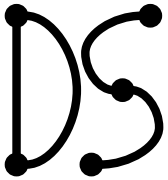


Research Group
Distributed Systems

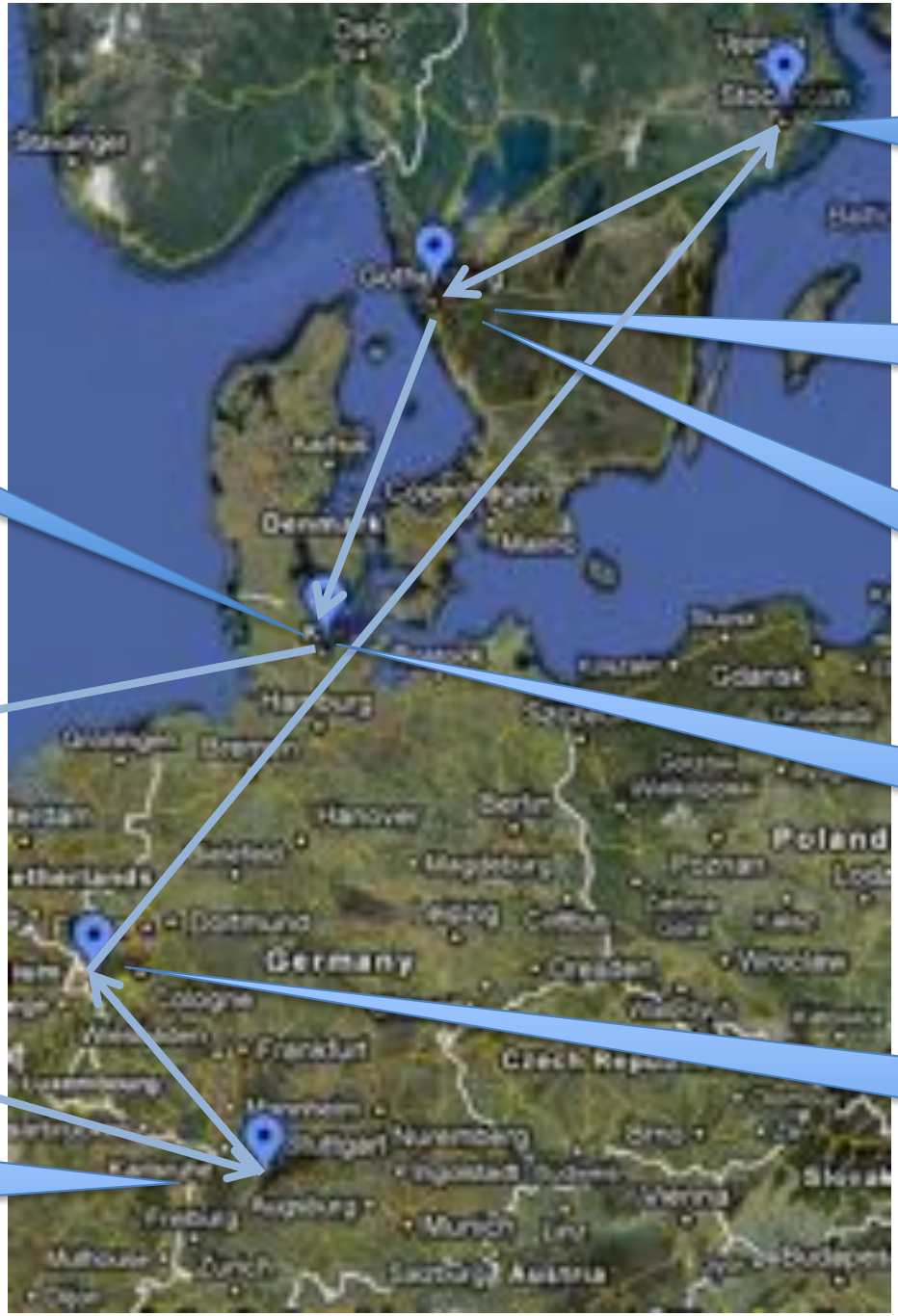
Distributed Systems
Introduction & Overview

Olaf Landsiedel

About Me: Olaf Landsiedel

- Research Group Distributed Systems
  - Since October 2018
  - Teaching elective courses
    - Advanced Computer Networks and Network Security
    - Distributed Systems
    - Wireless Networks and Internet of Things
    - Seminars, Master Projects, Master Theses

www.ds.informatik.uni-kiel.de



KTH
PostDoc
2010 - 2012

Chalmers
Assistant Prof.
2012

Chalmers
Associate Prof.
2016

CAU Kiel
Full Prof.
2018

RWTH Aachen
PhD Studies
2006 - 2010

Univ. of Tübingen
PhD Studies
2004 - 2006

Univ. of Kansas
MS Comp. Sc.
2002 - 2003

CAU Kiel
BS eq. Comp. Sc.
1999 - 2002

Team for this Course

Olaf Landsiedel

Valentin Poirot

About You

- Master
  - Computer Science?
  - Other?
- Bachelor
  - Computer Science?
  - Other?

- German speaking?

Introduction

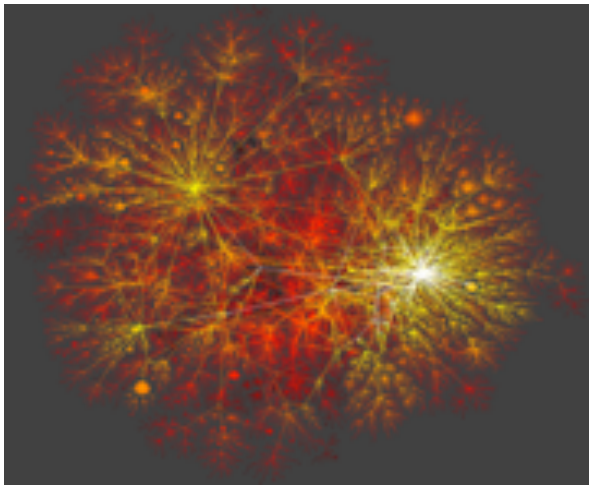
DISTRIBUTED SYSTEMS

Definition

- What is a **Distributed System?**
  - Have you ever seen one?
  - Have you ever used one?

- A Distributed System is characterized by?
  - Multiple devices
  - Connected by a network
  - Cooperating on some task

Examples



Internet

Facebook, etc.

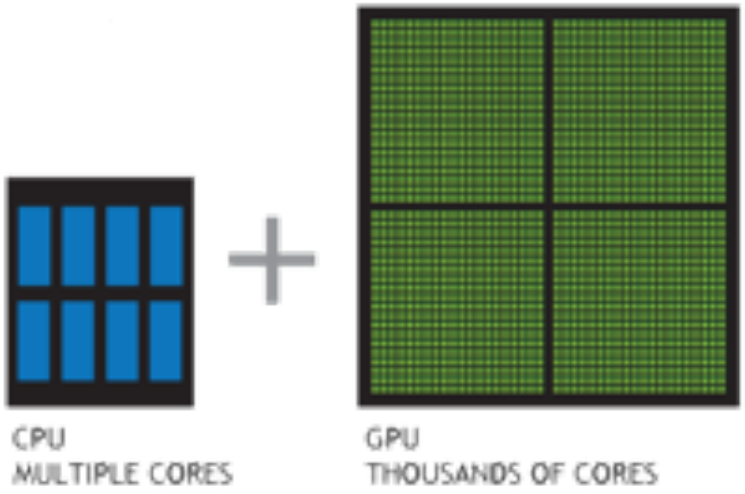
Modern Cars

Cloud Computing / Data Center

Phone Network

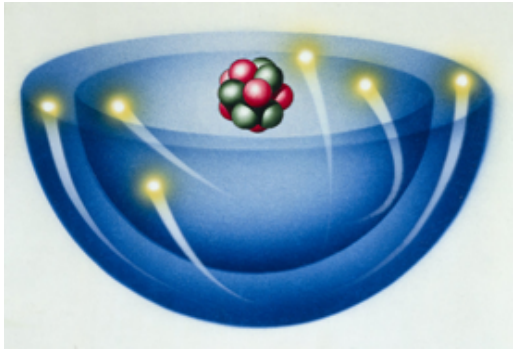
Power Grid

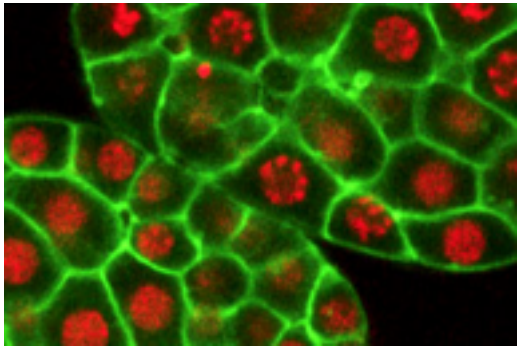
One more Example



- A modern computer is a distributed system
  - Multi-core CPU
  - Multi-core GPU
  - …
- Actually
  - Even a modern cell phone

