

# Cloud Computing

## Kapitel 8: Cloud-fähige Softwarearchitekturen

Dr. Josef Adersberger

2002

**WAL\*MART®** [Help](#) | [Your Account](#) | [Cart & Checkout](#)

[Electronics](#) [Photo](#) [Movies](#) [Music](#) [Books](#) [Toys](#) [Home](#) [Garden & Hardware](#) [Sporting Goods](#) [Gifts](#) [Jewelry & Watches](#) [See All Departments](#)

Search  Entire Site  for:

**Roses on Rollback! One Dozen Fresh-Cut, Long-Stem Roses Only \$24.88**

**Home Page**

- Electronics**
  - Computers & Printers
  - Internet Service
  - TV, DVD & Video
  - [More...](#)
- Photo**
  - Digital Cameras
  - Camcorders
  - Photo Center
  - [More...](#)
- Movies** *12,000 Movies!*
- Music** *80,000 CDs!*
- Books** *500,000 Titles!*
- Video Games**
  - PS2
  - Xbox
  - [More...](#)
- Toys**
  - Outdoor Play
  - Bikes & Scooters
  - [More...](#)
- Baby & Baby Registry**
  - Shop by Age or Stage
- Home**
  - Furniture
  - Appliances
  - Bed & Bath
  - [More...](#)
- Garden & Hardware**
  - Patio & Grilling
  - Lawn Equipment
  - Home Improvement
  - [More...](#)
- Tires**
- Sporting Goods**
  - Exercise & Hot Tubs
  - Camping & Hiking
  - [More...](#)
- Gifts**

**Find 300,000 Textbooks**  
Every Day Low Prices

Find it by [ISBN](#) or [category](#).

- [Business](#)
- [Education](#)
- [Test Prep](#)
- [Medical](#)
- [Engineering](#)
- [Computer](#)

[See all textbooks.](#)

**Furnish the Dorm**

Find the desk that fits your living space and study habits. [See all desks.](#)

Sleep well with our low prices on bedding. [See all college bedding.](#)

Make dorm life easier with refrigerators, dishwashers and more. [See all small appliances.](#)

**Back To College**

- [Portable Electronics](#)
- [Cameras](#)
- [Shopping Cards](#)
- [See More](#)

**K-12**

- [Clothing](#)
- [School Supplies](#)
- [See More](#)

**WAL\*MART Services**

- [Upload Digital Photos](#)
- [Find a Gift Registry](#)
- [Find and Order Tires](#)
- [Refill Prescriptions](#)

Kaufprozess

Artikel

2008

[See All Departments](#)

Search

[FIND](#) [Help](#)

[Cart](#)

[Apparel & Jewelry](#) ▸

[Baby](#) ▸

[Books, DVDs & MP3s](#) ▸

[Electronics](#) ▸

[For the Home](#) ▸

[Gifts & Registries](#) ▸

[Grocery](#) BETA ▸

[Health & Beauty](#) BETA ▸

[Outdoor Living](#) ▸

[Pharmacy](#) ▸

[Photo](#) ▸

[Sports & Fitness](#) ▸

[Toys & Video Games](#) ▸

[In Stores Now](#) ▸

[See All Departments](#)

**Savings Add Up**

[Clearance](#)

[Rollbacks](#)

[\\$4 Prescriptions](#)

[Video Games Under \\$20](#)

[Clearance on Baby Items](#)

**What's New**

[Classifieds](#) BETA

[Halloween](#)

[Elevenmoms](#)

[Better Homes](#)

## GREAT VALUE, NEW PRODUCTS

Download Flash Player version 7 or later for the ultimate Walmart.com shopping experience.



### Blu-ray Price Drop



★★★★★ [Magnavox](#) **from \$258**  
[Sony](#)  
[Samsung](#)

See all:  
[Blu-ray Players](#)

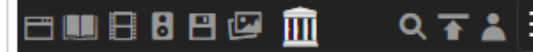
### 50% Off Top 50 Books



The 50 top sellers are 50% off — get great reads for the whole family, **from \$4**.

Shop:  
[50% Off Top 50 Books](#)

ADVERTISEMENT



INTERNET ARCHIVE  
**Wayback Machine**

[http://ad.yieldmanager.com/st?ad\\_type](http://ad.yieldmanager.com/st?ad_type)

[Logout](#)

### Save on Top-Brand Electronics



Zusätzliche  
Dienste

2011

See All Departments

- Electronics & Office >
- Movies, Music & Books >
- Home, Furniture & Outdoor >
- Apparel, Shoes & Jewelry >
- Baby & Kids >
- Toys & Video Games >
- Sports & Fitness >
- Auto & Home Improvement >
- Photo >
- Crafts & Party Supplies >
- Pharmacy, Health & Beauty >
- Grocery & Pets >
- College 2011**

Save Big Now

- Amazing Xbox 360 Savings
- Best-Selling Bikes from \$79
- Fragrances from \$5
- Furniture Rollbacks
- Save 10¢ per Gallon on Gas

What's New

- Captain America Is Here
- HP Dandelion Breeze Laptop
- Summer Fun

In the Spotlight

Search All Departments Search

Go

My Cart

Choose My Store

Must-Have College Essentials at Can't-Miss Prices

COLLEGE 2011

Bedding  
from  
**\$35**



Futons & Sofas  
from  
**\$99**



Storage  
from  
**\$19**



Desks  
from  
**\$39**



Laptops & Netbooks  
from  
**\$278**



Appliances  
from  
**\$57**



TVs  
from  
**\$149**



iPods  
from  
**\$49**

Shop for College 2011

ADVERTISEMENT

ADVERTISEMENT



Back to School Essentials

From uniforms to supplies, find everything for school at Every Day Low Prices.

School Supplies  
Uniforms



Have Your Fun in the Sun

Enjoy hot summer savings on everything you'll need for some outdoor family fun.

Pools & Waterslides  
Swing Sets  
Trampolines



Zusätzliche Dienste (z.T. personalisiert)

## CHALLENGE

Four years ago, the Walmart Global eCommerce system was a monolithic application, deployed once every 2 months. This represented an unsustainable rate of innovation given the competitive climate. Walmart recognized the need to fundamentally transform technology delivery to compete in the Digital Economy.

Walmart auf <http://www.oneops.com>

- “The company decided to replatform its old legacy system in 2012 since it was unable to scale for 6 million pageviews per minute and was down for most of the day during peak events.”
- “This is the multi-million dollar question which the IT Department of Walmart Canada had to address after they were failing to provide to their users on Black Fridays for two years in a row.”
- “It couldn’t handle 6 million pageviews per minute and made it impossible to keep any kind of positive user experience anymore.”

<https://blog.risingstack.com/how-enterprises-benefit-from-microservices-architectures>

