

# Düğümküme

Fatih Erikli

[fatiherikli@gmail.com](mailto:fatiherikli@gmail.com)

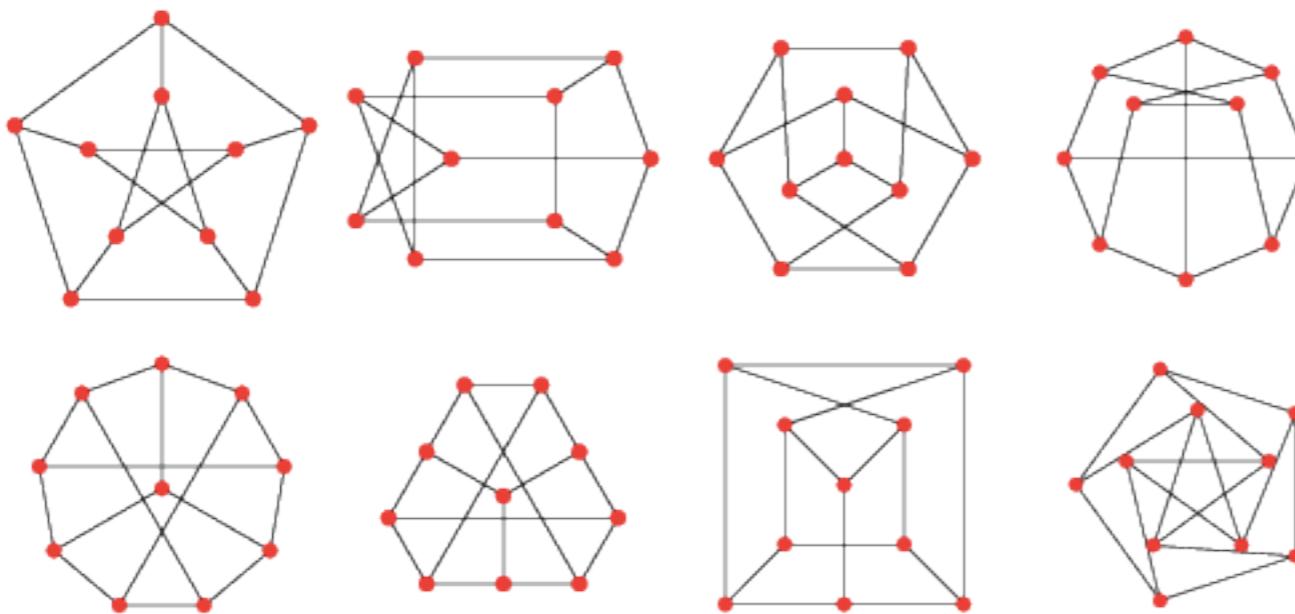
<http://fatiherikli.com>

---

software developer at [adphorus](#)

founder of [arguman.org](#)

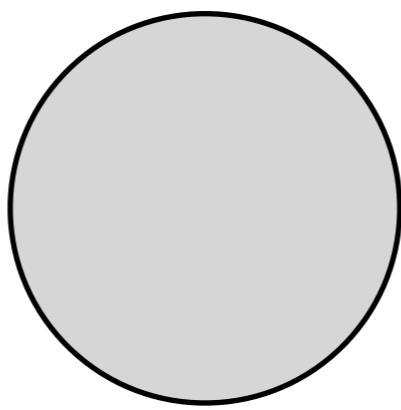
$\text{ağ} = \text{graph}(\text{düğümküme}^*)$



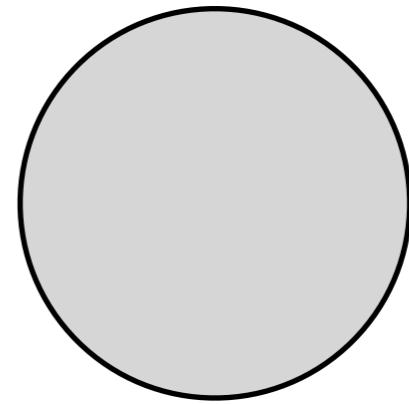
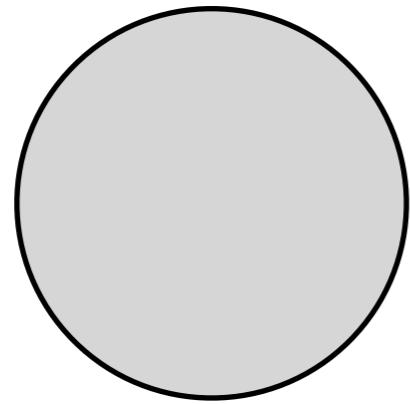
bir `graph` düğümler ve ilişkiler  
kümesidir.

