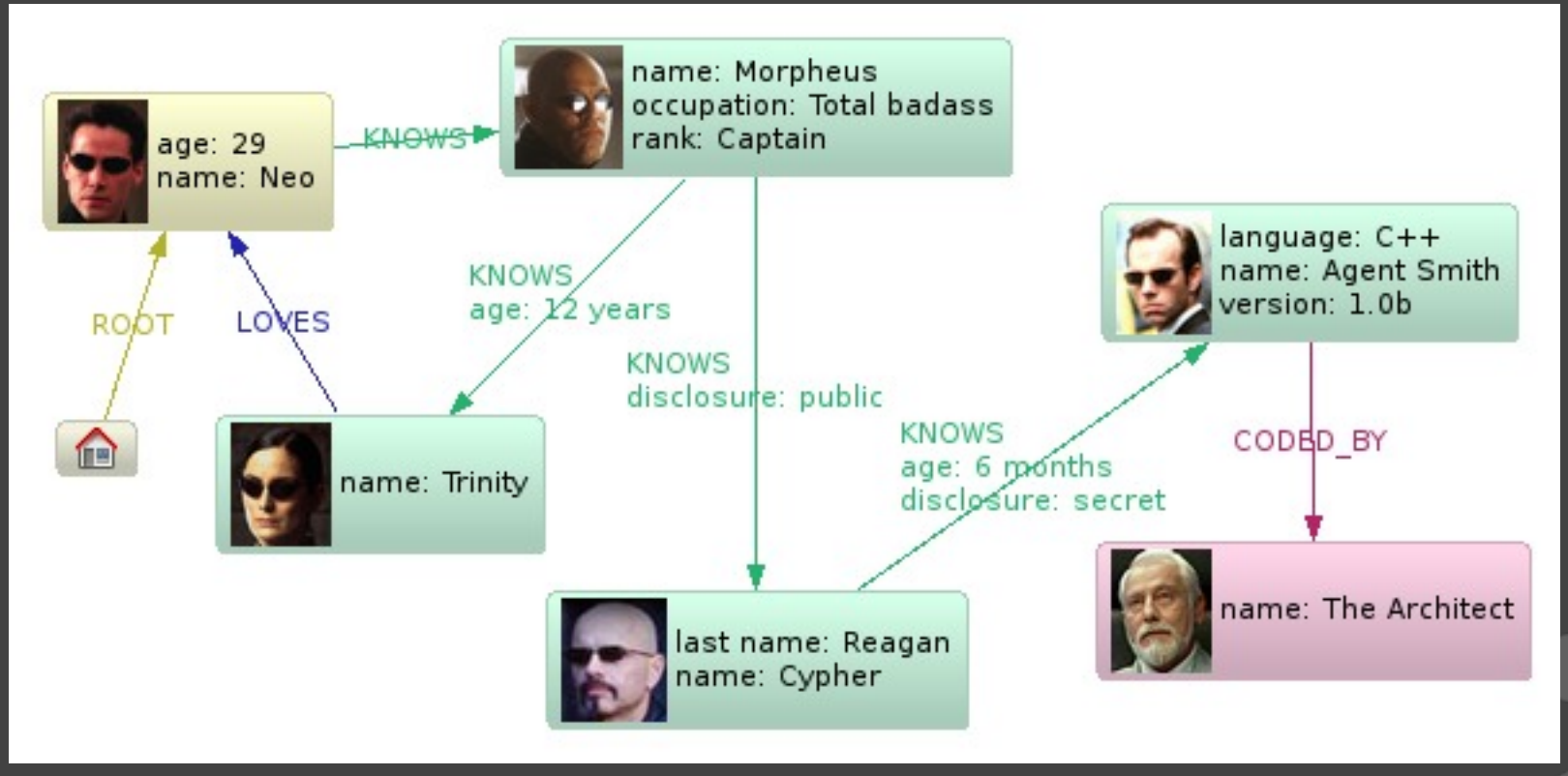
How to build a developer community with limited resources

GOTO CPH 2012

Peter Neubauer
Founder, Neo Technology

#neo4j
@peterneubauer
peter@neotechnology.com

What is Neo4j?



(friends)-[:create]->(project)-[:supports]->(community)

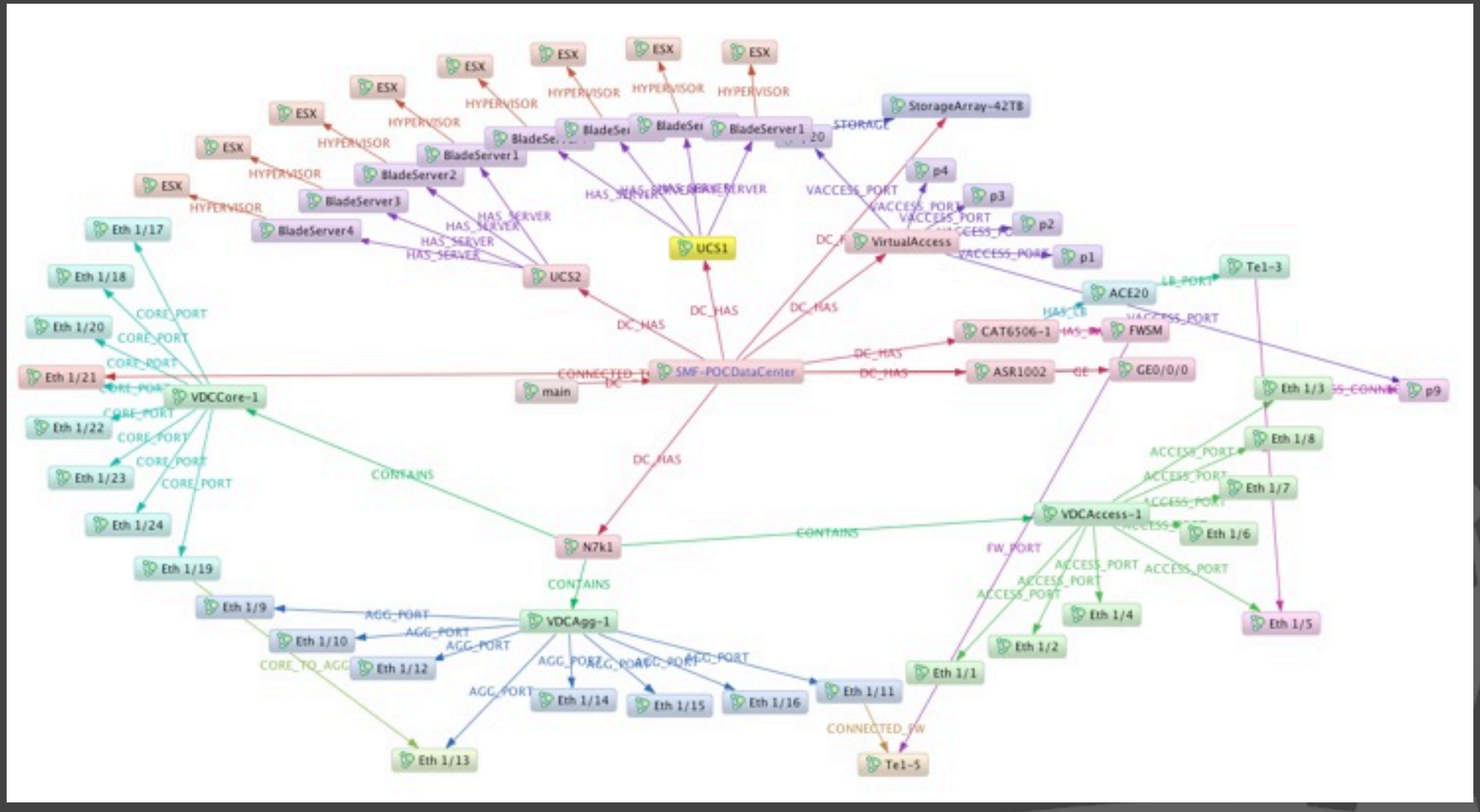
Neo4j
the graph database

What is Neo4j?

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

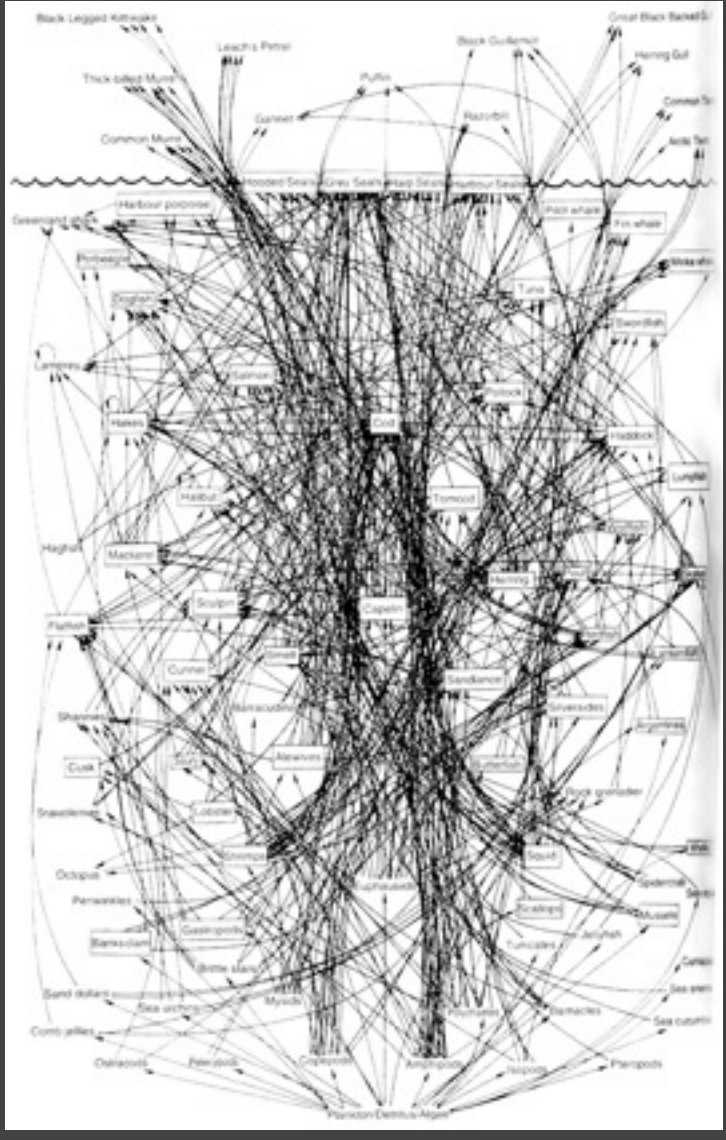
What is it good for?



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

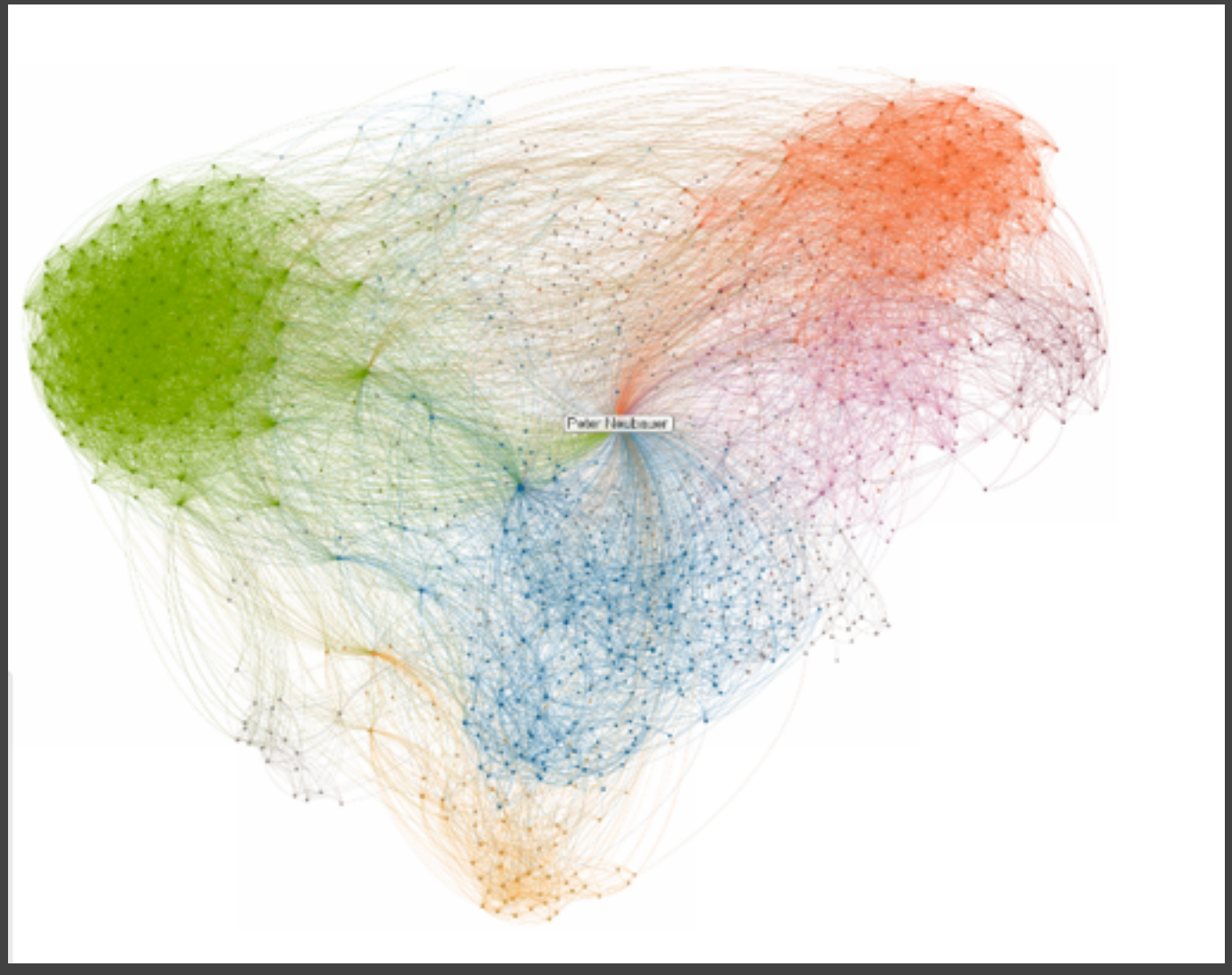
What is it good for?



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What is it good for?



