NDlib: a Python Library to Model and Analyze Diffusion Processes Over Complex Networks

IEEE DSAA 2018
Turin, Italy
1-4 October, 2018

Letizia Milli, Giulio Rossetti, Salvatore Rinzivillo
{name.surname}@isti.cnr.it

ISTI

KDD Lab

IN SUPREMÆ DIGNITATIS

1343

Horizon 2020, EU Projects

NDlib is funded by the European Community’s H2020 Program

Founding Scheme:
“FETPROACT-1- 2014: Global Systems Science (GSS)”

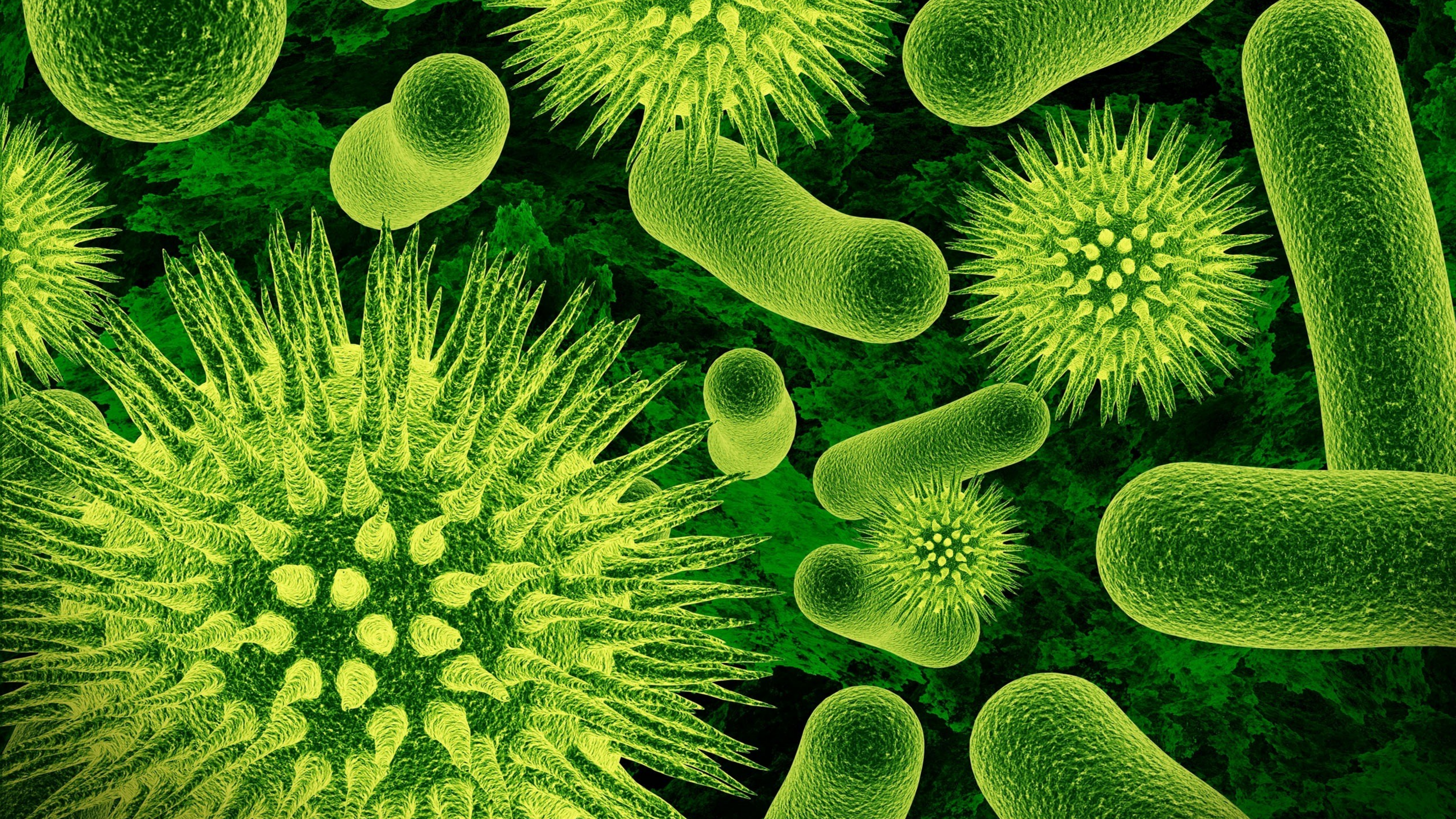
Project:
CIMPLEX, Bringing CItizens, Models and Data together in Participatory, Interactive SociaL EXploratories

Website:
https://www.cimplex-project.eu

Founding Scheme:
“INFRAIA-1-2014-2015: Research Infrastructures"

Project:
SoBigData, Social Mining & Big Data Ecosystem

Website:
http://www.sobigdata.eu

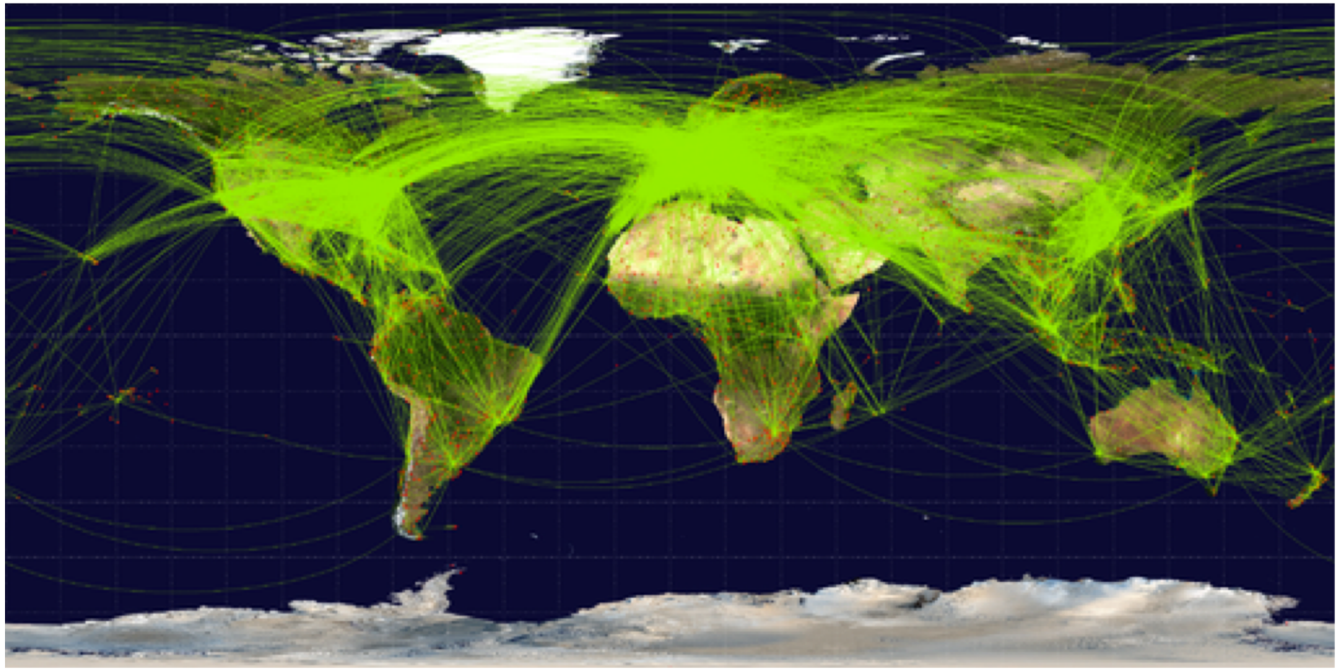



SAMSUNG
edit
Messages
Logs
Internet
Samsung Apps
Smart search
Settings
Calendar
Camera
Home
Alarm
Contacts
Dialer


Strategy Innovation
Vision
Creativity
Support


INNOVATION
BUSINESS
PLAN
IDEA

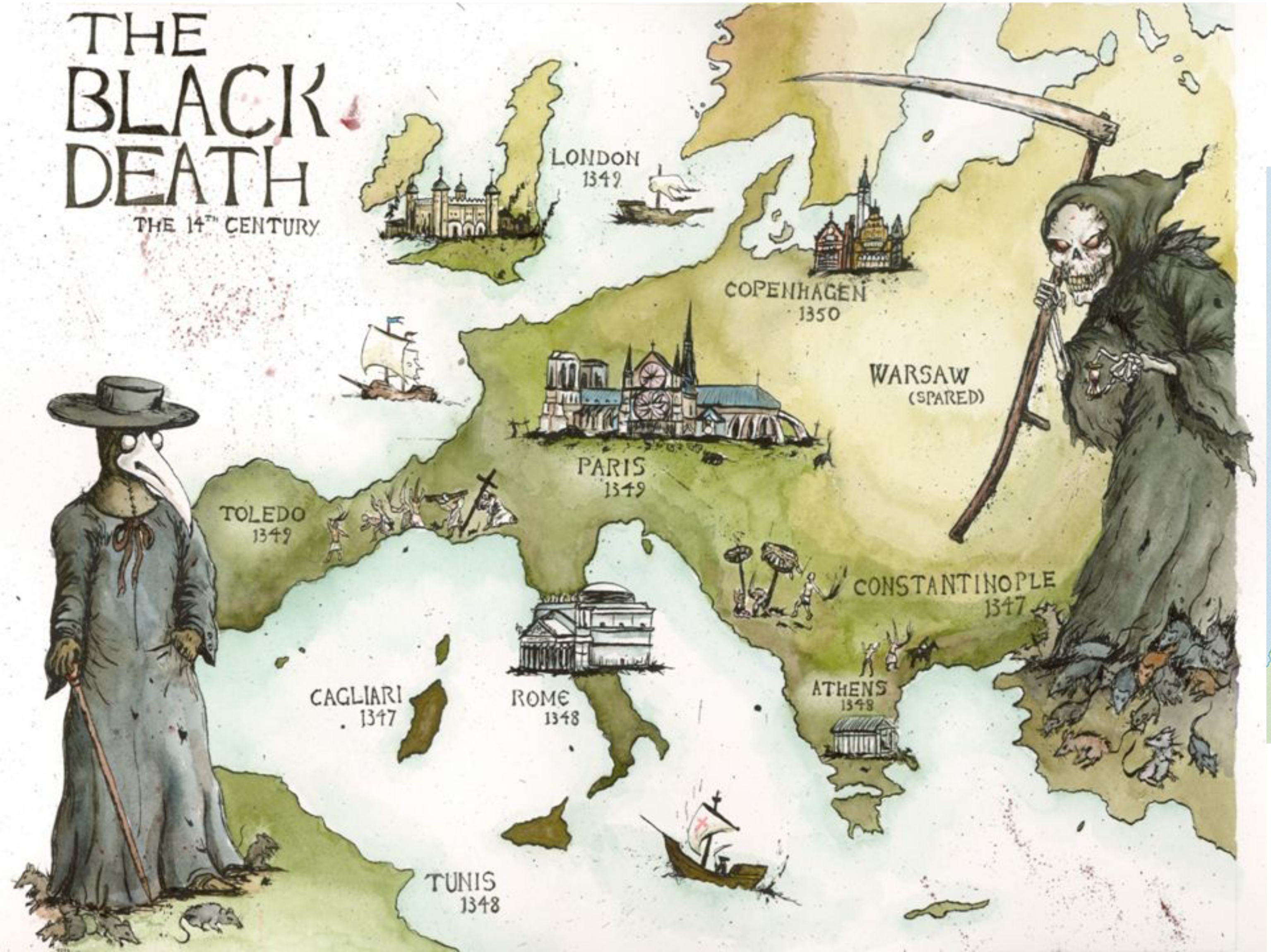
Why do diffusive processes matter now?