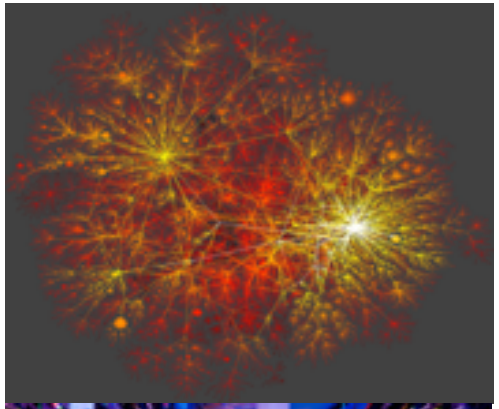
Note





- Non computer-driven "distributed systems"
  - Atoms
  - Molecules
  - Society
  - Animals (ants, bees, …)
  - …
- Not topic of this lecture

Distributed Systems vs. Networks



- Networking is worried about
  - Sending a message from here to there
  - Not what you do with the message
- Distributed Systems
  - Assume:
    There is a way to send messages
  - Focus: How you build a system using those messages
  - **We teach you what things to do with a network**

Course Goals in a Nutshell

- Lectures: Teach you Distributed Systems
  - What do they do?
  - How do they work?
- Labs: Give you hands-on experience
  - Feel the challenges
  - Master the techniques
- Have some fun!
  - Optional: you can pass this course without it

Will I learn something useful?

- We hope so!
  - This our key goal
- From an email we got from a former student
  - ”[…] I'm […] making a living out of building distributed systems, […] rest assured I've been finding the contents of your course very useful. :)”
    - Started working at Spotify
- We hope you will have a similar experience

Today: Introduction & Motivation

- Definition and Examples
- This course
  - Structure and Rules
- A bit of history
  - Why are there so many distributed systems (DSs)
- Challenges in DSs
- Course content

Your Background

- Suggested
  - Course on Computer Networks
    - TCP, UDP, IP, …
    - Most of you had such as course in their BS or MS programs
  - Course on Operating Systems or Concurrent Programming
    - Threads, locks, …
    - Most of you had such as course in their BS program
      - Or are taking the OS this period in parallel
  - Some programming experience for the labs
    - Most of you have this from their BS, too
    - We will use Python

Components in this Course

- Lectures
  - Usually Tuesdays, Thursdays,
  - Lecture notes uploaded (after class or before)
    - to iLearn
- Labs
  - Usually Thursdays
- Written Exam
  - Date?

Some notes…

- We are recording the teachers voice
  - To create screencasts for you
  - Let us know if this leads to privacy concerns

- Please interrupt me if you have questions
  - If you have a question, it is likely that others have a similar one…
  - Help yourself and your classmates and ask it

Some more notes…

- I am asking questions, too ;-)

- *Feedback & suggestions: very welcome!*
  - Talk to me after class, send me an email, …
  - Anonymous course evaluation

To Pass

- Exercises (graded with points)
  - 4 labs
  - Plus a small project in part two of the course
- Pre-Lab:
  - To get you started with Python
  - Need to pass, but does not give points
- Pass written exam, graded
  - Admission to exam: 50% of exercise points
  - Final grade: the better one of
    - exam grade or
    - 80% of the exam grade + 20% of the exercise grade

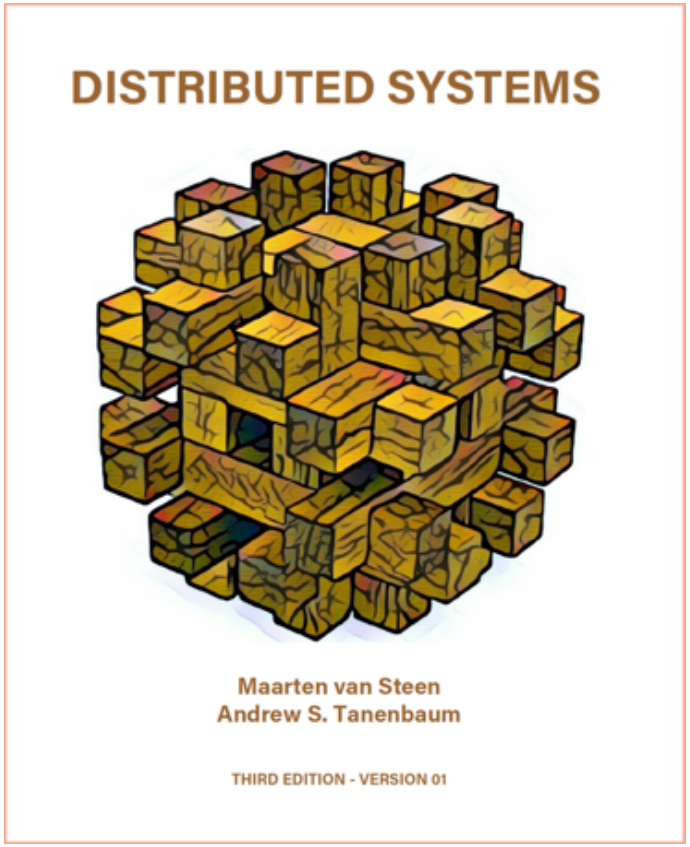
Lectures: *Please Come to Class*

- Lectures deepen and discuss
  - Difficult to do from just slides or book*: This is key!*
  - Ask questions: *Do not be shy*

- In general (statistically shown)
  - People who come to class have
    - Better grades
    - A higher probability of passing

Lectures: Book

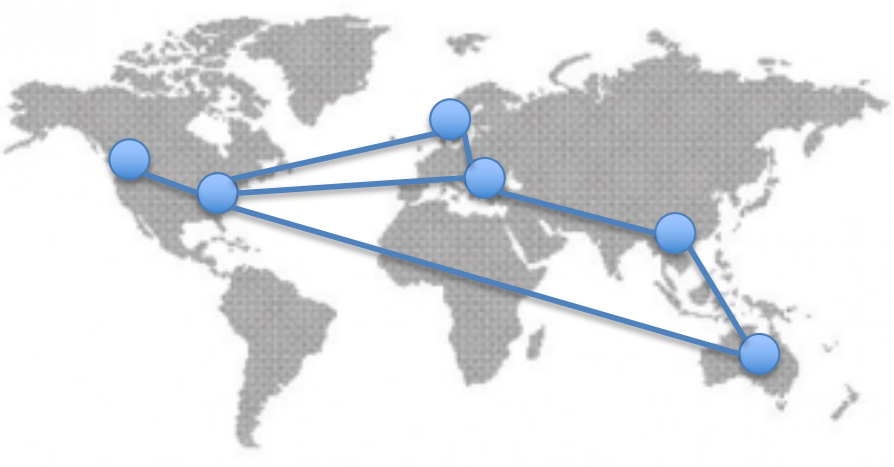
- Distributed Systems (**Third Edition**)
  - Andrew S. Tanenbaum, Marten van Steen
  - Free online copy available
    - See Link on iLearn page
  - Maybe available in the University bookstore
    - Check for cover picture and edition
    - Available through Amazon



- Nice book, well written
  - Book chapters match to lectures
  - **Do not forget:**
    - **No book (or video) can replace coming to class**

Labs

- Distributed Blackboard
  - Improve it each lab
  - In Python



- Run & Test: MiniNet
  - "Internet on your Laptop"
  - Experience
    - Slow links, dynamic links
    - …

