

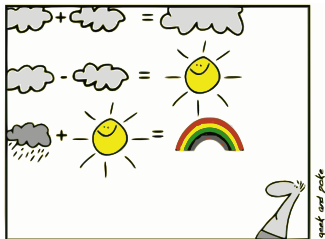
Technische Aspekte des Cloud Computing

Dr. Christian Baun

wolkenrechnen@gmail.com

16.3.2012

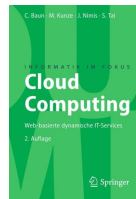
Cloud Computing – Definitionen



SIMPLY EXPLAINED - PART 17:
CLOUD COMPUTING

- Diverse Definitionen existieren
- Nicht alle sind hilfreich
- Gute Definitionen sind umfangreich

„Unter Ausnutzung virtualisierter Rechen- und Speicherressourcen und moderner Web-Technologien stellt Cloud Computing skalierbare, netzwerk-zentrierte, abstrahierte IT-Infrastrukturen, Plattformen und Anwendungen als on-demand Dienste zur Verfügung. Die Abrechnung dieser Dienste erfolgt nutzungsabhängig.“



Alles klar?

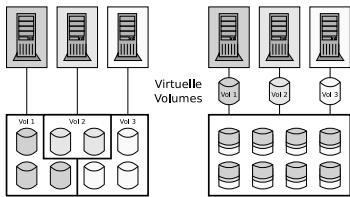
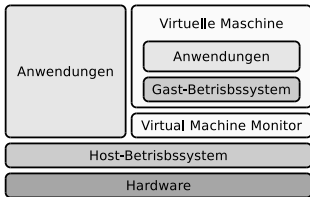
Cloud Computing – Schwerpunkte laut Definition

„*Unter Ausnutzung virtualisierter Rechen- und Speicherressourcen und moderner Web-Technologien stellt Cloud Computing skalierbare, netzwerk-zentrierte, abstrahierte IT-Infrastrukturen, Plattformen und Anwendungen als on-demand Dienste zur Verfügung. Die Abrechnung dieser Dienste erfolgt nutzungsabhängig.*“

- **Teil 1:** Grundlegende Technologien – Fundament des Cloud Computing
 - **Virtualisierung**
 - **Web-Services**
- **Teil 2:** Dienste und deren Eigenschaften
 - **Infrastrukturdienste, Plattformdienste, Softwaredienste**
 - **skalierbar** ⇒ „elastisch“
 - **netzwerk-zentriert** ⇒ Dienste/Ressourcen über das Netz erreichbar
 - **abstrahiert** ⇒ unabhängig von der konkreten Hardware
 - **on-demand** ⇒ zeitnahe Erfüllung von Anforderungen bzw. Nachfragen
 - **verbrauchsabhängige Abrechnung**

Grundlegende Technologien – Virtualisierung

- Erlaubt eine abstrakte, logische Sicht auf physische Ressourcen
 - Server, Speicher, Netzwerke
- Isoliert (versteckt) die physische Hardware
- Physische Hardware kann dank Virtualisierung...
 - ① transparent gemeinsam verwendet werden und
 - ② heterogene Hardwareressourcen können zu einem homogenen Ressourcenpool zusammengefügt werden



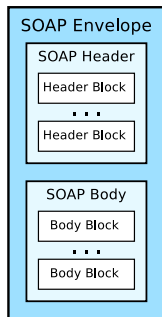
- Einige Vorteile
 - Serverkonsolidierung \implies Kostensenkung
 - Vereinfachte (kurzfristige) Bereitstellung
 - Flexibilität (Verschiedene Betriebssysteme auf einer Hardware)

