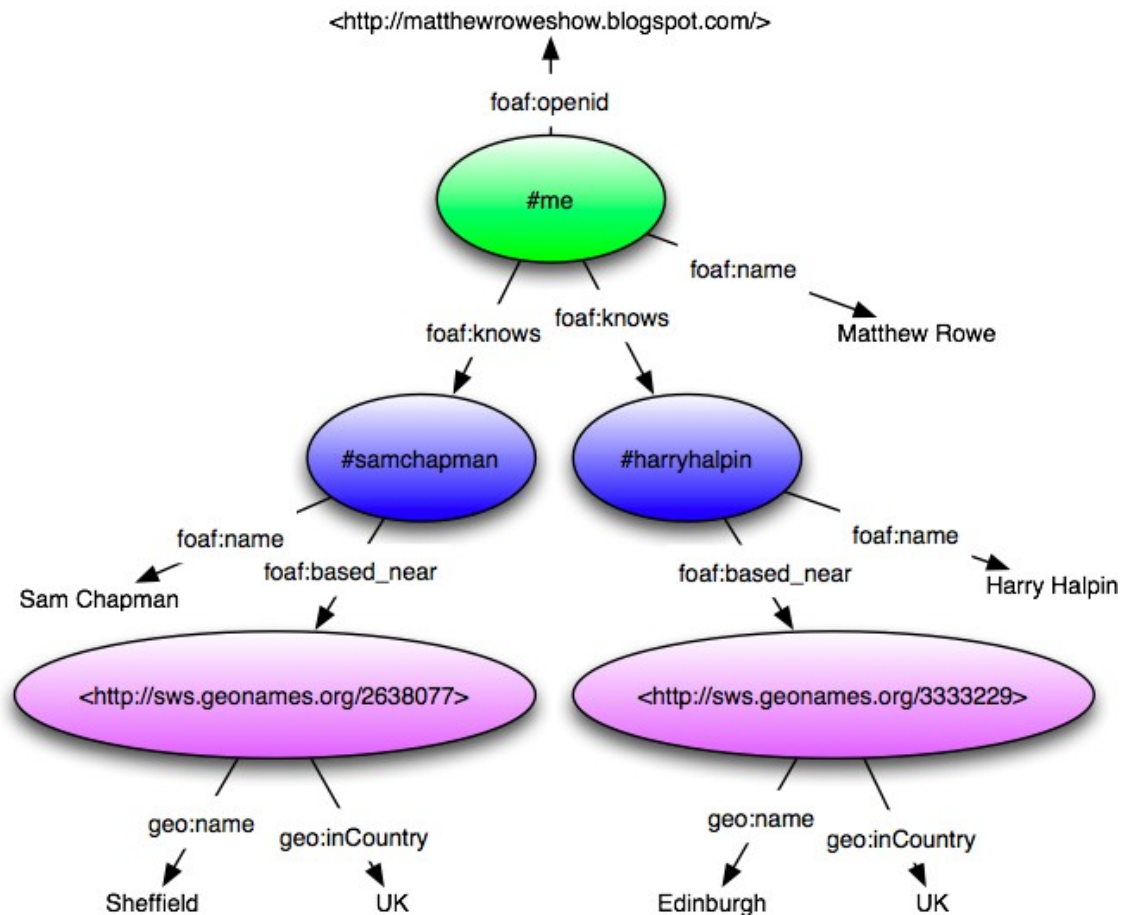
- To export a social graph in a semantic format:
 - Map components of the XML schema to necessary ontology concepts (FOAF, Geonames, etc)
 - Request the user for an OpenID (enabling person resolution and information linkage)
 - Assign URIs to people within the exported social graph
 - Using the user ID / username from the service
 - Assign URIs to location concepts from the Geonames Web Service
 - Query service using city and country

```
<foaf:knows>
  <foaf:Person rdf:about="#617555567">
    <foaf:name>Sam Chapman</foaf:name>
  </foaf:Person>
</ foaf:knows>
```

```
<foaf:knows>
  <foaf:Person rdf:about="#617555567">
    <foaf:name>Sam Chapman</foaf:name>
    <foaf:based_near>
      <geo:Feature rdf:about="http://sws.geonames.org/2638077">
        <geo:name>Sheffield</geo:name>
        <geo:inCountry>United Kingdom</geo:inCountry>
      </geo:Feature>
    </foaf:based_near>
  </foaf:knows>
```



Social Graph Exportation



Social Graph Aggregation

- Identify matching instances of `foaf:Person` in separate graphs and provide links between the instances using `owl:sameAs`
 - Provides a technique to produce linked data given two distributed social graphs
- A decision must be made when to create the link and when not to... Graph Reasoning:
 - Treat individual instances of `foaf:Person` and the accompanying properties as an individual graph
 - Compare graphs (essentially person objects) to derive a similarity measure
 - Should the measure exceed a set threshold, then provide a link between the instances of `foaf:Person`



Graph Reasoning

- When comparing instances of `foaf:Person`, the sole use of the `foaf:name` property to identify a match is insufficient (name ambiguity)
- Additional properties assigned to `foaf:Person` instances must be used to aid the reasoning process:
 - Unique identifiers
 - Inverse functional properties confirm a definite match between instances (e.g. `foaf:mbox`, `foaf:homepage`)
 - Geographical details
 - Compare `geo:Feature` instances from each person
 - Compare URI for a match
 - Compare semantic relation of the locations
 - » e.g. Crookes `dbprop:district` Sheffield
 - » Query a knowledge base to derive a relation (i.e. DBPedia)



Producing Linked Data

- A new RDF graph is created describing the interlinked content
- Information contained within separate social graphs is not duplicated