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The (G)Raffle

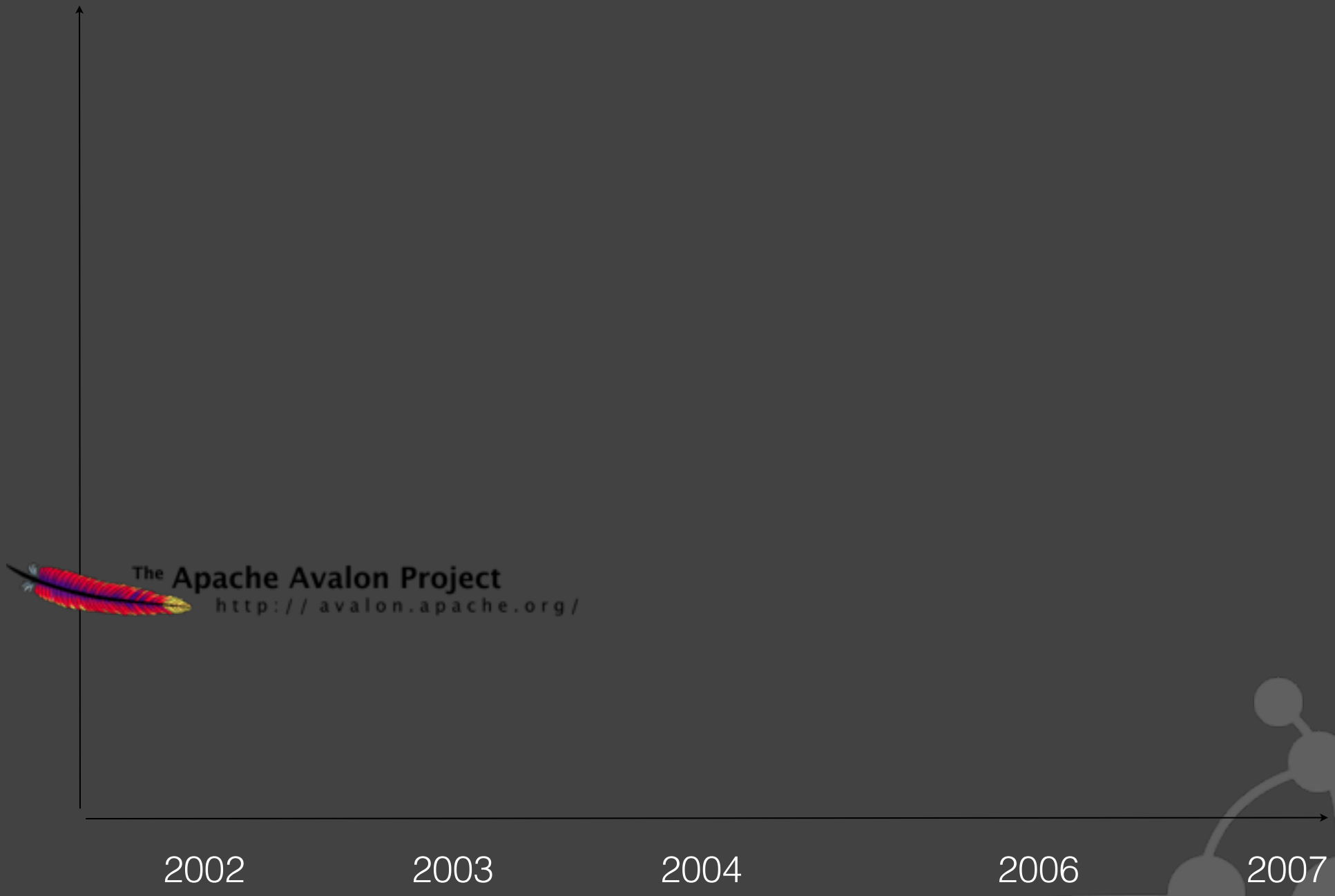
http://graffle-goto-cph.herokuapp.com/

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

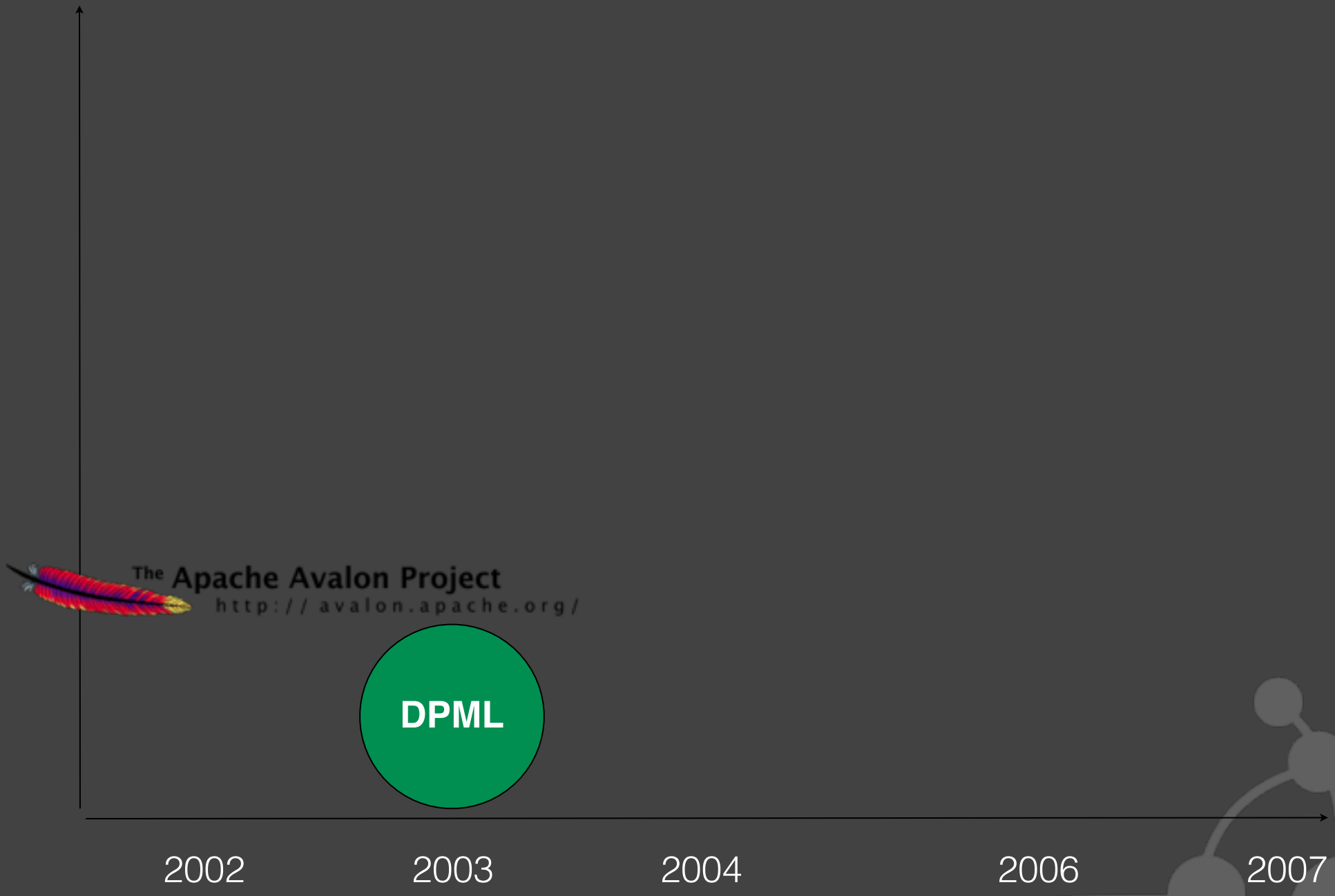
(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database



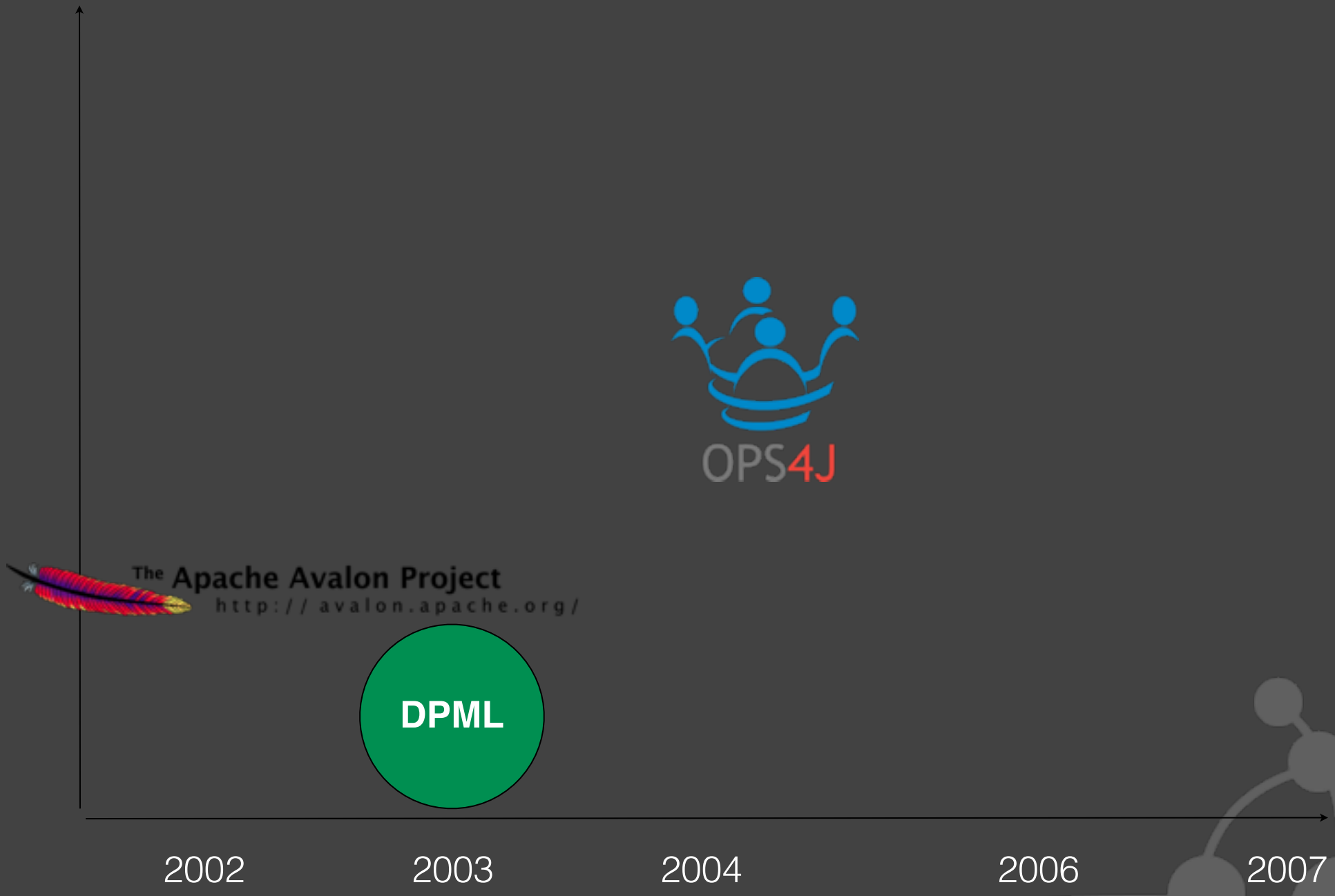
(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database



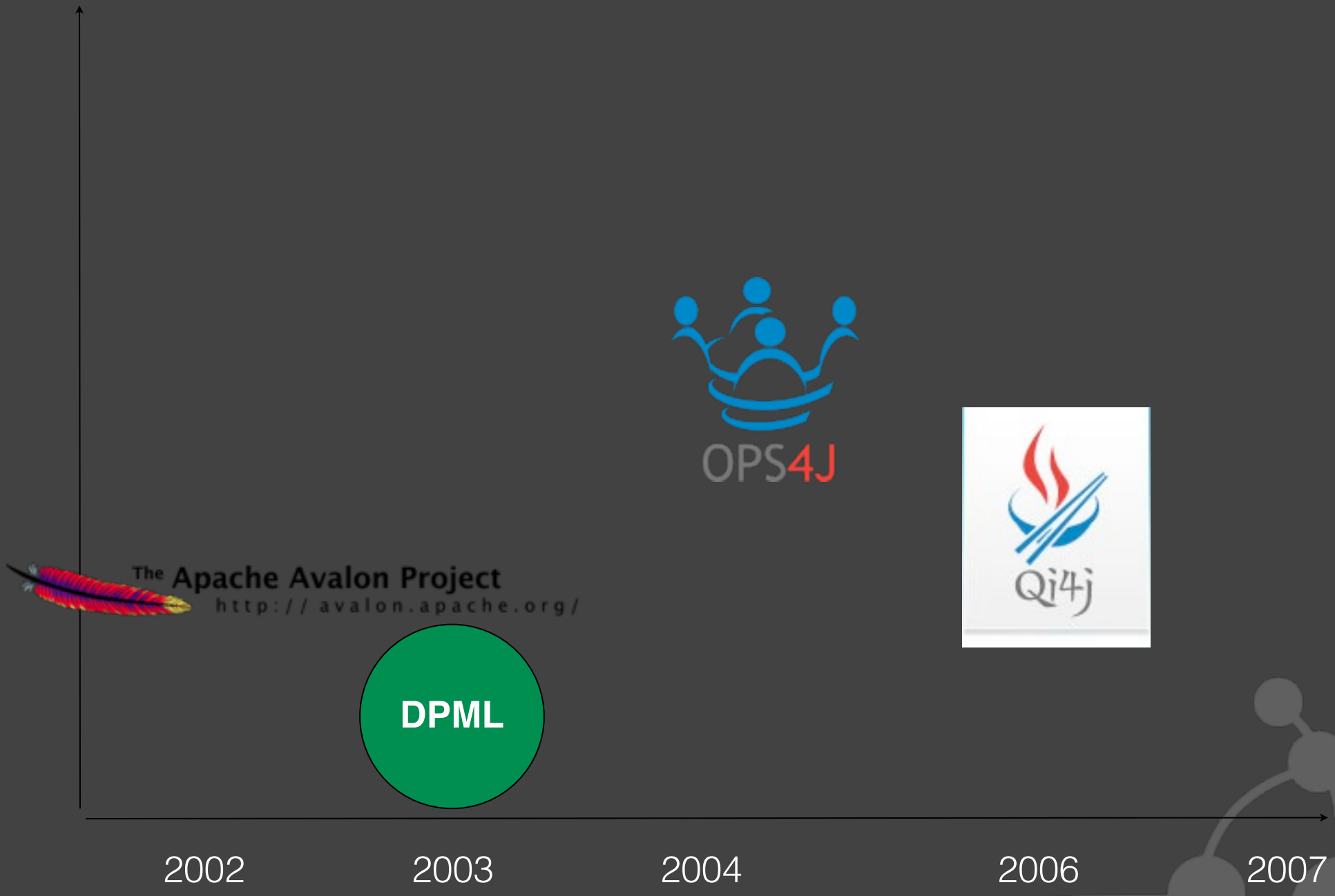
(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database



(friends)-[:create]->(project)-[:supports]->(community)

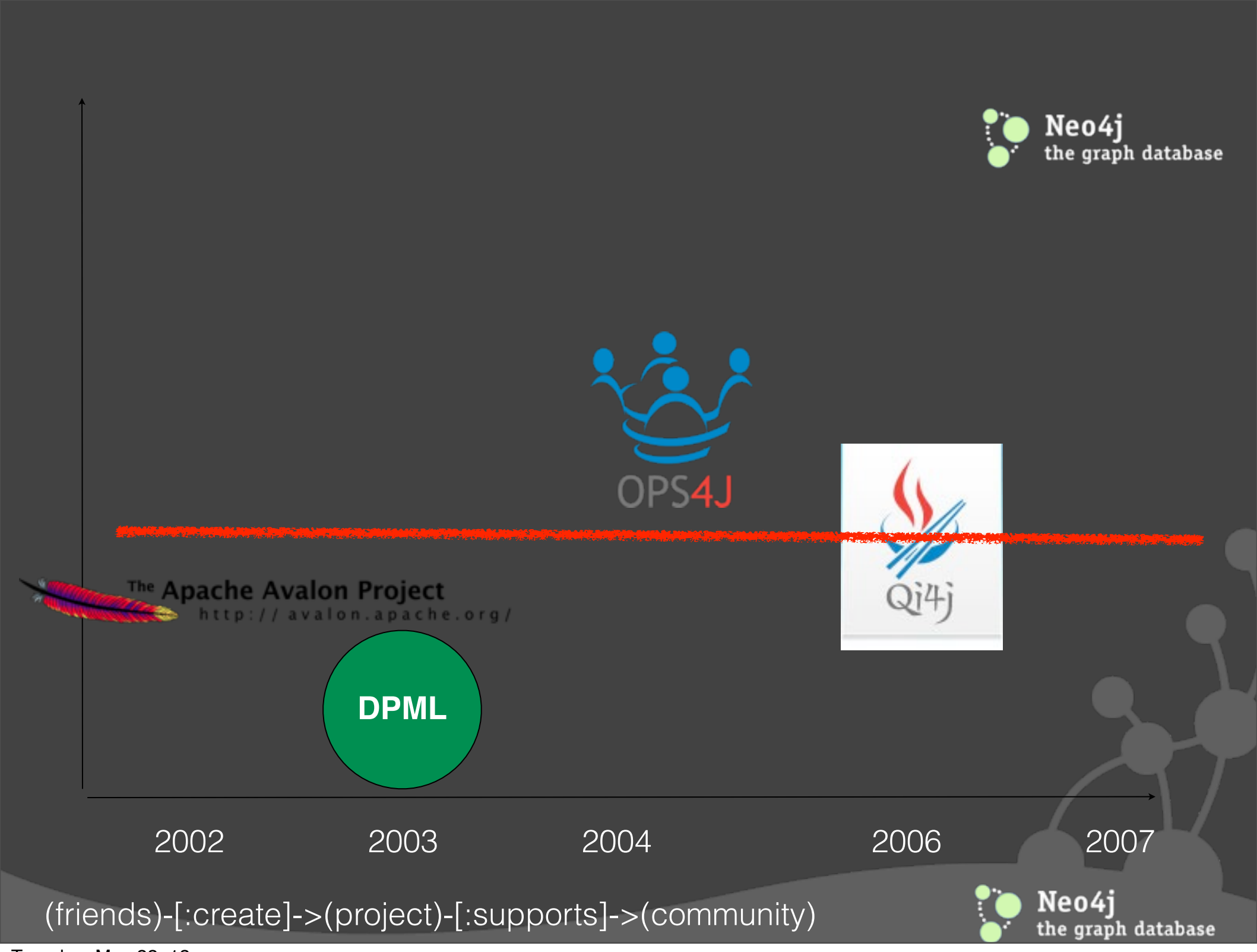
Neo4j
the graph database



(friends)-[:create]->(project)-[:supports]->(community)

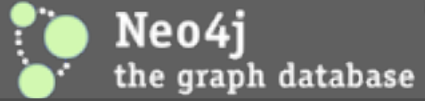
Neo4j
the graph database

(friends)-[:create]->(project)-[:supports]->(community)


