

Neo4

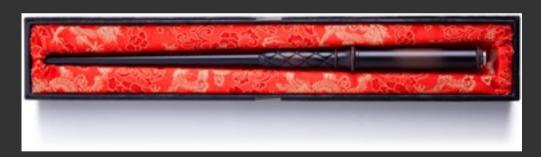
And some graph problems

Peter Neubauer COO, Neo Technology

#neo4j @peterneubauer peter@neotechnology.com



Any sufficiently advanced technology is indistinguishable from magic. Arthur C. Clarke



@kymerawand



First, a recap: NOSQL data models

Key-value stores Data size Column Family Document databases Graph databases We are here

Data complexity



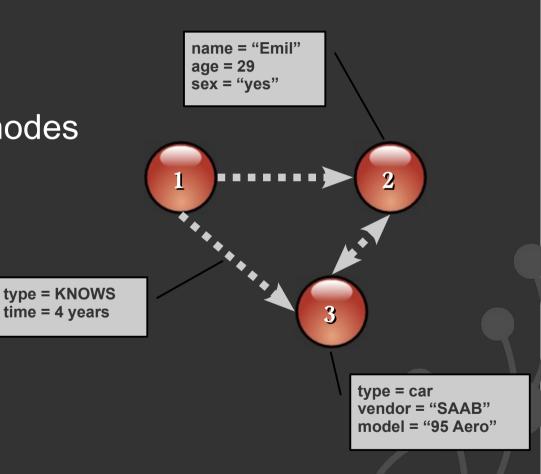
What makes a good DB?

- Separate Logic and Data
- Support Ad-Hoc queries
- Be persistent
- Scale good enough



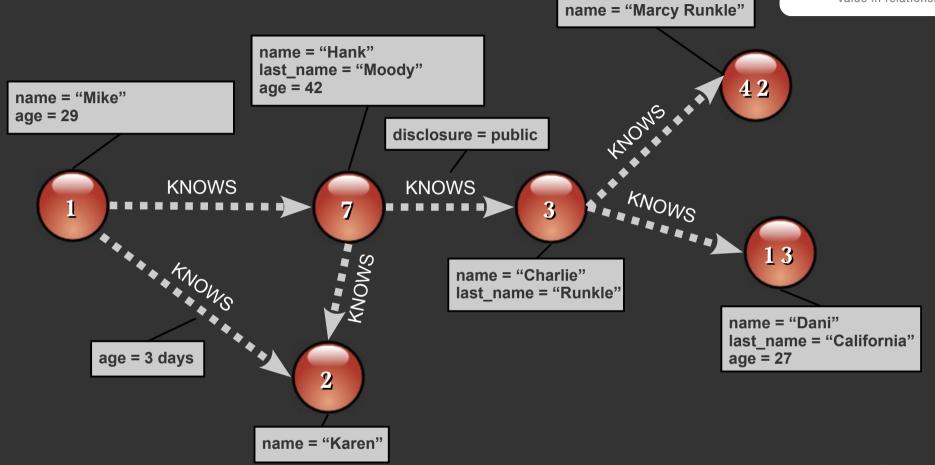
The property graph model

- Ore abstractions:
 - Nodes
 - Relationships between nodes
 - Properties on both
- Traversal framework
 - High performance queries on connected data sets
- Other bindings
 - Langs, REST, Gremlin, RDF



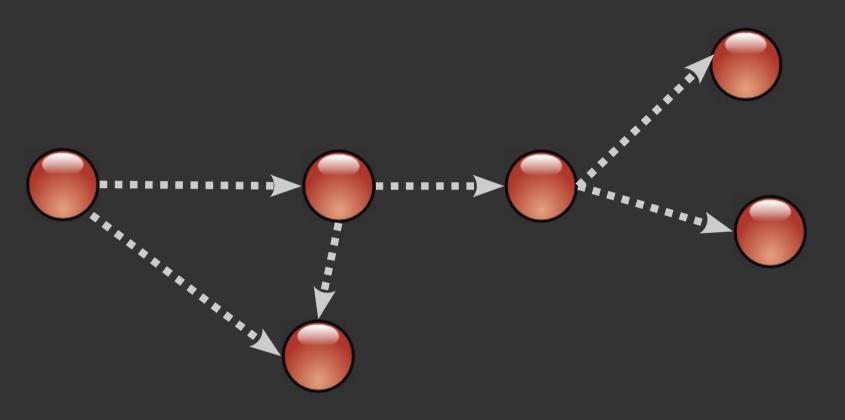
Social data (customer: brand-name social network)





