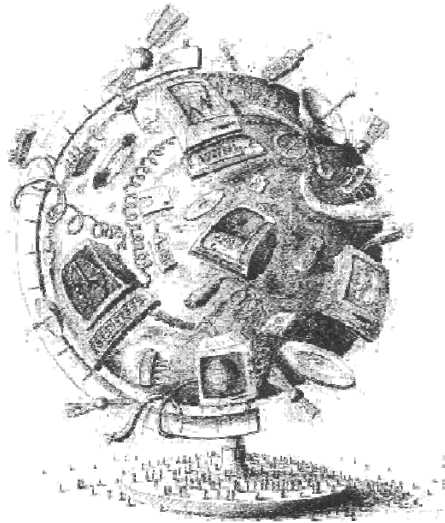


Verteilte Systeme

*Friedemann
Mattern*

ETH Zürich



Prüfungsrelevant ist der Inhalt der Vorlesung, nicht alleine der Text dieser Foliensammlung!

ETH Eidgenössische
Technische Hochschule
Zürich

© F. Mattern, HS 2012

Wer bin ich? Wer sind wir?



Prof. **Friedemann Mattern**

+ mehrere Assistenten
und Assistentinnen



Fachgebiet „Verteilte Systeme“
im Departement Informatik,
Institut für Pervasive Computing

Wer bin ich? Wer sind wir?



Prof. **Friedemann Mattern**

+ mehrere Assistenten
und Assistentinnen

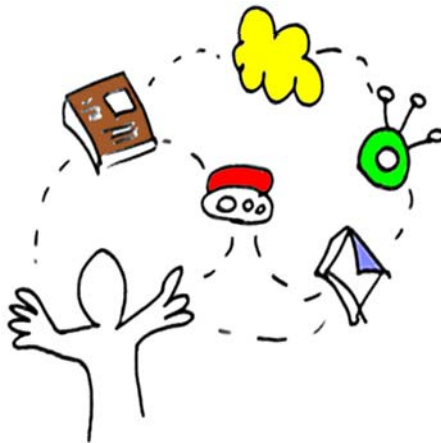
Matthias Kovatsch

Ansprechperson für or-
ganisatorische Aspekte
kovatsch@inf.ethz.ch



Fachgebiet „Verteilte Systeme“
im Departement Informatik,
Institut für Pervasive Computing

Mit was beschäftigen wir uns?



- Infrastruktur für verteilte Systeme
- Internet der Dinge
- Ubiquitous Computing
- Sensornetze
- Smart energy
- Verteilte Anwendungen und Algorithmen

Mehr zu uns:
www.vs.inf.ethz.ch

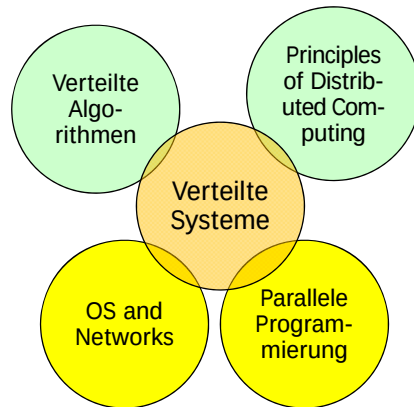
Organisatorisches zur Vorlesung

- Format: 6G+1A: **Vorlesung** und **Praktikum** integriert
 - Mo 9:15 - 12:00, Fr 9:15 - 12:00, jew. NO C 6
 - Praktikum ist inhaltlich *komplementär zur Vorlesung* (mobile Kommunikationsplattformen: Android, HTC Desire etc.)
 - Gelegentliche *Assistentenstunden* (zu den "Vorlesungsterminen") zur Besprechung der Praktikumsaufgaben und Vertiefung des Stoffes
 - Gelegentliche *Denkaufgaben* (ohne Lösung...) in der Vorlesung
- Sinnvolle **Vorkenntnisse** (Grundlagen)
 - 4 Semester der Bachelorstufe Informatik
 - Grundkenntnisse Computernetze und Betriebssysteme (z.B. Prozessbegriff, Synchronisation)
 - UNIX, Java ist hilfreich

Organisatorisches (2)

- **Folienkopien** jeweils einige Tage nach der Vorlesung
 - im pdf-Format bei www.vs.inf.ethz.ch/edu
- **Prüfung** schriftlich
 - bewertete Praktikumsaufgaben gehen in die Prüfungsnote ein
- Vorlesung 15.10. – 30.11.2012: Prof. **Roger Wattenhofer**

Einordnung der Vorlesung

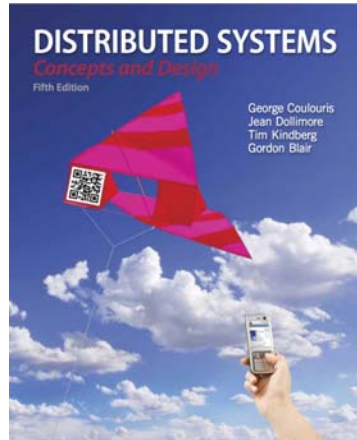


- „Verteilte Systeme“ ist ein **Querschnittsthema**
- Gewisse Überschneidungen mit anderen Vorlesungen unvermeidlich

Thematisch verwandte Veranstaltungen (Masterstufe)

- Ubiquitous Computing
 - Enterprise Application Integration - Middleware
 - Web Services and Service Oriented Architectures
 - Verteilte Algorithmen
 - Principles of Distributed Computing
-
- Einschlägige Seminare
 - Praktikum (“Lab”)
 - Semester- und Masterarbeit

