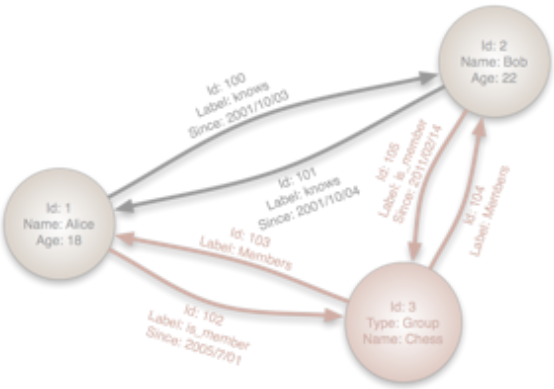
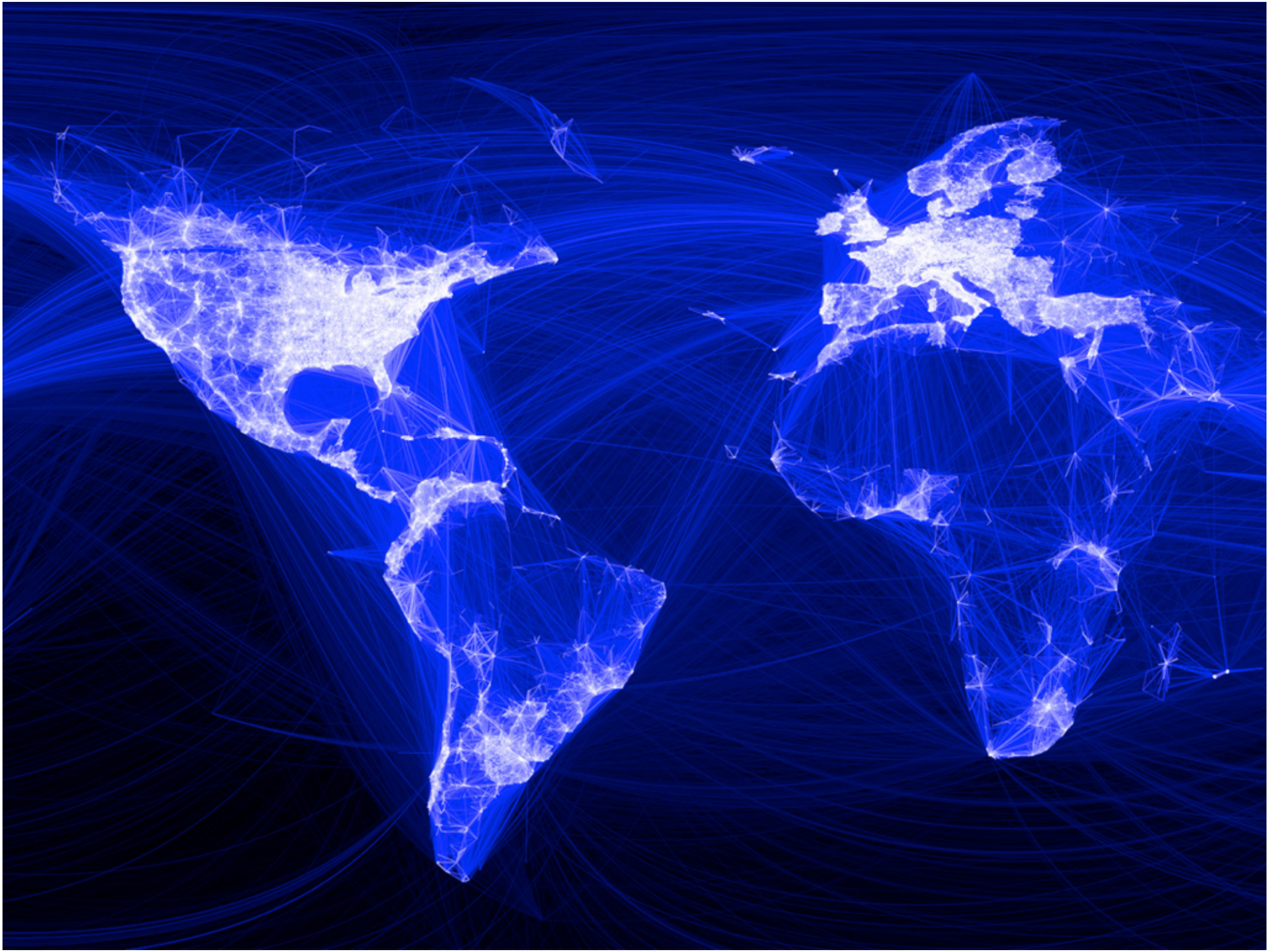
neo technology
graphs are everywhere

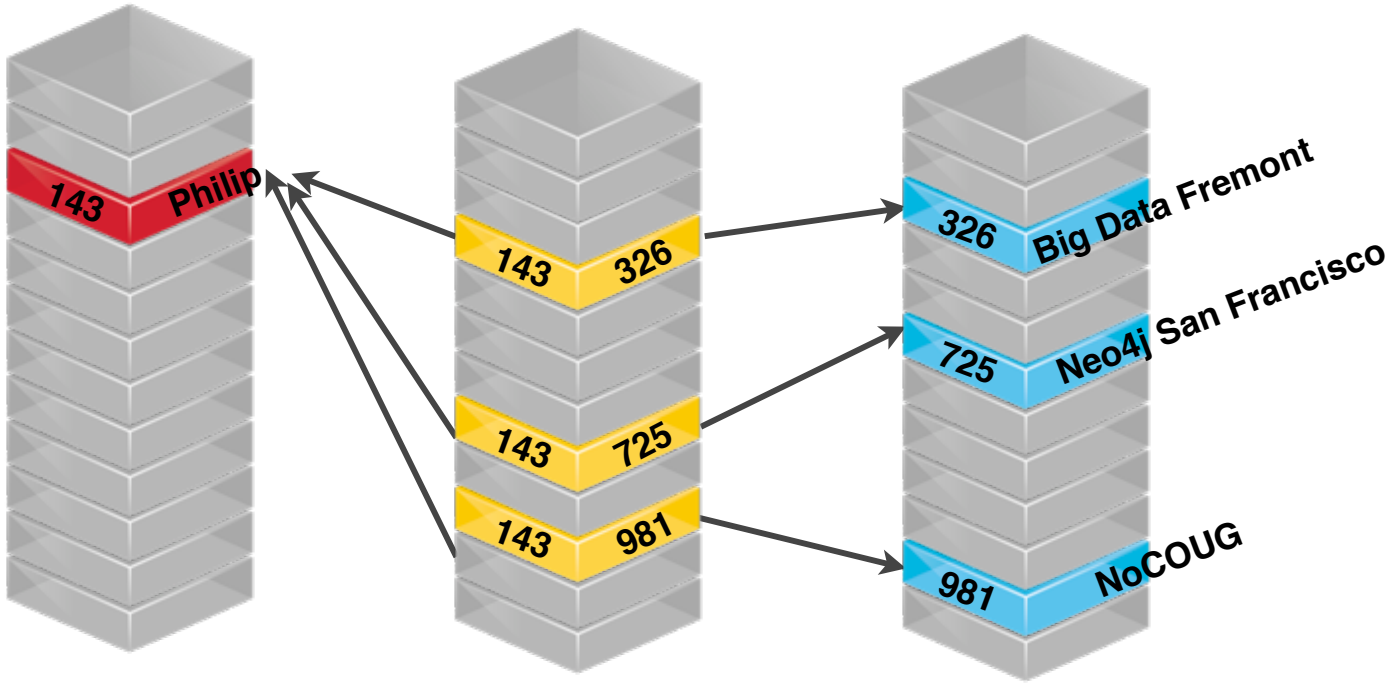
NoSQL Drill-Down: So What's a Graph Database?

NoCOUG Aug 2013

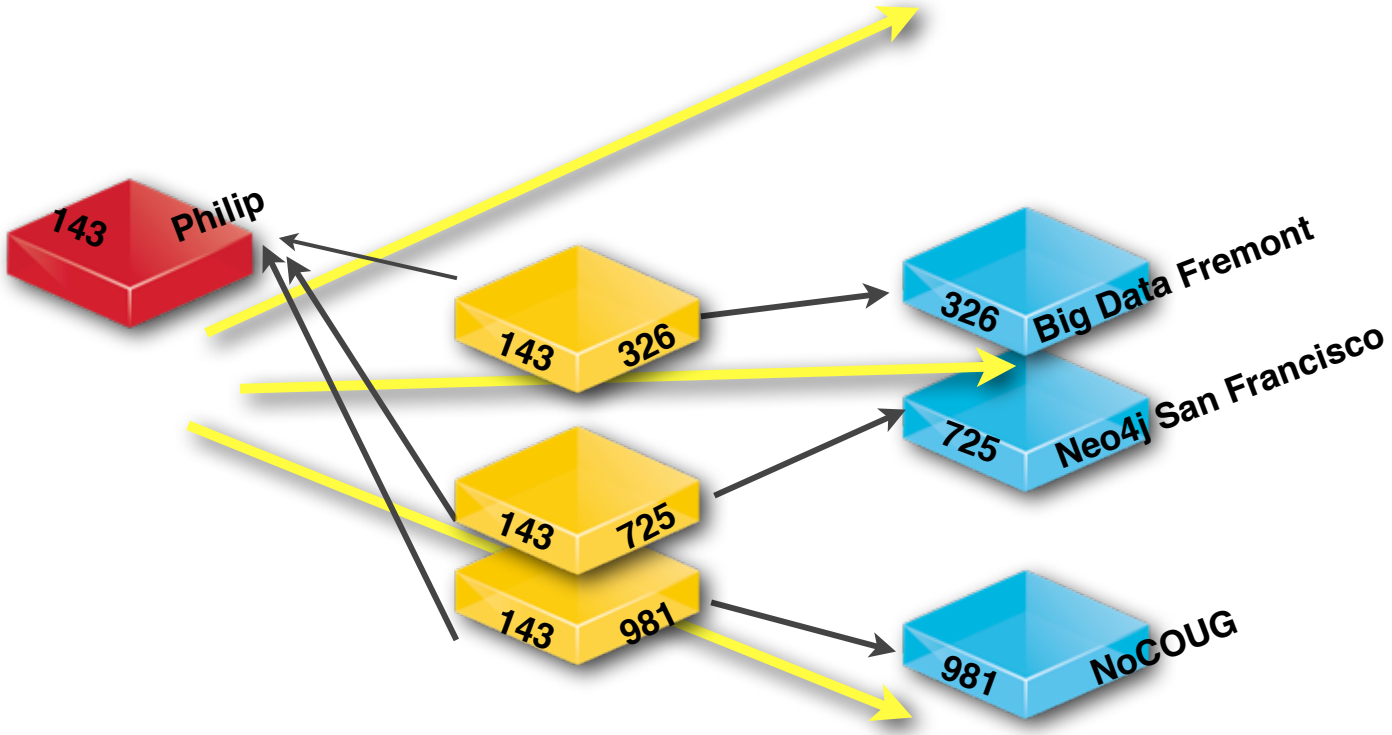
Philip Rathle
Sr. Director of Products for Neo4j
philip@neotechnology.com
@prathle



