

# Karmaşık Ağlar

NetworkX

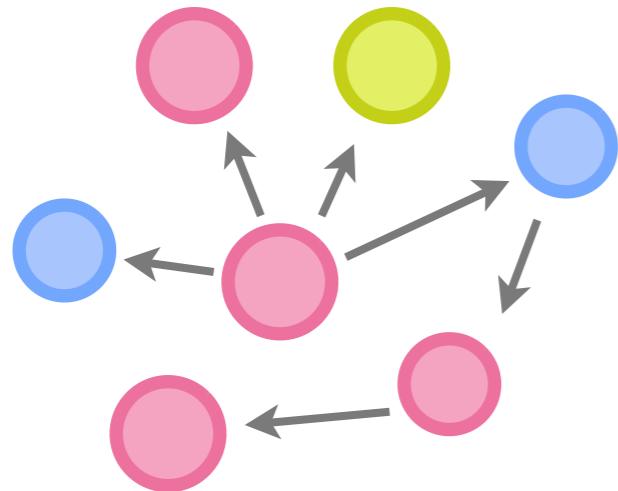
# selam

Fatih Erikli

<http://fatiherikli.com>

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

# ağ=düğümküme



noktalar ve çizgiler  
kümesidir.

nokta	çizgi	
köşe	kenar	matematik
düğüm	bağlantı	bilgisayar bilimi
yapıtaşısı	bağ	fizik
aktör	ilişki	sosyoloji