Grundlegende Technologien – Web-Services

- Verteilte Systeme integrieren häufig heterogene Ressourcen
 - Theoretische Ausbreitung dieser Ressourcen ist weltweit
- Nachteile von Verbindungen über weite Strecken im Vergleich zu LANs
 - Hohe Antwortzeiten
 - Geringe Datenübertragungskapazitäten
 - Potentiell unzuverlässige Verbindungen
- Web-Services ermöglichen **schwach gekoppelte, asynchrone** und **nachrichtenbasierte** Kommunikation auf Basis von HTTP und XML
- Die populärsten Anwendungsmöglichkeiten für Web Services sind
 - Entfernte Funktionsaufrufe (Remote Procedure Calls)
 - **SOAP** (früher: *Simple Object Access Protocol*)
 - **REST** (*REpresentational State Transfer* – Zustandsrepräsentationsübertragung)

Web-Services – SOAP

- SOAP-Nachrichten basieren auf der Auszeichnungssprache XML
 - Meist werden SOAP-Nachrichten im Body einer HTTP POST-Anforderung an eine URL geschickt



```
<?xml version="1.0" encoding="UTF-8" ?>
<env:Envelope xmlns:env="http://www.w3.org/2001/09/soap-envelope">
  <env:Header>
    <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Mary um 14 Uhr von der Schule abholen</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

Quelle: Tanenbaum, van Steen: *Verteilte Systeme*. Pearson Studium (2008)

- Die Nachricht sendet einen Text an einen Web Service
- Die Nachricht hat eine bestimmte Priorität (1) und wird verworfen, wenn Sie nach 14 Uhr beim Web Service ankommt

Web-Services – REST

- RESTful Web Services werden über die HTTP-Schnittstelle angesprochen
- Zustandslose Kommunikation
 - Jede HTTP-Nachricht enthält alle nötigen Informationen, um die Nachricht zu verstehen
 - Server hält keine Zustands- bzw. Sitzungsinformation über den Client vor
 - Jede Anfrage ist eine von anderen Anfragen unabhängige Transaktion
- 4 HTTP-Methoden genügen, um mit Ressourcen zu arbeiten

HTTP	CRUD-Aktionen	SQL	Beschreibung
PUT	Create	INSERT	Ressource erzeugen oder deren Inhalt ersetzen
GET	Read/Retrieve	SELECT	Ressource bzw. deren Repräsentation anfordern
POST	Update	UPDATE	Einer Ressource etwas hinzufügen
DELETE	Delete/Destroy	DELETE	Ressource löschen

- Weitere sinnvolle HTTP-Methoden:
 - HEAD – fordert Metadaten zu einer Ressource an
 - OPTIONS – fragt an, welche Methoden auf einer Ressource möglich sind

Cloud Computing – Dienste?

*„Unter Ausnutzung virtualisierter Rechen- und Speicherressourcen und moderner Web-Technologien stellt Cloud Computing skalierbare, netzwerk-zentrierte, abstrahierte **IT-Infrastrukturen, Plattformen und Anwendungen** als on-demand **Dienste** zur Verfügung. Die Abrechnung dieser Dienste erfolgt nutzungsabhängig.“*

- Cloud Computing ist ein Sammelbegriff für verschiedene Dienste

Was versteht man unter **Diensten**?

IT-Dienst – *Service*

- Dienstleistung aus dem Bereich der Informationstechnologie (IT)
- Stellt ein Dienstanbieter (*Service Provider*) für einen oder mehrere Kunden (*Customer*) bereit
- Wird ähnlich einem Produkt angeboten
- Sollte über ein Service-Level-Agreement (SLA) definiert sein