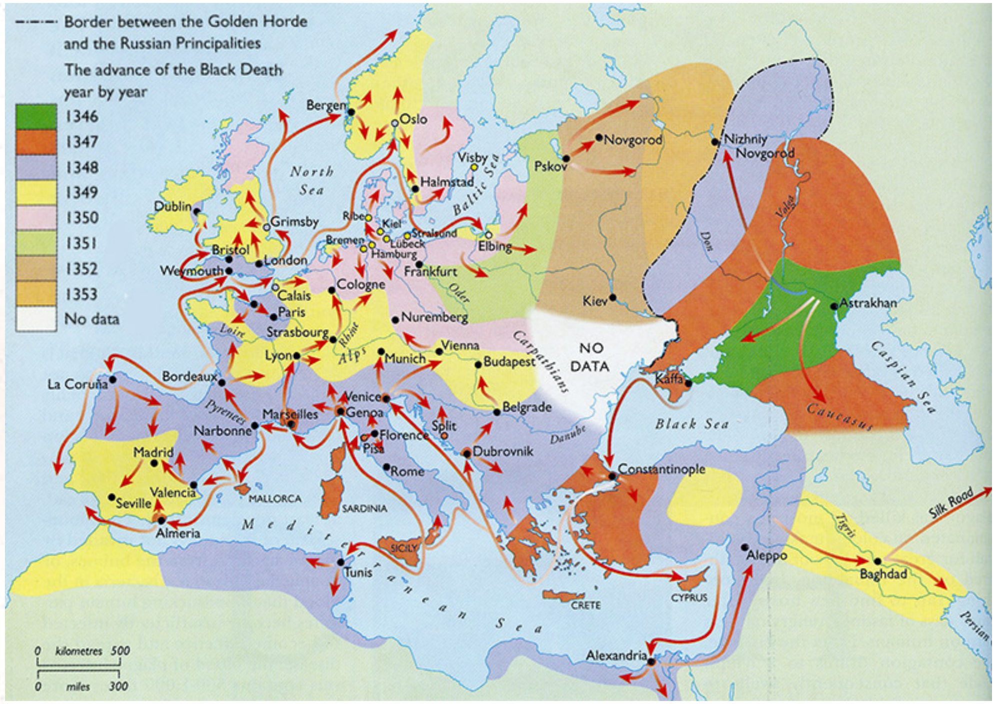


High mobility

High population density

The great plague in **14th century**

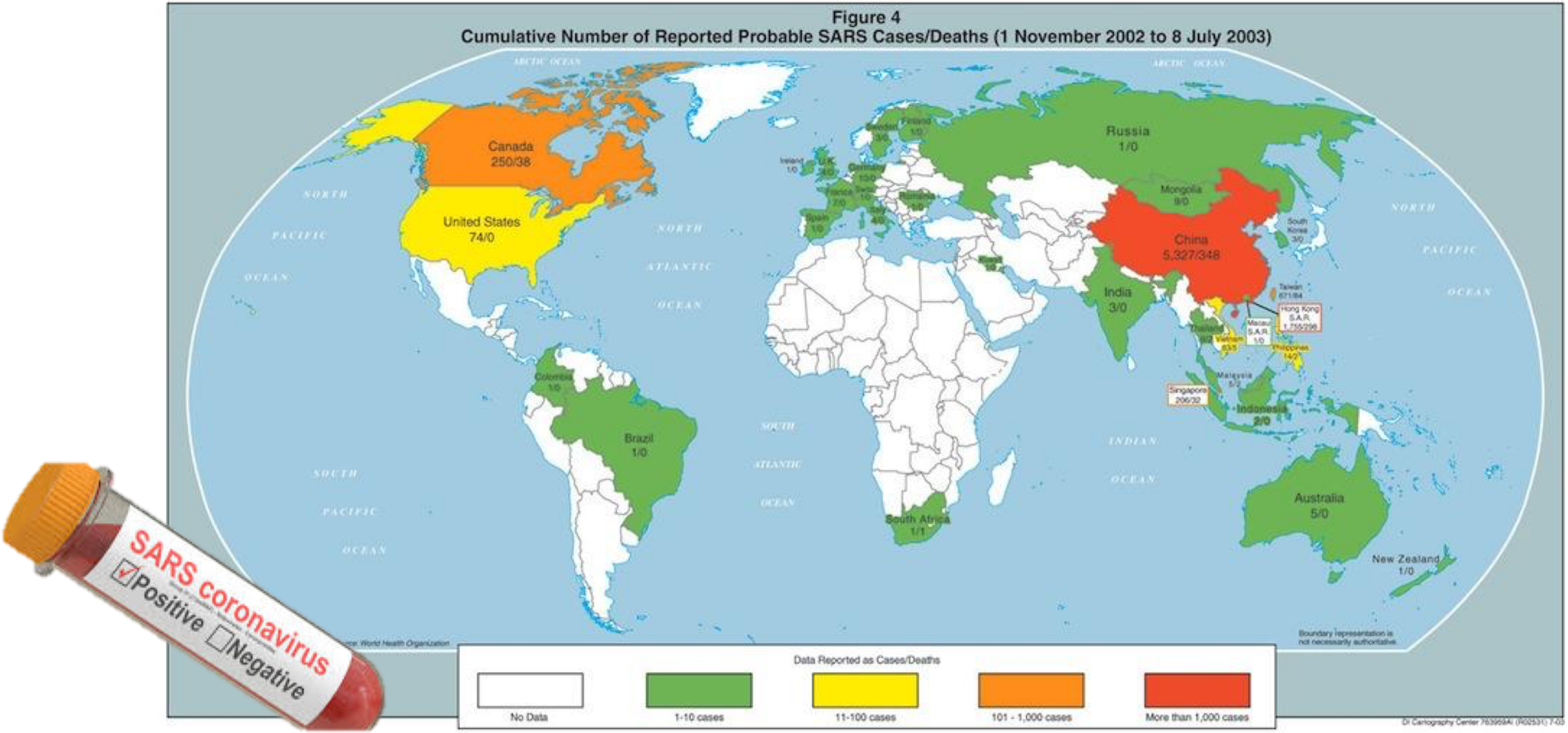




4 years from France to Sweden

SARS in 21th century

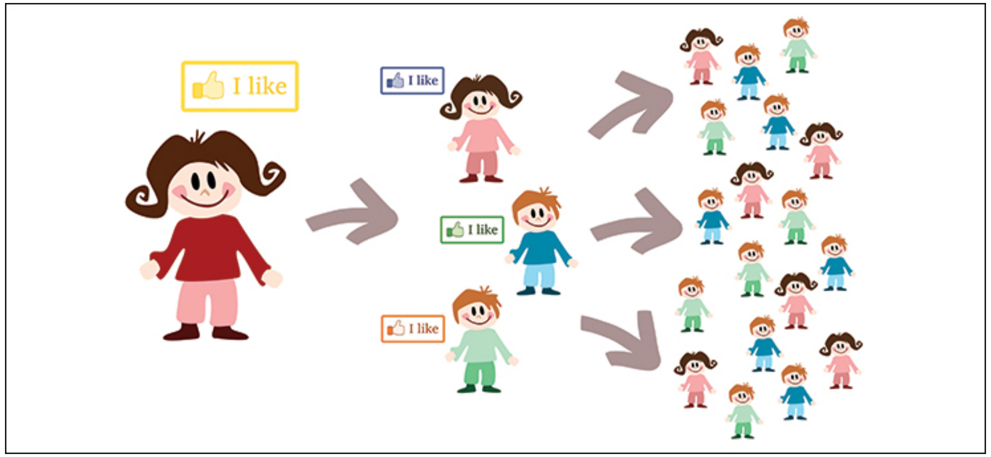
6 months…

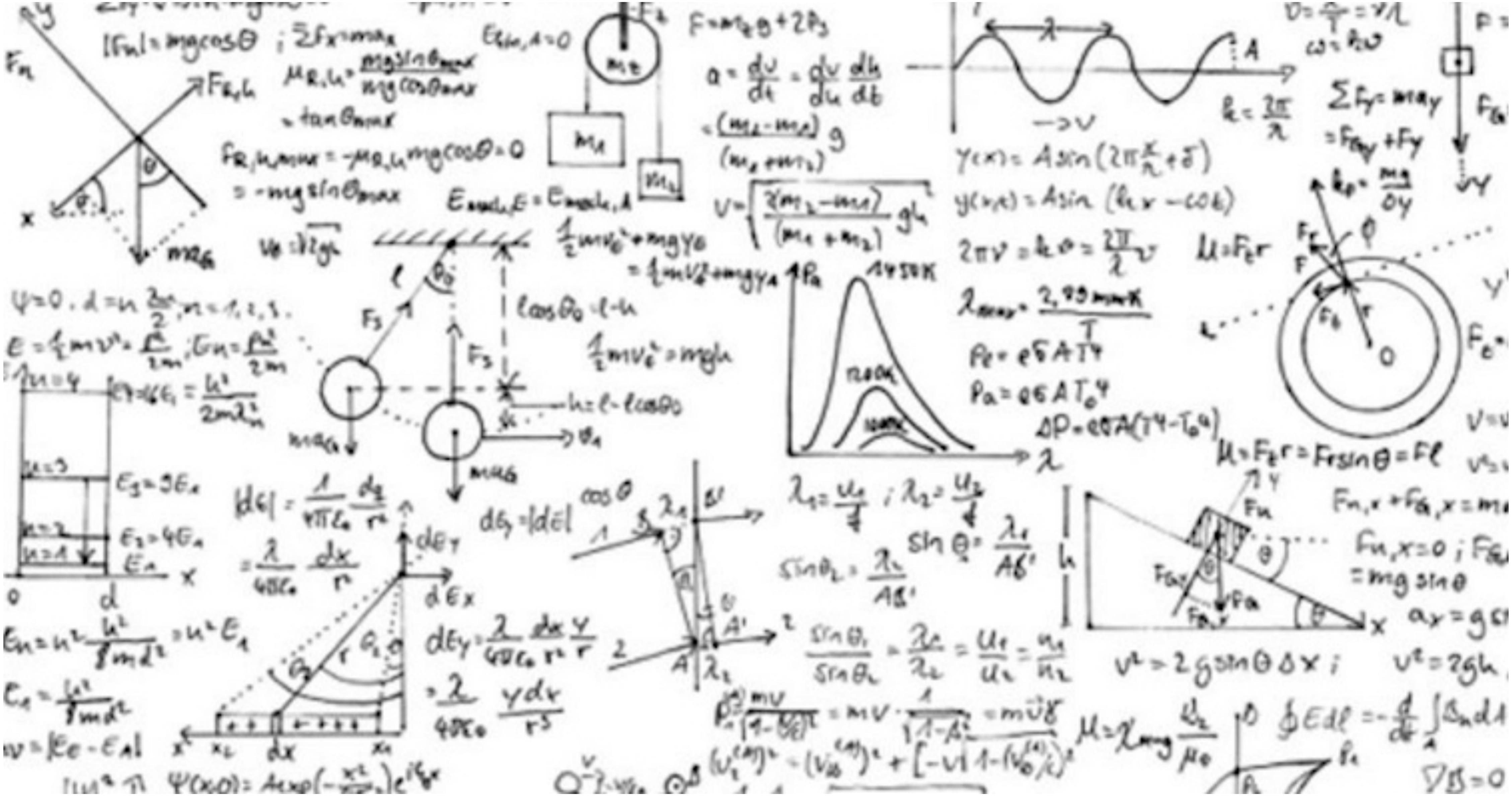


Epidemics, Opinions, Innovations…

The difference: the degree of *activeness* of the subjects they affect

- Passive: viruses…
- Active: ideas, adoptions…



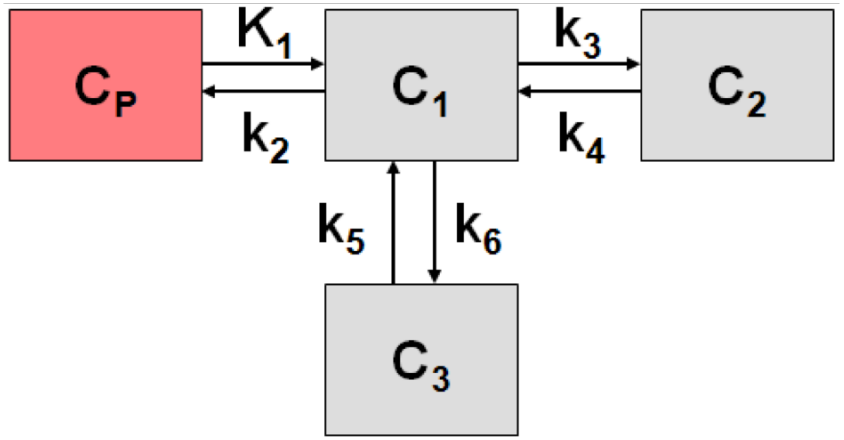


How can we model diffusive phenomena?

Compartmental Models
[Kermack & McKendrick, 1927]

The framework is based on two hypotheses:

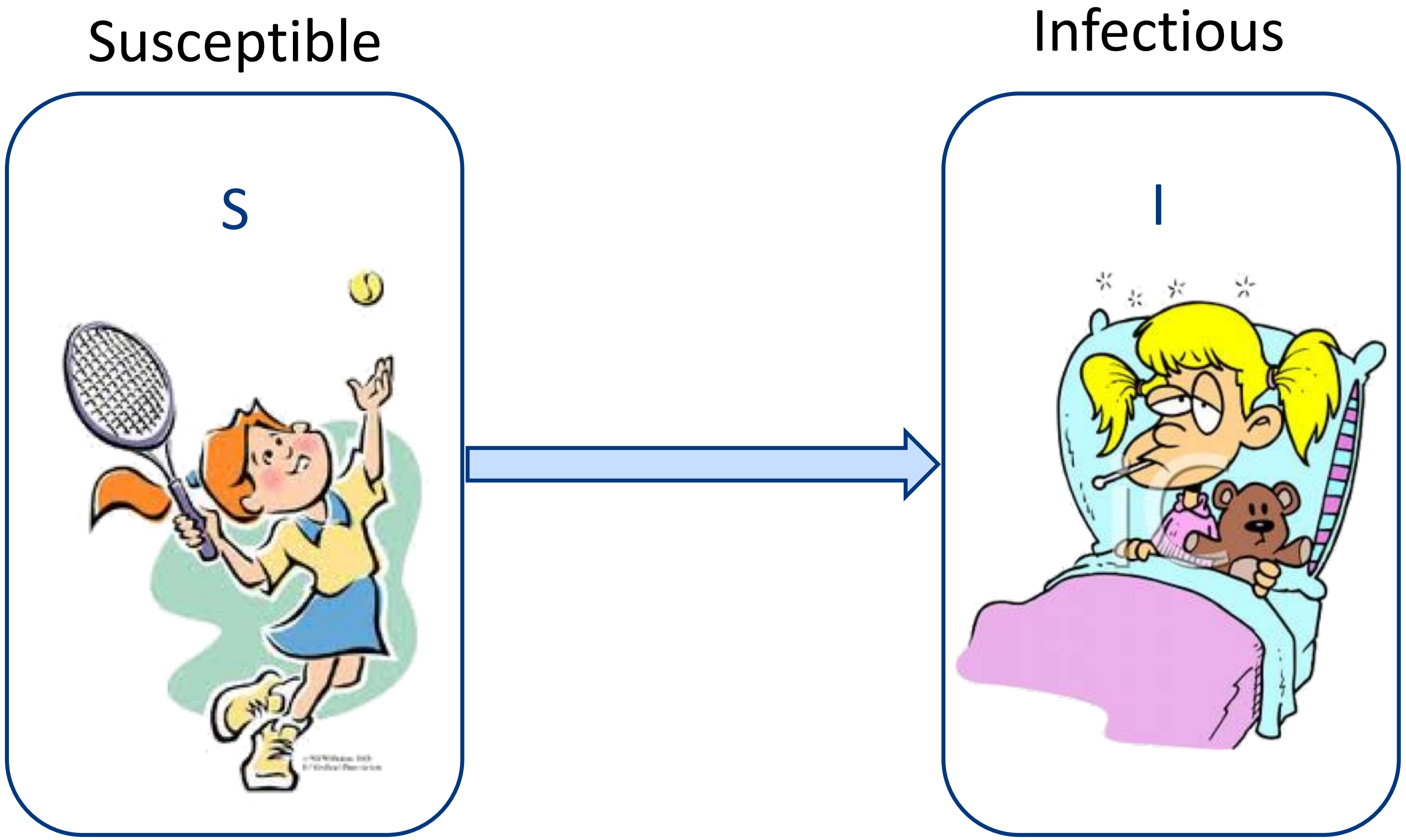
1. **Compartmentalization**: each individual is classified into distinct states. The simplest classification assumes that an individual can be in one of the states.
2. **Homogeneous Mixing**: each individual has the same chance of coming into contact with an infected individual.