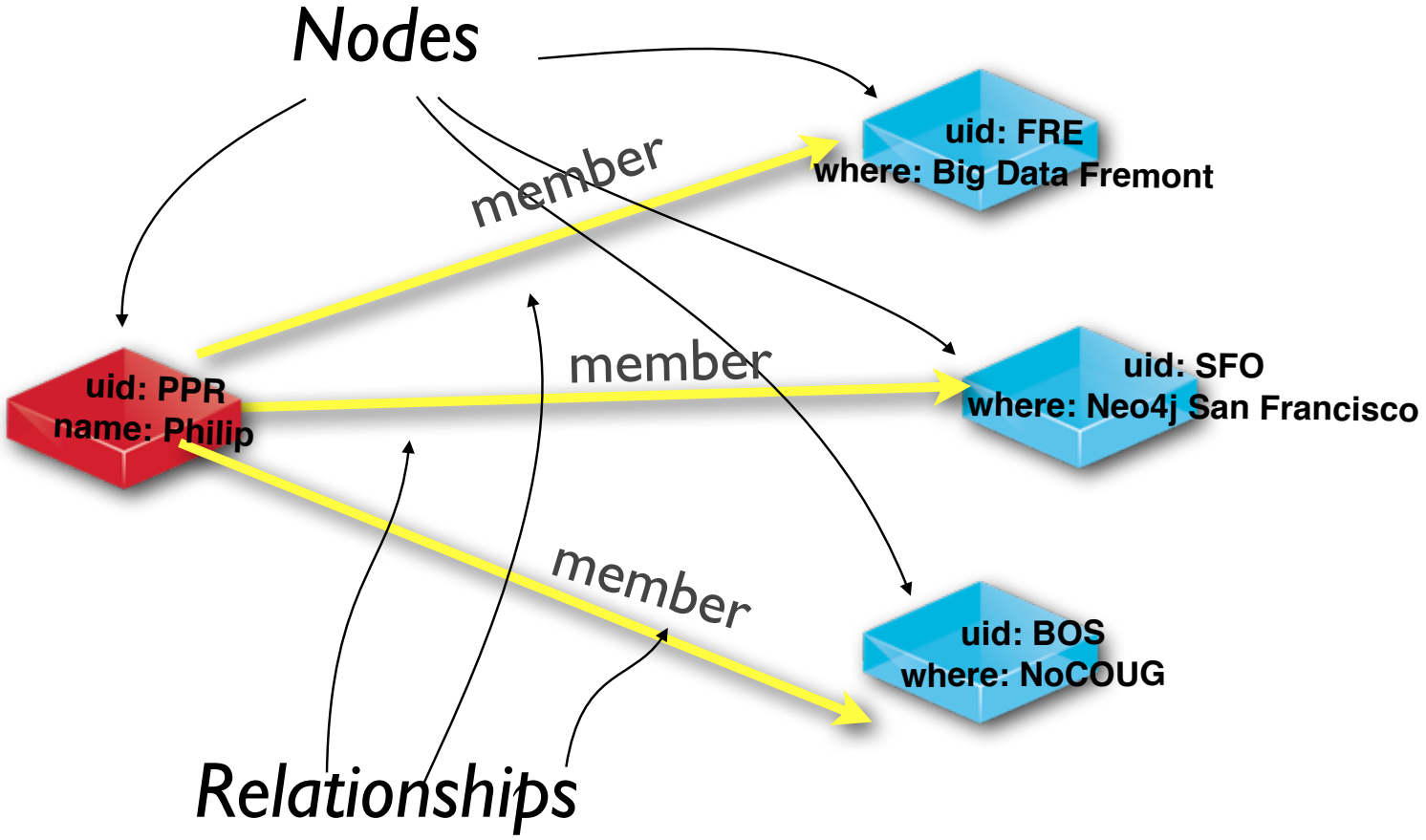




Member Member_Group Group


143
Philip
143
326
326
Big Data Fremont
725
Neo4j San Francisco
143
725
143
981
981
NoCOUG

A Property Graph



Sharif Nijim @snijim 3 Jul

Fired up working with graphs! #neo4j #graphdb #future #everywhere pic.twitter.com/nr3xC22474

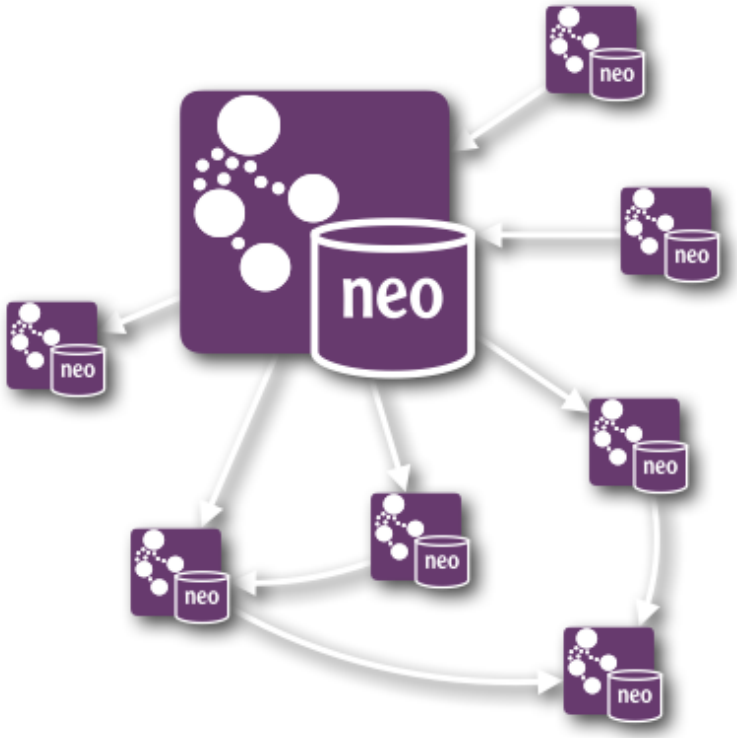
View photo

Domain-Driven Design Perspective

Put forth by Martin Fowler...