# graph theory

Euler

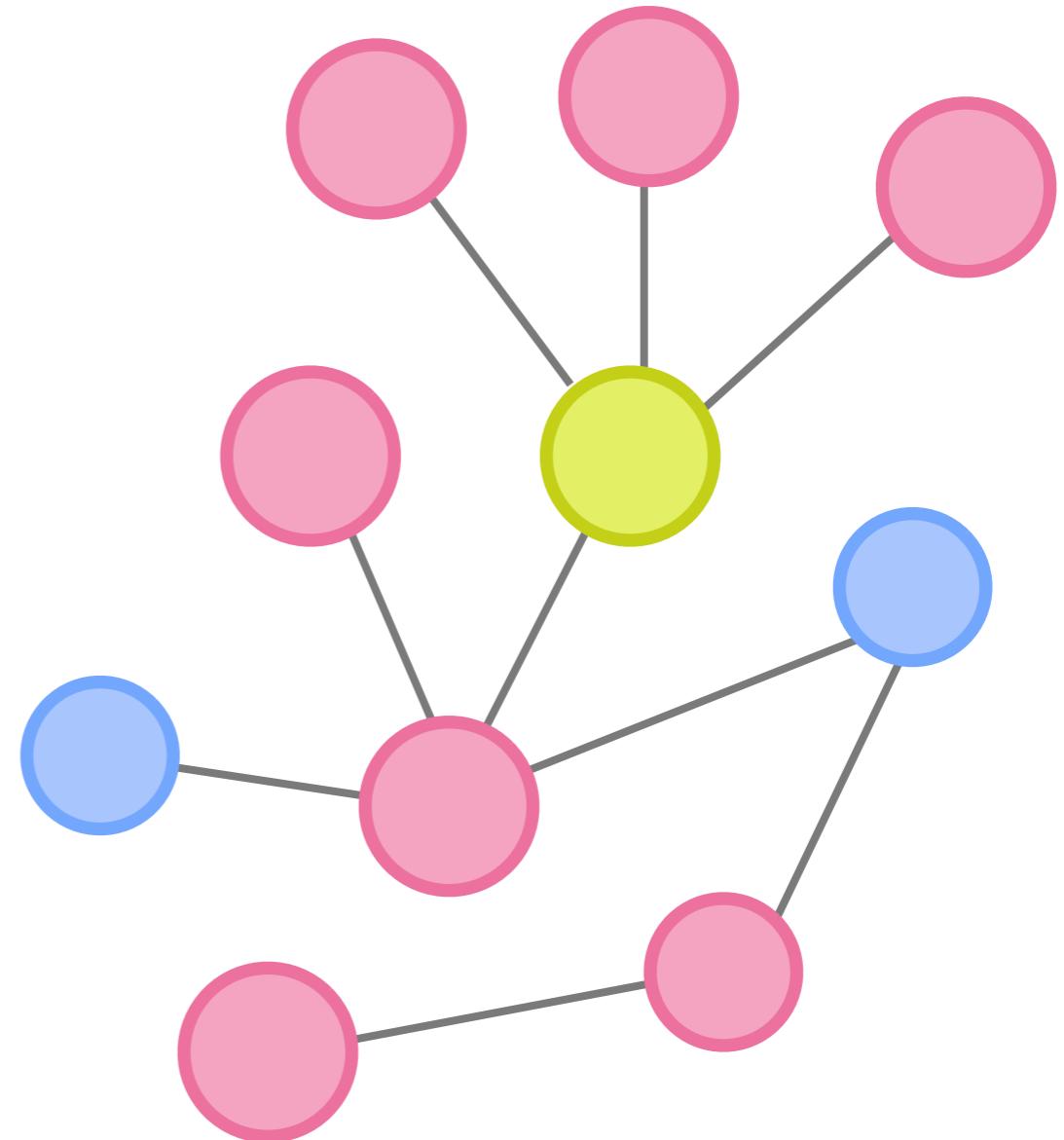
1736

Kaliningrad köprüleri

he used graphs  
before it was cool

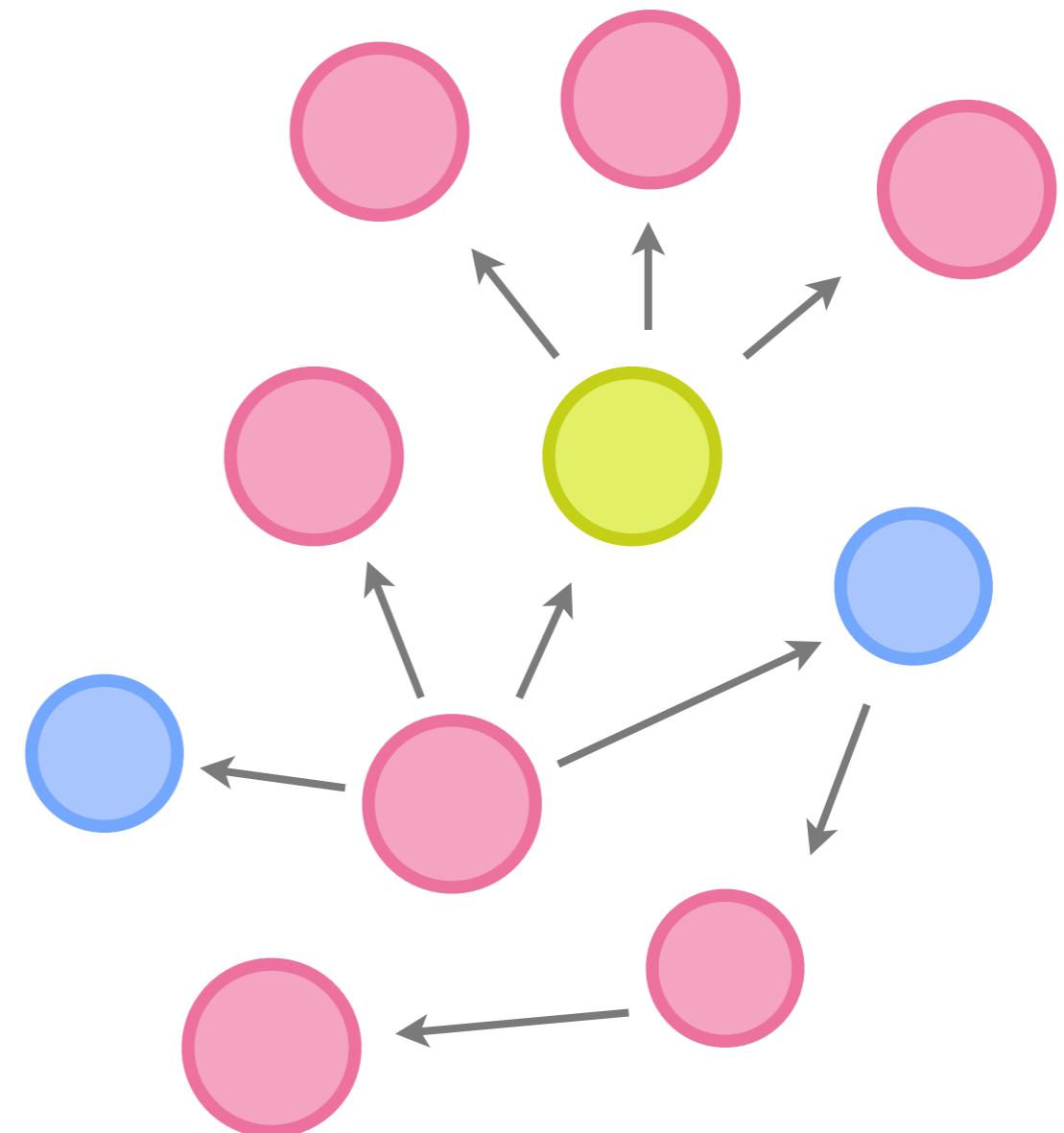
# undirected

- arkadaşlıklar
- savaşmalar
- çarpışmalar
- ulaşım ağları

