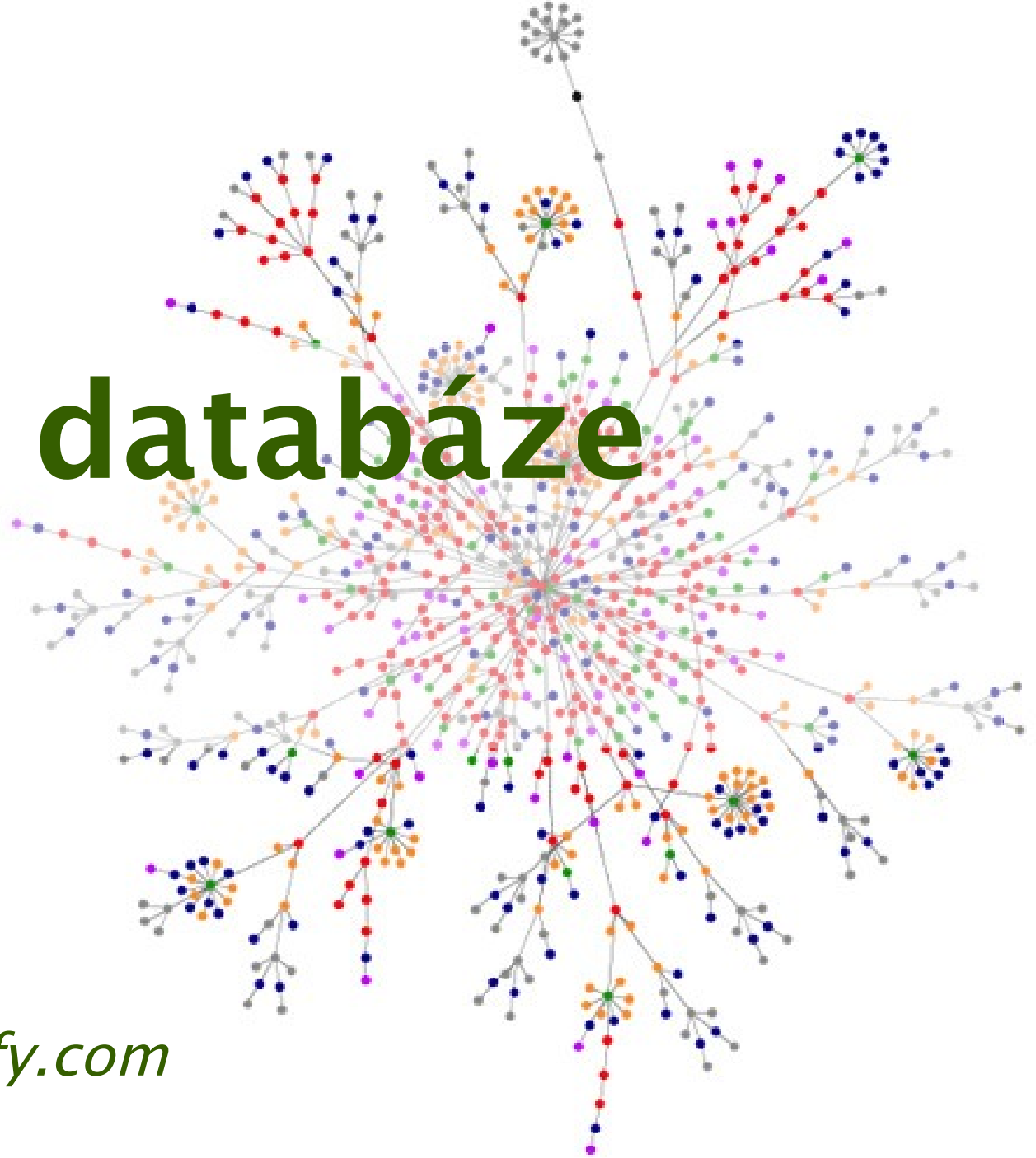


Grafová databáze neo4j

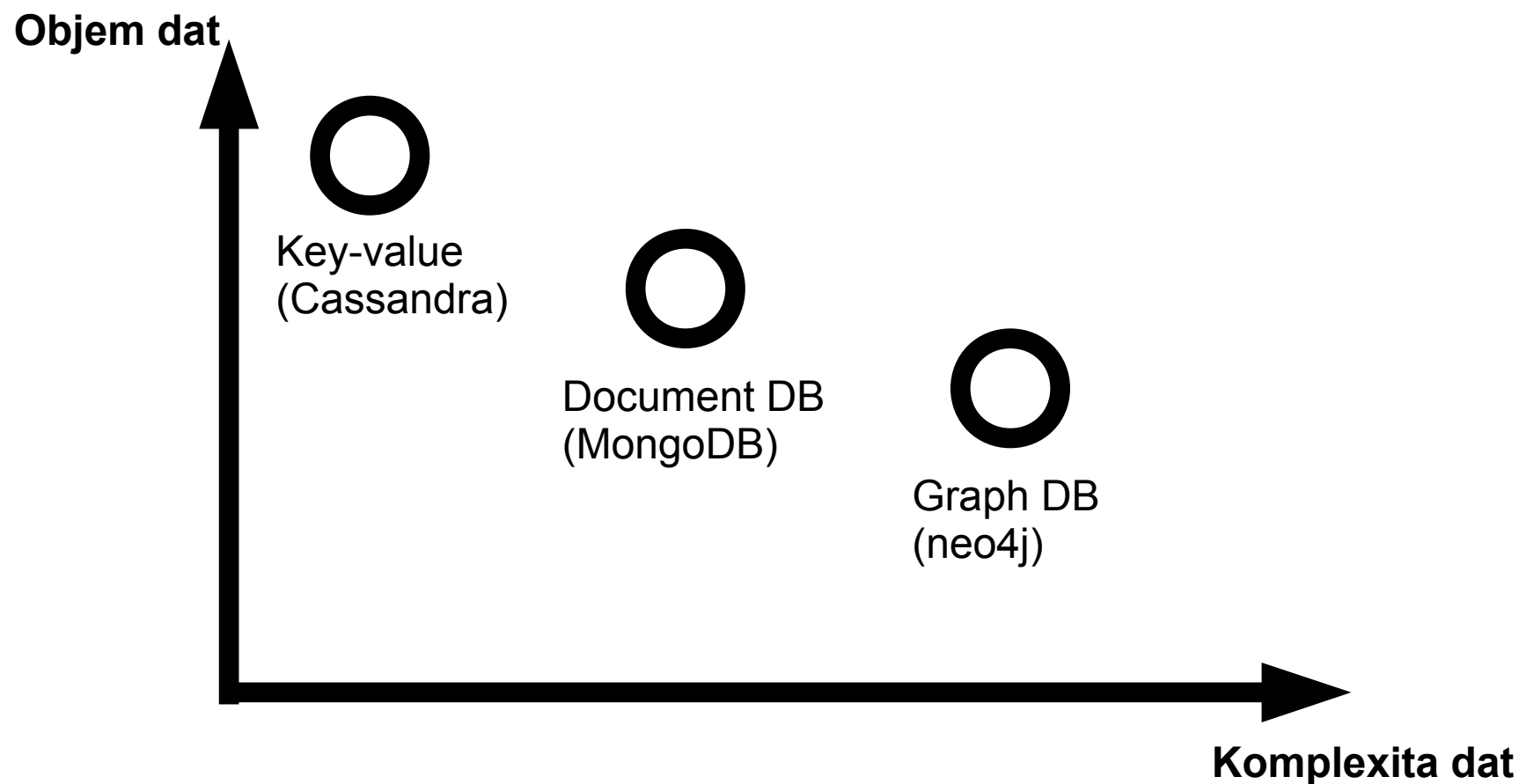
