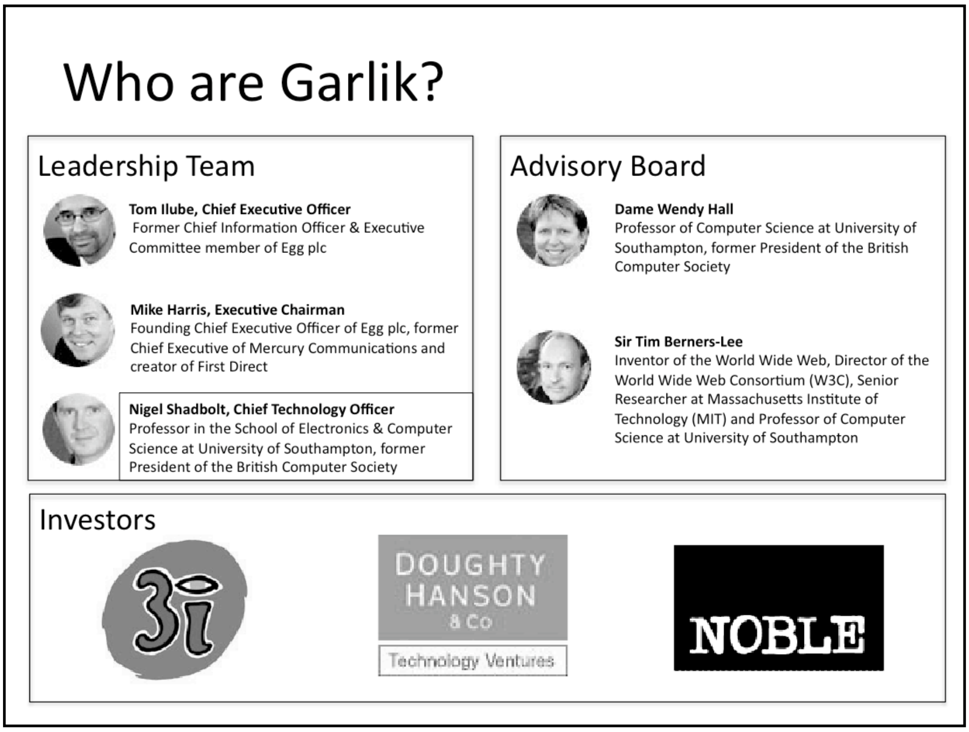
garlik
powerful stuff

Real World RDF Databases

February 2009

Contents

- Insight into Garlik
  - Who are Garlik?
  - Strategy
  - Garlik's services
- Garlik's Technology
  - What is RDF?
  - Technology platform

