A Little Graph Theory for the Busy Developer

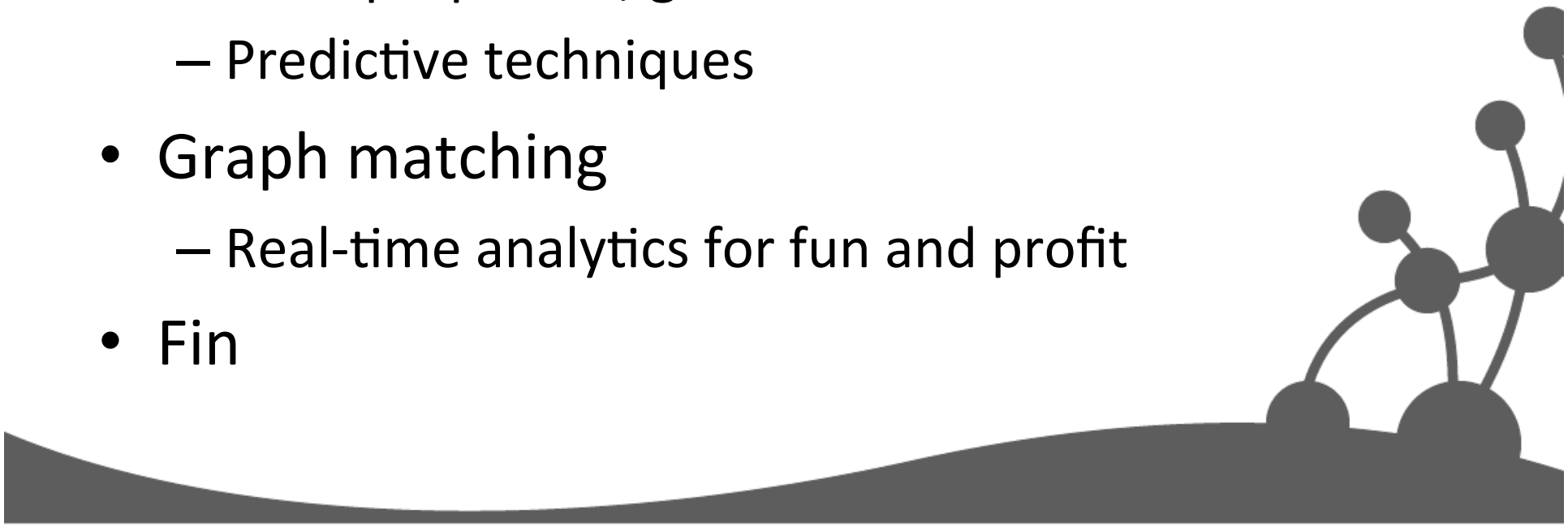
Jim Webber

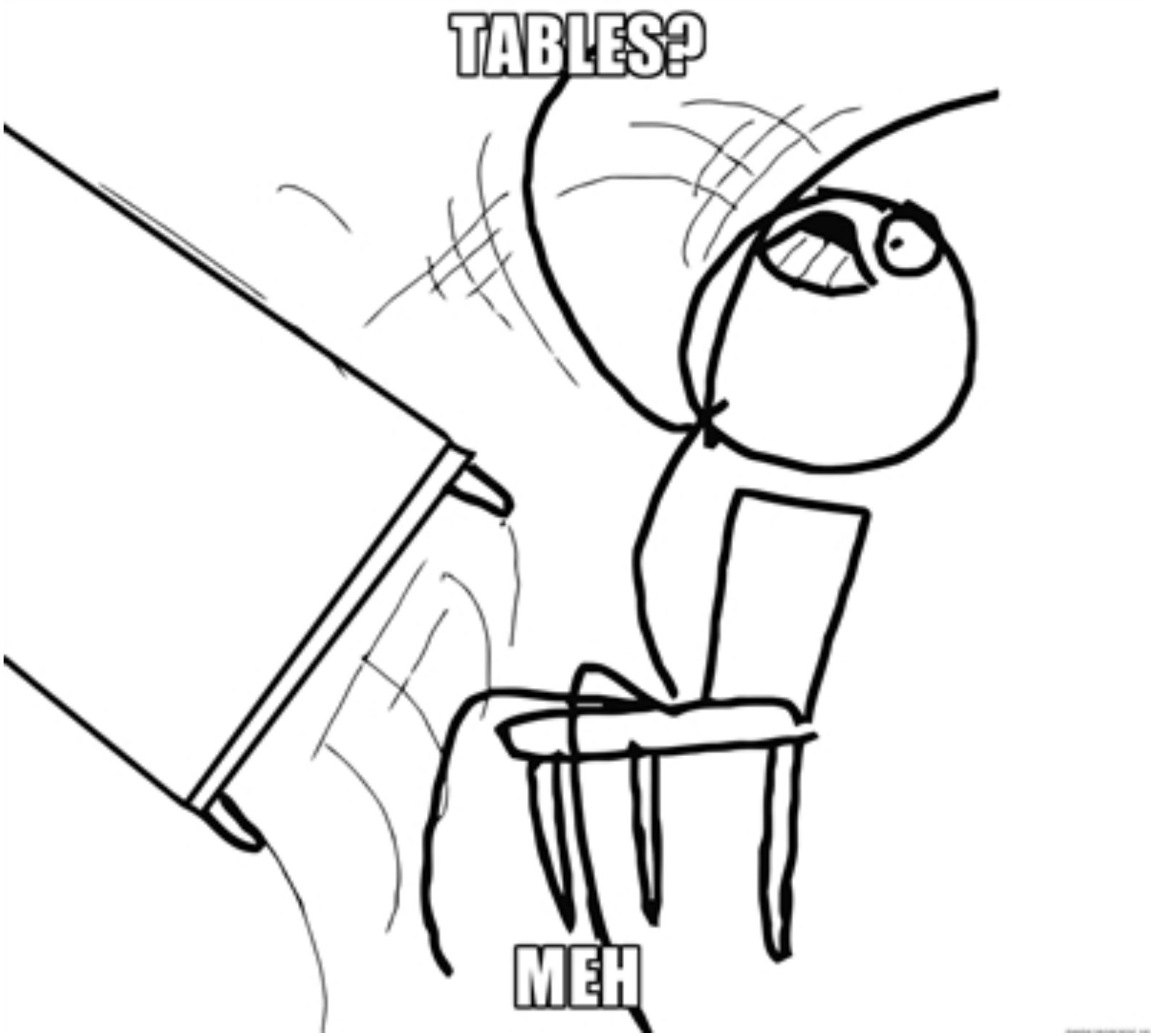
Chief Scientist, Neo Technology

@jimwebber

Roadmap

- Imprisoned data
- Graph models
- Graph theory
  - Local properties, global behaviors
  - Predictive techniques
- Graph matching
  - Real-time analytics for fun and profit
- Fin




TABLES?
MEH

http://www.flickr.com/photos/crazyneighborlady/355232758/

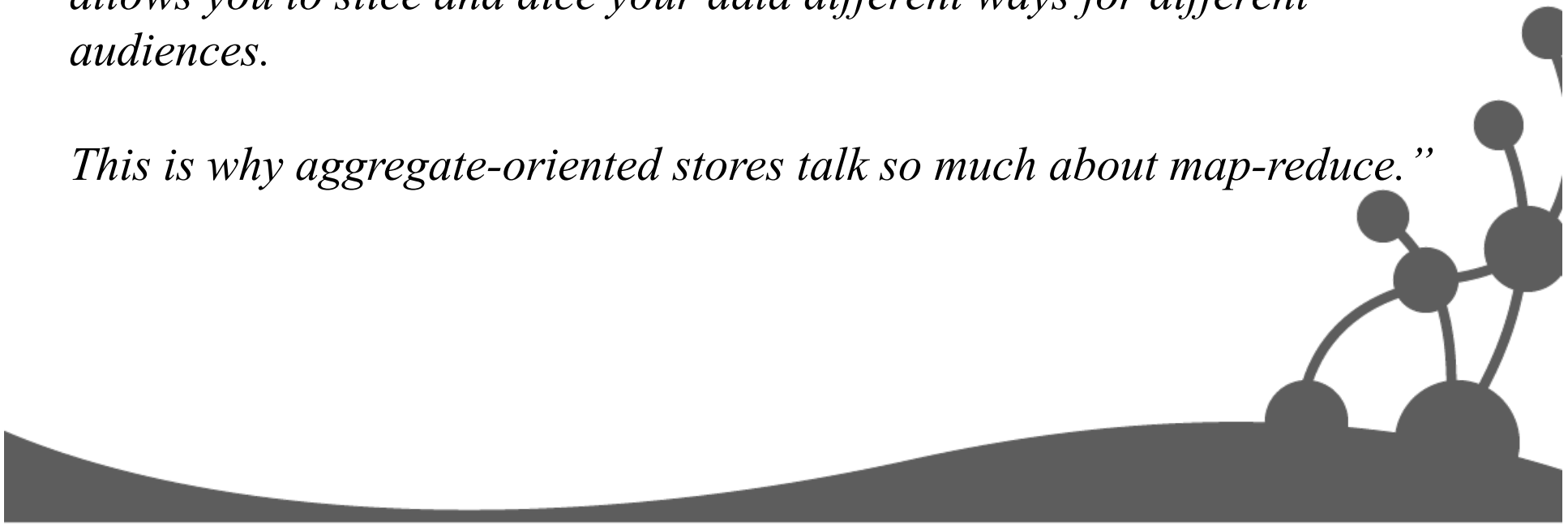

http://gallery.nen.gov.uk/image82582-.html

Aggregate-Oriented Data

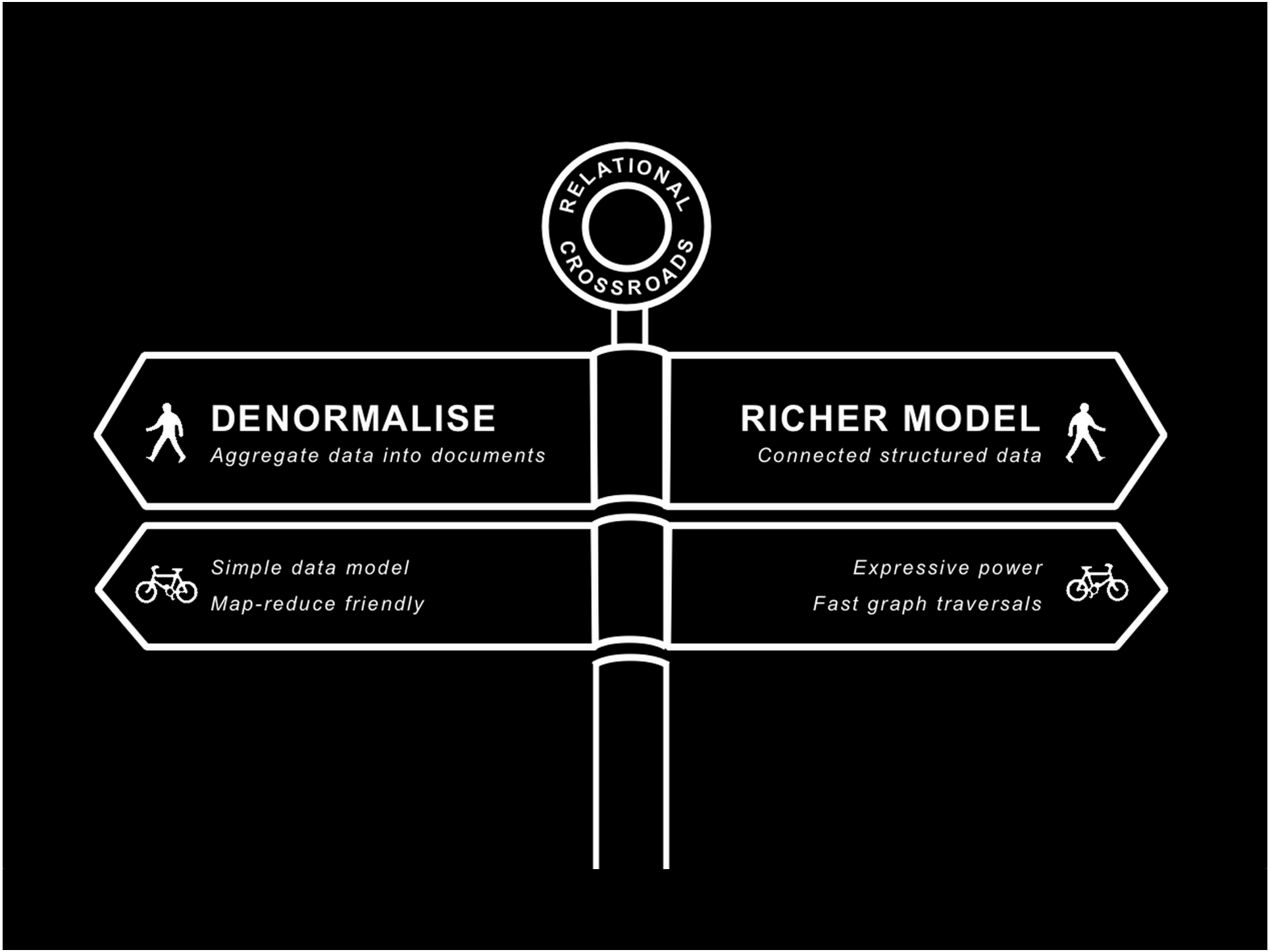
http://martinfowler.com/bliki/AggregateOrientedDatabase.html

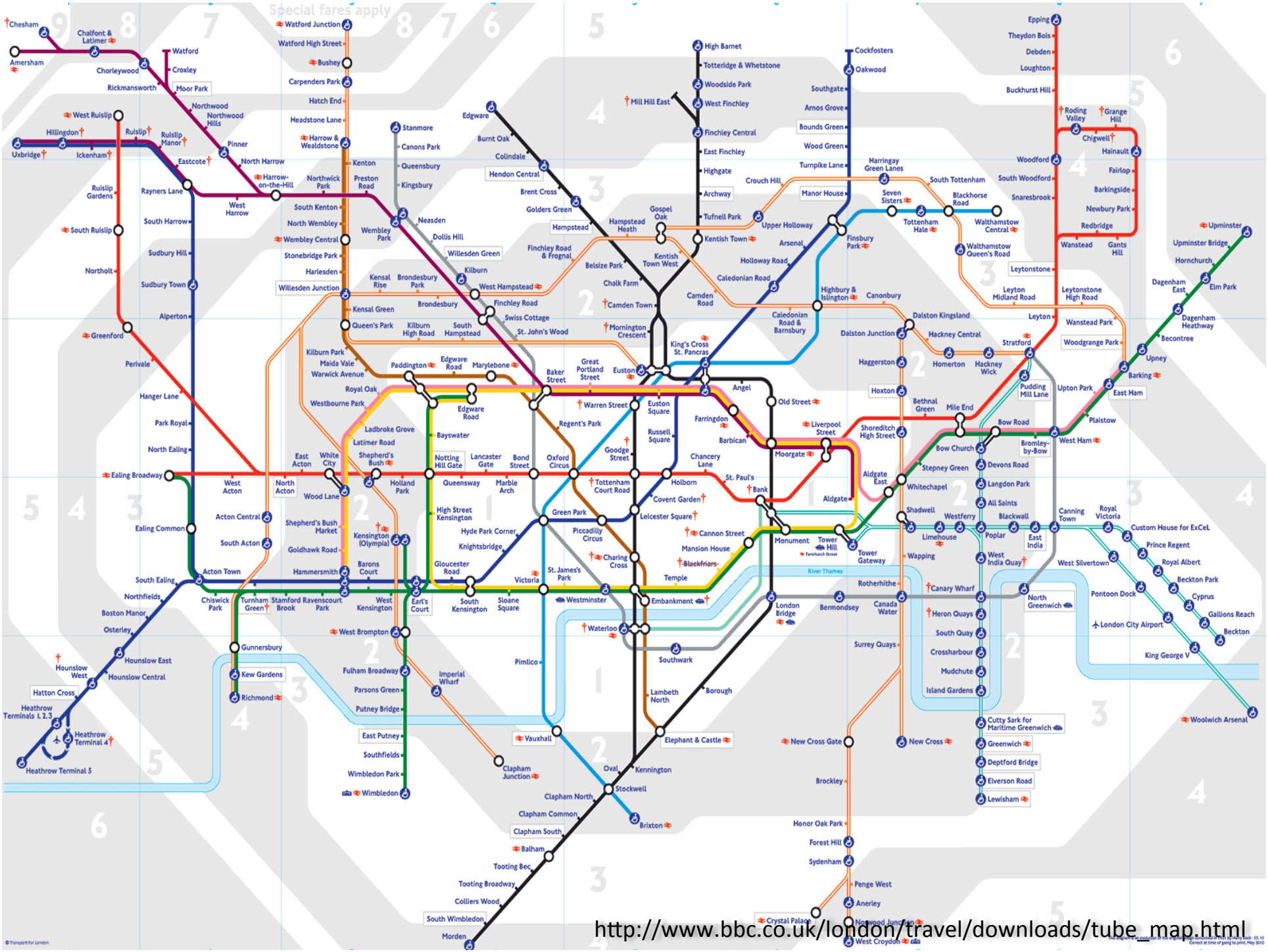
"There is a significant downside - the whole approach works really well when data access is aligned with the aggregates, but what if you want to look at the data in a different way? Order entry naturally stores orders as aggregates, but analyzing product sales cuts across the aggregate structure. The advantage of not using an aggregate structure in the database is that it allows you to slice and dice your data different ways for different audiences.