Josef Holý

@proactify

Josef.Holy@proactify.com

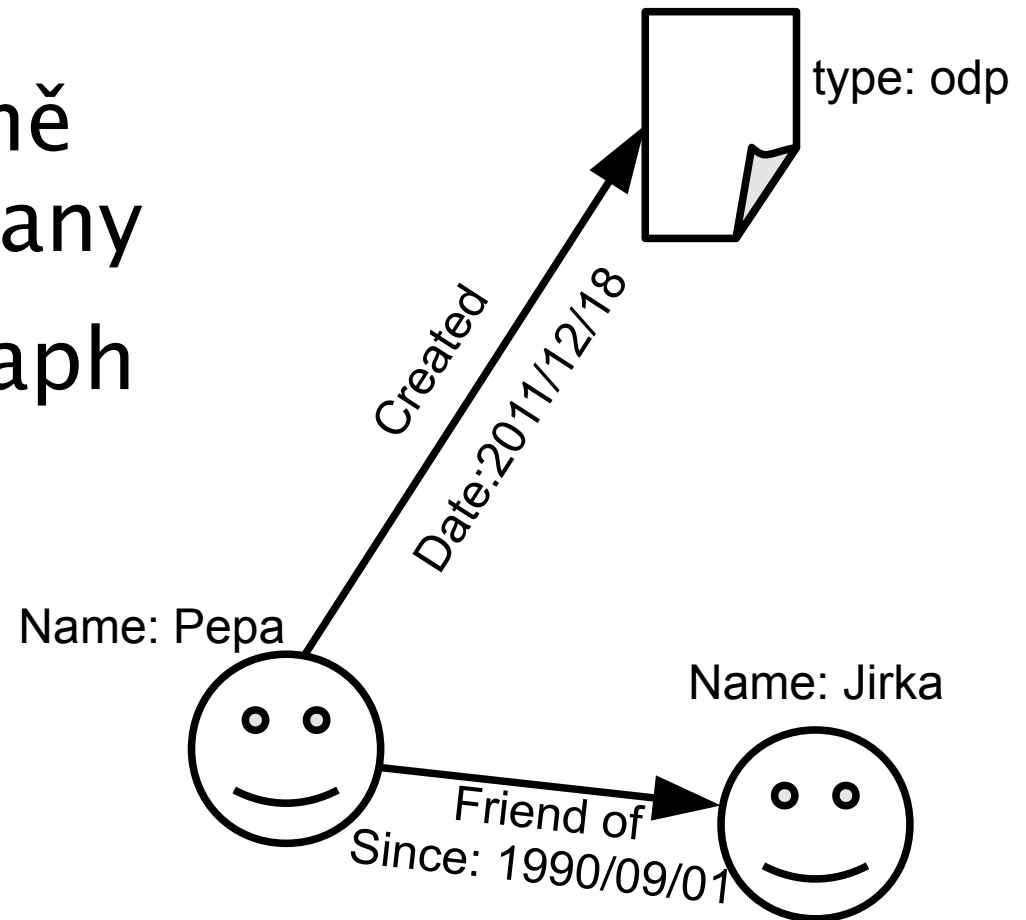


NOSQL Databáze