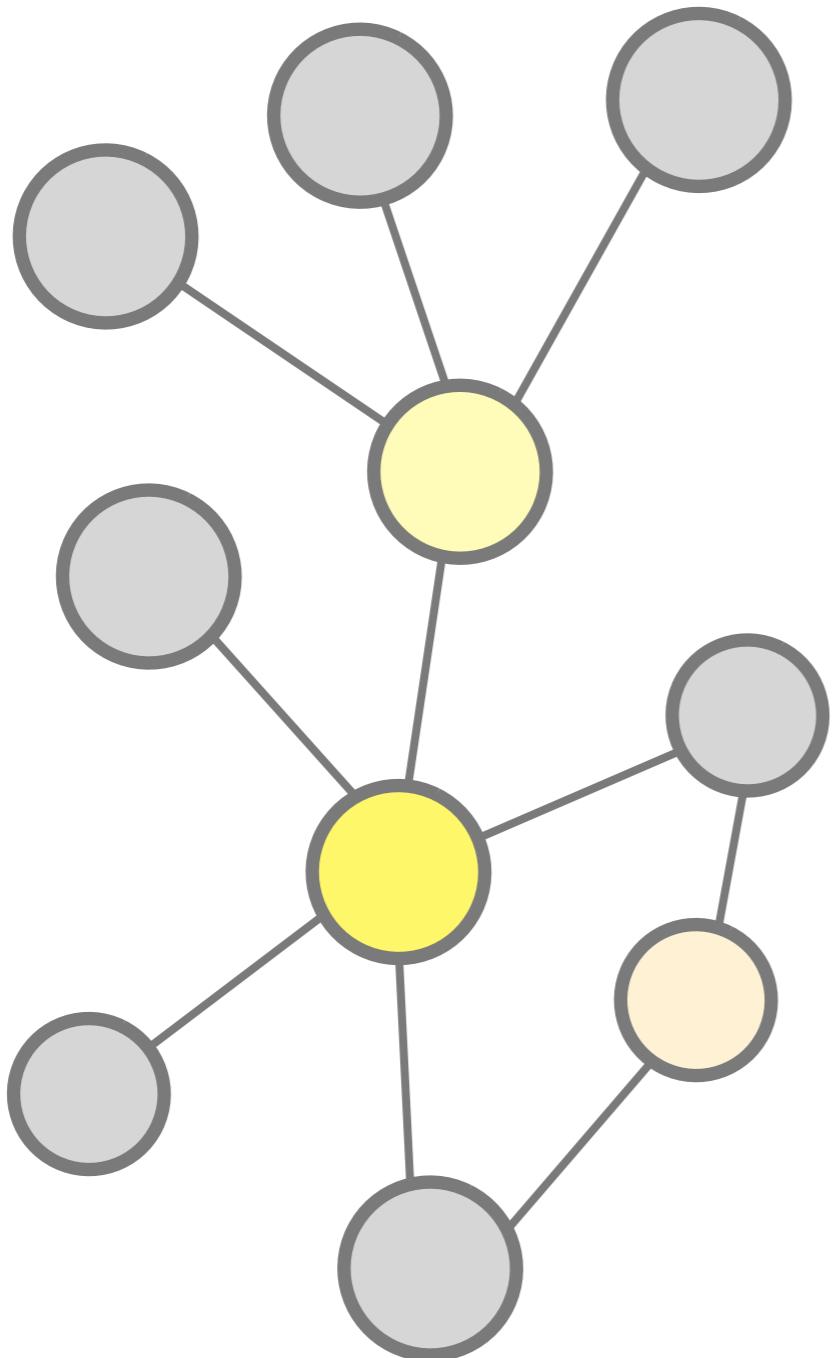


# directed

- takipçilermeler
- galibiyetler
- internet linkleri
- platonik sevgililikler

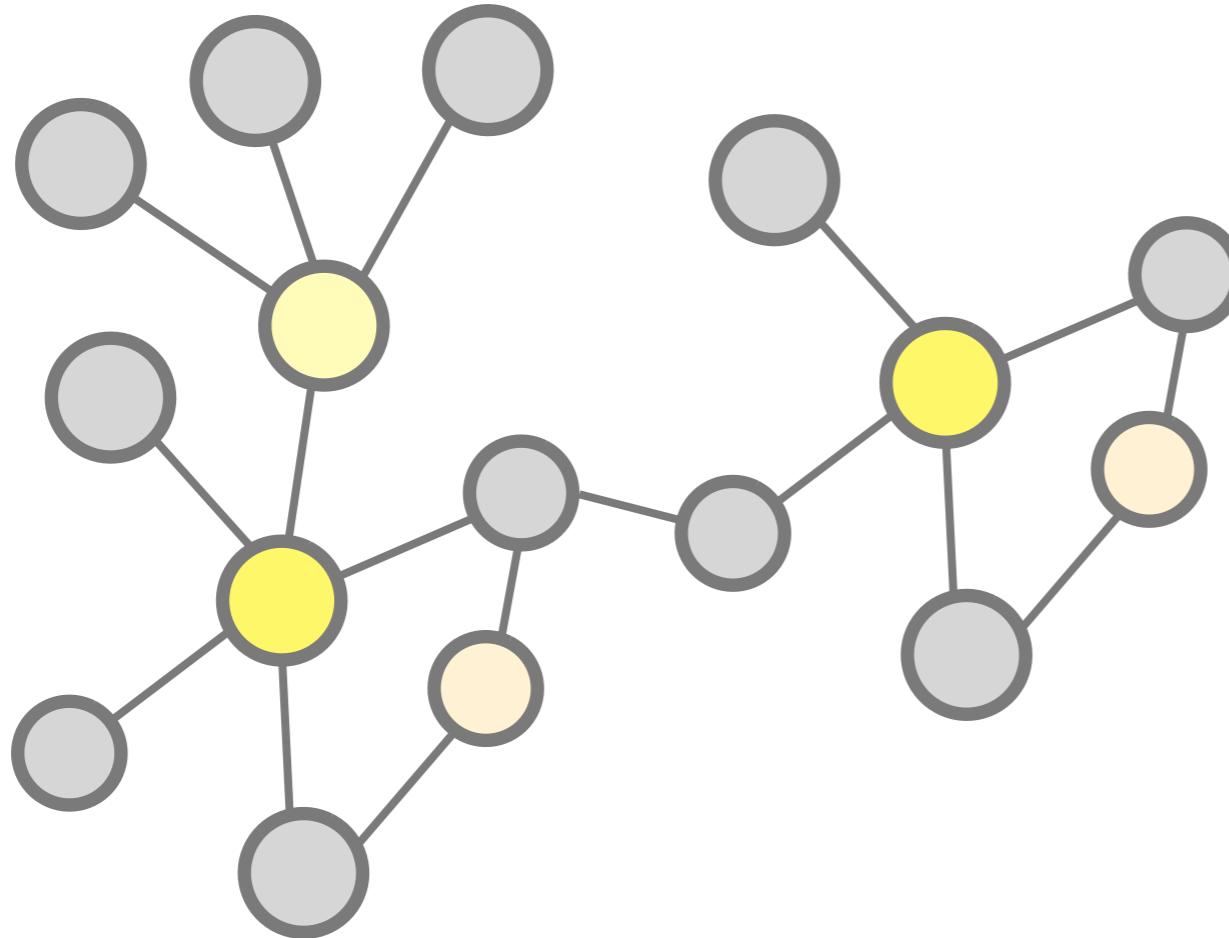


# merkeziyet

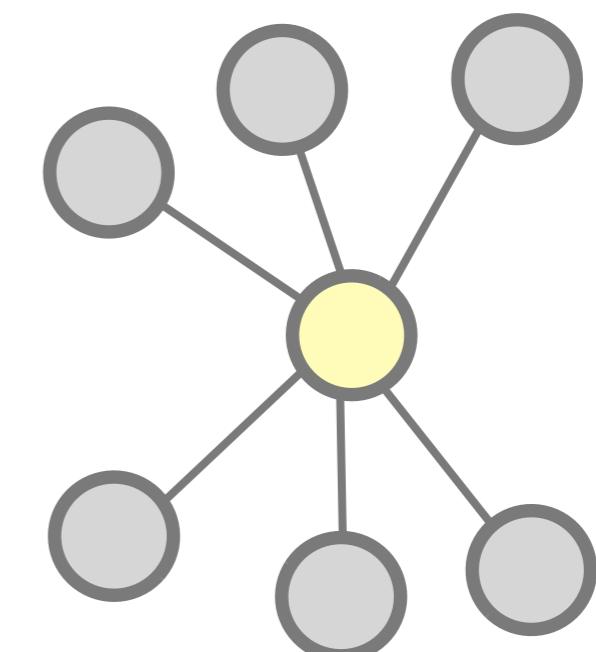


- Degree
- In degree
- Out degree
- Betweenness