This is why aggregate-oriented stores talk so much about map-reduce."



$$complexity = f(size, connectedness, uniformity)$$

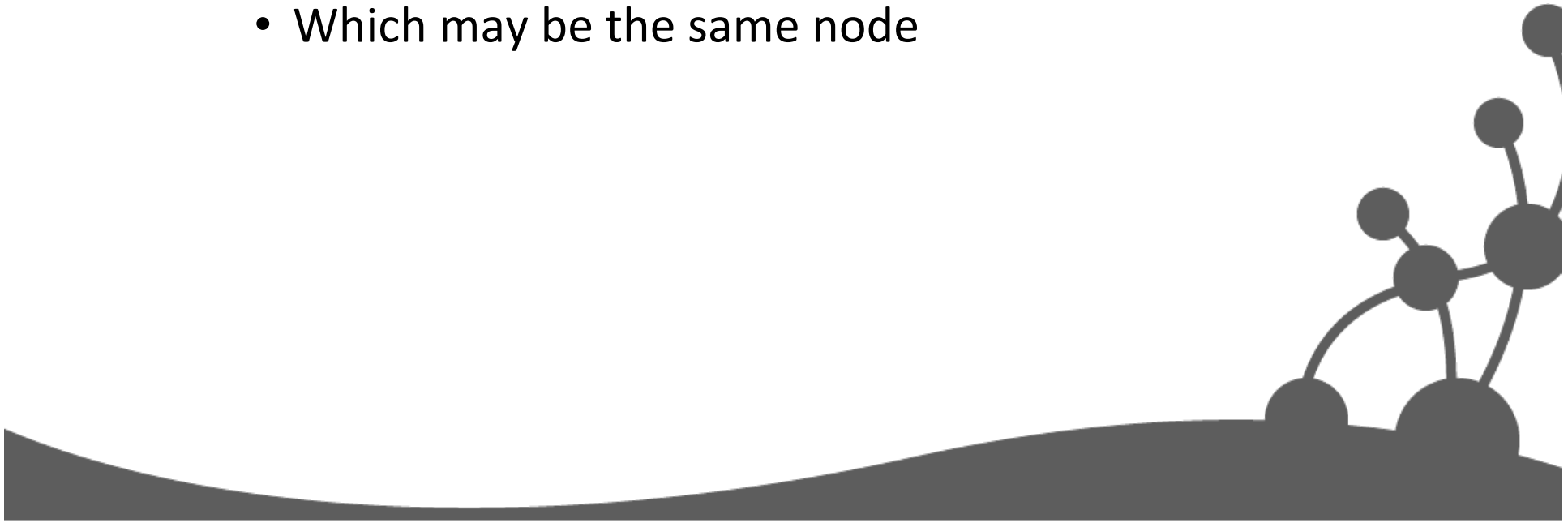

RELATIONAL CROSSROADS
DENORMALISE
Aggregate data into documents
RICHER MODEL
Connected structured data
Simple data model
Map-reduce friendly
Expressive power
Fast graph traversals

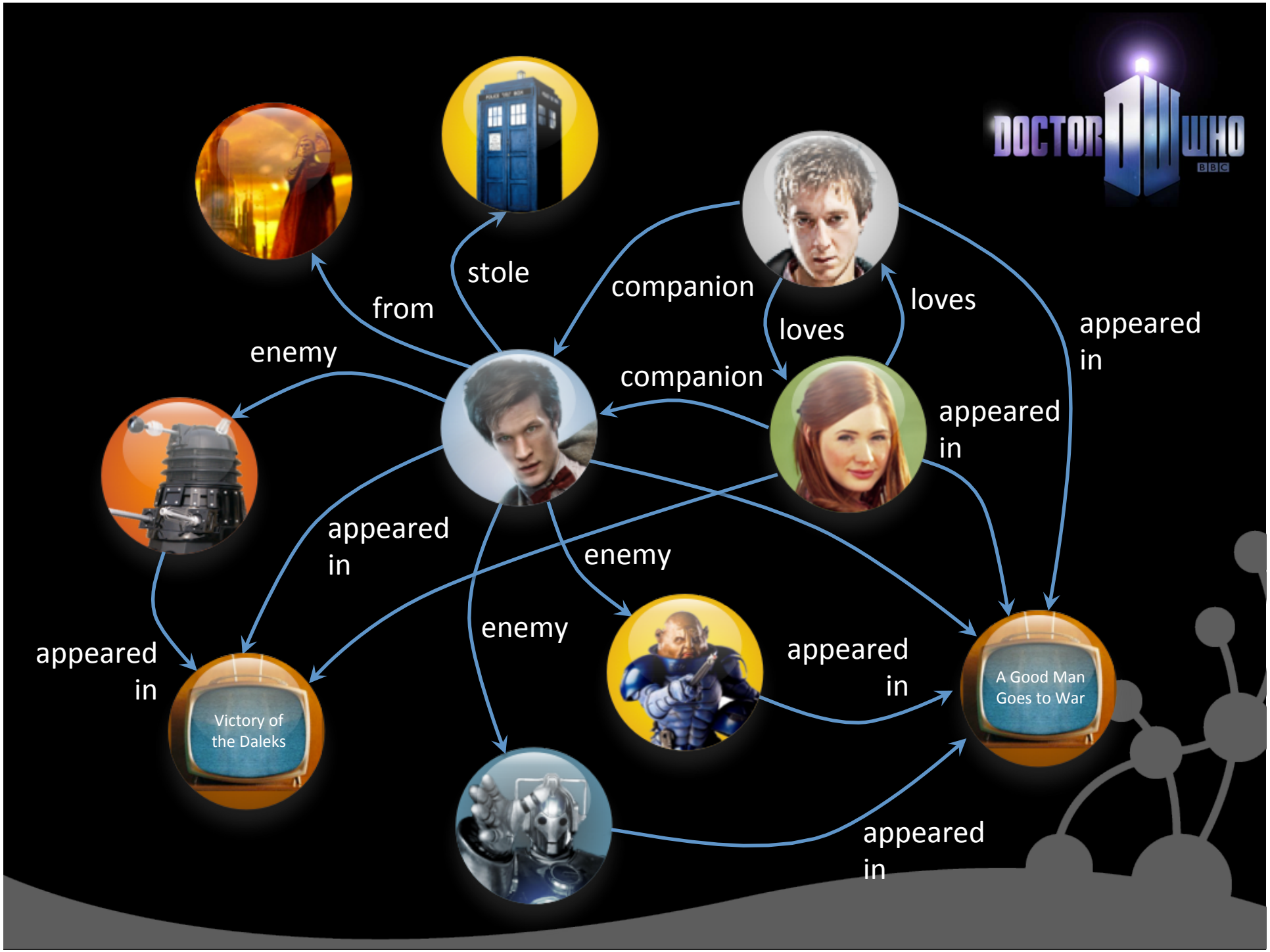


http://www.bbc.co.uk/london/travel/downloads/tube_map.html

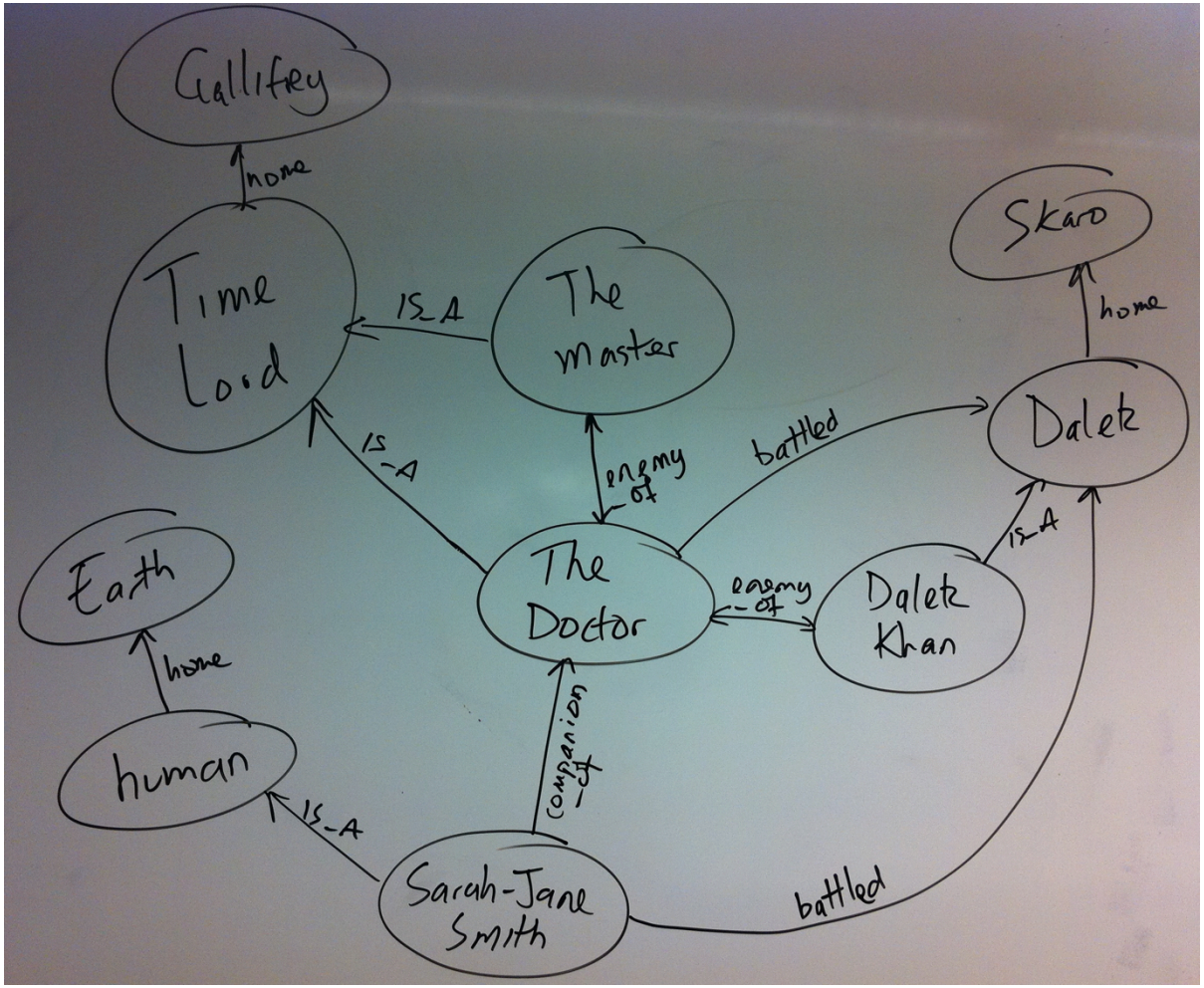
Property graphs

- Property graph model:
  - Nodes with properties
  - Named, directed relationships with properties
  - Relationships have exactly one start and end node
    - Which may be the same node




DOCTOR WHO
BBC
stole
from
companion
loves
loves
companion
enemy
appeared in
appeared in
appeared in
enemy
enemy
appeared in
appeared in
appeared in
Victory of the Daleks
A Good Man Goes to War

Property graphs are *very* whiteboard-friendly



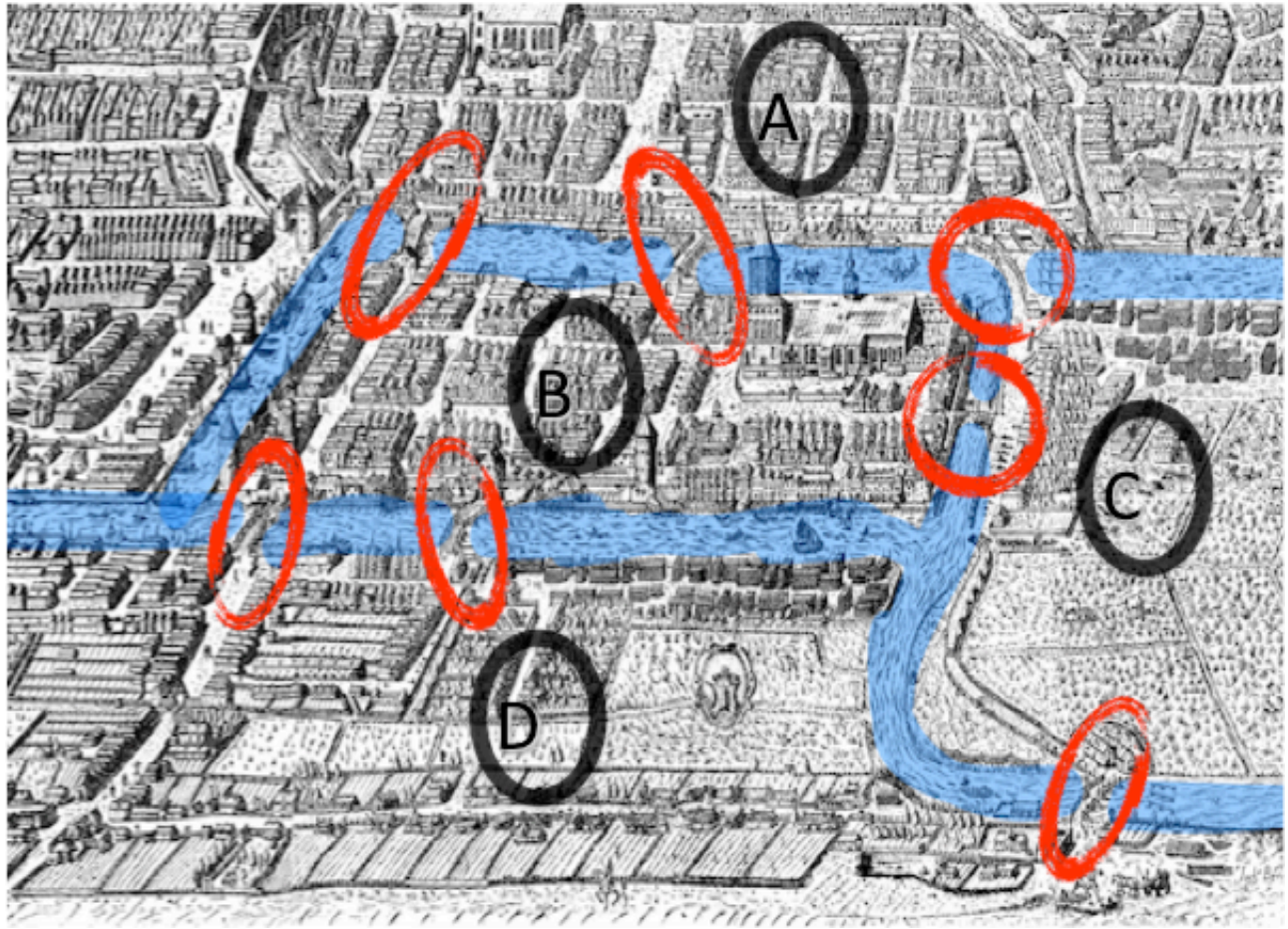
http://blogs.adobe.com/digitalmarketing/analytics/predictive-analytics/predictive-analytics-and-the-digital-marketer/

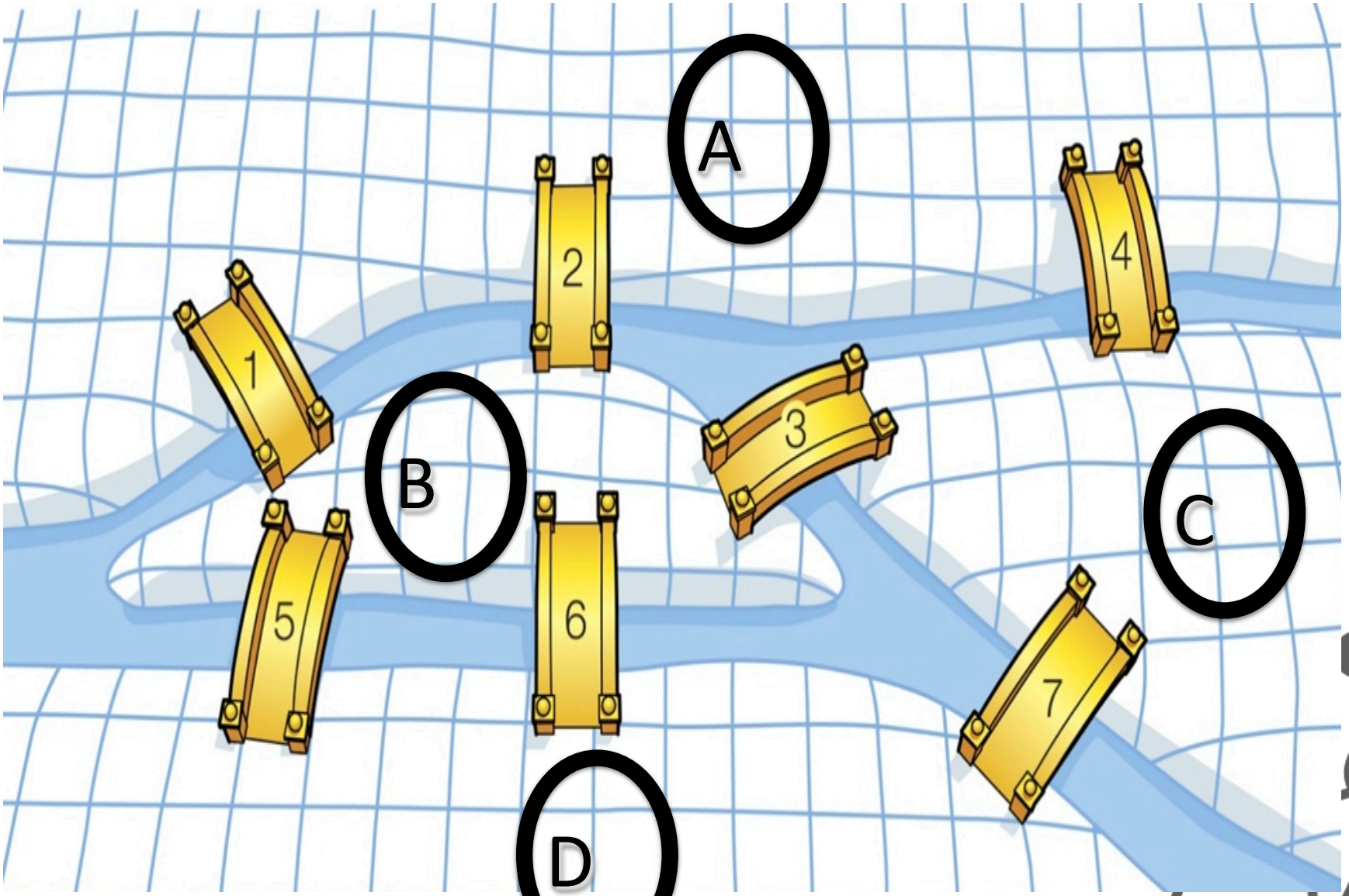
Meet Leonhard Euler

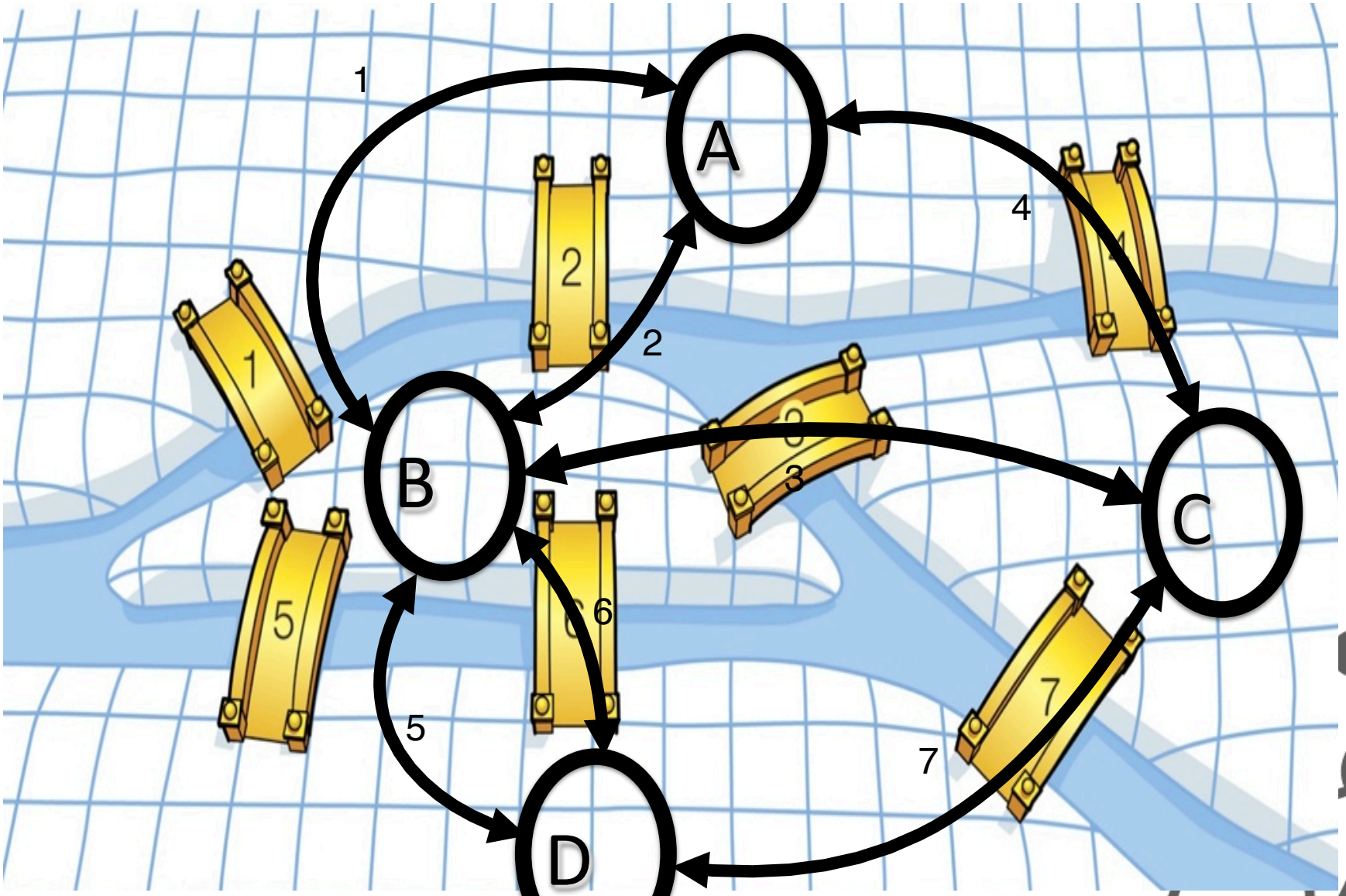
- Swiss mathematician
- Inventor of Graph Theory (1736)

