

# Algorithmen zur Visualisierung von Graphen

## Einführung

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Tamara Mchedlidze · **Martin Nöllenburg**  
22.10.2014



## Dozenten



- Tamara Mchedlidze
- `mched@iti.uka.de`
- Raum 307
- Sprechzeiten: nach Vereinbarung



- Martin Nöllenburg
- `noellenburg@kit.edu`
- Raum 319
- Sprechzeiten: nach Vereinbarung

## Dozenten



- Tamara Mchedlidze
- `mched@iti.uka.de`
- Raum 307
- Sprechzeiten: nach Vereinbarung



- Martin Nöllenburg
- `noellenburg@kit.edu`
- Raum 319
- Sprechzeiten: nach Vereinbarung

## Termine

- Montag 15:45 – 17:15 Uhr, Raum 301
- Mittwoch 15:45 – 17:15 Uhr, Raum 301
- genauer Zeitplan siehe Webseite

## Webseite

`i11www.iti.kit.edu/teaching/winter2014/graphvis/`

- aktuelle Informationen
- Vorlesungsfolien
- Übungsblätter
- Literatur & Zusatzmaterial
- Skript

## Webseite

`i11www.iti.kit.edu/teaching/winter2014/graphvis/`


- aktuelle Informationen
- Vorlesungsfolien
- Übungsblätter
- Literatur & Zusatzmaterial
- Skript

## Graphenvisualisierung im Master-Studium

Bachelor

Master

Algorithmen 1 & 2  
Theoretische Grundlagen  
Algorithmen für planare Graphen



Algorithmen zur  
Visualisierung von  
Graphen

⋮

VF Algorithmentechnik,  
Theoretische Grundlagen

**Lernziele:** Am Ende der Vorlesung können Sie

- Begriffe, Strukturen und Problemdefinitionen erklären
- behandelte Algorithmen ausführen, erklären und analysieren
- geeignete Algorithmen und Datenstrukturen auswählen und anpassen
- neue Graphenvisualisierungsprobleme analysieren und eigene effiziente Lösungen entwerfen

# Algorithmen zur Visualisierung von Graphen

**Lernziele:** Am Ende der Vorlesung können Sie

- Begriffe, Strukturen und Problemdefinitionen erklären
- behandelte Algorithmen ausführen, erklären und analysieren
- geeignete Algorithmen und Datenstrukturen auswählen und anpassen
- neue Graphenvisualisierungsprobleme analysieren und eigene effiziente Lösungen entwerfen

**Vorkenntnisse:** Algorithmen 1 & 2, Theoretische Grundlagen

**hilfreich:** Algorithmen für planare Graphen

**Lernziele:** Am Ende der Vorlesung können Sie

- Begriffe, Strukturen und Problemdefinitionen erklären
- behandelte Algorithmen ausführen, erklären und analysieren
- geeignete Algorithmen und Datenstrukturen auswählen und anpassen
- neue Graphenvisualisierungsprobleme analysieren und eigene effiziente Lösungen entwerfen

**Vorkenntnisse:** Algorithmen 1 & 2, Theoretische Grundlagen

**hilfreich:** Algorithmen für planare Graphen

**Anforderungen/Zeitbedarf:** 5LP = 150h

- Besuch von Vorlesung und Übung ca. 35h
- Vor-/Nachbereitung ca. 35h
- Bearbeiten der Übungsblätter ca. 40h
- Prüfungsvorbereitung ca. 40h



## Master Informatik

- Algorithm Engineering und Anwendungen (IN4INAEA)
- Design und Analyse von Algorithmen (IN4INDAA)
- Netzwerkalgorithmen (IN4INNWA)
- Algorithmen zur Visualisierung von Graphen (IN4INALGVG)
- **neues Modul:** Graphenvisualisierung+ (mehr dazu gleich)

## Master Informatik

- Algorithm Engineering und Anwendungen (IN4INAEA)
- Design und Analyse von Algorithmen (IN4INDAA)
- Netzwerkalgorithmen (IN4INNWA)
- Algorithmen zur Visualisierung von Graphen (IN4INALGVG)
- **neues Modul:** Graphenvisualisierung+ (mehr dazu gleich)

## Prüfungsmodalitäten

- aktive Übungsteilnahme zur Anmeldung erforderlich (d.h. eigene Lösungen an Tafel vorstellen)
- mündliche Einzelprüfung (ca. 20 Minuten)

# Graphenvisualisierung+

**Hintergrund:** International Symposium on Graph Drawing (GD)  
umfasst einen Graphenvisualisierungswettbewerb  
(Graph Drawing Challenge)



# Graphenvisualisierung+

**Hintergrund:** International Symposium on Graph Drawing (GD)  
umfasst einen Graphenvisualisierungswettbewerb  
(Graph Drawing Challenge)

**Aufgabe:** entwickle für im Vorfeld  
angekündigte Problemstellung  
eine Visualisierungssoftware



# Graphenvisualisierung+

**Hintergrund:** International Symposium on Graph Drawing (GD)  
umfasst einen Graphenvisualisierungswettbewerb  
(Graph Drawing Challenge)

**Aufgabe:** entwickle für im Vorfeld  
angekündigte Problemstellung  
eine Visualisierungssoftware



**Sieger 2014:** KIT-Team aus dieser Vorlesung im WS 2013/14!



# Graphenvisualisierung+

**Hintergrund:** International Symposium on Graph Drawing (GD)  
umfasst einen Graphenvisualisierungswettbewerb  
(Graph Drawing Challenge)

**Aufgabe:** entwickle für im Vorfeld  
angekündigte Problemstellung  
eine Visualisierungssoftware



**Sieger 2014:** KIT-Team aus dieser Vorlesung im WS 2013/14!



**2013/14:**

- Programmierprojekt als Teil der Vorlesung
- viel Spaß, aber recht zeitaufwändig

**diesmal:**

- eigene Veranstaltung im SS 2015
- aufbauend auf diese Vorlesung
- kombiniertes Modul mit insgesamt 10 LP

# Graphenvisualisierung+

**Hintergrund:** International Symposium on Graph Drawing (GD)  
umfasst einen Graphenvisualisierungswettbewerb  
(Graph Drawing Challenge)

**Aufgabe:** entwickle für im Vorfeld  
angekündigte Problemstellung  
eine Visualisierungssoftware



**Sieger 2014:** KIT-Team aus dieser Vorlesung im WS 2013/14!



Stimmungsbild?

**2013/14:**

- Programmierprojekt als Teil der Vorlesung
- viel Spaß, aber recht zeitaufwändig

**diesmal:**

- eigene Veranstaltung im SS 2015
- aufbauend auf diese Vorlesung
- kombiniertes Modul mit insgesamt 10 LP

## Medien:

- Folien & Tafel
- Übungsblätter zur Vertiefung und Anwendung
- (vorläufiges) Skript/Lehrbücher
- Original-Literatur



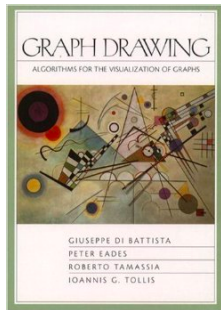
## Medien:

- Folien & Tafel
- Übungsblätter zur Vertiefung und Anwendung
- (vorläufiges) Skript/Lehrbücher
- Original-Literatur

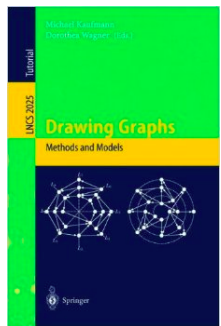
## Inhalte:

- Reduzierung der Visualisierung auf **algorithmischen Kern**
- **nicht:** grafisches Rendering
- Modellierung, Algorithmen, formale Analyse
  - Divide & Conquer / Rekursion
  - kombinatorische Optimierung (Flüsse, ILP, ...)
  - inkrementelle Algorithmen
  - Algorithmen für spezielle Graphenklassen
  - Zeichnen durch physikalische Analogien

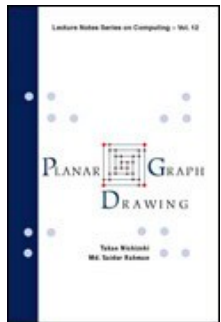
# Literatur (in Bibliothek verfügbar)



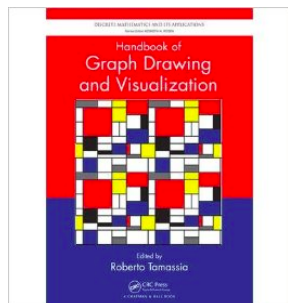
G. di Battista, P. Eades, R. Tamassia, I. Tollis:  
Graph Drawing  
Prentice Hall, 1998



M. Kaufmann, D. Wagner:  
Drawing Graphs: Methods and Models  
Springer, 2001



T. Nishizeki, Md. S. Rahman:  
Planar Graph Drawing  
World Scientific, 2004



R. Tamassia:  
Handbook of Graph Drawing and Visualization  
CRC Press, 2013

<http://cs.brown.edu/~rt/gdhandbook/>

## Basiswissen Graphentheorie:

- Graph, Knoten, Kanten
- Knotengrad, Nachbarschaft, adjazent, inzident
- Zusammenhang, Baum, Kreis, Pfad
- ...

## Basiswissen Graphentheorie:

- Graph, Knoten, Kanten
- Knotengrad, Nachbarschaft, adjazent, inzident
- Zusammenhang, Baum, Kreis, Pfad
- ...

## Basiswissen Algorithmik:

- asymptotische Laufzeit,  $O$ -Kalkül
- Komplexität, NP-Vollständigkeit
- Netzwerkflüsse
- Lineare Programmierung
- Rekursion
- Divide & Conquer
- Approximation
- ...

## Basiswissen Graphentheorie:

- Graph, Knoten, Kanten
- Knotengrad, Nachbarschaft, adjazent, inzident
- Zusammenhang, Baum, Kreis, Pfad
- ...

## Basiswissen Algorithmik:

- asymptotische Laufzeit,  $O$ -Kalkül
- Komplexität, NP-Vollständigkeit
- Netzwerkflüsse
- Lineare Programmierung
- Rekursion
- Divide & Conquer
- Approximation
- ...

**generell gilt:**  
bei Unklarheiten nachfragen

# Einführung Graphenvisualisierung

Was ist ein Graph?

## Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$



Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$

Darstellungsformen?

## Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$

## Darstellungsformen?

**Mengenschreibweise**

$$\begin{aligned} V &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \\ E &= \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \\ &\quad \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \\ &\quad \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \\ &\quad \{v_9, v_{10}\}\} \end{aligned}$$

## Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$

## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

```
v1 : v2, v8
v2 : v1, v3
v3 : v2, v5, v9, v10
v4 : v5, v6, v9
v5 : v3, v4, v8
v6 : v4, v8, v9
v7 : v8, v9
v8 : v1, v5, v6, v7, v9, v10
v9 : v3, v4, v6, v7, v8, v10
v10 : v3, v8, v9
```

## Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$

## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

**Adjazenzmatrix**

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

## Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$

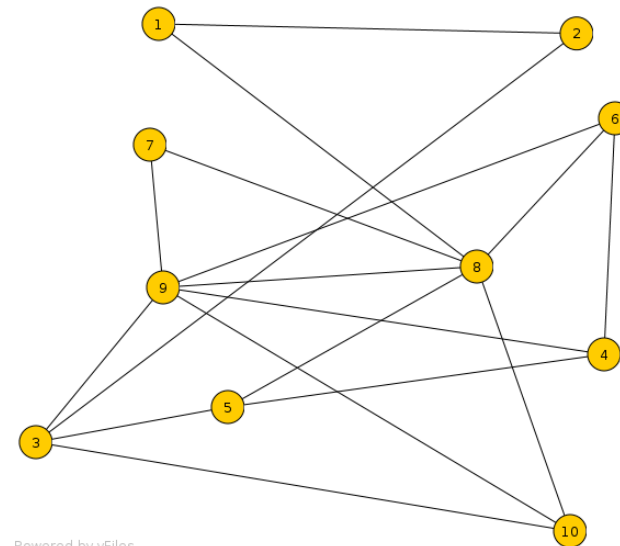
## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

Adjazenzmatrix

**Zeichnung**



Powered by yFiles

## Was ist ein Graph?

Tupel  $G = (V, E)$

Knotenmenge  $V = \{v_1, \dots, v_n\}$

Kantenmenge  $E = \{e_1, \dots, e_m\}$

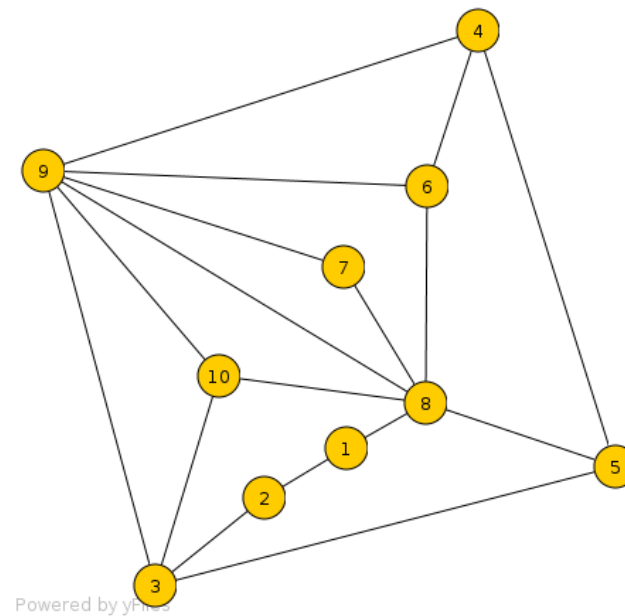
## Darstellungsformen?

Mengenschreibweise

Adjazenzliste

Adjazenzmatrix

**Zeichnung**

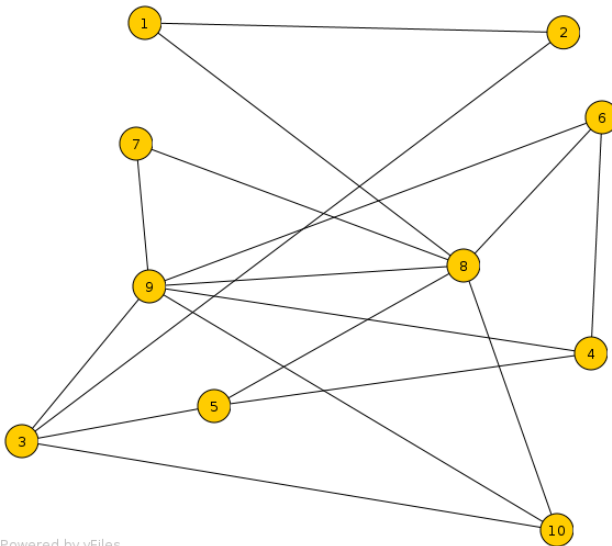


# Graphen und ihre Darstellung

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$$

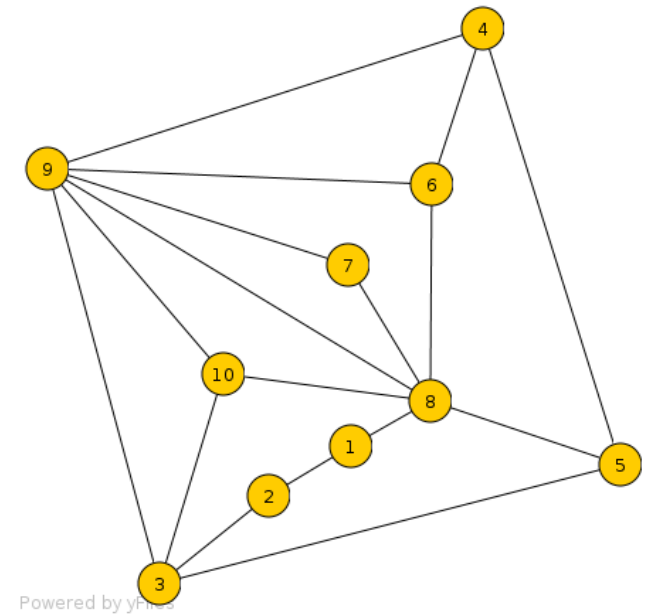
$$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\}$$

$v_1$  :  $v_2, v_8$   
 $v_2$  :  $v_1, v_3$   
 $v_3$  :  $v_2, v_5, v_9, v_{10}$   
 $v_4$  :  $v_5, v_6, v_9$   
 $v_5$  :  $v_3, v_4, v_8$   
 $v_6$  :  $v_4, v_8, v_9$   
 $v_7$  :  $v_8, v_9$   
 $v_8$  :  $v_1, v_5, v_6, v_7, v_9, v_{10}$   
 $v_9$  :  $v_3, v_4, v_6, v_7, v_8, v_{10}$   
 $v_{10}$  :  $v_3, v_8, v_9$



Powered by yFiles

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$



Powered by yFiles

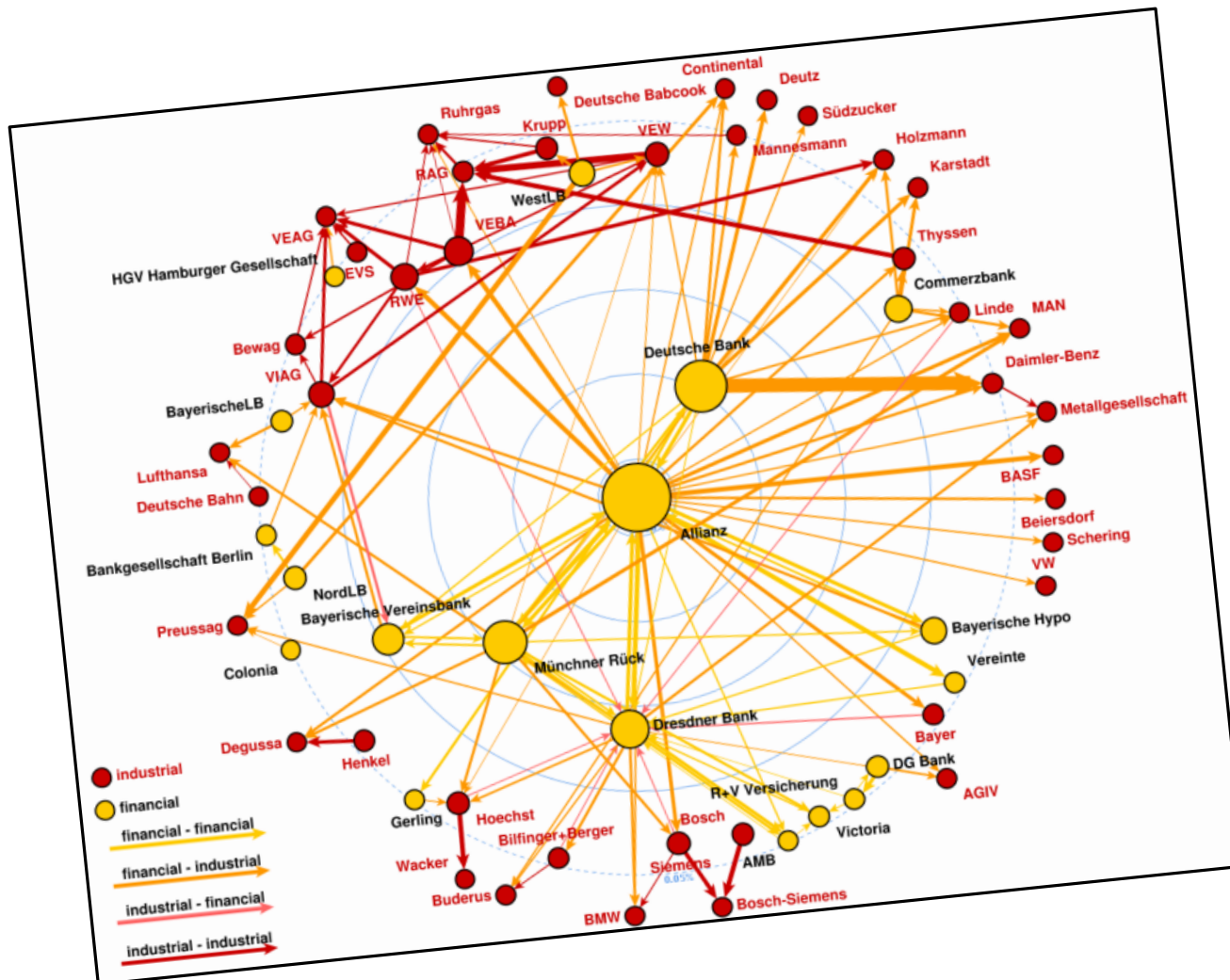
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



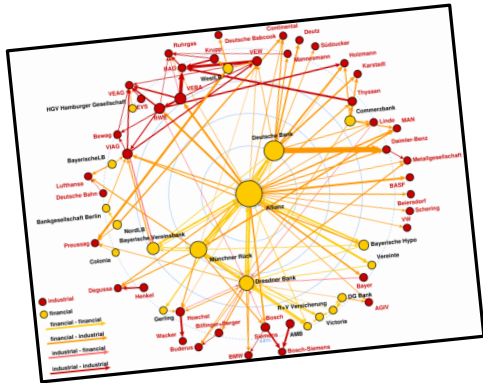
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



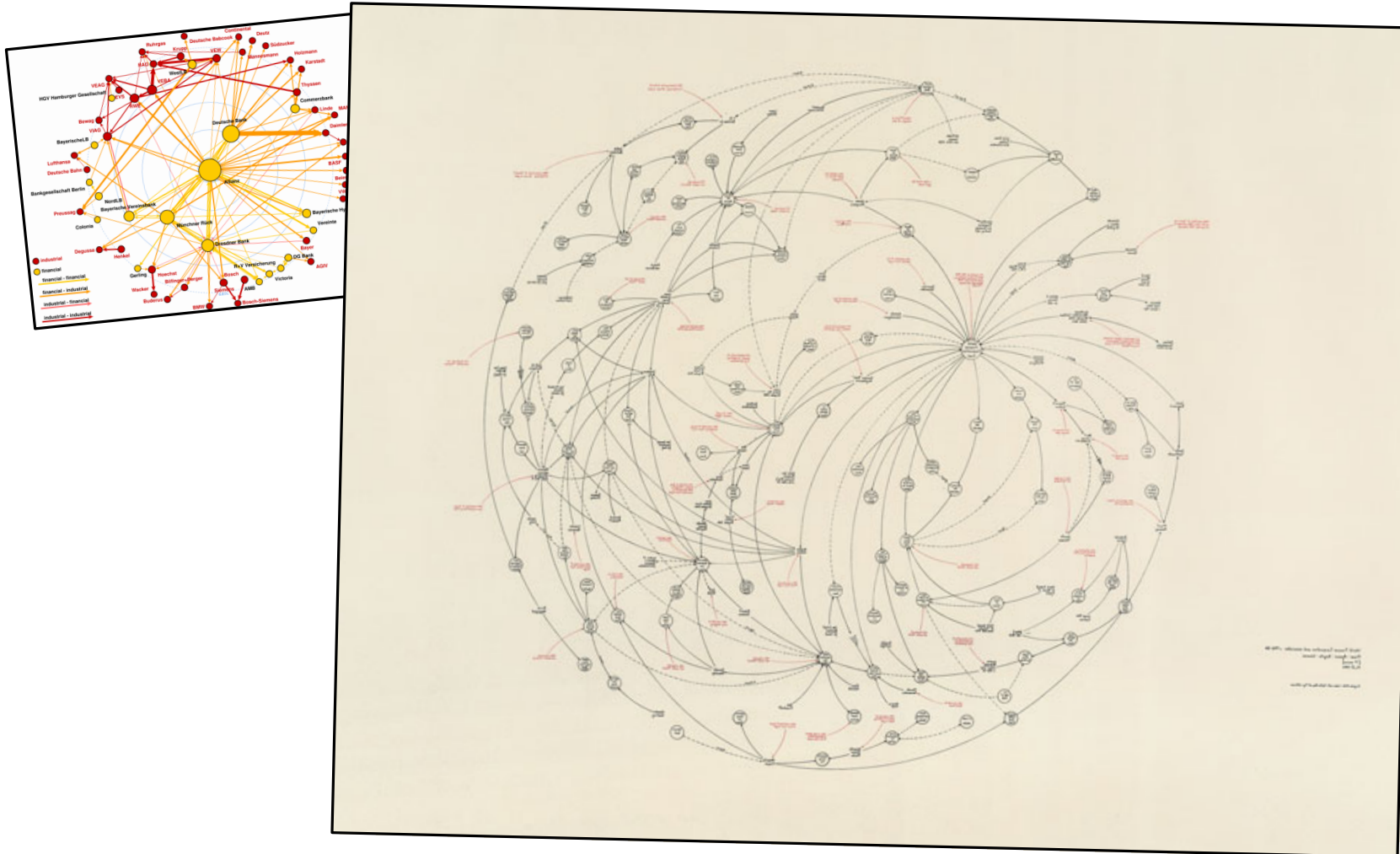
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