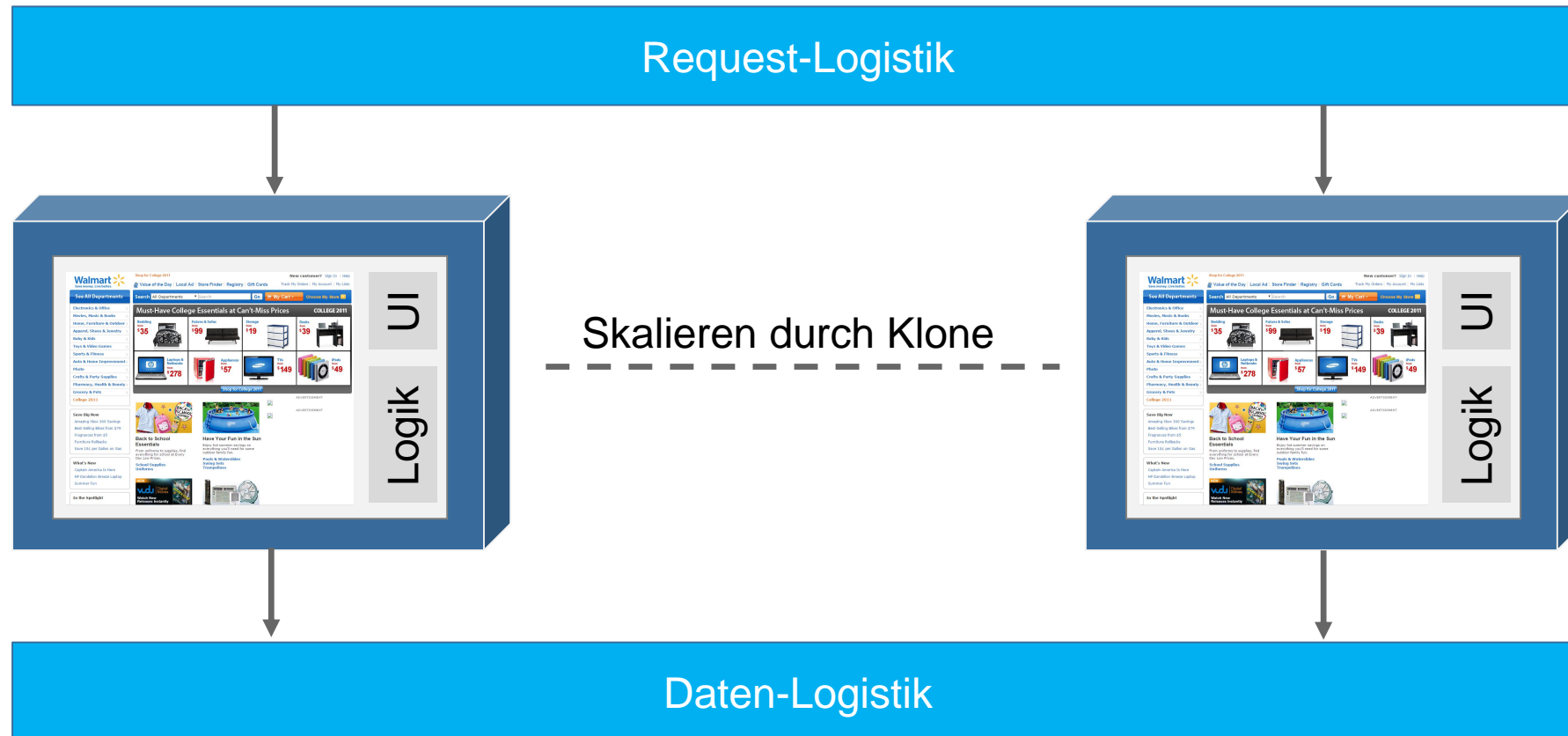
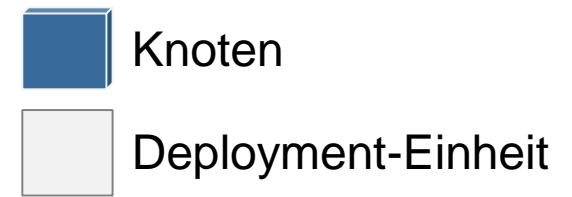
Lange Change-to-  
Production

Mangelnde  
Skalierbarkeit

Mangelnde Elastizität



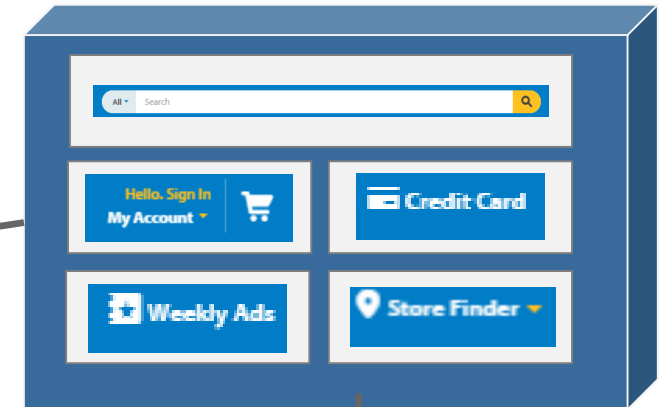
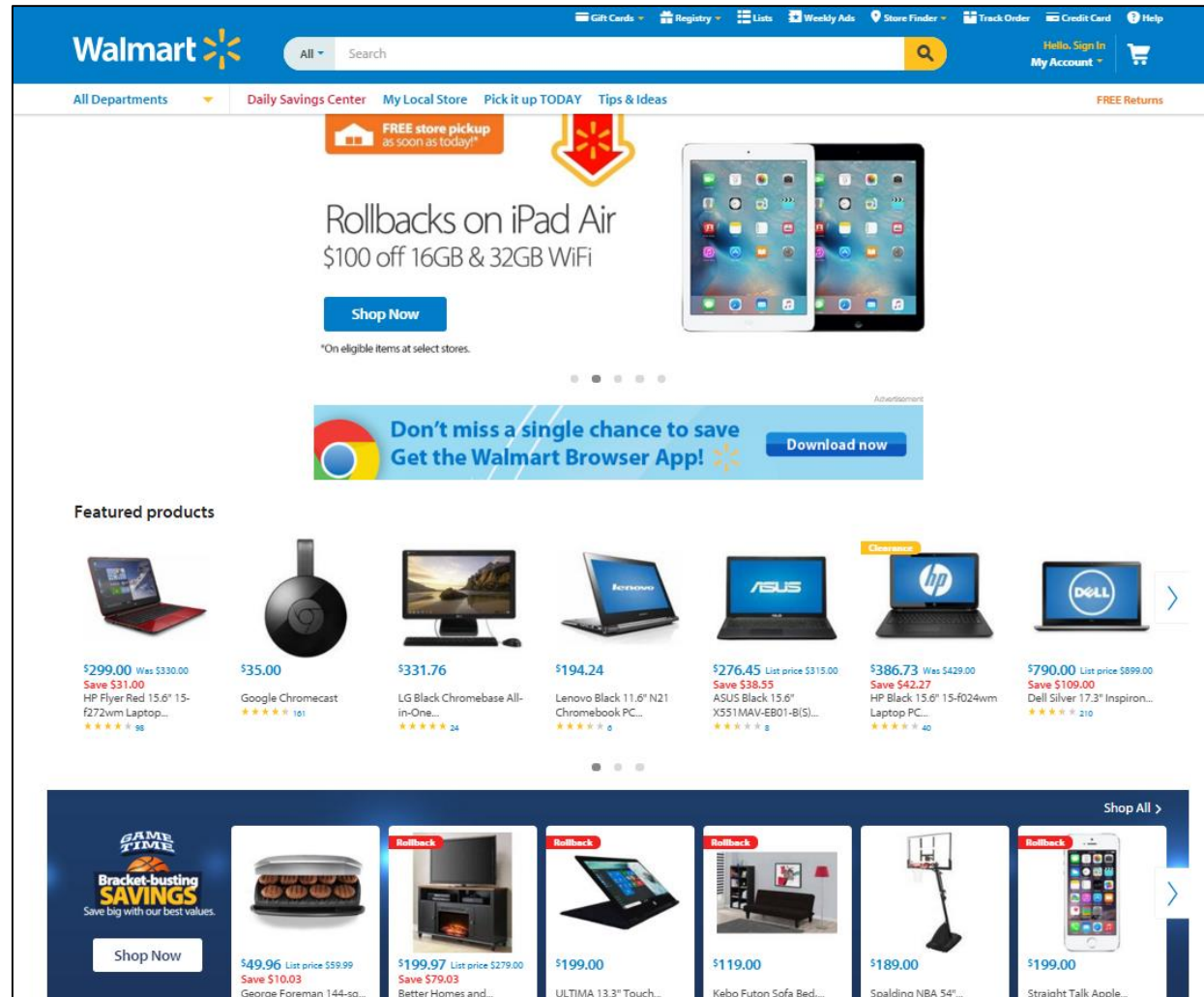
# Betriebsmonolithen



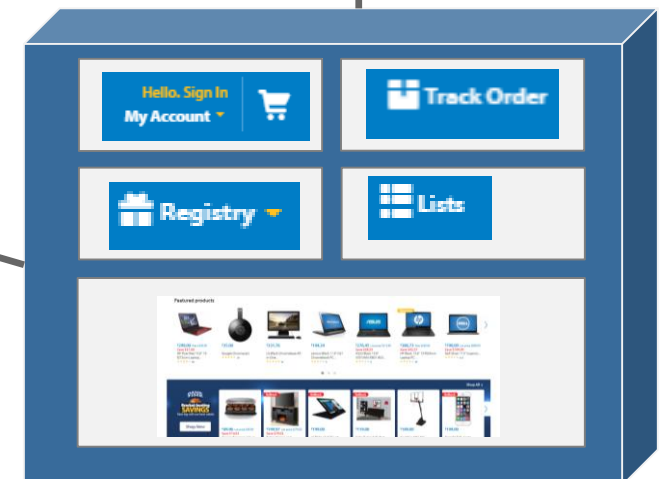
2016

Zusätzliche  
Dienste (z.T.  
personalisiert)

# Betriebskomponenten



Skalieren durch  
Verteilung und Klone





## RESULTS

Today the Walmart eCommerce platform is hosted in some of the largest OpenStack clouds and is managed exclusively via OneOps. On a typical day there are now over 1,000 deployments, executed on-demand by development teams, each taking only minutes on average to complete.

Walmart auf <http://www.oneops.com>

“They wanted to prepare for the world by 2020, with 4 billion people connected, 25+ million apps available, and 5.200 GB of data for each person on Earth. Walmart replatformed [...] with the intention of achieving close to 100% availability with reasonable costs.”

