



# Cloud Computing

Vorlesung an der Hochschule Karlsruhe -  
Technik und Wirtschaft im  
Sommersemester 2024

*Michael Fischer und Georg Magschok*  
mf@wanulator.de                      gio@eglikoe.de

Die Vorlesung im Web:

<http://pl.attitu.de/vorlesungen/cloudcomputing/>



# Roter Faden

- Einführung
- Grundlagen
  - Verteilte Systeme
  - Skalierungsdimensionen
  - Algorithmen
- Konkrete Cloud-Beispiele
- Lost&Found



# Wegweiser

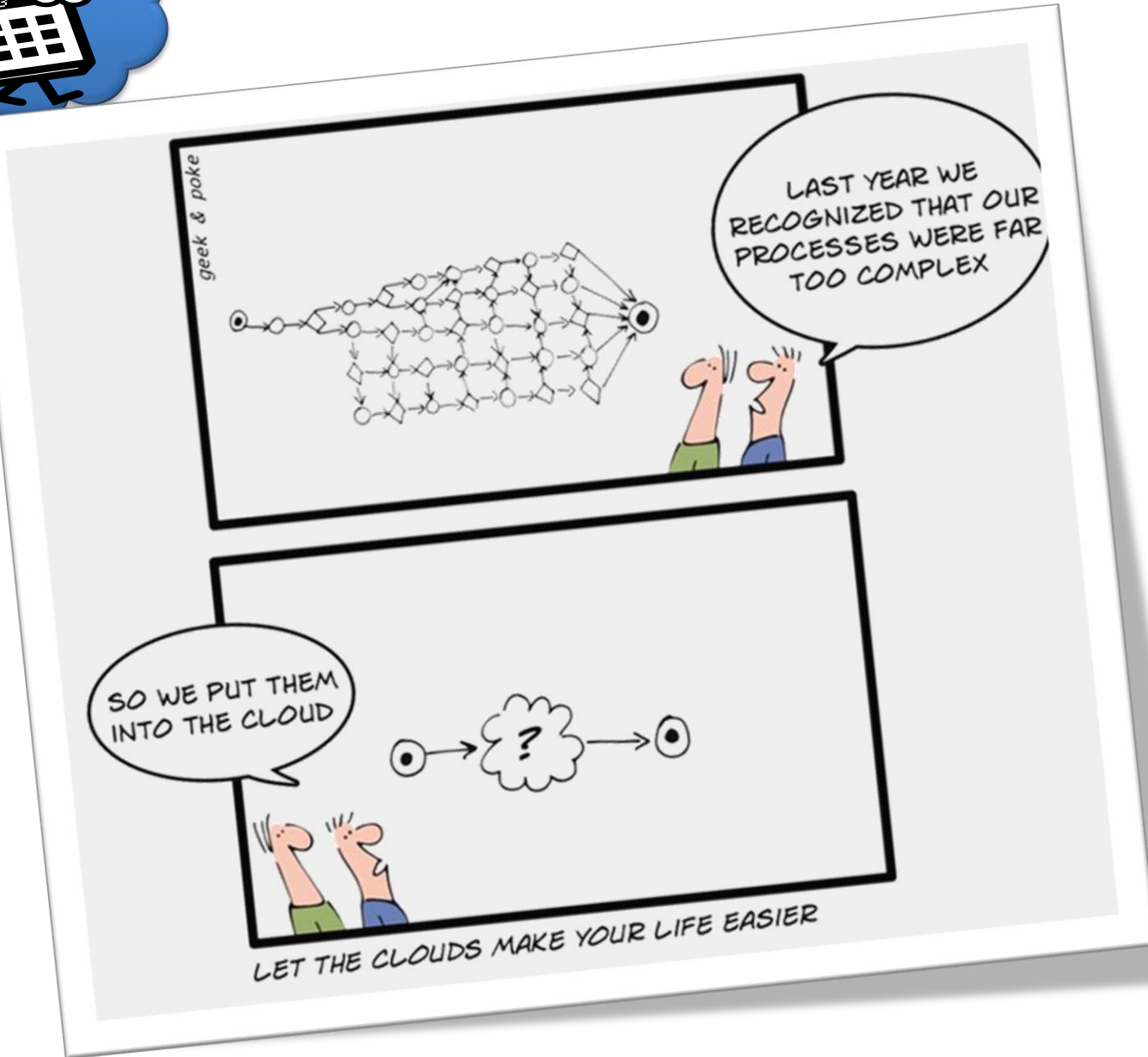
Einführung:

Was ist Cloud?



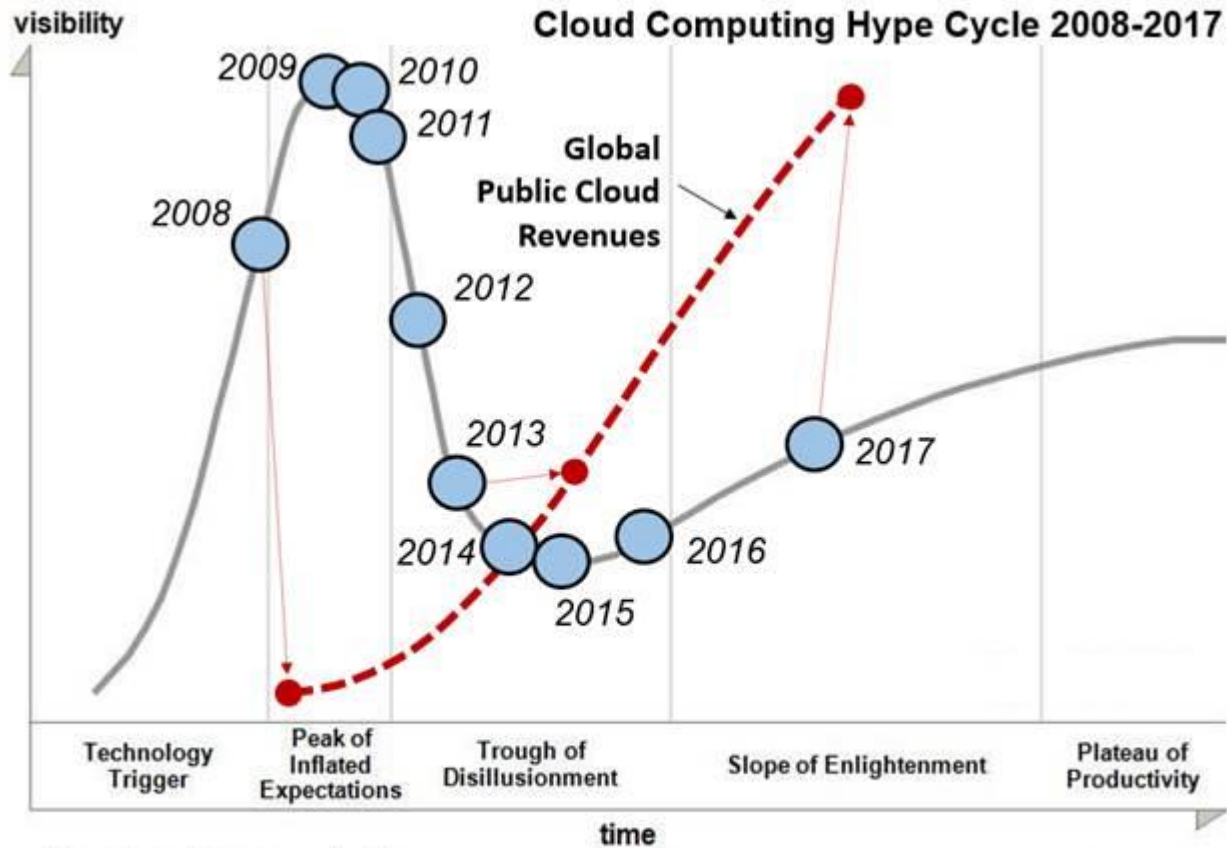


# Was ist Cloud?





# Buzzword?



Years to mainstream adoption:

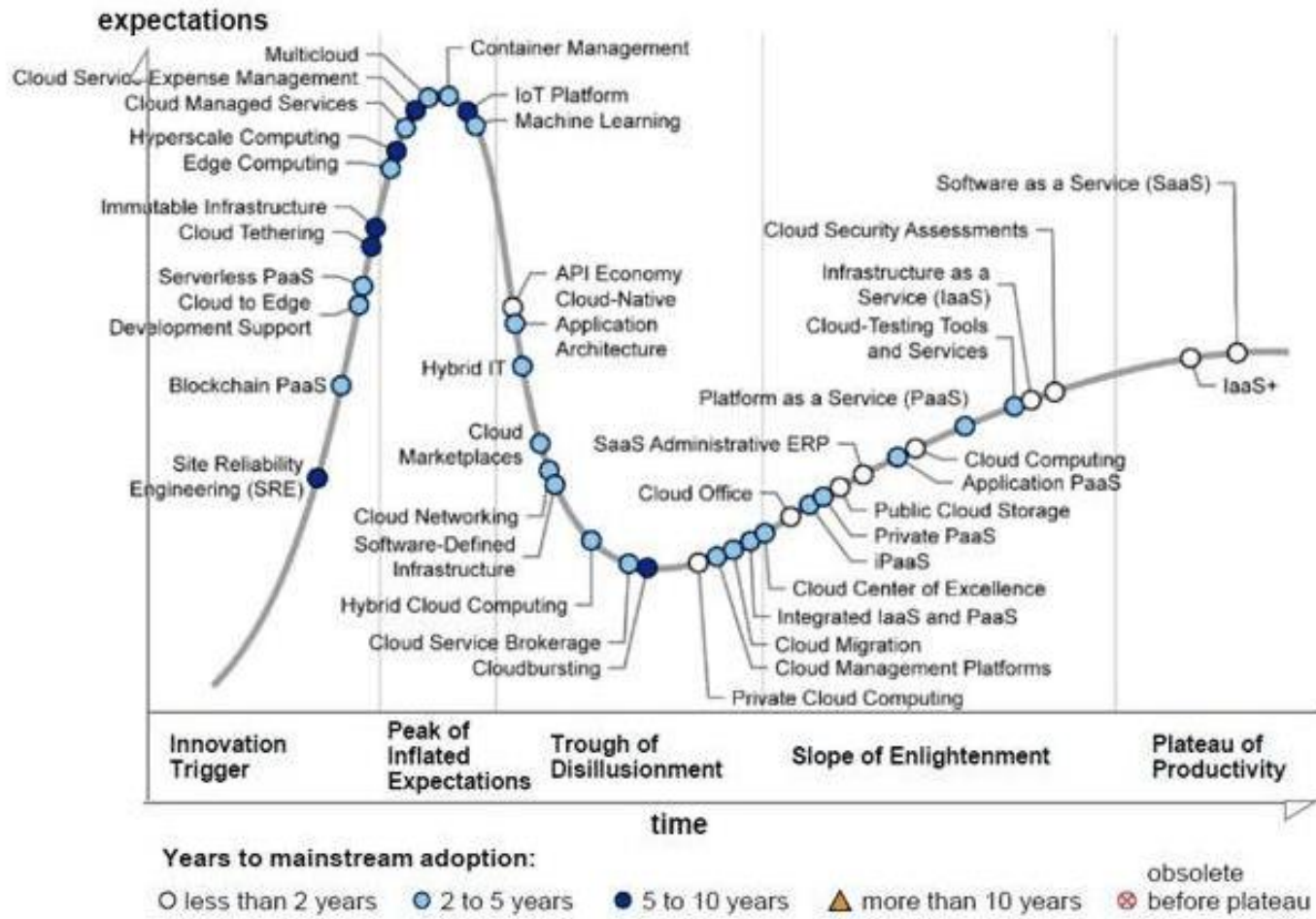
- less than 2 years
- 2 to 5 years
- 5 to 10 years
- ▲ more than 10 years
- ⊗ obsolete before plateau