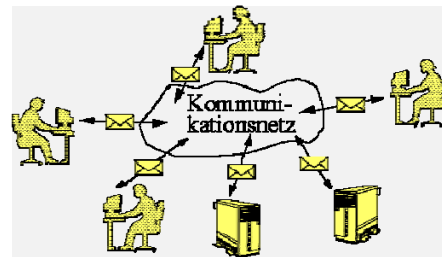
Literatur



- [G. Coulouris, J. Dollimore, T. Kindberg, G. Blair](#): Distributed Systems: Concepts and Design (5th ed.). Addison-Wesley, 2011
- [A. Tanenbaum, M. van Steen](#): Distributed Systems: Principles and Paradigms (2nd ed.). Prentice-Hall, 2007
- [O. Haase](#): Kommunikation in verteilten Anwendungen (2. Aufl.). R. Oldenbourg Verlag, 2008
- [A. Schill, T. Springer](#): Verteilte Systeme (2. Aufl.). Springer Vieweg, 2011

„Verteiltes System“ – zwei Definitionen

- A distributed computing system consists of multiple autonomous processors that do not share primary memory, but cooperate by sending messages over a communication network.
-- *H. Bal*

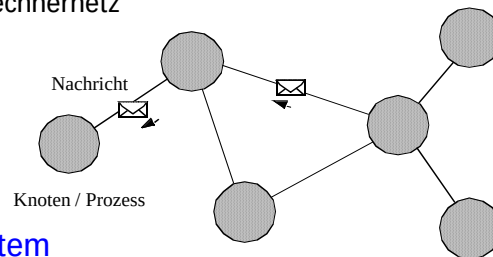


- A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable. -- *Leslie Lamport*
 - welche Problemaspekte stecken hinter Lamports Charakterisierung?

„Verteiltes System“

- Physisch verteiltes System

- Compute-Cluster ... Rechnernetz

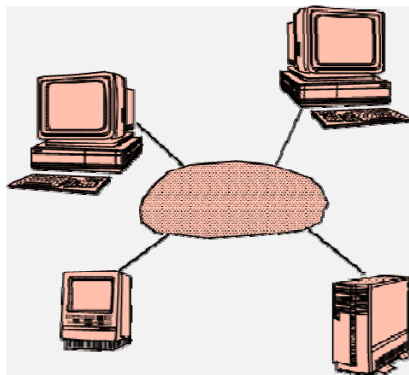


- Logisch verteiltes System

- Prozesse (Objekte, Agenten)
- Verteilung des Zustandes (keine globale Sicht)
- Keine gemeinsame Zeit (globale, genaue Uhr)

13

Sichten verteilter Systeme (1)



- Computernetz mit "Rechenknoten", z.B.

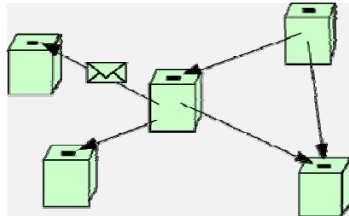
- Compute-Cluster
- Local Area Network
- Internet

- Relevante Aspekte:

- Routing, Adressierung,...

Zunehmende Abstraktion →

Sichten verteilter Systeme (2)

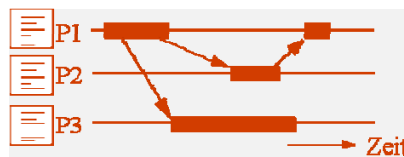


kommunizierende Prozesse,
kooperierende Objekte

- **Objekte** in Betriebssystemen, Middleware, Programmiersprachen
- “**Programmierersicht**”
 - z.B. Client mit API zu Server

Zunehmende Abstraktion →

Sichten verteilter Systeme (3)



- **Algorithmen- und Protokollebene**
 - Aktionen, Ereignisfolgen
 - Konsistenz, Korrektheit

- Man kann verteilte Systeme auf **verschiedenen Abstraktionsstufen** betrachten
- Es sind dabei jeweils **unterschiedliche Aspekte** relevant und interessant

Die verteilte Welt



Auch die “reale Welt” ist ein **verteiltes System**:

- viele gleichzeitige (“**parallele**”) Aktivitäten
- exakte globale **Zeit** nicht erfahrbar / vorhanden
- keine konsistente Sicht des **Gesamtzustandes**
- Kooperation durch explizite **Kommunikation**
- **Ursache** und **Wirkung** zeitlich (und räumlich) getrennt

Warum verteilte Systeme?

- **Es gibt inhärent geographisch verteilte physische Systeme**
 - z.B. Steuerung einer Fabrik, Zweigstellennetz einer Bank (Zusammenführen / Verteilen von Information)
- **Electronic commerce**
 - kooperative Informationsverarbeitung räumlich getrennter Institutionen (z.B. Reisebüros, Kreditkarten,...)
- **Mensch-Mensch-Telekommunikation**
 - E-Mail, Diskussionsforen, Blogs, digitale soz. Netze, IP-Telefonie,...
- **Globalisierung von Diensten**
 - Skaleneffekte, Outsourcing,...