\*söylem: <http://dugumkume.org/dugumkume>

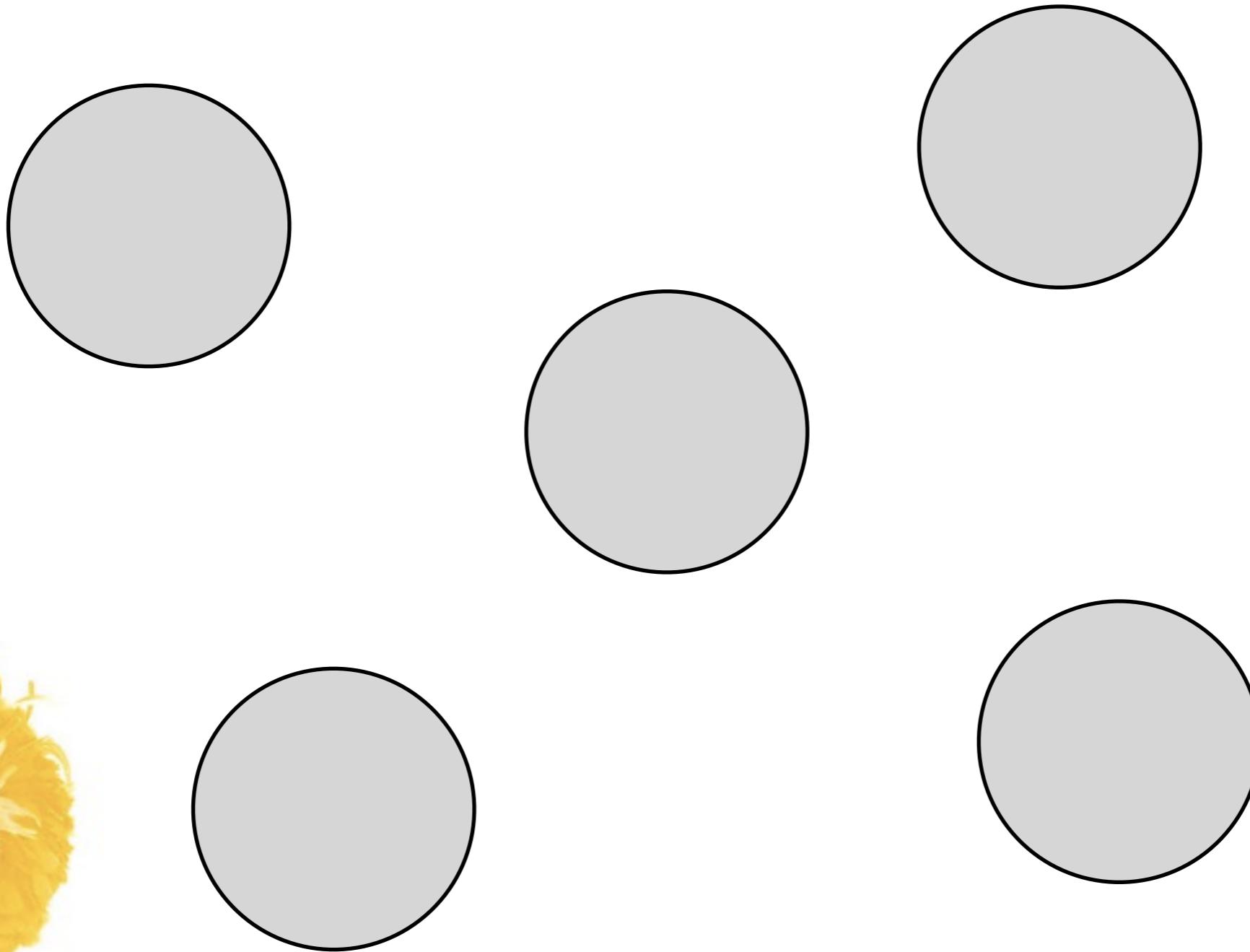
# düğüm



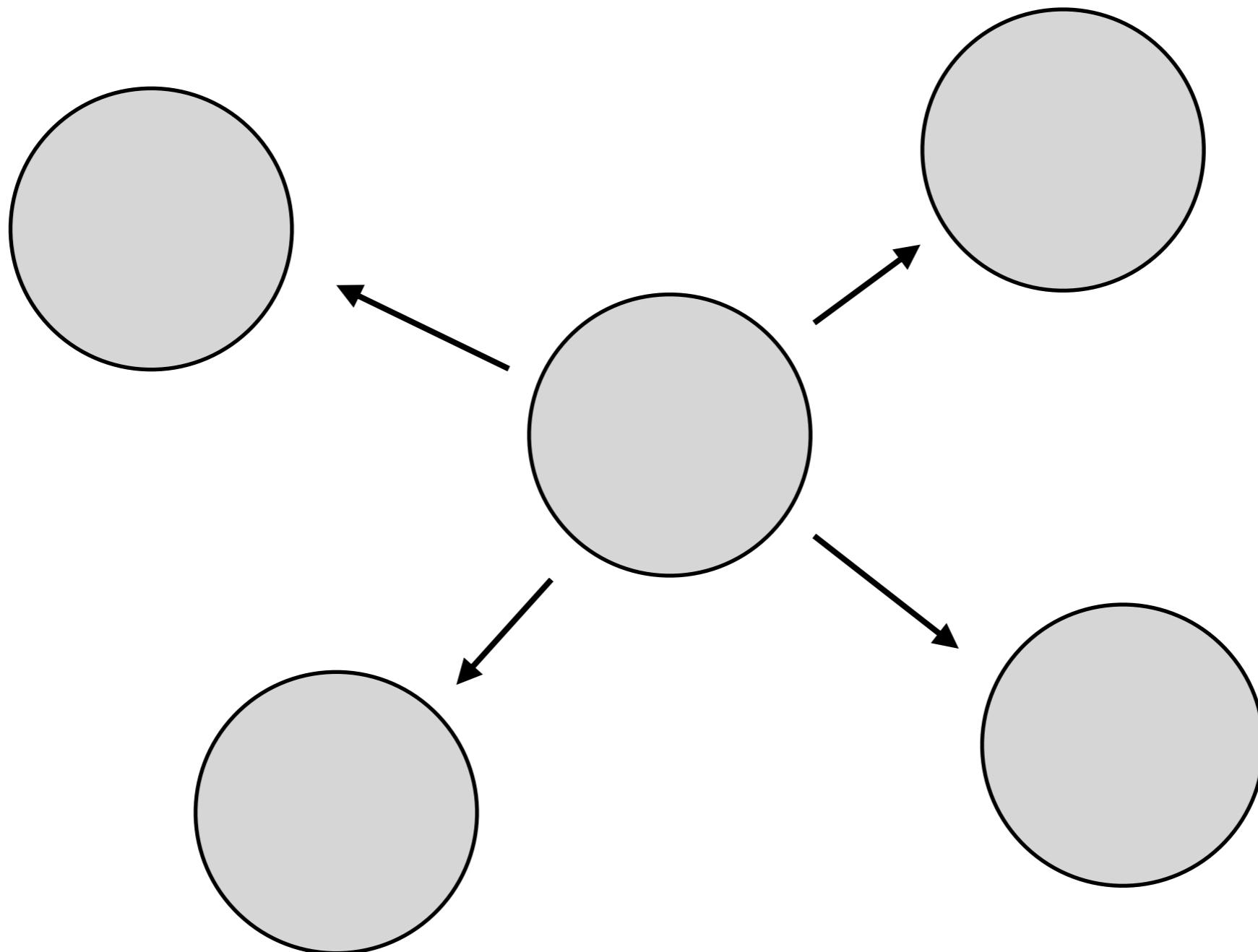
# düğümler



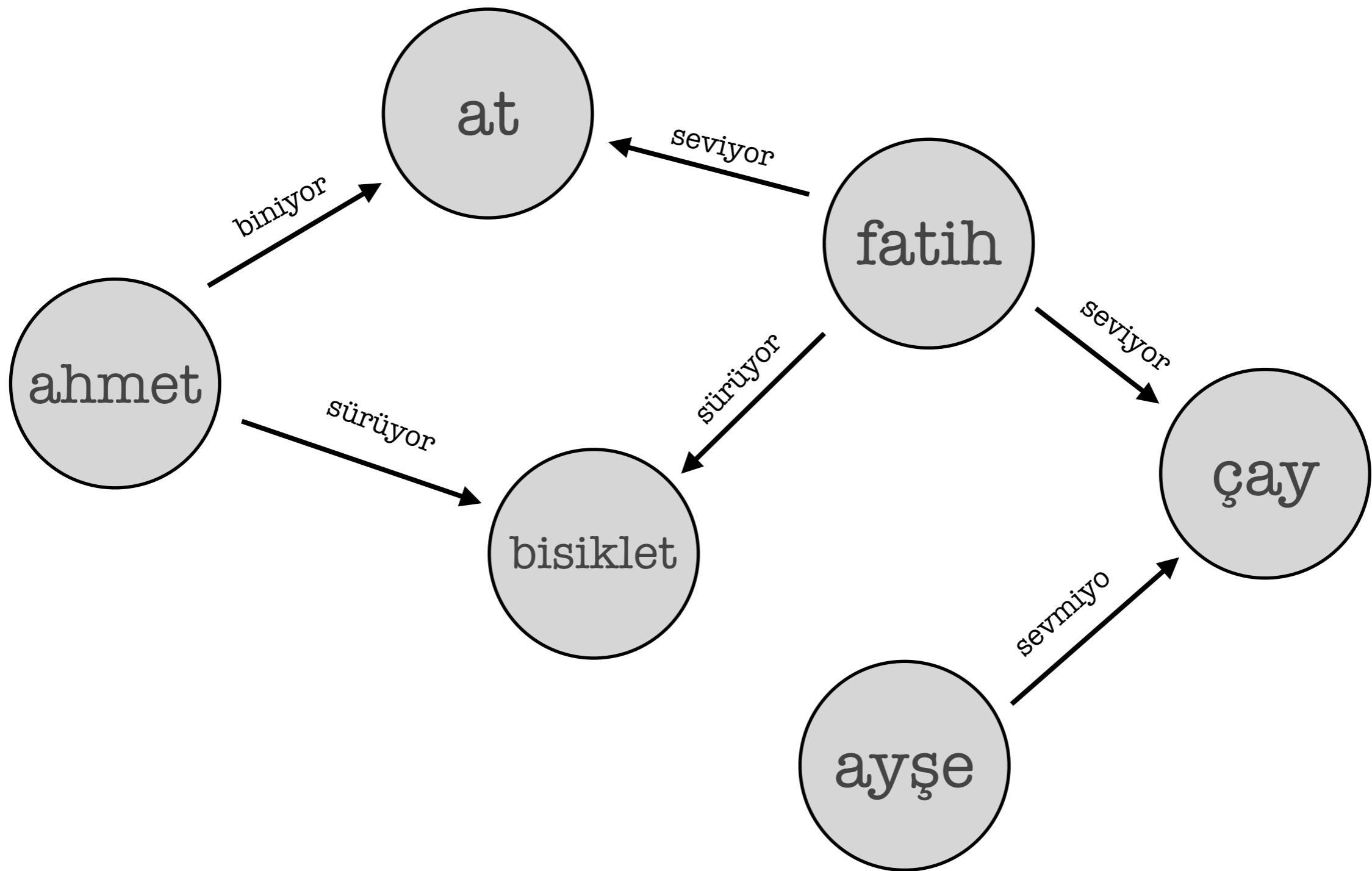
# birçok düğüm



# ilişkiler

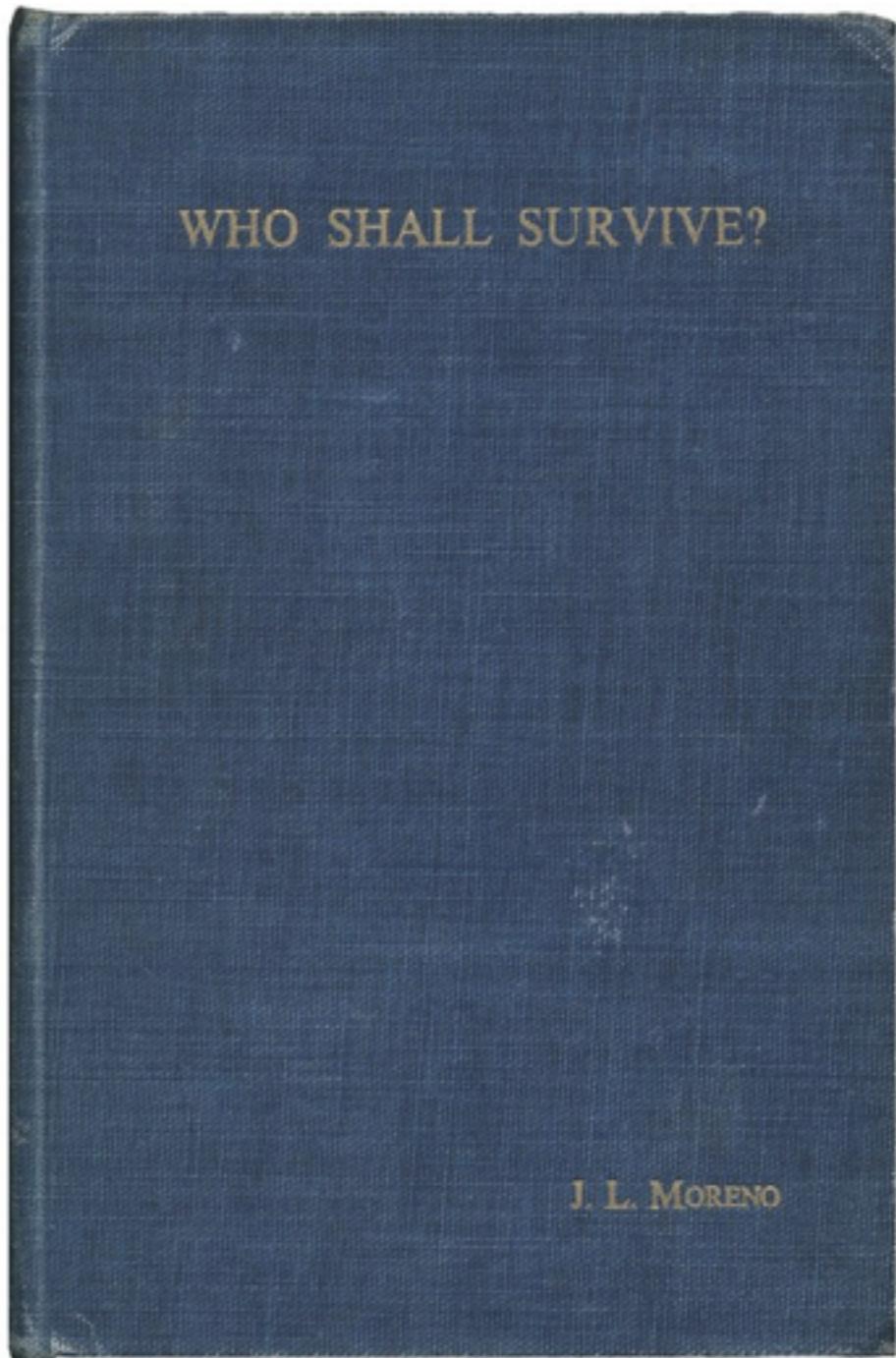


# ve bir düğümküme





Graph teorisinin temeli matematikçi Euler tarafından 1735 yılında Königsberg'in 7 Köprüsü probleminin çözümüne dayanıyor.



Sosyometri alanını öncüsü olan Moreno 1934 yılında **Who Shall Survive** kitabını yazar ve bir sosyoloji altdalını başlatmış olur:

Sosyal Ağ Analizi

# sosyometri

[f](#) [t](#) [araştır](#)

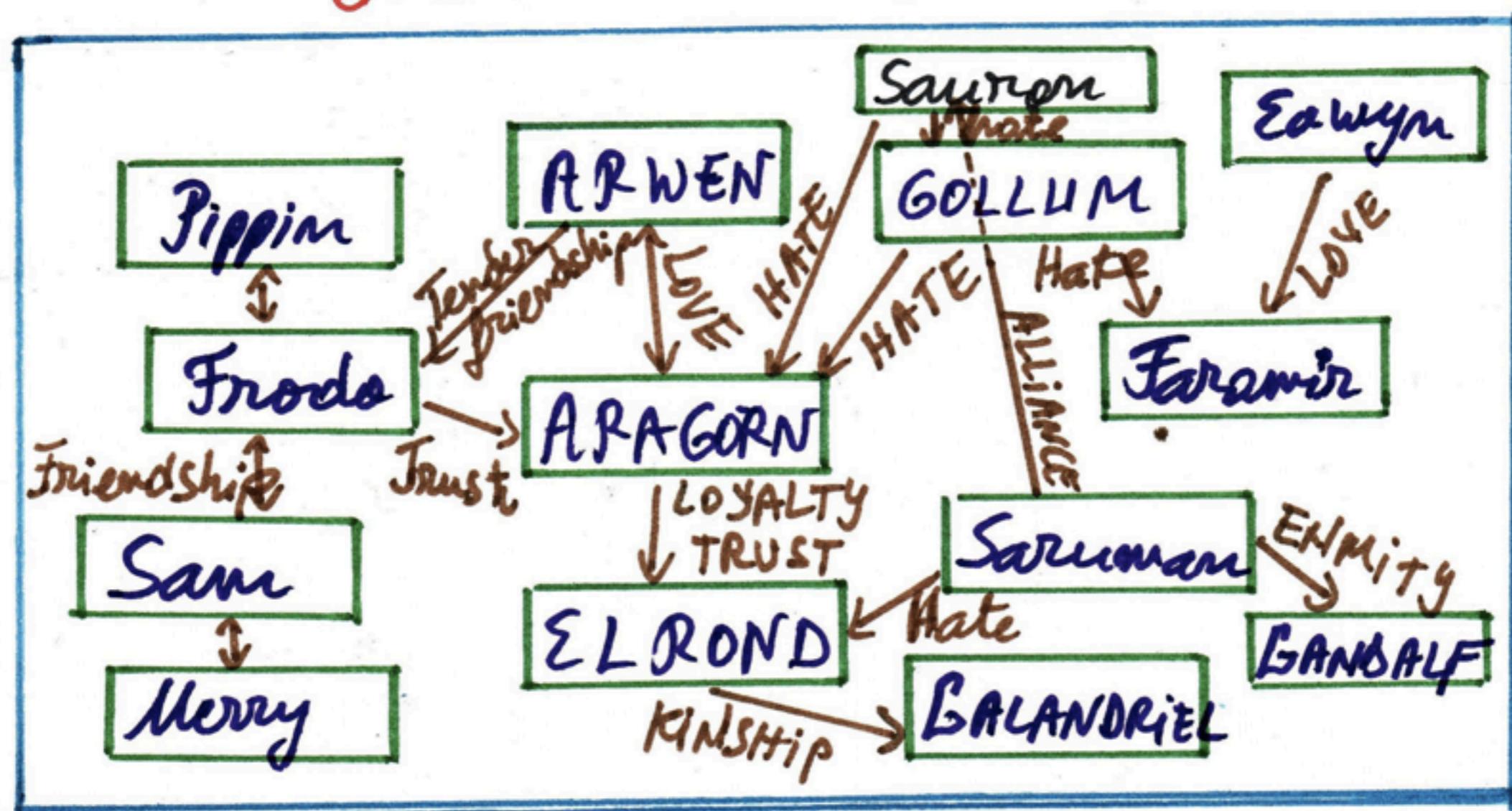
9 entry daha

grubun ilişki biçimlerini tespit edebildiği halde nedenleri hakkında bilgi veremeyen teknik. örnegin bu teknik sayesinde berkecan'ın gruptan dışlandığını öğrenebilirsiniz ama nedenini bilemezsiniz.