Sources: Gartner, Forrester Research, Author

Quelle: Philip Nones, <https://nonesnotes.com>



# Buzzword? -> Cloud Hype Cycle



Quelle: Computerwoche



# Nahe Verwandtschaften

- Cluster (n Systeme Verhalten sich wie 1)
  - Kommerzielles Umfeld
  - LoadBalancing, HA
  - Lokaler Standort
  
- Grid
  - Wissenschaftliches Umfeld
  - Dezentraler Zugriff auf Dezentrale Ressourcen
  - Heterogen, Skalierbar, APIs,
  - „Job“ orientiert



# Warum Cloud?

Zahlung nach Nutzung

24x7 Support

Lokations- und  
Geräteunabhängigkeit

Automatisierbarkeit,  
bes. Deployment

Geringer Kapitaleinsatz

Verfügbarkeit, auch  
Backup

Skalierbarkeit

Aufwandseinsparungen  
in der Administration

Oversubscription

Kleiner Ausflug: <http://www.cloudorado.com/>





# Warum nicht Cloud?

- Bestimmte Ansprüche an RZ, Hardware
- Verfügbarkeit und Sicherheit abhängig von Dritten
- Kosten (applikationsabhängig)
- Anbieterabhängigkeit unerwünscht
- Technische oder Support-Probleme

# Historie



Steinzeit 1950+

Salesforce.com  
1999

Microsoft Azure  
2008

Google Apps  
2009

IBM Smartcloud  
2011

Oracle Cloud  
2012

OpenStack 2010

Eucalyptus 2008

Amazon 2006

Großrechner  
1980er/90er



# NIST Definition: Eigenschaften

- **On-demand Self Service:** Die Provisionierung der Ressourcen (z. B. Rechenleistung, Storage) läuft automatisch ohne Interaktion mit dem Service Provider ab.
- **Broad Network Access:** Die Services sind mit Standard-Mechanismen über das Netz verfügbar und nicht an einen bestimmten Client gebunden.
- **Resource Pooling:** Die Ressourcen des Anbieters liegen in einem Pool vor, aus dem sich viele Anwender bedienen können (Multi-Tenant Modell). Dabei wissen die Anwender nicht, wo die Ressourcen sich befinden, sie können aber vertraglich den Speicherort, also z. B. Region, Land oder Rechenzentrum, festlegen.
- **Rapid Elasticity:** Die Services können schnell und elastisch zur Verfügung gestellt werden, in manchen Fällen auch automatisch. Aus Anwendersicht scheinen die Ressourcen daher unendlich zu sein.
- **Measured Services:** Die Ressourcennutzung kann gemessen und überwacht werden und entsprechend bemessen auch den Cloud-Anwendern zur Verfügung gestellt werden.

Quellen: [https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen\\_node.html](https://www.bsi.bund.de/DE/Themen/CloudComputing/Grundlagen/Grundlagen_node.html)  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>