- Closeness

# topluluk



Giant  
Component



Isolated  
Component

# mülksüzleştime ağları

### 3. Havalimanı

# programlama dilleri ağları

programming  
language  
network

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

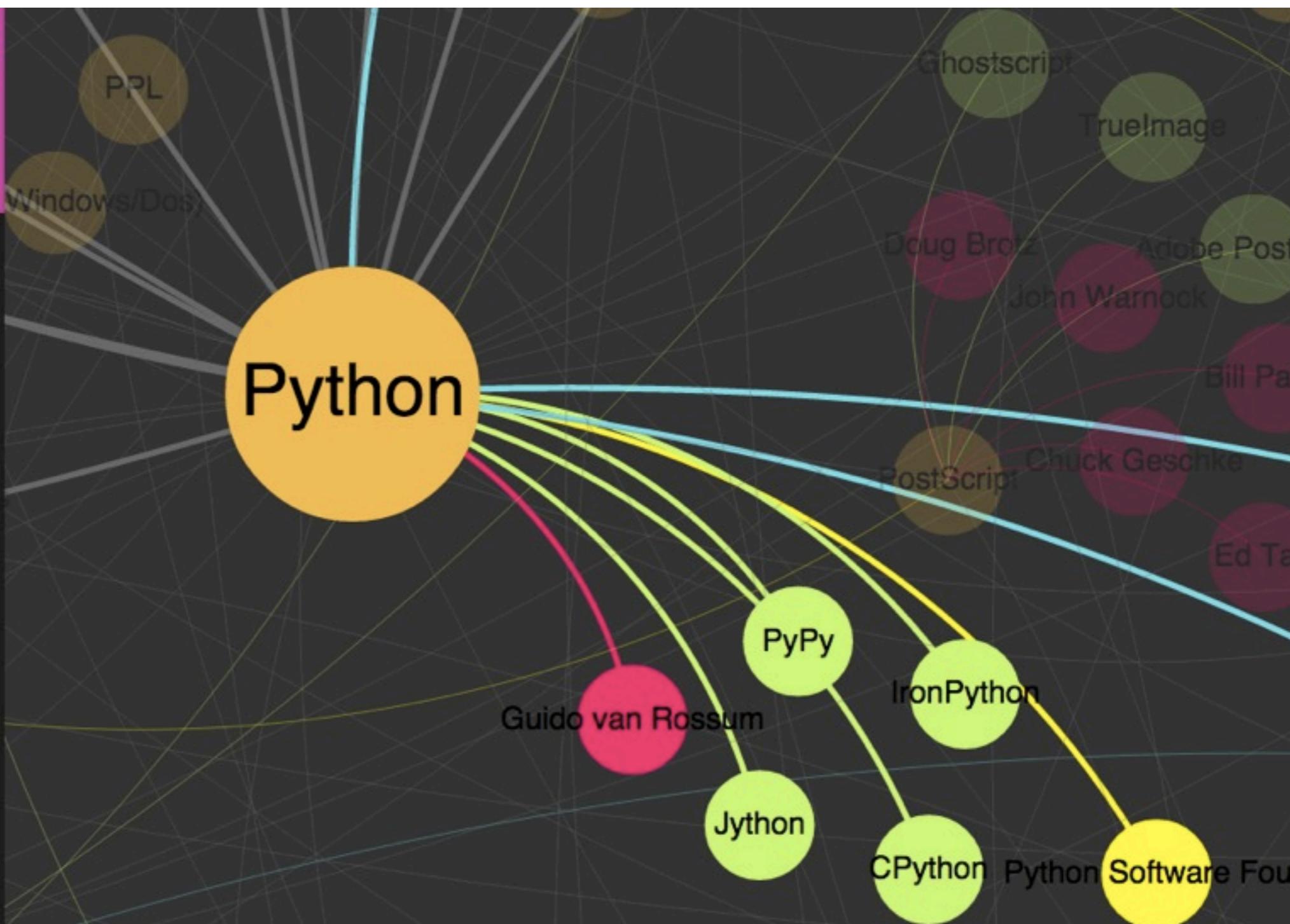