Neo4j
the graph database
OPS4J
Qi4j
The Apache Avalon Project
http://avalon.apache.org/
DPML
2002
2003
2004
2006
2007
(friends)-[:create]->(project)-[:supports]->(community)
Neo4j
the graph database

#oss dimensions

(friends)-[:create]->(project)-[:supports]->(community)



#oss dimensions

- **Free and Commercial**

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors
- Giving and Taking

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors
- Giving and Taking
- Distribution and Cloud

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors
- Giving and Taking
- Distribution and Cloud
- Freedom and Measurement

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors
- Giving and Taking
- Distribution and Cloud
- Freedom and Measurement
- Focus and Innovation

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors
- Giving and Taking
- Distribution and Cloud
- Freedom and Measurement
- Focus and Innovation
- Licensing and Adoption

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

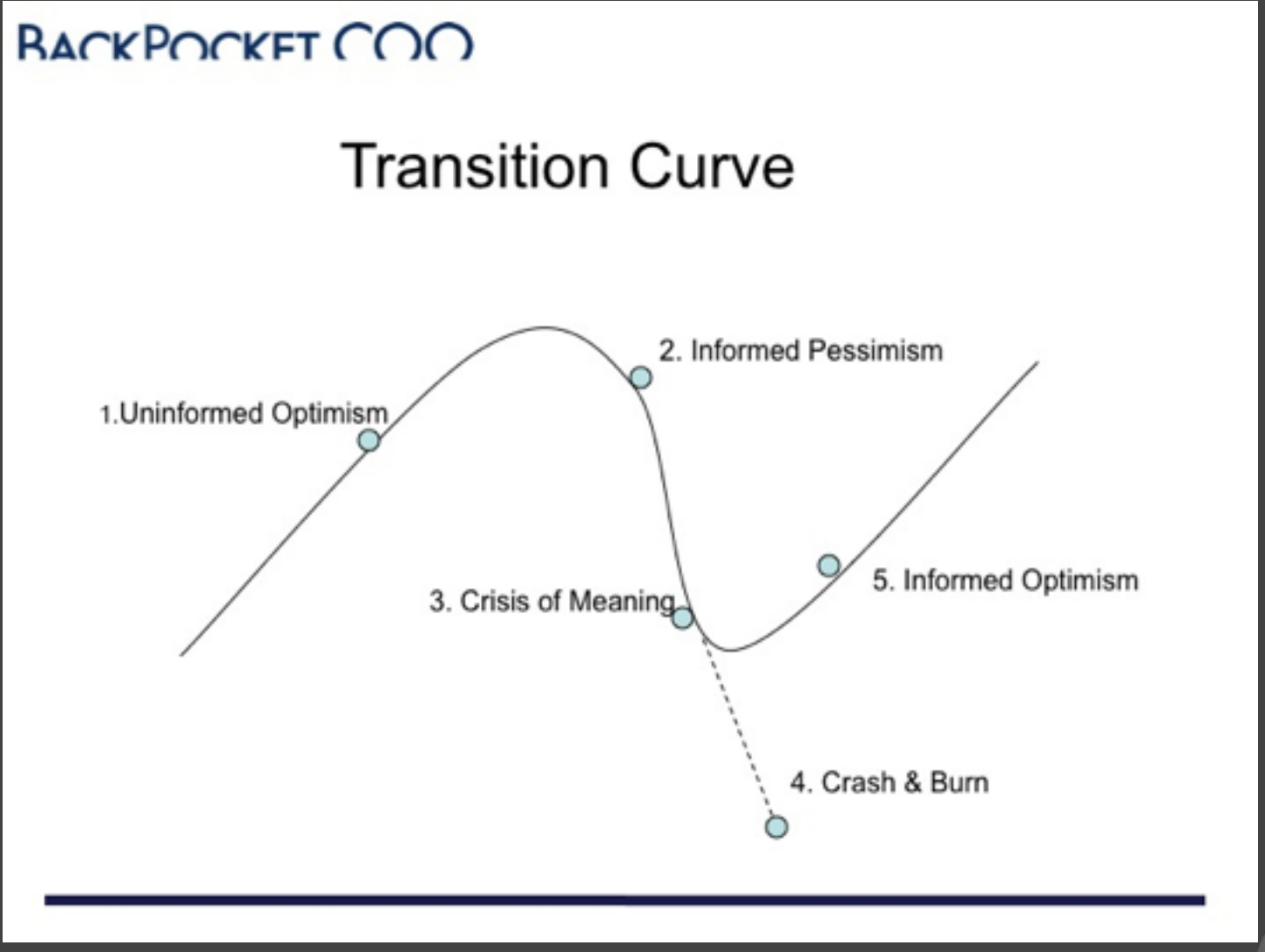
#oss dimensions

- Free and Commercial
- Acting and Saying
- Users and Contributors
- Giving and Taking
- Distribution and Cloud
- Freedom and Measurement
- Focus and Innovation
- Licensing and Adoption
- Code and Documentation

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The hype curve

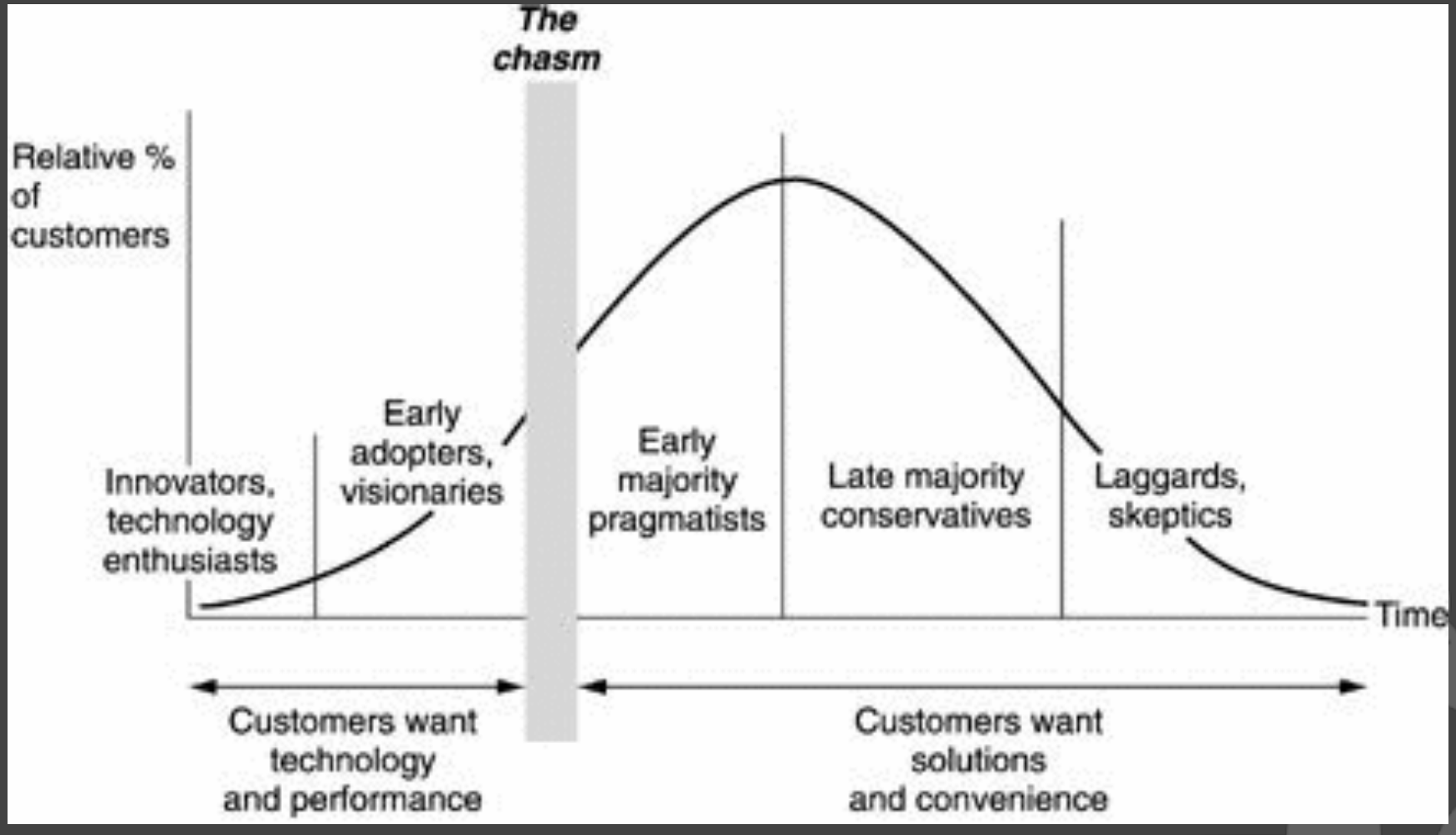


(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database


100% TOBACCO

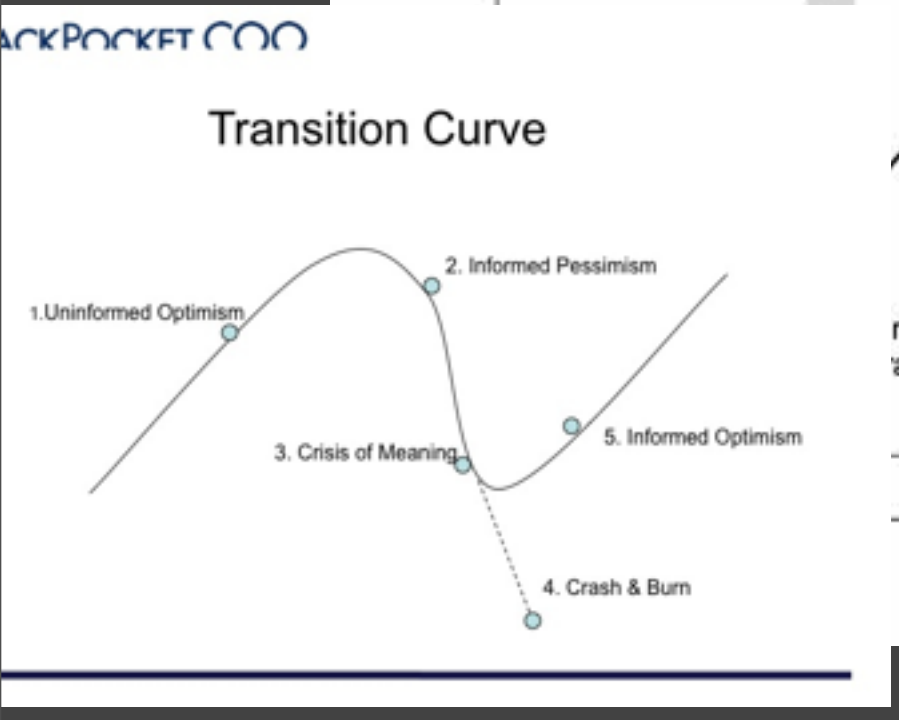
The chasm

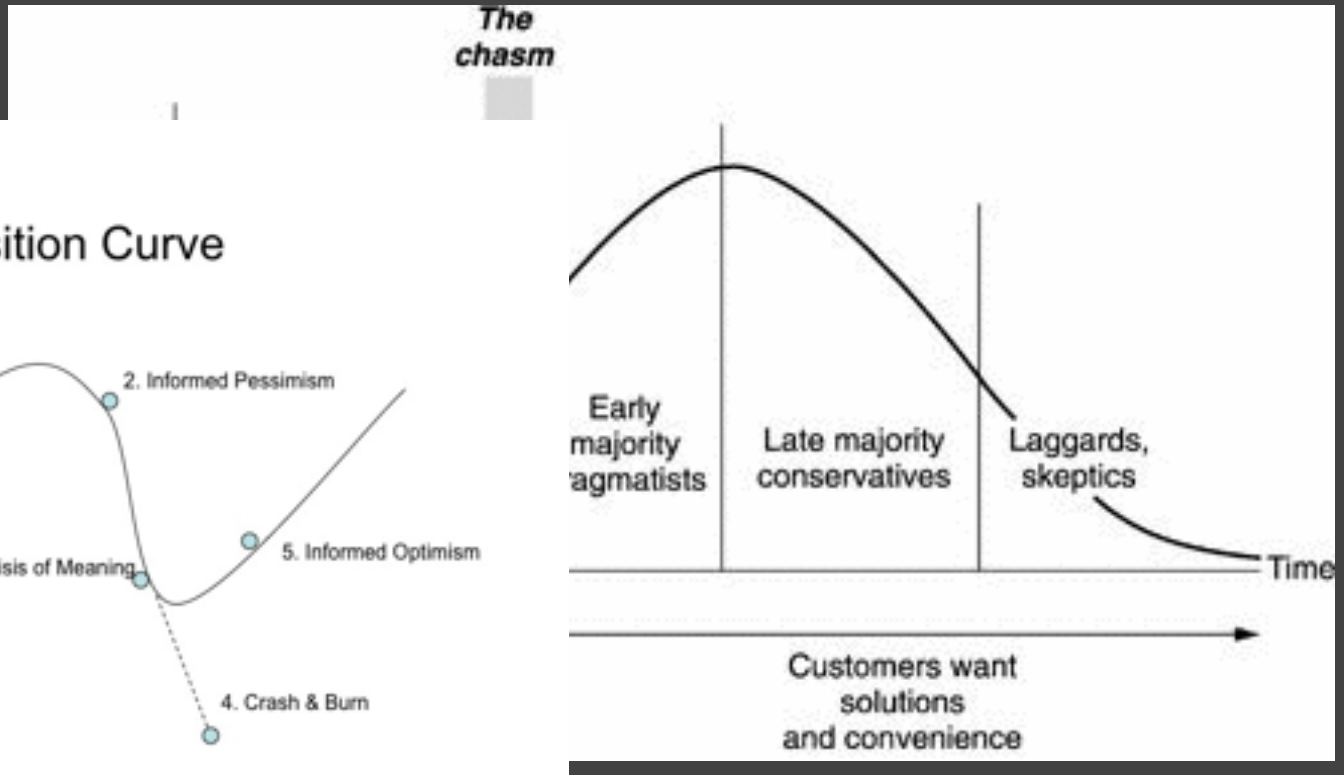


(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Theory





(friends)-[:create]->(project)-[:supports]->(community)

Neo4j the graph database

Reality

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Reality

(friends)-[:create]->(project)-[:supports]->(community)