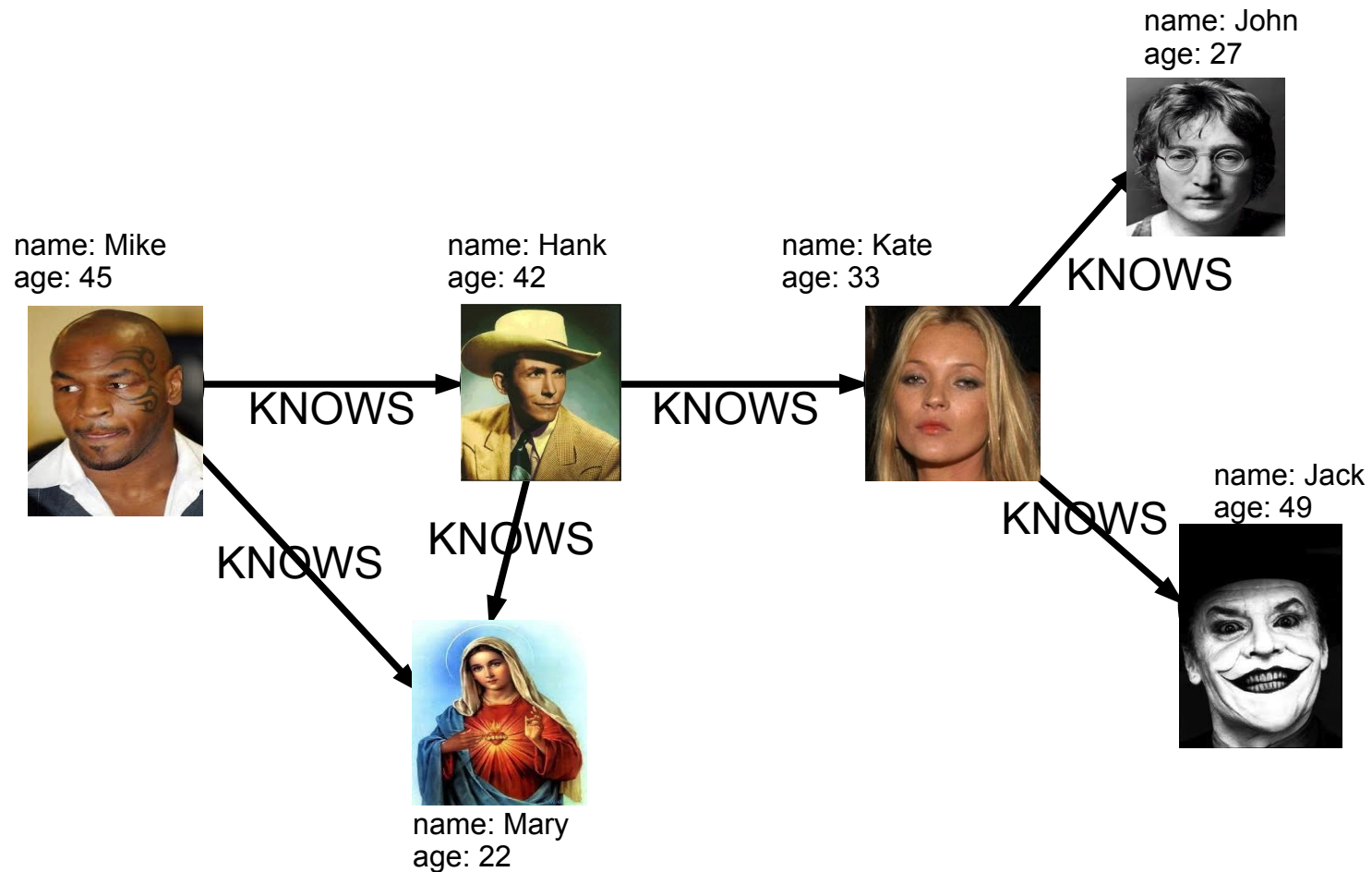


Graph DB?

- $G = (V, E)$
- Graph DB = explicitně definované uzly a hrany
- neo4j = Property Graph
 - Directed
 - Attributed
 - Multirelational
 - Schema-less