<https://blog.risingstack.com/how-enterprises-benefit-from-microservices-architectures>

- “In fact, the organization reports that some 3,000 engineers [...] drive 30,000 changes per month to Walmart software.”
- “Those new applications, which span everything from mobile devices to the Internet of things (IoT), are crucial weapons in a global e-commerce contest that pits Walmart against the likes of Amazon and Alibaba, as well as a host of other rivals that are emerging as the cost of entry into the online retail sector continues to decline in the age of the API economy.”

<http://www.baselinemag.com/enterprise-apps/walmart-embraces-microservices-to-get-more-agile.html>

- “The Walmart [...] servers [...] were able to handle all mobile Black Friday traffic with about 10 CPU cores and 28Gb RAM.”
- “On Thanksgiving weekend, Walmart servers processed 1.5 billion requests per day. 70 percent of which were delivered through mobile.”

<http://techcrunch.com/2014/12/02/walmart-com-reports-biggest-cyber-monday-in-history-mobile-traffic-at-70-over-the-holidays>

Change-to-Production  
im Tagesbereich

~100% Verfügbarkeit

Hohe Anzahl an  
Changes

Ressourcen-  
Sparsamkeit

Elastische  
Skalierbarkeit

# Softwarearchitektur = Komponenten + Schnittstellen

## Zweck:

- **Beherrschbarkeit & Parallelisierbarkeit:** Divide & Conquer
- **Isolation:** Problem- / Complexity-Hiding

## Entwurf



### De-Kompositionseinheiten

- Datenhoheit
- Hohe Kohäsion
- Lose Kopplung

## Programmierung

### Modul: Ausleihe

```
pom.xml  
de.qaware.bib.ausleihe.IAusleihe  
de.qaware.bib.ausleihe.impl.Ausleihe
```

- Entwicklungseinheiten
- Planungseinheiten

## Betrieb



?

- Release-Einheiten
- Deployment-Einheiten
- Laufzeit-Einheiten  
(Risikogemeinschaften,  
Einheiten der Isolation)
- Skalierungs-Einheiten
- Transport-Einheiten

# Die passende Granularitätsebene von Betriebskomponenten muss klug gewählt werden.

■ Der Tradeoff:



- Flexible Skalierungsmöglichkeiten
- Isolation zur Laufzeit
- Komponentenorientierung bis in den Betrieb
- Unabhängige Release und Deployments möglich



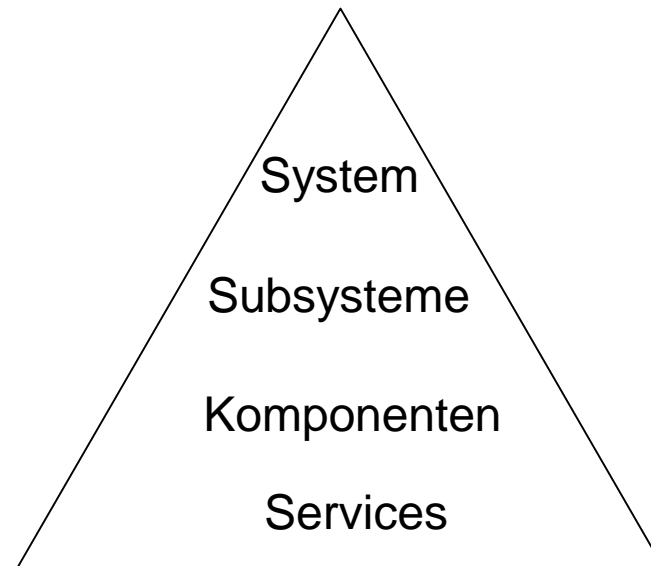
- Komplexere Betriebsumgebung
- „Verteilungs-Schulden“
- Schwierigere Diagnostizierbarkeit
- Komplexere Skalierbarkeit
- Komplexere Integration

Monolith

Macroservices

Microservices

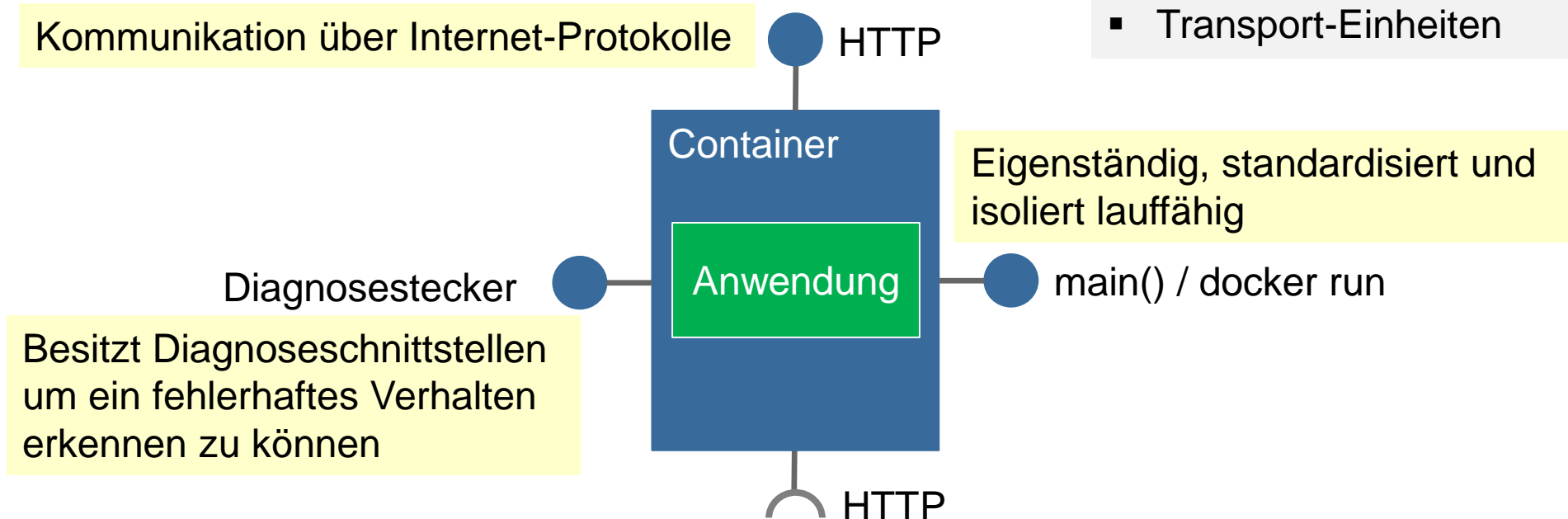
Nanoservices



# Die Anatomie von Betriebskomponenten

Die Betriebskomponenten sind:

- Release-Einheiten
- Deployment-Einheiten
- Laufzeit-Einheiten
- Skalierungs-Einheiten
- Transport-Einheiten



# Regel 1 für den Betrieb in der Cloud.

*“Everything fails all the time.”*

— Werner Vogels, CTO of Amazon





## Regel 2 für den Betrieb in der Cloud.

Soll nur der Himmel die Grenze sein, dann funktioniert nur horizontale Skalierung.



## Regel 3 für den Betrieb in der Cloud.

Wer in die Cloud will, der sollte Cloud sprechen.