[f](#) [t](#) [^](#) [v](#)

#13134213 24.04.2008 14:28 [have a cigar](#) ... ▾

## Character Sociogram



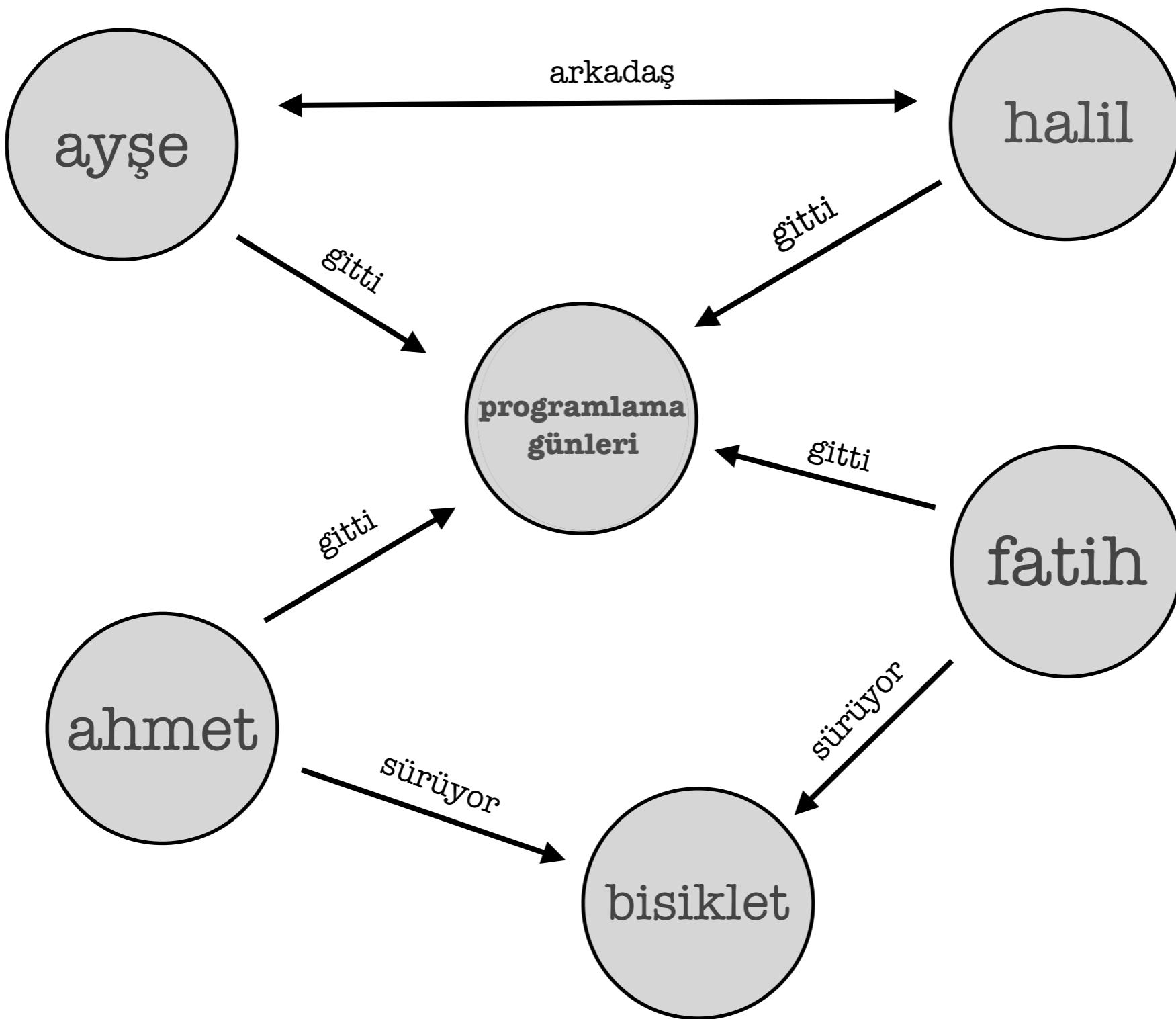
# neden popüler?

çünkü disiplinler arası.

çünkü internet.

çünkü günlük hayatı kullandığımız  
servislerin içinde aslında hepimiz birer  
düğümüz.