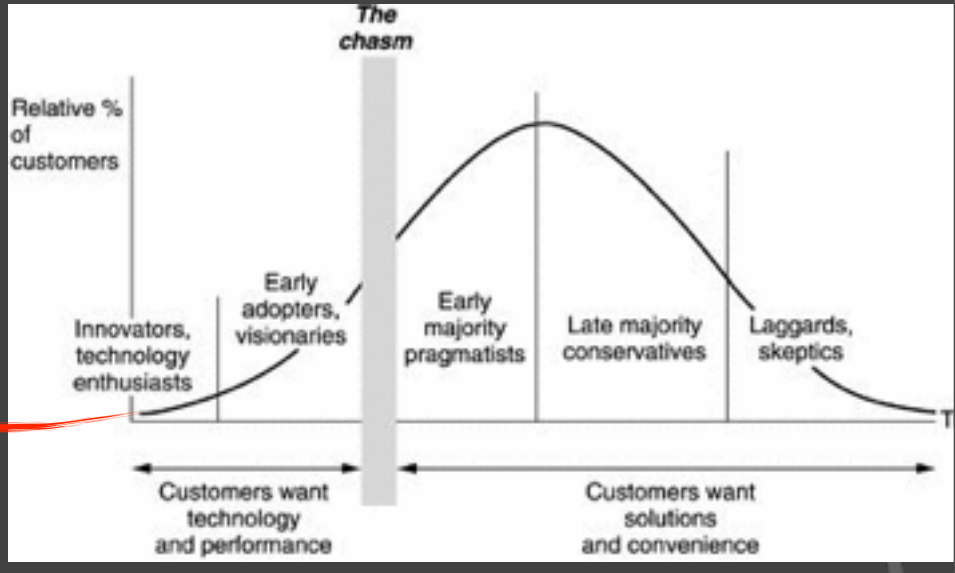
Neo4j
the graph database

Reality

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Reality

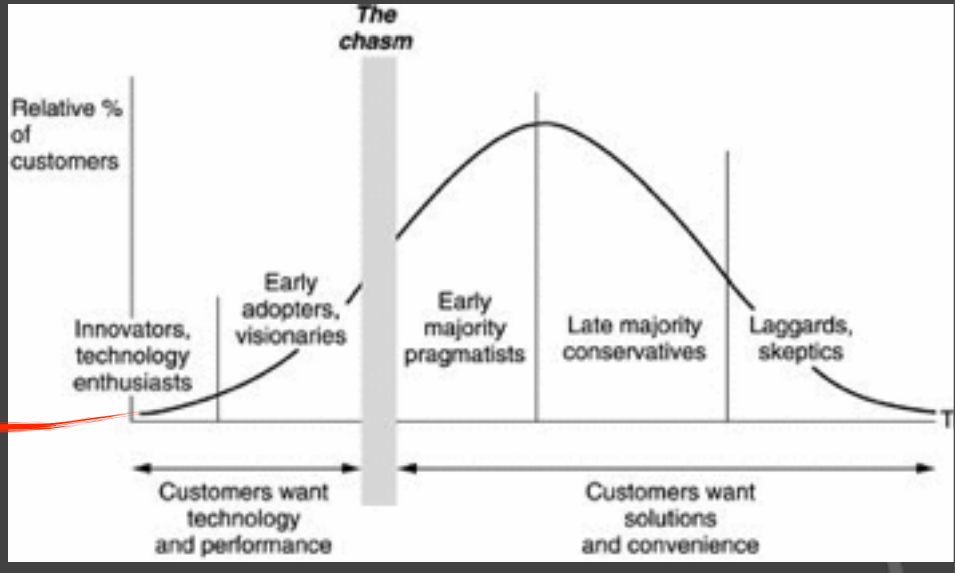


(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Reality

2000

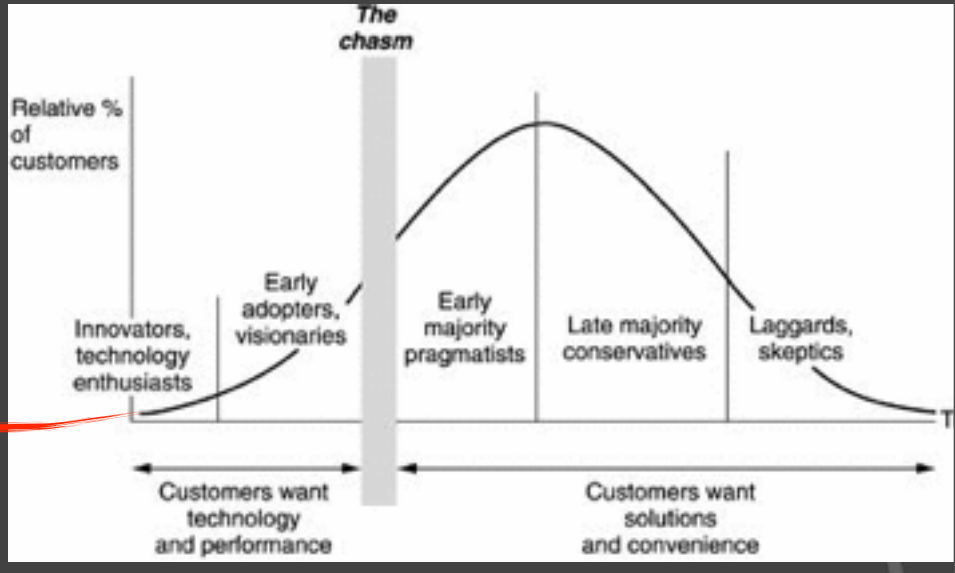


(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Reality

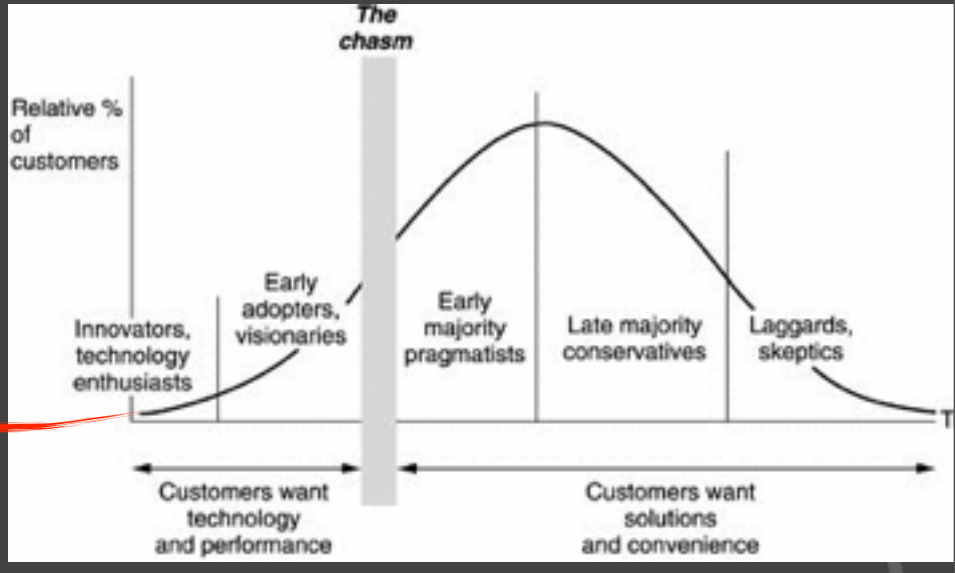
2000 2003



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j the graph database

Reality

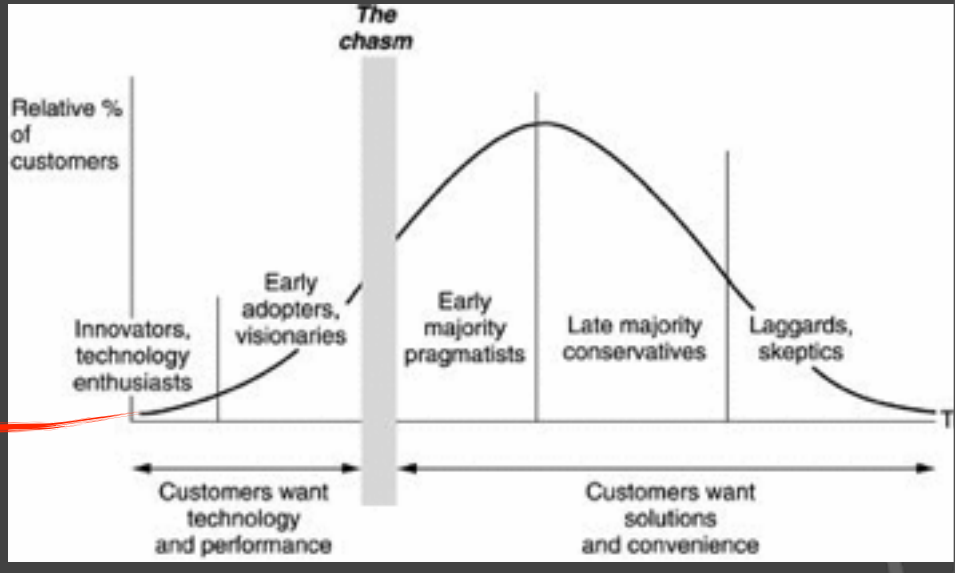


2000 2003 2007

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j the graph database

Reality

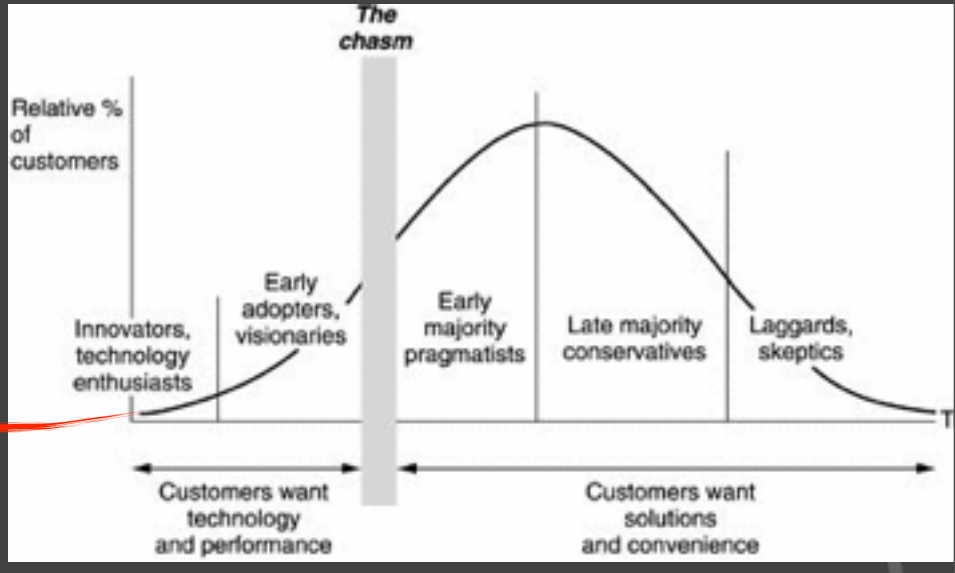


2000 2003 2007 2009

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j the graph database

Reality

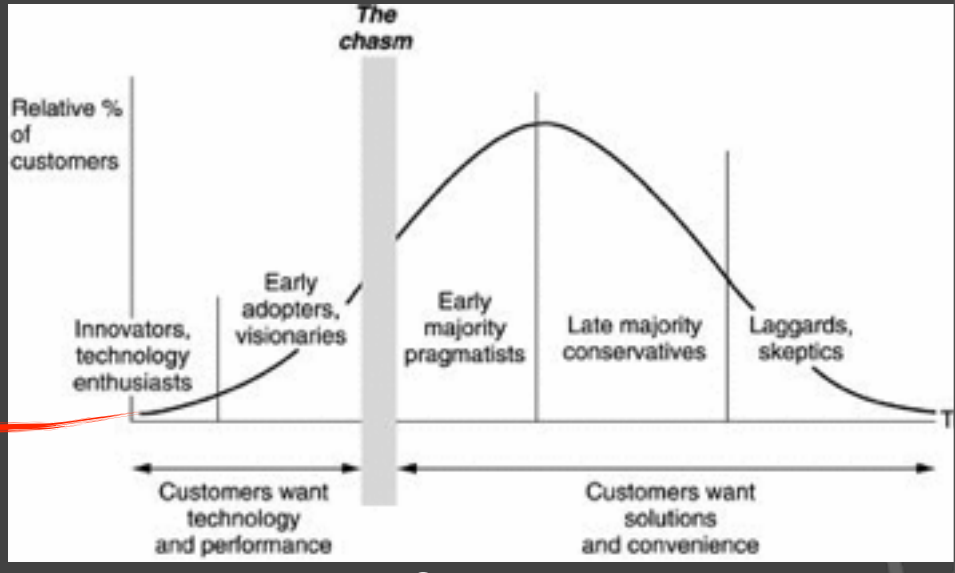


2000 2003 2007 2009 2013

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j the graph database

Reality



2000 2003 2007 2009 2013 ?

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Value in relationships

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Central teams don’t scale

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Document for the long term, Wikis suck.

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Empower others to produce content

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Be brave, test, measure, dogfood, automate

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Listen to all channels

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Support your contributors

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Beyond budgeting

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

No vendor talk

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

What we learned

(friends)-[:create]->(project)-[:supports]->(community)

What we learned

Usage feedback is ok, if honest.

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors
Issues

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors

Issues

Response

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors

Issues

Response

Onboarding

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors

Issues

Response

Onboarding

Core team engagement

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors

Issues

Response

Onboarding

Core team engagement

Mindshare

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Our programs

Contributors

Issues

Response

Onboarding

Core team engagement

Mindshare

Infrastructure

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The docs toolchain

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

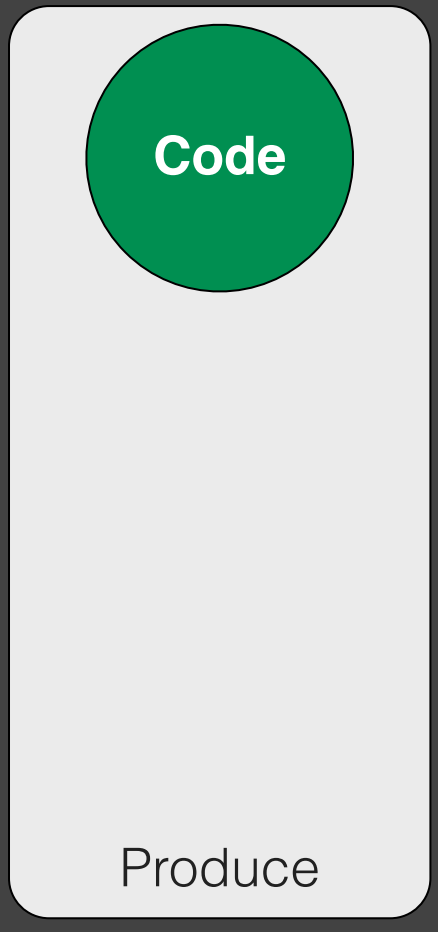
The docs toolchain

Produce

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

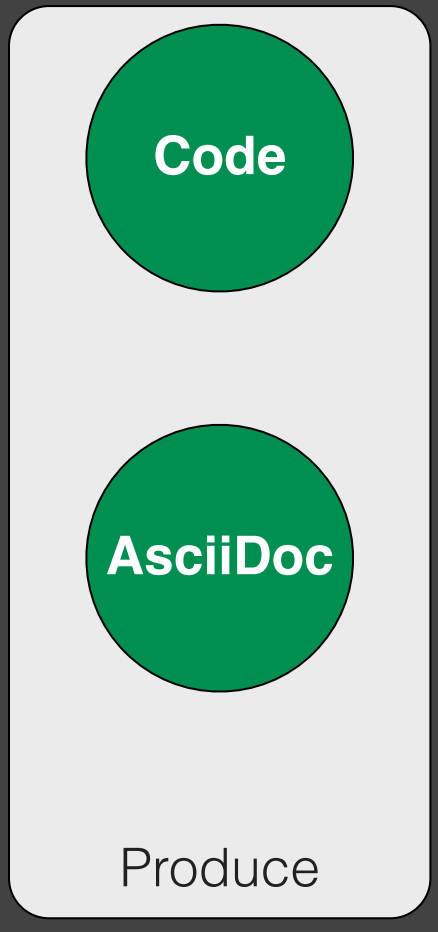
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

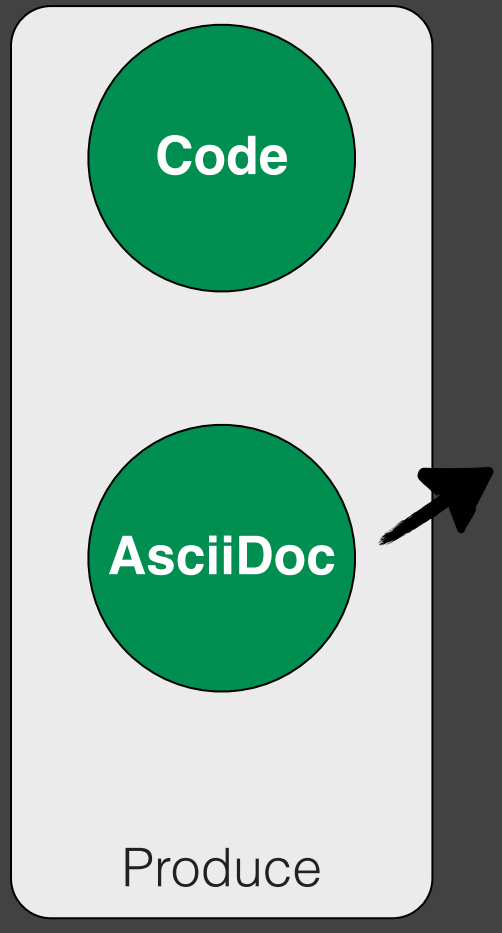
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

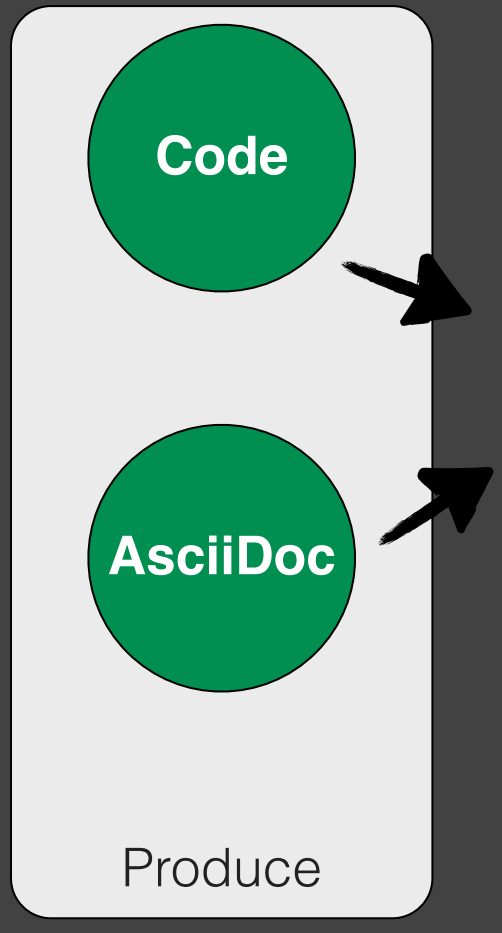
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