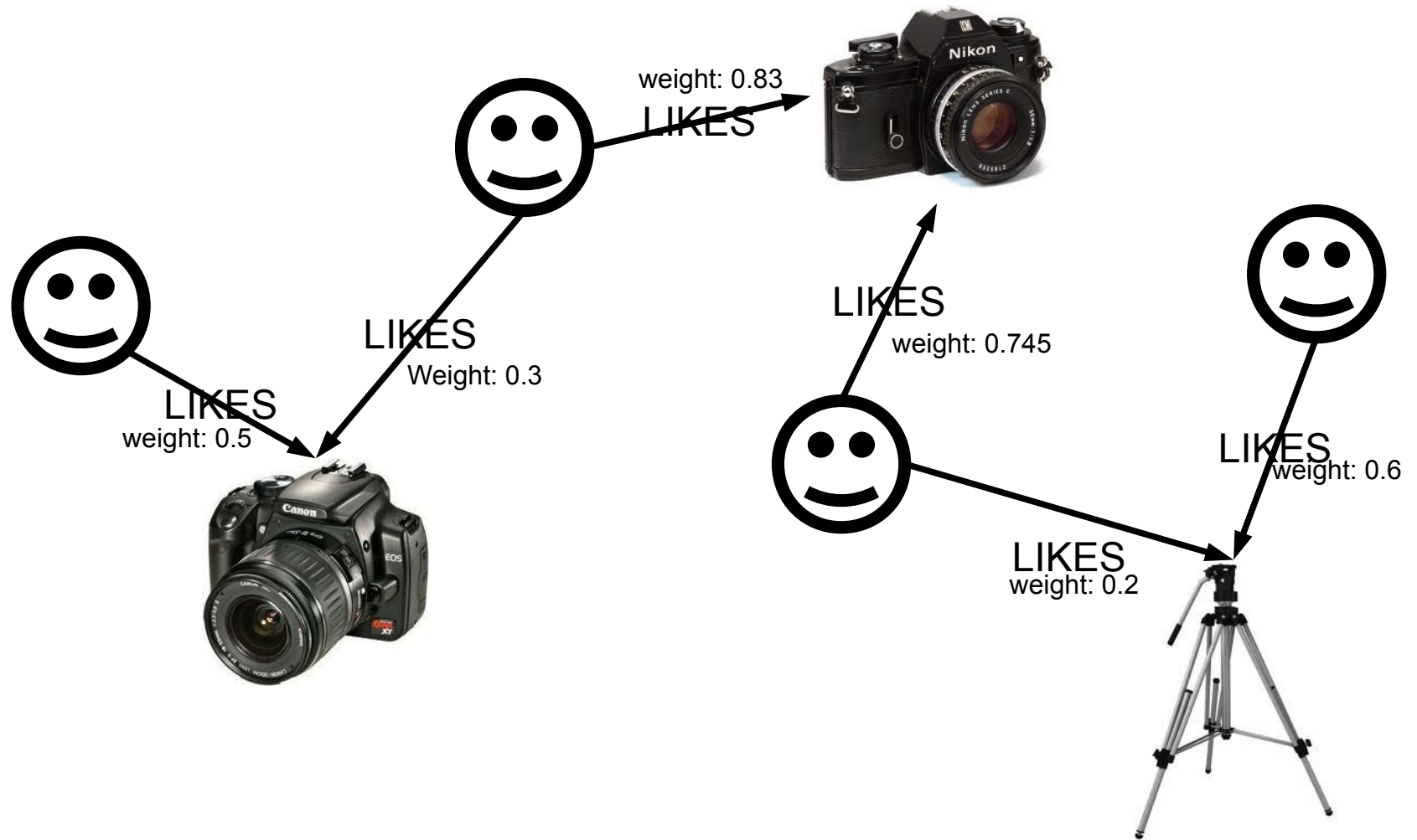
Použití: Social



Použití: Spatial



Použití: proactify.com



Použití: proactify.com

- po 7 měsících pilotního provozu
 - Embedded mode (Java, Linux, AWS)
 - 5M nodes
 - 20M+relationships
 - avg <30ms recommendations response time
- Výhled 2012:
 - >>100M nodes & 10G relationships
 - proactify.com/hiring