HISTORY

History

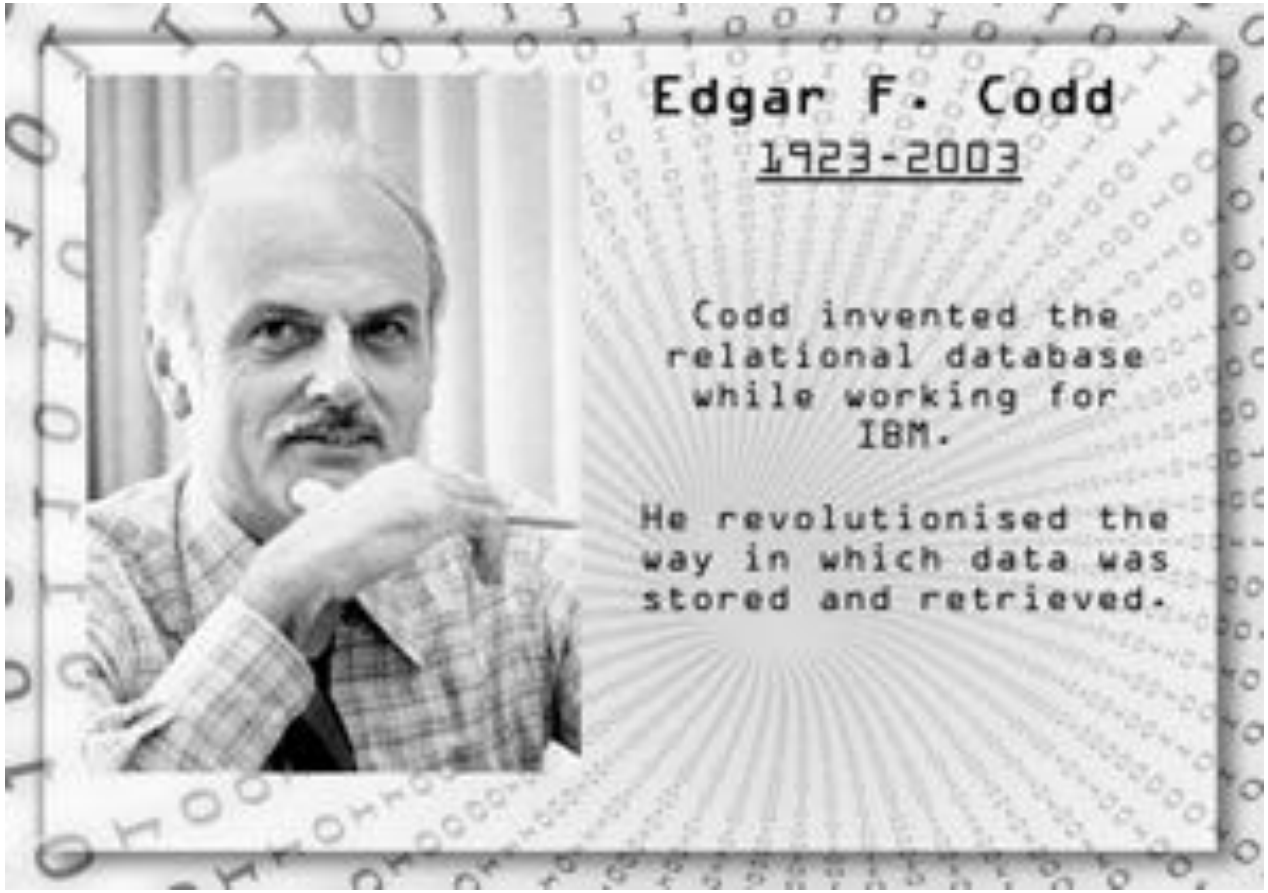
- In the examples
  - Many different distributed systems
- How did we get here
  - Where do all these DSs come from?
  - What is the trend?
    - Will their number increase even more?

1943

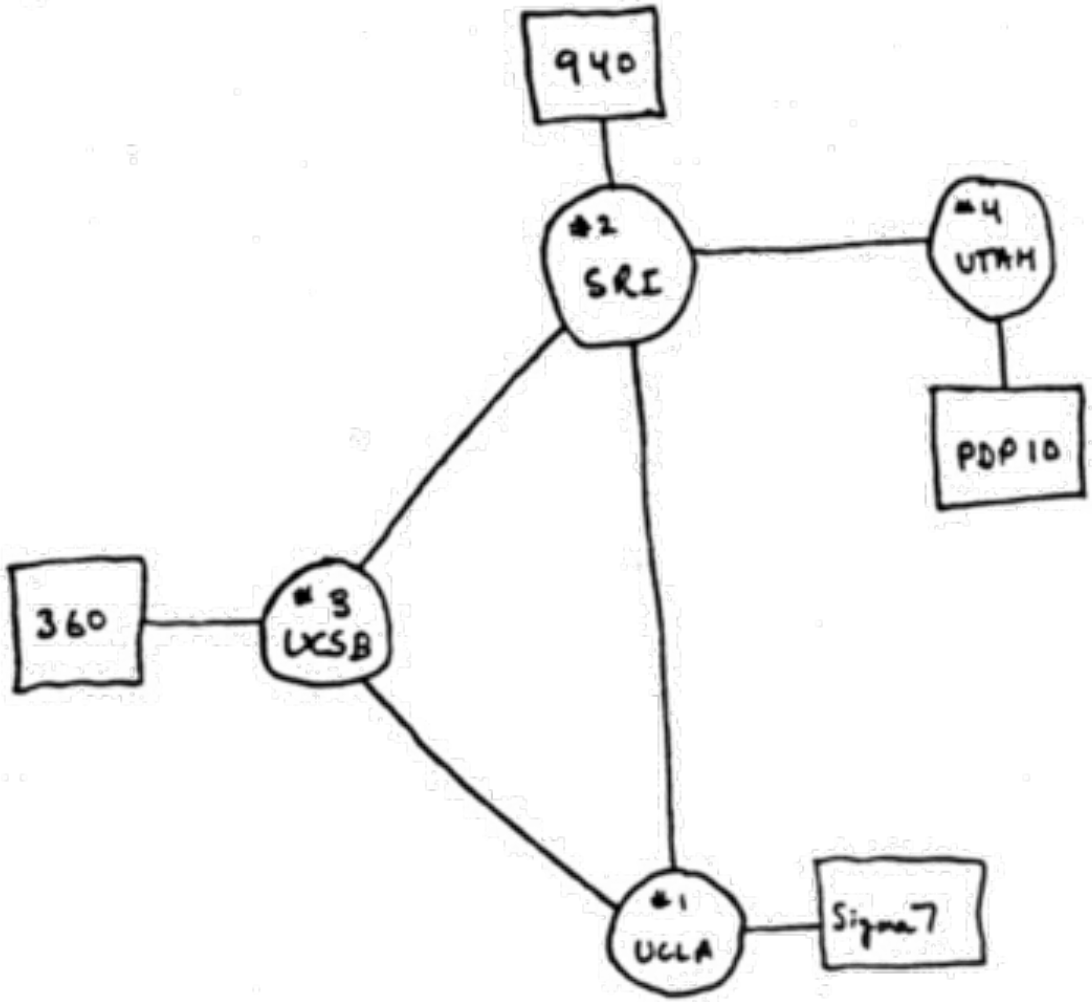
I think there is a world market for maybe five computers

Thomas J. Watson, 1943; Chairman and CEO of International Business Machines (IBM)

“In the mainframe’s early days, the 1950s, IBM sold the first era of mainframes to businesses that didn’t even know they needed them.”