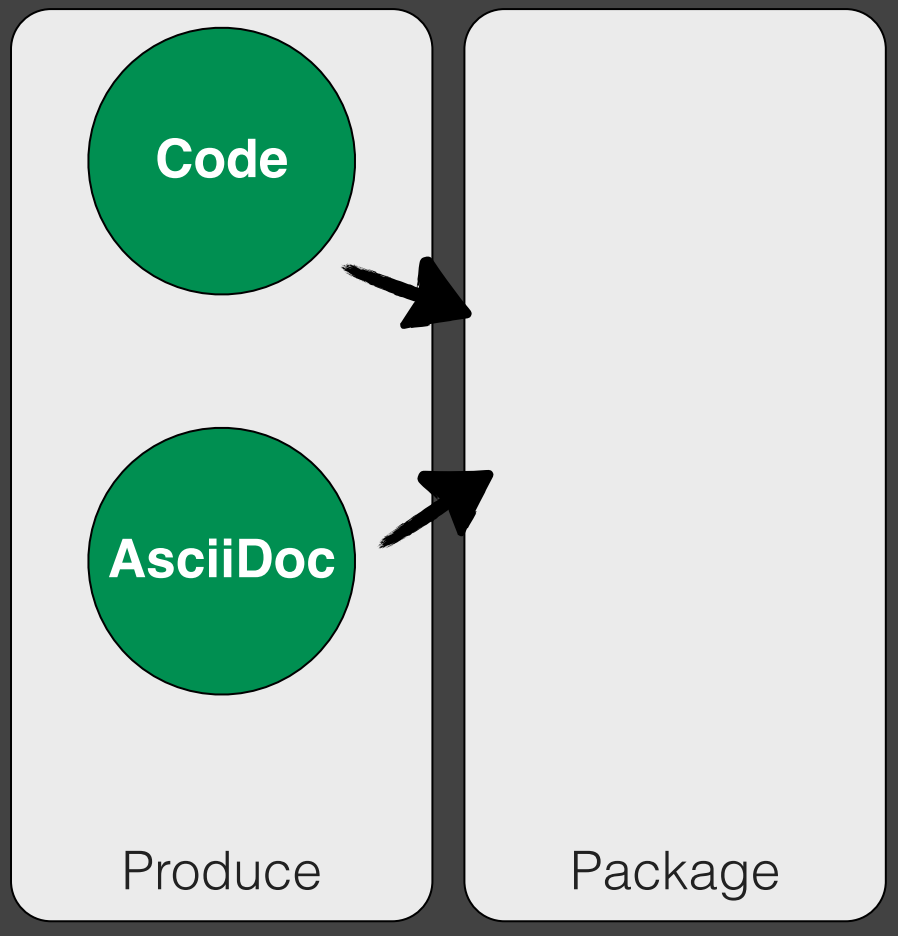
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

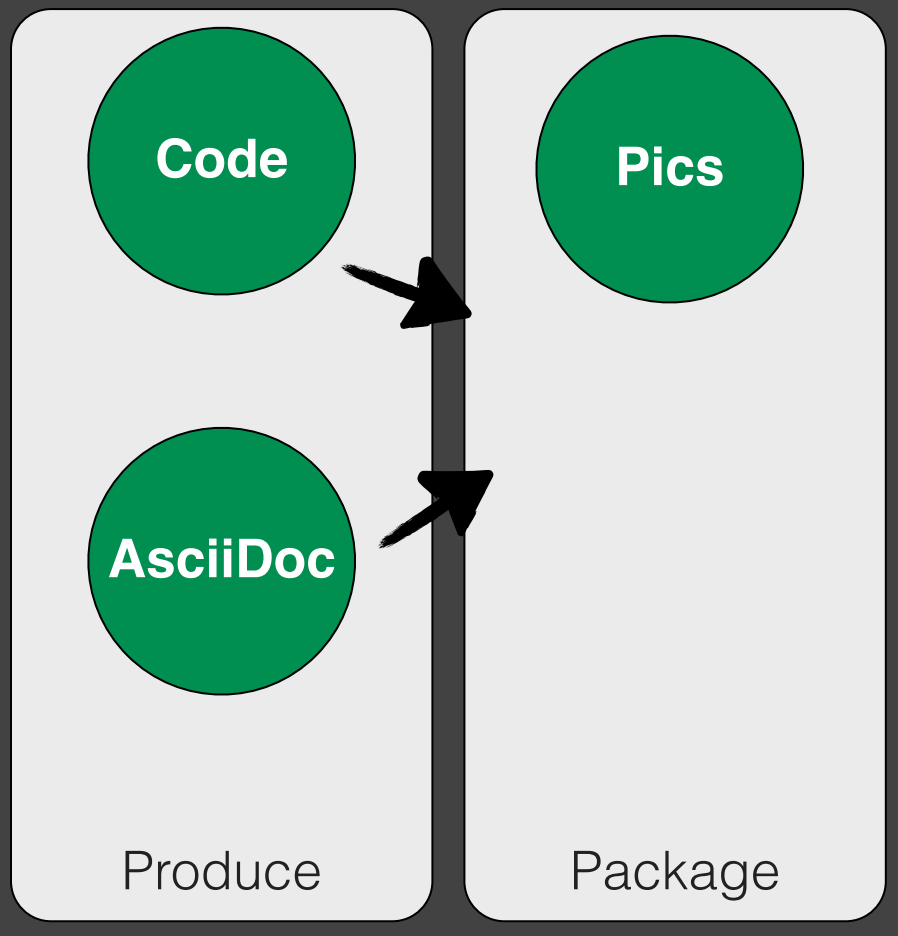
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

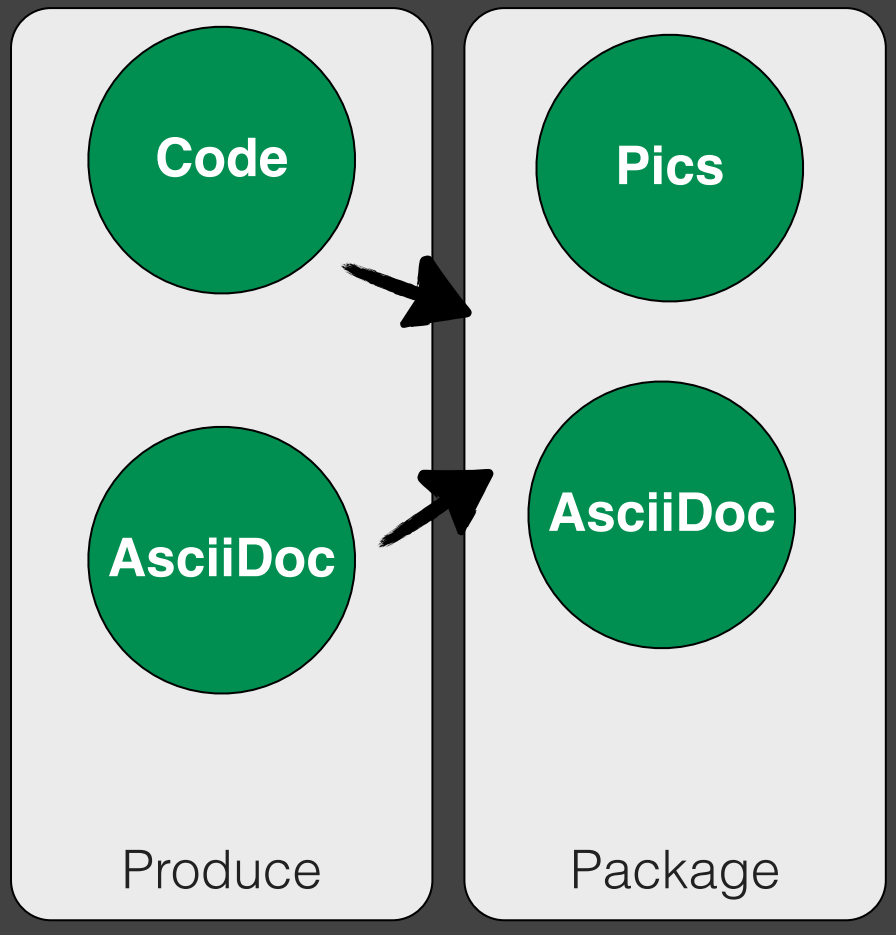
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

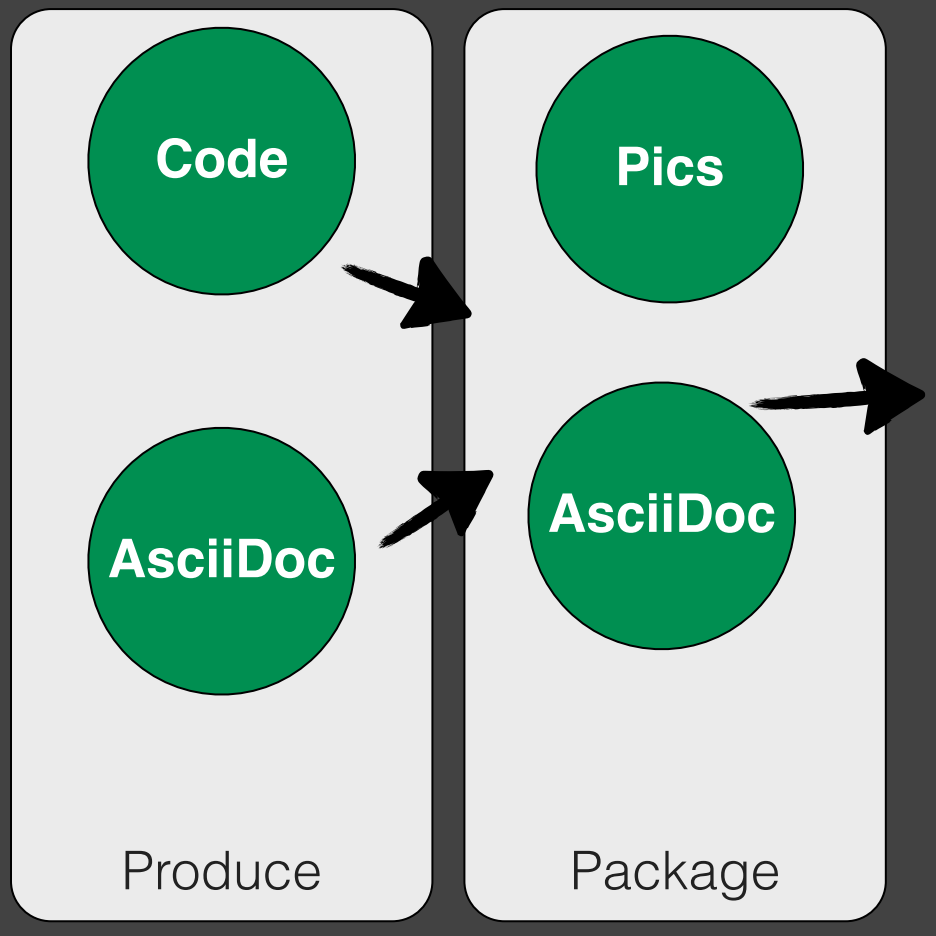
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

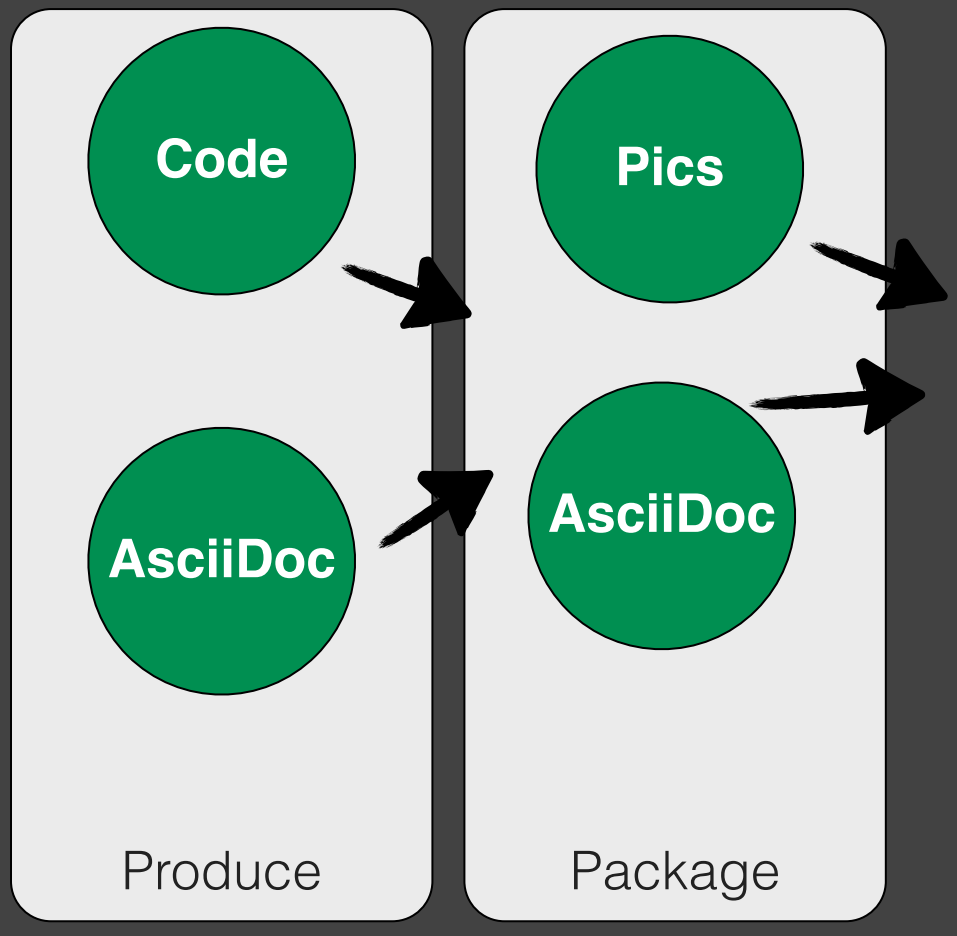
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

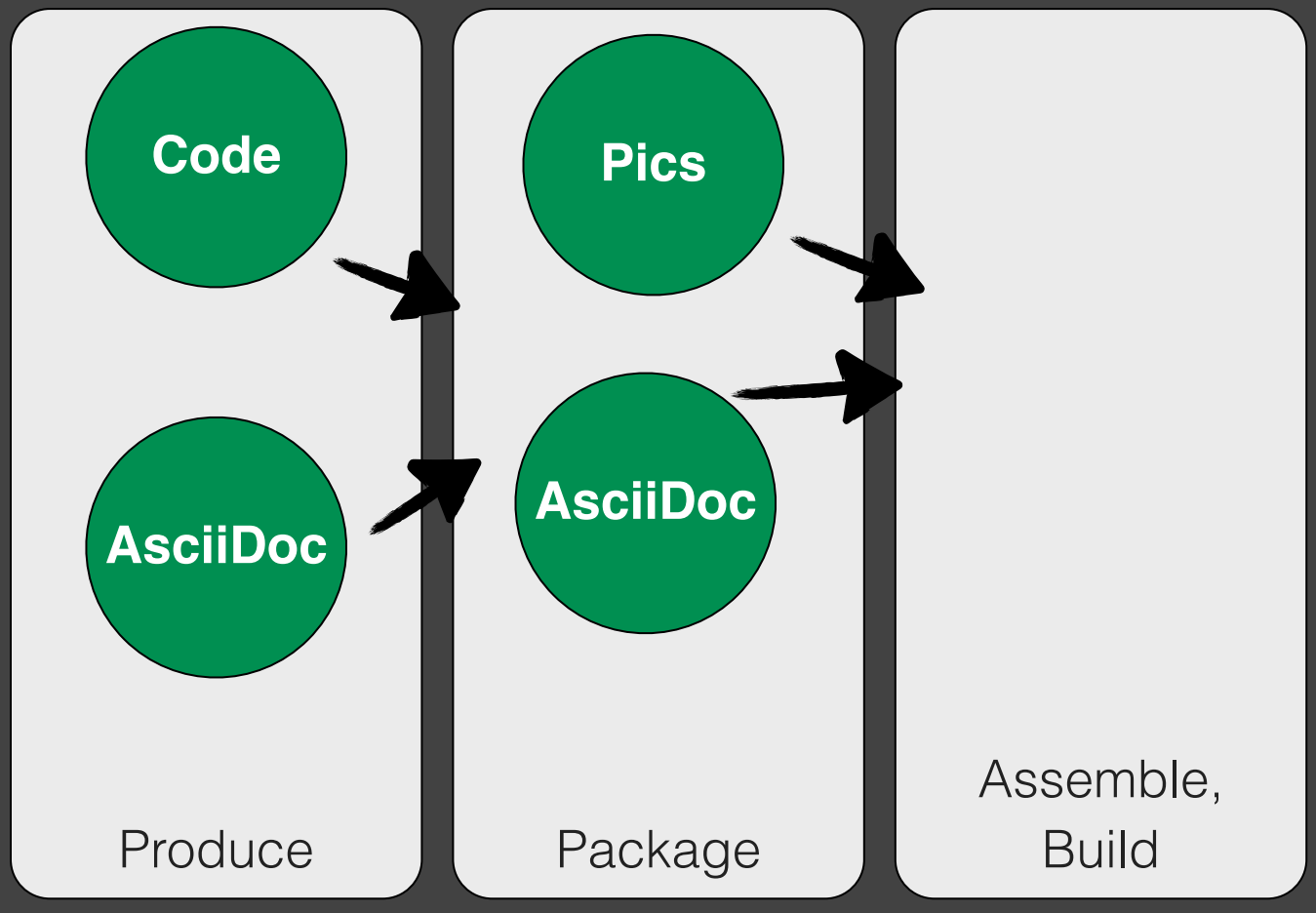
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