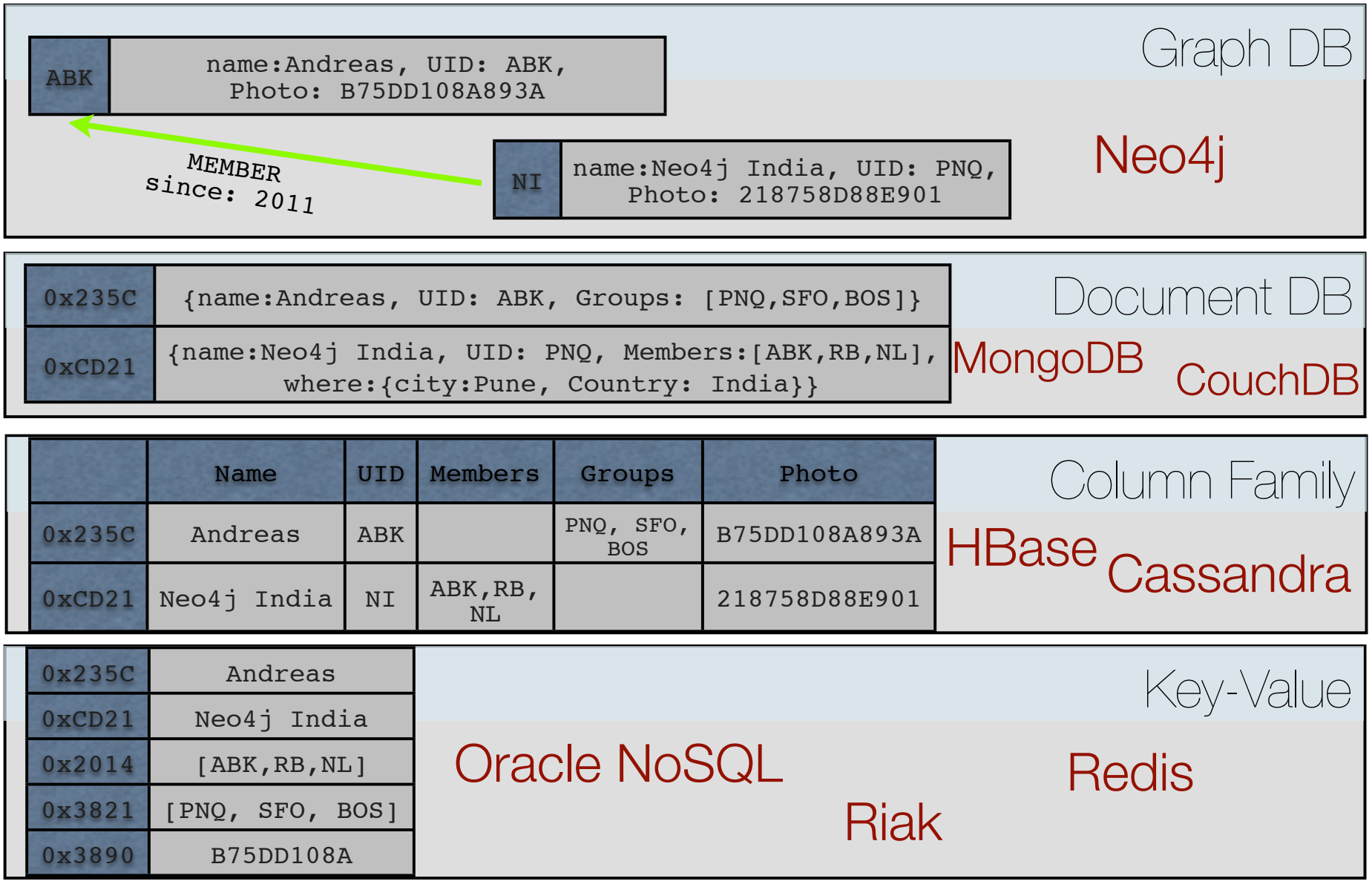
Aggregate oriented

Graph

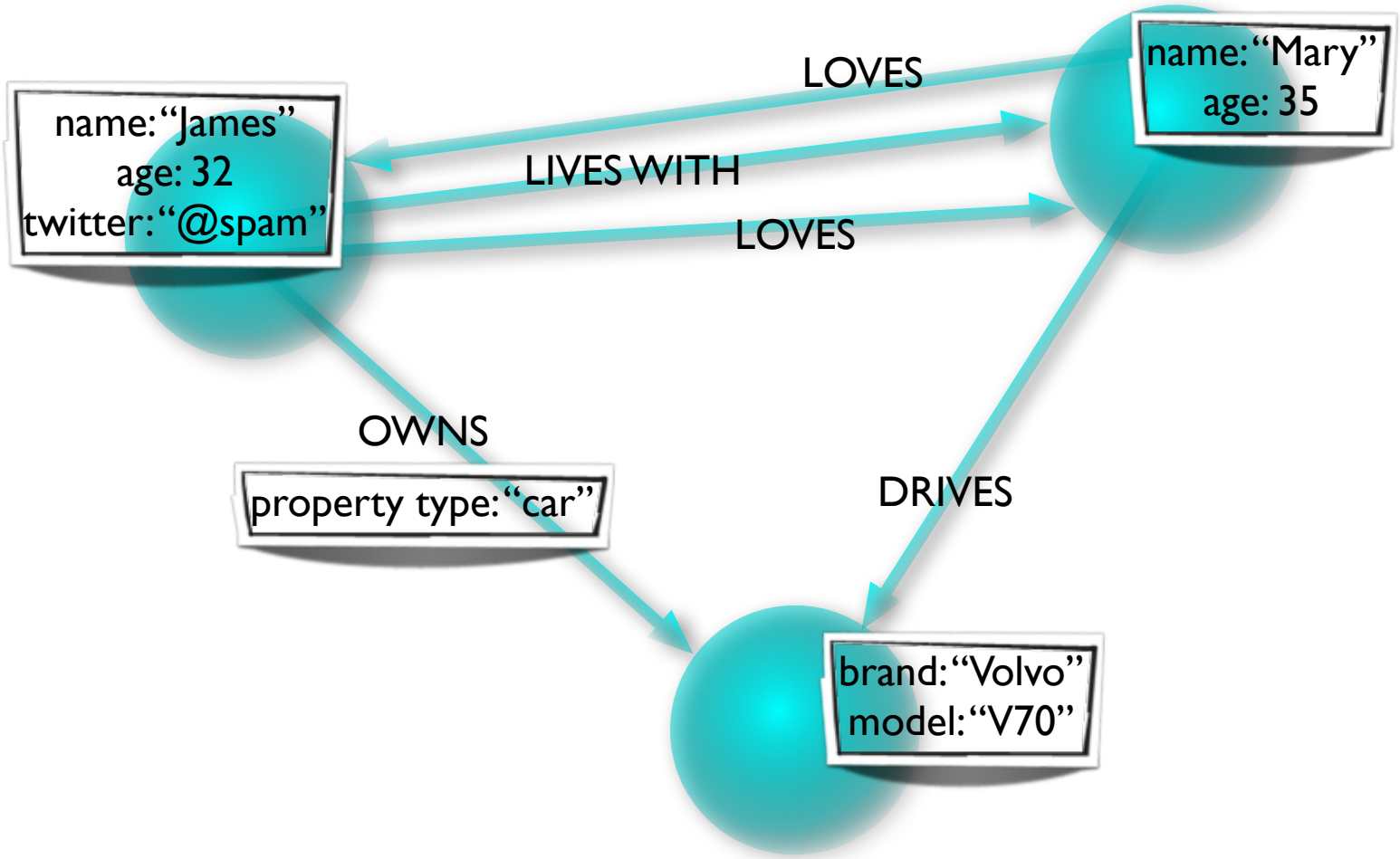


Source: NoSQL Distilled

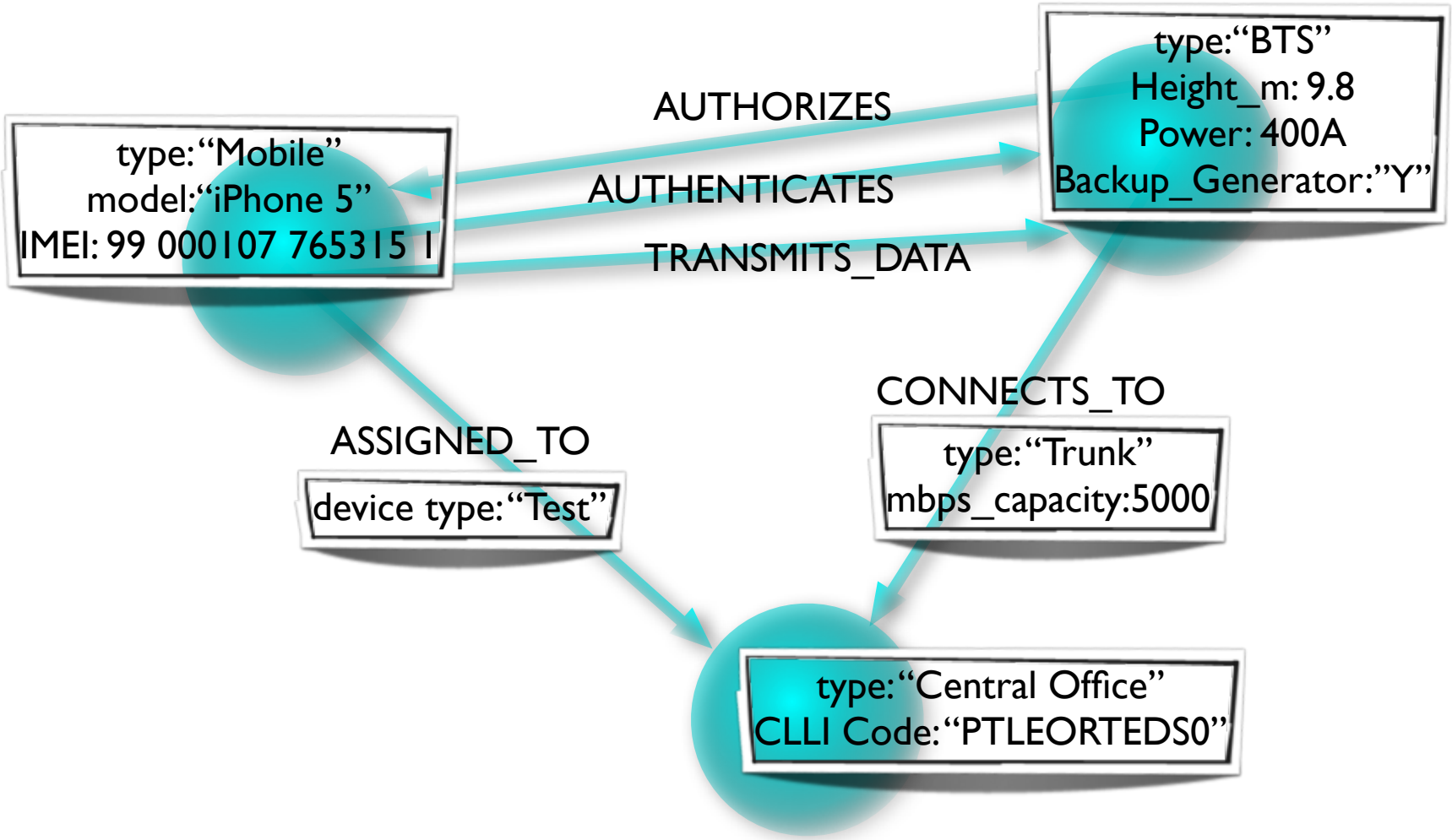
NOSQL - 4 Categories

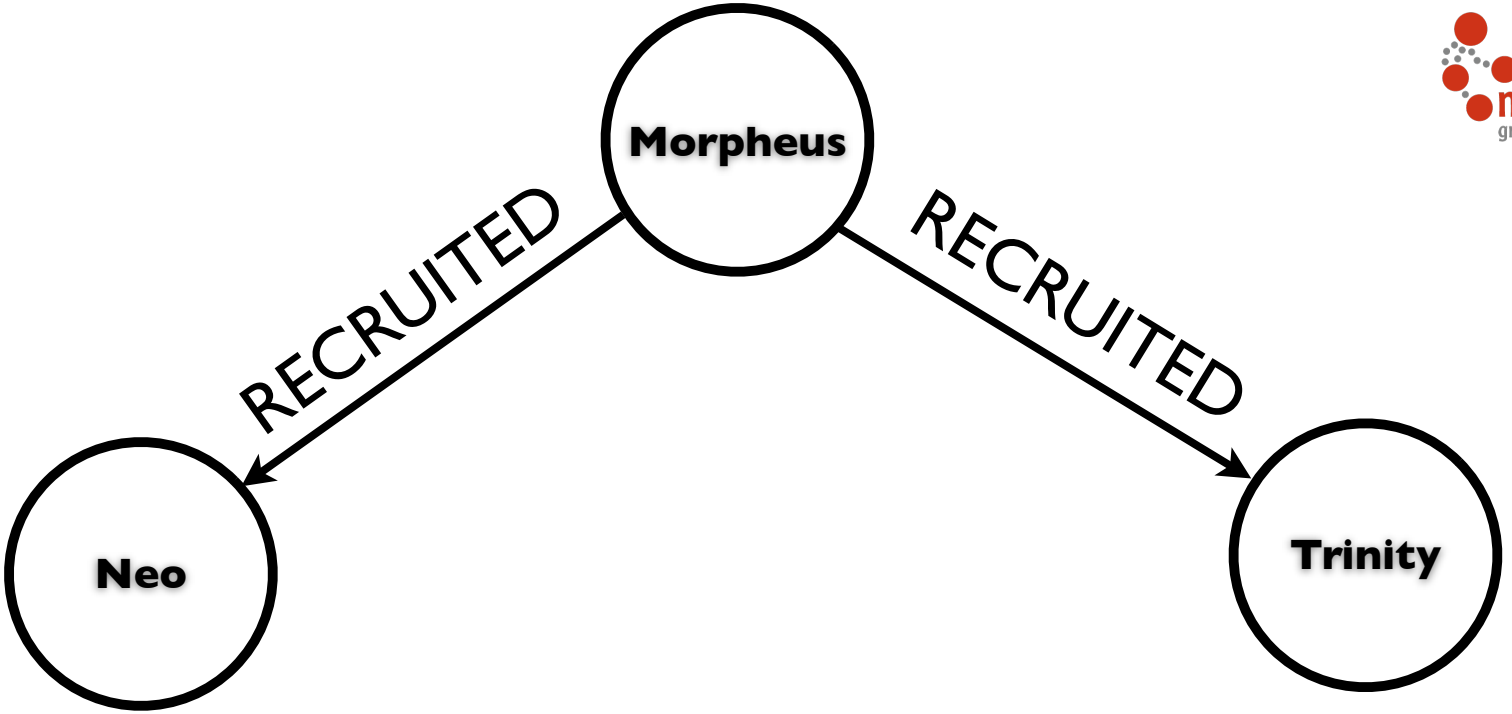


Graph data model



Graph data model





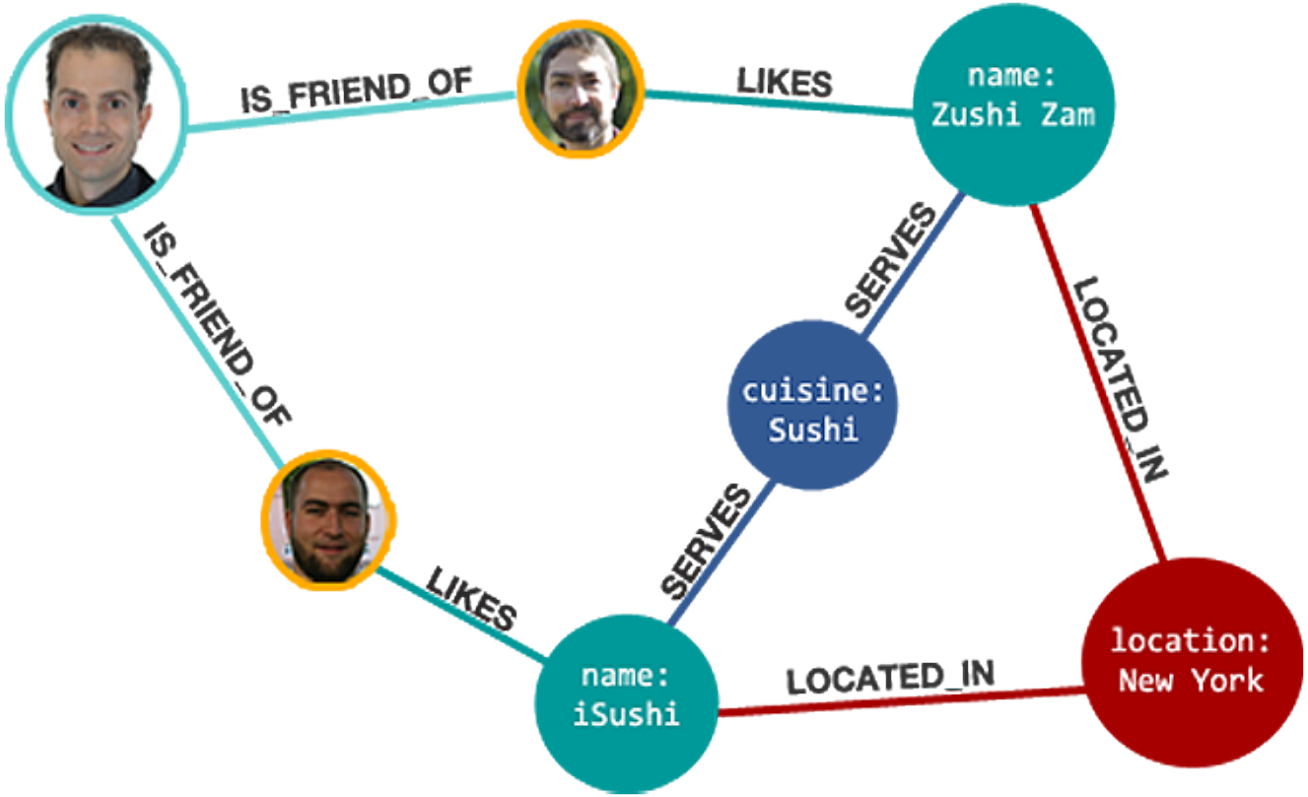

```
MATCH (a) -[:RECRUITED]-> (b)
WHERE a.name = “Morpheus”
RETURN b.name AS recruit
```

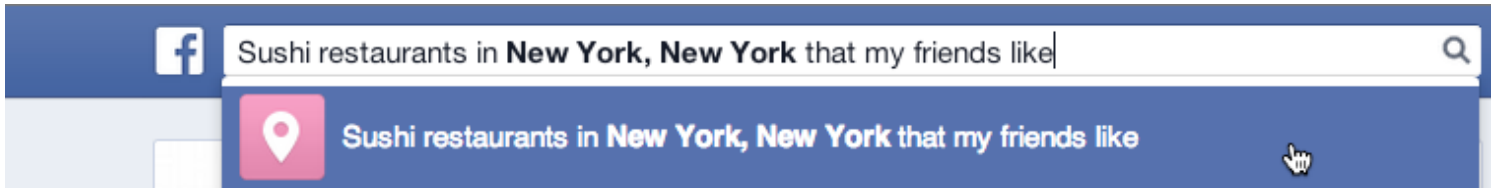
```
+---------+
| recruit |
+---------+
| Neo     |
| Trinity |
+---------+
 2 rows
```

neo technology
graphs are everywhere

Facebook Graph Search Example


Sushi restaurants in **New York, New York** that my friends like
Sushi restaurants in **New York, New York** that my friends like

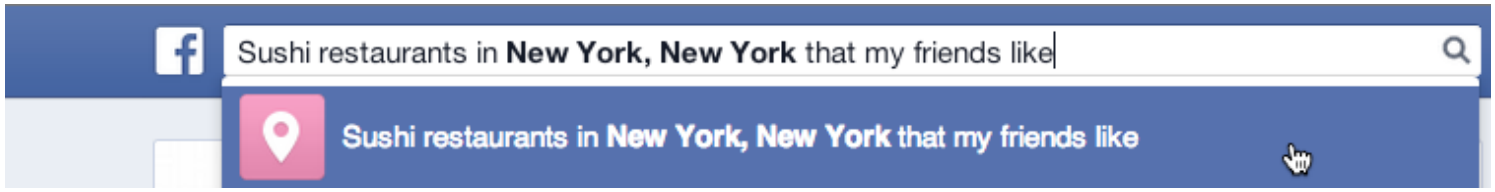

IS_FRIEND_OF
LIKES
name:
Zushi Zam