[1927] W. O. Kermack and Ag McKendrick. A Contribution to the Mathematical Theory of Epidemics

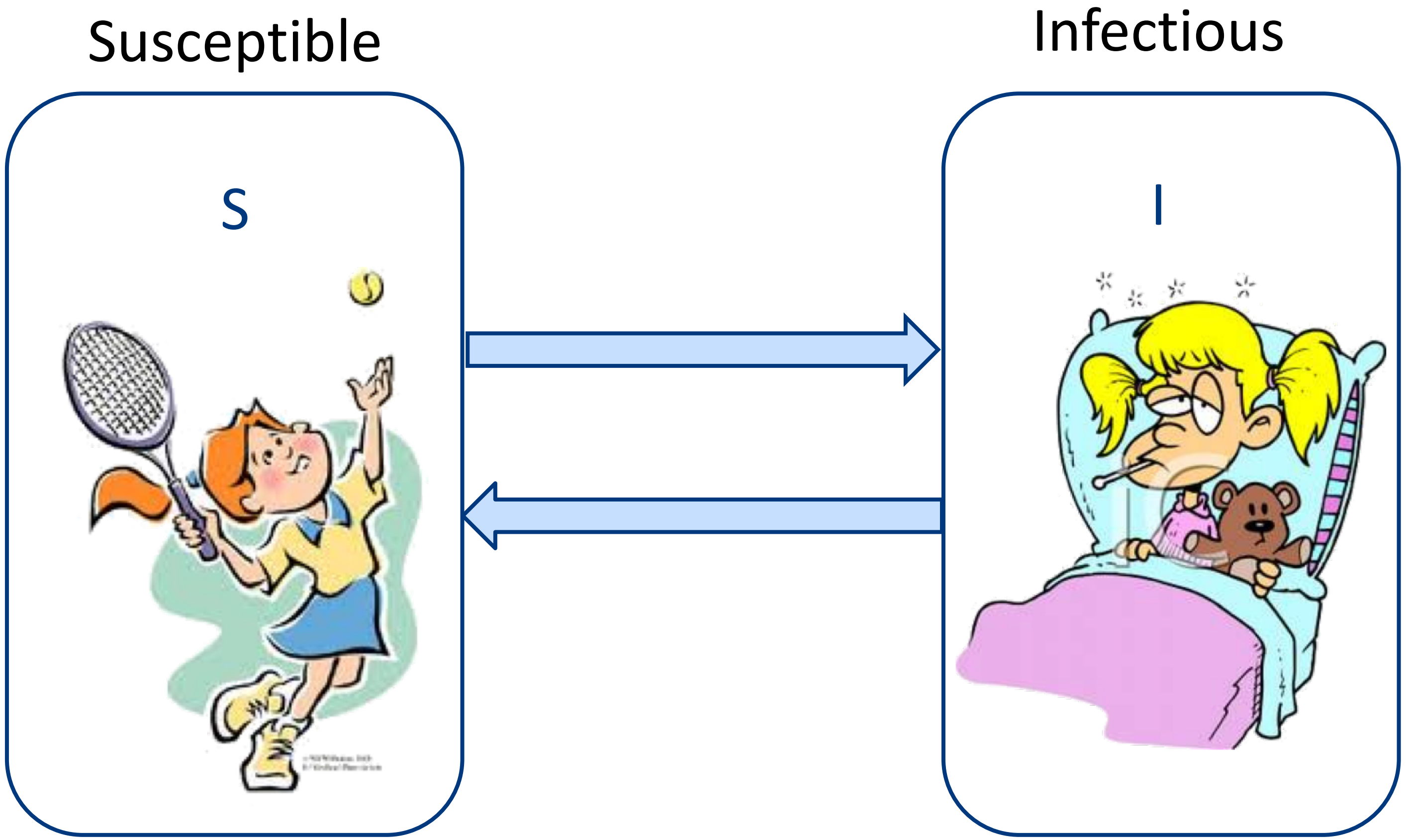
Compartmental Models
[Kermack & McKendrick, 1927]

SI MODEL



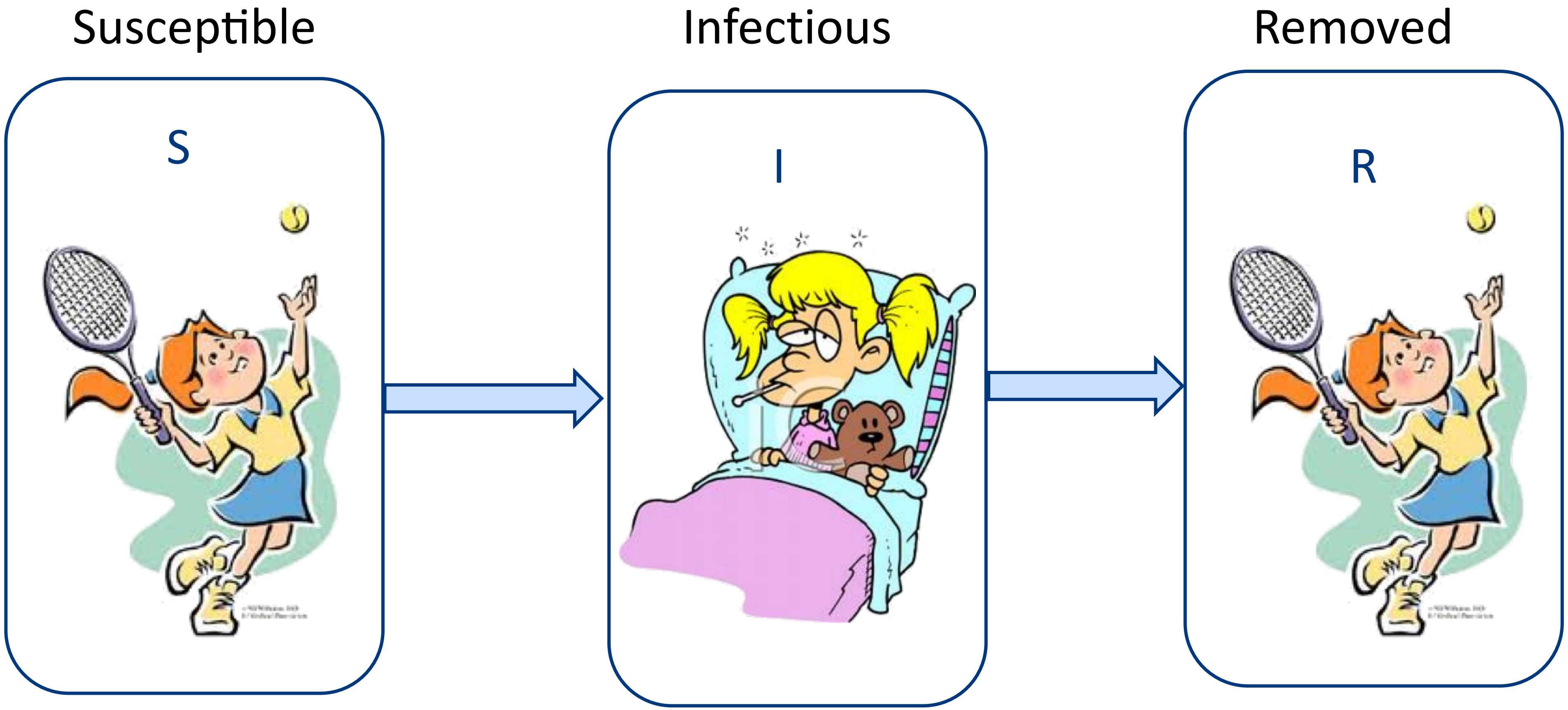
Compartmental Models
[Kermack & McKendrick, 1927]

SIS MODEL: for common cold



Compartmental Models
[Kermack & McKendrick, 1927]

SIR MODEL



Compartmental Models - Limits
[Kermack & McKendrick, 1927]

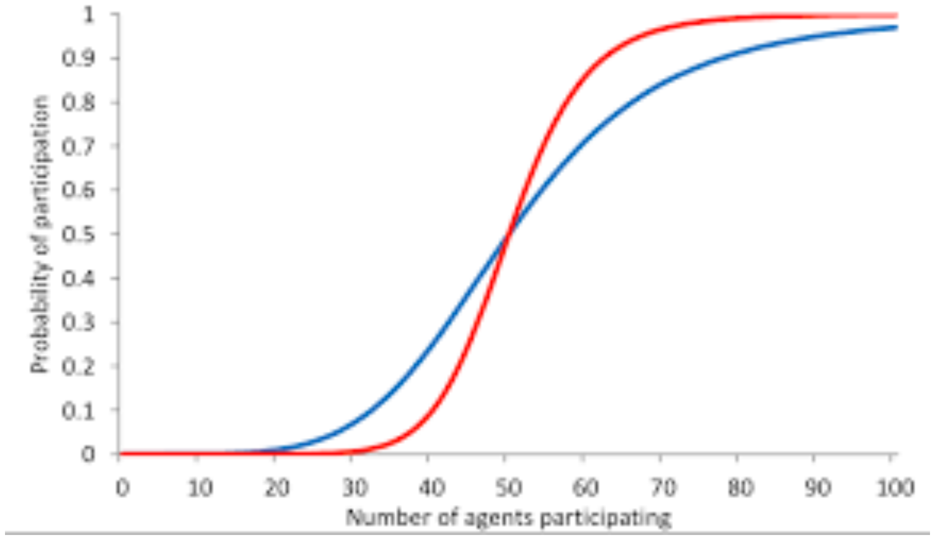
It does not take into account:

- Social structure (i.e., mean field scenario)
- Heterogenous mixing
    - An individual comes into contact only with a restricted set of peers

Threshold Models

[Granovetter, '78]

- Model for:
  - Riots, protests
  - Neighborhoods in cities changing ethnic composition
- Each person i has a threshold t_i
- Node i will adopt the behavior ⟺ at least t_i other people are adopters:
  - small t_i: *early adopter*
  - large t_i: *late adopter*



[1978] M. Granovetter. Threshold models of collective behavior.

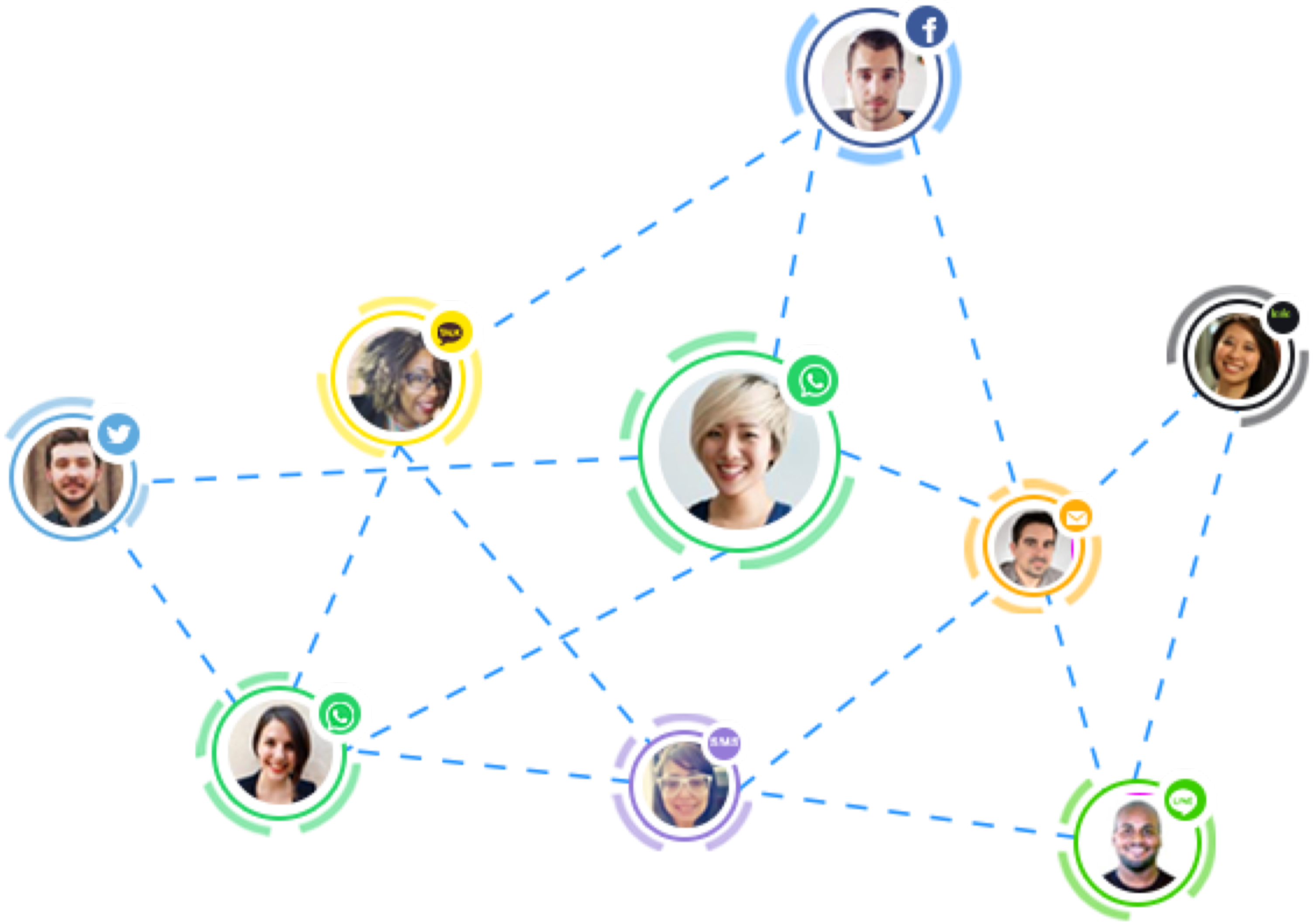
Threshold Model- Limits

[Granovetter, ‘78]

It does not take into account:

- Social structure (i.e., mean field scenario)
- It matters who the early adopters are, not just how many
- Non monotone behavior – dropping out if too many people adopt
- Modeling thresholds:
  - Richer distributions
  - Deriving thresholds from more basic assumptions

What about social interactions?