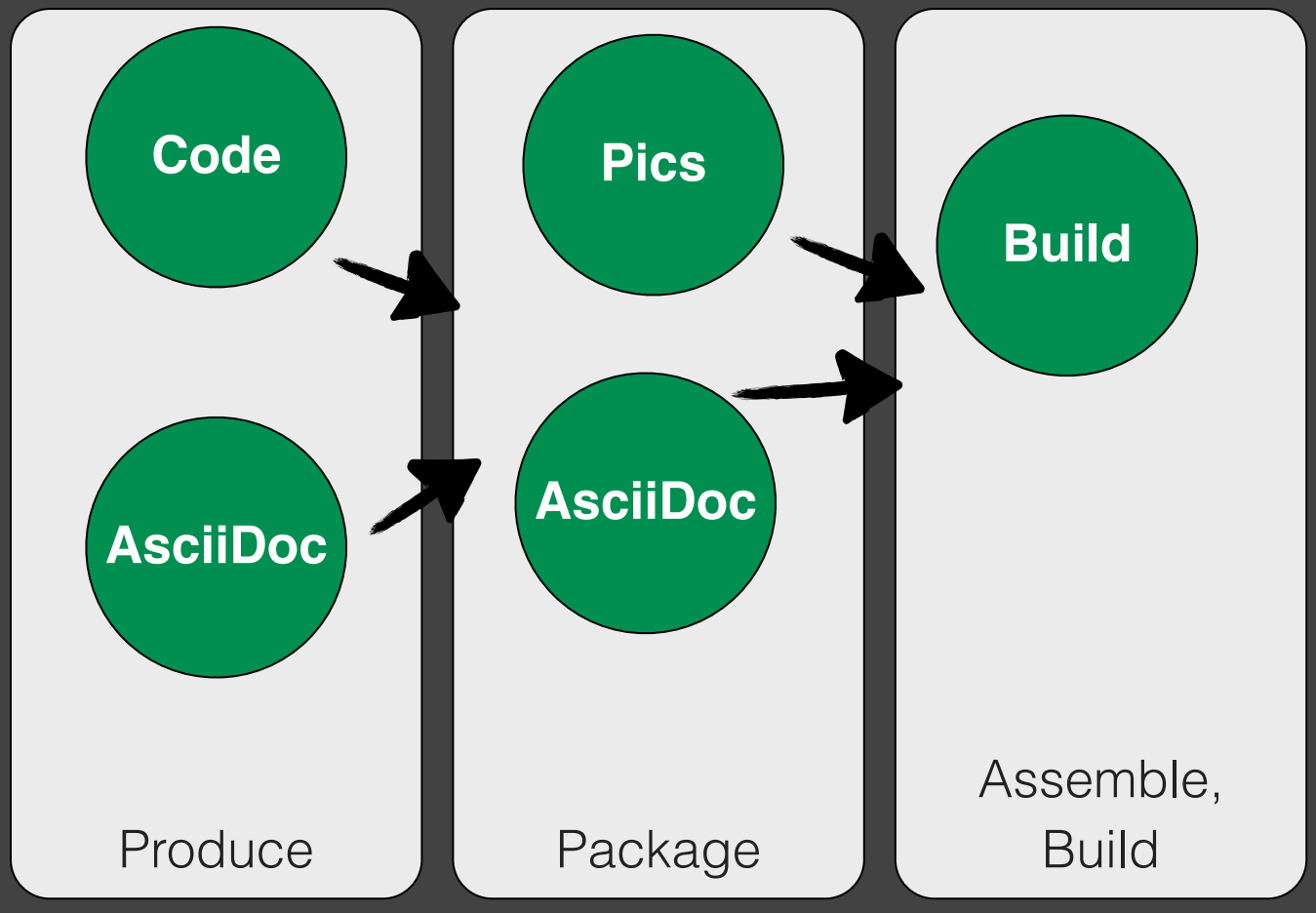
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

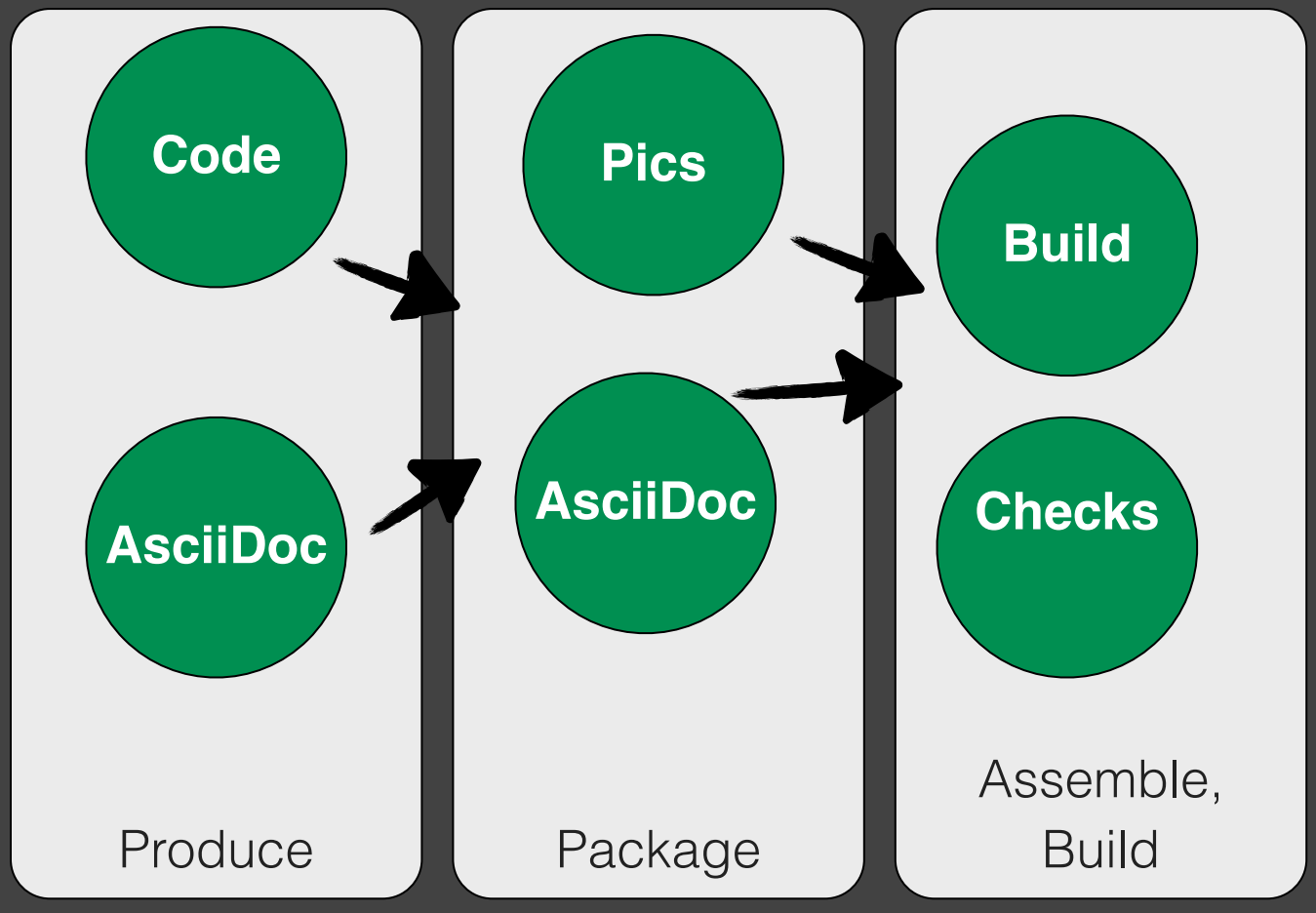
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

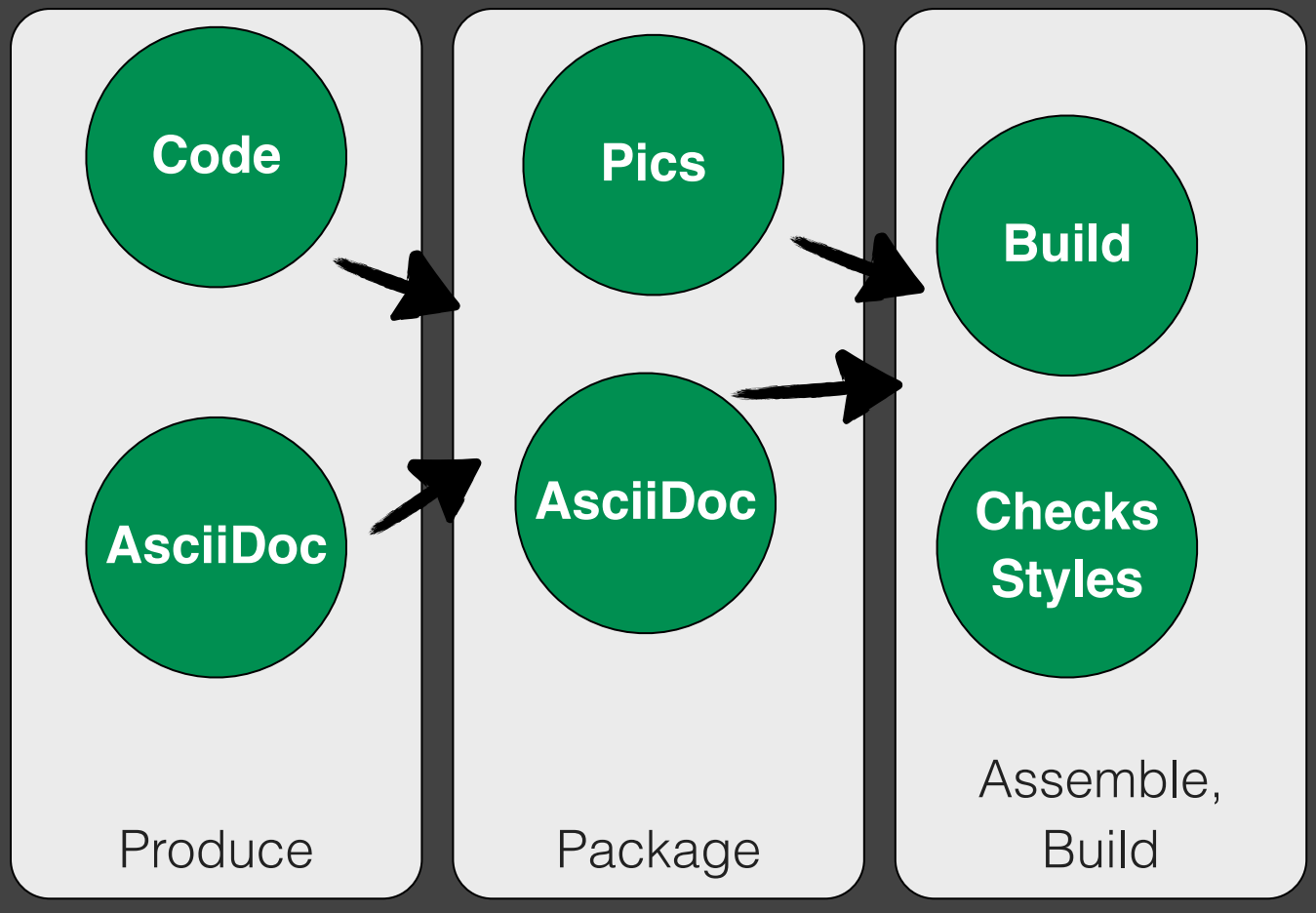
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

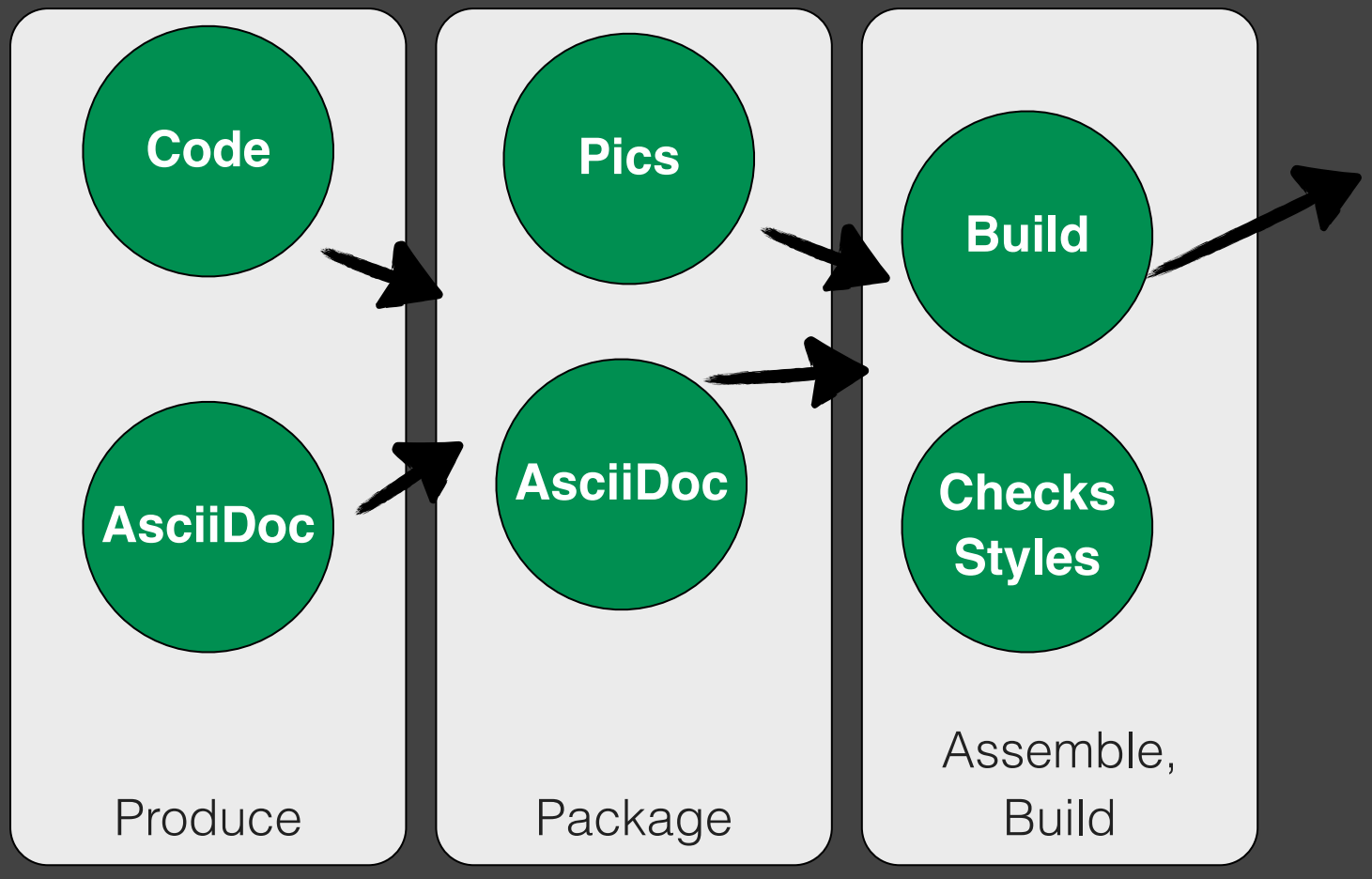
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