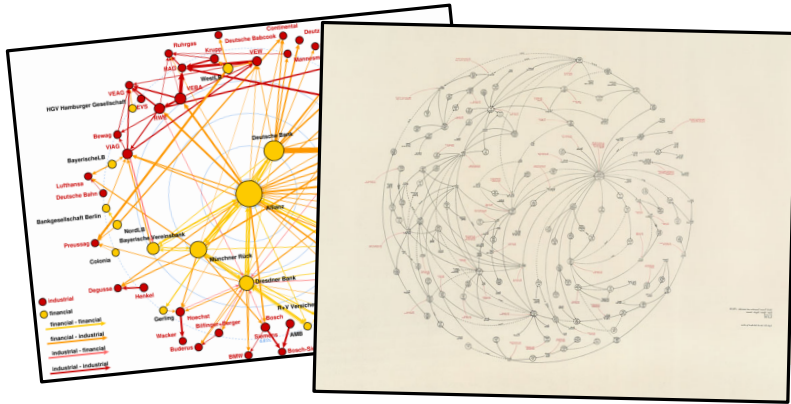
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



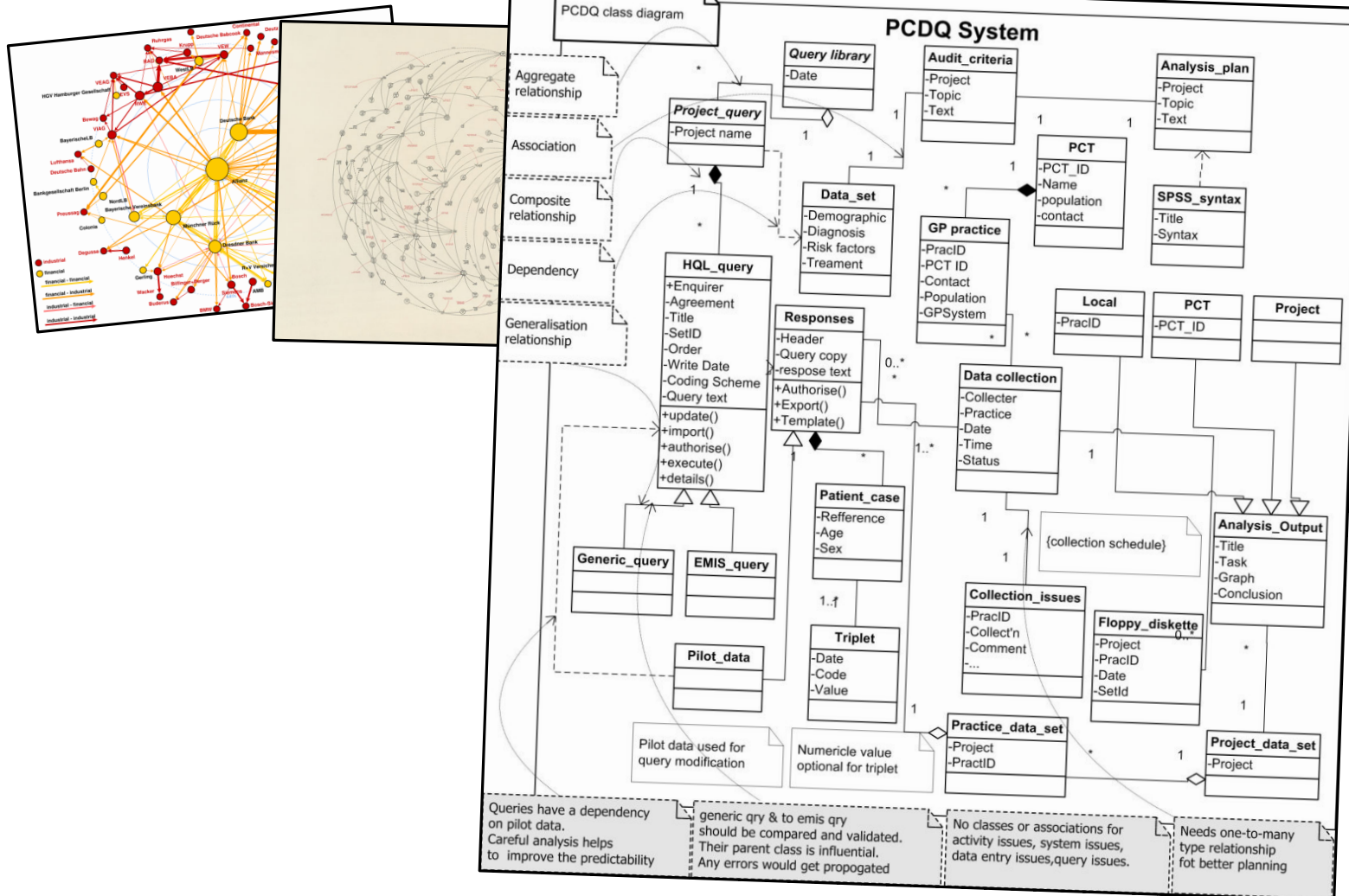
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



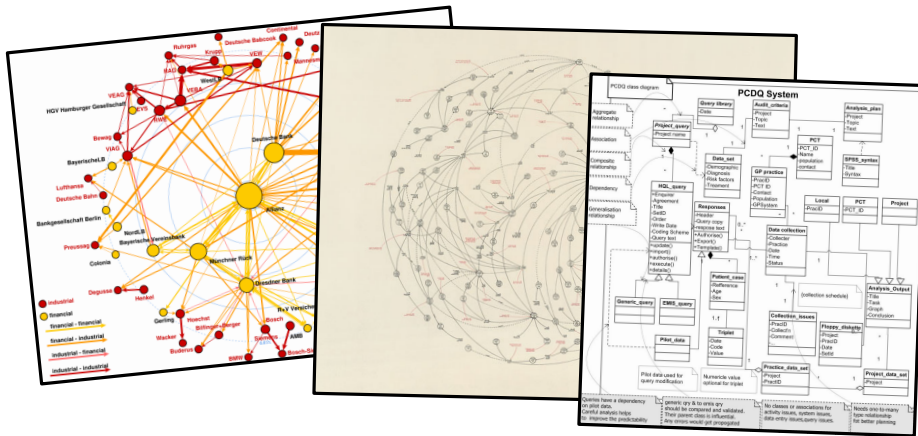
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



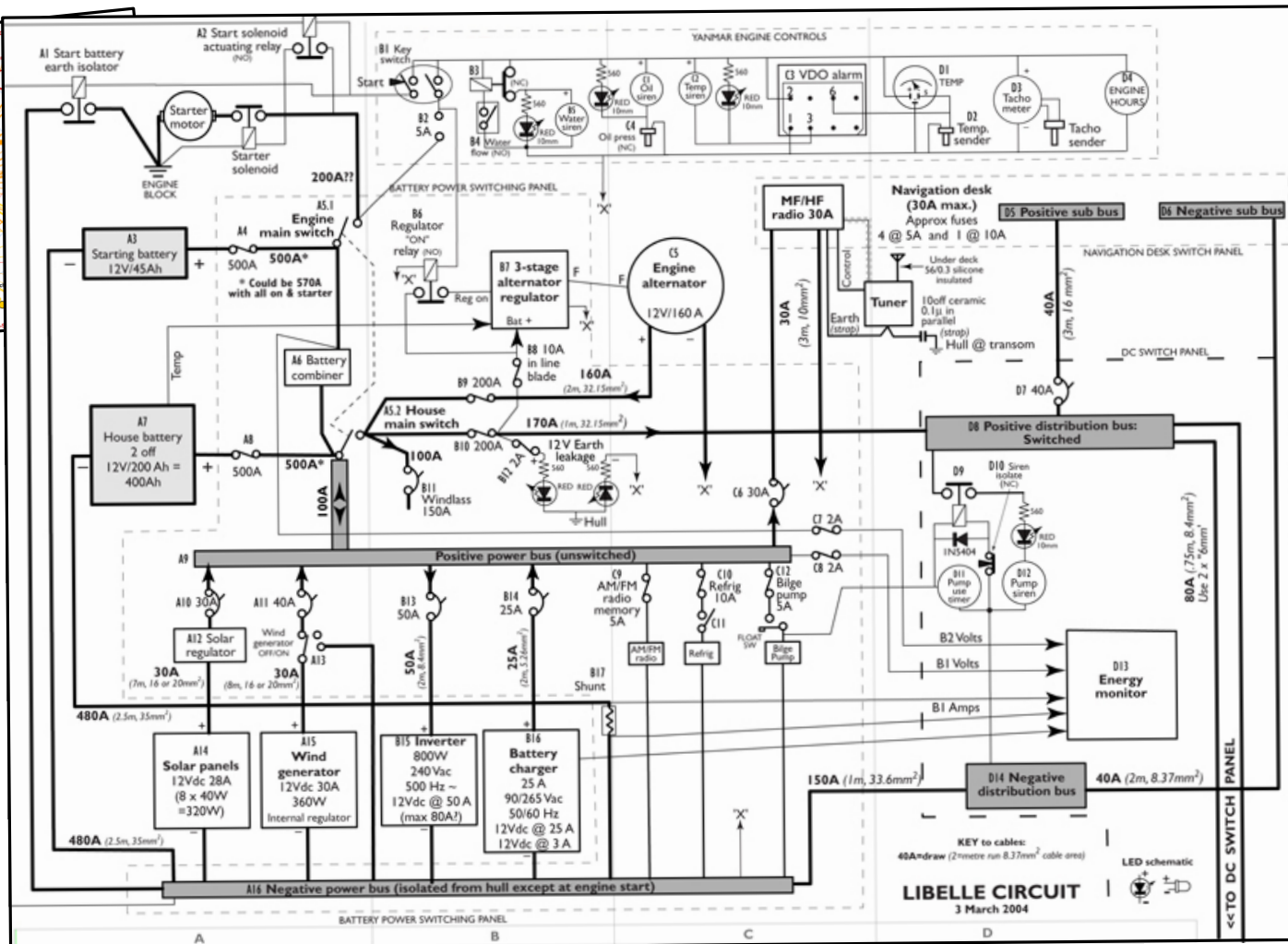
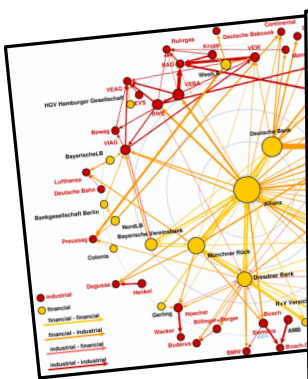
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



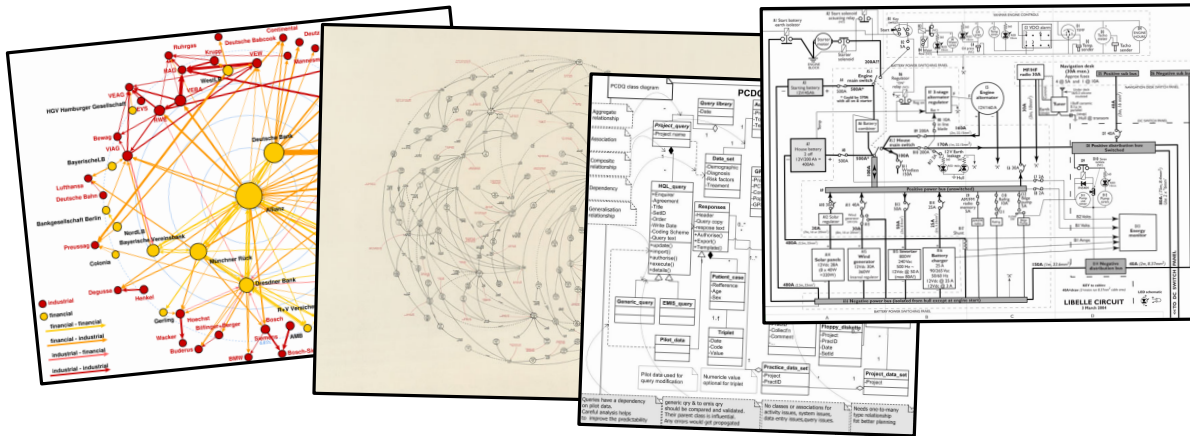
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



# Wozu Graphen zeichnen?

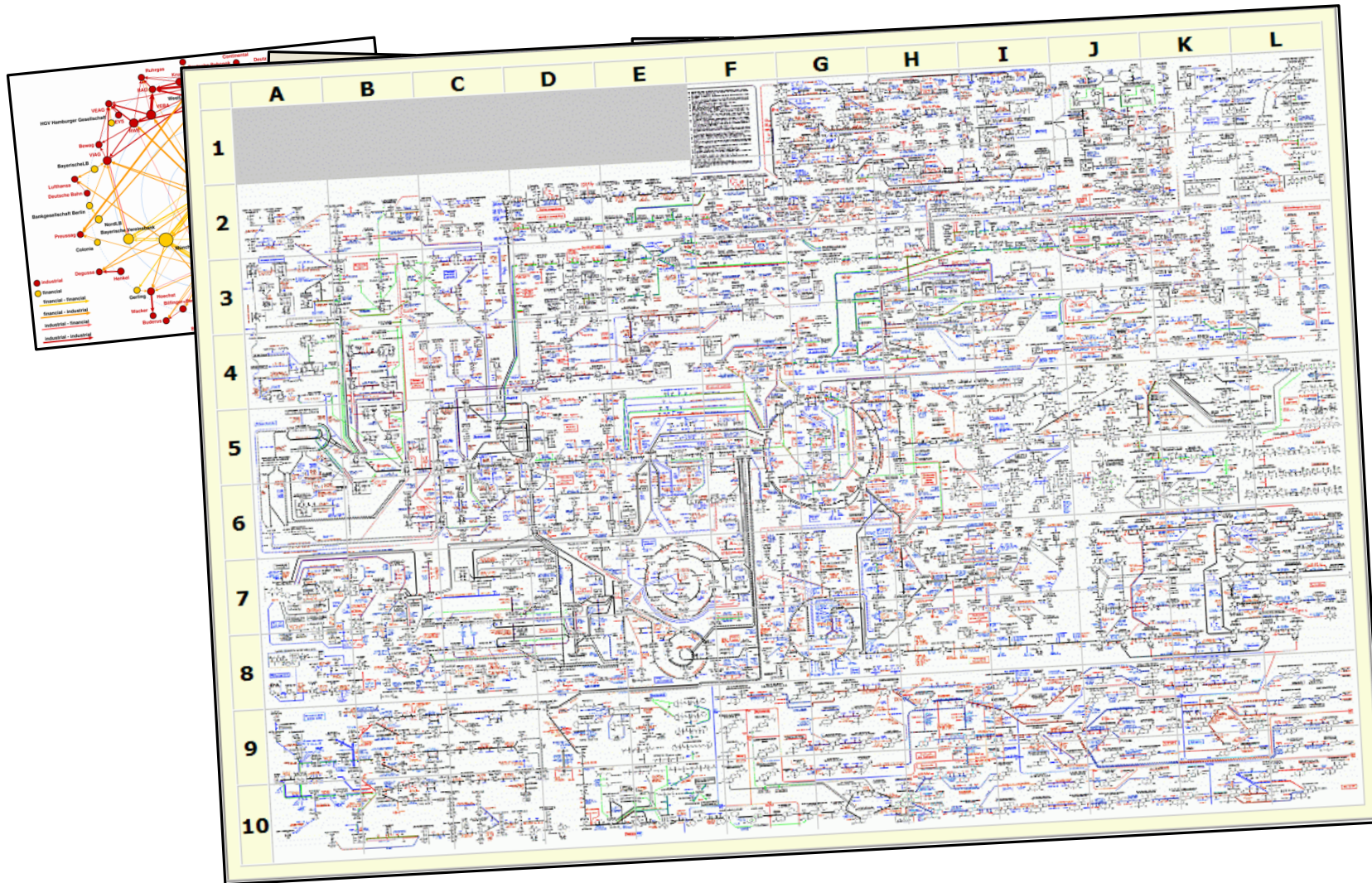
Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)





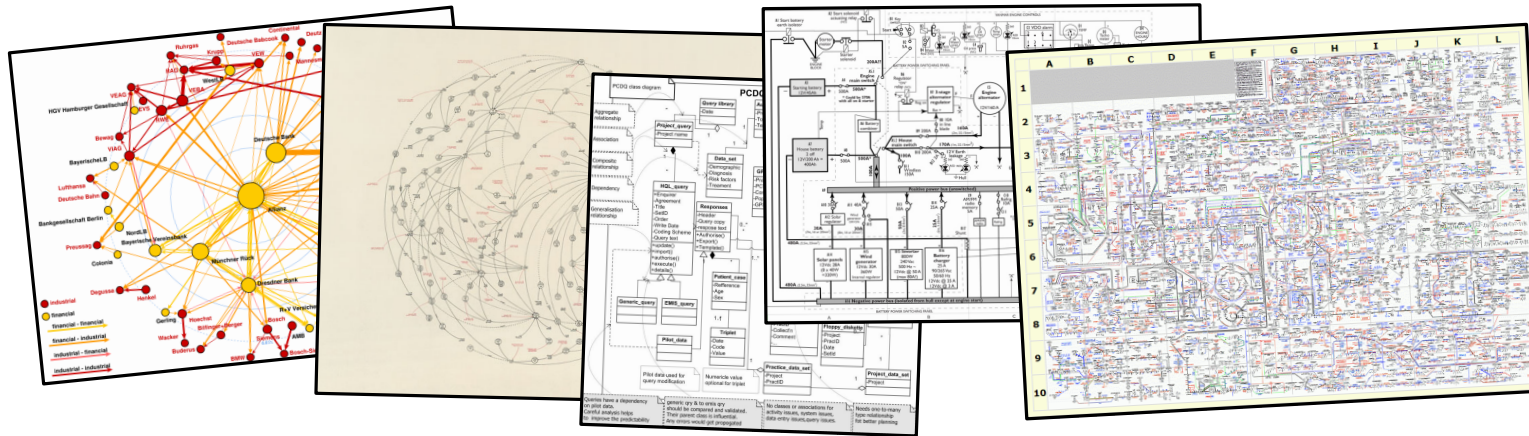
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



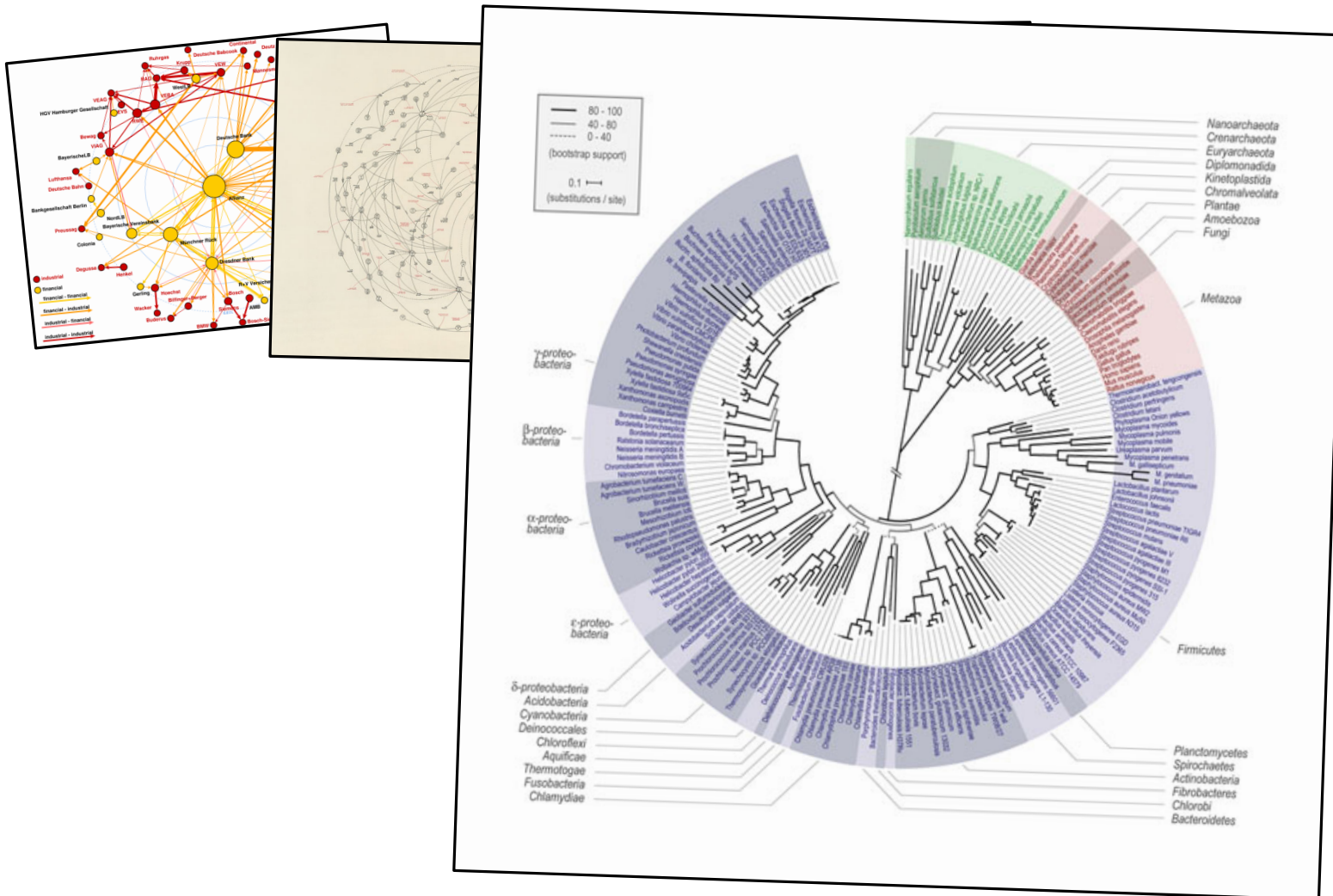
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



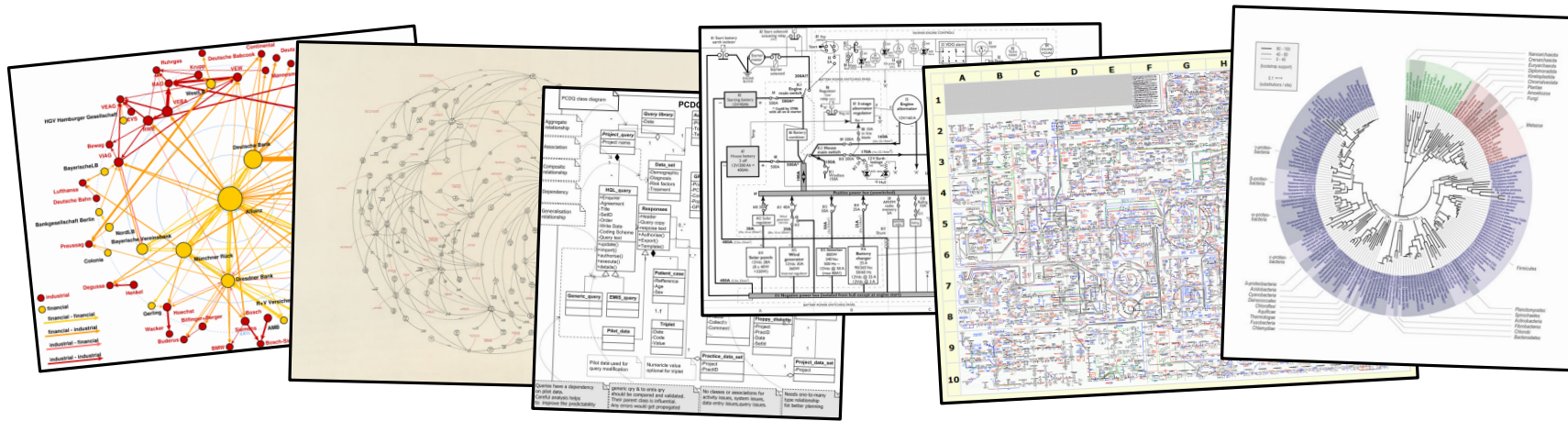
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