# NIST Definition: Servicemodelle

- **IaaS – Infrastructure as a Service** Rechnerwolken bieten Nutzungszugang von virtualisierten Computerhardware-Ressourcen wie Rechnern, Netzwerken und Speicher. Mit IaaS gestalten sich Nutzer frei ihre eigenen virtuellen Computer-Cluster und sind daher für die Auswahl, die Installation, den Betrieb und das Funktionieren ihrer Software selbst verantwortlich.
- **PaaS– Platform as a Service** Rechnerwolken bieten Nutzungszugang von Programmierungs- oder Laufzeitumgebungen mit flexiblen, dynamisch anpassbaren Rechen- und Datenkapazitäten. Mit PaaS entwickeln Nutzer ihre eigenen Software-Anwendungen oder lassen diese hier ausführen, innerhalb einer Softwareumgebung, die vom Dienstanbieter (Service Provider) bereitgestellt und unterhalten wird.
- **SaaS– Software as a Service** Rechnerwolken bieten Nutzungszugang von Software-Sammlungen und Anwendungsprogrammen. SaaS Diensteanbieter offerieren spezielle Auswahlen von Software, die auf ihrer Infrastruktur läuft. SaaS wird auch als *Software on demand* (Software bei Bedarf) bezeichnet.
- **(HaaS - Human as a Service)**

Quellen: <http://de.wikipedia.org/wiki/Cloud-Computing>  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>



# Beispiele IaaS

- Was:
  - CPU, RAM, Storage, Datenbanken, Netzwerkinfrastruktur (Router, Switches, Loadbalancer, ...)
- Produkte (kommerziell):
  - Amazon EC2/S3, IBM Smart Business Cloud, Jiffybox, 1&1 Dynamic Cloud Server
- Produkte (nicht kommerziell):
  - Open Stack, Eucalyptus, Open Nebula



# Beispiele PaaS

- Was:
  - Middleware – Java Runtime, Datenbanken, Entwicklungsumgebung, „APIs“
- Produkte (kommerziell):
  - Google AppEngine, Microsoft Azure
- Produkte (nicht kommerziell):
  - AppScale



# Beispiele SaaS

- Was:
  - Jede erdenkliche Applikation: ERP/CRM, Office, Mail, Photo/Video Bearbeitung, Cloud Gaming...
- Produkte (kommerziell):
  - Google Docs, Salesforce.com, Zoho Office, Adobe Photoshop Express,....



# NIST Definition: Deployment Modelle

- **Private Cloud** Bei „Private Clouds“ steht im Vordergrund, dass sich sowohl Anbieter als auch Nutzer im selben Unternehmen befinden, wodurch beispielsweise sämtliche Probleme aus dem Bereich Datensicherheit mehr oder minder hinfällig werden.
- **Public Cloud** Eine „Public Cloud“ ist eine „Cloud“, die öffentlich ist, d. h. von beliebigen Personen und Unternehmen genutzt werden kann und nicht mehr auf interne Anwendungen einer einzelnen Institution/eines Unternehmens beschränkt ist. Hierbei greifen dann auch vor allem Probleme, die mit Datensicherheit zu tun haben, und jeder Akteur muss sich selbst überlegen, wie viele und welche Daten er außerhalb seiner unmittelbaren Kontrolle halten möchte
- **Hybrid Cloud** Ein Unternehmen betreibt eine eigene „Private Cloud“ und nutzt zusätzlich als Failoverstrategie oder für Belastungsspitzen eine „Public Cloud“.

Quellen: <http://de.wikipedia.org/wiki/Cloud-Computing>  
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>





# (Load) Scaling Einführung

- Case-Study:  
„Wachstum eines Web-Services“
- Was machen eigentlich...
  - Facebook
  - Netflix
  - ...





# Motivation

- Erfolg:
  - Slashdot Effekt
- Problem Komplexität
  - Wissenschaftsbereich (CERN, ... )