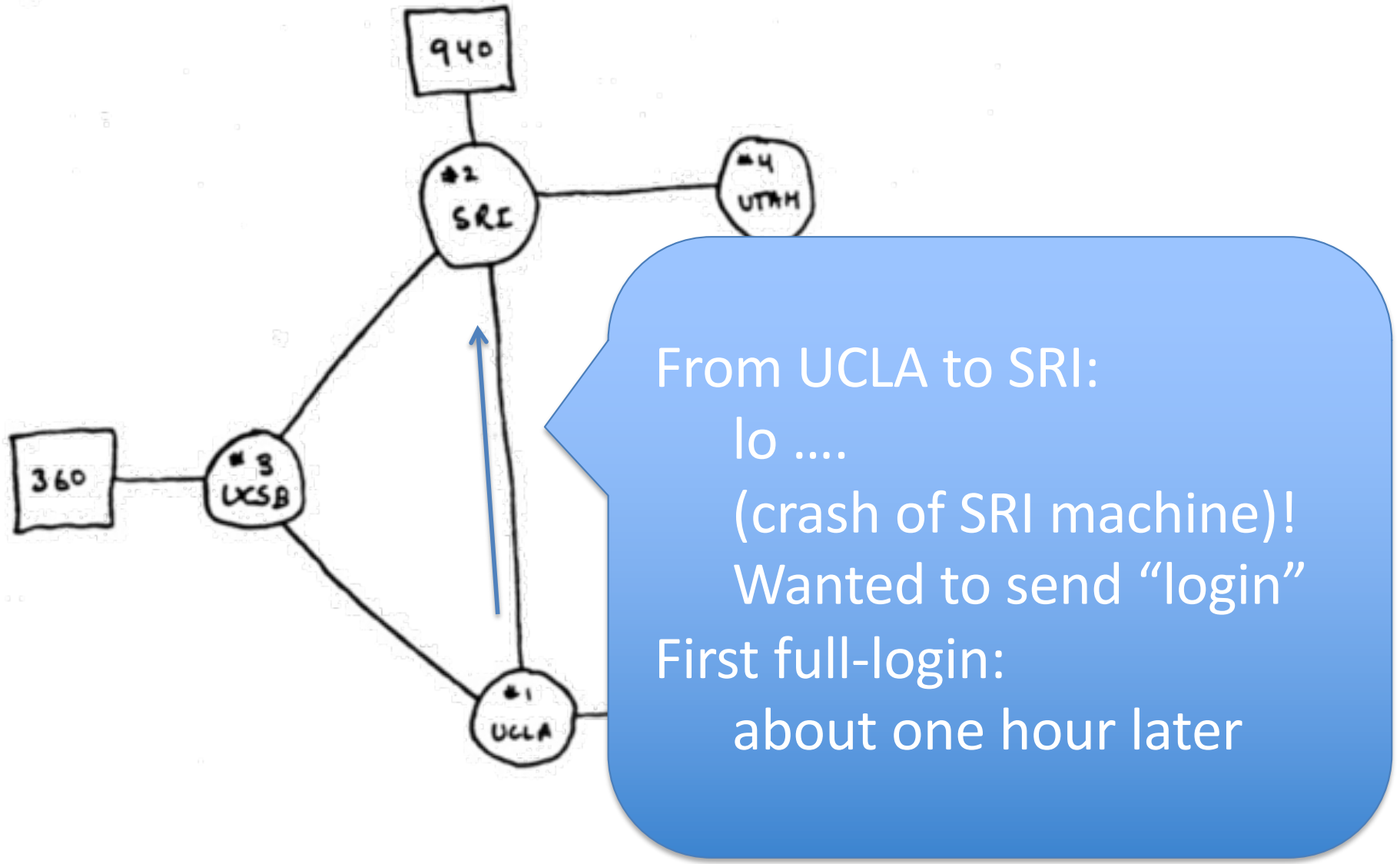

Edgar F. Codd
1923-2003
Codd invented the relational database while working for IBM.
He revolutionised the way in which data was stored and retrieved.

1969

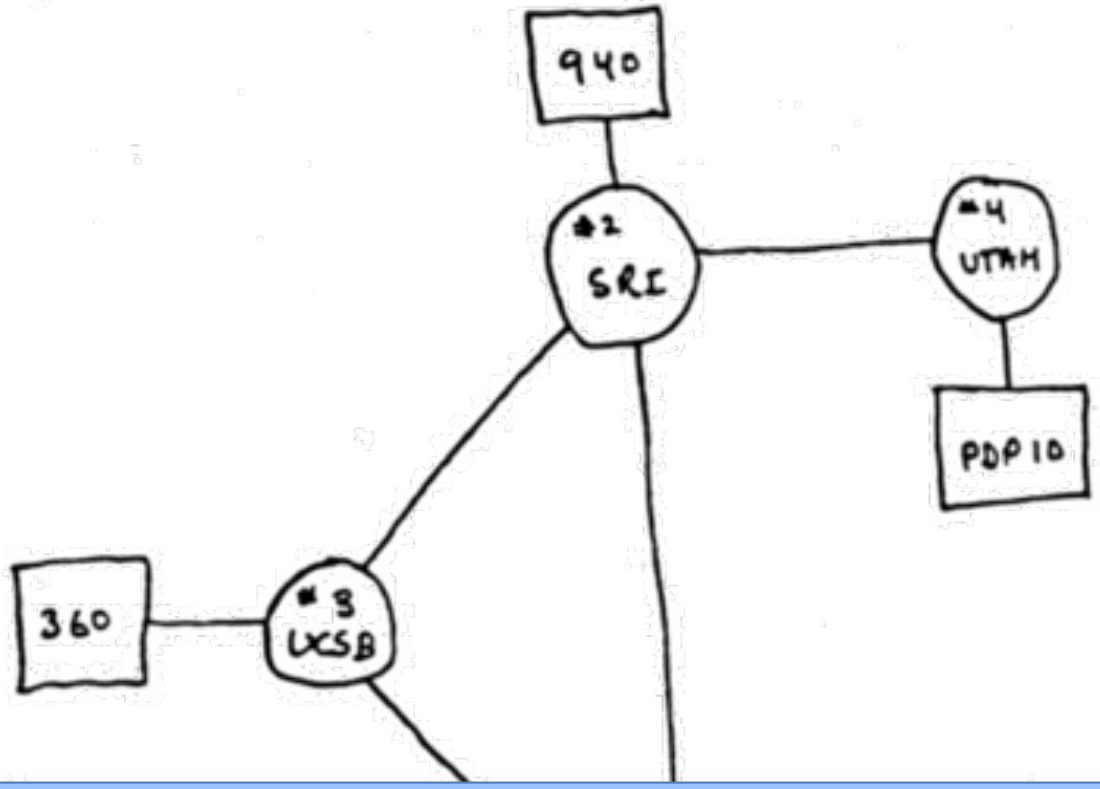


ARPANET begins…with a deployment at UCLA, Stanford, UCSB, and Utah (one computer per site)

1969, 29 Oct, 22:30:
First data on the Internet



1969, 29 Oct, 22:30:
First data on the Internet



Lessons Learned:

1. First words/letters on the Internet: “lo”
2. Not many things in the Internet work on the first try

Dezember 1969

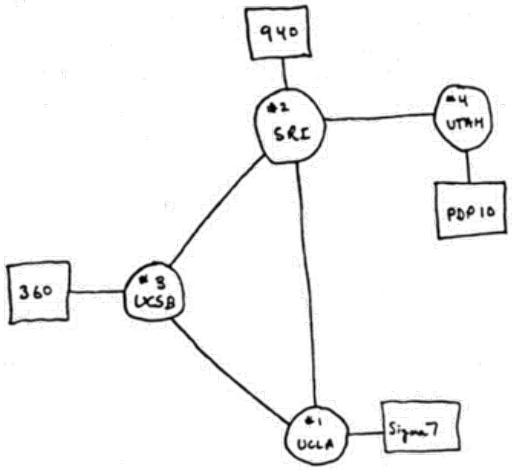
Juni 1970

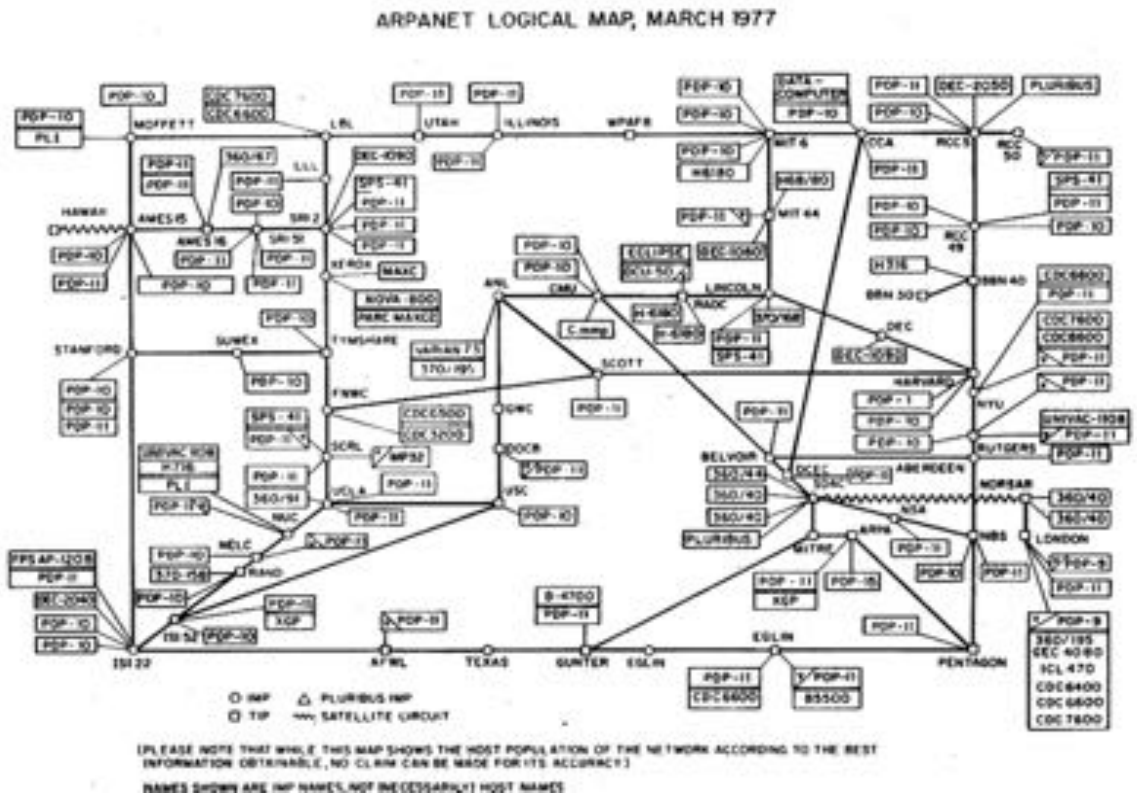
März 1972

Juli 1977

1977

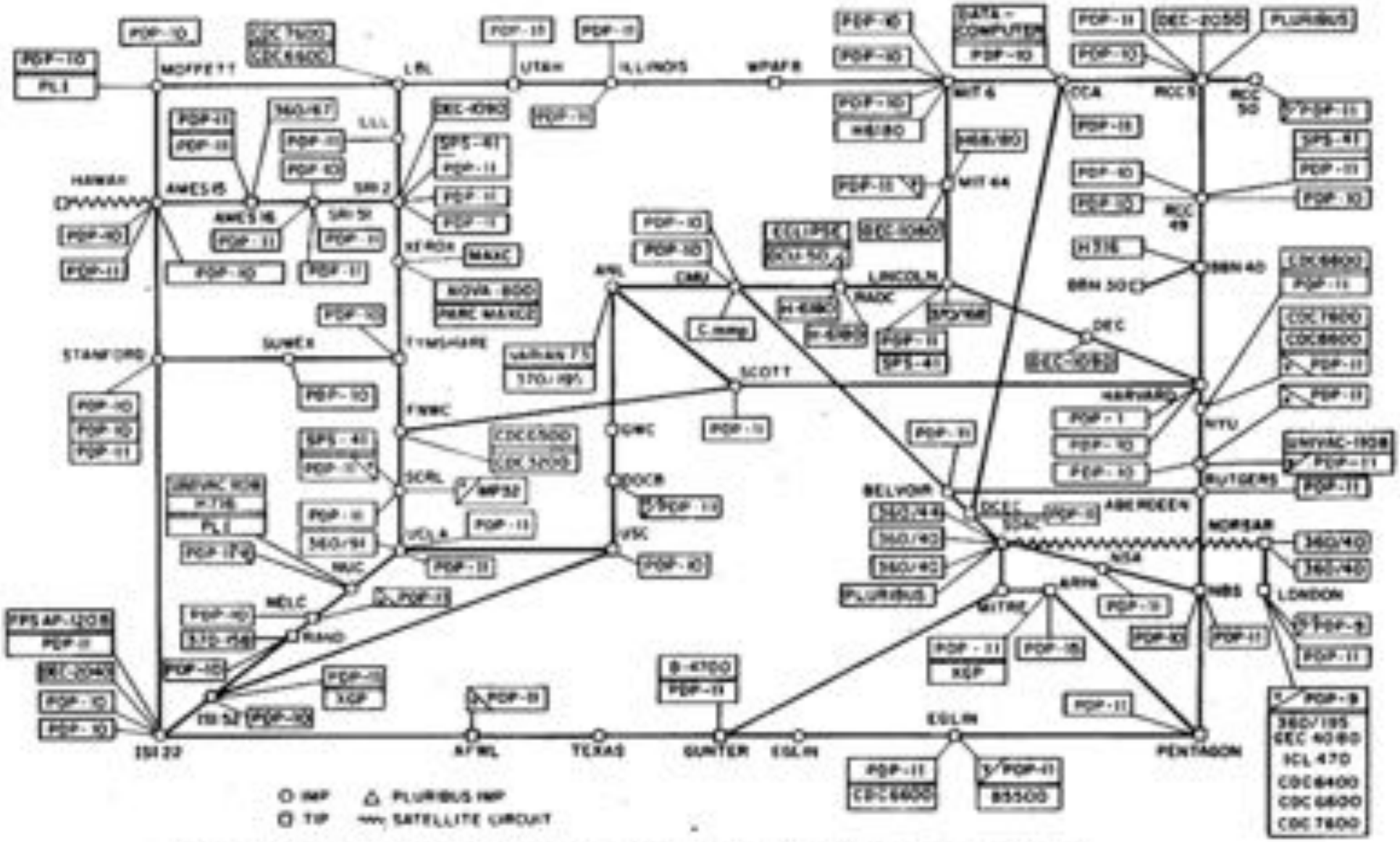
1969→





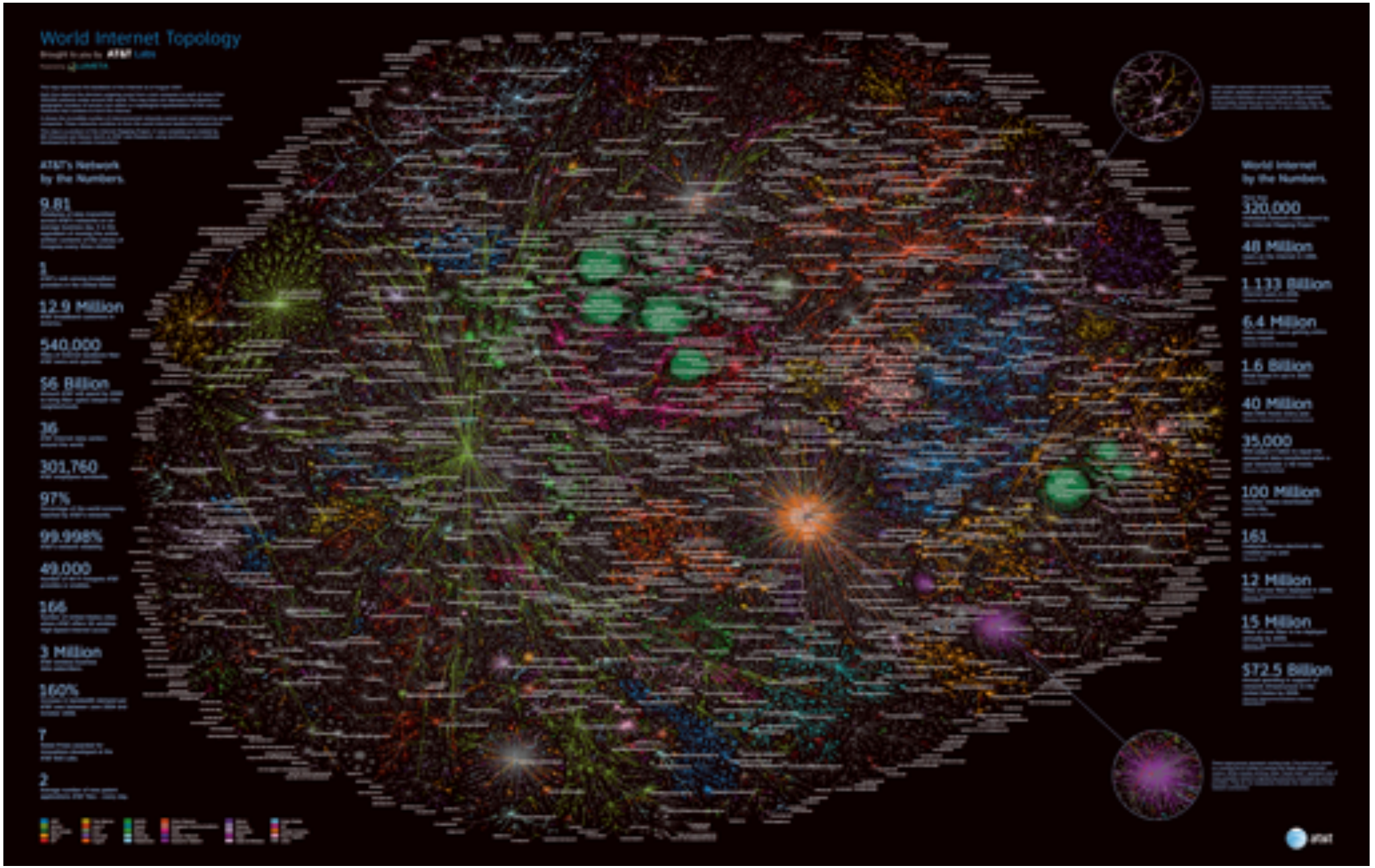
←1977

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



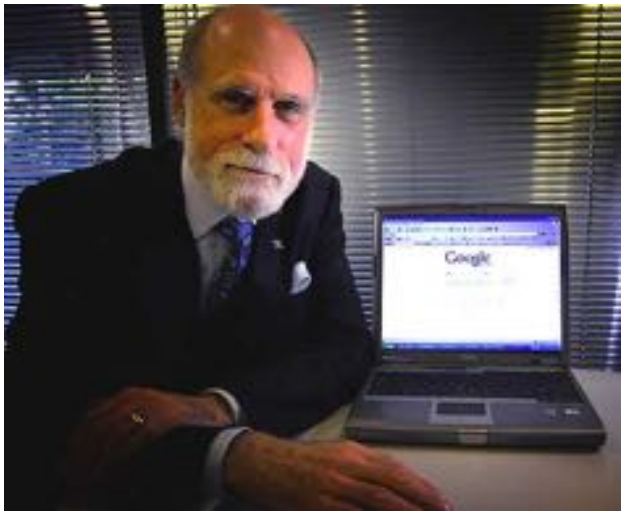
Internet 2007 (just the backbone)