Wirtschaftliche Aspekte

- Outsourcing von Diensten, Verlagerung in eine „Cloud“, kann günstiger sein als eine lokal-zentralisierte Lösung
- Compute-Cluster manchmal besseres Preis-Leistungs-verhältnis als Supercomputer

Verteilte Systeme als „Verbunde“

- Verteilte Systeme **verbinden** räumlich (oder logisch) getrennte Komponenten zu einem bestimmten **Zweck**
-
- | | |
|--|--|
| <ul style="list-style-type: none">▪ Systemverbund<ul style="list-style-type: none">▪ gemeinsame Nutzung von Betriebsmitteln, Geräten,...▪ einfache inkrementelle Erweiterbarkeit▪ Funktionsverbund<ul style="list-style-type: none">▪ Kooperation bzgl. Nutzung jeweils spezifischer Eigenschaften | <ul style="list-style-type: none">▪ Lastverbund<ul style="list-style-type: none">▪ Zusammenfassung der Kapazitäten▪ Datenverbund<ul style="list-style-type: none">▪ allgemeine Bereitstellung von Daten▪ Überlebensverbund<ul style="list-style-type: none">▪ Redundanz durch Replikation |
|--|--|

Historische Entwicklung (1)

- **Rechner-zu-Rechner-Kommunikation**
 - Zugriff auf entfernte Daten (“Datenfernübertragung”, DFÜ)
 - dezentrale Informationsverarbeitung war zunächst ökonomisch nicht sinnvoll (zu teuer, Fachpersonal nötig)
 - Master-Slave-Beziehung (“Remote Job Entry”, Terminals)
- **ARPA-Netz (Prototyp des Internet)**
 - “symmetrische” Kommunikationsbeziehung (“peer to peer”)
 - file transfer, remote login, E-Mail
 - Internet-Protokollfamilie (TCP/IP,...)

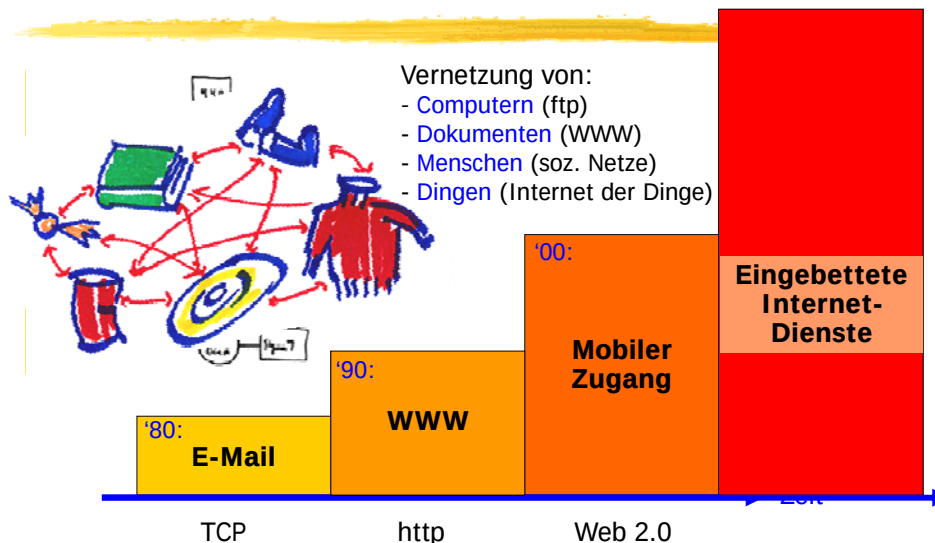
20

Historische Entwicklung (2)

- **Workstation- und PC-Netze (LAN)**
 - bahnbrechende, frühe Ideen bei XEROX-PARC (XEROX-Star als erste Workstation, Desktop-Benutzerinterface, Ethernet, RPC, verteilte Dateisysteme,...)
- **Kommerzielle Pionierprojekte als Treiber**
 - z.B. Reservierungssysteme, Banken, Kreditkarten
- **Web / Internet als Plattform**
 - für electronic commerce etc.
 - web services
 - neue, darauf aufbauende Dienste
- **Mobile Geräte**
 - z.B. Smartphones
- **Internet der Dinge**

21

Änderung der Vernetzungsqualität aus historischer Sicht



Historie von Konzepten

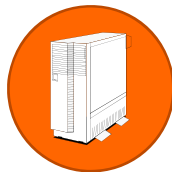
- **Concurrency, Synchronisation**
 - war bereits klassisches Thema bei Datenbanken und Betriebssystemen
- **Programmiersprachen**
 - „kommunizierende“ Objekte
- **Parallele und verteilte Algorithmen**
- **Semantik von Kooperation / Kommunikation**
 - mathematische Modelle für Verteiltheit (z.B. CCS, Petri-Netze)
- **Abstraktionsprinzipien**
 - Schichten, Dienstprimitive,...
- **Verständnis grundlegender Phänomene der Verteiltheit**
 - Konsistenz, Zeit, Zustand,...

Entwicklung “guter” Konzepte, Modelle, Abstraktionen etc. zum Verständnis der Phänomene **dauert oft lange** (notwendige Ordnung und Sichtung des verfügbaren Gedankenguts)

Diese sind jedoch für die Lösung praktischer Probleme **hilfreich**, oft sogar **notwendig!**

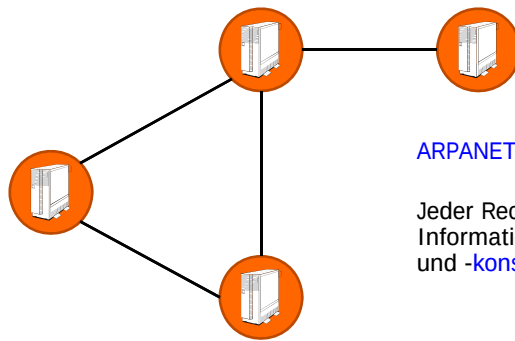
Architekturen verteilter Systeme

- Zu Anfang waren Systeme **monolithisch**



- Nicht verteilt / vernetzt
- **Mainframes**
- **Terminals** als angeschlossene „Datensichtgeräte“ („Datenendgerät“: Fernschreiber [tty], ASCII)