# Tag 1: im Leben eines Web Services

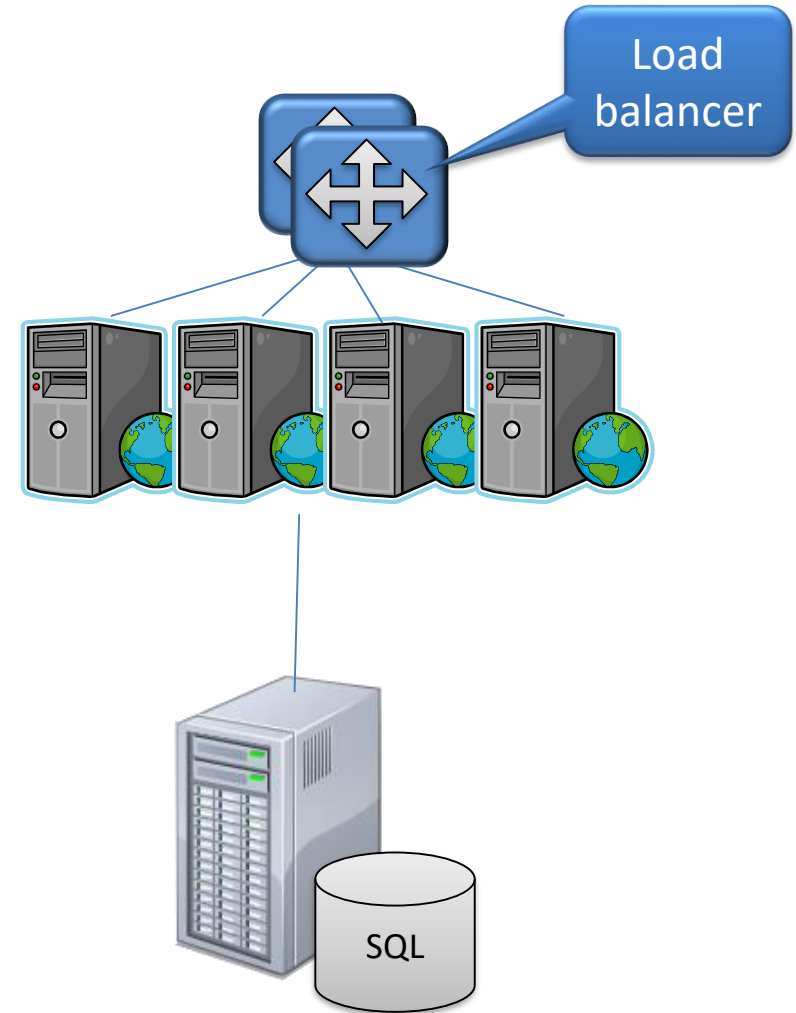
- Stats:
  - 40.000 PI/Tag
  - 0,5 Requests / s
  - ~2-3 Requests / s Peak
- Architektur:
  - Single Server
  - LAMP
    - Linux
    - Apache
    - MySQL
    - PHP