# facebook

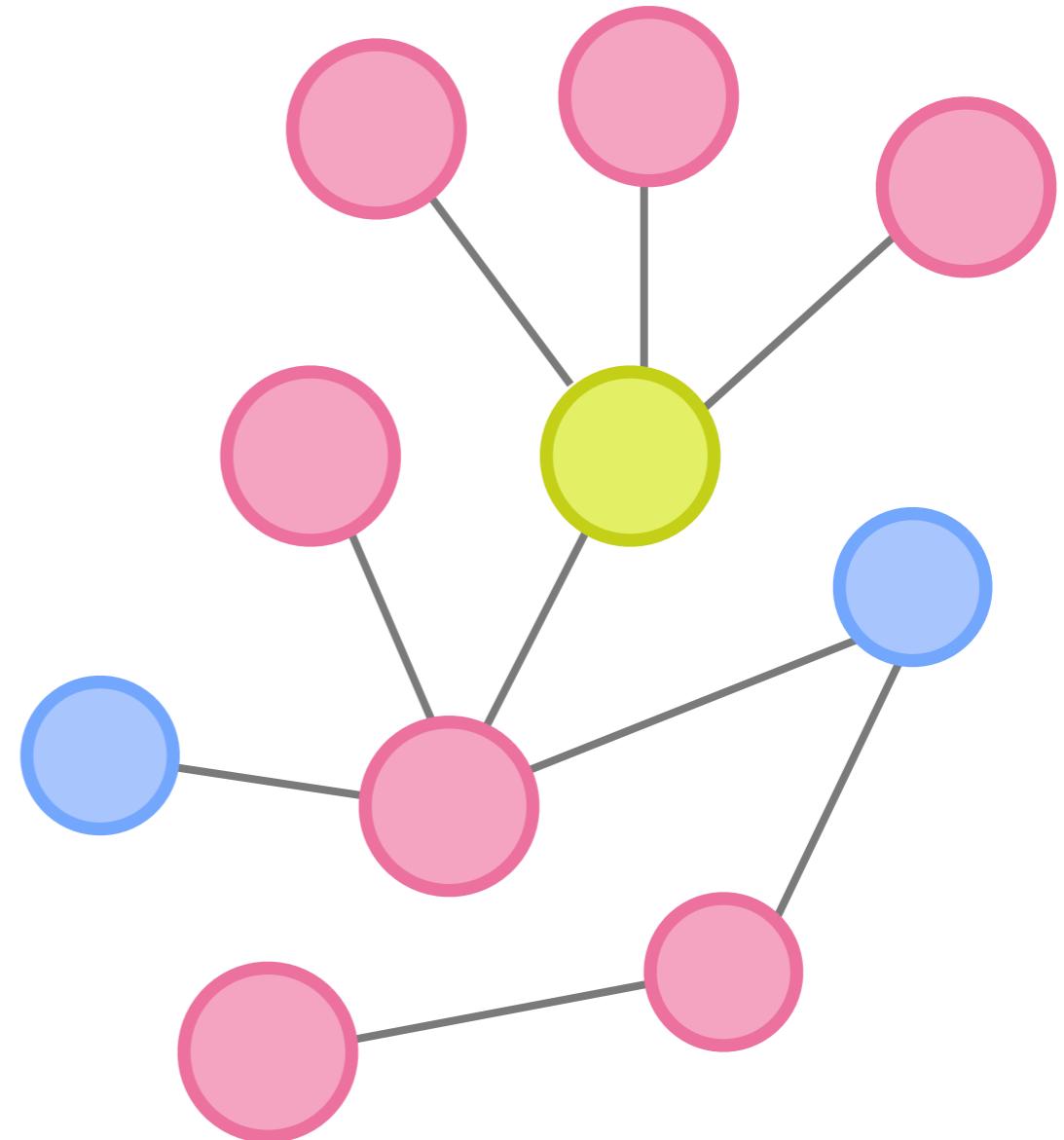




YIKIN  
TÜM  
DİKİLEN  
HEYKELLERİMİ

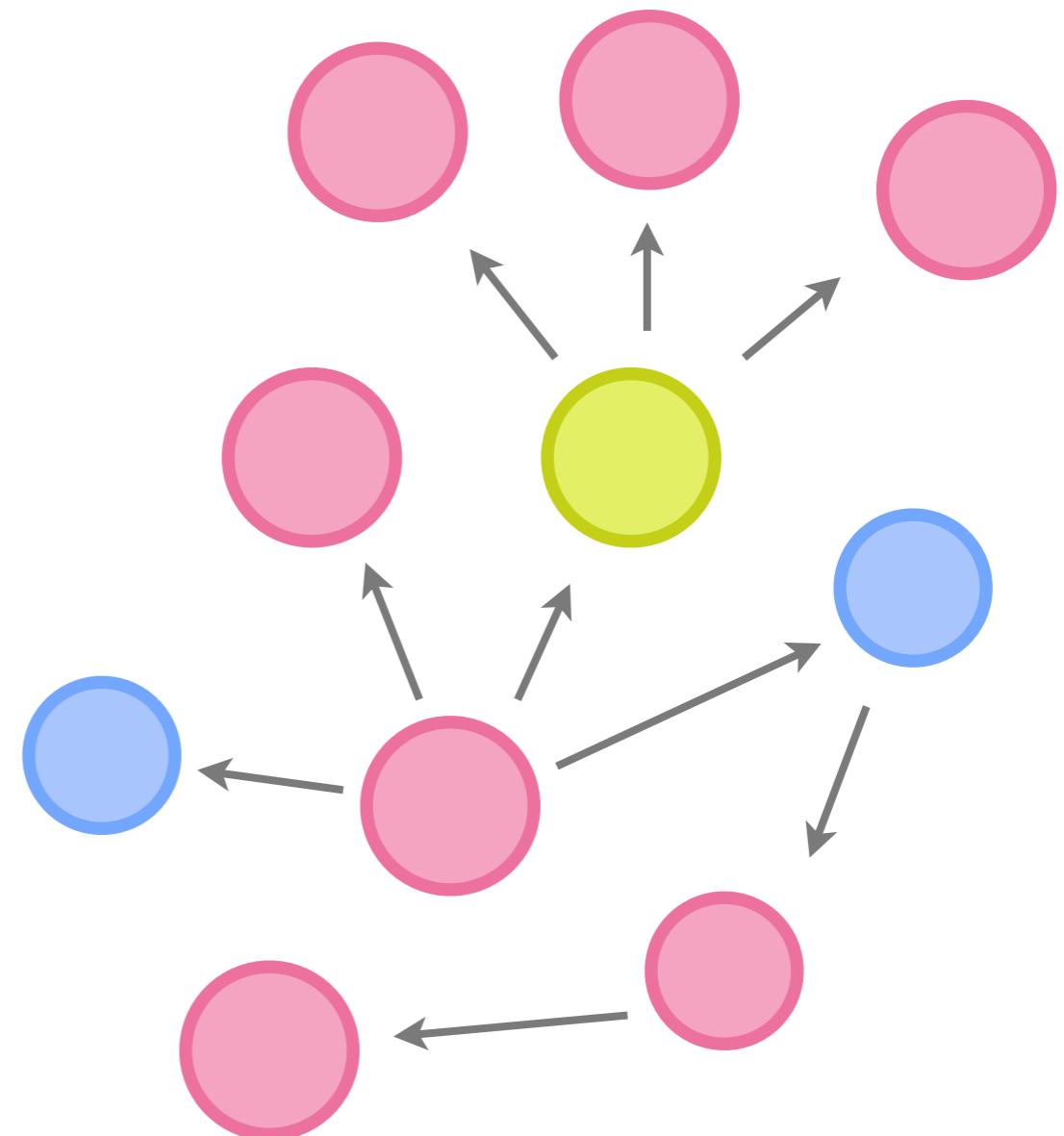
# undirected

- arkadaşlıklar
- savaşmalar
- çarpışmalar
- ulaşım ağları

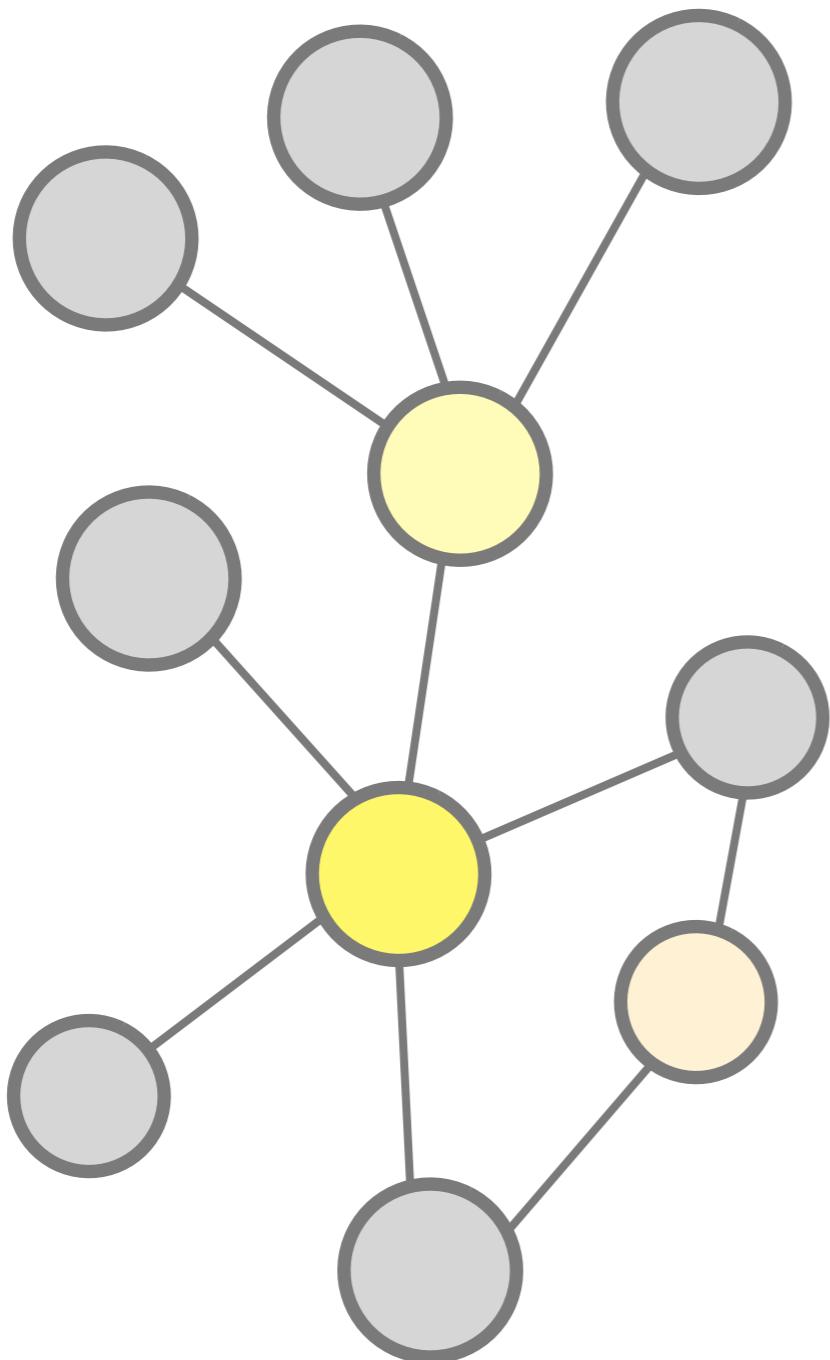


# directed

- takipçilermeler
- galibiyetler
- internet linkleri
- platonik sevgililikler

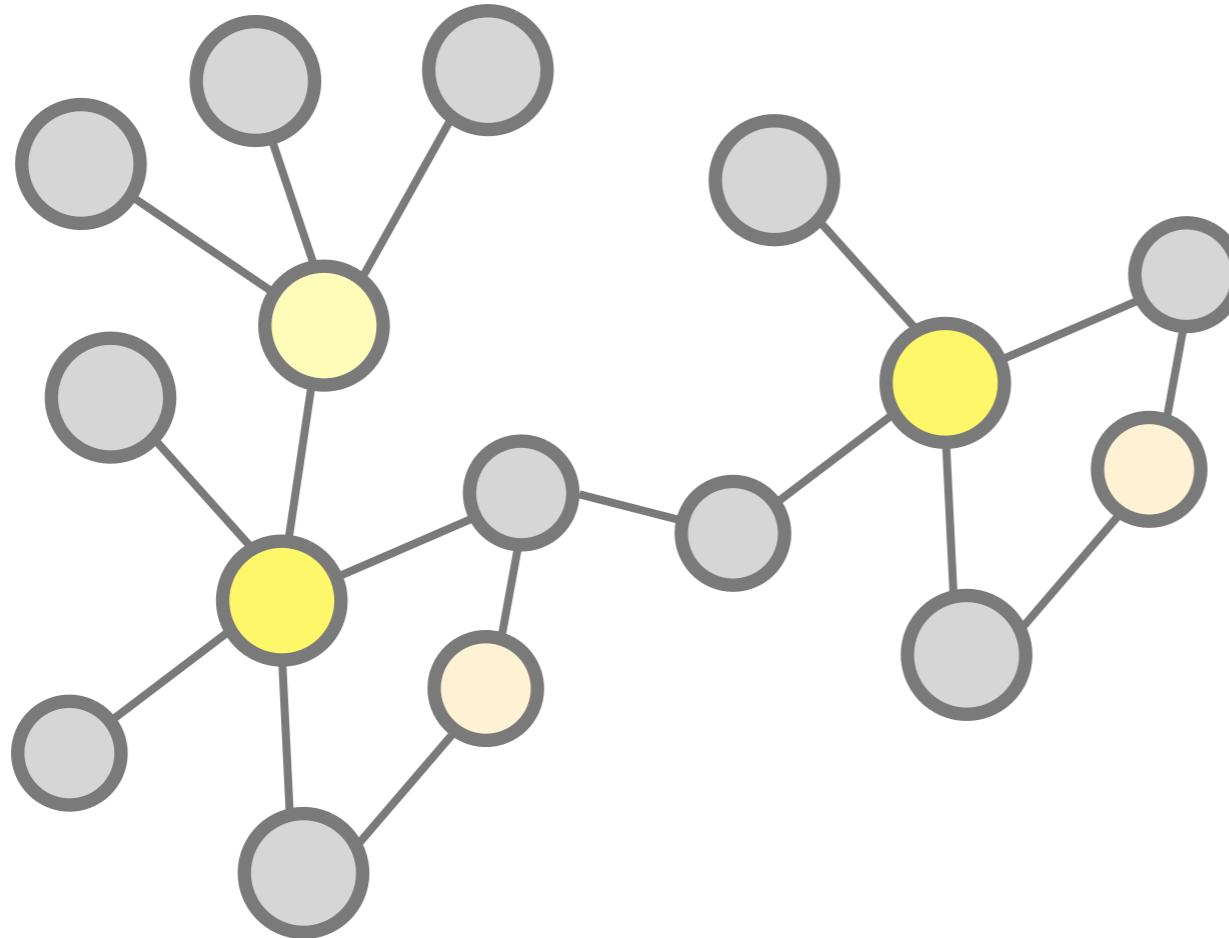


# merkeziyet

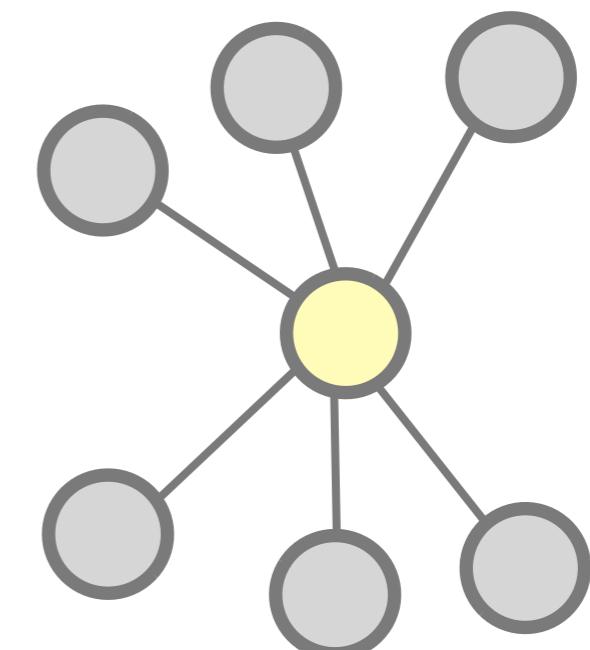


- Degree
- In degree
- Out degree
- Betweenness
- Closeness

# topluluk

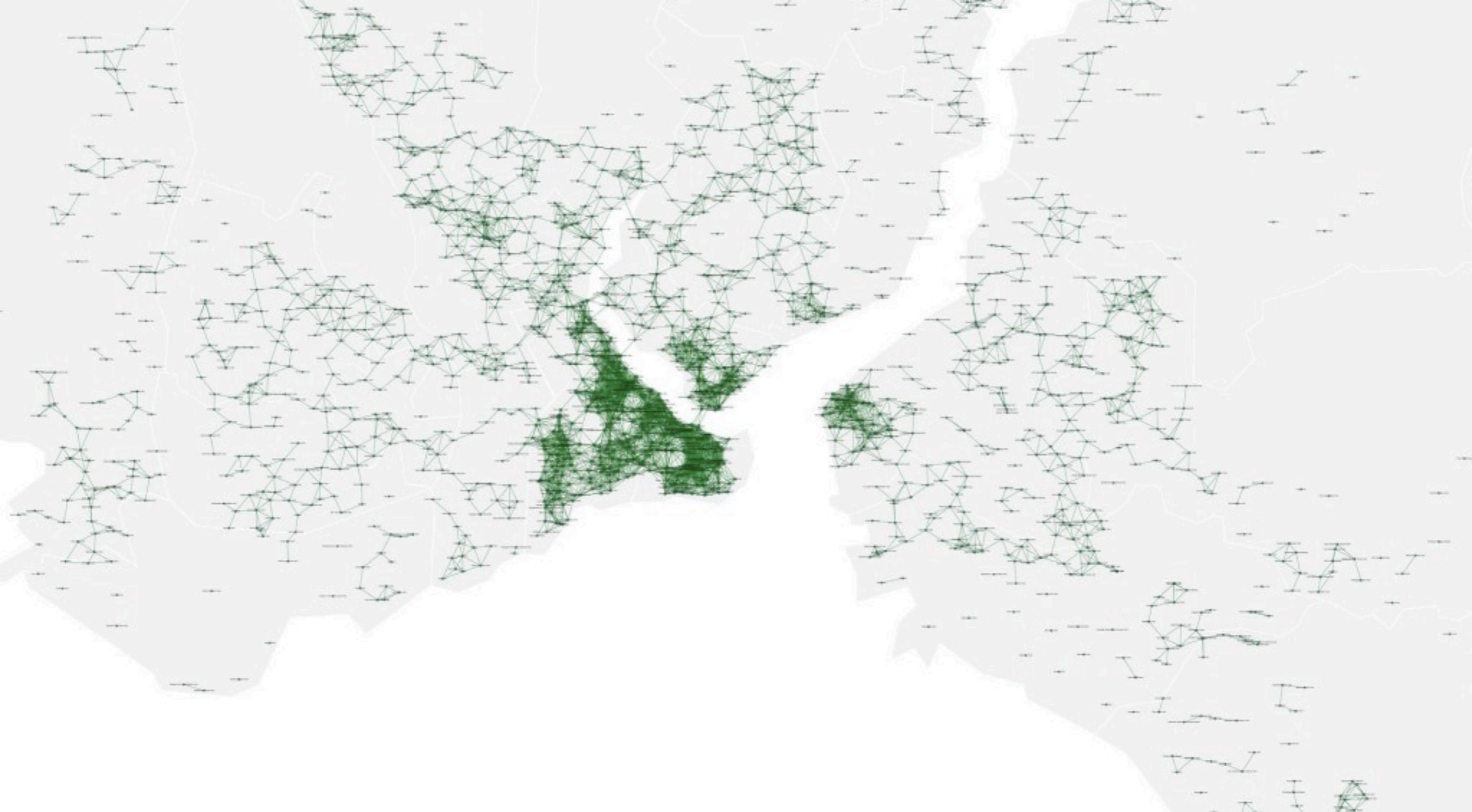