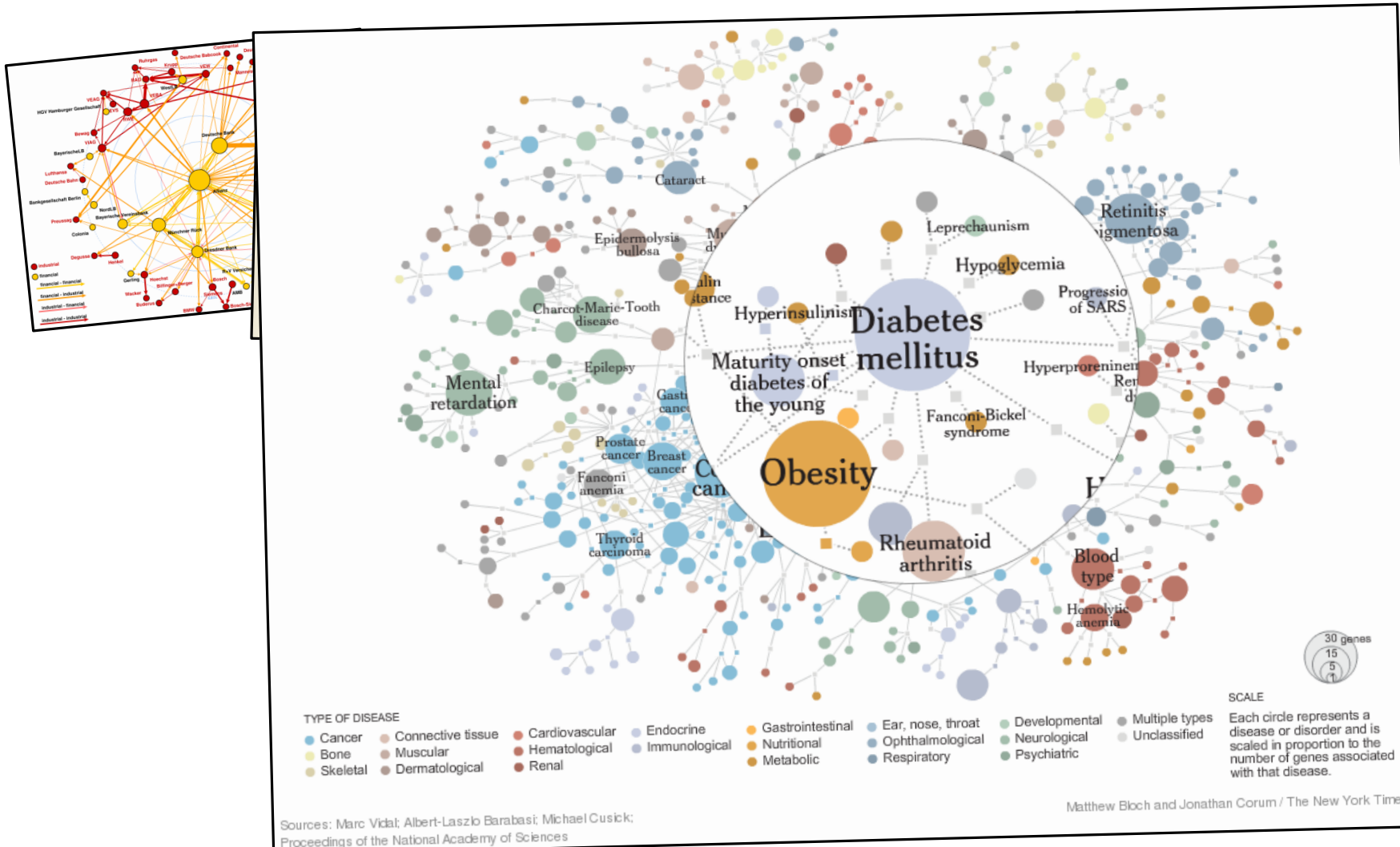
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



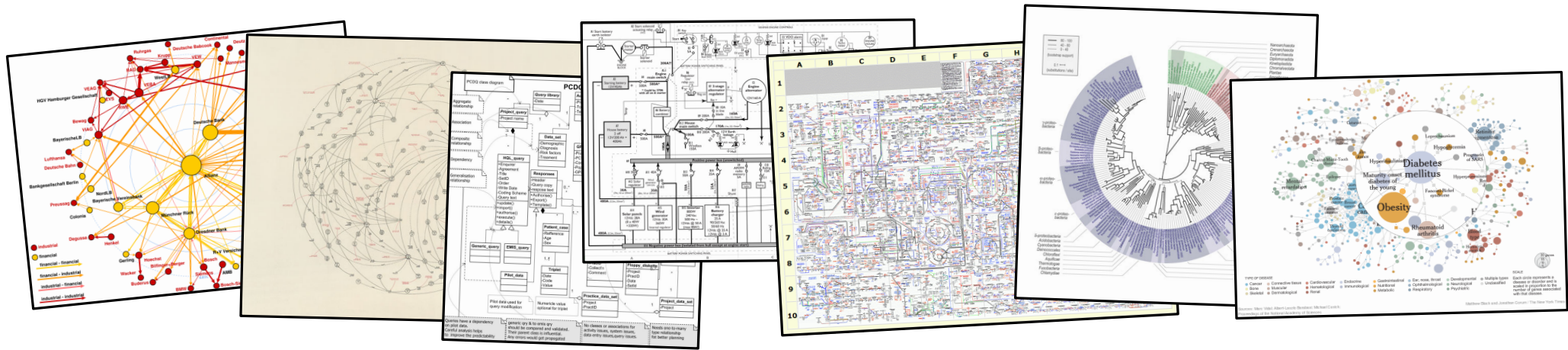
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



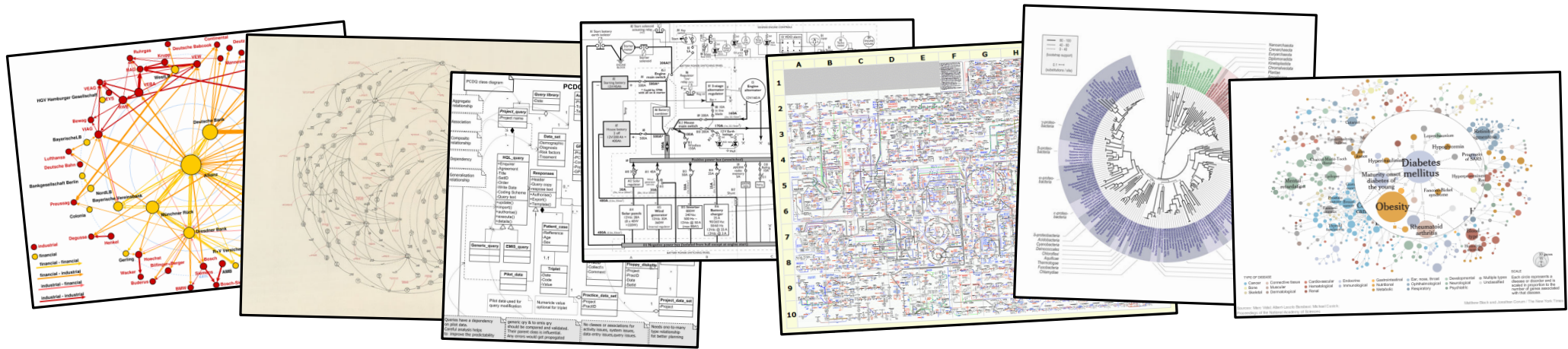
# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



# Wozu Graphen zeichnen?

Graphen sind mathematische Modelle realer physischer und abstrakter Netzwerke (soziale Netze, metabolische Netze, VLSI-Layout, UML Diagramme, Infrastrukturnetze, ...)



- **Menschen denken visuell** – ohne gute Visualisierung sind komplexe Graphen für uns unverständlich
- Visualisierungen helfen bei der **Kommunikation** und der **Exploration** von Graphen/Netzwerken
- wir brauchen **Algorithmen** zum Zeichnen von Graphen um Netzwerke dem Menschen zugänglich zu machen

# Wie Graphen zeichnen?

## Leitfragen

- was macht eine Zeichnung gut?
- was sollte man vermeiden?
- lassen sich Kriterien quantifizieren?



# Wie Graphen zeichnen?

## Leitfragen

- was macht eine Zeichnung gut?
- was sollte man vermeiden?
- lassen sich Kriterien quantifizieren?

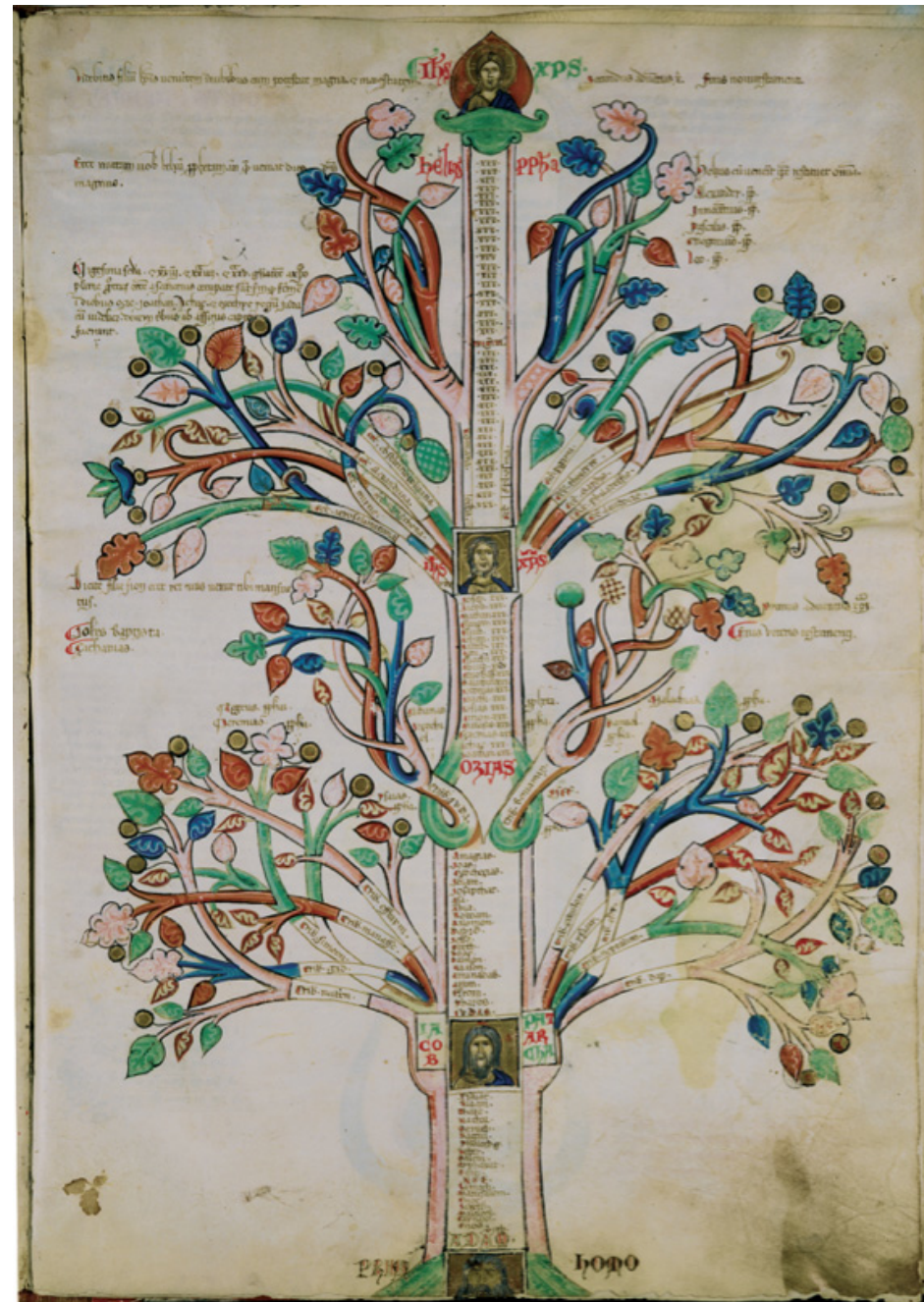
## Aufwärmübung WIR MALEN EIN PAAR GRAPHEN

- Zweier-Teams
- „Eingabe“ als Adjazenzmatrix/-liste
- 10–15 Minuten Zeit
- Vorstellen der Resultate und Sammeln der Kriterien