www2.research.att.com/~north/news/img/ATT_Labs_InternetMap_0730_10.pdf

1971

Ray Tomlinson creates first email program

1974



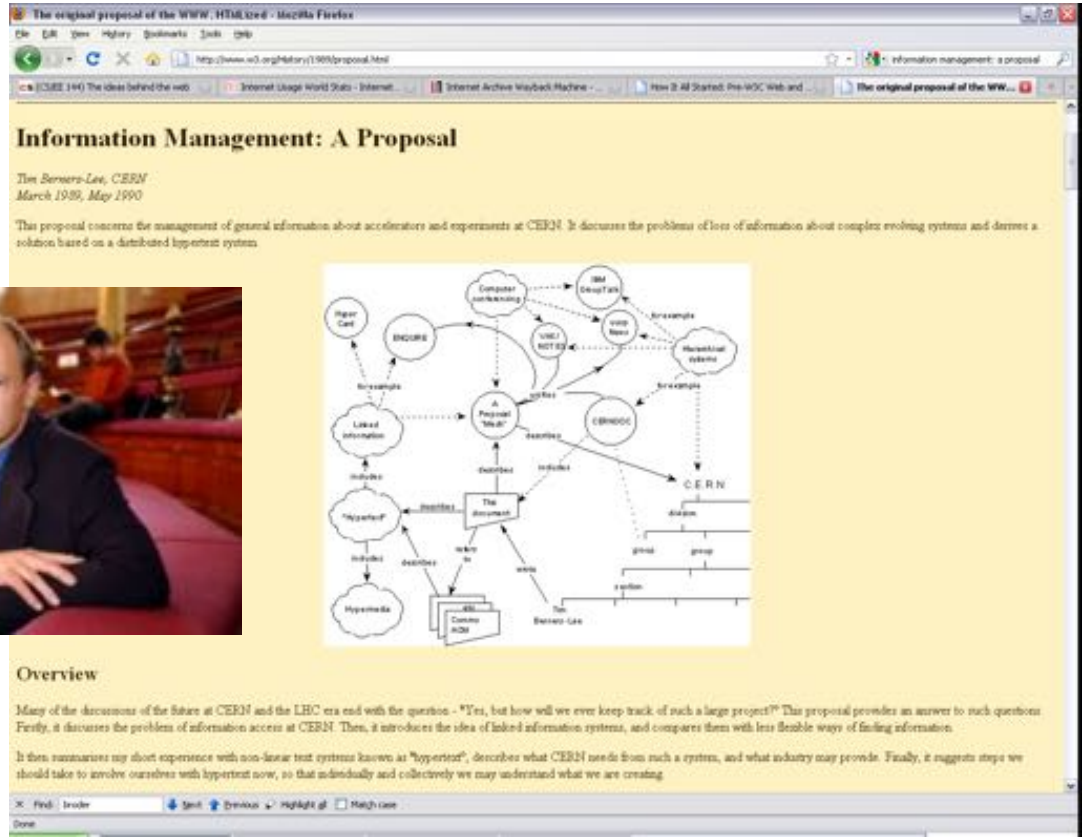
TCP / IP defined by Vint Cerf & Bob Kahn

2004: both received the Turing Award

1984

Paul Mockapetris introduces DNS

1989 – The Web Emerges



Information Management: A Proposal

Tim Berners-Lee, CERN
March 1989, May 1990

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

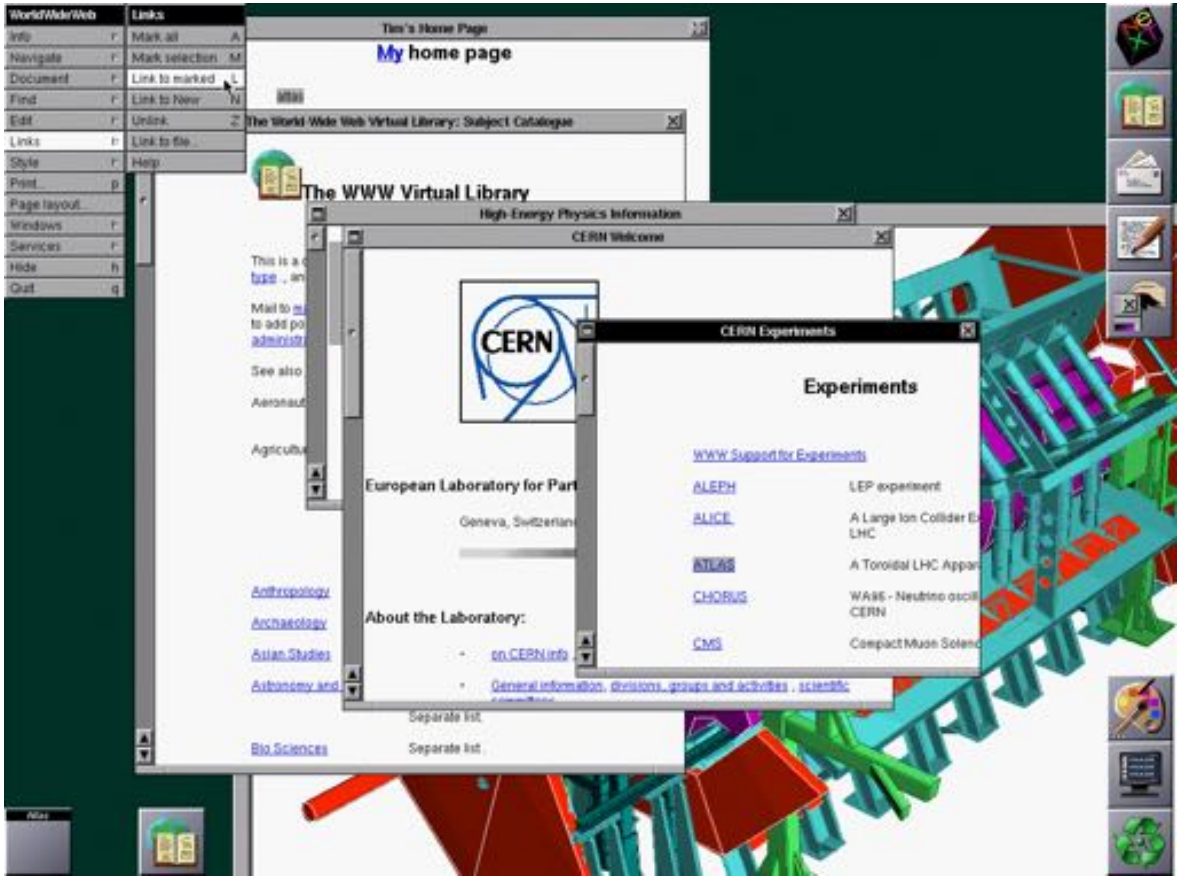
Overview

Many of the discussions of the future at CERN and the LHC era end with the question - "Yes, but how will we ever keep track of such a large project?" This proposal provides an answer to such questions. Firstly, it discusses the problems of information access at CERN. Then, it introduces the idea of linked information systems, and compares them with less flexible ways of finding information.

It then summarizes my short experience with non-linear text systems known as "hypertext", describes what CERN needs from such a system, and what industry may provide. Finally, it suggests steps we should take to involve ourselves with hypertext now, so that individually and collectively we may understand what we are creating.