Intro to:
Complex Networks

Complex

[adj., v. kuh m-pleks, kom-pleks; n. kom-pleks]

–adjective

1.

composed of many interconnected parts; compound; composite: a complex highway system.

2.

characterized by a very complicated or involved arrangement of parts, units, etc.: complex machinery.

3.

so complicated or intricate as to be hard to understand or deal with: a complex problem.

Source: Dictionary.com

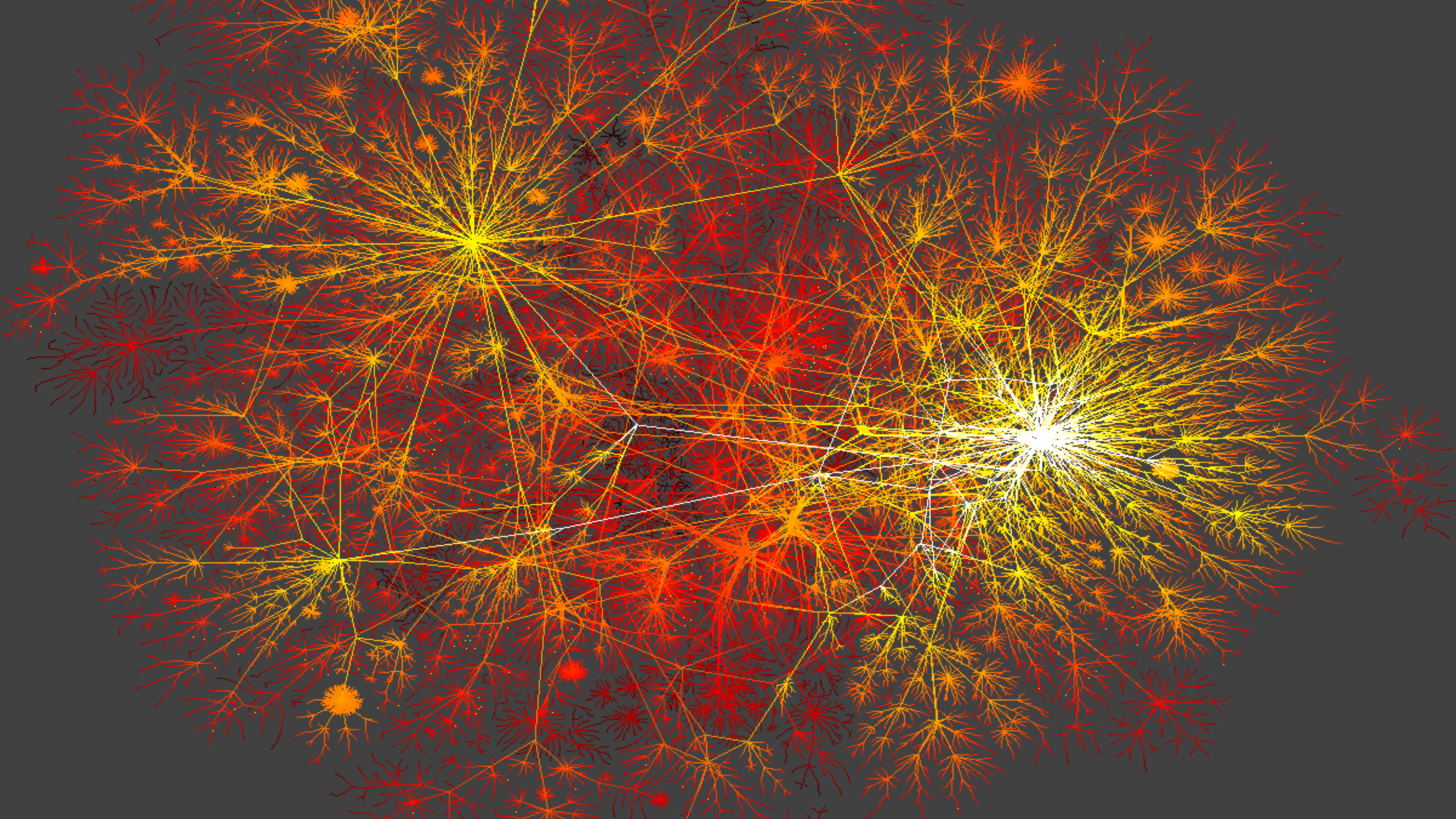
Complexity, a **scientific theory** which asserts that some systems display behavioral phenomena that are completely inexplicable by any conventional analysis of the systems' constituent parts. These phenomena, commonly referred to as **emergent behaviour**, seem to occur in many complex systems involving living organisms, such as a stock market or the human brain.

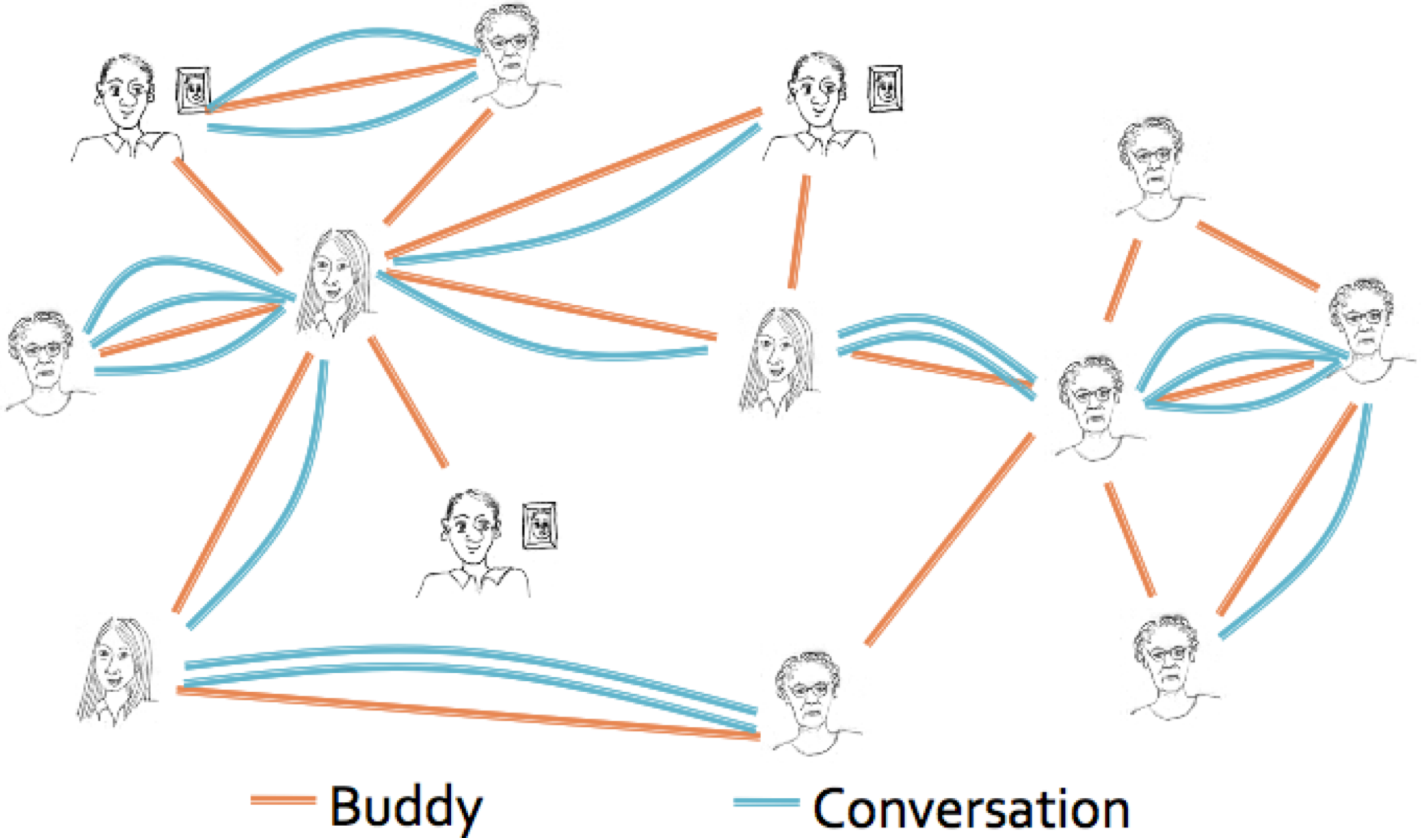
Source: John L. Casti, Encyclopædia Britannica

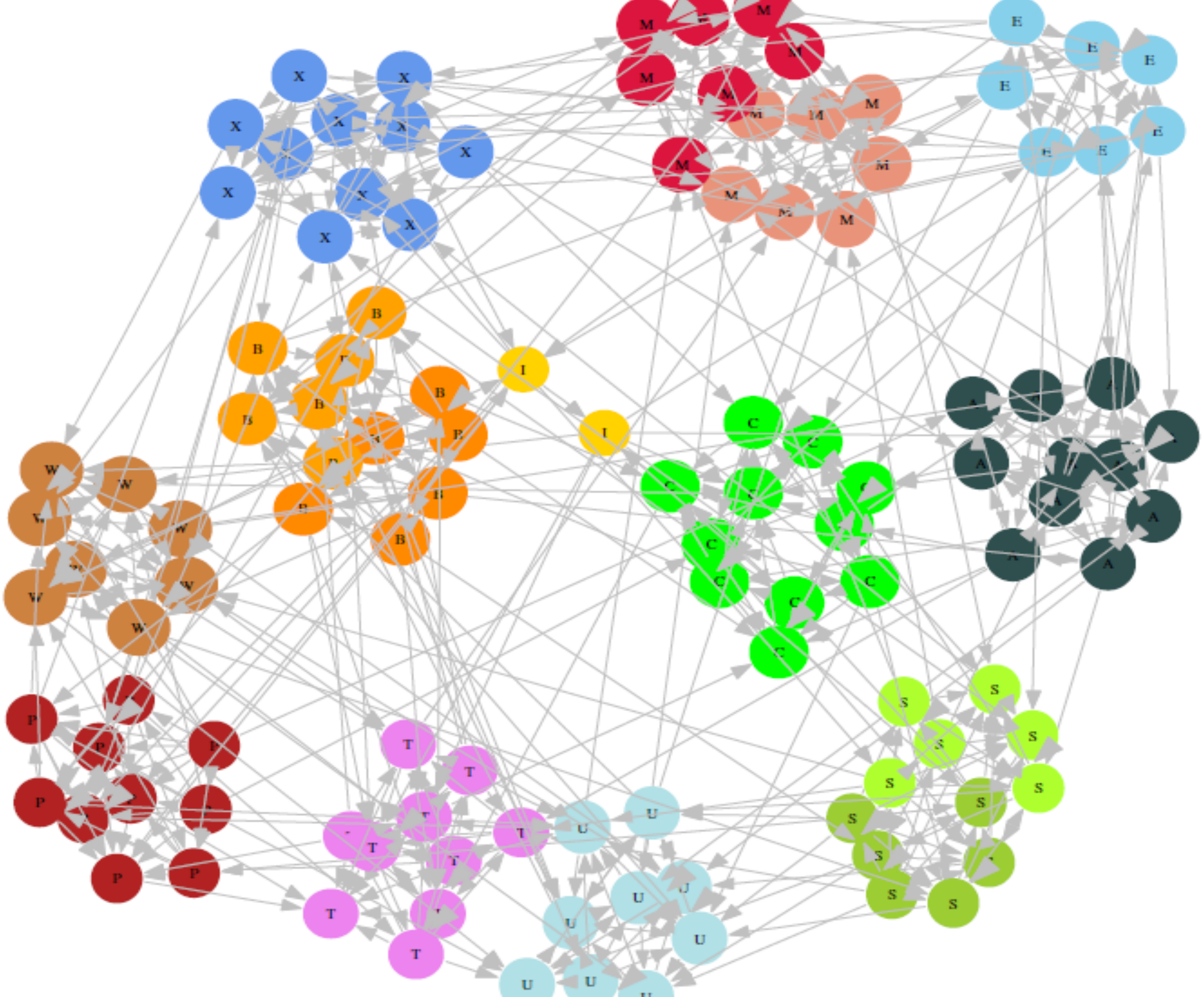
Complexity

Behind each complex system there is a **network**, that defines the interactions between the components.