# Tag 2: im Leben eines Web Services

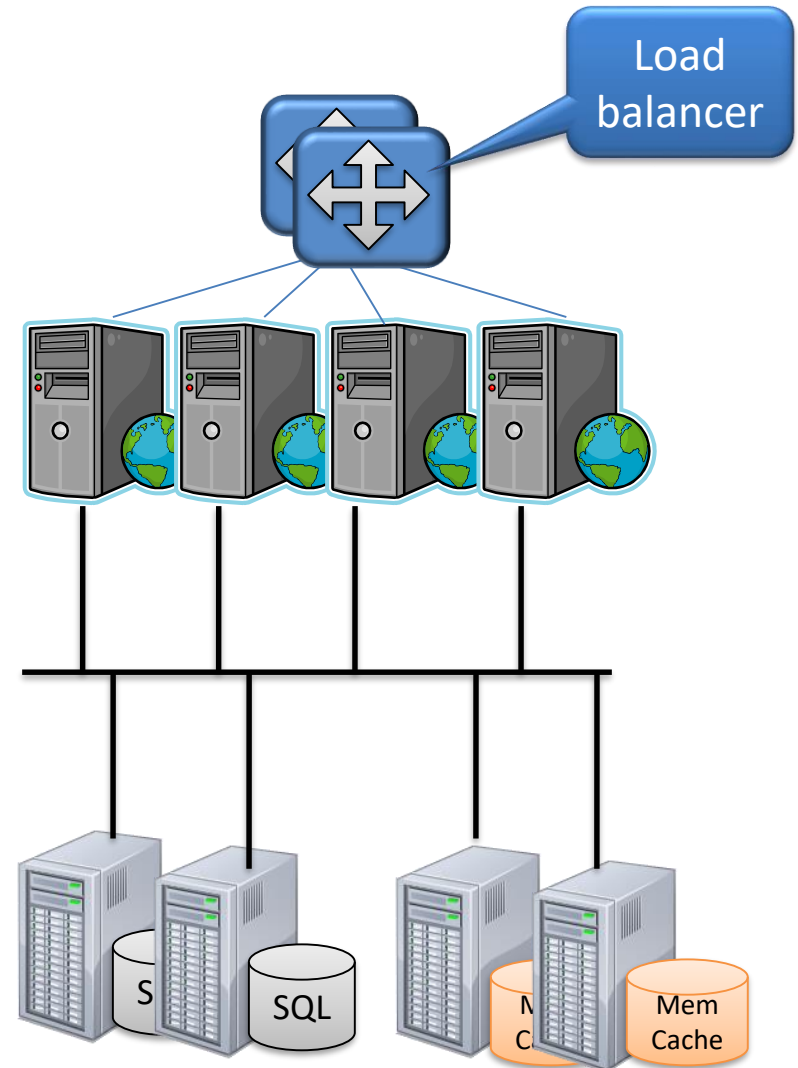
- Stats:
  - 2.000.000 PI/Tag
  - ~20 Requests / s
  - ~100 Requests / s Peak
- Architektur:
  - ~1-10 Server
  - Load Balancer
  - Webserver Farm
  - Apache
  - PHP
  - MySQL





# Tag 3: im Leben eines Web Services

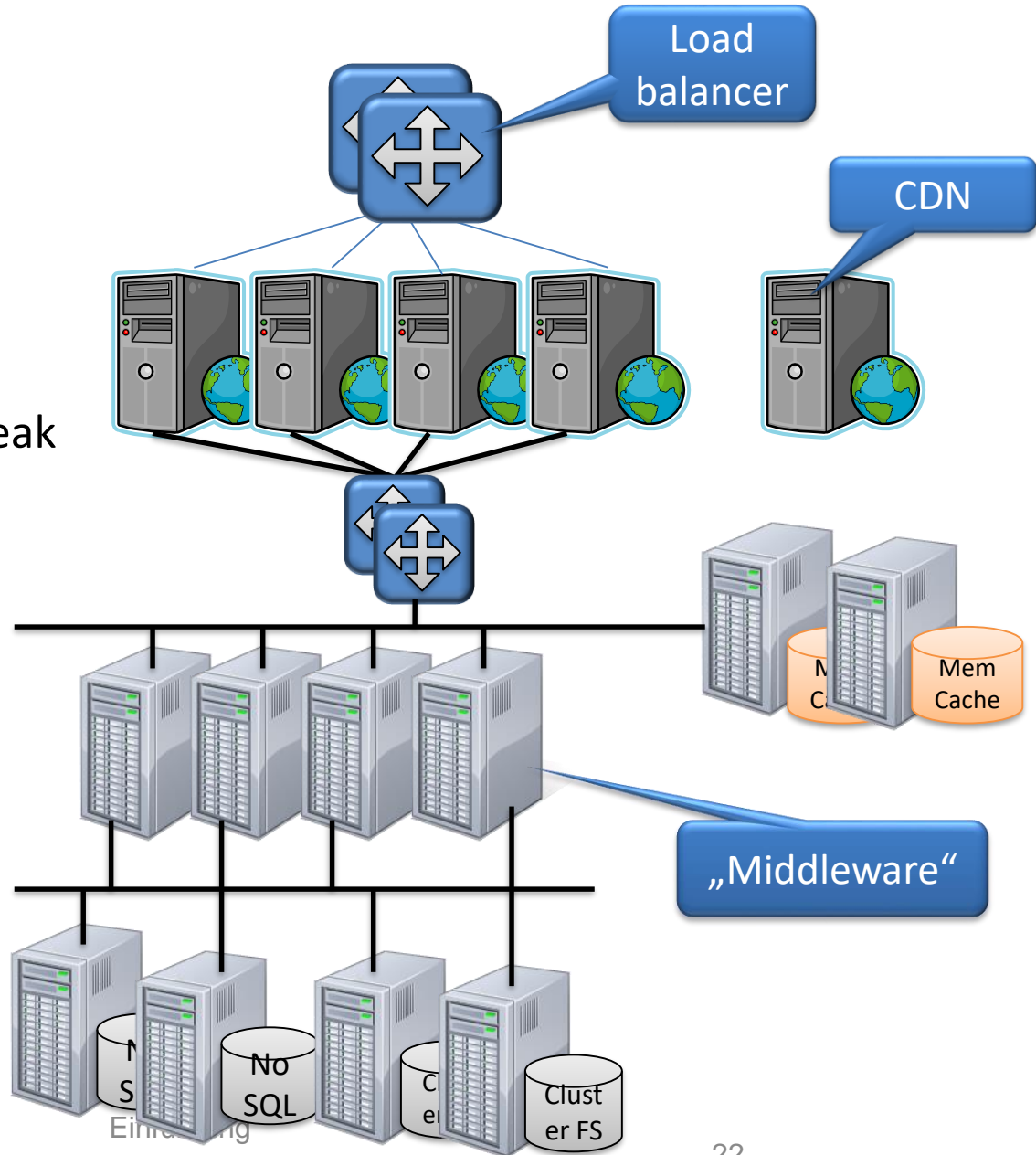
- Stats:
  - 10.000.000 PI/Tag
  - ~100 Requests / s
  - ~500 Requests / s Peak
- Architektur:
  - ~10-100 Server
  - MySQL Master Master Replikation, Read Slaves
  - Memcached
  - Änderung der Applikation!
    - Spezialisierung der Use-Cases