Tim Berners-Lee writes “Information Management: A proposal” at CERN

1990



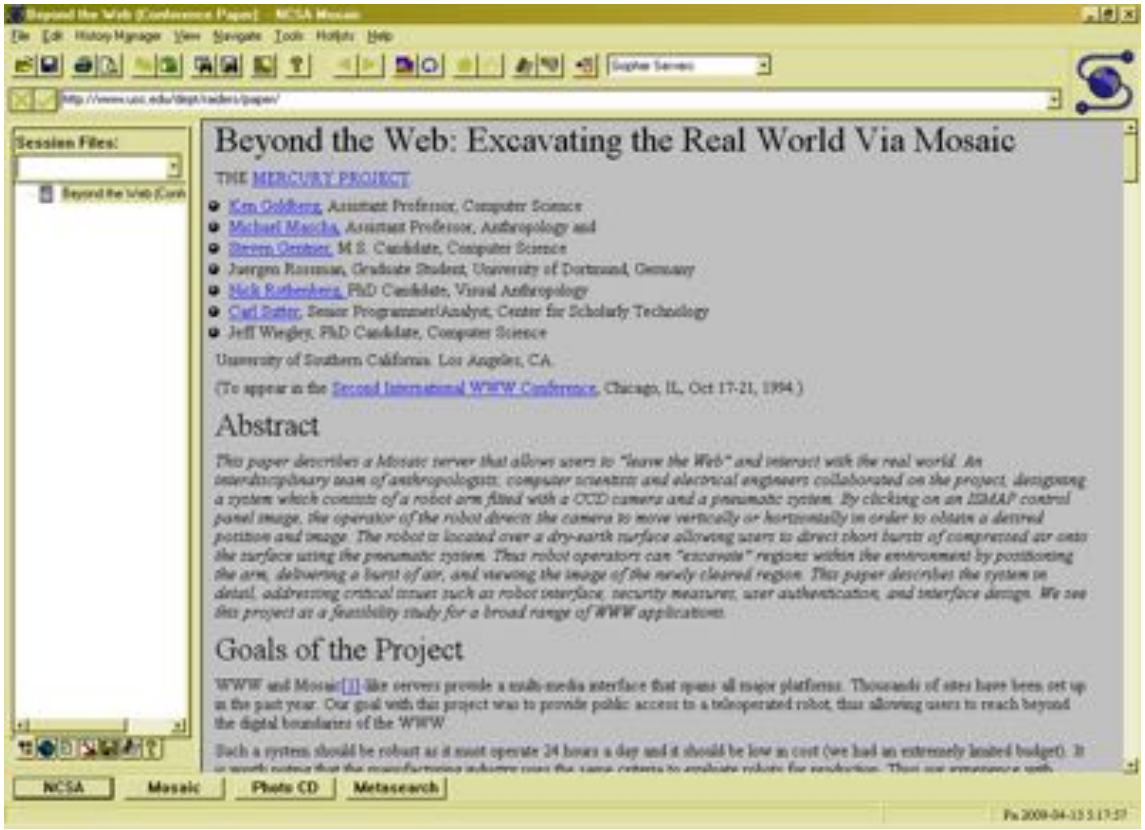
First browser developed at CERN

1991

First paper appears on the project at Hypertext conference

→ Only accepted as a poster!

1993



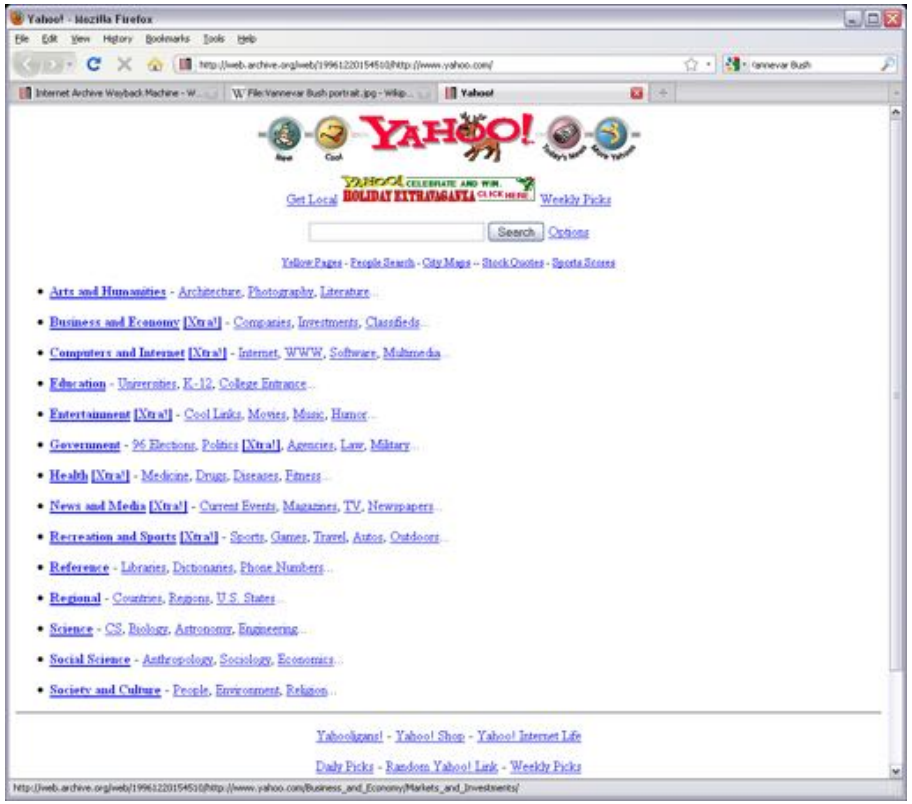
Mosaic became the first graphical browser

CERN agrees to allow public use of web protocol royalty-free!

1994

→ Mosaic goes commercial
(later becomes Netscape)

→ Traditional dialups (AOL, CompuServe, Prodigy) begin to sell Internet access.



Yahoo
circa
1996

"Jerry's Guide to the world wide web" started …
it eventually became Yahoo

1994

→ Mosaic goes commercial
(later becomes Netscape)

→ Traditional dialups (AOL, CompuServe, Prodigy) begin to sell Internet access.

“America Online (AOL)” start page …

1995+

Amazon arrives and the commercialization of the web begins

Amazon circa 1999

Today

Today

- How many connected devices do you have?

- Many!
  - Desktop
  - Laptop
  - (Smart)phone
  - Tablet
  - TV / gaming console
  - …