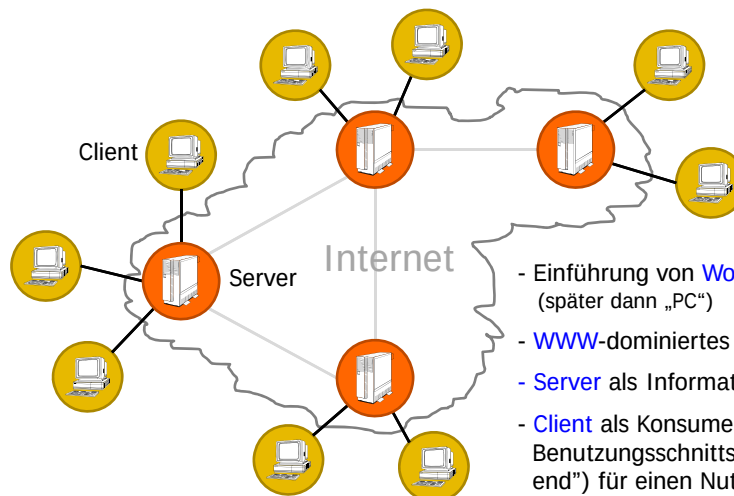
Architekturen verteilter Systeme: Peer-to-Peer



ARPANET 1969

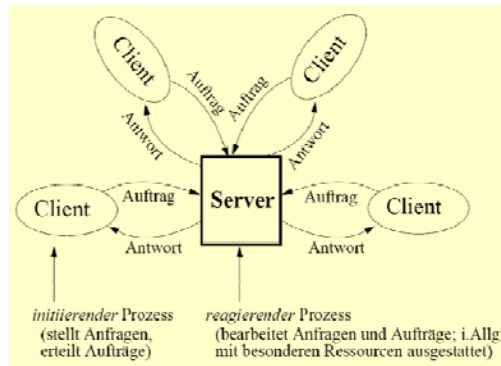
Jeder Rechner **gleichzeitig**
Informations**anbieter**
und **-konsument**

Architekturen verteilter Systeme: Client-Server



- Einführung von **Workstations** (später dann „PC“)
- **WWW**-dominiertes Internet
- **Server** als Informationsanbieter
- **Client** als Konsument; gleichzeitig Benutzungsschnittstelle („front end“) für einen Nutzer

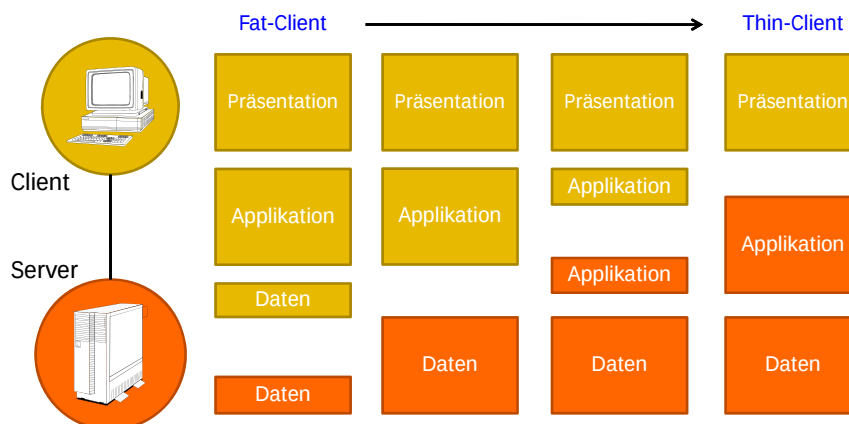
Client-Server



Populär auch wegen des eingängigen Modells

- entspricht Geschäftsvorgängen in unserer Dienstleistungsgesellschaft
- gewohntes Muster → intuitive Struktur, gute Überschaubarkeit

Architekturen verteilter Systeme: Fat- und Thin-Client



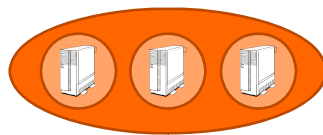
Architekturen verteilter Systeme: 3-Tier



Architekturen verteilter Systeme: Multi-Tier



Architekturen verteilter Systeme: **Compute-Cluster**



Vernetzung kompletter Einzelrechner

Räumlich konzentriert (wenige Meter)

Sehr schnelles Verbindungsnetz

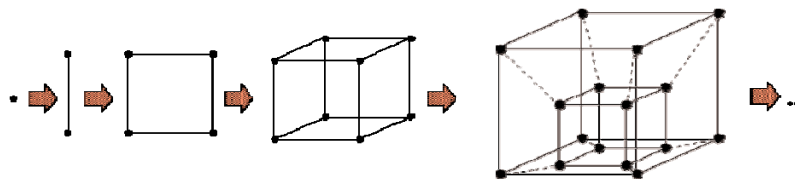
Es gibt diverse **Netztopologien**, um die Einzelrechner (*als Knoten in einem Graphen*) miteinander zu verbinden – diese sind unterschiedlich hinsichtlich

- Skalierbarkeit der Topologie
- Routingkomplexität
- Gesamtzahl der Einzelverbindungen
- maximale bzw. durchschnittliche Entfernung zweier Knoten
- Anzahl der Nachbarn eines Knotens
- ...