## Nodes

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

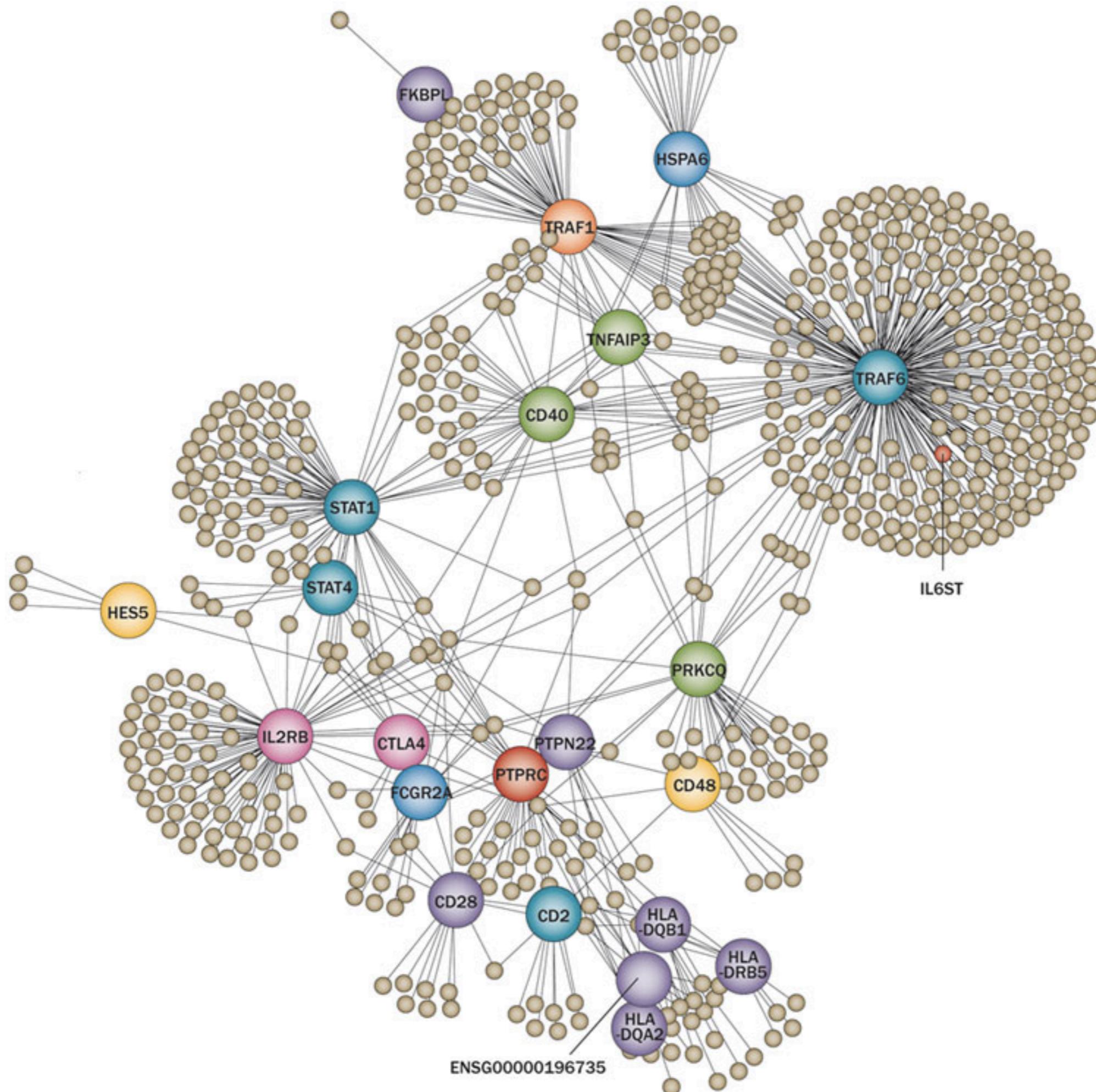
## Edges

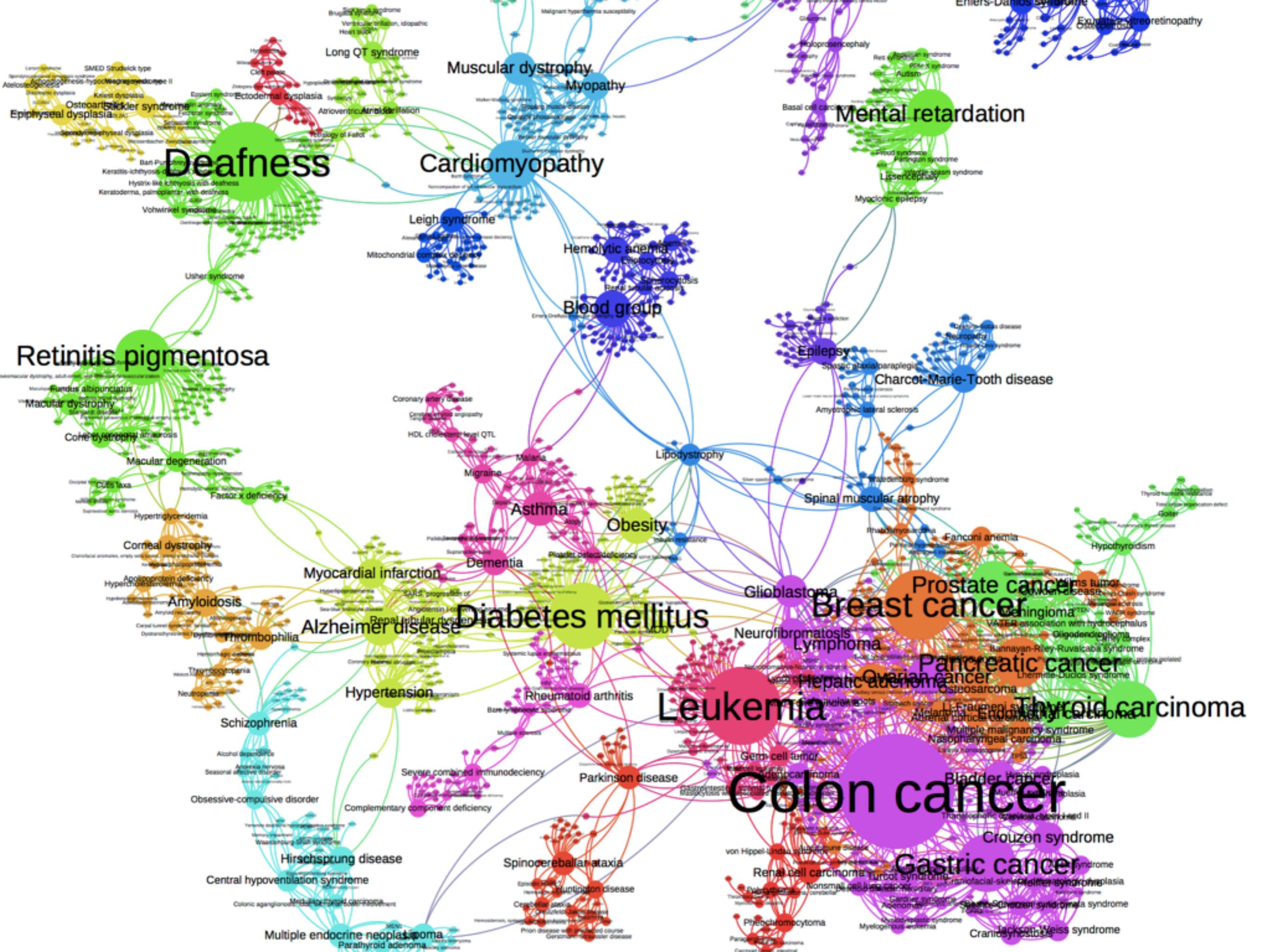
- Influenced by
- Designed by
- Developer
- Dialects
- Major implementations
- Implementation language