- Erbringt ein externer Anbieter oder eine unternehmenseigene Abteilung
- Outsourcing = Auslagern von IT-Dienstleistungen

Wie unterscheidet man Cloud-Dienste **organisatorisch**?

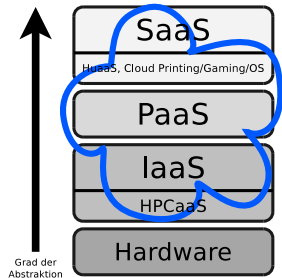
Organisatorische Unterscheidung der Dienste

- **Öffentliche Dienste** (Public Cloud)
 - Anbieter und Kunden gehören unterschiedlichen Organisationen an
⇒ Outsourcing
 - Keine Kosten für Anschaffung, Betrieb und Wartung eigener Hardware
 - Ressourcen sofort einsatzbereit und (fast) unbegrenzt verfügbar
- **Private Dienste** (Private Cloud)
 - Anbieter und Benutzer gehören der gleichen Organisation an
 - Kosten ähnlich einer nicht-Cloud-basierten Architektur
- **Hybride Dienste** (Hybrid Cloud)
 - Öffentlich verfügbare und private Dienste werden gemeinsam verwendet
 - Einsatzbeispiele:
 - Lastspitzen mit öffentlichen Diensten abfedern
 - Sicherheitskopien in öffentliche Dienste auslagern

Wie unterscheidet man Cloud-Dienste **funktional**?

Funktionale Unterscheidung der Dienste

- **Softwaredienste (SaaS)**
 - Anbieter betreibt Webanwendungen
 - Kunden brauchen nur einen Browser
- **Plattformdienste (PaaS)**
 - Anbieter betreibt skalierbare Laufzeitumgebung(en)
 - Kunden betreiben eigene Webanwendungen in der Infrastruktur des Diensteanbieters
- **Infrastrukturdienste (IaaS)**
 - Anbieter betreibt physische Server
 - Kunden betreiben VMs mit (fast) beliebigen Betriebssystemen und unveränderten Anwendungen
 - Kunden haben in ihren VMs Administratorenrechte und definieren die Firewall-Regeln



Beispiele für Softwaredienste

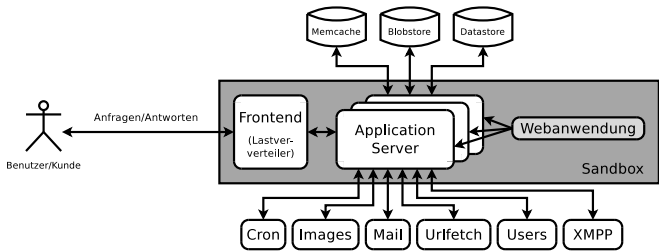


- (Freie) Lösungen zum Aufbau von Softwarediensten gibt es seit über 10 Jahren
 - Web-Server: Apache HTTP Server, nginx, . . .
 - Anwendungsserver für Webanwendungen: Apache Tomcat (Java), JBoss (Java), Zope (Python)
 - Skriptsprache für dynamische Webseiten: PHP
- Softwaredienste gibt es schon länger als den Begriff „Cloud Computing“

Beispiel für einen Plattformdienst – App Engine



- Kunden können eigene Webanwendungen in Python 2.5.2/2.7, Java 6 und Go betreiben
- Skaliert automatisch nach Bedarf
- Anwendungen können verschiedene Infrastruktur- und Speicherdienste nutzen



- Weitere öffentliche Plattformdienste: AWS Elastic Beanstalk, Engine Yard, Windows Azure Platform, Force.com

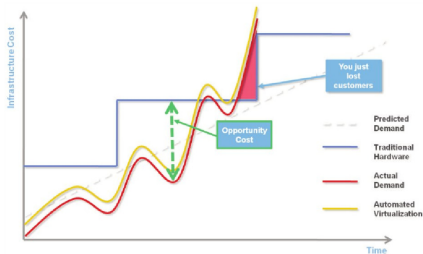
Beispiel für Infrastrukturdienste – Amazon Web Services

- Sammlung öffentlicher Cloud-Dienste
- Bekannte Dienste der AWS sind u.a.



Elastic Compute Cloud (EC2)	⇒	Infrastrukturdienst für virtuelle Server
Simple Storage Service (S3)	⇒	Speicherdienst für Webobjekte
Elastic Block Store (EBS)	⇒	Speicherdienst für virtuelle Speichervolumen
Elastic Load Balancing (ELB)	⇒	Dienst für virtuelle Lastverteiler
CloudWatch	⇒	Dienst zum Überwachen von AWS-Ressourcen
Auto Scaling	⇒	Dienst zum Skalieren von EC2-Kapazitäten

Take the Risk Factor out of Capacity Planning

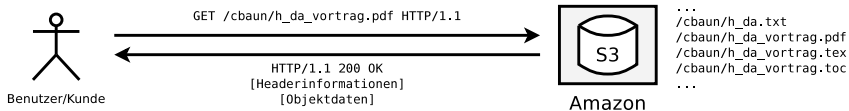


Source: Amazon Web Services