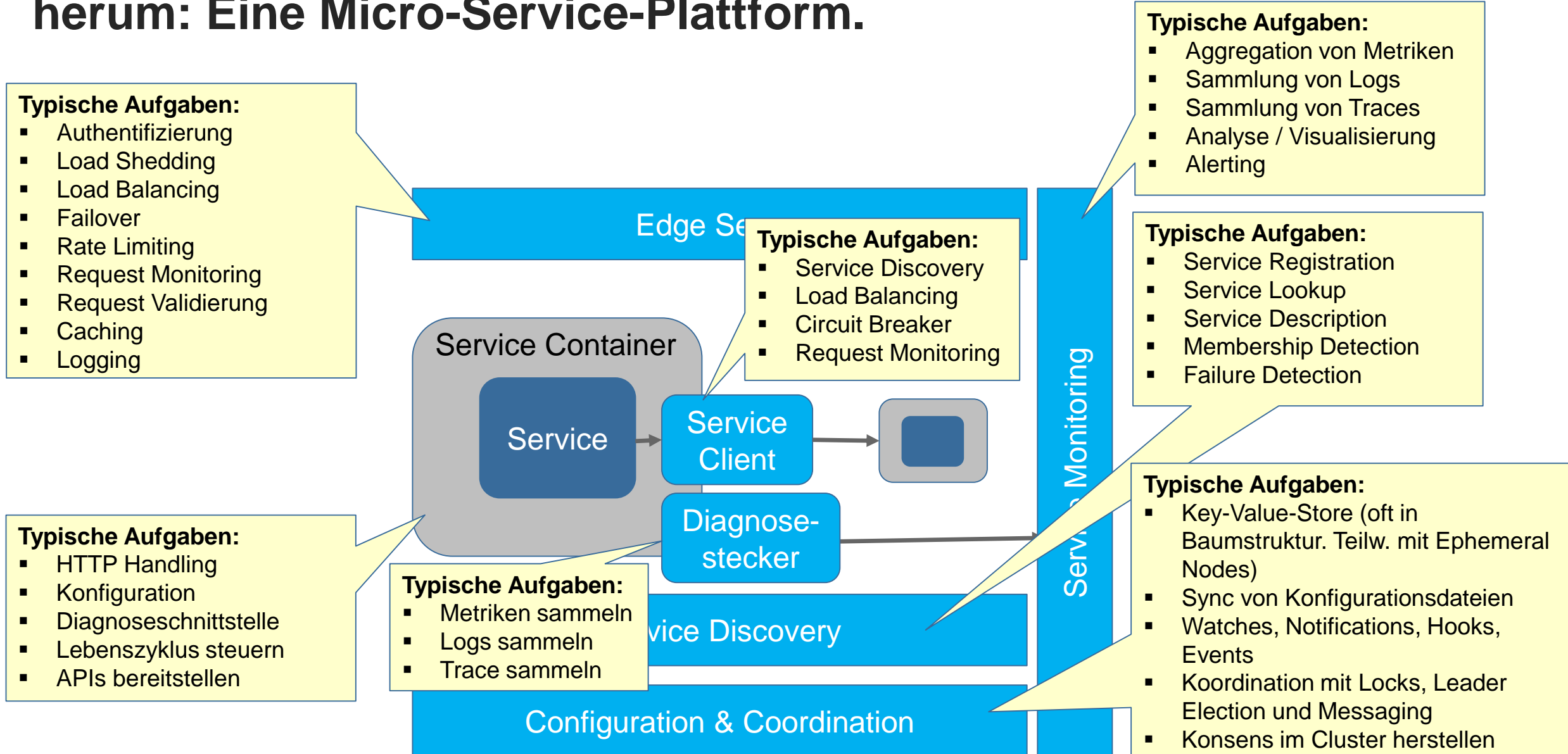
TCP

HTTP

DHCP

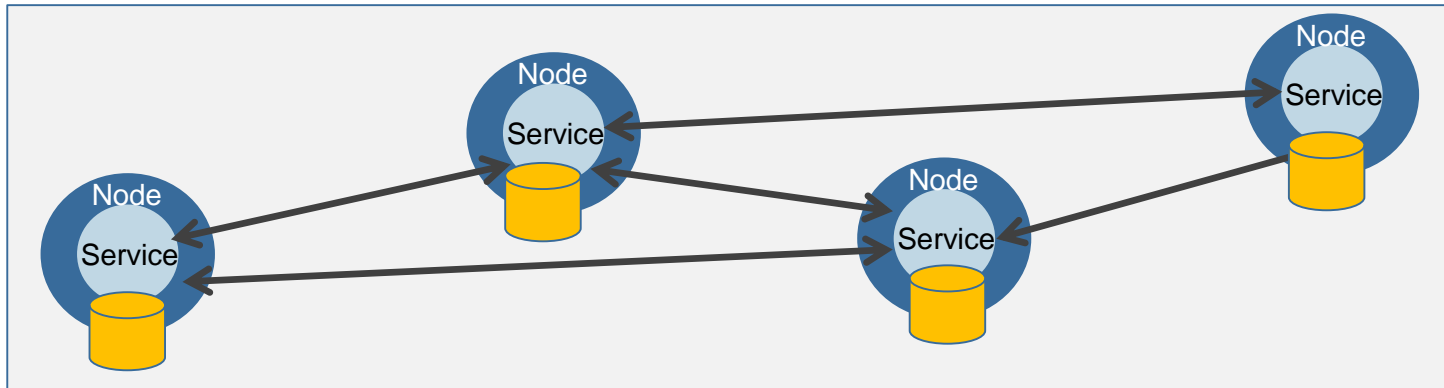
DNS

# Betriebskomponenten benötigen eine Infrastruktur um sie herum: Eine Micro-Service-Plattform.



# Configuration & Coordination: Verteilter Zustand & Konsens

# Ein verteilter Konfigurationsspeicher.



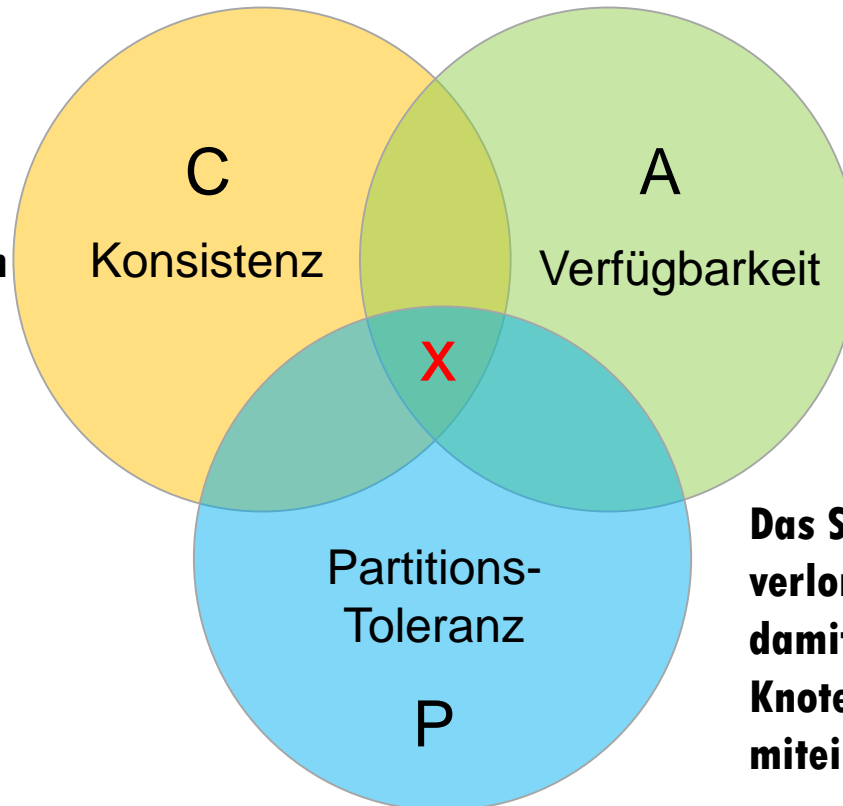
Wie wird der Zustand des Konfigurationsspeichers im Cluster synchronisiert?



# Das CAP Theorem

- Theorem von Brewer für Eigenschaften von zustandsbehafteten verteilten Systemen – mittlerweile auch formal bewiesen.  
Brewer, Eric A. "Towards robust distributed systems." *PODC*. 2000.
- Es gibt drei wesentliche Eigenschaften, von denen ein verteiltes System nur zwei gleichzeitig haben kann:

**Alle Knoten sehen die selben Daten zur selben Zeit. Alle Kopien sind stets gleich.**

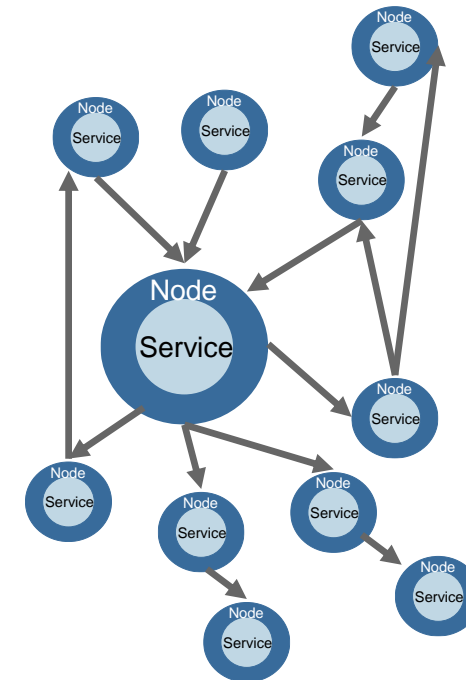


**Das System läuft auch, wenn einzelne Knoten ausfallen. Ausfälle von Knoten und Kanälen halten die überlebenden Knoten nicht von ihrer Funktion ab.**

**Das System funktioniert auch im Fall von verlorenen Nachrichten. Das System kann dabei damit umgehen, dass sich das Netzwerk an Knoten in mehrere Partitionen aufteilt, die nicht miteinander kommunizieren.**