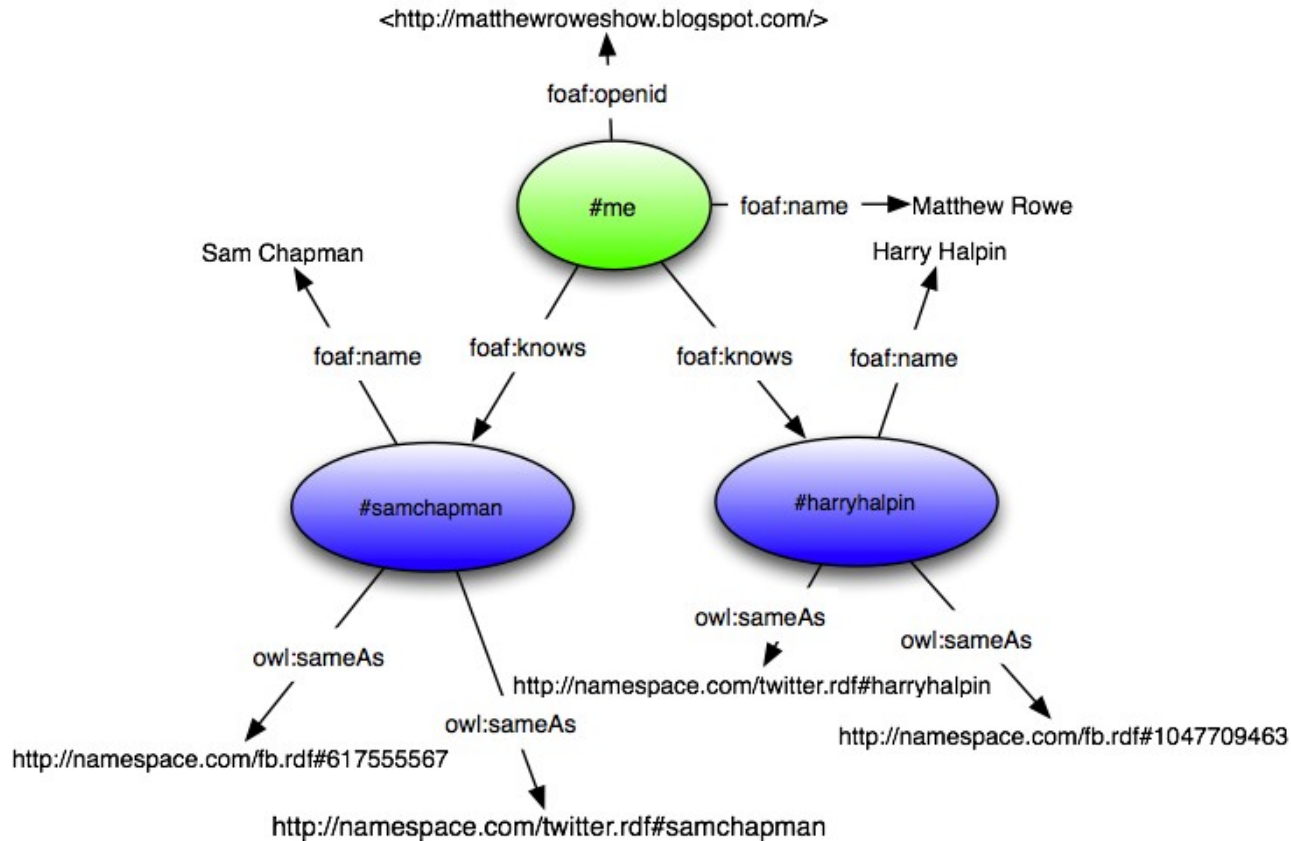
- Instead links are provided to additional information through URIs:

```
<foaf:knows>
  <foaf:Person rdf:about="#samchapman">
    <foaf:name>Sam Chapman</foaf:name>
    <owl:sameAs rdf:about="http://namespace.com/fb.rdf#617555567"/>
    <owl:sameAs rdf:about="http://namespace.com/twitter.rdf#samchapman"/>
  </foaf:Person>
</foaf:knows>
```

- Access to the linked data is now controlled by the hosting service
 - This allows access policies to be set accordingly and only grant access to relevant parties (FOAF+SSL, OAuth)



Producing Linked Data



Experiments

- Evaluate the accuracy of our graph reasoning method to provide links between `foaf:Person` instances
 - Accuracy is measured by minimising type I (false positives) and type II (false negatives) errors when creating links
 - Optimum result would be no type I or type II errors
- Datasets
 - Experiment 1: Social graphs exported from Twitter, MySpace and Facebook for one user
 - Experiment 2: Social graphs exported from Twitter and Facebook for ten separate users
 - The datasets contain overlap where links should be created



Experiments

- Results
 - Experiment 1:

	Fb' : MySp'	GS: Fb' : MySp'	Fb' : Twit'	GS: Fb' : Twit'
True Pos	11	11	5	10
True Neg	389	389	660	662
False Pos	0	0	2	0
False Neg	0	0	5	0

- Experiment 2:

	Fb' :Twit'	GS: Fb' : Twit'
True Pos	42	51
True Neg	2122	2136
False Pos	12	0
False Neg	9	0

Conclusions

- This approach to interlinking distributed social graphs:
 - Exports semantic information from walled garden data silos using existing ontologies
 - Links together instances of foaf:Person referring to the same real world person
 - Provides accurate linkage using low-level bespoke reasoning
 - Maximising correct links and minimising incorrect links
 - Produces a decentralised linked social graph
 - Maintains the access control to additional information of aggregated foaf:Person instances
- Future Work:
 - Releasing the service to allow web users to link their information together
 - Provide additional exportation tools for social web platforms





Questions?