http://olafland.polldaddy.com/s/connected-devices

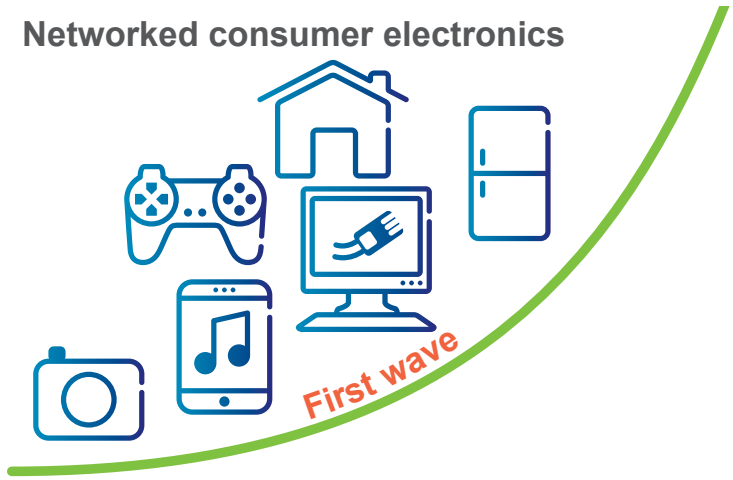
Summary: A bit of History

Mainframe age (60's & 70's):
One computer for many

PC age (80's & 90's):
One computer for each, partially networked

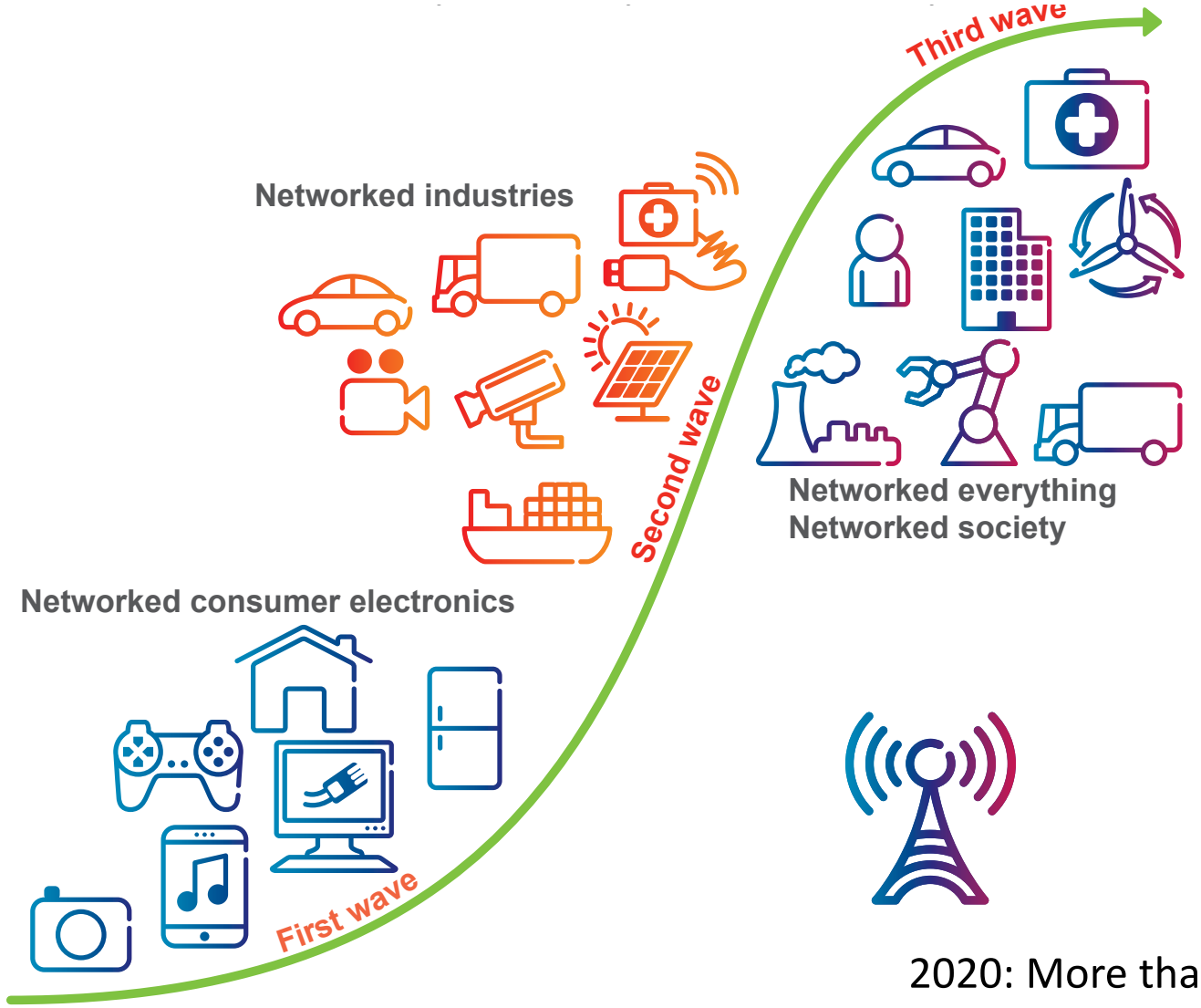
Cloud computing
Mobile, ubiquitous computir
(Today, > 2000):
Many computers for each, networked

Tomorrow?



Today

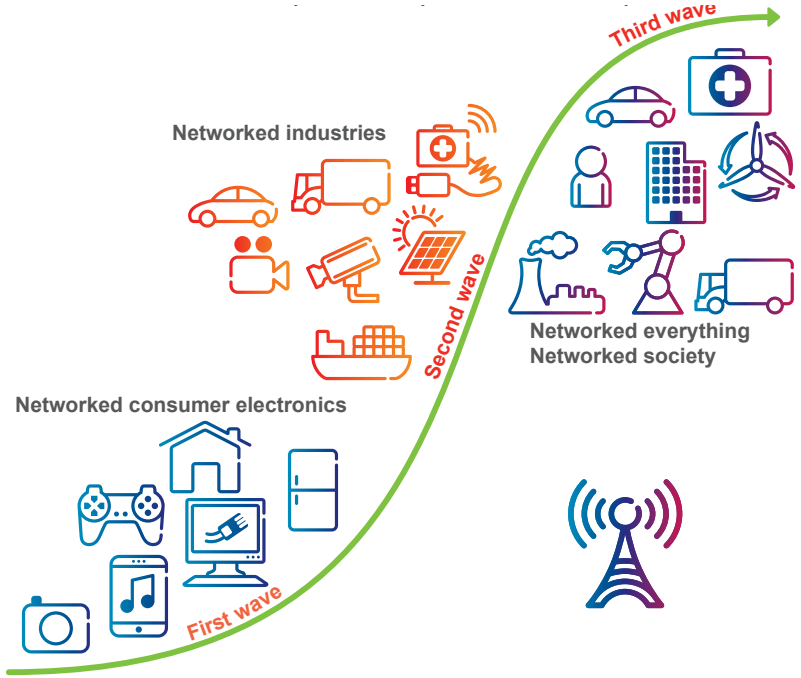
Tomorrow?



Today

2020: More than 50 Billion connected devices (Ericsson)

Tomorrow? Networked Society!



- Distributed Systems touch all aspects of daily life!
  - Integral building block for our networked society
  - Strongly increasing in numbers
  - Result: Very good topic to study ;-)

COURSE TOPICS

Course Topics: Motivation

- Assume: your task is to build
  - Facebook or
  - Amazon or
  - just a simple web application

- What challenges do you face?

The Eight Fallacies of Distributed Systems

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

Mechanisms

- This course
  - Mechanisms to deal with these challenges
  - Generic mechanisms
    - Not bound to the Internet
    - But: Examples mostly Internet bound
      - Easier to understand for most students
      - Compared to power grids, cars, …

Course Content: Mechanisms

- Architectures & Processes
- Mutual exclusion & Election
- Naming
- Clocks and Time
- Consistency & replication
- Fault tolerance

Course Structure

- *Intro / background*
  - *Overview, Communication*
- Architectures & Processes
- Mutual exclusion & Election
- Naming
- Clocks
- Consistency and replication
- Fault tolerance
- *Applications*
  - *Selected examples, incl. Bitcoin and Blockchain, Google Map Reduce, Amazon Dynamo, …*

Architectures & Processes

- High level overview

- How can we build distributed systems?
  - Key building blocks
  - Different approaches

Mutual Exclusion

- Grant and share access to resources
- Example: online store
  - Each item shall only be sold once
- Centralized and distributed approaches
  - Complexity?

Election

- Elect a leader
  - All nodes need to agree
  - Example: Elect resource manager
- Approaches
  - Wired and wireless networks
  - Small or large networks
  - Challenges: complexity, scalability

Naming

- Uniquely identify a resource
  - Computer, file, IP address, physical address, …
- Name lookup
  - Mechanism to map names
    - Domain Name System (DNS), phone book, map
  - www.google.com -> 74.125.224.72
  - Olaf Landsiedel -> Hermann-Rodewald-Str. 3, 24118 Kiel, Germany

Clocks and Time

- Why do we need clocks and time?
  - Example: to order events
    - Locally, globally
    - Facebook timeline
- Types of clocks
  - Physical Clocks
  - Logical Clocks
- Clock synchronization

Consistency and Replication

- Consistency:
  - All nodes shall have the same information
  - Difficult in distributed systems
    - Delay, packet loss, …
  - Mechanisms, complexity, tradeoffs
- Replication
  - Distribute “copies” among devices
  - Ensure consistency among them

Fault Tolerance

- Deal with faults
  - What can go wrong?
  - Expect faults
  - Understand impact
- Build systems that can deal with faults

Applications

- Google Map Reduce
- Google File System
- Amazon Dynamo
- TOR
- BitTorrent
- Bitcoin,
- Blockchain
- Smart contracts
- …

Not Part of this Course

- Computer Networks
  - We assume that there is network that can deliver packets from A to B
    - But not always reliably
    - With potential delays
    - … (see fallacies)

In Other Words

- Computer Networks Course
  - How? (the Internet / networks work)

- This Course
  - Why?
    - Why the Internet is built in the way it is
    - Why Facebook etc. are built the way they are
    - Mechanisms and Concepts

What is where? TODOs!

- iLearn (Please register to course)
  - Slides, exercise groups, videos, …
- Studi DB (Please register to course)
  - Official enrollment to the course, required for exam
- Module DB
  - Syllabus
- Univis
  - Rooms and dates
- www.ds.informatik.uni-kiel.de
  - Website of the research group
  - Email address etc. of us

Next Time: Communication

- Revisit the Internet architecture
- Focus on the why and not the how
  - We assume you know basic Internet protocols
    - TCP, IP, ..
  - Discuss why the Internet is built the way it is

Next Next Time: Lab Introduction
(Lab slot, Thursday afternoon)

- Introduce you to
  - MiniNet
  - Lab Tasks
  - Template for the Labs

- By the TA
- Lab 1 will be released after Lab introduction
- PreLab will be released today

Questions?

In part, inspired from / based on slides from Andrew Tanenbaum and many others