# Gossip Protokolle: Inspiriert von der Verbreitung von Tratsch in sozialen Netzwerken.

- Grundlage: Ein Netzwerk an Agenten mit eigenem Zustand
- Agenten verteilen einen Gossip-Strom
  - Nachricht: Quelle, Inhalt / Zustand, Zeitstempel
  - Nachrichten werden in einem festen Takt periodisch versendet an eine bestimmte Anzahl anderer Knoten (Fanout)
- Virale Verbreitung des Gossip-Stroms
  - Knoten, die mit mir in einer Gruppe sind, bekommen auf jeden Fall eine Nachricht
  - Die Top x% an Knoten, die mir Nachrichten schicken bekommen eine Nachricht
- Nachrichten, denen vertraut wird, werden in den lokalen Zustand übernommen
  - Die gleiche Nachricht wurde von mehreren Seiten gehört
  - Die Nachricht stammt von Knoten, denen der Agent vertraut
  - Es ist keine aktuellere Nachricht mit gleichem Inhalt vorhanden



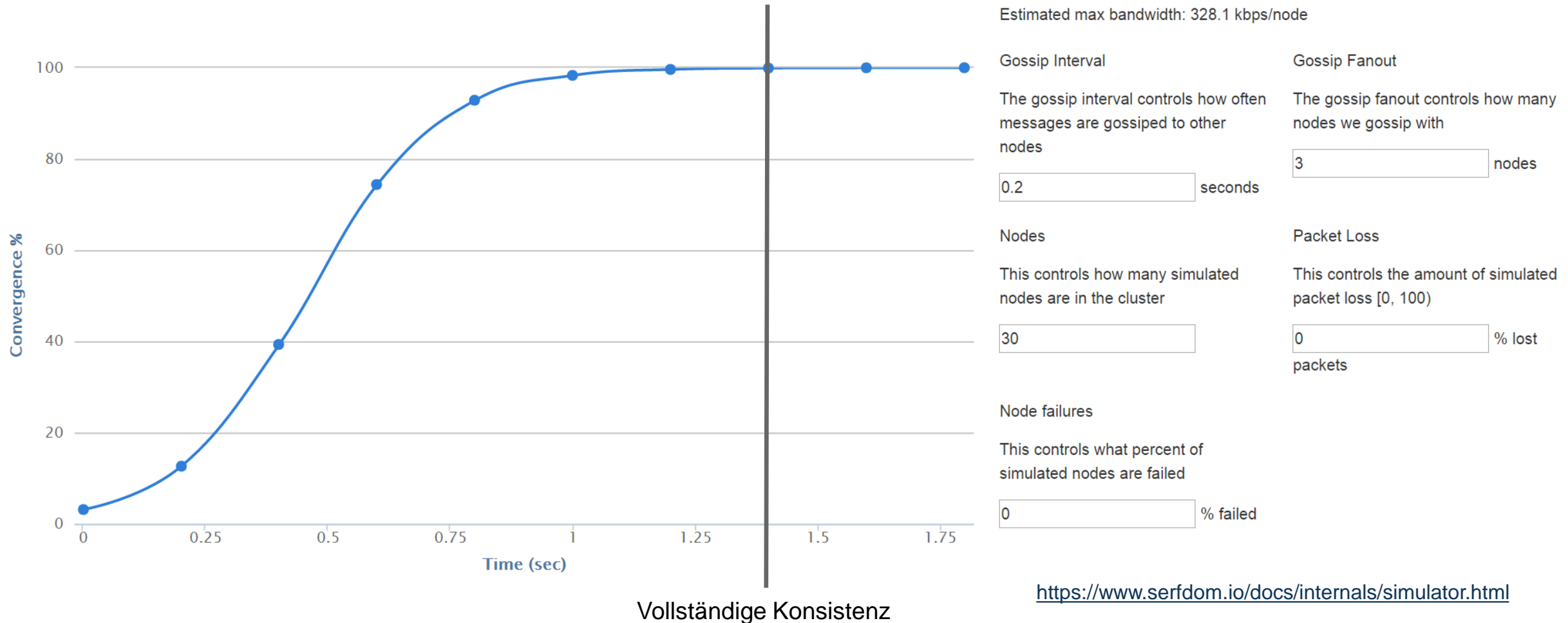
## Vorteile:

- Keine zentralen Einheiten notwendig.
- Fehlerhafte Partitionen im Netzwerk werden umschifft. Die Kommunikation muss nicht verlässlich sein.

## Nachteile:

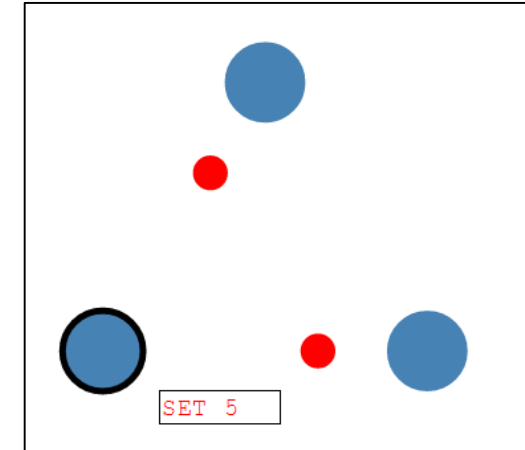
- Der Zustand ist potenziell inkonsistent verteilt (konvergiert aber mit der Zeit)
- Overhead durch redundante Nachrichten.

# Die Konvergenz der Daten und damit der Zeitpunkt der vollständigen Konsistenz ist berechenbar.



# Protokolle für verteilten Konsens sind im Gegensatz zu Gossip-Protokollen konsistent aber nicht hoch-verfügbar.

- Grundlage: Netzwerk an Agenten
- Prinzip: Es reicht, wenn der Zustand auf einer einfachen Mehrheit der Knoten konsistent ist und die restlichen Knoten ihre Inkonsistenz erkennen.
- Verfahren:
  - Das Netzwerk einigt sich per einfacher Mehrheit auf einen Leader-Agenten – initial und falls der Leader-Agent nicht erreichbar ist.
  - Alle Änderungen laufen über den Leader-Agenten. Dieser verteilt per Multicast Änderungsnachrichten periodisch im festen Takt an alle weiteren Agenten.
  - Quittiert die einfache Mehrheit an Agenten die Änderungsnachricht, so wird die Änderung im Leader und (per Nachricht) auch in den Agenten aktiv, die quittiert haben. Ansonsten wird der Zustand als inkonsistent angenommen.
- Konkrete Konsens-Protokolle: Raft, Paxos



## Vorteile:

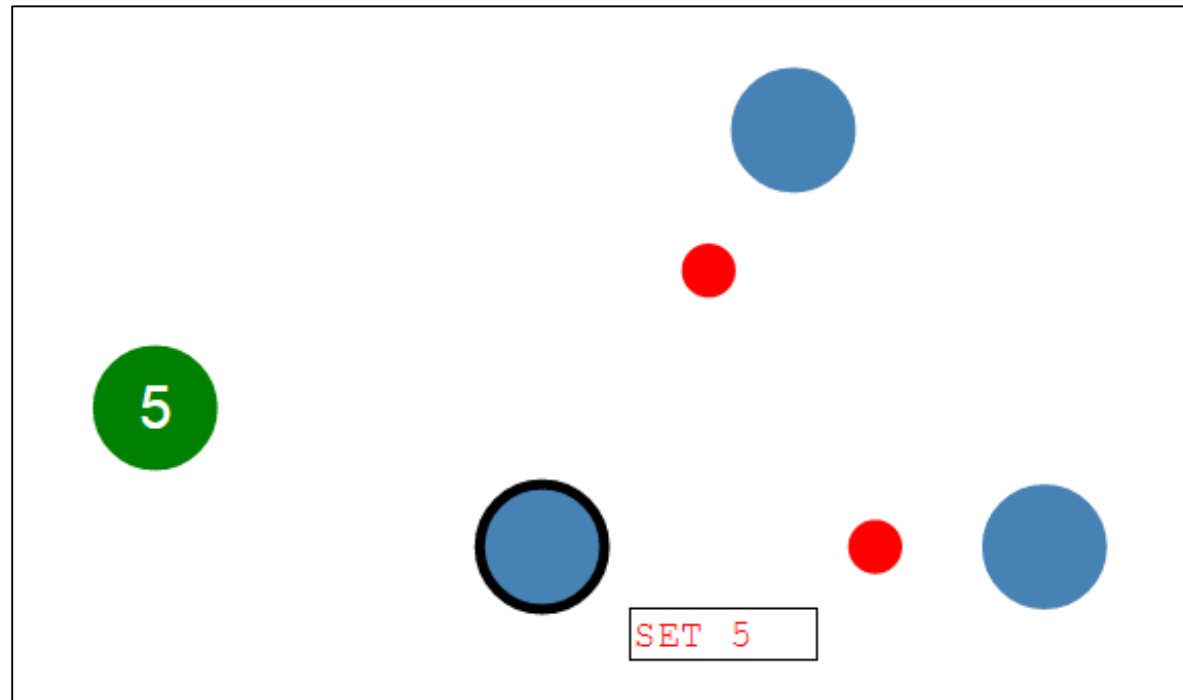
- Fehlerhafte Partitionen im Netzwerk werden toleriert und nach Behebung des Fehlers wieder automatisch konsistent.
- Streng konsistente Daten.