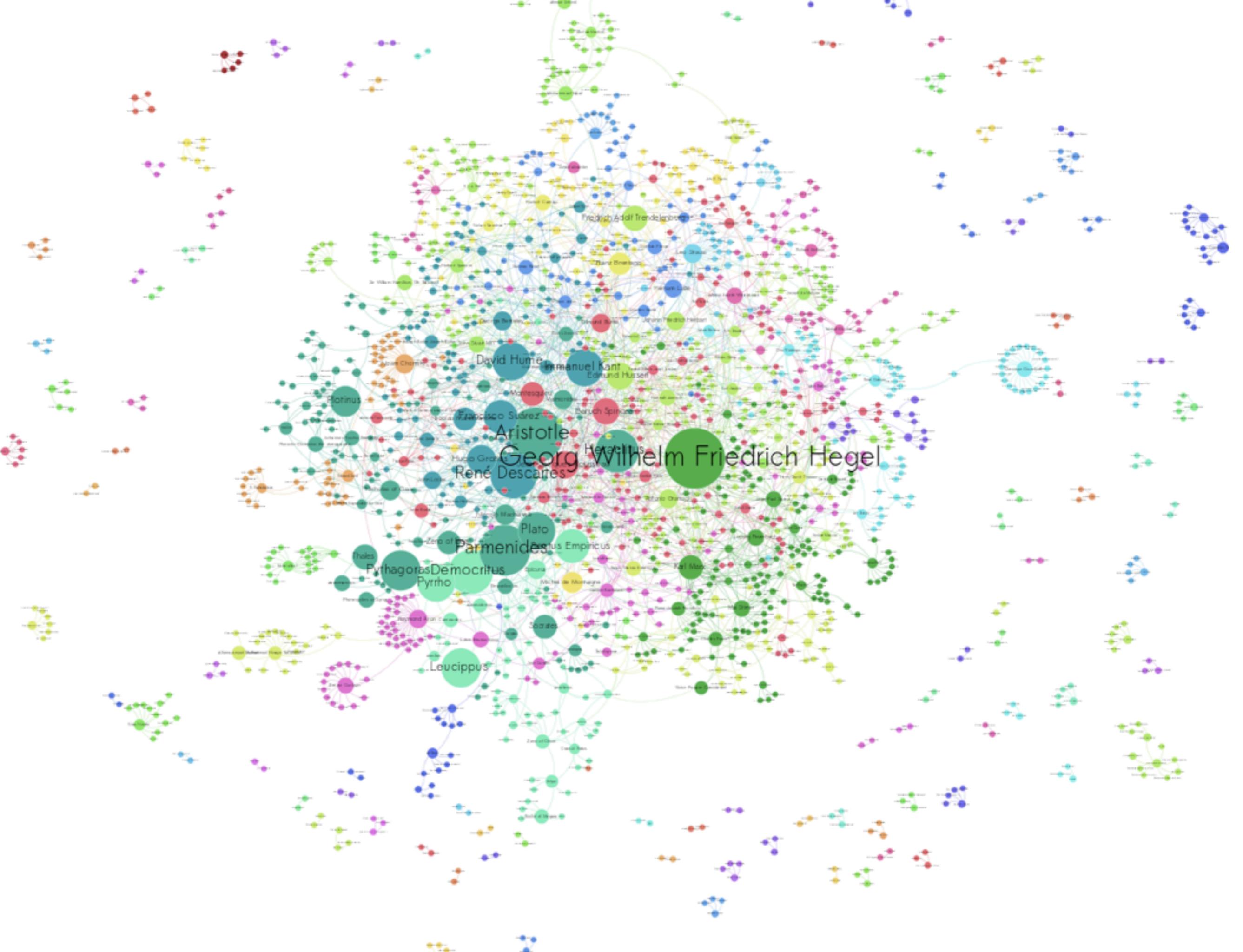
source code

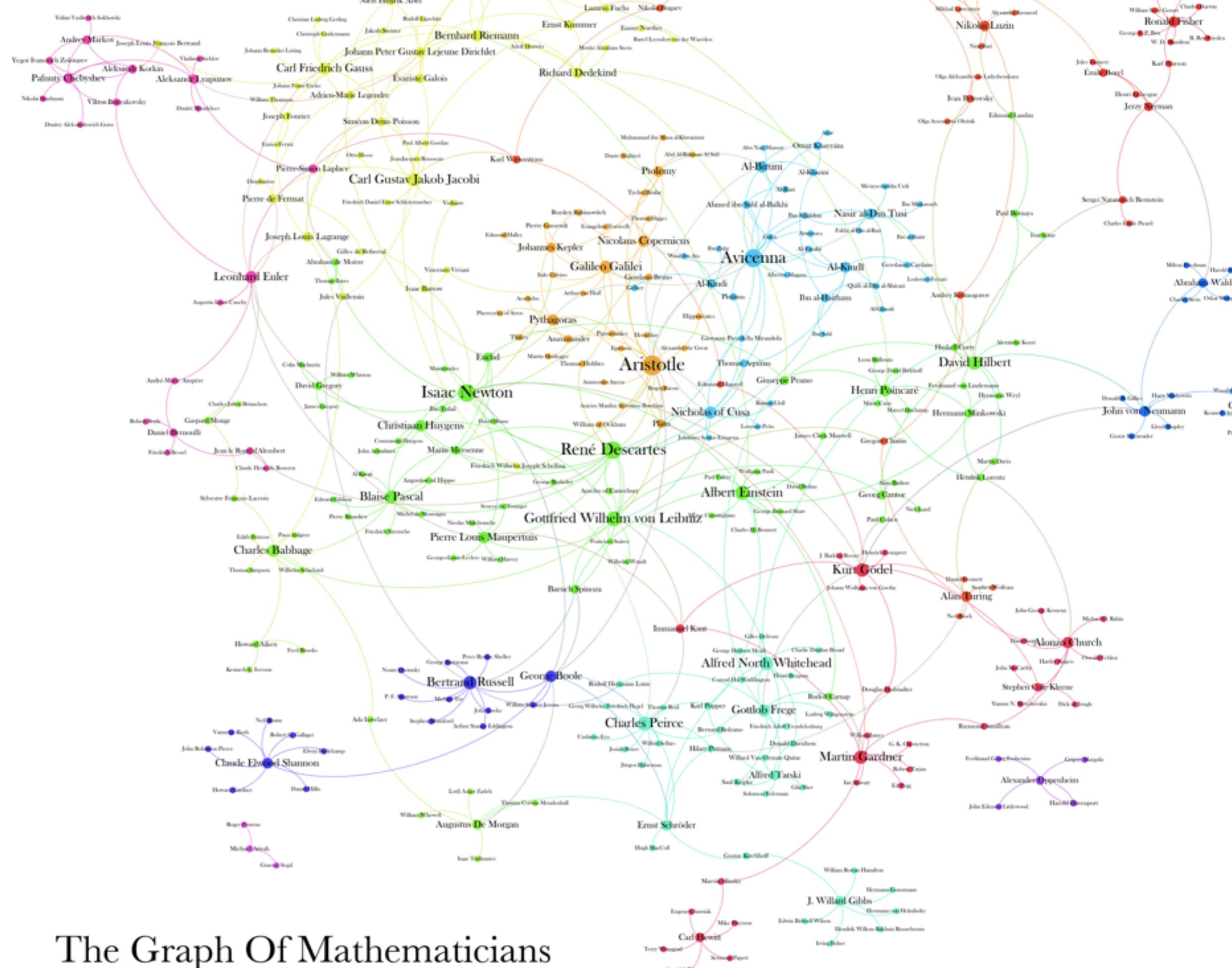


Disiplinlerarası



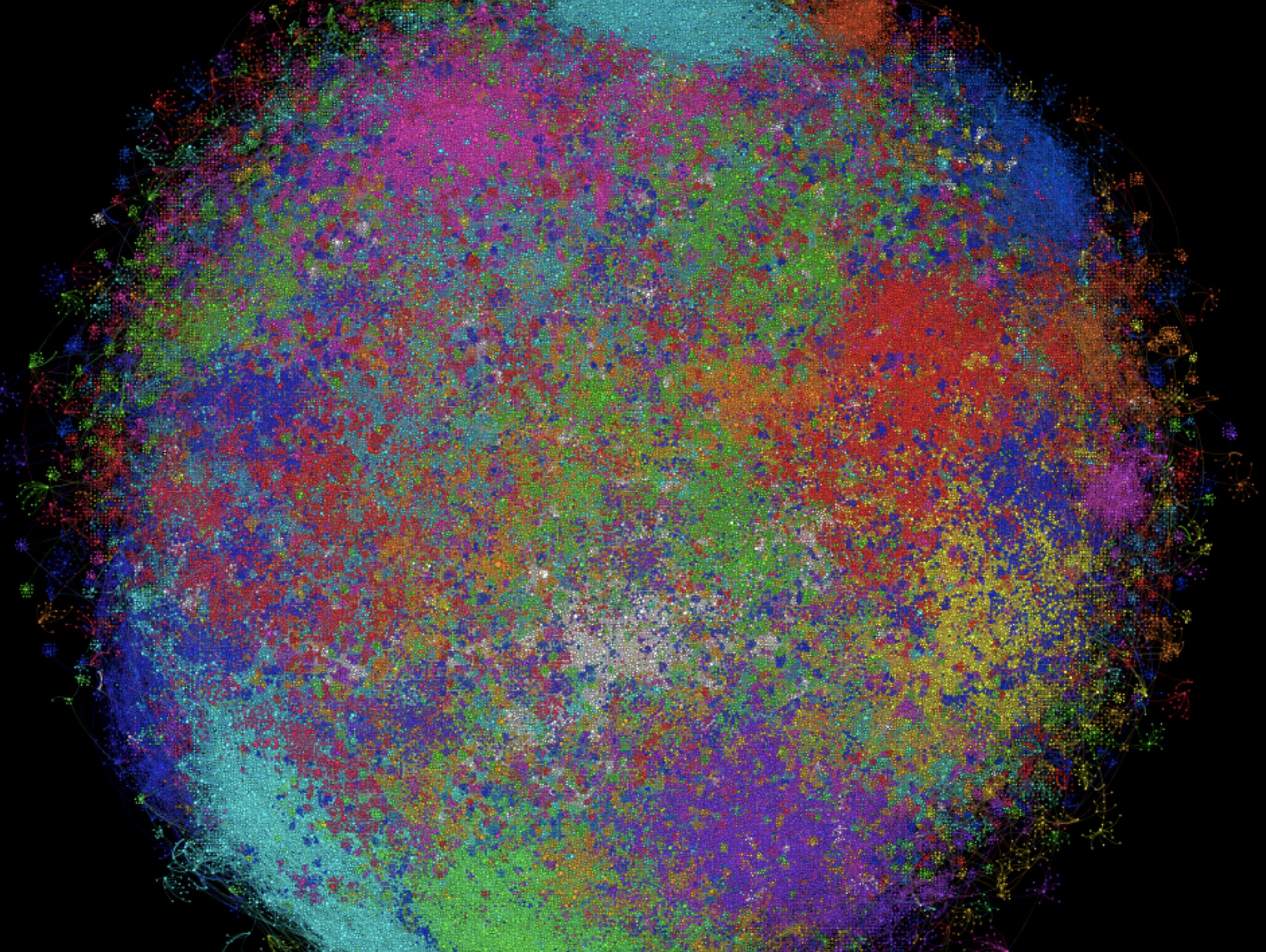






# The Graph Of Mathematicians

By Brendan Griffen [www.griffsgraphs.com](http://www.griffsgraphs.com)



Empirical  
and Quantitatif  
which means:

Empirik  
ve Kantitatif

Güzel kadınları severim  
İşçi kadınları da severim  
Güzel işçi kadınları  
Daha çok severim