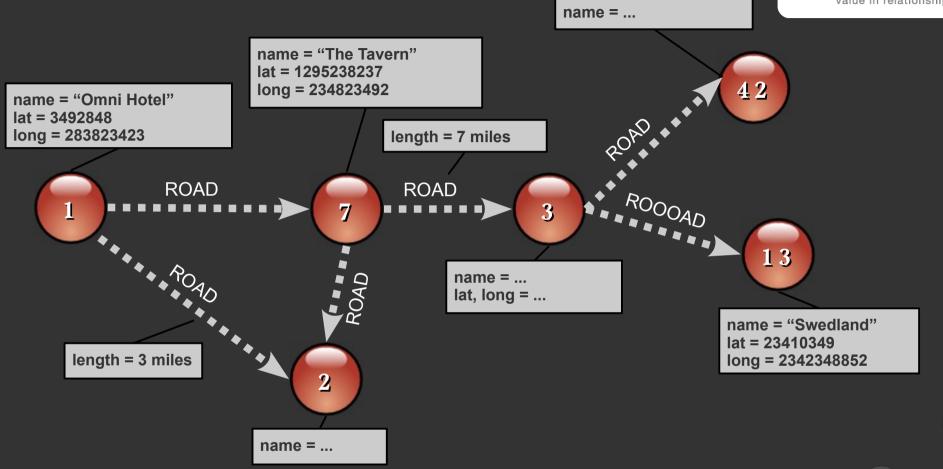
Just a social graph?





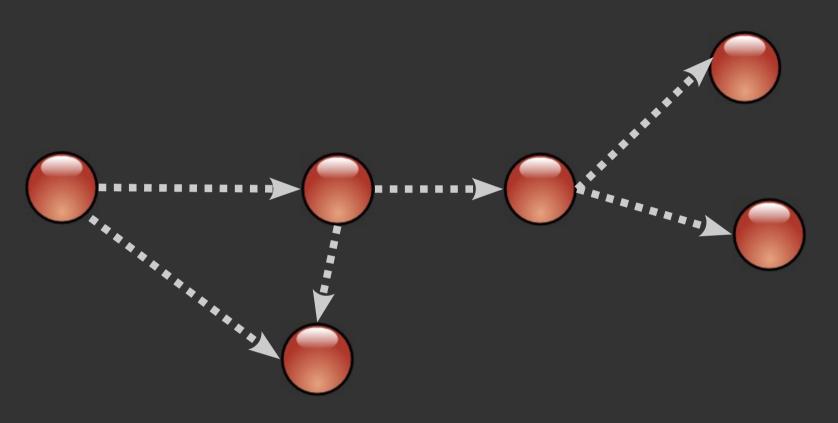
Spatial data (customer: large telecom company)





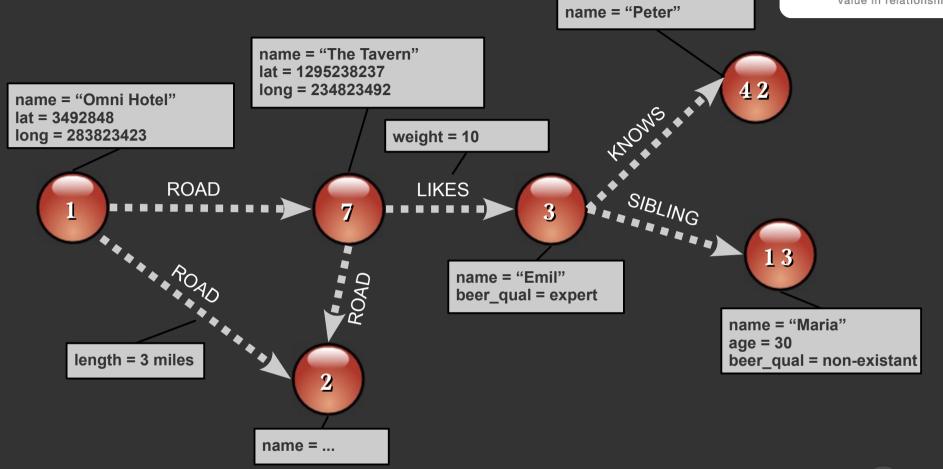
Social? Spatial? ... Social AND spatial!