## Nachteile:

- Der zentrale Leader-Agent limitiert den Durchsatz an Änderungen.
- Nicht hoch-verfügbar: Bei einer Netzwerk-Partition kann die kleinere Partition nicht weiterarbeiten.

# Das Raft Konsens-Protokoll

- Ongaro, Diego; Ousterhout, John (2013). "In Search of an Understandable Consensus Algorithm".



<http://thesecretlivesofdata.com/raft>



# Paxos als Alternative zu Raft

- Erstmalig 1989 beschrieben von Leslie Lamport zur Replikation von Zustandsautomaten in verteilten Systemen auf Basis (s)einer logischen Uhr.
- Unterschied zu Raft: Komplexer.  
Aber funktional gleichwertig zu Raft.
- Quellen:
  - <http://the-paper-trail.org/blog/consensus-protocols-paxos/>
  - <https://www.youtube.com/watch?v=JEpsBg0AO6o>
  - [http://www.uio.no/studier/emner/matnat/ifi/INF5040/h08/group/The Paxos Algorithm.pdf](http://www.uio.no/studier/emner/matnat/ifi/INF5040/h08/group/The_Paxos_Algorithm.pdf)
  - <http://research.microsoft.com/en-us/um/people/lamport/pubs/paxos-simple.pdf>

## Implementing Replicated Logs with Paxos

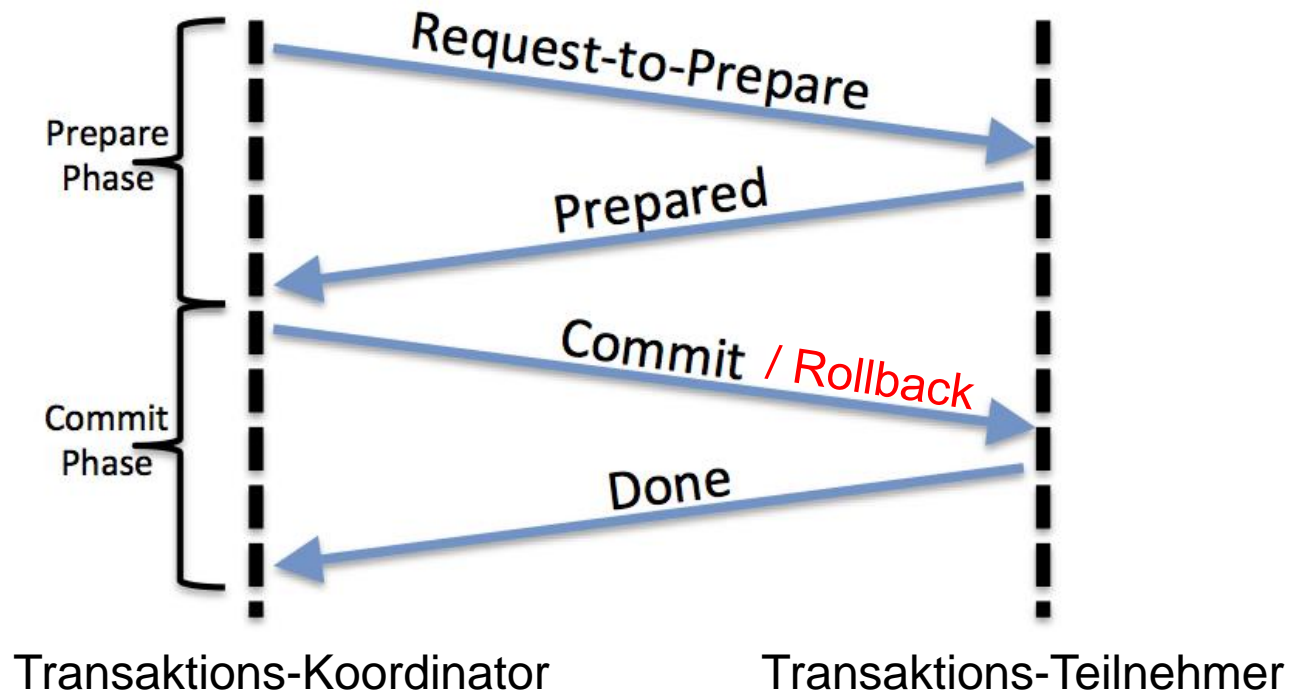
John Ousterhout and Diego Ongaro  
Stanford University



Note: this material borrows heavily from slides by Lorenzo Alvisi, Ali Ghodsi, and David Mazières

# Ist strenge Konsistenz über alle Knoten notwendig, so verbleibt das 2-Phase-Commit Protokoll (2PC)

- Ein Transaktionskoordinator verteilt die Änderungen und aktiviert diese erst bei Zustimmung aller. Ansonsten werden die Änderungen rückgängig gemacht.



## Vorteil:

- Alle Knoten sind konsistent zueinander.

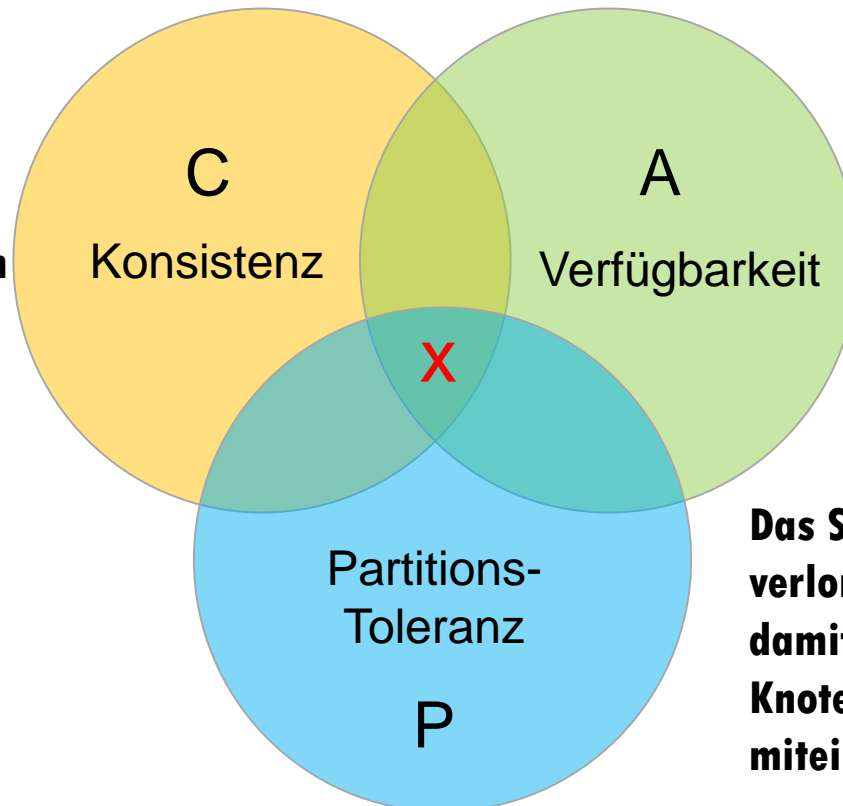
## Nachteile:

- Zeitintensiv, da stets alle Knoten zustimmen müssen.
- Das System funktioniert nicht mehr, sobald das Netzwerk partitioniert ist.