# Beispiele

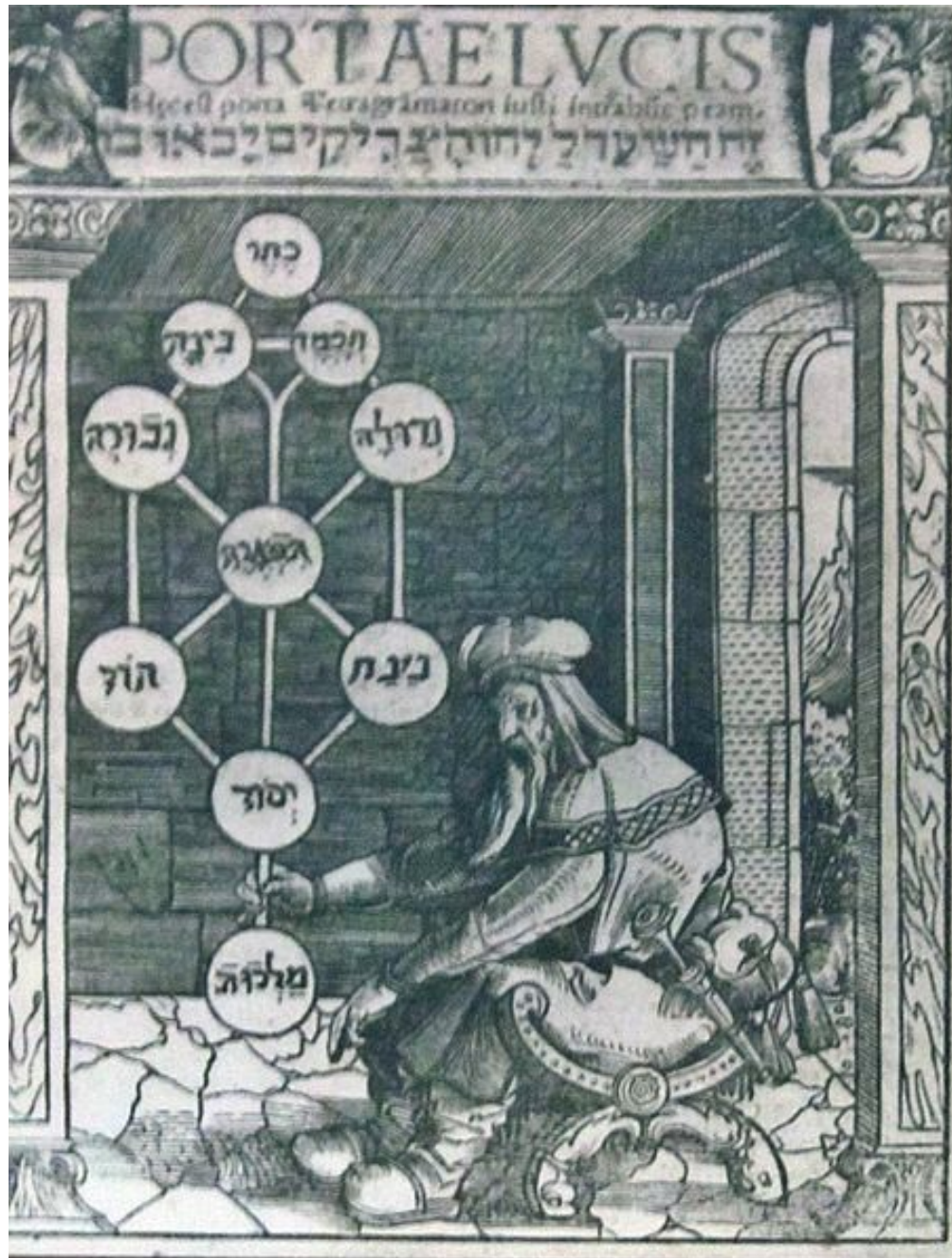
eine kleine Diaschau

# Biblische Charaktere und Ereignisse (1202)



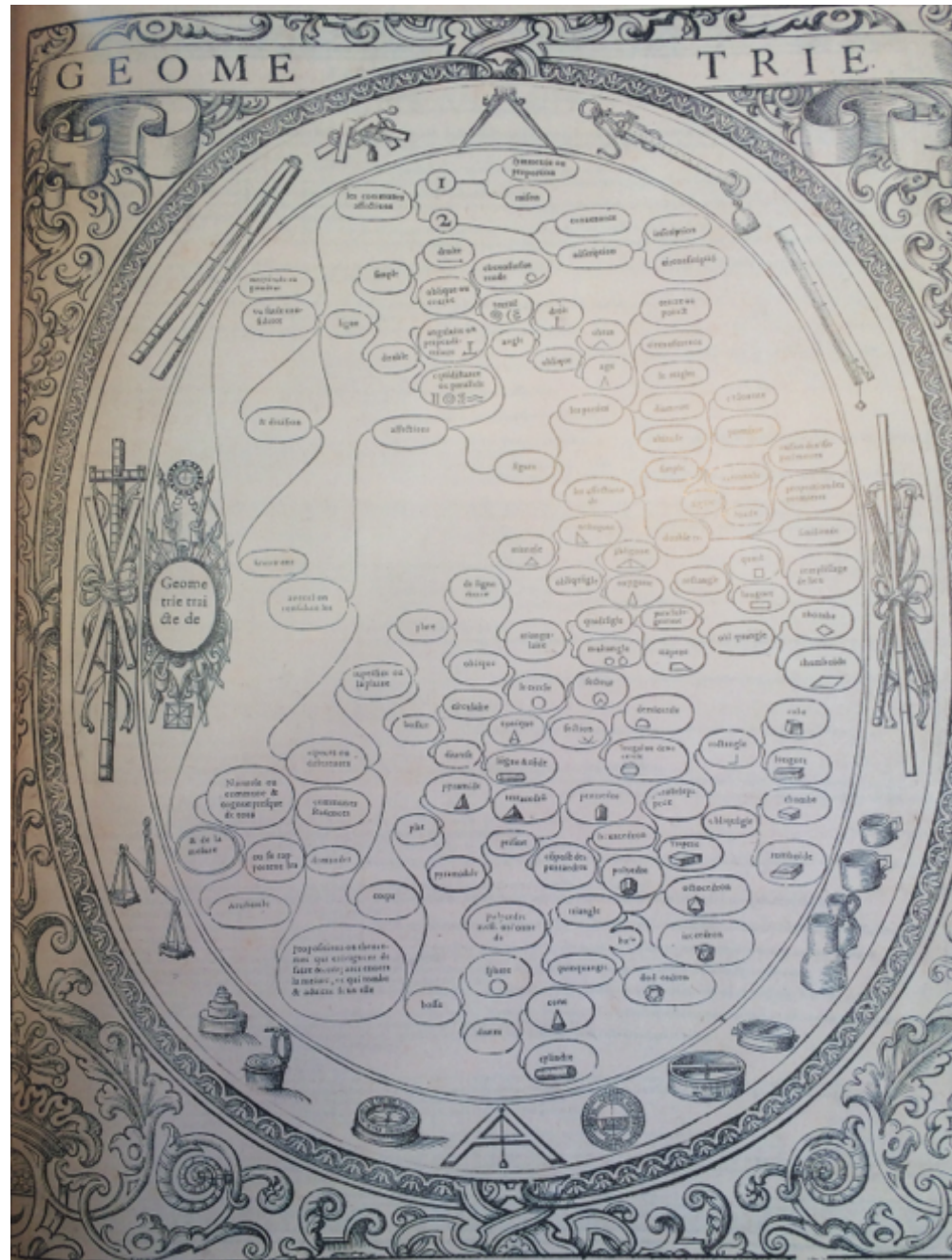
Quelle: Joachim de Fiore

# „Lebensbaum“ (1516)



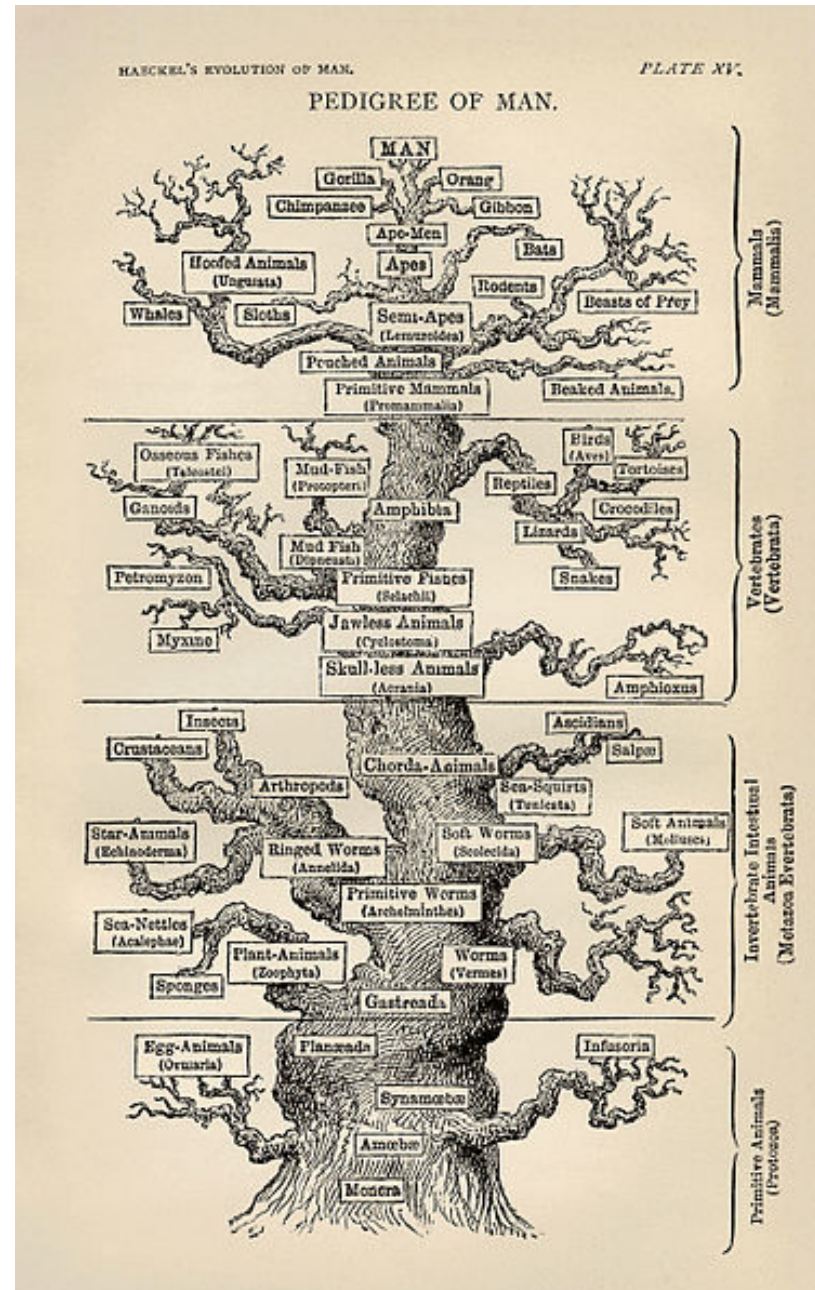
Quelle: Paul Riccius, Portae Lucis

# Geometrische Konzepte (1587)



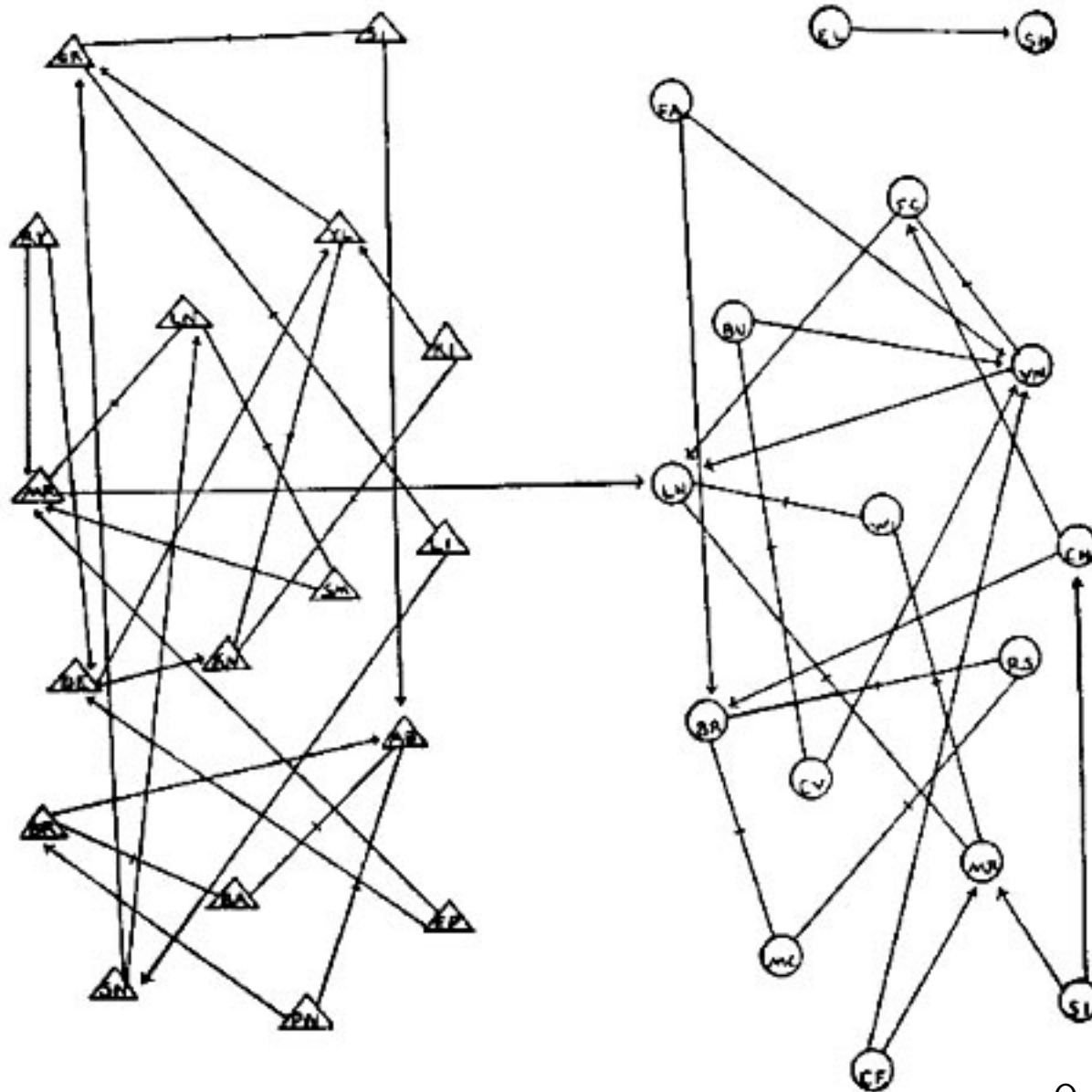
Quelle: Christophe de Savigny

# Abstammungsbaum (1879)



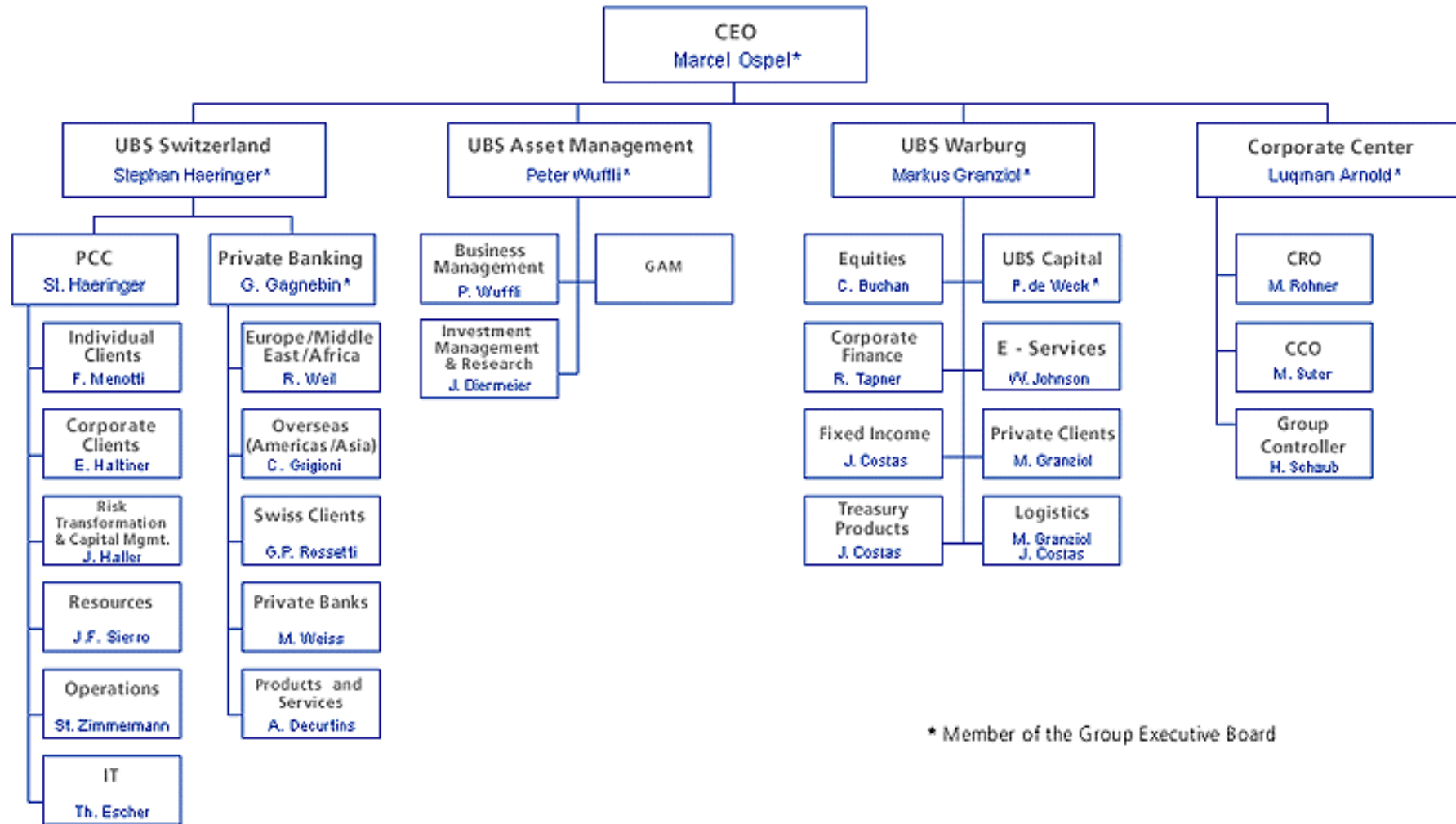
Quelle: Ernst Haeckel

# Soziogramm (1933)



Quelle: Moreno, 1933

# Soziale Netze – Organigramm UBS



\* Member of the Group Executive Board

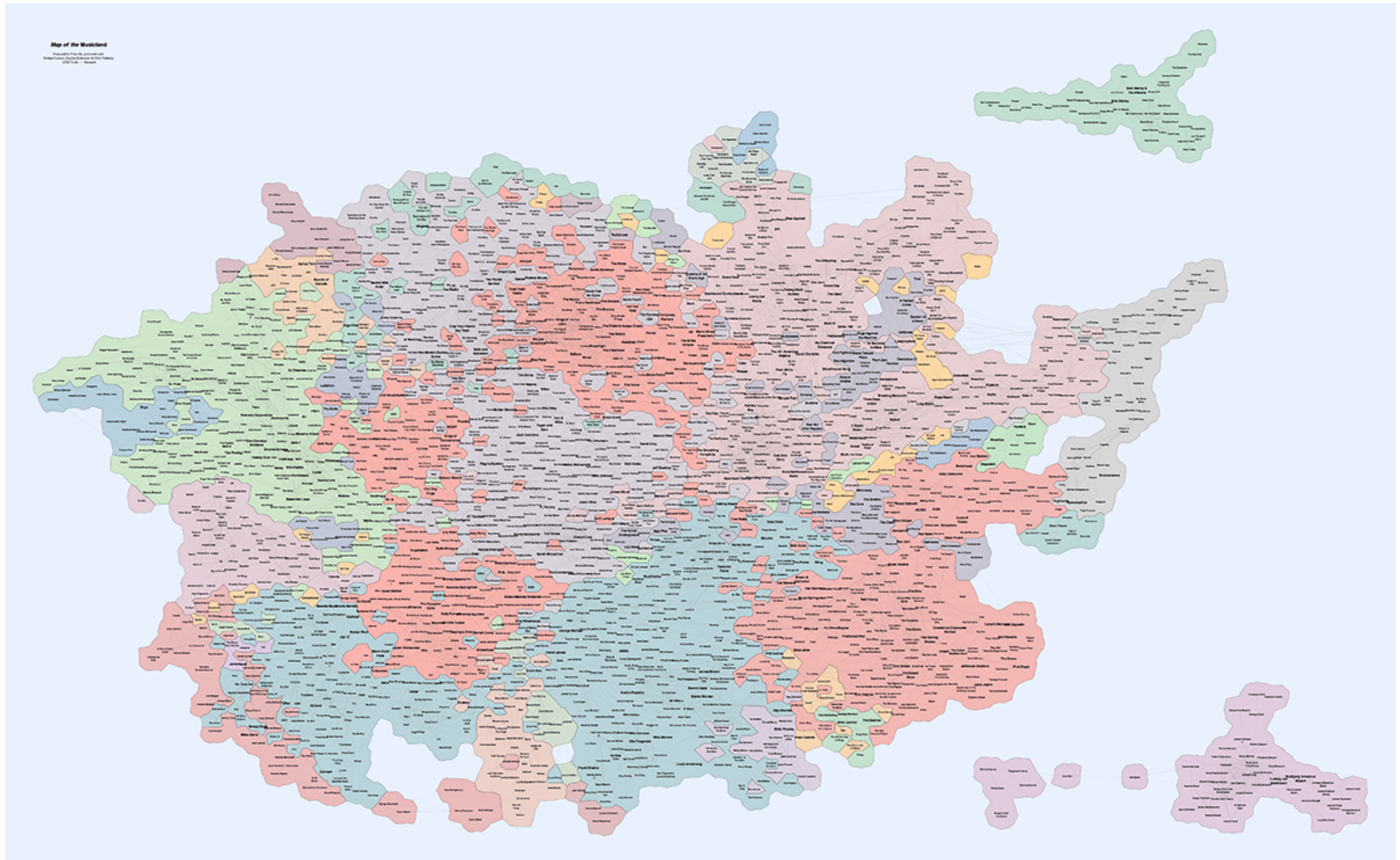


# CPAN Entwickler-Graph



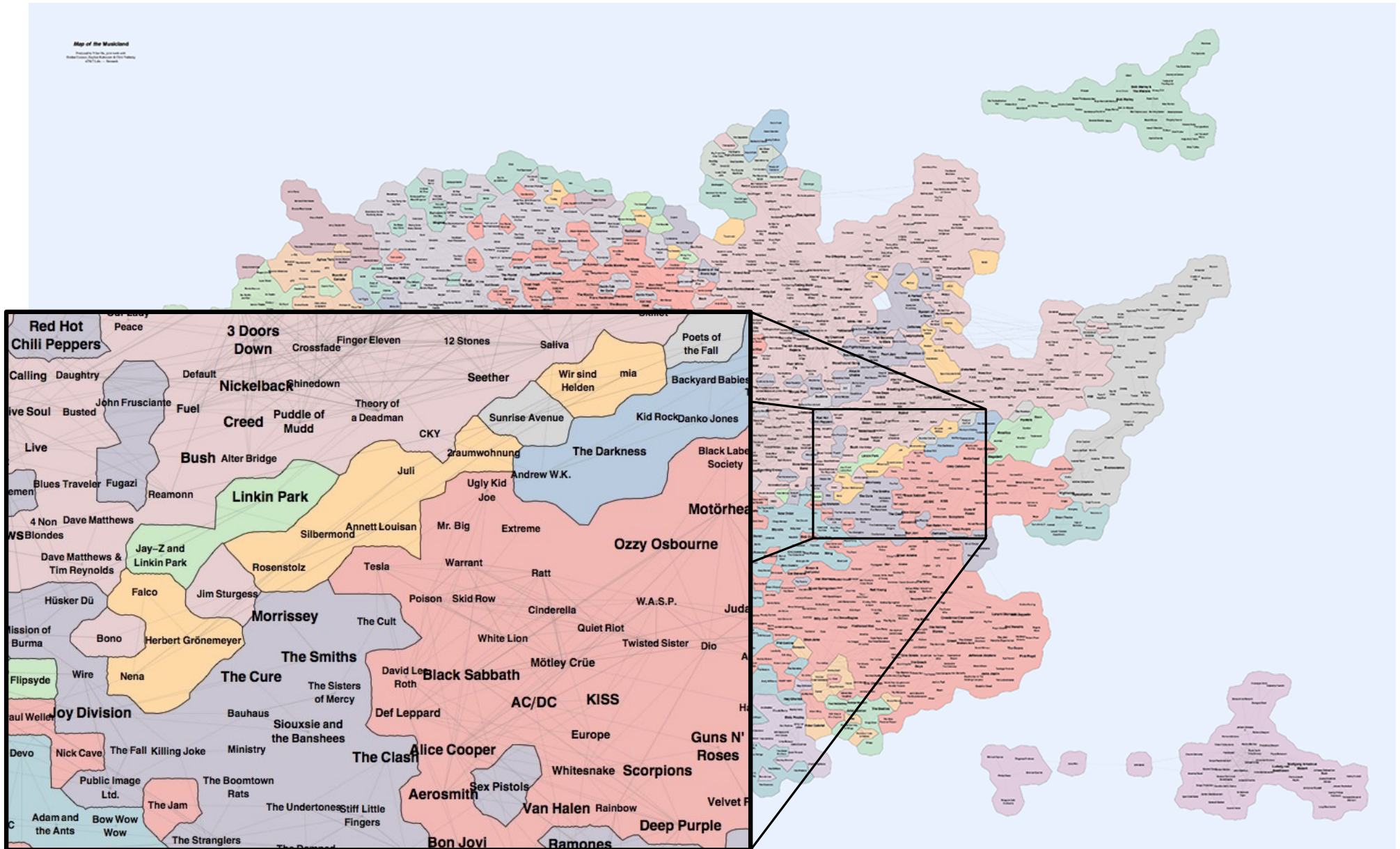
Quelle: [cpan-explorer.org](http://cpan-explorer.org)

# last.fm Musikergraph als politische Karte



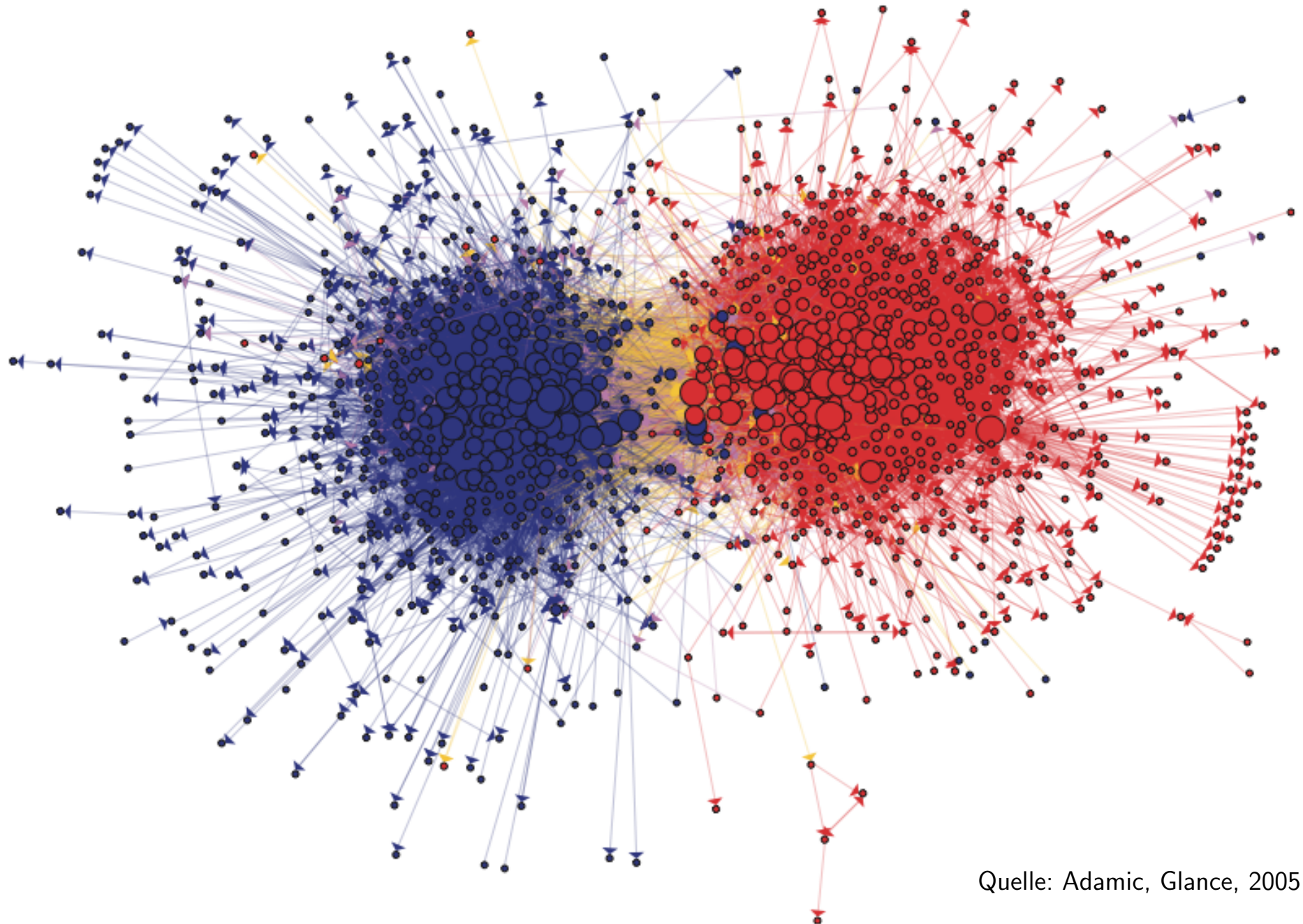
(Gansner, Hu, Kobourov: GMap, 2009)

# last.fm Musikergraph als politische Karte



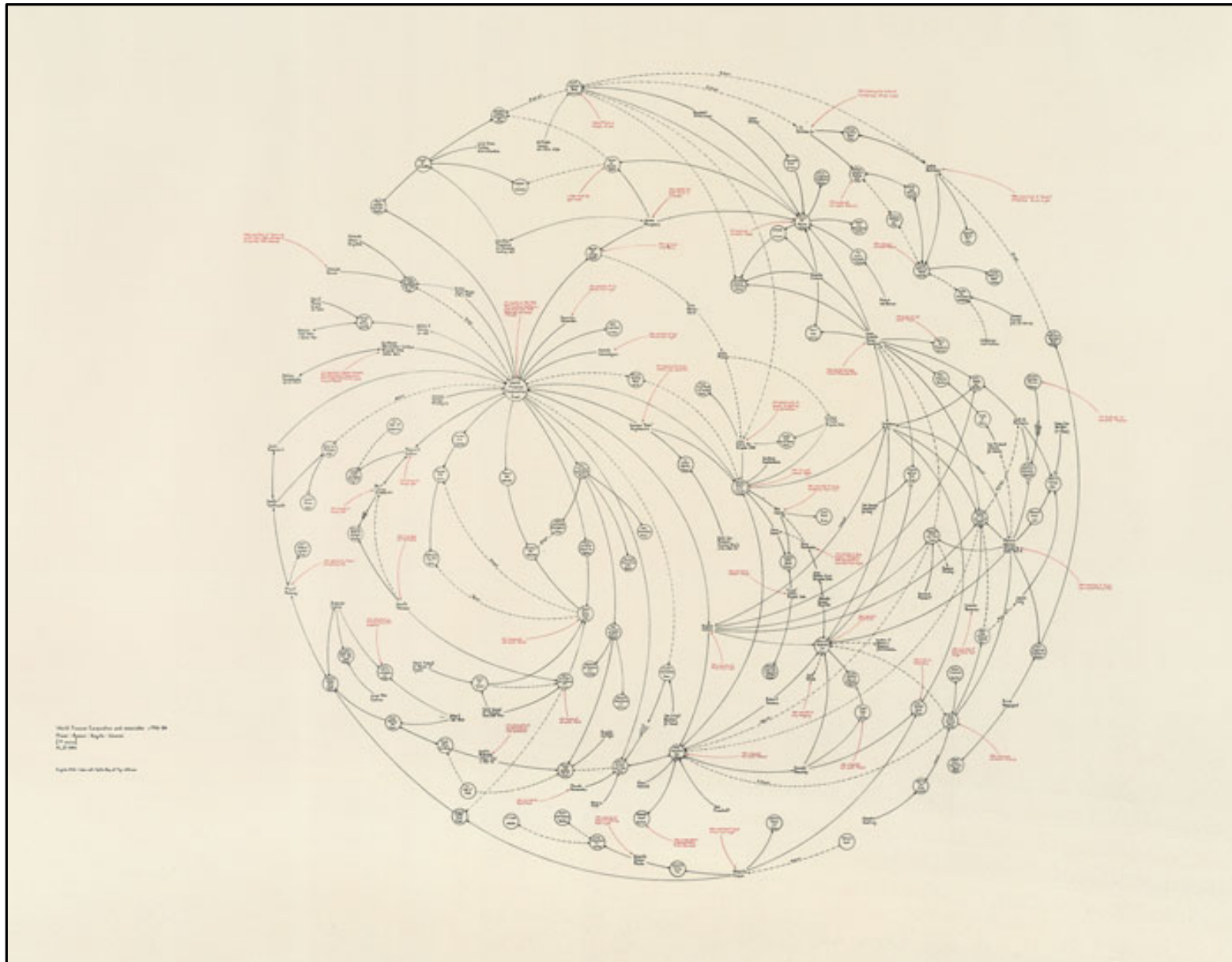
(Gansner, Hu, Kobourov: GMap, 2009)

# Blogosphere 2004 Elections USA



Quelle: Adamic, Glance, 2005

# Soziale Netze – Welt-Finanzsystem



World Finance Corporation  
© Mark Lombardi

# Soziale Netze – Staatsfonds

## FOLLOW THE MONEY

### The New Global Wealth Machine

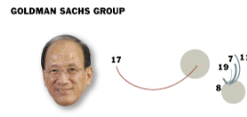
Sovereign wealth funds have emerged in recent months as the world's power brokers. They have used their tremendous wealth to make big cross-border investments and prop up some of Wall Street's best-known firms. The increased activity comes as other kinds of acquirers have been sidelined by the credit crisis. These funds are state-sponsored investment vehicles and have combined assets of \$2 trillion. With that much dry powder, sovereign funds dwarf the formerly booming private equity industry — and in some cases, compete directly with it. The Government of Singapore Investment Corporation has been the most active among the world's sovereign funds, making its deputy chairman, Tony Tan, a major center of gravity. Wall Street veterans always follow the money, so many of the big-name advisers in New York and London have found themselves traveling the globe playing international matchmaker to these funds. But sovereign funds have also learned the downside of deal-making: some of their blockbuster transactions have been big money losers so far. The question is where all that money will go next. **ANDREW ROSS SORKIN**

#### The Advisers

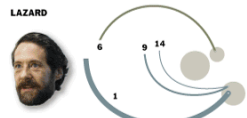
Selected financial advisers who worked on more than one of the top 20 deals.



**Michael Klein, Chairman, institutional clients group**  
One of the firm's highest-profile investment bankers, he advised Cayle in its stake sale to Mahabadi, as well as Citigroup in both of its deals with sovereign wealth funds.



**Richard Ong, Former managing director**  
Mr. Ong left Goldman early this year after the Chinese government refused to allow the firm to promote him to run its Beijing office. Mr. Ong's brother, Charles, was the chief investment officer of Temasek Holdings until 2006.



**Gary Parr, Deputy chairman**  
In addition to becoming the key adviser on many of the biggest sovereign wealth deals, Mr. Parr helped advise Bear Stearns on its distressed sale to JP Morgan Chase.

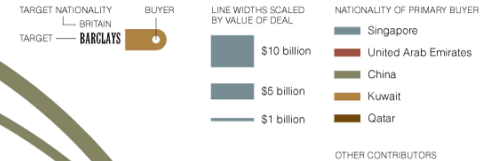


**Kate Richdale, Managing director**  
The head of Morgan Stanley's Asian general industries group, based in Hong Kong. She previously held a senior position in the investment bank's Southeast Asia group.

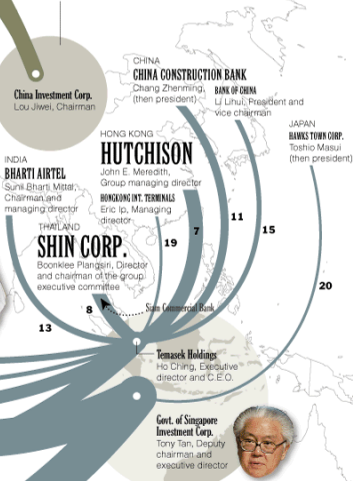
#### The Targets



### The 20 Biggest Cross-Border Sovereign Wealth Fund Deals Since 2005

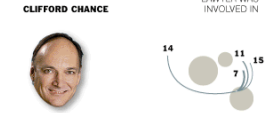


#### The Buyers

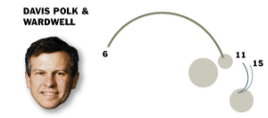


#### The Lawyers

Selected lawyers who worked on more than one of the top 20 deals.