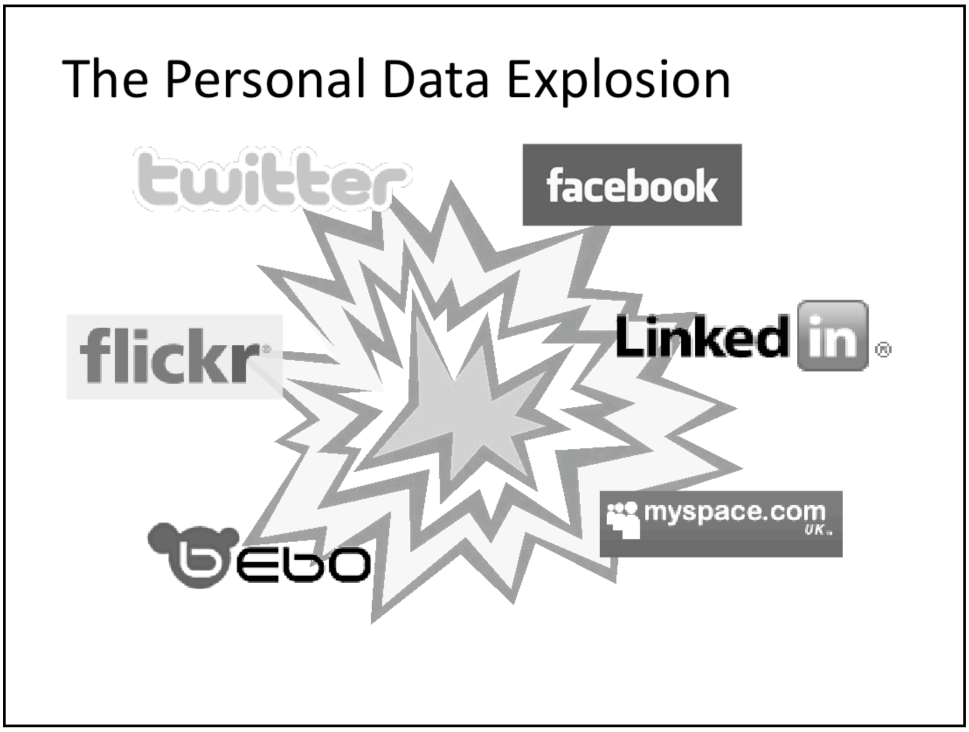


Garlik are the online personal identity experts

Set-up to give individuals and their families real power over the use of their information in the digital world

Garlik have assembled a world class Leadership Team, Advisory Board and partnered with leaders in private equity and venture capital

Over the past three years Garlik has secured over £10m in investment and built up a distribution partner network of blue chip companies



How did we get here?

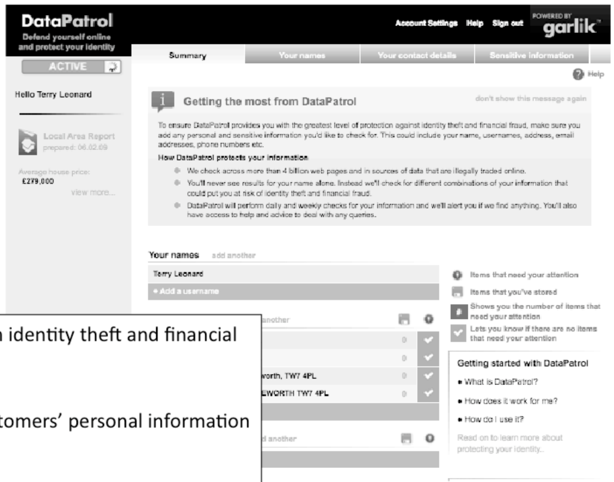
The volume of information about people online is growing

Websites and online services exposing personal information

Consumers beginning to understand the significance of their online presence

This data explosion has implications from a identity theft and a online identity perspective

Garlik's Services – DataPatrol



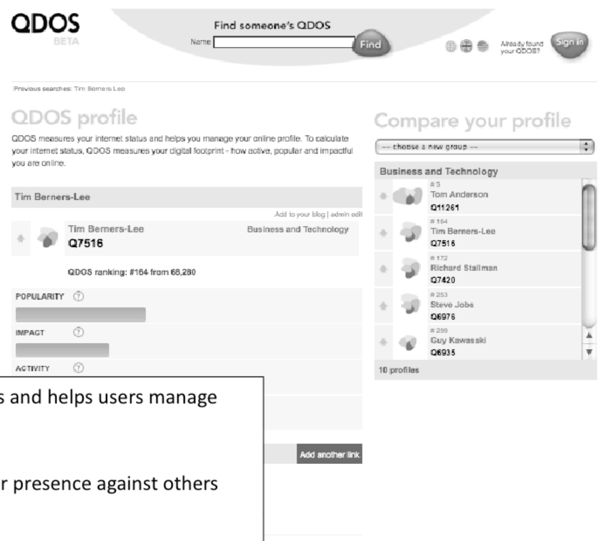
- Protects consumers from identity theft and financial fraud
- Active monitoring of customers' personal information online
- Alerts customers to potential threats and helping them take action

ID theft is growing & predicted to rise to £4bn in UK alone by 2010
3m cybercrimes in 2006 – 1 every 10 secs
DataPatrol undertakes daily and weekly searches for credit card and other compromised financial and sensitive information across

- billions of web pages
- millions of public records and commercial databases
- Compromised financial information from chat rooms

Users are immediately alerted if any of their sensitive details are found

Garlik's Services - QDOS



- Measures internet status and helps users manage their online profile
- Allows users to rank their presence against others
- A bit of fun