Bestimmt mit die Kosten und die Leistungsfähigkeit eines Systems

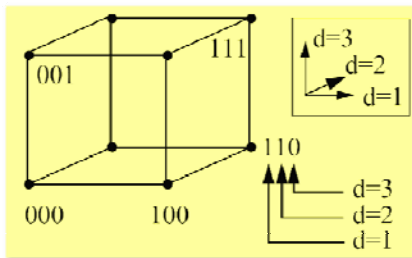
Beispiel: Hypercube-Verbindungstopologie

- **Würfel der Dimension d**
 - Vorteil: einfaches Routing, kurze Weglängen
 - Nachteil: Viele Einzelverbindungen ($O(n \log n)$ bei n Knoten)
- **Rekursives Konstruktionsprinzip**
 - Hypercube der Dimension 0: Einzelrechner
 - Hypercube der Dimension $d+1$: „Nimm zwei Würfel der Dimension d und verbinde korrespondierende Ecken“

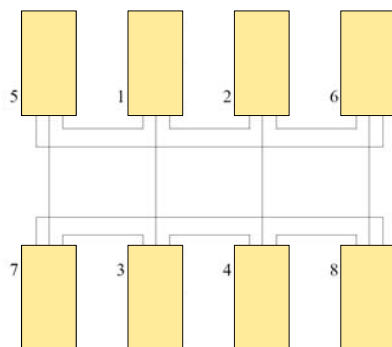


Routing beim Hypercube

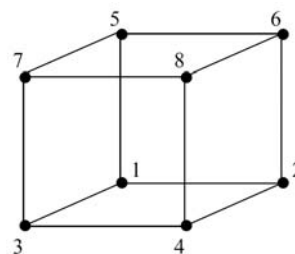
- Knoten **systematisch nummerieren** (entspr. rekursivem Aufbau)
- Zieladresse **bitweise xor** mit Absenderadresse
- Wo sich eine "1" findet, in diese Dimension muss gewechselt werden
- Maximale Weglänge: d ; durchschnittliche Weglänge = $d/2$ (Induktionsbeweis als einfache Übung)



Hypercube

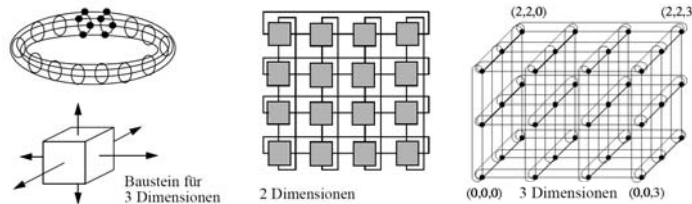


- Dies stellt auch einen **Hypercube** dar
 - ist nur nicht als Würfel gezeichnet



Eine andere Verbindungstopologie: Der d-dimensionale Torus

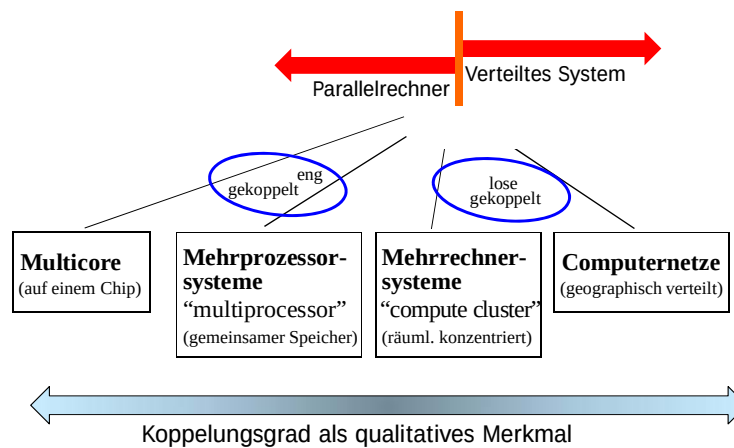
- Gitter in d Dimensionen mit “wrap-around”



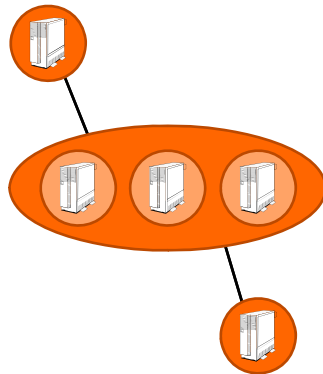
- Rekursives Konstruktionsprinzip:** Nimm w_d gleiche Exemplare der Dimension d-1 und verbinde korrespondierende Elemente zu einem Ring
 - Sonderfall **Ring**: $d = 1$
 - Sonderfall **Hypercube**: d-dimensionaler Torus mit $w_i = 2$ für alle Dimensionen i

Es gibt noch einige andere sinnvolle Verbindungstopologien (auf die wir nicht eingehen)

Parallelrechner ↔ verteiltes System



Architekturen verteilter Systeme: Service-Oriented Architecture (SOA)



Eine **Unterteilung** der Applikation in einzelne, unabhängige Abläufe innerhalb eines **Geschäftsprozesses** erhöht die Flexibilität weiter

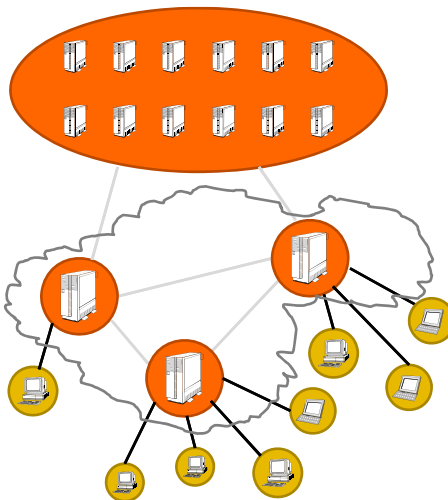
Lose Kopplung zwischen Services über Nachrichten und events (statt RPC)