- Anwendungsbeispiele:
 - Rechenzentren virtuell realisieren
 - Elastische Infrastruktur aufbauen (z.B. für ein Startup)
 - Kurzfristig Ressourcen akquirieren
- Weitere öffentliche Infrastrukturdienste: Rackspace, GoGrid, IBM SmartCloud Enterprise

Beispiel für einen Infrastrukturdienst – Amazon S3

- Simple Storage Service (S3)
- Daten werden als **Objekte** (1 Byte bis 5 TB) gespeichert
- Jedes Objekt ist einem **Bucket** zugeordnet
 - Jeder Bucket hat einen eindeutigen Namen und kann keinen weiteren Bucket enthalten
- Verzeichnisse sind nicht möglich
- Objekte sind online erreichbar
 - `http://s3.amazonaws.com/bucket/objekt`
- Zugriff auf Buckets und Objekte geschieht via REST oder SOAP



- Im Dezember 2011 waren 762 Milliarden Objekte in S3 gespeichert

Interaktion mit den Diensten

- Hauptunterschiede zwischen den Diensten untereinander:
 - Funktionalität
 - Verfügbarkeit bzw. Qualität
 - Preis
 - **Schnittstelle** \Leftarrow Wird häufig nicht erst genug genommen

Wie arbeiten die Benutzer/Kunden mit den Cloud-Diensten?

- Interaktion mit Diensten
 - SaaS \implies Browser
 - PaaS, IaaS \implies Protokollorientierte Programmierschnittstelle (API)
 - Werkzeuge der Dienst- und Drittanbieter verwenden die **jeweilige** API
- Gefahren
 - Datenschutz und Datensicherheit
 - Parental Computing
 - Lock-in (!!!)

Auswirkungen bei Wahl eines Dienstes

- Auswahl eines Betriebssystems oder einer Programmiersprache hat Auswirkungen u.a. auf:
 - Betriebssystem: Verwendbare Anwendungen, Sicherheit,...
 - Programmiersprache: Portabilität, verfügbare Bibliotheken,...
- Auswahl einer Schnittstelle hat u.U. noch gravierendere Auswirkungen

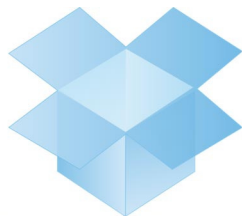
Ein Gedankenspiel

- Wechsel des Energieversorgers \implies Auswirkungen auf meine Geräte?
- Wechsel des Gasversorgers \implies Auswirkungen auf meine Heizung?
- Wechsel des Telefonanbieters \implies Auswirkungen auf mein Telefon?
- Wechsel des Cloud-Dienstanbieters \implies Auswirkungen auf meine Software bzw. mein Unternehmen?

Lock-in

- Entscheidet sich ein Kunde für einen öffentlich verfügbaren Dienst (PaaS oder IaaS), entscheidet er sich auch für eine Schnittstelle
- Gefahr des **Lock-in**
 - Abhängigkeit zwischen Dienstnutzer und -anbieter
- Denkbare Szenarien: Preiserhöhung, Änderung des Dienstangebots (Funktionalität), Insolvenz des Anbieters, . . .
- Wechsel des Anbieters nur bei gleichzeitigem **Verlust** der Infrastruktur (**Dienste**) und eventuell sogar der **Daten**
 - Auswirkungen für Kunden (insbesondere Unternehmen) u.U. fatal
- Verwendet man einen Dienst langfristig, **investiert** man in diesen
 - Dienste werden *veredelt*
 - Das eigene Geschäftsmodell wird darauf ausgerichtet
 - Mitarbeiter werden geschult

Auswirkungen des Lock-in



Dropbox

- Beispiel: Dropbox
- 2007 gegründeter Webdienst
- Stellt ein Netzwerk-Dateisystem für die Synchronisation von Dateien zwischen verschiedenen Rechnern und Benutzern bereit
- Verwendet zur Datenspeicherung Amazon S3