**James Baird, Partner and global head of private equity**  
Mr. Baird's firm, based in London, was one of the early firms to make a bet on Asia by staffing up there before some of the traditional white-shoe Wall Street firms ventured there.



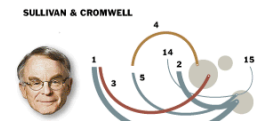
**Randall D. Guynn, Partner**  
As head of the firm's financial institutions group, he has advised on many international deals in Europe and Asia. He also worked on the team that advised Morgan Stanley in its \$5.5 billion stake sale to China's sovereign wealth fund.



**Richard Good, Partner**  
Based in Singapore, Mr. Good is the firm's man-on-the-ground in Asia. He has worked for Linklaters in Asia since 2000.



**Stephen M. Besen, Partner**  
A longtime hand in the Middle East, Mr. Besen's deep relationships have helped his firm carve out one of the strongest niches in the region.

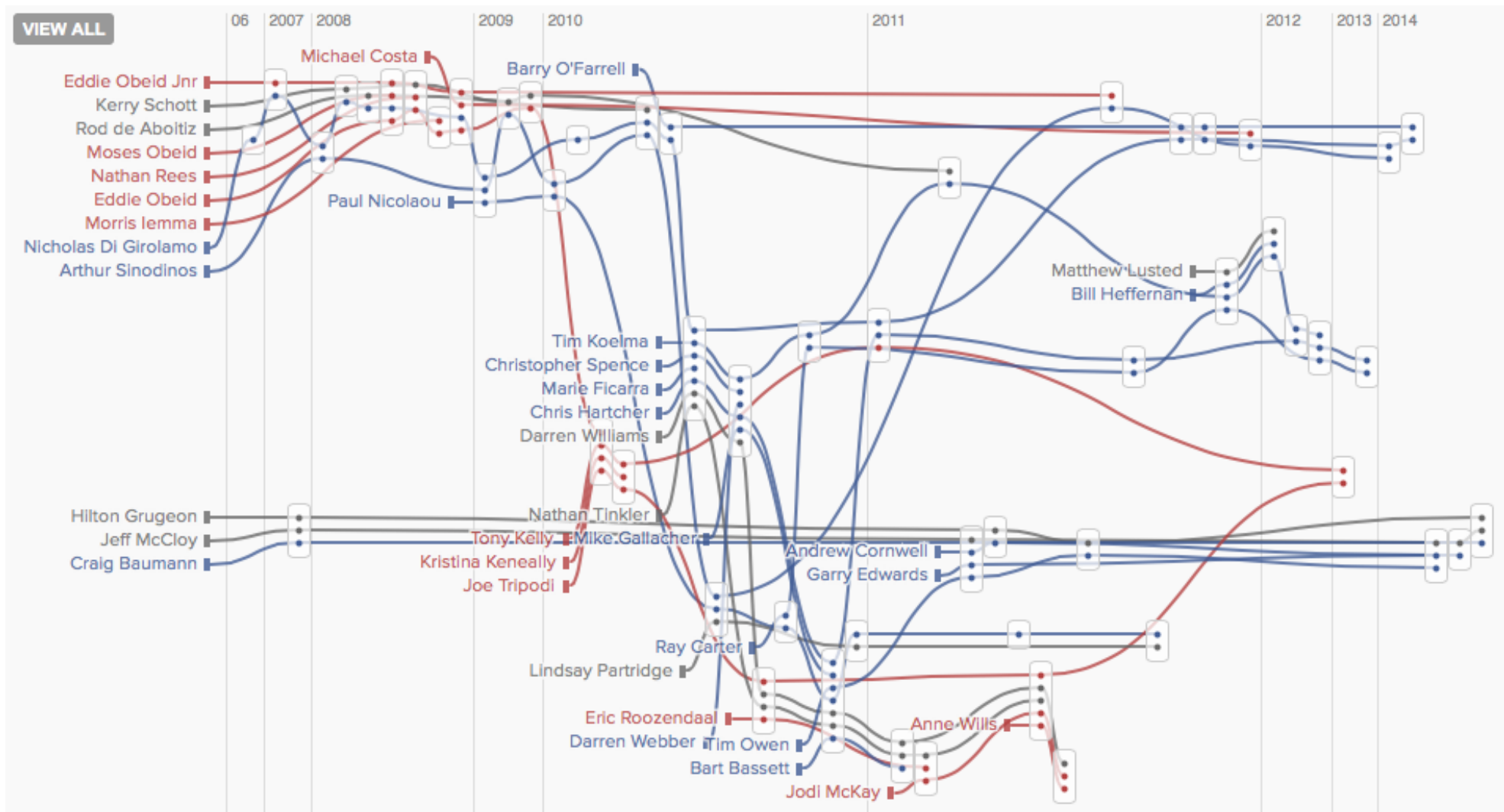


**H. Rodgin Cohen, Chairman**  
The world's go-to lawyer for sovereign wealth investments in financial services firms. He worked on twice as many sovereign wealth related deals than any other individual.

Source: Dealogic, the companies

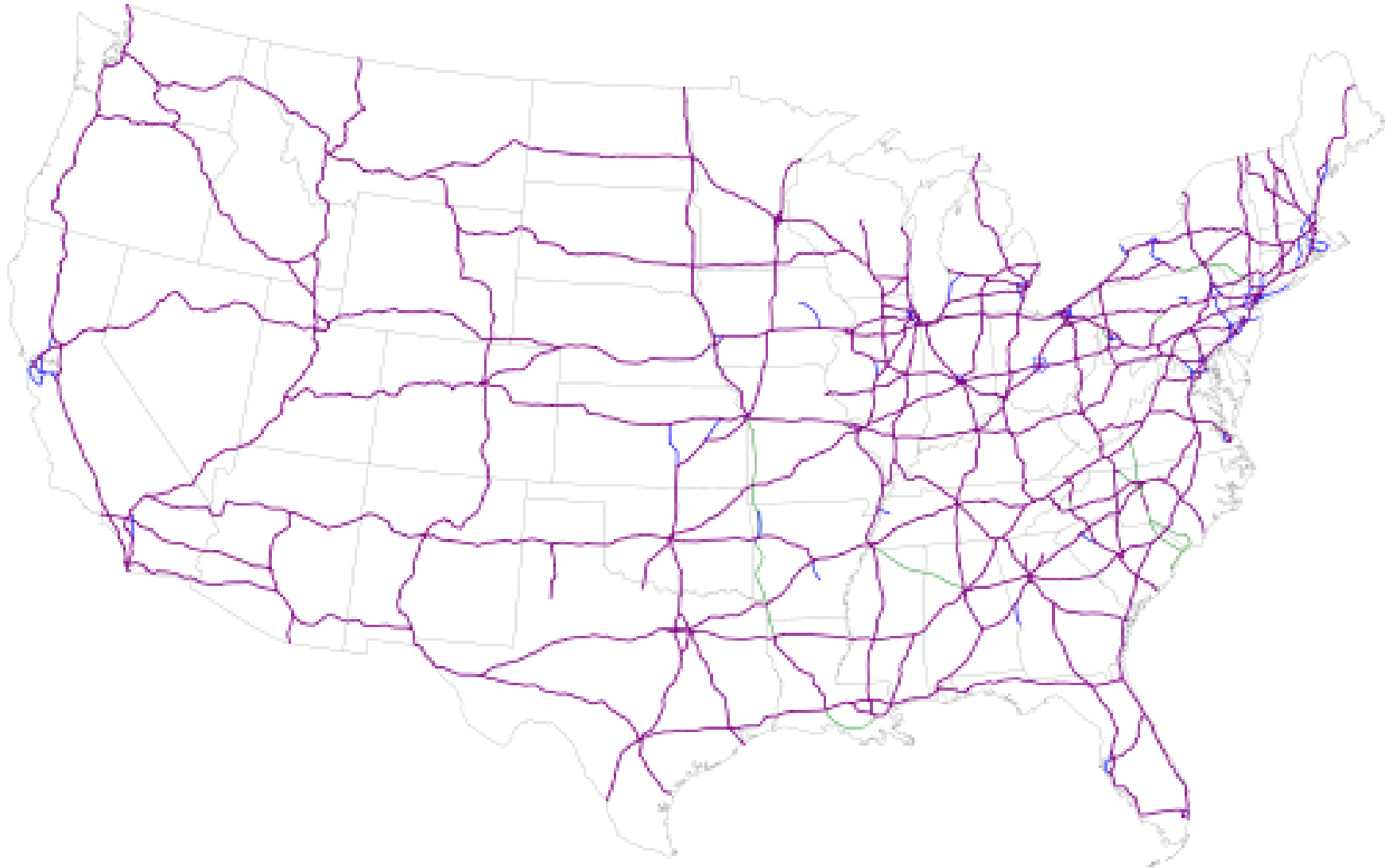
RESEARCH BY MICHAEL DE LA MERCEZ, GRAPHIC BY GILBERT GATES FOR THE NEW YORK TIMES