Services können bei Änderungen der Prozesse **einfach neu zusammengestellt** werden („development by composition“)

Services können auch von **externen Anbietern** bezogen werden

Oft in Zusammenhang mit **Web-Services**

Architekturen verteilter Systeme: **Cloud-Computing**

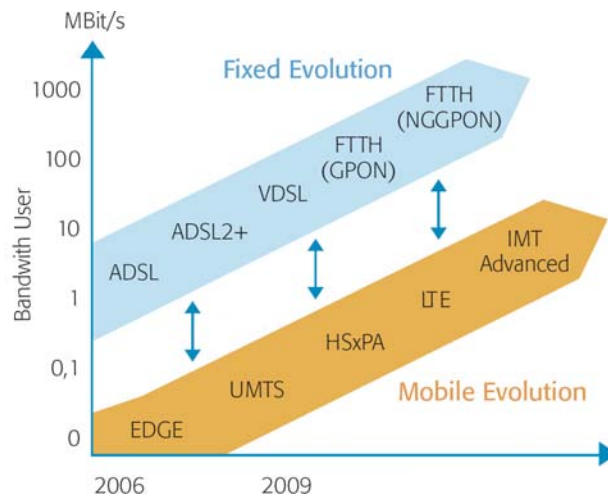


Massive **Bündelung der Rechenleistung** an zentraler Stelle

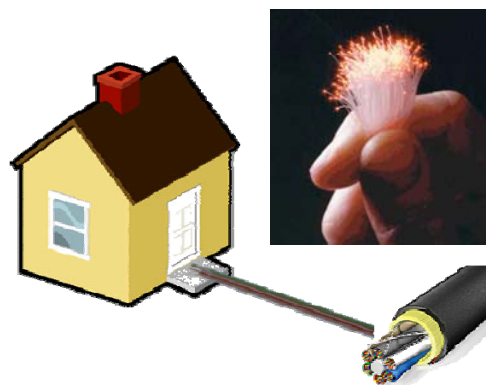
Outsourcen von Applikationen in die Cloud

Internet im Wesentlichen nur noch als **Vermittlungsinstanz**

Motivierender Trend: Stetige Erhöhung der Bandbreite für Endnutzer



Hochgeschwindigkeit ins Haus



- Konvergenz TV, Telekommunikation und Internet
- Technologiewechsel → erhebliche Investitionen
 - wirtschaftliche Faktoren und Bedingungen

▪ Telefondraht → Internet → TV
 ▪ TV-Kabel → Telefon → Internet

} → Glasfaser

Cloud-Computing



E-Mail wird beim
Provider gespeichert



Cloud-Computing



Fotos werden bei
flickr gespeichert



Cloud-Computing



Videos bei **You Tube**

Cloud-Computing



Private Dokumente
werden bei einem
Storage Provider
abgelegt

Das Tagebuch wird
öffentlich „im Netz“
als Blog geführt

Cloud-Computing



Informieren tut man sich im Netz

Google



Cloud-Computing



Vernetzen tut man sich bei „social networks“ oder „digital communities“ im Netz



Cloud-Computing



Plattformen im Netz
nutzt man zum

- Kaufen
- Spielen
- Kommunizieren
- ...

Google
Romance BETA

Cloud-Computing



Vorteile für Nutzer:

- von überall zugreifbar
- keine Datensicherung
- keine Softwarepflege

Kein PC, sondern
billiges Web-Terminal,
Smartphone etc.

Wie Wasserleitungen einst den eigenen Brunnen überflüssig machten...

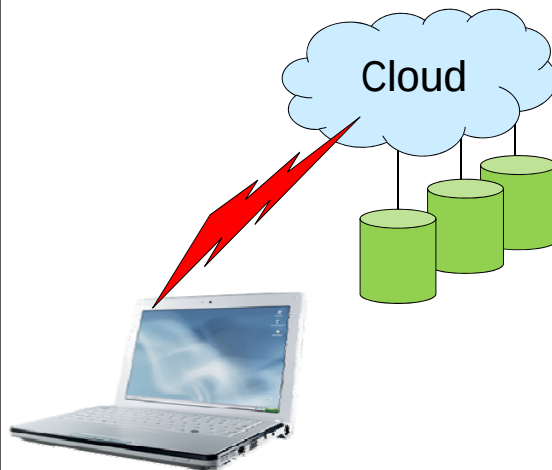
Cloud-Computing



Voraussetzungen?