 - Geschäftsmodell: Einen Cloud-Dienst veredeln

- Was passiert mit Dropbox, wenn S3 die Preise verdoppelt oder auf einmal nicht mehr existiert?
- Was wären die Auswirkungen für die Kunden von Dropbox und S3?
- Was kann man gegen die Gefahr des Lock-in tun?

Vermeidung des Lock-in

- **Wettbewerber**

- Bieten öffentliche Dienste mit gleicher Funktionalität und Schnittstelle an
- Beispiele
 - Für S3 API: Google Storage, Host Europe Cloud Storage

- **(Freie) Lösungen**

- Aufbau privater Dienste mit gleicher Funktionalität und Schnittstelle
- Beispiele
 - Für EC2 API: Eucalyptus, Nimbus, OpenNebula, CloudStack, OpenStack
 - Für S3 API: Walrus (Eucalyptus), Cumulus (Nimbus), Swift (OpenStack)
 - Für GAE API: AppScale, typhoonAE
- Idealerweise kann man damit hybride Clouds realisieren

- Existieren Wettbewerber und/oder (freie) Lösungen mit kompatibler Schnittstelle, eröffnen sich viele Möglichkeiten

Karlsruhe Open Application for cLoud Administration

- Benutzer interagieren direkt mit Infrastruktur- und Speicherdiensten
 - Dienste sind meist Insellösungen
- Werkzeuge der Anbieter unterstützen meist nur wenige Dienste

The screenshot shows the Amazon Management Console interface. At the top, there is a 'Logout' button and flags for Germany and the UK. Below that, a dropdown menu shows 'Amazon EC2 (US East)' and a 'Region wechseln' button. The active region is 'Amazon (us-east-1)'. A dropdown for 'Funktionalität/Dienst auswählen' is also visible. The main content area displays details for two EC2 instances:

Instance ID	Status
i-a7b3f1cb	running
i-e1b0f28d	running

Below the table, the details for the first instance (i-a7b3f1cb) are shown:

- Status:** running
- Typ:** m1.large
- Reservation ID:** r-5a489337
- Root:** instance-store
- Besitzer:** 178412210831
- Image:** ami-688c7801
- Kernel:** aki-427d952b
- Ramdisk:** None
- Zone:** us-east-1a
- Gruppe:** default
- Extern:** ec2-184-73-1-108.compute-1.amazonaws.com
- Intern:** ip-10-124-170-46.ec2.internal
- Schlüssel:** testkeypair
- Datum:** 2011-02-17 12:52:38

- Wünschenswert: **Marktplatzportal** \implies KOALA
 - Dienst, der öffentliche und private Dienste verschiedener Anbieter unter einer GUI integriert
- Kein Werkzeug unterstützt mehr Cloud-Dienste
- Entwickelt für die GAE
 - Läuft auch im privaten Kontext
- Dienst: <http://koalacloud.appspot.com>
- <http://code.google.com/p/koalacloud/>
- Herausforderungen bei der Entwicklung:
 - Fähigkeiten der (privaten) Dienste
 - Qualität der Dokumentationen

Octopus Cloud Storage System

- Dienst, der S3-kompatible Dienste zu einem RAID-1 zusammenfügt



- Vorteile:
 - Höhere Verfügbarkeit der Daten
 - Unabhängigkeit von einzelnen Anbietern
- Entwickelt für die GAE

- Kopiert Objekte zu den genutzten Diensten und kontrolliert deren Synchronität anhand der MD5-Prüfsummen
- Dienst: <http://cloudoctopus.appspot.com>
- Projektseite: <http://code.google.com/p/octopuscloud/>
- Herausforderung bei der Entwicklung:
 - Verhalten der Dienste nicht zu 100% identisch