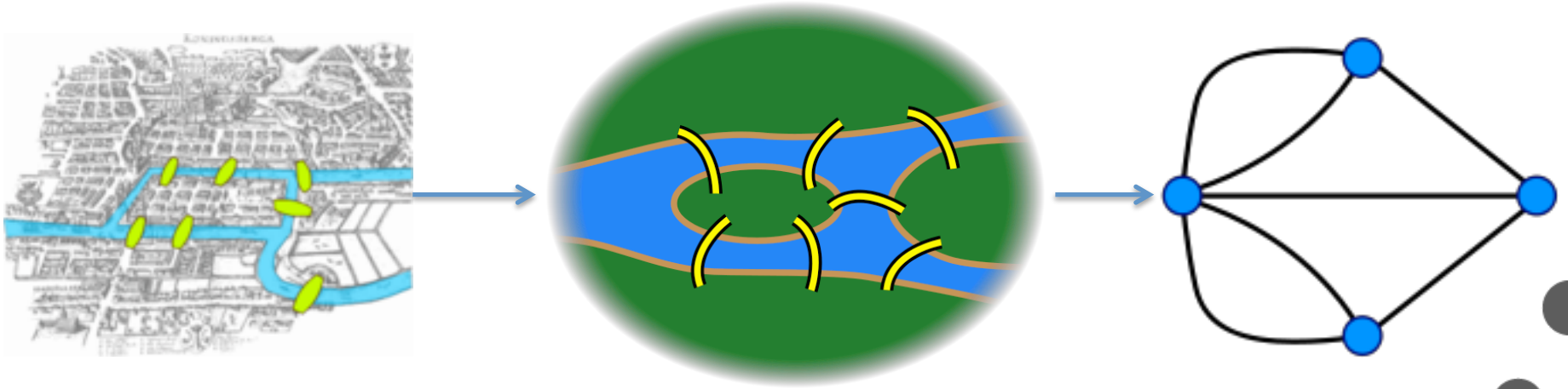
http://en.wikipedia.org/wiki/File:Leonhard_Euler_2.jpg

Königsberg (Prussia) - 1736




A
B
C
D
1
2
3
4
5
6
7

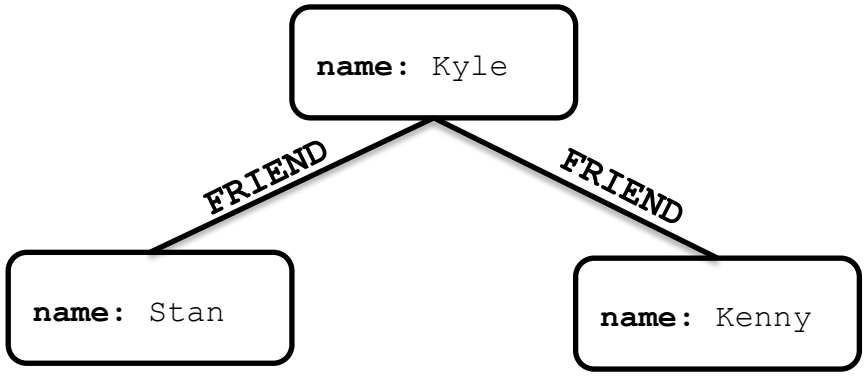

A
B
C
D
1
2
3
4
5
6
7



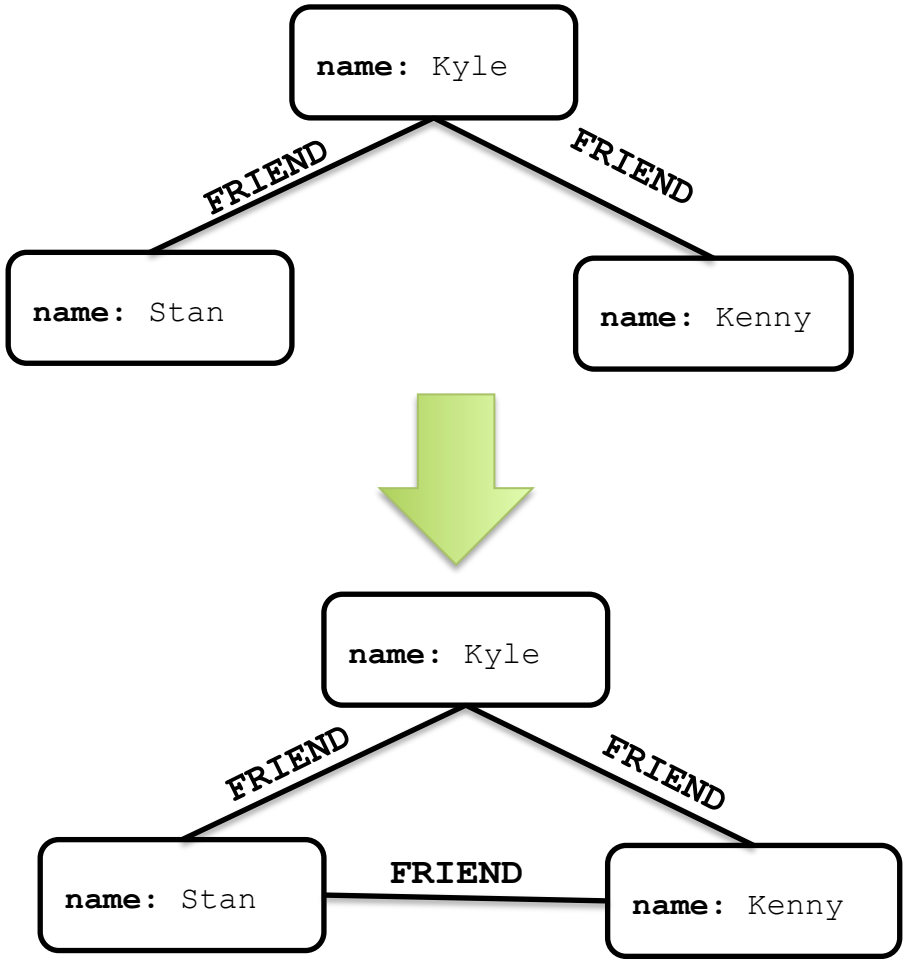
http://en.wikipedia.org/wiki/Seven_Bridges_of_Königsberg


DULWICH SUPERMARKET & OFF LICENCE
ENGLISH , TURKISH , GREEK , MEDITERRANEAN FOOD
18
DELICATESSEN & ORGANICS
TEL : 020 8299 2214
FRESHLY CUT SANDWICHES - BILTONG
BILTONG
BOEREWORS