# Das CAP Theorem

- Theorem von Brewer für Eigenschaften von zustandsbehafteten verteilten Systemen – mittlerweile auch formal bewiesen.  
Brewer, Eric A. "Towards robust distributed systems." *PODC*. 2000.
- Es gibt drei wesentliche Eigenschaften, von denen ein verteiltes System nur zwei gleichzeitig haben kann:

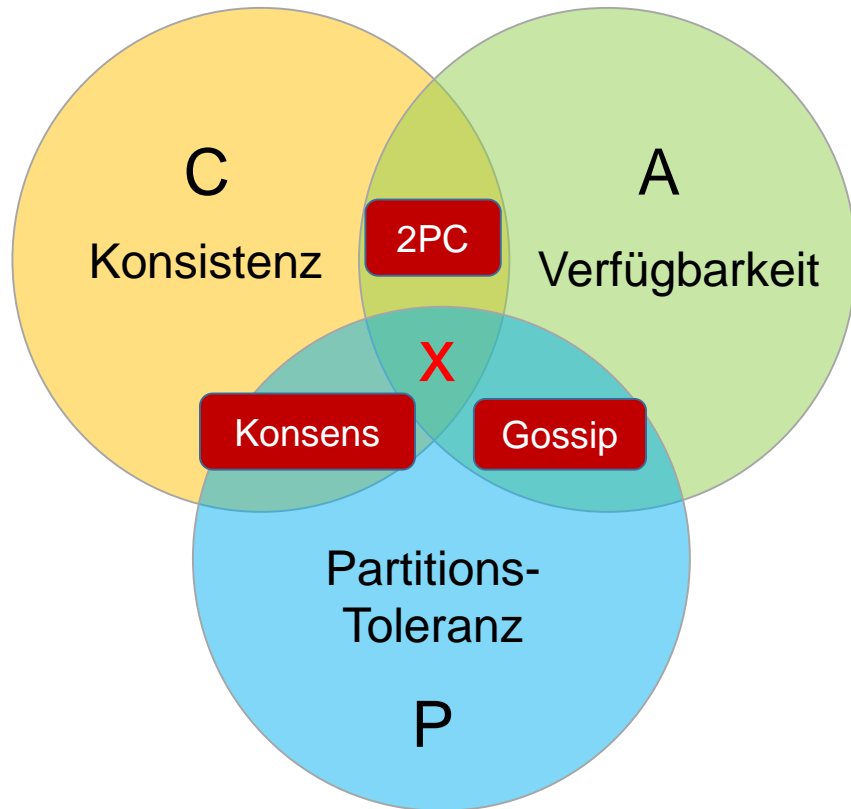
**Alle Knoten sehen die selben Daten zur selben Zeit. Alle Kopien sind stets gleich.**



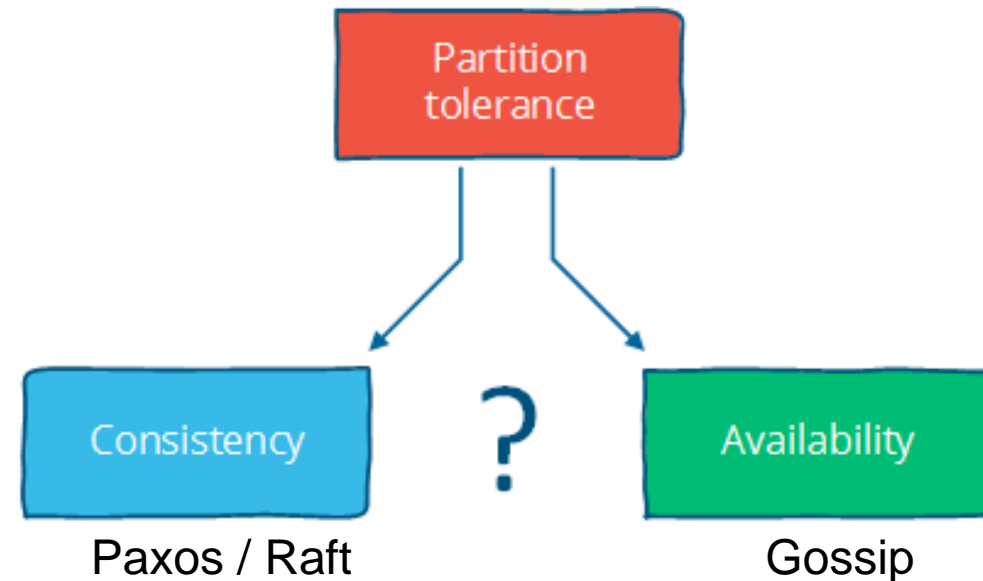
**Das System läuft auch, wenn einzelne Knoten ausfallen. Ausfälle von Knoten und Kanälen halten die überlebenden Knoten nicht von ihrer Funktion ab.**

**Das System funktioniert auch im Fall von verlorenen Nachrichten. Das System kann dabei damit umgehen, dass sich das Netzwerk an Knoten in mehrere Partitionen aufteilt, die nicht miteinander kommunizieren.**

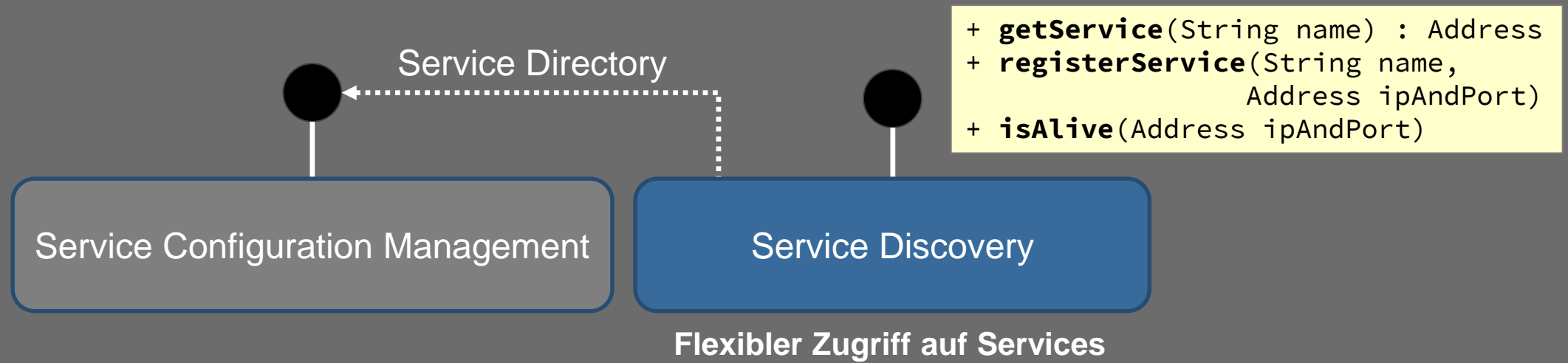
# Die vorgestellten Protokolle und das CAP Theorem.



In der Cloud müssen Partitionen angenommen werden. Damit ist die Entscheidung binär zwischen Konsistenz und Verfügbarkeit.

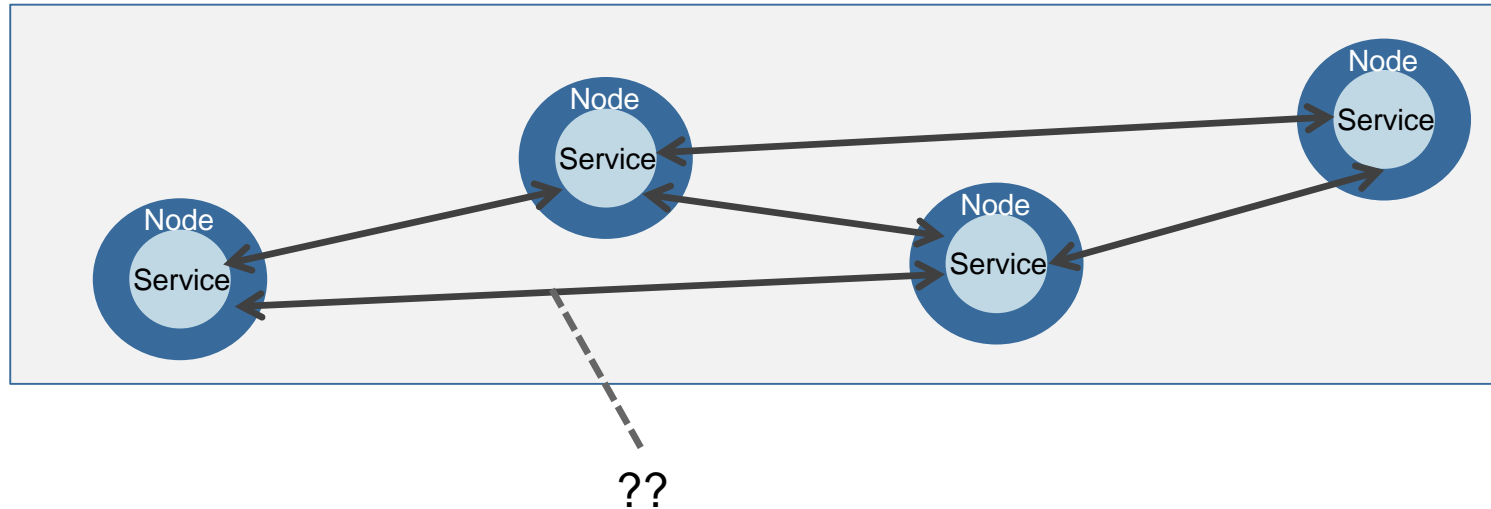


# Service Discovery





# Die Probleme einer klassischen Verknüpfung von Services in der Cloud.



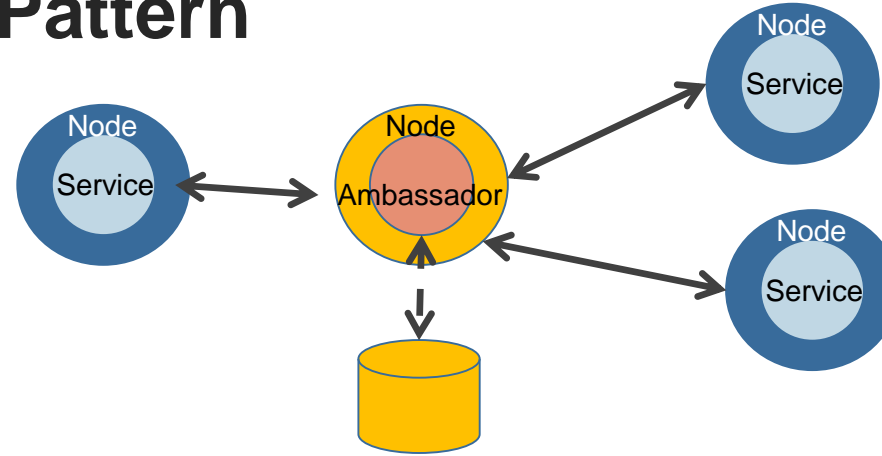
## Probleme:

- Mangelnde Redundanz: Jeder Service wird direkt genutzt. Er kann nicht unmittelbar in mehreren Instanzen laufen, die Redundanz schaffen.
- Mangelnde Flexibilität: Die Services können nicht ohne Seiteneffekt neu gestartet oder auf einem anderen Knoten in Betrieb genommen werden – oder sogar durch eine andere Service-Implementierung ausgetauscht werden.

## Lösungen:

- Dynamischer DNS
- Ambassador
- Dynamischer Konfigurationsdateien und Umgebungsvariablen

# Das Ambassador Pattern



Ein Ambassador-Knoten für jede Knoten-Art (z.B. NGINX-Webserver)

- **Service Registration:**
  - Beobachtet das Cluster und erkennt neue und kranke/tote Knoten in seiner Gruppe.
  - Hinterlegt die aktuell aktiven Knoten im Konfigurationsspeicher.
- **Service Discovery:** Der Client kommuniziert mit dem Ambassador-Knoten, der die Anfragen aber möglichst effizient an einen Knoten der Gruppe weiterreicht.

Der Ambassador-Knoten kann dabei eine Reihe an Zusatzdienste erweisen bei der Verbindung zum Service (**Service Binding**):

- Load Balancing inklusive Failover
- Service Monitoring
- Circuit Breaker Pattern
- Throttling

# Beispiel: Consul



<https://www.consul.io>

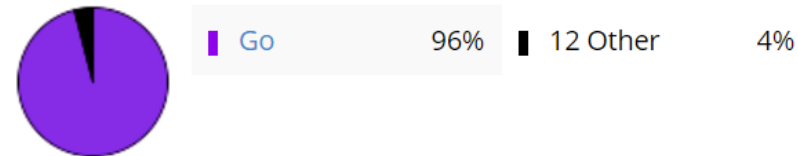
# Consul

- Infrastruktur für Configuration & Coordination (per Raft-Protokoll) und Service Discovery (per DNS oder HTTP-Lookups) inkl. Health Checking der Services.
- Open Source Projekt der Firma HashiCorp (<https://www.consul.io>), von dem auch eine kommerzielle Variante verfügbar ist.

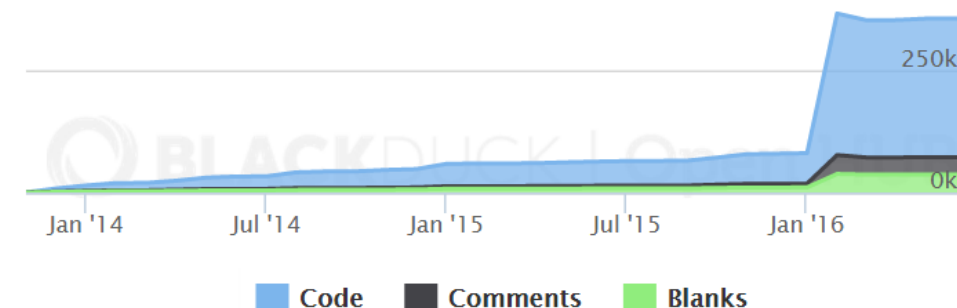
In a Nutshell, consul...

- ... has had 3,951 commits made by 238 contributors representing 285,317 lines of code
- ... is mostly written in Go with an average number of source code comments
- ... has a young, but established codebase maintained by a very large development team with stable Y-O-Y commits
- ... took an estimated 76 years of effort (COCOMO model) starting with its first commit in November, 2013 ending with its most recent commit about 1 month ago

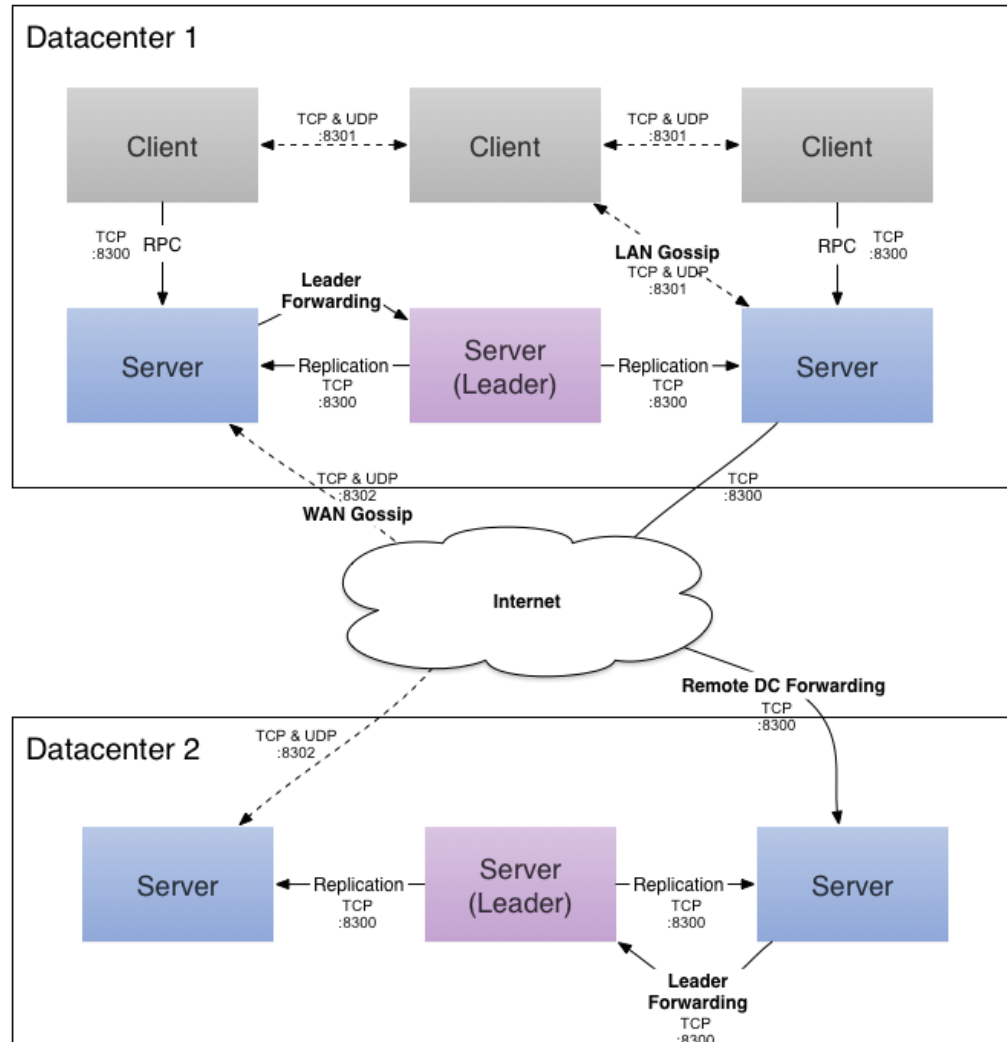
Languages



Lines of Code



# Consul Architektur



siehe <https://www.consul.io/docs/internals/architecture.html>

- Consul besteht aus einer einzigen Binary (*consul*), dem *Consul Agent*.
- Der Consul Agent kann sowohl als *Server* als auch als *Client* gestartet werden.
- Alle Agenten finden sich gegenseitig über ein Gossip-Protokoll und identifizieren so auch Fehler.
- Die Server synchronisieren ihren Zustand über ein Konsens-Protokoll (Raft). Empfohlen werden 3-5 Consul-Server pro Datacenter auf dedizierten Maschinen.
- Die Clients sind i.W. zustandslos und laufen auf allen Maschinen, die mit Consul interagieren. Die Kommunikation mit Consul läuft stets über den lokalen Agenten, der dann weiter vermittelt.
- Consul Server können sich über mehrere Datacenter hinweg synchronisieren.