IS_FRIEND_OF
SERVES
LOCATED_IN
cuisine:
Sushi
SERVES
LIKES
name:
iSushi
LOCATED_IN
location:
New York




```
MATCH (me:Person)-[:IS_FRIEND_OF]->(friend),
      (friend)-[:LIKES]->(restaurant),
      (restaurant)-[:LOCATED_IN]->(city:Location),
      (restaurant)-[:SERVES]->(cuisine:Cuisine)

WHERE me.name = 'Philip' AND city.location='New York' AND
      cuisine.cuisine='Sushi'

RETURN restaurant.name
```

http://maxdemarzi.com/?s=facebook

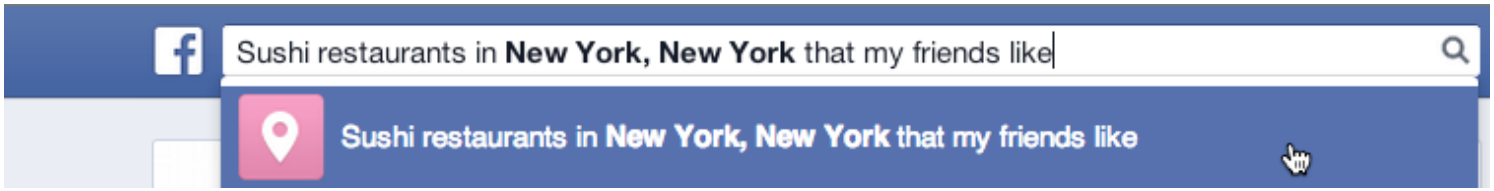
** Cypher query language example*

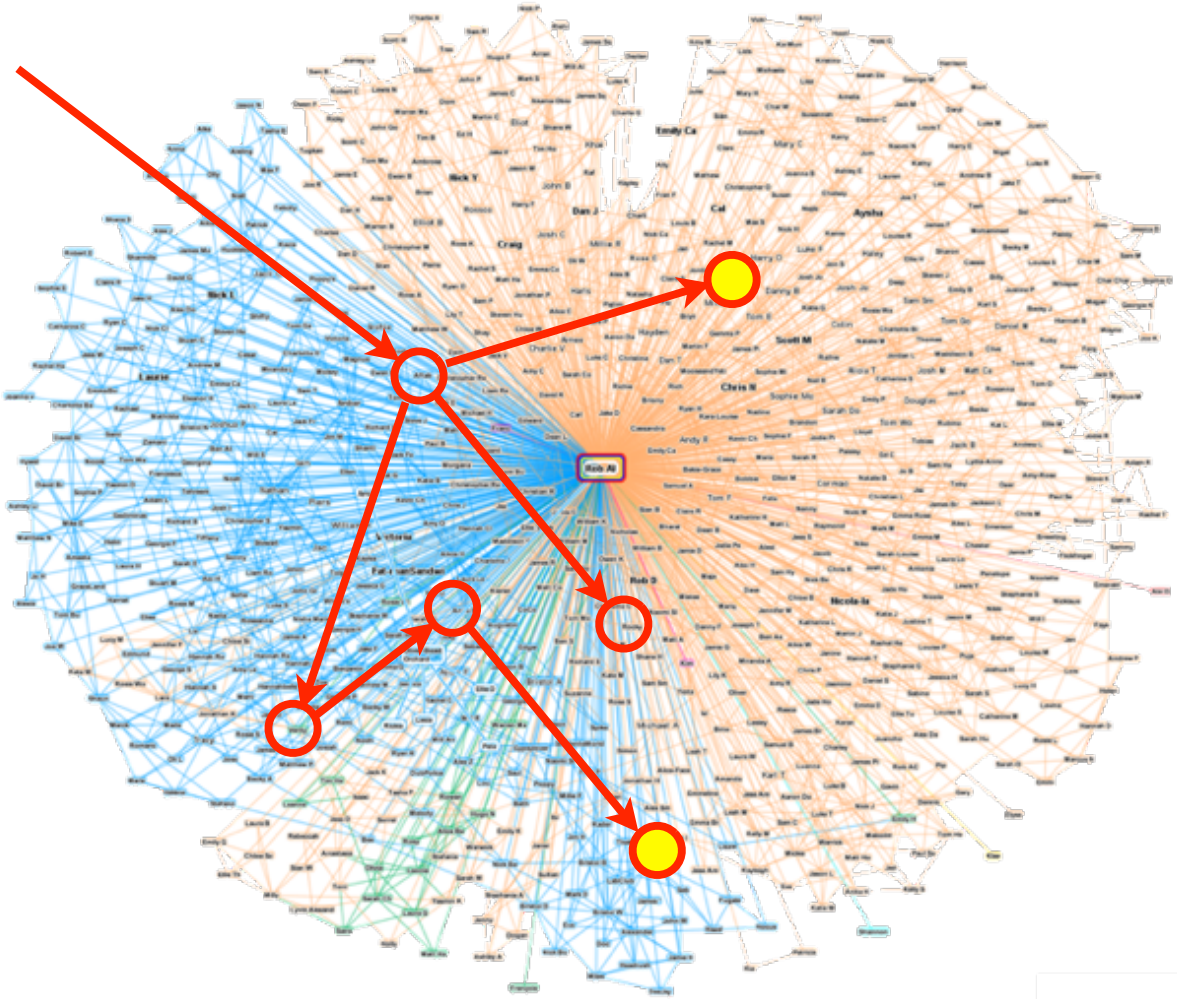



```
MATCH (me:Person)-[:IS_FRIEND_OF]->(friend)-[:LIKES]->(restaurant)-
[:LOCATED_IN]->(city:Location),
(restaurant)-[:SERVES]->(cuisine:Cuisine)

WHERE me.name = 'Philip' AND city.location='New York' AND
      cuisine.cuisine='Sushi'

RETURN restaurant.name
```