Write: ACID

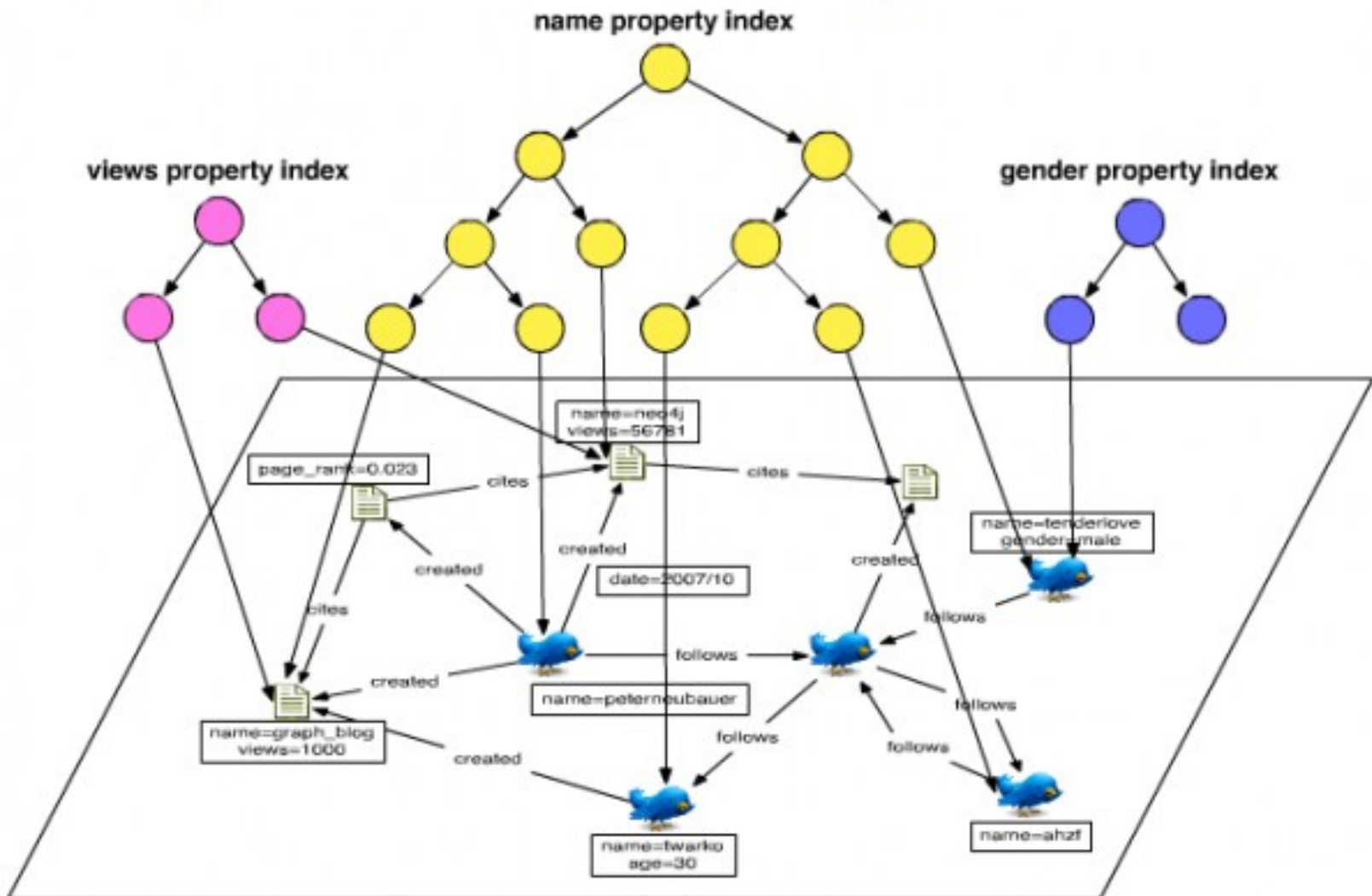
- JTA/JTS-compliant transaction manager
- 2-fázový commit

```
tx = beginTransaction
try{
    node1.setProperty („name“, „pepa“)
    node2.setProperty („name“, „jirka“)
    rel = node1.createRelationshipTo (node2, FRIEND_OF)
    rel.setProperty („since“, 1990/09/01)
    ...
    tx.success ()

} catch (ex) {
    tx.failure ()

} finally {
    tx.finish ()
}
```


Read: Lucene-index



Read: Pattern-matching

SPARQL:

```
SELECT ?y WHERE {  
    g:1 g:friend_of ?x .  
    ?x g:name ?y  
}
```

Read: Traversal

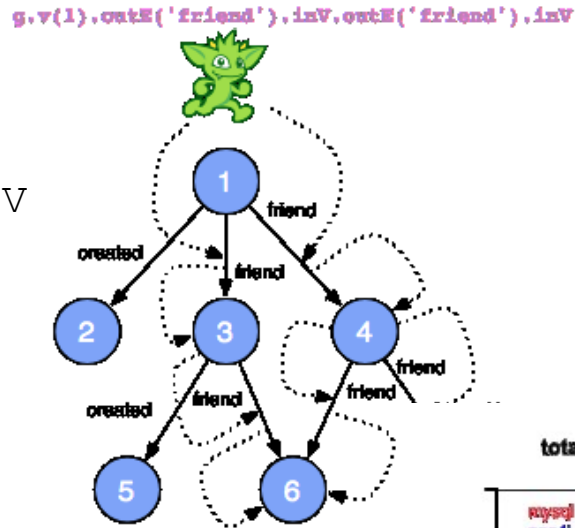
3-step traversal: Gremlin

```
g.V(?).outE.inV.outE.inV.outE.inV
```