More playful than DP
Measures internet status
Rich resource for users
Harvested and categorised Celebrity information to seed the service
Users set-up a QDOS profile to manage and maintain their online presence

Garlik’s Data Challenge

- Handle very large data sets
- Incomplete and irregular data
- Track data sources
- Make use of new data easily

Both QDOS and DataPatrol take large amounts of data and process this on behalf of users to provide them with the service

Large datasets
- Search results from WWW for many customers
- Use of large structured data sources

Incomplete data
-Compromised card data
  - fragments of addresses, credit card numbers
  - not all data types are present

Tracking data source
- Financial reporting and billing
- Presentation of data source to the user to give context

Using new data easily
- Incorporating new and interesting data to enrich user experience

?

There were several options, we went with RDF

Resource Description Framework (RDF)

- URIs

  `http://garlik.com/people#alice`

  protocol domain path fragment

- Literals

- Triples

W3C standard, part of Semantic Web technology stack - Language to write out graphs (as in graph theory, not as in charts)
Revolves around URIs and Triples.
RDF is serialised in text documents (often XML)
Literals are strings, numbers, dates and so on
URI's domain name and path give a namespacing mechanism, making it easy to create globally unique identifiers
This URI also happens to be a URL, so it can be dereferenced

The Triple

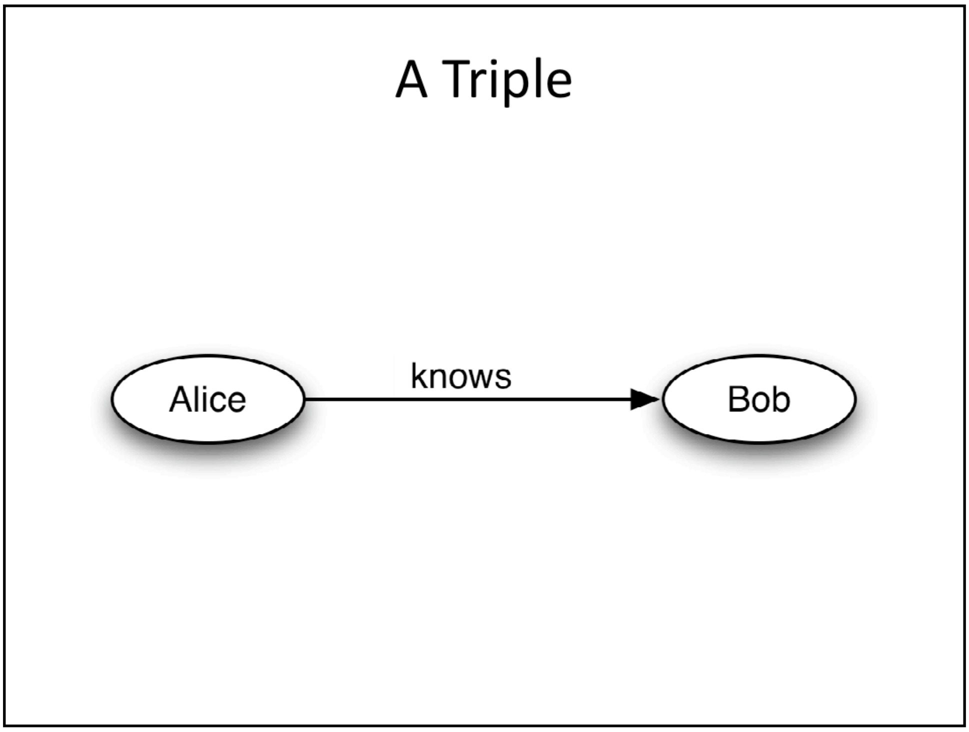
Alice knows Bob

(subject, predicate, object)

A triple is a 3-tuple consisting of two items of interest and a relation between them.