http://maxdemarzi.com/?s=facebook

* *Cypher query language example*

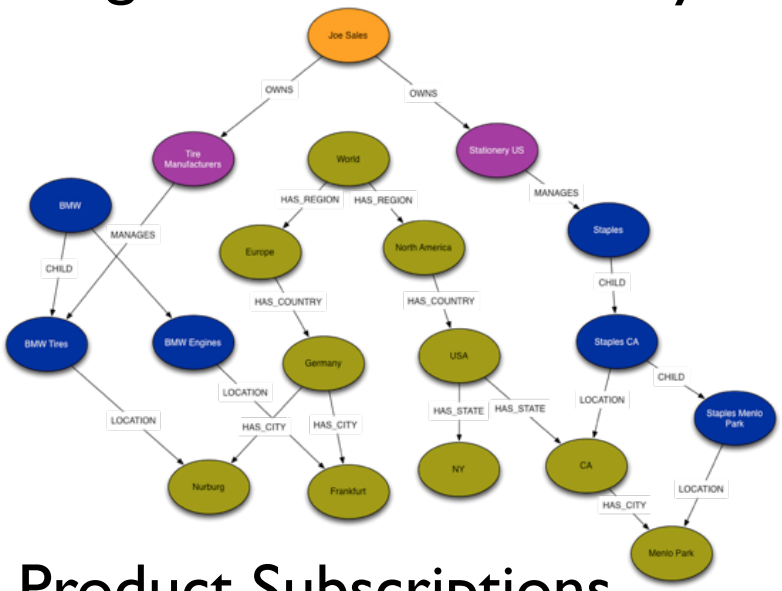




A Few Uses of Graphs in Industry

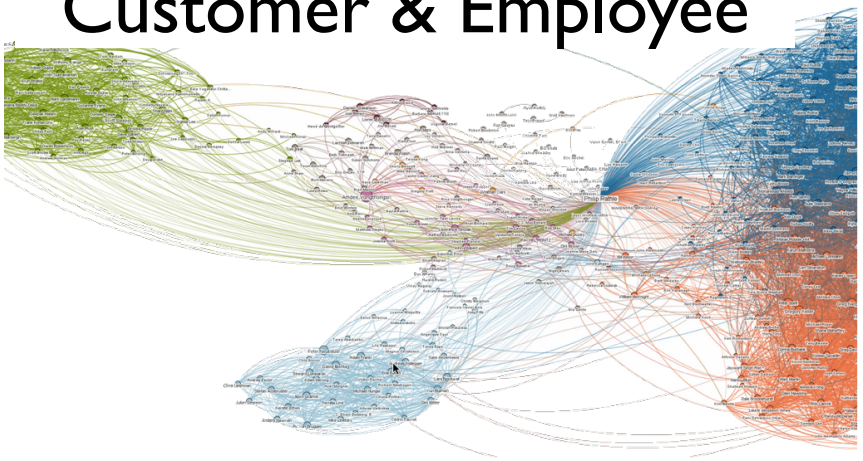
(Actual Neo4j Graphs)

Organizational Hierarchy

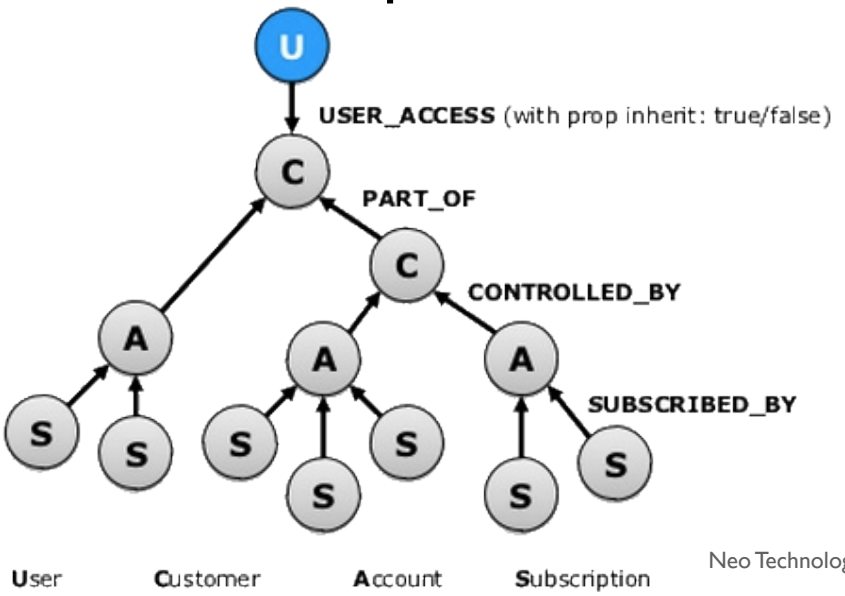


Social Networks

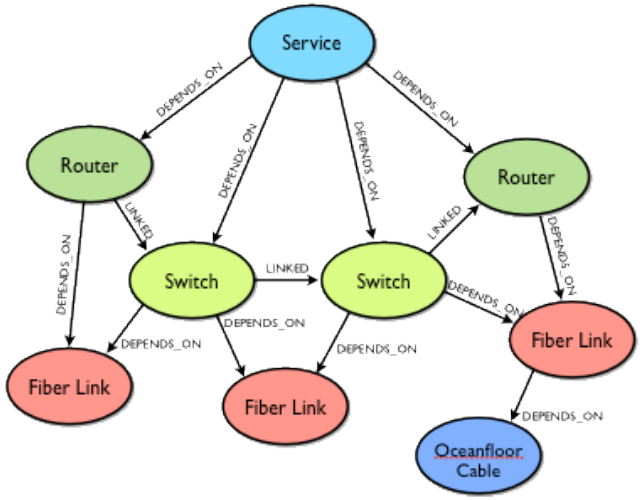
Customer & Employee



Product Subscriptions



CMDB (Network Inventory)



A Few Uses of Graphs in Industry

(Actual Neo4j Graphs)

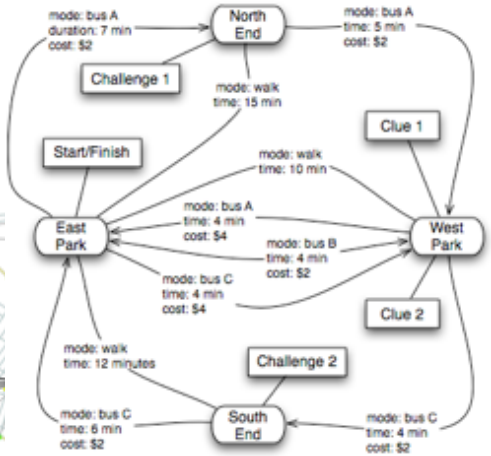
neo technology
graphs are everywhere

Entitlements & Identity Management

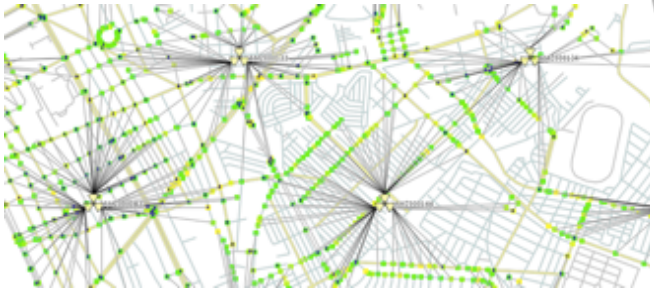
Insurance Risk Analysis

Geo Routing

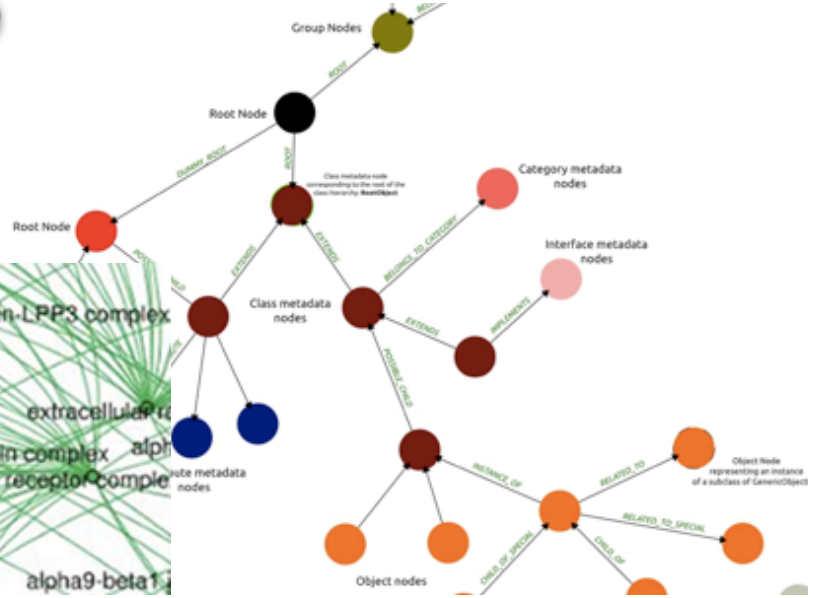
(Public Transport)