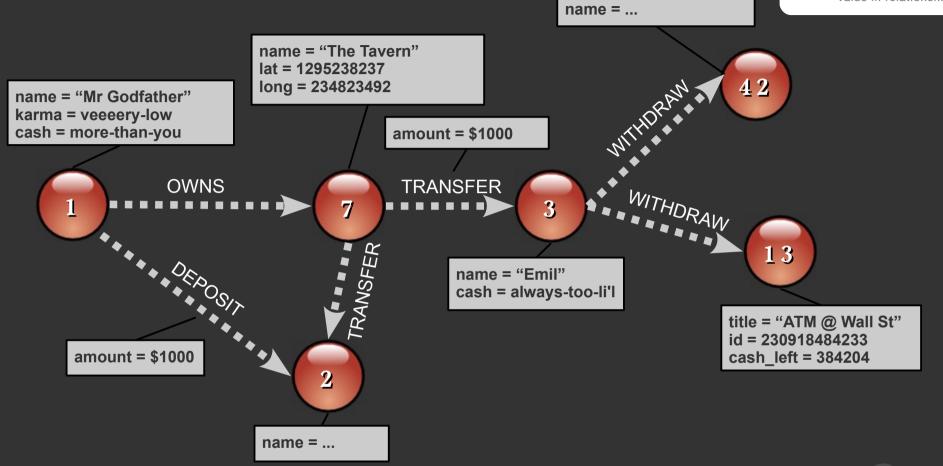
Social AND spatial data (customer: LBS)





Financial data (customer: international bank)





Use Cases



- REST in Neo4j
- The graph as an OODB − Roo, Grails Jruby, Jo4neo
- Deep graph algos Routing with A*
- Multiple indexes in the graph GIS
- Recommendation systems Gremlin and LinkedData



REST in Neo4j

- Build on self describing JSON
- Indexing
- Querying
- Bindings to PHP, C#, JS
- OpenGraph API
- Jruby, Scala

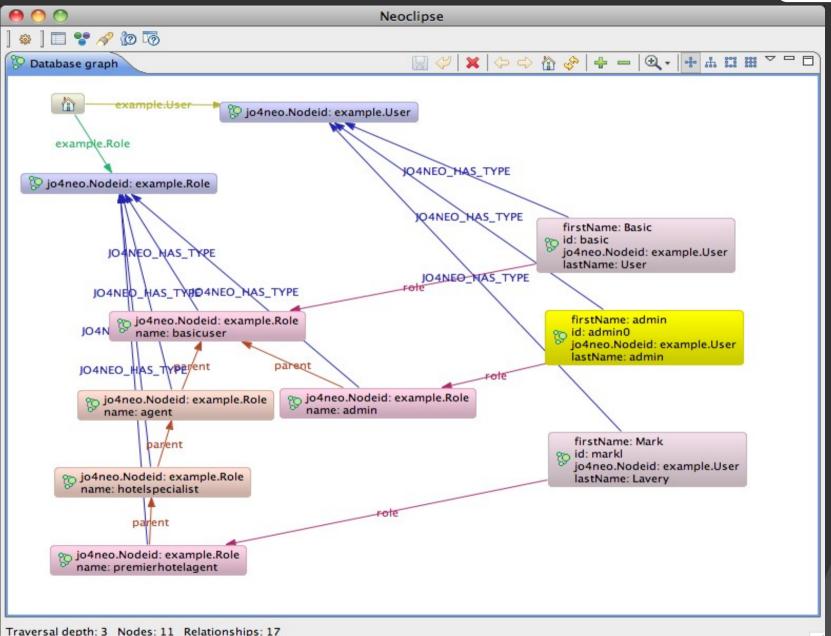
The Graph for objects



- Keep data clean!
- Annotation based
 - Grails
 - Jo4neo
 - Roo
 - JPA
- Mixin based
 - Django
 - JRuby
- © C# (REST)
- PHP (REST)