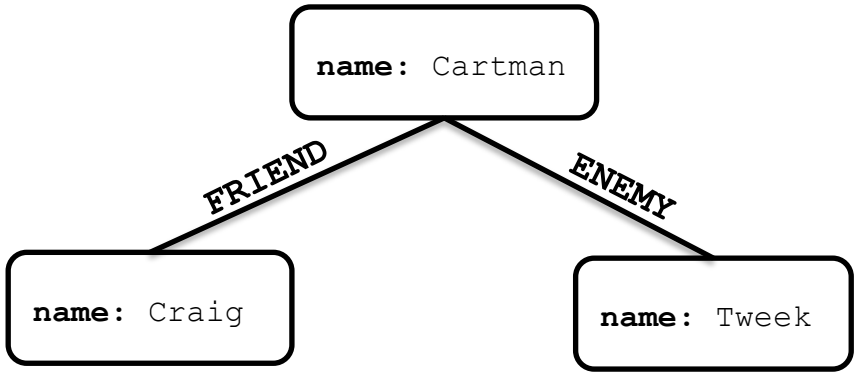
Triadic Closure



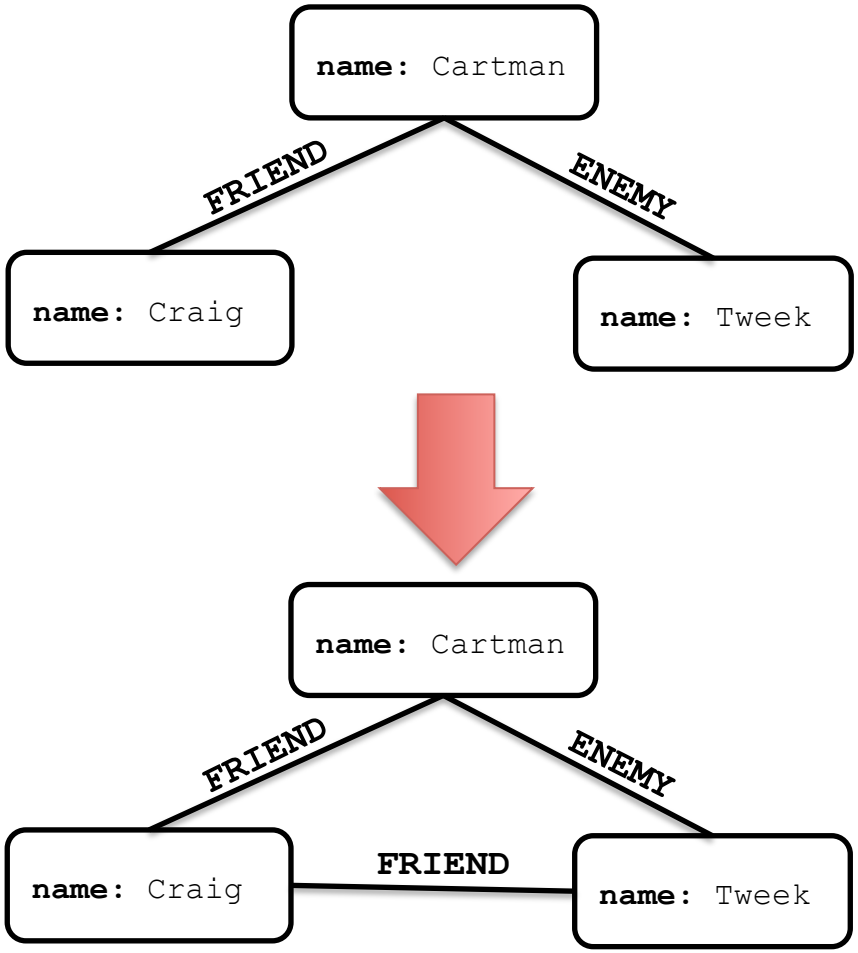
Triadic Closure



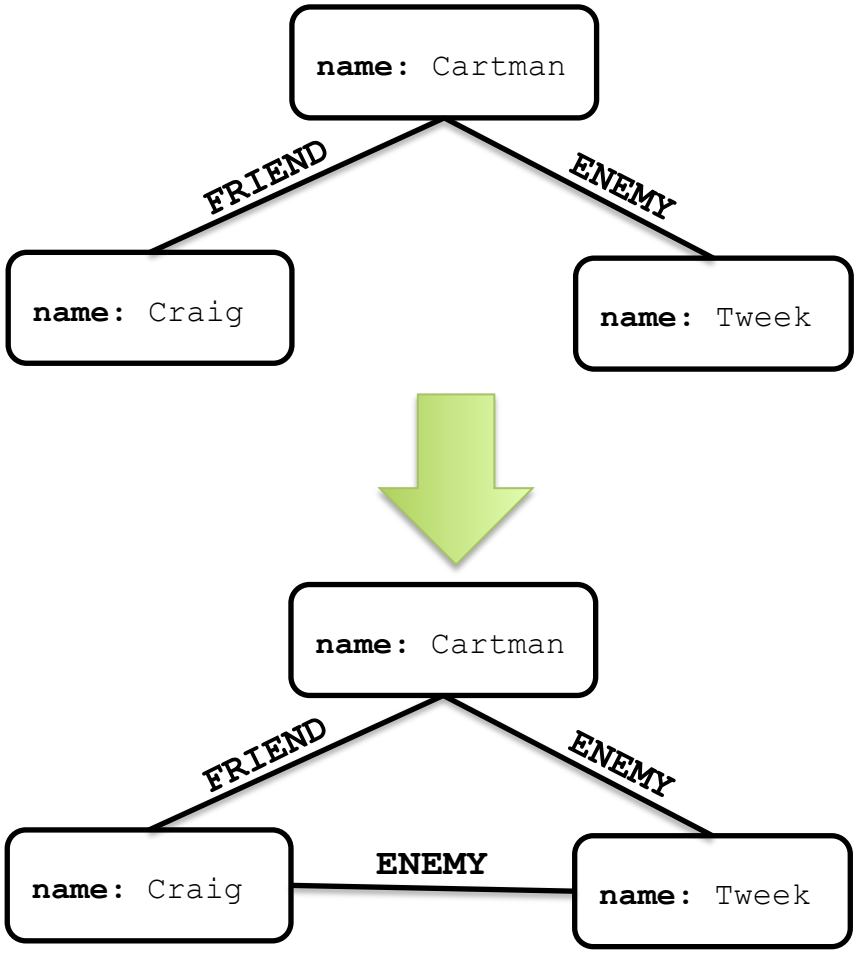
Structural Balance



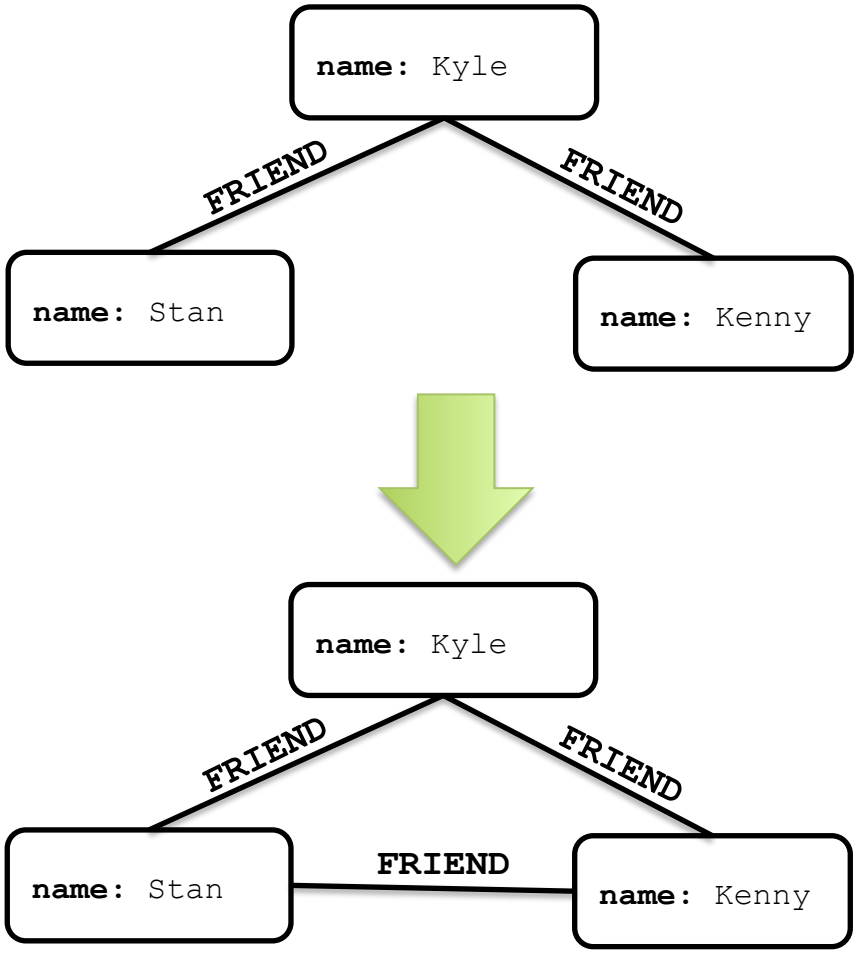
Structural Balance



Structural Balance

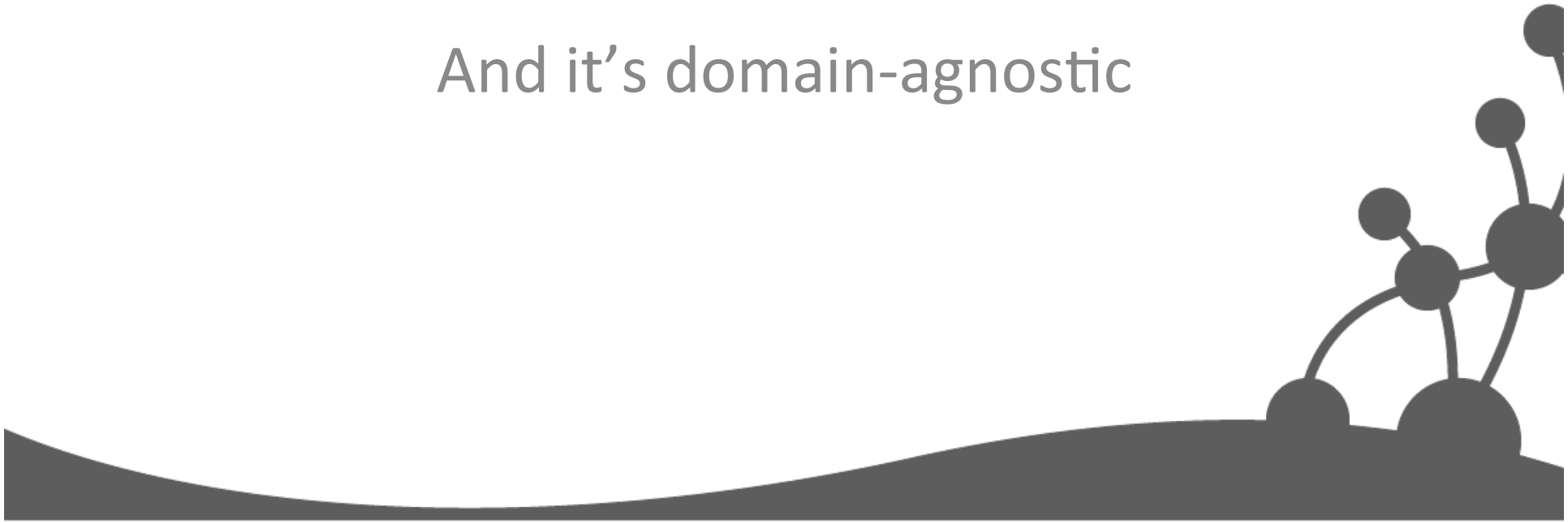


Structural Balance

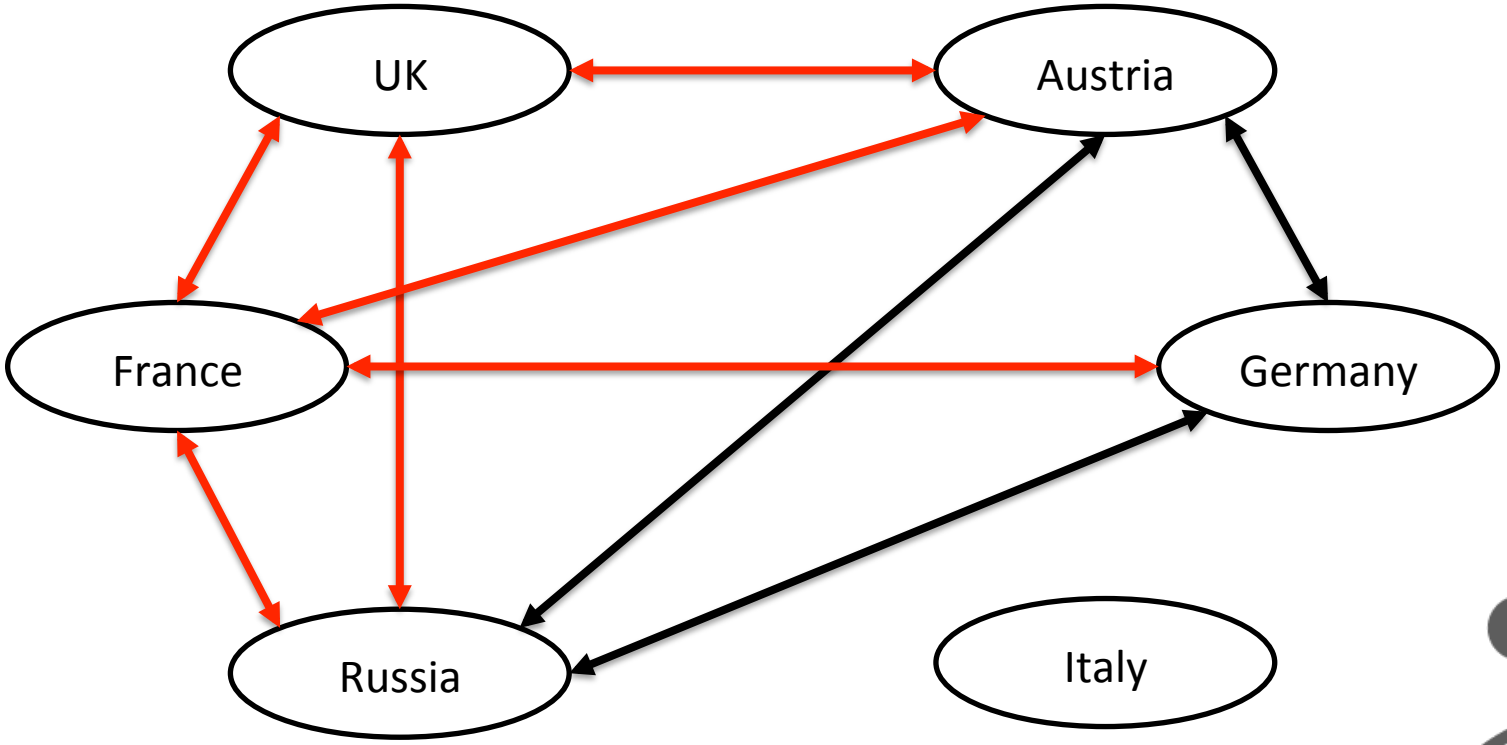


Structural Balance is a *key* predictive technique

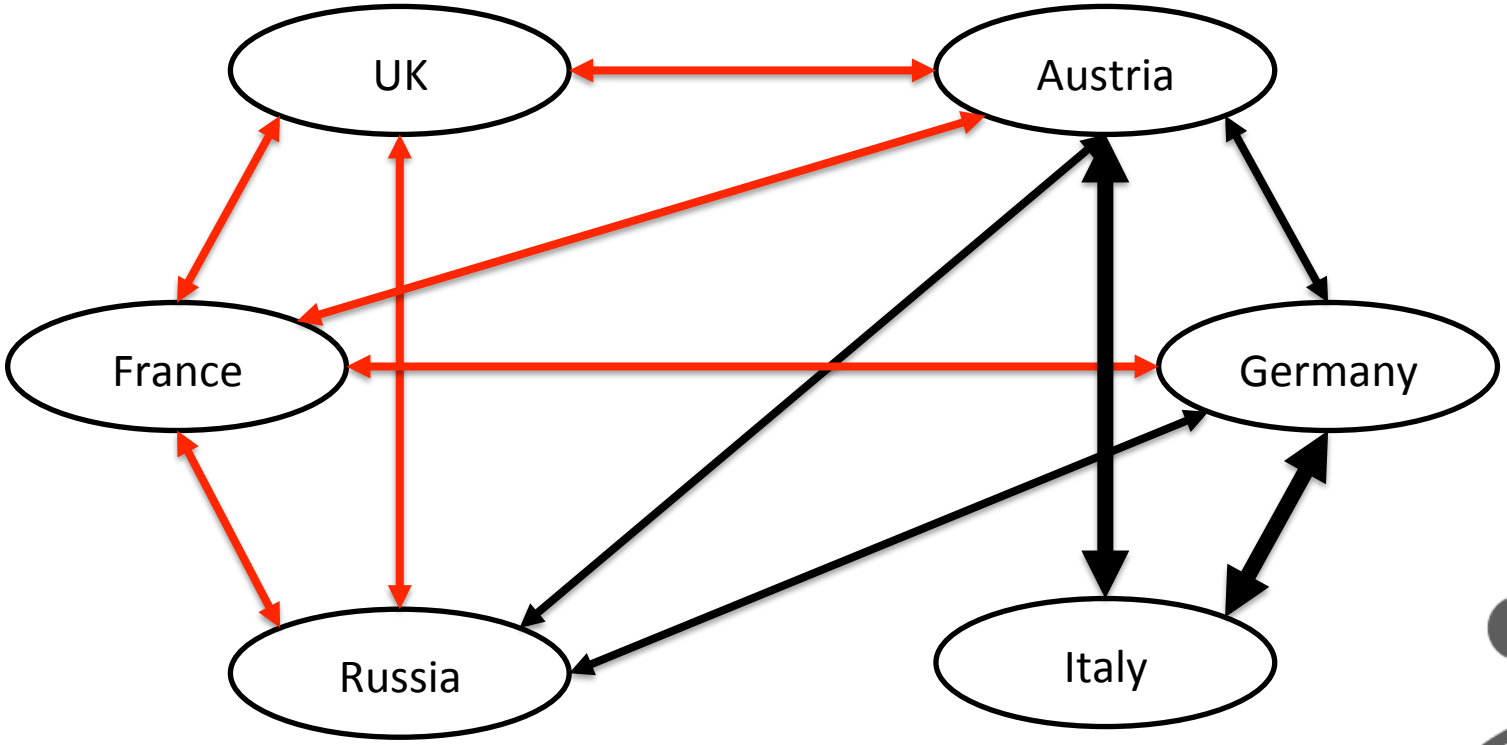
And it's domain-agnostic



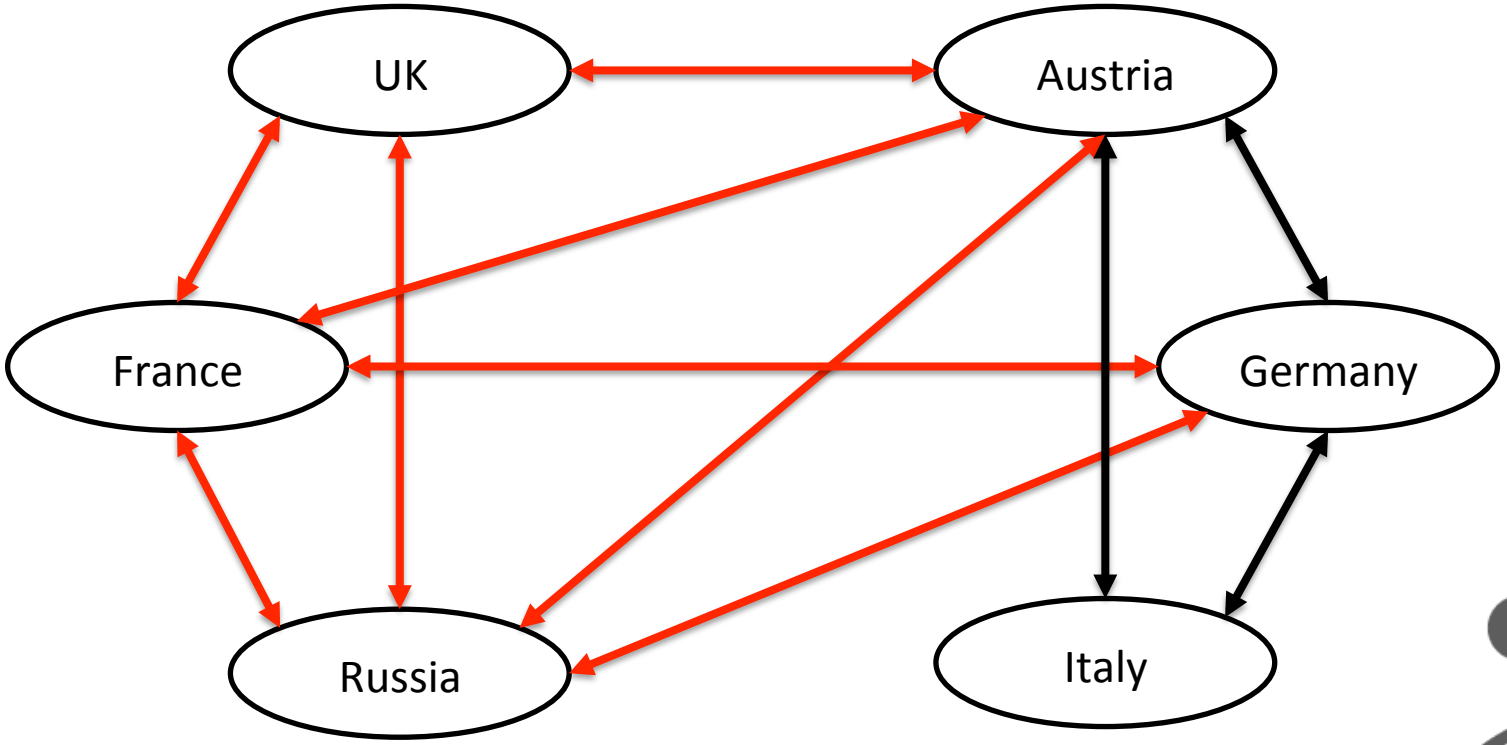
Allies and Enemies



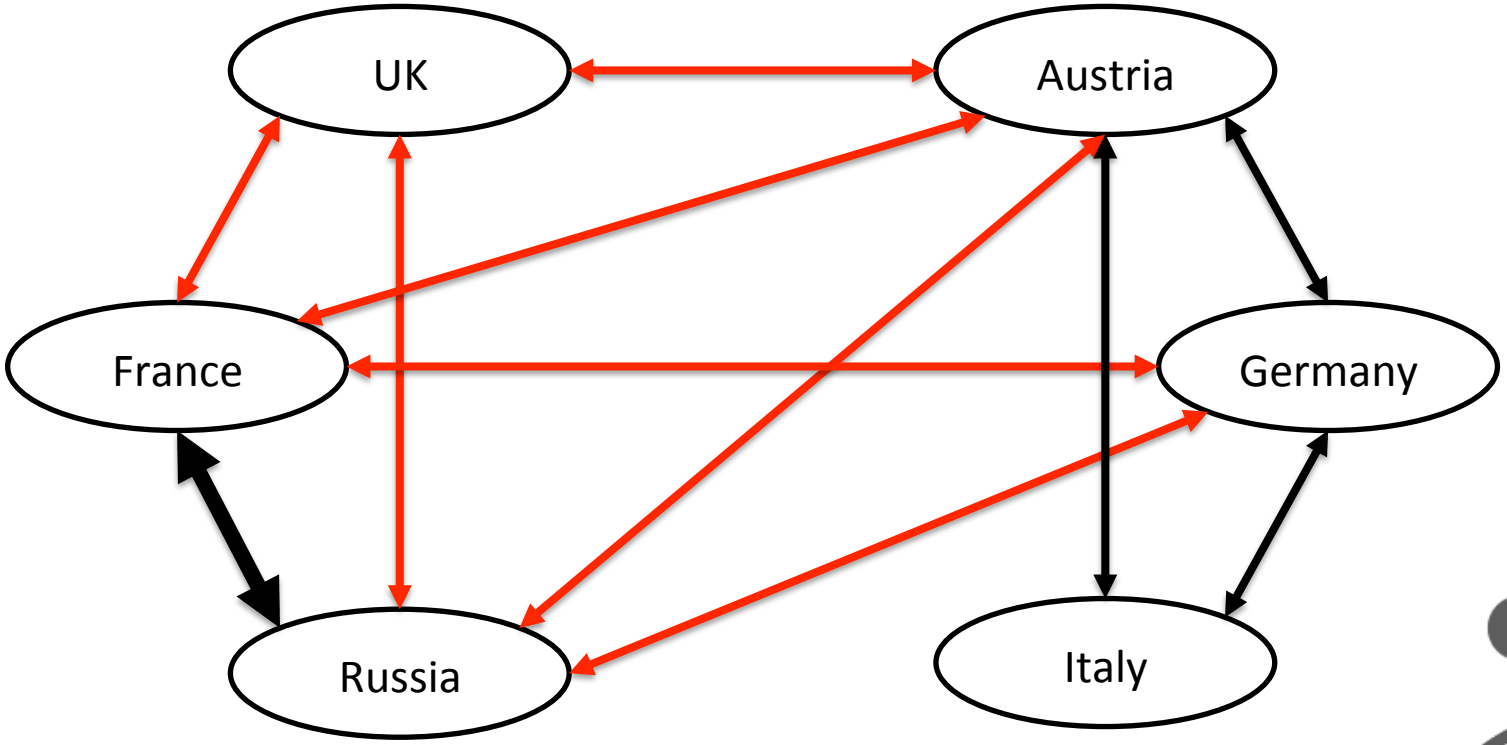
Allies and Enemies



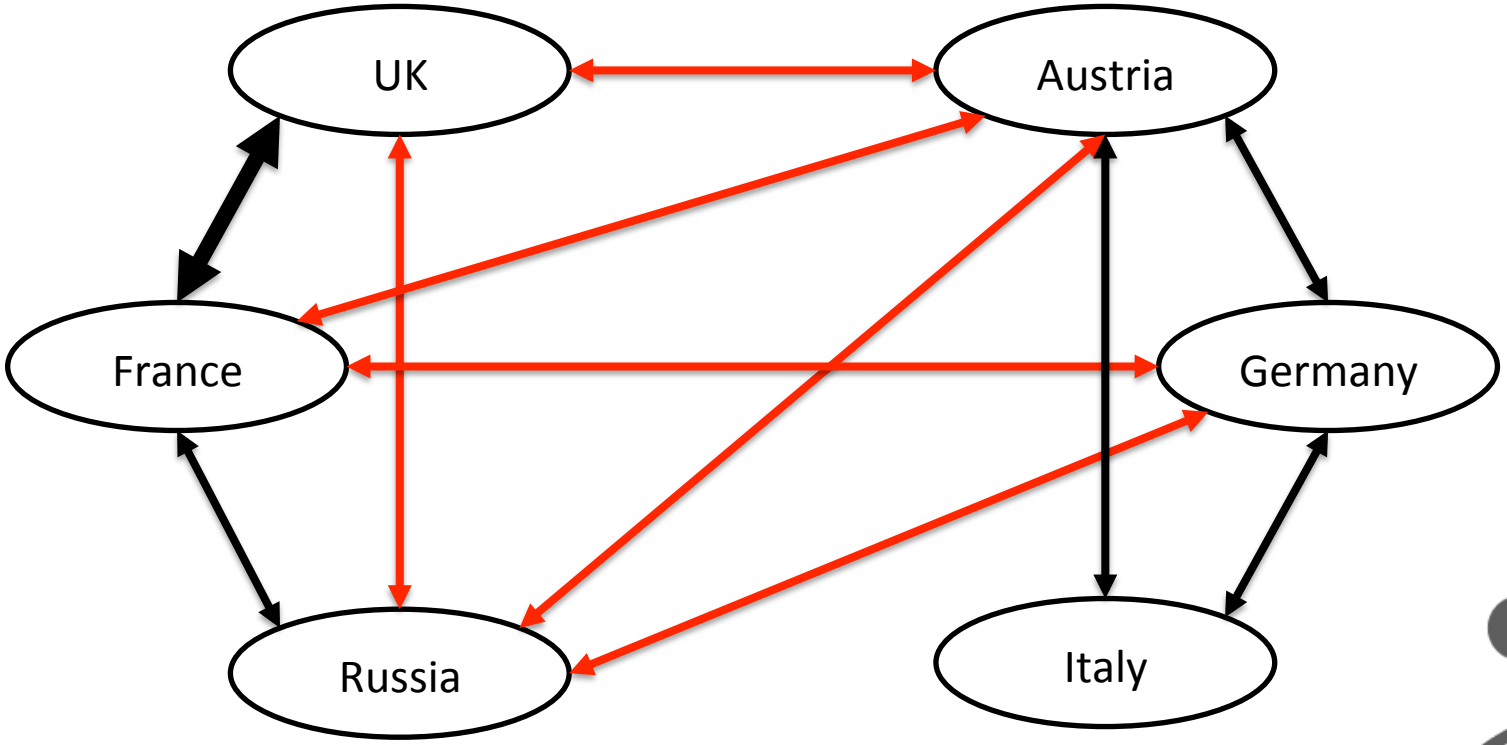
Allies and Enemies



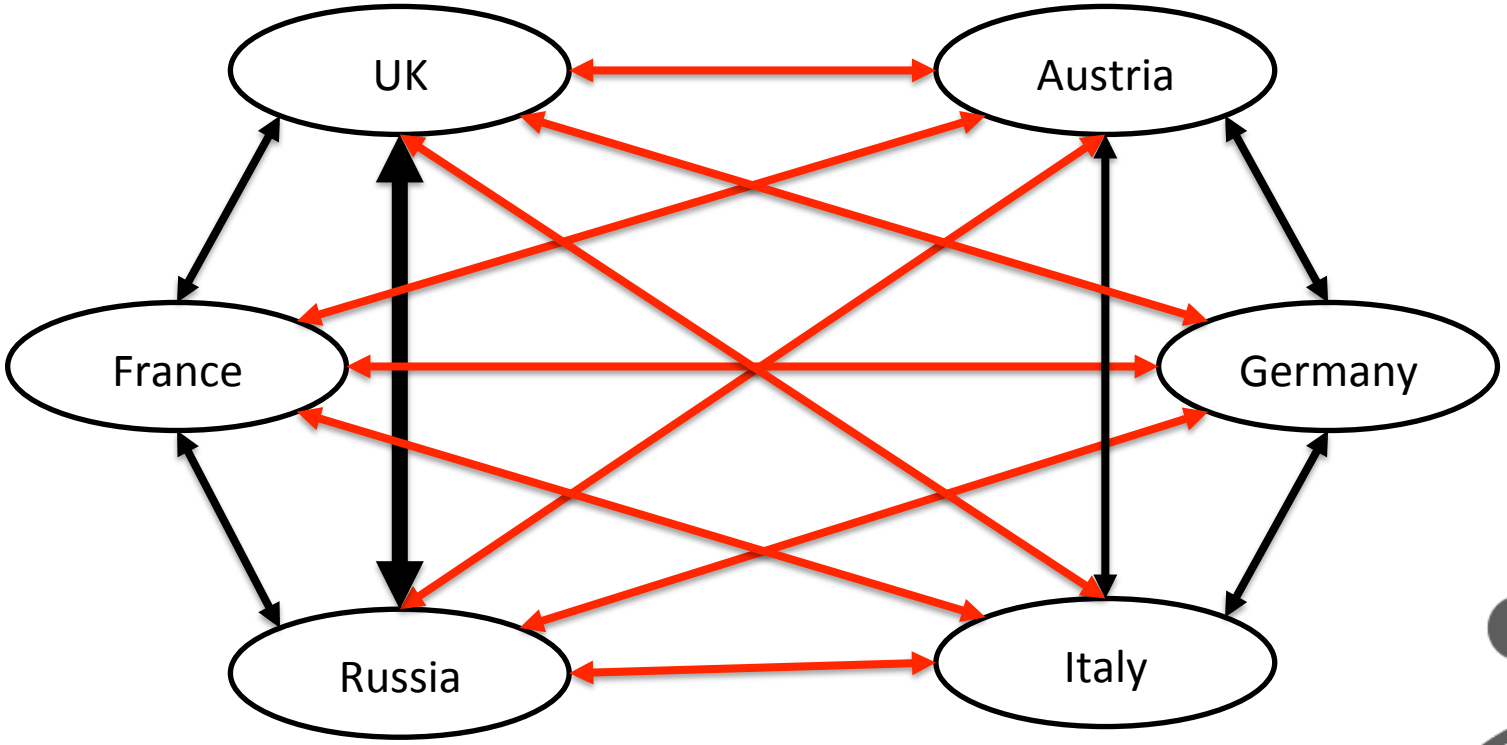
Allies and Enemies



Allies and Enemies

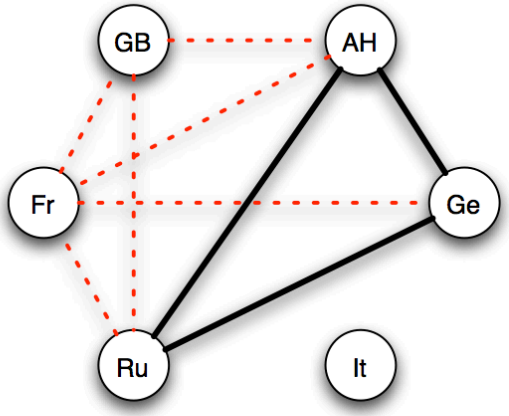


Allies and Enemies

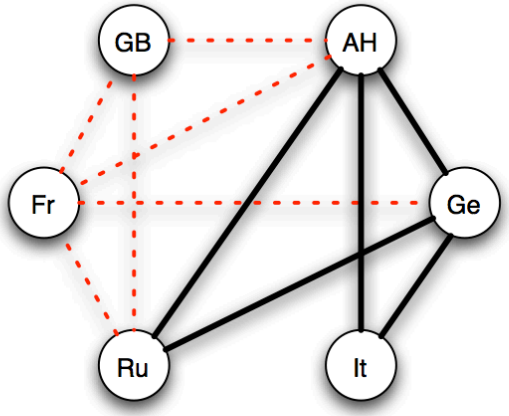


Predicting WWI

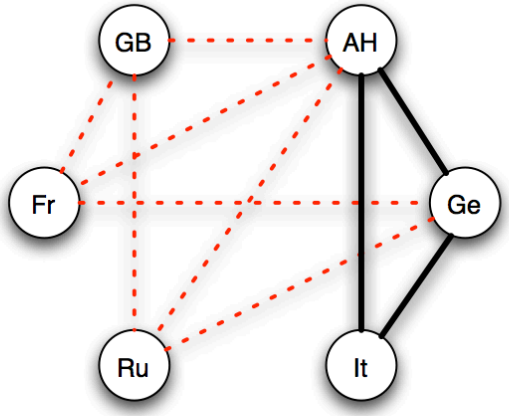
[Easley and Kleinberg]