Giant  
Component



Isolated  
Component

# örnekler



İstanbul'daki camilerin örtüßen ezan sesleri üzerine kurulmuş bir harita.  
<http://burak-arikan.com/islam-republic-neoliberalism>

# programming language network

A graph of programming languages that consists with their influences, companies, developers, dialects, implementations.

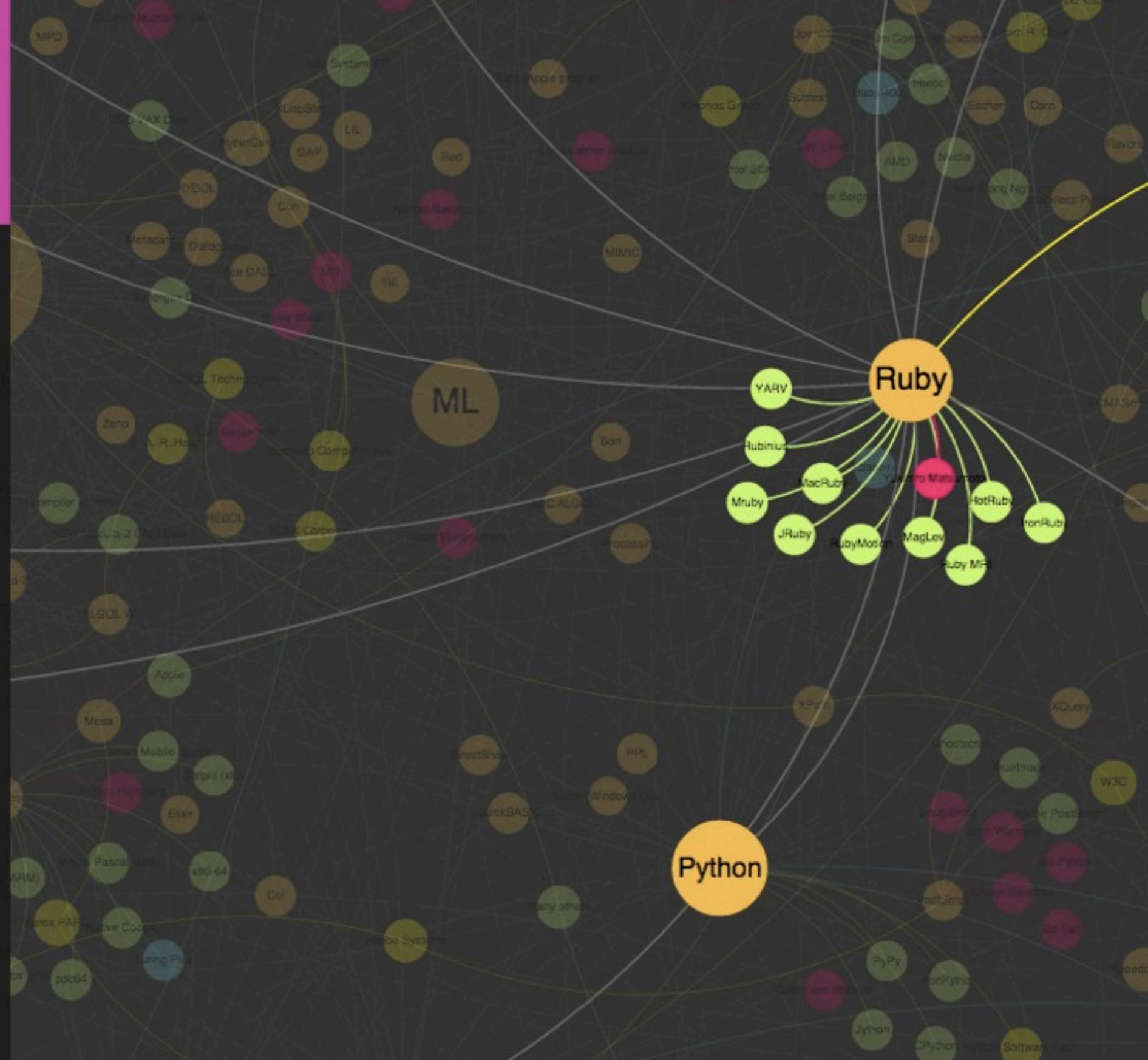
## Nodes

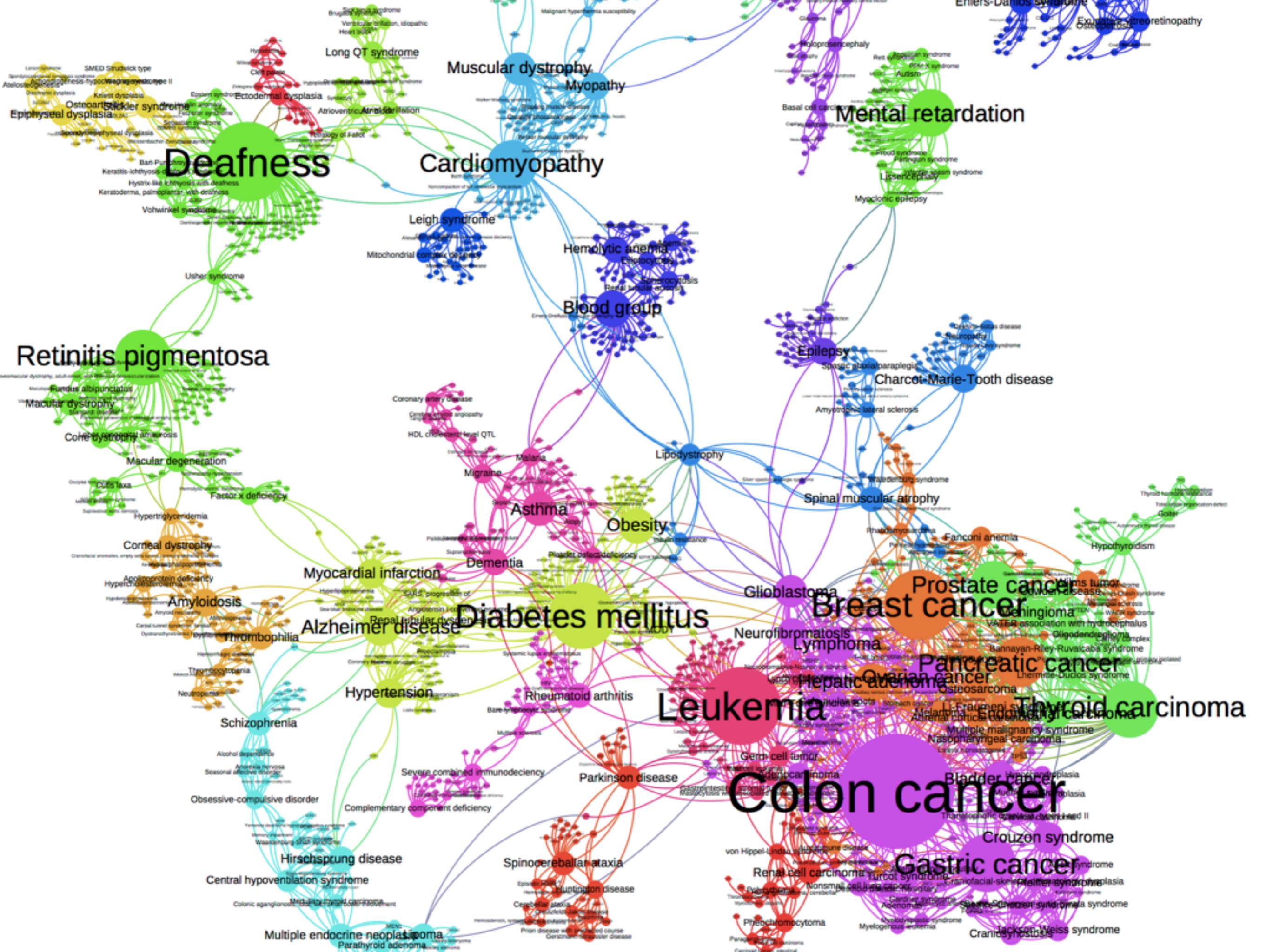
- Programming Language
- Computer Scientist
- Foundation
- Dialect
- Implementation