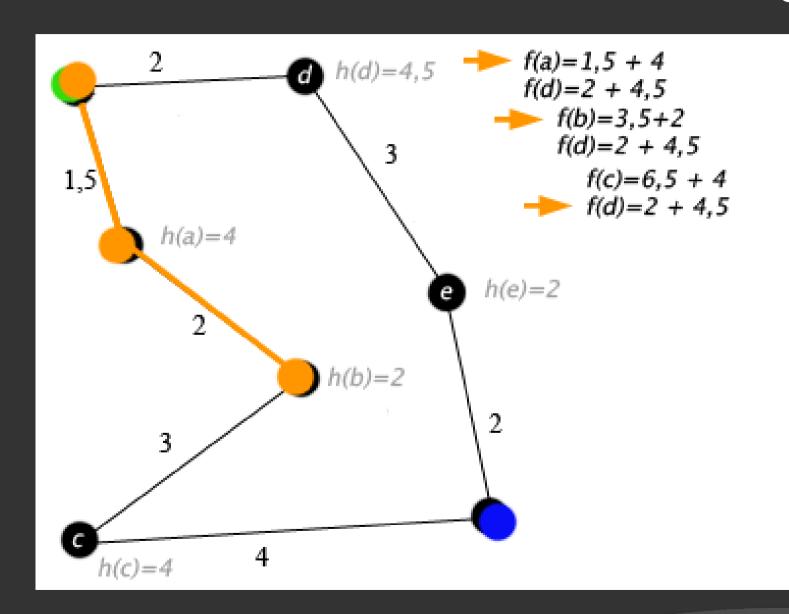
The Graph as an OODB





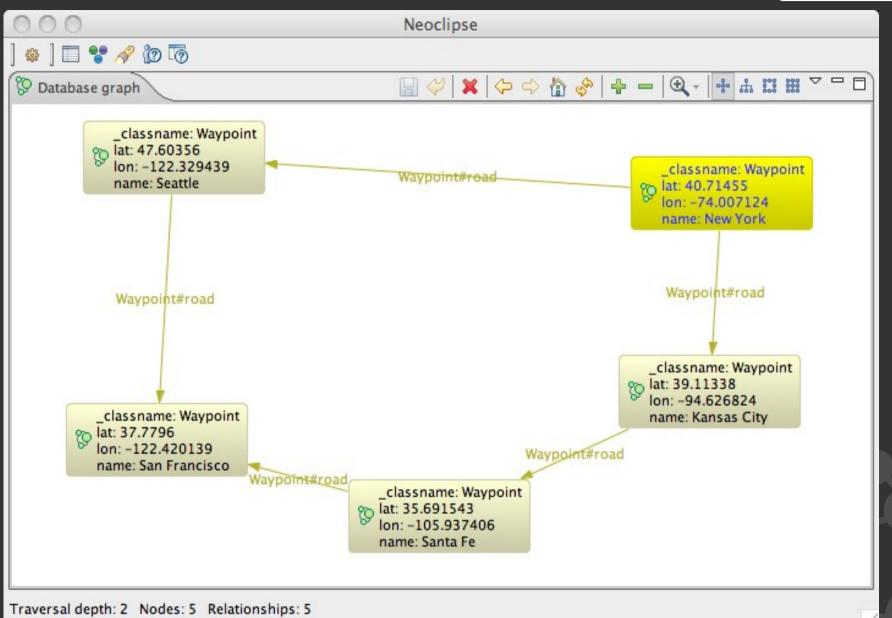
Routing with Neo4j and A*





Routing with Neo4j and A*





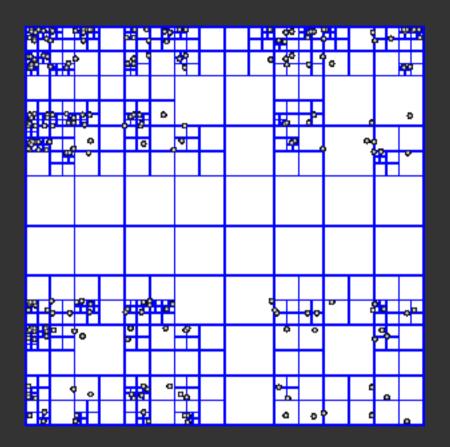
Multiple indexes - GIS

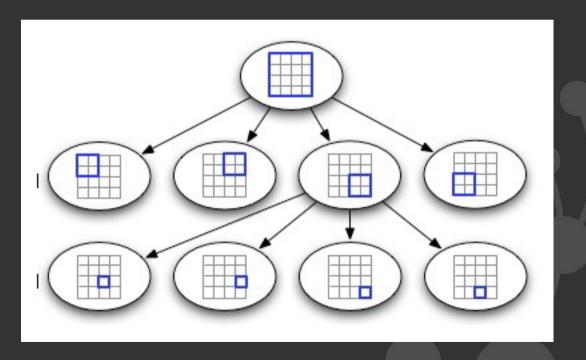
neotechnology
value in relationships

- QuadTree (2D)
- Hierarchy
- Time (1D)
- On-demand index

Multiple indexes - GIS

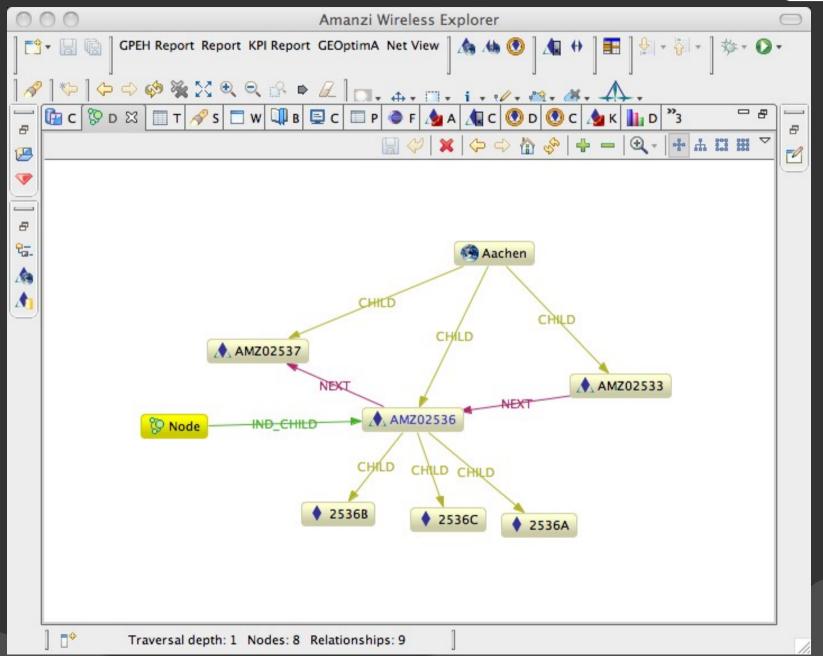






Multiple indexes - GIS

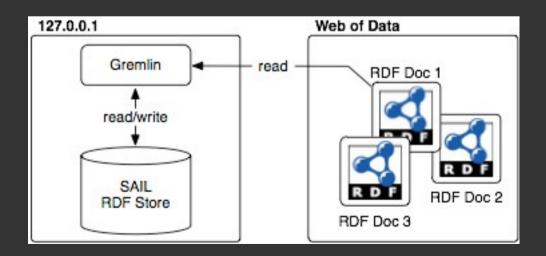






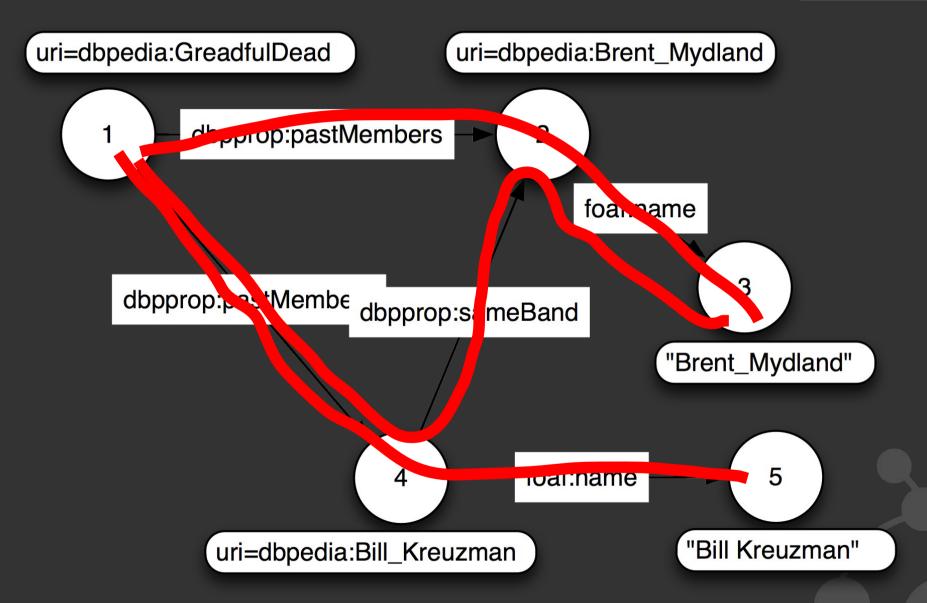
Recommendation - Gremlin

- gremlin.tinkerpop.com
- XPath based "Perl for Graphs"
- Property Graph Model
 - Neo4j, RDF-SAIL, Tinkergraph, MongoDB, FS
- LinkedData SAIL



Recommendation - Gremlin





Recommendation - Gremlin



```
algo.grml
$artist_uri := 'http://dbpedia.org/resource/Grateful_Dead'-
$_ := g:id($artist_uri)-
$e := 1.0¬
m := g:map() -
repeat 20-
  $_ := (./outE[@label=$labels]/inV)[g:rand-nat()][g:print(.)]-
  if count(\$_) = 1¬