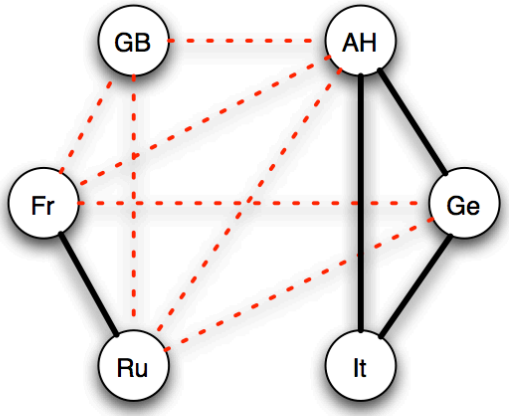
(a) *Three Emperors' League 1872–81*



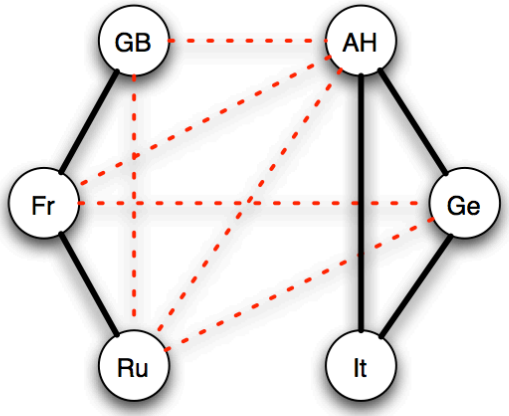
(b) *Triple Alliance 1882*



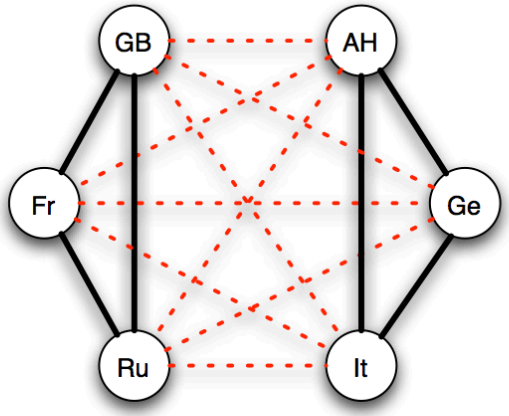
(c) *German-Russian Lapse 1890*



(d) *French-Russian Alliance 1891–94*



(e) *Entente Cordiale 1904*



(f) *British Russian Alliance 1907*

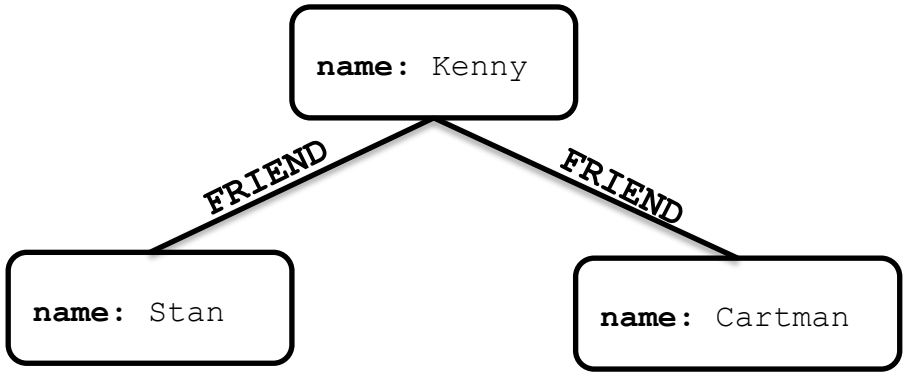
Strong Triadic Closure

It if a node has strong relationships to two neighbours, then these neighbours must have at least a weak relationship between them.

[Wikipedia]

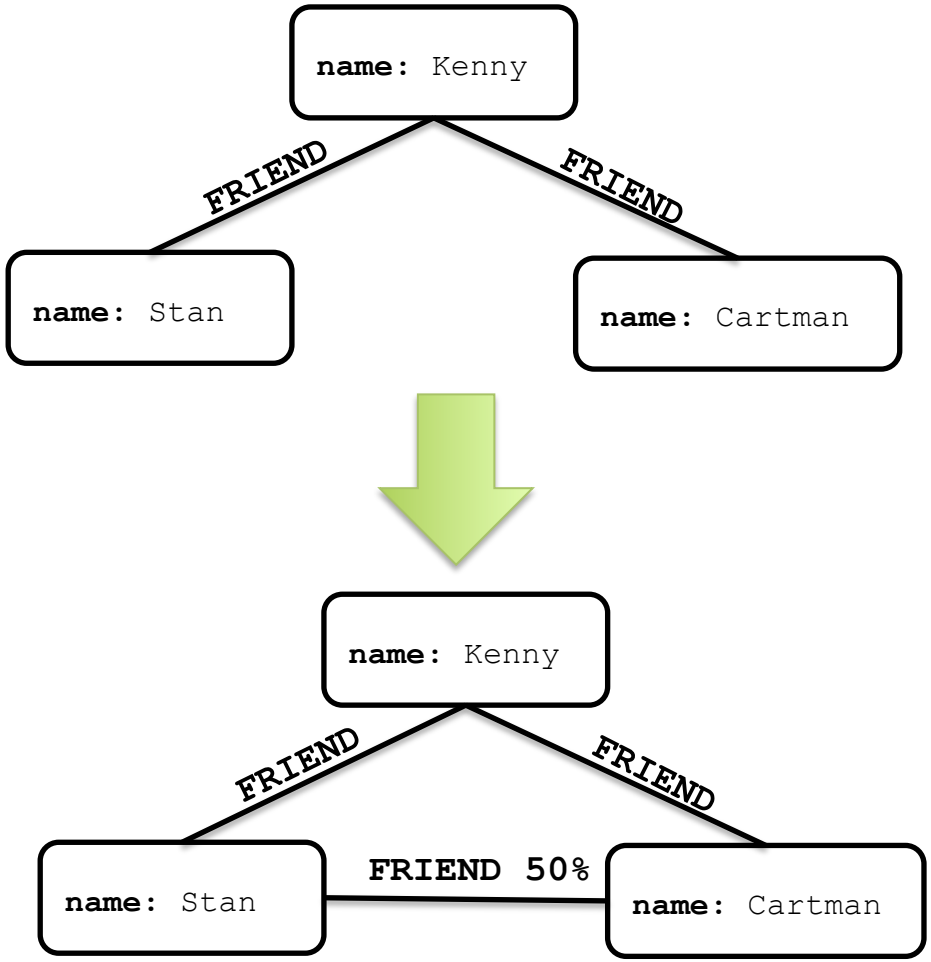
Triadic Closure

(weak relationship)



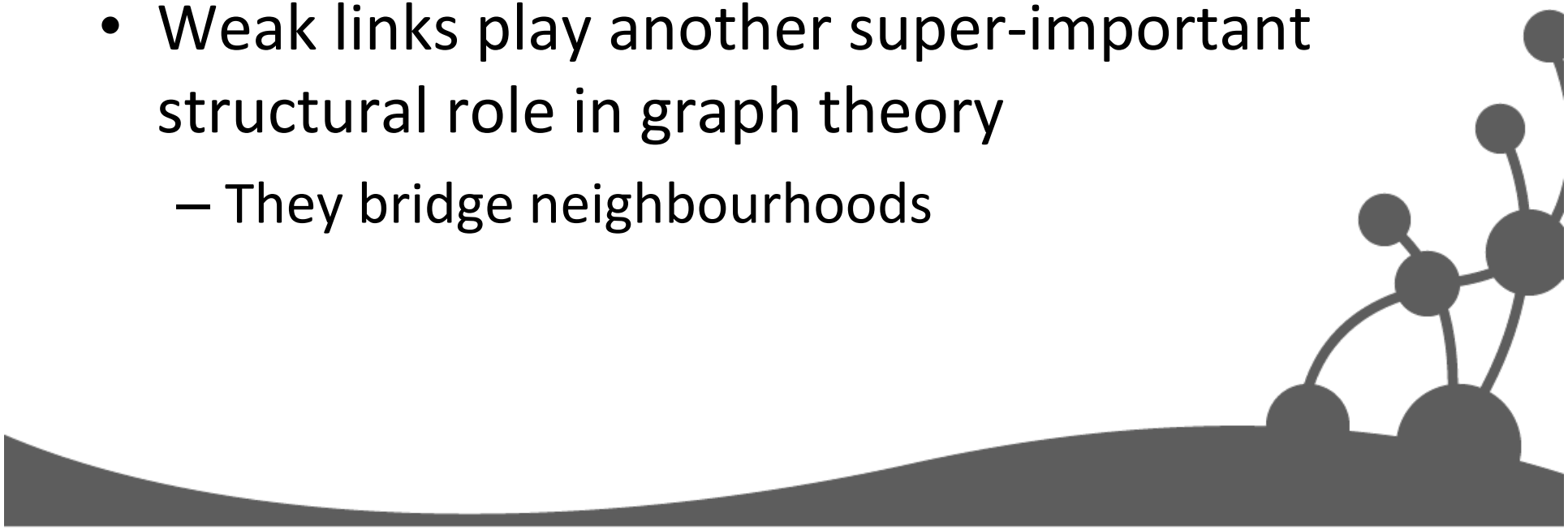
Triadic Closure

(weak relationship)

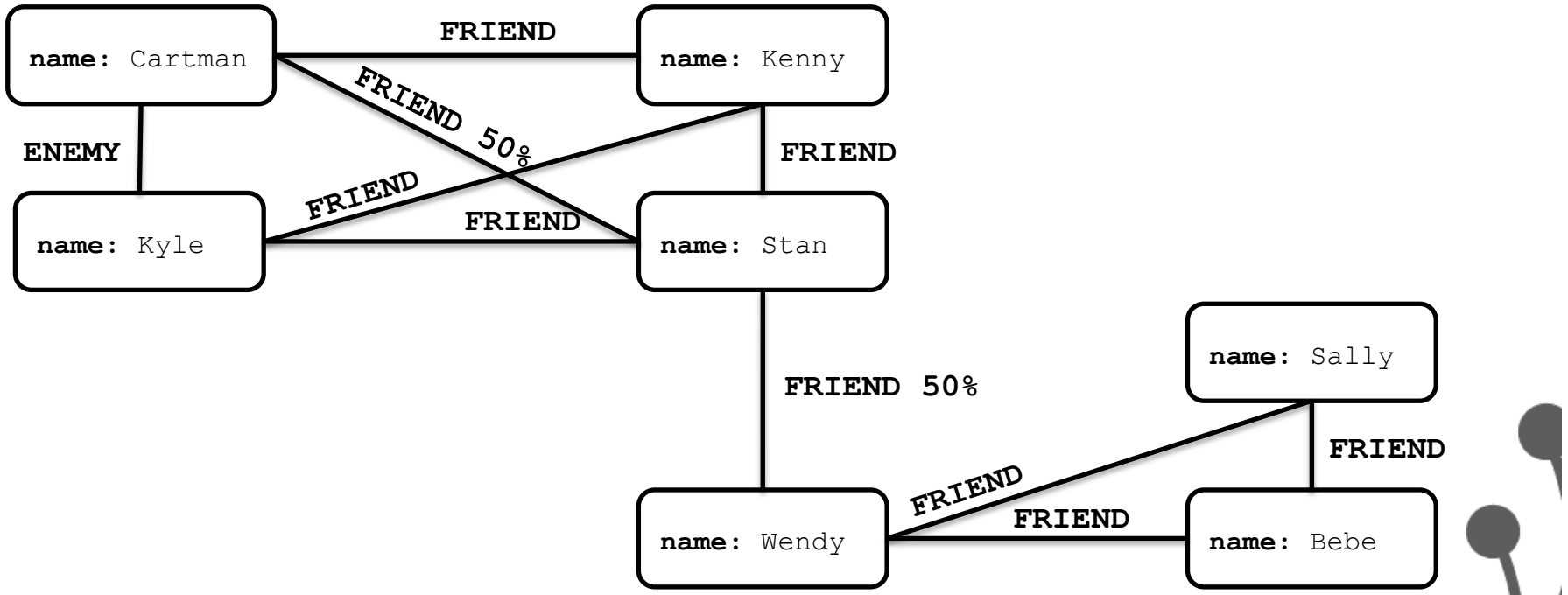


Weak relationships

- Relationships can have “strength” as well as intent
  - Think: weighting on a relationship in a property graph
- Weak links play another super-important structural role in graph theory
  - They bridge neighbourhoods