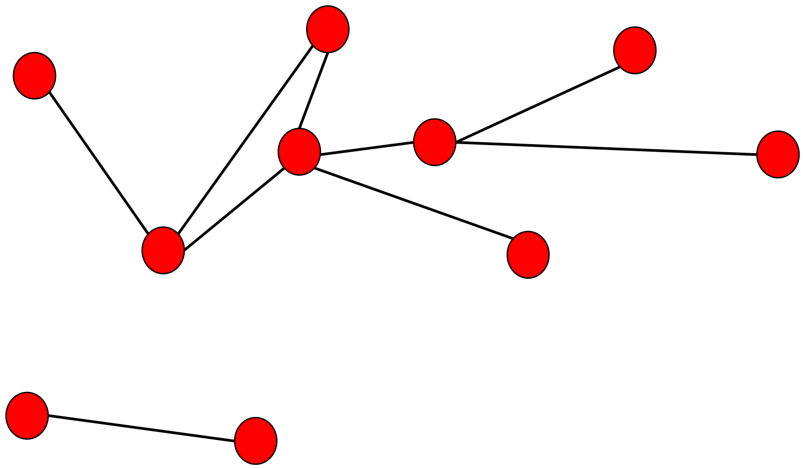
Keith Shepherd's "Sunday Best". http://baseballart.com/2010/07/shades-of-greatness-a-story-that-needed-to-be-told/




Buddy
Conversation


M
E
X
B
I
C
A
W
P
T
U
S

Networks in a nutshell

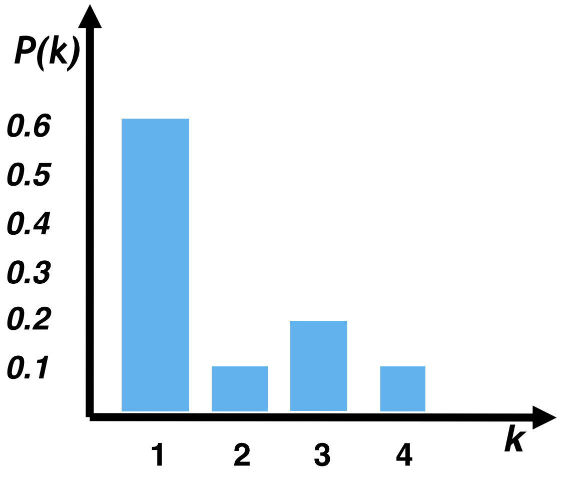


Degree distribution

P(k): probability that a randomly chosen vertex has degree k

N_k = # nodes with degree k

$P(k) = N_k / N$ ➔ plot



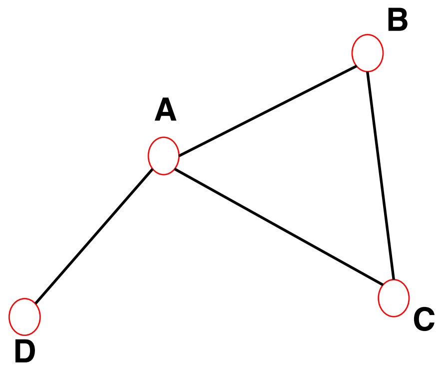
Network (graph)

a graph G=(N,L) is composed by a set of nodes, N, connected by links, L.

Degree

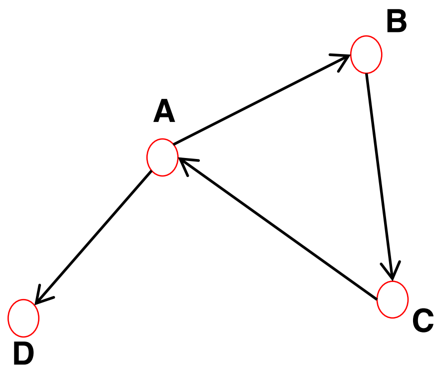
number of links incident to a given node.

Networks in a nutshell



The *distance (shortest path, geodesic path)* between two nodes is defined as the **number of edges along the shortest path connecting them**.

*If the two nodes are disconnected, the distance is infinity.



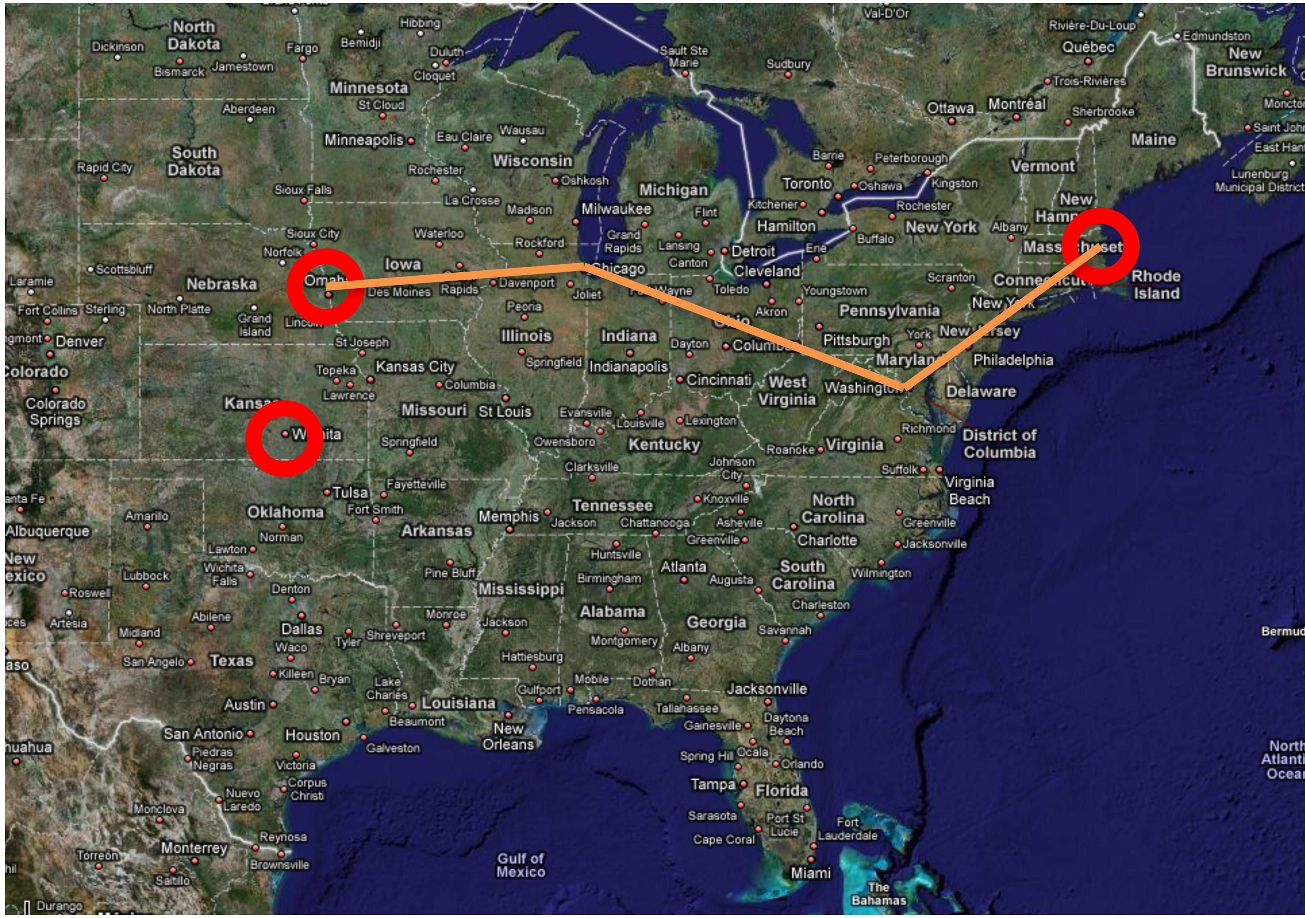
In directed graphs each path needs to follow the direction of the arrows. Thus in a digraph the distance from node A to B (on an AB path) is generally different from the distance from node B to A (on a BCA path).

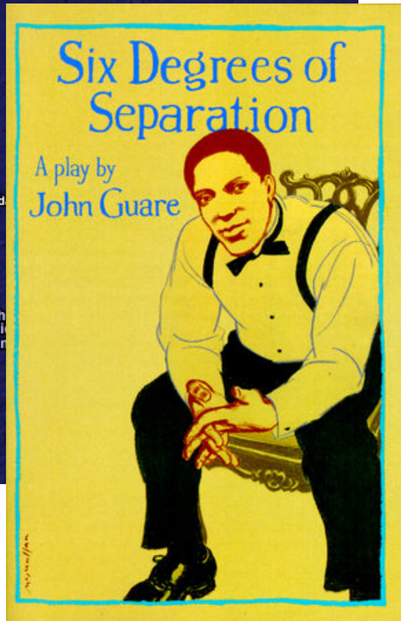
Real world Networks properties:
Six Degrees of separation…

"There exist a path of length 6 that connects any 2 nodes in a social graph"

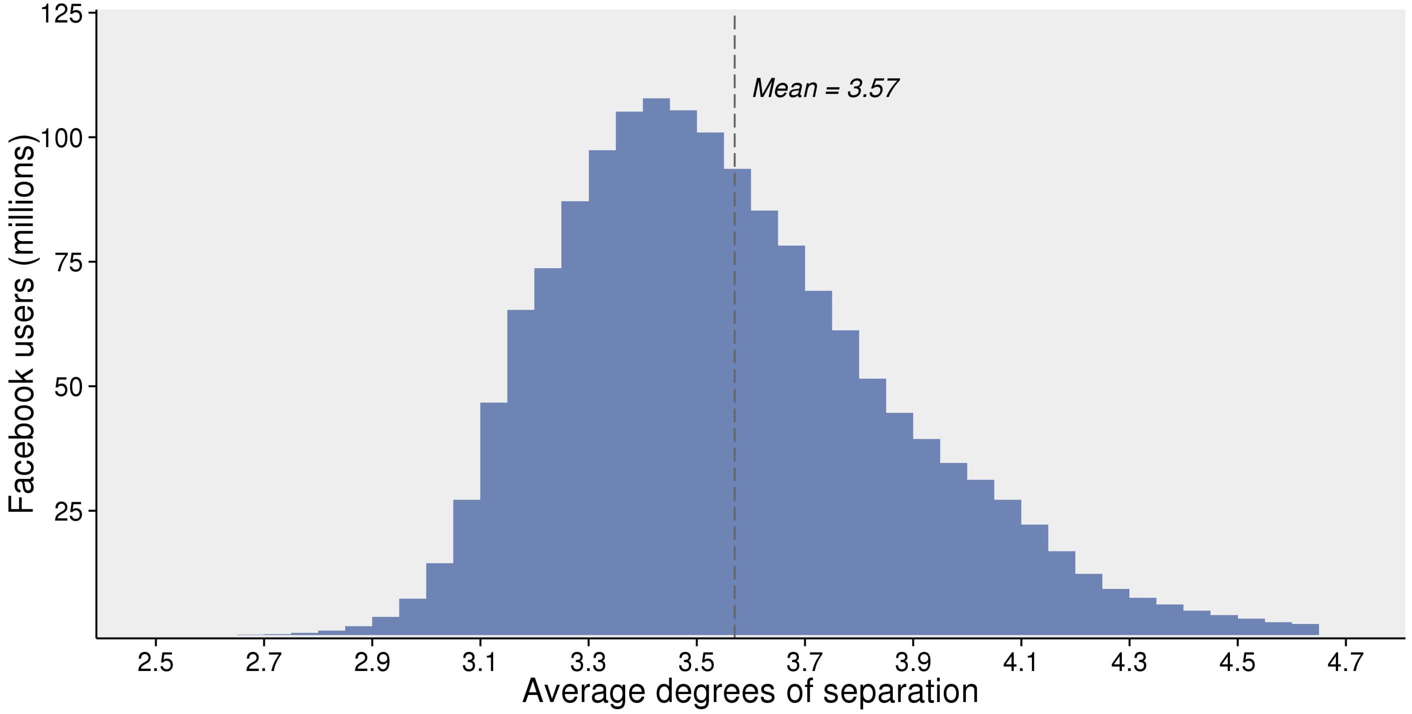
Stanley Milgram

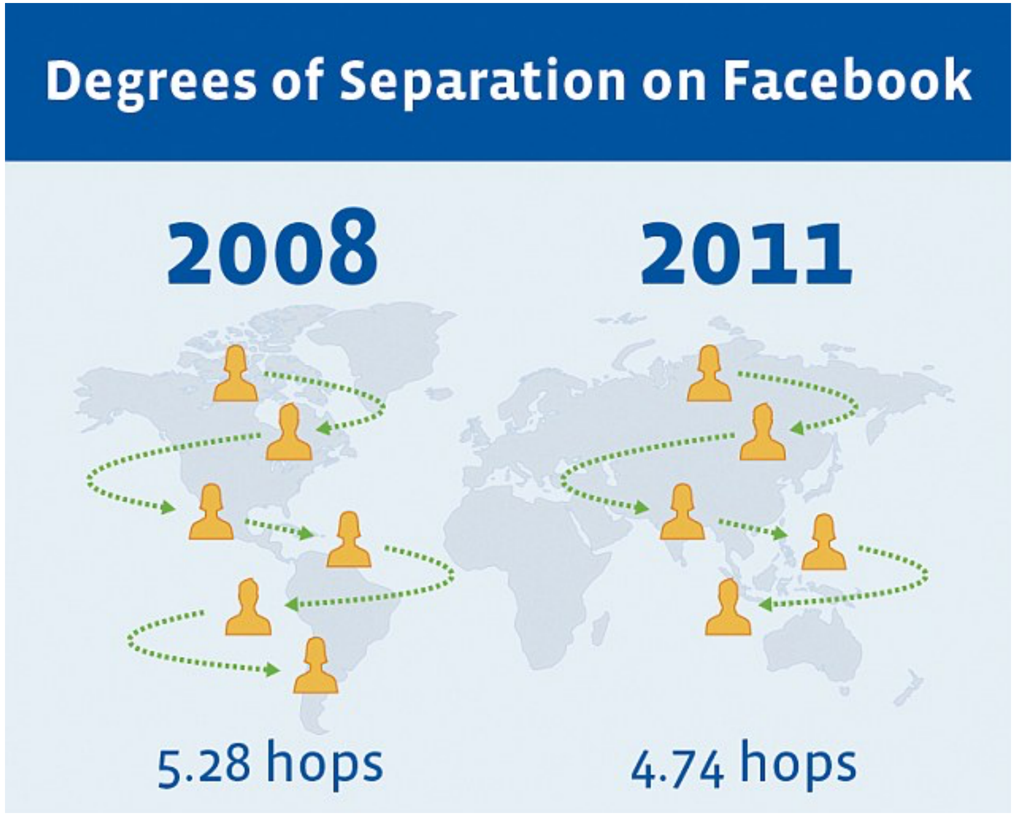
- 160 people in Omaha (Nebraska) & Wichita (Kansas)
- 1 recipient near Boston





… became three (and half) in Facebook



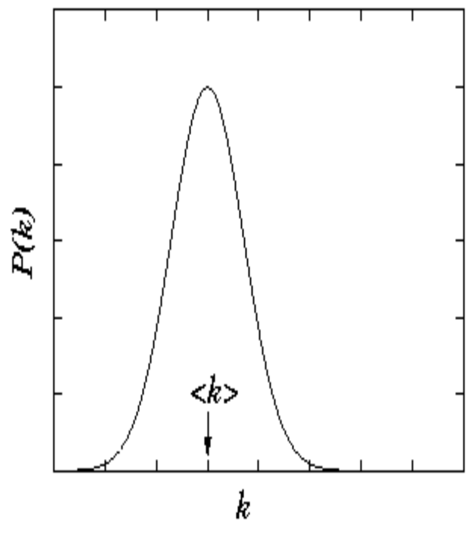


Backstrom, L., Boldi, P., Rosa, M., Ugander, J., & Vigna, S. (2012). Four degrees of separation. *Proceedings of the 4th Annual ACM Web Science Conference*, 33-42.