    $name := (./outE[@label=sail:ns('foaf:name')]/inV/@value)[1]-
    if count(\frac{1}{\sqrt{2}}
      m[@name=$name] := m[@name=$name] + $e^-
    end⊸
    else-
      $_ := g:id($artist_uri)-
    end⊣
  $e := $e * 0.85-
end⊣
g:sort($m,'value',true())-
 20 Column: 30 Plain Text
                          ‡ ③ ▼ Tab Size: 4 ‡
```

Other use cases



- Activity streams
- Process automation
 - Life-filtering of event streams
 - Web-of-Things data f ows
- Object oriented reporting
 - Jasper Reports
 - ReportAnywhere
 - Scripting

Why graph databases?





A web where the default is social → an explosion of graph-y data

"Just 3 years ago, the data sets our product handled were very simple. But the past 9-12 months have seen an explosion of the number of relationships between things."

"For anything with multiple relationships, multiple connections Neo4 j absolutely ROCKS!"

http://bit.ly/dys4nx



Manager of an enterprise product for the Fortune 5000



Werner Vogels CTO, Amazon



How ego are you? (aka other impls?)

- Franz' AllegroGraph (http://agraph.franz.com)
 - Proprietary, Lisp, RDF-oriented but real graphdb
- Sones graphDB (http://sones.com)
 - Proprietary, .NET, in beta
- Twitter's FlockDB (http://github.com/twitter/flockdb)
 - Twitter's (graph) database for large and shallow graphs
- Google Pregel (http://bit.ly/dP9IP)
 - We are oh-so-secret
- Objectivity's InfiniteGraph (http://infinitegraph.com)
 - New, closed OODB with Graph Layer on top

Why Neo4j?



- Most widely deployed graph db in the world
 - Robust: 24/7 production since 2003
 - Mature: lots of production deployments
 - Community: ecosystem of tools, bindings, frameworks
- Available NOW under AGPLv3 / commercial license
 - AGPLv3: "if you're open source, we're open source"
 - If you have proprietary software? Must buy a commercial license
 - But the f rst one is free! For ALL use-cases.
- Download
 - http://neo4j.org
- Feedback
 - http://lists.neo4j.org





Questions?



Image credit: lost again! Sorry :(