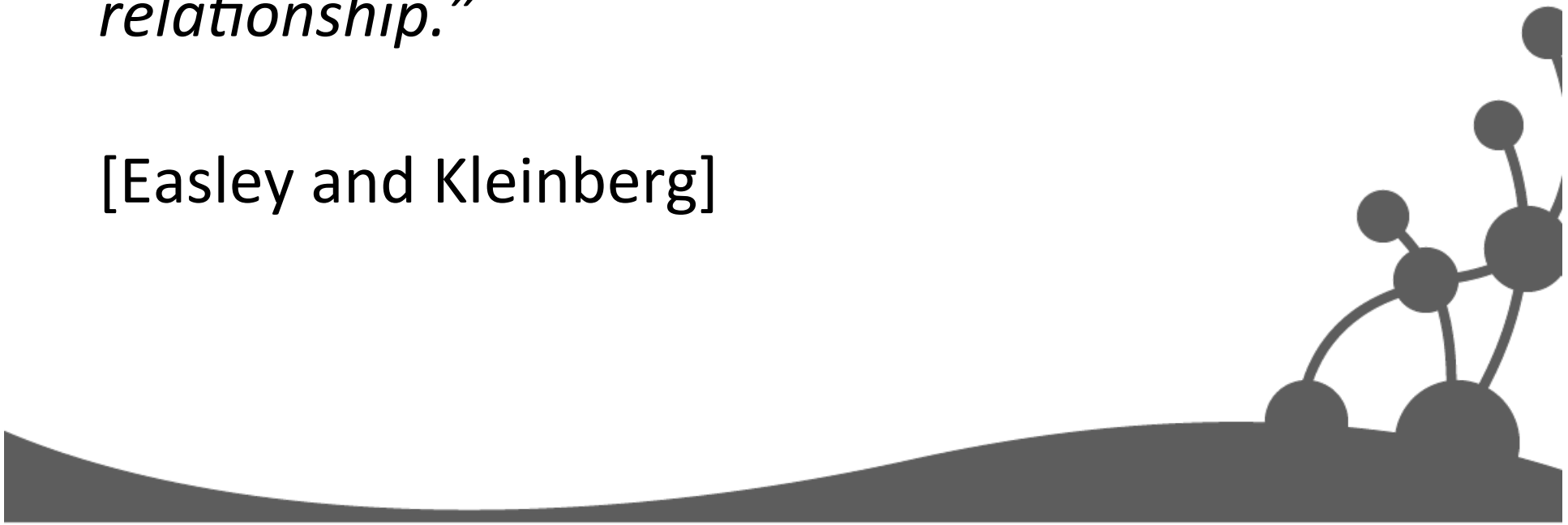
Local Bridges



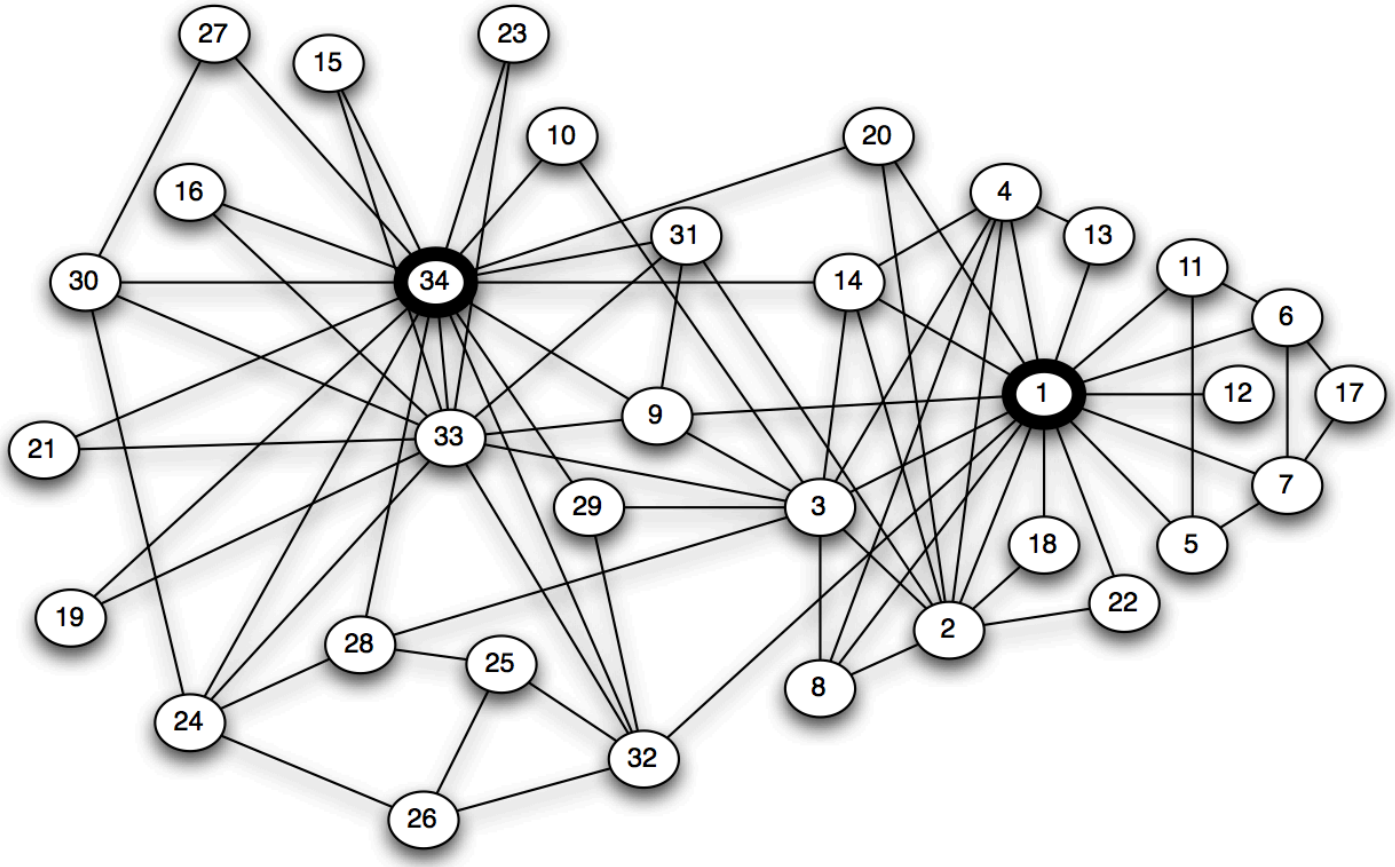
Local Bridge Property

*"If a node **A** in a network satisfies the Strong Triadic Closure Property and is involved in at least two strong relationships, then any local bridge it is involved in must be a weak relationship."*

[Easley and Kleinberg]

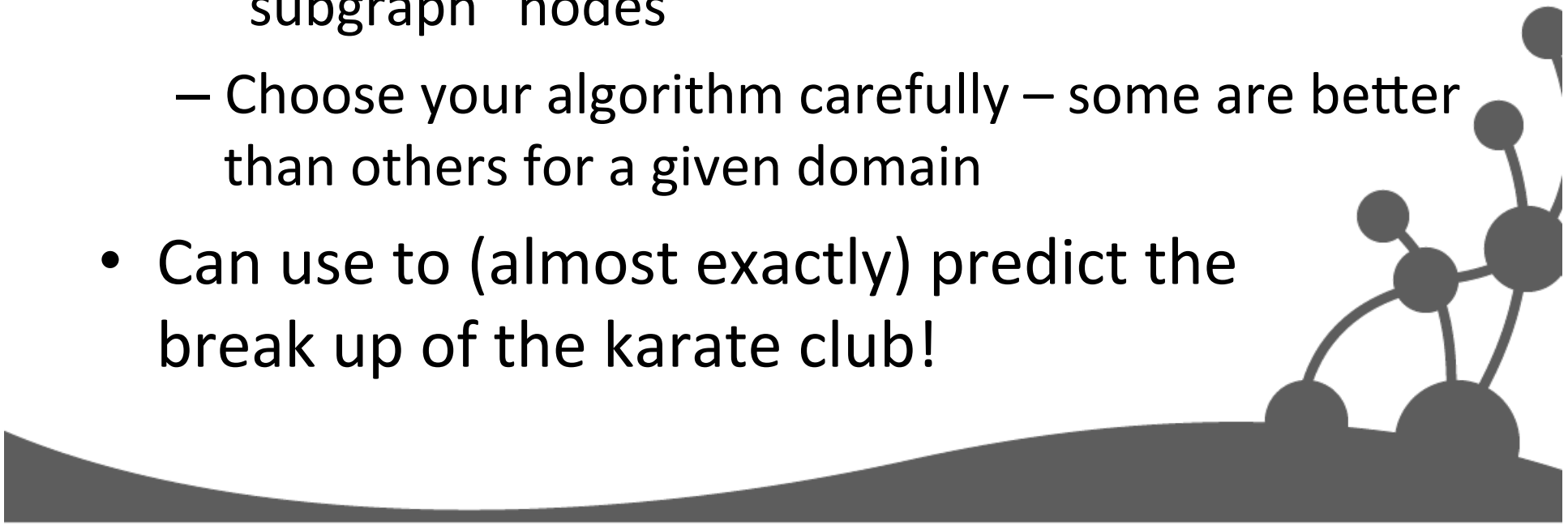


University Karate Club



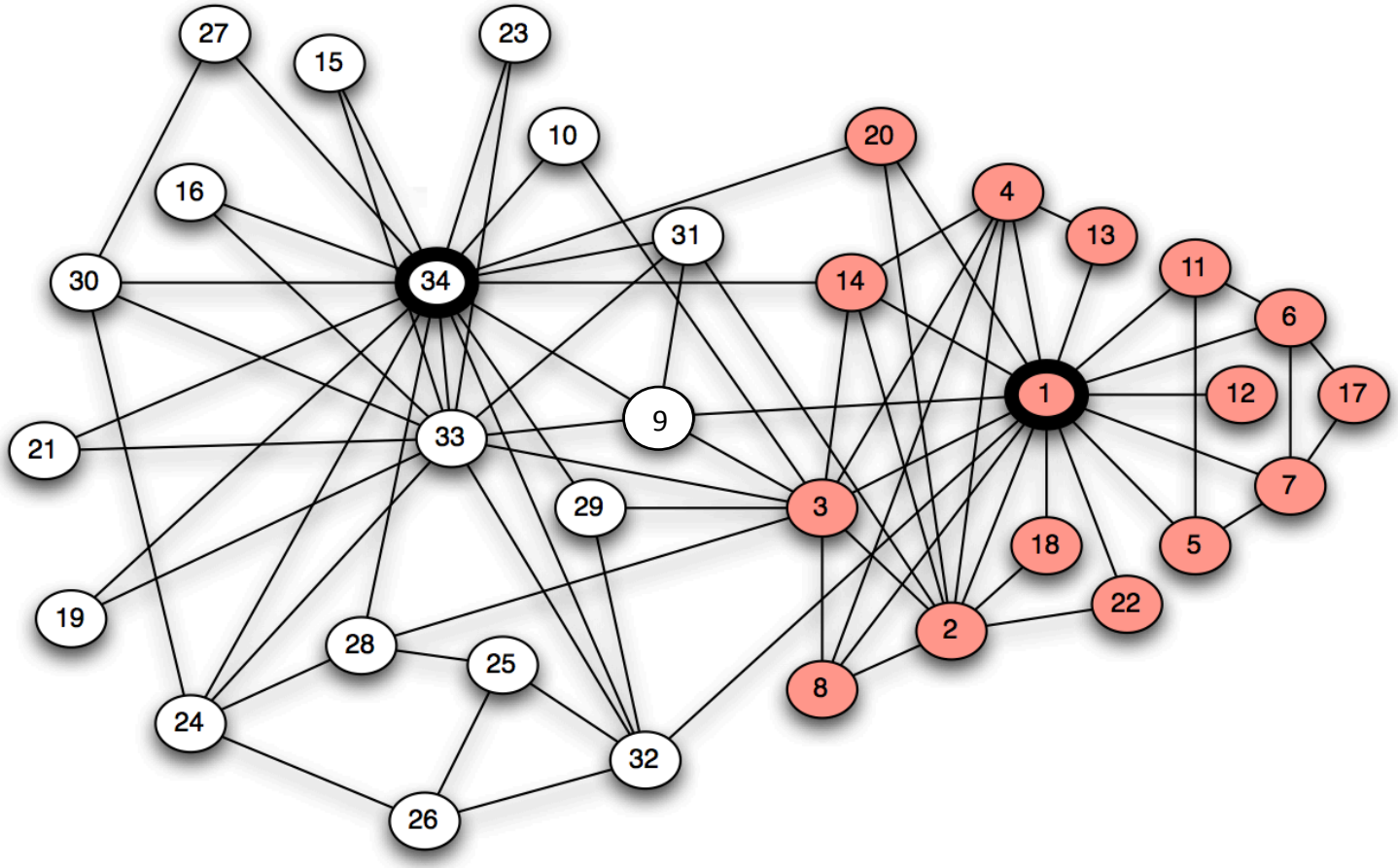
Graph Partitioning

- (NP) Hard problem
  - Recursively remove the spanning links between dense regions
  - Or recursively merge nodes into ever larger "subgraph" nodes
  - Choose your algorithm carefully – some are better than others for a given domain
- Can use to (almost exactly) predict the break up of the karate club!



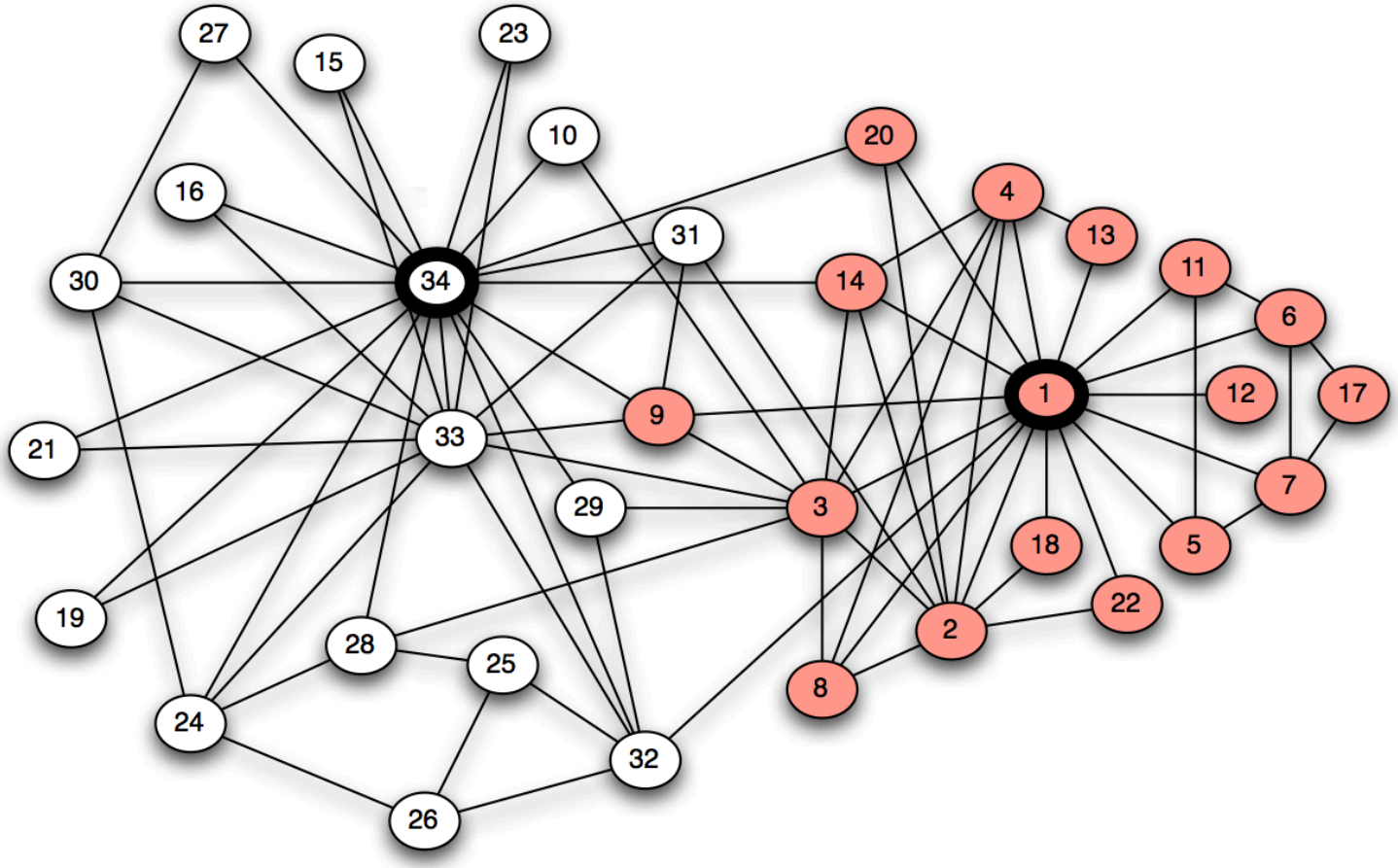
University Karate Clubs

(predicted by Graph Theory)



University Karate Clubs

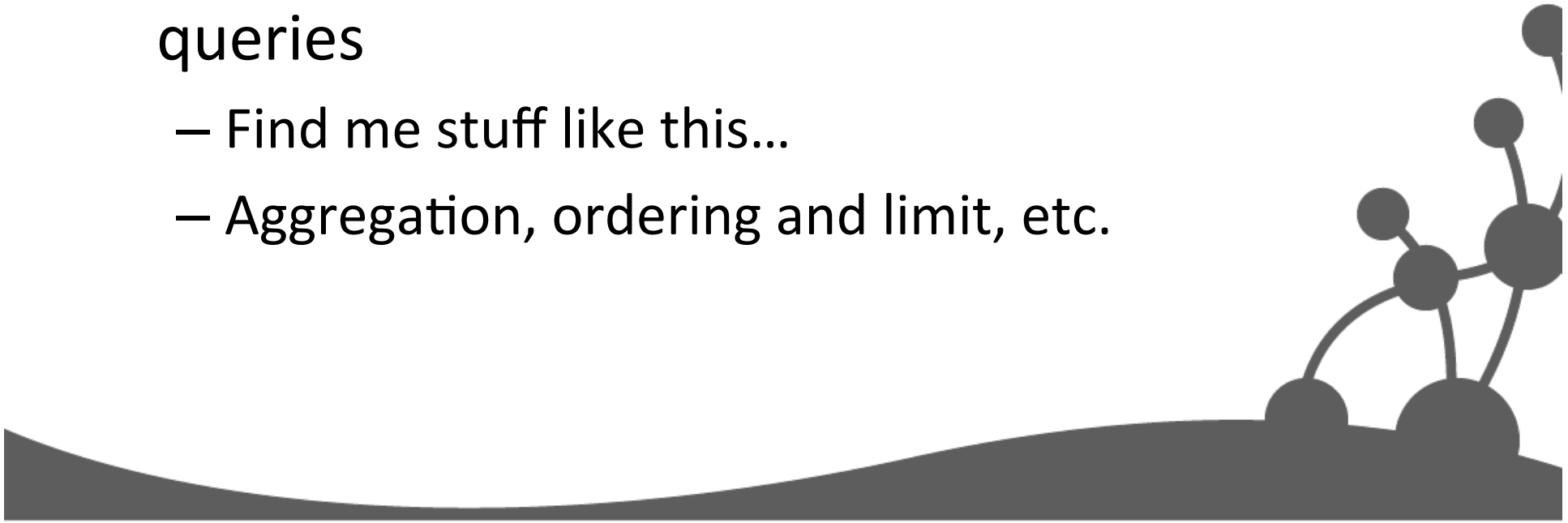
(what actually happened!)