orhan veli

karmasık  
ağlar

# scale-free networks

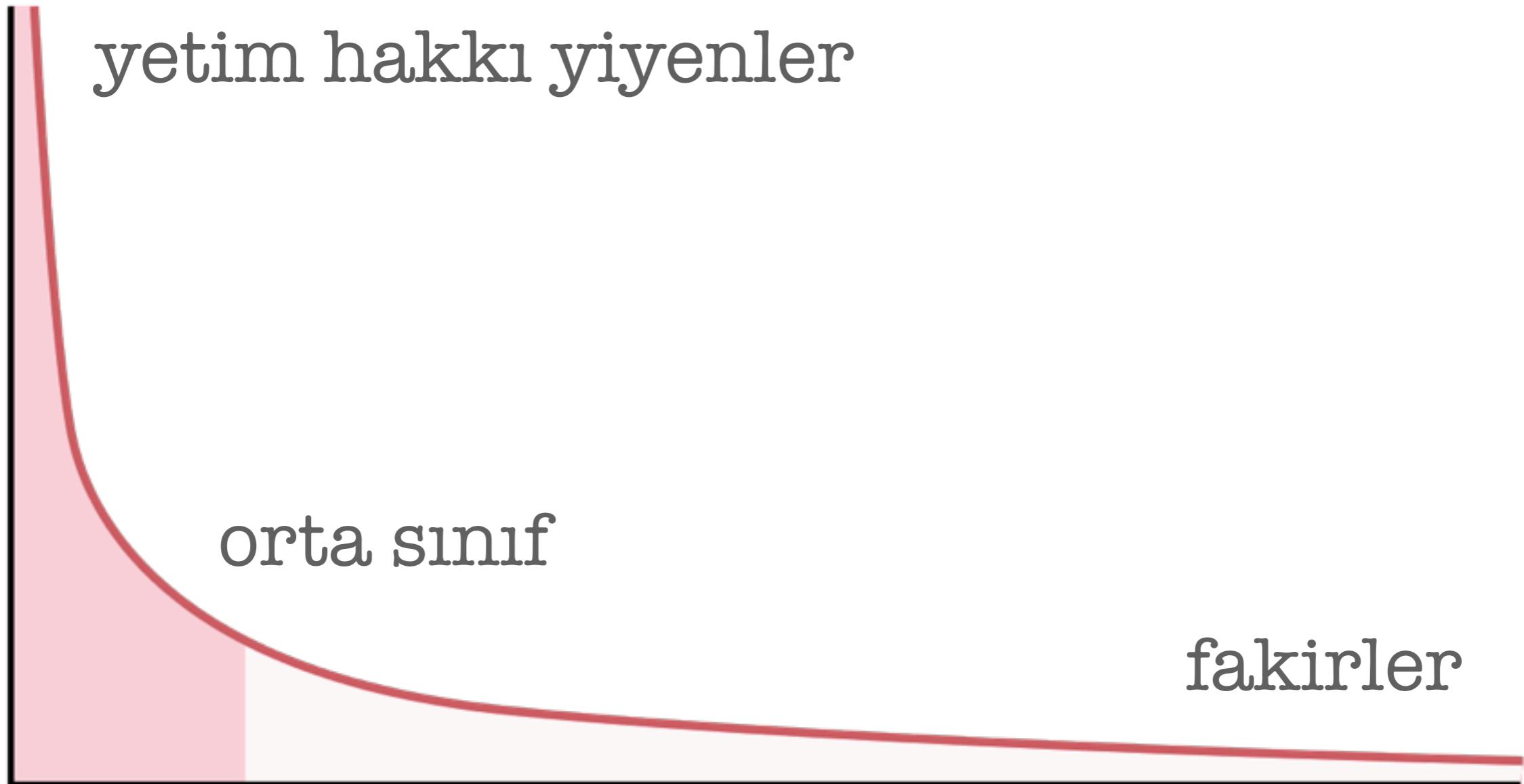
power  
law  
distribution

sosyal ağlar protein ağları

internet sinir ağları

www hava ulaşımı

# power-law



networkx

karmaşık ağ

analiz kütüphanesi

# temel kullanımlar

- Graph
  - DiGraph
  - MultiGraph
  - MultiDiGraph
- add\_node
- add\_edge

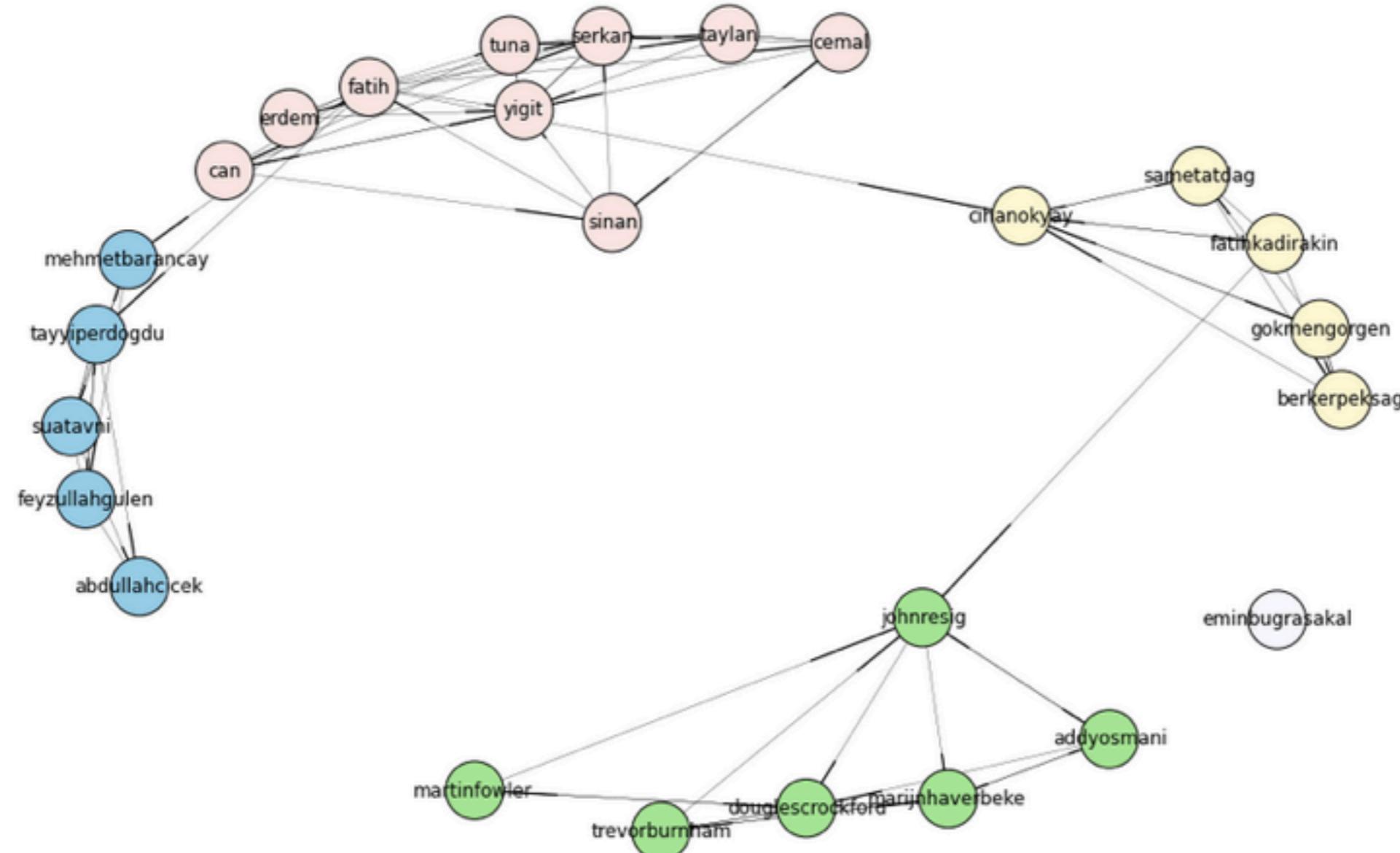
```
graph = Graph()  
  
graph.add_node("fatih")  
  
graph.add_node("ramazan")  
  
graph.add_node("fatma")  
  
graph.add_edge("ramazan", "fatih", "brother")  
  
graph.add_edge("fatma", "fatih", "sister")
```

```
>>> nx.density(graph)  
0.666666666667  
>>> graph.add_edge("ramazan", "fatma")  
>>> nx.density(graph)  
1  
>>> nx.degree("fatih")  
2
```

```
graph.add_edge("edi", "budu")
graph.add_edge("budu", "fofo")
graph.add_edge("hede", "budu")
graph.add_edge("budu", "fofo")
>>> graph.number_connected_components()
```

2

```
>>> list(nx.connected_components(graph))  
[  
    ['fofo', 'hede', 'edi', 'budu'],  
    ['fatma', 'ramazan', 'fatih']  
]
```



```
nx.draw(graph,
         with_labels=True,
         node_size=1400,
         node_color=node_colors,
         width=0.3)
```

çözülebilecek  
problemler

- Tanıyor olabileceğin kişiler
- Kaçincı dereceden tanıyorsun
- Ortak ata problemleri
- Gruplama, Clustering
- Öneri sistemleri

# kaynaklar

<http://barabasilab.neu.edu>

<http://graphcommons.com>

<http://mulksuzlestirme.org>

[http://fatiherikli.github.io/  
programming-language-network](http://fatiherikli.github.io/programming-language-network)

[jobs@hipo.biz](mailto:jobs@hipo.biz)