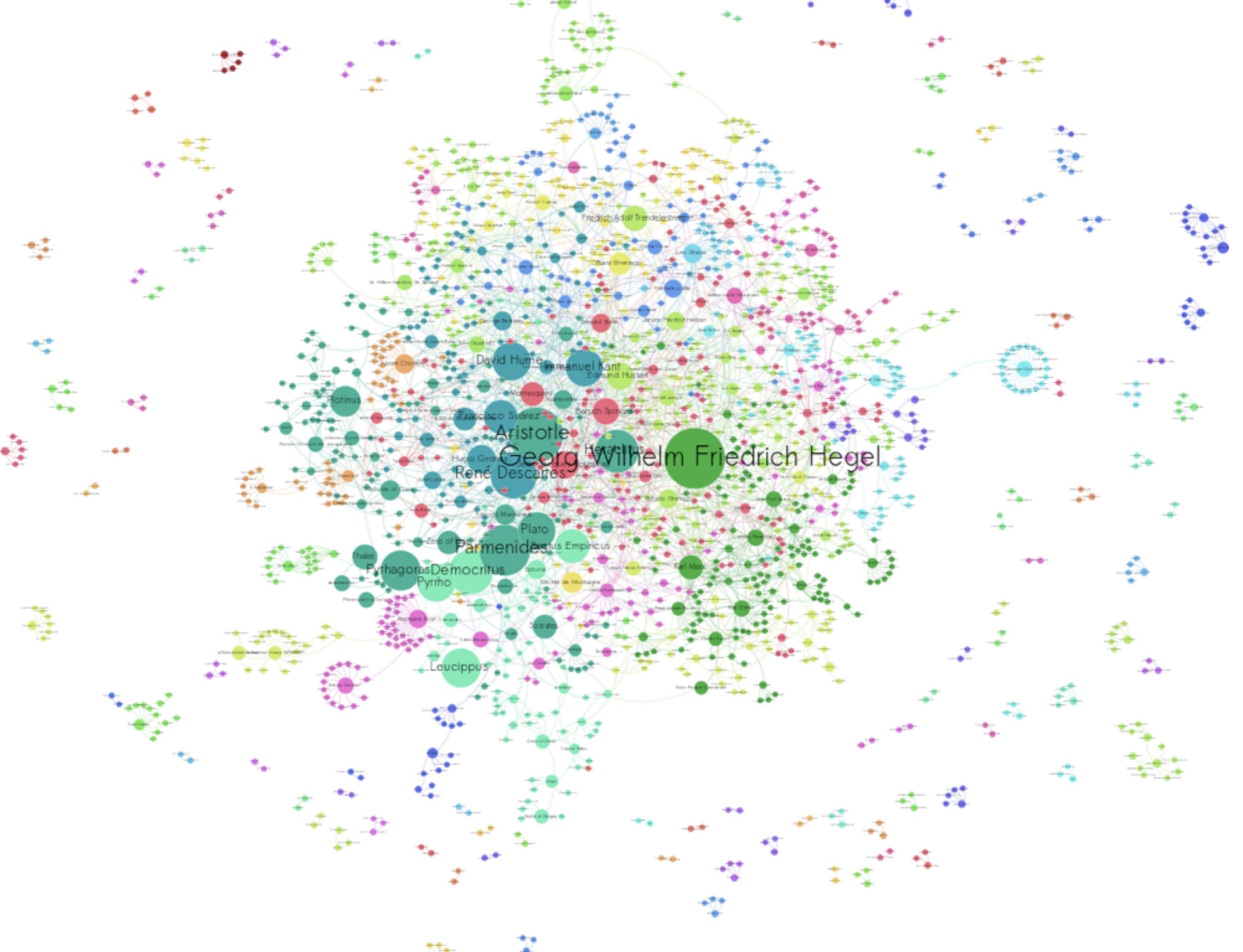
## Edges

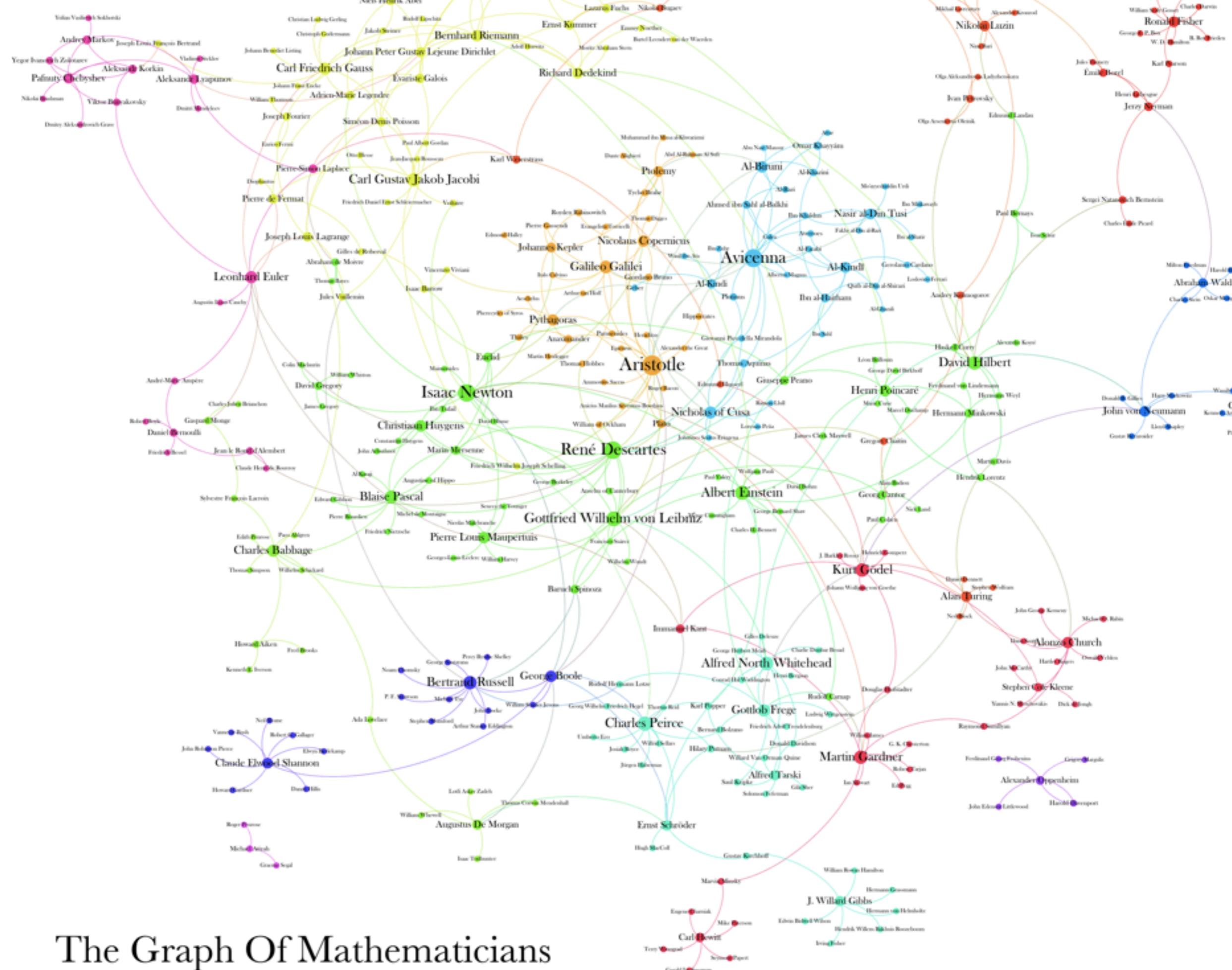
- Influenced by
- Designed by
- Developer
- Dialects
- Major implementations
- Implementation language

source code

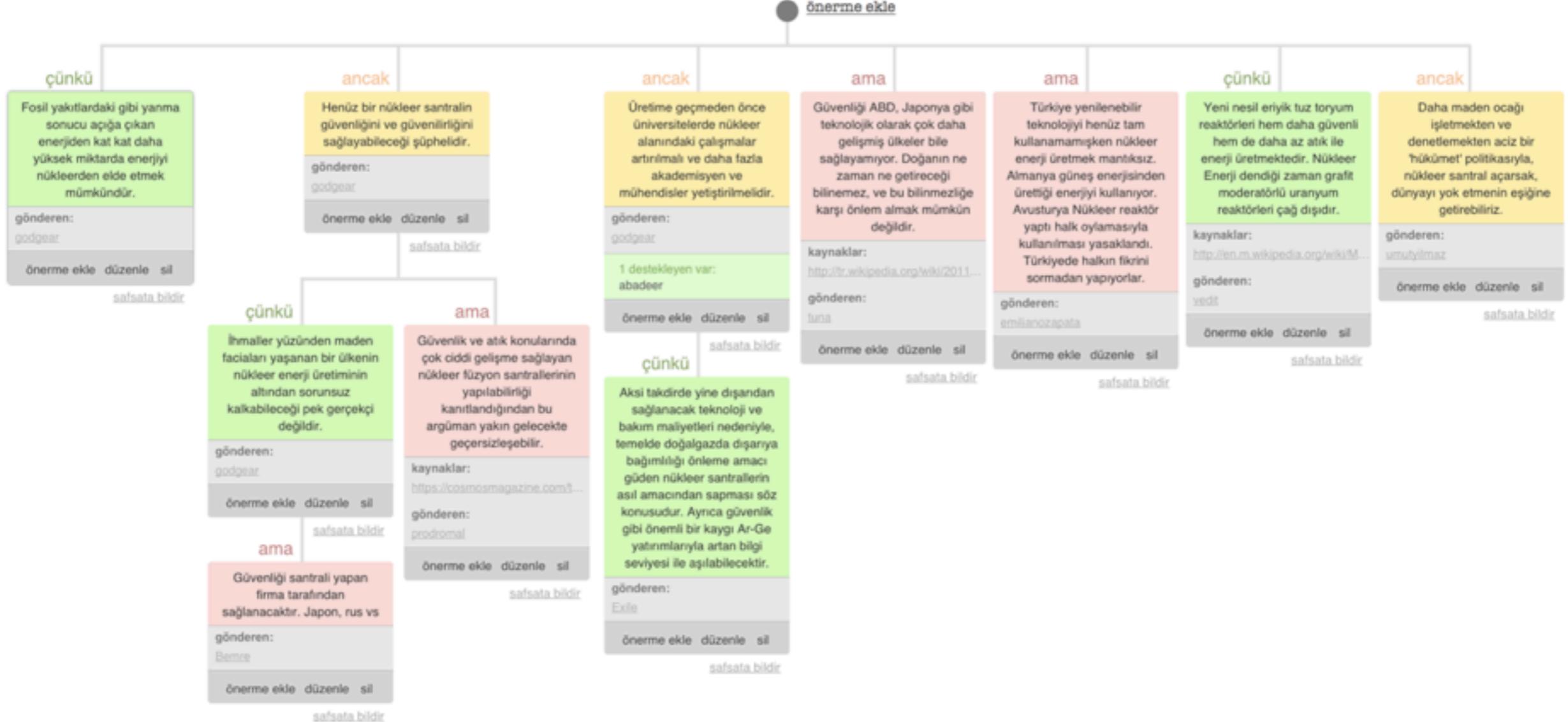








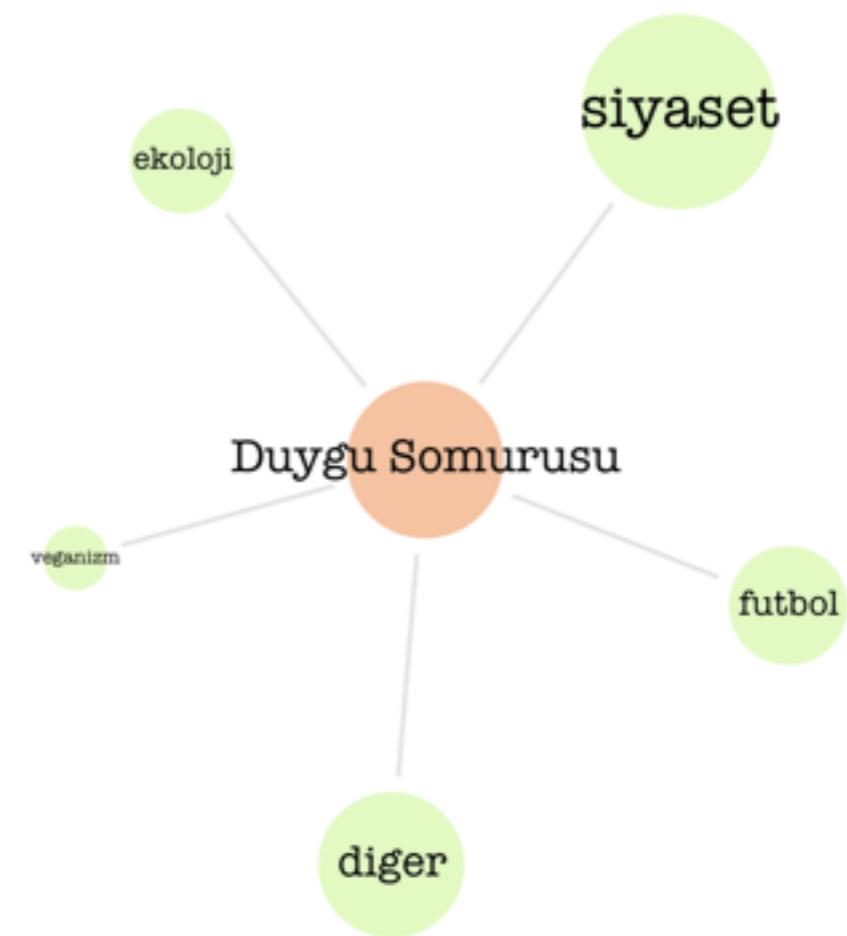
Türkiye nükleer santrallerden de enerji üretmelidir.