# Temporal Graph Layout: Storylines



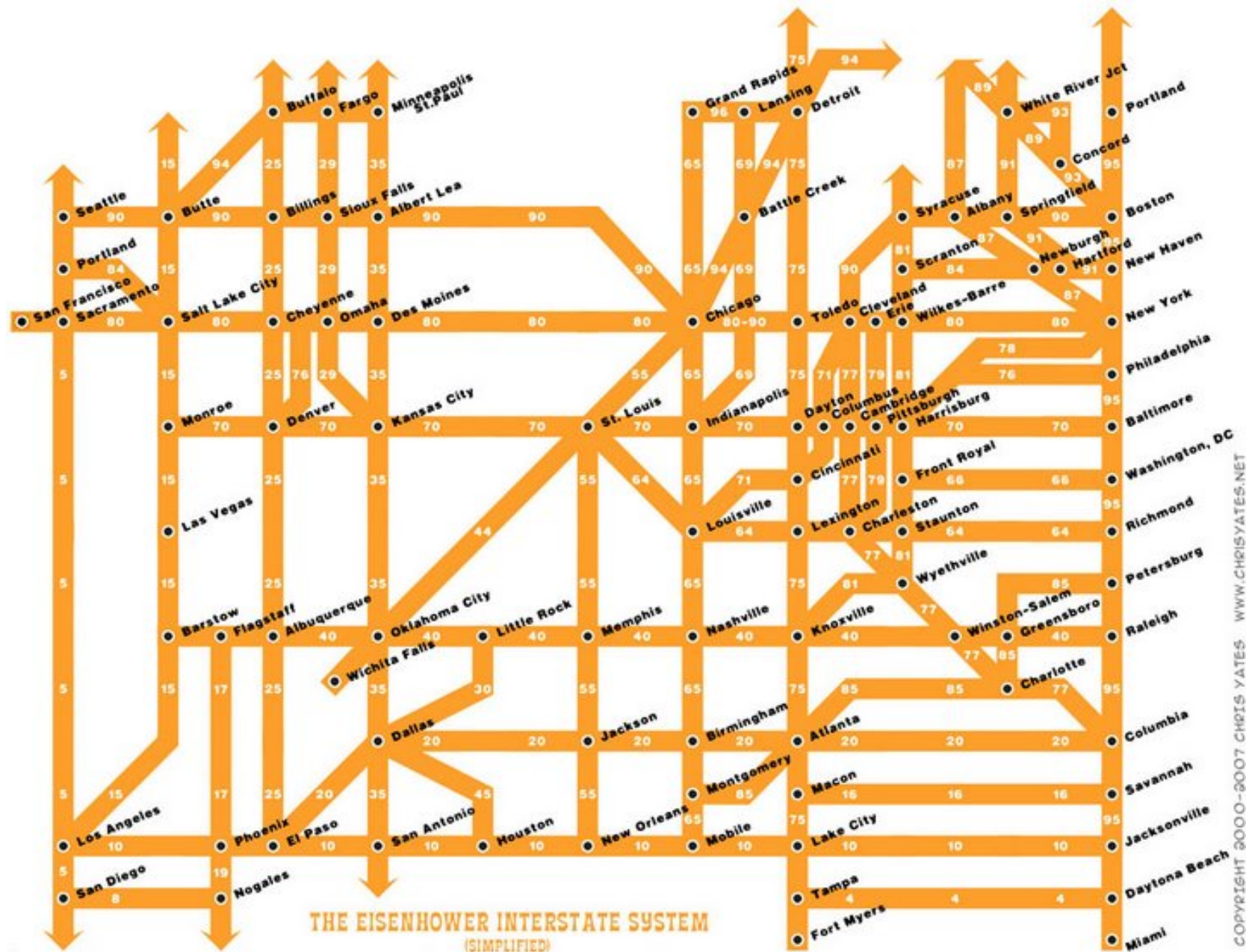
Quelle: ABC news, Australia

# Verkehrsnetze – Highways USA

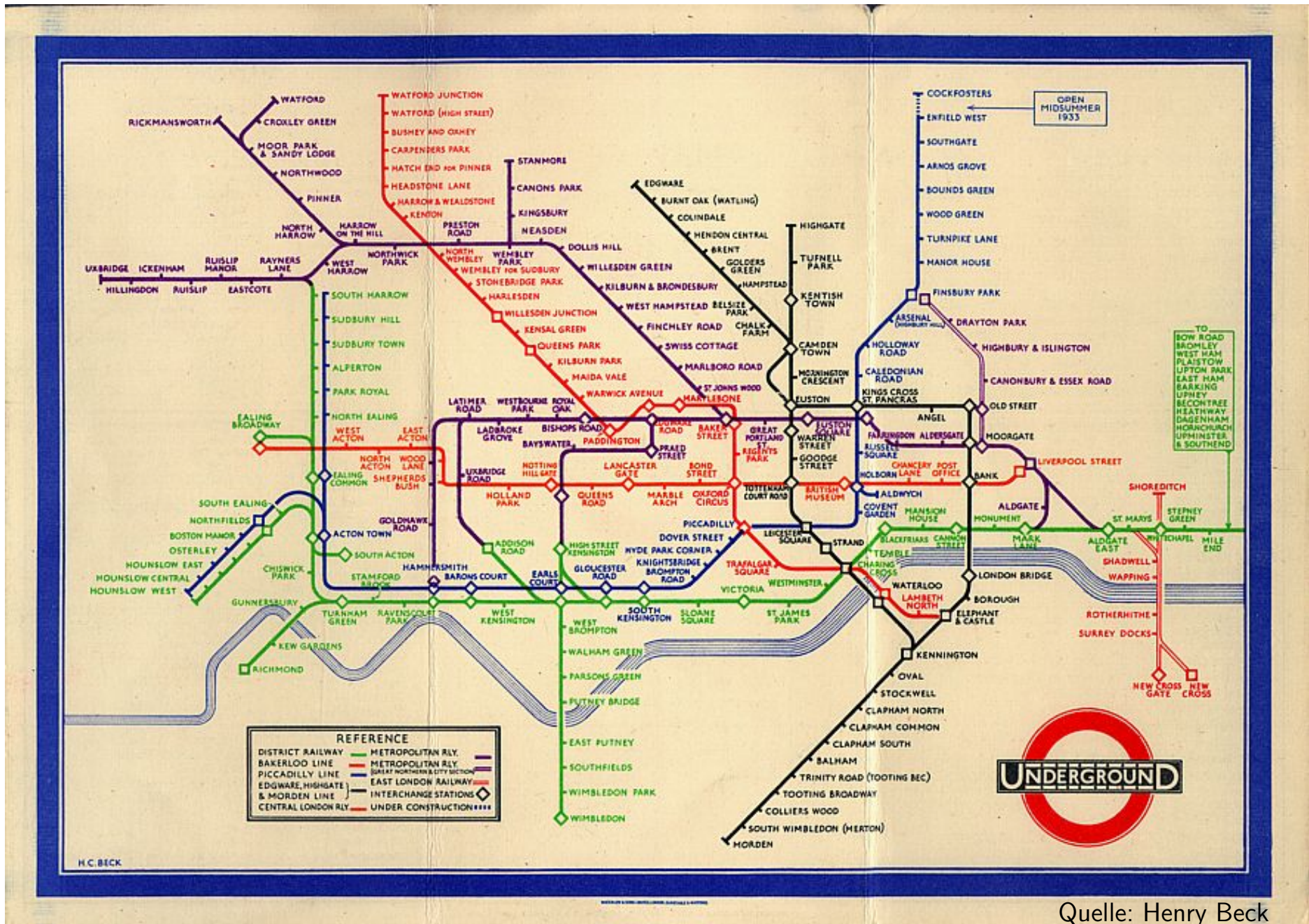




# Verkehrsnetze – Highways USA

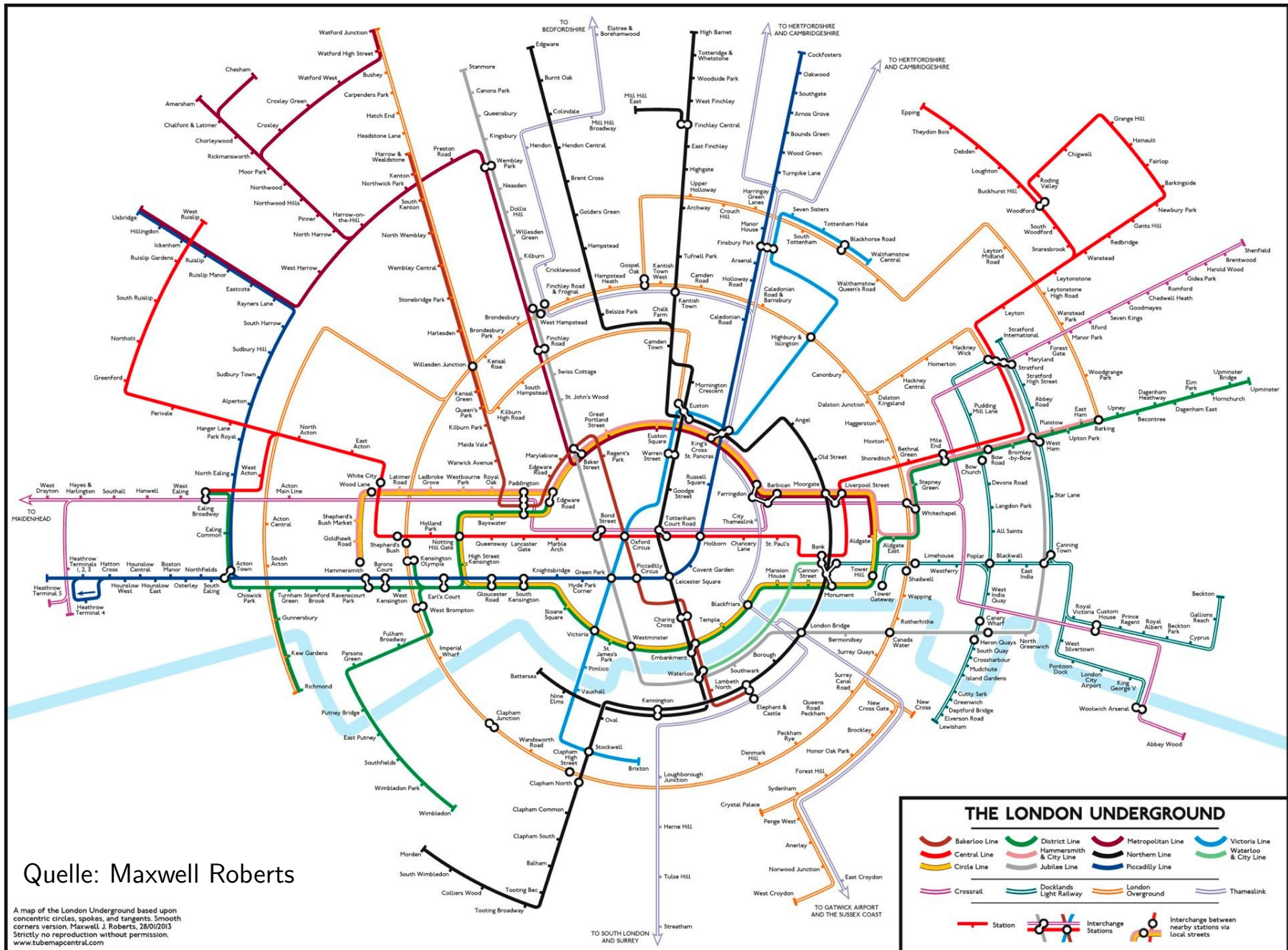


# London Tube Map (1933)

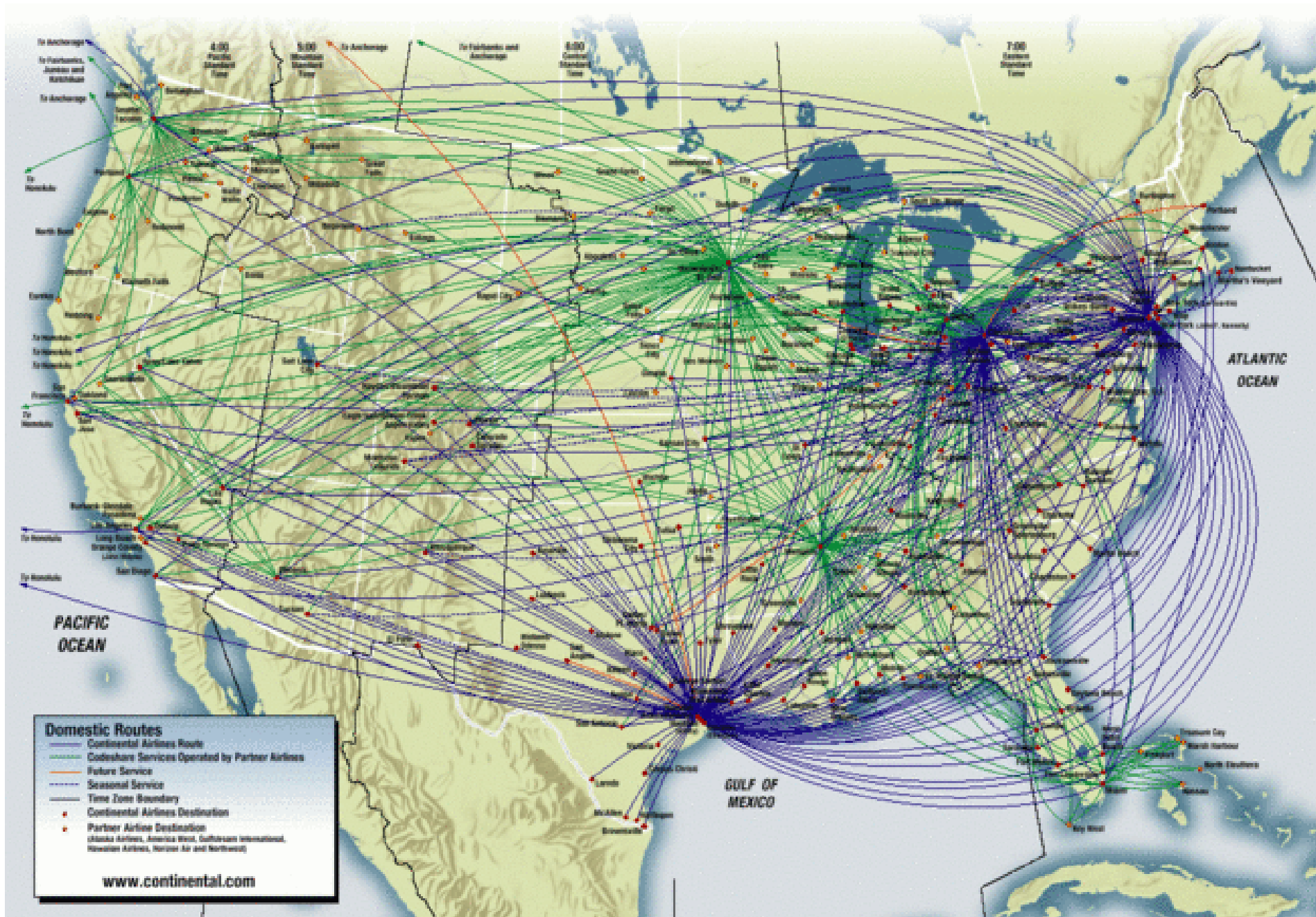


Quelle: Henry Beck

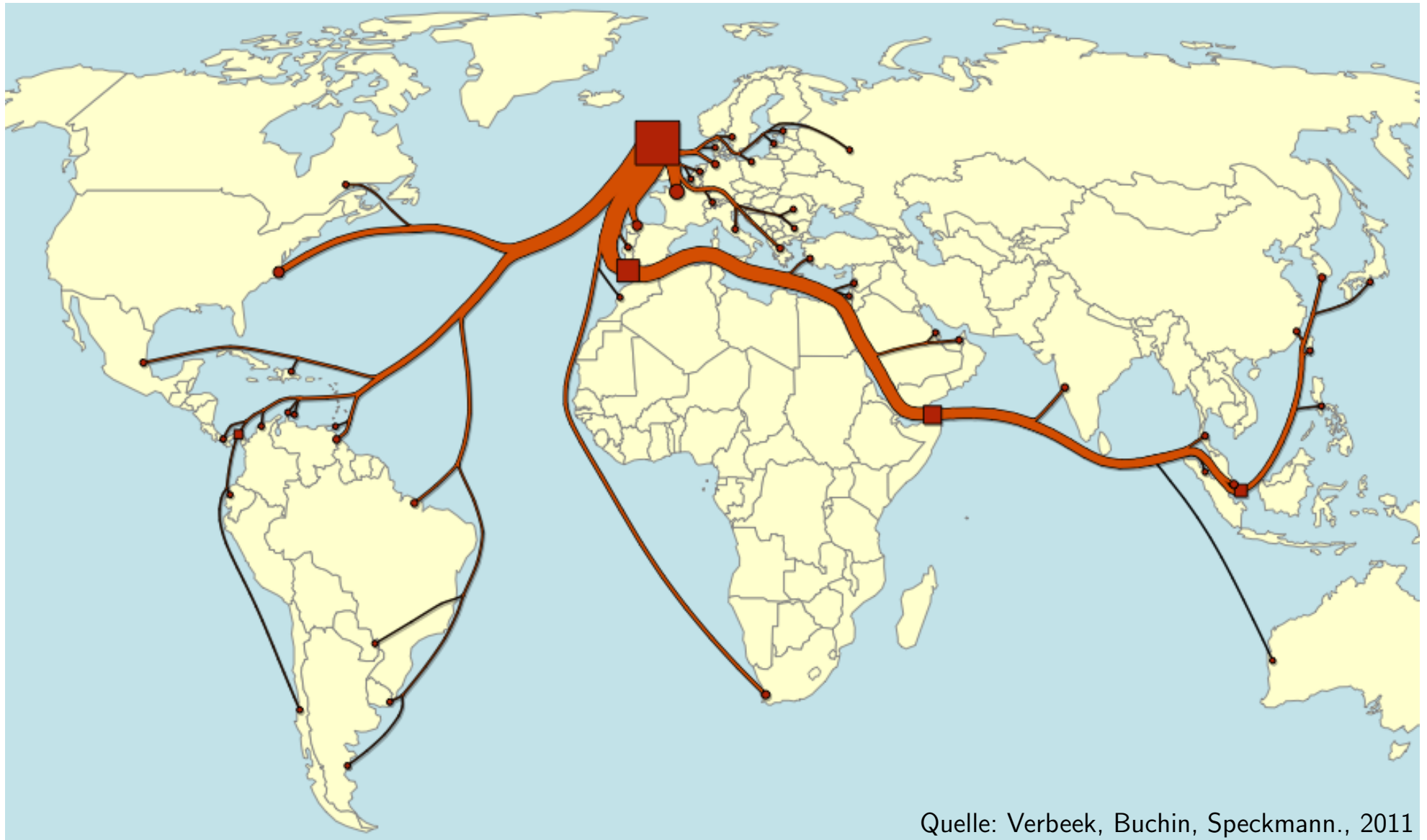
# Konzentrische Tube Map



# Verkehrsnetze – Flugverbindungen Continental

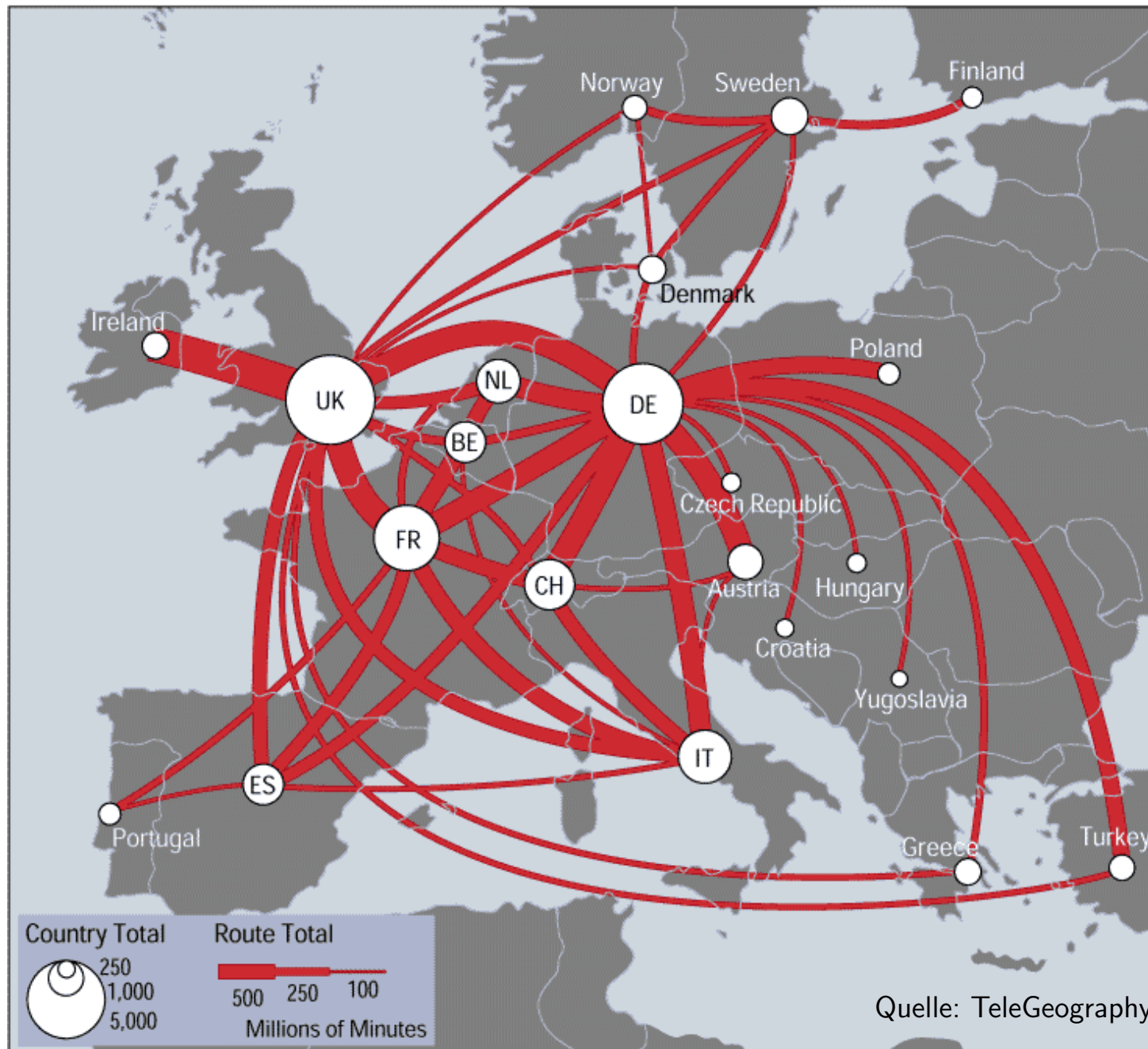


# Flow-Map: Whiskey Exporte



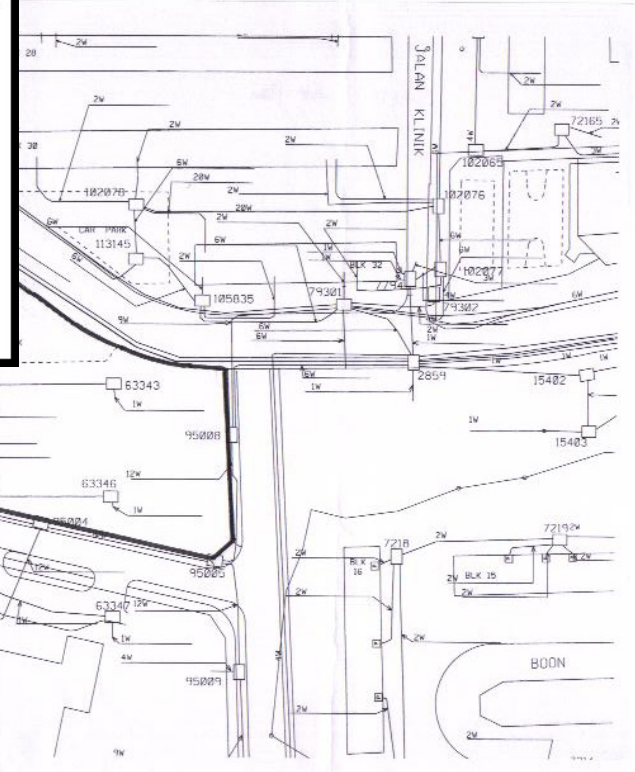
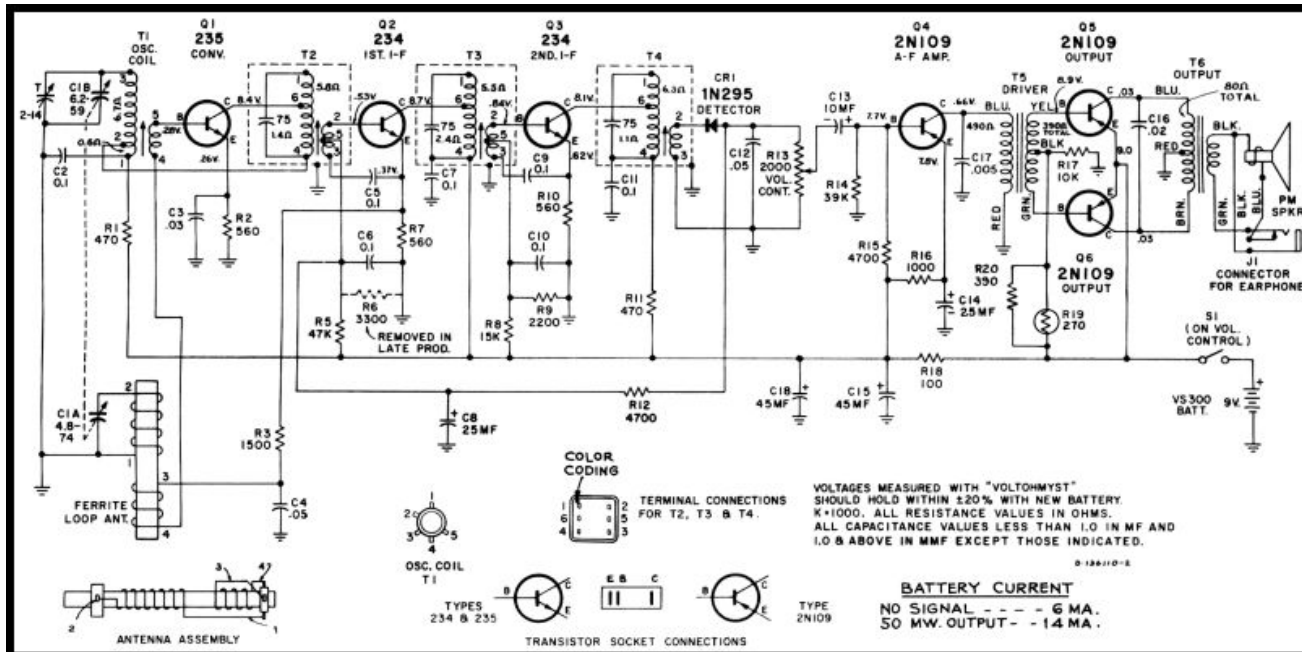
Quelle: Verbeek, Buchin, Speckmann., 2011

# Telefonie-Karte





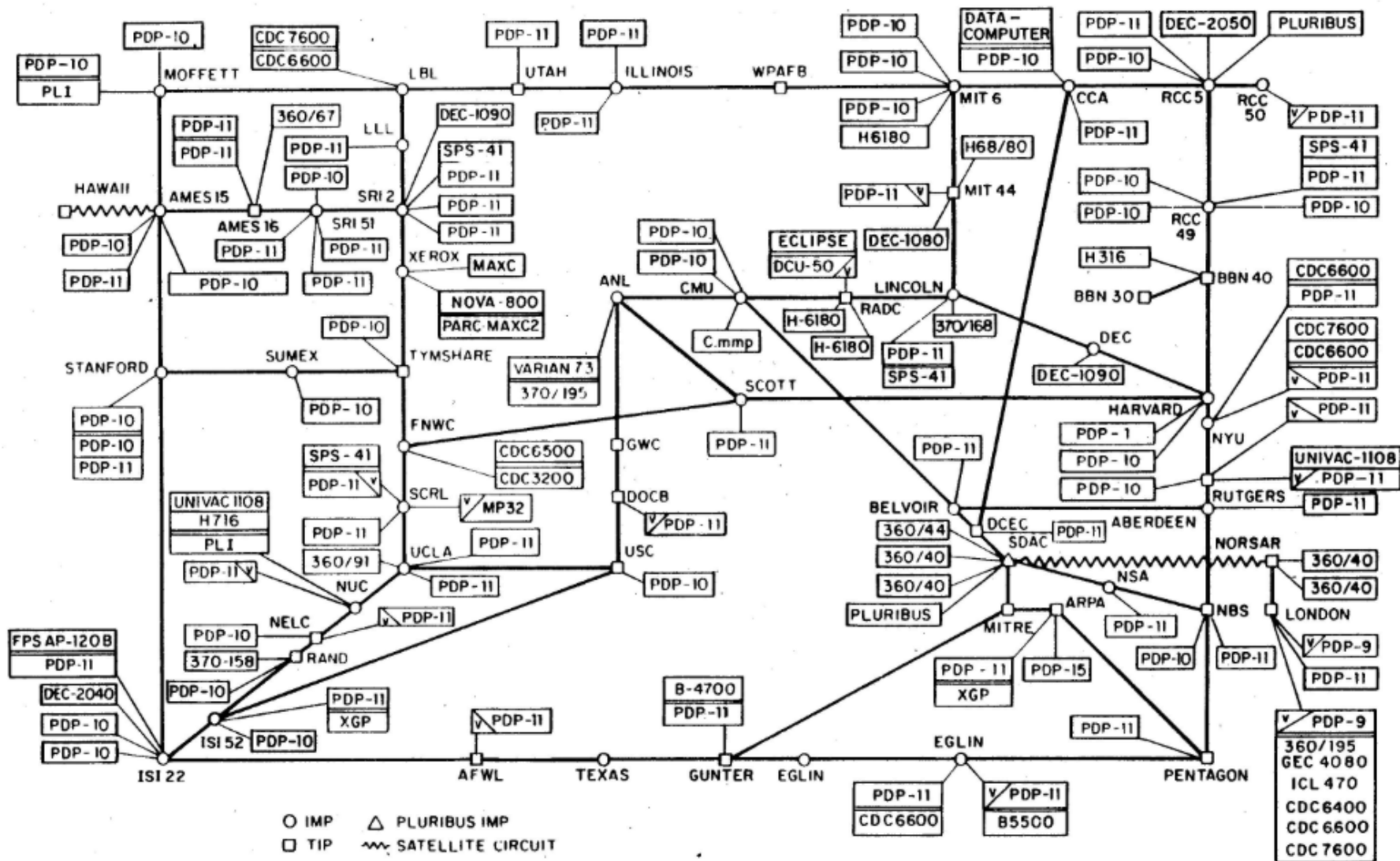
Quelle: Eir Grid, Irland





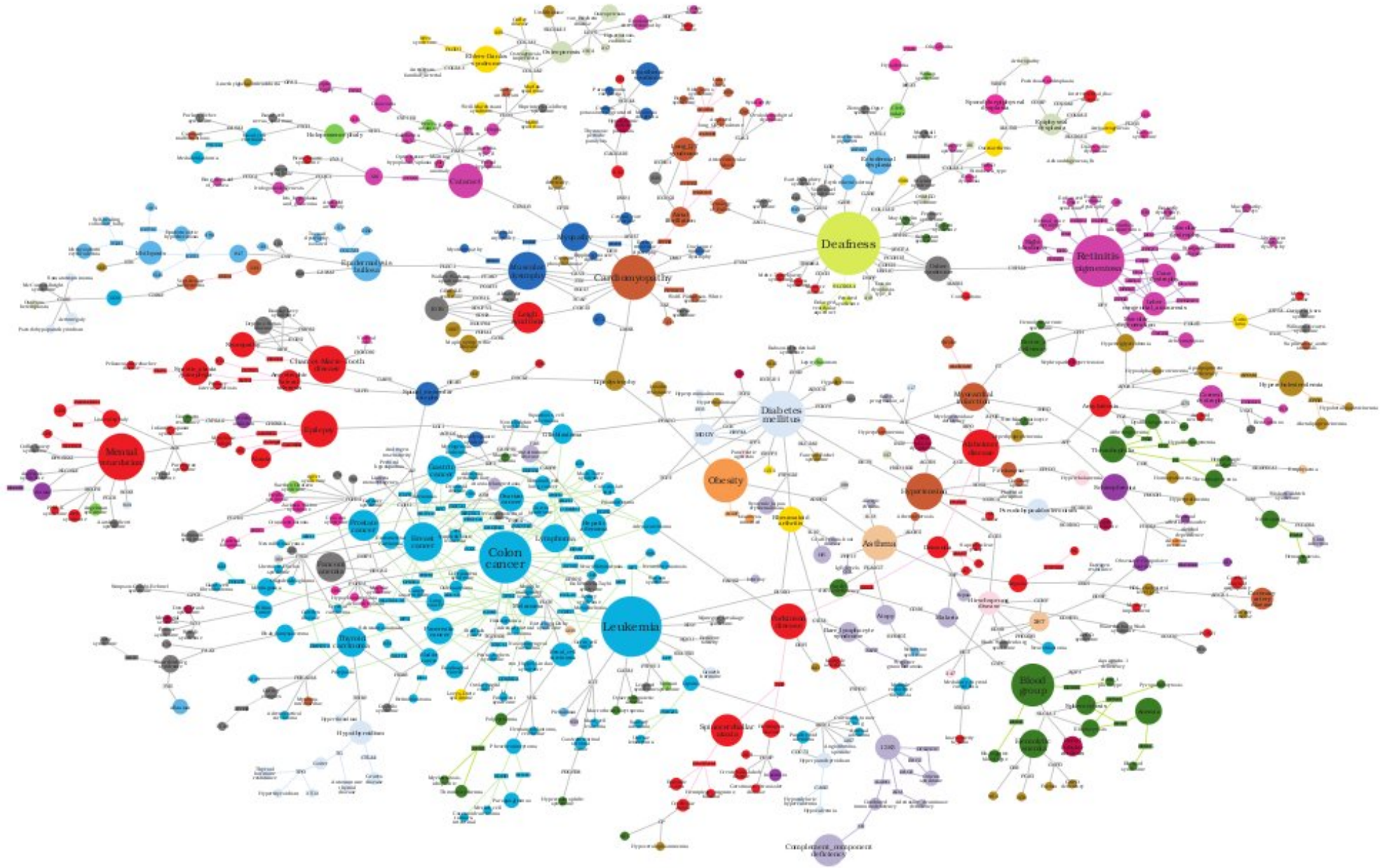
# ARPANET Skizze (1977)

ARPANET LOGICAL MAP, MARCH 1977

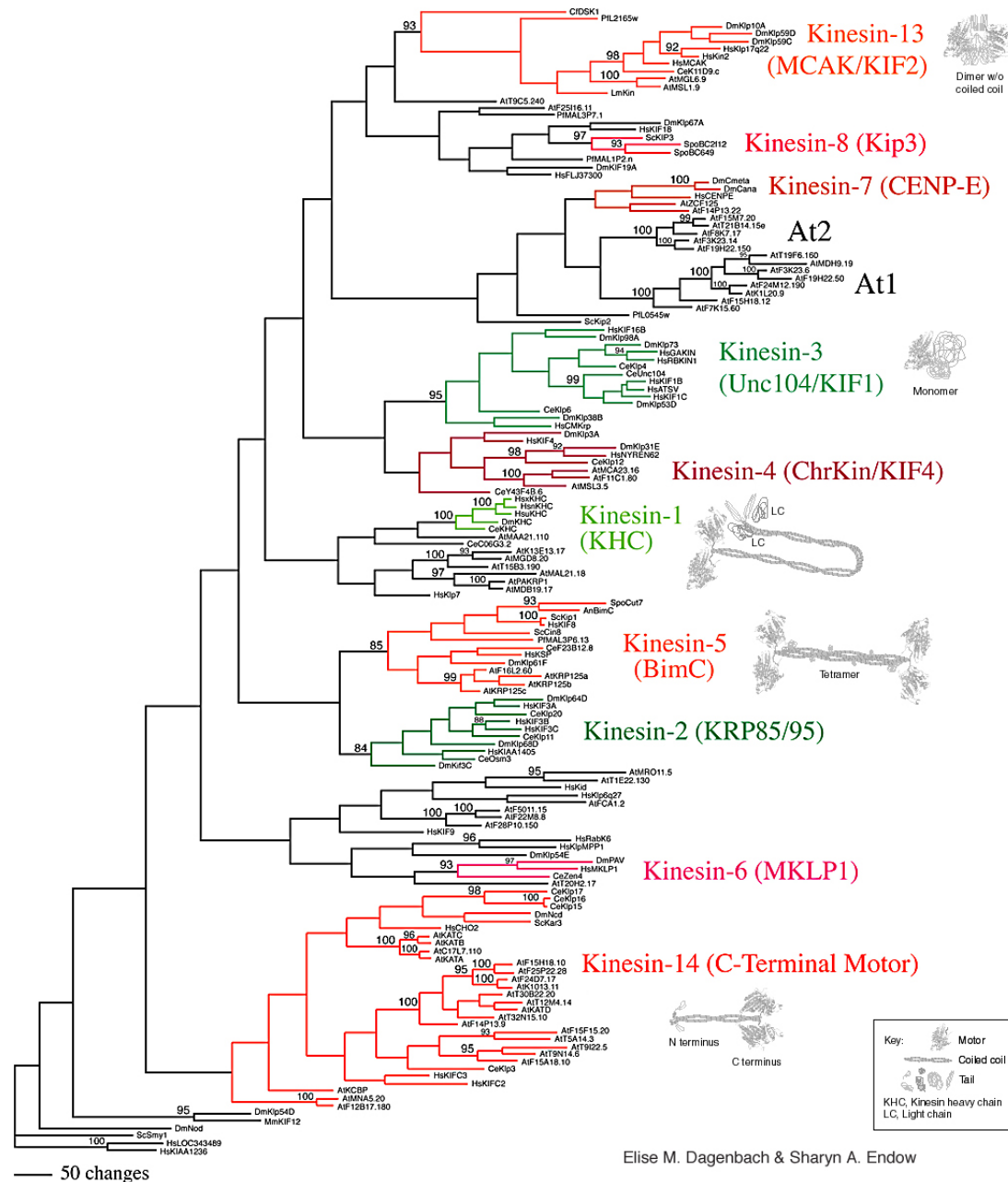


(PLEASE NOTE THAT WHILE THIS MAP SHOWS THE HOST POPULATION OF THE NETWORK ACCORDING TO THE BEST INFORMATION OBTAINABLE, NO CLAIM CAN BE MADE FOR ITS ACCURACY)