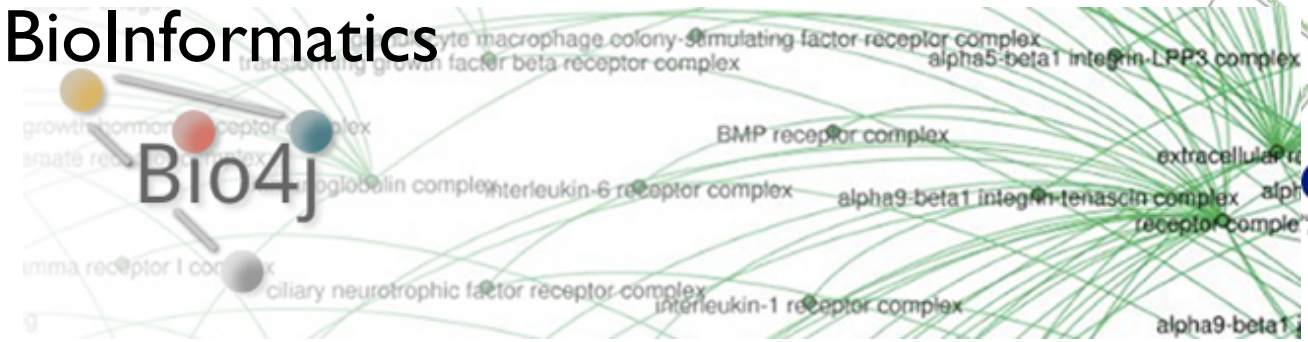
Network Cell Analysis



Network Asset Management



BioInformatics



What is a Graph Database

"A **graph database**... is an online database management system with CRUD methods that expose a graph data model"[1]

- Two important properties:
  - **Native graph storage** engine: *written from the ground up to manage graph data*
  - **Native graph processing**, including *index-free adjacency* to facilitate traversals

1] Robinson, Webber, Eifrem. *Graph Databases.* O'Reilly, 2013. p. 5. ISBN-10: 1449356265

Neo4j: Key Features

1. Property Graph Model
    - Nodes & relationships each comprise a set of key-value pairs
    - In this respect, relationships & nodes have equal status
2. Supports Native Graph Processing
    - Supports "Index-Free Adjacency", meaning that traversals are done through pointer chasing vs. indexed lookups
    - Has its own native graph query language
3. Uses Native Graph Storage
    - Built from ground up to support graph
4. Fully ACID*. All writes use transactions. XA support
    *Tunable consistency across cluster instances (eventual to strong)
5. High Availability Features incl. Clustering & Online Backups

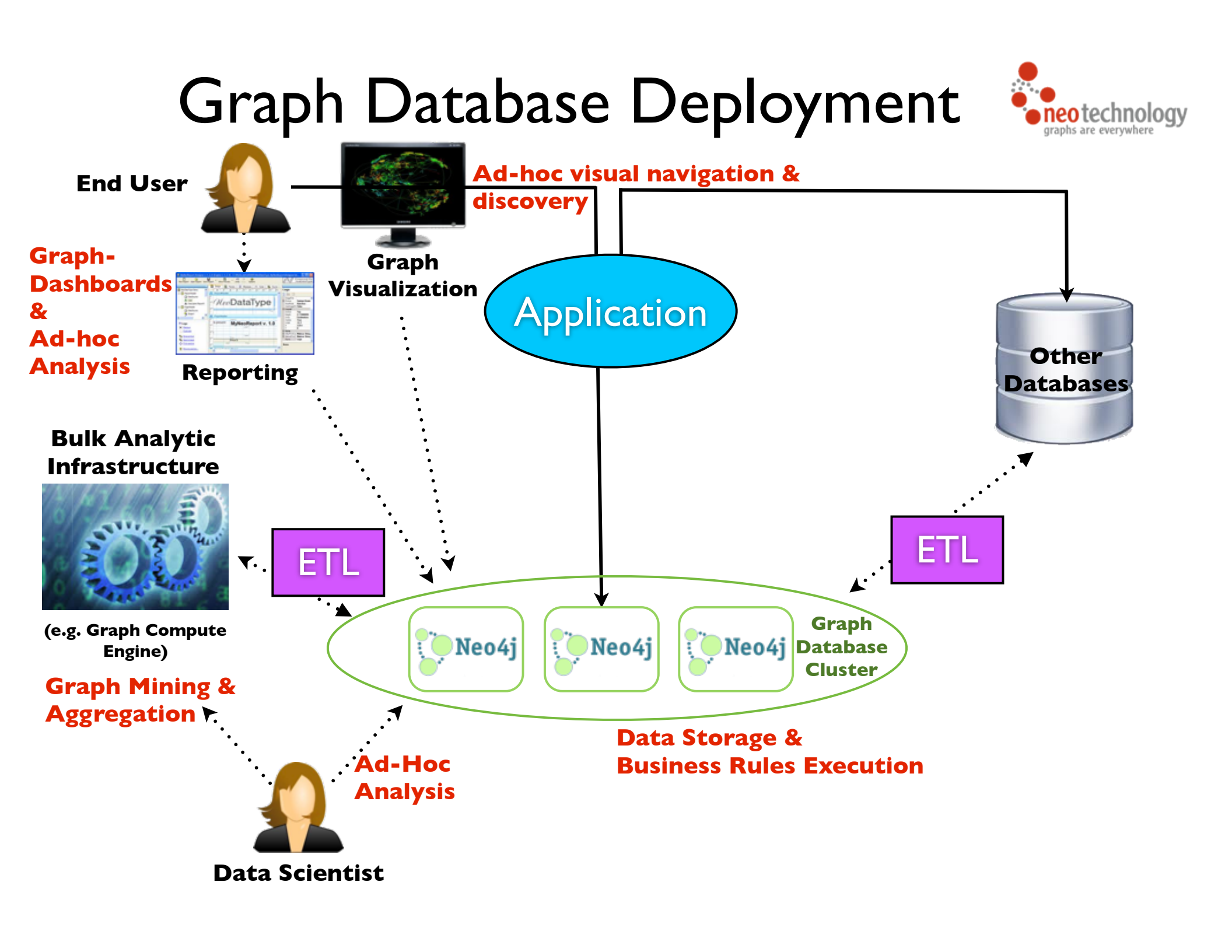
Top Reasons People Use Graph Databases

1. **Lots of Joins.**
2. **Continuously evolving data set** (often involves wide and sparse tables)
3. The **Shape of the Domain** is naturally a graph
4. **Open-ended business requirements** necessitating fast, iterative development.

Graph Databases are Designed to:

1. Store inter-connected data
2. Make it easy to make sense of that data
3. Enable extreme-performance operations for:
    - Discovery of connected data patterns
    - Relatedness queries > depth 1
    - Relatedness queries of arbitrary length
4. Make it easy to evolve the database

Graph Database Deployment



neo technology
graphs are everywhere

Network Management Example

vivendi

SFR

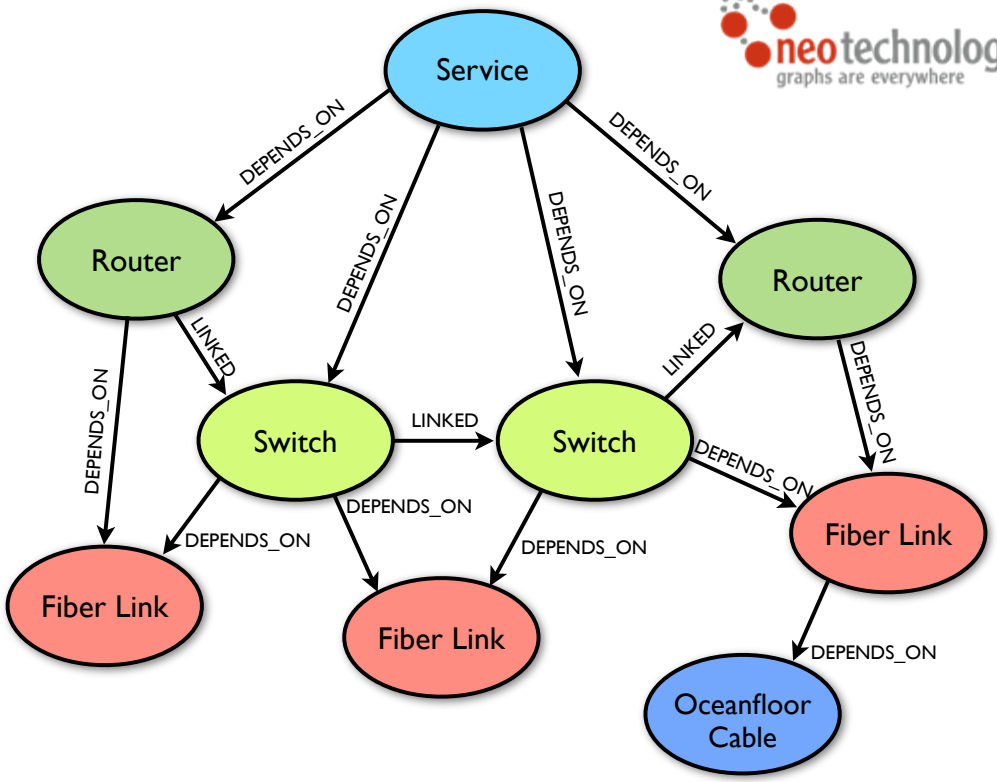
Industry: Communications
Use case: Network Management

Paris, France

neo technology
graphs are everywhere

Background

- Second largest communications company in France
- Part of Vivendi Group, partnering with Vodafone



Business problem

- Infrastructure maintenance took one full week to plan, because of the need to model network impacts
- Needed rapid, automated "what if" analysis to ensure resilience during unplanned network outages
- Identify weaknesses in the network to uncover the need for additional redundancy
- Network information spread across > 30 systems, with daily changes to network infrastructure
- Business needs sometimes changed very rapidly