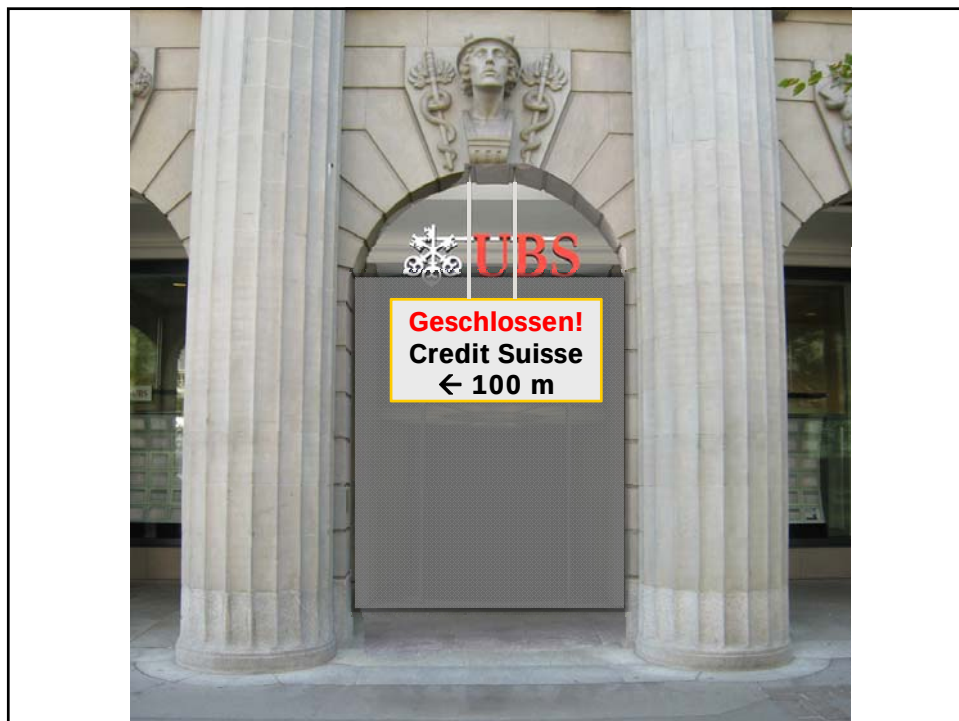
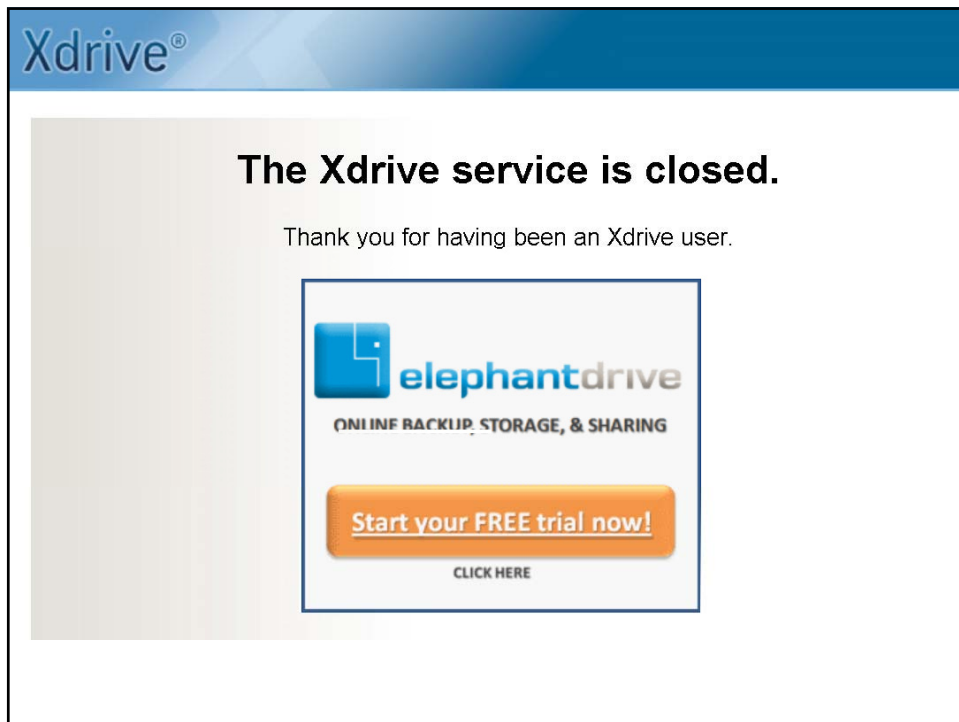
- Überall Breitband (fest & mobil)
- Netz-Verlässlichkeit (Versorgungssicherheit, Datenschutz,...)
- Wirtschaftlichkeit

Verlässlichkeit?



Voraussetzungen?

- Überall Breitband (fest & mobil)
- Netz-Verlässlichkeit (Versorgungssicherheit, Datenschutz,...)
- Wirtschaftlichkeit



Cloud-Computing



Plattformen:

- Wer betreibt sie? Wo?
- Wer verdient daran?
- Wer bestimmt?
- Wer kontrolliert?
- Welche Nationen profitieren davon?

Beispiel: Google-Datenzentren



- Jedes Datenzentrum hat **10 000 – 100 000 Computer**
- Kostet über **500 Mio \$** (Bau, Infrastruktur, Computer)
- Verbraucht **50 – 100 MW** Energie (Strom, Kühlung)
- Neben Google weitere (z.B. Amazon, Microsoft, Ebay,...)