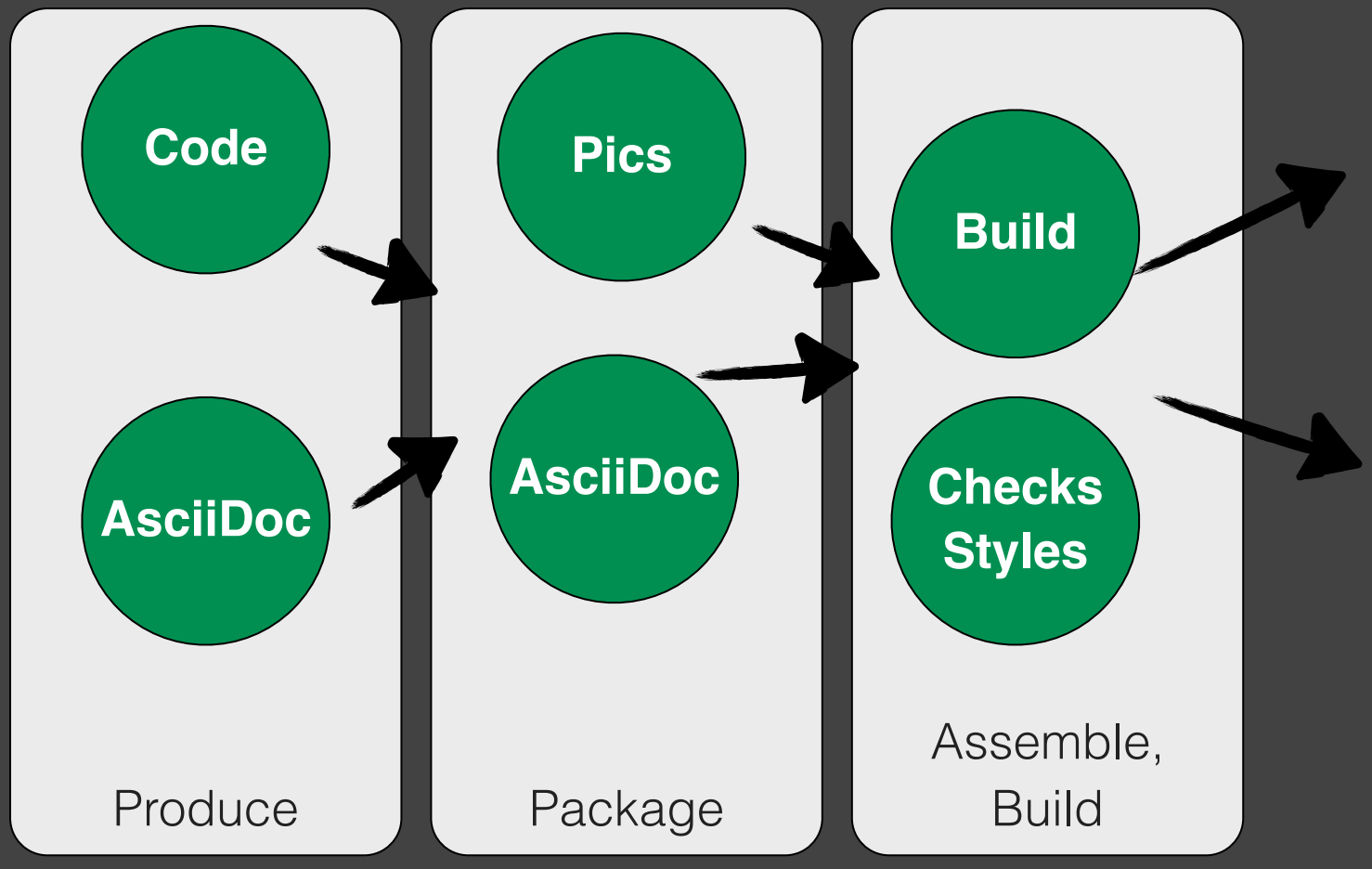
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

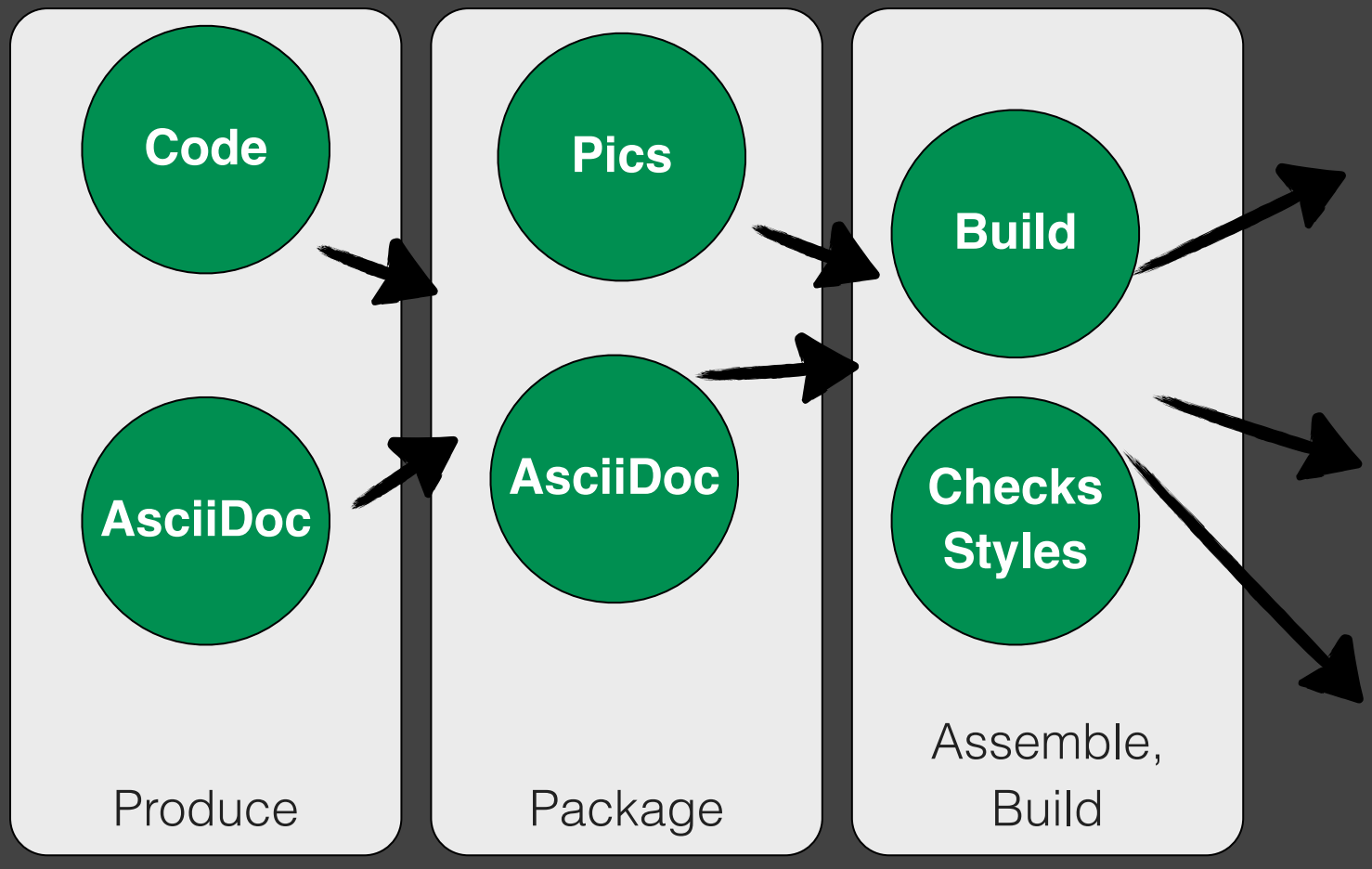
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

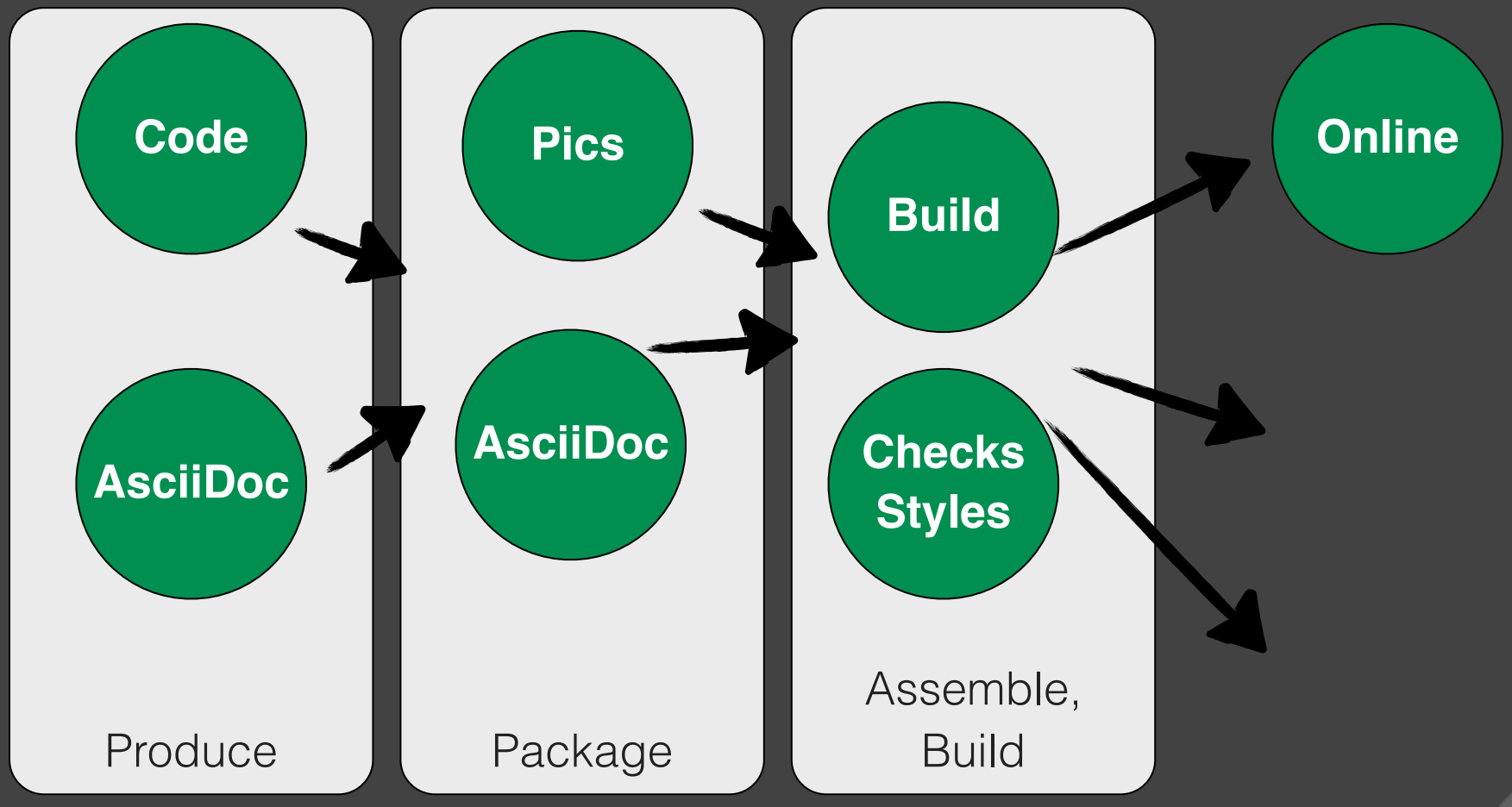
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

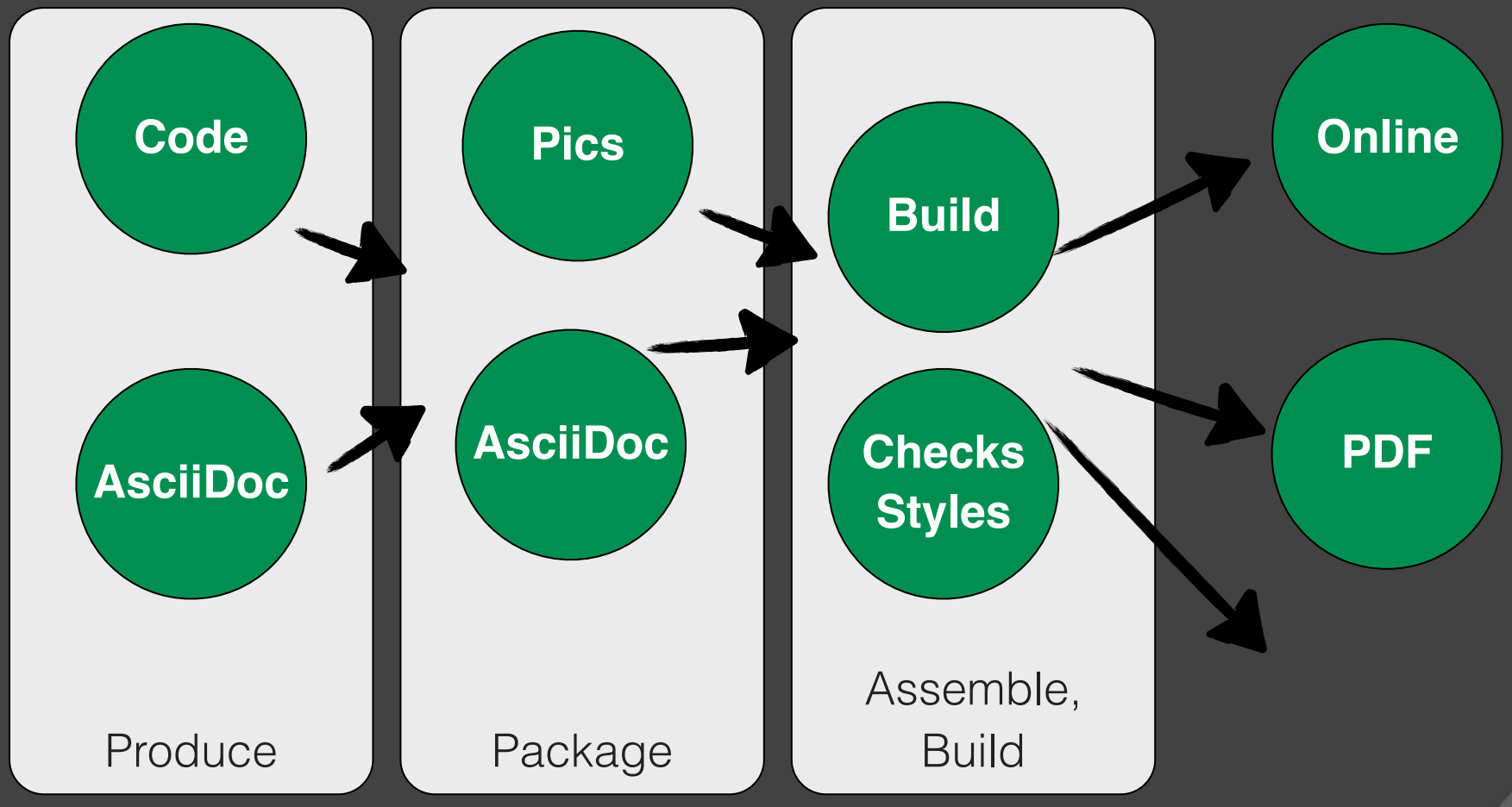
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