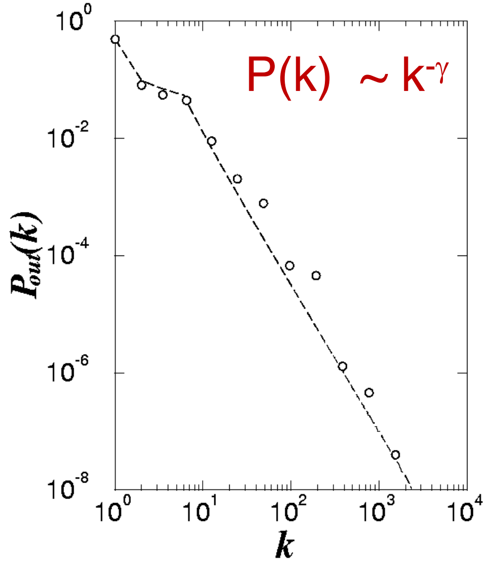
Real world Networks properties:
Networks are Scale-free

Often the degree distribution follow a power law:

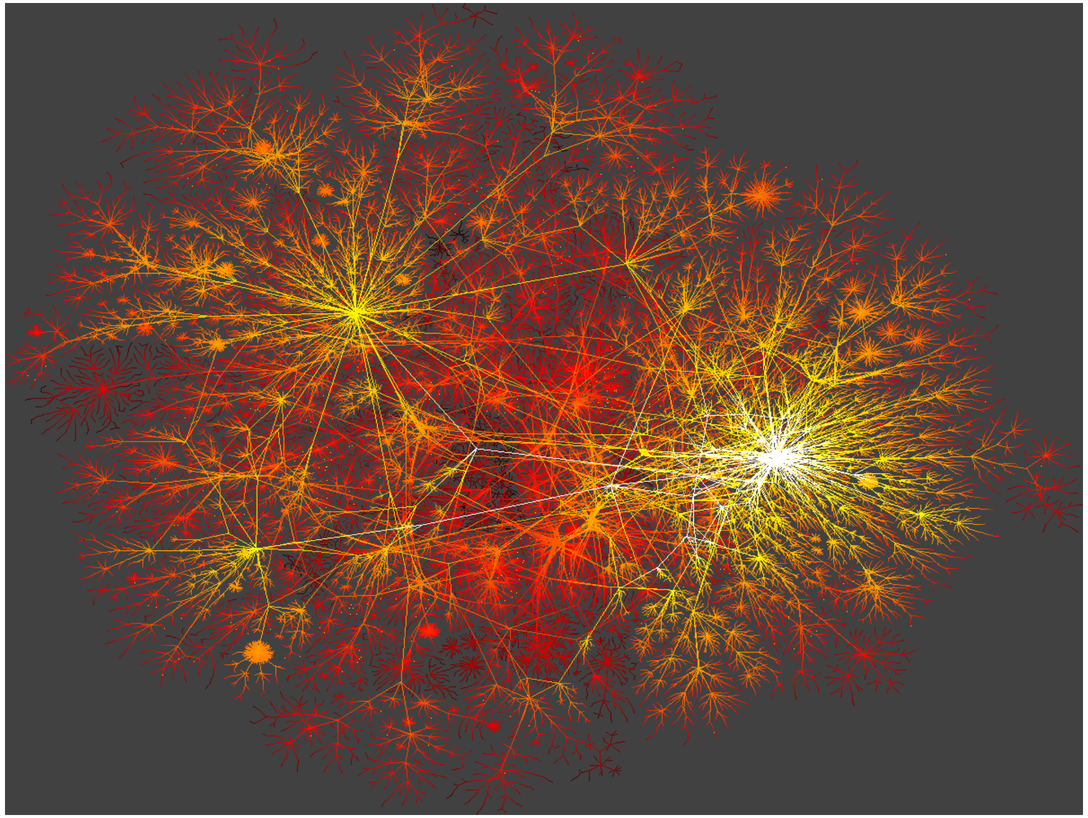
- *Presence of few power-nodes (hubs)*
- *Average degree is not a representative indicator*



Expected

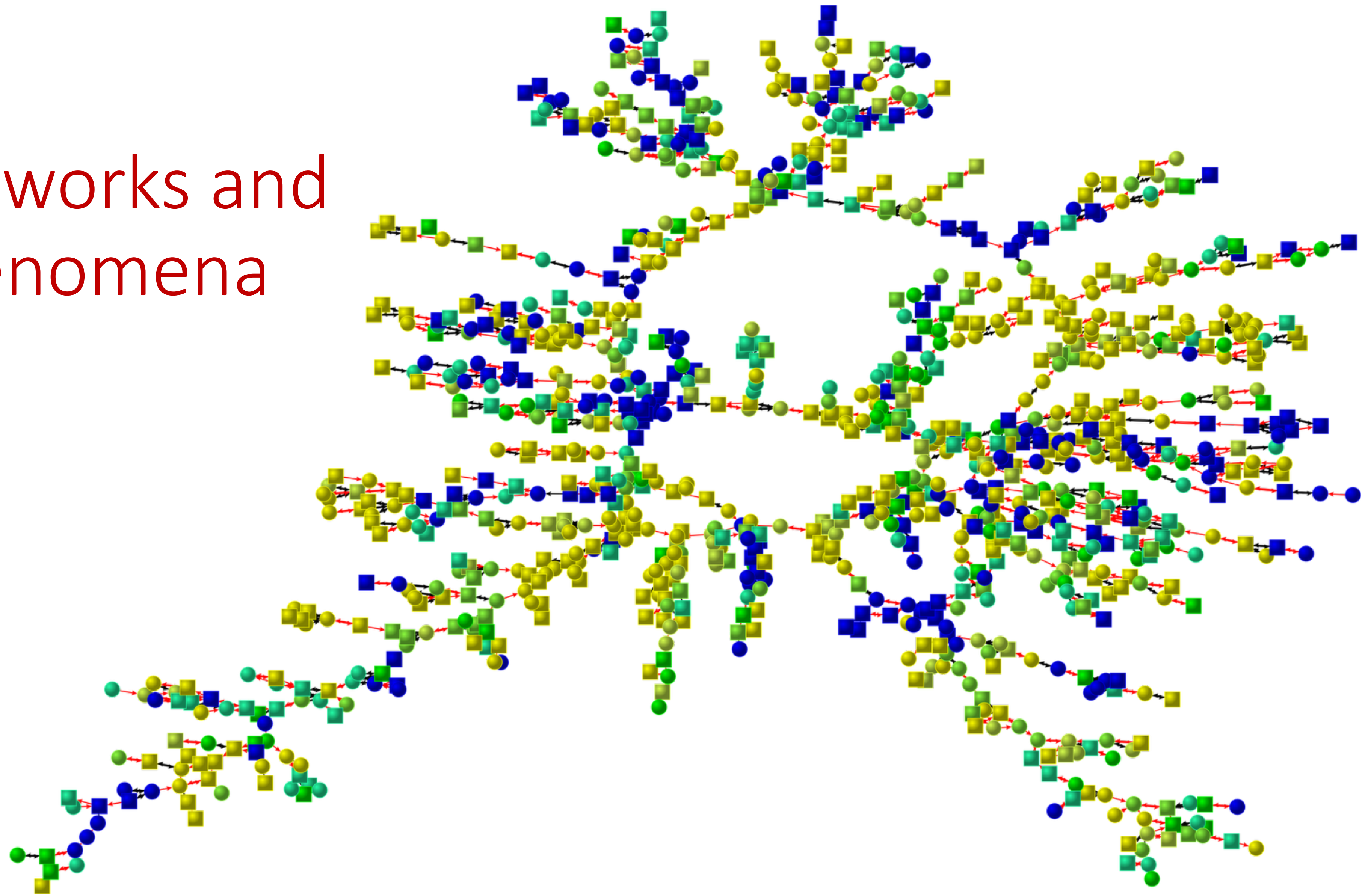


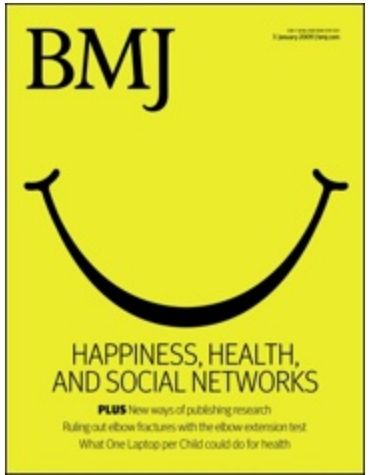
Found



R. Albert, H. Jeong, A-L Barabasi, *Nature*, 401 130 (1999).

Complex Networks and Diffusive Phenomena





James H. Fowler, Nicholas A. Christakis.
Dynamic Spread of Happiness in a Large Social Network: Longitudinal Analysis Over 20 Years in the Framingham Heart Study
British Medical Journal 337 (4 December 2008)

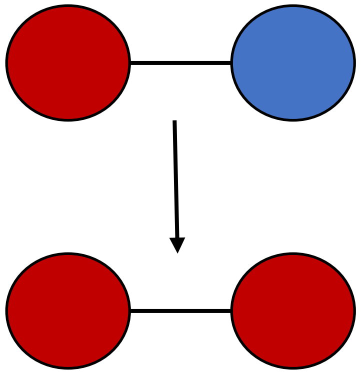
Breaking down a diffusion process

Diffusion *requires* network structure!

- It happens only when the carries of the diseases/virus/idea are **connected to each other**.

Diffusive phenomena can modeled by describing:

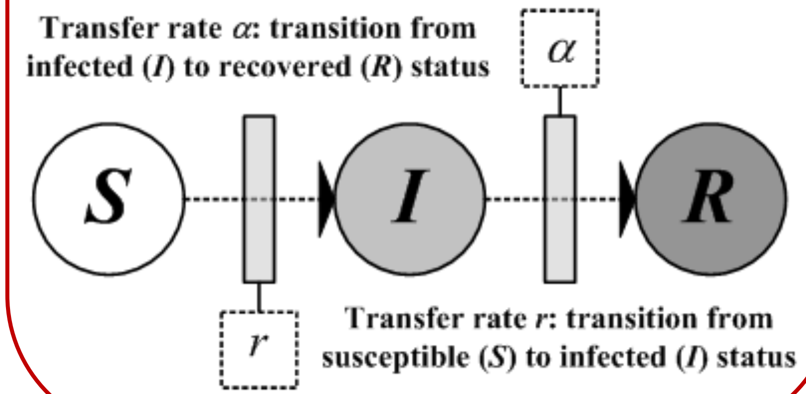
- *"node statuses"*
- *"transition rules"*



Example
SIR model

Three node statuses:
(**S**)usceptible,
(**I**)nfected,
(**R**)ecovered

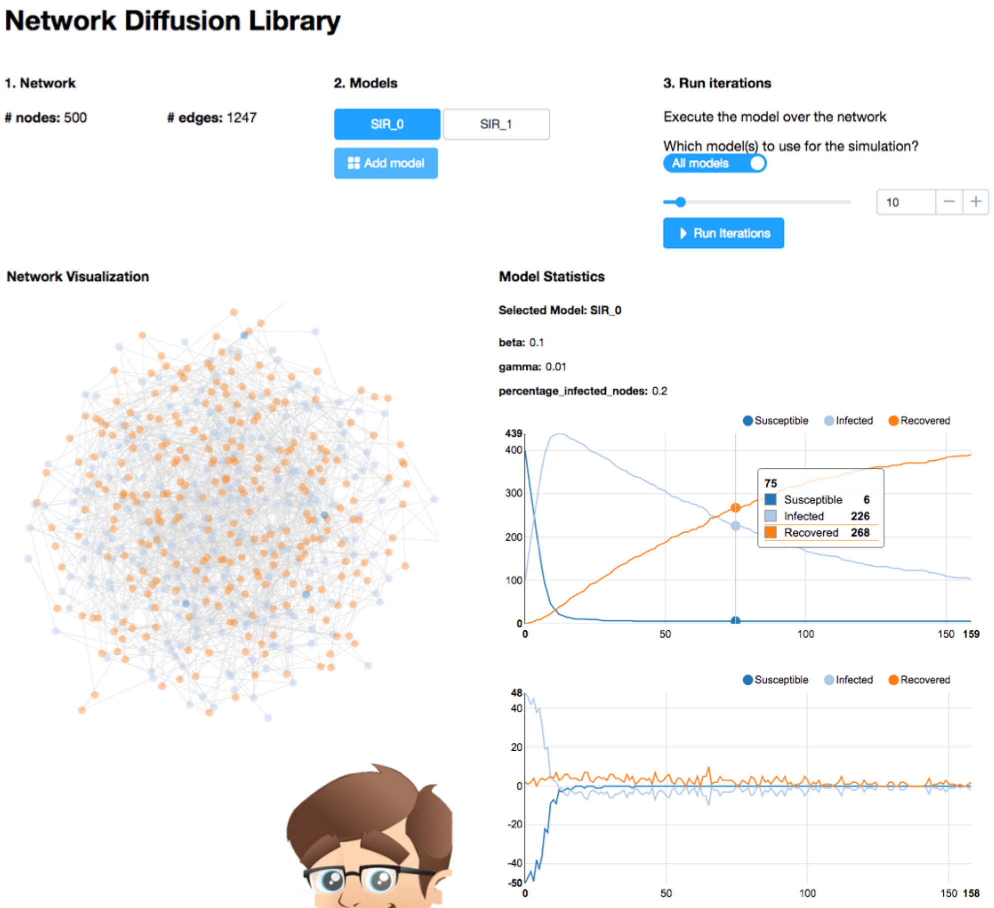
Two transition rules:
S->I;
I->R



A Network Diffusion Framework!