# argüman analizi platformu

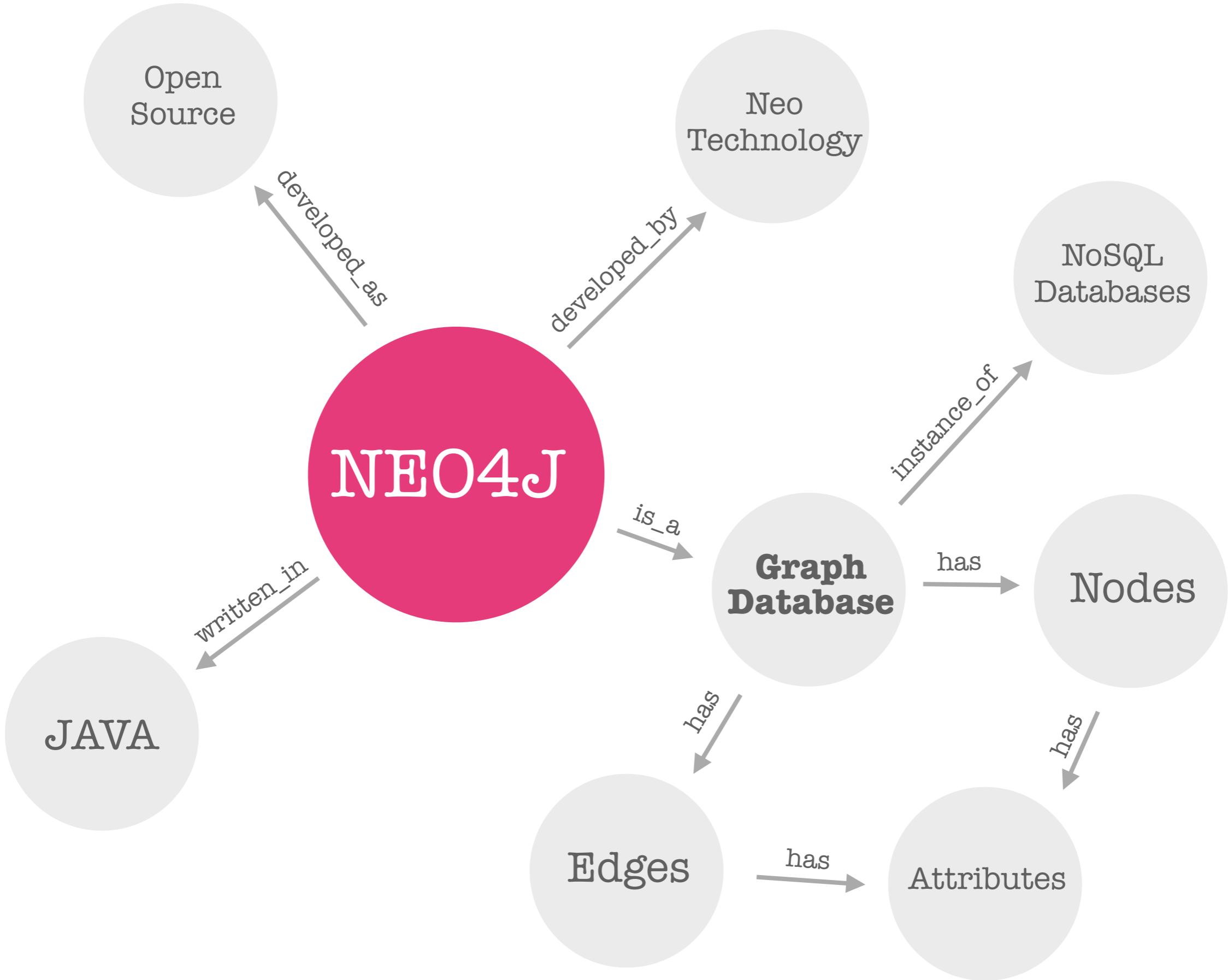
<https://arguman.org>

argüman haritaları hiyerarşik düğümkümelere rdir. önermeler birer node (düğüm), önerme tipleri ise bir edge (ilişki).

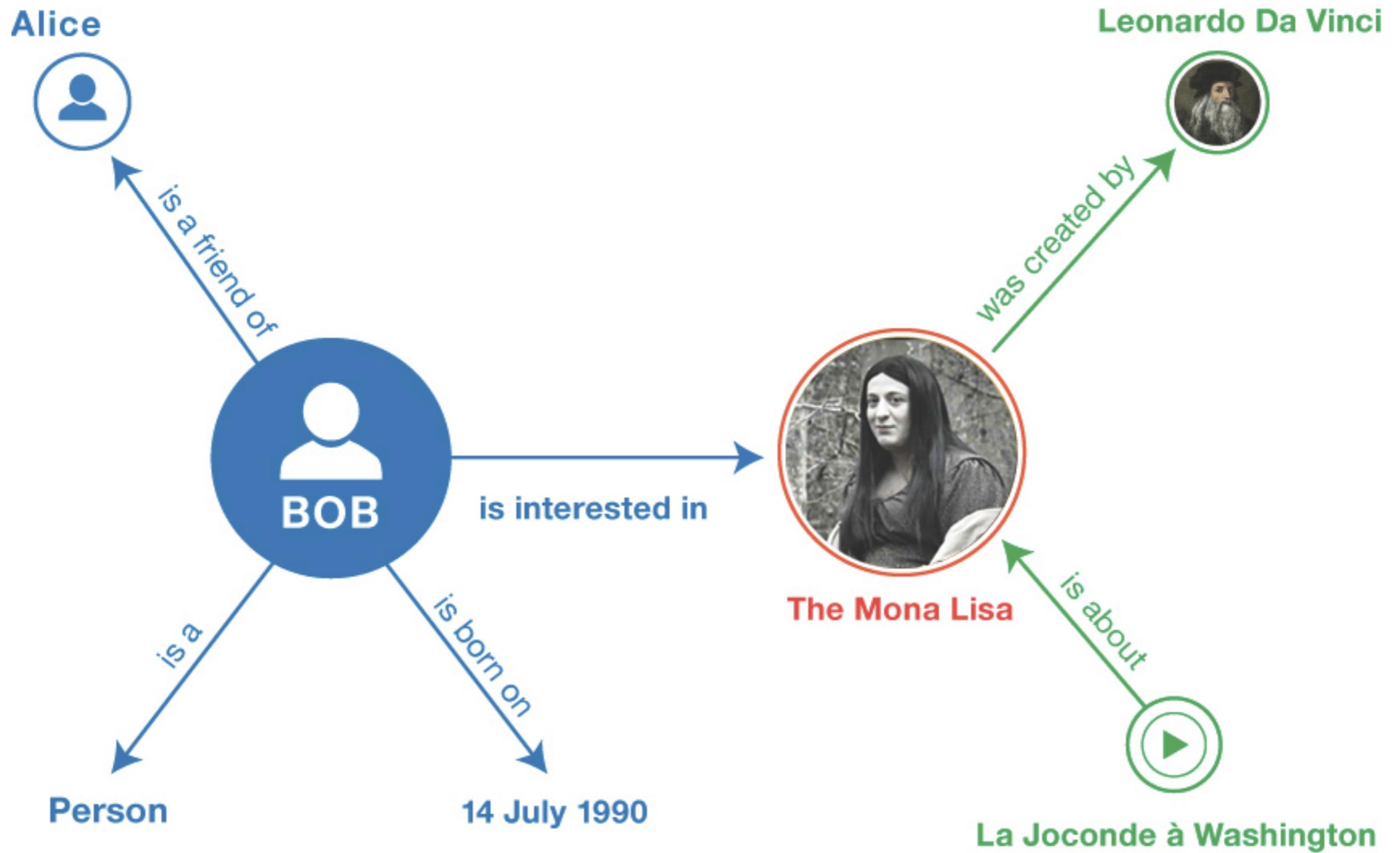


arguman.org'da bildirilen sfsatalar ve bulunduğu kanallar.  
<http://arguman.org/blog/sfsatalar/>

Ağ Veritabanları



- ilişkisel veritabanlarından farklıdır
- node (düğüm), edge (ilişki), ve bunlara bağlı veri setlerinden oluşurlar.
- kayıtlar (satırlar) için düğüm diyebiliriz.



# cypher

- Neo4j’nin kullandığı sorgu dilidir.
- SQL gibi düşünülebilir.

```
MATCH (me {name:"fatih"})-[:FOLLOWS]->(followees)  
RETURN followees
```

Neo4j data/graph.db

localhost:7474/browser/

```
1 MATCH (me {username:'fatiherikli'})  
2 WITH me  
3 MATCH (me)-[:FOLLOWS]->(followees)  
4 return followees;
```

CYPHER MATCH (me {username:'fatiherikli'}) WITH me MATCH (me)-[:FOLLOWS]->(followees) return followees;

Displaying 34 nodes, 16 relationships

# Sürücüler

- REST interface'ini kullanmak oldukça basittir.
- Eğer bir client isterseniz de birçok dilde implementation'i mevcuttur.

```
POST http://localhost:7474/db/data/transaction/commit
```

```
{  
  "statements": [ {  
    "statement" : "YOUR CYpher QUERY"  
  } ]  
}
```

# Düğümler

```
CREATE (User {username: 'ramazan'});  
CREATE (User {username: 'fatih'});  
CREATE (User {username: 'fatma'});  
CREATE (User {username: 'can'});  
CREATE (User {username: 'emine'});  
CREATE (User {username: 'gülzade'});  
CREATE (User {username: 'baki'});
```