Zusammenfassung

- Anbieter betreiben virtualisierte Ressourcen als Dienste und ermöglichen den Zugriff über das Netz via Web Services
- Fundament des Cloud Computing
 - Virtualisierung zur gemeinsamen und effizienten Ressourcennutzung
 - Web-Services zur Kommunikation zwischen Kunden und Diensten sowie zwischen den Cloud-Diensten selbst
- Organisatorische Unterscheidung der Dienste
 - Public Cloud, Private Cloud, Hybrid Cloud
- Funktionale Unterscheidung der Dienste
 - SaaS, PaaS, IaaS
- Gefahr des Lock-in
 - Wettbewerber und Lösungen für private Dienste vermeiden diese Gefahr
 - Möglichkeit zum Daten-Export sollte immer geprüft werden
 - Portabilität der Daten beachten

10 TB Daten sollen aus einer Cloud exportiert werden

- Wie lange dauert die Übertragung via DSL mit 16.000 Kbit/s?

Aufgabe zur Portabilität von Daten

Daten in der Cloud (10 TB) 10.000.000.000.000 Byte

Bandbreite des DSL (16.000 Kbit/s) 16.000.000 Bit/s

Bandbreite des DSL in Byte/s 2.000.000 Byte/s

$$10.000.000.000.000 \text{ Byte} / 2.000.000 \text{ Byte/s} = 5.000.000 \text{ s}$$

Dauer der Datenübertragung [s] = 5.000.000 : 60

Dauer der Datenübertragung [min] = 83.333, $\bar{3}$: 60

Dauer der Datenübertragung [h] = 1.388, $\bar{8}$: 24

Dauer der Datenübertragung [d] $\approx 57,87$

\implies ca. 57 Tage, 20 Stunden, 53 Minuten

Fakten zum Cloud-Seminar im SS2012

- Alle Informationen zum Seminar:
<http://www.informatik.hs-mannheim.de/~baun/SEM12/>
- Bei Fragen, Email an: wolkenrechnen@gmail.com
- Erfolgreiche Teilnahme am Seminar:
 - Anfertigung einer schriftlichen Ausarbeitung (max. 10 Seiten)
 - Präsentation mit Foliensatz (45 Minuten inklusive Demonstration)
 - Verwendung der L^AT_EX-Vorlage für die Ausarbeitung
- Heute:
 - Vergabe freier Themen
 - Reihenfolge der Präsentationen festlegen
 - Termine für Präsentationen festlegen
- Achtung: Die **Anmeldung** zum Seminar heute ist **verbindlich!**

Achtung!

- Das Seminar im Hauptstudium ist anspruchsvoller als das Proseminar im Grundstudium
 - Ausarbeitung und Präsentation dürfen keine Werbeshow mit inhaltslosem BlaBla sein!
 - Nicht ausschließlich Theorie aufschreiben
 - Immer Bezüge zur Realität bringen
 - Lassen Sie Ihre Erkenntnisse einfließen
 - Nach Möglichkeit mit den Anbietern/Entwicklern in Kontakt treten
 - Bringen Sie die Architektur und technischen Grundlagen in Erfahrung
- Die Quellen auf der Seminar-Webseite sind nur eine erste Anlaufstelle
 - Eigene Recherche ist unerlässlich
 - Die Ausarbeitung sollte nicht ausschließlich Web-Quellen enthalten
- Eine gute Ausarbeitung ist inhaltlich und sprachlich so, das der Leser in das Thema einsteigen kann und umfassend informiert wird

Danke für Ihre Aufmerksamkeit!

Die Folien zu diesem Vortrag finden Sie unter
<http://www.informatik.hs-mannheim.de/~baun/SEM12/>

Wie schreibt man eine gute Abschlussarbeit

http://www.informatik.hs-mannheim.de/~baun/hilfe_abschlussarbeit.html

Merke: Qualität kommt von Qual.
Quälen soll sich der Autor und nicht der Leser!