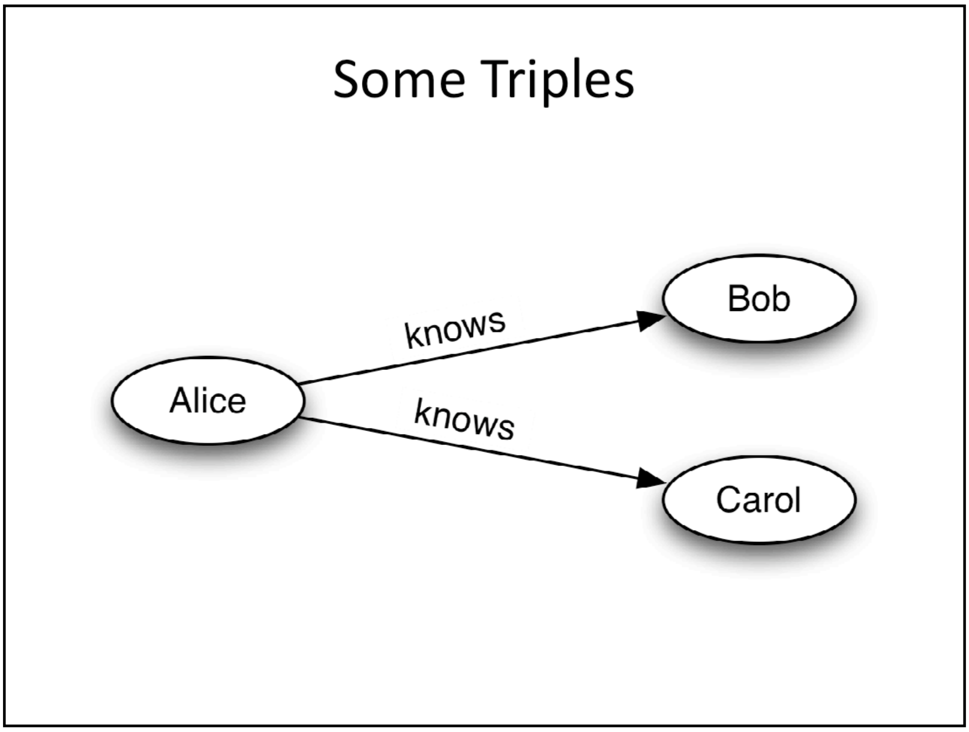
Subject, predicate, object is terminology from linguistics

The relation is non-symmetric (directed)