# İlişkiler

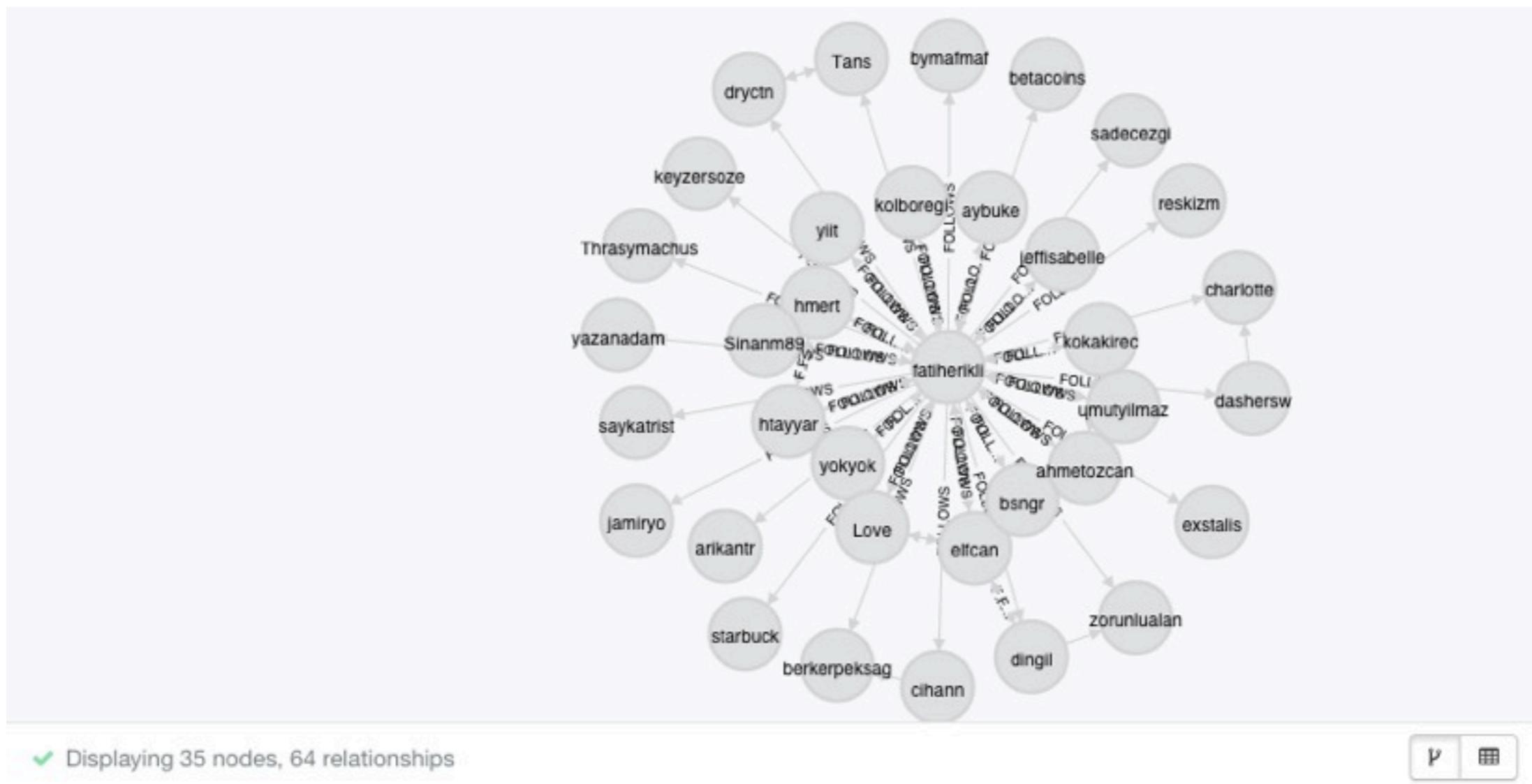
```
MATCH (user { username:'fatih' }),  
      (followee { username:'ramazan' })
```

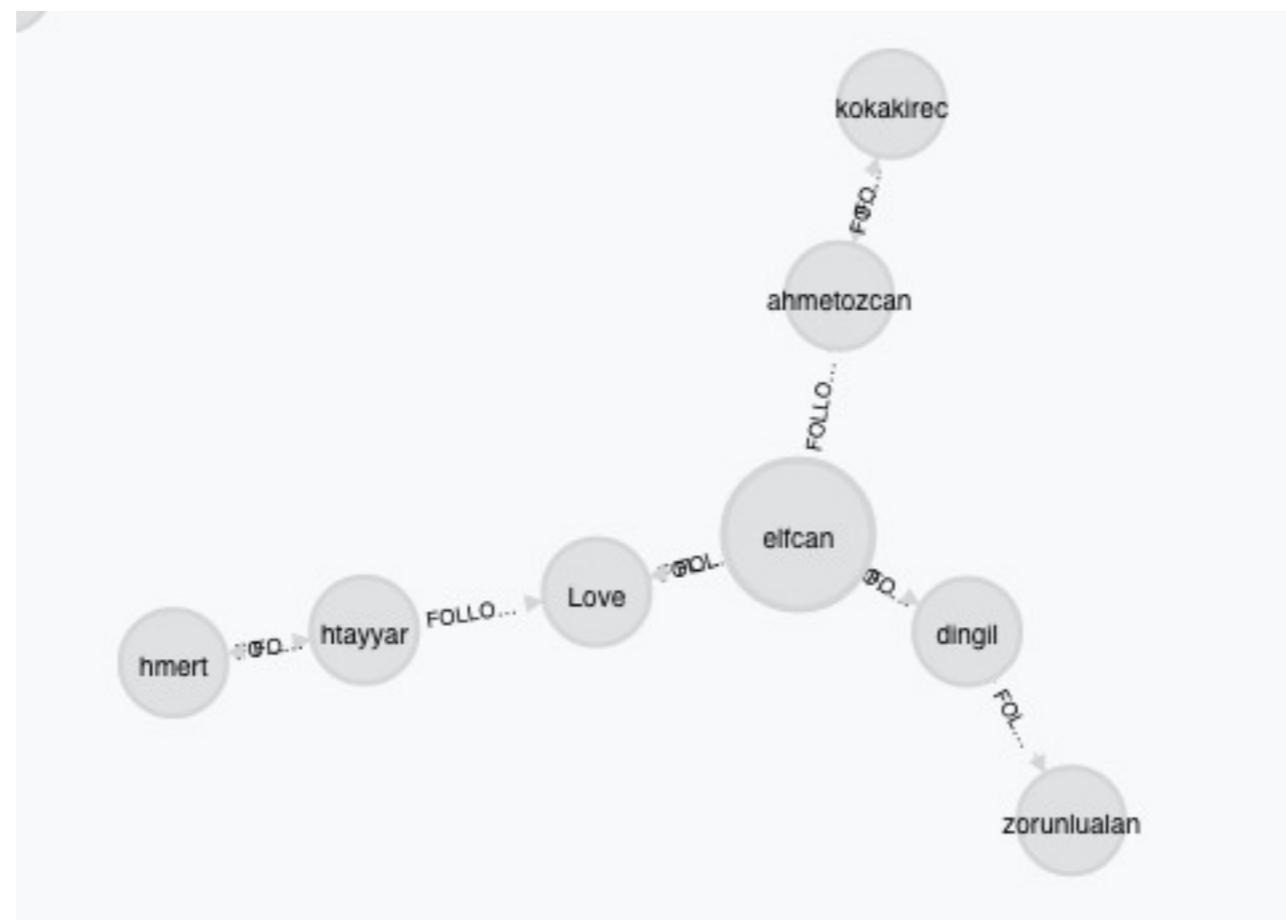
```
# user'ları match edip aralarında  
# bir edge (ilişki) yaratıyoruz
```

```
CREATE (user)-[:FOLLOWS]->(followee);
```

# Query

```
MATCH (me {username:'fatiherikli'})  
    -[r:FOLLOWS]->(followees)  
return me, followees
```





# Yönler

```
MATCH (me {username:'fatiherikli'})  
    <-[r:FOLLOWS]-(followers)  
return me, followers
```

# Oklar

(Kaynak) -[İLİŞKİ]-> (Hedef)

(user) -[:FOLLOWS]-> (followees)

(user)< -[:FOLLOWS]- (followers)

# Örnek:

fatih ve sepetle yumurta kisisinin ortak olarak takip ettiği kişiler:

```
MATCH (fatih { username:'fatiherikli' }),  
      (sepet { username: 'sepeth'})  
WITH fatih, sepet  
MATCH (fatih)-[:FOLLOWS]->(mutual)<-[:FOLLOWS]-(sepet)  
RETURN mutual
```

# ç1kt1S1

mutual



username berkerpeksag

id 2

username aybuke

id 211

✓ Returned 2 rows in 245 ms



# Örnek:

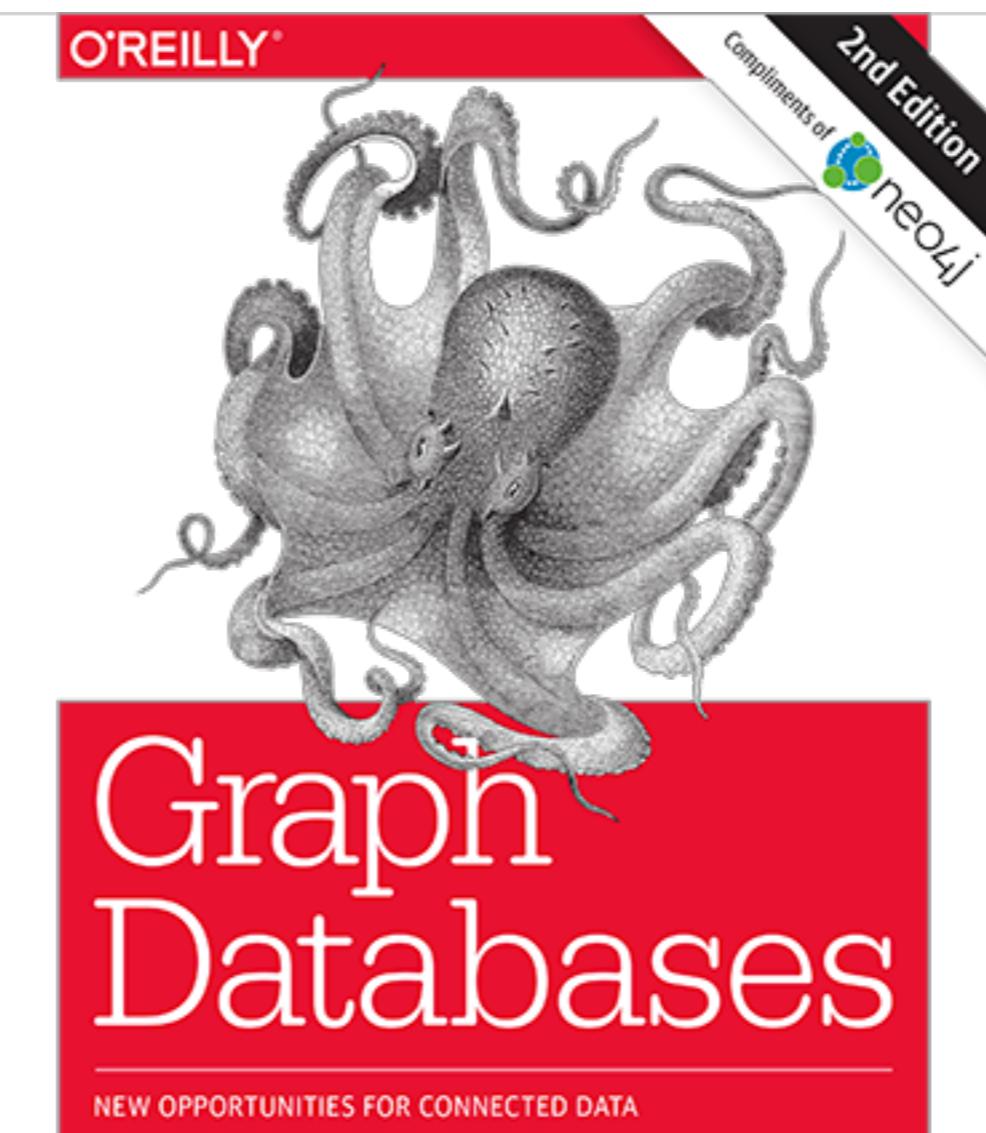
fatih kişisinin takip ettiği kişilerin yazdığını  
kanallar

```
MATCH (fatih { username:'fatiherikli' })
WITH fatih
MATCH (fatih)-[:FOLLOWS]->(followees)<-[ :POSTED_IN ]-(channels)
RETURN channels
```

# Örnek:

fatih kişisinin takip ettiği kanallar üzerinden takip edilecek kişi önerileri

```
MATCH (fatih { username:'fatiherikli' })
WITH fatih
MATCH (fatih)-[:SUBSCRIBER_OF]->(channels)
    <- [:POSTED_IN]-(contentions)
    <- [:POSTED_BY]-(authors)
RETURN distinct authors
```



Ian Robinson,  
Jim Webber & Emil Eifrem

[graphcommons.com](https://graphcommons.com)

Teşekkürler