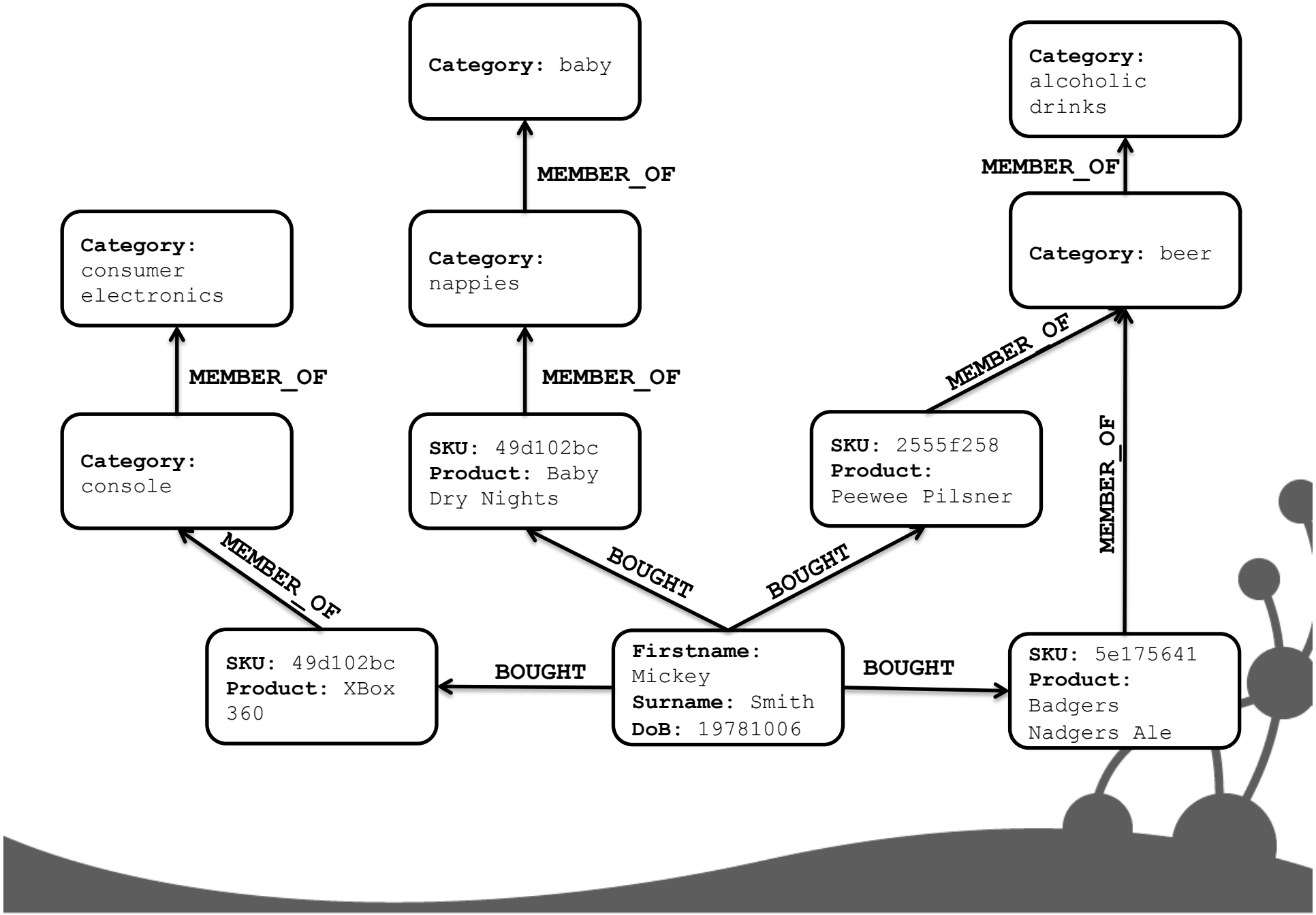


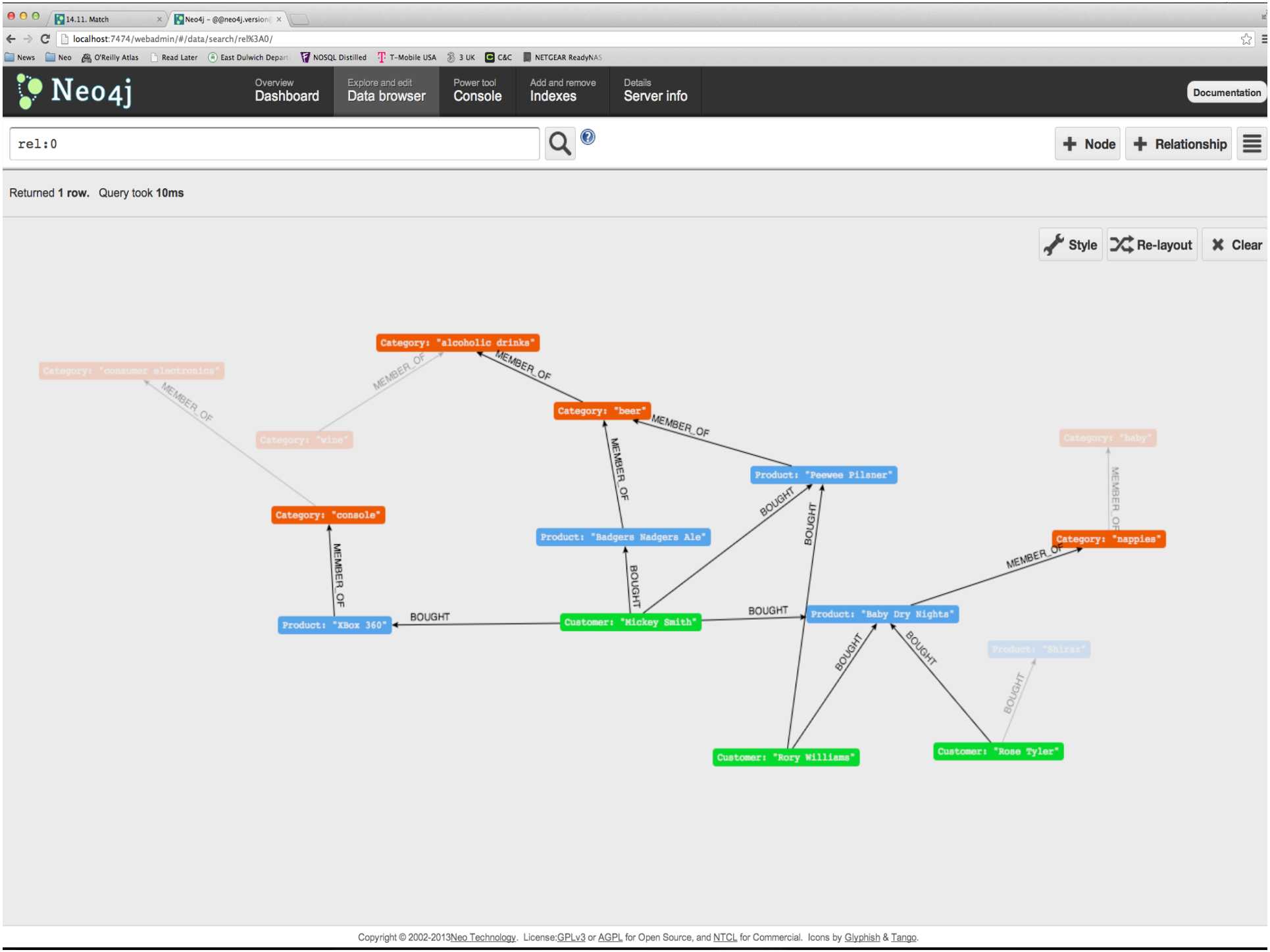

john
crane
Ltd

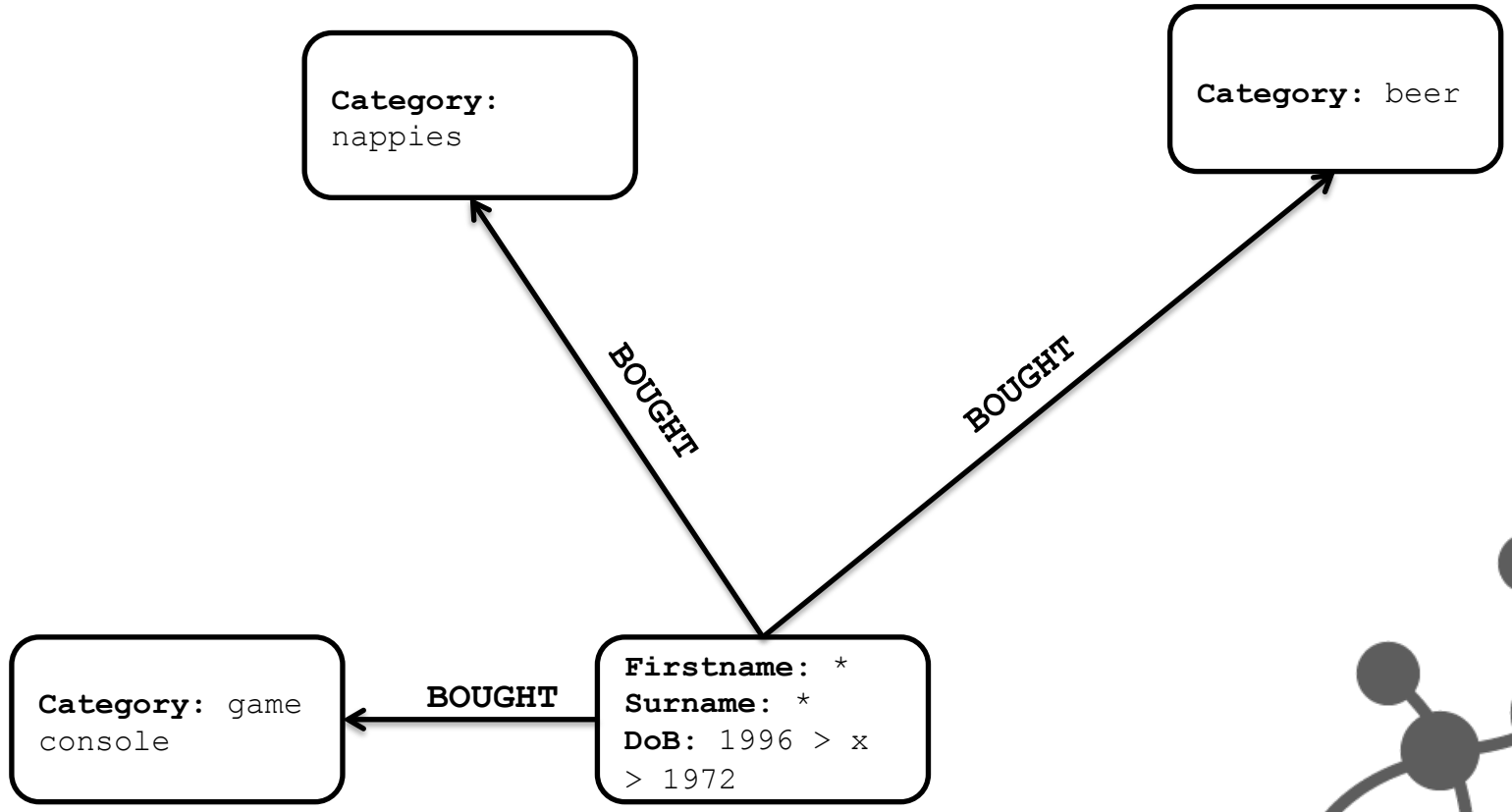
Cypher

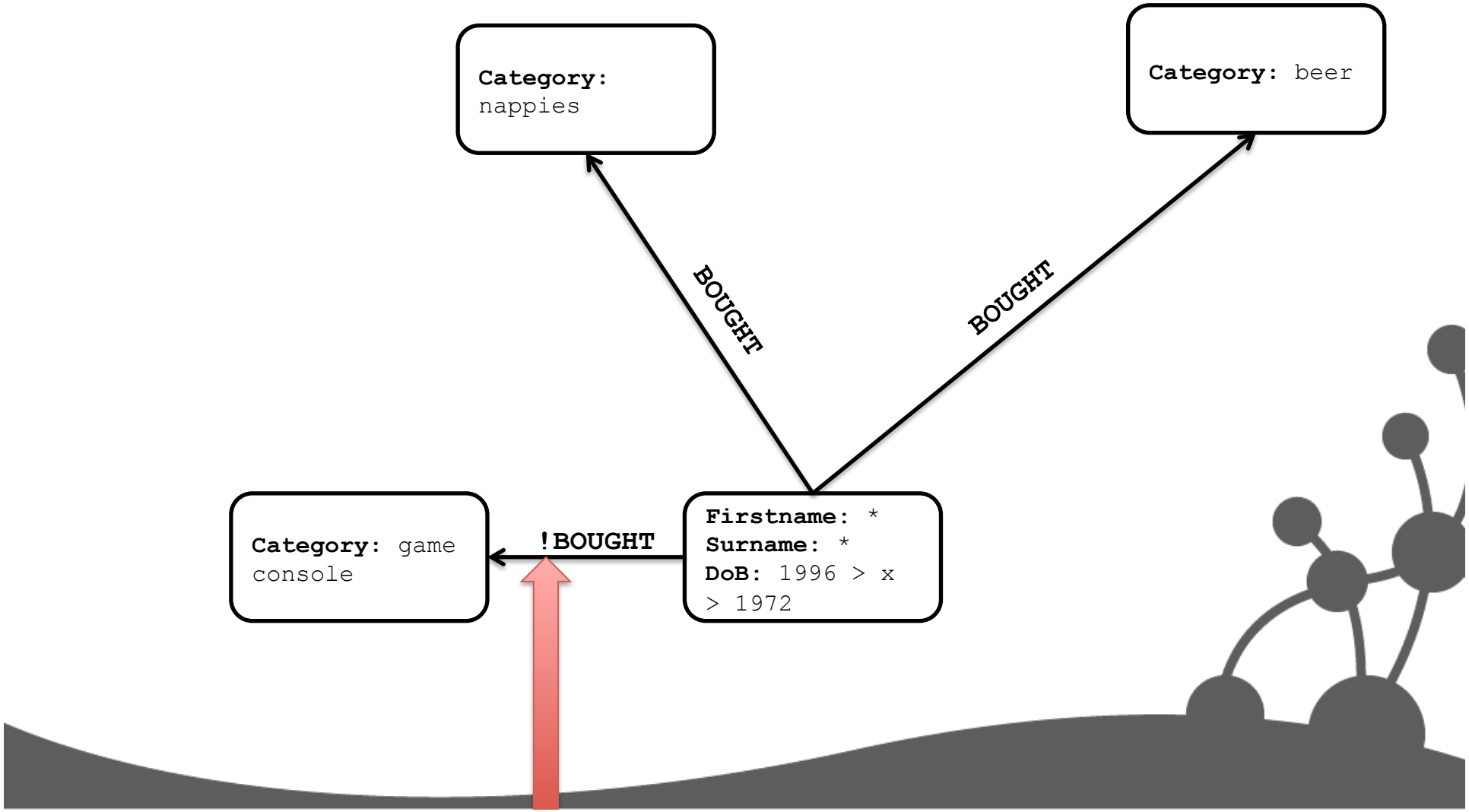
- Declarative graph pattern matching language
  - "SQL for graphs"
  - Columnar results
- Supports graph matching commands and queries
  - Find me stuff like this…
  - Aggregation, ordering and limit, etc.

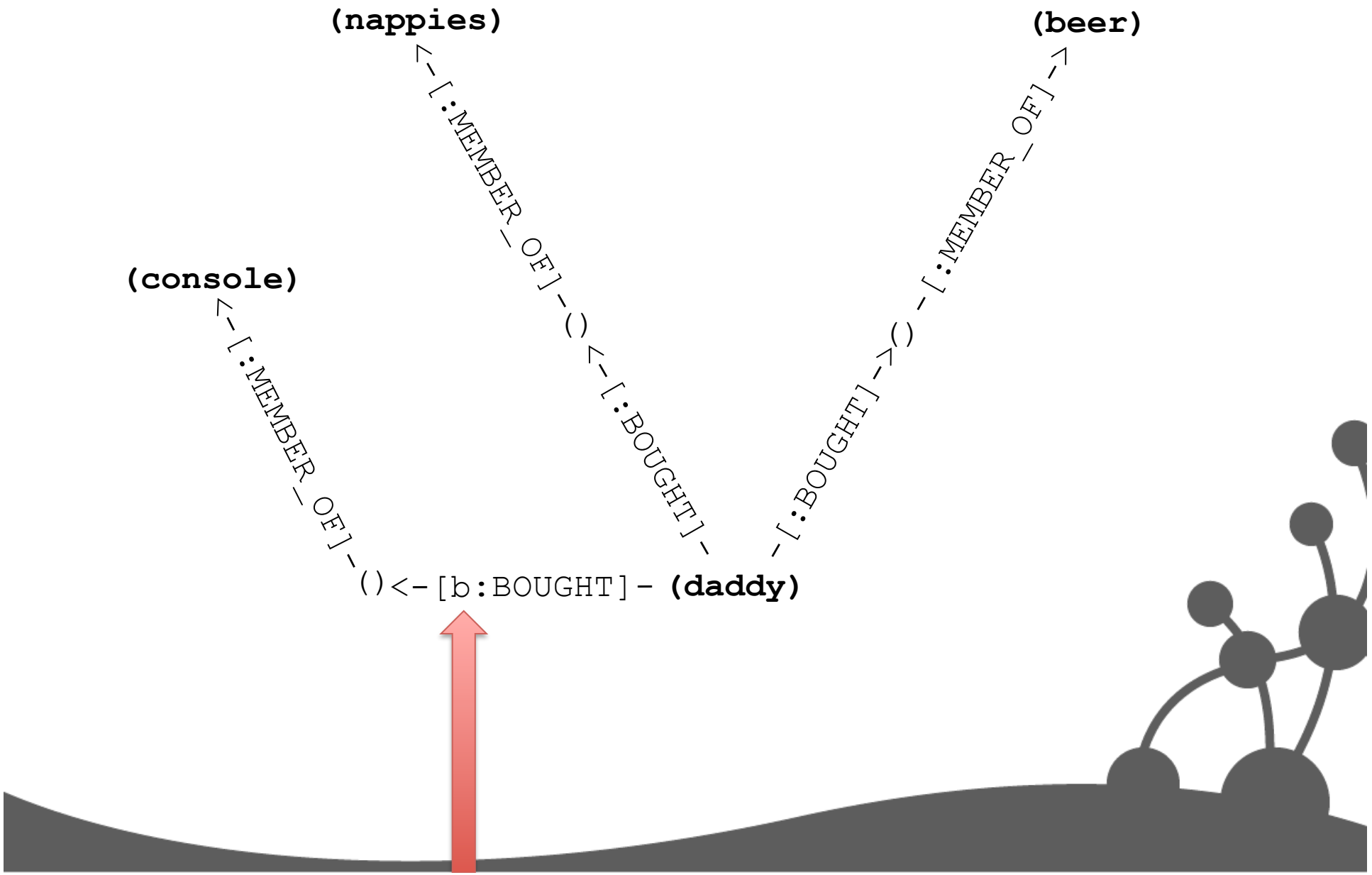



Category: baby
MEMBER_OF
Category: alcoholic drinks
MEMBER_OF
Category: consumer electronics
Category: nappies
Category: beer
MEMBER_OF
MEMBER_OF
MEMBER_OF
Category: console
SKU: 49d102bc Product: Baby Dry Nights
SKU: 2555f258 Product: Peewee Pilsner
MEMBER_OF
BOUGHT
BOUGHT
MEMBER_OF
SKU: 49d102bc Product: XBox 360
BOUGHT
Firstname: Mickey Surname: Smith DoB: 19781006
BOUGHT
SKU: 5e175641 Product: Badgers Nadgers Ale


14.11. Match
Neo4j - @@neo4j.version
localhost:7474/webadmin/#/data/search/rel%3A0/
Neo4j
Overview
Dashboard
Explore and edit
Data browser
Power tool
Console
Add and remove
Indexes
Details
Server info
Documentation
rel:0
Node
Relationship
Returned 1 row. Query took 10ms
Style
Re-layout
Clear
Category: "alcoholic drinks"
Category: "consumer electronics"
Category: "wine"
Category: "beer"
Category: "console"
Category: "baby"
Category: "nappies"
Product: "Peewee Pilsner"
Product: "Badgers Nadgers Ale"
Product: "XBox 360"
Product: "Baby Dry Nights"
Product: "Shiraz"
Customer: "Mickey Smith"
Customer: "Rory Williams"
Customer: "Rose Tyler"
MEMBER_OF
BOUGHT
Copyright © 2002-2013Neo Technology. License:GPLv3 or AGPL for Open Source, and NTCL for Commercial. Icons by Glyphish & Tango.