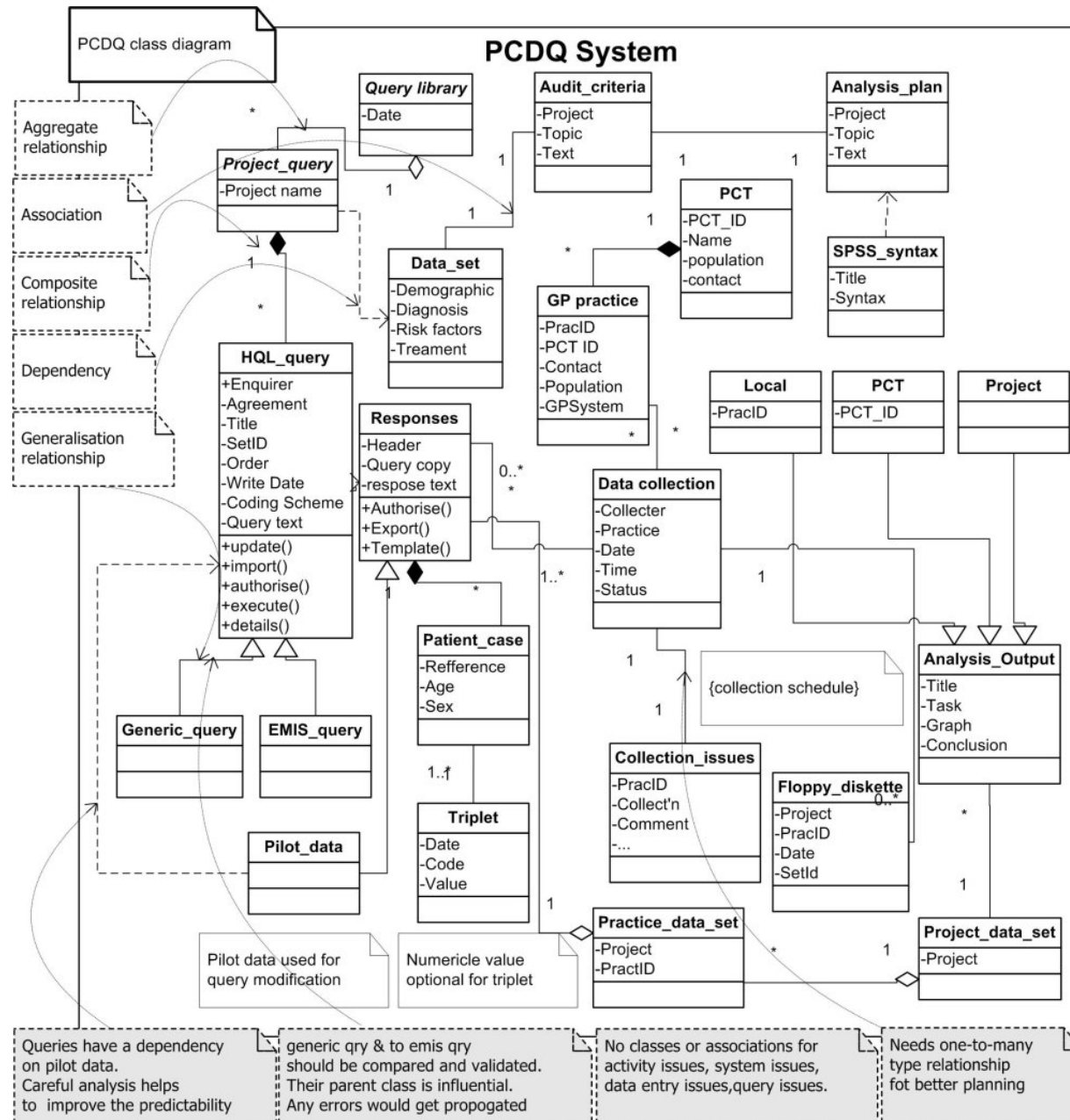
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES



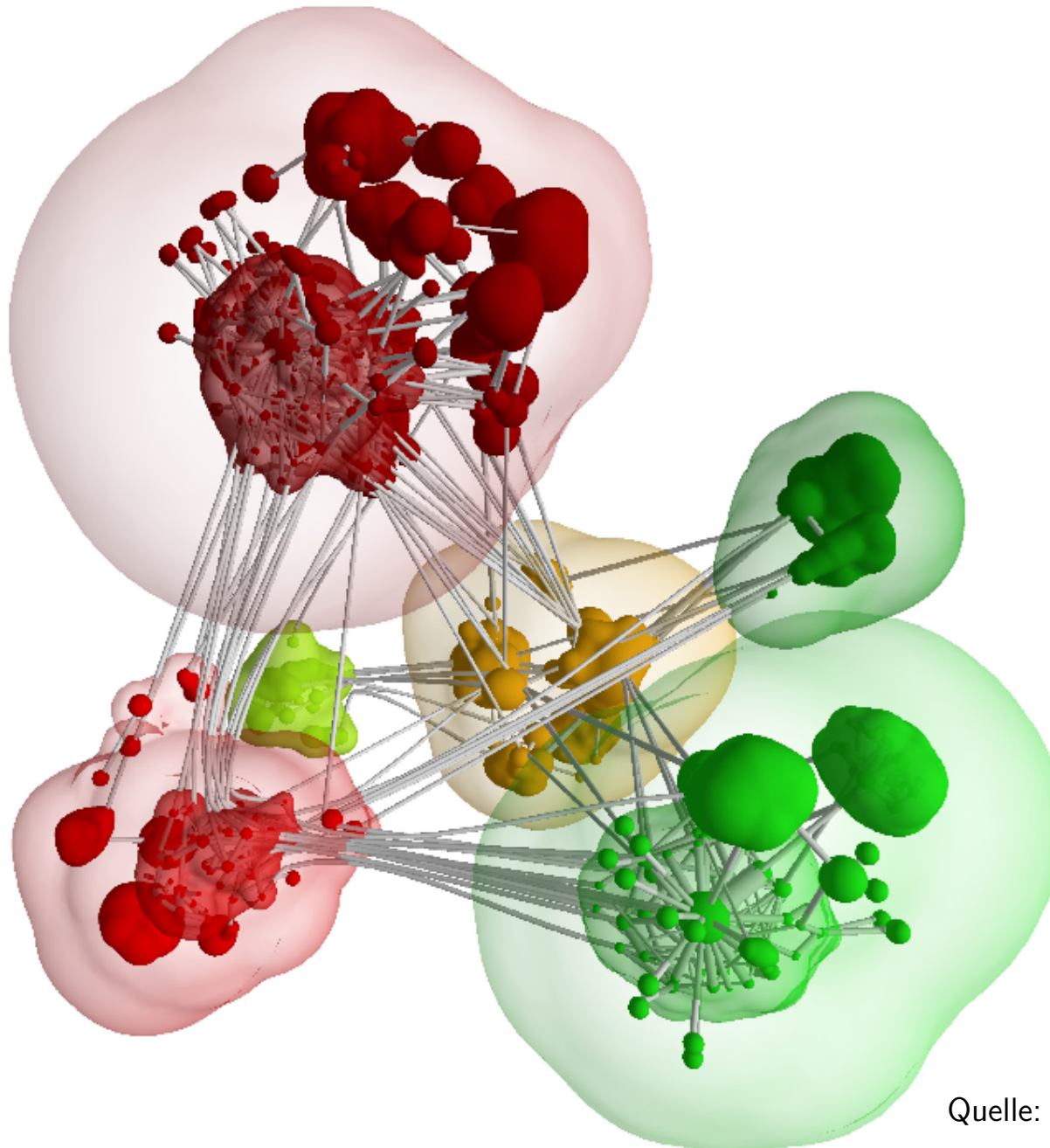
# Biomedizin – phylogenetische Bäume



# Software-Netze – UML Diagramme

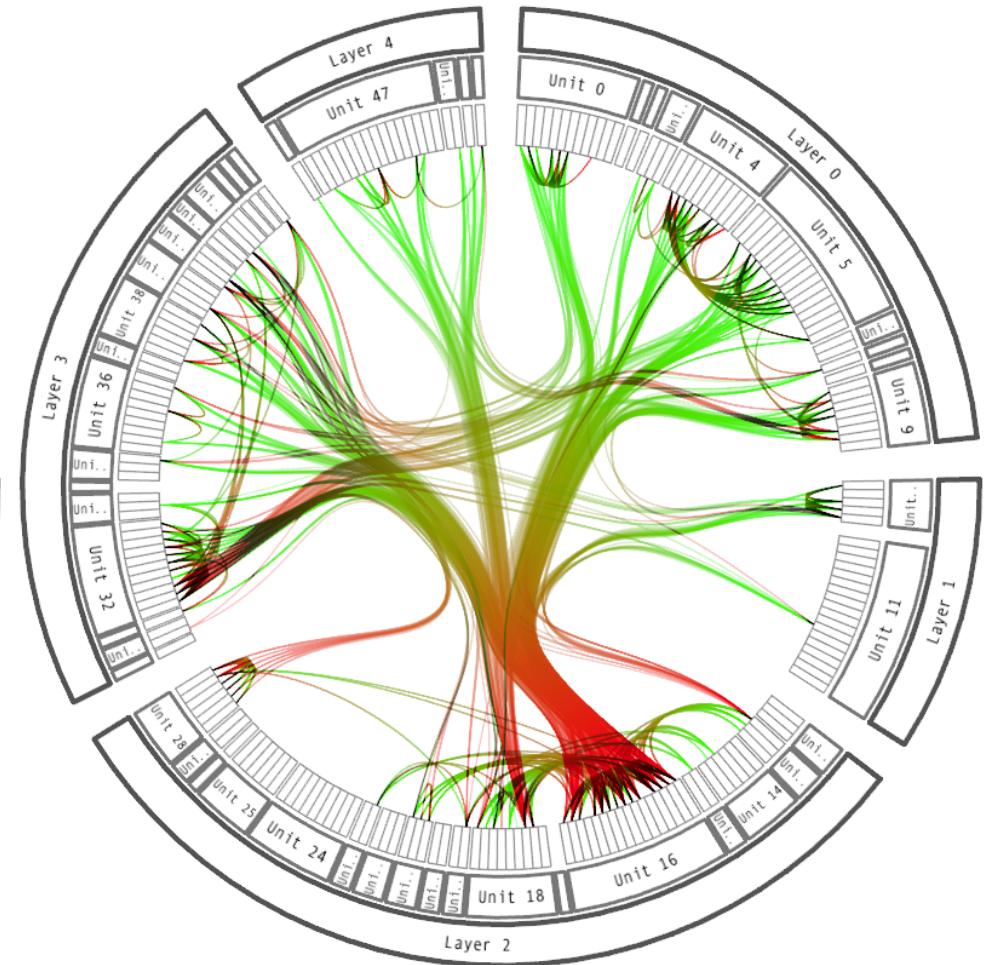
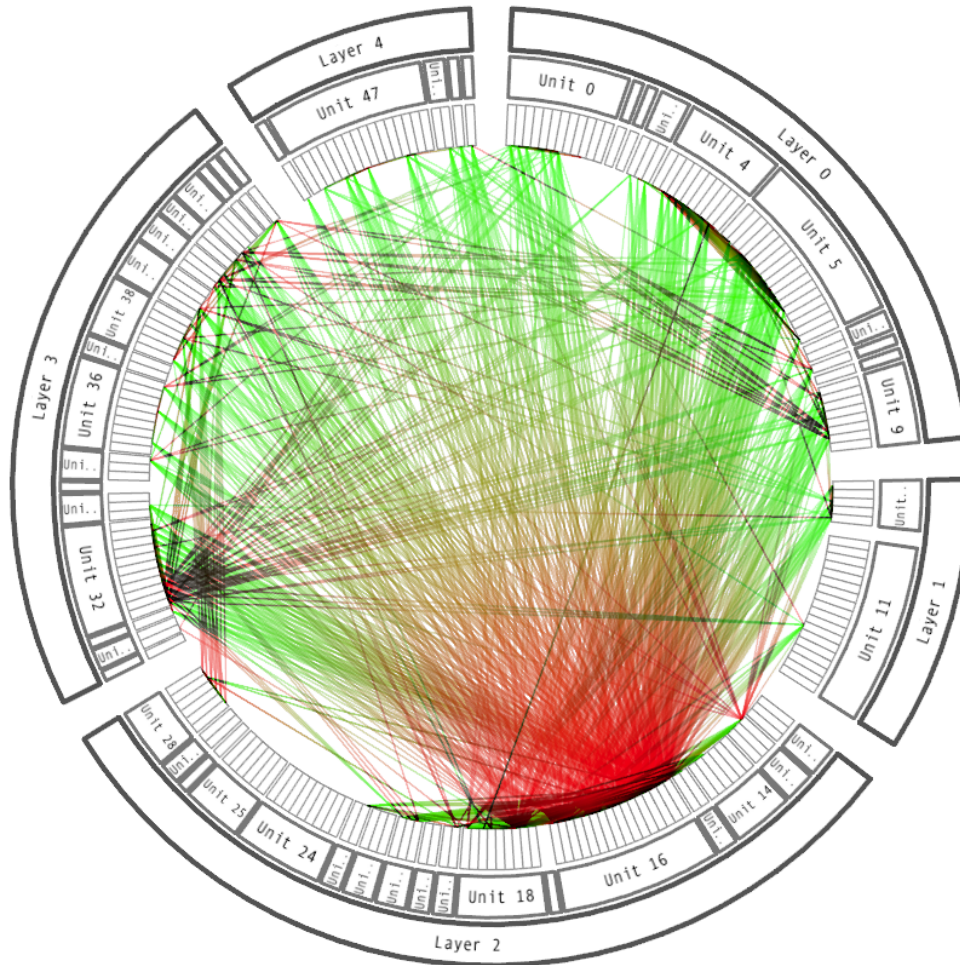