1

2

3



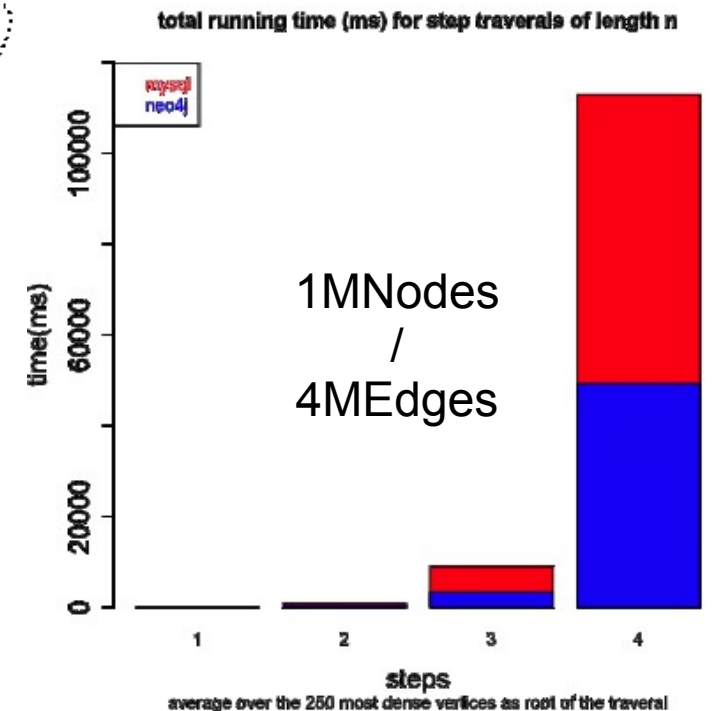
3-step traversal v MySQL – table JOINS

```
SELECT c.inV
FROM graph as a, graph as b, graph as c
WHERE
a.inV=b.outV AND b.inV=c.outV AND a.outV=?
```

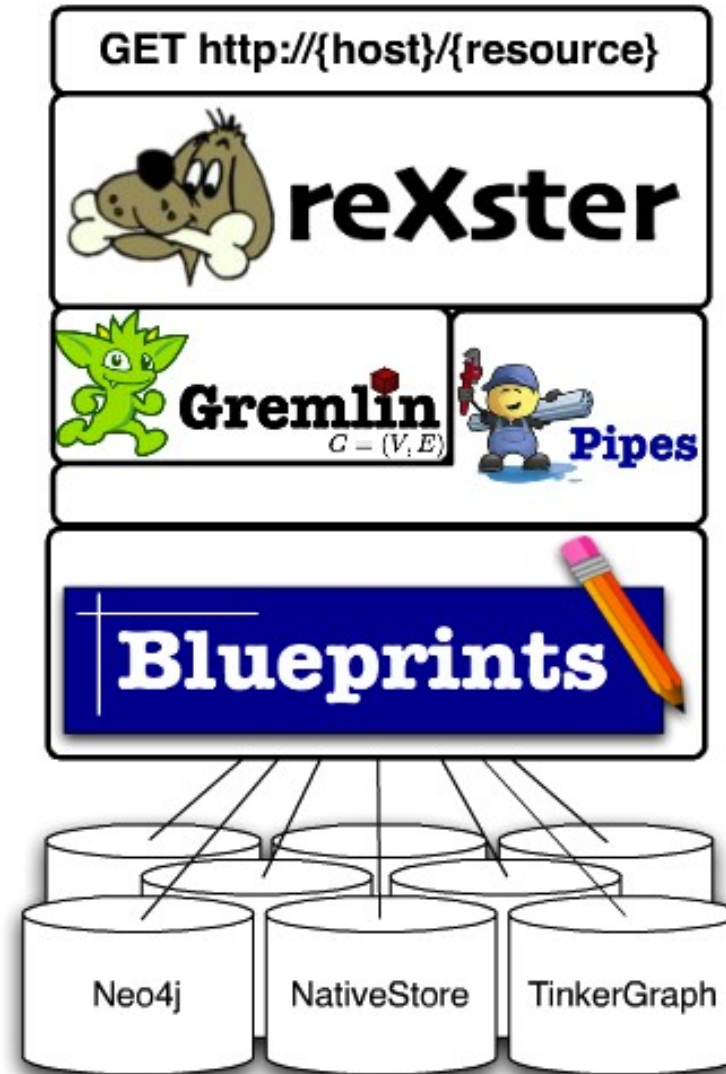
1

2

3



Tinkerpop.com technologies





We Want

Díky za pozornost!
Dejte se k nám!
proactify.com/hiring

Josef Holý
@proactify
Josef.Holy@proactify.com

proactify 