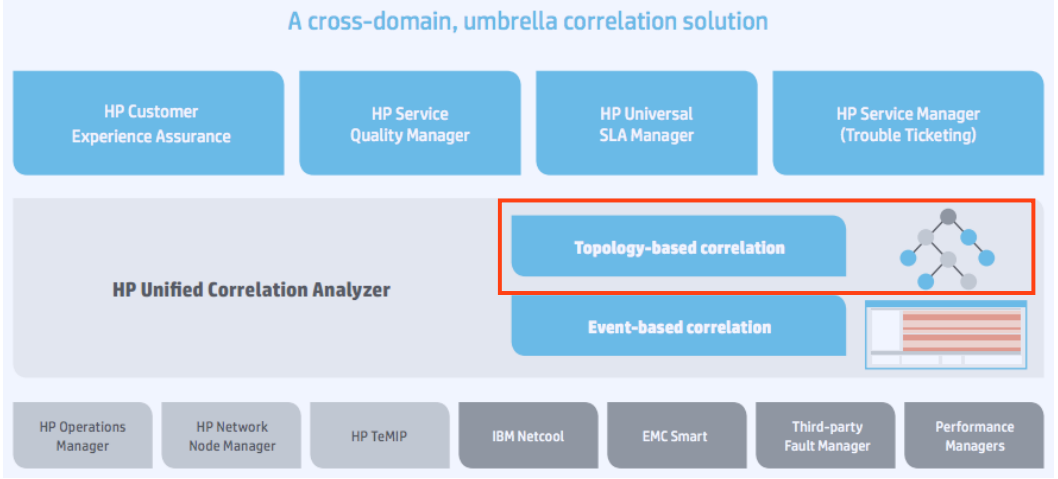
Solution & Benefits

- Flexible network inventory management system, to support modeling, aggregation & troubleshooting
- Single source of truth (Neo4j) representing the entire network
- Dynamic system loads data from 30+ systems, and allows new applications to access network data
- Modeling efforts greatly reduced because of the near 1:1 mapping between the real world and the graph
- Flexible schema highly adaptable to changing business requirements

Industry: Web/ISV, Communications
Use case: Network Management
Global (U.S., France)

Background

- World's largest provider of IT infrastructure, software & services
- HP's *Unified Correlation Analyzer* (UCA) application is a key application inside HP's OSS Assurance portfolio
- Carrier-class resource & service management, problem determination, root cause & service impact analysis
- Helps communications operators manage large, complex and fast changing networks



Business problem

- Use network topology information to identify root problems causes on the network
- Simplify alarm handling by human operators
- Automate handling of certain types of alarms Help operators respond rapidly to network issues
- Filter/group/eliminate redundant Network Management System alarms by event correlation

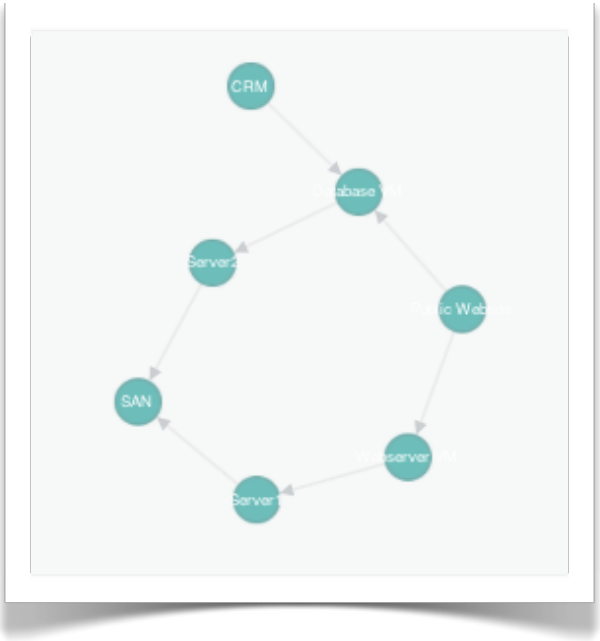
Solution & Benefits

- Accelerated product development time
- Extremely fast querying of network topology
- Graph representation a perfect domain fit
- 24x7 carrier-grade reliability with Neo4j HA clustering
- Met objective in under 6 months

Practical Cypher

Network Management - Create

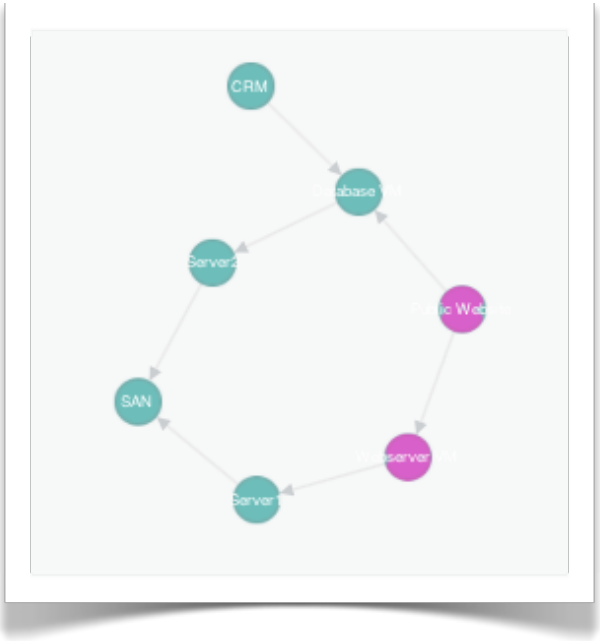
```
CREATE
    (crm {name:"CRM"}),
    (dbvm {name:"Database VM"}),
    (www {name:"Public Website"}),
    (wwwvm {name:"Webserver VM"}),
    (srv1 {name:"Server 1"}),
    (san {name:"SAN"}),
    (srv2 {name:"Server 2"}),

    (crm)-[:DEPENDS_ON]->(dbvm),
    (dbvm)-[:DEPENDS_ON]->(srv2),
    (srv2)-[:DEPENDS_ON]->(san),
    (www)-[:DEPENDS_ON]->(dbvm),
    (www)-[:DEPENDS_ON]->(wwwvm),
    (wwwvm)-[:DEPENDS_ON]->(srv1),
    (srv1)-[:DEPENDS_ON]->(san)
```



Practical Cypher

Network Management - Impact Analysis

```
// Server 1 Outage
MATCH (n)<-[:DEPENDS_ON*]-(upstream)
WHERE n.name = "Server 1"
RETURN upstream
```

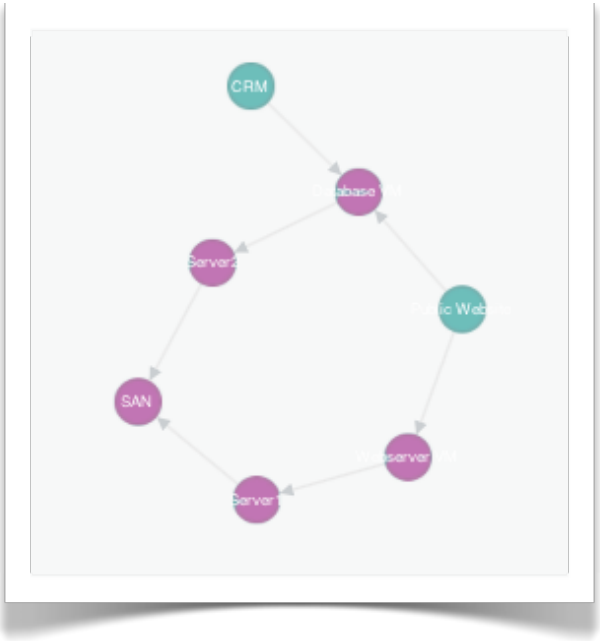


upstream
{name:"Webserver VM"}
{name:"Public Website"}

Practical Cypher

Network Management - Dependency Analysis

```
// Public website dependencies
MATCH (n)-[:DEPENDS_ON*]->(downstream)
WHERE n.name = "Public Website"
RETURN downstream
```

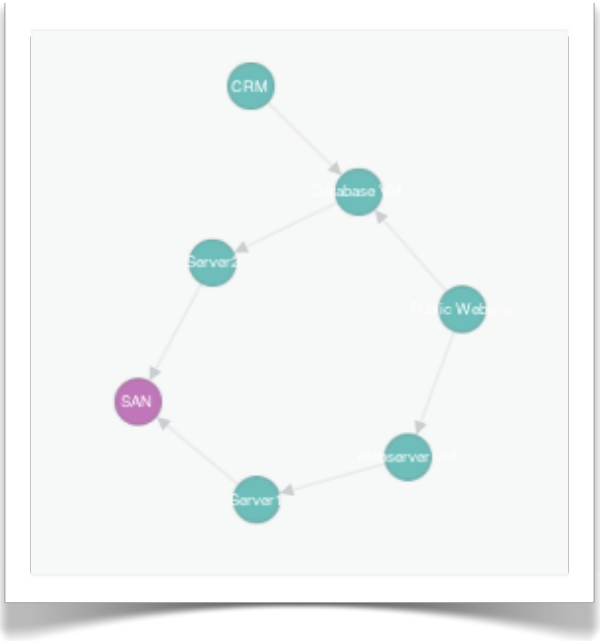


downstream
{name:"Database VM"}
{name:"Server 2"}
{name:"SAN"}
{name:"Webserver VM"}
{name:"Server 1"}