# Geclusterter Software-Graph in 3D



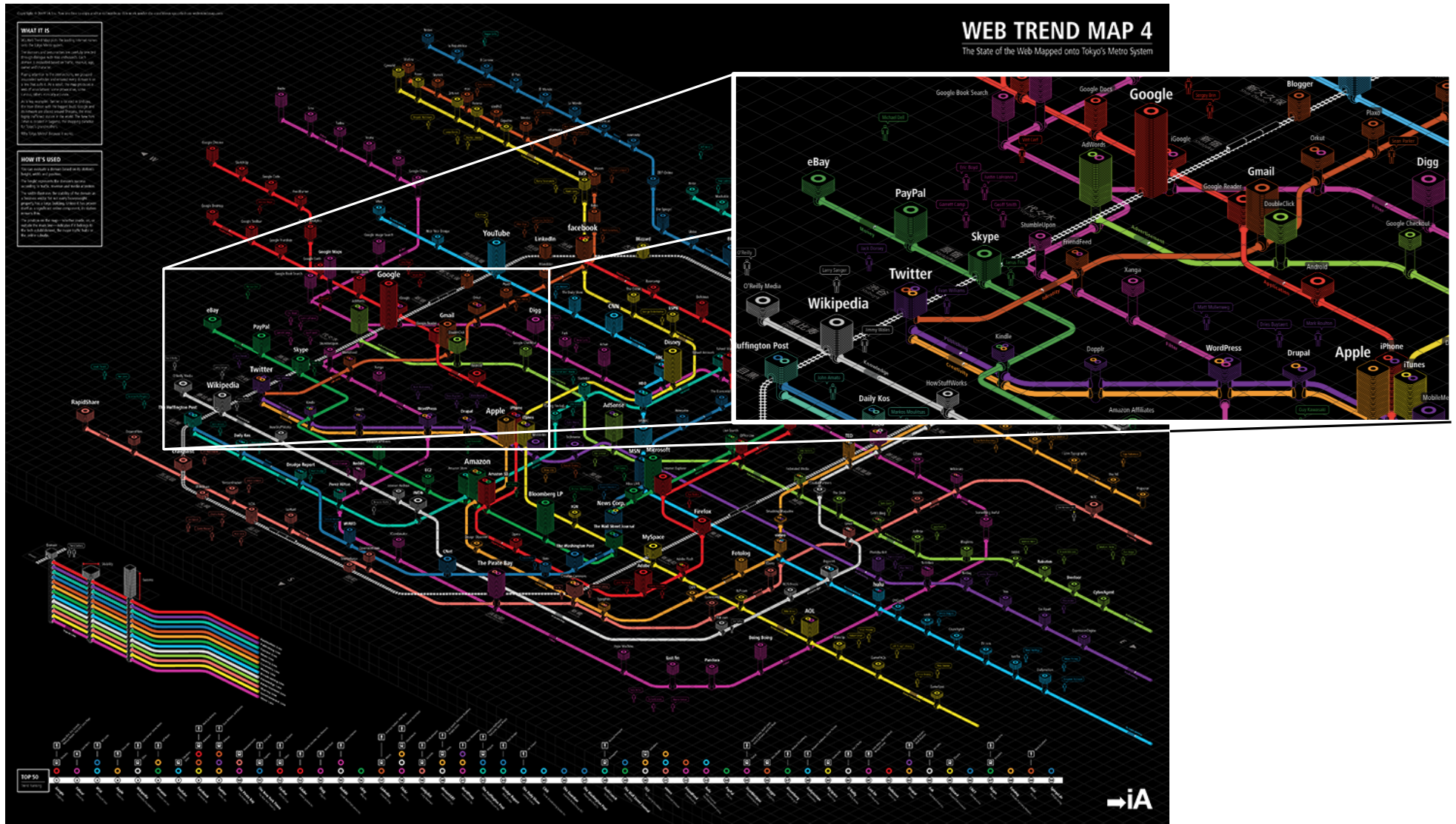
Quelle: Balzer, Deussen, 2007

# Software Call-Graph mit Kantenbündelung

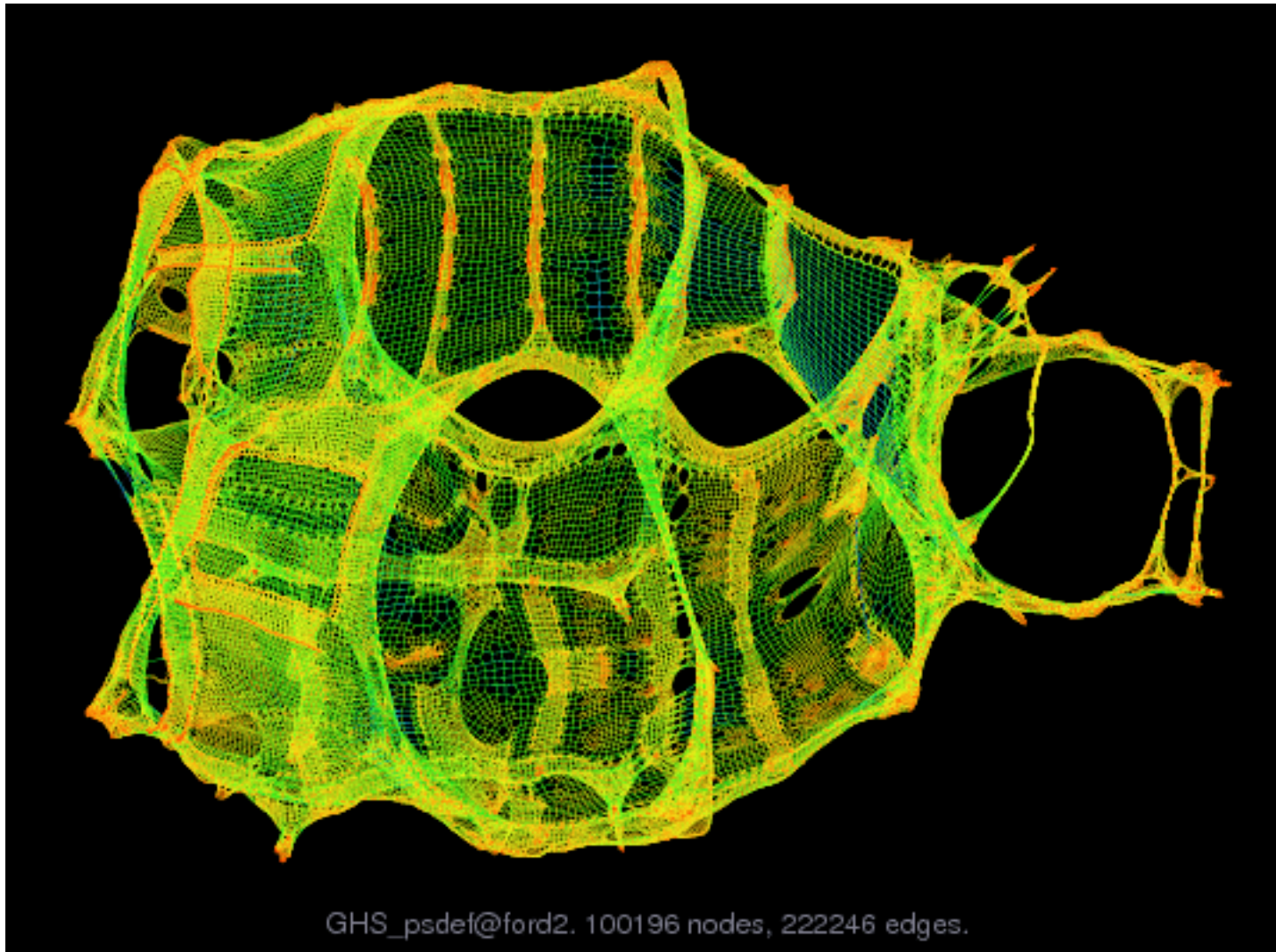


Quelle: Danny Holten, 2011

# Web Trend Map



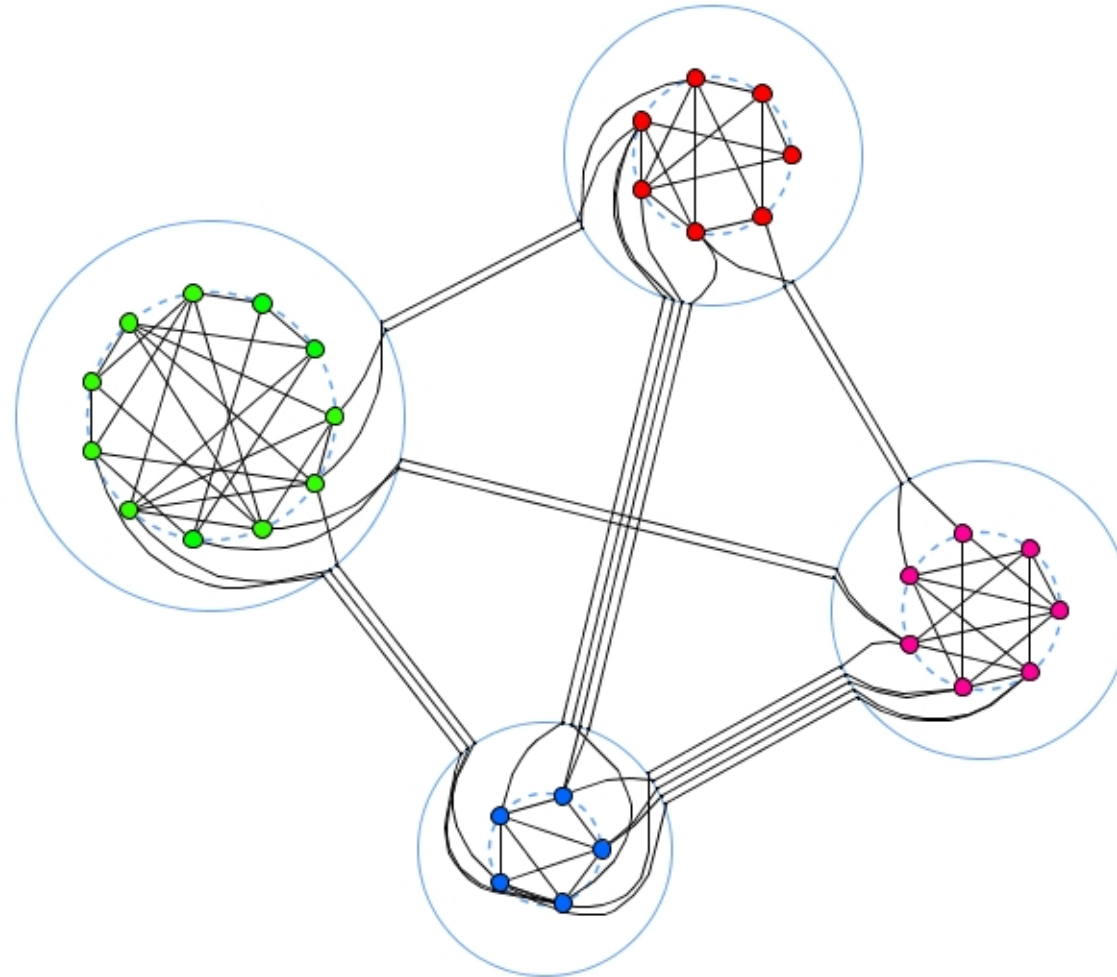
Quelle: information Architects, 2009



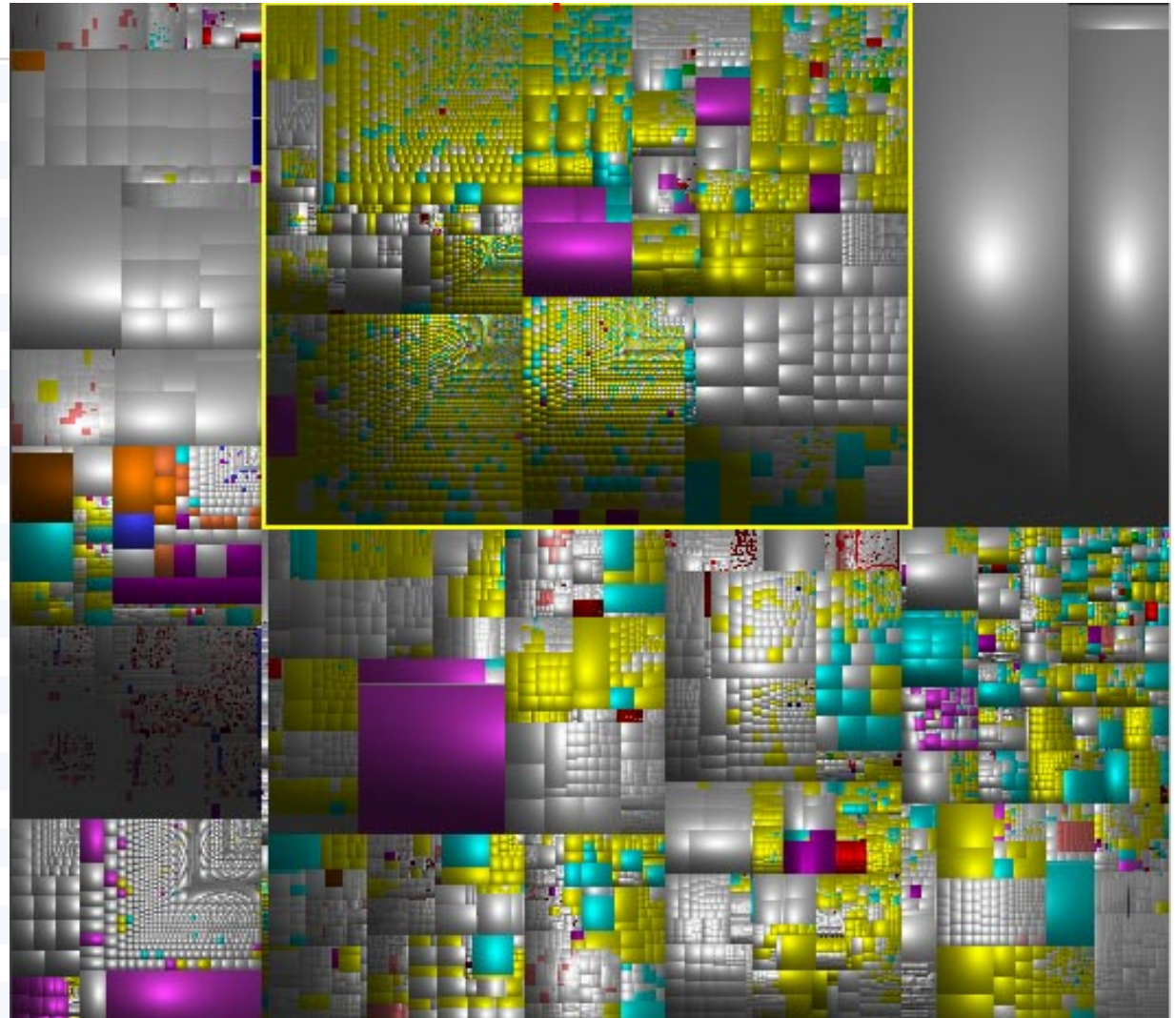
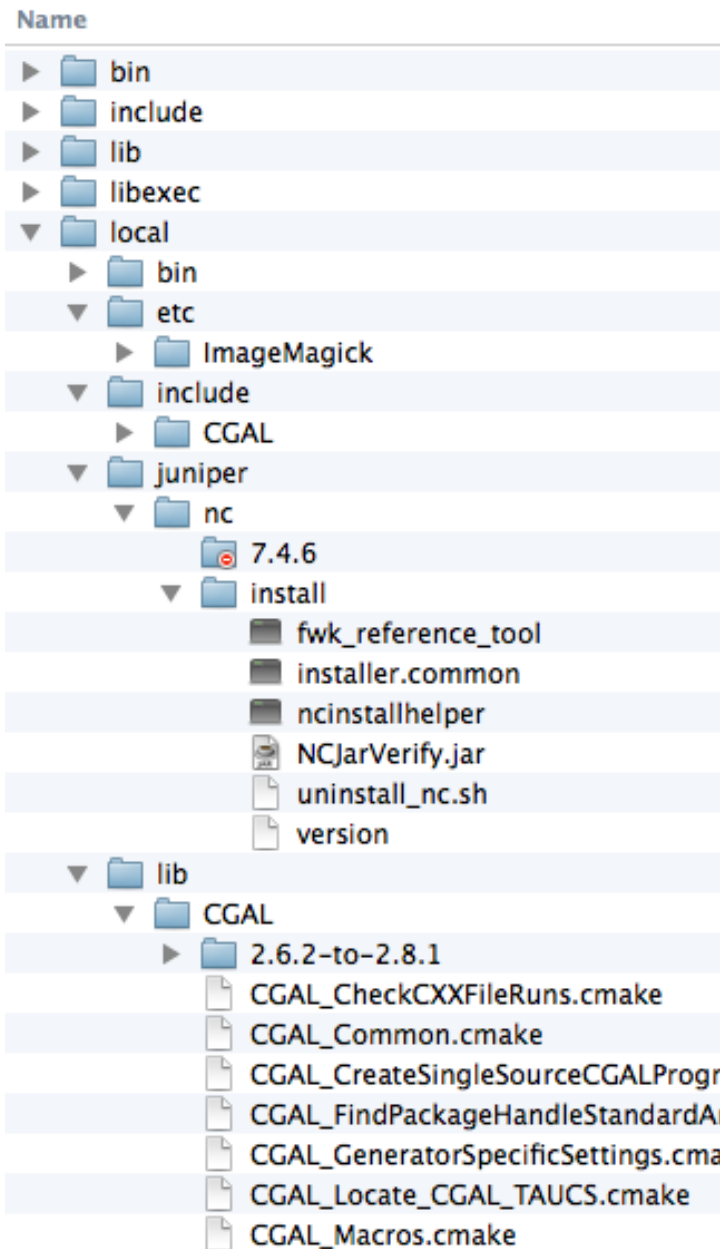
Quelle: Yifan Hu

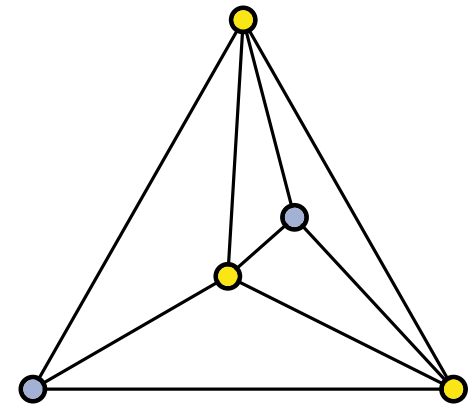
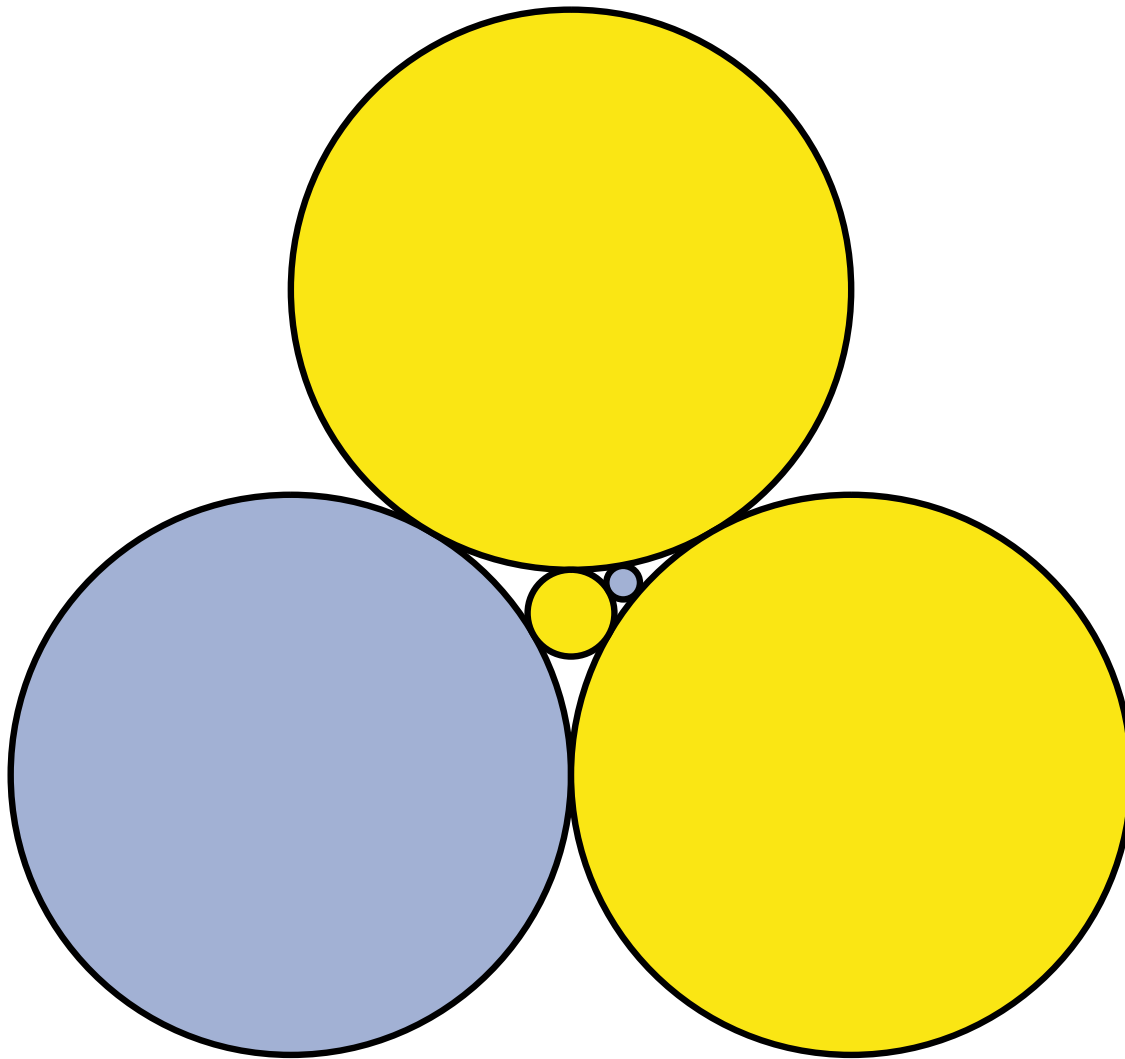


# Allgemeine Graphen – Mikro-Makro Layout



# Dateisystem: Explorer vs Treemap





## Bibliotheken zur Graphvisualisierung

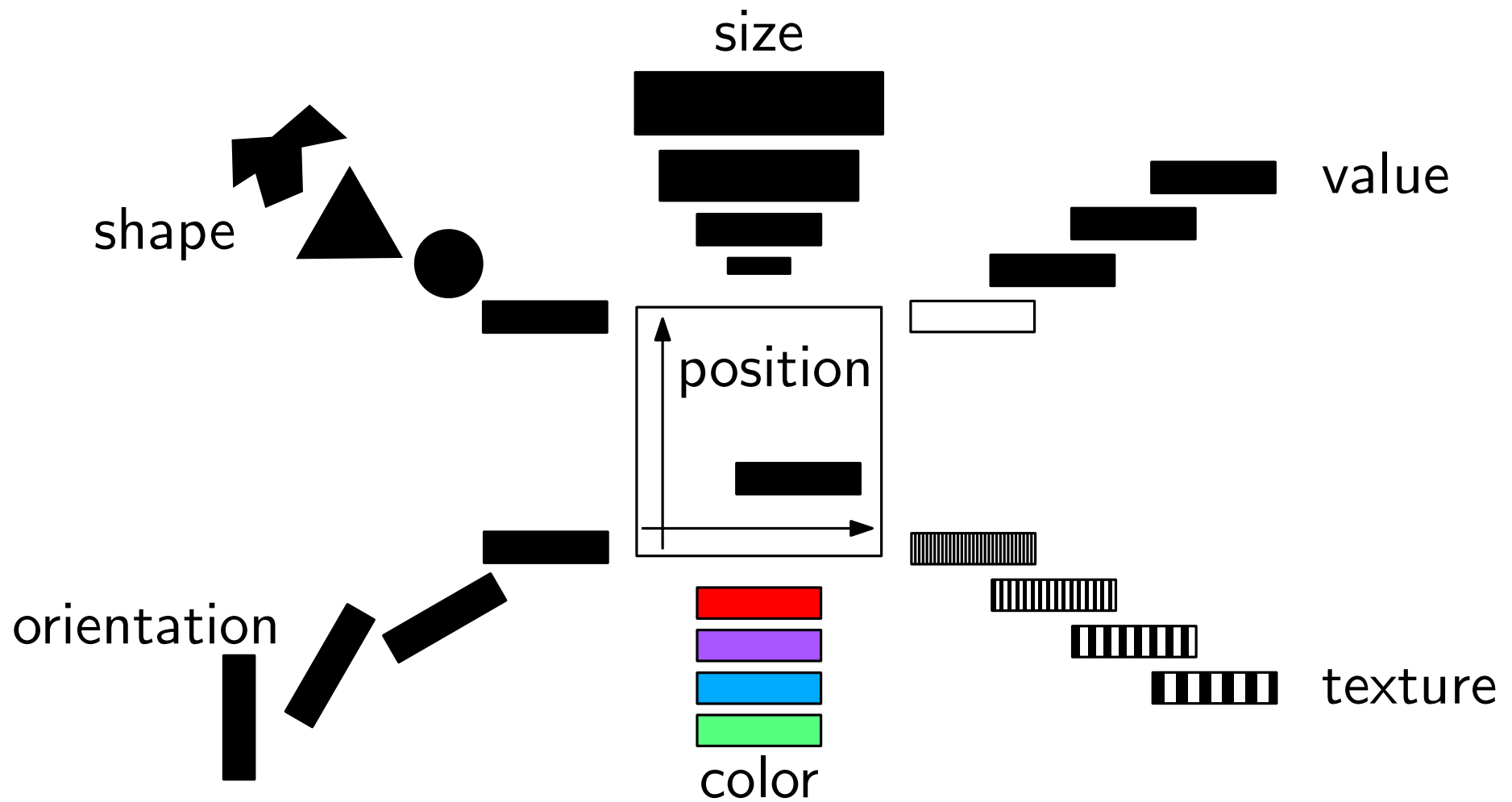
- JUNG [jung.sourceforge.net](http://jung.sourceforge.net) (Java)
- OGDF [www.ogdf.net](http://www.ogdf.net) (C++)

## Visualisierungs-Tools

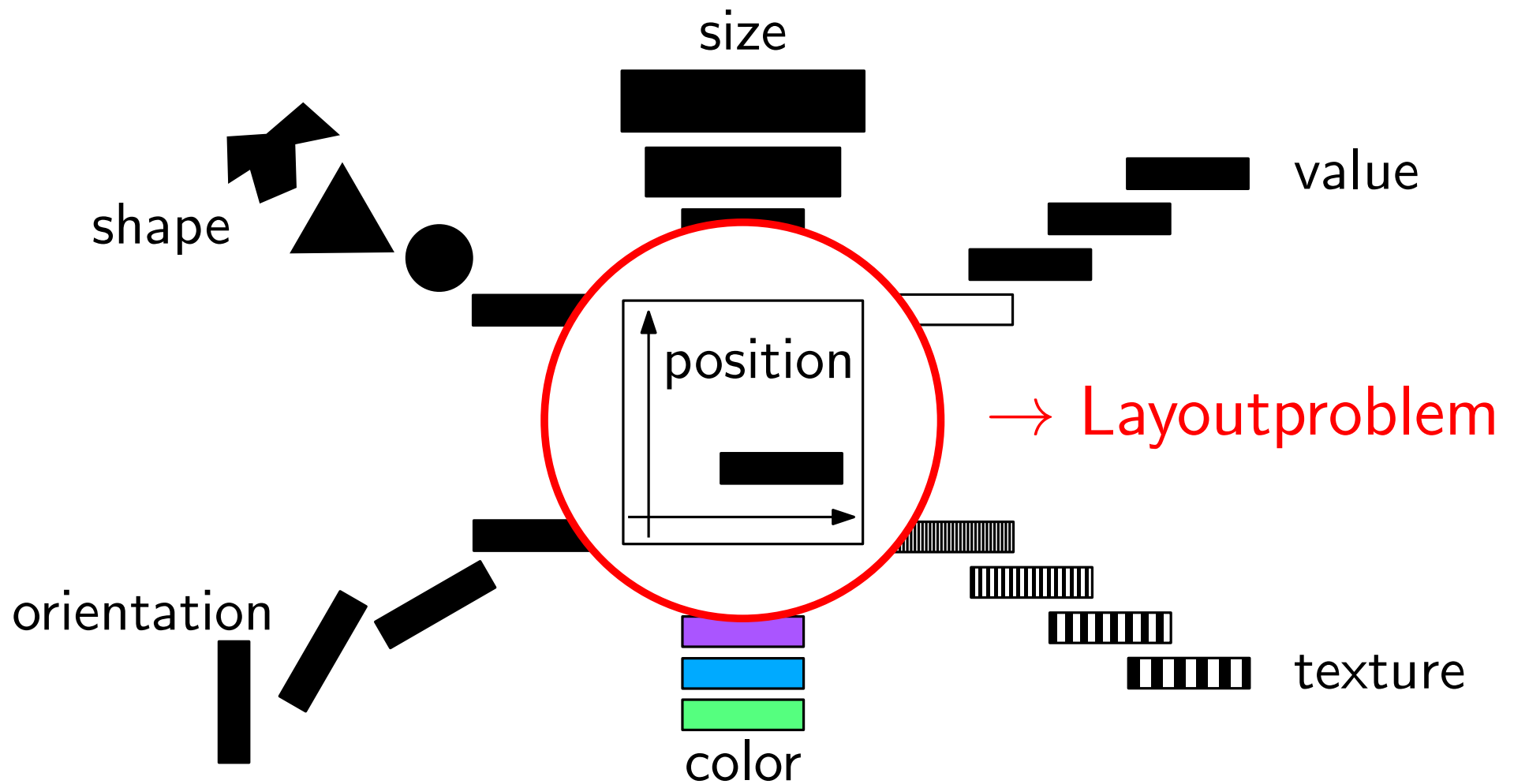
- visone [visone.info](http://visone.info)
- graphviz [www.graphviz.org](http://www.graphviz.org)
- yEd [www.yworks.com](http://www.yworks.com)
- Gephi [www.gephi.org](http://www.gephi.org)

# Grundlegende Definitionen

# Visuelle Variablen nach Bertin (1967)



# Visuelle Variablen nach Bertin (1967)



# Das Layoutproblem

*hier:* Beschränkung auf die sog. **Standardrepräsentation**  
(node-link diagram)

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** gute Zeichnung  $\Gamma$  von  $G$

- $\Gamma : V \rightarrow \mathbb{R}^2$ , Knoten  $v \mapsto$  Punkt  $\Gamma(v)$
- $\Gamma : E \rightarrow$  Kurven in  $\mathbb{R}^2$ , Kante  $\{u, v\} \mapsto$  einf. offene Kurve  $c_{uv} : [0, 1] \rightarrow \mathbb{R}^2$  mit  $c_{uv}(0) = \Gamma(u)$  und  $c_{uv}(1) = \Gamma(v)$



# Das Layoutproblem

*hier:* Beschränkung auf die sog. **Standardrepräsentation**  
(node-link diagram)

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** gute Zeichnung  $\Gamma$  von  $G$

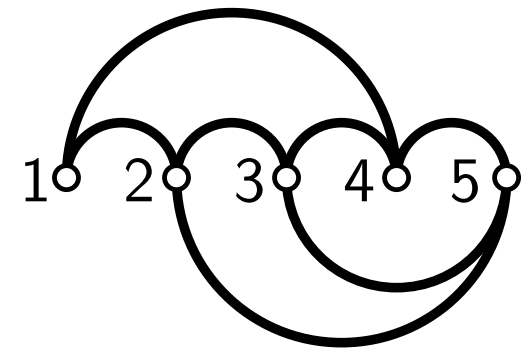
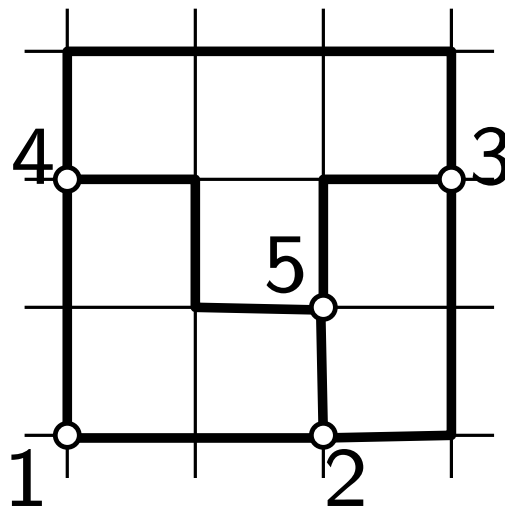
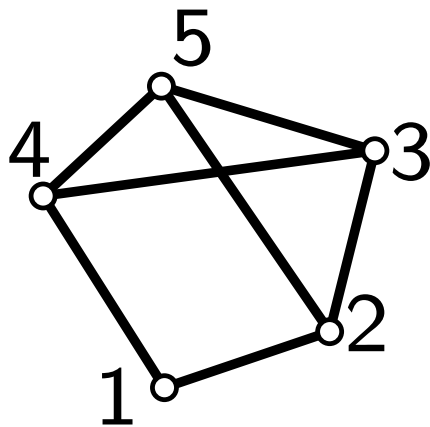
- $\Gamma : V \rightarrow \mathbb{R}^2$ , Knoten  $v \mapsto$  Punkt  $\Gamma(v)$
- $\Gamma : E \rightarrow$  Kurven in  $\mathbb{R}^2$ , Kante  $\{u, v\} \mapsto$  einf. offene Kurve  $c_{uv} : [0, 1] \rightarrow \mathbb{R}^2$  mit  $c_{uv}(0) = \Gamma(u)$  und  $c_{uv}(1) = \Gamma(v)$

Aber was ist eine gute Zeichnung?

# Anforderungen an ein Graphlayout

## 1) **Zeichenkonventionen**, erforderliche Eigenschaften, z.B.

- geradlinige Kanten mit  $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
- orthogonale Kanten (i.A. mit Knicken)
- Gitterzeichnungen
- kreuzungsfrei
- ...



# Anforderungen an ein Graphlayout

1) **Zeichenkonventionen**, erforderliche Eigenschaften

2) **Ästhetikkriterien** (zu optimieren), z.B.

- Kreuzungsminimierung
- Knickminimierung
- gleichmäßige Kantenlängen
- minimale Gesamtlänge/Fläche
- Winkelauflösung
- Symmetrie / Struktur
- ...

→ führen häufig zu NP-schweren Optimierungsproblemen!

→ oft mehrere konkurrierende Kriterien

# Anforderungen an ein Graphlayout

- 1) **Zeichenkonventionen**, erforderliche Eigenschaften
- 2) **Ästhetikkriterien** (zu optimieren)
- 3) **Lokale Nebenbedingungen**, z.B.
  - Positionseinschränkungen für Nachbarknoten
  - Einschränkungen für Gruppen von Knoten/Kanten

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** **gute** Zeichnung  $\Gamma$  von  $G$ , die

- Zeichenkonventionen erfüllt
- Ästhetikkriterien optimiert
- ggf. weitere Nebenbedingungen erfüllt

## Graphvisualisierungsproblem

**geg.:** Graph  $G = (V, E)$

**ges.:** **gute** Zeichnung  $\Gamma$  von  $G$ , die

- Zeichenkonventionen erfüllt
  - Ästhetikkriterien optimiert
  - ggf. weitere Nebenbedingungen erfüllt
- 
- führt zu algorithmisch interessanten Fragestellungen
  - nachgelagertes Renderingproblem bleibt außen vor

- **Zeichnen von Bäumen und weiteren rekursiv definierten Graphklassen**
- geradliniges Zeichnen planarer Graphen
- inkrementelle Layoutverfahren
- orthogonale Gitterzeichnungen
- Kontaktrepräsentationen planarer Graphen
- hierarchische Lagenlayouts gerichteter Graphen
- kräftebasiertes Graphenzeichnen
- ...

- **Zeichnen von Bäumen und weiteren rekursiv definierten Graphklassen**
- geradliniges Zeichnen planarer Graphen
- inkrementelle Layoutverfahren
- orthogonale Gitterzeichnungen
- Kontaktrepräsentationen planarer Graphen
- hierarchische Lagenlayouts gerichteter Graphen
- kräftebasiertes Graphenzeichnen
- ...

## Nächster Termin

Übung am Montag 27.10. 15:45 Uhr

→ bitte (falls vorhanden) Laptops mitbringen

→ Software yEd installieren ([www.yworks.com](http://www.yworks.com))