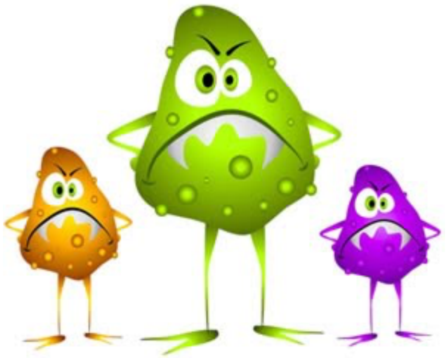
Simulate
Epidemics and
Opinion Dynamics
processes

Unfolding on top of
complex network structures



Programmatically
and **Visually**!

Available Models



Epidemics

(11 Models)

- SI / SIS / SIR
- SEIS / SEIR /SWIR
- Threshold / Profile / Profile-Threshold / Threshold-Blocked
- Independent Cascades

Opinion Dynamics

(6 Models)

- Majority Rule
- Voter / Q-Voter
- Sznajd
- Cognitive Opinion Dynamics
- Algorithmic Bias

A single workflow, two type of users!

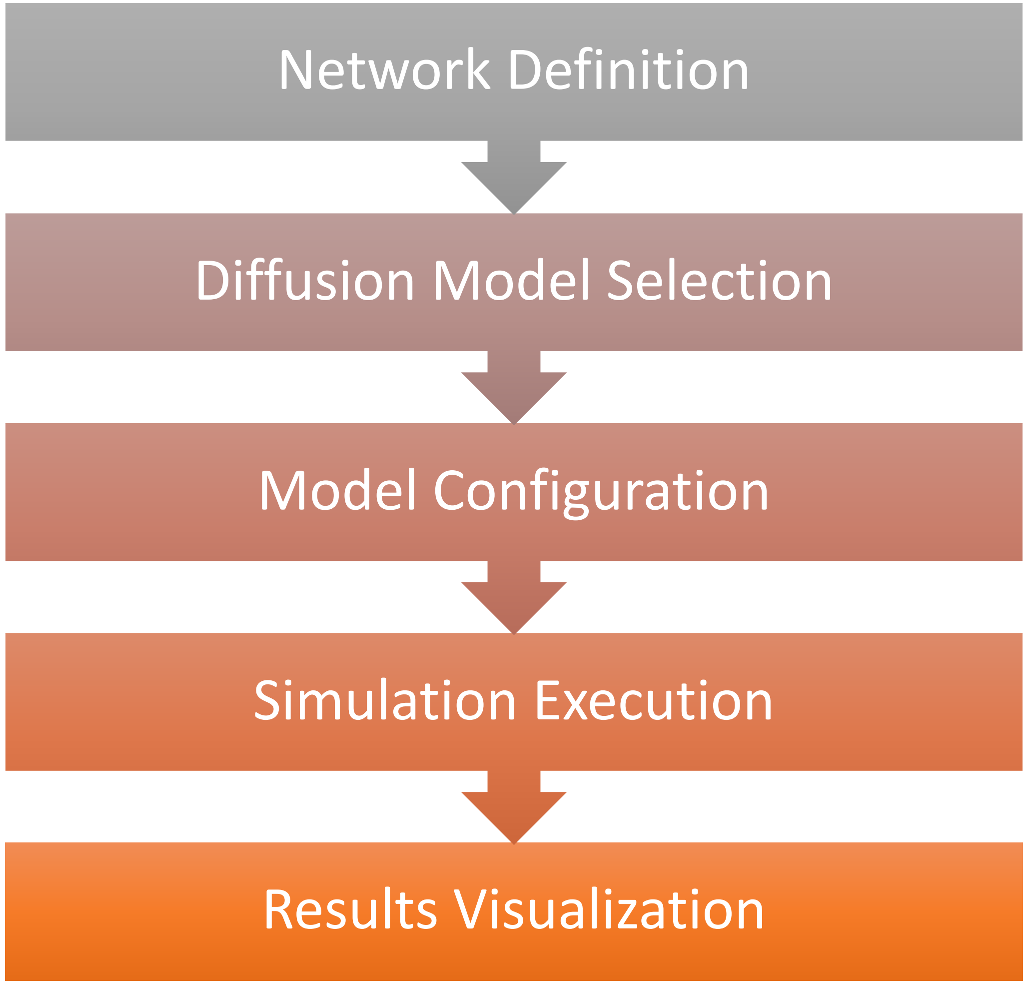
NDlib provide a common workflow to both programmers and analysts:

Programmers:

- Unified interface for several diffusion model
- Results Visualization facilities
- I/O standardization
- Extensibility

Analysts:

- Visual (web-based) platform
- Experiment configuration/execution
- Analytics as-a-service



NDlib Network Diffusion Library

Programmer:

SIR Code Example

A simple, unified, interface:

- Load the Graph
- Select and configure the model
- Run the simulation

All models follow the same programmatic pattern and produce standardized results

```
import networkx as nx
import ndlib.models.ModelConfig as mc
import ndlib.models.epidemics.SIRModel as sir

# Network topology
g = nx.erdos_renyi_graph(1000, 0.1)

# Model selection
model = sir.SIRModel(g)

# Model Configuration
cfg = mc.Configuration()
cfg.add_model_parameter('beta', 0.01)
cfg.add_model_parameter('gamma', 0.005)
cfg.add_model_parameter("percentage_infected", 0.05)
model.set_initial_status(cfg)

# Simulation execution
iterations = model.iteration_bunch(200)
```

Programmer:
Visual Analysis

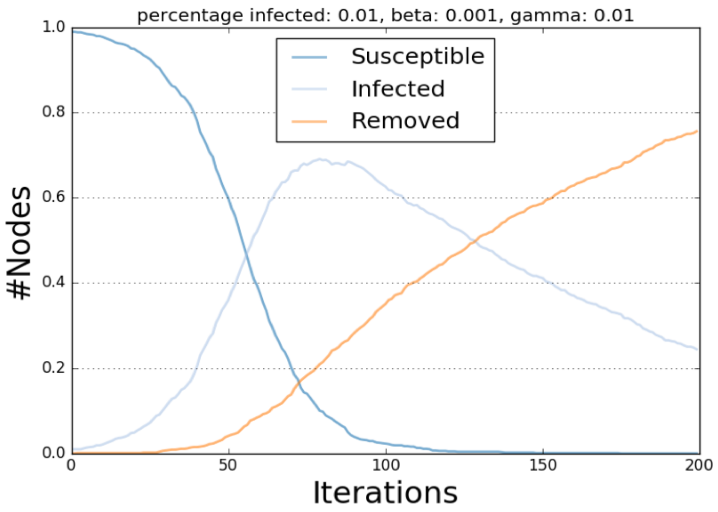
NDlib implements visualization facilities

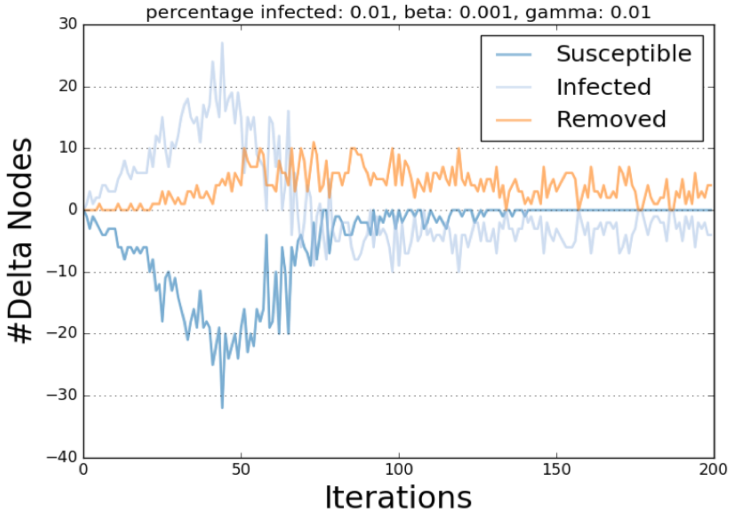
Base Viz

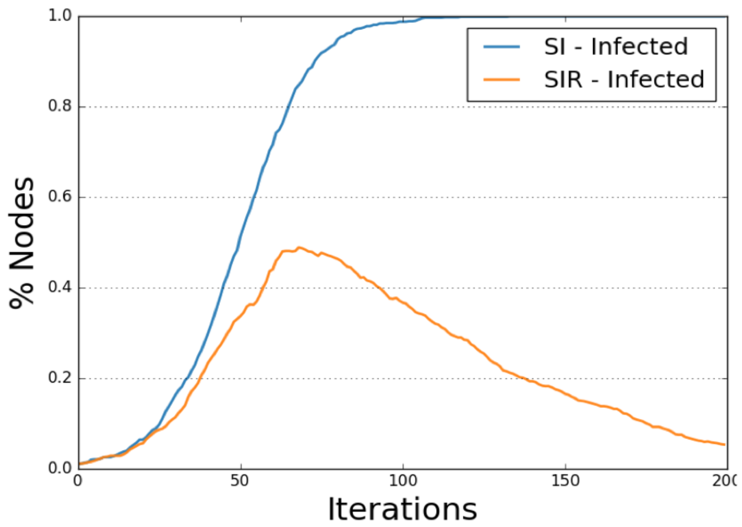
- Diffusion Trends
- Prevelence

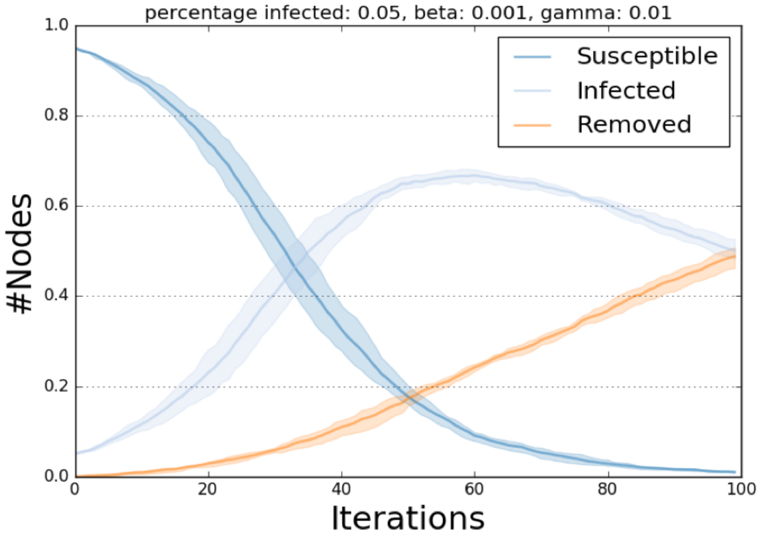
Advanced Viz

- Compare Models
- Multiple Run









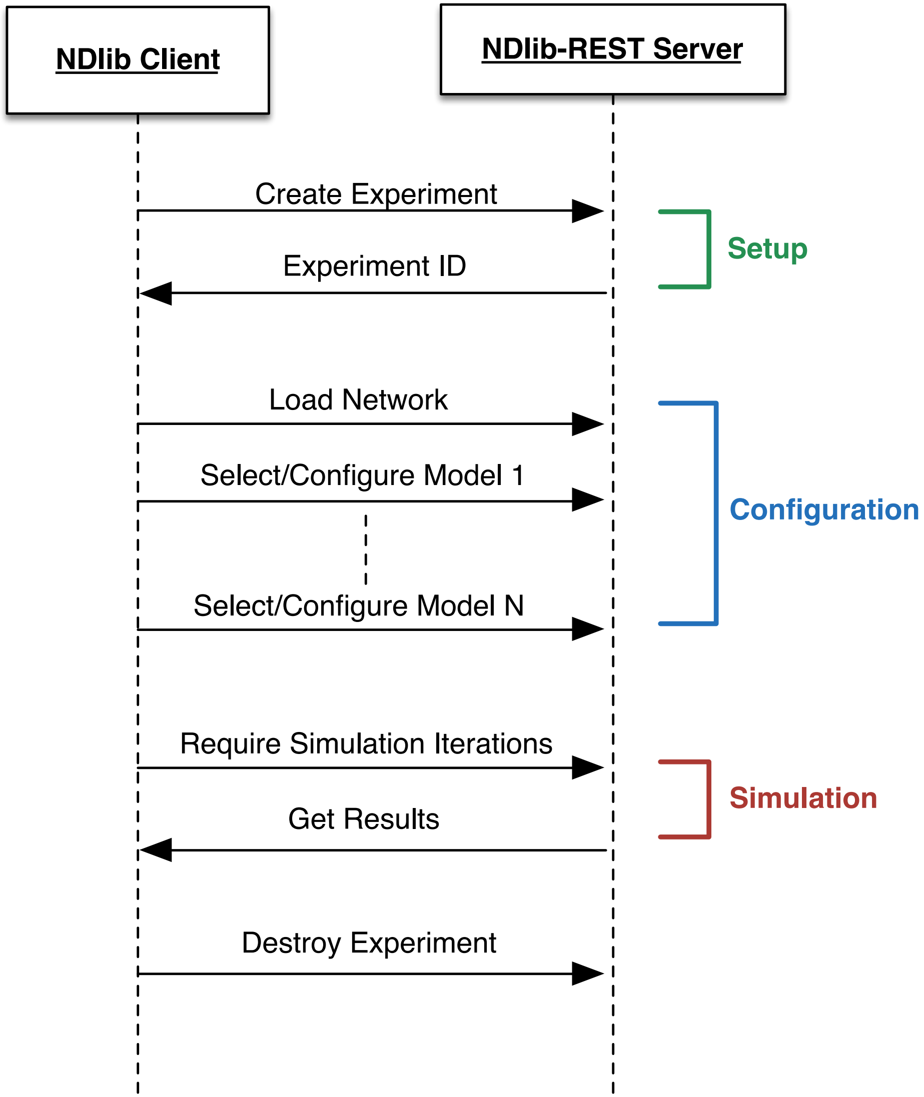
Programmer:
Remote Experiments

NDlib offers a **remote experiment server** that, using a REST-full API, allows to:

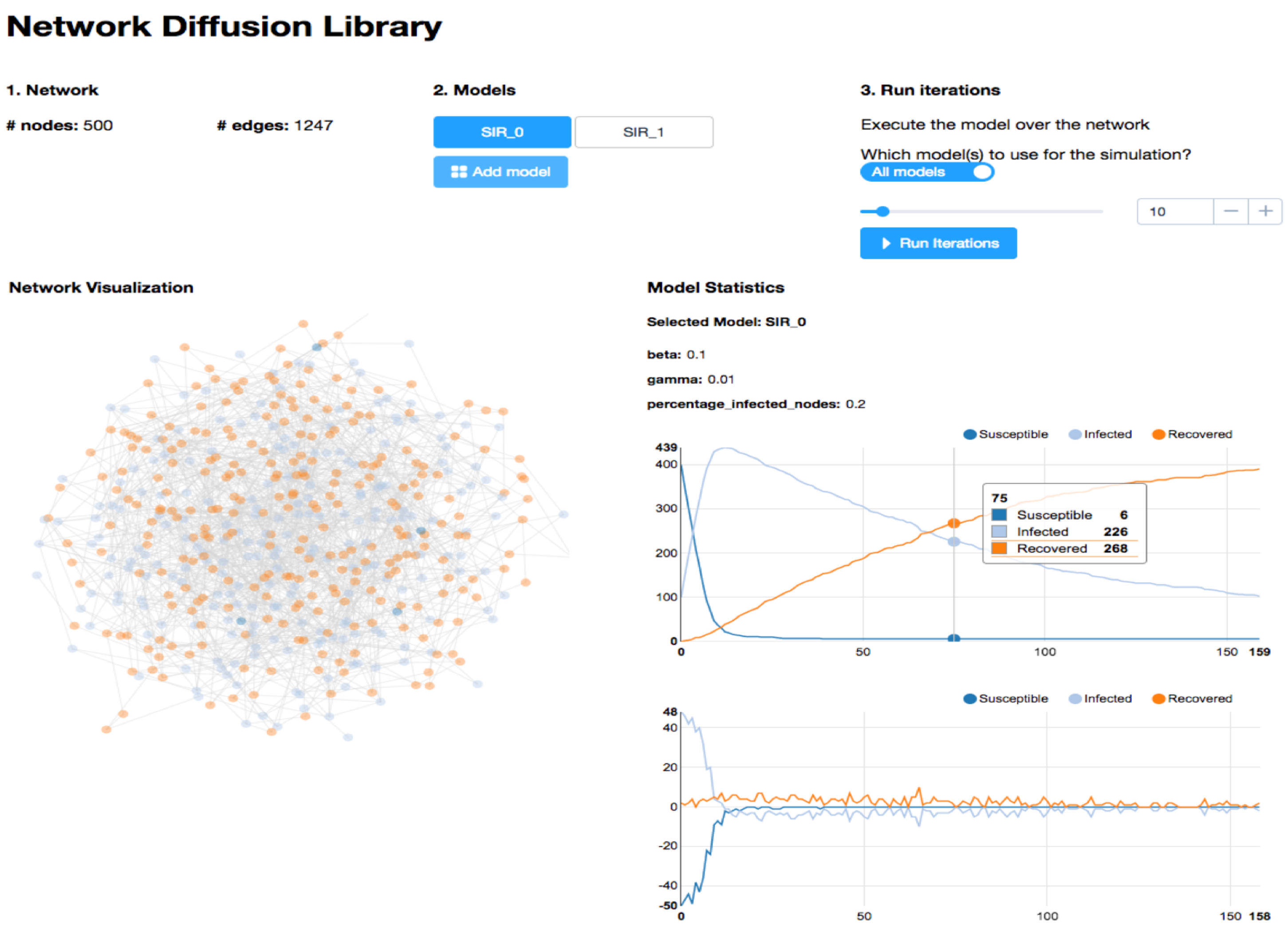
- Create Ndlib experiments
- Configure them
- Execute them remotely

NDlib-REST aims to:

1. Decouple experiment definition/execution
2. Increase scalability



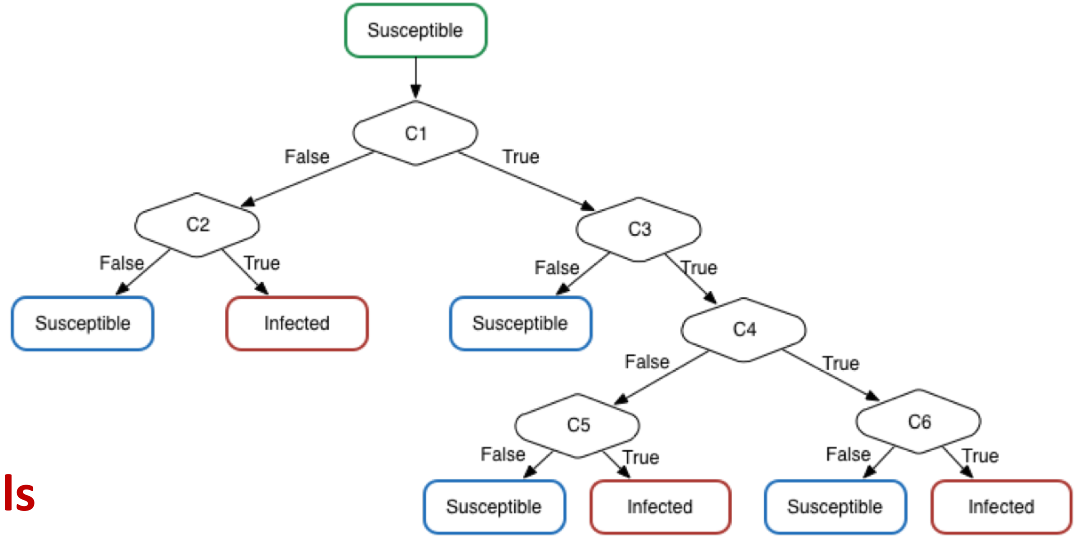
Analyst: Visual Simulation



Ndlib 4.0:
Advanced Features

Composite model definition

- Design diffusive models defining their transition rules as *trees* of atomic actions (compartments)

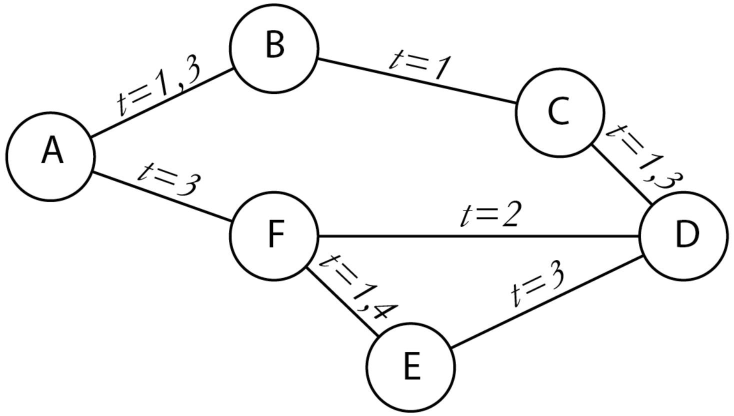


Support for Dynamic Network models

- Integration with DyNetX
  (ad-hoc library by CNR-UNIPI)

NDQL: Network Diffusion Query Language

- High-level query language
  for defining diffusion processes




```
CREATE_NETWORK g1
TYPE erdos_renyi_graph
PARAM n 300
PARAM p 0.1

MODEL SI

STATUS Susceptible
STATUS Infected

# Compartment definitions

COMPARTMENT c1
TYPE NodeStochastic
PARAM rate 0.1
TRIGGER Infected

# Rule definitions

RULE
FROM Susceptible
TO Infected
USING c1

# Model configuration

INITIALIZE
SET Infected 0.1

EXECUTE SI ON g1 FOR 100
```

State of art competitors analysis

Table 1 Diffusion libraries and tools. A c… on models, if any, natively implemented within each library

Name	Lang.	Net. Model	Active	License
NDLIB	Python	NetworkX DyNetX	✓	BSD
Epigrass	Python	NetworkX	✓	GPL
GEMFsim	Python	NetworkX	✓	-
Nepidemix	Python	NetworkX	✓	BSD
EoN	Python	NetworkX	✓	MIT
Epydemic	Python	NetworkX	✓	GPL
ComplexNetworkSim	Python	NetworkX		BSD
Nxsim	Python	NetworkX		Apache
EpiModel	R	iGraph	✓	GPL
RECON	R	adhoc	✓	Various
Sisspread	C	adhoc	✓	GPL
GLEaMviz	C++ Python	adhoc	✓	SaaS

Table 2 Runtimes comparison. SIR model (parameters: $\beta = 0.001$, $\gamma = 0.01$), initial infected 5%, number of iterations 25, network model Barabasi–Albert graph. The compared libraries are organized by programming language

Library	Graph size (nodes)			
	10^3	10^4	10^5	10^6
NDLIB	0.060s	0.655s	7.554s	90.443s
ComplexNetworkSim	0.264s	3.152s	43.145s	576.072s
Nepidemix	0.283s	3.241s	43.190s	525.768s
EpiModel	0.025s	0.141s	2.289s	45.725s

Publications

NDlib publications:

- **“NDlib: a Python Library to Model and Analyze Diffusion Processes Over Complex Networks”**
  G. Rossetti, L. Milli, S. Rinzivillo, A. Sirbu, D. Pedreschi, F. Giannotti. International Journal of Data Science and Analytics. 2017. DOI:0.1007/s41060-017-0086-6
- **“NDlib: Studying Network Diffusion Dynamics”**
  G. Rossetti, L. Milli, S. Rinzivillo, A. Sirbu, D. Pedreschi, F. Giannotti. IEEE International Conference on Data Science and Advanced Analytics, DSAA. 2017.
- **“NDlib: A Python Library to model and analyze diffusion processes over complex networks”**
  G. Rossetti, L. Milli, S. Rinzivillo. Demo @ The Web Conference, WWW, 2018.

Publications using NDlib:

- **“Information Diffusion in Complex Networks: The Active/Passive Conundrum”**
  L. Milli, G. Rossetti, D. Pedreschi, F. Giannotti International Conference on Complex Networks and their Applications, 2017. DOI:10.1007/978-3-319-72150-7_25
- **“Diffusive Phenomena in Dynamic Networks: a data-driven study”**
  L. Milli, G. Rossetti, D. Pedreschi, F. Giannotti. 9th Conference on Complex Networks, CompleNet, 2018.

NDlib
Network Diffusion Library

When:

Rigth now, NDlib v4.0.1 is out!

Where:

- Pypi: https://pypi.python.org/pypi/ndlib
- GitHub NDlib: https://github.com/GiulioRossetti/ndlib
- GitHub NDlib-REST: https://github.com/GiulioRossetti/ndlib-rest
- Documentation: http://ndlib.readthedocs.io/
- SoBigData: http://www.sobigdata.eu

Who:

python Package Index

GitHub

Read the Docs

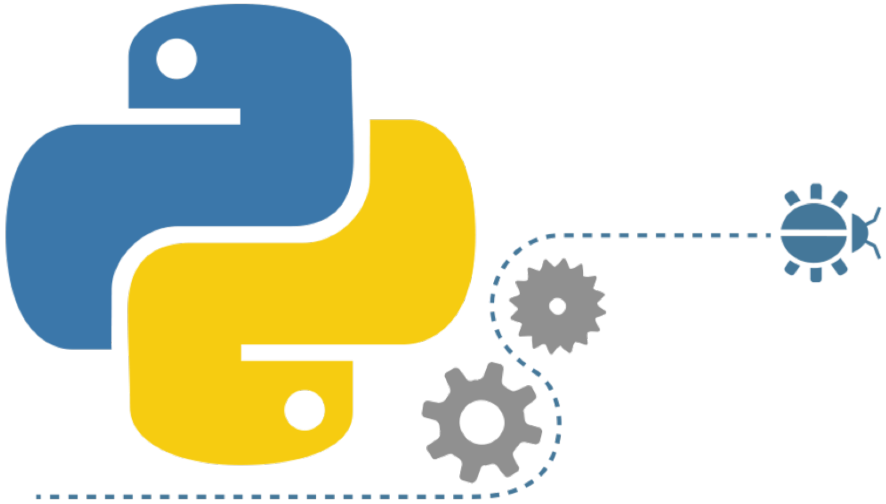
SoBigData

NDlib
Network Diffusion Library

Demo/Tutorial

How many people here know a programming language?

How many people here have used Python?



NDlib Network Diffusion Library

Where to start…

Install **Anaconda**
(Python 3 version!)
https://www.anaconda.com/download/

Anaconda is a python distribution for data scientists.

It provides out-of-the box support for:

- Data modeling
- Analysis
- Visualisation
- …

ANACONDA
Powered by Continuum Analytics

Resources

NDlib tutorial:
https://goo.gl/oGkuyk

NDlib-viz (testing server):
http://sobigdatata2.isti.cnr.it/NDLibViz/

Library Documentation:
http://ndlib.readthedocs.io/