# Tag 4: im Leben eines Web Services

- Stats:
  - 1.000.000.000 PI/Tag
  - ~10000 Requests / s
  - ~50000 Requests / s Peak
- Architektur:
  - ~100-500 Server
  - „NoSql“
  - N-Tier Architektur
  - Optimiertes Caching
  - Content Delivery Networks





# Was macht eigentlich: Facebook

- 2016:
  - 1.8 Milliarden User (>1 Milliarde täglich aktiv!)
  - 28 Milliarden Visits/Monat, >10 Million req/s Peak
  - Mehrere ExaBytes Daten
  - ...
- „LAMP“ optimiert bis zum Maximum
  - PHP => C++ (HipHop)
  - Spezielle Memcached Infrastruktur
  - MySQL nur key/value Store keine joins etc.
  - Backend angebunden über Thrift Business Logik in PHP/C++/Java/...
- Jede Menge „modernere“ Technologien für das drumherum:
  - Cassandra => mailsystem search
  - Hadoop/Hive => Logging & BI & Suggestions & ...
  - Erlang => Chat
  - Photos => Haystack (Object / Metadastore by Facebook)



# Was macht eigentlich...

## Netflix

- 2013:
  - 50 k req/s
  - Streaming
- Amazon AWS > 13000 AWS Instances
  - Java
  - Cassandra
  - Hadoop
  - Zookeeper
  - AppDynamics
  - Chaos Monkey
  - Hysterix





# Was macht eigentlich: Dropbox

- 2016:
  - 500 Mio Users
  - 500 PB Storage
- Kritische Masse erreicht: innerhalb von 2.5 Jahren von Amazon S3 auf eigene Infrastruktur migriert
  - Individualentwicklung/Custom
  - Peak 500Gbps Migrationsperformance

Quelle: <https://blogs.dropbox.com/tech/2016/03/magic-pocket-infrastructure/>