Practical Cypher

Network Management - Statistics

```
// Most depended on component
MATCH (n)<-[:DEPENDS_ON*]-(dependent)
RETURN n,
  count(DISTINCT dependent)
    AS dependents
ORDER BY dependents DESC
LIMIT 1
```



n	dependents
{name:"SAN"}	6

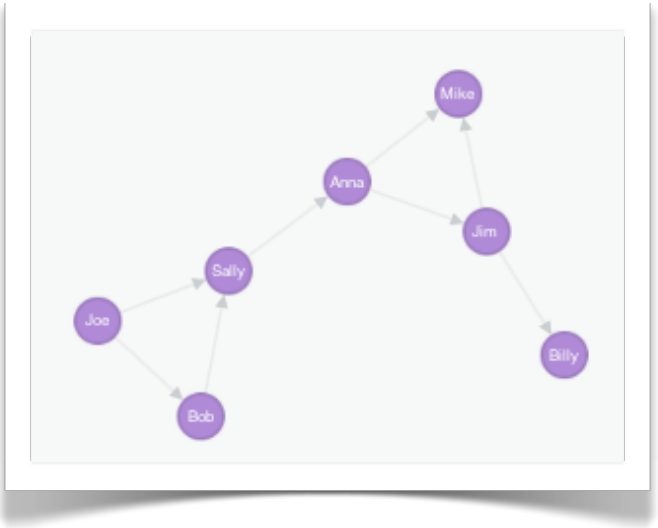
neo technology
graphs are everywhere

Social Example

Practical Cypher

Social Graph - Create

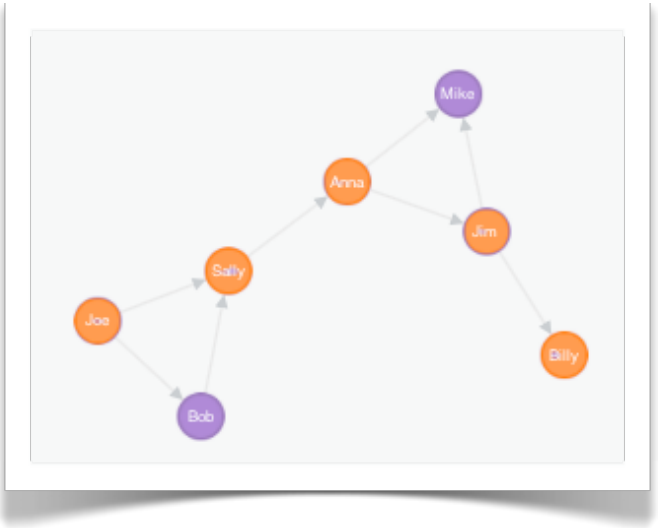
```
CREATE
    (joe:Person {name:"Joe"}),
    (bob:Person {name:"Bob"}),
    (sally:Person {name:"Sally"}),
    (anna:Person {name:"Anna"}),
    (jim:Person {name:"Jim"}),
    (mike:Person {name:"Mike"}),
    (billy:Person {name:"Billy"}),

    (joe)-[:KNOWS]->(bob),
    (joe)-[:KNOWS]->(sally),
    (bob)-[:KNOWS]->(sally),
    (sally)-[:KNOWS]->(anna),
    (anna)-[:KNOWS]->(jim),
    (anna)-[:KNOWS]->(mike),
    (jim)-[:KNOWS]->(mike),
    (jim)-[:KNOWS]->(billy)
```



Practical Cypher

Social Graph - Shortest Path

```
MATCH path = shortestPath(
  (person1)-[:KNOWS*..6]-(person2)
)
WHERE person1.name = "Joe"
   AND person2.name = "Billy"
RETURN path
```

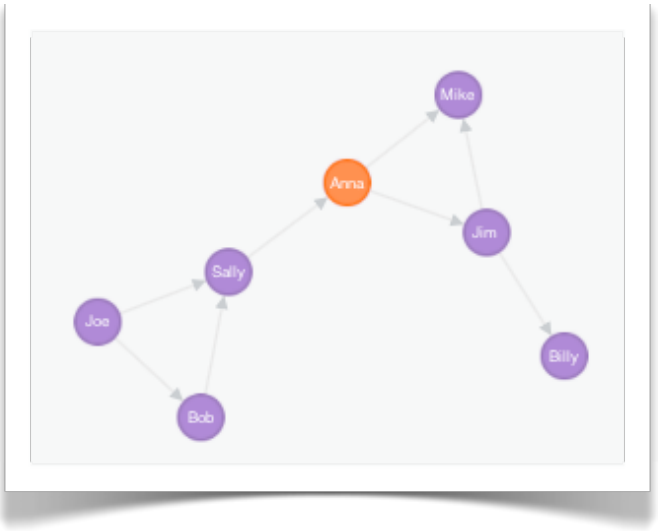


path
{start:"13759", nodes:["13759","13757","13756","13755","13753"], length:4, relationships:["101407","101409","101410","101413"], end:"13753"}

Practical Cypher

Social Graph - Friends of Joe's Friends

```
MATCH (person)-[:KNOWS]-(friend),
      (friend)-[:KNOWS]-(foaf)
WHERE person.name = "Joe"
  AND NOT(person-[:KNOWS]-foaf)
RETURN foaf
```

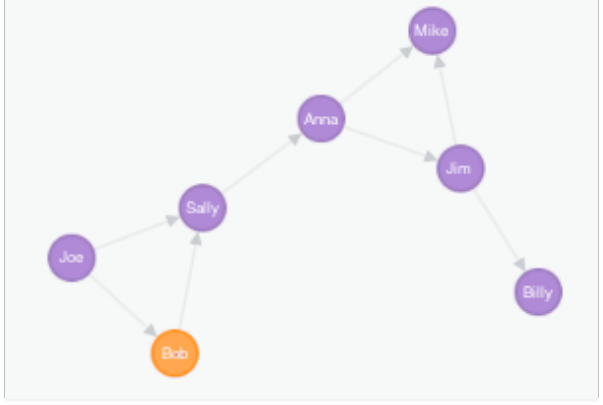


foaf
{name:"Anna"}

Practical Cypher

Social Graph - Common Friends

```
MATCH (person1)-[:KNOWS]-(friend),
  (person2)-[:KNOWS]-(friend)
WHERE person1.name = "Joe"
  AND person2.name = "Sally"
RETURN friend
```



friend
{name:"Bob"}

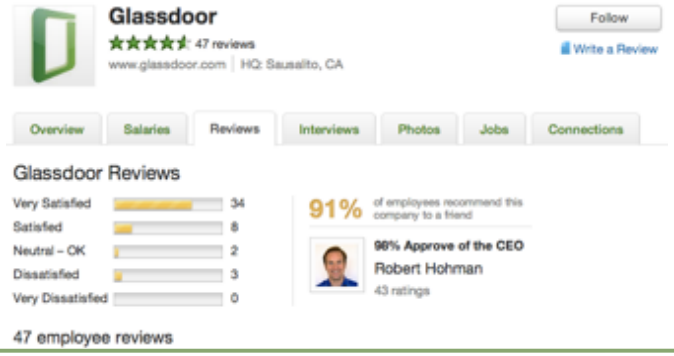
glassdoor.com

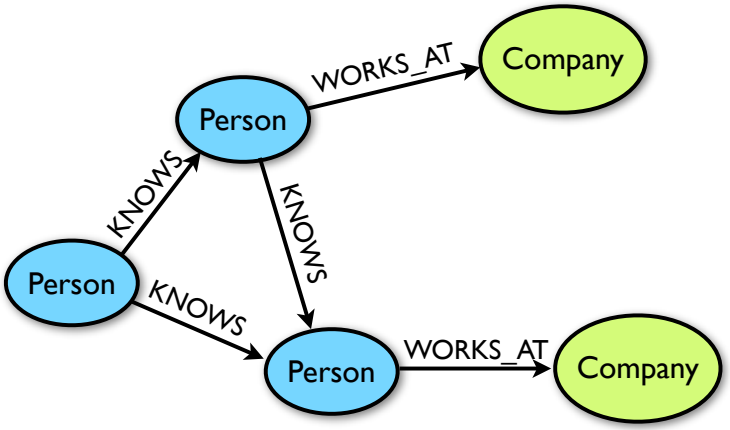
Industry: Online Job Search
Use case: Social / Recommendations
Sausalito, CA

neo technology
graphs are everywhere

Background

- Online jobs and career community, providing anonymized inside information to job seekers





Business problem

- Wanted to leverage known fact that most jobs are found through personal & professional connections
- Needed to rely on an existing source of social network data. Facebook was the ideal choice.
- End users needed to get instant gratification
- Aiming to have the best job search service, in a very competitive market

Solution & Benefits

- First-to-market with a product that let users find jobs through their network of Facebook friends
- Job recommendations served real-time from Neo4j
- Individual Facebook graphs imported real-time into Neo4j
- Glassdoor now stores > 50% of the entire Facebook social graph
- Neo4j cluster has grown seamlessly, with new instances being brought online as graph size and load have increased

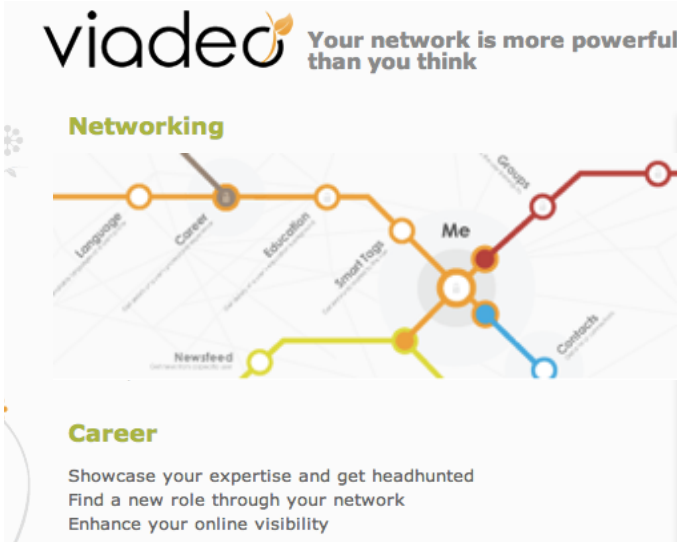
viadeo

Industry: Professional Social Network
Use case: Social, Recommendations
Silicon Valley & France

neo technology
graphs are everywhere

Background

- World's second-largest professional network (after LinkedIn)
- 50M members. 30K+ new members daily.
- Over 400 staff with offices in 12 countries



Business problem

- Business imperative for real-time recommendations: to attract new users and retain existing ones
- Key differentiator: show members how they are connected to any other member
- Real-time traversals of social graph not feasible with MySQL cluster. Batch precompute meant stale data.
- Process taking longer & longer: > 1 week!

Solution & Benefits

- Neo4j solution implemented in 8 weeks with 3 part-time programmers
- Able to move from batch to real-time: improved responsiveness with up-to-date data.
- Viadeo (at the time) had 8M members and 35M relationships.
- Neo4j cluster now sits at the heart of Viadeo's professional network, connecting 50M+ professionals

Recommended Reading & Next Steps for Learning About Graphs...

*Use Promo Code **NOCOUG** for 50% Off!!*

GRAPH CONNECT

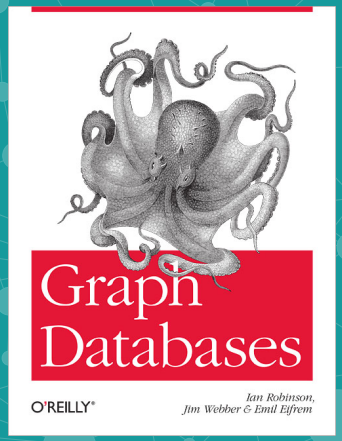
Innovate. Share. Connect.

San Francisco

October 3 - 4

www.graphconnect.com

(graphs)-[:ARE]->(everywhere)



Get the free ebook!
www.graphdatabases.com

www.neo4j.org