Category: nappies
Category: beer
BOUGHT
BOUGHT
Category: game console
BOUGHT
Firstname: *
Surname: *
DoB: 1996 > x > 1972


Category: nappies
Category: beer
BOUGHT
BOUGHT
Category: game console
!BOUGHT
Firstname: *
Surname: *
DoB: 1996 > x > 1972


(nappies)
(beer)
(console)
<-[:MEMBER_OF]-
<-[:BOUGHT]-
-[:BOUGHT]->
-[:MEMBER_OF]->
<-[:MEMBER_OF]-
()<-[b:BOUGHT]- (daddy)

Flatten the graph

```
(daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(nappies)
(daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(beer)
(daddy)-[b:BOUGHT]->()-[:MEMBER_OF]->(console)
```

Wrap in a Cypher MATCH clause

```
MATCH (daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(nappies),
(daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(beer),
(daddy)-[b:BOUGHT]->()-[:MEMBER_OF]->(console)
```

Cypher WHERE clause

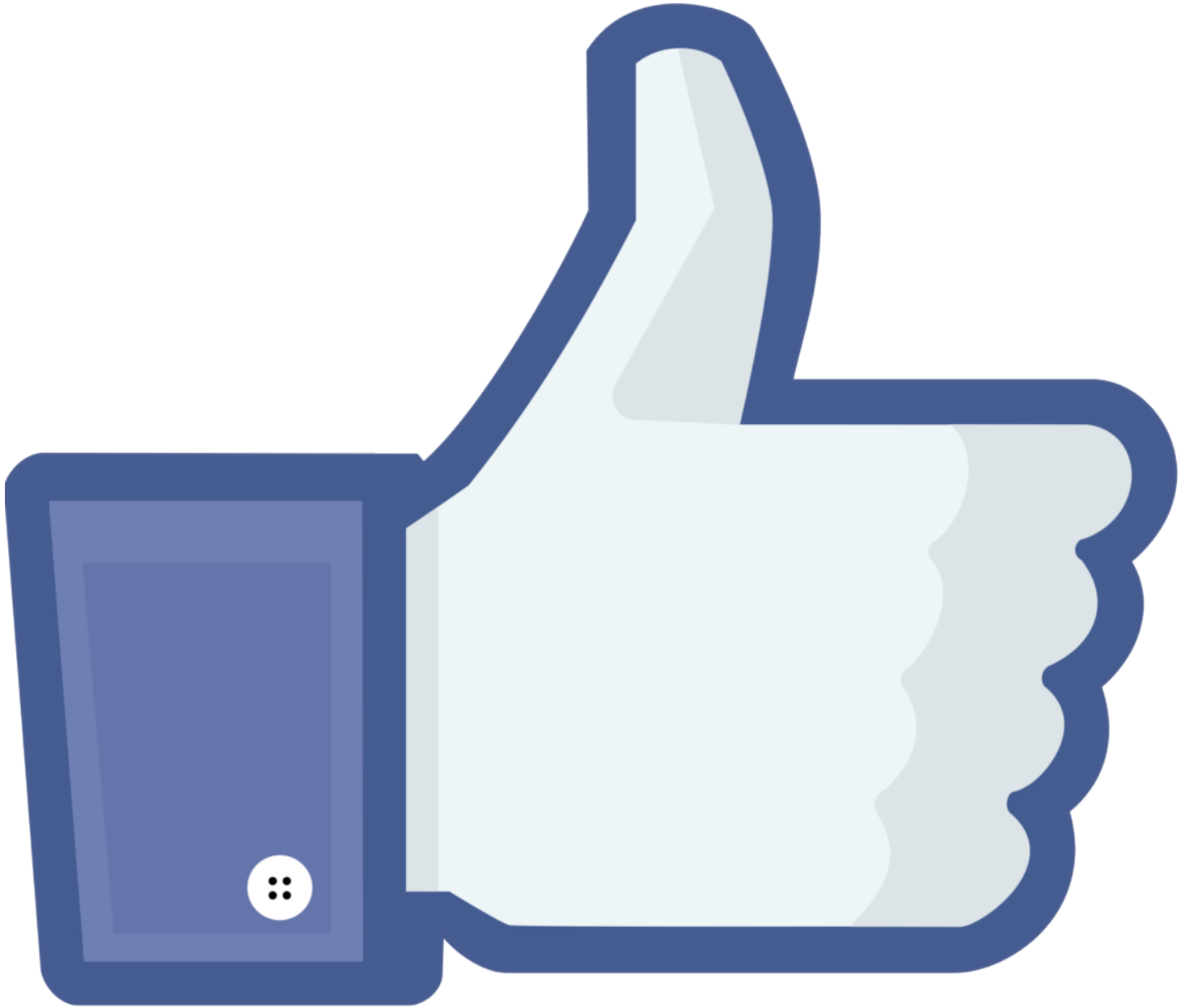
```
MATCH (daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(nappies),
(daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(beer),
(daddy)-[b:BOUGHT]->()-[:MEMBER_OF]->(console)
WHERE b is null
```

Full Cypher query

```
START beer=node:categories(category='beer'),
  nappies=node:categories(category='nappies'),
  xbox=node:products(product='xbox 360')

MATCH (daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(beer),
  (daddy)-[:BOUGHT]->()-[:MEMBER_OF]->(nappies),
  (daddy)-[b?:BOUGHT]->(xbox)

WHERE b is null

RETURN distinct daddy
```

Results

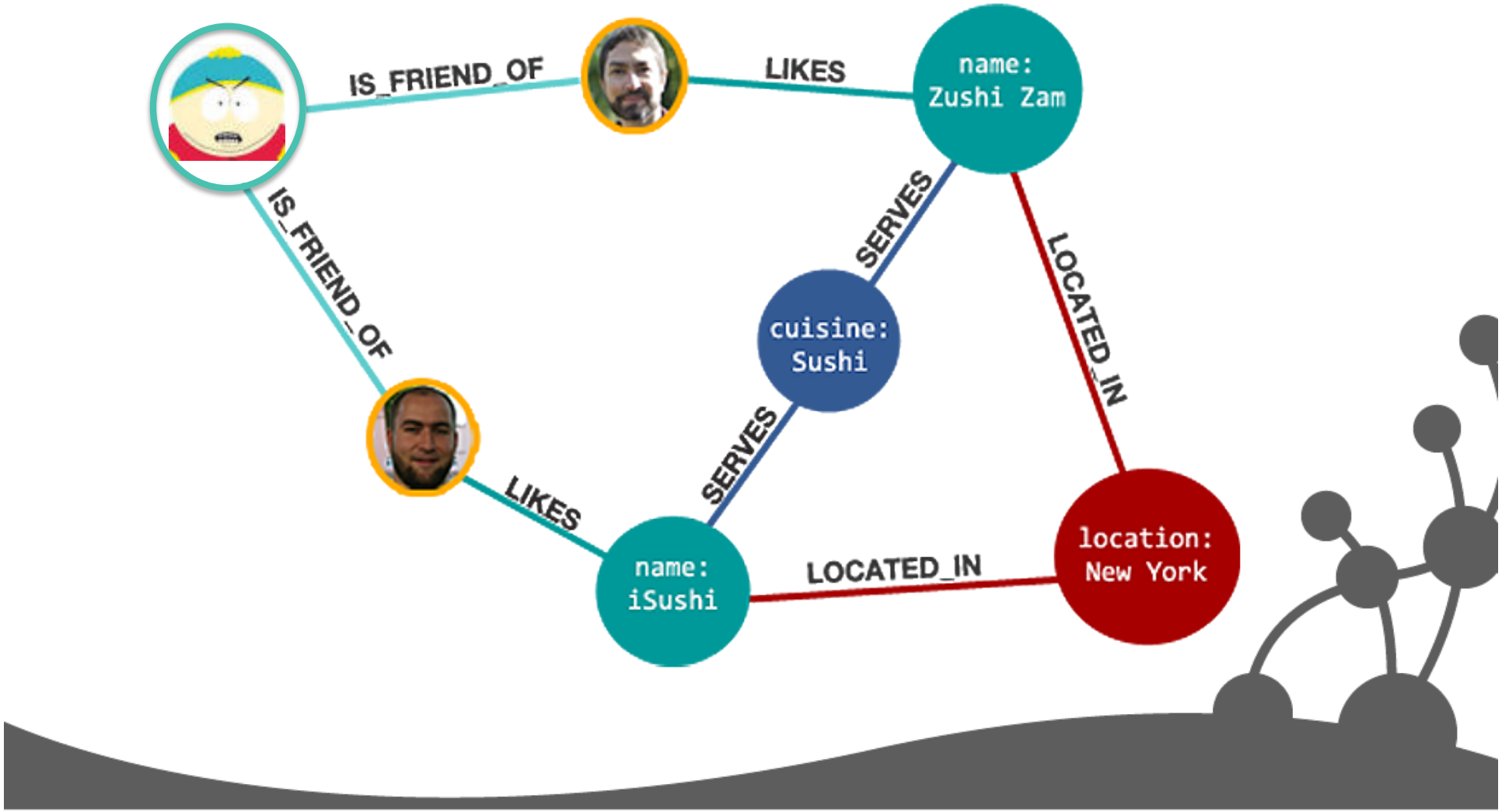
```
==> +-----------------------------------------+
==> | daddy                                   |
==> +-----------------------------------------+
==> | Node[15]{name:"Rory Williams",dob:19880121} |
==> +-----------------------------------------+
==> 1 row
==> 0 ms
==>
neo4j-sh (0)$
```



Facebook Graph Search

Which sushi restaurants in NYC do my friends like?

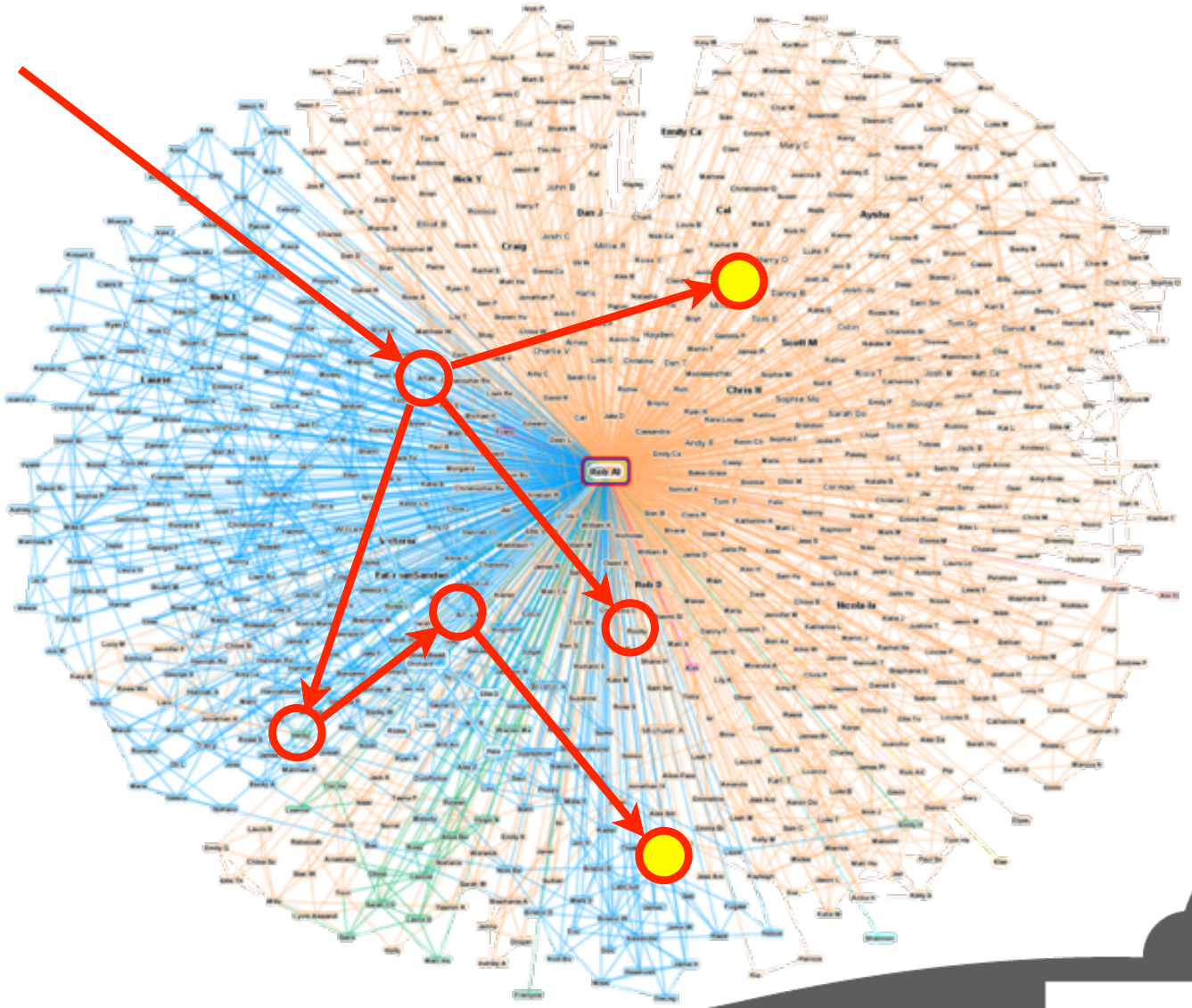
Graph Structure

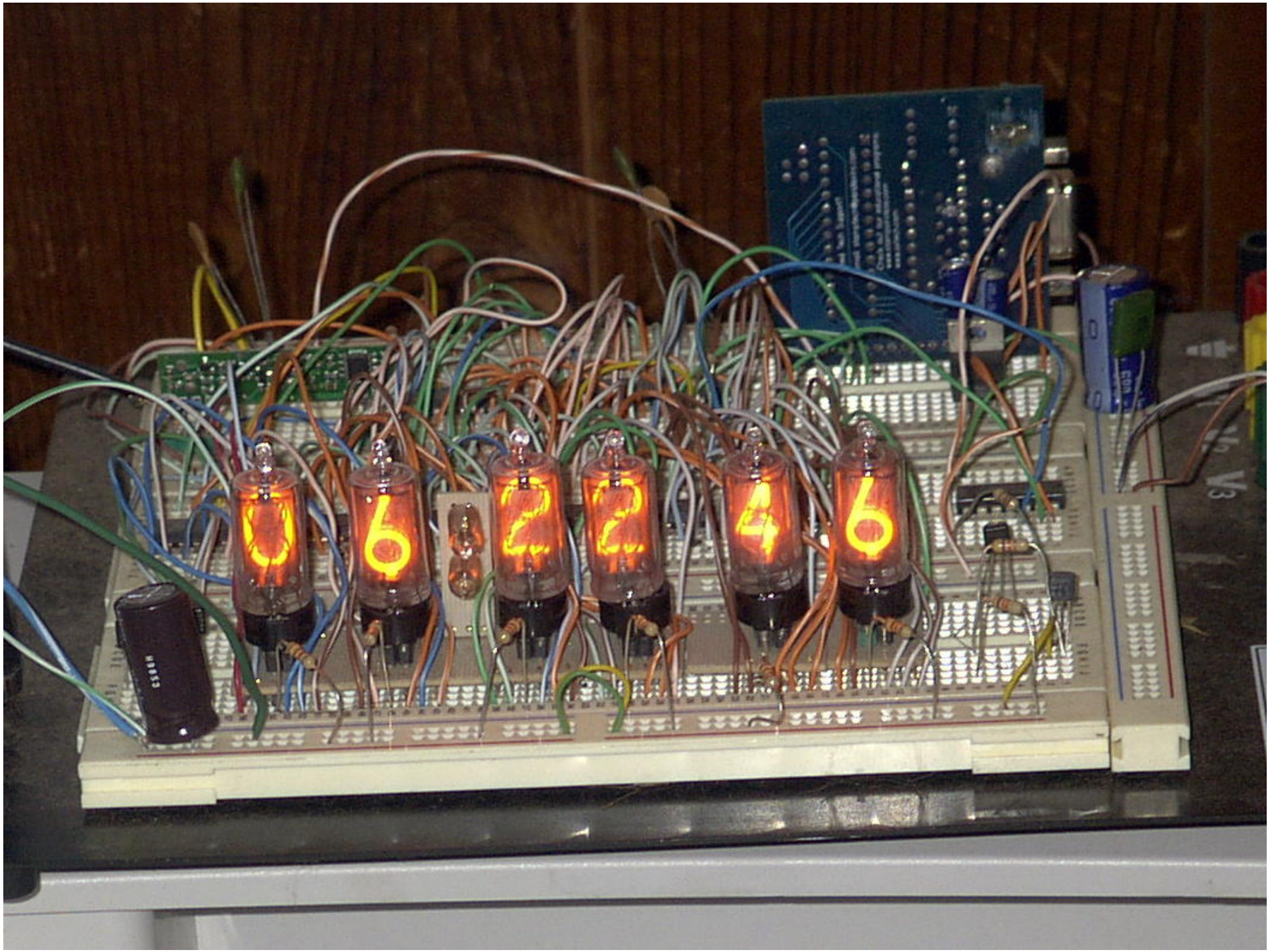


Cypher query

```
START me=node:person(name = 'Jim'),
      location=node:location(location='New York'),
      cuisine=node:cuisine(cuisine='Sushi')

MATCH (me)-[:IS_FRIEND_OF]->(friend)-[:LIKES]->(restaurant)
      -[:LOCATED_IN]->(location),(restaurant)-[:SERVES]->(cuisine)

RETURN restaurant
```

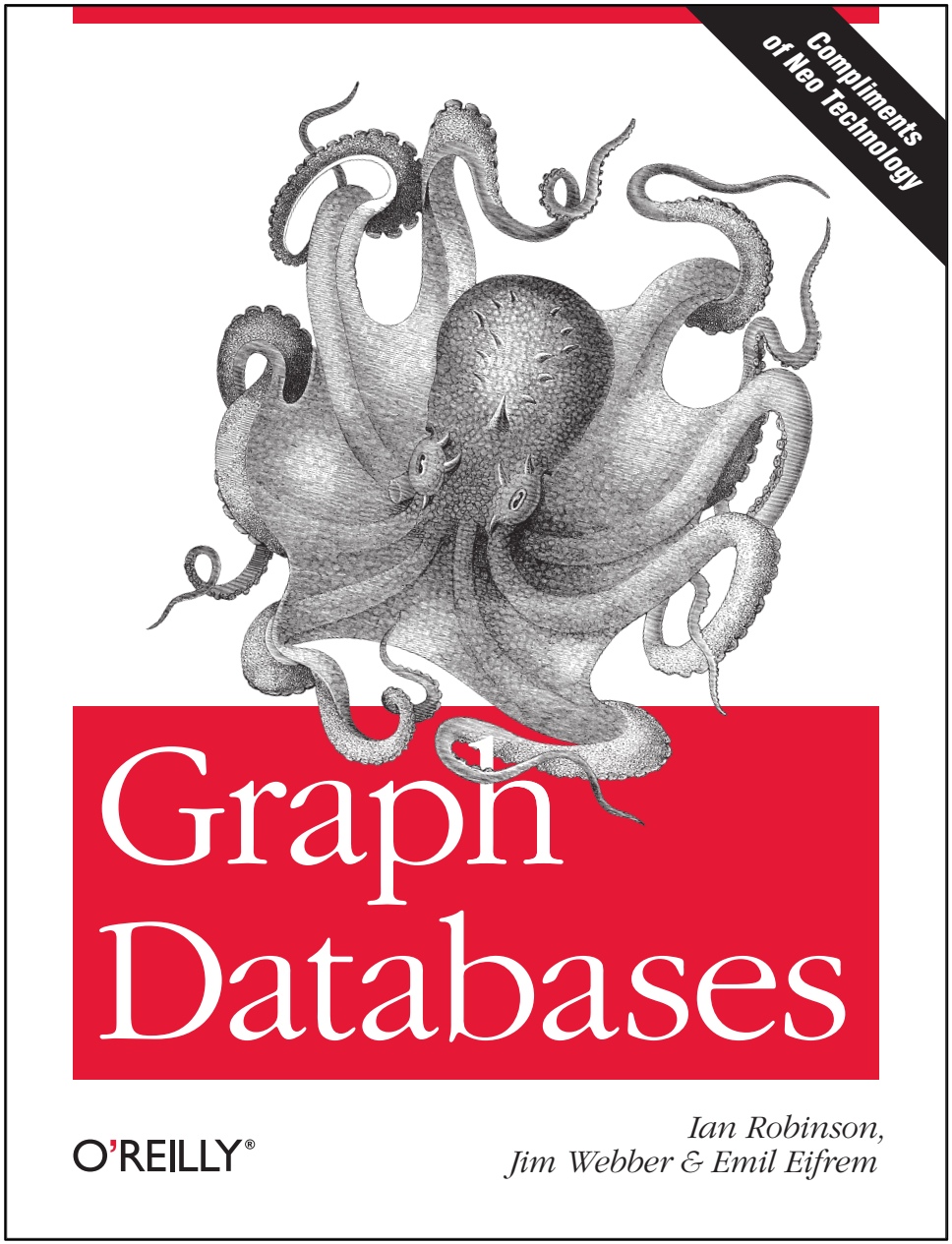
Search structure




0 6 2 2 4 6

What are graphs good for?

- Recommendations
- Pharmacology
- Business intelligence
- Social computing
- Geospatial
- MDM
- Data center management
- Web of things
- Genealogy
- Time series data
- Product catalogue
- Web analytics
- Scientific computing
- Indexing your *slow* RDBMS
- And much more!



Free O'Reilly book for everyone!

http://graphdatabases.com

Thanks for listening

Neo4j: **http://neo4j.org**

Me: **@jimwebber**