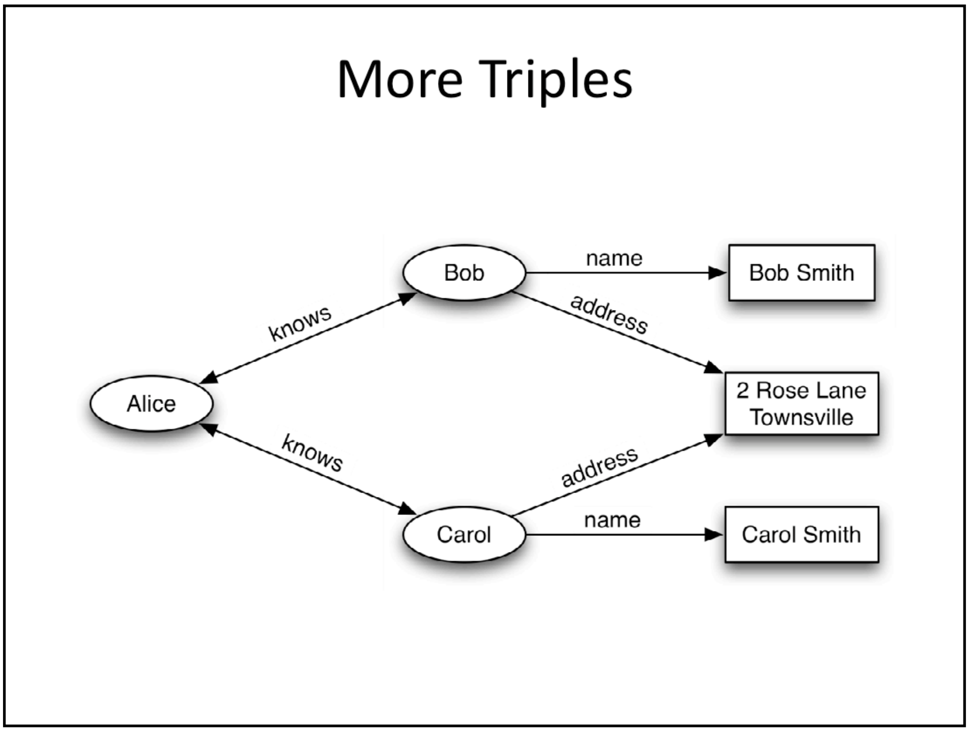


Subjects and Predicates are URIs
Objects can be URIs or Literals
Each triple forms an edge in a Directed Graph

Can use any predicate URI when writing data
No requirement to declare (in RDF itself)

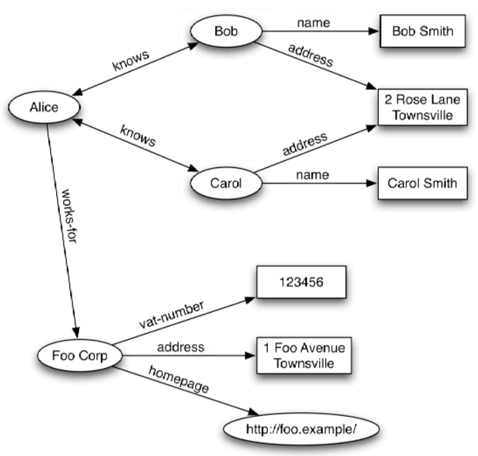


Build graphs using triples



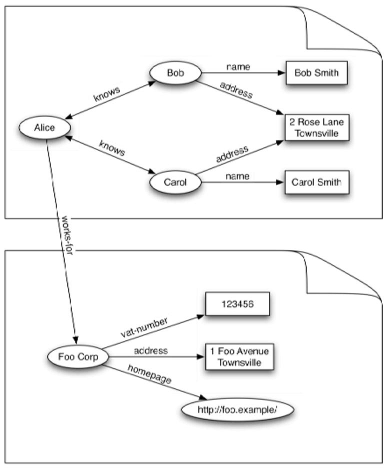
Literals are shown in boxes in the diagram
"2 Rose Lane…" is an example of a shared resource.
The knows relations are bi-directional in this example, done using two triples

Lots of Triples



Shows importance of schema as graph gets large
Graphs grow in any/every direction

Documents of Triples



[Diagram is a bit misleading, each triple is in 1+ documents]
Documents are in text formats, fetched over HTTP
Can often dereference eg. predicate URI to get schema
Context / Provenance can be handled using separate documents of triples – documents have URIs too

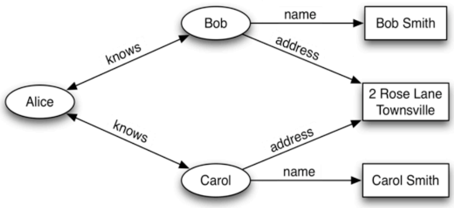
Relational Equivalence

people

pk	name	address
1	Ø	Ø
2	Bob Smith	2 Rose Lane Townsville
3	Carol Smith	2 Rose Lane Townsville

knows

a	b
1	2
2	1
1	3
3	1



Not direct equivalence, just highlights differences, there are better RDB schemas for this data
Triples express same information as rows
Shows “ragged” data in RDB – begin to see how RDB gets complex in this situation – with enough normalisation queries become long-winded to write

RDF Storage

Same challenges as for any system

Scale – 1GT expected, ended up with 10GT
Performance – hard to quantify, but needed to support interactive end users
Stability – support customer facing financial app
Features – transactions, online backup, SPARQL queries