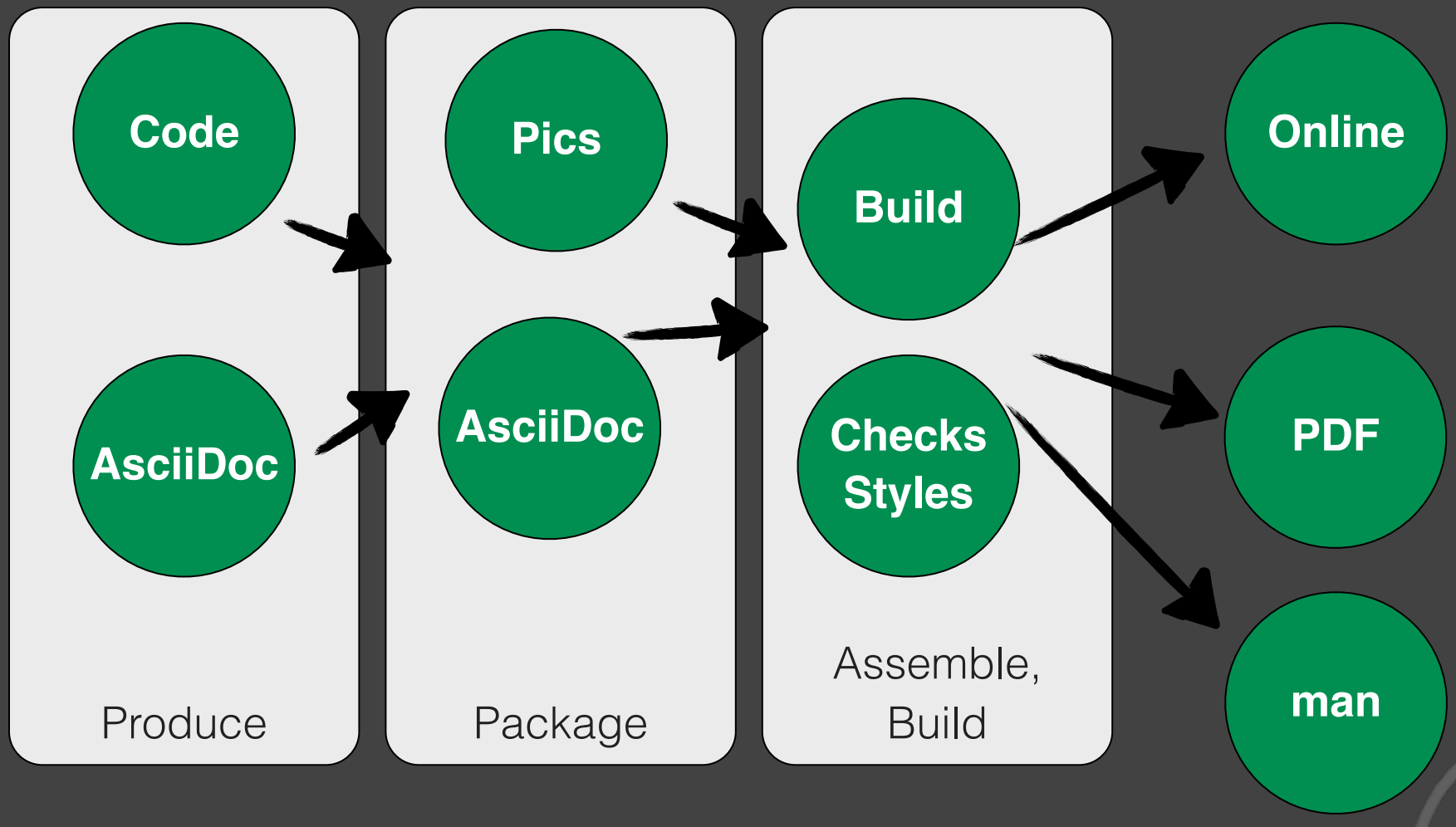
The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The docs toolchain



(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Docs #1

```scala
def graphDescription = List("A KNOWS B", "A KNOWS C", "A KNOWS D")


override val properties: Map[String, Map[String, Any]] = Map(
  "A" -> Map("property" -> 13),
  "B" -> Map("property" -> 33, "eyes" -> "blue"),
  "C" -> Map("property" -> 44, "eyes" -> "blue"),
  "D" -> Map("eyes" -> "brown")
)
```

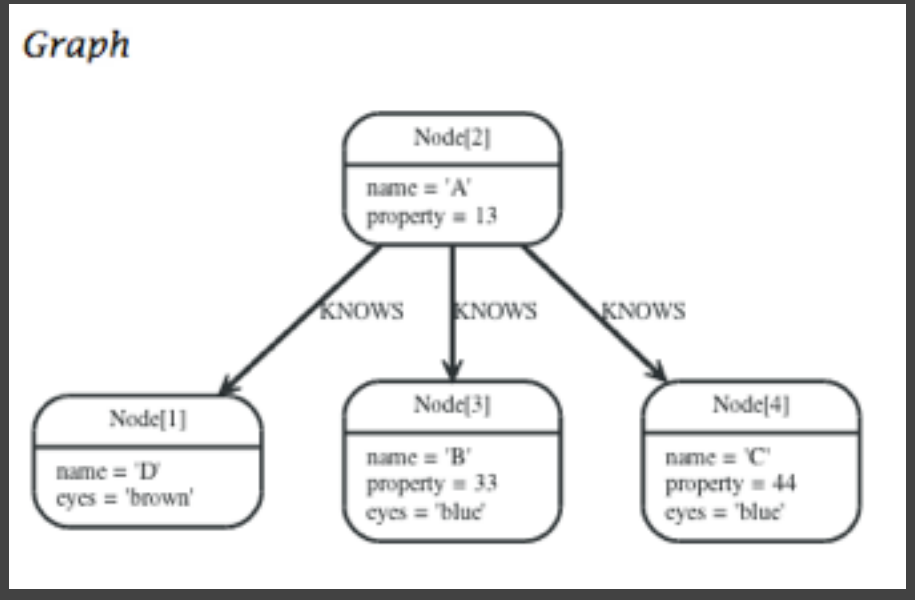
```scala
@Test def countRelationshipsByType() {
  testQuery(
    title = "Group Count Relationship Types",
    text = "To count the groups of relationship types, return the types and count them with `count(*)`.",
    queryText = "start n=node(%A%) match (n)-[r]->() return type(r), count(*)",
    returns = "The relationship types and their group count.",
    assertions = p => assertEquals(Map("type(r)" -> "KNOWS", "count(*)" -> 3), p.toList.head))
}
```

https://github.com/neo4j/community/blob/master/cypher/src/test/scala/org/neo4j/cypher/docgen/AggregationTest.scala

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Docs #2



16.12.4. Group Count Relationship Types

To count the groups of relationship types, return the types and count them with `count(*)`.

Query

```
START n=node(2)
MATCH (n)-[r]->()
RETURN type(r), count(*)
```

The relationship types and their group count.

Result

type(r)	count(*)
"KNOWS"	3
1 row	
0 ms	

Try this query live

```
(2)-[:KNOWS]->(3) {}
(2)-[:KNOWS]->(4) {}
(2)-[:KNOWS]->(1) {}

start n=node(2)
match (n)-[r]->()
return type(r), count(*)
+--------------------+
| type(r) | count(*) |
+--------------------+
| "KNOWS" | 3        |
+--------------------+
1 row
0 ms
```

You can modify and query this graph by entering statements in the input field at the bottom.
For some syntax help hit the button. If you want to share your graph, just do it with

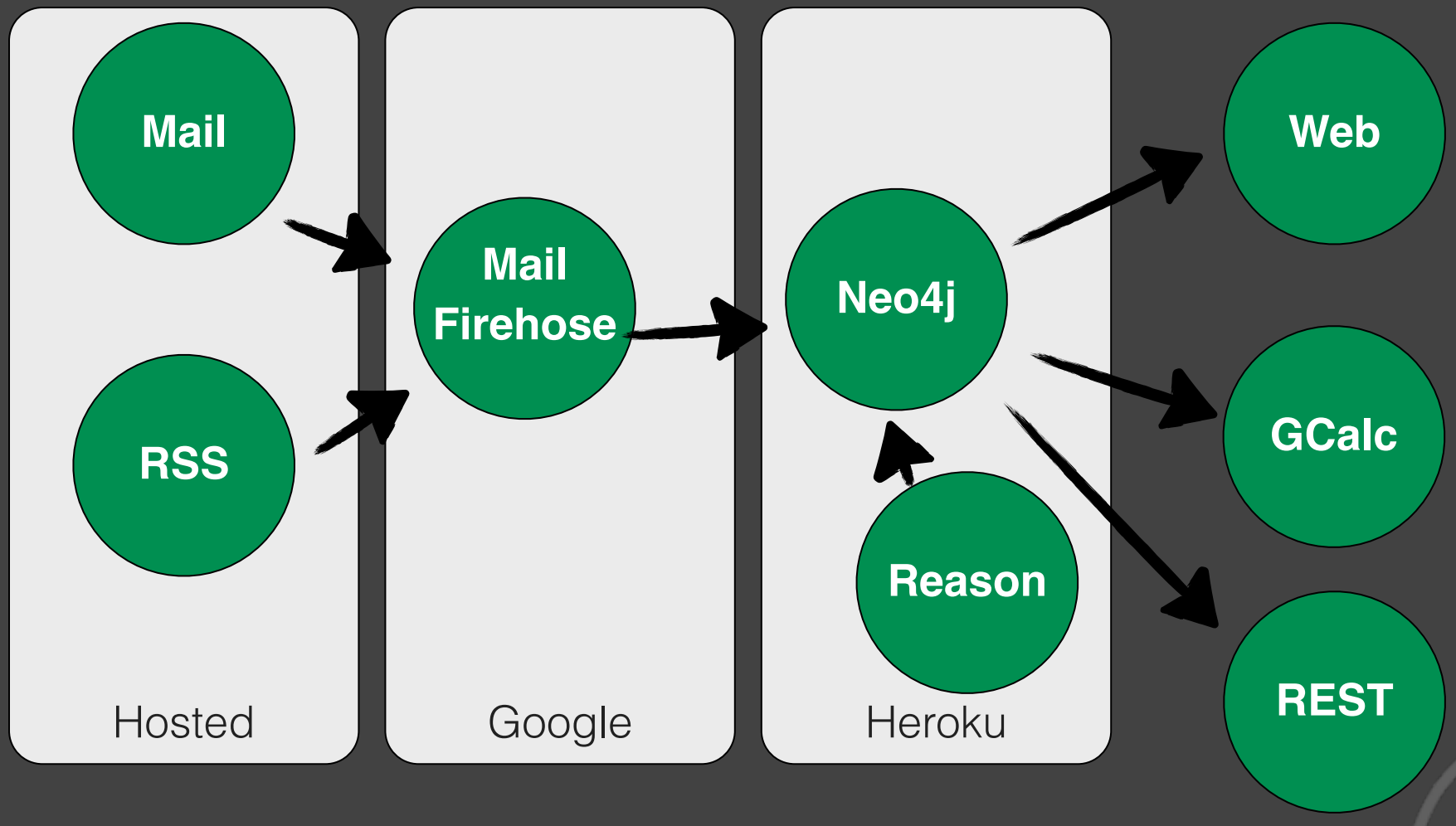
```
start n=node(2) match (n)-[r]->() return type(r), count(*)
```

http://docs.neo4j.org/chunked/snapshot/query-aggregation.html

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The Community graph

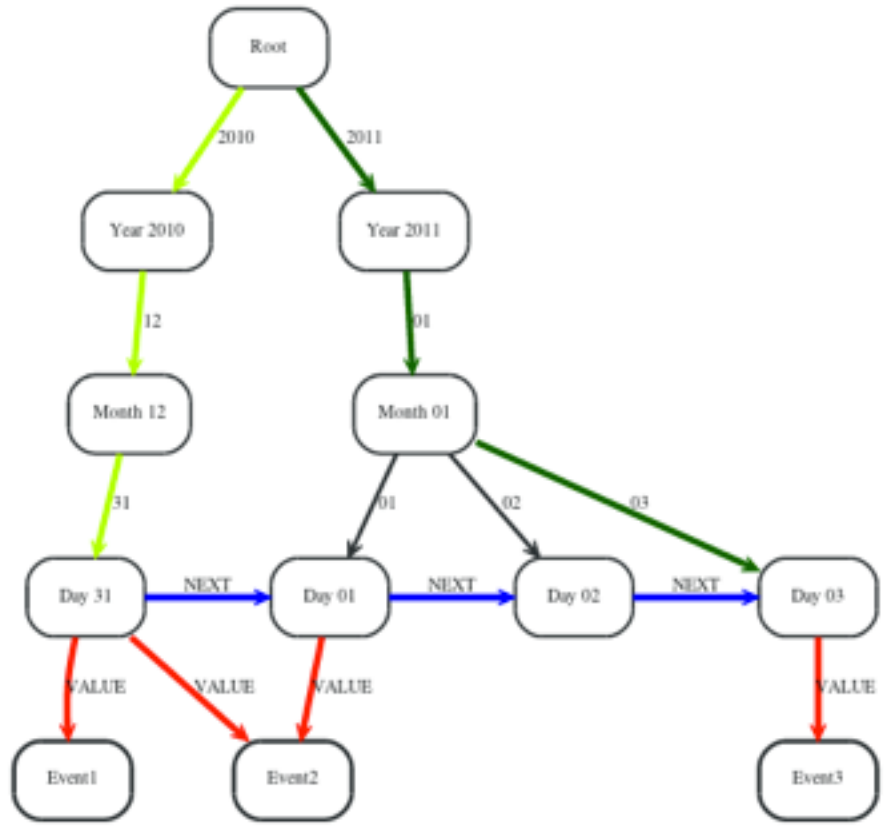


(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The Community graph

Graph



Query

```
START root=node:node_auto_index(name = 'Root')
MATCH startPath=root-[:`2010`]->()-[:`12`]->()-[:`31`]->startLeaf,
      endPath=root-[`:2011`]->()-[:`01`]->()-[:`03`]->endLeaf,
      valuePath=startLeaf-[:NEXT*0..]->middle-[:NEXT*0..]->endLeaf,
      values=middle-[:VALUE]->event
RETURN event.name
ORDER BY event.name ASC
```

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The Community graph

=getTopContentInMonth(A1,200)

A	B	C
May 1, 2012		
=getTopContentInMonth(A1,200)	base.name	count(*)
https://groups.google.com/d/topic/orient-database/pdzgSuCfoSM/discussion	https://groups.googl	48
http://www.springsource.org/node/3547	http://www.springso	30
http://www.meetup.com/sv-jug/events/63554162	http://www.meetup.c	27
http://info.neotechnology.com/0510-cloud.html	http://info.neotechnc	26

```
function getTopContentInMonth(date, limit) {
  var query = "start root=node(0) match "+
    "root-[:CATEGORY]->category-[:TWITTER_USER]->user-[:POSTED]->value<-[:LINKED]-link<-[:BASE_UR
    "value<-[:VALUE]-hour<-[h]-day<-[d]-month<-[:IDX_MONTH_"+
      lpad((date.getMonth()+1),2)+"]-year<-[:IDX_YEAR_"+date.getYear()+"]-root " +
    "where type(d) =~/IDX_DAY_.*/ and type(h) =~/IDX_HOUR_.*/ " +
    "return link.name, base.name,count(*) order by count(*) desc limit "+limit;
  return cellify(cypherFromCG(query));
}
```

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Teach

```
Graph Setup:

start root=node(0)
create (Neo {name:'Neo'}), (Morpheus {name: 'Morpheus'}), (Trinity {name: 'Trinity'}),
 (Cypher {name: 'Cypher'}), (Smith {name: 'Agent Smith'}), (Architect {name:'The Architect'}),
 root-[:ROOT]->Neo, Neo-[:KNOWS]->Morpheus, Neo-[:LOVES]->Trinity, Morpheus-[:KNOWS]->Trinity,
 Morpheus-[:KNOWS]->Cypher, Cypher-[:KNOWS]->Smith, Smith-[:CODED_BY]->Architect
--------------------------------------------------------------------------------

start n=node(*)
match n-[r?]->m
return n,type(r),m
+-------------------------------------------------------------------------------+
| n                                | type(r)     | m                              |
+-------------------------------------------------------------------------------+
| Node[0]{}                        | "ROOT"      | Node[1]{name->"Neo"}           |
| Node[1]{name->"Neo"}             | "KNOWS"     | Node[2]{name->"Morpheus"}      |
| Node[1]{name->"Neo"}             | "LOVES"     | Node[3]{name->"Trinity"}       |
| Node[2]{name->"Morpheus"}        | "KNOWS"     | Node[3]{name->"Trinity"}       |
| Node[2]{name->"Morpheus"}        | "KNOWS"     | Node[4]{name->"Cypher"}        |
| Node[3]{name->"Trinity"}         | <null>      | <null>                         |
| Node[4]{name->"Cypher"}          | "KNOWS"     | Node[5]{name->"Agent Smith"}   |
| Node[5]{name->"Agent Smith"}     | "CODED_BY"  | Node[6]{name->"The Architect"} |
| Node[6]{name->"The Architect"}   | <null>      | <null>                         |
+-------------------------------------------------------------------------------+
9 rows
0 ms
```

You can modify and query this graph by entering statements in the input field at the bottom.
For some syntax help hit the button. If you want to share your graph, just do it with

```
start n=node(*) match n-[r?]->m return n,type(r),m
```

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Engage

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Empower (Mattis)

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Empower (Mattis)

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Promote, Support

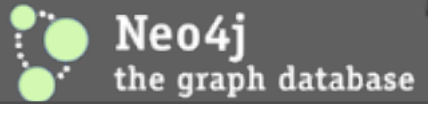
Till, Rene, Paul, http://www.rene-pickhardt.de/typology-using-neo4j-wins-2-awards-at-the-german-federal-competition-young-scientists/

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Summary

(friends)-[:create]->(project)-[:supports]->(community)



Summary

Have a cause

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Summary

Have a cause
Be brave

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Summary

Have a cause
Be brave
Value in Relationships

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

Summary

Have a cause
Be brave
Value in Relationships
Have fun!

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database

The (G)Raffle

http://graffle-goto-cph.herokuapp.com/

(friends)-[:create]->(project)-[:supports]->(community)

Neo4j
the graph database


Questions?