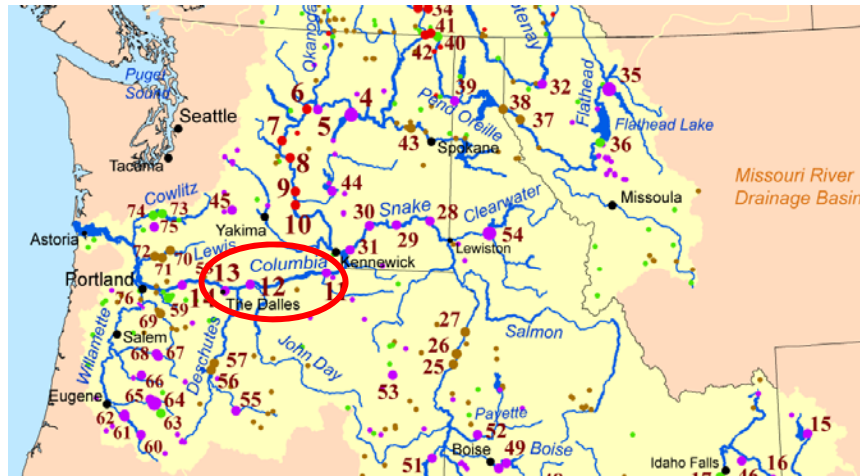
Google Data Center Groningen



Google Data Center Columbia River



The Dalles, OR, Columbia River



Google Data Center Columbia River



Energiezufuhr „Discovery Substation“ : 115 kV / 13.8 kV

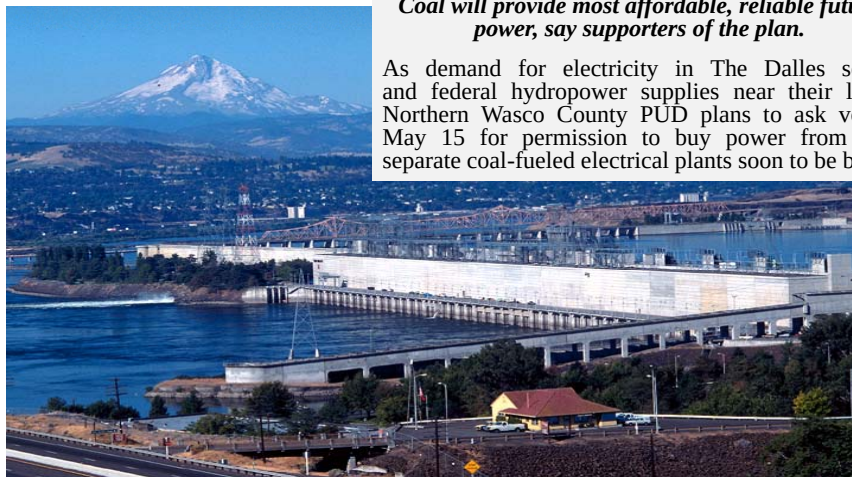


Nahes Kraftwerk: Dalles Dam Power Station, Columbia River

The Chronicle, March 1, 2007 –
PUD to seek vote on coal power

Coal will provide most affordable, reliable future power, say supporters of the plan.

As demand for electricity in The Dalles soars, and federal hydropower supplies near their limit, Northern Wasco County PUD plans to ask voters May 15 for permission to buy power from two separate coal-fueled electrical plants soon to be built.



Innenansicht eines Cloud-Zentrums

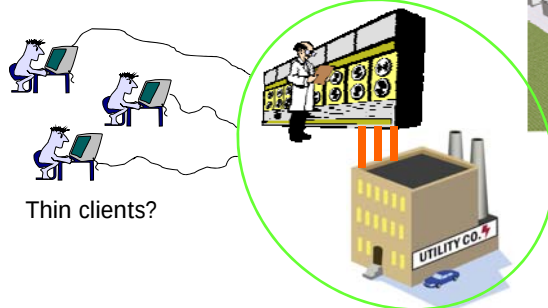


- Effizient wie **Fabriken**
 - Produkt: Internet-Dienste
- **Kostenvorteil** durch Skaleneffekt
 - Faktor 5 – 7 gegenüber traditionellen „kleinen“ Rechenzentren
- Angebot nicht benötigter Leistung auf einem **Spot-Markt**

Das entwickelt sich zum eigentlichen Geschäft!

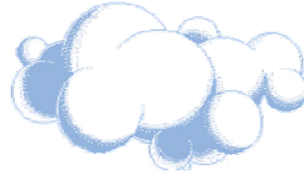
Zukünftige Container-Datenzentren

- Hunderte von **Containern** aus je einigen tausend Compute-Servern
 - mit Anschlüssen für Strom und Kühlung
- Nahe an **Kraftwerken**
 - Transport von Daten billiger als Strom



Cloud-Computing für die Industrie und Wirtschaft

- **Spontanes Outsourcen** von IT inklusive Geschäftsprozesse
 - Datenverarbeitung als Commodity
 - Software und Datenspeicher als Service
- Keine Bindung von Eigenkapital
 - **Kosten nach „Verbrauch“**
- **Elastizität**: Sofortiges Hinzufügen weiterer Ressourcen bei Bedarf
 - virtualisierte Hardware



Markt für „utility computing“
2010: ca. 95 Milliarden EUR

Architekturen verteilter Systeme: **Zusammenfassung**

- Peer-to-Peer
- Client-Server (Fat-Client vs. Thin Client)
- 3-Tier
- Multi-Tier
- Service-Oriented Architecture (SOA)
- Compute-Cluster
- Cloud-Computing

Resümee (1)

- **Verteilte Systeme:** Begriff, Sichtweisen, Eigenschaften,...
- **Warum** verteilte Systeme?
 - Kooperation von a-priori geographisch verteilten Einheiten
 - Verteilte Systeme als "Verbund"
- **Historische Entwicklung** von Systemen und Konzepten
- **Architekturvarianten**
 - Peer-to-Peer
 - Client-Server
 - Fat-Client vs. Thin Client
 - 3-Tier und Multi-Tier
 - Compute-Cluster
(Beispiel für Verbindungstopologie:
Hypercube und Torus)
 - Service-Oriented Architecture (SOA)
 - Cloud-Computing

Resümee (1b)

- **Cloud-Computing**
 - motiviert durch schnelle, ubiquitäre, verlässliche Netze
 - Trend: „alles“ irgendwo im Netz
 - Beispiele für Cloud-Datenzentren
 - wirtschaftliche Effekte: Skaleneffekte, Spot-Markt