Available Options

- DBMS mapping
- Off the Shelf solutions
- Build our own

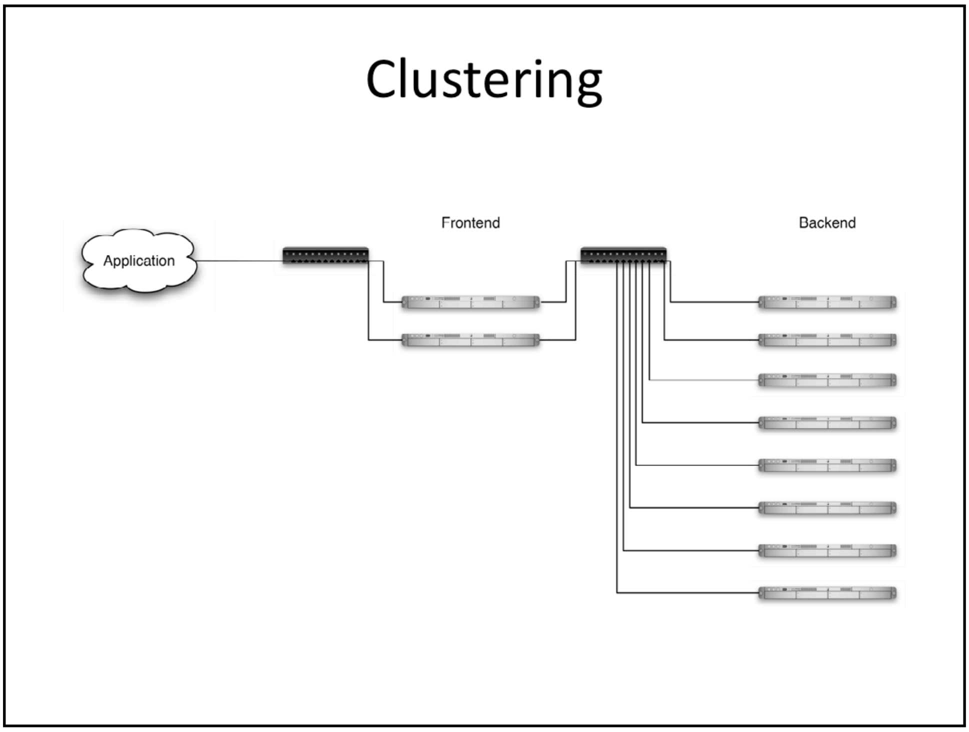
DBMS – inflexible, has overhead
COTS – not up to task at that time (three years ago) - now we have Jena, Oracle 11g, Top Quadrant, Virtuoso
BYO – only real option at the time

Different now, commercial/free offerings might work well enough

Garlik's Approach

Built new platform – slightly different problems to existing database technology

Clusters – starting from scratch with design, may as well
exploit availability of commodity gigabit ethernet



Shared nothing cluster
Frontend – routing, load balancing
Backend – backend tasks distributed across nodes, cloud-like, replication
Custom Protocol for backend – MDNS, TCP sockets
Application talks to DB over HTTP (SPARQL and PUT)

Design Approach

- Administrative
- Structure indexing
- Resource indexing
- Query processing
- Data processing

Admin – starting/stopping services, discovery, backups, routing, migrating data and processes
Structure – stores the underlying shape of data: triples, which triple's in what document
Resource – stores the actual values (URIs, Literals)
Query – orchestrates queries, collates results, handles query algebra
Data – parses RDF data into structure and resources

System written in C99 and C++

Achieving Performance

- Minimise indexes
- Data distribution
- Novel indexing algorithms

Indexes – down to 60-120 bytes / triple (depending on complexity of data)
Distribution – analyse sample data to pick good initial distribution across cluster
Indexing – classical relational DB indexes not particularly appropriate

Achieving Scalability

- Clustering
- Query algorithms
- Indexing

Clustering – spread data over machines
Query – designed to expect very large, incomplete data sets
Indexing – ways to consider random slices of relevant index, make use of multi cores

Efficiency - 250MT/node, 8 machines for 2BN triples

Conclusions

To recap the challenges that Garlik faced were: Scale, Incomplete and Irregular data and Provenance.

Advantages – flexible, in face of changing data.
expressive enough to model complex/ragged/irregular data
Possible – and much easier now COTS solutions exist
Practical – scales out as big as you need, performance is good

Because of that we feel that it's given us a clear advantage over going with traditional DBMS

Questions

Christian Davis
christian.davis@garlik .com

Steve Harris
steve.harris@garlik.com